An Adaptive TDMA Protocol for Soft Real-Time Wireless Communication among Mobile Autonomous Agents

Frederico Santos
DEE, Instituto Politécnico de Coimbra
Coimbra, Portugal
fred@mail.isec.pt

Luís Almeida, Paulo Pedreiras, Luís S. Lopes
LSE-IEETA/DET, Universidade de Aveiro
Aveiro, Portugal
{lda,pedreiras,lsl}@det.ua.pt

Tullio Facchinetti
DIS, University of Pavia
Pavia, Italy
tullio.facchinetti@unipv.it

Abstract

Interest on using mobile autonomous agents has been growing, recently, due to their capacity to cooperate for diverse purposes, from rescue to demining and security. Such cooperation typically requires the exchange of state data that is time sensitive and thus, applications should be aware of temporal data coherency. In order to provide such information a real-time communications protocol must be used to prevent unbounded latency of message delivery. This is, however, not a trivial task when using wireless links because of their poor reliability properties, exhibiting high bit error rates, omissions and inconsistencies. This paper describes such a communication protocol that adapts itself to the current conditions of the medium attempting to deliver an adequate timeliness while using the least bandwidth. The protocol operates over IEEE 802.11 networks in both managed and ad-hoc modes and it is fully distributed. It has been developed within the framework of the CAMBADA project at the University of Aveiro, Portugal, which aims at developing a robotic soccer team. This paper describes the protocol and presents some preliminary experimental results.

1. Introduction and related work

Coordinating several autonomous mobile robotic agents in order to achieve a common goal is currently a topic of intense research [6][7]. This problem can be found in many robotic applications, either for military or civil purposes, such as search and rescue in catastrophic situations, demining or maneuvers in contaminated areas.

The technical problem of building an infrastructure to support the perception integration for a team of robots and subsequent coordinated action is common to the above applications. One relatively recent initiative to promote research in this field is RoboCup [2] where several autonomous robots have to play football together as a team, to beat the opponent. We believe that researching ways to solve the perception integration problem in RoboCup is also very relevant to real-world applications.

Currently, the requirements posed on such teams of autonomous robotic agents have evolved in two directions. On one hand, robots must move faster and with accurate trajectories to close the gap with the dynamics of the processes they interact with, e.g., a ball can move very fast. On the other hand, robots must interact more in order to develop coordinated actions more efficiently, e.g., only the robot closer to the ball should try to get it while other robots should move to appropriate positions. The former requirement demands for tight closed-loop motion control while the latter demands for an appropriate communication system that allows building a global information base to support cooperation. Both cases are subject to time constraints that must be met for adequate performance.

However, the wireless medium has a relatively low coverage of timeliness and reliability properties due to high bit error rates, omissions and inconsistencies, among other possible negative effects. The difficulty in assuring timely communication over wireless links has also been motivating a substantial research activity, lately, either in the scope of mobile ad-hoc networks [11], sensor networks [10] and wireless access to Internet-based multimedia communications [12]. Despite the existence of some solutions for each of these application scopes, the problem is still open in general, with a large number of possibilities in terms of operational flexibility, energy consumption, bandwidth usage and communication coordination, while, at the same time there is a lack of commercial off the shelf (COTS) solutions delivering any kind of real-time guarantees.

The Cortex project [7] deserves here a particular reference because it bears several resemblances in its purposes with our project [9], but using different architectural options and protocols. Basically, one of the purposes in both approaches is to adapt dynamically the behavior of real-time cooperating entities according to the timeliness of the information received through the wireless links. Two example applications developed within Cortex are presented in [8], which rely on the Timely Computing Base (TCB) [5] to provide time-related services such as measurement of intervals and detection of timing violations. However, while the TCB uses an out-of-band control channel to support timeliness information across entities we propose an in-band adaptive synchronization scheme that attempts to improve the timeliness of the communication on an unreliable medium, such as the wireless one, minimizing the overhead bandwidth.

This paper presents the protocol developed to interconnect the robotic agents that constitute the CAMBADA middle-size robotic soccer team of the University of Aveiro,

This work was partially supported by the Portuguese Government (project CAMBADA, POSI/ROBO/43908/2002) and the European Commission (accompanying measure ARTIST, IST-2001-34820).

