NoSE: Neighbor Search and Link Estimation for a Fast and Energy Efficient Initialization of WSNs

TIK-Report No. 285, April 2008
Andreas Meier, Mischa Weise, Jan Beutel and Lothar Thiele
Computer Engineering and Networks Lab, ETH Zurich, 8092 Zurich, Switzerland

Abstract

With numerous application-specific low-power protocols being developed over the past years the design space has been widely covered. However, not much attention has been paid to the start up and initialization phase of a low-power protocol stack, e.g. when the actual start of the application is delayed by a long installation and deployment phase. In this paper we present NoSE, an extension to existing protocols that optimizes operation in the startup phase. NoSE uses a reduced signaling scheme to achieve an extremely efficient, yet highly reactive wakeup scheme for a delayed initialization of the main communication. Upon initialization, using a wakeup beacon, the protocol scheme wakes all neighbors and performs a neighborhood search with an integrated link assessment. Following the description of the protocol we present an evaluation using analytical methods and a testbed implementation concluding in a discussion including a comparison with other well known protocols.

1 Introduction

In the past years a multitude of sophisticated protocol and algorithms have emerged, allowing resource efficient, in particular energy efficient, operation for wireless sensor networks (WSNs). On the MAC layer this is achieved by minimizing idle listening, overhearing and transmission time, which can be achieved in many different ways leading to a vast amount of available MAC protocol for WSNs. On the network layer energy savings are possible by evenly balancing the traffic load among the nodes in a network and minimizing the routing overhead in particular by selecting well connected neighbors only. There are also custom solutions like Dozer [2] allowing a very energy efficient operation by combining the MAC and routing issues into a complex cross-layer protocol. On the application layer further energy optimizations are possible by aggregating both data packets and/or payload minimizing the load for the underlying layers.

While having numerous possibilities nowadays to assemble a very resource-efficient and power-aware distributed sensor network tailored to a specific application, the task of initializing the network has not yet attracted much attention. Especially the period required for deploying the nodes, i.e. the phase where nodes are being installed and commissioning is ongoing, is often neglected. For instance deployments in alpine regions or on a volcano, for measuring permafrost or seismic activity respectively, have shown that setting up a single sensor node easily requires a day. Even in well-accessible environments, the installation of a single sensor rather takes days than minutes. So the first nodes in a network that are being installed and turned on are most likely to be in solitude for a long time. If these are intensively listening for ongoing communication or actively broadcasting routing announcements, a lot of energy is wasted, long before the system could possibly start to become operational! More sophisticated protocols try to switch to energy saving modes after a certain time, reducing the listening and beaconing frequency, resulting in reduced responsiveness and are in the following only capable of waking up and rejoining the network with considerable delay. The responsiveness is usually not hard bounded and therefore the time for all nodes to become fully operational can be very long. This uncertainty is far reaching, since without further information at this point in time, if a node is not reporting, it is not known, if this particular node is not yet connected, has triggered an internal error, if it was deployed out of communication range or simply has run out of power.

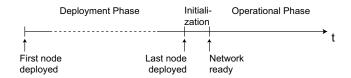


Figure 1. NoSE optimizes deployment and initialization phase in the lifetime of a WSN.

Protocol	NoSE	Dozer	B-day	T2
		[2]	[7]	CTP
Time Bound	✓	Х	Х	√
Link Assessment	✓	X	X	X
Topology stabiliza-	3	20	X	60
tion [min]				
Duty Cycle [%]				
- first 10min	0.3%	27.5%	1.0%	1.5%
- first 6h	0.3%	1.1%	1.0%	1.5%
Mean Current [mA]	0.28	_	0.41	0.67

Table 1. Performance of different protocols during initialization. The duty cycle indicates the energy consumption of a newly started node if neighbors are not yet present.

The communication range is another important but widely neglected issue during the initialization of the network. The low-power radios being used in a multi-hop deployment, where many nodes are located close within a node's communication range, poses severe problems for the robustness and reliability of WSN. This results in unpredictable communication channels (links) between nodes due to reasons of interference and fading. Various research groups analyzed this behavior [10-12, 14] and showed that the quality of different links varies greatly and that some links show a very unpredictable, non-deterministic behavior. This has implications on routing protocols, where a potential neighbor is selected and its link performance is assessed continuously during operation. Should a selected neighbor turn out to be poorly connected, i.e. requiring many retries, this particular neighbor must be replaced by one that is assumed offer a more stable link in the routing table. Hence, at the startup, the network topology can still be very unstable and a gradual refinement is only achieved over time. These dynamics do not only waste the most precious asset for WSN, energy and therefore reduce the sensor network lifetime, but also limit the network's performance right at the deployment. The situation is even more severe, when deployment or servicing personnel wants to gain a quick (first) impression of the network's performance as soon as all nodes have been deployed. An overview of some well known protocols and the performance measured is shown in Table 1 and also in Figure 2.

This paper proposes NoSE (NeighbOr Search and Estimation), a protocol specifically designed for an improved performance before and during initialization, minimizing energy usage in order to maximize system lifetime. The NoSE protocol is introducing a highly optimized protocol scheme for this specific problem that can be readily integrated with a number of widely used protocol stacks. In particular, the major contributions in NoSE are: (1) Energy-

efficient operation (0.3% duty cycle) right from the beginning of a node's powering up. (2) Allowing a fast and efficient network initialization with deterministic runtime that can be initiated at any node. (3) During initialization a complete list of all links available including respective quality metrics is assessed, providing a profound basis for the subsequent set up of routing tables in higher protocol layers.

For the remainder of this paper we use the following definition to describe the lifetime (see Figure 1) of a sensor network and define as follows: (1) With the first node in the network being turned on the so-called *deployment phase* starts. (2) As soon as all nodes are deployed and a wakeup beacon is sent, the so-called *initialization phase* begins that ends when all nodes have their routing table set up and (3) the network is *operational*, i.e. all nodes are ready for data.

