SIDA: Self-organized ID Assignment in Wireless Sensor Networks

Jialiu Lin
Carnegie Mellon University
jialiul@andrew.cmu.edu

Yunhuai Liu and Lionel M. Ni
Hong Kong Univ. of Sci. & Tech.
{yunhuai,ni}@cse.ust.hk

Abstract

Having an ID being unique within the application domain for each sensor node is the basic assumption in many WSN applications. In traditional assignments, fixed-length ID schemes, which are severely limited by scalability, flexibility and energy efficiency, are adopted. To solve these problems, we propose a novel variable-length ID scheme that allows different nodes to have different ID lengths. To realize this idea, we propose a fully localized ID assignment algorithm called emphSelf-organized ID Assignment (SIDA). SIDA presents great potential on scalability, flexibility and energy efficiency by enabling online ID assignment and various optimizations according to different communication paradigms. Analytical and simulation results show that SIDA has a comparable static ID length while the accumulated communication overhead for multi-hop transmissions can be saved by 20%; the control overhead of the new deployment assignment can be reduced to about 30% compared with fixed-length schemes.

1 Introduction

It is impractical and uneconomical to manually assign a unique ID to each sensor node in a large-scale deployment of Wireless Sensor Networks (WSNs) [1]. Though some specific protocols such as blind-flooding are able to operate without IDs, a well-designed ID scheme can significantly enhance system efficiency and application usage such as [2, 3]. For example, carrying the source and destination node IDs in packets allows efficient routing during communication, by which a large amount of energy cost can be saved compared with blind-flooding. Ideally, after being deployed to the application field, sensor nodes should self-organize to assign themselves each a unique ID in a fully distributed and energy-efficient manner, which is the main focus of our work.

One possible approach is to give each sensor node a globally unique ID such as the Medium Access Control (MAC) address in traditional networks [1]. However, such schemes introduce too much overhead on communications. In order to guarantee globally uniqueness, the IDs have to be sufficiently long, which is not affordable in resource-constrained WSNs. For example, the Berkeley Motes have a default fixed packet size of 36 bytes. Suppose the MAC address of 48 bits is applied and packets carry only the source and destination node IDs in their header. It results in an effective payload of, at most, \((36 - 12)/26 < 70\%\) of packets. On the other hand, globally uniqueness is not always necessary. In most cases, the low-priced sensor nodes are disposed after the application task is over. Therefore, an application dependent ID scheme, which is more flexible and efficient comparing with the globally unique ones, is sufficient and more suitable for WSNs.

According to the effective uniqueness domain, current localized ID schemes fall into two categories: ID reuse scheme and field-wide unique scheme. The ID reuse scheme allows the same ID to appear in the field in different locations or time frames. The ID conflict is tackled within a certain range, such as in \(k\)-hop neighborhoods or in clusters [4–7]. Current researches restrict \(k\) to a small value such as 2 or 3 in order to avoid large control overhead when solving ID conflicts. Such schemes however, often have very limited usage for other applications than the MAC design since the ID is of less meaning outside the uniqueness domain. On the contrary, a field-wide unique ID (e.g., [8,9]) guarantees that there is no ID conflict within the whole application field. It brings many advantages as it can be used in conjunction with any other protocols such as routing, MAC and network configurations. Each node is allowed to be uniquely identified and monitored without ambiguity. In this paper, we focus our attentions on the field-wide unique ID scheme and its assignment in WSNs.

Mainly there are four design objectives for an appropriate ID scheme in WSNs. First, the ID scheme should pose the smallest ID space and thus the shortest ID length to save the energy cost of communications. Second, the ID should be assigned with acceptable establishment overhead. Although ID assignments are usually performed only once, a
smaller energy cost on control information exchange is still preferred. Third, the ID scheme should present enough flexibility to handle dynamic environments. Sensor nodes are prone to failure due to harsh environments and limited energy supply. To replenish those failed nodes, new deployments of sensor nodes are needed and may be carried out frequently. In such scenarios, the new add-in nodes should be efficiently assigned IDs without invalidating those of existing ones. Finally, the ID assignment algorithm should be in a fully distributed manner in order to be scalable.

