

LON Technology in Wireless Sensor Networking Applications

Marek Miskowicz * and Ryszard Golanski

AGH University of Science and Technology, Department of Electronics, al. Mickiewicza 30, 30-059 Kraków, Poland. E-mails: miskow@agh.edu.pl, golanski@agh.edu.pl

*Author to whom correspondence should be addressed.

Received: 4 December 2005 / Accepted: 3 January 2006 / Published: 20 January 2006

Abstract: In the paper a discussion on how to optimize LonWorks/EIA-709 sensor networking technology for wireless applications, is presented. Main solutions offered by Local Operating Networks (LON, LonWorks) platform attractive for wireless communication, that is, the send-on-delta concept and the sleep mode, are displayed. The predictive p -persistent CSMA MAC protocol constituting the heart of the communication capability of LON networks is analysed in detail. Next, the message services are described, and the analytical evaluation of delivery reliability is derived. Performance evaluation based on simulation results for unicast traffic is presented first. In order to highlight the robustness of the predictive CSMA to overload situations, the saturation performance for a general case load scenario including multicast transactions is reported. The methods of effective management of energy consumption in LonWorks networks are discussed. Finally, the LON design tradeoffs are summarized.

Keywords: intelligent sensors, carrier sense multiaccess, communication system reliability, communication system performance, radio communication.

1. Introduction

Local Operating Networks (LON, LonWorks) is a name of one of the leading technologies in sensor/control networking, addressed to a wide range of applications. Due to architectural flexibility, LonWorks smart devices are intended not only to acquire data from the environment in a *sensor network*, but also to interact with the object being sensed in a feedback loop as a *control network*.

LON has become a classic solution in building automation, and home networking including all key building automation subsystems: heating, ventilating, and air conditioning, lighting, security, fire

detection, access control, energy monitoring, etc. Among others, LonWorks platforms are used also in semiconductor manufacturing, pulp and paper equipment, material handling, textile machinery, petrochemical, food and beverage, automotive, and wastewater treatment.

Most devices in LON-based building automation system are connected by wire (heating, ventilation, lighting, access control, etc.). Wireless transmission is useful to building-to-building communication, environmental monitoring of particular areas of the building or of materials stored in a building. Furthermore, wireless LON devices are used among others in public transportation monitoring systems, security camera monitoring and control, gas detection, fire valve control, smoke detection [22].

LonWorks technology allows us to develop networked systems (i) with a large number of nodes that do not require synchronization and cover relatively wide distances, (ii) peer-to-peer architecture, (iii) using multiple media including wireless communication, (iv) supporting battery-powered nodes, normally powered down, activated only for message sending or receiving.

In order to facilitate data the exchange between nodes, a programming language of LON applications (Neuron C) offers a special class of objects called *network variables* that can be considered as the logical input/output of a node [2]. In typical use, the change of state of a particular output network variable automatically evokes updating of input network variable(s) that are logically bound with it in order to inform them of a new value. By default, updating is done *implicitly* by triggering off message(s) without any explicit instruction in the application code dealing with message sending, buffering, addressing, or communication services. Furthermore, a set of *Standard Network Variable Types* (SNVTs) has been defined in order to report measurements of common physical magnitudes being sensed. Each SNVT is denoted by its identification number encoded in updates and dedicated to a particular physical magnitude. The definition of the SNVT includes also the unit of physical variable, range, and resolution. For example, illumination measurements can be reported by the network variable type called SNVT_lux, ranging from 1 to 65535 lux with 1 lux resolution. Today there are more than 140 SNVTs. However, the list of used SNVTs is open and new network variable types might be introduced.

The next interesting issue in LonWorks/EIA-709 technology is the strategy of reporting the environment temporal evolution. LonWorks smart sensor nodes monitor a state of the environment being sensed according to *send-on-delta concept* rather than *periodically* [2]. In the send-on-delta technique, the sensor node does not broadcast a new message until the input signal remains within a certain interval of confidence. The sampling occurs when the signal deviates by *delta*, defined as the *significant change* of input (e.g. 1°C), in relation to the most recent report. The use of send-on-delta technique instead of the traditional time-triggered updates reduces the mean rate of messages generated by a sensor node [8]. The reduction is significant for *burst signals*, defined as a type of signals that do not change, much or at all, for most of the time. Throttling the traffic originated out of sensor nodes improves the network performance and saves the energy consumption.