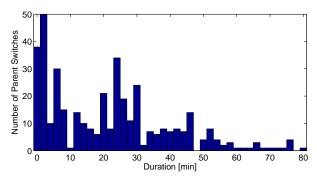
1.1 Related Work

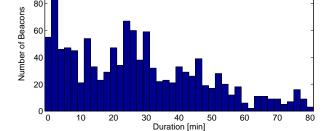
Resource efficient MAC protocol design has attracted a lot of attention in recent years, resulting in a vast amount of different protocol designs. All these protocol trade off latency and bandwidth for a reduced energy consumption by duty cycling the radio. However, the protocols vary greatly in their complexity and application domain, for instance resulting in protocols being specifically designed for dense networks, ultra low-data rate applications or adaptively accommodating different workloads. More complex MAC protocols are often based on a global structure (slot based, TDMA), whereas less complex designs are usually based on a random access scheme (CSMA).

Protocols based on a global structure, such as S-MAC, Z-MAC, LMAC, DMAC, Crankshaft or SCP-MAC, require to be set up before data can be sent. This requires the nodes to listen intensively (with some protocols having duty cycle of 100% in this phase) for any ongoing communication, which subsequently allows learning of the current channel policy. However, if no such communication is detected, the nodes usually start sending neighbor-announcement beacons in a regular interval, trying to find neighboring nodes. Naturally, this "best effort" practice requires energy, especially if only limited numbers of neighbors are present.

Another critical aspect in a network being turned on over a long period of time is the likelihood that multiple separated clusters emerge unintentionally; each of them having different channel access timings and not being visible for each other at all. Hence, it usually takes a long time before two separated clusters merge. Apart from the excess energy consumption, this is especially cumbersome in the case of a newly deployed network where status on the network operation as well as connectivity should be available quickly.

On the other hand, the family of random access protocols, does not have to deal with setting up a structure beforehand. These protocols, such as B-MAC, WiseMAC, X-





- (a) It takes the network about an hour to obtain a stable topology.
- (b) A lot of broadcasts are sent in the early phase of the operation; wasting a lot of energy and also channel bandwidth.

Figure 2. Dynamics of the network after start-up for a TinyOS-2.x CTP implementation (see Figure 7).

MAC or CSMA-MPS, listen periodically on the channel for ongoing communication and signal this to higher layers. As the MAC layer does not initiate message transmissions by itself it is up to the upper layer protocol to decide when the sending of the first message is being initiated. However, activity on the channel can only be observed within the own listening window and these windows are intentionally kept small in order to not waste energy on idle listening.

McGlynn and Borbash proposed the so-called Birthday Protocol [7] for discovery of a node's neighbors during the network's initialization, which they analyzed analytically and evaluated with simulation only. In order to save energy all the nodes start in the so-called BL mode, in which they listen to every p_L th slot in average (randomly distributed) and sleep otherwise. As soon as a first message is received, the node switches to a so-called PRR mode for a distinct number of slots N (e.g. N = 3000) sending every p_T th slot and listening otherwise (i.e. 100% duty cycle). Afterwards the node is switched back to BL mode. In order to init the protocol, the sink node is manually set into PRR mode. The neighbor discovery finishes when all nodes in the network reappear in BL mode, which is a non-deterministic duration depending mainly on the network characteristics. Furthermore, a complete list of neighboring nodes is generated, however, it is unknown if these are well connected.

Kuhn et al. [6] propose a clustering algorithm for an energy efficient initialization, setting up a clustered structure without any central control, i.e. there is no central trigger starting the initialization process. In the end, all nodes belong to one of the clusters, being either clusterhead or knowing the clusterhead's schedule. Moscibroda et al. [8] analytically determined the trade-off of the energy consumption while deploying a network and the reaction time for disseminating an external event (e.g. start beacon sent at the sink). In the algorithm they propose, the nodes arrange themselves in clusters, with the children requiring only very

little energy. Due to the existing structure, a wake-up beacon from the sink can be forwarded much faster. Both these clustering algorithm are too complex to be implemented on a resource limited node. Furthermore the clusterhead, coordinating its children, is operating very energy inefficient and is not suited for homogenuous networks. In [8] a second, so-called uniform algorithm is proposed, which is based on an exponentially increasing sending probability, which renders this algorithm not very energy efficient.

Keshavarzian [5] proposes a scheme for combining neighbor discovery and link assessment. He proposes different access schemes (random and code based) for sending, listening and sleeping in a slotted structure, in order to minimize interference. This approach however is based on the assumption that all nodes in a network are switched on synchronously, which is impossible to achieve in a real deployment.

In [13], Woo et al. investigate and evaluate various estimators for link-quality assessment while running in operational mode. Some of the estimators rely on overhearing all neighboring traffic; others only consider the local communication. The authors analyze how a finite, typically quite small neighbor table, providing connectivity and routing information, can be managed. Such an algorithm considerably improves network operation; however it does not solve the issue of estimating the link quality during initialization.

2 Initializing a WSN

Setting up the routing tables during the initialization requires a profound knowledge of the nodes' neighborhood. Gathering this information needs to consider the following five basic issues:

• *Neighbor Discovery:* At start up no information about the neighboring nodes is available, requiring message exchange.

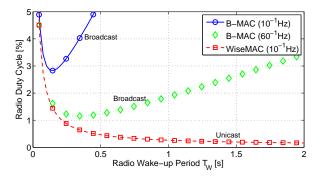
- Link-Quality Assessment: The link quality of the different neighbors might vary considerably. This is indicated in Table 2 showing that more than half of the links available in the network have a packet-reception rates of less than 85%. This imposes the necessity of a link-quality assessment during the initialization of the network, allowing to select high-quality neighbors during the set-up of the routing table only.
- Deployment Delay: The deployment phase takes considerable time, in select cases it can even take a couple of weeks. During this phase, it is important that the nodes do not drain a substantial fraction of their battery power. For instance, by desperately looking for neighbors, i.e. by broadcasting lots of beacons, while the neighboring nodes are not even installed.
- Energy vs. Responsiveness: The energy consumption is crucial for most sensor network deployments. On the other hand the time delay of the initialization is also a major concern as an immediate feedback of the networks performance is desirable, or at least a time bound when this feedback is available. For most current protocol stacks the requirements on the energy and time consumption is a trade-off: Either a protocol spends a lot of energy by aggressively looking for neighbors, allowing fast topology formation, or the radio communication is minimized decreasing the responsiveness. Many protocols follow a hybrid approach, at start-up the neighborhood is checked intensively, allowing for a high responsiveness. If no activity is detected, the node reduces their communication. However, if nodes are not yet ready to participate in the neighborhood, this approach is in vain.
- Compatibility A specialized initialization protocol is likely to require a complete second protocol stack for the start-up of the network. It is preferable if the initialization protocol fits (enhances) the current stack, minimizing the node's resource requirements and risk of errors: less program code, no transmissions required.

