A Simple Application Protocol for Embedded Controllers Over WPAN Networks

Rémy Feltrin, Stéphane Mocanu

A simple application protocol for embedded controllers over WPAN networks ⋆

Rémy Feltrin * **Stéphane Mocanu** **

* *GIPSA-lab, 961 rue de la Houille Blanche, BP 46 F - 38402 GRENOBLE, Remi.Feltrin@gipsa-lab.grenoble-inp.fr*
** *Stephane.Mocanu@gipsa-lab.grenoble-inp.fr*

Abstract: As a consequence of the new requirements for low energy consumption in buildings, intelligent embedded control systems will be widely deployed for domestic and office aerologic applications. Our work is concerned with the real-time tuning and application protocol synthesis for a wireless controller PAN (Personal Area Network) based on the IEEE 802.15.4 MAC (Medium Access Control) layer.

Keywords: Application protocol, 802.15.4 networks, networked control

1. INTRODUCTION

The new generation of embedded controllers for domestic and office applications are developed in a close relationship with the advance of the sensor wireless networks. For example, in the so called "intelligent building applications", which focus mainly on low energy consumption, controllers are required to be compliant with low rate, low consumption communication networks. One of such networks, which is of increasing interest for control developers, is the IEEE 802.15.4/Zigbee network (or, shortly, Zigbee). We focus our work on the use of Zigbee networks in control applications.

A long experience in fielbuses (mainly in wired real-time communication networks) shows that there are two main complementary approaches for network performance in control applications :

- Intrinsic real-time performance of the communication (including latency, end-to-end delay and predictability or determinism). This is achieved at the MAC layer level and involves real-time scheduling (time-triggered or absolute message priority)
- Readiness for control application deployment. Traditionally real-time communication networks provides an efficient data exchange support but application deployment (like SCADA interfaces or sensor/actuator integrations) are usually a difficult to develop due to a lack of application layer primitives for control (this is usually called an "application profile"). The test of time applied to some 50 fieldbuses used in the last 30 years shows that the second main reason for the abandon of a fieldbus is the lack of application profiles, i.e. a poor application layer.

Originally, Zigbee networks are not intended for control applications. The MAC layer (IEEE 802.15.4) is not designed especially for real-time performance and the numerous application profiles do not include control applications. However, the very low energy consumption of Zigbee networks together with the availability of cheap embedded communicators made them of great interest for home and office small control applications like light and temperature control.

Some, comparatively, important work was devoted in the last years for computing the real time performance of Zigbee networks (see Shin et al. (2007); Park et al. (2005) or Mišic et al. (2005) for example) but there are no attempts, to our knowledge, to specify the control application profiles.

In our work we focus on the co-design of the real-time tuning of the MAC parameters and the specification of the application layer. This co-design is necessary in order to respect the embedded application constraints : bidirectional communication of reduced amounts of data and program memory, fast application layer services.

The new protocol stack is implemented on embedded microcontroller devices and used for a centralized control of air conditioning application.

In the next section we present the sample control application. Although some application particularities are taken into account at application level we try to provide a more general control application profile not necessary limited to the actual application.

In section 3 we briefly present the main features of the Zigbee network. Section 4 is concerned with network parameter choice in order to achieve the best real-time performance. In section 5 we present our application layer together with a communication recovery protocol. The paper is concluded with some practical remarks and future work considerations.

2. THE SAMPLE CONTROL APPLICATION

Under floor air distribution (UFAD) solutions tend to be the modern alternative of the traditional ceiling-based ventilation (Center for the Built Environment (CBE) (2002)). An UFAD indoor climate regulation process is set with the injection of a fresh airflow from the floor and an exhaust

⋆ Work supported by the Carnot LSI institute.

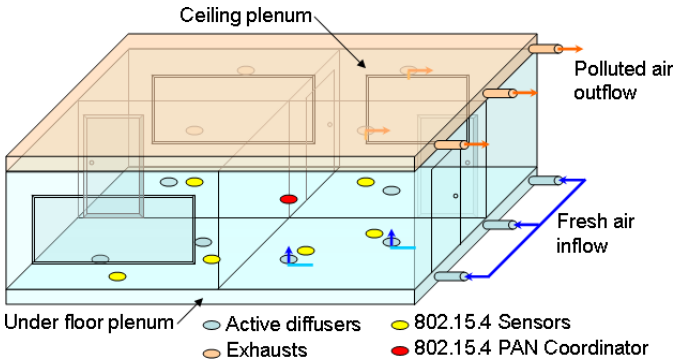

Fig. 1. UFAD ventilation

located at the ceiling level, as depicted in Fig. 1. Note that we consider the specific case where a common plenum is used at both the underfloor and ceiling levels. It has been established that well-designed UFAD systems can reduce life-cycle building costs, improve thermal comfort, ventilation efficiency and indoor air quality, conserve energy, and reduce floor-to-floor height. Feedback regulation is a key element for an optimized system operation and it can be achieved thanks to actuated diffusers and distributed measurements provided by a wireless sensor network (WSN) deployed in the ventilated area.

