Analysing and Improving Energy Efficiency of Distributed Slotted Aloha

Haidi Yue, Henrik Bohnenkamp, Malte Kampschulte and Joost-Pieter Katoen

Software Modeling & Verification
Dept. of Computer Science 2, RWTH Aachen University, Aachen, Germany
{haidi.yue|henrik|Malte.Kampschulte|katoen}@cs.rwth-aachen.de

Abstract. This paper is concerned with the formal modelling and simulative analysis of an energy-efficient MAC protocol for gossip-based wireless sensor networks. This protocol is a variant of classical slotted Aloha in which the number of active TDMA slots is dynamically changed depending on the number of neighbours of a node. We provide a formal model of this protocol, and analyse energy consumption under the signal-to-interference plus noise ratio (SINR) radio model. We propose an amendment of the distributed slotted Aloha protocol by a simple dynamic power assignment scheme, and show that this significantly reduces the energy consumption (30%) and speeds up the message transmission.

Keywords: WSNs; Aloha, formal modelling; SINR; simulation; energy;

1 Introduction

We consider the setting of gossip-based wireless sensor networks (WSNs), in which battery-powered mobile sensors interact via a wireless communication network. The sensors are extremely simple, cheap to produce, and have limited data and processing capacities. The system lifetime is mainly determined by the battery lifetime, as recharging is typically not possible. To support the reliable communication between mobile sensors a medium access protocol (MAC) is needed. In order to extend the lifetime of a sensor, the protocol must be energy-efficient – it should attempt to idle once in a while, not only to save power, but also in order to profit from the so-called recovery effect [10]. To meet all these requirements, the Dutch company CHESS, experts on developing gossiping WSNs for various applications, has developed a variant of slotted Aloha, called distributed slotted Aloha. This protocol aims to significantly simplify previously proposed energy efficient MAC proposals such as gMAC [15], A-Mac [12] and the TDMA-W [6] protocols.

In contrast to the well-known classical slotted Aloha protocol [9], a node in distributed slotted Aloha has a *dynamic* number of active slots. Each node keeps track of a list of neighbour nodes, and adapts the number of its active slots when its neighbourhood increases or decreases. On the one hand, this is aimed to maximise throughput; on the other hand it aims to adapt the idle period of a frame in order to reduce energy consumption. The protocol is relatively simple to realise, and has moderate memory usage. The parameters of the protocol are tuned to extensive experiments in real networks. The aim of our study is to provide an abstract and easy-to-grasp formal model of the protocol, and analyse its energy-preservation capabilities.

In distributed slotted Aloha, each node has a constant signal power during its lifetime, and this is equal for all nodes. As signal power is a critical parameter in our studies, we use the signal-to-interference plus noise ratio (SINR) radio model as radio propagation model. The SINR model [11] is an intuitive model in which there is a relation between signal strength and distance, and the receipt of a message depends on the strength of interfering transmissions. In the SINR model, a node can decode a message only if the received signal strength divided by the strength of concurrent interfering senders (plus the noise) exceeds a threshold. This is more realistic compared to the unit disk model [15].

We model the distributed slotted Aloha protocol in the MoDeST language [3], a formalism that supports the modular specification of distributed systems in a mathematically rigorous, though user-friendly, manner. The simulation is performed using the Möbius [7] tool-set. The main advantage over the usage of standard simulation packages such as NS2, Opnet or OMNET, is that we obtain semantically sound simulation runs. Together with the fact that we do not model entire protocol stacks but rather abstract from lower layer effects, this avoids many of the credibility problems of standard simulations [5, 1].

An important outcome of our analysis studies is that the constant and identical signal strength may result in a huge interference in densely populated areas, whereas nodes become disconnected in sparse areas. To overcome these unfavourable situations, we propose an amendment of the distributed slotted Aloha protocol in which nodes can adapt their signal power strength dynamically based on the number of neighbours of a node. In sparse areas, the strength is increased, whereas in dense areas, it is reduced. We show that this scheme substantially reduces energy consumption and improves message propagation.