A common feature shared by most existing ID schemes is the fixed-length format. All node IDs in the network are mandated to have the same length, which is determined by the ID space. The major limitation of fixed-length ID schemes is the lack of flexibility. Before assigning an individual node with an ID, the length of IDs should be determined. Once the length of IDs is agreed by all the sensor nodes in the network, the ID space is fixed. In other words, topology changes cannot be handled efficiently afterward. Especially when the network size exceeds the capability of the original ID spaces due to new deployment, the scheme needs to be performed again from the very beginning. In addition, in order to define a suitable ID space to guarantee the uniqueness feature of the ID scheme, certain global information such as the network size is needed [8]. As a result, a large amount of energy will be used in controlling information exchange which compromises the communication efficiency. Moreover, the fixed-length ID scheme cannot alleviate the accumulated effect on communication overhead caused by multi-hop routing. It should be noticed that different nodes may have different impacts on the communication overheads associated with the IDs. For most applications where the majority of communications are performed between sensor nodes and the sink node, a distant node will experience more transmissions to deliver a packet to the sink than a node nearby the sink. Therefore, the communication overhead of a distant node is aggravated because of the length of transmission paths.

Comparing with the fixed-length scheme, variable-length ones can effectively conquer these drawbacks. With the guaranteed network-wide uniqueness of the ID scheme, variable-length schemes have no ID space defined in advance of the assignment. Instead, they can expand the ID space in real-time when more nodes are probed and involved in the network. Therefore, they can be performed on the fly without the knowledge of how many nodes are deployed and how they are deployed. After the first-run assignment, many IDs are still available for newly joined nodes whose assignment can be done locally in a distributed manner. Since the IDs can be arbitrarily long (as long as it can be carried by a single transmission packet), the ID space can be easily extended without network-wide synchronization when new nodes are deployed. Furthermore, in order to compromise the cumulative effect on communication overhead, shorter IDs can be assigned to the distant nodes while longer IDs are assigned to those nearby nodes of the sink, which can significantly save the energy consumed on transmitting ID fields. This cannot be achieved by fixed-length schemes. The main challenge of variable-length ID scheme design is the parsing of the packets. One possible solution to the problem is to apply a prefix-free coding scheme so that the variable-length IDs can be self parsed, such as the well-known Huffman coding. However, prefix-free coding schemes have a strict ID structure which lacks flexibility either. In a highly dynamic environment, it is also a challenge to assign IDs to newly deployed nodes without invalidating the existing prefix-free architecture unless the whole ID assignment is run again, which requires a large amount of control information exchanges.

To offer a flexible variable-length ID scheme which can be parsed easily, in this paper we propose a self-organized ID assignment (SIDA). SIDA aims to optimize communications between the sensors and the sink. Its basic idea is to construct an overlay binary tree structure. Each position on the tree structure is mapped to a unique ID which is then assigned to a node. A simple example is illustrated in Fig. 1, where twelve nodes are deployed including the sink. The dotted lines represent the wireless links between nodes. When a traditional fixed-length ID scheme is applied, a space of 12 IDs has to be defined which requires 4 bits for each node ID. Using variable-length on the contrary, nodes have IDs ranging from 1 bit to 3 bits. In addition, the length of the node ID is roughly inversely proportional to their distances to the sink in order to reduce the accumulative effect of communication overhead. The shortest ID “1” is assigned to a three-hop node which forms the level 1 group (the other shortest ID “0” is assigned to the sink node). The level 2 group, whose IDs are of two bits, consists of two other three-hop nodes and two 2-hop nodes. The left nodes form the level 3 group with 3 bit IDs. In order to correctly parse the packets, a fixed-length field called Length Indicator (LI) is introduced to explicitly indicate the length of the ID. LI is placed ahead of the

Figure 1. An example of SIDA algorithm of different lengths;
variable-length part. The size of the length indicator field is determined during assignment and agreed by all nodes in the network. Although the size of length indicator also puts a constraint on the ID space and ID length, comparing with the constraint in the fixed-length scheme, it is much looser.

