Fadip: Lightweight Publish/Subscribe for Mobile Ad Hoc Networks

Koosha Paridel, Yves Vanrompay, and Yolande Berbers

K.U. Leuven, Department of Computer Science
Celestijnenlaan 200A, Heverlee, Belgium
{koosha.paridel,yves.vanrompay,yolande.berbers}@cs.kuleuven.be
http://www.cs.kuleuven.be

Abstract. Publish/Subscribe is an interesting communication paradigm because it fosters a high degree of decoupling between the communicating parties and provides the ability to communicate in an asynchronous way. Publish/Subscribe systems have been extensively studied for wired networks but designing a Publish/Subscribe system for mobile ad hoc networks is still a challenge. In this paper we propose a lightweight Publish/ Subscribe system for mobile ad hoc networks which uses a limited gossip mechanism to match the published messages with the subscriptions. The goal of this work is to reduce the number of exchanged messages used for communication and maintenance while keeping an acceptable delivery ratio. Experimental results show that Fadip achieves an acceptable delivery ratio even in high mobility rates.

Keywords: Publish/Subscribe, MANET.

1 Introduction

Publish/Subscribe is an asynchronous communication model in a network. It has been a popular model because of its decoupling and asynchronous communication properties. The decoupling property allows the communicating parties to exchange information without knowing each other. The asynchronous communication property also allows them to communicate even though they are not online at the same time.

Mobile ad hoc networks (MANETs) on the other hand are networks of mobile nodes which can move freely and arbitrarily. MANETs have several characteristics[5]: (i) dynamic topologies, (ii) bandwidth-constrained, variable capacity links, (iii) energy-constrained operation and (iv) limited physical security. The need for mobile ad hoc networking technology is increasing and consequently there is a growing need for scalable and robust communication mechanism for MANETs.

Publish/Subscribe systems have been intensively studied for wired networks and infrastructured mobile networks. However, there are not many solutions for Publish/Subscribe systems in mobile ad hoc networks and all of these solutions suffer from scalability issues. These scalability issues can be categorized into:

R. Meersman et al. (Eds.): OTM 2010, Part II, LNCS 6427, pp. 798–810, 2010.

(i) large number of nodes, (ii) large network area, (iii) high mobility rates and (iv) large number of messages. In this paper, our focus is on the number of nodes and on mobility rate which all the current solutions are struggling with.

We propose a lightweight Publish/Subscribe systems for MANET called Fadip (Publish/Subscribe using FADing GossIP). In designing Fadip our goal is to make it lightweight in terms of the logical network structure (i.e repairing and maintenance needed) and the number of messages exchanged for communication. Meanwhile, we want to achieve a reasonable deliver ratio of the messages. We also aim to make a system suitable for high mobility rates and extremely volatile networks.

To achieve these goals, we propose to use a hybrid model by using bounded subscription propagation and publication propagation at the same time. The idea is to propagate subscriptions and publications as bounded as possible and make matchings in the intermediary nodes. This approach is to some extent similar to the idea of having rendezvous points which is used in Publish/Subscribe systems for wired network because the matching is done in some intermediary nodes which can be assumed as a kind of a rendezvous point. However, the difference is that unlike wired networks we cannot rely on any node as a dedicated rendezvous point. Thus, neither the publishers nor the subscribers have any information about where their publications and subscription might be matched. In this paper we describe the basic protocol which includes subscription propagation, publication propagation, matching and delivery. Subsequently, we introduce Fadip or *fading gossip* technique to further limit the number of messages. We also did extensive simulation studies to determine the effectiveness of the basic protocol and the effect of applying the Fadip technique.

The rest of this paper is structured as follows: We present the Fadip protocol in Section 2, then we describe the results of simulation in Section 3. Next, we discuss the related work in Section 4 and finally we conclude in Section 5.

2 The Fadip Protocol

The idea of this work is to use bounded propagation of both publish and subscribe messages and match them in intermediary nodes between a publisher and a subscriber. The goal is to increase the delivery ratio of the publications and decrease the number of multicasts. Although there are other important factors for a publish/subscribe system such as matching techniques and event expressiveness, we do not deal with them in this paper.