Portugal [9]. We start in section 2 by briefly presenting the architecture of the robotic agents, including computing, software and communications architectures. Section 3 describes the wireless communication protocol used among agents and section 4 presents preliminary experimental results. Finally section 5 concludes the paper.

2. Overview of the team agents architecture

The computing architecture of the robotic agents follows the biomorphic paradigm [4], being centered on a main processing unit (*the brain*) that is responsible for higher-level behaviors coordination. This main processing unit handles external communication with other agents via a wireless IEEE 802.11b network and has high bandwidth sensors, namely 2 cameras, directly attached to it. Finally, this unit receives low bandwidth sensing information and sends actuating commands to control the robot attitude by means of a distributed low-level sensing/actuating system (*the nervous system*) (Fig. 1).

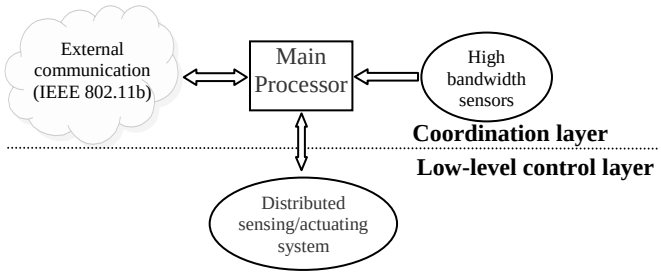


Figure 1. The biomorphic architecture of the CAMBADA robotic agents

The main processing unit is implemented either on a laptop or on a mini PC motherboard that runs the Linux operating system complemented with the RTAI kernel for timeliness support, namely for time-stamping, periodic transmissions and task temporal synchronization. This use of RTAI follows a similar paradigm as the Timely Computing Base proposed in [5]. The distributed sensing/actuating system uses a set of microcontrollers of the PIC18Fxx8 family interconnected by a Controller Area Network and handles specialized functions such as the closed-loop control of each motor speed, the holonomic attitude control, the kicker control and the odometry.

The software architecture is developed around a distributed real-time database (RTDB), as presented in [3], which holds the state data of the local agent together with local images of the state data of the other team members. Then, several Linux and RTAI tasks work over the RTDB updating its contents and defining the robot behavior at each instant, which is then transmitted to the lower control layer (Fig. 2). The replication of the state data of each robot in the RTDBs of the others supports an easy access to remote sensing favoring cooperative behaviors. Moreover, the access to remote sensing information is carried out locally with fast non-blocking functions. The communication system manages the refreshing of the data in an automatic way, in the background, by triggering the update transactions at an adequate rate, both at the coordination and low-level control layers. Such transactions are triggered under the control of the RTAI kernel so that the transmission instants are respected within small tolerances, contributing to achieve better closed-loop control of the robots motion. This is particularly important in the communication with the low-level control layer where periods in the range of few tens of milliseconds are used.

In what concerns the wireless communication, it is handled within Linux by a high-priority task, with SCHED_FIFO scheduler, due to unavailability of RTAI device drivers for certain wireless cards. Nevertheless, this task is also synchronized by RTAI. The periodicity of these transmissions is in the order of 100ms, a value that establishes a compromise between the bandwidth used by the system and the temporal coherency of the remote data inside the RTDB. Notice, however, that the requirement for temporal coherency of the remote data is not particularly stringent since it is not used within high-speed closed-loop control.

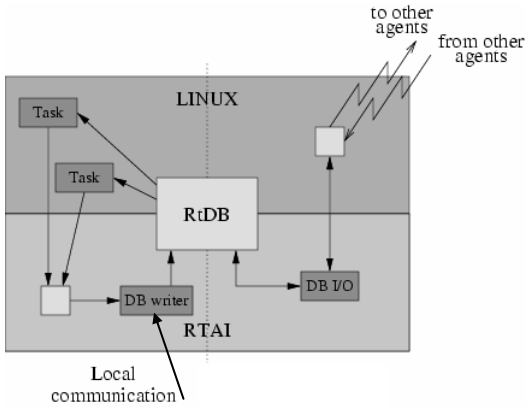


Figure 2. The main processor software architecture

An important feature is that the communication follows the producer-consumer co-operation model, according to which each robot regularly broadcasts, i.e. produces, its own data while the remaining ones receive, i.e. consume, such data and update their local structures (Fig. 3). In this paper we focus on the communication protocol that is used to refresh the remote state data of the RTDB.