The major contribution of this paper is to identify the drawbacks of the fixed-length ID assignment scheme and the potential of variable-length schemes. We propose a self-organized variable-length ID assignment algorithm (SIDA) to assign IDs to every node in the network in a distributed manner. The experimental results show that there is approximately 20% percent improvement in saving communication overhead and a dramatic improvement in flexibility comparing with fixed-length schemes. The remaining of this paper is organized as follows. Section 2 gives a brief introduction to some related work in literature. In Section 3, the SIDA algorithm is described. Simulation results are presented in Section 4. Section 5 concludes this paper and suggests a number of future research directions.

2 Related Works

ID assignments in WSNs can be classified as ID reuse assignment and field-wide unique ID assignment, based on the uniqueness domain of the assigned IDs. In this section, we introduce them respectively.

2.1 ID reuse assignment

Early in 2001, Elson and Estrin pointed out the problem of large overhead caused by the ID portion in a packet in WSN [4]. They gave a clear analysis on the usage and cost of IDs and advocated the temporal and spatial reuse of IDs. The basic idea is that node IDs are distinguished in the spatial and temporal dimension, rather than in the ID space. Nodes that are far apart are allowed to share an ID at the same time, while nearby nodes share an ID at different times. Schurgers et.al turned this idea into a proactive ID assignment scheme [5, 6]. They proved that the uniqueness of node IDs within 2-hop neighborhood suffices for transmitting data from nodes to the sink. By exchanging the neighbor tables, the ID conflict within 2-hop neighborhood could be solved. To further reduce the average length of address field, the schemes encode the address by using Huffman coding. Zhou et.al further improved the idea by a reactive approach [7]. Comparing with the proactive ones, this scheme released the requirement of the embedded neighbor table in control messages. Conflict resolution was postponed until data transmission. The simulation results showed the correctness and the low overhead comparing with the proactive method. All these works however, assumed that data traffic came from the sensor nodes towards the sink in a directed diffusion communication paradigm. Ali realized ID reuse by clustering the network [10]. The scheme guaranteed uniqueness within a cluster and no address conflict between two-hop neighbors. Simulation and experimental results showed that their scheme has a great increase in efficiency comparing with the global unique address. However, the naming scheme only favored cluster-based sensor networks.

The main limitation of ID reuse schemes is that individual nodes cannot be specifically addressed. Such schemes have fairly limited usage for applications and other protocols.

2.2 Field-wide unique ID assignment

Valls et.al identified that in a sensor network every node is a potential destination [8]. Their work mainly aims at maximizing the usage of the ID space while guaranteeing the field-wide uniqueness. A tree structure was built starting from the sink in order to determine the network size. Then ID assignment responsibilities were distributed from the root to the leaf nodes to complete the final ID assignment. The mathematical analysis and simulation results showed that the algorithm has a good performance in terms of the high success rate and short execution time. The main drawback of the scheme is the requiring for the knowledge of the network size in the initial phase. Due to the large scale of the network and harsh application environment, sensors are often deployed in an asynchronous, autonomous manner (e.g., by unmanned aerial vehicles [11]). They may wake up at different time and go back to the sleep mode in order to save energy. It is still a challenging job to obtain the precise information of the network scale. Moreover, Vall’s work assumed that the assignment was one-off without considering the maintenance and recovery. Changes in the topology (e.g., due to the node failure and new deployment) will introduce a large amount control overhead as the re-assignment in the whole network field is inevitable.

In our previous work, we also proposed a field-wide unique ID assignment algorithm [9]. It was based on a virtual coordinate system that partitions the network to many virtual cells. Each cell has a unique ID according to its
3 Self-Organized ID Assignment

3.1 Basic concepts

Final SIDA IDs are composed of two parts, the fixed-length field $LI(v)$, and the variable-length field $PID(v)$, as illustrated in Fig. 2. The length indicator $LI(v)$ used for parsing indicates the length of the $PID(v)$ field, while the $PID(v)$ field uniquely identifies each node. The assignment of SIDA IDs has three phases which are associated with $IID(v)$ (Initial ID), $TID(v)$ (Temporal ID) and $PID(v)$ (Permanent ID), respectively. In the first phase, each node generates an $IID(v)$ to allow control information to be exchanged between neighboring nodes. Once the $IID(v)$ is available, $TID(v)$ is assigned to each node. In the second phase, the maximal length of $TID(v)$ over the network is collected to the sink and broadcasted to the whole network. In the last phase, $PID(v)$ the permanent ID is assigned based on the $TID$ and the maximal length of $TID$. An example of these phases is illustrated in Fig 3.

