

A LINK LAYER PROTOCOL FOR SELF-ORGANIZING ULTRA WIDE BAND IMPULSE RADIO NETWORKS

Nan Shi, Liang Xia and Ignas G. Niemegeers

*Center for Wireless and Personal Communications, EWI, Delft University of Technology,
Mekelweg 4, 2628 CD, Delft, The Netherlands {n.shi;l.xia;i.niemegeers@ewi.tudelft.nl}*

Abstract: Ultra Wide Band (UWB) impulse radio, promises to be suitable for short-range, low-power, low cost and high data rate applications. While most UWB research is concentrating on the physical layer, little research has been published on the link layer. A novel self-organizing link layer protocol (SDD) based on time hopping impulse radio was proposed by the authors. In this paper, the SDD protocol is further developed and specified in detail. The simulations are carried out using the GloMoSim simulation environment.

1. INTRODUCTION

Future Personal Area Networks (PANs) should be able to support a large variety of personal applications. Some of the more demanding ones will be, for instance, video conferencing or interactive games. Ultra Wide Band impulse radio (UWB-IR) promises to be suitable for such short-range, low-cost and high data rate applications while having a very low energy consumption. IEEE 802.15 is the standardization body which covers the link layer and physical layer technologies for PANs. UWB-IR is one of the technologies considered within IEEE 802.15.3a.

PANs that meet these expectations will have a hybrid character, consisting mainly of ad hoc networks, using IEEE 802.15 air-interfaces, with occasional access to infrastructures. Because of the mobility of a person and the devices that constitute a PAN, and the dynamics of the applications [1], nodes will join or leave the network, and radio links will be broken or

established. Therefore the network should be able to quickly configure and reconfigure itself. These operations should be done without the intervention of a user or a system administrator, and therefore will have to be self-organizing. Self-organization in this context implies the discovery of neighbors, the creation of connections, the scheduling of transmissions and the formation and re-configuration of the network topology.

In [2] we introduced a novel link layer protocol, for PANs based on UWB-IR, i.e., Self-organizing Device discovery and Data transmission (SDD) protocol. The SDD protocol makes use of time hopping multiple access, which is an intrinsic feature of UWB-IR. This protocol is able to automatically discover nodes within radio range, form a distributed link layer topology and assign channel resources for collision-free transmissions.

In this paper we highlight the key aspects in the design of the SDD protocol, present a further developed and optimized version of the protocol and analyze its behavior by means of simulation. In Section 2, we discuss issues related to the design of UWB-IR based networks. In Section 3, we describe the sub-processes of the SDD protocol. In Section 4, the performance parameters which characterize the quality of self-organization in the SDD protocol are specified and analyzed by means of simulation in Section 5.

2. DESIGN ISSUES OF UWB-IR BASED NETWORKS

In order to design a high data rate and fast medium access in UWB ad hoc networks, a number of facts and design issues related to the UWB-IR technique need to be kept in mind.

- TH impulse radio: Low duty cycle and low power density are crucial properties of TH impulse radio. Multiple access can be achieved by TH spread spectrum, which uses a pseudorandom time hopping sequence, called the TH code. TH impulse radio also has in principle the capability to determine its location with great accuracy, i.e., of the order of sub-centimeter [3].
- Collisions: As a result of the low duty cycle of TH impulse radio, the probability that packets overlapping in time collide is decreased. When multiple packets transmitted on different codes arrive at a node simultaneously, the node is able to decode the information on its own code while the information on other codes is perceived as noise [5]. This enables simultaneous in-range transmissions and receptions on different codes. This improves the aggregate capacity of the network. We base our design on the assumption that no more than 10 nodes communicate simultaneously. This allows us to neglect the multiple access interference

(MAI) in this instance and ensure that the achievable data rate of a node is not affected [4]. Furthermore, when overlapping packets sent on the same code arrive (Fig.1), multiple pulses are received in one frame and the pulses may not exactly overlap. The result is an increase in MAI but not a collision since packets are still successfully received. By using a novel blind synchronization method [6], the receiver is able to decode the overlapping packets of the same code. A packet collision happens only when two or more packets with the same code exactly coincide in time at a node; the physical layer will detect this collision.

