15

Using a Meeting Channel and Relay Nodes to Interconnect Mobile Robots

Nassima Hadid, Alexandre Guitton
and Michel Misson
*LIMOS CNRS / Clermont Université, Campus des Cézeaux, 63 173 Aubiere Cedex
France*

1. Introduction

Mobile robots are increasingly used in industrial applications (generally in order to convey merchandises), in healthcare applications (for example, to transport patients or medicines) or in domotic applications (for instance, to sweep a floor). These robots are often mounted with several sensors and actuators, in order to carry out their tasks (*e.g.*, grabbing an object of interest, and bringing it a destination without colliding with obstacles). To reduce the complexity and weight of the wiring of all these sensors and actuators, these devices can be equipped with wireless capability. A central entity, called the central unit, is usually in charge of collecting the data from the sensors, integrating part of the data in a models, computing a behavioral response and operating the actuators accordingly. Another entity, called the coordinator, is in charge of managing the wireless entities, retrieving data from sensors and sending orders to actuators. The same device can be simultaneously the central unit and the coordinator. In the following, the wireless communications between sensors, actuators and coordinator are called intra-robot communications.

Real applications often require the collaboration of mobile robots in order to perform required tasks. For instance, when they are used in airports to transport luggage, mobile robots can cooperate to avoid colliding with each other. In search and rescue operations, mobile robots can inform their neighbors that an area has already been explored (Ferranti & Trigoni, 2008). Robots can also follow tracks on the floor (Hogg et al., 2002) or monitor an area collaboratively (Lambrou & Panayiotou, 2009). An allegorical example is shown on Fig. 1, where the waiter robot *Dro* has to collaborate with the waitress robot *Ide* in order to serve hot coffee to customers. These communications used for collaborations are called inter-robots communications.

The design of a network architecture that allows mobile robots to cooperate efficiently, without negatively impacting the performance of each intra-robot communication, is a challenging issue, and is the aim of this chapter.

The remainder of the chapter is the following. In Section 2, we introduce a multi-channel approach where intra-robot communications are performed on a robot-specific channel, and inter-robot communications are performed on a common meeting channel. In this way, mobile robots can communicate without impacting intra-robot communications. In Section 3, we discuss the protocols that allow mobile robots to communicate directly, when they are

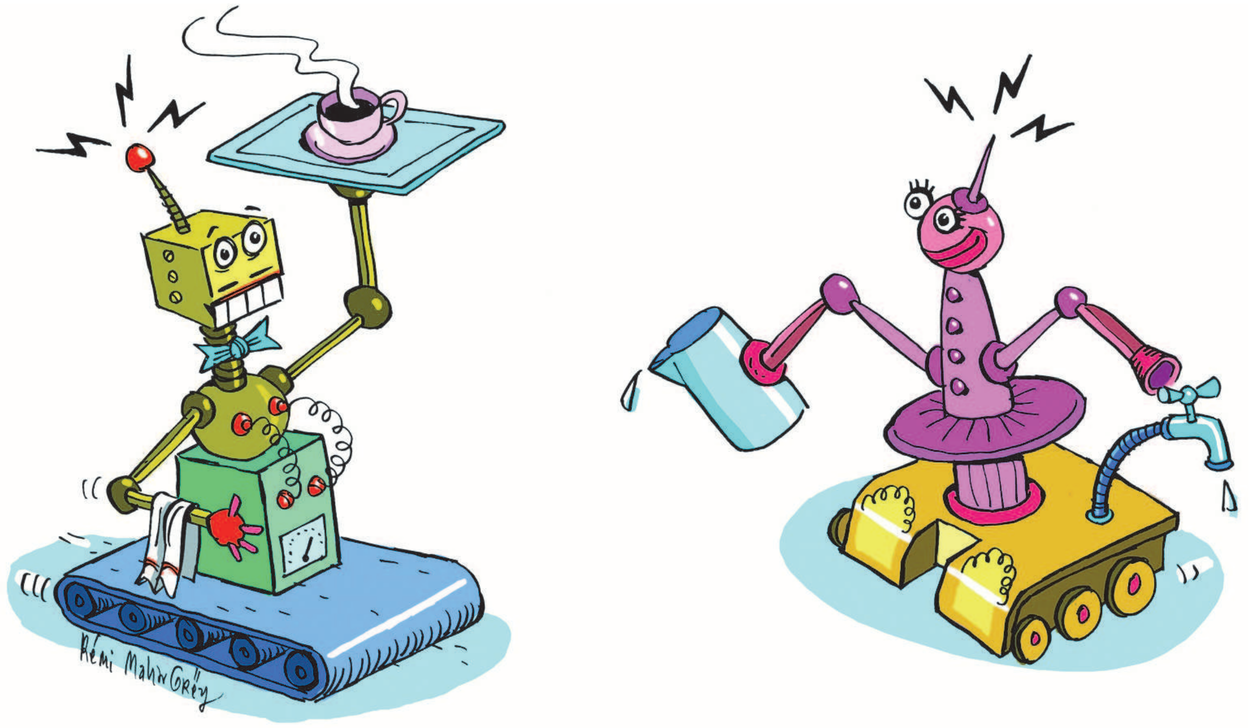


Fig. 1. Mobile robots *Dro* (on the left) and *Ide* (on the right) have to collaborate to serve hot coffee to customers.

in range of each other. These protocols are based on several mechanisms: a data storage mechanism (which allows data to be stored until the intended destination mobile comes in range), a neighbor discovery mechanism (which detects mobiles in range), an association mechanism (which allows neighbors to identify each other), and a data exchange mechanism (which transfers the data). All these mechanisms are described in details. The concepts behind these protocols come from delay tolerant networks (DTN), as the mobility of robots is often difficult to integrate to applications, or even unpredictable in some applications. In Section 4, we discuss how to achieve indirect communications between mobile robots. We explain how road-side units can help mobile robots to communicate. These road-side units can also be required in some applications, where a message has to be disseminated in a specified geographical area. A generic addressing mechanism is useful in this case, when the ID of the destination mobiles cannot be known beforehand. We show that the overhead of implementing the generic addressing is limited. Finally, in Section 5, we conclude this chapter by summarizing the benefits of the meeting channel.

2. Interconnection of mobile robots using a meeting channel

Most applications using mobile robots require the mobile robots to communicate together in order to perform the required tasks. In the following, we consider a generic application where mobile robots follow tracks and carry items (such as robots carrying luggage in airports). The mobile robots communicate in order to avoid collisions, and to detect priority when crossing intersections. Each mobile is equipped with sensors and actuators that communicate with the central unit embedded on the mobile.

Although the communications between mobile robots can be performed on the same communication channels as the intra-mobile communications, this tends to reduce the performance of the whole system. Indeed, if the communications between mobiles and the intra-mobile communications are on the same channel, the competition for the medium is high. Reducing the bandwidth of intra-mobile communications can have a disastrous impact on the robot behavior, for instance if the coordinator is not able to inform quickly the actuators (such as brakes) to be operated.

Thus, it is important to interconnect mobile robots using a multi-channel approach.

2.1 Multi-channel approaches

Multi-channel approaches have the main advantage to reduce the contention on each communication channel, and thus to improve the performance of the system. In the literature, several multi-channel approaches have been proposed (Bahl et al., 2004; Jovanovic & Djordjevic, 2007; Nasipuri et al., 1999; Pathmasuntharam et al., 2004). There are two types of multi-channel approaches: multi-channel approaches with several radio interfaces and multi-channel approaches with a single radio interface.

2.1.1 Multi-channel approaches with several radio interfaces

HMCP (Hybrid Multi-Channel Protocol) (Kyasanur & Vaidya, 2005) divides radio interfaces into two groups: static interfaces operating on static channels, and dynamic interfaces able to change the channel they use. Nodes periodically inform their neighbors of the channel of their static interfaces. Notice that neighboring nodes do not necessarily have their static interfaces on the same set of channels. When a sender has to transmit data to a neighbor, the sender switches one of its dynamic interface to a channel corresponding to one of the static interface of the neighbor. HMCP requires periodic exchange of control packets to maintain the neighbor tables, and is not suited for mobile nodes.

The DCA (Dynamic Channel Assignment) protocol (Wu et al., 2000) requires each node to have at least two radio interfaces. One of the interface is used for control frames, and the others for data frames. The control interface always communicates on the same channel, while the data interfaces can change channels dynamically. Neighboring nodes collaborate to reserve available channels for transmissions. The amount of messages exchanged in DCA makes it not suitable when contacts between mobile entities last for a short duration.

The DCSS (Dynamic Channel Selection with Snooping) protocol (Seo et al., 2008) also uses two radio interfaces: the control interface is always listening on a common given channel, while the data interface changes its channel depending on the reservation. Nodes perform reservations depending on their view of the channel states. Nodes overhear control packets in order to determine the set of channels locally used. They assess the state of all the channels using a round-robin algorithm while they are neither transmitting nor receiving a frame. Thus, they periodically update their view of the channel states. The main drawback of DCSS is that nodes spend a lot of energy in assessing the channels.

In all these protocols, nodes are required to have two radio interfaces, which consumes a lot of energy. Indeed, radio interfaces are the main source of energy consumption in low rate wireless solutions. Moreover, the control radio interface is often active all the time. Additionally, having several radio interfaces increases the cost of the hardware. Thus, having a multi-channel approach operating with several radio interfaces is not suited in our applications.

2.1.2 Multi-channel approaches with a single radio interface

The McMAC (Multi-channel Medium Access Control) (So & Walrand, 2007) protocol uses a channel reservation mechanism based on the periodic diffusion of beacons. The beacons allow nodes to be synchronized. The interval between successive beacons is divided into two periods: a control period, and a data period. During the control period, all the nodes communicate on the same channel in order to plan the data period. During the data period, node switch to and communicate on the channels reserved during the control period. The main drawback of McMAC in our applications is that it is difficult to maintain a synchronization between mobile entities. Also, nodes need to be identified before the control period in order to be able to reserve channels: if the delay to identify new mobile nodes is large, these new nodes might be unable to communicate during a long duration.

The TMMAC (TDMA-based Multi-channel Medium Access Control) (Zhang et al., 2007) protocol is similar to the McMAC protocol. It uses a channel allocation bitmap and a channel usage bitmap in order to reserve channels. The channel allocation bitmap is filled during the control period, and the channel usage bitmap is maintained by each node and transmitted to its one-hop neighborhood. The TMMAC protocol has the same drawbacks as the McMAC protocol.