When $PID$ is ready, the content of the $LI$ field is calculated by $Val(LI(v)) = Len(PID(v))$. Upon receiving a data packet, applications should first read in the length indicator $LI(v)$ to understand the structure of the packet. Applications then can present a correct parsing. In later evaluation section we consider the whole ID format as a union including both $LI(v)$ and $PID(v)$. For convenience, in the following sections, we present the wireless sensor network as a directed graph $G(V,E)$, where $V$ is the set of nodes and $E$ is the set of edges. The unit disk model is assumed to be valid that is, there is a link $e = (u,v)$ between node $u$ and node $v$ as long as the Euclidean distance $d_{u,v}$ between $u$ and $v$ is less than the transmission range $T_r$.

We further assume wireless links are bi-directional and reliable. This can be achieved by removing the single directional links plus providing a simple acknowledgement mechanism. The overlay pseudo binary tree structured during assignment is denoted as $T(V,E_T)$. We use $parent(v) \in V$ to denote a node vs parent node and $children(v) \subseteq V$ to denote the set of its children nodes in the tree structure. Let $H(v)$ be the number of hops along the shortest path from node $v$ to the sink in $G(V,E)$. Notations are summarized in Table 1.

3.2 Initial ID assignment

At the very beginning, in order to allow information exchange between neighboring nodes, we assign an $IID$ to each node, so that neighboring nodes can be uniquely identified and communicated with. Towards this end, an ID of two-hop uniqueness is sufficient. In our work, we randomly generate a 10-bit long string as the $IID$. The 10-bit length constrains the ID conflicts which can be detected by beacon message exchange in a low probability. Once a conflict oc-
3.3 Maximal TID length determination

Upon receiving the TID, a node should determine whether it has the maximal length within its one-hop neighbors. This can be done by a single beacon message or by overhearing others TID distribution messages. As along as a node holds the maximal length within adjacent areas, it would initiate a report message which contains its length and sends it back towards the sink. The report message can be aggregated along the path and finally the sink can determine \( \max(\text{Len}(\text{TID})) \). The sink then disseminates an advertisement message through a simple network wide broadcast to notify the whole network the value of \( \max(\text{Len}(\text{TID})) \). From \( \max(\text{Len}(\text{TID})) \), nodes can determine the length of the length indicator, i.e.,

\[
\text{Len}(\text{LI}) = \log_2(\max(\text{Len}(\text{TID})))
\]

which is identical within the network field.

3.4 Permanent ID conversion

Once both TID and \( \max(\text{Len}(\text{TID})) \) are available, the conversion of P.ID(v) is ready. Roughly speaking, the major task in this phase is to reverse the length of IDs. Specifically, a long TID should be converted to a short P.ID and vice versa. The main challenge during this operation is to preserve the uniqueness. We perform the conversion based on the following procedures. We define that for two given nodes \( u \) and \( v \), \( \text{ID}(u) \) is larger than \( \text{ID}(v) \) when \( \text{ID}(v) \) is longer. If they are of the same length, then the numerical values are compared. This definition is expressed as:

\[
\text{ID}(v) > \text{ID}(u) \iff (\text{Len}(\text{ID}(u)) > \text{Len}(\text{ID}(v))) \lor \\
(\text{Len}(\text{ID}(u)) = \text{Len}(\text{ID}(v))) \land \\
\text{Val}(\text{ID}(u)) > \text{Val}(\text{ID}(v))
\]

We arrange the IDs in the ID space in an ascending order. The whole ID space has \( N = 2^{\max(\text{Len}(\text{TID}))} + 1 - 3 \) available IDs (we do not allow an ID of zero length and 0 is reserved for the sink). The one in the complement position in the space will be assigned to the node during the conversion from TID to PID. More clearly, suppose a node \( v \) has the TID(v) = “\( b_0b_1\ldots b_l \)” as variable lengths are allowed, \( v \) will be ranked in the ID space at the position: \( \text{rank}(\text{TID}(u)) = 2^{l+1} - 1 + \text{Val}(“\( b_0b_1\ldots b_l \”)) \). The complement position of \( v \), supposing it is node \( u \) with TID(u) = \( N - \text{rank}(\text{TID}(u)) \), will be assigned to \( v \) as P.ID(v), i.e., P.ID(v) = TID(arg\{u|TID(u) = N – \text{rank}(\text{TID}(u))\}).

