

# Dynamic Connectivity Analysis of ABF-based Ad-hoc Networks

Fei Liu, Geert Heijenk

University of Twente, P.O. Box 217,  
7500 AE Enschede, The Netherlands

{fei.liu, geert.heijenk}@utwente.nl<sup>1</sup>

**Abstract.** This paper evaluates the performance of a context discovery protocol using attenuated Bloom filters while nodes appear, disappear, and are temporary unreachable in ad-hoc networks through analytical analysis. Further we verified obtained results with simulation analysis. The analytical results are most accurate when node density is high. Further, we discover that adding a node generates less traffic than removing a node in our protocol. Some proposals are given to reduce the traffic generated by propagating updates due to removing a node.

## 1 Introduction

Context-aware ad-hoc networks adapt their behavior, based on the context in which they operate. For this purpose, nodes use information from context sources. To discover these sources, a context discovery protocol is needed. Ad-hoc networks are severely limited in resources, such as communication bandwidth, energy usage, and processing power. To save communication resources, we have proposed to perform context discovery using attenuated Bloom filters (ABFs) [8]. We have proven that using ABFs can well save the traffic compared with conventional approaches in realistic situations.

Another important feature of ad-hoc networks is dynamics in connectivity. In this paper, we present an investigation of the impact of network dynamics on our ABF-based discovery protocol through an analytical approach. In general, three categories of causes of network dynamics can be identified: nodes may be mobile; battery-supplied devices might exhaust their batteries; the quality of the wireless transmissions might be varying due to varying propagation conditions.

Because of the random position and movement of the nodes, it is too complex to quantify the network traffic in a mobile environment mathematically. Thus, simulation is a good approach to study this problem. [5] has examined the net-

---

<sup>1</sup> This work is part of the Freeband AWARENESS project (<http://awareness.freeband.nl>). Freeband is sponsored by the Dutch government under contract BSIK 03025.

work traffic generated by updating the ABFs while nodes are moving in a low density network, and the reach-ability of the required services through simulations. In this paper, we present an analytical modeling of the dynamics due to the limited battery-supply and unstable transmission quality in very high density networks. First, we will consider node disappearance and appearance. When a node is powered off, it disappears from the network. After it switches on again, it joins the network again. We quantify the network load through analytical study and verify obtained results with simulations. Further, we observe a special case where the packets transmitted by a node get lost for certain time due to the poor propagation conditions. In this scenario, the node is considered as disappearing and reappearing in the network. We obtain simulation results for various packet loss periods.

This paper is structured as follows. Section 2 gives a brief introduction of the ABF-based context discovery protocol for ad-hoc networks. Section 3 introduces discusses related work, the assumptions we use in our analysis, and an approximation for the basic notion of  $i$ -hop node degree. Section 4 presents the analysis of network traffic when nodes appear and disappear in the network, or when a series of consecutive advertisement packets are lost. Section 5 concludes the study and discusses the future work.

## 2 ABF-based Ad-hoc network

### 2.1 Attenuated Bloom filters (ABF)

Bloom filters [3] have been proposed in 1970s to represent a set of information in a simple and efficient way. They use  $b$  independent hash functions to code the information. The hash results are over a range  $\{1..w\}$ , where  $w$  denotes the width of the filter. In the filter, which has a length of  $w$  bits, every bit is set to 0 by default. Only the bit positions associated with the hash results will be set to 1. The resulting Bloom filter can be used to query the existence of certain information. If all the bit positions related to the hash results of the queried information are 1 in the filter, the information exists with small chance of false positive.

Attenuated Bloom filters (ABFs) are layers of basic Bloom filters. We use ABFs to represent information regarding the presence of context sources on a hop-distance basis [8]. The  $i$ th layer of an ABF ( $0 \leq i < d - 1$ ) aggregates all information about context sources  $i$  hops away. The depth of the ABF,  $d$ , also stands for the total propagation range of the information. Note that context sources reachable in  $i$  hops may also be reachable via longer paths. As a result, hash results at larger  $i$  will often be repeated in lower layer  $j$  ( $j > i$ ).

Fig.1 exemplifies the context aggregation operation for a node with two neighbors. Each node has an ABF with 8 bits width ( $w=8$ ) and the depth of 3 ( $d=3$ ). The node uses two hash functions ( $b=2$ ) to encode its local context sources “temperature” and “humidity” into  $\{2,8\}$  and  $\{2,5\}$  respectively. If we set the corresponding bit positions, we can obtain *filter\_local* as shown in Fig.1. When

the node receives the incoming filters  $filter\_in[1,..]$  and  $filter\_in[2,..]$  from its neighbors, it shifts the received filters one layer down and discards the last layer.  $filter\_in[1,..]$  and  $filter\_in[2,..]$  are obtained. We perform a logical OR operation on each set of corresponding bits of  $filter\_local$ ,  $filter\_in[1,..]$ , and  $filter\_in[2,..]$ .  $filter\_out$  can be obtained as the ABF that the node broadcasts to its neighbors. This filter contains the local information of the node on layer 0; one hop neighbors' information on layer 1; and two hop neighbors' information on layer 2.