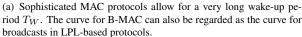
In order to tackle these issues, one of the most important design paradigms for NoSE is the compatibility with widely used protocol stacks (in particular MAC protocols), allowing to enhance the current system with minimal overhead. Nowadays, most MAC protocols used are based on a concept referred to as *low power listening (LPL)* that is being used in MAC protocols such B-MAC [9], WiseMAC [4], X-MAC [1] and many others. With LPL based protocols, the nodes have the radio turned off, only switching it on every T_W (e.g. $T_W = 300ms$) in order to sample the channel for a short time T_{cs} (e.g. $T_{cs} = 2ms$) and so performing a carrier sense. Should such a carrier exist, the node keeps listening, otherwise the radio is switched off immediately. This

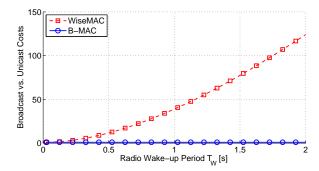
concept results in a very energy efficient operation (duty cycle = T_{cs}/T_W) if there is no or little communication in the network. Sending a packet on the other side is much more expensive, since the respective receiver's wake-up period is not a priori known. Hence a long wake-up preamble, or alternatively a long packet stream, on the order of a nodes wake-up period T_W has to be broadcasted. This does not only require a lot of energy for the sending node, but also puts a lot of load on the channel for a small packet that could be transmitted in a few milliseconds. WiseMAC extends this scheme with link based synchronization, allowing the sending node to predict the receivers wake-up time and therefore minimizing the length of the wake-up preamble. Should the information not be available or outdated, WiseMAC falls back to B-MAC's long wake-up preamble. The synchronization information required for the wake-uptime estimation is provided with two additional bytes in the acknowledgment, which is very effective and efficient during the operation, but rather inefficient during the initialization. This is due to the fact, that three packets with long preambles have to be exchanged for each neighbor before the link is synchronized. It is therefore suggested to extend this synchronization scheme by adding the information to broadcast messages, allowing to synchronize all neighbors upon sending one single broadcast.

For a synchronization based communication the preamble length depends on the message frequency (clock drift) and not on the wake-up period. This is further detailed in Figure 3(a), showing a well defined point of operation (i.e. minimum) for B-MAC given a certain message frequency. WiseMAC on the other hand, does not show such a minimum and hence a wake-up period in the order of one or two seconds is preferable, if the data rate used allows this. While such a long wake-up period allows for a very efficient unicast communication, the costs for a broadcast (requiring a node to be sending for a whole wake-up interval) increases significantly. As indicated in Figure 3(b), for a wake-up period in the order of one and a half seconds, the energy equivalent of sending a single broadcast would be to send about 75 unicast messages! Broadcast messages are mostly required for announcing changes in the network topology. Thus using such a protocol during initialization, favoring long wake-up periods, the amount of broadcasts should be reduced, requiring a careful selection of neighbor links.

While there are many extensions to the original concept of low-power listening, the initialization concept offered by NoSE can be easily integrated with any of them. In particular, NoSE can be used with bit stream based radio as well as with newer packet based radios. This means that NoSE can be used in combination with the prevailing number of MAC protocol being used in TinyOS 1.x and 2.x.







(b) The costs for a broadcast are the same for B-MAC and WiseMAC. Compared with a unicast, a single broadcast can get more than 100 times as expensive if a long wake-up period is chosen.

Figure 3. Broadcast and unicast performance analysis of B-MAC and WiseMAC.

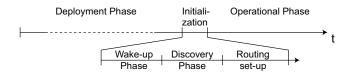


Figure 4. NoSE operational phases.

3 NoSE Protocol

A node being installed and powered on has no information about the neighborhood. The node starts exploring the neighborhood by either intensively listening or transmitting. Setting up the neighborhood information when not all surrounding nodes are yet available is of limited use. Especially if the goal is to retrieve a well assessed and complete neighbor information, this cannot be done before all nodes are deployed, rendering any efforts made to find neighboring links a waste of resources. Therefore, in NoSE nodes do not send when being powered on but only listen to the channel every wake-up interval to detect for any ongoing communication. Since all nodes in the network follow this paradigm, no message exchange takes place during the deployment phase. This of course requires a user intervention as soon as the network is deployed and wants to be used. The start can be triggered manually at the sink node or by triggering a specially programmed node, launching the so-called wake-up call. This is basically a flood into the network making all nodes ready for the upcoming 'synchronous' discovery phase. Synchronous in this respect does not mean that the nodes synchronize their wake-up times, but that they rather synchronize on a common time window during which the neighbor search and link assessment is taking place. Hence a rather loose synchronization in the order of a second is sufficient. This whole process is shown in Figure 4.

During the discovery, all nodes have to send a well-defined number of messages, making it beneficial to adapt the wake-up period to a so called high-alert mode during this phase. This saves a lot of energy as due to the temporarily increased channel bandwidth the duration is shortened. The discovery phase ends at the same time for all neighbors. It results in complete and well-assessed neighbor information available at every node. Following this is the set-up phase of the routing tables before the operation of the network can actually start.

