

A Self-organizing Link Layer Protocol for UWB Ad Hoc Networks ¹

N. Shi and I.G.M.M. Niemegeers

Center for Wireless and Personal Communication
Delft University of Technology, Delft, the Netherlands
{n.shi, i.niemegeers}@ewi.tudelft.nl

Abstract. Ultra Wide Band (UWB) impulse radio is a promising technology for future short-range, low-power, low cost and high data rate ad hoc networks. The technology is being explored in a number of research projects. While most UWB research for this class of networks is concentrating on the physical layer, little research has been published on link layer protocols which exploit the specifics of UWB impulse radio. In this paper, we focus on the self-organization concept and the peculiarities of UWB technology from a physical and a link layer point of view. A novel self-organizing link layer protocol based on time hopping spread spectrum is proposed in this paper. This protocol promises to be an efficient and collision-free mechanism that enables the devices to discover neighbor nodes and arrange the access to communication resources shared among the nodes. The adjustable parameters of the protocol enable the network to adapt to a dynamic environment.

1. Introduction

A range of services supporting future mobile applications are expected to require high data rates, high communication quality and efficient network access. A case in point is mobile interactive gaming, where fast transmission of image and voice in dynamic environments is a prerequisite.

Wireless networks that meet these expectations will have a hybrid character, consisting mainly of ad hoc networks with occasional access to infrastructures, in order to reach remote nodes or infrastructure-based servers. They will have to operate completely automatically without the intervention of system administrators, and therefore will have to be self-organizing. Self-organization in this context implies the automatic finding of neighbor nodes, the creation of connections, the scheduling of transmissions and the determining of routes. This should be performed in a distributed manner so that all nodes in the network are able to exchange information and reconfigure the network when nodes join or leave, or when radio links are broken or established.

A promising but, because of implementation difficulties, not well explored radio technology is UWB impulse radio. This technology has a lot of potential for high-

¹ This research is part of the AIRLINK project funded by the Dutch Ministry of Economic Affairs under the Freeband Impulse Program.

bandwidth, low-power, low-cost and short-range communication in self-organizing ad hoc networks. Self-organization is helped by flexible radio resource management. UWB impulse radio has this intrinsic flexibility [1] because of the parameters of the physical layer that can be explicitly controlled by the upper layers. Furthermore it offers the potential for achieving high data rates with low spectral density and immunity to interference [2]. Its radar like pulses can in principle also be used for high precision positioning.

The research on UWB-based ad hoc networks is still in its infancy. Currently, IEEE 802.15.3a [3] is considering UWB technology for the physical layer. A debate is going on whether this should be based on OFDM or on impulse-based radio. OFDM technology is well understood, for impulse-based radio on the other hand, research is needed to understand the practical limitations and to come up with technical solutions. This paper intends to contribute to this goal.

An ongoing project, which explores UWB impulse radio technology for use in short-range ad hoc networks, is the AIR-LINK project [4]. The research reported on here was carried out in this project. We approach the UWB ad hoc network for the applications that need to efficiently establish connections and exchange data at high speed within a short range. We consider the scenarios where the transmission distance is about 10 to 100 meters, the data rate is at least 100 Mbps and the time needed for establishing a network is of the order of 0.1s.

In this paper the key issues involved in UWB ad hoc and self-organization are analyzed and a new link layer device discovery protocol is proposed to discover disconnected nodes within radio range and establish links among them. We assume that the network layer and higher layers will be based on IETF protocols. We also assume that the communicating devices are energy constrained, since many of them will be portable and battery powered.

The paper is organized as follows. In Section 2, we introduce the concept of self-organization. In Section 3, we describe an UWB ad hoc network and address the physical and link layer issues. In Section 4, a new self-organization device-discovery protocol (SDD) is proposed and specified. A conclusion is made in Section 5.

2. Self-organization

Self-organization refers to the ability of a system to achieve the necessary organizational structures without human intervention [5]. In our case it is the process network nodes go through to autonomously organize and maintain a network topology either at network initialization or during operation. During the latter the topology may change due to nodes joining or leaving the network, and links appearing and disappearing. Note that, given a set of connected nodes, the topology may change because extra (radio) links become operational, making the topology more interconnected. One should also envisage the partitioning of a given topology into two, no longer connected networks and the merger of two networks when a radio link is established between them.

Self-organization plays a role both at link and network layer. The issues that need to be addressed are neighbor discovery and connection setup, link scheduling and

channel assignment, network topology formation and re-configuration, control and routing information adaptation and mobility management [6].

