Published in:
Industrial Robot International Journal, vol. 28, no. 3, 2001, pp. 220–232 (13)

A Self-Configuring Network for Mobile Agents

Thomas Bräunl, Peter Wilke
The University of Western Australia
Centre for Intelligent Information Processing Systems (CIIPS)
http://robotics.ee.uwa.edu.au {braunl, wilke}@ee.uwa.edu.au

Abstract

In a continuously changing environment with mobile agents coming and going and without any fixed infrastructure (no base stations), a self-configuring network is the only option. In addition, wireless communication for mobile agents is intrinsically non-reliable. Data may be lost due to an agent driving out of reach or interference with on-board devices (e.g. motors). The "EyeNet" self-configuring network for mobile agents fulfills these requirements. Agents communicate via a wireless virtual token ring network without central control and no need for a fixed master node.

1 System Concept

Our autonomous agents are a group of mobile robots, independent of any global sensor systems and without the need for remote computing on a host system. All sensor data evaluation, control and planning is done locally on-board each robot. In this scenario, there are a number of tasks, a self-configuring network could serve for:

1. To allow robots to communicate with each other
   E.g. sharing sensor data or cooperating on a common task or devise a shared plan.
2. To remote-control one or several robots
   E.g. giving low-level driving commands or specifying high level goals to be achieved.
3. To monitor robot sensor data
   E.g. display camera data from one or more robots or record a robot's distance sensor data over time
4. To run a robot with off-line processing
   E.g. combining of the two previous points, each sensor data packet is sent to a host where all computation takes place, the resulting driving commands are relayed back to the robot
5. To create a (multi-) robot monitoring console
   E.g. monitoring each robot's position, orientation and status in a multi-robot scenario in a common environment. This will allow a post-mortem analysis of a robot team's performance and effectiveness for a particular task

The network needs to be self-configuring. This means there will be no fixed or pre-determined master node. Each agent can assume the role of master. Each agent must be able to sense the presence of another agent and establish a communication link. New incoming agents must be detected and integrated in the network, while exiting agents will be deleted from it. An special error protocol is required because of the high error rate of mobile wireless data exchange.

2 Communication Details

We are considering a local wireless communication network for a large number of mobile agents. Using different frequencies for each agent is not viable, because not enough different channels are available in commercial modules and also because cross-talk and reflections would create a very high noise level that resulted in frequent drop-outs.

For the reasons mentioned above, we opted for a solution where all robot agents and optional base stations share the same transceiver module with the same frequency. On the physical level, we are using standard radio modules for half-duplex data transmission at a speed of 40 kb/s and up to 30 m distance. The on-board controller actively switches between sending/receiving and also performs data coding/decoding.

A number of protocol requirements have to be met:

- Frame control: Delimiting frames and character count to differentiate frames.
- Error control: Detection of errors and acknowledgment of correctly received messages. The error detection method used are cyclic redundancy check [Peterson] and sequence control.
- Initialization control: Establishment of an active data link over a communications facility.
- Flow control: Acceptance or rejection of information transfers.
- Link management: Controlling the transmission direction and identification of sender station.

- Transparency: Transmission of data streams without interfering of information and link control functions.
- Abnormal recovery: Handling of illegal sequences, loss of response etc.

2.1 Communication Protocol

In principle, a wireless network can be considered as a logical fully connected network, with every node able to access every other node in one hop, i.e. data is always transmitted directly without the need of a third party.

However, if we allowed every node to transmit data at any point of time, we would need some form of collision detection an recovery, similar to CSMA/CD [Wang94]. Since the number of collisions increases quadraticly with the number of agents in the network, we decided to implement a time division network instead. Several agents can form a network using a TDMA protocol (time division multiple access) to utilize the available bandwidth. In TDMA networks, only one transceiver may transmit data at a time, which eliminates data loss from transmission collisions.