The SMC MAC (Sensor Multi-Channel Medium Access Control) (Ramakrishnan & Vanaja Ranjan, 2009) protocol based on RTS (Request To Send) and CTS (Clear To Send) control frames. RTS and CTS frames are sent on the control channel. The RTS contains the list of available channels from the sender point of view. The CTS contains the chosen channel from the destination point of view, which is the first channel of the intersection of the available channels of the sender and the receiver. When transmitting and receiving a CTS, both the sender and the receiver switch to the chosen channel, and the other nodes mark this channel as busy. The main drawback of SMC MAC is that all the nodes in range share the same communication channels: the performance of intra-robot communications can be reduced by other intra-robot communications or by inter-robot communications.

2.2 Interconnection using a meeting channel

Our goal is to allow several mobile robots to communicate without requiring an expensive hardware. Thus, we plan to use a multi-channel approach with a single radio interface. Moreover, the performance of the intra-robot communications of a robot should not be reduced by the proximity of other robots in the neighborhood. Finally, the mechanism should be compatible with existing low-power wireless personal area networks standards.

To achieve these goals, we propose to consider two types of channels: the robot-specific channels and the meeting channel. Each robot is allocated its own robot-specific channel, in order to be able to perform intra-robot communications without suffering from severe and systematic interferences from other robots that are moving in the same area. In the case where there are more robots than available channels, it is possible that several robots share the same channel. However, it is important to allocate the same channel to robots having disjoint trajectories, in order to maintain a low level of interferences. Additionally, all the robots share the same meeting channel. This channel is used solely for communications between robots.

2.2.1 Energy-efficiency

Mobile robots are often equipped with a battery that powers the engine and the coordinator. Sensors and actuators are also powered by a battery (either by the main battery or by their own). In order to maximize the lifetime of the robot, it is crucial that all embedded devices save energy, including network devices.

As the radio module of network entities consumes a significant amount of energy, most standards for low-power wireless personal area networks, such as IEEE 802.15.4 (IEEE 802.15, 2006), deactivates the radio module of nodes periodically. During this time, the nodes are sleeping and can neither receive nor transmit. They are activated at specific instants, usually during small fractions (typically less than 1%) of time.

2.2.2 Compatibility with standards

Our proposition is to allow some of the nodes to use the meeting channel for inter-mobile communications, while other nodes are sleeping. Note that most standards for low-power wireless personal area network define an inactivity time of the standards (such as IEEE 802.15.4 standard for example). The nodes that remain awaken are called energy-rich (ER) nodes, and the nodes that go to sleep are called energy-limited (EL) nodes. In this way:

- inter-mobile communications are transparent to EL nodes,
- inter-mobile communications do not degrade intra-mobile communications,
- the activities on the robot-specific channel are compliant with the standard,
- ER nodes use the inactivity time, which is otherwise wasted.

Our mechanism does not require mobile robots to be synchronized so that their inactivity period occur at the same time. Such a synchronization among mobile entities would be very difficult to achieve and induce a significant amount of control traffic, especially as the number of robots in the fleet increases.

Our mechanism uses the fact that inactivity is typically larger than the activity time. Table 1 shows the expected time ratio during which all the n mobiles are simultaneously inactive, as the activity ratio varies. It can be seen that when the activity ratio is low, the simultaneous inactivity ratio becomes very high. For example, for an activity ratio of 1%, 10 mobile robots are simultaneously inactive (and thus potentially on the meeting channel) during more than 90% of the time.

Number of mobile robots n	Activity ratio	Expected simultaneous inactivity ratio
2	0.5	0.25
2	0.25	0.5625
2	0.01	0.9801
3	0.5	0.125
3	0.25	0.4219
3	0.01	0.9703
10	0.5	0.0010
10	0.25	0.0563
10	0.01	0.9044

Table 1. The ratio of time during which n mobile robots are simultaneously inactive is very high when the activity ratio of each mobile is low.

Allowing mobile robots to communicate during their inactivity period has several advantages. As inactivity periods are usually long, the probability that two different robots are simultaneously inactive is high. By activating a subset of network devices during the inactivity period of each robot, it is possible to have these ER devices communicate with each other while the EL nodes sleep.

The remaining of this chapter describes how mobile robots can communicate using the meeting channel, either directly (when the destination robot becomes in range with the source robot) or indirectly (when the destination robot does not become in range with the source robot, or when the destination robot address is unknown by the source robot). Notice that intra-mobile communications can be performed according to the ZigBee and IEEE 802.15.4 standards for instance, and is not the focus of this chapter.

3. Direct communications between mobile robots

To allow direct communications between mobile robots, a DTN paradigm has to be used (*Delay-Tolerant Networking Architecture*, 2006). Indeed, communications between mobile robots are opportunistic in nature: they depend on the mobility pattern of each mobile robot and on the wireless propagation conditions, both of which are considered here as unpredictable. Moreover, the number of mobile robots in the area is small compared to the area size. Thus, mobile robots meet each other infrequently.

We propose to adapt the *store, carry and forward* principle from the DTN literature to our generic application. In DTNs, a new layer called the bundle layer is introduced between the application layer and the transport layer of the OSI model. The main role of this bundle layer is to mask to applications the effects of node mobility, and mainly the sporadicity of the network connectivity. To do so, data from the application layer is encapsulated in a composite structure, called a bundle. A bundle contains the application data, and additional data to deal with sporadic connectivity, such as timeouts and destinations. Using a bundle layer implies the need of a transport layer, as the bundle structure is often larger than the maximum frame size in a low-power wireless personal area network.

According to the DTN principle, messages are stored in bundle in specific nodes called custodians. When two mobiles meet, their custodians can exchange bundles. The bundles can circulate through the mobility of robots or through contacts between mobiles. The detection of a contact between mobiles is performed by specific nodes called gateways.

To summarize, we can consider the following types of nodes.

- EL nodes are only active on the robot-specific channel. They sleep during the robot inactivity period, and do not contribute to communications between mobile robots.
- There is one custodian node per robot. It is an ER node with larger storage capabilities than the other nodes. During the robot activity period, it works as the other nodes. During the robot inactivity period, it switches to the meeting channel, and manages bundles.
- There is at least one gateway node per robot. Gateways are ER nodes whose role is to detect neighboring robots on the meeting channel, during the robot inactivity period.
- There can be intermediate ER nodes per robot. Intermediate nodes are ER nodes that help interconnecting the custodian and the gateways on the meeting channel, during the robot inactivity period.

Fig. 2 describes the network stack of our architecture.

In this section, we first describe how the presence of ER nodes can improve the network performance of intra-robot communications. This feature is important as the mobile robots often have no neighbors in range. Second, we study the neighbor discovery mechanism and

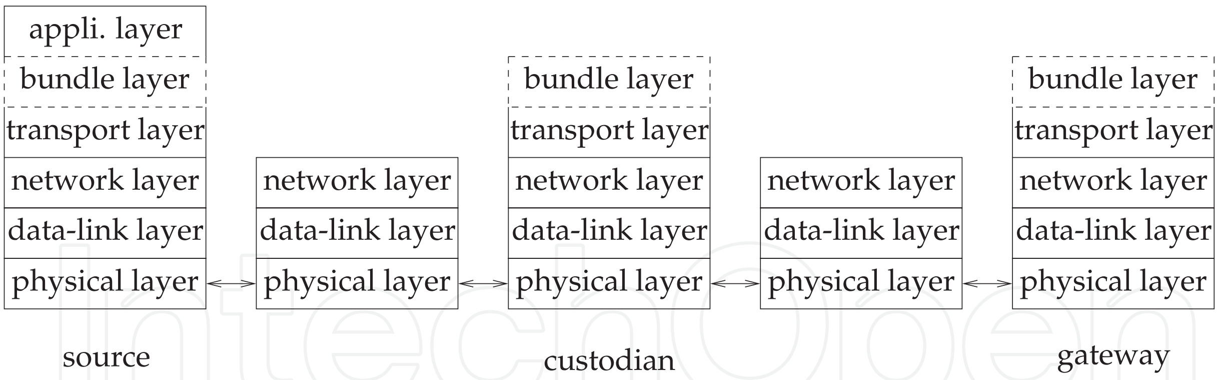


Fig. 2. Network stack and communication from a source to a gateway (on the same mobile), including the bundle layer.

we present the trade-off between the energy-efficiency of the neighbor discovery mechanism and the duration of the neighbor detection. Third, we show how communication links can be established between robots, and how the custodians of two neighbor robots can know each other. Fourth, we discuss direct communications between mobiles.

3.1 Optimized intra-robot communications

During the robot inactivity period, all ER nodes are active on the meeting channel. The gateways are trying to detect potential neighboring robots. However, most of the time, no robots are found in the vicinity. Instead of being idle while waiting for encounters, ER nodes can help relaying the intra-robot traffic.

3.1.1 Description

During the robot inactivity period, the ER nodes form a network called the ER network. Thus, ER nodes have two addresses: one during the robot activity period, and one during the robot inactivity period. EL nodes have a single address, for the robot activity period. We assume that there exists a routing function $\mathcal{R}$, such that $\mathcal{R}$ takes as input the destination addresses (in the robot network and/or in the ER network) and returns the next hop addresses (in the robot network and/or in the ER network). In the ZigBee standard, the function $\mathcal{R}$ is based on the hierarchical allocation of addresses. We also assume that there exists a neighbor address function $\mathcal{N}$ that takes as input the address of a next hop (in the robot network and/or in the ER network) and returns, if possible, the address of the next hop in both networks. It is not mandatory that functions $\mathcal{R}$ and $\mathcal{N}$ are always able to translate the address of a node from one network to another, but they can improve the global behavior of the mechanism.

To optimize intra-robot communications on the meeting channel, the MAC relaying scheme has to be modified in the following way. Let us assume that an ER node n has a frame for destination d. If the address of d in the ER network is known, this address can be used to compute the next hop h on the ER network, as $h = \mathcal{R}(d)$. n can send the frame to h. If the address of d in the ER network is not known, it is possible that the next hop h' of d computed on the robot network is also a neighbor of n in the ER network. If this is the case, $h = \mathcal{N}(h')$ is known: n can send the frame to h (in fact, both addresses h and h' are present in the frame). Otherwise, n is unable to send the frame during this period. Thus, n proceeds to the next frame in queue (without dropping the previous frame, as it will be sent during the next robot activity period).