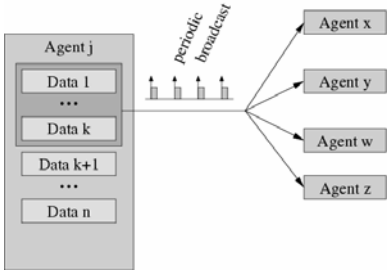


Figure 3. Each agent broadcasts periodically its subset of state data that might be required by other agents

3. Communication protocol among agents

As referred in the previous section, agents communicate using an IEEE 802.11 network, sharing a single channel with the opposing team and using managed communication

(through the access point). This raises several difficulties because the access to the channel cannot be controlled [1] and the available bandwidth is roughly divided by 2.

Therefore, the only alternative left for each team is to adapt to the current channel conditions and reduce access collisions among team members. This is achieved using an adaptive TDMA transmission control that uses the frames reception instants to setup and maintain the slot and round synchronization. The round has a predefined period called *team update period (Ttup)* that sets the responsiveness of the global communication. Within such round, there is one single slot allocated to each team member so that all slots in the round are separated as much as possible (Fig. 4). This allows calculating the target inter-slot period *Txwin* as *Ttup/N*, where N is the number of robots in the team.

The transmissions generated by each agent are scheduled within the RTAI task *DB_IO* (Fig. 2) according to the production periods specified in the RTDB records. Currently a rate-monotonic scheduler is used. When the respective TDMA slot comes, all currently scheduled transmissions are piggybacked on one 802.11 frame and sent to the channel.

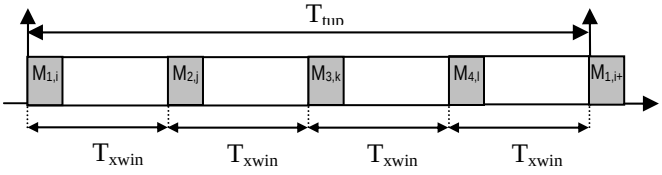


Figure 4. TDMA transmission control of wireless communication within the team

When a robot transmits at time t_{now} it sets its own next transmission instant $t_{next} = t_{now}+Ttup$, i.e. one round after. However, it continues monitoring the arrival of the frames from the other robots. When the frame from robot *k* arrives, the delay δ_k of the effective reception instant with respect to the expected instant is calculated. If this delay is within a validity window $[0, \Delta]$, with Δ being a global configuration parameter, the next transmission instant is delayed according to the longest such delay among the frames received in one round (Fig. 5), i.e.,

$$t_{next} = t_{now} + Ttup + max_k\,(\delta_k)$$

On the other hand, if the reception instant is outside that validity window, or the frame is not received, then δ_k is set to 0 and does not contribute to update t_{next}.

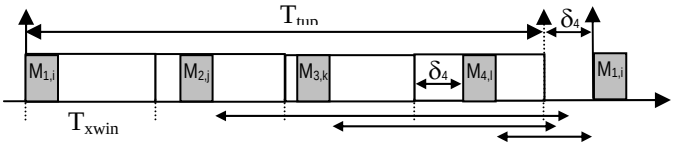


Figure 5. Adaptive TDMA synchronized on the reception of the frames

The practical effect of the protocol is that the transmission instant of a frame in each round may be delayed up to Δ with respect to the predefined round period *Ttup*. Therefore, the effective round period will vary between *Ttup* and $Ttup+\Delta$. When a robot does not receive any frame in a round within the respective validity windows, it updates t_{next} using a robot specific configuration parameter β_k in the following way

$$t_{next} = t_{now} + Ttup + \beta_k \quad \text{with} \quad 0 \le \beta_k \le \Delta$$

This is used to prevent a possible situation in which the robots could all remain transmitting but unsynchronized, i.e. outside the validity windows of each other, and with the same period *Ttup*. By imposing different periods in this situation we force the robots to resynchronize within a limited number of rounds because the transmissions will eventually fall within the validity windows of each other.