We propose a content-based publish/subscribe system for mobile ad hoc networks called Fadip. It is a light-weight protocol in the sense that it does not maintain any kind of logical structure for the mobile nodes such as a routing tree[10] or grouping clusters[14]. Fadip is content-based because it routes the publications to the subscribers by matching their content with the subscriptions. Because we are not dealing with expressiveness here we choose the publications to be in a range of integers specified by $value_{min}$ and $value_{max}$.

The network is considered to be an ad hoc network of homogeneous mobile nodes that are moving freely in a 2-dimensional playground with an area of $size_x$

by $size_y$. Every node has a transmission range and when a broadcast message is sent, all the nodes in the transmission range can receive that message. All nodes can have the roles of a publisher and a subscriber.

2.1 The Basic Protocol

The Fadip protocol consists of four operations: (i) subscription propagation, (ii) publication propagation, (iii) matching, and (iv) delivering. In the following we describe these operations but first we describe the properties of a message and basic propagation rules.

Messages and Maximum Hop Count: Each subscription is propagated up to a certain hop count which is called $hcMax_{subscribe}$. On the other hand, each publication is published up to hop count limit which is called $hcMax_{publish}$. We name these two parameters *Subscribe Penetration* and *Publish Penetration* recpectively. Depending on the number of nodes and the size of the playground these hop count limits should be chosen in a way to provide an acceptable probability for publications to be matched with subscriptions. As we show in Section 3 a proper hop count can be easily calculated after a few runs of the protocol. Each message (publish or subscribe message) has a variable which indicates the number of hops that this message is away from its producer. This variable is called *hc*. Each node, upon receiving a publish or subscribe message checks the value of *hc* with $hcMax_{publish}$ or $hcMax_{subscribe}$ respectively. If the value is smaller than the limit the node forwards the message by broadcasting it to the neighbors. This node also remembers forwarding this special message to avoid forwarding it again and make forwarding loops.

Subscription Propagation: If the forwarding message is a subscription, an intermediary node $node_i$ checks if it already received this subscription or not. Suppose that the originating node of this subscription is $node_a$. $node_i$ can receive a subscription earlier from $node_a$ or it can receive a similar subscription but from another node $node_b$ sooner. In this case $node_i$ checks the *hc* variable of the message which shows the number of hops that this message has been forwarded. If *hc* of the message is lower than the minimum *hc* value that $node_i$ has seen until this moment for this specific subscription, $node_i$ will replace his old value for this subscription with the newer and smaller value. For example if $node_i$ has a *hc* value of 5 for a subscription S and it receives a new subscription with *hc* value of 4, it will change the value of hc_S from 5 to 4. Finally $node_i$ will increase *hc* value of the forwarding messages and sends it to the neighbors.

Publication Propagation: If the forwarding message is a publication $node_i$ checks if it has already forwarded the same publication originating from the same publisher $node_a$. It prevents unnecessary and redundant forwarding of publications. If it is the first time that $node_i$ is receiving this message it increases the *hc* value of the messages and forwards it to the neighbors.

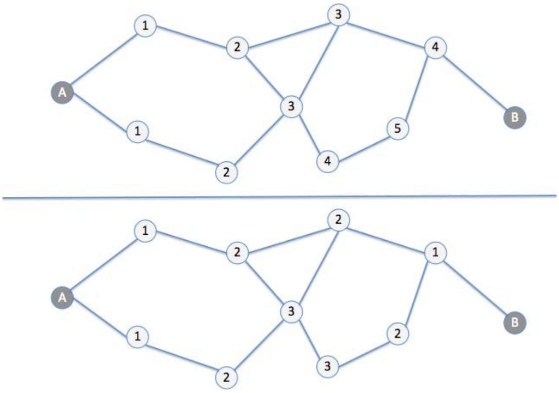


Fig. 1. Subscription Forwarding in Fadip. The upper figure shows hop count values for the intermediary nodes after a subscription propagation by Node A. The lower figure shows the same network after the propagation of the same subscription by Node B.

