

# Communication Techniques and Architectures for Bluetooth Networks in Industrial Scenarios

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

## Abstract

*The Bluetooth (BT) standard, originally designed as a cable replacement technology for low-cost, effortless connection of electronic devices, is becoming quite an attractive option even in industrial communication, as it allows low-power, cheap and easy-to-build solutions to be obtained using the available BT modules and application profiles. The great recent interest in using Bluetooth in Distributed Process Control Systems (DPCSs) provides the motivation for this work, which has a twofold aim. Firstly, investigating scenarios and architectures which enable full exploitation of BT capabilities in industrial communication. Secondly, exploring technological enhancements which could improve the performance of the protocol even further.*

## 1. Bluetooth in industrial communication

The Bluetooth (BT) standard [1], originally developed for short-range ad-hoc wireless interconnection in Personal Area Networks, is becoming quite appealing in the industrial environment, where a number of applications, such as remote control and diagnostics, process supervision, etc. can benefit from replacing traditional wired connections with wireless ones [2][3][4]. 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 using a Time Division Multiple Access/Time Division Duplex (TDMA/TDD) scheme. Scheduling is handled by the Master, alternating a Master transmission with one by a designated Slave, based on 625- $\mu$ s slots. This means that scheduling occurs in pairs of slots (i.e., the Master/Slave pair).

BT offers several advantages for industrial communication, such as:

- **Easy installation.** BT can be successfully used to implement distributed control systems where cabling is difficult or only a temporary installation is needed.
- **Easy configuration.** BT allows for extremely easy configuration of wireless systems and stations, thus allowing the fast creation of communication infrastructures without the need for expensive cabling. Moreover, in the process

control environment, BT network set-up and configuration is lightweight as compared with the general case. As the Master and Slave roles are defined by the application, they are known a priori, so discovery procedures can be simplified and an operator can feed the Master node with the Slave node data during the configuration phase.

- **Centralized Medium Access Control.** The BT Master/Slave (M/S) centralized medium access control protocol avoids collisions inside a piconet and has a great potential for use in the control of transmission times, to the benefit of time-constrained traffic. The piconet Master can schedule traffic in a deadline-aware way according to well known real-time scheduling algorithms, also taking into account, already existing solutions in the literature, to deal with message release jitter, variable periods etc. For example, an innovative approach to support both periodic and aperiodic message transmissions on BT networks has been recently proposed in [10]. As it is shown in that work, the piconet Master can combine the Earliest Deadline First (EDF) [11] algorithm with a dynamic server mechanism such as the Total Bandwidth Server (TBS) [12][13] to guarantee deadline meeting for periodic messages while providing a best-effort service for aperiodic traffic exchanged on the same piconet.
- **Slotted Channel.** The slotted channel approach implemented in BT greatly simplifies clock synchronization between network nodes. The adoption of fixed-size slots may not be considered very efficient in the case of short payloads, but this could be solved by enabling a station to multiplex data in a single message.
- **Reduced interference.** Thanks to the frequency hopping scheme it implements, BT offers a notable resilience to interferences.
- **Low power.** BT features a fairly low level of energy consumption as compared with other wireless technologies (e.g. 802.11).

The above features explain the great interest in using Bluetooth in Distributed Process Control Systems (DPCSs), and provides the motivation for this work which has a twofold aim. The first is highlighting the

capabilities and suitability of BT networks for supporting control loops in DPCSSs. The second is investigating a novel communication model, based on Slave/Slave communication, and suitable network architectures based on such a model, which improve the performance of the BT protocol in industrial communication even further. In order to show the advantages of introducing Slave/Slave communication in the BT protocol, a case study is presented, in which two sample network configurations based on Slave/Slave communication are examined and compared. The case study shows the benefits that Slave/Slave communication and the derived architectures introduce in BT-based industrial communication networks. Finally, the paper addresses the benefits for real-time periodic traffic, in terms of enhanced schedulability, introduced by S/S communication through analytical argumentations and an example.

## 2.2. Architectures and techniques to exploit Bluetooth capabilities in industrial communication

Fig. 1 shows a typical scenario for DPCSSs. A PLC polls sensors and processes their data to derive the command for the actuator. The BT communication protocol fits this simple and quite popular control architecture very well.

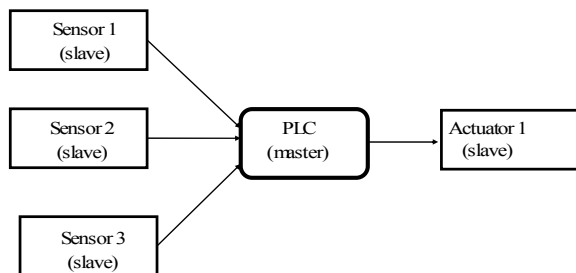


Fig.1 A typical control scheme for DPCSSs

A piconet can be built connecting the PLC to the Master node, all the other devices being Slaves. Within a scan cycle, the Master polls the sensors in sequence, thus enabling them to transmit their data, and then sends the command in the payload of a frame sent to the actuator. Fig.2 shows a simple schedule for the piconet shown in Fig.1.