3.1 Wake-up Call

The aim of the wake-up call is to prepare all nodes in the network for the subsequent discovery phase. To be more precise, all nodes have to know the point in time the discovery phase begins, its length and further discovery parameters such as the number of packets being exchanged and the adapted channel-polling interval. This information has to reach all nodes reliably and in time, before the start of the discovery phase. While the synchronous phase is launched synchronously for all nodes, i.e. with a granularity on the order of a second, the wake-up call is received asynchronously by the individual nodes in the network. NoSE's wake-up call is therefore not just a simple flood of the network but a combined approach leveraging speed and reliability, by allowing every node to send multiple broadcasts. This of course is paid by an increase in energy consumption; however this is only of minor concern since the wakeup only takes place once during the lifetime of the node and cannot be compared with a repeated flooding during the operational phase. The responsiveness of the wake-up call is achieved by using a short back-off timer after a node receives its first wake-up beacon. This back-off timer will increase the chances of a collision due to hidden terminals; however, the goal of the first wake-up call is not primarily

to reach all nodes, and rather is meant as an effort to reach the border of the network as soon as possible. What follows is a second broadcast having a much larger back-off time which greatly minimizes the chances for collisions and allows to reliably reach all other nodes. The parameters for the wake-up call, i.e. the number of beacons and the respective back-off times, are also sent with the wake-up beacons, allowing to parameterize the wake-up call to deployment characteristics should this be necessary.

Missing the wake-up call, despite the dual broadcasts from all surrounding nodes means, that a particular node will not take part in the NoSE discovery phase. However, at some point during the operation the node will pick up a message and learn that the network is already initialized. This response to neighbor announcement beacons however is exactly what most routing protocols do exclusively during their start up. In the approach presented the responsiveness of the small fraction of nodes that are missed on initialization is not increased; however they profit from a minimized energy consumption during the deployment phase. Furthermore it should be stressed, that missing the wake-up call in NoSE usually means that the node is very badly connected and a manual interaction is required, i.e. relocation of the node or the addition of an additional relay node for a better connectivity.

3.2 Discovery Phase

The discovery phase starts synchronously for all nodes in the network. During this time (e.g. 3 min) all nodes are required to send exactly N broadcast messages containing their node identifier. All nodes keep a neighborhood table with an entry for every single node they received a message from during the discovery. Furthermore, for every single node, the number of received messages as well as the maximum RSSI value of a message received by this node is kept in the neighborhood table. By the end of the discovery phase, the number of received packets and the RSSI value allows estimating the link quality as discussed in detail in Section 5.4.2. This estimation has then to be transferred to the network layer's routing table, making it advisable to have a combined neighbor list allowing a smooth transition and saving on the memory requirements.

During the discovery phase a lot of messages are being exchanged in a short time increasing NoSE's responsiveness. However, shortening the phase increases the probability for messages to collide and thus affecting the quality of the link estimation. Therefore the channel load needs to be capped, which affects the minimal length of the discovery.

Besides the responsiveness the energy consumption is also of major concern. As indicated in Figure 3, a wellknown communication pattern, in particular the amount of messages being sent in a given time frame, allows optimizing the MAC protocols energy efficiency by minimizing the wake-up period. This is illustrated in an analysis in Figure 5 assuming a node having a mean of 10 neighbors.

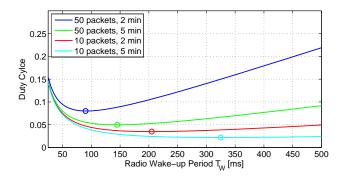


Figure 5. Model of NoSE's energy consumption during the discovery phase.

Besides minimizing the energy consumption, the channel occupancy is also minimized due to the shortened broadcasts. This in turn allows a shorter wake-up period resulting in a maximal channel load, saving additional energy. During the discovery every node sends N packets, evenly distributed over time. This is achieved by first partitioning the discovery phase into N subslots, and then selecting a randomly chosen start time for each of them.

Since all nodes select their sending times independent of each other, there is a considerable probability that the long broadcasts partially overlap. Such collisions compromise the link-estimation performance and must be avoided. This is achieved by using the simple, yet effective trick of a carrier sense prior to the broadcast. There is still the possibility of (hidden terminal) collisions; however these collisions occur only rarely in both simulation and on the testbed. In particular this holds for the good links in the network, which we are most interested in finding and assessing during the discovery. Should the broadcast be delayed due to a busy channel, the node selects a new random starting time according to a back-off scheme. Chances for a collision right after a positive carrier sense are slightly increased if several nodes are blocked. Chances for this to happen are very small, since the duration of the discovery is selected in such a way that the channel offers ample bandwidth. A problem occurs if a nodes broadcast is blocked close to the end of the discovery phase, where the time for starting the broadcast is very limited. Therefore there is a buffer slot at the end of the discovery, for which the nodes must not schedule any broadcasts beforehand. During the start-up of the node, there is no information about the neighborhood available; in particular it is not possible to send the packets with a predefined TDMA scheme as this would require to assign and distribute a global schedule.

4 Implementation and Testbed

NoSE has been implemented in TinyOS-2.x on a Tmote Sky node, featuring the Chipcon CC2420, 2.4 GHz, packet based, low-power radio. For the communication stack, NoSE has been embedded in TinyOS's LPL layer. This LPL implementation is a packet-based implementation of the original B-MAC protocol, sending a packet strobe instead of a long preamble as is the case for a bit-stream based radio. While the basic concept remains, i.e. the sending node has to send a long packet burst in order to ensure hitting the receivers channel poll, there are some details worth noticing. On the one hand, the radio does not allow sending packets directly one after another. Instead, there is gap (1ms) between two consecutive packets, which requires a slightly prolonged carrier sense in order to check for activity on the channel, i.e. there is an increase in energy consumption for the idle listening over the model. On the other hand, the packet stream allows to receive the packet right away, without the necessity of listening until B-MAC's end of the long preamble before the packet is sent. This makes the reception of a message very cost efficient, i.e. requiring almost as little time as the regular channel poll.