Matching and Delivering: Upon receiving a publication message m_p, $node_i$ checks if it has already received a subscription S which matches m_p[1]. If a match is found, m_p will be forwarded to the neighbors even if it has hc value more than $hcMax_{publish}$. However, it is still necessary to control the propagation depth to avoid unnecessary forwarding. Thus, for a matched publication, instead of limiting hc value to $hcMax_{publish}$ we limit it to $hcMax_{publish}+hcMax_{subscribe}$ so that it can have enough steps to reach the subscriber. Note that if for a message hc is exceeding $hcMax_{publish} + hcMax_{subscribe}$, it means that it is going on a wrong direction, because m_p should be able to reach from $node_i$ to the subscriber using the same path that the subscription was routed which is shorter or equal to $hcMax_{subscribe}$. When m_p is matched at the subscriber node it will not be forwarded anymore.

2.2 Fading Gossip

In addition to limiting data dissemination to a certain hop limit, to decrease the number of broadcasts we use another technique which we call it *fading gossip*. In fading gossip we decrease the fanout of each broadcast (which makes it a multicast) at every hop level. For example, the fanout for the message sender is 100% of its neighbors, while for the next level the fanout is 80% of neighbors, for the next level 60% and so on. This means that the message will be forwarded to a random fraction of neighbors and this fraction is equal to the fanout percentage. On the other hand we do not want to decrease fanout less than a certain threshold such as 50% of the neighbors. Hence, we define a formula to adjust the fanout

[1] As mentioned earlier we are not dealing with expressiveness in our Publish/Subscribe system. Thus, subscriptions and publication are positive integers ranging from $value_{min}$ and $value_{max}$.

at each hop level which starts from 100% and leans towards 50%. We use the formula:

$$fanout = \frac{1}{2} + \frac{1}{2}e^{-c.hc} \tag{1}$$

In this formula c is the coefficient used to speed up the fanout decrement and hc is the hop count of the forwarding message. For example, when $c = 1$ which means that we are using normal fanout decrement speed and $hc = 1$ which means a message is on first hop level (immediate neighbor of the message's source) we have a fanout equal to 0.684. This means that the second level neighbors will receive this message with a probability of 68.4%. Using similar calculation, the second level fanout is 0.568 and it will continue decreasing each level down to 0.5. However, in sparse networks where nodes do not have many neighbors, it is possible that a forwarding message gets eliminated in the early stages. For example, suppose a subscriber in a sparse network where it has only one neighbor. There is a good probability that using the fading gossip technique the subscription get stopped at the first neighbor and never reaches the other nodes in network. Therefore, there is a need to change the proposed formula so that it can adapt itself with sparse networks. In order to do that, we can suspend the formula from acting until a certain hop limit (e.g. first or second level of neighbors) and apply it after that limit. Thus, the adjusted formula becomes:

$$fanout = \begin{cases} 1 & \text{if } hc \leq \text{limit} \\ \frac{1}{2} + \frac{1}{2}e^{-c.hc} & \text{if } hc > \text{limit} \end{cases} \tag{2}$$

Using this formula to compute fanout, we can avoid the forwarding message to get eliminated in the beginning levels. For example, by setting limit to 1 we can make sure that the message will be broadcasted to all the second level neighbors and after that fading gossip technique takes effect.

In the following section we show the performance of basic protocol and the effect of fading gossip through extensive simulations.

3 Simulation

Fadip is simulated in OMNeT++ [13], an open source discrete event simulator, and using a mobility framework for OMNeT++ called MiXiM [9]. The nodes move in a playground with an area of 1500 meters by 1000 meters which resembles a campus area. The movement model used is the Random Waypoint model [3] which is considered as a benchmark mobility model to evaluate routing protocols in MANET [1]. The speed and the number of nodes changes in different simulations and will be given for each case.