Self-organization has been investigated in ad hoc networks, Bluetooth scatternets, and sensor networks. Recently a number of protocols have been proposed in these areas. An example of a device discovery protocol for short-range wireless ad hoc networks is the DD protocol proposed in [7]. It has been designed to be time-efficient when the number of devices in the network is large. The protocol is designed to perform well when devices that are in-range of each other become active in a short time interval. Some Bluetooth scatternet construction protocols [8] contain self-organization procedures that cater for the case of nodes frequently joining and leaving the scatternet, leading to dynamic topologies. However these protocols rely on the Bluetooth random-access inquiry mechanisms, which are reputed to be slow; delays of the order of 10s [9] are not unusual. Therefore, these protocols are unsuitable for very dynamic environments. Sensor networks [10] are formed by collections of distributed sensor nodes in order to sense the environment and inform users. The self-organization process in this type of network is designed with an emphasis on prolonging the lifetime of the network without the need for human assistance. This is achieved by initiating a communication-link schedule among the nodes and maintaining this schedule periodically in the long-term. Protocols belonging to this class have been proposed in [11], e.g., SMACS and EAR. Because of the static nature of the sensor nodes, the self-organization process is mostly a one-time initialization effort, except in case nodes fail.

3. UWB Ad hoc Networks

The UWB ad hoc network we envisage is composed of a set of nodes, each of which is assumed to be equipped with an UWB transceiver. The nodes are personal digital devices such as notebooks, PDAs, mobile phones or, in principle, any device with computing and UWB communication capabilities. An example is shown in Fig. 1.



Fig. 1. UWB ad hoc network

3.1 Communication Architecture

We assume that the network has a multi-hop architecture. This implies that we can distinguish two domains: the link domain corresponding to sets of nodes that are

within radio range of each other, and the network domain corresponding to multi-hop interconnection of nodes in different link domains (Fig. 1). Protocols for device discovery, the establishment of UWB physical connections between a node and its neighbors and MAC protocols belong to the link domain. The network domain is where end-to-end multi-hop connectivity is realized. In this domain, each node is assumed to have equal functionality. In particular, each node is able to initiate end-to-end communication and to forward packets on behalf of other nodes.

3.2 Impulse Radio

The UWB impulse radio signal consists of a spreading pulse train in a framed period. The duration of each frame is equal and is divided into multiple time bins. Each pulse belonging to one signal is repeated per frame in a randomly positioned time bin (Fig. 2). The position sequence, which is called *time hopping sequence* or *TH code* [12], is based on a pseudorandom (PN) process. The pulses are further dithered within a bin based on the information bit of the signal.

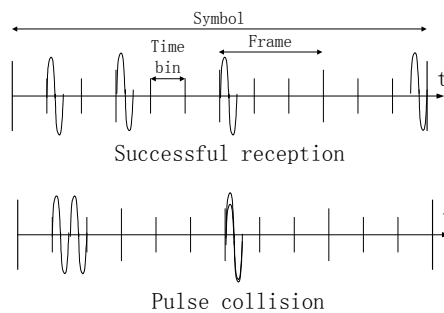


Fig. 2. Impulse radio signal

The reception of an impulse radio signal is performed by a correlator [13]. A template signal is applied to the received signal to retrieve the original pulse train. The expected TH code is used to generate the template signal. When pulses of two signals (Fig. 2) arrive in one time bin, a pulse collision happens. When two signals encoded by the same TH code are received simultaneously, a collision happens if they overlap exactly.

Impulse radio offers a flexibility that can be exploited in the design of a self-organizing ad hoc network. First the parameters, e.g., the TH code, can be used to identify the network nodes and to indicate the required radio resource. Second, because a TH code is applied to the original pulse train, only a small fraction of the frame is occupied (typically one percent or less) [14]. Thus, a collision hardly happens in multiple packets transmission. If a device uses two orthogonal TH codes for transmitting and receiving, the signals can be received without interfering with the signals that are being transmitted; if a device simultaneously transmits or receives multiple packets which are transmitted on orthogonal TH codes, collisions are avoided.

However, the potential drawback of impulse radio is its long signal acquisition time. This is the time required for a receiver to collect sufficient energy for bit synchronization. Synchronization between two devices requires one or more signal acquisition times. Because the transmitted pulse train has an extremely low power spectral density, impulse radio receivers inherently have a longer signal acquisition time than conventional receivers [15]; this can be of the order of milliseconds. Our physical layer design aims at a value of 0.1ms.