The NoSE protocols needs to synchronize nodes for a simultaneous start of the discovery phase, requiring network wide 'very granular' synchronization in the order of about a second. This information is added to the data packet of the wake-up call. The packet stream (B-MAC's preamble), which can have a duration on the order of a second, adds an increasing delay as it progresses. This delay needs to be compensated for every data-packet of the stream, which has been implemented by adapting the radio buffer after receiving the radio's start-frame delimiter interrupt.

NoSE data packets, namely wake-up and discovery packets, have to be recognized as such. This is achieved by adding two user defined frame types, namely IEEE154_TYPE_WAKEUP 0b100 IEEE154_TYPE_DISCOVERY 0b101 indicating wake-up and discovery packets respectively being compliant with the IEEE 802.15.4 standard used by the CC2420 radio. Based on this information, NOSE can decide, as part of the MAC layer, whether the packet needs to be handled by itself, or if the message has to be passed to the routing layer. NoSE further maintains a state machine, shown in Figure 6 containing three states: idle, wake-up and discovery. During the idle state, regular data traffic from or to the network layer is forwarded to the appropriate interface. As soon as a wake-up packet is received, which occurs asynchronously for the different nodes in the network, NoSE switches to the wake-up state setting a timer for the start and the end of the discovery state, after which the node switches back to the idle state. In the discovery and the wake-up state, no regular data traffic will be forwarded.

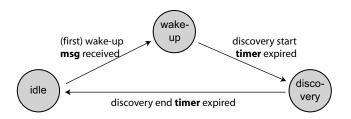


Figure 6. NoSE state chart.

Furthermore, discovery packets are only accepted in the discovery state, wake-up packets only in the idle state.

This scheme allows to run NoSE not only one single time right after the deployment, but also at a later stage during the operation, e.g. for maintenance purposes. For instance, the network can be set back to sleep, saving energy and waiting for another wake-up call. This would then allow to suspend the sensor network, for instance when replacing nodes, and restarting it at a later point in time, without adding much complexity and saving large amounts of energy. Another interesting application scenario is the possibility to efficiently rebuild an existing network topology.

During the discovery, packets are exchanged to find neighbors and to assess the link quality. The latter issue however, has to ensure, that the condition of the link assessment are as similar to the normal operation as possible. Chackeres et al. [3] showed, that the discovery beacons should have a similar packet size to the data beacons, being used during operation, for achieving a better performance in the link estimation. NoSE therefore sends a discovery beacon containing the node's identifier and also some filler data. However, instead of just adding garbage data, useful data such as the nodes battery level is being transmitted, that can be used by the routing protocol, for selecting its parent node according to the additional information.

4.1 Experimental Setup

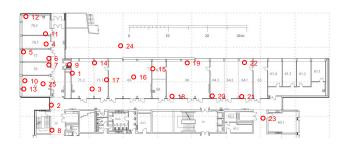
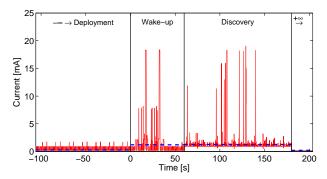
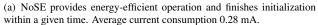
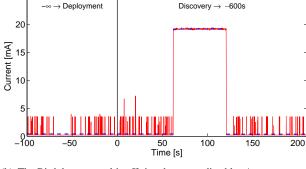


Figure 7. Testbed with 25 Tmote Sky nodes.

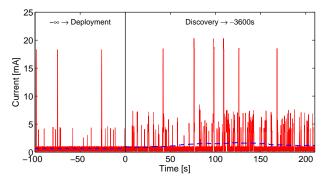
NoSE is designed to improve the effectiveness of real deployments and therefore needs to be tested in such a realistic environment, in particular not in an artificial node



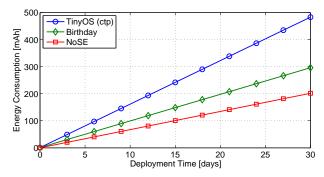




(b) The Birthday protocol is efficient but unpredictable. Average current consumption $0.41\ mA.$



(c) TinyOS-2.x (CTP) does not distinguish phases for deployment and initialization. Average current consumption 0.67 mA.



(d) Energy consumption of all three protocols as a function of the duration of the deployment phase.

Figure 8. Measured power and responsiveness analysis for different protocols during initialization.

line-up such as a grid on a tabletop. The performance evaluation has been conducted in a 25 node scenario, deployed over multiple offices as illustrated in Figure 7. The node density and location was chosen to be similar to the characteristic of the fire-detectors deployed in the building, i.e. the application scenario assumed is that of a wireless multi-hop fire-detection network initialized with NoSE.

Besides saving energy and time during the initialization, the purpose of NoSE is to find and assess all neighboring nodes. In order to check the quality of this neighbor search and estimation, it is required to have a profound knowledge of the network characteristics. In particular it is essential to know all neighbors and the according link qualities. This information however is susceptible to change. Therefore the network characteristics of the network was measured regularly over a period of six weeks, alternating with the performance evaluation of the NoSE protocol. For evaluation, 558 test runs were performed. For the link measurements, the link quality was assessed 18 times during these six weeks, for which every node sent 1000 messages randomly distributed over a period of three hours. Based on this data, two different metrics are used in the remainder

of this paper, namely the *Packet Reception Rate* (PRR) and the *Long Term PRR* (LTPRR). The term PRR reflects the snapshot of the link measurements being closest in time. The LTPRR on the other hand expresses the overall link performances over the whole six weeks, i.e. over all link measurements. Furthermore, the terminology *High-Quality Links* (HQ Links), refers to links having a (LT)PRR >95% (see also Table 2).

5 Performance Evaluation

The metrics used to benchmark NoSE's performance, evaluate three major concerns of the initialization: (1) Energy-efficiency during the deployment phase, (2) responsiveness and energy-efficiency of the initialization and (3) the completeness of the neighbor search and the quality of the link assessment. These three metrics are evaluated and compared for NoSE, the Birthday and the TinyOS-2.x collection tree protocol (CTP) which we have implemented. The Birthday protocol has been implemented according to the protocol description in [7] and evaluated with the suggested parameter allowing to find 95% of the avail-

able links. CTP is a low-power optimized adaptation of the standard collection tree protocol, running with a 100% radio duty cycle. The MAC layer has been replaced with the same LPL stack, NoSE is based on. CTP is parameterized for an optimized operational efficiency, and has proven to work efficiently and reliably in the testbed for months.