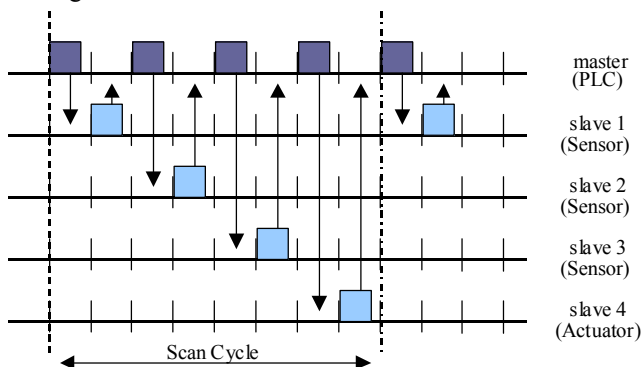


Fig.2 Schedule for the piconet supporting the control scheme in Figure 1

As said before, the piconet Master polls each Slave thus enabling it to transmit. This straightforward approach however has some shortcomings. If only either the Master or the Slave has data to send, the BT MAC protocol 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 DPCSSs, where exchanged messages, especially those featuring stringent timing constraints (i.e. periodic variables), are small in size. The transmission overhead also affects the message delay experienced as seen in the scheduling sequence shown in Fig.2.

These limitations could be solved by introducing Slave/Slave (S/S) communications. The current BT protocol does not allow for direct S/S communications, so if a pair of Slaves (e.g. intelligent sensors/actuators) wish to communicate, they can only do so through the Master. However, in DPCSSs periodic traffic is quite common and the Slaves' transmission requirements are known a priori, so allowing direct communications between Slaves is in principle possible and beneficial, as it would significantly reduce the protocol overhead and save bandwidth. S/S communication would also shorten the scan cycle duration, thus enabling the protocol to support faster dynamics.

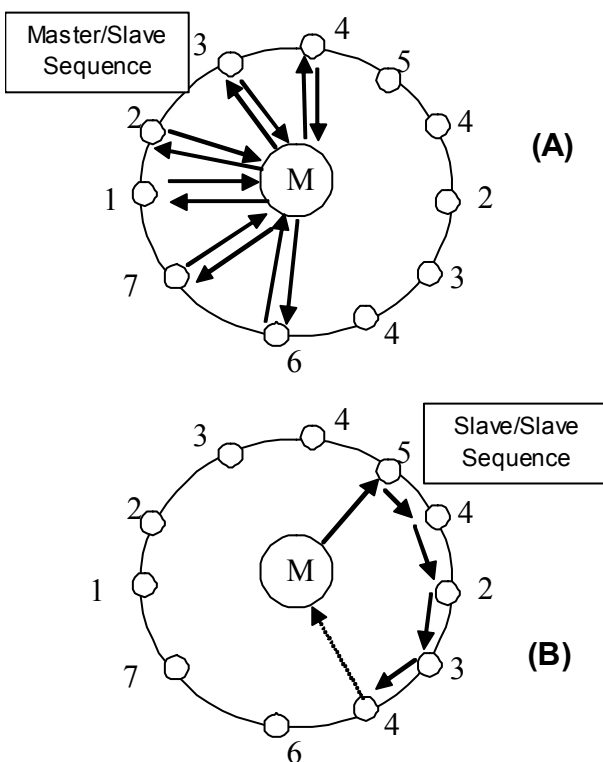
S/S communications, which are not currently provided for in the Bluetooth standard, were recently proposed in [5][6][7][8]. Zhang et al in [7], for example, proposed an approach called *Time-Slot Leasing (TSL)*, where temporary piconets *lease* slots from the original piconet to support S/S communications. This approach permits the existence of multiple temporary piconets, thus enabling multiple communications at the same time. Temporary piconets remain synchronized as regards both time and frequency with the original Master, whose task it is to designate a Slave as the *temporary master* of a temporary-piconet.

Another interesting approach to S/S communication in BT was presented in [5], where the concept of dynamic structure/role management was proposed. According to this approach, when a Slave needs to communicate frequently with another Slave, a Master/Slave switching or a piconet partitioning is activated. An improvement on this technique was presented in [6], where Pseudo Role Switching and Pseudo Partitioning techniques were proposed to reduce the switching bottleneck of the previous approach.

However, all these approaches only enable communications between pairs of Slaves, thus restricting the applicability of such techniques to file transfer or similar applications. On the contrary, the approach proposed in [8], adopted here to support some of the scenarios and architectures dealt with in this paper, allows Slaves to communicate with any member of a group of Slaves. This feature makes the proposed

approach highly suitable for supporting the typical communication exchanges in DPCSs.

An issue which could arise refers to how to implement S/S communication. In fact, the addition of the S/S operating mode should not preclude the presence of traditional devices and compatibility with devices operating in the traditional Master/Slave mode has to be maintained. No significant modifications to the standard should be required and changes should as far as possible, be confined to the software. These two goals are addressed by the approach proposed in [8], where S/S transmission takes place by means of scheduling handled by the Master, which has previously to configure a group of Slaves as belonging to a certain logical ring. The Master specifies the order of the various Slaves in the Ring via a Broadcast message and then transfers 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. This can all be done in a single Broadcast message. As can be seen in Fig.3A, the Master communicates with some nodes (6, 7, 1, 2, 3 and 4 in the example) in sequence in the M/S mode (according to its scheduling) and then authorizes S/S communication between nodes 5, 4, 2, 3 and 4, which have previously been configured as a logical ring (see Fig.3B).



