Poster Abstract: A MAC Protocol for Industrial Process Automation and Control

Petcharat Suriyachai, James Brown, Utz Roedig

InfoLab21, Lancaster University, UK
{p.suriyachai, j.brown, u.roedig}@lancaster.ac.uk

Abstract. Industrial process automation and control is a promising application domain of wireless sensor and actuator networks (WSAN). This type of application requires reliable and timely data delivery which is inherently difficult to achieve in wireless communication. In this paper, we present a MAC protocol for the aforementioned application domain. The protocol excels in simplicity compared with WirelessHART standard as no central network management is needed.

1 Motivation and Background

In industrial process automation and control applications, control loops (for example, proportional–integral–derivative (PID) controllers) are mapped onto a WSAN. For these control loops to function, measurement data need to be delivered reliably and within a given time bound D_S from sensors to a sink. Thereafter, a command needs to be transferred reliably and within a given time bound D_A from the sink to an actuator.

As WSNs are considered to be relatively unreliable, they so far have seen little use in such applications. However, given the potentially large cost savings wireless deployments in industrial settings can have over their wired counterparts, the feasibility of wireless process automation and control has began to see recent investigations. Most notably, the HART communication foundation and the EU funded project GINSENG [1] are both currently working towards supporting process automation and control applications. The established HART protocol used for cabled industrial process automation and control was recently extended to support wireless devices (WirelessHART). GINSENG uses the MAC protocol described within this paper. WirelessHART uses at its core a MAC protocol called TSMP [2]. TSMP is a protocol combining TDMA and Frequency Division Multiple Access (FDMA). A central entity called Network Manager is used to assign collision free transmission slots and to select redundant routing paths through a mesh network to ensure reliable transmission. GINSENG aims to avoid the complexity and increased communication overheads that are introduced by this Network Manager. Before deployment a generic topology and its associated TDMA schedule are computed; during deployment nodes select their position in this topology, whose position indicates which TDMA slots they should become active within.

2 Protocol Description

The protocol operation is divided into dimensioning and deployment phases. The purpose of the dimensioning phase is to determine a generic topology and associated

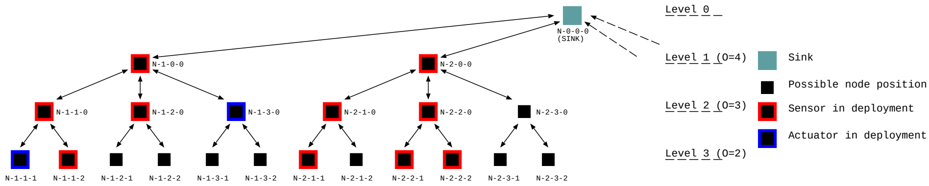

Fig. 1. Generic topology and example deployment of $N_A = 2$ and $N_S = 10$.

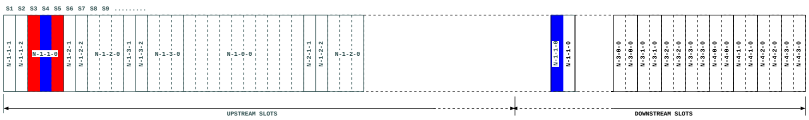

Fig. 2. Slot allocation in generic topology and slot usage of N-1-1-0 in the example deployment.

TDMA schedule for all nodes such that the required delay bounds D_S and D_A can be met. During the deployment phase a node can assume a position in the generic topology for which it has the pre-computed TDMA schedule.

2.1 *Dimensioning Phase*

For dimensioning a generic tree topology rooted at the sink of whose size can support the required delay bounds for the application is assumed. The tree is described by the two parameters: maximum hop distance H and fan-out degrees $O_h\,(0 < h \leq H)$ at each tree level h. Each node is identified by an address of the format N-{Level1ID}-{Level2ID}-{Level3ID} signifying the position of the node in the tree. An example topology is shown in Fig. 1 with $H = 3$ and $O_1 = 4, O_2 = 3, O_3 = 2$. The topology can accommodate a maximum number of N^{max} nodes and in the example above $N^{max} = 41$. However, in the actual deployment a number of nodes $N \leq N^{max}$ may be used. The network consists of N_S sensing nodes and N_A actuator nodes. We assume that no TDMA slot can be shared by two nodes as the transmissions of one node may interfere with transmissions of other nodes. A number of slots S^{up} is required to accommodate traffic flowing from nodes to the sink; a number of slots S^{down} is required to accommodate traffic flowing from the sink to actuators. We provision the system such that within one epoch all nodes can send one data packet to the sink. Moreover, within this epoch the sink can send one broadcast data packet to all nodes (for configuration) and one command data packet to each actuator in the network. The required TDMA schedule for the example topology is shown in Fig. 2. It has to be noted that non-leaf nodes require transmission slots to support the transmissions for all descendants/child nodes in additional to their own transmission.

2.2 *Deployment Phase*

When deployed a node first listens continuously for any transmissions from already deployed nodes. These packets are used to obtain time synchronization and to learn of free position(s) in the generic topology the node could assume. Transmission timing is strictly managed, transmissions occur at T microseconds into the TDMA slot. Receiving nodes can calculate the start time of the current slot $slot_t$ by subtracting T and the transmission time TX from the reception time R giving $slot_t = R - TX - T$. Nodes only perform time synchronization and correct their clock when receiving packets from nodes that are closer to the root of the tree. Each slot is $10ms$ in length which includes time for message processing and the transmission of a maximum sized data packet and acknowledgment. Nodes are not limited to assuming a single network position, a node might decide to assume several positions in the virtual tree at once to improve data transport reliability. Assuming multiple positions allows nodes to use multiple routes for reliability or to send more traffic than provisioned. This method of operation requires that the general topology is dimensioned for more nodes than will be physically present at deployment time. Depending on the position(s) a node assumes, it has to be active in different slots of the precomputed TDMA schedule. Fig. 2 shows which slots are activated by node n-1-1-0 if the generic topology is occupied by nodes in the deployment as shown in Fig. 1.

3 Evaluation

The MAC protocol was implemented on TinyOS 2.0.2 for the TelosB platform, which incorporates a CC2420 radio transceiver. As an initial evaluation, we deployed a network of 15 nodes structured as a binary tree with $H = 3$. Each node periodically generated packets to the sink, and the end-to-end delays were recorded and compared with the calculated delay bounds. The results indicated that packets were delivered within these delay bounds. Hence, the MAC protocol could provide the necessary time guarantees.

4 Conclusion

The initial evaluation validated that data were transferred within the delay bounds. Our proposed protocol is thus a simple solution to support WSAN. Additional evaluation is being carried out in a lab environment, and these results will be further verified in a real-world industrial setting of an oil refinery.

Acknowledgements

This work has been partially supported by the European Commission under the contract FP7-ICT-224282 (GINSENG).

References

1. C. Sreenan, J. S. Silva, L. Wolf, R. Eiras, T. Voigt, U. Roedig, V. Vassiliou, and G. Hackenbroich. Performance control in wireless sensor networks: the ginseng project. Communications Magazine, 47(8), August 2009.
2. K.S.J. Pister, and L. Doherty. TSMP: time synchronized mesh protocol. Proceedings of the IASTED International Symposium on Parallel and Distributed Computing and Systems, 2008.