Running head: THE IMPACT OF NETWORK DYNAMICS ON A DISCOVERY PROTOCOL

On the impact of network dynamics on a discovery protocol for ad-hoc networks

Fei Liu and Geert Heijenk

University of Twente, P.O. Box 217,

7500 AE Enschede, The Netherlands

{fei.liu, geert.heijenk}@utwente.nl[1]

ABSTRACT

A very promising approach to discovering services and context information in ad-hoc networks is based on the use of Attenuated Bloom filters. In this paper we analyze the impact of changes in the connectivity of an ad-hoc network on this approach. We evaluate the performance of the discovery protocol while nodes appear, disappear, and move, through analytical and simulative analysis. The analytical results are shown to be accurate when node density is high. We show that an almost linear relation exists between the density of the network and the number of update messages to be exchanged. Further, in case of nodes moving, the number of messages exchanged does not increase with the speed of movement.

KEYWORDS

mobility, context discovery, ad-hoc networks, Bloom filters, performance evaluation

INTRODCTION

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. Such a protocol disseminates information on context information that can be provided by nodes to nodes that might want to use the information. 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) (Liu & Heijenk, 2007). We have proven that using ABFs our discovery protocol can provide discovery, while exchanging far less information than conventional approaches.

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 not feasible to quantify the network traffic of our discovery protocol in a mobile environment mathematically. Thus, simulation is a good approach to study this problem. (Goering, Heijenk, Haverkort, & Haarman, 2007) has examined the network traffic generated by updating the ABFs while nodes are moving in a low density network, and the reachability 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. Finally, we study the effect of node movement on network traffic for various network densities using simulations.

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 discusses network structures, 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, when a series of consecutive advertisement packets are lost, or when nodes are moving. Section 5 concludes the study and discusses the future work.

A DISCOVERY PROTOCOL FOR AD-HOC NETWORKS

Attenuated Bloom filters (ABF)

Bloom filters (Bloom, 1970) have been proposed in the 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 (Liu & Heijenk, 2007). The *i*th layer of an ABF ($0 \le i < d-1$) 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 layer *i* will often be repeated in lower layer *j* (*j*>*i*).

Figure 1 exemplifies the context aggregation operation for a node with two neighbors. In this example, each node has an ABF with 8 bits width (*w*=8) and a 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 Figure 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. Thus, *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.

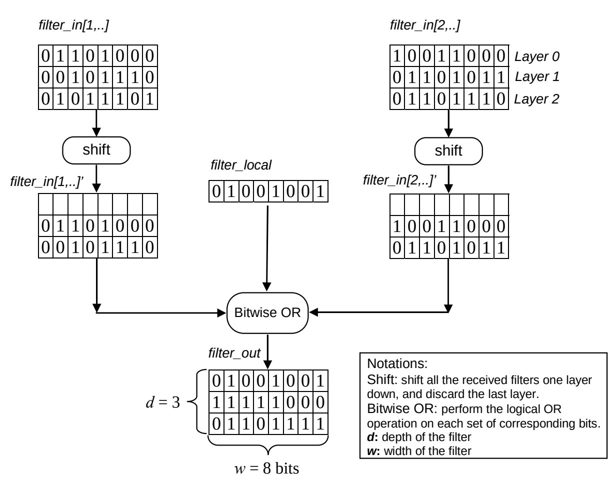

Figure 1. An example of ABF aggregation

Protocol Specification

Our ABF-based context discovery protocol distinguishes 3 phases (Liu, Goering, & Heijenk, 2007): 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. This broadcast will be received by the nodes within communication range. Any neighboring 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 node is looking for a specific context source, and a query is generated in that 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 unicasted 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. State information will be temporarily stored to be able to route responses back to the query initiator.

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-id*s. 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 period*s, it considers the node has left and removes its neighbor's information.

MODELING PRELIMINARIES

In this section, we introduce background knowledge regarding modeling of connectivity in multi-hop ad-hoc networks. We start with our modeling assumptions regarding the network structures. Further, we introduce 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.

Network Structures

To be able to analyze the performance of the proposed system, we model two typical network structures, which we will refer to as grid structure and circle structure. The grid structure represents a simple and regular network in which each node has four neighbors. In the circle structure placement of nodes and connectivity of nodes is based on the notion of a random geometric graph.

Grid Structure. In grid structures, each node has 4 direct neighbors within communication range. The distance of each pair of connected nodes equals to the communication range of the nodes r. The connections between nodes can be considered as series of intersecting vertical and horizontal axes that form a two-dimensional grid structure with $r \times r$ size square-shaped grids. The structure is shown in Figure 2a. The regularity of the grid structure and the fixed number of neighbors for each node enables a rather simple analysis of models using this structure.