**Fig.3 – Master/Slave communication (A) and Slave/Slave Communication (B)**

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 in the slot in which they possess the virtual token, i.e. when it is their turn according to the schedule that the Master has broadcast. 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 in the BT standard maintains synchronization between the various nodes and prevents overlapping between two consecutive transmissions. Each station can address a message to any (active) Slave in the virtual ring according to local scheduling. When the logical ring is over, the Master resumes normal M/S transmission based on its own scheduling. Even in the S/S operating mode, scheduling is always up to the Master, which has to acquire the knowledge needed for correct bandwidth allocation in the logical 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 by an operator or acquire a configuration file from a database. Once the Master has acquired the time requirements of all the traffic, it can perform a scheduling analysis and compute a suitable scheduling table, containing the time slots assigned to each S/S transmission. The transmission sequence is repeated after a time defined on the basis of the control dynamics. Due to the need to build a static schedule, the S/S communication 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. The non-mobility constraint is not, however, a real drawback in DPCSs, where devices are rarely moved once allocated and mobility is not an issue.

## 2.1 Implementation details for S/S Communication

S/S communication has to be compatible with Master/Slave communication and co-exist with it in a single piconet. It will thus be the task of the Master to establish which Slaves to enable for S/S transmission, when to activate the communication and for how long. It is therefore always the Master which determines the order of transmission in the logical ring, maintaining its role as the only traffic manager in the piconet.

Once it has collected, during the start up phase, the information required to establish scheduling in the piconet, the Master will set up and schedule the logical ring, configuring the various Slave units to be included in it. During normal execution, all transmissions in the piconet take place in the M/S mode, according to the scheduling algorithm chosen by the Master. When the Master decides to set up the logical ring, on the basis of the information it possesses, it sends the Slave units a Broadcast message.

Fig. 4 shows the format of the Broadcast message.

|                      |     |        |      |           |
|----------------------|-----|--------|------|-----------|
| AM_ADDR <sub>1</sub> | ... | N_SLOT | SKEW | TIME_SLOT |
|----------------------|-----|--------|------|-----------|

**Fig. 4 Broadcast message fields**

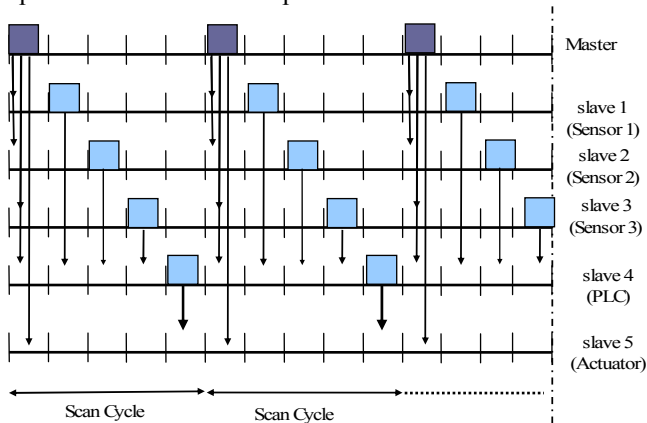
**AM\_ADDR<sub>i</sub>**: the address list giving the scheduling of the Slaves belonging to the ring (i.e. the sequence of the virtual token).

**N\_SLOT**: the number of slots the S/S sequence will be repeated before the M/S mode is resumed.

**SKEW**: this is a key parameter. It represents the shift in the frequency hopping sequence needed to avoid collisions with nodes operating in the M/S mode. The S/S ring, in fact, may operate in parallel with the main piconet (which remains in the M/S mode), so in order to avoid collisions, transmissions on the logical ring have to take place at frequencies other than those used for M/S transmissions.

**TIME\_SLOT**: the number of slots during which the Slaves will have to wait before starting the S/S sequence.

Under the hypothesis assumed here – that messages exchanged in the logical ring are one slot long – a one-hop shift in the hopping sequence is sufficient to avoid collisions. A shift in the hopping sequence only requires a modification to the Clock OFFSET value which is used to maintain synchronization with the Master [1]. 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 is more beneficial than having multiple unsynchronized collocated piconets close to one another. The impact of co-channel interference from other BT piconets on the packet error probability is addressed in [9]. This aspect is particularly critical in environments, like the factory floor, where multiple wireless networks operate in a common air space.