Organisation of the paper. Section 2 introduces the distributed slotted Aloha protocol investigated in this paper. Section 3 explains the SINR radio model. Section 4 describes our modelling idea and the simulation setup. Section 5 contains results and Section 6 concludes.

2 Distributed slotted Aloha

The well-known slotted Aloha protocol [9, 13] has a fixed number of slots, and has the property that its throughput reaches the maximum when a node has the same number of neighbours as the number of the TDMA slots. The distributed slotted Aloha (dsA for short) strategy is an extension of classical slotted Aloha, uses the same principle but has a dynamic behaviour depending on the number of direct neighbours. When the number of neighbours increases (or decreases), a node will increase (or decrease) its number of active slots, to make the number of neighbours and the number of the current TDMA slots more or less equal, and hence attempts to achieves the maximum throughput as close as possible.

In the dsA protocol, a *frame* is the basic unit of time (Figure 1). A frame is subdivided into an active and idle period. Radio communication occurs in the active period and the idle period is considered for energy conservation. The active period is divided in slots of equal length, we call these slots *virtual slots*. The virtual slots are merged into several blocks of an equal number of slots, each block is called a *schedule*. A schedule can be either active or passive. Slots in a passive schedule are considered as idle slots. The active schedules are at the begining of a frame, virtual slots in active schedules are called *active slots*. At the begin of each frame, every node randomly chooses a sending slot (the so called TX-slot) from all its active slots. The receive action occurs also in active slots, however, not among all the active slots, but in a certain schedule. We will explain later how it works. In the following, we denote by S the number of active schedules.

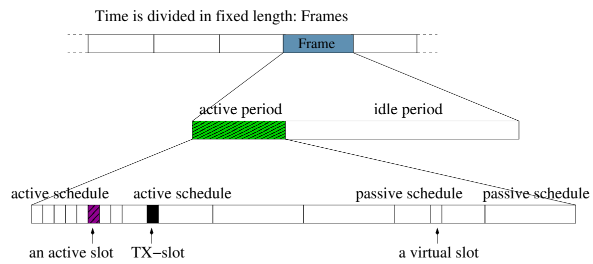


Fig. 1. Basic structure of distributed slotted Aloha

The distributed slotted Aloha algorithm has a fixed number of virtual slots per frame. In the current implementation at CHESS, this parameter is set to 80, and each schedule contains 8 virtual slots, i.e. there are 10 schedules in total. A node with S ($1 \leq S \leq 10$) active schedules has hence $8 \cdot S$ active slots, which are settled at the beginning of a frame. Per frame, a node will send in one of the active slot, and receive in one of the S active schedules.

Assume a node has 3 active schedules (Figure 2), i.e., it has 24 active slots. Since a node can receive only in one active schedule per frame, if it received in the first active schedule at the nth frame, then in the $(n+1)$th frame, it receives in the second active schedule, in the $(n+2)$th frame, in the third active schedule, and in the $(n+3)$th frame, it receives again in the first active schedule, and so on and so forth. But no matter in which active schedule it receives, it chooses randomly a send slot from all the active slots. In general, a node with S active schedules needs S frames to complete a *receiving cycle* over all $8 * S$ active slots. Active slots that are neither used to receive nor to send are considered idle.

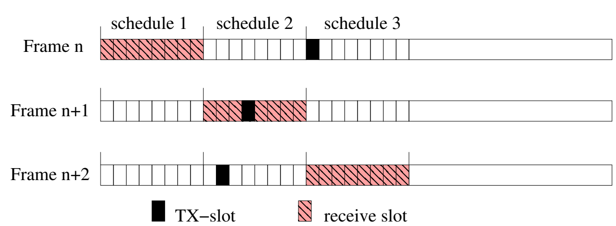


Fig. 2. Distributed slotted Aloha example of a node with 3 active schedules.