Circle Structure. To model the network structure of an ad-hoc network, based on random location of nodes, we can model it as a random geometric graph. Given a graph $G=G(V,E)$, where nodes are vertices (V), while links between nodes are edges (E), a unit disk graph (Clark, Colbourn, & Johnson, 1990) models an ideal network where the two-dimensional radio coverage of a mobile node is a circle (with range r). This can be extended to a random geometric graph, where vertices (nodes) are located at random, uniformly and independently in a region, and an edge between two vertices exists if and only if the distance between them is at most r. Random geometric graphs are often used to model ad-hoc networks (Penrose, 2003), and we will use them as the basis for our analysis.

In Section 3.4, we will argue that, from the point of view of a selected node, the set of areas in which nodes at i hops distance can be located can be approximated by a set of concentric circles, one for each value of i. For this reason, we will refer to this model as circle structure, shown in Figure 2b.

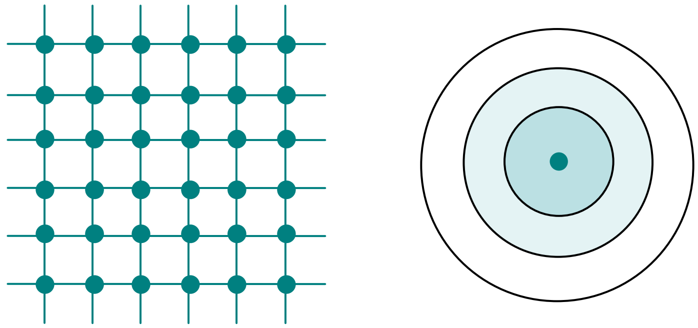

Figure 2. Network structure (a) grid structure; (b) circle structure.

Connectivity in Ad-hoc Network Models

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. The degree of a vertex can be defined as the number of edges incident to it. Further, a graph is called *k*-connected if the graph remains connected when fewer than *k* vertices are removed from the graph. (Clark, Colbourn, & Johnson, 1990) has discussed the connectivity problems of unit disk graph. (Bettstetter, 2002) 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. (Hekmat & Mieghem, 2003) 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. (Albero, Sempere, & Mataix, 2006) 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 (Trajanov, Filiposka, Efnuseva, & Grnarov, 2004) and test bed (Lenders, Wagner, & May, 2006).

We can generalize that 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. 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 a selected node can reach in exactly *i* hops, but not fewer than that. Let us denote the *i*-hop node degree as $\underline{D_i}$.

For grid-structured network, it is straightforward to determine the *i*-hop node degree, D_i^g. By definition, we define the node degree at 0 hops, i.e. the number of nodes 0 hops away as

$$D_0^g = 1. \qquad (1)$$

Here the superscript *g* denotes the grid structure. For larger *i*, it can easily be seen from Figure 2a. that 4*i* nodes become reachable when increasing the maximum number of hops from the central node from *i-1* to *i*. The node degree of *i*th hop, D_i^g, can be derived as

$$D_i^g = 4 \cdot i. \qquad (2)$$

This value increases linearly with the distance (the number of hops). The total number of nodes reachable in at most *i* hops, N_i^g, including the central node can be easily found by summation:

$$N_i^g = \sum_{j=0}^{i} D_j^g = 1 + 2i(i+1). \qquad (3)$$

For a random geometric graph, the distribution of D_i^c can be derived from D_{i-1}^c (the superscript *c* denotes the circular structure), conditioned on the position and the number of nodes reachable in *i-1* hops. Theoretically, we can derive D_i^c in this way. However, the expression is going to be computationally infeasible. Therefore, we will take a step back and observe the upper bound of this problem.

Modeling Assumptions of Circular Structured Networks

As we mentioned above, we model our network based on random geometric graph, where nodes are uniformly and independently 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 N, 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 (Bettstetter, 2002):

$$r_0 \geq \sqrt{\frac{-\ln\left(1-p_c{}^{1/N}\right)}{n\pi}}\,. \tag{4}$$

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 assume for each node a fixed communication range r. To simplify our analysis, and to achieve with high probability a network without isolated nodes, we assume a very high-density network.

i-hop Communication Range in Circular Structured Networks

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 Figure 3a. 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. 3a. Because the distance between A and B is between r and $2r$, the intersection of circles 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 and independently 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}:

$$P(B \text{ is a 2-hop neighbor of } A \mid B \text{ is in } \mathrm{R}_2) = P(N_{AB} > 0) = 1 - P(N_{AB} = 0). \qquad (5)$$

Since the number of nodes is Poisson distributed, Eq. 5 can be rewritten as:

$$P(N_{AB} > 0) = 1 - e^{-\lambda_{AB}} = 1 - e^{-S_{AB} \cdot n} \qquad \left(0 < S_{AB} < \frac{2}{3}\pi r^2 - \frac{\sqrt{3}}{2}r^2\right). \qquad (6)$$

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:

$$\lim_{n\to\infty} P(B \text{ is a 2-hop neighbor of } A \mid B \text{ is in } \mathrm{R}_2) \to 1. \qquad (7)$$