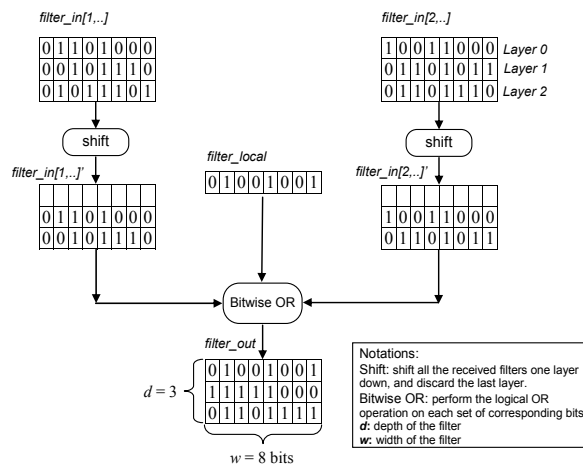


Fig. 1. An example of ABF aggregation

## 2.2 Protocol specification

Our ABF-based context discovery protocol distinguishes 3 phases [9]: context exchange, context query, and context update and maintenance.

**Context exchange:** every node stores two kinds of ABFs: incoming ABFs for each neighbor and an aggregated outgoing one with all local and neighboring information. When a new node joins the network, it will broadcast an ABF with only the local information first. Any node receiving this ABF will update its outgoing ABF with the new information and broadcast it. Once the newly joined node receives the neighboring information, it updates its outgoing ABF and broadcasts it. Every neighboring node will aggregate this update into its outgoing ABF as well. If there is any change in the outgoing ABF, the updated ABF will be broadcast to the network. After the exchange of ABFs, every node will have a clear view of the context information present within  $d$  hops.

**Context query:** whenever a query is generated in a node, the node first looks for presence of the information locally. If the required information is not available locally, it will hash the query string and check it against the stored neighboring ABFs. If there is no match, the query will be discarded. If there is any match, a query message will be uni-casted to that neighbor with a *hop-counter* set to  $d$ . The neighbor will perform the same action. It checks the query against the locally

available context sources. If there is any match, a response message will be sent back to the querying node. If nothing matches, it will check the stored neighboring ABFs. Whenever there is a match, the query will be propagated to that node with the *hop-counter* decreased by 1. When the *hop-counter* equals 0, the query will stop spreading. If a node receives the same query multiple times, as identified by a unique query ID, the query will simply be dropped.

**Context update and maintenance:** if there is no change in the context sources offered by a node, a keep-alive message will be sent out periodically. A keep-alive message is a short message with a *generation-id* of the last broadcasted ABF from this node. A node can identify the freshness of the stored ABFs by comparing *generation-ids*. Once it notices the *generation-id* is different from that of the stored ABF for this neighbor, an update request is sent out. The neighbor replies back with its latest ABF. If a node does not receive keep-alive messages from a certain neighbor for two consecutive *keep-alive periods*, it considers the node has left and removes its neighbor's information.

### 3 Modeling Preliminaries

In this section, we introduce related work and background knowledge regarding graph theory. Further, we introduce our modeling assumptions, and the important notion of the multi-hop node degree. We conclude with an approximation of the mean multi-hop node degree, which is an essential component of our further analysis.

#### 3.1 Related graph theory

Networks can be represented as a graph  $G=G(V,E)$ , where nodes are vertices ( $V$ ), while links between nodes are edges ( $E$ ). The degree of a vertex can be defined as the number of edges incident to it. If each vertex has degree of at least  $k$ , the graph can be called  $k$ -connected. Random geometric graphs [10] and unit disk graphs [4] are often used to model wireless ad-hoc networks. In ad-hoc networks, nodes are only directly connected to each other if they are within each others vicinity. The vicinity of those wireless nodes mostly depends on the geographic distance and transmission power. In both of the graph types mentioned above, vertices are connected, only if the distance between them is at most a threshold  $r$  (referred to as the communication range of wireless nodes). Both graphs are considered to be among the most realistic methods to model wireless ad-hoc networks. A random geometric graph is a graph in which the vertices or edges are associated with geometric objects or configurations. In a random geometric graph, vertices are placed random uniformly and independently in a geographic area, while in unit disk graph there is no specific restriction for vertices distributions. In here, we assume nodes are uniformly distributed in the network. Therefore, in this paper, we analyze ad-hoc networks with random geometric graph theory.