To summarize, ER nodes can always relay frames to ER nodes.

3.1.2 Simulation results

To quantify the performance improvement when the meeting channel is used for intra-robot communications, we performed simulations using the network simulator NS2. We used the IEEE 802.15.4 physical layer and a custom IEEE 802.15.4 MAC layer that supports our meeting channel architecture and our relaying scheme. We used the ZigBee hierarchical routing protocol for both the robot network and the ER network. The propagation model used is the ITU indoor propagation model (*Propagation data and prediction method for the planning of indoor radio communication systems and local area networks in the frequency range of 900 MHz to 100 GHz*, 2005), with a path loss exponent of 3. The transmission power of nodes is set to -25 dBm. Frames have a size of 119 bytes at the MAC layer, which corresponds to 1000 bits at the physical layer. Queue lengths are set to 50 frames.

Simulations are performed on a network of 6 nodes with four pairs of source *s* and destination *d*, as depicted on Fig. 3. Solid nodes represent EL nodes, half-striped nodes represent ER nodes. The source is always three hops away from the destination.

In order to evaluate the advantages of communications on the meeting channel, we compare the scenario where all the nodes are ER nodes with a scenario where the two central nodes are ER nodes, and the source and destination both have a 50% probability of being ER nodes. Simulation results are averaged over 100 runs.

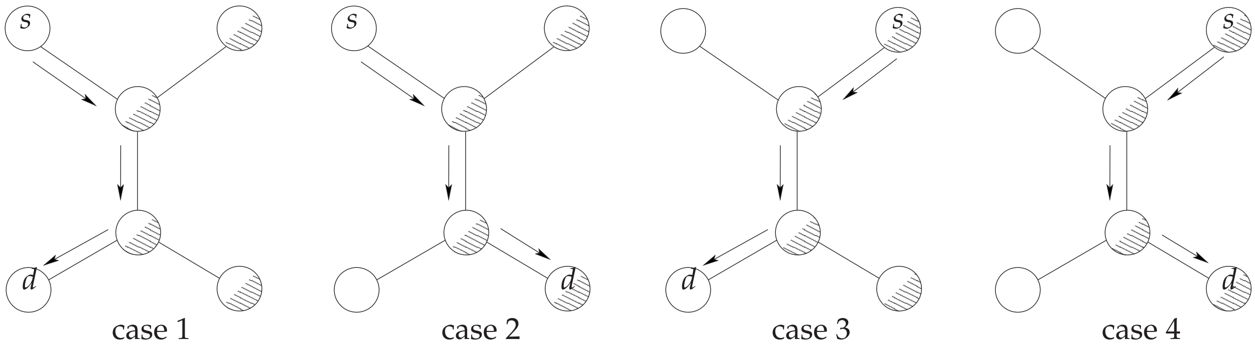


Fig. 3. Simulations are performed on a small topology, where sources and destinations both have a 50% probability to be ER nodes.

We considered two metrics: (1) the goodput, which is the number of frames correctly received by the destination node per second, and (2) the end-to-end delay.

Fig. 4 displays the goodput as a function of the frame generation rate, for several scenarios: one where the nodes are always active on the robot-specific channel, in dotted lines, one with only EL nodes and for various duty cycles on the robot-specific channel (from 12.5% to 50%), in solid lines, and one with both EL and ER nodes (as shown on Fig. 3) for various duty cycles on the robot specific channel (from 12.5% to 50%), in dashed lines. First, it can be noticed that the goodput increases with the frame generation rate, until it reaches a maximum. This behavior is typical of IEEE 802.15.4. As expected, the goodput with EL nodes only is strongly related to the duty cycle: it is high when the duty cycle is 100% (EL nodes are always active), and is reduced when the duty cycle goes from 50% down to 12.5%. When ER nodes are added, the goodput is significantly increased with respect to the scenarios with only EL nodes.

Fig. 5 displays the average end-to-end delay as a function of the frame generation rate, for the three same scenarios, and for various duty cycles. The average end-to-end delay increases with the frame generation rate, until it reaches a maximum. This maximum can be explained in the following way. The delay is computed based on the received frame only. Frames cannot

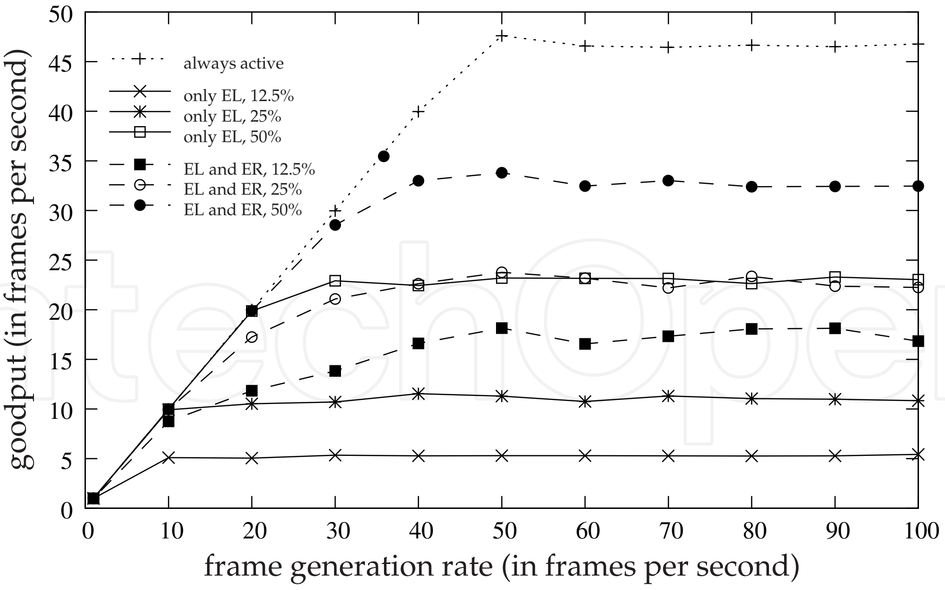


Fig. 4. The presence of ER nodes significantly improves the performance of the network, for all duty cycles.

wait for a long time in the frame queue, because of the limited size of the frame queue and because slotted CSMA/CA drops frames after a small number of attempts. When nodes are always active, the delay is very low. The delay increases when the network is saturated, which occurs at about 50 frames per second (which can be seen on Fig. 4). When there are only EL nodes, the end-to-end delay is large: indeed, EL nodes spend a large amount of time sleeping, which increases the delay. In the scenario with ER nodes, the end-to-end delay is improved compared to the scenario without ER nodes: indeed, some frames can be relayed during the inactivity period of EL nodes, which helps reducing the delay.

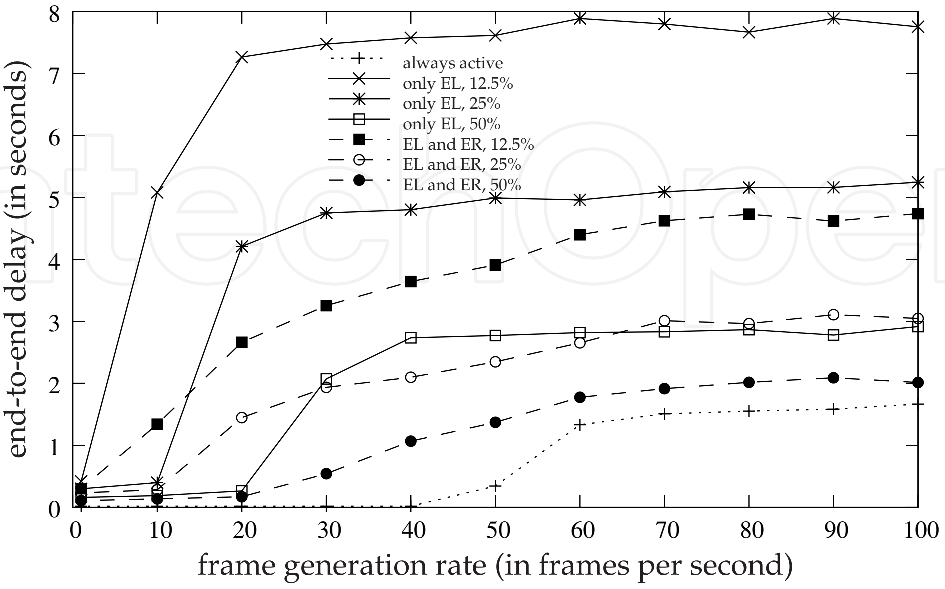


Fig. 5. The presence of ER nodes significantly reduce the end-to-end delay, for all duty cycles.

ER nodes consume more energy than EL nodes, because they are more often active. Thus, ER nodes allow to achieve a trade-off between energy consumption and performance.

3.2 Neighbor discovery

When a robot moves in range of another, the two robots need to discover each other before being able to communicate. The neighbor discovery mechanism Hadid et al. (2010) occurs during the inactivity period, and is performed by gateway nodes only.

Gateway nodes send periodic gateway advertisement (GW-ADV) frames. Each GW-ADV frame contains the identification of the gateway, and the identification of the robot that carries it. When receiving a GW-ADV frame, a gateway replies with a gateway reply (GW-REP) frame. Each GW-REP frame contains the identification of the local gateway, the identification of the distant gateway, and the identification of the robot that carries it. Upon receiving a GW-ADV or GW-REP frame, both gateways update their neighbor table. An entry in this table contains the identification of the neighbor mobile, the address of the distant gateway, and a time to live. It is possible for the table to contain two different entries for the same mobile, if the mobile is reachable through two gateways.

The neighbor discovery is not instantaneous, as gateways transmit GW-ADV periodically, and only when they are operating on the meeting channel. The neighbor discovery delay depends on the periodicity of the GW-ADV frames, and on the robot activity period. To study the impact of these two parameters, we carried out two sets of simulations. The first set of simulation concerns the neighbor discovery delay from the moment the two mobiles become in range. The second set of simulation concerns the total contact duration, between the moment the gateways identify each other, until the time the contact is broken due to robots mobility.

