This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE Globecom 2010 proceedings.

Scalable Publish/Subscribe Service in Wireless Mesh Networks

Juan Li, Weiyi Zhang, and Xiaojun Xia
Department of Computer Science
North Dakota State University, Fargo, USA
{j.li, weiyi.zhang, xiaojun.xia}@ndsu.edu

***Abstract*—Wireless Mesh Networks represent a promising technology to provide wireless Internet connectivity over a large community. This new technology not only allows a fast, easy and inexpensive network deployment, but also enables many new applications including community-scale peer-based communication or sharing of network resources and services. In this work, we propose an effective publish/subscribe communication paradigm to facilitate efficient data dissemination over wireless mesh networks. The proposed publish/subscribe scheme dynamically routes and delivers events and services from sources to interested users with minimum communication overhead and memory storage. The effectiveness of the system is demonstrated through comprehensive simulation studies.**

Keywords-publish/subscribe system; routing; wireless mesh network

I. INTRODUCTION

Wireless mesh networks (WMNs) ([2, 16, 17]) have emerged as one of the key technologies for next-generation wireless networking. Because of their potential for extending the coverage of public Internet access and enabling savings on cabling, deployment and maintenance costs, WMNs are undergoing rapid progress and inspiring numerous applications. Besides connecting users to the Internet, WMNs are increasingly used for connecting users in their communities to share resources and information. In such a wireless mesh community, every user in this community neighborhood can contribute resources and cooperate. The mesh community then allows faster and easier dissemination of information that is relevant to the local community. Content exchange and data dissemination in these networks, therefore, become increasingly important. The event-driven or notification-based interaction pattern provided by publish/subscribe (pub/sub) communication paradigm, because of its decoupling and asynchrony properties, is inherently suited for these wireless mesh community networks.

A pub/sub system connects information publishers with subscribers by delivering published events from publishers to interested subscribers. Subscribers express their interested contents using a set of subscriptions. Publishers publish new information into the system using a set of publications. Upon receiving a publication, the system searches for matching subscriptions and notifies the interested subscribers. This pub/sub model decouples time, space, and flow between publishers and subscribers, allowing for greater scalability and a more dynamic network topology as required by WMNs. Although pub/sub communication paradigm brings many benefits to distributed applications and has been widely used in distributed systems, it is challenging to apply pub/sub to WMNs. A WMN is limited by its network capacity due to the requirement that nodes forward each other's packets. Therefore, the pub/sub systems should be designed in a way that minimizes the number of messages propagated by the mesh routers. Moreover, mesh networks may involve either fixed or mobile clients. In the presence of mobility of the mesh clients, it may be expensive in term of message overhead for keeping track of the location of mobile publishers and subscribers.

Because of the special properties of WMNs, previous pub/sub solutions proposed for wired networks [3, 4, 18, 19] are not suitable for WMNs. For example, P2P-based pub/sub cannot be directly applied to WMNs, since these approaches are optimal in terms of overlay messages and they do not consider the underlying mesh (router) topology. Pub/sub approaches for mobile ad hoc networks (MANETs) [1, 11, 13, 20], on the other hand, assume all nodes are highly dynamic and do not take advantage of the relative static property of mesh routers. Therefore, pub/sub schemes for MANETs do not fit for WMNs either.

In this paper we propose an efficient service/event delivery protocol that adapts pub/sub communication paradigm to WMNs. By applying a hash-based mapping, we dynamically determine brokers that perform a store and forward function to route messages from publishers to subscribers. Notifications are delivered to subscribers once the published events/services match the subscriptions. Our protocol guarantees that notification messages are only transmitted to interested subscribers without propagating to unrelated nodes. In order to exploit all the advantages that the wireless mesh medium offers, we propose a new routing metric for our pub/sub communication. Instead of using a virtual overlay as existing pub/sub systems did, our routing metric comes from a cross-layer approach in order to make the pub/sub layer aware of the local issues of the underling routing layer.

The rest of this paper is organized as follows: Section II introduces the related work. Section III presents the proposed pub/sub protocol and the notification multicast algorithm. Section IV discusses the experimental results. Finally, conclusions are drawn in Section V.