Figure 2 illustrates the simulation environment with mobile nodes and the wireless links between the nodes represented by the images of laptops and black arrows respectively. In this example, there are 75 nodes in a playground of 1500 meters by 1000 meters. The mobile nodes move in the playground according to the movement model. When two nodes enter each other's wireless communication range a wireless link is established, and when they go out of this range the

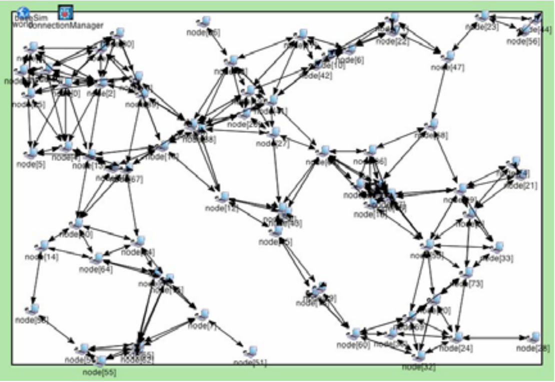
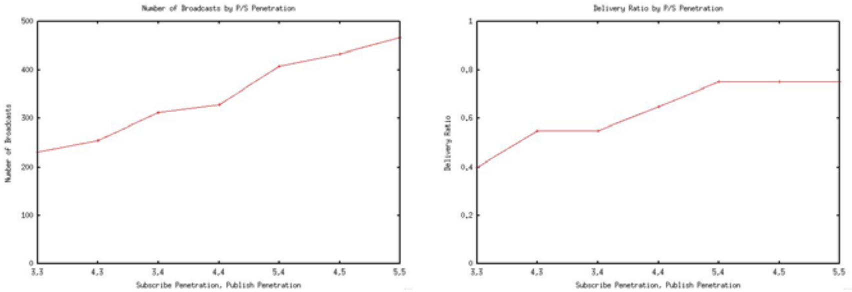


Fig. 2. Simulation in OMNeT++ using MiXiM. Each laptop icon represents a mobile node and arrows show direct wireless links between nodes. A link is established when two nodes are in each others wireless communication range. A link is broken when the two nodes are not in each other's wireless communication range anymore.

wireless link is broken. In the following, we describe the different simulating experiments that we have done to evaluate Fadip.

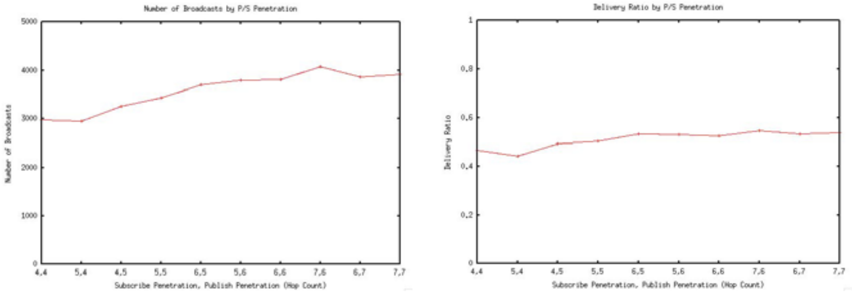
Propagation Limit: The goal of this simulation is to see the effect of propagation limit in terms of hop counts on delivery ratio of the messages and to observe the costs of it in terms of the number of sent messages. In this setup we use 75 nodes in the same playground of 1500 by 1000 meters. There is one random subscriber and 20 random publishers which each publisher publishes only one message. Figure 3 shows the effect of different values of $hcMax_{publish}$ and $hcMax_{subscribe}$ on delivery ratio and number of broadcasts. We did the same experiment with 150 nodes and 49 random publishers too (randomly chosen by 20% chance which is our simulation turned out to be 49). Figure 4 shows the result of using different publish and subscription penetration for a network of 150 nodes in the same playground which we used for the network of 75 nodes. However, this time we used multiple subscribers (5% of total nodes) instead of a single subscriber. The results show that increasing the penetration after a certain limit has a neutral or even negative effect on delivery ratio and at same time it increases the number of broadcasts rapidly. For example for the network of 75 nodes and value of 4 and 4 for $hcMax_{publish}$ and $hcMax_{subscribe}$ respectively gives about 65% delivery ratio and uses fewer broadcasts comparing to larger penetration (hop count limit) values. Also for the network of 150 nodes a penetration value of around 5, 5 or 6, 6 perform very well in term of delivery ratio and number of broadcasts. These results suggest that for every network, based on the size of playground and the number of nodes there is an efficient setting for the penetration limit which can be calculated easily. Next we see the effect of publish and subscribe penetration separately.