We used a simple mobility model[1]: two robots move on parallel lines, in opposite directions, with a constant speed of 2 meters per second. They start far away, at a position where they are not in range of each other. They eventually meet and discover each other, while continuing their movement. After some time, they are too far away to keep communicating, and the contact is broken. The minimum distance between robots is 3 meters, which is loss than the range of a node. The placement of gateways is shown on Fig. 6, where gateways are represented with nodes connected to a black rectangle.

We used a duty cycle of about 1 second[2]. Each simulation lasts for 100 second, and the results are averaged over 500 runs.

3.2.1 Neighbor discovery delay

The first simulation concerns the quantification of the neighbor discovery delay. It is computed as the difference between the time when the two mobiles become in range and the time when they are not in range anymore. To model the fact that mobiles are not synchronized, the activity (on the robot-specific channel) of mobile M_1 starts at t_0, and the activity (on the robot-specific channel) of mobile M_2 starts at $t_0 + \Delta$, where Δ is chosen randomly in the cycle. Moreover, we consider that the periodicity of GW-ADV might vary between mobiles, due to

[1] Our focus here is not to focus on the mobility model, but on the quality of the neighbor discovery procedure. That is why we prefer to use a simple mobility model rather than a more complex one. A discussion on the impact of the mobility model can be found in Bai et al. (2003).

[2] The duty cycle is in fact equal to 983.04 ms, which is the closest value for a duty cycle of 1 second for IEEE 802.15.4 solutions.

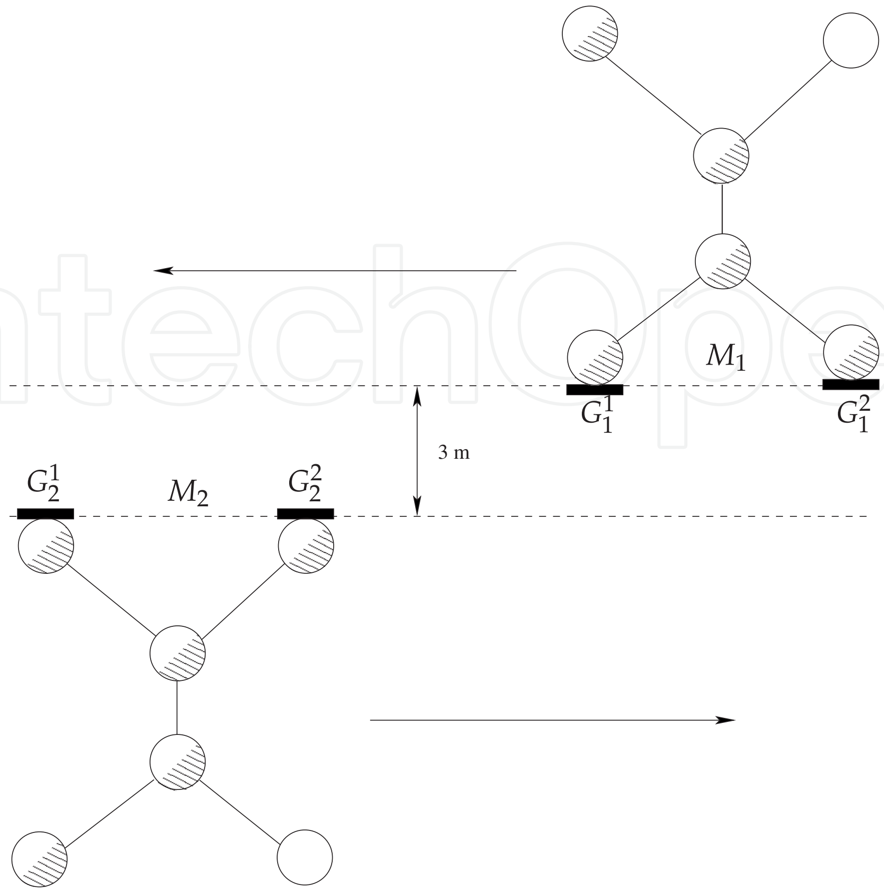


Fig. 6. Two mobiles M_1 and M_2 cross each other. The minimum distance between the gateways of M_1, G_1^1 and G_1^2, and the gateways of M_2, G_2^1 and G_2^2, is 3 meters.

clock drifts. The periodicity of GW-ADV for a mobile M_i is given by $T(1 + \delta_i)$, where T is the basic periodicity, and δ_i is chosen randomly in $[-0.3; 0.3]$ (according to a uniform distribution). Fig. 7 shows the neighbor discovery delay as a function of the GW-ADV periodicity T, and as a function of the duration of the activity period on the robot-specific channel. The largest the periodicity, the longest the duration. However, the duration of the activity period has the most significant impact on the neighbor discovery delay: when the nodes spend 50% of their time on their robot-specific channel, the gateways of a mobile often miss the GW-ADV messages sent on the meeting channel by the gateways of the other mobile.

3.2.2 Contact duration

The second simulation concerns the quantification of the contact duration. It is computed as the difference between the time when the mobiles discover each other and the time when there is no gateway of one mobile in contact of one gateway of the other mobile. Notice that having several gateways per mobile might be a way to (slightly) improve the contact duration.
Fig. 8 shows the contact duration as a function of the GW-ADV periodicity T, and as a function of the duration of the activity period on the robot-specific channel. The largest the periodicity, the smallest the duration, as it takes some time for the nodes to discover each other. Again, the duration of the activity period has the most significant impact on the contact duration.

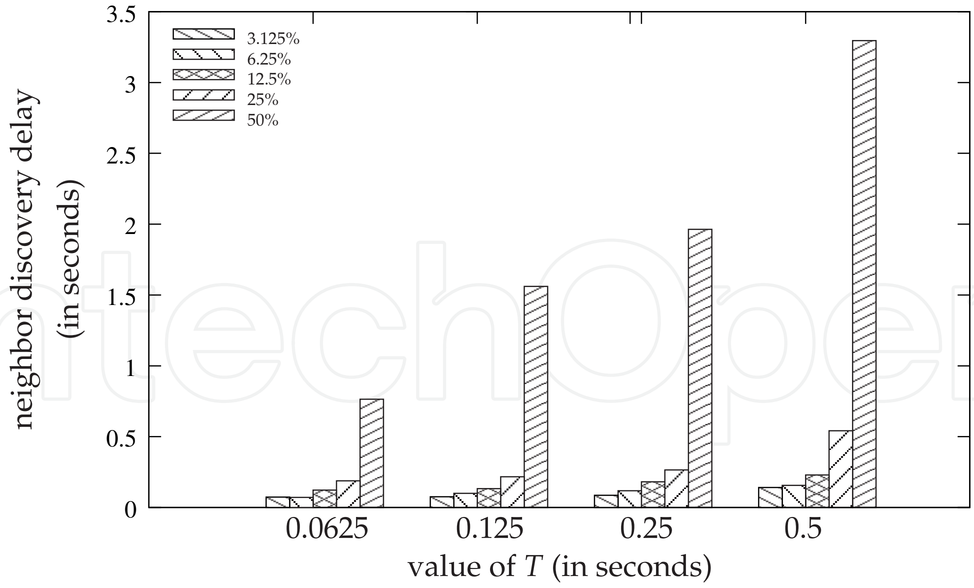


Fig. 7. The neighbor discovery delay is greatly impacted by the duration of the inactivity period.

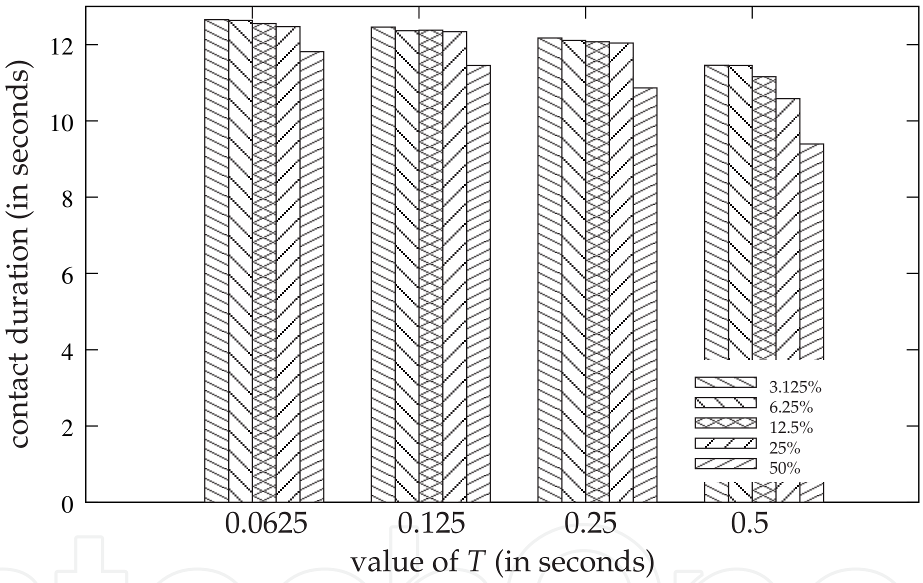


Fig. 8. The neighbor discovery delay is greatly impacted by the duration of the inactivity period.

3.3 Link establishment

Once a local gateway discovers a distant gateway, the local gateway notifies its custodian of its discovery using a notification (NOTIF) message. This is done either when the local gateway received a GW-REP, or received the GW-ADV from another gateway. The NOTIF message contains the identification of the distant mobile, the address of the distant gateway and the address of the local gateway. Upon receiving the NOTIF message, the custodian updates its contact table. An entry of the contact table contains the identification of the distant mobile, the address of the distant gateway and the address of the local gateway. There might be several local and distant gateways that lead to the same mobile.

If a gateway stops receiving GW-ADV messages during a period equal to the time to live of the neighbor entry, the gateway sends a message to its custodian, in order to remove the entry from the contact table. The mobile might still be reachable, through other local or distant gateways.

3.4 Direct data exchange between mobiles

The data exchange between a source node on a source mobile and a destination node on a destination mobile is performed according to the following steps:

- the source node sends the bundles to the custodian node of the source mobile,
- the custodian node of the source mobile sends the bundles to a gateway node of the source mobile, once a link between these two mobiles has been established,
- the gateway node of the source mobile sends the bundles to the gateway node of the destination mobile,
- the gateway node of the destination mobile sends the bundles to the destination node.

The transmission from the source node to the custodian node of the source mobile corresponds to an intra-robot communication, as both nodes are on the same mobile. This part of the communication can be achieved even if the two mobiles are not in range. The same goes for the transmission between the gateway of the destination mobile to the destination node.