978-1-4244-5637-6/10/$26.00 ©2010 IEEE

II. Related Work

Most of the previous works on pub/sub [5, 9, 10, 15] focus on increasing efficiency over traditional wired networks. These pub/sub systems can be implemented centrally or in a distributed manner. Centralized schemes [14, 12, 4] have a global view of the system, which enables efficient optimizations during the matching process. However, the central server can be a potential bottleneck and the system may be lack of scalability and fault tolerance. Distributed pub/sub systems [8, 19] have been introduced to address these problems. For example, P2P-based solutions [18, 3] implement pub/sub in a fully decentralized manner. Most P2P pub/sub approaches construct an overlay to transmit messages. They do not consider the underlying topology, for instance by mesh routers.

Many studies have been performed to apply the pub/sub paradigm to MANETs. They can be categorized into two schemes by routing methods: destination-based routing scheme [1, 11] and content-based routing scheme [20, 13]. Both schemes have to build dissemination trees to deliver publications to subscribers who are interested in the publications. The pub/sub schemes proposed for MANETs are not optimal for WMNs, because unlike the highly mobile MANET nodes mesh routers have minimal mobility and form the mesh backbone for mesh clients. Therefore, pub/sub in WMNs should take advantage of this property and make the communication more efficient.

To the best of our knowledge, there are only a few works on constructing pub/sub system over WMNs [21, 7]. In [21], a subscriber subscribes to a broker (mesh router). A broker broadcasts the subscription to all other brokers in the network. The matched event is propagated back using reverse path. This approach is very expensive in terms of the subscription messages, because subscriptions have to be broadcasted to all the routers in the network. SPINE [7] implements the pub/sub scheme by establishing vertical and horizontal paths within a two-dimensional space. Subscriptions are propagated along a horizontal path, while publications are forwarded along a vertical path. The matching happens at the intersections. Compared with the broadcasting-based approach [21], SPINE [7] reduces the overhead of pub/sub by only propagating publications and subscriptions to one row/column. Compared with SPINE, our approach further reduces the matching overhead and more importantly, dramatically reduces the storage overhead for storing subscriptions.

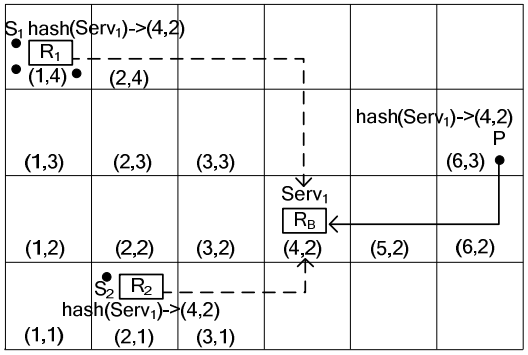

Figure 1. An example of hash-based publishing and subscribing routing

III. Scalable Pub/Sub Protocol

In our system, we assume that both the publishers and the subscribers are from client nodes. Client nodes can provide services to other client nodes and at the same time, they may consume services from other client nodes. Mesh routers act as brokers to assist matching publications with subscriptions.

A. Efficient subscribing and publishing

We assume the mesh network is deployed in a two-dimensional space that can be partitioned into equal sized zones. Each zone is assigned with a coordinate pair containing a latitude value and a longitude value, as shown in Fig. 1. Each zone contains at least one mesh router and the clients in a zone can directly communicate with the router located in the same zone. The router is called the home router of the clients in the same zone. In addition, a router can directly communicate with another router in the neighboring zones sharing one side. We assume each router knows its approximate location in the network. Routers are also brokers that store subscriptions and match subscriptions with publications.

