Approaches to Support Real-Time Traffic over Bluetooth Networks

Lucia Lo Bello, Mario Collotta, Orazio Mirabella
RETISNET Lab, Dipartimento di Ingegneria Informatica
e delle Telecomunicazioni, University of Catania, ITALY
lucia.lobello@diit.unict.it, omirabel@diit.unict.it

Thomas Nolte
MRTC, Dept. of Computer Science and Electronics
Mälardalen University, Västerås, SWEDEN
email: thomas.nolte@mdh.se

Abstract

The paper addresses the possibility of real-time data communications on the factory floor over wireless networks based on the Bluetooth (BT) standard. Originally designed as a cable replacement technology for low-cost, effortless connection of electronic devices, BT does not provide real-time support for data packets. However, employing BT on the factory floor is quite an attractive option, because low-power, cheap and easy-to-build solutions can be obtained using the available BT modules and application profiles. The great interest recently shown in using BT for supporting factory communication is therefore a good reason for investigating deadline-aware scheduling mechanisms that will allow Bluetooth networks to meet real-time constraints. The paper presents and discusses two approaches to schedule real-time traffic on Bluetooth networks used in Distributed Process Control Systems (DPCSs).

1 Introduction

The Bluetooth (BT) standard [1], originally developed for short-range ad-hoc wireless interconnection, is becoming quite appealing in the industrial environment, where a number of applications, such as remote control, diagnostics, process supervision, etc. can benefit from replacing traditional wired connections with wireless ones [10, 11, 5]. BT makes it extremely easy to configure wireless systems and stations, allowing the fast creation of communication infrastructures in industrial plants without the need for expensive cabling.

BT is also gaining ground among wireless systems in the context of sensor networks because of its resilience to interferences (notably in the 2.4 GHz band), thanks to the Frequency Hopping scheme it implements. Other benefits which BT offers for sensor networks are software support (network self-assembly, multihop routing, in-network processing), and fairly low energy consumption. Moreover, the mass production of BT modules guarantees robustness and decreasing costs.

The BT communication topology is point-to-multipoint, where a Master node communicates with up to seven Slaves forming what is called a piconet. To resolve contention over the wireless links, the BT channel is organized according to a Time Division Multiple Access/Time Division Duplex (TDMA/TDD) scheme based on 625-μs slots. Scheduling is handled by the Master, alternating a Master transmission with one by a designated Slave. This implies that the communication occurs in pairs of slots (i.e., the Master-Slave pair). The approach does not allow for direct Slave/Slave communications. If a Slave wishes to communicate with another Slave it can only do so through the Master. The Master also establishes the sequence of transmission/reception frequencies, which will be known to all the piconet Slaves. They will calculate the hopping sequence autonomously, on the basis of the Master address and its clock. As all the nodes in the piconet agree on the next transmission/reception frequency, they can all take part in communications by synchronizing on the appropriate frequency.

Bluetooth supports both voice and data traffic [1] through two types of links, i.e., Synchronous Connection-Oriented (SCO) links for voice and Asynchronous ConnectionLess (ACL) links for data. Voice traffic occupies fixed slots assigned a priori by the Master. In Distributed Process Control Systems (DPCSs), both periodic and aperiodic data traffic is present, and the communication protocol has to be flexible and reliable. For this reason, we will focus here on ACL links.As said before, the piconet Master polls each Slave thus enabling it to transmit. This straightforward approach has, however, some shortcomings for the kind of traffic typically found in DPCSs. First, if only one of the Master or the Slave has data to send, this approach introduces a non-negligible overhead, as a slot gets wasted. The protocol efficiency, and consequently the actual throughput, is therefore limited, especially when short messages (i.e., one slot long) are exchanged. This is often the case with DPCSs, where exchanged messages, especially those featuring timing constraints (i.e., periodic variables), are small in size. Moreover, the transmission overhead also affects the message delay experienced. Second, the polling schemes implemented by the Master to address the Slaves, such as the One Round Robin (1-RR), are not deadline-aware, so they are unable to deal effectively with time-constrained data traffic.