(a) Number of broadcast for different subscription/publish penetrations

(b) Delivery ratio for different subscription/publish penetrations

Fig. 3. The effect of subscription and publish penetration on number of broadcasts and on delivery ratio for a single subscriber and a network of 75 nodes



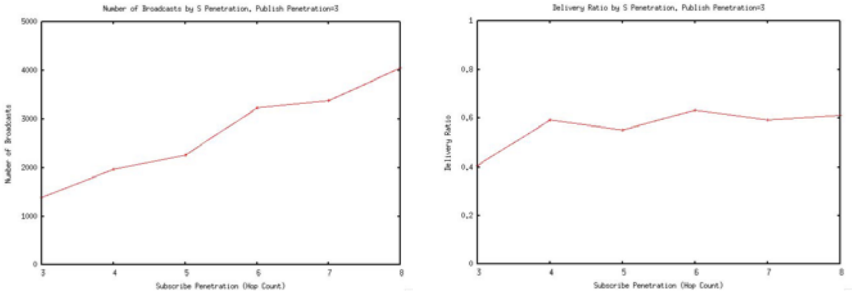
(a) Number of broadcast for different subscription/publish penetrations

(b) Delivery ratio for different subscription/publish penetrations

Fig. 4. The effect of subscription and publish penetration on number of broadcasts and on delivery ratio for a multiple subscribers and publishers in a network of 150 nodes

Subscribe Penetration: We did simulations having a constant publish penetration and a varying subscription penetration. The reason is to see the effect of changing a subscription penetration alone. Again we use a single subscriber and 49 random publishers. The network consists of 150 nodes and the playground size is the same as previous setups. Figure 5 shows the result of changing subscription propagation from 3 to 8 hops. The results suggest that increasing the subscription penetration after a certain limit does not have a positive effect on delivery ratio but increases the number of broadcast rapidly. Hence, there is an efficient threshold for subscription penetration that can be determined for a network easily through simulations or experiments.

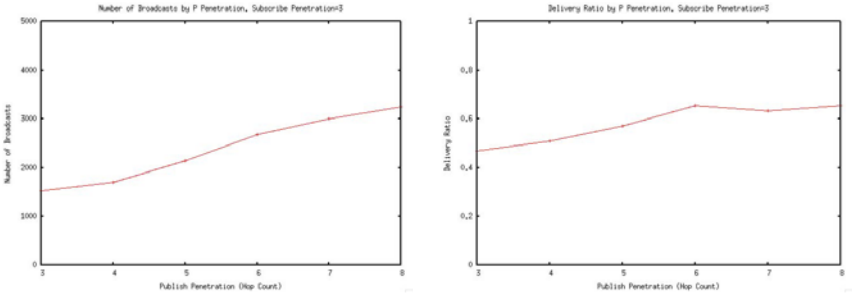
Publish Penetration: We also did simulations having a constant subscription penetration and a varying publish penetration. Figure 6 shows the result of changing publish propagation from 3 to 8 hops. The results again suggest that increasing the subscription penetration after a certain limit increases the number of broadcast sharply but does not provide more delivery ratio.



(a) Number of broadcast for different subscription penetrations

(b) Delivery ratio for different subscription penetrations

Fig. 5. The effect of subscription penetration on number of broadcasts and on delivery ratio for a single subscriber and a network of 150 nodes



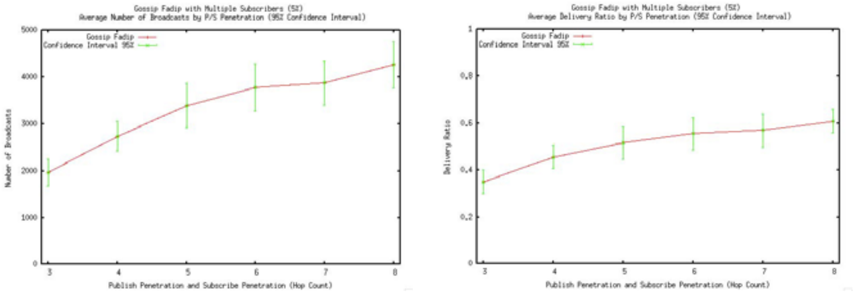
(a) Number of broadcast for different publish penetrations

