NETWORK PLANNING TOOL WITH TRAFFIC-ADAPTIVE PROCESSING FOR WIRELESS SENSOR NETWORKS

José Cecílio and Pedro Furtado
University of Coimbra, Coimbra, Portugal

Keywords: WSN, Network Planning, Measurement, Performance, Reliability.

Abstract: Some applications of Wireless sensor networks (WSNs), especially in industrial sense and react scenarios, require fairly fast sampling rates. Considering that a few sensors may share a common sink, sharing part of their path on the way to the sink may result in undesirable message losses and delays that cannot be solved without modifying data communication rates. Our research focuses on planning a WSN to avoid excess traffic during sensing and acting to guarantee the minimal delay for critical scenarios. In this paper we propose an integrated approach to plan, test and reconfigure a network. Initially, our approach gives guidance for a base-plan for the network. With this first-cut plan we test the performance of the network and if necessary reconfigure it. The results of tests given are followed by traffic-level adjustments of the system by several possible techniques: adjustment of number of nodes, network partitions, reduction of the sampling rate or in-network processing with strategies such as aggregation techniques or in-node closed control loops. We evaluate experimentally the proposed approach with two different mechanisms of communication, and different levels of traffic, showing that our planning and reconfiguration allows users to make the best choices for the application context.

1 INTRODUCTION

When deployed in an industrial setting for monitoring-and-control applications, latencies and message losses can become paramount, as some industrial applications need high sampling rates and may apply closed loop control. Then the question arises as whether one can provide guarantees to urgent message delivery. One way to try to provide added guarantees is to deploy a WSN network with real-time specific algorithms that would include at least completely pre-planned synchronous time-division mechanisms. There is extensive research into synchronous time-division protocols for WSN. Among all the work, protocols based on time division multiple access (TDMA) attract much attention, since TDMA inherently avoids collisions, the primary factors causing message loss. TDMA-based protocols are very efficient when network link traffic is kept below a certain level, resulting in neglectable packet losses, where the whole system will guarantee delivery for messages. However, CSMA-based protocols are more flexible for dynamic networks, where data traffic generated at each node can change over time.

One issue to deal with is how to keep network traffic below a certain limit, in order to guarantee minimal losses and delays. We start by defining a simple network planning approach assuming a planned deployment of nodes. Then we provide a tool for users to test the deployment concerning metrics such as packet and message losses and latencies. If the test results in non-conformance to user-dictated application requirements, there is a need to modify some parameter(s) and to re-test the solution until it conforms to requirements. Perhaps it is absolutely necessary to decrease the sensing rate, remove nodes from the network (or to create network partitions), or perhaps the sending rate, if some extra delay can be tolerated.

The proposed approach consists on creating a planned network by taking into account a set of constraints. The approach includes: a Network Planning algorithm to deal with high-rate sampling; a Network Status module offering simple network status tests that provide information to the user, such as the message loss ratio, delays and the degree of difficulty in delivering application-level messages; a module that provides commanding / reconfiguring the WSN to adjust the system until the network

Cecílio J. and Furtado P..
NETWORK PLANNING TOOL WITH TRAFFIC-ADAPTIVE PROCESSING FOR WIRELESS SENSOR NETWORKS.
DOI: 10.5220/0003815600590064
In *Proceedings of the 1st International Conference on Sensor Networks* (SENSORNETS-2012), pages 59-64
ISBN: 978-989-8565-01-3

status tests are satisfied.

The rest of the paper is organized as follows: section 2 discusses related work; section 3 presents the tool for planning, reconfiguring and evaluating the network performance; section 4 shows the experimental results obtained for TDMA (schedule-based) and X-MAC (contention-based) protocols and section 5 concludes the paper.

2 RELATED WORK

In this section we first review related work on MAC protocols and its adaptation, planning and monitorization. One important key issue in WSNs that influences whether the deployed system will be able to save battery power or to handle high sampling rates gracefully is the MAC protocol and its configurations. WSN MAC protocols can be classified into two main families or their combination: Carrier Sense Multiple Access (CSMA), and Time Division Multiple Access (TDMA). TDMA protocols will schedule the activity of the network in a period in which all nodes will be active. In the idle times between data gathering sessions, nodes can turn off the radio interface and lie in a sleep state. Thus, the main and most important advantage of TDMA is time critical and low power consumption. There are innumerous works addressing TDMA protocols. Several protocols have been designed for quick broadcast/convergecast, others for generic communication patterns. The greatest challenges are the spatial-reuse of the time-slots, interference avoidance, low-latencies, and energy-efficiency.