Using graph theory can help us to analyze some specific network characteristics. Most studies have been done in the area of node degree and connectivity. [4]

has discussed the connectivity problems of unit disk graph. [2] has investigated the relationship between required range  $r$ , node density, and almost certainly  $k$ -connected networks, assuming random geometric graphs. The results provide the principles to choose practical values of those parameters for simulations and design. [6] has shown that the degree distribution in wireless ad-hoc networks, modeled as a random geometric graph, is binomial for low values of the mean degree.

Besides using graph theory, some other approaches have also been used to investigate the connectivity problems in ad-hoc networks. [1] has examined the connectivity of a certain number of mobile nodes within a certain area by using a stochastic activity model. However, the study is limited to low density networks due to the limitation of the stochastic model. Besides this analytical analysis, some studies have been done by means of simulation [11] and test bed [7].

### 3.2 Multi-hop node degree

The current research of connectivity mostly focuses on the following two major questions: (1) how to achieve a  $k$ -connected network; (2) what is the degree distribution of a node. For instance, [6] has studied the degree distribution of one node in ad-hoc networks through a combination of analytical modeling and simulation. It is binomial for low value of the mean degree. Therefore, by given network density and communication range, we can obtain the distribution of number of direct neighbors one node has.

However, we have not found any research describing the degree distribution multiple hops away. In here, we define the  $i$ -hop node degree to be the number of nodes one node can reach within exactly  $i$  number of hops, but not fewer than that. Let us observe the neighbors of one node. Let  $N_i$  denote the  $i$ -hop node degree. For a random geometric graph, the distribution of  $N_i$  can be derived from  $N_{i-1}$ , conditioned on the position and the number of the nodes reachable in  $i-1$  hops. Theoretically, we can derive  $N_i$  in this way. However, the formula is going to be too complex to calculate. We cannot obtain the exact distribution formula for more than one hop. Therefore, we will take a step back and observe the upper bound of this problem.

### 3.3 Modeling assumptions

As we mentioned above, we model our network based on random geometric graph, where nodes are uniformly distributed in a certain area. To observe the entire network as one graph, we assume the graph is connected, which implies that no node is isolated. For a given node density  $n$ , total number of nodes  $Num$ , and the probability that the network has no isolated node  $p_c$ , we can obtain the minimum communication range  $r_0$  for which there is no node isolated in the network [2]:

$$r_0 \geq \sqrt{\frac{-\ln(1 - p_c^{1/Num})}{n\pi}}. \quad (1)$$

In our model, we abstract from the fact that communication between two nodes is subject to various kinds of time- and place-dependent propagation effects, which would imply that the communication range is also varying with time and place. Therefore, we fix the communication range for each node to  $r$ . To simplify our analysis, and to achieve with high probability a network without isolated nodes, we assume a very high-density network.

### 3.4 *i*-hop communication range

In line with our assumptions, a node can reach all the nodes located within the circle with the radius of  $r$  whose center is the position of the node A as shown in Fig. 2a. Node B, which is located within the annulus  $R_2$  with outer circle radius  $2r$  and the inner circle radius  $r$ , will reach the center node A, if and only if there is a node C located within the intersection area  $S_{AB}$  of the communication range of A and B, as shown in Fig. 2a. Because the distance between A and B is between  $r$  and  $2r$ , the intersection of circle A and B  $S_{AB}$  is between 0 and  $\frac{2}{3}\pi r^2 - \frac{\sqrt{3}}{2}r^2$ .

Since we have assumed that nodes are uniformly distributed in the network, the number of nodes located in the intersection area  $S_{AB}$  fits the Poisson distribution with  $\lambda_{AB} = S_{AB} \cdot n$ . We set the probability that there is at least one node located in the area  $S_{AB}$  as  $P(N_{AB} > 0)$ , which is also the probability of having a path between A and B. This probability equals 1 minus the probability that no node is located in the area  $S_{AB}$ :

$$P(B \text{ is a 2-hop neighbor of } A \mid B \text{ is in } R_2) = P(N_{AB} > 0) = 1 - P(N_{AB} = 0). \quad (2)$$

Since the number of nodes is Poisson distributed, formula (2) can be rewritten as:

$$P(N_{AB} > 0) = 1 - e^{-\lambda_{AB}} = 1 - e^{-S_{AB} \cdot n} \quad \left( 0 < S_{AB} < \frac{2}{3}\pi r^2 - \frac{\sqrt{3}}{2}r^2 \right). \quad (3)$$

We can observe that if  $n$  is sufficiently large,  $P(N_{AB} > 0)$  goes to 1. This implies that with almost 100% probability there is a path between node A and B if the node density is sufficiently high:

$$\lim_{n \rightarrow \infty} P(B \text{ is a 2-hop neighbor of } A \mid B \text{ is in } R_2) \rightarrow 1. \quad (4)$$