The reasoning behind the protocol definition is that when the medium is loaded, causing network-induced delays, the system reduces its transmission pressure by increasing the round period, instead of increasing the pressure as any retransmission protocol would do, further consuming the channel bandwidth. Nevertheless, the system always tries to bring the round period down to *Ttup*. Moreover, using broadcasts instead of unicasts has the benefit of strongly reducing the bandwidth used by the team while improving energy efficiency by reducing the number of transmissions.

This mechanism is adaptive in nature but within defined bounds so that we can still reason about the temporal coherency of the data inside the RTDB. Moreover, transmitting frames as far apart as possible makes the protocol more tolerant to deviations either caused by temporary loss of communication or by interference from other network load. Particularly, when this network load is periodic, e.g. related to control/streaming information, with harmonic period regarding *Ttup*, the protocol is able to adapt and completely avoid contention with that load.

4. Experimental results

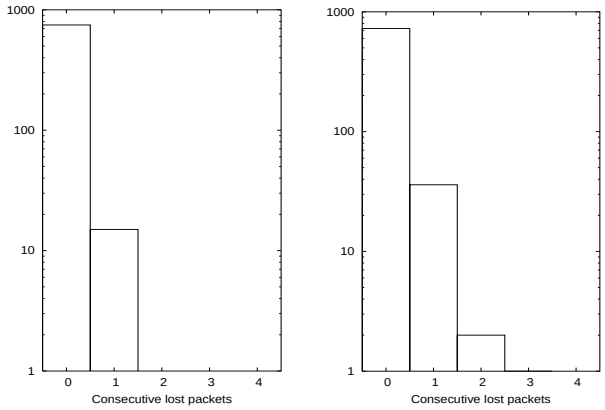
In order to test the protocol described in the previous section a few experiments were conducted. These are, however, preliminary results and more experiments will be carried out in future work. In this case, the experiments were conducted to assess the number of lost packets (recall that there is no collision detection with broadcast packets) with and without additional traffic load and with and without the synchronization scheme.

The experimental setup comprised 4 robots and a monitoring station time-stamping and logging frame receptions in promiscuous mode. The monitoring station did not transmit and was not included in the TDMA round. The round length was set to *Ttup* =100ms corresponding to an inter-slot period *Txwin*=25ms. The frames used in the protocol were all carrying 640B of data payload and were transmitted in raw mode, directly accessing the network driver. On the other hand, the additional traffic load was generated using the *ping* command addressed to the access point (AP) with 1000B of payload and at 200 packets/s. Due to space constraints, we present below the results referring to the transmissions from robot 1, only, as captured by the monitoring station. However, they were similar to the results obtained for all the other robots.

The initial tests were carried out with a clean environment, i.e. without any other source of traffic beyond the team. We captured near 150 seconds of traffic, with and without synchronization among robots. In the former case,

we captured 1430 packets without any loss, while in the latter case we captured 1482 packets, with 9 packets lost (0.6%). This result already indicates the benefit of synchronizing the transmissions of the team to reduce loss of packets, in this case caused, mainly by self interference within the team.

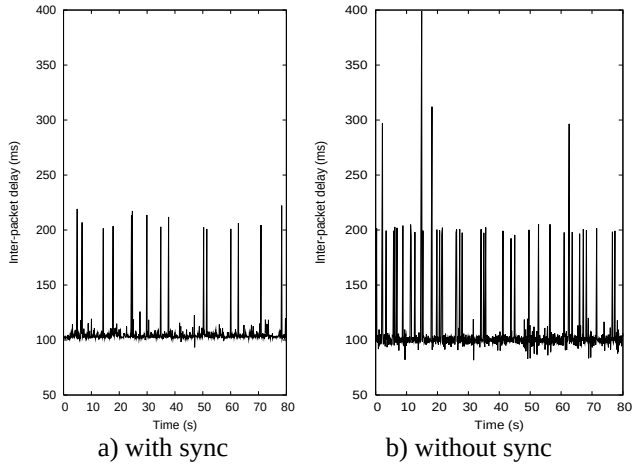
Then, we assessed the protocol under heavy additional traffic load, as quantified above. We captured near 80 seconds of traffic, again with and without synchronization. In the former case, 748 packets were captured while 15 were lost, representing a percentage of near 2%. On the other hand, without synchronization, we captured 726 packets and observed 39 packet losses (5.1%). Figure 6, a) and b), shows the respective histograms of the number of consecutive lost packets. As expected, this shows a higher benefit of using synchronization to improve the resilience of broadcast packets and consequently, the timeliness of the system.