SS-TDMA (Kulkarni) is a TDMA protocol designed for broadcast/convergecast in grid WSNs. The slot allocation process tries to achieve cascading slot assignments. Each node receives messages from the neighbours with their assigned slots.

In RT-Link [Rowe] and PEDAMACS (Ergen) protocols, time-slot assignment is accomplished in a centralized way at the gateway node, based on the global topology in the form of neighbour lists provided by the WSN nodes.

CSMA protocols may be suited for event-driven WSN applications with dynamic topologies. Some protocols such as S-MAC, B-MAC, WiseMAC and X-MAC are frequently used in WSN. S-MAC (Ye) defines periodic frame structure divided into two parts, with nodes being active in the first fraction of the frame and asleep for the remaining duration. The length of each of the frame parts is fixed according to the desired duty-cycle.

B-MAC (Polastre) and WiseMAC (El-Hoiydi) are based on Low-Power Listening (LPL) (Polastre) that is a very simple mechanism designed to minimize the energy spent in idle listening. X-MAC (Buettner) also is based on Low-Power Listening but reduces the overhead of receiving long preambles by using short and strobed preambles allowing unintended receivers to sleep after receiving only one short preamble and the intended receiver to interrupt the long preamble by sending an ACK packet after receiving only one strobed preamble.

Monitorization tools can be used to evaluate network performance. It is necessary to have information on the fraction of packets and application messages losses, latencies and other simple metrics that provide enough information about the network health. It is also necessary to have alternative in-network processing approaches.

We have reviewed existing tools to monitor network health (SNMS, SNIF, DiMo), and designed our own simplified tool adapted to our planning objectives. Our current tool does not include some important parts that the reviewed tools do include and which we plan to add later, such as node failure detection mechanisms.

To avoid congestion, our work uses in-network data processing. Data processing in sensor networks has been studied extensively, and in-network processing is the general term used for techniques that process data on a node or group of nodes before forwarding it to the user.

Our study is related to these ones in that in-network data processing approaches are used to decrease the amount of communication that is needed. But in our work these approaches are part of an integrated system, with guaranteed delivery of messages with minimal loss, and monitorization and configuration to reduce network traffic to acceptable levels.

3 NETWORK PLANNING

In this section we devise the basic network planning approach, considering schedule-based with a fixed topology and simple slot-based planning. We also discuss why this approach can also be a basis for a first-cut plan on contention-based protocols, although more testing is required in that case to determine whether the system will behave gracefully with strict sampling rates.

3.1 Planning for Schedule-based Protocols

TDMA protocols create a schedule for network activity: each node is assigned at least one slot in a time frame, which is considered to be the number of slots required to get a packet from each source to the sink. In this work we consider a simple protocol with a global network frame where all nodes have the same length in the frame. The time axis is divided into fixed-length base units called epochs. Each epoch is subdivided into *k.n* time slots, where *k* is the number of slots required for successful transmitting a message and *n* is the number of nodes in the network. There are various techniques to determine the value of *k* depending on application requirements. We used *k=3* slots to provide added guarantee for delivery of messages. The first slot is used to send a message, the second is reserved to receive an acknowledgment from receiver and the last slot is used to retransmit a message if the node doesn't receive an ack. The slot assignment of each node can be pre-configured, as a planned deployment is assumed. Each node is identified by a unique identification number *i* and can only transmit any message on the assigned *ki* slot. Figure 1 displays an example for two nodes (mote 1 and 2) in a star topology.

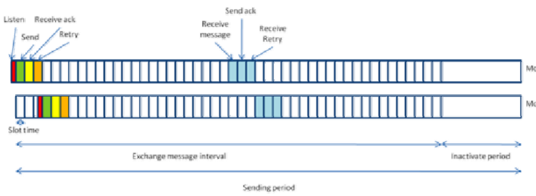

Figure 1: Slot usage for Mote 1 and 2.

One important issue on planning a deployment to use a TDMA protocol is related with the time slot. The slot size should be as small as possible to reduce the epoch size and consequently the end-to-end delay.

To determine the slot time, we take into account the following times:

- Time to transfer a message from the MAC layers data FIFO buffer to buffer of radio transceiver (t_{ts});
- Time to transmit a message (t_{xm});
- Time a receiver needs to process the message and initiate the transmission of an acknowledgment message (t_{pm});
- Time to transmit an acknowledgment (t_{xa});
- Time to transfer and process the acknowledgment from the radio transceiver and to perform the associated actions for received/missed acknowledgment (t_{pa}).