The number of active schedules of a node is dynamic and depending on the number of neighbours. To achieve the dynamic management of its neighbourhood, each node has an unique Id, and maintains a list containing the Ids of its neighbours. Each message contains the sender's Id. On receiving a message, if the sender is not yet in the neighbourhood list, it will be added to the list and the number of neighbours b will be increased by one. Furthermore:

1. At the end of each frame, every node checks its neighbourhood list. If there are some nodes that have not seen for 49 frames, remove those nodes from the list. For each removed node, b is reduced by one.
2. At the end of a receiving cycle, each node eventually updates S by the following rules:
   - if $(8 \cdot S < 2 \cdot b)$ and $S < 10 \rightarrow S + =1$
   - if $(8 \cdot S > 2 \cdot b)$ and $S > 1 \rightarrow S - =1$.

   Intuitively speaking, if the number of active slots $(8 \cdot S)$ of a node is not sufficient to receive all its neighbours $(2 \cdot b)$ and the maximal number S is not yet reached, increases S by one, so that eventually more neighbours can be found. Otherwise, reduce S by one. If $8 \cdot S$ is equal to $2 \cdot b$, let S be unchanged. The factor 2 for b comes from the consideration that not all messages from the neighbourhood can be received, and nodes assume that the actual number of neighbours is approximately two times the number of nodes in their neighbourhood list.

The dsA is basically a random access protocol with dynamic active slots. The idea behind this dynamic scheduling is to keep the real active phase of a node as short as possible, but obtain the same effect as if there would be more active slots. Assume a node has 24 one-hop neighbours, in a random medium access protocol with a fixed number of active slots, at least 24 receive slots per frame are necessary so that the node can get messages from every neighbour. This costs per frame three times more energy, than if there would be 8 receive slots per frame, and the messages are received over 3 frames. In other words, the dsA strategy delays the message transmission with the gain of longer node energy preservation and hence extends the network life. Later we compare the dsA strategy against the random access protocol with a fixed number of active slots (simple slotted Aloha, ssA for short), and show that it is indeed the case that dsA consumes much less energy (58.75%)to finish the all-to-all message communication, at the cost of slower propagation speed (2.5 times longer).

The dsA protocol uses constant and equal signal power for all nodes. The transmission power of nodes should be high enough to reach the intended receiver while causing minimal interference at other nodes. Hence the most critical parameter of this protocol is the value of the power. We choose SINR [11] as our radio propagation model which is more realistic than the unit disk model [15]. With this physical model, we first estimate the effectiveness of dsA. Later, we modify the protocol by letting nodes be able to regulate their signal power dynamically. The way how a node dynamically manages its signal power, and a comparison between constant power vs. dynamic power are presented in the section on results.

3 SINR

Receiving a message in a wireless context mainly depends on two things: the distance from which signals come, and the power with which they are transmitted. Those two variables form the basis of the SINR radio model [11].

Let p_i be the sending power of node i and x_i its position. Then the relative signal strength of a message from node i at node j ($j \neq i$) is determined by

$$r_i(x_j) := \frac{p_i}{d(x_i, x_j)^\alpha} \tag{1}$$

where $d(x_i, x_j)$ is the distance between x_i and x_j and α the path loss exponent, which determines the power loss over distance. $r_i(x_j) = 0$ if i is not sending. Depending on the environment, it is usually assumed that α has a value between 2 and 5 [8]. In an ideal vacuum, we have $\alpha = 2$ and we use this value for our experiments because it provides the upper bound of $r_i(x_j)$ with fixed p.

A signal can only be received if its relative signal strength is significantly higher than the strength of all other signals (for example, signals from other sending nodes) combined. This allows us to formulate a noninterference condition. Node j will receive a signal from node k if

$$r_k(x_j) > \beta(\sum_i r_i(x_j) + \nu(x_j)) \tag{2}$$

where β determines the minimal share of the whole signal which is needed for a successful transmission and $\nu(x_j)$ is the background noise at node j. The value of β must be between 0.5 and 1.

4 Modelling and simulation details

The modelling of the protocol described above is done in MoDeST, the "MOdeling and DEscription language for Stochastic and Timed systems" [3]. It allows us to describe the behaviour of discrete event systems in a compositional manner. As the possible slot assignments at each frame are of order $|slots|^{|nodes|}$, the situation we want to analyse are too complex to be solved in a formal way. Therefore, we simulate them and average the results. For this we will use Möbius [7], an integrated tool environment for the analysis of performability and dependability models. The conjunction between MoDeST and Möbius is realised by MOTOR [2, 4], a tool that is integrated into the Möbius framework and aims to facilitate the transformation and analysis of MoDeST models.