(b) Delivery ratio for different publish penetrations

Fig. 6. The effect of publish penetration on number of broadcasts and on delivery ratio for a single subscriber and a network of 150 nodes

Multiple Subscribers: In order to verify the results we showed earlier with a single subscriber we also did simulations with multiple subscribers. In this setup we randomly chose 5% of nodes as subscribers and 15% of nodes as publishers in a network of 150 nodes. We ran 3 simulations for each publish/subscribe penetration setting and computed the average and a 95% confidence interval for the results. Figure 7 shows the results of running Fadip with multiple subscribers with the calculated confidence interval. The results show a similar trend with the single subscriber scenario.

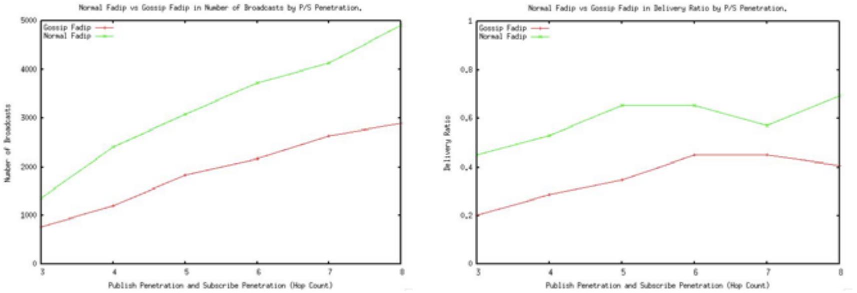
Fading Gossip: As described in section 2.2, we use the fading gossip technique to control the number of broadcasts. We did simulation to compare the basic protocol and fading gossip and to see the effects on the number of broadcasts and delivery ratio. Figure 8 compares the basic protocol and fading gossip in terms of the two mentioned parameters. The results show that using fading gossip technique limits the number of broadcasts considerably while loosing a



(a) Average number of broadcast by publish and subscribe penetration for Fadip with multiple subscribers

(b) Average delivery ratio by publish and subscribe penetration for Fadip with multiple subscribers.

Fig. 7. Fadip with multiple subscribers (5% of total nodes) and multiple publishers (15% of total nodes). The results are shown with 95% confidence interval. Network size is 150 nodes.



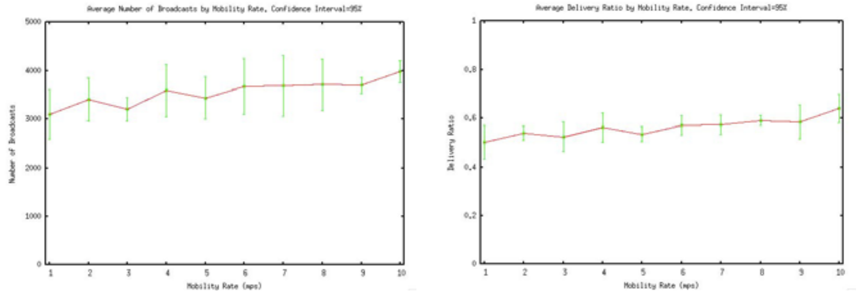
(a) Number of broadcast for basic protocol vs. Fadip with different publish and subscribe penetrations

(b) Delivery ratio for basic protocol vs. Fadip with different publish and subscribe penetrations

Fig. 8. Basic protocol vs. Fadip with different publish and subscribe penetrations for a network of 150 nodes

small percentage of delivery ratio, specially for higher publish and subscribe penetration values.

Mobility Ratio: An important factor for a Publish/Subscribe system for MANET is its behavior as mobility ratio increases. Higher mobility ratio invalidates routing information and logical network structures (e.g. clusters, routing trees) more rapidly. Therefore, as mobility ratio increases we expect to have a decrement in delivery ratio. In our simulations, we use a network of 150 nodes and change the mobility speed of nodes from 1 meter per second up to 10 meters per second. We a have 5% of nodes as subscribers and 15% of nodes as publishers. We ran each experiment 3 times and calculated the average with a confidence interval of 95% for the number of broadcasts and average delivery ratio. Figure 9 shows the effect of mobility ratio on number of broadcasts and