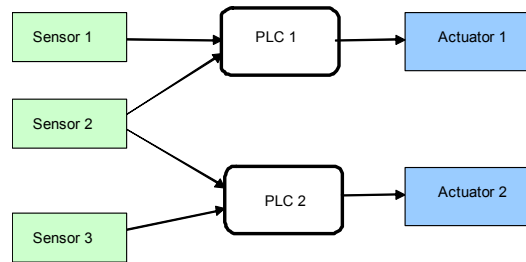
**Fig.5 Schedule for the piconet supporting the control scheme in Figure 1, using the S/S approach**

The S/S transmission sequence can be repeated a certain number of times (N\_SLOT), as established by the Master. As can be seen, it is possible to generate long S/S transmission sequences without loss of synchronization with the Master. For further details, the reader is referred

to [8]. 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 thus restored and the Master can resume transmitting to all of them in the M/S mode, according to the scheduling algorithm being used. Fig. 5 shows the schedule built for the piconet supporting the control scheme in Figure 1, with the S/S approach.

### 3. Case study

We will show the benefit of the proposed approach in a typical control scenario, with a network made up of two PLCs, three sensors and two actuators, like the one in Fig. 6. Sensor 2 is shared between the two PLCs. This arrangement can be made to avoid unnecessary replication of components in the control system. A shared sensor, however, places some constraints on traffic scheduling, as it must be included in the schedules of both PLCs. In this case, two choices are possible. The first choice is to build a scatternet made up of two piconets. In each piconet, one of the two PLCs is the Master, while Sensor 2 is a Slave



**Fig. 6. Case-study network**

acting as bridge between the two piconets. However, this solution has some drawbacks. First, it is difficult to schedule traffic exchanges involving the shared sensor, as it belongs to two different piconets and thus has to meet the timing requirements of both PLCs (e.g. periodic transmission of data according to the scan cycle of each PLC). Scheduling calculation is further complicated by the time overhead due to the sensor switching back and forth from one piconet to another, which may impact on time-constrained traffic. Finally, building several different piconets, operating close to one another in common air space, may be critical due to the potential for co-channel interference.

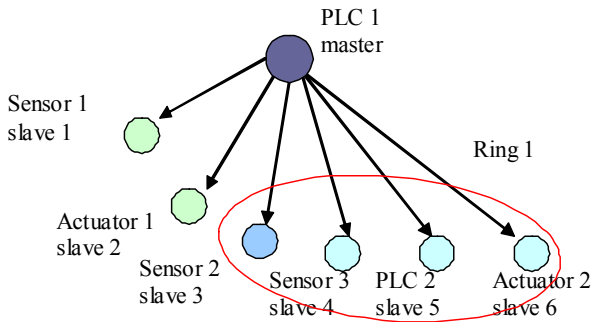
An alternative solution is to organize all the devices in a single piconet working according to the traditional M/S approach. However, this solution also suffers from some limitations, i.e. the overhead introduced by the slot sent by the Master to enable the Slave to transmit, which introduces a 50% overhead if the Master does not have payload data to transmit. In the case study addressed here, this happens when the PLC polls the sensors, to the detriment of protocol efficiency.

The limitations of the above mentioned solutions provide the motivation for a third approach, which

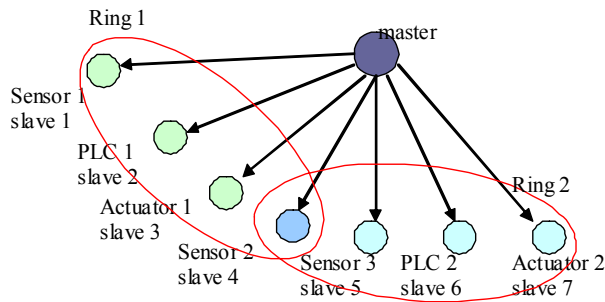
consists in organizing a single piconet where nodes operate combining M/S and S/S modes.

Two different configurations may be adopted. In the first one, henceforward called Configuration A and depicted in Fig.7, one PLC is the piconet Master, while the remaining devices are Slaves. In this case, the piconet Master is the PLC, so it is an internal device in the control loop. For this reason this configuration will henceforward be called Configuration A - Internal Master. In the second configuration, shown in Fig.8, and henceforward called Configuration B- External Master, all the devices in the control system are Slaves, while the Master node is a device not involved in the control loop which is in charge of handling initial set-up and scheduling.

In both configurations, the Master has to schedule Sensor 2 twice, as it produces data which is used by two different consumers, i.e. PLC 1 and PLC 2. Referring to Fig. 7, the first data pulled out of Sensor 2 will be consumed by the Master (i.e. PLC 1), while the second sample will be forwarded to PLC2, which is involved in



**Fig. 7: Configuration A: Internal Master**



**Fig. 8: Configuration B: External Master**

S/S communications over the logical ring. In Configuration B, the Master has to forward the data received from Sensor 2 first to one PLC and then to the other one, as will be shown below in Fig.13.