So at no time, two transceiver modules may send data at once. There are basically two techniques available to satisfy this condition:

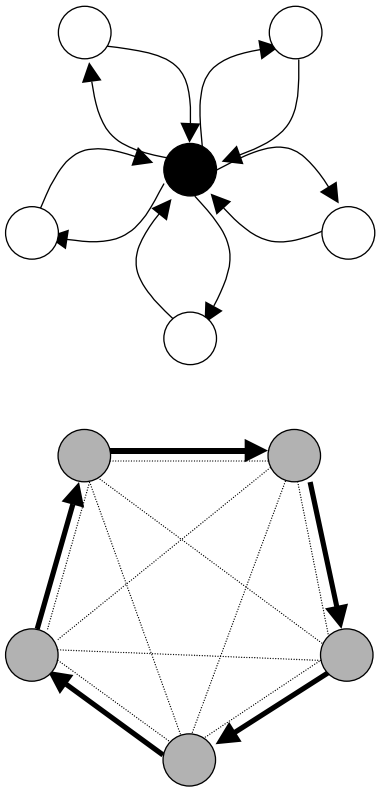


Figure 1. Polling versus Virtual Token Ring

- Polling:
  One agent (or a base station) is defined as the "master" node. In a "round-robin" fashion, the master talks to each of the agents subsequently to let it send one message, while each agent is listening to all messages and can filter out only those messages that apply to it.
- Virtual Token Ring:
  Like before, one agent becomes "master", initiating the ring mechanism and also takes over if a problem occurs like the loss of a token (see section on fault tolerance). The master sends the token to the next agent, which may send its messages and then passes the token on to the next agent in the list, and so on.

While both approaches are rather similar, the token approach has the advantage of less overhead, yet has the same functionality and safety. Therefore, we implemented the "Virtual Token Ring" approach.

1. The master has to create a list of all active robot nodes in the network by an initialization polling process. This list has also to be maintained during the operation of the network (see below) and has to be broadcast to all robots in the network (each robot has to know its successor for passing on the token).
2. The master has to monitor the data communication traffic with a time-out watchdog. If no data exchange happens during a fixed amount of time, the master has to assume that the token got lost (e.g. the robot that happened to have the token ``died''), and create a new one.
3. If the token is lost e.g. three times by the same agent, the master decides that this agent is malfunctioning and takes it off the list of active nodes
4. The master periodically checks for new agents that have become available (e.g. just switched on or recovered from an earlier malfunction) by sending a "wild card" message. The master will then update the list of active agents, so they will participate in the token ring network.

All agents (and base stations) are identified by a unique ID number, which is being used for addressing an agent. A specific ID number is used for broadcasts, which follows exactly the same communication pattern. In the same way, subgroups of nodes can be implemented, so a group of agents receive a particular message.

The control token is passed from one network node to another according to the set of rules that have been defined allowing the node with the token to access the network

medium. At initialization, a token is generated by the master station and passed around the network once to ensure that all stations are functioning correctly.

2.2 Data Transfer

A node may only transmit data when it is in possession of the control token. A node must pass the control token after it has transmitted an allotted number of frames. The procedure is as follows:

1. A logical ring is first established that links all nodes connected to the wireless network, and the master control station creates a single token.
2. The token is passed from node to node around the ring until that node which is waiting to send a frame receives it.
3. The waiting node then sends its frame using the physical medium, after which it passes control to the next node in the logical ring.

2.3 Recovery from Error

The major error recovery tool in the network layer is the timer in the master station that was mentioned earlier. The timer monitors the network to ensure that the token is not lost. If the token is not passed across the network for a certain period of time, a new token is generated at the master station and the control loop is effectively reset.

2.4 Base Stations

We assume to have a network of about 10 mobile agents (robots), which should be able to talk directly to each other without the need for a base station. However, it is possible to include a base station (PC or workstation) as one of the network nodes.