(a) Number of broadcasts for different mobility speeds

(b) Delivery ratio for different mobility speeds

Fig. 9. The effect of mobility ratio on number of broadcasts and on delivery ratio for a network of 150 nodes. (5% Subscriber, 15% Publisher).

on delivery ratio. Interestingly, there is no negative effect on delivery ratio. The reason for such a good behavior towards high mobility ratio is the fact that Fadip is not dependent on any logical network structure which can be damaged quickly by high mobility ratios. On the other hand, because of not relying on logical structures there is no need for maintenance communication and therefore there is no bad effect on the number of exchanged messages. That is an important result because Publish/Subscribe systems for MANET suffer heavily from the negative effects of increasing the mobility ratio.

4 Related Work

There are already many decent solutions for Publish/Subscribe systems in wired networks [2] and infrastructured mobile networks[4][7][8]. Recently, several pub/sub systems have been proposed for mobile ad hoc networks. In this section we describe several Publish/Subscribe systems for mobile ad hoc networks and we discuss their scalability in terms of network size and mobility ratio.

In Yuan et al. [15], the authors propose a pub/sub protocol for mobile ad hoc networks involving the construction of dynamic Voronoi regions. All nodes are divided into broker and non-broker categories, and each region is represented by a broker node. The authors propose mechanisms for constructing these Voronoi regions and for message deliveries.

Yoo et al. [14] also propose a pub/sub system for mobile ad hoc networks involving a hybrid model which uses flooding and content-based routing techniques hierarchically. They divide the network into clusters of nodes and use the event flooding technique for inter-cluster communication and content-based routing for intra-cluster communication. They try to utilize the advantages of content-based routing to achieve more efficient subscription propagation and maintenance and also reduce the costs of document delivery, while avoiding the high topology maintenance costs of content-based routing by using clusters and using flooding techniques for inter-cluster communication.

Also in Mottola et al. [10], the authors propose a protocol to organize the nodes of a mobile ad hoc network in a single, self-repairing tree structure which can be used for efficient content-based routing in the network. Their idea of maintaining the tree topology is inspired by a protocol for multicast over mobile ad hoc networks called MAODV.

Costa et al. [6] present a semi-probabilistic content-based pub/sub system in which the authors propose to use deterministic decision for event routing when there exists subscription information and to use probabilistic decisions when lacking subscription information. To do this, they propagate the subscription in the surrounding area of a subscriber. Upon receiving a forwarding event if subscription information is available, the event will be routed along the link that the subscription was received. If no subscription information is available, the event will be routed to a randomly chosen set of neighbors. The authors argue that their technique performs better than a fully deterministic and a fully probabilistic approach.

In Rezende et al. [12] the authors propose a pub/sub system for mobile ad hoc networks which does not build routing trees or other logical structures that need to be maintained. However, their goal is just to deliver publications using the minimum amount of broadcasts and they do not assume any time constraints for that. As they state in their paper their system is only suitable for applications that can tolerate delays of dozens of minutes. They also assume that mobile devices move and stop continuously and they are able to know if they are moving or not.

While [14] and [10] provide some analysis based on mobility ratio, [6], [15] and [12] do not deal with mobility ratio in their evaluations. In [6] the authors do simulation with a single moving speed of 2 m/s. In [12] nodes move with a randomly chosen speed of 1.5 to 4.0 m/s. It is also a similar case with [15] in which the range of randomly chosen speeds is between 10 to 20 m/s. In [14] the authors study the effect of mobility ratio on delivery ratio by changing the maximum speed of nodes from 1 m/s to 10 m/s. However, their results show a rapid decrement in delivery ratio as nodes move faster. In [10] the authors do not evaluate the performance of their system based on mobility rate. They only state that because the results of their simulations for nodes moving at 1 m/s and nodes moving at 10 m/s are not much different, their system is reasonably independent of node speed.