5.1 Energy Efficiency

The energy consumption during the deployment phase is most crucial in order to prevent a substantial amount of the available energy to be lost before becoming operational. The start-up behavior is analyzed measuring a single node's energy consumption, if no neighboring nodes are in communication range. This is shown in Figures 8 (a)-(c) showing the traces of the different protocols power consumption (the details sampled at 1 msec are integrated and plotted over periods of 100 msec each) during the deployment and the initialization phase. For this setup, only the last 100 seconds of the deployment phase are presented, whereas this phase usually takes orders of magnitudes longer in real scenarios. However, longer measurements showed that these trace excerpts are representative for the node's behavior.

NoSE's deployment phase shows a very regular and low energy pattern, with an average current drain of 0.28mA. This is due to NoSE's low duty cycle, i.e. sampling the channel once every second, not sending any neighbor announcement beacons. The Birthday protocol, also following this paradigm of total radio silence, shows an increased current drain of about 0.41mA. This can be attributed to a rather long channel sampling time of 20 msec waiting for a message to be received, instead of having the sender sending for 20 msec and letting the receiver perform sporadic carrier senses only. Another interesting artifact can be seen around in the interval -80 to -60 sec, where the Birthday protocol did not sample the channel for about 20 sec. This is due to the use of random sampling times sometimes allowing the nodes not to listen to the channel for a long time and in effect making the protocol highly non-deterministic. The CTP on the other hand is not following the rule of total silence and is broadcasting a neighbor announcement beacon every once in a while, resulting in an increased average current consumption of 0.67mA.

Comparing the average deployment phase's energy consumption for the different protocols shows a more than doubled current consumption for the actively messaging CTP protocol, clearly showing the disadvantage of this approach. The long term effect of this increase is illustrated in Figure 8(d), showing the CTP requiring 490mAh over a period of 30 days, i.e. 22% of the available energy of a standard single alkaline AA battery (2200mAh). NoSE on the other hand reduces this amount to 9%, i.e. a reduction by 40%.

5.2 Responsiveness

The second important aspect of the initialization is the responsiveness of the network. Ideally the network should be ready as soon as the last node is deployed and powered on. This behavior is shown in Figures 8 (a)-(c), whereas the time equal t = 0s indicates the begin of the initialization phase. In particular this means that all nodes are powered on and that the beginning of the initialization is triggered at the sink node. CTP is designed and parametrized for an efficient overall operation, at the price of a delayed responsiveness that can be seen in more detail in Figure 2(a). In particular it shows a lot of dynamics, i.e. channel switches distributed througout the network's startup which all in all, takes the about an hour to build a stable topology in our 25 node testbed. The frequent channel switches, each come for the price of a lot of control beacons, as can be seen in Figure 2(b), announcing the changes in the topology. All these beacons however, cost an order of magnitude more from the energy budget than regular unicast packets (see Figure 3), making the initialization of CTP very costly. The Birthday protocol, although having the advantage of being initialized by a trigger, requires an unknown time for discovery. Moreover, the birthday protocol lacks a link-assessment mechanism, allowing to estimate the link-quality to the different neighbors. Hence a routing protocol following Birthday's discovery, has to deal with similar issues like CTP: a huge amount of channel switches until finally finding stable links.

NoSE on the other hand features a deterministic discovery time, the length of which depends on the user-defined length of wake-up and discovery. Minimizing these two phases directly minimizes the responsiveness. Nevertheless, it needs to be ensured that this reduction in time is not jeopardizing the discovery and link assessment.

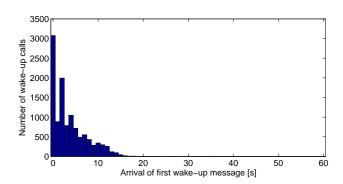


Figure 9. Time of reception of the first wakeup call after being triggered at the sink.

During NoSE's wake-up phase, all nodes have to receive the wake-up call, which is designed to be fast and reliable using a two-wave scheme. Out of all 558 runs, only in one occasion the wake-up call has not been received by all nodes in the network, showing that the two-wave scheme of a first aggressive and second slow, but reliable wave performs as expected. Concerning the delay of the asynchronous wake-up call, Figure 9 shows the delay between sending the trigger and reception of the (first) wake-up beacon. According to this illustration a wake-up time of 30 sec allows a reliable wake-up of all nodes in the network. Nevertheless a wake-up period of 60 sec is chosen, on the safer side for larger deployments. For the discovery phase, a period of 2 min has shown to be an adequate value. This aspect is further detailed in Section 5.4.1. Overall NoSE provides a well-assessed neighborhood information in just about 3 minutes; after the wake-up call has been triggered.

5.3 NoSE vs. Birthday

PRR [%]	> 95	85 - 95	50 - 85	< 50
#Links	155	45	42	75
NoSE	97.8%	88.8%	80.8%	59.3%
Birthday	97.0%	91.9%	82.5%	70.4%

Table 2. Comparison of the neighbor discovery performance. Both protocols find almost all high-quality links, but also find a substantial amount of the available low-quality links.

NoSE and Birthday both aim at finding all available neighbors. In Table 2 it is shown, how well the two perform this task, by sending 20 discovery broadcasts. The table segments the number of found neighbors according the measured link quality (PRR). Of all the available high quality (HQ) links, both protocols find almost all links available. The same holds for links with a link quality of 85-95%. However, both protocols also find a substantial amount of links, with a poor link quality. These links should not be included, emphasizing again the necessity of a link assessment prior to setting up of the routing tables.