As seen in Fig. 1 each slave board handles one sensor and one actuator (a temperature sensor, respectively a fan controller). The regulation is achived at the PAN-coordinator in a centralized controller architecture.

The full details on the UFAD modeling and controller synthesis are provided in Witrant et al. (2009)

3. NETWORK ARCHITECTURE

Our network setup is based on Silicon Laboratories development kits build around a precision mixed-signal 8051 MCU and a Chipcon CC2420 2.4 GHz 802.15.4 transceiver. Sensors are set in a star topology around a Personal Area Network (PAN) coordinator. Several works (Kumar et al. (2008); Salles and Krommenacker (2007)) had shown that the best real-time performance in 802.15.4 is achieved in a coordinated network using 2.4GHz frequency band and O-QPSK modulation which are both supported by Chipcon tranceiver.

3.1 IEEE 802.15.4 real-time support

In Salles and Krommenacker (2007) an exhaustive study of the real-time performance of IEEE 802.15.4 was presented. Our conclusions are based mostly on this study and partly on the performance analysis presented in Zheng and Lee (2006). Two medium access methods are available in coordinated 802.15.4 low-rate wireless personal area networks (LR-WPAN) : beacon-enable mode and non-beacon-enable mode. Non-beacon-enabled mode uses only Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) medium access method quite similar to 802.11 WI-FI networks, which is not deterministic, and therefore not suitable for real-time applications. In beacon-enabled mode a Contention Free Period (CFP) is defined which implements a fairly good deterministic support for real-time applications.

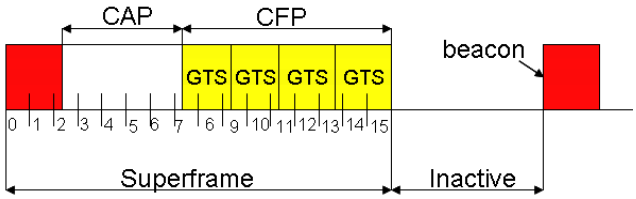

Fig. 2. Superframe structure in beacon-enabled mode

3.2 Beacon-enabled mode

While in beacon-enable mode the WPAN behavior is driven by synchronisation signals (beacons) sent by the PAN coordinator. The beacon interval defines the transmission period of the network. According to 802.15.4 standard the beacon interval is divided between an active period (called superframe) and a power saving period (inactive). The superframe is divided in 16 slots which are allocated to the beacon frame, the Contention Access Period (CAP) and the Contention Free Period. The CAP period is a CSMA/CA transmission and has no real-time interest. The CFP period is divided in reserved slots (called Guaranteed Time Slots - GTS) which are allocated to devices by the PAN coordinator. The overall structure of the superframe and the beacon interval is presented in Figure 2. The existence of GTS provides the real-time support of 802.15.4 networks.

Unfortunately there are three limitations in the 802.15.4 superframe which decreases the real-time performance. Firstly a minimal length of the CAP period has to be preserved. Secondly, the maximal number of allocatable GTS is limited to 7. Thirdly, the GTS slots are unidirectional and they expire after four inactive slots.

3.3 Network and Application Layer

In a IEEE 802.15.4/Zigbee network the Zigbee part actually covers the Network and Application Layer. Unfortunately, Zigbee application layer is not designed for control application. On another hand for a small PAN with a single coordinator the network layer is useless. In order to produce a lighter and faster code we avoid the use of the Zigbee layers. As in traditional real-time (wired) networks our networks will consist on an application layer over the 802.15.4 MAC layer.

4. FINE TUNING OF REAL-TIME COMMUNICATION

A first fact that has to be taken into account is that, given the unidirectionality of the GTS slots one cannot use GTS transmission for data acquiring and actuator commands.

The real-time performance of the beacon-enabled LR-WPAN networks depends on several parameters which controls the superframe structure and length. We examine the choice of the various parameters from the point of view of the control system.

The main performance measures of interest are : the end-to-end propagation delay, the bit error rate and the maximal polling rate. While we assume only CFP transmission there are no back off frames. It follows that the only source of delay is the pure radio waves propagation delay which, given the low distances (around 15m) and the high radio waves propagation speed, it is of order of nanoseconds and can be neglected.