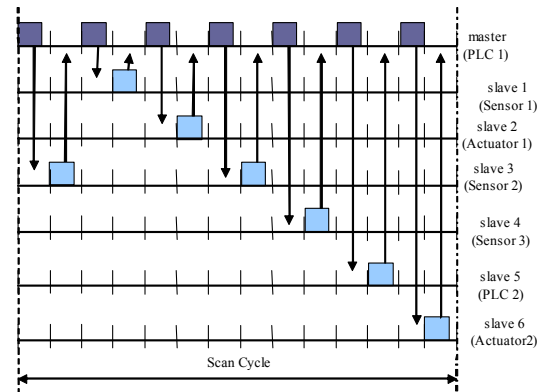
The piconet traffic is distributed as shown in Table 1, which describes the traffic flows and highlights the data exchange in terms of producer/consumer relationships.

| Producer | Consumer   |
|----------|------------|
| Sensor 1 | PLC 1      |
| PLC 1    | Actuator 1 |
| Sensor 2 | PLC 1      |
| Sensor 2 | PLC 2      |
| Sensor 3 | PLC 2      |
| PLC 2    | Actuator 2 |

**Table 1: Cyclic traffic distribution**

### 3.1 Configuration A: Internal Master

In this configuration, the piconet comprises 7 units, including the Master (i.e. PLC 1). A possible M/S schedule for this piconet is shown in Fig. 9. As can be seen, in this case the scan cycle is 14 slots long, as each traffic exchange always requires two slots. This straightforward schedule is therefore not very efficient in terms of bandwidth exploitation, but is shown here for comparison purposes.



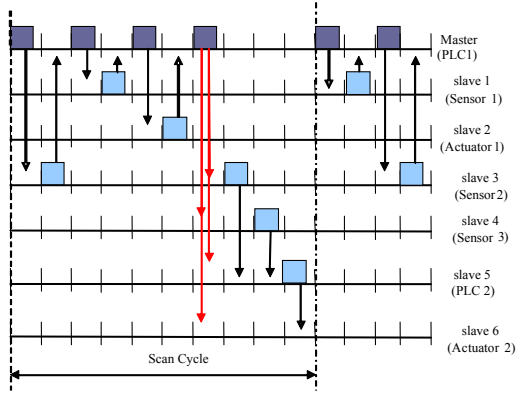
**Fig.9: Configuration A- Scan cycle for M/S communication**

The configuration shown in Fig.7 can be more effectively scheduled combining M/S and S/S communication. In this way, the original piconet is split into:

- a smaller piconet, working in the M/S mode, made up of PLC 1, Sensor 1, Sensor 2 and Actuator 1, and
- a logical ring, which comprises PLC 2, Sensor 2, Sensor 3 and Actuator 2, working according to the S/S communication model.

As Fig. 10 shows, this combined approach reduces the scan cycle to 10 slots.

The only limitation is due to Sensor 2, which limits the level of parallelism between the piconet and the logical ring, because it has to work with both PLCs. In this configuration, therefore, the Master first works in the M/S mode with the Slaves involved in the control loop of PLC 1, and then sends the broadcast message to all the other Slaves thus setting up the logical ring. As a result, in this configuration, M/S and S/S communications cannot occur in parallel, but in strict alternation.

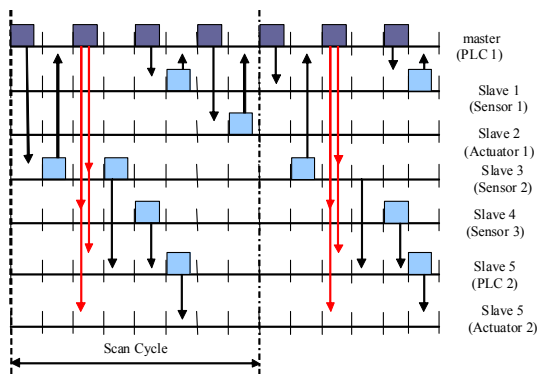


**Fig.10: Configuration A- Scan cycle for S/S communication**

As there is no need for the members of the logical ring to exchange data with the Master, with the only exception of keeping themselves synchronized with it, in this scheme the control of the logical ring goes back to the Master at the end of each scan cycle. This means that all the piconet Slaves, including the ones which were involved in the logical ring, realign themselves on the same frequency and hopping sequence as the Master. It is then up to the Master to set up the next cycle of the logical ring, preparing and sending a new broadcast message to the relevant Slaves.

Further improvements in terms of reduced scan cycle can be obtained with careful planning of the schedule. For example, Fig. 11 shows a schedule where the Master activates the logical ring after reading Sensor 2 data. In this way, the logical ring can work in parallel with the remaining part of the originary piconet, thus fully exploiting the benefits of introducing the S/S communication model.

In this way the scheduling cycle reduces to 8 slots. The result is notable, as it means that the scan cycle can be significantly shortened as compared with that achievable in a single piconet using M/S communications only.