NoSE as well as the Birthday protocol both feature high activity during the discovery phase. For the Birthday protocol, the duration differs for all nodes in the network. In particular, during discovery a node does not know about the state of the network wide discovery, it's finishing time and when to start the next step of the initialization, the set-up of the routing tables. NoSE on the other hand features a synchronous, bounded and deterministic discovery phase, allowing for a smooth transmission to the subsequent routing set-up phase. During the discovery, both protocols show an increased activity. Even though discovery time is short, the energy consumption requires considerations. In Birthday, as illustrated in Figure 8(b), the radio is always turned on, running with a 100% duty cycle. NoSE on the other

hand is using the low-power mechanism, allowing for a reduced energy consumption as indicated in Figure 8(b). In order to save time and energy, the LPL's wake-up period is reduced during the discovery. The effect of this reduction is analyzed in Figure 10, showing the nodes duty cycle in dependence of the discovery and channel-polling time, which has been measured and averaged over all nodes in the network. Despite the high communication density during the discovery, NoSE allows for a reduced duty cycle of less then 20% while having the same duration of 1 min as the Birthday's discovery phase, thus being 5 times as efficient. In Section 3.2, it has been shown, that in dependence on the discovery time and the numbers of packets, an optimal wake-up period T_W can be determined. This is also reflected in the measurements depicted in Figure 5, showing this optimum and suggesting a similar wake-up period to be used. Therefore it is possible to use the model to determine the optimum operating point for NoSE. While both protocols perform well in finding the available links, the concept of the two protocols differ greatly. In particular, the Birthday protocol's non-deterministic behaviour, the requirement of building a second radio stack for the operational phase and the lack of an integrated link-quality estimation, clearly shows its limited usability when being integrated into a system.

5.4 NoSE Link-Assessment Quality

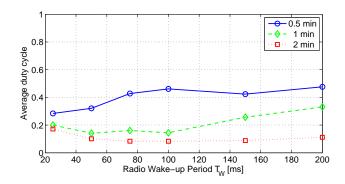


Figure 10. The energy consumption can be optimized by carefully adapting the wake-up period (T_W) of the MAC protocol during the discovery phase.

The link assessment of NoSE is a unique feature that is not found in other initialization protocols. In the following it is discussed, how long the discovery phase should be chosen to allow a good link estimation to be made. Further it is detailed how the link assessment can be improved using the radio *Radio Signal Strength Indicator* (RSSI) value.

5.4.1 Discovery Duration

NoSE link assessment is based on the knowledge that all nodes in the network send N discovery beacons. A node receiving N beacons from a particular neighbor can therefore be sure to be well connected. If only a small fraction of the possible N beacons are received, the node has to assume to be badly connected with this particular node. This assumption however, is made based that beacons did not arrive due to bad link quality and not due to collisions.

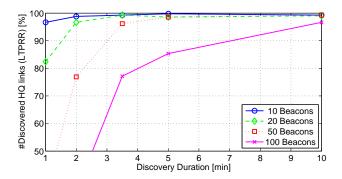


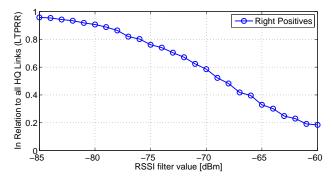
Figure 11. NoSE discovery phase: Beacon channel saturation.

The influence of collisions on the link estimation is analyzed in Figure 11. The plot shows, for how how many of the high-quality (LTPRR) links at least 90% of the N beacons were received. The plot shows that for 50 beacons, a discovery time of at least 5 minutes should be chosen. As a rule of thumb, for every 10 beacons being sent, the discovery should last an additional minute. However, this value needs to be increased for denser network deployments and also adapted (linearly) to the chosen wake-up period $T_W = 100 \text{ms}$. In the current test environment, the maximal number of neighbors of a single node is L=12. This requires that this particular node can exchange (L+1)N (N to be sent by the node itself) messages during the discovery time T_D . Using the rule of thumb mentioned above the channel utilization should not be larger than $N(L+1)T_W/T_D = 0.21$, i.e. the channel utilization should not exceed 20% during the discovery in order to ensure a well-assessed link estimation. The discovery phase can therefore be reduced if a very short wake-up period is chosen. However, a reduced discovery phase shortens the channel assessment time, making the estimation very fragile to short-term link fluctuations.

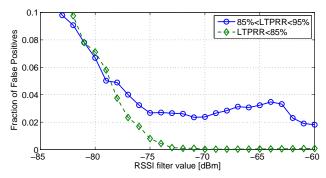
5.4.2 Tweaking the Link Assessment

As indicated, the link estimation is only a snapshot in time, raising the question how well the links are evaluated considering the long-term behaviour (LTPRR) of the links. In [12],

Srinivasan et al. indicated that the RSSI value can help estimating high-quality links, arguing that a link can be estimated based on a single packet, if the RSSI is very high. NoSE on the other hand, already has information on the current link quality after the discovery phase. However, the RSSI value is being used to minimize the number of false positives, i.e. the number of wrongly assessed high-quality links (LTPRR), which is shown in Figure 12. For this analysis, the discovery is parametrized to exchange N=20 beacons. A link is assessed having a high-quality if at least 19 of the 20 were received and if the maximal RSSI value of the received beacon exceeds a certain threshold $RSSI_{min}$, plotted against the x-axis.



(a) HQ links where assessed based on the criteria that at least 19 of 20 beacons needs to be received. The second criteria is the maximum RSSI, allowing to detail the assessment.



(b) Even though all the selected links show currently a good performance, many of these links are not stable on the long term. Using the RSSI as a second criteria allows minimizing the number of wrongly assessed links.

Figure 12. RSSI's impact on the estimation.