To overcome these limitations, this paper will address innovative scheduling algorithms to enable BT networks to support real-time traffic in DPCSs scenarios. Two approaches are presented. The first - EDF/TBS - combines Earliest Deadline First (EDF) with a Total Bandwidth Server (TBS) [8, 9]. The second - EDF/AS - is based on EDF and the insertion of an Additional Slot (AS) for aperiodic traffic at the end of each periodic traffic polling cycle.

The proposed approaches are motivated and described, and their possible implementations discussed.

2 System model

The first scheduling level is Local Scheduling (LS), performed inside the local queues of each BT device. We assume that, to support real-time traffic, LS is handled in each device in an EDF way. The second level, called Intra-Piconet Scheduling (IPS), refers to the polling scheme used within the piconet. As in [2], here an IPS is defined as the set of rules which determines when the piconet Master switches from one Slave to another. In order to support real-time communication, deadline-aware policies should also be implemented at the IPS level.

As in [7], here we use the notion of *knowledge* to schedule messages, where knowledge means the information used by the Master to build the IPS scheme. As the context dealt with here is typical of DPCSs, we assume that traffic exchanges are known a priori, at least for periodic variables. The Master could be configured by an operator or acquire a configuration file from a database. As far as aperiodic traffic is concerned, we assume a *signalling scheme*, where Slaves can communicate their queue status while transmitting packets to the piconet Master. This information is entered into a specific packet field of the slot sent to the Master, e.g., the flow bit in the header, as in [3]. For the sake of simplicity, we assume that each Slave produces one periodic variable (with a fixed period and deadline equal to the period) and one aperiodic variable, both of known size. Each Slave therefore manages two distinct queues (buffers) for periodic and aperiodic variables. However, this simplifying assumption is not mandatory: if several different periodic and aperiodic variables are produced by each Slave, several distinct traffic classes, together with the relevant message queues, will be implemented. In this case, QoS support (i.e., differentiated services for different traffic classes) can be provided by assuming a priority-based mechanism to poll the different queues, where the priority of each queue reflects the importance of messages from the application point of view.

3 The EDF/TBS approach

In the EDF/TBS approach, transmissions of periodic variables by each Slave are handled by the Master according to the EDF algorithm, while aperiodic requests are managed through a Total Bandwidth Server (TBS) [8, 9], a simple to implement as well as efficient aperiodic service mechanism. Originally proposed for processor scheduling of aperiodic tasks, the idea behind the exploitation of TBS here is that whenever an aperiodic request arrives (through the signalling scheme discussed before), the piconet Master assigns to it the requested bandwidth in such a way that the overall utilization by aperiodic traffic never exceeds a given maximum value U_s, determined so as to make the following inequality hold:

$$U_p + U_s \leq 1 \tag{1}$$

where U_p and U_s are the utilization factors for periodic and aperiodic traffic, respectively. U_p is determined taking into account that each periodic transmission from a Slave entails an extra slot, i.e., the slot sent by the Master to address the Slave. We therefore have to refer to the bandwidth utilization of a periodic *transaction*, where not only the transmission time of the periodic variable, but also the slot transmitted by the Master is accounted for. As a result, in a piconet with seven Slaves, U_p is given by:

$$U_p = \sum_{i=1}^{7} U_i = \sum_{i=1}^{7} \frac{C_{iP} + C_M}{T_i} \tag{2}$$

where C_{iP} is the transmission time (in slots) of the periodic message (period = T_i) containing the periodic variable generated by Slave i and C_M is the transmission time of the polling message sent by the Master (i.e., 1 slot).