Let us now have a look at node F in Fig. 2b, which is located within the annulus  $R_3$  with outer circle radius  $3r$  and the inner circle radius  $2r$ . Node F can reach the center node A, if and only if there is at least one node E located within the communication range of node F, and that node has a connection to node A. Therefore, the probability that node F is a 3-hop neighbor of A can be derived as:

$$\begin{aligned} & P(F \text{ is a 3-hop neighbor of } A \mid F \text{ is in } R_3) \\ &= P(\exists E : d(E, F) \leq r \wedge E \text{ is a 2-hop neighbor of } A \mid F \text{ is in } R_3) \end{aligned} \quad (5)$$

From Fig. 2b, we can observe that if node E is located outside ring  $R_2$ , the probability that E is a 2-hop neighbor of A is 0. Moreover, when the network density goes to infinite, from formula (4), we can obtain:

$$\lim_{n \rightarrow \infty} P(F \text{ is a 3-hop neighbor of } A \mid F \text{ is in } R_3) \rightarrow \lim_{n \rightarrow \infty} P(\exists E : d(E, F) \leq r \wedge E \text{ is in } R_2 \mid F \text{ is in } R_3) \quad (6)$$

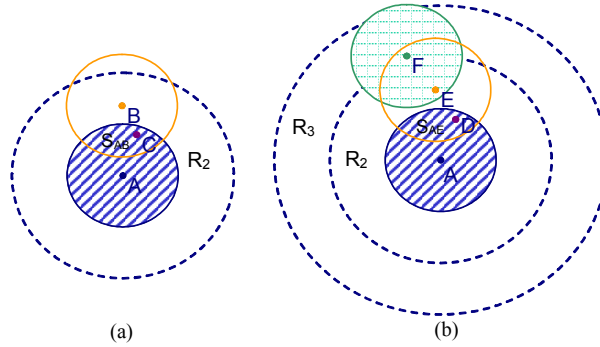
Since  $n$  goes to infinite, we have:

$$\lim_{n \rightarrow \infty} P(\exists E : d(E, F) \leq r \wedge E \text{ is in } R_2 \mid F \text{ is in } R_3) \rightarrow 1. \quad (7)$$

In a similar way, we can deduce the formula for the probability of node X located in  $R_i$  is a  $i$ -hop neighbor node A as:

$$\lim_{n \rightarrow \infty} P(X \text{ is a } i\text{-hop neighbor of } A \mid X \text{ is in } R_i) \rightarrow \lim_{n \rightarrow \infty} P(\exists Y : d(Y, X) \leq r \wedge Y \text{ is } (i-1)\text{-hop neighbor of } A \mid X \text{ is in } R_i) \quad (8)$$

$$\rightarrow \lim_{n \rightarrow \infty} P(\exists Y : d(Y, X) \leq r \wedge Y \text{ is in } R_{i-1} \mid X \text{ is in } R_i) \rightarrow 1$$



**Fig. 2.** (a) A and B are connected through C; (b) A and F are connected through E and D.

This shows that in a high-density network, when node X is located in  $R_i$ , X is an  $i$ -hop neighbor of A with almost 100% probability. Therefore, we assume a high density network in this paper. This implies that with very high probability, the network is connected. Further, adding or removing a node in the network will not influence the length of the shortest path between any two other nodes in the network. Formula (8) shows that in a high-density network, the probability of any node located in  $R_i$  being node A's  $i$ -hop neighbor goes to 1. That implies the  $i$ -hop communication range of node A goes to  $ir$  in a high-density network, which can be represented as:

$$\lim_{n \rightarrow \infty} (i\text{-hop communication range of } A) \rightarrow ir. \quad (9)$$

Therefore, in this paper we can use  $ir$  as our approximate  $i$ -hop communication range of node A with the assumption of a high-density network. The accuracy of this approximation depends on the actual network density, being highest at very high density.

### 3.5 Mean multi-hop node degree

We define the number of nodes that can be reached within  $i$  hops, but not fewer than  $i$  hops, as random variable  $N_i$ . Given formula (8), those nodes are located in the annulus with outer circle radius as  $ir$  and the inner circle radius as  $(i-1)r$  in high density networks. The expected value of  $N_i$  can be written as:

$$\lim_{n \rightarrow \infty} E[N_i] = \begin{cases} 1 & i = 0 \\ n\pi r^2(i^2 - (i-1)^2) = (2i-1)n\pi r^2 & (i > 0) \end{cases} \quad (10)$$