The bit error rate parameter can be taken in account as a variable delay in the information transmission. For the moment we ignore this parameter. This is not a very strong assumption while some works Shin et al. (2007) show that the main source of packet errors is the CSMA/CA transmission.

We turn out to the study of the maximal polling rate (equivalently, the minimal sampling period). While a communication cycle is defined by the beacons, a high polling rate is achieved by short beacon intervals. However, there is a trade-off between the superframe duration and the number of polled sensors in a superframe. The optimization of the maximal polling rate is dependent on the number of sensors managed by the PAS coordinator.

The following system and application accessible variable are controlling the superframe (we consider unacknowledged transmission only) :

System variables (cannot be modified) :

- Minimal CAP Length : 440 bytes
- Base Slot Duration : 60 bytes
- Beacon maximal size : 127 bytes
- interframe spaces : 40 bytes for long frames, 12 for short frames (¡ 18 symbols)
- MAC frame overhead : 5 to 25 bytes depending on addressing mode. Most common 13 for 2 bytes MAC adresses.
- Physical frame overhead : 6 bytes

Application variables :

- Superframe duration (controlled by the Superframe Order - SO)

  $$SD = \#slots \times BaseSlotDuration \times 2^{SO} = 960 \times 2^{SO}$$

  The legal values of SO can range between 0 and 14.

One can easily see from the above formula for the superframe duration that a 802.15.4 frame is between 15.36 ms (for SO=0 at 256 kbps transmission speed) and 251.6 s (for SO=14 also at 256 kbps).

We made our dimensioning in order to compute the minimal superframe order which allows the data transfer from our sensors. Once the minimal order of the superframe is determined, the choice of the actual order is a compromise between sampling rate and energy consumption.

It easily follows that the minimal sampling period that can be used is 15.36 ms. However, achieving this small sampling period is not obvious because SO=0 frames assume an important protocol overhead (more than 50% of the frame is used by the beacon and CAP). Then a very important step in communication tuning is to verify that the available capacity of the superframe can acquire the necessary communication data.

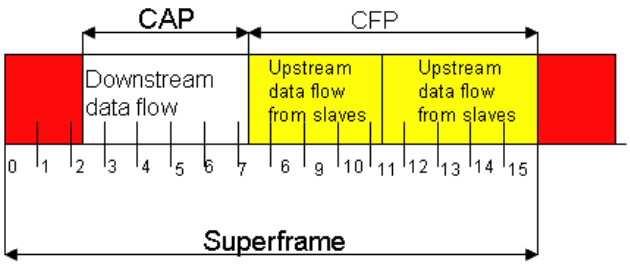

Fig. 3. Data exchange embeding in Superframe

In our setup the sensor use accurate 24-bits A/DC. Then, each GTS needs between 25 and 72 bytes (2 payload, 11 to 31 overhead, 12 or 40 interframe). As a Base Slot is 60 bytes long, a GTS longer than 60 bytes will occupy two entire Base Slots. Or in a superframe of order 0, only 5 or 6 base slots are available (8 are used by CAP, and 2 at worse 3 are used by the beacon). In order to insure that our sensors can pe polled in a single order zero superframe we have simply to chose short (2 bytes) MAC addressing which induces 31 bytes overhead, and requires then a single base slot GTS.

4.1 Full data transfer application protocol

The previous simple calculations insure that for two bytes data acquisition applications a order zero superframe can be used for up to 6 slave boards. Actually, the maximal payload of a GTS is of 35 bytes. This allow us to define several APDU (application protocol data units) for data transfer and network management.

We firstly achieve the definition of bidirectional data exchange protocol and then, in the next section we define the application management APDUs.

As stated previously, GTS transfer allows only for unidirection date acquisition. In theory the GTS direction can be reversed but this will require three more superframes (reverse request, confirmation and data transfer), then this will multiply the network period by 6 (three superframe for data acquisition and three other for controls transmission). Fortunately there is an alternative in the case of centralized control: while there are as many sources of trafic as slave boards for the upstream data flow (data acquisition), there is only one trafic source for the downstream data flow (the controller). Then one can safely use the CAP period for the downstream data flow. It follows that we can use the following data exchange protocol :

- Upstream : each slave transmits data to coordinator during GTS (up to 35 bytes).
- Downstream : controller broadcasts commands during CAP (up to 440 bytes).

The data embedding in the supertrame is represented in Fig. 3

5. COMPLETE APPLICATION PROFILE

The complete application layer is defined in order to handle the following issues :

- Bidirectional data exchange.
- GTS expiration handling.