Let us now have a look at node F in Figure 3b, 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:

$$\begin{aligned} &P(F \text{ is a 3-hop neighbor of } A \mid F \text{ is in } \mathrm{R}_3) \\ &= P(\exists E : d(E,F) \le r \wedge E \text{ is a 2-hop neighbor of } A \mid F \text{ is in } \mathrm{R}_3) \end{aligned} \qquad (8)$$

From Figure 3b, 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 Eq. 7, we can obtain:

$$\begin{aligned} &\lim_{n\to\infty} P(F \text{ is a 3-hop neighbor of } A \mid F \text{ is in } \mathrm{R}_3) \\ &\to \lim_{n\to\infty} P(\exists E : d(E,F) \le r \wedge E \text{ is in } \mathrm{R}_2 \mid F \text{ is in } \mathrm{R}_3) \end{aligned} \tag{9}$$

Since n goes to infinity, we have:

$$\lim_{n\to\infty} P(\exists E : d(E,F) \le r \wedge E \text{ is in } \mathrm{R}_2 \mid F \text{ is in } \mathrm{R}_3) \to 1. \tag{10}$$

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:

$$\begin{aligned} &\lim_{n\to\infty} P(X \text{ is a i-hop neighbor of } A \mid X \text{ is in } \mathrm{R}_i) \\ &\to \lim_{n\to\infty} P(\exists Y : d(Y,X) \le r \wedge Y \text{ is (i-1)-hop neighbor of } A \mid X \text{ is in } \mathrm{R}_i) \\ &\to \lim_{n\to\infty} P(\exists Y : d(Y,X) \le r \wedge Y \text{ is in } \mathrm{R}_{i-1} \mid X \text{ is in } \mathrm{R}_i) \to 1 \end{aligned} \tag{11}$$

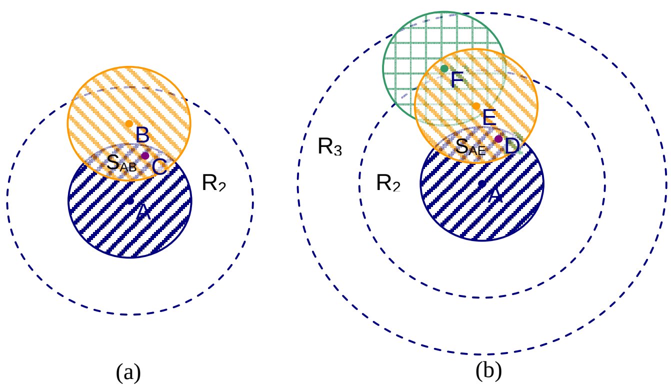

Figure 3. (a) A and B are connected through C; (b) A and F are connected through E and D.

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.

Eq. 11 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 that the i-hop communication range of node A goes to ir in a high-density network, which can be represented as:

$$\lim_{n\to\infty}\left(i\text{ - hop communcation range of } A\right)\to ir\ . \tag{12}$$

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.

Mean Multi-hop Node Degree of Circular Structured Networks

For a circular structured network, we denote the number of nodes that can be reached within i hops, but not fewer than i hops, as random variable D_i^c. Given Eq. 11, 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 D_i^c can be written as:

$$\lim_{n\to\infty} E\left[D_i^c\right]=\begin{cases} 1 & i=0 \\ n\pi r^2\left(i^2-(i-1)^2\right)=(2i-1)n\pi r^2 & (i>0)\end{cases}. \tag{13}$$

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

$$\lim_{n\to\infty} E\left[N_i^c\right]=\lim_{n\to\infty} E\left[\sum_{j=0}^{i} D_j^c\right]=\lim_{n\to\infty}\sum_{j=0}^{i} E\left[D_j^c\right]=1+n\pi r^2 i^2. \tag{14}$$

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. For grid structures, we study the effect of a single node disappearing, appearing, and moving across the network, analytically. We study the same phenomena for circular structures, and add also the case where packets from a specific node are lost for a period of time. We quantify the network load generated due to the dynamic connectivity through analytical study and verify obtained results with simulations. Nodes disappearing and (re-)appearing can be caused by a (temporary) lack of energy supply, e.g., in case the needed energy has to be harvested from the environment. 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 node's packet will be received again. As a result, it reappears in the network. When a node is moving across the networks, the set of neighbors it can reach with its broadcasts is changing continuously.

In this section, we will first present the analysis for the grid structure. Next, in Section 4.2 and 4.3, we will present the analysis, and both simulation and analytical results for the grid and circular structure, respectively. We conclude the section with a discussion of our findings in Section 4.4.

Grid Structure

Node Disappearance. Besides insufficient battery supply, a node might disappear from the current network, due to various other reasons, such as un-functional antenna, system crash, personal opinions like switching off the mobile or leaving the network.