2.5 Message Structure

Messages are transmitted in a frame structure, containing a number of fields that carry specific information. These are:

- Start byte
  A specific bit pattern as frame preamble
- Message type
  e.g. user data, system data (see next section)
- Destination ID
  ID of the message's receiver agent or ID of a subgroup of agents or broadcast to all agents
- Source ID
  ID of the sending agent
- Next sender's ID
  ID of the next agent from the list of active agents, whose turn it is to send data
- Message length
  Number of bytes contained in message (may be 0)
- Message contents
  Actual message contents as specified by message length (may be empty)
- Checksum
  Cyclic redundancy checksum (CRC) over whole frame; automatic error handling for bad checksum

So each frame contains the id numbers of transmitter, receiver and the next-in-line transmitter, together with the data being sent. The "message type" is used for a number of organizational functions, described in the next section.

2.6 Message Types

As described above, each message sent between agents has a specific message type set in its packet. We have to distinguish three major groups of messages, which are:

- Messages exchanged at application program level
- Messages exchanged for enabling remote control at system level (i.e. keypresses, printing to LCD, driving motors)
- System messages in relation to the wireless network structure, which require immediate interpretation.

Distinguishing between the first two message types was required because the network was to serve more than a single task. Application programs should be able to freely send messages to any other agent (or a group of agents). Also, we wanted to be able to remote control or just monitor the behavior of an agent by sending messages, *transparent* to the application program.

This required the maintenance of two separate receive buffers, one for user messages and one for system messages, plus one send buffer for each agent. That way it can be guaranteed that both application purposes (remote control and data communication between agents) can be executed transparently at the same time.

The message types implemented are:

- USER
  A message sent between agents.
- OS
  A message sent from the operating system, e.g. transparently monitoring an agent's behavior on a base station.
- TOKEN
  No message contents is supplied, i.e. this is an empty message.
- WILD CARD
  The current master node periodically sends a wild card message instead of enabling the next agent in

the list for sending. Any new agent that wants to be included in the network structure has to wait for a wild card message.

- ADDNEW
  This is the reply message to "wild card" of a new agent trying to connect to the network. With this message the new agent tells the current master its ID number.
- SYNC
  If the master has included a new agent into the list or excluded a non-responding agent from the list, it generates an updated list of active agents and broadcasts it to all agents.

3 Self-Configuration of EyeNet

3.1 Ring Master

The token ring network is a symmetric network so during normal operation there is no need for a master node. However, a master is required to generate the very first token and in case of a token loss due to a communication problem or a agent malfunction, a master is required to restart the token ring.

There is no need to have the master role fixed to any particular node, in fact the role of master can be assigned temporarily to one of the nodes once the situation arises to perform the function mentioned above.

3.2 Network Startup

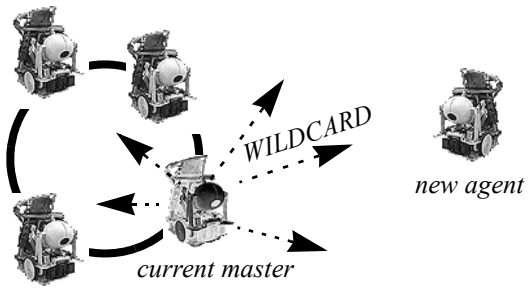
Since there is no master node, the network has to self-configure at initialization time and whenever a problem arises.

When an agent is activated (i.e. a robot is being switched on), it only knows its own id-number, but neither the total number or IDs of other agents, nor who the master is. Therefore, the first round of communication is performed only to find out:

- How many agents are communicating,
- What are their ID numbers and
- Who will be temporary master for starting the ring or recovery.

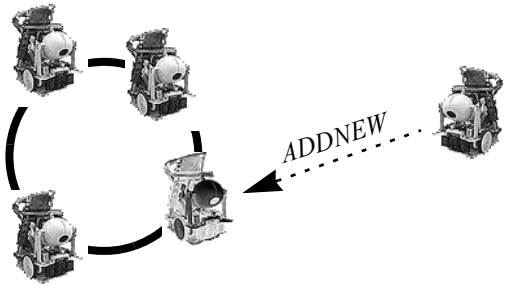
Agent ID numbers are used to trigger the initial communication. When the wireless network initialization is called for an agent, it first listens to any messages currently being sent. If no messages are picked up within a certain time interval multiplied by the nodes own ID number, the agent assumes it is the lowest ID number and therefore becomes master.