Fig. 9 depicts this data exchange on a small example. Custodians are shown with a small database on the side, to indicate that they possess storage capabilities. As shown below the figure, several communication layers are involved in this process. The source node S generates data at the application layer, which is encapsulated within bundles at the bundle layer. The communication from S to C_1 is done at the bundle layer, and can require the transmission through intermediate nodes, such as X_1. Then, the bundles wait in C_1 for a notification concerning mobile M_2. Once this happens, C_1 communicates with its gateway G_1 at the bundle layer. G_1 and G_2 are able to communicate directly. Finally, G_2 sends the bundles to D, at the bundle layer, potentially involving intermediate nodes, such as X_2. Finally, D can extract the data at the application layer.

One bundle can correspond to several packets. The transport layer is in charge of splitting large bundles into several packets.

3.4.1 Description

The direct data exchange between mobiles raises four issues: (1) the choice of the source gateway by the source custodian, (2) the choice of the distant gateway by the source gateway, (3) the algorithm used by the custodian to send the bundles, and (4) the policy to retransmit or drop bundles.

When there are several gateways that can reach the destination mobile, the source custodian has to choose to which source gateway the bundles have to be sent. The custodian can apply different policies of load balancing, depending on the application, by choosing several different source gateways to allow parallel transmissions of the bundles. Thus, several bundles at a time are sent, one per gateway that discovered the destination mobile.

A given source gateway can be in range of several destination gateways. The source gateway has to choose to which destination gateway the bundles should be sent to. In this paper, we propose that source gateways choose destination gateways at random, to reduce

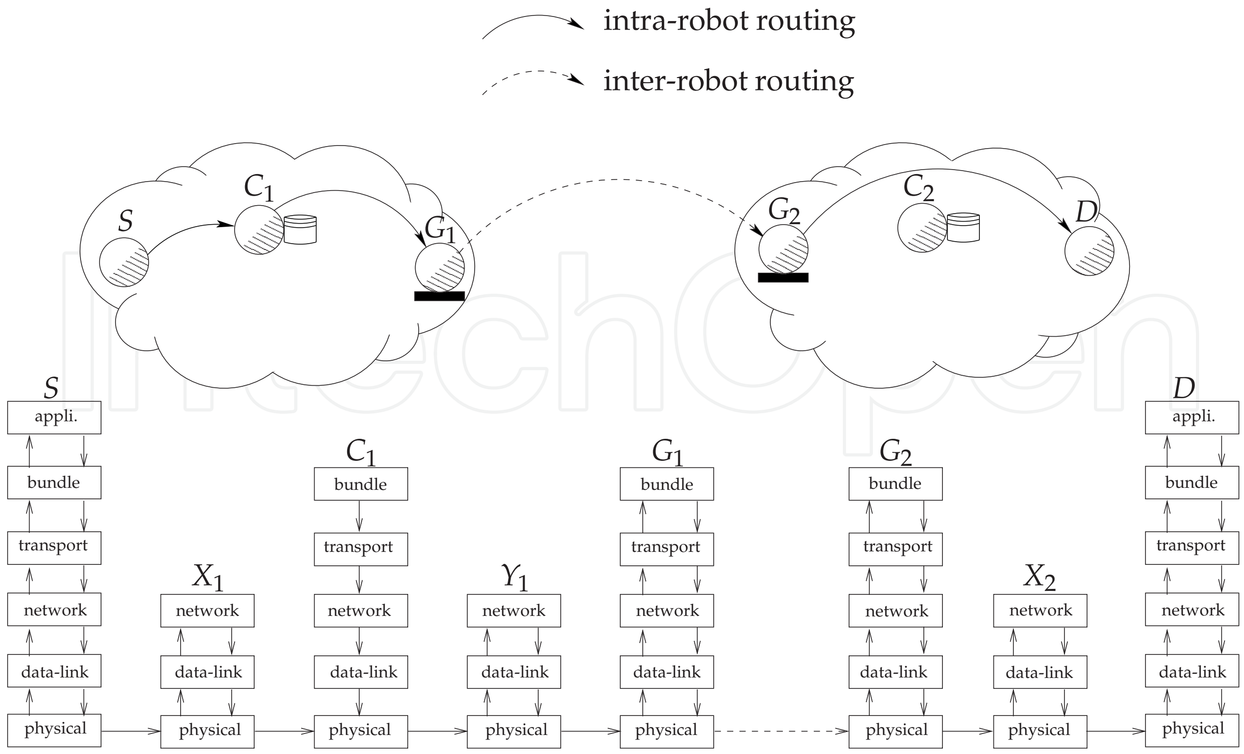


Fig. 9. The bundles sent by mobile M_1 to mobile M_2 first go from the source node S to the custodian C_1. Once M_1 and M_2 have established a link through their gateways, the bundles go from C_1 to G_1, from G_1 to G_2, and finally from G_2 to D.

the probability that two different source gateways choose the same destination gateway repeatedly.

The algorithm applied by the custodian is the following. When it receives a bundle, it checks whether there is an entry in its contact table corresponding to the destination mobile or not. If there is no entry, the custodian stores the bundle. Otherwise, the custodian computes the set of source gateways that can reach the destination mobile.

Bundles are not kept permanently in custodians. Indeed, they are associated to a time to live, which is decided at the application layer. Bundles are also removed by the source custodian when they are successfully transmitted. When a local gateway sends a bundle to a distant gateway, it waits for the acknowledgment of the transmission. Once the local gateway receives the acknowledgment, it sends a ROUTE-SUCCESS message to the custodian. Upon receiving this message, the custodian removes the bundle from its memory. If the local gateway does not receive acknowledgments from the distant gateway after a number of retries, the local gateway sends a ROUTE-FAILURE message to the custodian. The custodian can decide to transmit the bundle to another source gateway. The custodian does not delete bundles after ROUTE-FAILURE messages: it simply waits for another opportunity to transmit it, or for the time to live to expire. Notice that the ROUTE-SUCCESS and ROUTE-FAILURE messages only contain a bundle ID: these messages can fit in short frames.

3.4.2 Simulation results

We evaluated the performance of this data-exchange mechanism according to three metrics: (1) the bundle reception rate, (2) the average delay from the custodian of the source to the destination, and (3) the average path length for bundles.

Simulations are performed on a scenario close to the one depicted on Fig. 9. We assume that each mobile has two gateways, that the two gateways of the source mobile are one hop away from the custodian, and the destination is one hop away from the two gateways of the destination mobile. We assume that each bundle can fit in a single frame, and thus bundles do not require fragmentation and reassembly. The frame size for a bundle is set to 119 bytes at the MAC layer, so that a bundle is 1000 bits long at the physical layer. The duty cycle is set to about one second, and gateways send their GW-ADV frame with a periodicity of $T = 0.25$ seconds (when they operate on the meeting channel). This value of T is a trade-off between the discovery delay and the overhead of the GW-ADV frames.

Fig. 10 shows the bundle reception rate as a function of the bundle generation rate. The bundle reception rate increases with the bundle generation rate, until it reaches a maximum (again, this behavior is typical of IEEE 802.15.4). When the duty cycle on the robot-specific channel is low (for instance, when the duty cycle is equal to 12.5%), the ER nodes spend most of their time on the meeting channel. Thus, the bundle reception rate is large. Contrarily, when nodes spend a significant part of their activity time on the robot-specific channel (for instance, when the duty cycle is 50%), the ER nodes cannot spend most time on the meeting channel, and the gateways of the different robots have less time to communicate (on average, the gateways of the two robots are on the same meeting channel during only 25% of the time), which negatively impacts the bundle reception rate.

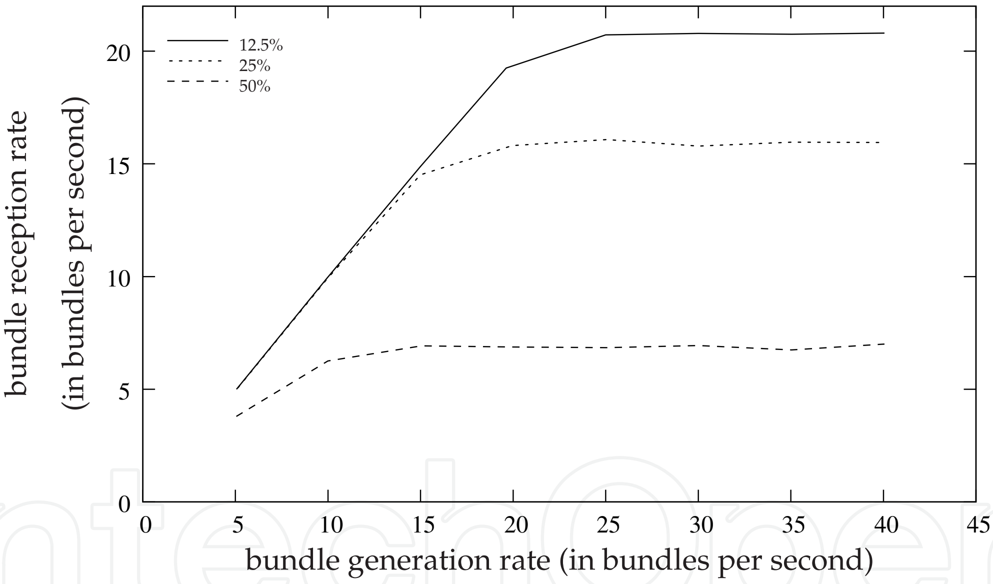


Fig. 10. The bundle reception rate (in bundles per second) increases as the duty cycle on the robot-specific channel decreases.