In the current protocol, the disappearance of a neighbor is noticed if no keep-alive message has been received from this node for two consecutive keep-alive periods. The neighbors of this absent node will generate a new ABF, removing the hash results relating only to the absent node from layer 1, and broadcast it. Due to the change, all nodes that receive the updated filter will also regenerate a filter and broadcast it. This process will continue till *(d-1)* hops away from the absent node. Note that there is always more than one path from any node within range to the disappearing node due to the special structure of grid network. Especially, any node within range can reach the disappearing node in every two additional hops starting from the shortest path length. Therefore, nodes need to clean up every second layer after the first time of cleaning up one by one.

When removing the context sources of some 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. Liu and Heijenk (2007) have already defined and derived this probability. We use $P_{fp,i}$ to represent the false positive probability of layer *i*. Liu and Heijenk (2007) has proved that

$$P_{fp,i} \approx \left(1 - e^{\frac{-bx_i}{w}}\right)^b \tag{15}$$

where x_i denotes the total number of represented context sources in layer *i*. We use x_i^g for the grid structures and x_i^c for the circular structures. We can obtain that $x_i^g = s \cdot N_i^g$ and $x_i^c = s \cdot \left(1 + n\pi r^2 i^2\right)$. $P_{fp,i}$ 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, the number of updates for different maximum hop count *d* can be represented as follows:

$$N^{g}_{update\_ndisa}(d) = \sum_{i=1}^{d-1}(1 - P^{s}_{fp,i}) \sum_{j=0}^{\left\lfloor \frac{i-1}{2} \right\rfloor} D^{g}_{i-2j} \ . \qquad (16)$$

In this equation, for a certain value of i, we count all the transmissions, done i nodes away from the node that disappeared, and also the transmissions done at *i-2*, *i-4*, etc., nodes away, because of alternative longer paths to the node that disappeared. The number of transmissions depends on the number of nodes present at the relevant distance, and weighed by the probability that there is no false positive. This is done for every possible value of i.

Node Appearance. In this subsection, we continue with the analysis of a node appearing in the network. In here, we assume that every node, including the new one which just appears, knows about the format and hash functions of attenuated Bloom filters used in the network. Further, the new node does not have any information about the neighbors. In here, we will refer to the appearing node as the new node. The new node broadcasts keep-alive message periodically. The direct neighbors who receive a keep-alive message from an unknown neighbor initialize a link with this new neighbor and send an update-request back to the new node. As the answers to the update-request, the new node will broadcast its filter (of size $w \times d$ 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 broadcast it. The direct neighbors update their filters after receiving the filter, and broadcast them. The other nodes within range will also update their filters and broadcast them.

Please be aware that because of the duplication (see Section 2.2), every node in range only needs to update once. The network traffic depends on how many neighbors the appearing node has, which is directly related to the location of the new node appearing. In a grid structure, we can divide the space of a grid into 3 different areas as shown in Figure 4, based on the number of surrounding neighbors that have the area within their transmission range. Area 1, can be reached by all 4 surrounding nodes of the grid, area 2 can be reached by 3 nodes, and area 3 can only be reached by 2 nodes.

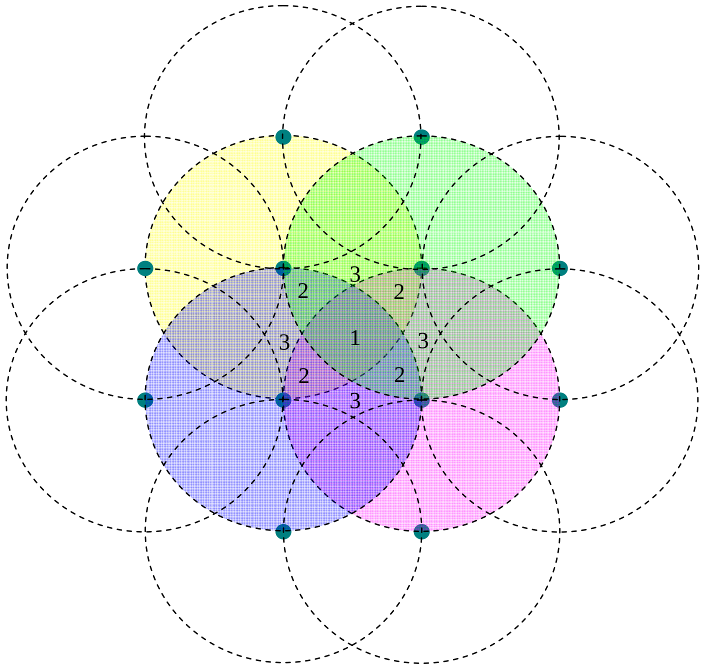

Figure 4. Node appears in a grid network.