3.3 Link Layer Issues

UWB-based medium access control (MAC) can in principle be achieved by contention-based and by explicit methods. Both methods are impacted by the long acquisition time. The impact is more obvious in the contention-based method. For instance, the CSMA/CA protocol has a high packet delay because the signal acquisition occurs twice per packet transmission [16].

Since we consider a dynamic environment, access schemes with fixed periodical schedules, e.g., TDMA as used for sensor networks [11] are clearly not applicable. Consistent with the requirement that the ad hoc communication should be completely distributed, link establishment should be performed in a symmetrically distributed way; link information should be exchanged with neighbor nodes. Hence, synchronization should not be required in the network domain but only in the link domain between transmitters and receivers. Finally, since time hopping spread spectrum is used, the nodes have to have different TH codes so that they are able to initialize conflict-free links with neighboring nodes.

4. The Device Discovery Protocol

Self-organization of an UWB ad hoc network requires new link layer protocols to be able to automatically discover the in-range nodes, form a distributed link-layer topology and dynamically arrange the access to communication resources shared among the nodes.

4.1 Protocol Assumptions

We consider the situation that a node has no prior knowledge about its surroundings. When a node is powered on, the node is able to establish direct links with in-range nodes after it has exchanged its time hopping pattern with these nodes. When a node loses most of its direct links, this is an indication that the topology of the network has significantly changed; the node then needs to discover its neighbors again. We also need a link level mechanism for data transmission over established links. An RTS-CTS handshake is used for this purpose.

In the sequel we describe a self-organizing distributed protocol on top of the TH-based impulse radio physical layer. We also describe the essentials of the protocol for data transmission.

We consider a scenario where a variable number of nodes within radio range of each other are expected to form and maintain a single-hop ad hoc network. It is assumed that the nodes can move at most at walking speed. Let $K(t)$ denote the number of the nodes in-range at time t . $K(t)$ increases when nodes that are powered on or move into the radio range join; $K(t)$ decreases when nodes that are powered off or move out of the radio range leave. Each node keeps a list of the TH codes of its neighbors. This list is denoted by $L_i(t)$ (i can be any node, $i \leq K(t)$). We assume that the TH codes are sufficient to identify a node. $L_i(t)$ is updated whenever a node discovers a new TH code or detects the disappearance of a neighbor; it records the TH codes of its neighbors.

A node is not aware of the value of $K(t)$; it only knows $L_i(t)$. When a node i initially joins, it has no information about the other $K(t)-1$ nodes, i.e., $L_i(t)$ is empty, unless it contains outdated information about its surroundings. If a node j other than i leaves an established network, the information in each $L_i(t)$ is updated only after node i detects the absence of node j . In addition, in order to avoid an explicit removal procedure, a TH code in $L_i(t)$ expires after it has not been used for a period longer than a timeout T_{\max} .

We assume omni-directional antennas, a negligible propagation delay and errorless channels. The nodes know their own TH codes. We assume that the UWB device consumes very little power when it is continually monitoring a TH code.

We define a packet collision on a TH code if two or more packets transmitted on the same TH code arrive at one node at the same moment; the physical layer detects this. When multiple packets transmitted on different TH codes arrive at a node simultaneously, the node is able to decode the information on its own TH code while the information on other codes is perceived as noise.

4.2 Self-organizing Device Discovery Protocol (SDD)

The device discovery protocol, SDD, is based on a RTS-CTS type of signaling dialog on the link layer.

4.2.1 Overall Protocol Procedure

The SDD protocol is a link layer protocol. It directly interfaces with the physical layer and the network layer. Via the physical layer interface the following information is communicated:

Physical layer to link layer:

- A collision indication.
- The TH code of the received packet, i.e., the identification of the sender.
- The information decoded from the packet received on a particular TH code.

Link layer to physical layer:

- The information to be sent out via the physical layer

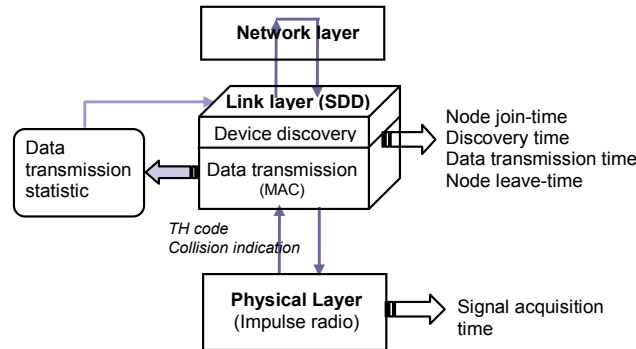


Fig. 3. SDD overall protocol procedure

The performance parameters of the SDD protocol, in particular, the discovery time, the node join-time, the node leave-time and the data transmission time, are impacted by the parameters of the physical layer, in particular by the signal acquisition time.