The master will keep sending tokens to the "wild card" address at regular time intervals, allowing a new incoming robot to enter and participate in the ring. The master will receive the new agent's ID and assign it a position in the ring. All other agents will be notified about the change in total robot number and any changes in the virtual ring neighborhood structure, via a broadcast SYNC message.

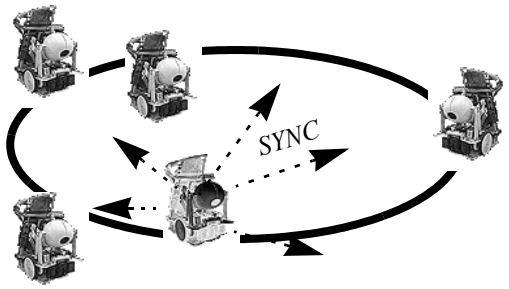


established network

Step1: Master broadcasts WILDCARD



Step2: New agent replies with ADDNEW



Step3: Master updates network with SYNC broadcast

Figure 2. Adding a node to the network

3.3 Fault-Tolerant Network Recovery

Each agent has an internal timer process. If, for a certain time, no communication has taken place, it is assumed that the token has been lost, e.g. due to a communication error, a agent malfunction or simply because one agent has been switched off. In the former case, the message can be repeated, but in the latter case (if the token is repeatedly lost by the same agent), that particular agent has to be taken out of the ring.

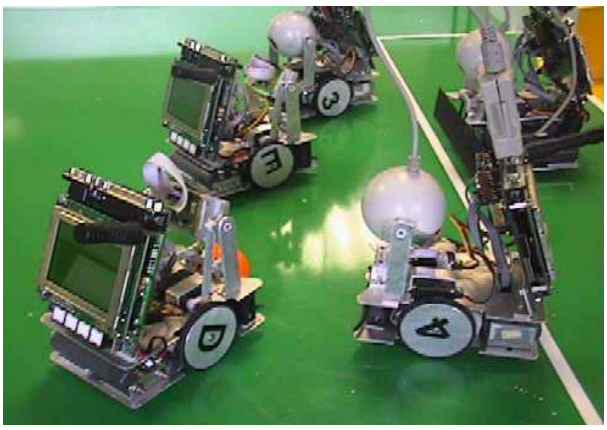


Figure 3. Group of EyeBot mobile robots

If the master node (assigned at a previous occasion) is still active, that agent recreates a token and monitors the next round for repeated token loss at the same node. If that is the case, the faulty agent will be decommissioned and a broadcast message updates all agents about the reduced total number and the change in virtual ring neighbor structure.

If there is no reply after a certain time period, each agent has to assume that it is the previous master itself which became faulty. In this case, the previous master's successor in the virtual ring structure now takes over its role as master. However, if this process does not succeed after an additional time period, the network startup process is repeated and a new master is negotiated.

4 User Interface

As has already been mentioned before, the wireless robot communication network *EyeNet* serves two purposes:

- Message exchange between robots for application specific purposes under user program control and
- Monitoring and remote controlling of one or several robots from a host workstation.

Both communication protocols are implemented as different message types using the same token ring structure, transparent to both the user program and the higher levels of the RoBIOS operating system itself.

Here we discuss the design and implementation of a multi-robot console that allows monitoring robot behavior, their sensor readings, internal states, and current position and orientation in a shared environment. Remote controlling of individual robots, groups of robots or all robots is also possible by transmitting input button data, which is interpreted by each robot's application program.