The routing mechanism used in Fadip is to some extent similar to Gradient Routing which is presented in [11]. In Gradient Routing each node maintains a cost table which keeps the estimated cost of sending a message to a target node. When a node forwards a message, only those neighbors which can deliver the message with a lower cost will participate in forwarding it. The concept of maintaining a cost table based on hop count is somehow similar to what we do in Fadip by keeping the distance of intermediary nodes in terms of hop count from a subscriber. Gradient Routing is a routing mechanism and it routes a message to a particular destination, and the originator is aware of the existence of the destination and its identification information. However, in Fadip a publisher is not aware of the subscribers. A publisher propagates its message in the network up to certain limit and hopes that its message would be matched in some places in the network with subscriptions. Moreover, for each publication in Fadip multiple destinations can exist. Thus, a publication can be forwarded in several branches to reach different subscribers, in contrast with Gradient Routing in which a message only follows the shortest route to its single destination. After the first matching, we forward a publication through all neighbors which have received a matching subscription and not only through those that have a shorter distance.

5 Conclusion

In this paper we presented Fadip, a light-weight Publish/Subscribe system for mobile ad hoc networks. Fadip uses controlled propagation of both publications and subscriptions and does the matching in the intermediary nodes. First, we described the basic protocol in which the propagation is done to all neighboring nodes at each level and next we introduced the fading gossip technique to further limit the number of broadcasts. Results show that Fadip tolerates high mobility rates without loosing any considerable delivery ratio.

References

1. Bai, F., Helmy, A.: A survey of mobility models in wireless adhoc networks. In: Wireless Ad Hoc and Sensor Networks. Kluwer Academic Publishers, Dordrecht (2004)
2. Baldoni, R., Virgillito, A.: Distributed event routing in publish/subscribe communication systems: a survey. DIS, Universita di Roma" La Sapienza", Tech. Rep. (2005)
3. Broch, J., Maltz, D.A., Johnson, D.B., Hu, Y.C., Jetcheva, J.: A performance comparison of multi-hop wireless ad hoc network routing protocols. In: Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking, p. 97. ACM, New York (1998)
4. Burcea, I., Jacobsen, H.A., De Lara, E., Muthusamy, V., Petrovic, M.: Disconnected operation in publish/subscribe middleware. In: 2004 IEEE International Conference on Mobile Data Management, Citeseer
5. Corson, S., Macker, J.: Mobile ad hoc networking (manet): Routing protocol performance issues and evaluation considerations. Request for comments 2501 (1999)
6. Costa, P., Picco, G.: Semi-probabilistic content-based publish-subscribe. In: International Conference on Distributed Computing Systems, vol. 25, p. 575. Citeseer (2005)
7. Fiege, L., Gartner, F.C., Kasten, O., Zeidler, A.: Supporting mobility in content-based publish/subscribe middleware. LNCS, pp. 103–122. Springer, Heidelberg (2003)
8. Huang, Y., Garcia-Molina, H.: Publish/subscribe in a mobile environment. Wireless Networks 10(6), 643–652 (2004)
9. Köpke, A., Swigulski, M., Wessel, K., Willkomm, D., Haneveld, P.T., Parker, T.E.V., Visser, O.W., Lichte, H.S., Valentin, S.: Simulating wireless and mobile networks in omnet++ the mixim vision. In: Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), p. 71 (2008)

10. Mottola, L., Cugola, G., Picco, G.P.: A self-repairing tree topology enabling content-based routing in mobile ad hoc networks. IEEE Transactions on Mobile Computing, 946–960 (2008)
11. Poor, R.D.: Gradient routing in ad hoc networks (2000)
12. Rezende, C.G., Rocha, B.P.S., Loureiro, A.A.F.: Publish/subscribe architecture for mobile ad hoc networks. In: Proceedings of the 2008 ACM symposium on Applied computing, pp. 1913–1917. ACM, New York (2008)
13. Varga, A., et al.: The omnet++ discrete event simulation system. In: Proceedings of the European Simulation Multiconference (ESM 2001), pp. 319–324 (2001)
14. Yoo, S., Son, J.H., Kim, M.H.: A scalable publish/subscribe system for large mobile ad hoc networks. The Journal of Systems & Software 82(7), 1152–1162 (2009)
15. Yuan, Q., Wu, J.: Drip: A dynamic voronoi regions-based publish/subscribe protocol in mobile networks. In: IEEE INFOCOM, Citeseer (2008)