The other solution dedicated to efficient energy management is to put a LonWorks sensor node into a *sleep mode*, when it is not engaged in data processing [5]. Sleeping node wakes up when some activity is detected on I/O port and/or in the network. The sleep mode is attractive for battery-powered sensor nodes. Use of sleep mode is especially efficient if the node does not interact with the other nodes and only reports rare events occurring in its local environment. Sleep mode can be also used together with the send-on-delta concept if the mean sampling interarrival time is long. However,

unlike in most wireless sensor networking technologies, the energy consumption is *not* the main constraint in LonWorks design since the batteries, powering devices, are in general replaceable.

Sensor networks are designed in a variety of ways to address different priorities and make the appropriate technology tradeoffs according to specific application requirements. Recent technological improvements and most research activity for the last five years have been devoted to development of *dense wireless sensor networks* consisting of small, inexpensive, low-power, randomly deployed distributed devices, capable of local processing and wireless communication. Due to enormous amount of research, the term wireless sensor networks is identified in most literature with the concept of *dense wireless sensor networks*.

Unlike dense wireless sensor networks that are used commonly in environmental monitoring, LonWorks technology has not been designed for detection of event of interest at the same time by a large number of nodes in a sensor field. The dense networks are *data-centric*, meaning that data is requested based on certain attributes such as, which area has temperature more than 20°C [20]?. Meanwhile, LON sensor networks are *node-centric* since data is requested from a specific node. Furthermore, LON networks use *statically-assigned addresses* that are guaranteed to be unique, i.e. each node has a unique identification number (ID), used for routing. Instead, *attribute-based addressing*, where nodes are described by attributes other than addresses are appropriate for dense sensor networks [21].

However, the LON node processing and storage capabilities are in general greater than that offered by a typical sensor node in a dense network. Consequently, the functionality of a LON typical device provides a variety of application programming techniques: detection of predefined events in hardware (e.g. message arrival), concept of network variables, event-driven task scheduling, authentication, client/server support, etc. For the more detailed description of engineering solutions used in LON technology, please see references, e.g. [2], [5], [10]. Summing up, the character of the LonWorks networked systems is more *intensive*, than *extensive*. The latter is represented by dense wireless sensor networks that reveal an innovative approach in last years.

The paper addresses the issue of how to optimize LonWorks/EIA-709 sensor networking technology for wireless applications. In Section 2, the predictive *p*-persistent CSMA MAC protocol constituting the heart of the communication capability of LonWorks networks is analysed in detail. In Section 3, its performance evaluation based on simulation results for unicast traffic, is presented. Next, in order to highlight the robustness of the predictive CSMA to overload situations, the saturation performance for a general case load scenario including multicast transactions, is reported. In Section 4, the message services are described, and the analytical evaluation of delivery reliability is derived. The methods of effective management of energy consumption in wireless LonWorks are discussed in Section 5. Finally, the conclusions useful for the network design are drawn.

2. LonTalk media access control and transaction control

All LonWorks devices communicate with one another using LonTalk protocol, embedded in Neuron Chip microcontroller that is found in each LonWorks device. Neuron Chip, manufactured by Toshiba and Cypress, is a system-on-chip, including three 8-bit CPUs. Two of them (Network

Processor and MAC Processor) handle LonTalk protocol stack, the third one (Application Processor) executes the node application program developed by a user.

LonTalk is defined in ANSI approved standard as EIA-709.1 for communication between smart devices. It follows guidelines for open systems interconnection and provides services at all seven layers of the OSI reference model. In order to support versatility of LonWorks/EIA 709 technology, LonTalk is structured as a collection of services that may be optionally used. A strict layering organization makes easier LonTalk/IP interconnection [11,12]. Since LonTalk protocol does not narrow the communication only to media with the propagation characteristics of wire, the entire LON network or some subsystems may use wireless communication.

On the MAC sublayer of LonTalk/EIA-709.1 protocol, the *predictive p-persistent CSMA* scheme is used. Like all CSMA-based protocols, the predictive *p*-CSMA belongs to random access schemes basing on contention. Rather than reserve the bandwidth, nodes compete for a shared channel, resulting in probabilistic coordination. When a node wins a competition, it transmits at the full channel bandwidth. Neither a priori coordination among the nodes, nor clocks synchronization is required. The nodes can join, and leave the network without extra operations (e.g. system reconfiguration). Unlike simple *p*-CSMA, the predictive *p*-CSMA has been developed for networked sensor/control applications.