In essence, since the complement positions are symmetric, nodes \( u \) and \( v \) actually exchange their TIDs in
the last phase. Notice that the sink is a special case: its PID will not change in this phase and its complement position will be considered as reserved. All computations of phase 3 can be done locally without any control message exchange. The LI field is also calculated at this stage: \( \text{Val}(LI(v)) = \text{Len}(LI(v)) \).

3.5 Maintenance

Maintenance is another key issue in SIDA. Since battery replacement is almost impossible in reality, deploying new nodes into the network is a common way used to recover the network topology. The newly deployed nodes can substitute the defunct ones or cover the uncovered regions. In order to handle the dynamic topology, SIDA also provides a scheme to help with maintenance of the ID assignment.

We employ soft-state mechanism to recycle node IDs. More specifically, when no message has been heard from node \( v \) for a period of time, node \( v \) is regarded as defunct. parent(\( v \)) will accordingly recycle PID(\( v \)) for future use. When a new node \( u \) joins the network, the adjacent nodes within 2-hop neighborhood will assign the recycled PID, if they have any, to \( u \). Otherwise the closest node \( w \) who has less than two children will assign its TID appended by one bit to node \( u \) and also inform \( u \) of the max(Len(TID)). Then node \( u \) can figure out its PID(\( u \)) according to the two pieces of information.

Two possible scenarios may happen during maintenance. One is that the ID length of newly deployed node \( u \), TID(\( u \)), exceeds max(Len(TID)). In this situation, \( u \) will report its own Len(TID(\( u \))) to the sink as the new max(Len(TID)) and the sink will disseminate this information again. The PID of each node needs to be recomputed accordingly. The other scenario is that the length of TID(\( u \)) even if it exceeds the range of the current LI can be present. In this case, the length of LI will be calculated again as well.

3.6 Analysis

In this chapter, the correctness in terms of the uniqueness of IDs will be proved. Performance of the best case of the SIDA algorithm will be presented as well.

3.6.1 Field-wide uniqueness

Theorem 3.1 The nodes that have different positions on the overlay tree must have different TIDs.

Theorem 3.2 The nodes with different TIDs must have different PIDs.

Theorem 3.3 The ID assigned by SIDA is network-wide unique.

Due to the space limitation and the simplicity, we omit the proof of these theorems.

3.6.2 Communication overhead

We define the communication overhead \( \varepsilon \) as the total bits used to transmit the ID fields of packets in the network if every sensor node sends one packet towards the sink, i.e.,

\[
\varepsilon = \sum_{v \in V} \text{Len}(\text{PID}(\text{sender})) + \text{Len}(\text{LI}(\text{sender})) + \text{Len}(\text{PID}(\text{sink})) + \text{Len}(\text{LI}(\text{sink})) \times H(\text{sender})
\]

\( \varepsilon \) measures the communication overhead associated with the ID fields in the network. The best case of SIDA is that a perfectly complete balanced binary tree can be established, so that there is no waste in the ID space. Since \( H(v) \) is topology dependent, we average the best case results over 10 different topologies while network density is fixed at 6. The numerical results are shown in Fig. 4.

4 Performance Evaluation

In this section, the performance of SIDA under different simulation settings will be presented in order to study the impact of network size and network density on the communication overhead as well as control overhead.

4.1 Experimental setups

We generate topologies by BRITE v2.1b [12] with a random distribution over a square region. One sink is also randomly placed in the network field. No collision or congestion is considered. Network size varies from 1000 to 10000.
and the network density varies from 6 to 16, which are the two key parameters in the settings. Data are averaged over 10 independent runs with the same parameters. We measures two metrics during performance evaluation.

1. Communication overhead $\varepsilon$, which is defined above;
2. Control overhead $\rho$: It refers to the number of total control messages needed during the assignment, including the initial assignment and the latter maintenance.