Via the network layer interface the following information is communicated.

Link layer to network layer:

- Current link information used for routing
- Service data unit that needs to be forwarded

Network layer to link layer:

- Service data unit when data transmission is requested from the application layer of the device
- Service data unit when data is requested to be forwarded

The SDD protocol is initialized when the node is powered on and terminates when the node is powered off. The protocol consists of two types of sub-processes: *device discovery* and *data transmission* (Fig. 3).

A device discovery process can be driven by a discovery process timer with an adaptive time interval, denoted by T_{interval} . The short time interval is used when a device is powered on to form a network and join a network or, in case it has reason to believe it has lost its connectivity with its surroundings, e.g., due to movement. In the former case, the neighbor list $L_i(t)$ is empty. A node could assume the latter has happened, e.g., when the data transmissions to most of its neighbors on $L_i(t)$ keep failing persistently. In addition when a destination node one-hop away from node i is not included in $L_i(t)$, a device discovery process has to be executed before the data transmission process can be initialized. The discovery process will execute in a long interval, T_{interval} , set to a large value, when the number of the discovered nodes $L_i(t)$ is larger than a maximum value, L_{MAX} , or no more neighbors are discovered in three continuous discovery processes.

The data transmission process in a node associated to one of its neighbors executes whenever there is a request from the higher layers of the node to send data to its neighbor.

The device discovery and data transmission processes contend in using the physical channel resources. For example a node could be attempting to discover other nodes while a data-sending request arrives from upper layer or one neighbor node. We will show later that both actions require the use of the same TH code. However one process cannot preempt another one and has to wait until the other process is completed and resources are released again.

4.2.2 Definitions

We assume each node is assigned three basic types of TH codes [17]. The codes are used in the device discovery process and the data transmission process.

- *Common code C*: a fixed short TH access code known by all nodes. It is used to initialize device discovery and data transmission through a broadcast operation.
- *Receiver-based code $C_{i,R}$* : the TH code of node i for receiving packets from neighboring nodes. It is used in device discovery for an inquiring node to receive responses from the nodes inquired by it.
- *Transmitter-based code $C_{i,T}$* : It is the TH code of node i for transmitting packets to neighboring nodes. This code is used in both device discovery and data transmission. In *half-duplex* communication (e.g., i to j), the code in use is always associated with $C_{i,T}$, which is the transmitter-based TH code of the node that needs to send data. *Full-duplex* communication between nodes i and j is initialized by exchanging the transmitter-based codes of both nodes, i.e., $C_{i,T}$ and $C_{j,T}$.

The receiver-based and transmitter-based TH codes are generated by a pseudo random process. The unique 48-bit IEEE MAC address of a device can be used as the key of the process.

The protocol defines seven PDU types defined as follows:

- *IS packet* (Inquiry): The purpose of this packet is to inquire the as yet undiscovered neighboring devices. It contains packet type, the TH codes of the inquiring node, i.e., $C_{i,R}$ and $C_{i,T}$, and the *response scan window* size $N_i(t)$. It is transmitted on the C code.
- *IR packet* (Inquiry Reply): The purpose of this packet is to reply to the IS packet. Apart from the packet type, it contains the transmitter-based code of the inquired node $C_{j,T}$. It is sent using the receiver-based TH code $C_{i,R}$ of the inquiring node.
- *LRS packet* (Low Rate Synchronization): This packet is used to keep synchronization between the inquiring node and the undiscovered neighbors it is inquiring. It is a short packet with packet type information. It is sent by the inquiring node using code $C_{i,T}$.
- *RTS packet* (Request to Send): The purpose of this packet is to synchronize and inform the potential receiving node to be ready for data transmission. It contains packet type, the transmitter-based TH codes of the receiving node(s) and transmitting node, e.g., $C_{j,T}$ and $C_{i,T}$, assuming that the data should be transmitted from i to j . It is sent using the C code.