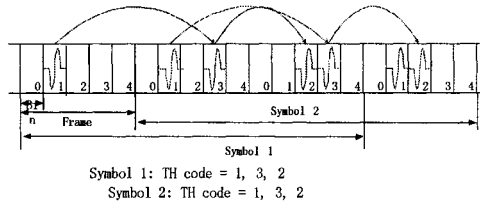


Figure 1. Packets overlapping on the same TH code.

- Carrier-sensing: The low power density and low duty cycle of TH impulse radio however implies that it is difficult to distinguish idle periods from packet transmission periods in an UWB receiver [7] and then a long channel detection delay can not be avoided. As a result, MAC protocols based on carrier-sensing would result in poor performance.
- Acquisition time: The signal acquisition time of impulse radio receivers is longer than conventional receivers (of the order of ms). The transmission of one data packet between two devices normally requires one or more signal acquisitions. The long signal acquisition time may degrade the achievable throughput [8]. To mitigate this effect we have a physical layer design which targets a relatively short acquisition time, of the order of 0.1ms [6]. Moreover, the link layer protocol design should as much as possible avoid frequent synchronizations.
- Spreading code: TH impulse radio has a large and adjustable code family; it enables each node to be assigned different codes, e.g., a TH sequence [5]. We define three basic types of codes: *common*, *transmitter-based* and *receiver-based* codes, denoted respectively by C , $C_{i,T}$ and $C_{i,R}$. The receiver-based and transmitter-based codes can be derived from a node's unique 48-bit IEEE MAC address by using a pseudo random process, e.g., using the method proposed in [9]. The common code is defined as a signaling channel for exchanging specific control messages.
- Neighbor discovery: Using TH spread spectrum as the multiple access method implies that data transmission is not possible before the communication codes are known by the sending and receiving nodes.

Neighbor discovery is the process during which nodes get to know the presence and identity of their neighbors.

3. OPERATIONS OF THE SDD PROTOCOL

In this section we discuss performance related aspects and optimizations of the SDD protocol. For more details on the operation of the protocol we refer to [1]. We consider a scenario where a number of nodes within radio range of each other are expected to form and maintain a single-hop ad hoc network. The protocol consists of two sub-processes: *Device Discovery* (DD) and *Data Transmission* (DT).

3.1 Device Discovery Process

The DD process can discover multiple nodes per synchronization. This is unlike Bluetooth and it clearly speeds up the network formation. For example, Bluetooth needs up to 10.24s to finish one discovery cycle; on average, it spends 5~6s finding only one device (see Bluetooth specifications). We will show that (see Section 5) our protocol uses around 50ms to find 10 nodes and 70ms to find 50 nodes.

We illustrate how the DD process works. To simplify the explanation, the propagation delay is not shown in the following figures. As shown in Fig. 2, if node i is switched on, it first stays in *Idle* state for a random period with uniform distribution. Then, node i broadcasts an Inquiry (IS) packet on the common code C . The acquisition header of the IS packet allows the nodes in range and listening, e.g. node j , to synchronize with node i . After sending the IS packet, node i starts a response scan operation which is able to receive multiple responses from the inquired nodes. During this operation, node i periodically sends Low Rate Synchronization (LRS) packets on code $C_{i,T}$ of node i to keep all inquired nodes synchronized.

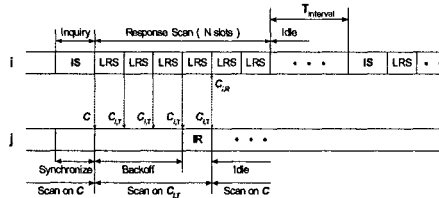


Figure 2. Device Discovery process.

If the link i - j already exists, j doesn't have to reply and returns to *Idle* state monitoring code C . Otherwise, after a random number of time slots, denoted by $T_{j,backoff}$, node j responds with an Inquiry Response (IR) packet on

the code $C_{i,R}$ of node i . $T_{j,backoff}$ is determined by node i 's scanning window size $w_{i,scan}$. It is uniformly distributed in the range $[0, w_{i,scan}]$. Afterwards, node j returns to *Idle* state. If node j is successful in sending the IR packet containing node j 's code $C_{j,T}$, node i and j have discovered each other and are ready for data transmission.