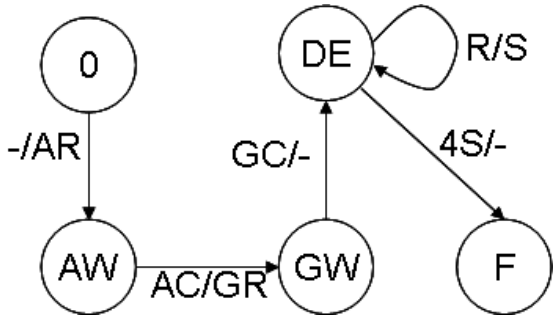

Fig. 4. Mealy Automata modeling the slave behavior

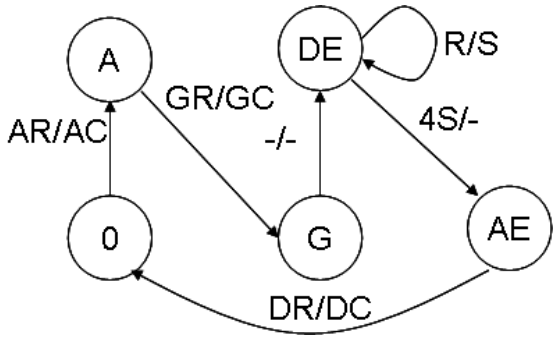

Fig. 5. Mealy Automata modeling the coordinator behavior

Bidirectional data exchange is handled at application level as described in the previous section.

The GTS expiration problem is a bit more complex. As specified by the 802.15.4 standard a GTS will expire if it is not used during four superframes. The slave is not informed of the GTS expiration but it can made a new GTS request if needed. At high sampling rates a slave in power slave mode can easily miss GTS deadlines and then go to a blocked state (an expired GTS is not deleted from the GTS list while the slave is still associated so the number of new GTS requests is very limited). Our protocol application implements a recovery procedure in case of GTS expiration by a globad network refresh and association reset of expired slaves.

First lets investigate the behavior of a slave in absence of recovery mechanism. the normal behavior of a slave using GTS is represented by the Mealy automata in Fig 4. Starting from the inactive state 0 the slave made an Association Request (AR) and enters the Wait for Association (AW) state. Once the Association Confirm (AC) is received the slave is issuing the GTS Request signal and waits for confirmation (GW). Once the GTS allocation confirm (GC) is received the normal Data Exchange (DE) state is entered and repeated Send/Receive data signals are issued. Again, the 802.15.4 standard was not designed for real-time applications. Normally the use of GTS is very limited and not intended for periodic data exchage. For this reason is a non-disconnecting slave misses fours consecutive superframes (4S event) the GTS expires and from the point of view of the control applications it enters a faulty state (F). Even if, theoretically, the slave can renew the GTS request this is not a solution in the case of periodic data exchange. Moreover, the 4S event is not observable by the slave and the GTS expiration can be detected only by a transmission error.

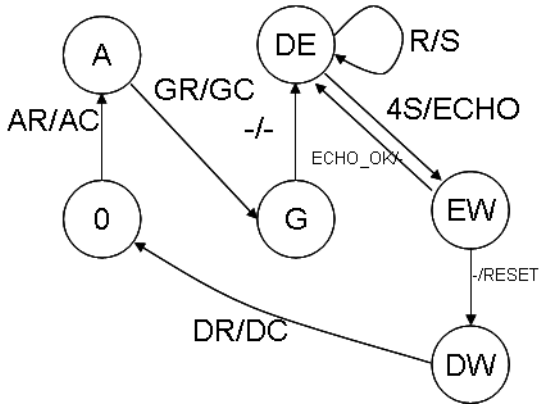

Fig. 6. Coordinator controlled behavior

The corresponding Mealy machine corresponding to coordinator behavior is presented in Fig 5. Once the Association and GTS allocation requests are satisfied (AR/AC and GR/GC requests) during the Data Exchange period if the inactive GTS superframes counter expires (4S) the coordinator will mark the slave as GTS expired and associated waiting for the disassociation requests to issues the disassociation confirm (DR/DC) in order to reset the connection. Again, in a periodic GTS exchange data, according to the 802.15.4 standard this will be very costly as the slaves are not supposed to disconnect and has to wait for a transmission error before deciding that the GTS is expired.

5.1 GTS recovery

We wish to define an application level procedure to recover the GTS expiration during a periodic data exchange. Obviously the problem can be solved directly at MAC level but we wish to keep the compatibility with the standard 802.15.4 network. then we'll define a procedure at application level for GTS recovery.