- *CTS packet* (Clear to Send): The purpose of this packet is to confirm the request for data transmission. It contains the code of the transmitting node, i.e., $C_{j,T}$, to match to the RTS. It is sent using the C code.
- *Data packet*: This packet carries the data that needs to be sent to the receiver. The packet is transmitted using code $C_{i,T}$ assuming node i is the sender.
- *ACK packet*: The purpose of this packet is to confirm the successful reception of a Data packet. The packet is sent using the $C_{j,T}$ code.

An operational node has eight states. These states correspond to different phases of the protocol, which are explained as follows:

- *Inquiry*: The node is initiating an inquiry request in order to discover the as yet unknown neighboring nodes.
- *Inquiry response scan*: An inquiring node stays in this state while waiting for the inquiry responses from its not yet discovered neighboring nodes.
- *Inquiry response*: An inquired node stays in this state until it replies to the inquiring node.
- *Data send request*: The node is broadcasting a request to send a packet.
- *Data request response*: A node is replying to the sender to be clear to send.
- *Data send*: The data is being sent from this node.
- *Data receive*: The node is receiving data.
- *Idle*: The node state is idle, when there is no ongoing operation on the node. At this time, common code is always free and available for initiating new operations.

4.2.3 Device Discovery Process

The function of the device discovery process is to automatically discover new in-range nodes and quickly form a distributed link-layer topology. The discovery process adapts itself to the number of nodes that are in-range.

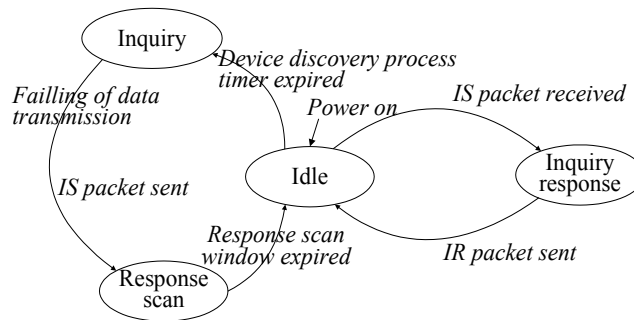


Fig. 4. State transition diagram of the device discovery process

The device discovery process includes the operations of *inquiry* and *response scan* running on the inquiring node, and *inquiry response* running on the inquired nodes. Fig. 4 shows the state transition diagram of the device discovery process.

After a node is powered on, it enters the *Idle* state and waits for a random period T_{start} . Then it moves to the *Inquiry* state. A transition from the *Idle* state to the *Inquiry* state can also occur when either the device discovery process timer expires or the node detects during data transmission that the number of consecutive failing transmissions to is above a threshold or the destination node is not a neighbor of the node. In *Inquiry* state an IS packet is generated by the node and is broadcast. After sending the IS packet, the inquiring node moves to the *Response scan* state to receive the responses from its neighbor nodes. All neighbor nodes in *Idle* state will receive the IS packet and synchronize with the inquiring node. Then, each inquired node gets into the *Inquiry response* state and schedules an IR packet at a randomly chosen time to reduce the probability of packet collisions. After these operations, the inquiring and inquired nodes return to the *Idle* state.

We use an example with two nodes i and j to illustrate how the process works. As shown in Fig. 5, when node i is switched on, node i first stays in *Idle* state for a random period, T_{start} . The discovery process starts when node i broadcasts an IS packet on code C . The acquisition header of the IS packet allows node j , which is in range and listening, to synchronize with node i . After sending the IS packet, node i starts a response scan operation for receiving responses from the inquired nodes. The number of the slots during the response scan operation is called the *response scan window size* $N_i(t)$. During this operation, node i will periodically send LRS control packets on code $C_{i,T}$ to keep all inquired nodes synchronized.

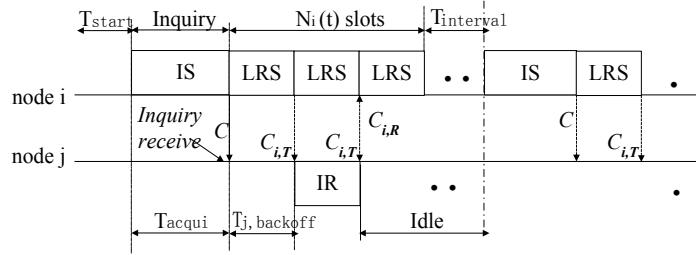


Fig. 5. Device discovery process

If the link i - j already exists, j doesn't have to reply and returns to *Idle* state monitoring code C . Otherwise, j starts listening on code $C_{i,T}$ to receive periodically LRS packets, which keep it synchronized with node i . After a random number of time slots, denoted by $T_{j,backoff}$, node j responds with an IR packet on code $C_{i,R}$. $T_{j,backoff}$ is determined by node i 's scanning window size $N_i(t)$. It is randomly chosen in the range $[0, N_i(t)]$. Afterwards, node j stops listening to code $C_{i,T}$ and returns to *Idle* state. If node j is successful in sending the IR packet containing node j 's TH code $C_{j,T}$, node i and j have discovered each other and are ready for data transmission.