According to the TBS algorithm, when an aperiodic request arrives (i.e., it is signalled to the Master) at time $t = r_i$, it receives a deadline d_i which depends on the bandwidth allocated to previous aperiodic variable transmission requests, the transmission time of the request itself and the server bandwidth, according to the following rule:

$$d_i = \max\left(r_i, d_{i-1}\right) + \frac{C_{iAp} + C_M}{U_s} \tag{3}$$

where C_{iAp} is the transmission time (in slots) of the aperiodic message containing the aperiodic variable generated by Slave i, and C_M is the transmission time of the polling message (i.e., 1 slot) sent by the Master. As in the periodic case, we will therefore refer to an aperiodic transaction, which takes into account the extra slot sent by the Master. Although not crucial for the mechanism, for the sake of simplicity here we assume that $C_{iP} = C_{iAp} = 1$ slot. This assumption is realistic as data exchanged in DPCSs are usually small in size. Once the deadline has been assigned to the aperiodic request, the Master inserts the Slave generating the request into the EDF-based IPS scheme, and the Slave will be addressed as any other Slave generating periodic traffic. When receiving the slot sent by the Master, the Slave will be allowed to transmit its aperiodic variable in the next slot.

4 The EDF/AS approach

The EDF/AS approach handles the transmission of periodic variables in an EDF way, while aperiodic traffic is transmitted in Additional Slots (AS), added at the end of each polling cycle and statically reserved to one of the Slaves. The EDF/AS approach is based on an innovative transmission mode for the BT standard - the Slave/Slave transmission proposed in [6]. Sect. 4.1 illustrates how this communication is achieved.

4.1 Slave/Slave communication in BT

If the traffic is periodic and the Slaves transmission requirements are known a priori, as is typical of DPCSs, the protocol overhead can be significantly reduced by allowing direct communications between Slaves without mediation by the Master, to the benefit of protocol efficiency

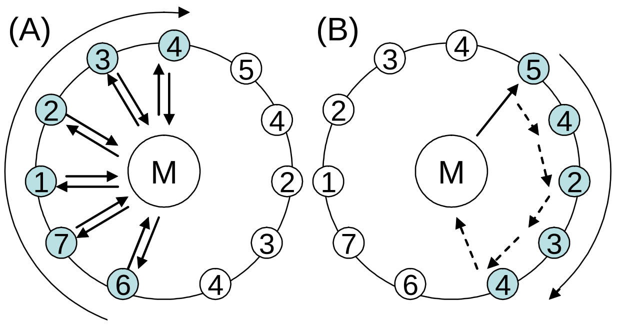

Figure 1. Master/Slave communication (A), and Slave/Slave Communications (B).

and system exploitation. The greater amount of bandwidth available can be used to accept new flows of periodic traffic and to transmit aperiodic traffic. However, the addition of the Slave/Slave operating mode should not preclude the presence of traditional devices, so careful design is needed. To ensure compatibility with devices operating in the traditional Master/Slave mode, no significant modifications to the BT standard should be required and changes should be confined to the software as much as possible. As explained in [6], Slave/Slave transmission can take place by means of scheduling handled by the Master, which has to previously configure a group of Slaves as belonging to a certain logical ring (specifying the order of the various Slaves in the Ring: 5, 4, 2, 3, 4 in the example shown in Fig. 1.) via a Broadcast message and then transfer to these stations the right to transmit in sequential order, specifying the starting slot and the number of times the sequence is to be repeated. As can be seen in Fig. 1A, the Master communicates with some nodes (6, 7, 1, 2, 3 and 4 in the example) in sequence (according to its scheduling) and then authorizes communication between nodes 5, 4, 2, 3 and 4, which have previously been configured as a logical ring (see Fig. 1B).

The approach is thus a token passing one, based on a virtual token, i.e., the token does not need to be passed physically because each station knows its position in the logical ring and therefore its turn to transmit. Slaves belonging to the logical ring transmit when it is their turn according to the schedule that the Master has broadcasted. A Slave recognizing itself as being the addressee of a message receives and copies it, but does not reply in the slot immediately following (as would happen in normal Master/Slave communications), unless this is contemplated in the scheduling of the virtual token ring configured by the Master. The use of a fixed-length slot foreseen in the BT standard maintains synchronization between the various nodes and prevents overlapping between two consecutive transmissions. Time lags between subsequent transmissions are not needed, because the time sequence of the slots is known to all the nodes.