The total number of reachable nodes in  $i$  hops can be derived as  $Totalnumofnodes_i = \sum_{j=0}^i N_j$ . The expected total reachable nodes in  $i$  hops can be derived as:

$$\lim_{n \rightarrow \infty} E[Totalnumofnodes_i] = \lim_{n \rightarrow \infty} E\left[\sum_{j=0}^i N_j\right] = \lim_{n \rightarrow \infty} \sum_{j=0}^i E[N_j] = 1 + n\pi r^2 i^2. \quad (11)$$

## 4 Analysis of Dynamic Connectivity

In this section, we analyze the effect of dynamic connectivity, e.g., due to the limited battery supply and unstable transmission on the performance of our ABF-based discovery protocol. We study three different cases: a node disappears, appears, or is temporarily unreachable. When a node runs out of battery for instance, it disappears from the network. After it replaces the battery, it appears again as a brand new node. When for some reason the propagation conditions of the wireless medium are bad, some packets of a node might get lost. If these packets are a number of consecutive keep-alive messages, other nodes in the network will consider this node disappeared. After some time, the propagation conditions may improve so that the nodes packet will be received again. As a result, it reappears in the network. We will quantify the extra traffic load caused by those three different cases through an analytical approach and verify with the results from simulation.

Our model has been implemented with the discrete event simulator OPNET Modeler version 11.5. We observe the node density influence on the traffic load in all three experiments below. We place 25, 61, 100, 125, 150 nodes randomly into a  $1700 \times 1700 \text{m}^2$  area with 300 meters communication range,  $r$ , for every node. Note that the 61 nodes scenario can generate networks that are 1-connected graphs with 90% probability (see formula (1)). Therefore, we consider the 25 nodes scenario as a low density network, and the 150 nodes scenario as a high-density network. We expect that simulation results obtained for this high-density network are close to our analytical analyses, as this analysis was based on the assumption of a very high node density. The node, which disappears, or appears, or is temporarily unreachable, is located in the center of the area to avoid border



effects. For each parameter setting of the simulations introduced below, 30 independent runs will be done to calculate a 90% confidence interval.

Some basic ABF parameters are set as follows: number of hash functions per service,  $b = 10$ ; ABF width,  $w = 1024$  bits; ABF depth,  $d = 3$ ; number of context sources advertised per node,  $s = 1$ .

#### 4.1 Node disappearance

We start the analysis with the case of one node disappearing from the network. We assume that a node disappears at the moment the network has reached the stable state, i.e. all ABFs are up-to-date at the moment the node disappears. Possible reasons for node disappearance are insufficient battery supply, entering deep power saving mode, system shut down, un-functional antenna, system crash, etc. The absence of node A will be discovered by its direct neighbors when no keep-alive messages have been received for two consecutive keep-alive periods. Since the keep-alive period is unsynchronized, one of the direct neighbors B will notice this first. This node will remove the incoming ABF from node A. As a result, the representation of node A's context sources will be removed from layer 1 of B's outgoing ABF. For ease of explanation we will write that a node is advertised (or removed), where we actually mean that the context sources of a node are advertised (or removed). Since the other direct neighbors of A have advertised A in layer 1 of their ABF, node B will continue to advertise A in layer 2 of its ABF. As a matter of fact, we duplicate the local context sources of each node to every lower layer from layer 2 in the advertisement and maintenance phase of our protocol. This is because in a very high density network, if a path exists to a node, there are always longer paths to the same node. By duplicating a node's context sources to all lower layers in the ABF, we avoid that extra advertisements are exchanged to announce these longer path. So in the situation above, node B will still think that it can reach the services of node A via other neighbors. Therefore, those absent services are only removed from Layer 1 of the outgoing ABF from B.

There are two kinds of nodes that are the direct neighbor of B: direct neighbors of A and two-hop neighbors of A. The other direct nodes which receive this information will not take any action since they still think there are other routes to. As the last direct neighbor of A notices the disappearance of node A, it will realize that it cannot reach the absent services within one or two hops. It will start sending out an updated ABF with layer 1 and layer 2 cleaned up. Nodes which are two-hop neighbors of A that receive the updated ABF from direct neighbors of A will also take no action, until all their direct neighbors which are also the direct neighbors of A notice the absence of A. Only then these nodes will realize there is no path to A with 2 hops. They will send out an ABF with layer 2 cleaned up (Note that these two-hop neighbors do not have node A's information on layer 1). This clean up will be spread till  $(d-1)$ -hop neighbors based in a similar way.