The number of slots during the response scan is called the *response scan window size* $w_{i,scan}$. The larger $w_{i,scan}$, the lower the probability of response packet collisions but on the other hand the longer the response scan duration. Two parameters are related to $w_{i,scan}$, of which one is the initial value of the scan window size W after a node is powered on and the other is the adaptation factor β . β is to scale the changes of $w_{i,scan}$ each time. The information of the discovered node is stored in the neighbor list. The maximum size of the neighbor list is denoted as l_{max} . The discovery interval T_{int_short} and T_{int_long} determines the frequency with which the DD process is executed.

3.2 Data Transmission Process

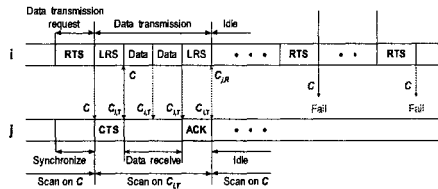


Figure 3. Data Transmission process.

The DT process is initiated by an RTS-CTS handshake using code C . Fig.3 illustrates the unicast data transmission process for two nodes. Node i needs to send data to node j . 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 packet on code C . In the meantime, node j begins to monitor code $C_{i,T}$. Subsequently, the data packet is transmitted on code $C_{i,T}$ from node i to node j . If it succeeds, 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 in T_s interval for at most m times. This is illustrated in Fig.3. If there is still no response after m number of RTS packets, it means that either node j has left the radio range of node i , node j has been powered off, or the connection between node i and j has been disturbed for a long time.

4. PERFORMANCE EVALUATION

1) Discovery time:

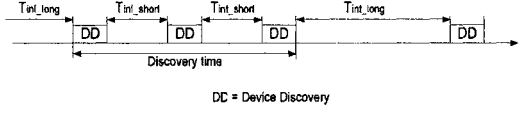


Figure 4. SDD discovery time.

We define *discovery time* \bar{T}_{disc} as the average time spent by a self-organization protocol to enable any newly entering node or group of nodes to discover and incorporate with all or most neighboring nodes. It should be short enough to allow timely configuration and re-configuration of the link topology relative to the dynamics of the network. A relevant measure is the time for a particular node i to detect a neighboring node j has joined or left, denoted by $T_{i,j}^{join}$ and $T_{i,j}^{depart}$.

The discovery time $T_{i,disc}$ of the SDD protocol is defined as the duration of a set of successive DD processes with short average interval. It is counted from the moment that a node using T_{int_short} interval to the moment it changes to T_{int_long} interval mean (see Fig.4). The period of one DD attempt is denoted as $T_{i,dd}$, $T_{i,dd} = v_{IS} + w_{i,scan} \times \sigma$ where v_{IS} is IS packet transmission time and σ is the time slot used in response scan window. The average discovery time \bar{T}_{disc} is as follows:

$$\bar{T}_{disc} = \frac{1}{N_{total}} \sum_{l=1}^{N_{user}} \left(\sum_{k=1}^r T_{i,dd}(k) + \sum_{k=1}^{r-1} T_{int_short} \right) \quad (1)$$

There are k ($1 \leq k \leq r$) times DD attempts within the discovery time for a single node. r is obtained from the simulation.

2) Discovery ratio:

We define the *discovery ratio* $\bar{\rho}$ as the average percent of the number of neighbors on the list in the total number of nodes within the radio range. Note that this measure relates to the network as a whole and not to an individual node. Discovery ratio is calculated as the average percentage of discovered neighbors after one discovery time. d_i is the degree of any node i .

$$\bar{\rho} = \frac{1}{N_{total}} \sum_{i=1}^{N_{total}} \frac{d_i}{N_{total} - 1} \quad (2)$$

3) Effect on throughput:

When network nodes frequently join and leave, the self-organization processes, i.e., device discovery and connection setup, need to be performed more often. We define the *effect on throughput* as the achievable throughput measured when discovery processes are performed to the throughput measured when discovery processes are not performed. We measure the

achievable throughput when DD process is performed as well as when DD process is not performed. The comparison of the two measurement results can show the effect of DD process on throughput. Throughput is defined as the average successful transmission rate.

5. SIMULATION RESULTS

We did an initial performance evaluation of the SDD protocol using GloMoSim. Based on an interactive game scenario described in [1], we consider a number of nodes forming a single-hop network within an area of $10\text{m} \times 10\text{m}$, i.e., corresponding to an in-room environment. The nodes are less than 50 and are uniformly distributed over the simulation area. Node mobility is neglected in this scenario. We assume an error-free channel, hence all packet losses are due to collisions or buffer overflow.