Even in the logical ring, scheduling is always up to the Master, which has to acquire the knowledge needed for correct bandwidth allocation in the ring. This knowledge is simple to acquire in DPCSs, which typically feature well-defined, repetitive information exchanges (with mostly periodic producer/consumer traffic). The Master could therefore be configured during the startup phase either by an operator or reading a configuration file from a database. Once the Master has acquired the time requirements of all traffic, it can perform a scheduling analysis and compute a suitable scheduling table, containing the time slots assigned to each Slave/Slave transmission. The transmission sequence in the logical ring is repeated after a time, called a scan cycle, defined on the basis of the control dynamics. Due to the need to build a static schedule, the scheduling approach described is very suitable for fixed (or rarely mobile) systems in which the piconet nodes do not change and scheduling remains valid for long periods of time. However, the non-mobility constraint is not a real drawback in DPCSs, where devices are rarely moved once allocated.

4.2 Implementing the EDF/AS approach

In the EDF/AS approach, the Master forms a logical ring with n equal scan-cycles, each having one additional slot reserved for aperiodic traffic. Such a slot is statically assigned to one of the n Slaves in the logical ring in such a way that each of them in turn will have a slot to transmit aperiodic traffic. At the end of the first scan cycle, for example, the AS will be reserved for Slave 1, at the end of the second for Slave 2 and so on. The transmission of each aperiodic variable is thus encapsulated in the additional slot each Slave is cyclically assigned.

The Master will resume Master/Slave transmission with the Slaves involved in the logical ring at the end of the n scan-cycles forming the ring. It is even possible to have the Master operating in parallel with the logical ring, communicating in the Master/Slave mode with piconet nodes not involved in the logical ring. In this case, to avoid collisions, transmissions on the logical ring have to take place at frequencies other than those used for Master/Slave transmissions. Under the hypothesis assumed here that messages exchanged in the logical ring are of a length equal to one slot, a one-hop shift in the hopping sequence is sufficient to avoid collisions. Such a shift only requires a modification to the Clock OFFSET value which is used to maintain synchronization with the Master. In general cases, the number of hops in the frequency hopping sequence needed to avoid collisions with nodes operating in the Master/Slave mode will be specified in the broadcast message used to configure the ring. Thanks to the robust adjacent channel filtering implemented in Bluetooth, an inner parallelism inside a piconet using shifted frequencies on the same frequency hopping sequence could be more beneficial than having multiple unsynchronized collocated piconets close from one another. The impact of co-channel interference from other BT piconets on the packet error probability was addressed in [4]. This aspect is particularly critical in environments such as the factory floor, where multiple wireless networks operate in a common air space. In any case, being the Master either active in parallel with the logical ring or idle for the duration of the logical ring, the Slave/Slave transmis-

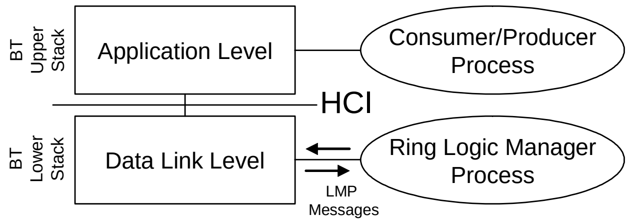

Figure 2. Functional architecture

sion sequence will be repeated a certain number of times established by the Master. This number must be less than the maximum duration of a Slave/Slave sequence before a loss of synchronization with the Master occurs. Such a limit (i.e., 800 slots) was derived in [6], and shows that it is possible to generate long Slave/Slave transmission sequences without loss of Master synchronization. When the time allocated to the logical ring expires, the last Slave realigns with the Master hopping frequency. From the next slot onwards, the original piconet, comprising all the Slaves, is restored and the Master/Slave transmissions are started again.

As compared with the EDF/TBS approach, in the EDF/AS approach the duration of any transaction in the logical ring is reduced to transmission by a Slave and thus to useful traffic, to the advantage of schedulability, which also means that periodic traffic with more urgent deadlines can be supported.