Also, a small guardian time is required at the beginning and end of each slot to compensate for clock drifts between nodes (t_g). Thus, the minimum size of a transmission slot is given as:

$$T_{st} = t_g + \max\{(t_{ts} + t_{xm}), (t_{pm} + t_{xa}), t_{pa} + t_g\} \quad (1)$$

The information content of messages and sampling rate should be planned as soon as epoch and slot assignments are known.

The epoch size (including a built-in inactivity period) defines the minimum delay of messages, and a maximum packet size defines the maximum information length of a single message.

Based on a set of network constraints that can be defined by the user, it is possible to determine adequate values for parameters such as network size or minimum delay, depending on which variable is to be determined.

Minimum Sending Period: The minimum sending period is the minimum time between two messages are sent by a node and it depends on the network size (*n*), the slot time ($slot_{time}$) and the number of slots reserved for each node (*k*):

$$\min\{sending_{Period}\} = t_{Node} \cdot n \quad (2)$$

Where t_{Node} represents the time needed by each node and it is given by: $t_{Node} = k \cdot slot_{time}$.

The maximum sending rate is the inverse of this value. Likewise, the maximum network size is the maximum size of the network measured in number of nodes. Given a sending rate, this value is determined as a function of the minimum sending period:

$$n_{Max} = \frac{sendingPeriod}{slot_{time} \cdot k} \quad (3)$$

The maximum delay measures is the time taken from the moment when a sample is collected to the moment when it is delivered to the client. This measure includes the time it takes for a sample to wait for the next sending instant.

$$\max\{Delay\} = Epoch_{Size} + TransmissionTime + ReceiveTime \quad (4)$$

Where epoch size is the time needed to do complete a cycle in schedule time line. It is defined by:

$$Epoch_{Size} = n \cdot k \cdot slot_{Time} + inactivityTime \quad (5)$$

We can also estimate the closed-loop maximum delay. The closed-loop maximum delay measures the amount of time from the instant when a sample is collected to the moment an actuation is enacted that already carries a response to the sampled value. The closed loop control can be applied on motes or in client workstation, depending where it is applied, we can define the maximum delay as:

$$\max\{D_{Closedloop}\} = ComputationTime + w \cdot (2 \cdot \max\{Delay\} + ComputationTime_{Client}) \quad (6)$$

Where w indicates the place where closed loop control is applied. This variable is a binary variable. If $w=0$, closed loop control is applied on mote, otherwise closed loop control is applied on client workstation.

Other parameter that can be estimated is the radio duty cycle of each node. It is measured as:

$$Radio_{Dutycyle}[\%] = \frac{k \cdot Slot_{Time}}{Epoch_{Size}} \quad (7)$$

The above planning is a simple way to plan a flat wireless sensor network. If we are interested to plan a tree network with several levels each epoch should be subdivided into

$$\sum_{l=1}^{N_{Levels}} \left(\sum_{i=1}^{n_{Nodes_{Level=l}}} \left(k \cdot n_{descendent|i,l} + k \right) \right) \quad (8)$$

time slots. Where:

- N_{Levels} is the number of levels used by the topology.
- $n_{Nodes_{Level=j}}$ is the number of nodes that exists in level j.
- $n_{descendent|i,j}$ is the number of nodes below level j connected to node i.
- k is the number of slots required by each node.

In this case, the minimum sending period is given by:

$$\min\{sendingPeriod\} = \min\{Epoch_{Size}\} = \sum_{l=1}^{N_{Levels}} \left(\sum_{i=1}^{n_{Nodes_{Level=l}}} \left(t_{Node} \cdot n_{descendent|i,l} + t_{Node} \right) \right) \quad (9)$$

3.2 Planning for Contention-based Protocols

While planning for schedule-based protocols can be based on deterministic formulas, contention-based protocols have to deal with probabilistic collisions and other network factors. In this work we consider the X-MAC as an example of contention-based protocol, and apply a planned topology similar to the one present in the previous section. X-MAC uses a strobe preamble that consists of a sequence of short preambles prior to DATA transmission, as illustrated in figure 3.

These short preambles indicate that a node has data to transmit. Assuming a star topology and that the sink is always active, when a receiver receives a strobe, it sends an acknowledgment to the sender which indicates that the transmission can start. When the sender receives this ack, it stops to send strobes and starts transmitting the message immediately.