1) Subscribing. To subscribe an interested service (or event), the subscriber maps the interested service onto a point with coordinate pair $<x_1, y_1>$ in the two-dimensional space using a uniform hash function. Then the subscription is routed from the home router (with coordinates $<x_0, y_0>$) to a broker router (with zone coordinates $<x_1, y_1>$). The routing is performed by greedily forwarding the subscription message horizontally and then vertically. It is obvious that the number of hops needed for the routing is the Manhattan distance ($|x_1-x_0| + |y_1-y_0|$) from the subscriber's home zone to the broker's zone. After the subscription reaches the broker zone, the router at that zone stores the subscription, which includes the subscriber's ID, the coordinate pair of its home zone, and the information about the interested event/service. For example, in Fig. 1, two subscribers S_1 and S_2 both subscribe their interest in service $Serv_1$, which is hashed to a coordinate pair <4, 2>. Then S_1 and S_2 will route their subscription from their respective home router R_1 and R_2 to the broker router R_B at zone <4, 2>.

2) Publishing. To publish a service, the publisher will apply the same deterministic hash function as the subscribers used to map the service to a zone coordinate pair. The publisher will use the same routing strategy as subscribers used to locate the broker that is in charge of the zone. For example, in Fig. 1, publisher P who publishes/updates a service $Serv_1$, will be able to locate the broker R_B at zone <4, 2> based on the hash value of $Serv_1$. Then broker R_B will notify the publication to interested subscribers, i.e., S_1 and S_2 in this case. As shown from the example, the broker router R_B works as a bridge for the publishers and subscribers.

3) Multi-attribute support. Because of the use of hashing to map services/events to brokers, the proposed strategy supports topic-based pub/sub best. Our system can also support multi-attribute content-based pub/sub with a little extra work. For a publication with multiple attributes, each attribute is mapped to a hash pair which corresponds to a broker. For example, a service S with three attributes $<S.attr_1, S.attr_2, S.attr_3>$ will be hashed three times by applying a

simple uniform hash function, SUH, to $S.attr_1$, $S.attr_2$, and $S.attr_3$ respectively.

For a multi-attribute subscription, we do not need to subscribe it to multiple brokers. Instead, the subscription is only subscribed to one broker based on one attribute. Any attribute of a multi-attribute subscription can be used as the subscription key. Because each service/event is published redundantly at multiple brokers, the subscription can be matched successfully at any one of the brokers.

4) Pub/sub at the network layer. The strategy of pub/sub using distributed brokers has been used in P2P systems [18, 3]. These existing P2P pub/sub systems construct a distributed hash table-based (DHT) overlay as a virtual broker overlay to support efficient communication. Our pub/sub scheme adopts the idea of distributed brokers as well. However, instead of constructing the virtual DHT-based rendezvous overlay, we utilize the topology of the mesh routers. Our pub/sub routing is performed on top of physical routers, i.e., on the network layer. Because of the approximate locality of the neighboring mesh routers, our routing scheme is much more efficient compared to application-level overlay routing.

B. Notification multicast tree

Multiple subscribers may subscribe the same service; notifying them one by one may waste lots of valuable bandwidth. Because the routes from the broker to the subscribers may share many common parts, a multicast tree structure can be more efficient for notification propagation. A multicast tree can be constructed for all client nodes that have subscribed to a certain service/event. The root of the multicast tree is the broker, and its branches are formed along the routes from the root to the subscriber nodes. The tree is formed on the fly during the process of propagating the notification.

C. Mobility and fault tolerance

In a mesh community, client nodes are normally mobile. Since both publishers and subscribers are client nodes, the system should support the pub/sub service seamlessly in spite of the clients' mobility. In our pub/sub system, brokers record the location of subscribers in order to deliver the notifications. When a subscriber moves to a different location, the original subscription is no longer valid because of the change of the location of the subscriber. Therefore, the system needs to update the subscription. A subscriber who moves from one zone to another connects to the mesh router in the new zone and re-subscribes with its new zone information. The broker(s) of this subscriber will update the subscriber's location information. Then future notification will be forwarded to the new location of the subscriber. An advantage of our system is that the broker is located according to the hash of the service/event, which is unrelated to the subscriber or publisher's location. Therefore, although subscribers move, their brokers are still the same.