Therefore, the network traffic can be obtained as the sum of the number of updates $N^{g}_{updates\_na,i}$ when the new node is appearing in area i ($i \in \{1,2,3\}$) with the probability p_i that the node appears there. We have:

$$N^{g}_{update\_na} = N^{g}_{update\_na,1} \times p_1 + N^{g}_{update\_na,2} \times p_2 + N^{g}_{update\_na,3} \times p_3 \,. \quad (17)$$

When the node appears in the area 1, it has 4 direct neighbors. All the nodes within range need to update their filters. We can obtain the number of updates with the probability that no changes are need to certain layers in ABF in this situation as:

$$N^{g}_{udpates\_na,1} = \sum_{i=0}^{d-1} D^{g}_{i} \left(1 - P^{s}_{fp,i}\right). \quad (18)$$

Similarly, we can obtain the number of updates when the node appears in the area 2 as:

$$N^{g}_{updates\_na,2} = 1 + \sum_{i=1}^{d-1} \left(D_i^g - 1\right)\left(1 - P_{fp,i}^{s}\right). \tag{19}$$

The number of updates when the node appears in the area 3 can be obtained as:

$$N^{g}_{updates\_na,3} = 1 + \sum_{i=1}^{d-1} \left(D_i^g - 2\right)\left(1 - P_{fp,i}^{s}\right) \tag{20}$$

The probabilities of a node appearing in area 1, 2, and 3 equal the ratio of these three specific areas over the entire grid. They can be denoted as:

$$\begin{aligned} p_1 &= \left(\frac{\pi}{12} - \frac{1}{4}\right) \times 4 + \left(2 \times \sin\frac{\pi}{12}\right)^2 \approx 0.3151 \\ p_2 &= \left(\left(\frac{\pi}{4} - \frac{1}{2}\right) \times 2 - p_1\right) \times 2 = \frac{\pi}{3} - 8\sin^2\frac{\pi}{12} \approx 0.5113 . \\ p_3 &= 1 - p_1 - p_2 = 2 - \frac{2\pi}{3} + 4\sin^2\frac{\pi}{12} \approx 0.1736 \end{aligned} \tag{21}$$

One Node Moving. Now we study the scenario that a node is moving horizontally through the network, crossing one of the rows of nodes. Theoretically, when the mobile node is in the same location as one of the nodes in the row, it will establish three new direct connections to the nodes located right on top of, under, and next to it along the direction of movement. However, we assume the side of the grid is exactly 300 meters, which is also the maximum transmission range of a node. Therefore, when the mobile node is in the same location as one of the nodes along its trace, it is the only position where the mobile node is in reach of the nodes above and under it vertically. Theoretically, the mobile node only spends 0 second staying in that exact position. It is not possible for the mobile node to establish a direct connection to the nodes right above and below it in the grid within 0 second. Therefore, we assume only one new direct connection will be established. That is the link between the mobile node and the next node right along the moving direction.

This new connection results in the nodes in orange in Figure 5a to update (for d = 3). Similarly, when the mobile node passes the point mentioned above, it loses the direct connection with the node behind in the direction of moving. This also results in the updates of those orange nodes in Figure 5b. Each orange node needs to update once, because the alternative paths of longer length are already/still available, and represented in the Bloom filters. In both of the case, the number of the updates is the same, and is given by:

$$N^{g}_{udpates\_hori} = \frac{(1+(2d-1))\cdot d}{2} = d^2 . \qquad (22)$$

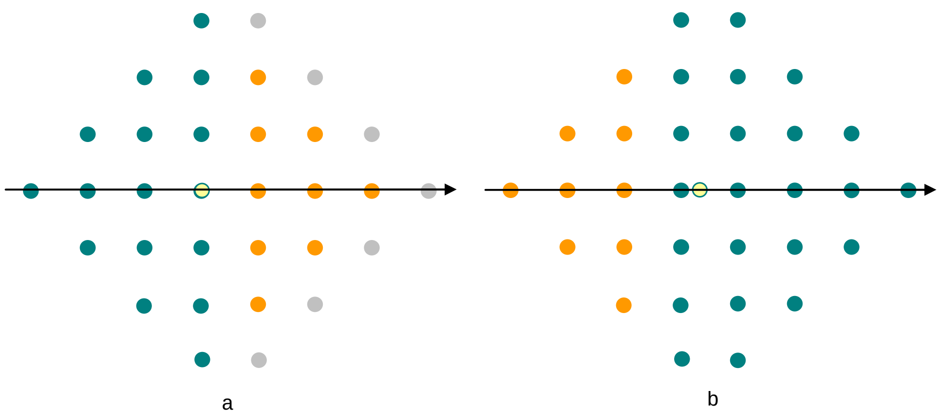

Figure 5. Horizontal move

Circular Structure

Simulation Setup. Besides the approximate analysis, based on the assumptions stated in Section 3, the circular structured model has also been implemented in the discrete event simulator OPNET Modeler version 11.5 ("OPNET Modeler", n.d.). We observe the node density influence on the traffic load in first three experiments below. We place 25, 61, 100, 125, 150 nodes randomly into a 1700×1700m^2 area with 300 meters communication range, r, for every node. Note that the 61 nodes scenario generates networks that are 1-connected graphs with 90% probability (see Eq. 4). 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 that 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.