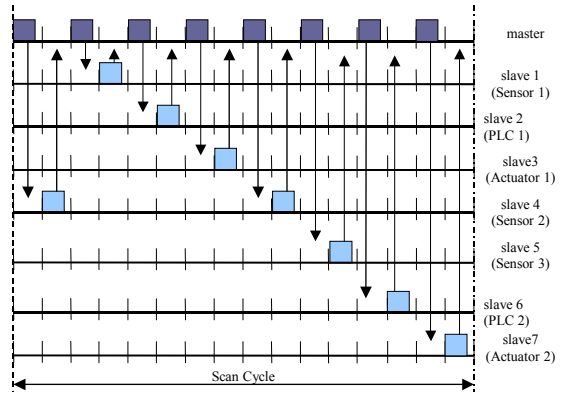


**Fig.11: Configuration A- Scan cycle for S/S and M/S communications occurring in parallel**

### 3.2 Configuration B

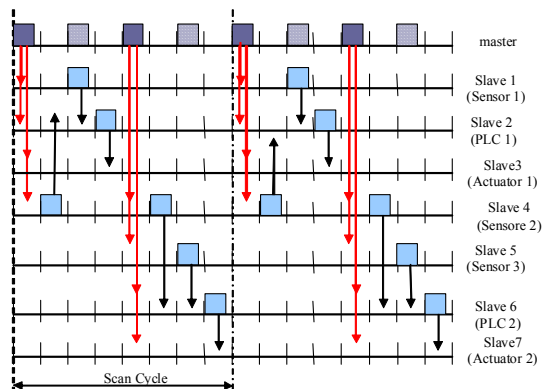
In Configuration B, the two PLCs work inside different logical rings, while the Master, which is not involved in any control loop, only performs logical rings configuration and scheduling activities.

Fig. 12 shows the scan cycle obtained for Configuration B working in the M/S communication mode.



**Fig.12: Configuration B- Scan cycle for M/S communication**

As the figure shows, the external Master has to poll 8 Slaves (as Sensor 2 has to be polled twice), so it takes 16 slots to complete the scan cycle. Fig. 13 on the other hand, shows the scheduling obtained for the same configuration using S/S communication. As it can be seen, in this case only 8 slots are needed to perform the complete scan cycle. Although the existence of Slave (Sensor 2) shared between the two logical rings does not allow for real parallelism between them, the duration of the scan cycle is halved as compared with that obtained in the M/S mode. This proves the efficiency of the S/S approach. Such a reduced scan cycle also allows faster control dynamics to be supported, thus proving beneficial for the control application.



**Fig.13: Configuration B- Scan cycle for S/S communication**



### 3.3 Comparison between Configuration A and Configuration B

A comparison between Configuration A and B shows that the former outperforms the latter when both work in the M/S mode. On the other hand, combining both M/S and S/S communication, the duration of the scan cycle is significantly reduced in both configurations. As said before, a shorter scan cycle enables the network to support faster control loops and, as it will be shown in Sect.4, improves real-time traffic schedulability. It should be emphasized that the delays introduced by the single devices in processing the relevant data have not been considered. For example, the execution time of the algorithm the PLC implements has not been taken into account. However, this does not alter the results obtained, as this delay is the same in all the configurations, working in either M/S or the S/S mode, so it can be considered a constant offset to be added.

Another interesting point to highlight is the reduced ‘thinking time’ introduced in the scan cycle when S/S communication is used instead of M/S communication.

When, in M/S communication, data exchanged between two Slaves has to go through the Master, there is an extra reception/transmission overhead introduced by the baseband and the protocol stack (which include correctness checking, buffering times, packet assembling, etc.) inside the Master node. This overhead has to be taken into account in building the schedule to meet any application constraints on the duration of the scan cycle. When the Master receives a packet to be forwarded to another slave, a too long thinking time could prevent the transmission of the packet in the slot immediately after the received slot. By using the S/S communication, this overhead inside the Master is avoided, as the communication is direct, and only the delays introduced by the two Slaves have to be considered. This point further stresses the advantage of, and the need for, S/S operating modes in the BT protocol.

As far as fault tolerance is concerned, if the channel used by one of the slaves during its transmission is subject to interference, the on-going packet is lost, but the system continues to work because the virtual token mechanism is not affected by such an error. Moreover, it has to be underlined that usually, in process control applications, when a sample is lost, it is preferable to disregard it instead of trying to retransmit the data, in order to avoid modifications in the transmission schedule. The same approach holds in M/S scheme, where the Master could retransmit a lost message, but at the cost of consuming additional slots, thus extending the schedule cycle duration.

From what has been said above, one could infer that the approaches in Figs. 7 (Configuration B -Internal Master) and 8 (Configuration B- External Master) are equivalent from the point of view of the results obtained when S/S communication is introduced. This does hold for the scenario considered here (Fig. 6), but different scenarios are possible where complex processes may

require the co-existence of several simple control loops. This is the case for smart sensors/actuators handling a control loop without the need for a PLC. In these scenarios, the External Master Configuration, where the Master is not involved in the control loop, is more effective than the Internal Master one, as in the first case the Master only has to handle logical rings scheduling, so it has more freedom in choosing when to send the Broadcast messages to the relevant rings, thus better serving/meeting the requirements of the different control loops. On the other hand, in configuration B the Master schedules only the activity of the network without taking part in the process control. In this case it is an additional device that has to be counted up in the number of the piconet active devices.