2.1. Predictive *p*-CSMA versus the persistent CSMA

MAC protocols for sensor networks have to be able to support variable and highly correlated traffic. Unlike in data networks, this traffic is generated not by human activity, but is determined by the dynamics of physical processes. Thus, the traditional CSMA-based schemes are without modifications inappropriate for sensor networks because they make a fundamental assumption of stochastically distributed traffic. Meanwhile, the traffic in sensor networking is extremely bursty, especially in case of the event-triggered application architecture that LonWorks technology exemplifies. The *event-triggered architecture* is prone to *event showers*, i.e. burst of events, often released by a single physical event that causes congestion of the system [10].

Therefore, in order to cope with overload situations, the predictive *p*-persistent CSMA has been designed as the adaptive version of pure *p*-CSMA protocol. In the *p*-persistent CSMA, introduced by Kleinrock and Tobagi in 1975, a node transmits with a fixed probability *p* if the channel is idle, and defers a transmission with probability (1-*p*), when the channel is busy [13]. In the predictive *p*-CSMA, the probability *p* is variable and dynamically adjusted to the expected traffic load. The latter is evaluated on a basis of the *derivative traffic* (i.e. traffic of acknowledgements and retransmissions). The prediction built in the CSMA protocol realizes *collision avoidance*, a basic task of media access control in sensor networks. It is based on backoff scheme that has been studied since the beginning of the 70s and successfully applied in data networks, e.g. Ethernet and WLANs. Backoff reduces contention, which helps to desynchronize the traffic. The predictive *p*-CSMA uses the *memoryless backoff* in contrast to IEEE 802.11, where backoff has memory, i.e. new backoff depends on the previous one. Thus, the predictive *p*-CSMA exemplifies a class of MAC protocols derived from traditional CSMA-based schemes for data networks and improved for sensor/control networking applications.

2.2. LonTalk/EIA-709.1 packet cycle

LonTalk/EIA-709.1 media access control packet cycle consists of two phases (Fig. 1). A type of *reservation protocol* is employed during the first phase and the *predictive p-CSMA* is used during the other one. Access to the LonTalk/EIA-709.1 channel is limited to the specified instants called *slots*.

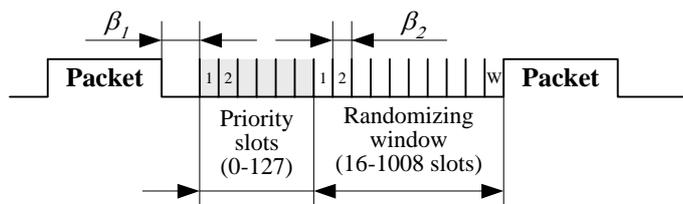


Figure 1. LonTalk/EIA-709.1 packet cycle.

In the LonTalk MAC sublayer the *implicit* reservation is applied since neither the reservation subchannel is established, nor reservation packets are used to decide, which node can transmit. The reservation phase in the LonTalk packet cycle is optional and dedicated to priority messages.

2.3. Priority slots in LonTalk packet cycle

Some applications in sensor networking require the guaranteed delivery. Priority slots follow immediately the interpacket gap of β_1 duration and are dedicated to improve the response time of critical packets. The number of priority slots per channel ranges from 0 to 127 (Fig. 1) and is configurable. Lower priority numbers indicate higher priority levels. The priority slots are in general uniquely assigned to particular nodes. Because there is no contention for the channel during the priority portion of a packet cycle, the priority transactions have a better response time than non-priority ones. In particular, a use of the same priority slot for all the nodes in a channel enables the implementation of the *multimaster* architecture for special purposes in LonWorks network (e.g. network management, polling, etc.), where the single master cycles around and sends messages to receivers [1]. This centralized architecture might be then placed on the top of the background peer-to-peer activity. However, priority slots should be used carefully and for special purposes in order to avoid channel monopolization by privileged node(s).

2.4. Predictive p-persistent CSMA algorithm specification

The predictive *p*-CSMA, like some others CSMA algorithms, splits the time axis into segments, called *slots* of β_2 duration. All nodes are forced to start transmission only at the beginning of a slot. When two packets conflict, they will overlap completely rather than partially, which greatly reduces the probability of collision and provides an increase of channel efficiency. This idea in any case goes back to the middle of the 70s when slotted-ALOHA protocol, doubling pure ALOHA channels throughput, was invented.