When a subscriber loses its interest to a particular event/service, it can actively unsubscribe its original subscription from the corresponding broker. Alternatively, the subscriber does not need to unsubscribe. The broker will know the changing of interests when the notification fails. A broker may know that a subscriber silently leaves the network or loses interest in a service/event from the returned error message.

The movement of publishers would not affect the system. When publishing a new event/service, the publisher will dynamically route the event to related brokers and notify interested subscriber. The publishing and notification process is not related to the publisher's location. Therefore, the proposed pub/sub communication inherently supports the movement of publishers.

In order to improve the robustness of the system, subscriptions are replicated. Each broker replicates its local subscriptions to its neighboring routers. Therefore, when one broker goes down, the request will be sent to the backup brokers.

IV. EXPERIMENT

We conducted a set of simulations to evaluate our pub/sub scheme. In this section, we present the experimental setup and analyze the results obtained.

A. Setup

A simulated WMN environment was considered. The underlying topology was composed of sets of fixed mesh routers and mobile mesh clients. In the evaluation we considered a typically evaluated grid topology, in which routers were placed within an area, so that each grid contains at least one router. The size of the area and the number of routers were adjusted throughout the simulations. Mobile clients were randomly distributed in the network. The ratio of the number of routers and clients was 1:5, i.e., on average, each router was connected with 5 clients. The propagation model used was Two-Ray Ground Reflection, the router's transmission range was 100m and the data rate was 11MB. The mobility of the mesh clients was similar to that of the "random waypoint" model as reported in [6]. In the random waypoint model, initially, nodes are randomly distributed within the 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. In the simulation, subscriptions and publications are represented as randomly-generated strings.

We also simulated two other protocols [21, 7] for performance comparison. As mentioned in Section II, [21] uses a subscription broadcasting approach (SB), and SPINE [7] uses a row/column-based pub/sub propagation approach. These three approaches (our hash-based approach (Hash), the SPINE approach, and the subscription broadcasting approach (SB)) 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) communication overhead including subscription, publication, and notification overhead, (b) memory storage overhead needed for storing subscriptions, (c) computing overhead for matching subscription with publications. All of the experiments performed are based on topic-based pub/sub unless otherwise noted.

978-1-4244-5637-6/10/$26.00 ©2010 IEEE

B. *Result and discussion*

Fig. 2 compares the total communication overhead of the three pub/sub approaches in networks with varied number of nodes (routers). The communication overhead is represented in terms of the number of messages propagated in the network. In this experiment, there are 1000 randomly distributed subscriptions and 1000 randomly distributed notifications. We increase the area from 10×10 zones to 60×60 zones. The communication overhead includes the subscribing, the publishing and the notification overhead. As expected, the SB approach generates the highest message overhead, while our Hash approach creates the lowest overhead among these three approaches. SB approach broadcasted the subscriptions to the whole network; therefore the message overhead is high. Our Hash approach could directly locate related brokers, thus saves lots of traffic.

Because SB approach always generates much more message overhead compared with Hash and SPINE, we only compare Hash and SPINE in the following experiments. Fig. 3 presents the subscription overhead of our Hash approach with SPINE approach. Fig. 4 compares the notification overhead of these two approaches. In these two experiments, the number of zones is 50×50, and the number of subscriptions and the number of notifications are 1000. Compared with SPINE, our Hash approach generates fewer subscription messages, while the notification messages are similar.

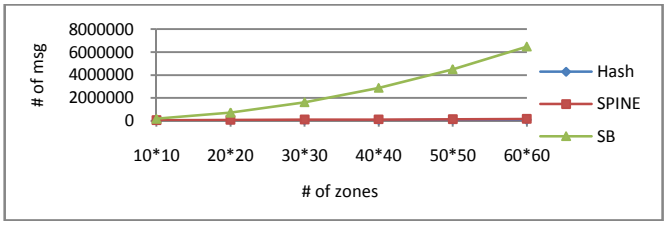

Figure 2. Total message overhead vs. the number of zones.