Fig. 11 shows the average delay to transmit a bundle from the custodian of the source mobile to the destination. Note that this delay does not take into account the time required to transmit a bundle from the source to the custodian. Indeed, most of the time, all the bundles are accumulated in the custodian rather than in the source node. As expected, the average delay increases with the bundle generation rate. When the bundle loss rate is too high, most bundles are lost, and the delay becomes stable (as it involves only bundles that are correctly received). The average path length for the bundles is computed as the average amount of transmission of a bundle. The minimum value for this metric is three in our topology, as the destination is

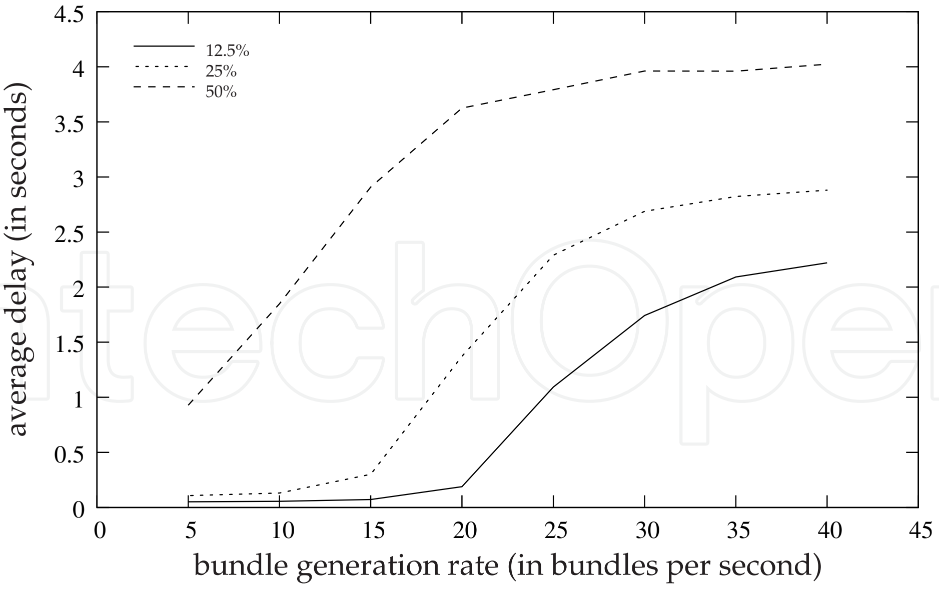


Fig. 11. The average delay from the source custodian to the destination node decreases when the duty cycle on the robot-specific channel decreases.

three hops away from the custodian of the source mobile. Path lengths larger than three occur in the following two cases.

- When one of the nodes on the path from the custodian to the destination has to repeat the transmission (usually due to the loss of a frame or of its acknowledgment), the path length increases accordingly.
- When the custodian C_1 sends the bundle to a local gateway G_1^1, it is possible that this local gateway is not connected anymore to the distant gateway. Indeed, this local gateway can be waiting for the time to live of the neighbor to expire in order to remove the entry in its neighbor table and to notify the custodian. After having tried several times to transmit the bundle, G_1^1 notifies the source custodian with a ROUTE-FAILURE message. The custodian can try to send the bundle to another local gateway G_1^2 which is still in contact with the destination mobile. In this case, the path length is composed of four hops: (C, G_1^1), (C, G_1^2), (G_2^2) and D. Note that while bundles are long frames, the ROUTE-FAILURE message is a simple notification. Thus, it is not counted in the path length.

Note that to simplify, we assume in our simulations that each bundle is sent in a single frame. Thus, the path length increases only by one each time a data frame is transmitted.

Fig. 12 shows the average path length for the bundles. When the duty cycle on the robot-specific channel is low (for instance, when it is only 12.5%), the average path length varies slightly around 3.2 hops. As the minimum path length is 3, this means that, on average, the probability for a bundle to be retransmitted (due to collision, loss of a frame or of its acknowledgment, or due to the mobility of mobiles) is only 6%. However, this percentage increases up to 26% when the duty cycle on the robot-specific channel is high.

4. Indirect communications between mobile robots

Direct communications occur when the following three conditions are met.

- A source robot has to communicate with a destination robot.

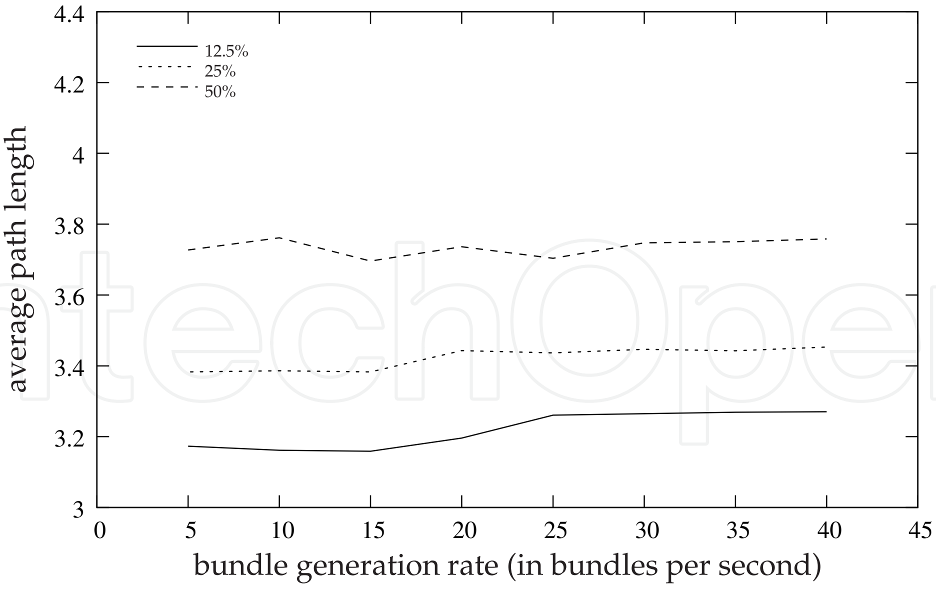


Fig. 12. The average path length for the bundles decreases when the duty cycle on the robot-specific channel decreases.

- The source robot knows the ID of the destination robot.
- The source robot and the destination robot meet directly.

In practice, the two last conditions are hard to fulfill. For example, when a robot has to perform an emergency brake, it has to inform the following robots. The ID of those robots might not be known by the source robot. Moreover, it is possible that a robot and the one following it on the same trajectory never meet: in this case, they are unable to communicate using direct communications.

Indirect communications help removing these constraints. In this section, we first consider how fixed relays can improve the overall connectivity of the network, and can enable indirect communications without requiring sophisticated routing protocols. Then, we consider how to identify mobile robots and nodes based on their role rather than based on their ID or address.

4.1 Communications through fixed-relay networks

Fixed-relay networks can be used to improve the communication opportunities between mobiles (Groenevelt et al., 2005; Zhao et al., 2006). When these relay networks are placed along the road, or at crossings, they can be used to store bundles from robots passing by. In turn, the bundles can be sent to other mobiles traveling in range of the relays.

4.1.1 Description

Fixed-relay can be used to improve the connectivity of the network. If they are deployed strategically at the intersection of mobile trajectories, they can store bundles temporarily, and retransmit them to mobiles passing by.

Each fixed-relay is composed of several nodes forming a network. There is a custodian node in each fixed-relay, and several gateways. The role of the custodian node and of the gateways is the same as for mobile robots.

The main difference between fixed-relays and mobile is that the nodes in fixed-relays are always working on the meeting channel. Thus, they are always available on this channel, and are thus easier to contact than mobiles.

4.1.2 Simulation results

To show the benefits of fixed-relays, and to evaluate the advantage of remaining on the meeting channel on the performance of the network, we performed numerous simulations. The simulation settings are mainly the same as in Section 4. Fixed-relays have a six-node topology, which is identical to the topology of the mobiles. They are static.

Fig. 13 shows the average delay for a mobile to discover a relay node, as a function of the periodicity *T* of the GW-ADV of both the mobile and the fixed-relay. The relay discovery delay increases as the periodicity *T* decreases and as the duty cycle on the robot-specific channel increases. The duty cycle on the robot-specific channel is the key parameter: its impact is the most significant. When comparing with Fig. 7, the advantage of fixed-relays on the discovery delay is obvious. In the case where the duty cycle on the robot-specific channel is 50%, and for a GW-ADV periodicity *T* of 0.5 seconds, the discovery delay of a mobile is 13.2 times larger than the discovery delay of a fixed-relay.

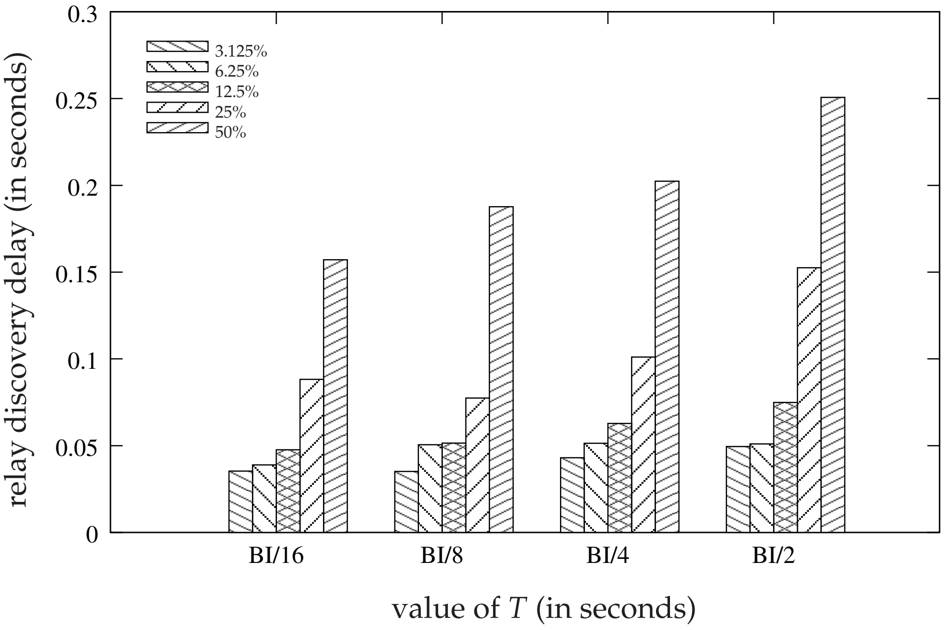


Fig. 13. The relay discovery delay for a mobile is very short, and depends mostly on the duty cycle on the robot-specific channel.

Fig. 14 shows the contact duration between a mobile and a fixed-relay. The contact duration decreases as the GW-ADV periodicity increases because it takes more time to discover the fixed-relay with a large periodicity. The contact duration also decreases as the duty cycle on the robot-specific channel increases, for the same reason. When comparing with Fig. 14, the contacts are about twice larger, as expected since only one robot is moving in this scenario with fixed-relays.