Adaptation algorithm of response scan window

A node i can adjust the response scan window size $N_i(t)$ that will be used for the next device discovery process based on the reception statistics during the response scan operation. The result of every time slot as observed by node i can be *null*, *success* or *collision*; null implies that no IR packet was received nor a collision occurred, success

means that an IR packet was successfully received and collision means that a collision was detected. Node i counts the number of the three results during $N_i(t)$ time slots. At the end of the response scan operation, if the number of *null* results is much larger than the other two, it means that the response scan window size $N_i(t)$ is too large for the number of in-range nodes and can be reduced without significantly increasing the probability of collisions. If the number of *success* results is dominating, it implies that $N_i(t)$ is a suitable number. Finally, if the number of *collision* results is much larger than the others, it indicates that $N_i(t)$ is too small and should be enlarged for the next device discovery process.

4.2.4 Data Transmission Process

The data transmission process is initiated by an RTS-CTS handshake using code C. Unicast and multicast are both supported. In multicast operation, multiple TH codes of the destination nodes are included in one RTS packet. Our design allows simultaneously transmission and reception on several orthogonal TH codes. Multicast operation is done by encoding the same data information into the multiple TH codes and sending them at the same time. A successful multicast operation is confirmed if the source node receives the acknowledgments from all the destination nodes.

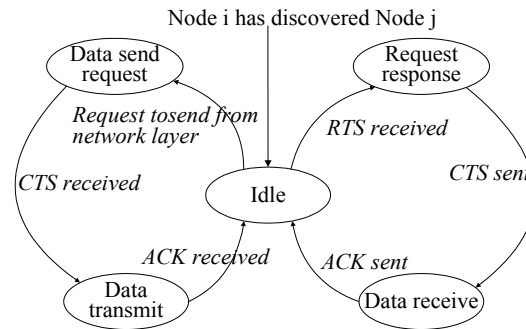


Fig. 6. State transition diagram of the data transmission

Data transmission can only happen after the destination node has been discovered by the source node. Data transmission is invoked by a request from the network layer or a reception of a RTS packet. In Fig. 6, when a node is in *Idle* state and needs to send data to a neighbor node, it makes a transition to the *Data send request* state and sends an RTS packet. If the destination node is in *Idle* state, it responds with a CTS packet and moves to the *Data receive* state. The source node then begins sending data and moves to the *Data send* state. After the source node receives confirmation of successful data transmission by receiving an ACK from the destination node, both nodes make a transition to the *Idle* state.

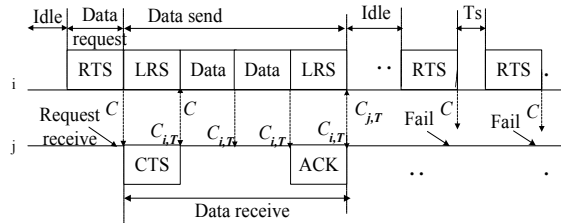


Fig. 7. Data transmission

Fig. 7 illustrates this process. Combining two half-duplex links, i.e., by having a handshake in both directions, forms a full-duplex link. Let us assume that node i needs to send data to node j . As shown in Fig. 6, if node i is *Idle*, it first broadcasts an RTS packet on code C . Without waiting for the response, node i continues to send an LRS packet on code $C_{i,T}$ to keep node j synchronized. Right after node j receives the RTS packet, it replies to node i by sending a CTS on code C . In the meantime, node j begins to monitor code $C_{i,T}$. Subsequently, the data packets are transmitted on code $C_{i,T}$ from node i to node j . After all the data packets are transmitted, node j replies to node i with an ACK packet on code $C_{j,T}$. Finally both nodes i and j return to *Idle* state.

When node i does not receive a CTS packet from node j , node i will periodically attempt to resend an RTS packet with period T_s on code C for at most m times. This is illustrated in Fig. 7. If none of the RTS packets are replied to, it means that either node j has left the radio range of node i , has been powered off or radio communication between i and j is disturbed for a longer time.