In terms of supervisory control, the required procedure is a discrete event controller whose objective is to avoid the fault state. The control is defined at the coordinator level and it consist in early detection of the GTS expiration and slave state reset. Such that when the coordinator detects the expiration of a GTS, instead of simply deleting the GTS, it will also broadcast an ECHO application request to the slaves. Active slaves will answer by an ECHO_OK data unit, while unsynchronized slaves (expired GTS) will not answer. The coordinator will refresh the slaves list then will send the RESET application request to the lost slaves. On RESET message the slaves will send a Disassociation Request to the coordinator followed by an Association Request, GTS Request sequence (i.e. the normal restarting sequence). The final (controlled behavior) is described in the charts Fig 6 (coordinator side) and Fig 7 (slave side).

It is easy to see that the recovery procedure is not blocking. Once the ECHO request is issued by the coordinator if the ECHO_OK is received both slave and coordinator goes back to the Data Exchange status. Otherwise, on RESET the slave and the coordinator will synchronize on Disassociation Request/ Confirm (DR and DC signals)

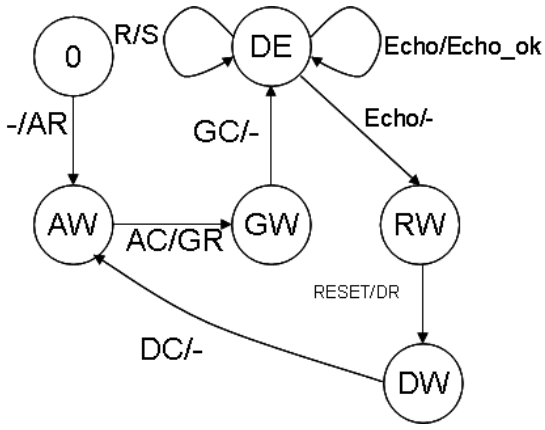

Fig. 7. Fast GTS recovery on slave side

which allows a fast restart of the slave (no timeout needed) and let the coordinator to keep a proper list of associated slaves.

6. CONCLUSIONS AND FURTHER WORK

The main contribution of this work is the proposal of an application layer for control applications over IEEE 802.15.4 networks. The main features of the application layer are an optimized data exchange protocol using GTS slots for upstream data and CAP periods for downstream transmission and an included fast recovery procedure when GTS synchronisation is lost. The application layer is operational and was implemented on a demonstrator. Empirical measures in our experimental networks shows that on a five boards networks the rate of lost packets is around 10/day at a 300ms sampling rate while the original network (without fast recovery) expected blocking.

The future work consist in extending the application profile by adding remote configuration requests and remote sensor calibration. In a last step we'll attempt to submit an draft proposal for control profile to the Zigbee Alliance.

ACKNOWLEDGEMENTS

This work is supported by a Carnot LSI grant on Intelligent Buildings.

REFERENCES

Center for the Built Environment (CBE) (2002). Design guide on underfloor air distribution systems. Technical report, ASHRAE.

Kumar, P., Güneş, M., Al Mamou, A., and Schiller, J. (2008). Real-time, bandwidth, and energy efficient IEEE 802.15.4 for medical applications. In *7. GI/ITG KuVS Fachgesprch "Drahtlose Sensornetze"*. FU Berlin, Germany.

Mišic, J., Shafi, S., and Mišic, V.B. (2005). The impact of mac parameters on the performance of 802.15.4 pan. *Elsevier Ad hoc Networks Journal*, 3, 509528.

Park, T., Kim, T., Choi, J., Choi, S., and Kwon, W. (2005). Throughput and energy consumption. analysis of IEEE 802.15.4 slotted CSMA/CA. *IEEE Electronics Letters*, 41(8), 1017–1019.

Salles, N. and Krommenacker, N. (2007). Performance Analysis of IEEE 802.15.4 Contention Free Period through Real-Time Industrial Maintenance Applications. In *22nd IAR annual meeting*. Grenoble France.

Shin, S.Y., Park, H.S., and Kwon, W.H. (2007). Throughput and delay analysis of unslotted IEEE 802.15.4. *IEICE Transactions on Communications*, E90(10), 2961–2963.

Witrant, E., Mocanu, S., and O., S. (2009). A hybrid model and MIMO control for intelligent buildings temperature regulation over WSN. In *8th IFAC Workshop on Time Delay Systems*. Sinaia, Romania.

Zheng, J. and Lee, M.J. (2006). *Sensor Network Operations*, chapter 4.3 Comprehensive Performance Study of IEEE 802.15.4., 218–237. Wiley – IEEE Press.