### 4. Benefits for real-time periodic traffic introduced by S/S communication

In this Section we will show how S/S communication is very beneficial even from the real-time schedulability point of view. Here we deal with message transmission, which is non-preemptive, so we can refer to non-preemptive processor scheduling literature (in our case, bandwidth is the scheduled resource). In [14] Jeffay et al. showed that a set of  $n$  periodic tasks is schedulable according to the non-preemptive EDF algorithm, iff two conditions hold [14]. The first one is relevant to the processor utilization, while the second one refers to the processor demand. Here we focus on the first condition, which can be expressed as

$$U_{tot} \leq 1 \quad (4.1)$$

where  $U_{tot}$  is the utilization factor for periodic traffic, defined as

$$U_{tot} = \sum_{i=1}^n U_i = \sum_{i=1}^n \frac{C_i}{T_i} \quad (4.2)$$

where  $U_i$  is the utilization factor for a given periodic variable  $i$ ,  $C_i$  is the relevant transmission time and  $T_i$  the period. As BT is a time-slotted protocol, here both  $C_i$  and  $T_i$  will be henceforward expressed in slots.

According to the BT M/S operating mode,  $U_{tot}$  is determined taking into account that each periodic transmission from a Slave also entails the transmission of the relevant polling slot from the Master. 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  $n$  slaves,  $U_{tot}$  is given by:

$$U_{tot} = \sum_{i=1}^n U_i = \sum_{i=1}^n \frac{(C_{Si} + C_M)}{T_i} \quad (4.3)$$

where  $C_{Si}$  is the transmission time for the periodic variable generated by Slave  $i$  and  $C_M$  is the Master polling slot transmission time.

From formula (4.3) it is evident that the polling slot the Master sends at each slave affects the overall system utilization  $U_{tot}$  and may impact on schedulability according to condition (4.1). On the other hand, the S/S operating mode reduces  $U_{tot}$ , thus favoring schedulability, as the Master sends only the slot containing the broadcast message with the logical ring configuration and then only Slave transmissions occur. Assuming that in a piconet with  $n$  slaves only  $k$  slaves work according to the M/S mode, while the remaining  $n-k$  are involved in the logical ring, the overall system utilization obtained using S/S communication is:

$$U_{tot} = \sum_{i=1}^k \frac{(C_{Si} + C_M)}{T_i} + \frac{C_B}{M_B} + \sum_{i=k+1}^n \frac{C_{Si}}{T_i} \quad (4.4)$$

where  $C_{Si}$  is the transmission time for the periodic variable generated by Slave  $i$ ,  $T_i$  is the relevant period,  $C_B$  is the transmission time (1 slot) for the broadcast message sent by the Master and  $M_B$  is the interarrival time between two consecutive broadcast messages sent by the Master. Similarly to  $C_i$  and  $T_i$ , even  $C_B$  and  $M_B$  are expressed in slots. As the broadcast message already provides in the  $N\_SLOT$  field the number of times (in slots) the S/S sequence has to be repeated before slaves in the logical ring resume the M/S mode, a new broadcast message has to be sent only when either the Master has to change (or quit) the logical ring due to specific application requirements (this may seldom occur) or before M/S synchronization is lost. As we calculated in [8], it is possible to generate long S/S transmission sequences before synchronization is lost (about 800 slots). In both cases, the contribution of the second addend in formula (4.4) is negligible. So, as shown in formula (4.4), the higher the number of devices in the logical ring (i.e. the smaller  $k$ ), the lower the utilization factor is. This shows that S/S communication, as compared with the M/S case, not only saves bandwidth, but also favors schedulability (according to formula (4.1)) and allows for a better exploitation of the piconet, thus enabling a higher number of periodic variables being exchanged on the same piconet, and thus a higher number of control devices being supported.

We made some tests using periodic message sets. Results from utilization-based schedulability test showed that, in some cases, message sets which failed in satisfying formula (4.1) and so were not schedulable using the M/S operating mode, were able to satisfy (4.1) using S/S communication.

Here we report a sample case. Tables 2 and 3 show the set of periodic variables chosen and the result of the utilization-based schedulability tests performed. In Table 2,  $U_{tot}$  is calculated according to formula (4.3), as the whole piconet works according to the M/S mode. In Table 3, formula (4.4) has been used, with  $k=1$ , as the piconet is split in such a way that the slave producing variable 1 works in M/S mode, while the slaves

producing variables 2 and 3 are involved in a logical ring working in S/S mode. In Table 3, the parameters of the Master broadcast message are also shown.

|       | $T_i$ (slot) | $C_i$ (slot)                | $C_M$ (slot) |
|-------|--------------|-----------------------------|--------------|
| Var.1 | 4            | 1                           | 1            |
| Var.2 | 3            | 1                           | 1            |
| Var.3 | 7            | 1                           | 1            |
|       |              |                             |              |
| Utot  | 1.45         | Utot>1<br>(not schedulable) |              |

**Table 2: Utilization-based schedulability test using M/S communication**

|                   | $T_i$ (slot) | $C_i$ (slot) | $C_M$ (slot) |
|-------------------|--------------|--------------|--------------|
| Var.1             | 4            | 1            | 1            |
| Var.2             | 3            | 1            | 0            |
| Var.3             | 7            | 1            | 0            |
|                   |              |              |              |
|                   | $M_B$ (slot) | $C_B$ (slot) |              |
| Broadcast message | 100          | 1            |              |
|                   |              |              |              |
| Utot              | 0.986        | Utot<1       |              |

**Table 3: Utilization-based schedulability test using S/S communication**

In the S/S mode, the value of 100 chosen for  $M_B$  (the interarrival time of the broadcast message) is just an example of a possible one, and is well lower than the threshold beyond which M/S synchronization is lost (about 800 slots) [8]. A higher  $M_B$  value would improve schedulability even further, thus allowing the addition of new periodic variables in the piconet.