4.2.5 Collision Analysis in Multiple Nodes Example

In the SDD protocol, we assume that a packet collision only happens when two or more packets arrive at the receiver at the exact same moment on the same TH code. The collision problems of SDD protocol on the common code and on other codes can be illustrated by using multiple nodes examples of device discovery process.

Collision avoidance on the common code

If a node successfully accesses code C , the other nodes in its radio range are aware of this. Every node uses code C only in the beginning of the device discovery process and the data transmission process. A collision happens on code C only when two or more IS or RTS packets arrive at a node at the same time. When the packets partly overlap, collisions don't happen. The random backoff procedure (with backoff time T_{start}) has been introduced in the SDD protocol (see Section 4.2) to avoid potential collisions when all the in-range nodes are powered on at the same time. Therefore, the probability of a collision on the common code C is minimized.

Collision on code $C_{i,R}$

In the Inquiry Response state of the device discovery process, the number of successful transmissions depends on the mismatch between the response scan window size $N_i(t)$ and the number of neighbor nodes that receive the inquiry, denoted by $N_{i,inquired}(t)$. When $N_{i,inquired}(t)$ is much larger than $N_i(t)$, i.e., the maximum number of responses node i can receive, there is a higher probability of collisions. In Fig. 8, node i sends an IS packet on code C and starts to scan code $C_{i,T}$. The in-range nodes j and k successfully receive node i 's inquiry. Node j and k only scan code $C_{i,T}$ to stay synchronized but do not scan code C any more. Thus, the IS or RTS packets from other nodes, e.g., node l , cannot be received by node j or k . Once their random back-off periods, i.e., $T_{j,backoff}$ and $T_{k,backoff}$, are expired, they send IR packets to node i on code $C_{i,R}$. When $T_{j,backoff}$ is equal to $T_{k,backoff}$, a collision happens between their IR packets (see Fig. 8). Node i detects this collision on $C_{i,R}$ and sets the reception result of the time slot as *collision*. Afterwards, node j and k return to monitor code C .

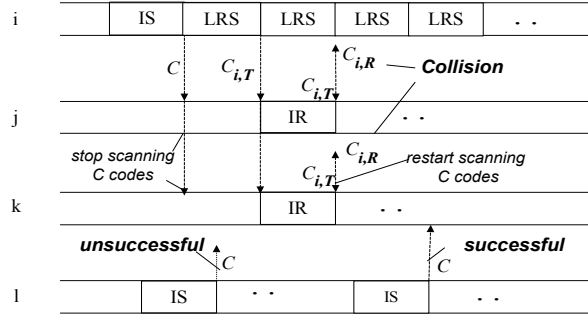


Fig. 8. Collision on $C_{i,R}$

4.3 System Parameters and Variables

The following system parameters and variables are used and need to be determined in the SDD protocol:

- $K(t)$: The number of nodes in radio range at a time.
- $L_i(t)$: The list of the TH codes that node i has found at a time.
- L_{MAX} : The maximum number of neighbors or TH codes that a node can store.
- $T_{interval}$: The time interval between two adjacent device discovery processes.
- T_{lmax} : The maximum time that a TH code can be kept unused in $L_i(t)$ before it expires.
- T_{start} : The random period that a node has to wait before moving to the Inquiry state after it is powered on. It is used to avoid collisions caused by simultaneous power on.
- $N_i(t)$: Response scan window size at time t .
- T_{slot} : The duration of a time slot, a typical value could be 0.05ms
- $N_{i,inquired}(t)$: The number of nodes that receive an inquiry from node i as a result of a single inquiry.

- $T_{i,backoff}$: A random number of time slots that a node waits before sending a response to an inquiry.
- T_s : The repetition period of an RTS message, in case a neighbor does not answer with a CTS packet.
- m : The maximum number of times that a device repeats an RTS packet, in case a neighbor does not answer.

We are presently investigating the performance of the protocol by means of a Glomosim [18] simulation model. The most important performance parameters are the node join-time, the discovery time of the device discovery process, the data transmission time of a packet, the node leave-time and the throughput.

5. 5. Conclusion

In this paper we have presented a novel self-organizing link layer protocol for short-range ad hoc networks based on impulse radio UWB. The protocol exploits the specific features of time hopping spread spectrum. This protocol promises to be an efficient and collision-free mechanism that enables the devices to discover neighbor nodes and arrange the access to communication resources shared among the nodes. The adjustable parameters, e.g., the time interval of the device discovery processes and the adaptive response scan period enable the protocol to adapt to dynamic environment. Finally, the main system and performance parameters were identified.