The algorithm of *Carrier Sense Multiple Access* (CSMA) is presented in Fig. 2. Let us assume that there are no priority slots. A node attempting to transmit monitors the state of the channel. If the channel is busy the node continues sensing, but when it detects no transmission during the β_1 period, the node delays a random number of slots of β_2 duration. If the channel is still idle when the random delay expires (i.e. no other nodes are currently transmitting), the node transmits. Otherwise, the node receives the incoming packet and competes for the channel again.

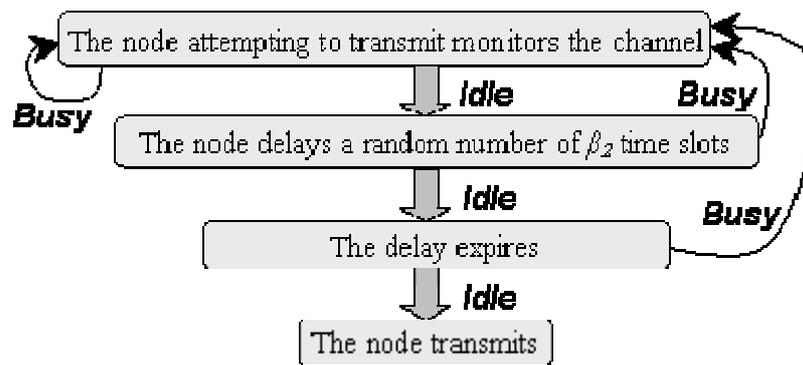


Figure 2. Carrier Sense Multiple Access (CSMA) algorithm.

If more than one node chooses the same slot, and where that slot is the earliest one selected by all the contending nodes, then a *collision* happens. All the packets involved in a collision are corrupted. The time by which the competing node defers the transmission is expressed as a pseudorandom number of time slots β_2 drawn from the uniform distribution between 0 and W , where W is the *size of the randomizing (contention) window*.

The size of the contention window is dynamically adjusted to the *expected channel load*. When the channel is idle, the window consists of 16 time slots. When the expected channel load increases, the number of slots grows by factor BL , called the *estimated backlog*. The backlog BL ranges from 1 to 63 so the size of the randomizing window varies from 16 to 1008 time slots, because:

$$W = BL \cdot W_{base} \quad (1)$$

where W_{base} is the size of the *basic* randomizing window (16 slots).

Thus, the persistence of predictive p -CSMA has either the upper ($p_{max} = 1/16 = 0.0625$), or the lower bound ($p_{min} = 1/1008 = 0.0009$).

2.5. Backlog concept

Each node calculates the backlog autonomously based on the backlog counter implemented in LonWorks node firmware. The channel backlog estimation is based on the calculation of the number of packets expected in a competition for the channel during the next packet cycle. The prediction takes into account only the *derivative traffic* (i.e. traffic of acknowledgements and sometimes also retransmissions). The current value of the backlog counter BL varies from one to the next packet cycle and relies on the accumulation of consecutive backlog increments and decrements.

The backlog counter is modified according to the following rules:

- BL is incremented by Δ_{BL} as a result of sending or receiving the message,

- BL is decremented at the end of each packet cycle,
- BL is incremented by *one* if the optional collision detection is enabled.

Δ_{BL} is the 6-bit long data field in the packet Link Layer header (Fig. 3), which expresses the number of acknowledgements generated by recipient(s) as a result of a message reception. Thus, $\Delta_{BL} = 1$ for unicast messages and $1 < \Delta_{BL} \leq 63$ for multicast transactions since the maximum size of a group equals 63. In particular, if a message is sent using unacknowledged service, then $\Delta_{BL} = 0$ is set at the Link Layer header.

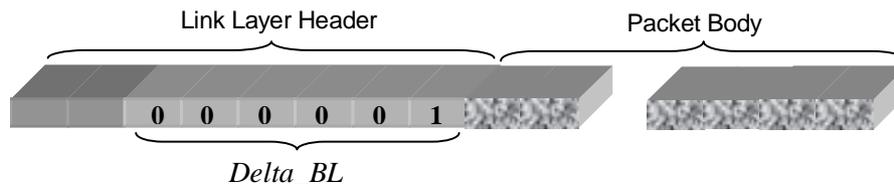


Figure 3. Δ_{BL} is the 6-bit long data field in the 8-bit Link Layer header.