Table 1. Parameter specifications

Parameter	Value	Parameter	Value
Achievable data rate	110 Mbps	Initial time T_{int}	[1ms, 20ms]
Traffic model	CBR	Slot duration for DD σ	20 μs
Packet size	2000 bytes	T_{int_short}	10ms
Buffer size	10000 packets	T_{int_long}	1s
Channel acquisition time	0.1 ms	Retransmission interval mean $E[T_r]$	0.2 ms
IS packet size	36 bytes	Retransmission times m	3
IR packet size	20 bytes	Response scan window initial value W	15 or 32 slots
RTS packet size	36 bytes	Neighbor list size l_{max}	10 to 100 entries
CTS packet size	36 bytes	Neighbor list time out T_{lmax}	100 s

The system parameters used to evaluate the SDD protocol are listed in Table 1. We set the achievable data rate to 110Mbps corresponding to the IEEE 802.15.3a standard. The signal acquisition time is set to 0.1ms, which can be further improved [6]. In order to avoid frequent synchronizations, we use a large packet length of 2000 bytes. The choice of protocol parameters is mostly based on preliminary experiments.

5.1 Discovery time

We first investigate the influence of the response scan window size on *discovery time*. We assume l_{max} is larger than the number of nodes and hence does not impact the discovery time.

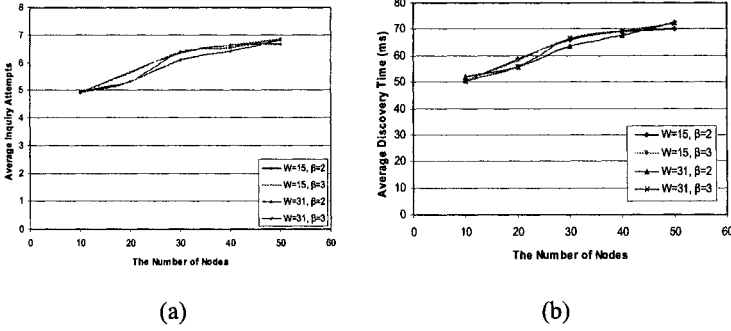


Figure 5. (a) The average number of DD processes in one discovery; (b) The average discovery time.

In Fig. 5, we observed a certain degree of linearity in terms of the number of nodes. Moreover, the maximum discovery time for a 50-node setting is less than 75ms which is much lower than the magnitude of the discovery time in Bluetooth. We also observe that as we use different initial scan window size W and the adaptation factor β (see Fig. 5), \bar{T}_{disc} is slightly different within a certain range. The difference is caused by the different adaptation parameters W and β . The difference is not so dramatic because in the initial setting, T_{int_short} is one or two orders of magnitude larger than the response scan window period which is from 0.2 ms to 0.7ms. We now investigate the influence of the neighbor list size l_{max} on average discovery time \bar{T}_{disc} for different numbers of in-range nodes. In this experiment, the initial scan window size W is chosen as 31, and the adaptation factor β is 2.

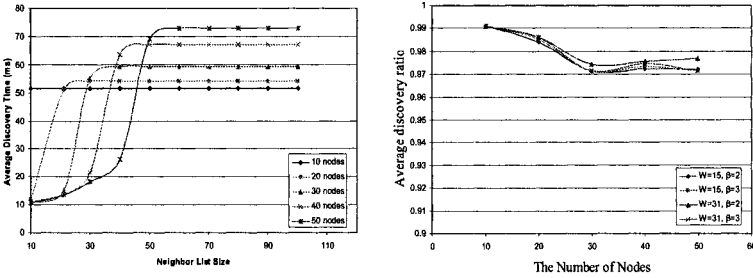


Figure 6. The average discovery time for different neighbor list sizes.

Figure 7. Average discovery ratio.

As shown in Fig. 6, we can see that when the neighbor list size l_{max} is smaller than the number of nodes, the discovery time increases as l_{max} increases. The reason is that the DD processes with T_{int_short} interval terminate only according to one condition that the neighbor list is full. More time is needed by each node to fill in its neighbor list when l_{max} increases. In this case, the degree d_i of each node is equal to l_{max} . If l_{max} is equal to or

larger than the number of adjacent nodes, when the network is fully connected, the discovery time reaches a steady maximum value. This is because the DD processes with T_{int_short} interval terminate due to the fact that no new neighbor has been found in the last three contiguous discovery processes.