It is a very complex procedure to clean up the services. Every node  $i$ -hop away has to clean up  $(d-i)$  layers in total. It will clean up layer  $j$  ( $i \leq j \leq d-1$ ) once it real-

izes there is no route to the service within  $j$  hops. Every node cleans up the service layer by layer. Only the last node in the  $i$ th hop that realizes the absent service will clean up two layers at once. When removing the context sources of node A from the ABF at a certain layer, there is a slight chance that no changes to the ABF are required, because the bits that would have to be set to zero have to remain one, as they also represent other context sources in other nodes. This is the same property that causes a false positive when querying context sources. In [8] we have already defined and derived this probability. We use  $P_{fp,i}$  to represent the false positive probability of layer  $i$ . [8] has proved that

$$P_{fp,i} \approx \left(1 - e^{-\frac{bx_i}{w}}\right)^b \quad (12)$$

where  $x_j$  denotes the total number of  $i$  layers context sources in layer  $i$ , with  $x_i = s \cdot (1 + n\pi \cdot i^2)$ .  $P_{fp,i}$  is the probability that no changes have to be made to layer  $i$  of an ABF, upon the disappearance of node A, provided that node A has only one context source advertised. If node A advertises  $s$  context sources, the probability that no changes have to be made to layer  $i$  is raised to the power  $s$ , i.e.  $P_{fp,i}^s$ . Therefore, we can derive the expected number of clean-up updates while one node disappears,  $E[N_{updates-disa}]$ , as:

$$\begin{aligned} E[N_{updates-disa}] &\approx \sum_{i=1}^{d-2} \left( (E[N_i] - 1) \cdot \sum_{j=i}^{d-1} (1 - P_{fp,j}^s) + \sum_{j=i+1}^{d-1} (1 - P_{fp,j}^s) \right) + E[N_{d-1}] \cdot (1 - P_{fp,d-1}^s) \\ &\approx \sum_{i=1}^{d-1} E[N_i] \cdot \sum_{j=i}^{d-1} \left( 1 - \left(1 - e^{-\frac{bx_j}{w}}\right)^{sb} \right) - \sum_{j=2}^{d-1} \left( 1 - \left(1 - e^{-\frac{bx_j}{w}}\right)^{sb} \right) \end{aligned} \quad (13)$$

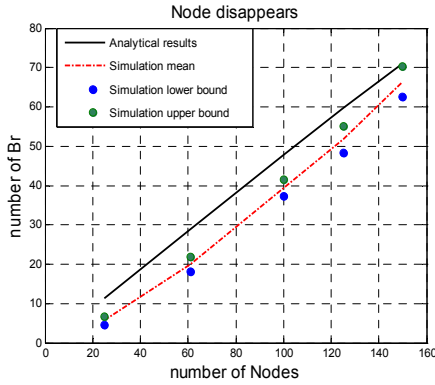


Fig. 3. Node disappears

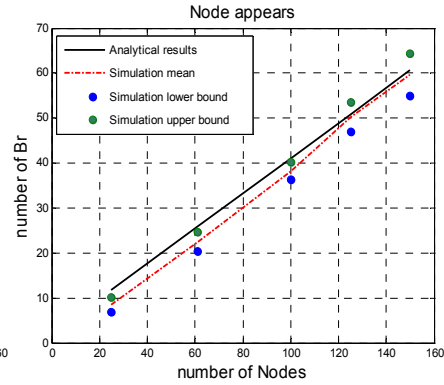


Fig. 4. Node appears

We verify the results of this approximation with simulations. In here, we study the traffic load generated by one node disappearance under different network

density see Fig.3. We found that the analytical results are slightly higher than the simulation results. This is because our analytical analysis is based on the assumption of a very high density network, so that the  $i$ -hop node degree is slightly over-estimated. In that respect, our analysis provides an upper bound to the expected number of updates. In our scenarios, there are fewer nodes involved in updating and more nodes clean up more than one layer of filter at once. This results in fewer update packets sent out than we estimate in formula (13). We also observe that the higher the network density, the smaller difference there is between analytical and simulation analysis as we expected from our assumption.

#### 4.2 Node appearance

In this section, we consider the scenario that one new node appears in the network. The reason could be that the node just switches on. We assume that the new node is familiar with the standardized format and hash functions of attenuated Bloom filters used in the current network. However, the node does not have knowledge about nodes and context sources in the network. First of all, this node will broadcast its filter (size of  $w \times d$  bits) with its local services. The direct neighbors who receive this filter will update theirs. Those new filters will be broadcasted around. The new node waits for a short moment, till it receives all the neighbors' replies. It aggregates all incoming filters, updates its own filter and broadcasts it. Note that, the network is assumed to have a high node density, which means that the appearance of the new node will not generate any shorter path between any pair of existing nodes. Therefore, for any node up to  $d-1$  hops away from the new node, only the appearance of the new node will be added into the existing filters. Further, since the local information is duplicated to every layer of the ABF before it is sent out, there will not be a loop between neighbors to add the information layer by layer. After the initial broadcast, every node, including the new one, will only update once. The expected number of updates can be quantified as the total number of nodes within range plus one:

$$E[N_{update-a}] = E[Totalnumofnodes_{d-1}] + 1 = 2 + n\pi(r \cdot (d-1))^2 \quad (14)$$