Modelling assumptions. Per receive slot, a node can receive at most one message. Collisions occur when more than one node is sending to the same node at the same time. We assume there are no message loss during propagation, hence the only reason that a node received nothing is due to a collision.

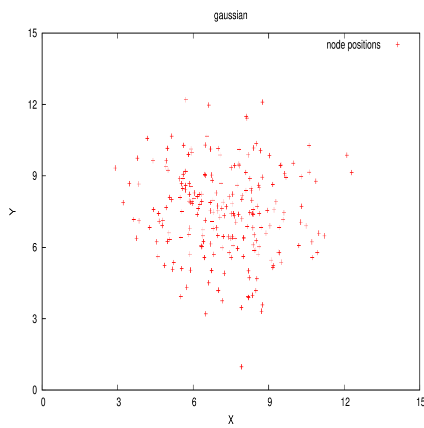
Distributed slotted Aloha incorporates a mechanism to synchronise clocks of neighbouring nodes. And with MoDeST we can treat every node as a single process equipped with a different clock. However, since clock synchronisation is not the focus of this paper, we assume in our model that all nodes are perfectly synchronised to each other.

Important simulation parameters for the experiments are explained below.

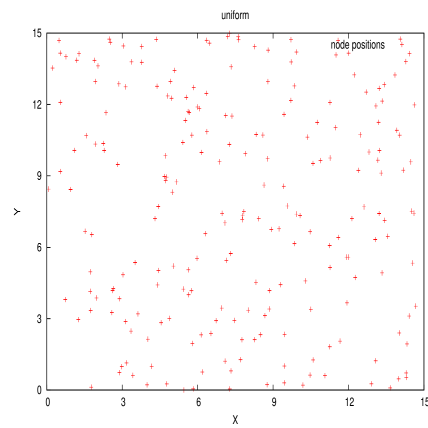
4.1 Node arrangement

In order to investigate the protocol behaviours under distinct network connectivity conditions, we consider the following topologies.

1. Grid network of size 15×15 with 225 nodes, each node is placed at the vertex of the grid. It is large enough to create most of the interesting situations and at the same time small enough to become no burden on our computation time.
2. We generate 225 points inside a 15×15 area around the point $(7.5, 7.5)$, so that the choice of the coordinates of those points is governed by a Gaussian distribution (see Figure 3(a)). This kind of node arrangement builds a much more clustered structure comparing to the centre-less grid network.



(a) Gaussian



(b) uniform

Fig. 3. Two random networks

3. The last graph represents a uniform distribution, i.e. both x and y coordinates of the 225 nodes are generated by a uniform distribution from 0 to 15. A typical graph generated by this setting is shown in Figure 3(b).

4.2 Simulation parameters and return data

As the basic purpose of any network is to transfer information, we focus our analysis on measure its ability and particularly the cost for this transfer. The cost parameters of interest in our study are time and energy consumption. In the experiments, we consider the all-to-all communication, and measure the energy consumed by the whole network. We equip each node with a distinct flag. Initially, every node only has its own flag set. Every time when node j receives a message from node k, all flags of node k are set for node j. All networks considered in this paper consist of 225 nodes, which means there are

$225 \times 225 - 225 = 50400$ new flags to set. So the return data of the experiment are the energy consumption E by the whole network when m ($m \leq 50400$) flags are set. We consider the propagation task to be successfully finished if 99.9% of the 50400 new flags are set. Since the number of maximal possible active slots in the dsA protocol is 80, we vary the number of active slots in the ssA from 16 to 80.

Two important parameters in the SINR model are the signal power p and the background noise ν. We first observe from the in-equation (2) for the radio model that it suffices to vary one of these parameters only. Increasing the background noise by a factor η has the same effect as decreasing the power of all senders by a factor $\frac{1}{\eta}$, as has the scaling of all distances by η^2. So to keep things simple, we fix the background noise ν and vary the signal power p.