A critical parameter in UWB impulse radio systems is the potentially long acquisition time. This parameter has an impact on the device discovery process and the efficiency of data transmission. In the AIRLINK project we expect that by using optimized signal processing technology a significant reduction of the acquisition time can be obtained.

Some issues are to be solved in the future research. The effort on the reduction of the acquisition time will be carried out. The optimized protocol procedure will also be worked out to reduce the performance degradation caused by the long acquisition time.

The protocol is being implemented in a Glomosim simulator, which will be used to analyze the performance and to optimize the system parameters.

References

1. M.G. Di Benedetto, P. Baldi, "A Model for Self-organizing Large-scale Wireless Networks", Invited paper, Proceedings of the International Workshop on 3G Infrastructure and Services, Athens, July 2001, pp. 210-213
2. J. Foerster, E. Green, S. Somayazulu, D. Leeper, "Ultra-Wideband Technology for Short- or Medium-range Wireless Communications", Intel Technology Journal Q2, 2001
3. <http://grouper.ieee.org/groups/802/15/pub/TG3a.html>
4. <http://www.freeband.nl/projecten/airlink/ENindex.html>
5. H. Balakrishnan, S. Seshan, P. Bhagwat, M.F. Kaashoek, "Self-organizing Collaborative Environments", (Position paper) NSF/DARPA/NIST Workshop on Smart Environments, Atlanta, GA, July 1999.

A Self-organizing Link Layer Protocol for UWB Ad Hoc Networks

6. A.N. Zadeh, B. Jabbari, "Self-organizing Packet Radio Ad Hoc Networks with Overlay (SOPRANO)", *Communications Magazine, IEEE*, vol. 40, issue 6, June 2002, pp. 149 - 157
7. P. Popovski, T. Kozlova, L. Gavrilovska, R. Prasad, "Device Discovery in Short-Range Wireless Ad Hoc Networks", *Proc. 5th Conf. Wireless Personal Multimedia Communications (WPMC)*, October 2002
8. T. Salonidis, P. Bhagwat, L. Tassiulas, R. LaMaire, "Distributed Topology Construction of Bluetooth Personal Area Networks", *INFOCOM 2001, Proceedings. IEEE*, vol. 3, April 2001, pp. 1577 - 1586
9. D. Groton, J.R. Schmidt, "Bluetooth-based Mobile Ad Hoc Networks: Opportunities and Challenges for a Telecommunication Operator," *Vehicular Technology Conference*, Spring 2001, vol. 2, pp. 1134-1138.
10. I.F. Akyildiz, Weilian Su; Y. Sankarasubramaniam, E. Cayirci, "A Survey on Sensor Networks", *Communications Magazine, IEEE*, vol. 40, Aug 2002, pp. 102 - 114
11. K. Sohrabi, J. Gao, V. Ailawadhi, G.J. Pottie, "Protocols for Self-organization of a Wireless Sensor Network", *IEEE Personal Communications*, October 2000
12. M.Z. Win, R.A. Scholtz, "Ultra-Wide Bandwidth Time-Hopping Spread-Spectrum Impulse Radio for Wireless Multiple-Access Communications", *IEEE Transactions on Communications*, vol. 48, no. 4, April 2000
13. M.Z Win, R.A. Scholtz, "Impulse Radio: How It Works", *IEEE Comm. Letters*, vol. 2, no. 2, February 1998, pp. 36-38
14. F. Cuomo, C. Martello, A. Baiocchi, "Radio Resource Sharing for Ad Hoc Networking With UWB", *IEEE Journal on Selected Areas In Communications*, vol. 20, no. 9, December 2002
15. S.S.Kolenchery, J.K.Townsend, J.A. Freebersyser, "A Novel Impulse Radio Network for Tactical Military Wireless Communications", *Proceedings of IEEE Milcom '98, Boston*, October 1998.
16. J. Ding, L. Zhao, S.R. Medidi, K.M. Sivalingam, "MAC Protocols for Ultra-Wide-Band (UWB) Wireless Networks: Impact of Channel Acquisition Time", *ITCOM*, 2002
17. M. Joa-Ng, I.T. Lu, "Spread Spectrum Medium Access Protocol with Collision Avoidance in Mobile Ad-hoc Wireless Network", *INFOCOM '99. Proceedings. IEEE*, vol. 2, March 1999, pp. 776 - 783
18. <http://pcl.cs.ucla.edu/projects/glomosim/>