Fig. 15 displays the bundle reception rate as a function of the bundle generation rate, for several duty cycles on the robot-specific channel. The bundle reception rate increases with the bundle generation rate, until a maximum is reached. When comparing with Fig. 15, the results are significantly improved and show that using fixed-relays is highly beneficial. For instance, when the duty cycle is 50%, the maximum bundle reception rate is about 7 when two mobiles meet, and is about 14 when a mobile meets a fixed-relay (as the bundle reception rate is given in bundles per second, the increase of the contact duration in the scenario with

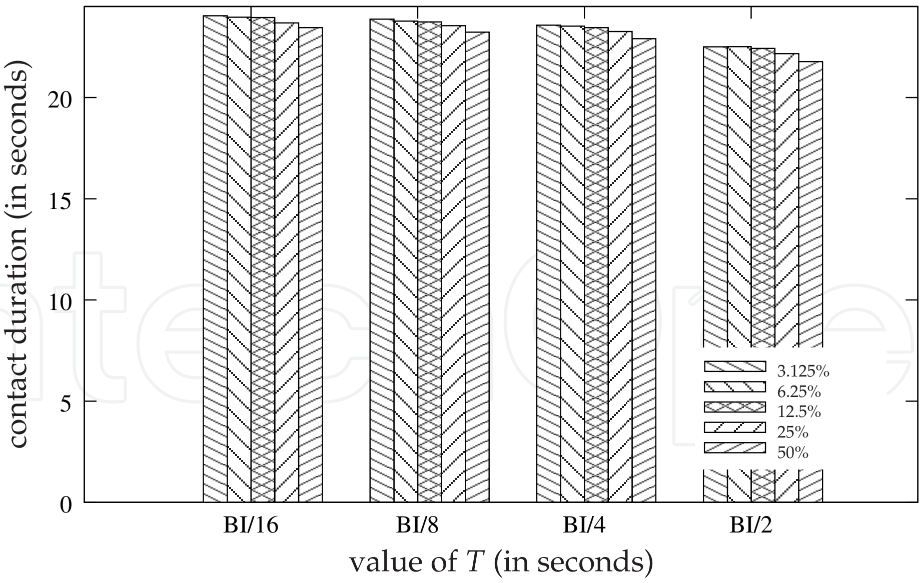


Fig. 14. The contact duration between a mobile and a fixed-relay depends mostly on the GW-ADV periodicity.

fixed-relays has little impact on this number). When the duty cycle is 12.5%, the maximum bundle reception rate is 21 when two mobiles meet, and 24 when a mobile meets a fixed-relay.

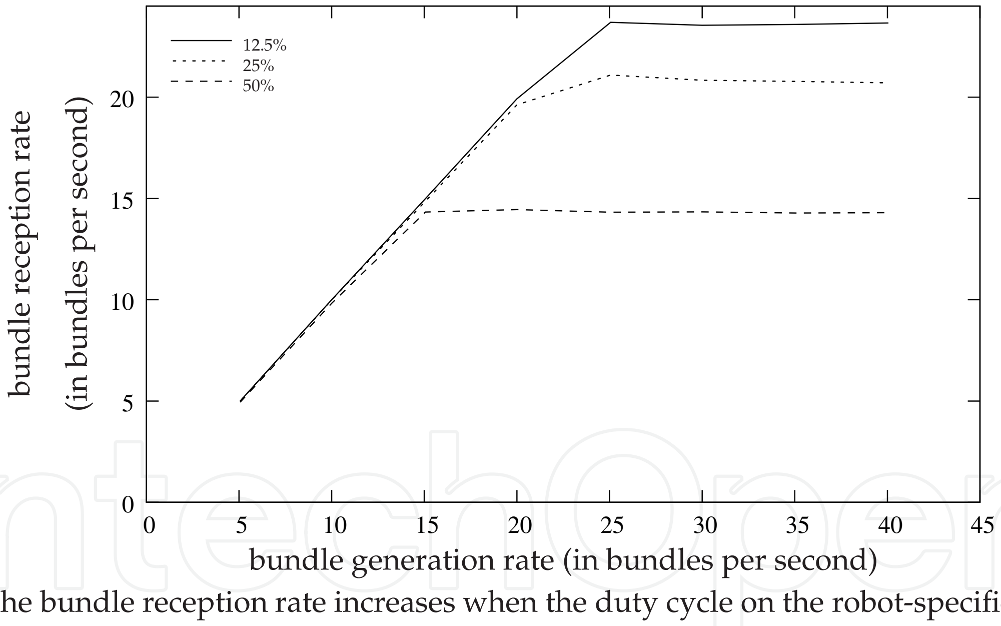


Fig. 15. The bundle reception rate increases when the duty cycle on the robot-specific channel decreases.

Fig. 16 shows the average delay as a function of the bundle generation rate, for several duty cycles on the robot-specific channel. The delay is the time difference between the first transmission of the bundle by the custodian of the source mobile, and the first reception of the bundle by the custodian of the fixed-relay. When comparing with Fig. 16, the results are similar. The gain in delay is not significant, except for the duty cycle on the robot-specific channel, where the delay is reduced by 25%.

Fig. 17 shows the average hop count as a function of the bundle generation rate, for several duty cycles on the robot specific channel. The y-axis is the same as in Fig. 17. It can be seen

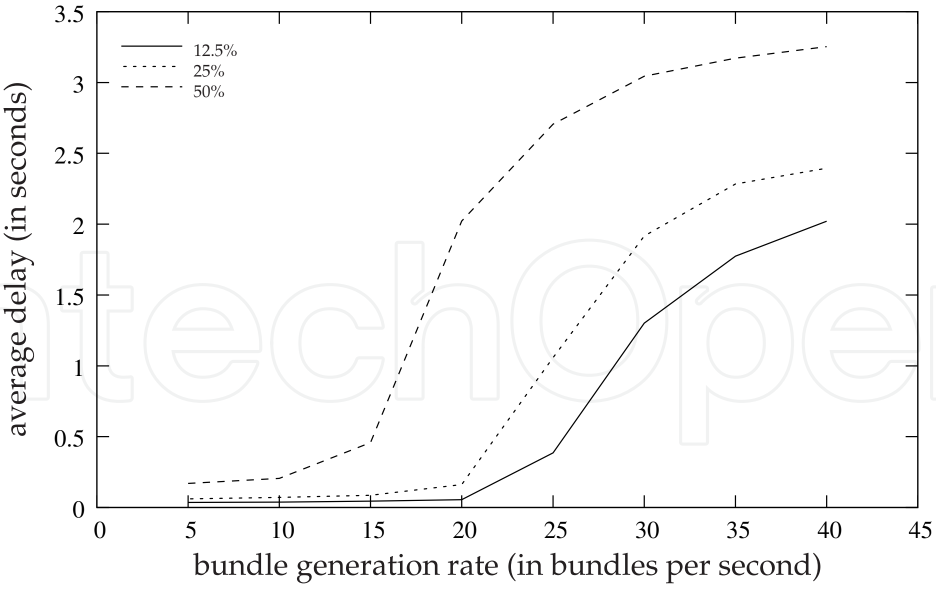


Fig. 16. The average delay from the custodian of the source to the custodian of the fixed-relay decreases when the duty cycle on the robot-specific channel decreases.

that the average hop count varies from 3 (which is the minimum value in our scenario) to 3.2, which constitutes an important improvement.

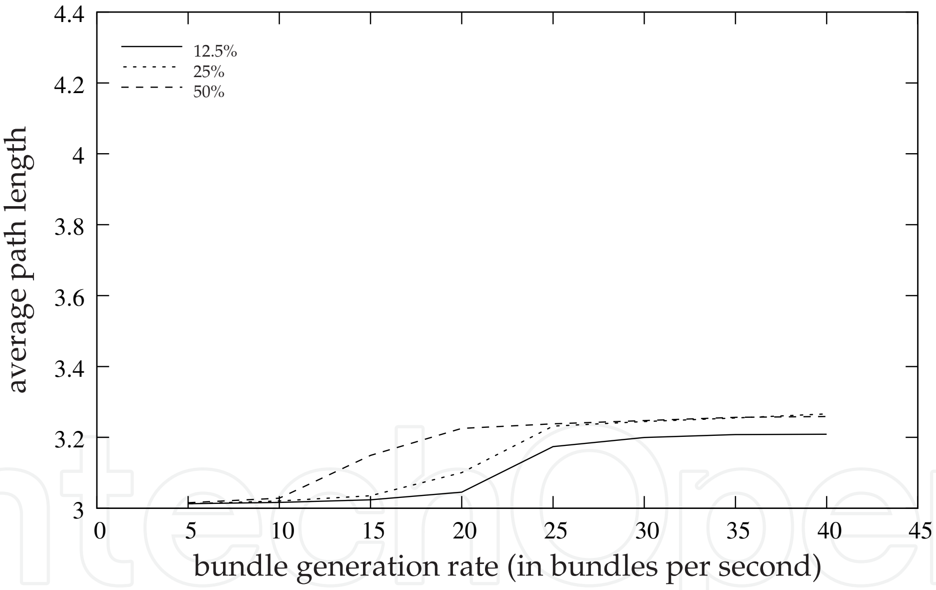


Fig. 17. The average length of the paths followed by bundles is close to the minimum value of 3 when a mobile meets a fixed-relay.

4.2 Generic addressing

The generic addressing consists in addressing network entities by their role rather than by their ID. There are two types of generic addressing: one for the mobile robots, and one for the nodes. These two types can be used simultaneously: it is possible to address a generic node on a generic mobile.

The address space has to be divided into two parts: the part for network addresses and the part for generic addresses. In the ZigBee protocol for instance, when the distributed address allocation mechanism is used, the whole address space is not used for nodes. Among the 2^{16} available short addresses, only n can be allocated, where n is the maximum number of nodes in the hierarchical tree. n depends on the ZigBee network parameters R_m, C_m and L_m. It is possible to use the unused addresses to represent roles. The same is true for mobiles: it is unrealistic to consider that there are 2^{16} mobile robots in the network. Generally, the number of roles is much lower than the number of entities (either mobiles or nodes) in the network: it is not required to allocate a large number of generic addresses.

The definition of generic addresses is application-dependent, and it depends on the communication requirements. For mobiles, we identified the following requirements.