To determine the range of values of ν that are of interest, we consider of which potentially effective transmission distances are reasonable, or in other words, how many neighbours on average a node should have, so that we can examine both sparse and dense networks. Considering the grid network, if we choose the maximum possible range for a message to be transmitted be 2.1, 4.1 and 6.1, each node has 12, 48 and 112 direct neighbours, respectively. Now we can deal with the choice of background noise. We determin it by consider the case for a single sender in the noninterference condition, as explained in the following.

Recall that the physical model declares a message from node k to node j as received if (2) holds. Under the ideal case that the only node sending was k, this inequality simplifies to $r_k(x_j) > \beta(r_k(x_j)+\nu(x_j))$. We know that the strength of the received signal from node k at node j is given by (1). Using this and solving for the distance $d(x_k, x_j)$ we get

$$d(x_k, x_j) = \sqrt[\alpha]{\frac{p_k(1-\beta)}{\beta\nu}}$$

as the maximum distance. Since there are always some other nodes interfering with k, we can only view this as an upper bound.

Using this equation, if we choose $p = 10$, $\alpha = 2$ and $\beta = 0.7$, we get noise levels 0.971, 0.255 and 0.115, respectively, which loosely approximate the range 2.1, 4.1 and 6.1 mentioned before. Accurate values for α and β are not relevant, since from the above equation, we can always manipulate the signal power p and obtain an equivalent result of distance.

mode	current
transmit	11.3 mA
receive	12.3 mA
idle	0.9 μA

Table 1. Energy demands of the nRF24L01 radio

The sensor nodes developed by CHESS are equipped with an ATMega64 micro-controller and a Nordic nRF24L01 [14] packet radio. The energy demands of the Nordic nRF24L01 radio are summarised in Table 1. In the MoDeST model, we use the multiplication of those real values and the signal power p as the energy consumption in each send, receive, and idle slot. All simulations ran at least 500 times. The confidence level of all simulations is set to 0.95 and the relative confidence interval is 0.1.

5 Results

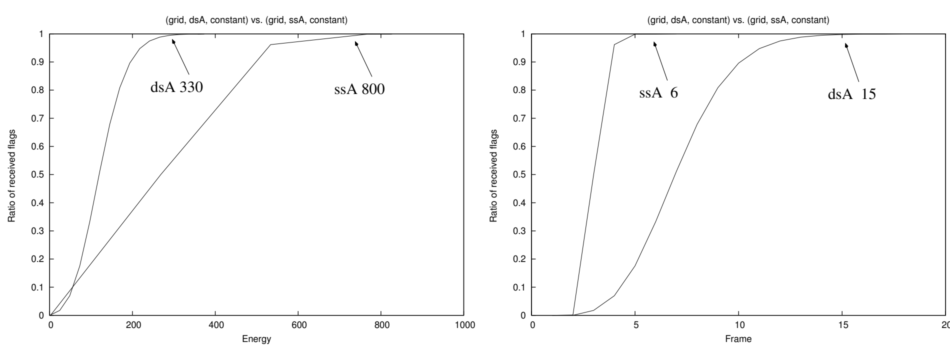
In this section we present the results of our analysis of the two protocols ssA and dsA, in the different topologies grid, Gaussian and uniform. Furthermore we investigate the influence of dynamic send-power management in dsA. We denote these different network configurations as a triple (*network*, *protocol*, *power*), where *network* can either be grid, Gaussian or uniform; *protocol* can be dsA or ssA; and *power* is either constant or dynamic.

5.1 DsA vs. ssA

To compare dsA and ssA, we choose $p = 15$, $\nu = 0.255$ and vary the number of active slots in configuration (grid, ssA, constant) from 16 to 80 in steps of 16. As we can see in Figure 4(a), the energy consumption is much lower in (grid, dsA, constant) than (grid, ssA, constant). When the propagation finishes, (grid, dsA, constant) consumes on average 330 energy units, whereas the best result in (grid, ssA, constant) consumes 800 energy units.

This means a 58.75% of energy saving with dsA. Thus, dsA saves energy, but as a penalty, throughput is slower. This effect is illustrated in Figure 4(b). There we see the fraction of flags distributed through the network vs. the time it took: with dsA, 15 frames, with ssA, just 6.