a) with sync b) without sync

Figure 6. Histograms of the number of consecutive lost frame

Figure 7, a) and b), exhibits the timeline of the inter-packet delay corresponding to the histograms of Figure 6.



a) with sync b) without sync

Figure 7. Timeline of the inter-packet delay

5. Conclusion

Cooperating robots is a field currently generating large interest in the research community. RoboCup is one example of an initiative developed to foster research in that area. This paper refers to the CAMBADA middle-size robotic soccer team being developed at the University of Aveiro and briefly discusses the robots architecture. The focus of the paper is on the wireless communication protocol used among robots in the team, which is based on an IEEE 802.11b network. The protocol is presented as well as some preliminary experimental results that show the protocol properties.

The protocol is flexible to accommodate changes in the conditions of the wireless medium and in the communication requirements of the team. It is also energy and bandwidth efficient by reducing the number of transmissions while attempting to deliver adequate timeliness. More experiments are being carried out to better characterize the protocol performance and discuss its optimal configuration for a broader set of applications.

References

[1] Decotignie, J.-D., et al., "Architecture for the Interconnection of Wireless and Wireline Fieldbuses", FeT'01 - IFAC Conf. on Fieldbus Technologies, Nancy. November, 2001.

[2] Kitano, K., M. Asada, Y. Kuniyoshi, I. Noda, E. Osawa, "RoboCup: The Robot World Cup Initiative", *Proc. of IJCAI-95 Workshop on Entertainment and AI/Alife,* Montreal. 1995.

[3] Kopetz, H., "Real-Time Systems Design Principles for Distributed Embedded Applications", Kluwer, 1997.

[4] Proc. of the "NASA Workshop on Biomorphic Robotics", Jet Propulsion Laboratory, California Institute of Technology. USA. August 14 - 16, 2000.

[5] Veríssimo, P., Casimiro, A., "The Timely Computing Base Model and Architecture", IEEE Transactions on Computers, 51(8), August 2002.

[6] Weiss, G. "Multiagent systems. A Modern Approach to Distributed Artificial Intelligence" MIT Press, 2000.

[7] Veríssimo, P., V. Cahill, A. Casimiro, K. Cheverst, A. Friday and J. Kaiser, "Cortex: Towards supporting autonomous and cooperating sentient entities", in Proceedings of European Wireless 2002, pp: 595-601. Florence, Italy, Feb. 2002.

[8] Martins, P., P. Sousa, A. Casimiro, P. Veríssimo, "Dependable Adaptive Real-Time Applications in Wormhole-based Systems", Proceedings of the Int. Conf. on Dependable Systems and Networks, Florence, Italy, June 2004.

[9] Almeida, L., F. Santos, T. Facchinetti, P. Pedreiras, L. S. Lopes, "Coordinating distributed autonomous agents with a real-time database: the CAMBADA project", Proceedings of the ISCIS 2004, the 19th Int. Symp. on Computer and Information Sciences, Kemer-Antalya, Turkey, Oct. 2004.

[10] J. Stankovic, T. Abdelzaher, C. Lu, L. Sha, and J. Hou. Realtime communication and coordination in embedded sensor networks. In *Proceedings of the IEEE*, volume 91, pages 1002–1022, July 2003.

[11] C.-T. Chou, K.G. Shin. Analysis of Adaptive Bandwidth Allocation in Wireless Networks with Multilevel Degradable Quality of Service. IEEE Transactions on Mobile Computing, Vol. 3, N. 1, pp 5-17, IEEE Press, Jan. 2004.

[12] S. Mangold, S. Choi, P. May, O. Klein, G. Hiertz, L. Stibor, IEEE 802.11e Wireless LAN for Quality of Service (invited paper), in Proceedings of European Wireless 2002, pp: 25-28. Florence, Italy, Feb. 2002.