### 4.2 Static comparisons

We compare SIDA with the fixed-length ID assignment proposed in [8]. Fig. 5 shows the ID length distribution of the both algorithms. For a fixed-length ID assignment, the length of all IDs are identical, which is equal to $\log_2(\text{network size})$. By SIDA, the shortest length is always $1 + \text{Len}(LI)$, and the longest length is approximately $\log_2(\text{network size}) - 1 + \text{Len}(LI)$. When the network size is small, the advantage of SIDA is not very evident. This is because the length indicator dominates the ID field, which compromises the saving of bits in the variable-length part. When the network size becomes larger and larger, the size of the length indicator increases. However, the increasing speed of the size of length indicator is not as fast as the increasing speed of the variable-length part. In other words, the portion of the length indicator becomes smaller and smaller. In this way, the advantage of SIDA becomes more and more visible.

### 4.3 Impact of network size

Fig. 6 shows the communication overhead $\varepsilon$ of the SIDA algorithm in terms of total ID bits consumed in the ID fields. In this set of simulations, network densities are fixed at 6; which means each node has 6 neighbors on average. Along with the increment of the network size, the advantage of SIDA becomes more and more evident comparing with Ould-Ahmed-Valls method. The benefit comes from the variable-length scheme and the approach of allocating distant nodes with shorter IDs. Assigning the shortest ID to the sink also improves the performance. When the network size reaches 10000, SIDA can outperform Ould-Ahmed-Valls method by more than 20%. At the same time, the results from SIDA is very close to the ideal case, which means that the binary trees built are close to a balance binary tree.

Fig. 7 plots the average number of control messages needed in different network sizes. It shows that the control overhead has approximately a linear relationship with the network size. The slop of the curve is about 4, which means every node needs to transmit approximate 4 messages during the assignment procedures on average. The control overheads of SIDA and Ould-Ahmed-Valls method are almost at the same level. Under the situation of having similar control overheads but less communication over-
head, SIDA outperforms Ould-Ahmed-Valls method as the network size increases.

4.4 Impact of network density

Fig. 8 and Fig. 9 plot the relationship between communication/control overheads and network density. The performance of SIDA improves in terms of less communication overhead when network density increases. This is because the larger the network density, the easier it is to achieve balance, and this reduces the waste of ID spaces. While the average number of neighbors increases, more control messages are involved in the two-hop neighborhood during children-parent selection. This explains the increase of control overheads along with network density.

5 Conclusions

In this paper, we studied the node ID assignment related issues in WSNs. Due to the lack of flexibility, traditional algorithms of fixed-length IDs assignment usually have a large control overhead during assignment and are hard to reduce the cumulated communication overhead in multi-hop routing. Motivated by these observations we propose a novel variable-length ID scheme that allows different nodes to have various ID formats. To realize such an idea we design and implement a fully localized and distributed ID assignment algorithm called SIDA. The major distinctive feature of SIDA is the flexible ID format which enables online assignment and maintenance without field-wide knowledge. By mathematical proof we show that SIDA holds the field-wide uniqueness feature which is needed by many applications. Simulation results show that SIDA significantly outperforms the existing fixed-length ID assignment in terms of the cumulated communication overhead while keeping the control overheads at a similar level.

Our future work will focuses on three aspects. First, SIDA is based on the unit disk model. Its performance in real world settings is still unknown and needs further investigations. Second, SIDA is mainly designed for the communications between sensor nodes and the sink. Different communication paradigms may have different requirements and optimal ID schemes. Finally, we need to note that SIDA does not minimize the communication overhead or control overhead although it attempts to reduce both of them. There is still ample room for future study. More research can be done to find other variable-length encoding methods which could further reduce the communication overhead while keep the property of easy parsing.

ACKNOWLEDGEMENTS

This research was supported in part by Hong Kong RGC Grant HKUST6183/06E, the Key Project of China NSFC Grant 60533110, the National Basic Research Program of China (973 Program) under Grant No. 2006CB303000, the Key Project of Guangzhou Municipal Government Guangdong/Hong Kong Critical Technology Grant 2006Z1-D6131 and the HKUST Nansha Research Fund NRC06/07.EG01.

References