The same experiments, repeated with various other power parameters, show similar results: dsA propagates messages slower than ssA, but with less energy.



(a) Energy comparison (b) Latency comparison

Fig. 4. dsA vs. ssA

We repeat the experiment for Gaussian and uniform networks and the results show again that dsA propagates messages slower than ssA, but cost less energy.

5.2 Optimal signal power

In this section we investigate whether there is an optimal power level in terms of message propagation speed (thus, also energy).

To that end, we investigate the all-to-all message communication in (grid, dsA, constant), (Gaussian, dsA, constant) and (uniform, dsA, constant). We set again the background noise $\nu = 0.255$ for all the experiments and vary the signal power p. The results of the simulations for the Gaussian network are shown in Figure 5 (Note that the plot is focused to the $95\% - 100\%$ interval. Otherwise the curves would be too hard to distinguish). The curve for $p = 6$ represents the most efficient result, because it reaches the 99.9% mark first. For lack of space, the graphs for the grid and uniform network are not shown. All networks have in common that there exists an optimal signal power for each network. Any other values of p that are larger or smaller than this value yield a greater energy consumption when the all-to-all communication is done. Where they differ is the optimal value of signal power, and the total energy consumed by the network for this power.

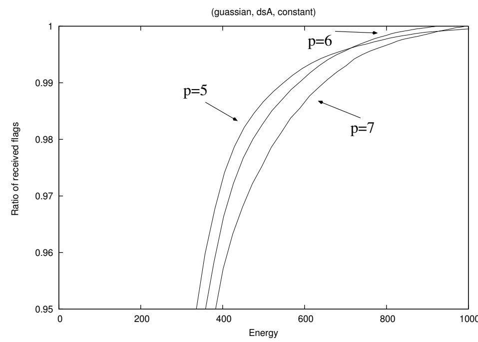


Fig. 5. Optimal transmission power for (Gaussian, dsA, constant)

We can summarise that for the (Gaussian, dsA, constant) scenario, the optimal power is 6, whereas in (grid, dsA, constant) and (uniform, dsA, constant), the value is 15 and 16, respectively. Our explanation why the power in (Gaussian, dsA, constant) is smaller is that nodes are more densely distributed. The average distances between nodes in grid and uniform are larger, requiring higher send power. This also explains why the optimal values of signal power for these two node arrangements are similar.

5.3 Dynamic power management

Introduction. Given a certain transmission power p and background noise ν, the number of nodes inside the maximal reachable transmission range of each node in the network is usually individual (except in the grid network). This means the dsA strategy with the identical transmission power for all nodes may cause a lot of interference in dense clusters of the network, which in sum might even interfere with nodes far away outside the cluster. In sparse areas in the network, on the other hand, nodes may disconnect from the network due to the interference, and need to send with higher power to make themselves heard.

One idea to deal with these two unfavourable situations is to vary the transmission power of nodes dynamically, so that nodes in dense areas are able to lower send power to reduce interference, and nodes in sparse areas can increase send power to maintain network connectivity.

This forms the basic idea of dynamic power management in dsA. There are two questions to answer: first, how to determine whether a node is in a dense or a sparse area of the network; second, how to adjust the power levels. The first question we answer as pragmatically: we use the neighbourhood list that is already present in the dsA protocol and define a low- and high-water-mark L and H which, when tripped, cause an increase or decrease in power. The answer to the second question is inspired by the Nordic nRF24L01 radio. This radio has for different power levels of 0dBm, -6dBm, -12dBm, and -18dBm, *i.e.,*full power, and 1/4, 1/16 and 1/64 power. The radio has thus an exponential decrease of send power with each level.

The mechanism we use to adjust the power is then the following: for node i with size of neighbourhood list b_i and current power level p_i, at the beginning of each frame: if $b_i > H$, then $p'_i = p_i/n$; if $b_i < L$, then $p'_i = np_i$, where $n \in \{2, 3, 4, 5, \ldots\}$ the factor to increase or decrease the power and p'_i the new power level.