We chose short periods for the variables in Tables 2 and 3 to show the S/S communication potential to serve periodic variables with quite fast dynamics.

## 5. Conclusions

This paper has investigated the capabilities and suitability of BT networks for supporting control loops in DPCSs. It has been shown that BT offers several interesting features, such as centralized medium access control, slotted time, clock synchronization and frequency hopping, which enable it to effectively support industrial applications. One shortcoming is the protocol overhead due to the strict M/S communication model, which could be overcome if direct S/S communication is allowed. This type of communication could be realized by implementing a logical ring on the piconet, working either in parallel or in strict alternation with the M/S operating mode, according to the application needs. The paper has also shown the advantages in terms of reduced scan cycle which can be obtained by organizing traffic on different logical rings co-existing on the same piconet using different configurations. These configurations have



been proved to be beneficial from several perspectives, as they allow easy data exchange between devices in the same piconet, improve protocol efficiency, and avoid the proliferation of a large number of piconets working close to each other, thus limiting the potential for interference when complex control systems require the handling of several regulation loops. Finally, the paper has analytically addressed the benefits for real-time periodic traffic in terms on enhanced schedulability, also providing an example.

It should be emphasized that the S/S operating mode described in the paper is a superset of the functionalities already present in the BT standard and can co-exist with them. The implementation of new functionalities does not require any HW modification to the BT Baseband chip. Only modifications in the firmware are needed to realize the functionalities needed for S/S communication. These modifications could be steadily implemented in the BT chip and then used by applications which can take advantage of them.

## References

- [1] Bluetooth SIG. Specification of the Bluetooth System – Version 1.1B, Specification Vol. 1 & 2, February 2001.
- [2] P.A. Wiberg, U. Bilstrup. Bluetooth in Industrial Environment. In Proc. of IEEE WFCS'00, Porto, Portugal, September 2000.
- [3] P.A. Wiberg, U. Bilstrup. Wireless Technology in Industry- Applications and User Scenarios. In Proc. of IEEE ETFA'01, pages 123–131, Antibes - Juan les Pins, France, October 2001.
- [4] Leine & Linde. Wireless health monitoring systems for encoders, July 2001. <http://www.leinelinde.se>.
- [5] W. Zhang, H. Zhu, G. Cao. On Improving the Performance of Bluetooth Networks through Dynamic Role Management. Technical Report CSE-01-018, Pennsylvania State University.
- [6] S. Abhyankar, R. Toshiwal, C. Cordeiro, D. Agrawal. On the Application of Traffic Engineering over Bluetooth Ad Hoc Networks. ACM MSWiM'03, San Diego (CA), Sept. 2003.
- [7] W. Zhang, H. Zhu, G. Cao. Improving Bluetooth Network Performance through a Time-Slot Leasing Approach. In Proceeding of the 2002 IEEE Wireless Communications and Networking Conference WCNC 2002, Orlando, Florida, March 2002.
- [8] L. Lo Bello, N. Torrisi, O. Mirabella. New operating Modes for Bluetooth Networks in Distributed Process Control System. In Proc. of IEEE LCN'04, Tampa, Florida, USA, November 2004.
- [9] A. El-Hoiydi, J. Decotignie. Soft deadline bounds for two-way transactions in Bluetooth piconets under co-channel interference. In Proc. of IEEE ETFA'01, Antibes- Juan les Pins, France, October 2001.
- [10] L. Lo Bello, M. Collotta, O. Mirabella, T. Nolte. Approaches to support Real-Time traffic over Bluetooth. In Proceed. of the 4<sup>th</sup> Intl. Workshop on Real-Time Networks, RTN 05 Satellite Workshop of the 17<sup>th</sup> Euromicro Intl. Conference on Real-Time Systems, ECRTS 05, Palma de Mallorca, Spain, July 2005.
- [11] C. L. Liu, J. W. Layland. Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment. Journal of ACM, Vol. 20, No. 1, Jan. 1973.
- [12] M. Spuri, G. C. Buttazzo. Efficient Aperiodic Service under Earliest Deadline Scheduling. In Proceedings of the 15th IEEE Real-Time Systems Symposium, San Juan, Puerto Rico, 1994.
- [13] M. Spuri, G. C. Buttazzo. Scheduling Aperiodic Tasks in Dynamic Priority Systems. Journal of Real-Time Systems, Vol. 10, No. 2, pp. 1-32, 1996.
- [14] K. Jeffay, D.F. Stanat, C.U. Martel. On Non-Preemptive Scheduling of Periodic and Sporadic Tasks. In Proceedings of the 12th IEEE Real-Time Systems Symposium, pp.129-139, San Antonio, TX, Dec. 1991.