We observe the traffic load generated by one node appearing in networks with various densities. We expected fewer broadcasts to be generated in the simulation than in the analytical study. Because the node density is not infinitely high, there are fewer nodes in reach than we expected in the analytical study. However, it turns out that the load is quite accurately predicted by the analytical model. This is because in a very high density network as we assumed, one nodes' appearance will not generate new shortest paths between any pair of nodes. In the lower density networks we simulated, quite some nodes update more than once, because extra new indirect neighbors are discovered due to the appearance of one node. Interestingly, as shown in Fig 4, we found that the number of extra updates compensates the number of extra nodes we estimated in the analytical study. Of course, the higher the network density is, the more accurately the  $i$ -hop node de-

gree is approximated, and fewer new neighbors are discovered due to one node appearance.

### 4.3 Packet loss

In this section, we study the situation when some packets of a node get lost, due to unfavorable propagation conditions. If at least two keep-alive messages from a node are lost consecutively, the other neighbors of the node consider the node disappeared. They start cleaning up their ABFs as we described in Section 4.1. After some time, the transmission quality of the node gets better, and keep-alive messages will be received again. The neighbors think there is a new node appearing in the network. The actions as addressed Section 4.2 will be taken. Here, we assume that the packet loss only occurs in one direction, i.e., we assume the node can still receive packets from its neighbors. Therefore, it keeps updated information of the neighbors. This is a slight difference from the scenario in Section 4.2. In Section 4.2, the appearing node does not have any knowledge about the network. Therefore, in here, one update less is generated than in section 4.2. The number of updates generated in this scenario,  $N_{packet\_loss}$ , can be obtained as:

$$N_{packet\_loss} = N_{update-disa} + N_{updates-a} - 1 \quad (15)$$

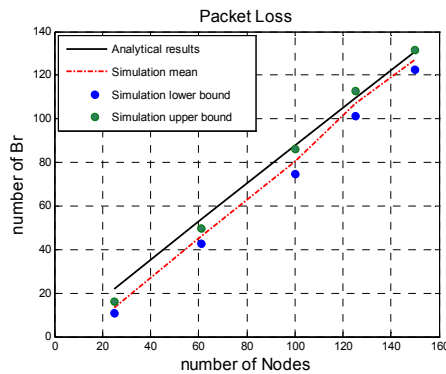


Fig. 5. Effect of Packet loss

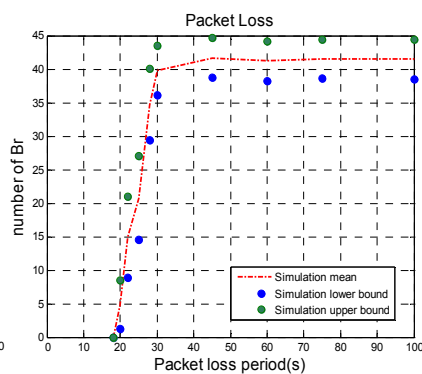


Fig. 6. Effect of Packet loss periods

We compared this with simulation results in networks of various densities, as shown in Fig.6. The keep-alive period, i.e. the time between two consecutive keep-alive messages is distributed uniformly in the interval [15, 17] seconds. Packet loss period is 45 sec, which guarantees at least two continuous keep-alive messages are lost. Simulation results are slightly lower than analytical results. All the reasons we mentioned above in Section 4.1 and 4.2 influence the results of this experiment as well. We also did an experiment to study the effect of different packet loss periods, which is shown in Fig.6. We use the 61-node scenario. We vary the packet loss period from 18sec to 100 sec. When there is only one packet lost, there is no update needed in the network, since nodes are only considered disappeared if no keep-alive messages have been received for two keep-alive

periods. When there are at least two packets lost consecutively, updates are generated. When reappearance period is between 20 and 45 sec, some of the nodes notice the node disappearance, but the node reappears before its information has been removed from the network totally. The longer the period is, the more updates can be done before the reappearance of the node. Therefore, the number of updates is growing in this period. After 45 sec period, the number of packets generated is almost constant. This is because the nodes have enough time to complete the updates for disappearance within this time period.

#### ***4.4 Summary and discussion***

We have derived analytical expressions for the number of additional broadcasts in an ad-hoc network using ABF-based context discovery. These expressions approximate the expected number of additional broadcasts, in case the network density is sufficiently high. From the comparisons above, we observed that analytical formulas are indeed more accurate for higher density network. This fits our hypothesis. The higher the network density is, the more accurate our approximation of the  $i$ -hop node degree. Further, since the proposed protocol automatically duplicates the Bloom filter representing its own context sources to all lower layers of the ABF, the appearance of a new node can be handled in a single pass of advertisement. No advertisements have to go up and down to propagate the availability of indirect paths to the new node into the lower layers of the ABF. However, in the case of removal of a node, multiple passes are needed to remove its representation completely from all ABFs. Removal of context sources has to be done layer by layer, as the equivalent of duplication cannot be performed. Therefore, adding a node generates less traffic than removing a node.