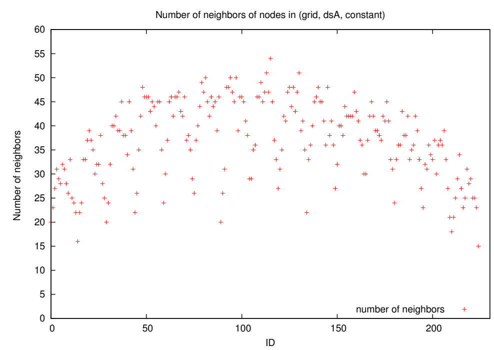
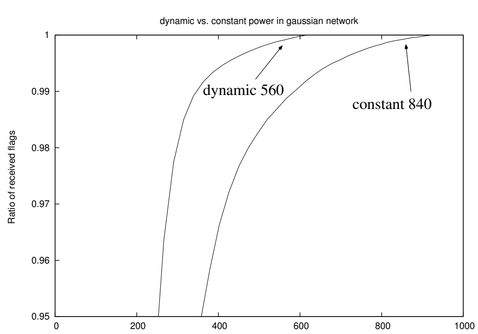
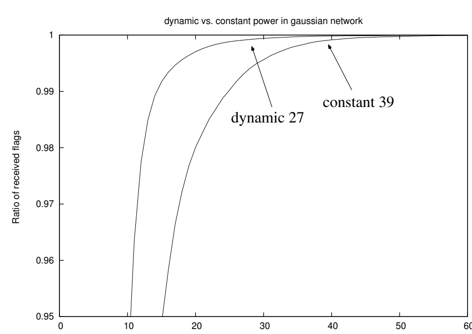


Fig. 6. Number of neighbours of all 225 nodes in (grid, dsA, constant) with $p = 15$ and $\nu = 0.255$

The question is how to determine H and L. For this, consider Figure 6, which shows the number of neighbours of all the 225 nodes in (grid, dsA, constant) with background noise $\nu = 0.255$ and signal power $p = 15$, at the end of the 500th frame. Note that this p is the optimal value of signal power for (grid, dsA, constant), as determined in Section 5.2. As we can see, most of the nodes in Figure 6 have between 20 and 50 neighbours, so we use these two values as the lower and upper bounds in dynamic power management.



(a) Gaussian, energy comparison



(b) Gaussian, latency comparison

Fig. 7. Comparison of constant and dynamic power assignment in Gaussian network

Experiments and Results. We are again interested in the speed of flag propagation in the whole network. The results for the uniform and grid network topology show no influence of dynamic power management on the results compared to the ones in Section 5.2. We explain this with the relatively uniform structure of the topologies, where no area is much denser than the other.

output power	current
0 dBm	11.3 mA
−6 dBm	9.0 mA
−12 dBm	7.5 mA
−18 dBm	7.0 mA

Table 2. Output power setting for the nRF24L01

We thus concentrate on the Gaussian case, *i.e.,*(Gaussian, dsA, dynamic). We set $\nu = 0.255$ and $p = 6$ as the maximum power level (which is the optimal value determined in Section 5.2). As the energy consumption of the radio we use Table 2 of the Nordic radio.

We conducted two different experiments for two different values of n, the factor used in the power adaption: $n = 2$ and $n = 4$. The case $n = 4$ thus mimics very closely the power levels of the Nordic radio. Interestingly, the result shows no improvement over dsA without power-management. However, with $n = 2$, it does, as is shown in Figure 7. In Figure 7(a), we compare the energy consumption between (Gaussian, dsA, constant) and (Gaussian, dsA, dynamic). Obviously, with the same initial power value, (Gaussian, dsA, dynamic) consumes 30% less energy than (Gaussian, dsA, constant). The results with other p are similar and not shown here. In Figure 7(b), we see that, the dynamic power management not only reduces energy consumption but also accelerates propagation speed.

Apparently the choice of the factor n is important to affect an improvement in the energy consumption and latency. The conditions under which such improvement can be achieved requires more investigation and is subject of our future work.