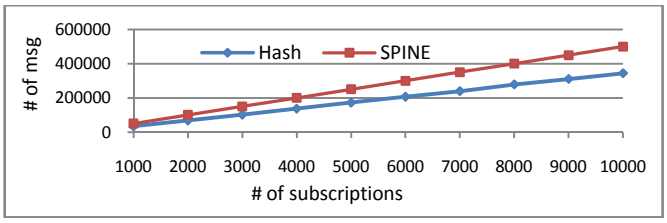

Figure 3. Subscription overhead vs. the number of subscriptions.

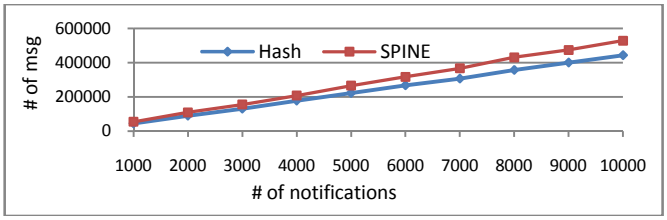

Figure 4. Notification overhead vs. the number of notifications.

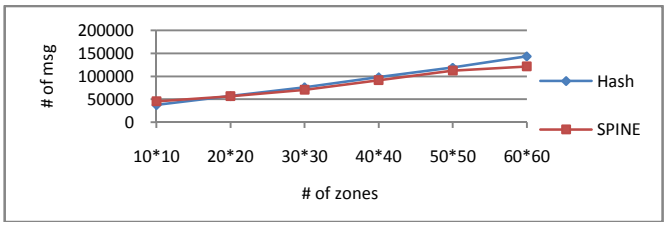

Figure 5. Total message overhead of the multi-attribute content-based pub/sub vs. the number of zones.

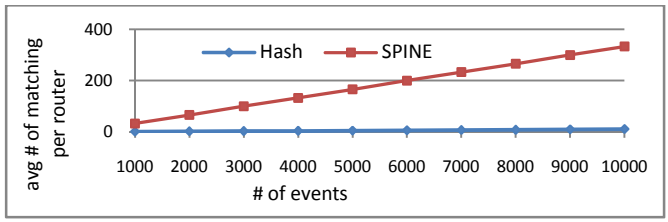

Figure 6. Matching overhead vs. the number of events/services published.

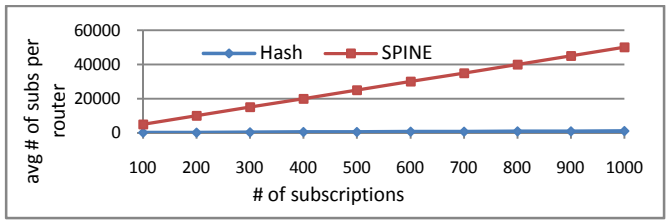

Figure 7. Memory overhead vs. the number of subscriptions.

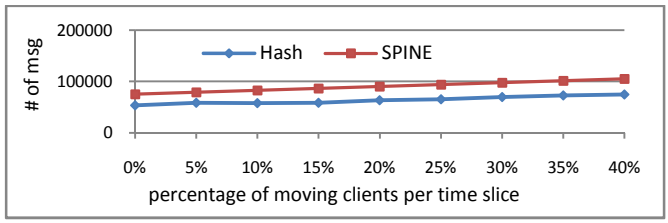

Figure 8. Subscription/re-subscription overhead vs. degree of client mobility.

As explained in Section III.A, our Hash approach performs best for topic-based pub/sub. All the experiments performed above are based on topic-based pub/sub. In this experiment, we evaluate the performance of multi-attribute content-based pub/sub. With the same network configuration as the first experiment (Fig. 2), we compare our Hash approach with SPINE in terms of the total message overhead for content-based multi-attribute pub/sub. In the experiment, the average number of attributes for each content-based subscription and publication is four. As can be seen from Fig. 5, our Hash approach performs very well: the message overhead is similar to that of SPINE's. For subscriptions, multi-attribute does not affect the system performance. The publication overhead increases, because the system has to publish to multiple brokers for each attribute. The notification overhead is not affected by multi-attribute. Note that multi-attribute will not bring any extra memory storage overhead to our scheme, because we do not subscribe to multiple brokers; although we publish to multiple brokers, publications are not stored at brokers.