Figure 12(a) shows that a conservatively chosen RSSI threshold greatly minimizes the number of links found. For instance, choosing $RSSI_{min} = -80 \mathrm{dBm}$ selects 92%, whereas $RSSI_{min} = -70 \mathrm{dBm}$ will only select 59% of all available HQ-links. This poor efficiency is influenced by the fraction of wrongly assessed HQ-links as indicated in Figure 12(b). The analysis shows that choosing an $RSSI_{min}$ threshold of -73dBm, allows to be quite certain that no bad links (LTPRR< 50%) are selected. However

there is still the possibility of roughly 3% of selecting a medium link (LTPRR \leq 50%). Even by further increasing threshold, this latter fraction cannot be minimized. With a decreasing threshold on the other hand, the chances for wrong link-assessment increases rapidly. Considering that the number of found HQ-links does not substantially increase with a $RSSI_{min} <$ -80dBm, but the number of false positives significantly do. The threshold $RSSI_{min}$ value is therefore suggested to be set to -80dBm, if the application designer is looking for long-term stability in the links.

6 Conclusion

In this paper we have addressed the issue of initialization of a low-power protocols in a novel context. Although a large body of work exists on these protocols, their design, optimization and application most related work has focused on an efficient operating phase. For this operating phase, of course metrics such as the convergence and long-term stability have been studied but only few have formulated the initialization as an isolated problem and proposed substantial solutions. The work presented on NoSE in this paper is motivated by a scenario, where actual usage of the WSN is envisaged only late after the first deployment efforts have been made. Nose offers superior solutions that outperform standard and specialized approaches known in the literature by at least 50% energy savings during this deployment time and superior startup time. NoSE makes use of an adaptive protocol scheme that utilizes the right amount of resources where necessary but is highly efficient otherwise. In the presentation we have discussed a number of trade-off issues between optimal reactivity required e.g. by an interactive user and efficient and reduced usage of the energy resource. The implementation and evaluation underpins the benefit gained from NoSE and demonstrates the possibility to integrate NoSE with a number of other protocols to create true custom solutions based on the exact application requirements. Application areas for NoSE are abundant as the energy savings are evident even starting with very short deployment periods. It seems like a bonus that due to the nature of the initialization problem and the architecture of the solution offered by NoSE the assessment of the link quality and establishment of reliable neighborhood tables comes almost for free.

References

[1] M. Buettner, G. V. Yee, E. Anderson, and R. Han. X-mac: a short preamble mac protocol for duty-cycled wireless sensor networks. In *Proc. 4th ACM Conf. Embedded Networked Sensor Systems (SenSys 2006)*, pages 307–320, New York, NY, USA, 2006. ACM Press.

- [2] N. Burri, P. von Rickenbach, and R. Wattenhofer. Dozer: ultra-low power data gathering in sensor networks. In Proc. 6th Int'l Conf. Information Processing Sensor Networks (IPSN '07), pages 450–459, New York, NY, USA, Apr. 2007. ACM Press.
- [3] I. D. Chakeres and E. M. Belding-Royer. The utility of hello messages for determining link connectivity. In *Proc. 5th Int'l Symp. Personal Wireless Multimedia Communications* (WPMC 2002), volume 2, pages 504–508, Oct. 2002.
- [4] A. El-Hoiydi and J. Decotignie. WiseMAC: An ultra low power MAC protocol for multi-hop wireless sensor networks. In S. Nikoletseas and J. Rolim, editors, *Proc. 1st Int'l Workshop Algorithmic Aspects of Wireless Sensor Networks (ALGOSENSORS 2004)*, volume 3121 of *Lecture Notes in Computer Science*, pages 18–31. Springer, Berlin, June 2004.
- [5] A. Keshavarzian, E. Uysal-Biyikoglu, F. Herrmann, and A. Manjeshwar. Energy-efficient link assessment in wireless sensor networks. In *Proc. 23rd Ann. Joint IEEE Conf. Computer Communication Soc. (Infocom 2004)*, volume 3, pages 1751–1761, Mar. 2004.
- [6] F. Kuhn, T. Moscibroda, and R. Wattenhofer. Initializing newly deployed ad hoc and sensor networks. In *Proc. 10th ACM/IEEE Ann. Int'l Conf. Mobile Computing and Networking (MobiCom 2004)*, pages 260–274. ACM Press, New York, 2004.
- [7] M. J. McGlynn and S. A. Borbash. Birthday protocols for low energy deployment and flexible neighbor discovery in ad hoc wireless networks. In *Proc. 2nd ACM Int'l Symp. Mo*bile Ad Hoc Networking and Computing (MobiHoc 2001), pages 137–145, New York, NY, USA, 2001. ACM Press.
- [8] T. Moscibroda, P. von Rickenbach, and R. Wattenhofer. Analyzing the energy-latency trade-off during the deployment of sensor networks. In *Proc. 25th Ann. Joint IEEE Conf. Computer Communication Soc. (Infocom 2006)*, Apr. 2006.
- [9] J. Polastre, J. Hill, and D. Culler. Versatile low power media access for wireless sensor networks. In *Proc. 2nd ACM Conf. Embedded Networked Sensor Systems (SenSys 2004)*, pages 95–107. ACM Press, New York, 2004.
- [10] N. Reijers, G. Halkes, and K. Langendoen. Link Layer Measurements in Sensor Networks. In Proc. 1st Int'l Conf. on Mobile Ad-hoc and Sensor Systems (MASS '04), Oct. 2004.
- [11] D. Son, B. Krishnamachari, and J. Heidemann. Experimental study of concurrent transmission in wireless sensor networks. In *Proc. 4th ACM Conf. Embedded Networked Sensor Systems (SenSys 2006)*, pages 237–250. ACM Press, New York, 2006.
- [12] K. Srinivasan and P. Levis. RSSI is under appreciated. In *Proc. 3rd IEEE Workshop on Embedded Networked Sensors* (*EmNetS-III*), May 2006.
- [13] A. Woo, T. Tong, and D. Culler. Taming the underlying challenges of reliable multihop routing in sensor networks. In *Proc. 1st ACM Conf. Embedded Networked Sensor Systems* (SenSys 2003), pages 14–27, New York, NY, USA, 2003. ACM Press.
- [14] J. Zhao and R. Govindan. Understanding packet delivery performance in dense wireless sensor networks. In First Int'l Workshop on Embedded Software (EMSOFT 2001), pages 1–13, 2003.