5.2 Discovery ratio

With the same parameter settings as in the initial experiments, the average discovery ratio was measured. We set the neighbor list size l_{max} much larger than the total number of node. From Fig.7, we observe that the average discovery ratio $\bar{\rho}$ is always higher than 0.97 for different number of nodes. It shows that most nodes can find most of their neighbors in one discovery time. Thus, the degree of any node is very close to the number of in-range nodes. Therefore, the node similarity can be satisfied in the network topology formed by SDD protocol. In Fig.7, when the number of nodes increases, $\bar{\rho}$ slightly decreases. This is because when more nodes come up in the network, the probability that more DD processes overlap with each other increases, as well as the probability of collisions of packets.

5.3 Throughput behavior

This experiment evaluates the influence of the DD processes on the system throughput. We only consider 4 nodes in this experiment and thus the effect of MAI and routing protocol can be neglected. In both the situations of without and with DD processes, the initial DD process has to be executed.

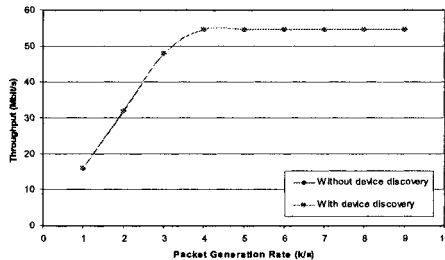


Figure 8. Throughput behavior with/without device discovery.

When the packet generation rate is smaller than 4000 pkts/sec and the network is lightly loaded, the throughput increases with the packet generation rate (see Fig.8). For higher packet arrival rates, the network is saturated and the throughput stays above 54 Mbit/s. When the network is either lightly loaded or saturated, we observe that the device discovery has very little influence on the system throughput. It benefits from the high data rate and capability of multiple transmissions offered by TH impulse radio.

The fast transmission of control packets on the common code reduces the probability of collision occurrence between the DD and the DT processes. The multiple transmissions on different codes enable the two simultaneous processes to be successfully executed.

6. CONCLUSION

In this paper, the key characteristics in the design of the self-organizing link layer protocol SDD for impulse radio UWB were described. The SDD protocol was implemented in a GloMoSim simulator. Results show that the SDD protocol has good performance in the operations of neighbor discovery and medium access. In the current experiment conditions and assumptions, the average discovery time is much lower than the discovery time in Bluetooth. The node similarity can be satisfied in the single-hop network topology. The device discovery process has very little influence on the system throughput. Presently, we are working on the multi-hop case and on performance analysis using different traffic models.

REFERENCES

1. W. Lu, A. Lo and I. Niemegeers, "On the Dynamics and Self-organization of Personal Networks", MAGNET Workshop in Shanghai, November, 2004.
2. N. Shi and I. Niemegeers, "A Self-organizing Link Layer Protocol in UWB Ad Hoc Networks", Proceedings of Personal Wireless Communications: IFIP TC6 9th International Conference, PWC 2004, Delft, September 2004, pp. 248-261.
3. M.G.M. Hussain, "Principles of Space-time Array Processing for Ultrawide-band Impulse Radar and Radio Communications", IEEE Transactions on Vehicular Technology, vol. 51, issue 3, May 2002, pp 393-403.
4. L. Zhao, A.M. Haimovich, "The Capacity of an UWB Multiple Access communication System", Communications, 2002. ICC 2002. IEEE, vol.3, 28 April-2 May 2002, pp.1964 - 1968.
5. 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, pp. 679-689.
6. R. Djapic, G. Leus and A-J. van der Veen, "Blind Synchronization in Asynchronous Multiuser UWB Networks Based on the Transmit-reference Scheme", The Asilomar Conference on Signals, Systems, and Computers, Pacific Grove CA, November 7-10, 2004.
7. I. Oppermann, "The Role of UWB in 4G", Special Issue on Adaptive Global Net: Vision Towards a Modern Communication Infrastructure, Wireless Personal Communications, April 2004, vol. 29, no. 1-2, pp. 121-133(13).
8. 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.
9. J.J. Garcia-Luna-Aceves, J. Raju, "Distributed assignment of codes for multihop packet-radio networks", MILCOM 97 Proceedings, vol.1, November, 1997, pp. 450-454.