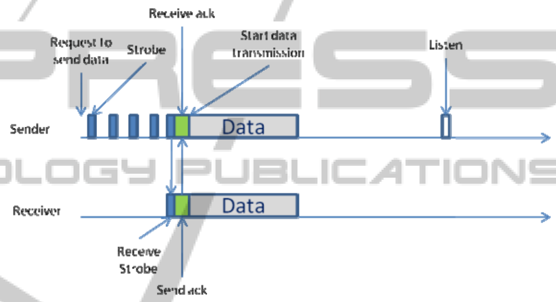

Figure 2: Operation of X-MAC.

After transmission concludes, the sender goes to sleep until it has new messages to send or until the next listen period. This listen period is used to check if another node (the sink) wants to send messages to the node. In our prototype we limit the X-MAC period to the same slot time used on schedule-base protocol. Since the sink is always on, we assume that when a node wishes to communicate to send values, it only needs to send a small strobe and gets an ack back immediately from the sink (single hop and sink always with power). As a first-cut approximation, our planning assumes no collisions (best-case scenario). With this scenario, exactly the same slot-based logic used in schedule-based planning is assumed. We assume that each node will transmit in a slot, sending a small probe, getting ack, sending the data packet, getting an ack and possible sending retry. This planning will result in the following timings:

- Time to transmit a strobe (t_{xs});
- Time to receive and process a strobe (t_{rs});
- Time to transmit an acknowledgment (t_{xa});
- Time to transmit a message (t_{xm});

- Time a receiver to process the message and initiate the transmission of an acknowledgment message (t_{pm});
- Time to transmit an acknowledgment (t_{xam});
- Time to perform the associated actions for received/missed acknowledgment (t_{pa}).

Thus, the minimum time per node is given as:

$$T_{node} = t_{xs} + t_{rs} + t_{xa} + t_{xm} + t_{pm} + t_{xam} + t_{pa} \quad (10)$$

The above plan is assuming no collisions, which is an acceptable assumption when there is little congestion in the medium. If the collision probability is significant, due to high sampling rates, the planned slot time per node should be multiplied by a factor *α*, where *α* is a percentage of slot time increase that accommodates backoff periods due to collisions. For instance, if each node takes 21ms, this is multiplied by 1.04 to assume an average 4% increase in total slot time.

To determine the percentage of increase we consider the probability of any node transmitting in the same slot time as another node. We use the following expression to determine that:

$$\frac{t}{T} \cdot (n-1) \cdot \frac{t}{T} \quad (11)$$

Where *t* is the total time reserved for each node, *n* is the number of node and *T* is a Period (sending period).

For example, if each node occupies 10% of the period, and there are 10 nodes, the overhead increase will be 9%. So we assume a basic slot-like plan and we add 9% to the period to better account for collisions.

For contention-based protocols also it is possible to determine adequate values for parameters such as:

- The maximum delay

$$\max\{Delay\} = Strobe_{Time} + backoff_{Time} + TransmissionTime + ReceiveTime \quad (12)$$

Where $Strobe_{Time}$ is the time needed to receive an ack from receiver and $backoff_{Time}$ is the time needed to start the transmission of the message.

- The closed-loop maximum delay can also be estimated by (6) where Delay is given by expression (12).

3.3 Assessing and Reconfiguration

Network planning determines a fist-cut layout of the network. After network planning and tests, reconfiguration may be necessary to improve the performance of the system. The user can change configuration parameters until the desired characteristics are obtained. In order to do this, the reconfiguration requires a set of configurations that should be used in the successive test procedure, until the desired characteristics are obtained.

The reconfiguration module collects information provided by network status, considering metrics such as message loss ratio, delays, number of fail ack and battery consumption. If any metrics fail to provide desired guarantees it is necessary to reconfigure the system. The reconfiguration module allows changing the number of nodes and network partitions, decouple the sending rate from the sampling rate and join few reading into same packet or perform aggregations (Summarize information).

4 EXPERIMENTAL EVALUATION

In this section we report the evaluation of our approach using schedule and contention based protocols. The objective is threefold: to test network planning, to characterize and compare alternative protocols and configurations under different traffic conditions; to show that excessive traffic intensity is promptly characterized by the tests and test reconfiguration works.

In our approaches we are interesting to give a tool for planning a network to lead with high rate and give guarantees of alarm detection. With the tool we can plan a network with schedule-based and contention-based protocols.

In tests we used a slot time equal to 7ms, corresponding to 21 ms of the period reserved to a single node. Assuming the sampling rate and sending rate are the same, if a node needs 21ms, and we deploy a network with 32 nodes, we conclude through eq.(3) that the minimal guaranteed delay is 672 ms and the minimal period is equal to that too.

In figure 3 we show the message loss ratio obtained during the tests of our network planning tool. The results were obtained for contention-based and schedule-based protocols with different sampling rates.