Fig. 6 presents the matching overhead of the two approaches with respect to the number of events/services published. As can be seen, our Hash approach saves the matching overhead dramatically, because this approach can locate the right brokers directly and only do matching at the

right brokers. SPINE, on the other hand, has to match on all the nodes along the whole vertical line of the WMN area.

Fig. 7 illustrates the average memory overhead of storing subscriptions at each router. We increased the number of subscriptions in this experiment. It can be seen that our Hash approach uses much less memory compared with SPINE. Our approach stores each subscription only at the corresponding broker and a few neighboring backup routers for fault tolerance reasons. SPINE stores each subscription at all routers in the same row of the subscriber.

Fig. 8 illustrates the subscription/ re-subscription overhead with the increasing degree of mobility. Both approaches respond well for increased mobility. Mobility of the client nodes would not affect the system performance too much.

The simulation results demonstrate the advantages of our proposed pub/sub approach: it is simple, and it generates less network traffic, consumes less memory storage, uses less computing power, and is highly mobility-tolerant.

V. CONCLUSTION

In order to deploy distributed applications inside WMNs, service/event notification plays an important role. In this paper, we presented a pub/sub communication paradigm to enable faster and easier dissemination of information inside WMN communities. In this communication paradigm, subscribers and publishers utilize a hash function to accurately locate related brokers at the WMN area. Brokers can then match the publications with the subscriptions and efficiently propagate notifications to the right subscribers through a multicast algorithm. The experimental evaluation shows that our approach is scalable and efficient.

ACKNOWLEDGMENT

The research developed in this paper is supported by NSF ND EPSCoR under the Infrastructure Improvement Program FAR-0015846.

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] I.F. Akyildiz, X. Wang, W. Wang, “Wireless mesh networks: asurvey, Elsevier Journal of Computer Networks”, Vol. 47, Issue 4, 2005, pp. 445- 487.

[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] I. Burcea, Muthusamy, V., Petrovic, M., Jacobsen, H.A., de Lara, E., “Disconnected operations in publish/subscribe”. In Proceedings of: IEEE Mobile Data Management, 2004, pp. 39-50.

[5] 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.

[6] C. Bettstetter and C. Wagner. “The spatial node distribution of the random waypoint mobility model”. IEEE Transactions on Mobile Computing, Volume 5, Issue 6, June 2006, pp.680 - 694.

[7] J.A. Briones, B. Koldehofe, and K. Rothermel. “SPINE: Publish/subscribe for wireless mesh networks through self-managed intersecting paths”. In Proceedings of the 8th IEEE International Conference on Innovative Internet Community Systems. 2008.

[8] A. Carzaniga, D.S. Rosenblum, A.L. “Wolf: Achieving scalability and expressivenessin an Internet-scale event notication service”. In: Proc. of ACM Symp. On Principles of Distributed Computing (PODC), Portland, OR , 2000, pp. 219-227

[9] 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.

[10] 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.

[11] D. Frey, G. Roman, “Context-aware publish/subscribe in mobile ad hoc networks”. In ninth international conference on coordination models and langrauges, 2007, pp. 37-55.

[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] 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.

[15] 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.

[16] A. Raniwala, T. Chiueh, “Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network”, 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE, Vol. 3 (22 August 2005), pp. 2223-2234.

[17] A. Raniwala, K. Gopalan, T. Chiueh, “Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks”, ACM Mobile Computing and Communications Review, Vol. 8, 2004, pp. 50-65.

[18] 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.

[19] P. Trianta llou, A. Economides, “Subscription summaries for scalability and effciency in publish/subscribe”. In: Proc. of Workshop on Distributed Event-BasedSystems, Vienna, Austria, 2002, pp. 619-624.

[20] 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.

[21] Y. Zheng, J. Cao, M. Liu, J. Wang, "Efficient event delivery in publish/subscribe systems for wireless mesh networks" 8th IEEE Wireless Communications and Networking Conference, 2007, pp. 4327-4332.