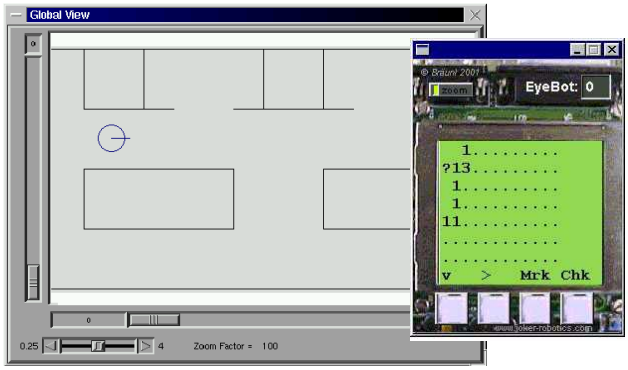


Figure 4. User interface

The *optional* host system is a workstation running Unix or Windows, accessing a serial port linked to a wireless module, identical to the ones on the robots. The workstation behaves logically exactly like one of the robots, which make the other network nodes. I.e. the workstation has a unique ID number and also receives and sends the token. System messages from a robot to the host are transparent from the user program and update the robot display window and the robot environment window. All messages being sent in the entire network are monitored by the host without the need to explicitly send them to the host. The EyeBot Master Control Panel has entries for all active robots.

The host only then actively sends a message, if a user intervention occurs, e.g. by pressing an input button in one of the robot's windows. This information will then be sent to the robot in question, once the token is passed to the host. The message is handled via low level system routines on the robot, so for the upper levels of the operating system it cannot be distinguished, whether the robot's physical button has been pressed or whether it was a remote command from the host console. Implementation of the host system largely reuses communication routines from the EyeBot and only makes a few system-dependent changes where necessary.

5 Related Work

5.1 Simple Position Sensing Networks

In the approach of Fung [Fung94] the position of a robot is gathered from infra-red sensor data and is then transmitted to other robots. Communication is transmitted via a two-way radio link. Robots are identified by a unique code which leads the transmitted position data. The received information is decoded and processed by the on-board microprocessor of the robot. The radio link uses an UHF transmitter/receiver and an encoding IC which detects if the address matches the robot's ID and then transfers the data directly to the microprocessor.

The communication network as suggested by Fung et. al. is tailor made to communicate very specific data between agents. The system is therefore inflexible to any changes of the communication system.

5.2 Inter-Robot-Network IRoN

IRoN is a robot communication network with modest cooperation and flexible enough to support a range of tasks [Rude97]. Cooperation is achieved by exchanging state variables using implicit communication. In addition explicit communication routines are available.

The signal to noise ratio is increased at the expense of bandwidth by using a spread spectrum technique with 19200 bits/second using a commercial radio modem, i.e. the energy of each single bit is spread over the entire frequency band.

Each team member stores a set of every other agent's state variables and is able to transmit them. The net-server module is a process in a multi-tasking environment of each robot and handles all transmissions. This process has two interfaces; one for the modem and the other for the processes exchanging data. The modem is connected to the RS232 interface on the host robot processor. Implicit communication is realized by updating the robot's state variable copy in each team member in regular intervals. Overflow and deadlock are avoided by giving implicit communication priority over explicit communication.

IRoN uses a token ring communication to perform time division multiplexing which is relatively easy to implement.

IRoN is a high performance robot network and is able to perform implicit and explicit communication. The drawback is that the state variables implicitly exchanged may not be suitable and/or sufficient for all robot applications, i.e. the complexity of the user processes increased while the net-server module is kept simple. Implicit communication is based on explicit communication, i.e. can be implemented on every system with explicit communication. IRoN can easily be modified to either maintain additional state variables and/or adding new processes to handle the communication on each host.

5.3 CSMA/CD

The Carrier Sense Multiple Access with Collision Detection (CSMA/CD) protocol divides time between the various agents. This sophisticated technique is more difficult to implement than the token ring communication but has a smaller overhead in each message [Wang94]. It is based on the CSMA/CD protocol and allows wireless inter-robot communication among multiple autonomous mobile robots with a centralized supervisor.