Message loss ratio is one of our key measures that is used as an indicator of excess traffic conditions. In figure 3 we observe high message loss ratio for high traffic rate (sampling rate below 800 ms). During the plan, our tool gives 672ms as minimum sampling rate. When we can verify through the figure 3, our planning is correct, because if we used sampling rates less than minimum, many messages are lost. Therefore if we consider a sampling rate greeter than 672ms, who corresponds to a one reading per slot time period, all messages are successful delivery.

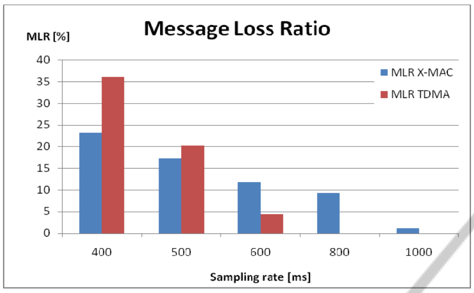

Figure 3: Message loss ratio with X-MAC and TDMA protocols.

When we use a contention-based protocol (X-MAC), the minimum sampling rate obtained to schedule-based protocol is insufficient to guarantee minimal losses. In our planning tool we added a tolerant factor (more 25% of needed time) to prevent collisions. In figure 3 we can also verify that is factor is important when we lead with high rate. For low sampling rates, X-MAC protocol have little losses with good performance, but TDMA is better with losses equal zero.

Based on these results and in the planning if a user wants to increase the sampling rate to 500ms, he has to create partitions on the network or reduce the number of nodes using a single network only. Applying the planning tool is obtained for a schedule-based protocol 18 motes as limit. Who is needed a tolerant factor for contention-based protocol the limit is less.

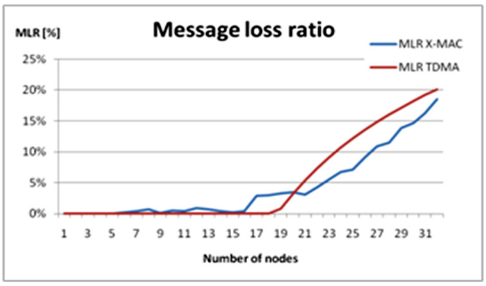

Figure 4: Evaluation of message loss ratio versus number of nodes.

Figure 4 shows the influence of adding nodes in the network performance. The test was run with 500 ms of sampling rate. The test starts with a single mote and after 15 minutes we add more one node and store the influence of then in the message loss ratio.

The results show an increase in the message loss with the increase in the number of active motes. When the number of nodes is less than 18 (number given by eq.(4)), the message loss is zero for schedule-based protocols, but for contention-based protocols is closed zero until get 16 motes. If the number of nodes is greater the message loss increase significantly. When the number of nodes is greater than 22, the message loss becomes critical (higher than 5%).

5 CONCLUSIONS

In order to make WSNs more reliable in practical contexts with constraints such as high sense and control rates, there is a need for approaches to help a user to correctly plan a WSN. Sensor networks tend to have congestion problems at high sampling rates, which may endanger the timely delivery of messages. In this paper we have proposed a tool to plan and test a WSN. Our approach allows to configure the network traffic (taking into consideration application requirements such as sampling rate, maximum delivery delay and whether the data should be detailed or can be summarized) and measure their performance.

We proposed a module to plan, configure and reconfigure the network. Our experimental results study the traffic characteristics of the approaches under different conditions, and to conclude on their usefulness.

REFERENCES

Kulkarni, S. S. and Arumugam, M., "SS-TDMA: A Self-Stabilizing MAC for Sensor Networks", *Sensor Network Operations*, IEEE Press (2005).

Rowe, A., Mangharam, R., Rajkumar, R. Rajkumar, "RT-Link: A Global Time-Synchronized Link Protocol for Sensor Networks", *Elsevier Ad hoc Networks, Special Issue on Energy Efficient Design in Wireless Ad Hoc and Sensor Networks*, 2007.

Ergen, S. C. and Varaiya, P., "PEDAMACS: Power Efficient and Delay Aware Medium Access Protocol for Sensor Networks," *IEEE Trans. Mobile Comput.*, vol. 6, July 2006.

Polastre, J., Hill, J., Culler, D., "Versatile low power media access for Wireless Sensor Networks", in *ACM SenSys*, 2004.

Buettner, M., Yee, G. V., Anderson, E. and Han, R., "X-mac: A short preamble mac protocol for duty-cycled wireless sensor networks", in ACM Sensys, 2006.