6 Conclusions and Future work

In this paper, we reported on the simulative analysis of the distributed slotted Aloha protocol, aimed for gossiping-based application in sensor networks. Our analysis shows that, comparing to a simple slotted Aloha strategy with a fixed number of active slots, the distributed slotted Aloha with dynamic number of active slots significantly reduces the energy consumption (almost 60%) for all-to-all communication, although throughput decreases. We showed that the optimal transmission power is different from network to network, which indicates a necessity of dynamic power management. We proposed a modification of the distributed slotted Aloha protocol by a simple dynamic power assignment scheme, and show that this not only reduces the energy consumption (30%) but also speeds up the message propagation.

Our future investigation will focus on different schemes for power-management and on determining the circumstances which make dynamic power management effective.

Acknowledgement

We thanks Bert Bos, Frits van der Wateren, Marcel Verhoef (all of CHESS) for discussions on the protocol and suggestions on the modeling. This research has been founded by the EU under grant number FP7-ICT-2007-1 (Guasimodo) and the DFG Excellence Cluster UMIC.

References

1. T.R. Andel and A. Yasinac. On the credibility of MANET simulations. *IEEE Computer*, 39(7):48–54, 2006.
2. H. Bohnenkamp, T. Courtney, D. Daly, S. Derisavi, H. Hermanns, J.-P. Katoen, V. Vi Lam, and B. Sanders. On integrating the Möbius and MoDeST modeling tools. In *Proceedings of the 2003 International Conference on Dependable Systems and Networks*, volume 4424. IEEE Computer Society Press, 2003.
3. H. Bohnenkamp, P.R. D'Argenio, H. Hermanns, and J.-P. Katoen. MoDeST: A compositional modeling formalism for real-time and stochastic systems. *IEEE Trans. on Software Engineering*, 32(10):812–830, 2006.
4. H. Bohnenkamp, H. Hermanns, and J.-P. Katoen. Motor: The MoDeST tool environment. In *Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, volume 4424, pages 500–504. Springer-Verlag, 2007.
5. D. Cavin, Y. Sasson, and A. Schiper. On the accuracy of MANET simulators. In *ACM Workshop On Principles Of Mobile Computing*, pages 38 – 43, 2002.
6. Z. Chen and A. Khokhar. Self organization and energy efficient TDMA MAC protocol by wake up for wireless sensor networks. *In Sensor and Ad Hoc Communications and Networks*, pages 335–341, 2004.
7. D. D. Deavours, G. Clark, T. Courtney, D. Daly, S. Derisavi, J. M. Doyle, W. H. Sanders, and P. G. Webster. The Möbius framework and its implementation. *IEEE Trans. on Software Engineering*, 28(10):956–969, 2002.
8. A. Fanghänel, T. Kesselheim, H. Räcke, and B. Vöcking. Oblivious interference scheduling. In *ACM Symposium on Principles of Distributed Computing (PODC)*, pages 220–229, 2009.
9. J. L. Hammond and P. J.P. O'Reilly. *Performance Analysis of Local Computer Networks*. Addison-Wesley, 1986.
10. M. R. Jongerden, A. Mereacre, H. Bohnenkamp, B. R. Haverkort, and J.-P. Katoen. Computing optimal schedules for battery usage in embedded systems. *IEEE Trans. Industrial Informatics*, 6:276–286, 2010.
11. P.Gupta and P. R. Kumar. The capacity of wireless networks. In *IEEE Transactions on Information Theory*, volume 46, pages 338–404, 2000.
12. R. A. Rashid, W. Mohd, A. Ehsan, W. Embong, A. Zaharim, and N. Fisal. Development of energy aware TDMA-Based MAC protocol for wireless sensor network system. *In European J. of Scientific Research*, pages 571–578, 2009.
13. L. G. Roberts. Aloha packet system with and without slots and capture. In *ACM SIGCOMM Computer Communication Review*, volume 5, pages 28–42, 1975.
14. Nordic Semiconductors. *nRF2401 Single-chip 2.4GHz Transceiver Data Sheet*, 2002.
15. H. Yue, H. Bohnenkamp, and J.-P. Katoen. Analyzing energy consumption in a gossiping MAC protocol. In *Measurement, Modelling, and Evaluation of Computing Systems and Dependability and Fault Tolerance*, volume 5987, pages 107–119, 2010.