Two difficult problems have to be solved to used CSMA. First CSMA/CD relies on a centralized mechanism to detect collisions and second the detection of the collision itself is difficult for a sending and receiving unit.

These problems are solved by two changes to the protocol. First the network software ensures that all nodes have an individual message length which they transmit. Second the network software monitors the channel before and after an attempt to send. This is done to avoid start sending while the channel is in use and to detect that one or more nodes started sending at the same time.

Therefore, the nodes that are involved will be able to detect the collision except the one transmitting the longest message which must be notified by the others.

CSMA requires a transceiver, a modem, HDLC controller and a microcontroller to realize the protocol. The processes running on the microcontrollers receive the message and deliver the message frames to the robot system, i.e. frames are identified within the continuous data stream and those addressed to the current node are forwarded to the robot system. The nodes are in receive mode except when they are sending. The transmission mechanism is responsible for the transmission of message from the robot and the detection of collisions.

This modified CSMA protocol is quite complicated and difficult to implement. It requires significant amounts of processing capacity on the robot system. Its major advantages are the smaller overhead in each message and the non-centralized collision detection.

6 Summary

We presented a new wireless communication schema which can be applied for groups of autonomous mobile robot agents. The network is able to perform an automatic self-configuration not only when establishing the network, but also in case of changes to the nodes in the network or for transmission errors. The network is completely homogeneous without the need for a fixed master. Individual nodes will negotiate and take on the master role whenever necessary. Incoming new nodes will automatically be integrated into the network via "wild card" and "sync" messages, while exiting or faulty nodes will be automatically eliminated.

The physical communication port is a serial interface. The wireless transceiver operates on a single frequency using time division multiplex. A graphical user interface enables remote robot control, remote debugging, and monitoring of experiments of groups of robots. More information on the wireless network and the robot family is available from:

`robotics.ee.uwa.edu.au/eyebot/`

References

[1] [Agah95] Agah, A.; Bekey, G. In a Team of Robots the Loudest is not Necessarily the Best.. published in Proceedings of the International Conference on Systems, Man and Cybernetics, 1995, pp. 584-592.

[2] [Balch95] Balch, T.; Arkin, R. Communikation in Reactive Multiagent Robotic Systems. Autonomous Robots, 1:27-52, 1995.

[3] [Bräunl97] Bräunl, Thomas. Improv and EyeBot - Real-Time Vision on-board Mobile Robots, published in 4th Annual Conference on Mechatronics and Machine Vision in Practice (M2VIP), Toowoomba QLD, Australia, Sep. 1997, pp.131-135 (5).

[4] [Bräunl98a] Bräunl, Thomas. CIIPS Glory - Autonomous Robot Soccer Players with Local Intelligence, Proceedings RoboCup Workshop, 2nd Robot World Cup Soccer Conference, Paris, 1998, pp. pp. 497-502 (6).

[5] [Bräunl98b] Bräunl, Thomas; Graf, Birgit. Robot Soccer with Local Vision. published in Pacific Rim International Conference on Artificial Intelligence, Singapore, Nov. 1998, pp. 14-23 (10).

[6] [Cao95] Cao, Y.; Fukunaga, A.; Kahng, A.; Meng, F. Cooperative Mobile Robotics: Antecedents and Directions, IEEE Workshop on Intelligent Robots and Systems, 1995, pp. 226-234.

[7] [Conard89] Conard, J. Character-Oriented Link Control, chapter 2, pp. 83-107. published in [Sunshine89], 1989.

[8] [Dudek93] Dudek, G.; Jenkin, M.; Milios, E.; Wilkes, D. A Taxonomy for Swarm Robots. published in IEEE, RSJ, Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 1993, pp. 441-447.

[9] [Franklin95] Franklin, R.; Harmon, L. Elements of Cooperative Behavior. Technical Report ERIM TR 655404-1-F, Environmental Research Michigan, USA, 1995. page 698-709.