An important issue to improve the performance of ABF-based ad-hoc networks in dynamic environments is to reduce the traffic while removing context sources. One of the possible solutions to improve the protocol is to be more conservative when adding information regarding new context sources to the ABFs. We could add certain policies to restrict adding new context sources, based on the quality level of the source, such as stability, bandwidth, and distance, etc. Only “good quality” and “valuable” information will be added into ABFs. By restricting incoming information, we can reduce the traffic for removing context sources that are most probably not used during their presence. In section 4.2, we have studied the update traffic caused by node appearance without outdated information which needs to be cleaned up. However, in reality this is not always the case. Nodes might appear in an environment where its advertised services may still be present somewhere in some of the ABFs, especially in the case when a node is moving. Based on our study, we found it is mathematically unachievable to quantify the update traffic load caused by this type of clean up. We can consider a moving node as a more complex action, which extends the actions of nodes appearing and disappearing.

## 5 Conclusions and Further Work

In this paper, we have extended the performance analysis of ABF-based ad-hoc networks to a dynamic environment where nodes appear and leave the network, or are temporarily unreachable due to poor propagation conditions. We have used random geometric graphs to model our network topology. In order to be able to approximate the  $i$ -hop node degree accurately, we assume that the network density is very high. The study has been done analytically to quantify the update traffic caused by a node appearing, disappearing, or being temporarily unreachable. We verify the analytical results with simulations. The analytical formulas give more accurate results when the network density is higher. We discovered that it is easier to add context information than to remove it. Especially, in the case when context information moves out of the range of some nodes but still can be reached by other nodes, there are many dynamic parameters, such as node positions and network topology, needed to compute the exact network traffic. This part of work cannot be done analytically.

In the experiments, we observed that there is much less traffic generated by adding context information than removing. Therefore, reducing broadcast traffic for removing context sources is an important topic of further study.

## References

1. T. Albero, V. Sempere, J. Mataix, "A Study of Mobility and Research in Ad Hoc Networks Using Stochastic Activity Networks", in *Proc. of the 2<sup>nd</sup> Conference on Next Generation Internet Design and Engineering, 2006, NGI '06*, Valencia, Spain, April 2006.
2. C. Bettstetter, "On the Minimum node Degree and Connectivity of a Wireless Multihop Networks", in *Proc. of the 3<sup>rd</sup> ACM International Symposium on Mobile Ad Hoc Networking and Computing, MOBIHOC'02*, EPF Lausanne, Switzerland, June 2002.
3. B. H. Bloom, "Space/Time Trade-offs in Hash Coding with Allowable Errors", *Communications of the ACM* 13(7): 422-426.
4. B. Clark, C. Colbourn, D. Johnson, "Unit Disk Graphs", *Discrete Mathematics*, vol. 86, no. 1-3, pp. 165-177, December 1990.
5. P.T.H. Goering, G.J. Heijenk, B. Haverkort, R. Haarman, "Effect of Mobility on Local Service Discovery in Ad-Hoc", in *Proc. of Performance Engineering Workshop, 2007*, Berlin, Germany, September, 2007.
6. R. Hekmat, P. Van Mieghem, "Degree Distribution and Hopcount in Wireless Ad-hoc Networks", in *Proc. of the 11<sup>th</sup> IEEE ICON2003*, Sydney, Australia, September 2003.
7. V. Lenders, J. Wagner, M. May, "Analyzing the impact of mobility in Ad Hoc Networks", in *Proc. of the 2nd international workshop on Multi-hop ad hoc networks: from theory to reality 2006, REALMAN2006*, Florence, Italy, May, 2006.
8. F. Liu, G. Heijenk, "Context Discovery Using Attenuated Bloom filters in Ad-hoc Networks", in *Journal of Internet Engineering*, 2007, Vol. 1, No. 1, pp. 49-58.
9. F. Liu, P. Goering, G. Heijenk, "Modeling Service Discovery in Ad-hoc Networks", in *Proc. of the 4th ACM International Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks, PE-WASUN 2007*, Chania, Crete Island, October 2007.
10. M. Penrose, "Random Geometric Graphs", *Oxford University Press Inc.*, 2003.
11. D. Trajanov, S. Filiposka, M. Efnuseva, A. Grnarov, "Ad Hoc Networks Connection Availability Modeling", in *Proc. Of the 1st ACM international workshop on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks, PE WASUN'04*, Venice, 2004.