Node Disappearance. We start the analysis again 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 A. 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 in a similar way.

It is a very complex procedure to clean up the services. Every node *i* hops away has to clean up *(d-i)* layers in total. It will clean up layer *j* $(i \leq j \leq d-1)$ once it realizes 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. Similar to the analysis of the grid structure, we have to take into account the probability that no changes are needed to a certain layer in the ABF, because the relevant bits have to remain set, because of context sources of other nodes. We have defined the false positive probability in Eq. 15. Further, we have defined for the circular structure $x_i^c = s \cdot \left(1 + n\pi r^2 i^2\right)$.

Therefore, we can derive the expected number of clean-up updates while one node disappears, $E[N^{c}_{updates-disa}]$, as:

$$E[N^{c}_{updates\text{-}disa}] \approx \sum_{i=1}^{d-2}\left(\left(E[D_i^c]-1\right)\cdot \sum_{j=i}^{d-1}\left(1-P^{s}_{fp,j}\right) + \sum_{j=i+1}^{d-1}\left(1-P^{s}_{fp,j}\right)\right) + E[D^{c}_{d-1}]\cdot\left(1-P^{s}_{fp,d-1}\right). \qquad (23)$$

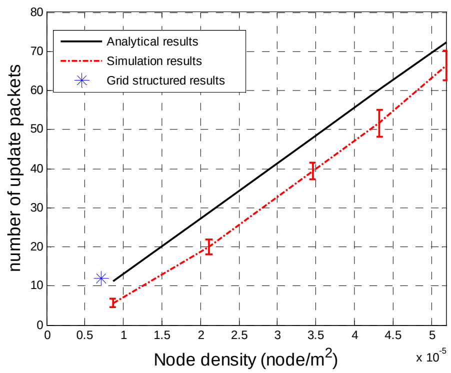

Figure 6. Node disappears

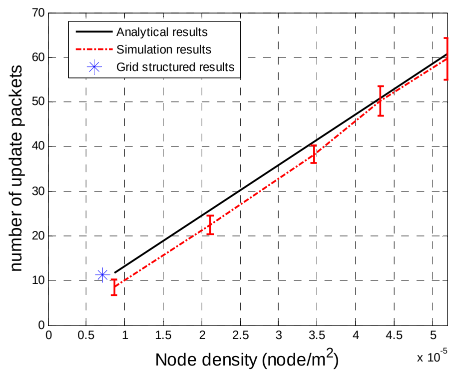

Figure 7. Node appears

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 Figure 6. 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 overestimated. 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 Eq. 23. 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.

We obtained the number of updated packets for the grid structure from Eq. 16, with the related network density estimated as in (Liu & Heijenk, 2007). The result is 12.0 updates transmitted, at a node density of 0.7×10^{-5}. This has also been plotted in Figure 6, and matches the analytical circular structured results.

Node Appearance. In this section, we again 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 the 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. Similarly, with the probability that no changes are needed to certain layers in the ABF, the expected number of updates can be quantified as the total number of nodes within range plus one:

$$E\left[N^{c}_{update-a}\right]=\sum_{i=0}^{d-1}E\left[D^{c}_{i}\right]\cdot\left(1-P^{s}_{fp,i}\right)+1 \qquad (24)$$

We observe the traffic load generated by one node appearing in networks with various densities. 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 Figure 7, 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 degree is approximated, and fewer new neighbors are discovered due to one node appearance.

The corresponding results of the cost for the grid structure, as obtained from Eq. 17, with again a node density of 0.7×10^{-5} has been plotted in Figure 7. This result is also very close to the analytical results for circular structured network.

Packet Loss. In this section, we study the situation where some packets of a node get lost, e.g., 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 4.2.1. 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 in Section 4.2.2 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 4.2.2. In Section 4.2.2, the appearing node does not have any knowledge about the network. Therefore, in here, one update less is generated than in section 4.2.2. The number of updates generated in this scenario, $N^c_{packet\_loss}$, can be obtained as:

$$N^c_{packet\_loss} = N^c_{update-disa} + N^c_{updates-a} - 1 \qquad (25)$$

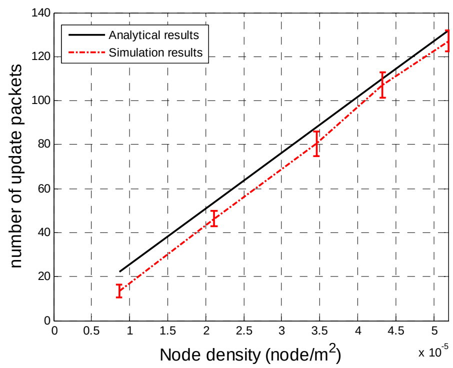

Figure 8. Effect of Packet loss

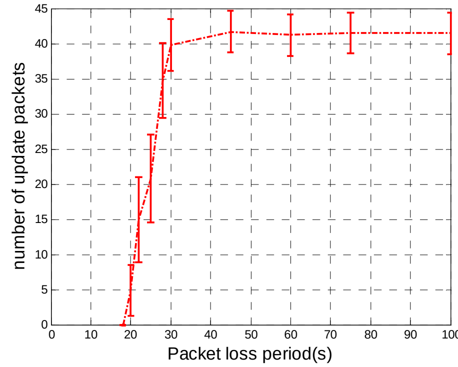

Figure 9. Effect of Packet loss periods

We compared this with simulation results in networks of various densities, as shown in Figure 8. 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 again slightly lower than analytical results. All the reasons we mentioned above in Section 4.2.1 and 4.2.2 apply to 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 Figure 9. We use the 61-node scenario. We vary the packet loss period from 18 sec 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. If the packet loss period is larger than 45 sec, 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.

A Moving Node. A moving node is a more complex action, which extends beyond the actions of nodes appearing and disappearing. In this section, we continue our study of the extra updated traffic generated by a moving node via simulations. The node is moving from one spot to another in a straight line with steady speed. The start and end point of the journey are far enough that two nodes located in both positions do not share any neighbour within *d* hops. This guarantees that the nodes in range (*d* hops) of the moving node need to update their filters at last. This offers the chance for us to compare the simulation results with analytical results in the extreme case of a node disappearing in one location, and reappearing in a different location. The simulation has been done in a two-dimensional area of $4200\times1800\text{m}^2$. We assume *d* is equal to 3. The mobile node with 300 meters communication range is set to move 2400 meters from point (900m, 900m) to (3300m, 900m). That guarantees the moving node to avoid border effects. The additional update traffic is highly related to the speed of the node and the network density with fixed keep-alive period. Therefore, we first fix the network density and observe the traffic load under different speed. Afterwards, we observe the extreme case in which the mobile node is moving with extremely fast speed, so that the other nodes along the path will not have time to notice the movement. The network behaviour resembles the case where the mobile node disappears from one point and reappears at the other point far away from the previous position. We compare the simulation results with our analytical results for different network density.

First of all, the simulations have been done with two different densities 8.07×10^{-6} (61 nodes in the experiment area) and 1.98×10^{-5} node/m^2 (150 nodes in the experiment area). The mobile node is moving with different speeds from 0.1m/s to 20m/s. 20m/s can be considered the average speed of a car; 5m/s as the average speed of a bicycle; 1m/s as the average speed of a walking adult. The results are shown in the Fig.10. The traffic load decreases as the speed increases. This is because the faster the node is moving, the more nodes it misses when updating information. Therefore, fewer updates will be generated. Further, the higher the network density, the more nodes are involved in the update.

Finally, we simulate the extreme case with a very high speed of 24000 m/s. Ideally, at this speed, the nodes along the path will not have time to notice the move. This scenario can be considered as the node disappears from one position and reappears at another. The effect of the disappearance of the node is exactly the same as discussed in Section 4.2.1. However, the reappearance is slightly different from Section 4.2.2. In Section 4.2.2, we assumed that the node appears in a new environment without any previous information stored in ABF. In here, the node appears in a new environment with an ABF filled with the information of the previous position. Therefore, the updates will take place twice. When any node notices the existence of the mobile node, the first round of updates will happen to add information still present in the ABF of the mobile node. After two continuous keep-alive periods, the mobile node notices the loss of the connection to the neighbors at the previous position; it will clean up those neighbors from the filter. The second round of updates will take place. Therefore, we can generalize the total number of updates due to the moving node, $N^c_{update-move}$, as:

$$N^c_{update-move} = N^c_{update-disa} + 2 \cdot N^c_{update-a} \,. \qquad (26)$$

We compare the simulation results with the analytical results for different network densities: 8.07×10^{-6}, 1.98×10^{-5}, 2.38×10^{-5}, 3.17×10^{-5}, and 3.97×10^{-5} nodes/m^2. The results are shown in Figure 11. We can see that the higher the network density, the more updates are generated. When the network density is 8.07×10^{-6}, the analytical results are higher than the confidence interval of simulation results. This is because this is a low density scenario, there are much less nodes connected as we expected in the model. Therefore, the simulation generates less updates as we expected. As we can see, higher the network density, more accurate is our analytical analysis. In high density scenarios of 2.38×10^{-5}, 3.17×10^{-5}, and 3.97×10^{-5} node/m^2, the analytical results are always within the confidence interval of the simulation results.