5 Implementation issues of the Slave/Slave communication

Management of the logical ring to implement S/S communication does not require any modification to the current BT specifications. However, some additional functionalities are needed, which require integration with processes performing link management in the baseband layer. As the communication functionalities available between the lower parts of the BT stack (UART, USB, PCMI) are not fast enough to allow synchronization between scheduling processes located in different parts of the stack, some functionalities have to be moved, possibly down to the baseband layer. Fig. 2 shows a possible functional system organization.

On the basis of the temporal constraints of application processes, the Master node schedules the Slave/Slave communications to be activated. The scheduling decisions are made at the Application level, on top of the HCI, where processing resources are available for the computation of scheduling sequences. Then the schedules are transferred to the Data Link level (baseband) where an ad-hoc process will manage the logical ring (or rings). The hardware resources required for logical ring management are only a memory register storing the current Slave/Slave scheduling and one pointer to shift the Master/Slave frequency hopping sequence. Some memory space is also required to store the logical ring management process. The new functionalities required for Slave/Slave communication entail a few changes at the baseband level, so they require some modifications in the firmware which have to be made by firmware developers. However, such changes do not entail major modifications in the BT chip architecture. One needed change is the access mode to the register containing the Master/Slave offset. The current Read_only permission should become a Read/Write one. Another change is enabling a Slave to transmit even without the explicit polling slot from the Master. This can be achieved by adding firmware modules implementing such a feature at the baseband level. The same holds for implementing the scheduling approaches proposed in the paper.

6 Conclusions and further work

Preliminary results from utilization-based schedulability tests (which are not presented here for reasons of space) have shown improvements in terms of enhanced schedulability for periodic traffic, obtained using both the proposed approaches. In particular, the EDF/TBS mechanism proved to be highly suitable for scenarios featuring both aperiodic and periodic exchanges, with less imminent periodic deadlines, while the EDF/AS mechanism is more suitable for scenarios where the exchanges are mainly periodic with stringent deadlines. Further work will deal with improvements and extensive simulations of the proposed approaches.

References

[1] Bluetooth SIG. Specification of the Bluetooth System - Version 1.1B, Specification Vol. 1 & 2, February 2001.

[2] A. Capone, M. Gerla, and R. Kapoor. Efficient polling schemes for Bluetooth picocells. In *Proceedings of IEEE ICC'01*, Helsinki, Finland, 2001.

[3] A. Das, A. Ghose, A. Razdan, H. Saran, and R. Shorey. Enhancing performance of asynchronous data traffic over the Bluetooth wireless ad-hoc network. In *Proceedings of IEEE INFOCOM'01*, Alaska, USA, 2001.

[4] A. El-Hoiydi and J. Decotignie. Soft deadline bounds for two-way transactions in Bluetooth piconets under co-channel interference. In *Proceedings of IEEE ETFA'01*, France, October 2001.

[5] Leine & Linde. Wireless health monitoring systems for encoders, July 2001. http://www.leinelinde.se.

[6] L. Lo Bello, N. Torrisi, and O. Mirabella. New operating Modes for Bletooth Networks in Distributed Process Control System. In *Proceedings of IEEE LCN'04*, Tampa, Florida, USA, November 2004.

[7] A. Mercier, P. Minet, and L. Gorge. Introducing QoS support in Bluetooth Piconet with a Class-based EDF scheduling. Technical report, Rapport de Recherche RR-5054, INRIA, Le Chesnay Cedex, France, December 2003.

[8] M. Spuri and G. C. Buttazzo. Efficient Aperiodic Service under Earliest Deadline Scheduling. In *Proceedings of IEEE RTSS'94*, San Juan, Puerto Rico, December 1994.

[9] M. Spuri and G. C. Buttazzo. Scheduling Aperiodic Tasks in Dynamic Priority Systems. *Real-Time Systems*, 10(2):179–210, March 1996.

[10] P.-A. Wiberg and U. Bilstrup. Bluetooth in Industrial Environment. In *Proceedings of IEEE WFCS'00*, Porto, Portugal, September 2000.

[11] P.-A. Wiberg and U. Bilstrup. Wireless Technology in Industry - Applications and User Scenarios. In *Proceedings of IEEE ETFA'01*, pages 123–131, France, October 2001.