- Concerning unicast communications (that is, from one mobile to another mobile), two types of requirements can be considered.
  - Role *unicast-next*: the mobile has to communicate with the next mobile. The next time the source mobile meets another mobile, this new mobile becomes the destination of the bundles.
  - Role *unicast-next-geographical*: the mobile has to communicate with the next mobile entering a given area. This role makes use of the fixed-relay to transmit bundles. With this role, the source mobile sends the bundles to the first mobile or fixed-relay it meets. Once the bundles have reached a fixed-relay, it sends the bundles to the first mobile it meets.
- Concerning multicast communications (that is, from one mobile to several mobiles), three types of requirements can be considered. They are a generalization of the roles for unicast communications.
  - Role *multicast-next-all*: the mobile has to communicate with n next mobiles. The next time the source mobile meets another mobile, it decreases the value of n and sends all the bundles to this new mobile. If n is equal to zero, the bundles can be removed from the memory of the source mobile. Otherwise, the bundles are stored for the future encounters. Note that the source mobile has to maintain a list of mobiles it has already met: if a source mobile meets twice the same mobile, the source does not send the same bundles twice.
  - Role *multicast-next-chain*: the mobile has to communicate with a a chain of n next mobiles. This requirement is important when several mobiles are following each other. When the source mobile meets the first mobile of the chain, the source sends all the bundles, and does not keep a copy of them (once they are acknowledged by the gateway of the first mobile of the chain). Then, the first mobile becomes in charge of transmitting the bundles to the $n - 1$ next mobiles. Additional care has to be taken in order to avoid loops: a mobile should not receive bundles it has previously received.
  - Role *multicast-next-geographical*: the mobile has to communicate with n next mobiles entering a given area. When a mobile has to transmit k times the bundles and meets a fixed-relay, the mobile transmits all its bundles to this fixed-relay. The fixed-relay becomes in charge of transmitting $k - 1$ times the bundles. When a mobile has to transmit k times the bundles and meets another mobile, the former mobile become in charge of transmitting $\lfloor (k-1)/2 \rfloor$, and the latter mobile becomes in charge of transmitting $\lfloor k/2 \rfloor$. In this way, the bundles are disseminated rapidly in the area.

For nodes, several generic roles can be identified.

- the custodian of the mobile,
- one of the gateways of the mobile,
- all the gateways of the mobile,
- one of the sensors (for each type of sensor),
- all of the sensors (for each type of sensor),
- one of the actuators (for each type of actuator),
- all of the actuators (for each type of actuator).

As it can be seen, it is possible to represent several nodes through the same generic role. The duplication of messages from one generic message into several messages is performed in the node that translates the generic addresses, that is in the custodian node.

The generic addressing mechanism requires address translations (one for the generic mobiles, and one for the generic nodes). The translation for generic nodes is always performed at the custodian of the destination mobile, as only this custodian know the exact sensors and actuators deployed on its mobile. The translation for generic mobiles is performed by the custodian of a mobile that depends on the type of generic address.

- Role *unicast-next*. The translation is performed by the custodian of the source mobile.
- Role *unicast-next-geographical*. The translation is performed by the custodian of the source mobile, and potentially by the custodian of the fixed-relay.
- Role *multicast-next-all*. The translation is performed by the custodian of the source mobile.
- Role *multicast-next-chain*. The translation is performed by the custodian of each mobile of the chain, in turn.
- Role *multicast-next-geographical*. The translation is performed by all the custodian of the mobiles involved as intermediate mobiles, as well as by the custodian of all fixed-relays.

5. Conclusion

Several new applications require robots to move within an area, in order to monitor the environment while performing a task. The applications are likely to require that these mobile robots communicate, to avoid obstacles or to improve the monitoring for example. In this context, we have proposed a multi-channel architecture: each robot uses a robot-specific channel for intra-robot communications, and a common channel, called the meeting channel, is used to interconnect mobile robots that are in range of each other.

The use of a meeting channel allows mobiles to perform intra-mobile communications without interferences from neighbor robots, and it allows robots to communicate while some of their nodes save energy by sleeping. When there is no neighbor mobile in the vicinity, the meeting channel can be used to improve the intra-mobile communications. We conducted several simulations in order to quantify the benefits of the meeting channel, and we studied the performance of the network mainly in terms of throughput, delay and path length.

We proposed to use fixed-relays to improve the connectivity of the network, and to allow information to be stored at specific location in the environment. Fixed-relays are always on the meeting channel: this feature allows significant performance benefits.

Finally, we proposed the use of generic addresses in order to identify mobiles and nodes based on their role, rather than based on their address. This concept is especially important when a mobile robot has to communicate with the next mobiles.

6. References

Bahl, V., Chandra, R. & Dunagan, J. (2004). SSCH: Slotted seeded channel hopping for capacity improvement in IEEE 802.11 ad-hoc wireless networks, *ACM International Conference on Mobile computing and Networking*, pp. 216–230.

Bai, F., Sadagopan, N. & Helmy, A. (2003). IMPORTANT: a framework to systematically analyze the impact of mobility on performance of routing protocols for adhoc networks, *IEEE INFOCOM*, pp. 525–535.

Delay-Tolerant Networking Architecture (2006). *Request For Comments 4838*, IETF.

Ferranti, E. & Trigoni, N. (2008). Robot-assisted discovery of evacuation routes in emergency scenarios, *ICRA*, pp. 2824–2830.

Groenevelt, R., Nain, P. & Koole, G. (2005). The message delay in mobile ad hoc networks, *Performance Evaluation* 62(1-4): 210–228.

Hadid, N., Guitton, A. & Misson, M. (2010). Exploiting a meeting channel to interconnect 802.15.4-compliant mobile entities: discovery and association phases, *IEEE International Symposium on Computers and Communications*.

Hogg, R. W., Rankin, A. L., Roumeliotis, S. I., McHenry, M. C., Helmick, D. M., Bergh, C. F. & Matthies, L. (2002). Algorithms and sensors for small robot path following, *IEEE International Conference on Robotics and Automation*, pp. 11–15.

IEEE 802.15 (2006). Part 15.4: Wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (WPANs), *Standard 802.15.4 R2006*, ANSI/IEEE.

Jovanovic, M. D. & Djordjevic, G. L. (2007). TFMAC: Multi-channel MAC protocol for wireless sensor networks, *International Conference on Telecommunications in Modern Satellite, Cable and Broadcasting Services*, pp. 23–26.

Kyasanur, P. & Vaidya, N. H. (2005). Routing and interface assignment in multi-channel multi-interface wireless networks, *IEEE Wireless Communications and Networking Conference*, pp. 13–17.

Lambrou, T. P. & Panayiotou, C. G. (2009). Collaborative area monitoring using wireless sensor networks with stationary and mobile nodes, *EURASIP Journal on Advances in Signal Processing* .

Nasipuri, A., Zhuang, J. & Das, S.-R. (1999). A multi-channel CSMA MAC protocol for multi-hop wireless networks, *IEEE Wireless Communications and Networking Conference*, pp. 1402–1406.

Pathmasuntharam, J. S., Das, A. & Gupta, A. K. (2004). Primary channel assignment based MAC (PCAM) - a multi-channel MAC protocol for multi-hop wireless networks, *IEEE Wireless Communications and Networking Conference*, pp. 1110–1115.

Propagation data and prediction method for the planning of indoor radio communication systems and local area networks in the frequency range of 900 MHz to 100 GHz (2005). *Recommendation ITU-R P 1238-4*, ITU.

Ramakrishnan, M. & Vanaja Ranjan, P. (2009). Multi channel MAC for wireless sensor networks, *International Journal of Computer Networks and Communications (IJCNC)* 1(2): 47–54.

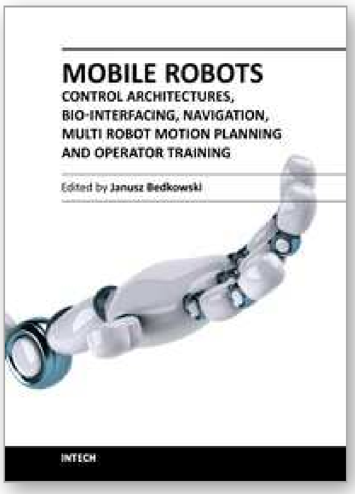
Seo, M., Kim, Y. & Ma, J. (2008). Multi-channel MAC protocol for multi-hop wireless networks: handling multi-channel hidden node problem using snooping, *IEEE Military Communications Conference*, pp. 1–7.

So, H. W. & Walrand, J. (2007). McMAC: A multi-channel MAC proposal for ad-hoc wireless networks, *IEEE Wireless Communications and Networking Conference*.

Wu, S.-L., Lin, C.-Y., Tseng, Y.-C. & Sheu, J.-P. (2000). A new multi-channel MAC protocol with on-demand channel assignment for multi-hop mobile ad hoc networks, *International Symposium on Parallel Architectures, Algorithms and Networks*, pp. 232–237.

Zhang, J., Zhou, G., Huang, C., Son, S. H. & Stankovic, J. A. (2007). TMMAC: an energy efficient multi-channel MAC protocol for ad hoc networks, *IEE ICC*, pp. 3554–3561.

Zhao, W., Chen, Y., Ammar, M., Corner, M., Levine, B. & Zegura, E. (2006). Capacity enhancement using throwboxes in DTNs, pp. 31–40.


MOBILE ROBOTS
CONTROL ARCHITECTURES,
BIO-INTERFACING, NAVIGATION,
MULTI ROBOT MOTION PLANNING
AND OPERATOR TRAINING
Edited by Janusz Bedkowski
INTECH

Mobile Robots - Control Architectures, Bio-Interfacing, Navigation, Multi Robot Motion Planning and Operator Training
Edited by Dr. Janusz Będkowski

ISBN 978-953-307-842-7
Hard cover, 390 pages
Publisher InTech
Published online 02, December, 2011
Published in print edition December, 2011

The objective of this book is to cover advances of mobile robotics and related technologies applied for multi robot systems' design and development. Design of control system is a complex issue, requiring the application of information technologies to link the robots into a single network. Human robot interface becomes a demanding task, especially when we try to use sophisticated methods for brain signal processing. Generated electrophysiological signals can be used to command different devices, such as cars, wheelchair or even video games. A number of developments in navigation and path planning, including parallel programming, can be observed. Cooperative path planning, formation control of multi robotic agents, communication and distance measurement between agents are shown. Training of the mobile robot operators is very difficult task also because of several factors related to different task execution. The presented improvement is related to environment model generation based on autonomous mobile robot observations.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821