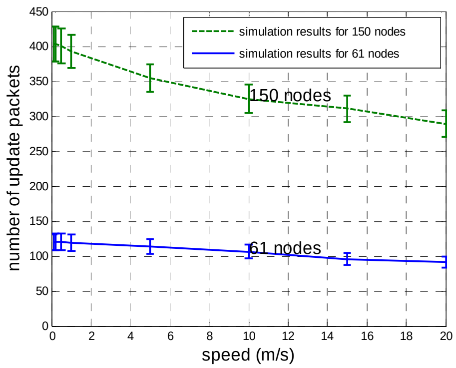

Figure 10. One node moving at different speeds

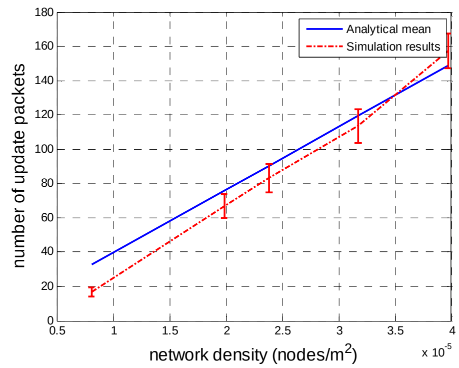

Figure 11. One node moving for various densities

Summary and Discussion

We have derived analytical expressions for the number of additional broadcasts in an ad-hoc network using ABF-based context discovery. We have done this for two different network models, a grid structure, and a circular structure, based on the assumption of a random geometric graph. For the grid structure, we found exact expressions for the expected number of update messages caused by the disappearance and appearance of a node, and by the movement of a node along a straight, horizontal line. For the circular structure, the expressions found approximate the expected number of additional broadcasts, in case the network density is sufficiently high. From the comparisons above, we observed that the analysis is indeed more quite accurate for high-density networks. 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 advertisements. 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 4.2.2, we have studied the update traffic caused by node appearance without outdated information that 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 infeasible to quantify the update traffic load caused by this type of clean up.

We studied a simple mobile case of one node moving in a straight line in the network. For increasing speed of the moving node, less traffic is generated. However, network density has more influence on the number of updates generated than the speed of the mobile node. In the extreme case of node movement at very high speed, the higher the network densities, the better our analytical results fit the simulation results.

CONCLUSIONS AND FUTURE 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, are temporarily unreachable due to poor propagation conditions, or move through the network. The analysis has been performed for two different network models. We have found exact expressions for a grid-structured model where nodes are located at the line crossings of a regular grid, whereas the transmission range is such that two neighboring nodes on a line of the grid can just reach each other. The second network model assumes that nodes are located and connected, based on a random geometric graph. In order to be able to approximate the i-hop node degree accurately for this network model, we assume that the network density is very high, which results in a circular structure, consisting of concentric circles to denote the i-hop node degree. The study has been done analytically to quantify the update traffic, measured in the number of broadcast packets with ABFs. For this circular structure, we verified the approximate results with simulations. The analytical expressions 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.

General conclusions with respect to the performance of the ABF-based discovery protocol in a dynamic environment are that network load increases linearly with the node density. Furthermore, the network load decreases slowly with increasing node speed.

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

Albero, T., Sempere, V., & Mataix, J. (2006). A Study of Mobility and Research in Ad Hoc Networks Using Stochastic Activity Networks. *Proc. 2nd Conference on Next Generation Internet Design and Engineering*.

Bettstetter, C. (2002). On the Minimum node Degree and Connectivity of a Wireless Multihop Networks. *Proc. 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing*.

Bloom, B. H. (1970). Space/Time Trade-offs in Hash Coding with Allowable Errors., *Communications of the ACM, 13(7),* 422-426.

Clark, B., Colbourn, C., & Johnson, D. (1990). Unit Disk Graphs. *Discrete Mathematics, 86(1-3)*, 165-177.

Goering, P.T.H., Heijenk, G.J., Haverkort, B., & R. Haarman (2007). Effect of Mobility on Local Service Discovery in Ad-Hoc. *Proc. Performance Engineering Workshop*.

Hekmat, R., & Van Mieghem, P. (2003). Degree Distribution and Hopcount in Wireless Ad-hoc Networks. *Proc. 11th IEEE International Conference on Networks*.

Lenders, V., Wagner, J., & May, M. (2006). Analyzing the impact of mobility in Ad Hoc Networks. *Proc. 2nd international workshop on Multi-hop ad hoc networks: from theory to reality*.

Liu, F., & Heijenk, G. (2007). Context Discovery Using Attenuated Bloom filters in Ad-hoc Networks. *Journal of Internet Engineering, 1(1)*, 49-58.

Liu, F., Goering, P., & Heijenk, G. (2007). Modeling Service Discovery in Ad-hoc Networks. *Proc. 4th ACM International Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks*.

OPNET technologies Inc., *OPNET Modeler Accelerating Networks R&D.* (n.d.). Retrieved 2008, from http://www.opnet.com/products/modeler

Penrose, M. (2003). *Random Geometric Graphs*. New York, USA: Oxford University Press Inc..

Trajanov, D., Filiposka, S., Efnuseva, M., & Grnarov, A. (2004). Ad Hoc Networks Connection Availability Modeling. *Proc. 1st ACM international workshop on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks*.

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