[10] [Fukuda94a] Fukuda, F.; Sekiyama, K. Communication Reduction with Risk Estimate for Mul-tiple Robotic Systems, IEEE. Proc. of the IEEE Conference on Robotics and Automation, 1994, pp. 2864-2869.

[11] [Fukuda94b] Fukuda, T.; Ueyama, T. Cellular Robotics and Micro Robotic Systems. World Sci-entific, Singapore, 1994.

[12] [Fung94] Fung, C.; Eren, H.; Nakazato, Y. Position Sensing of Mobile Robots for Team Operations, Proceedings of IMTC, 1994 , pp. 185-188.

[13] [Jablonski85] Jablonski, J.; Posey, J. Robotics Terminology, pp. 1271-1303. In Nof, S., Hrsg. Handbook of Industrial Robotics. J. Wiley, NJ, USA, 1985.

[14] [Kitano + 97] Kitano, H.; Tambe, M.; Stone, P.; Veloso, M.; Coradeschi, S.; Osawa, E.; Matsu-bara, H.; Noda, I.; Asada, M. The RoboCup synthetic agent challenge 97. Proc. of IJCAI-97, 1997, pp. 24-29 (6).

[15] [Lin94] Lin, F.; Hsu, J. Cooperation and Deadlock-Handling for an Object Sorting Task in an Multi-Agent Robotic System., IEEE. Proc. of the IEEE Conference on Robotics and Automation, 1994, pp. 2580-2585.

[16] [MacLennan91] MacLennan, B. Synthetic Ecology: An Approach to the Study of Communica-tion. published in Farmer al. et., Hrsg. Artificial Life II. Addison-Wesley, MA, USA, 1991, 1991.

[17] [Mataric92] Mataric, M. Integration of Representation into Goal-Driven Behavior Based Ro-bots. IEEE Transactions on Robotics and Automation, pp. 304-312, 1992.

[18] [Peterson61] Peterson, W.; Brown, T. Cyclic Codes for Error Detection, Proceedings of the IRE, 1961, pp. 228-235.

[19] [Premvuti90] Premvuti, S.; Yuta, S. Consideration on the Cooperation of Multiple Autonomous Mobile Robots, IEEE Workshop on Intelligent Robots and Systems, 1990, pp. 59-63.

[20] [Rude + 97] Rude, M.; Rupp, T.; Matsumoto, K.; Sutedjo, S.; Yuta, S. IRoN: An Inter Robot Network and Three Examples on Multiple Mobile Robots' Motion Coordination, published in IEEE Workshop on Intelligent Robots and Systems, 1997, pp. 1437-1444.

[21] [Sunshine89] Sunshine, C., (Ed.). Computer Network Architectures and Protocols. Plenum Press, Santa Monica, CA, USA, 1989.

[22] [Wang94] Wang, J.; Premvuti, S. Resource Sharing in Distributed Robotic Systems based on a Wireless Medium Access Protocol, EEE, RSJ, GI. Proceedings of the IEEE/RSJ/GI, 1994, pp. 784-791.

[23] [Werner90] Werner, G.; Dyer, M. Evolution of Communication in Artificial Organisms. Tech. Rep. UCLA-AI-90-06, University of California at Los Angeles, UCLA, CA, USA, June 1990.

[24] [Whiting75] Whiting, J. An Efficient Software Method for Implementing Polynomial Error Detection Codes. Computer Design, pp. 73-77, 1975.

[25] [Yoshida + 95] Yoshida, E.; Yamamoto, M.; Arai, T.; Ota, J.; Kurabayshi, D. A Design Method of Local Communication Area in Multiple Mobile Robot Systems. published in IEEE Proc. of the IEEE Conference on Robotics and Automation, 1995, pp. 2567-2572.

[26] [Yuta93] Yuta, S. Cooperative Behavior of Multiple Autonomous Mobile Robots. published in Proceedings Workshop on Needs for Research in Cooperating Robots, Atlanta, USA, 1993.