$\Delta_{BL} = 1$ is set in the Figure, which corresponds to the unicast acknowledged message.

Backlog counter is decremented either during busy packet cycles, or idle cycles (i.e. if no node occupies the channel and all the slots go by without activity in the channel). Either the backlog increments based on Δ_{BL} , or decrements at the end of each packet cycle, are calculated by built-in Neuron Chip firmware.

In order to increment the backlog counter in case of collisions, dedicated hardware in the transceiver, is needed. If a collision is detected by the transceiver, then Neuron Chip is signaled using a particular pin in the communication port [2].

Note that the traffic of original messages (i.e. messages that initiate transactions) is not included in the backlog estimation. If original messages present a significant percentage of the total traffic in the channel (i.e. many transactions are unacknowledged), the protocol prediction becomes less powerful.

All the nodes in the network modify the backlog in the same way if the packets are not corrupted during the transmission, that is, if the channel is noise-free, and collision-free. Otherwise, backlog states can differ between the nodes at the same time since the recipients cannot read Δ_{BL} from packets with invalid CRC.

Strictly speaking, only the idealized channel ensures the backlog consistency. The inconsistency in backlog counting is the undesirable effect since it causes unfair access to the channel between contending nodes. However, the real channel behavior is close to idealized channel operation if the collision rate is low, that is, the number of contending nodes is small.

2.6. Collision detection problem

The predictive p -persistent CSMA might be equipped with optional collision detection in wired sensing systems. In wireless networks, detection of collisions presents nowadays a difficult problem and is not commercially available. Because the transmitted signal is always much stronger than any received signal, a transmitting node cannot efficiently distinguish incoming weak signals from its own transmission. Thus, a node is not able to transmit and receive at the same time. However, the most

recent achievement, reported by Whitehouse *et al.* [16], shows that collision detection and recovery for wireless networks is possible using the *capture effect*, i.e. the ability of some radios to receive a signal from one transmitter despite interference from another transmitter, even if relative strengths of two signals are comparable.

If the collision detection is absent, the prediction implemented in the LonTalk MAC protocol is less powerful, because the negative feedback between the collision rate and the size of a competition window is deactivated. Instead, the size of a window is tuned only by the evaluation of the expected traffic of acknowledgements.

Next, if Physical Layer does not perform collision detection, then Transport Layer of LonTalk/EIA-709.1 has to detect the unsuccessful transmissions. Transport Layer deduces collision occurrences implicitly on a basis of a lack of the *positive acknowledgement(s)* from the recipient(s) within a fixed period of time. Then, a node assumes that the transmitted packet has been lost and retries. Transport Layer reliability mechanism is cheaper than Physical Layer hardware-based detection but slower, since time-out technique is used.

2.6. Transaction control

In LonWorks networks, the messages can be sent to a *single* receiver (*unicast*), a *group* of receiving nodes (*multicast*), and to all the nodes within a *domain*, or *subnet* (*broadcast*). In case of multicast, a sender has not to be a member of a group being addressed. Multicast/broadcast transactions save communication bandwidth since a single multicast message substitutes a set of unicast messages addressed to each recipient individually.

LonTalk/EIA-709.1 provides the following message services: acknowledged service, unacknowledged service, unacknowledged repeated service, request/response service.

The *acknowledged service* (ACK) is a default type of the message service in CSMA-based networks. As a response to acknowledged message reception, each receiver generates a *positive acknowledgement* to the sender. If acknowledgement(s) are not all received, the sender times out and retries the transaction. The number of retries and the time-out are configurable. The acknowledged transaction is completed if all the expected acknowledgements are successfully received by a sender, or the timer controlling the transaction time expires. From the communication point of view the *request/response service* (REQ-RESP) is similar to the acknowledged service. The only difference is that a response unlike an acknowledgement can carry application data.

In *unacknowledged repeated service* (UNACK_RPT), acknowledgements are not applicable. The acknowledged repeated transaction is completed after sending all message repetitions. The particular case of the unacknowledged repeated service is the simple *unacknowledged service* (UNACK), where repetitions of the message are not sent.

The unacknowledged service conserves more energy than the acknowledged one because the latter generates *control packets overhead* in the form of acknowledgements. This feature can be important in wireless applications.

The multicast/broadcast transactions are quite frequent in sensor networks. In case of multicast to large groups or broadcast transactions, the unacknowledged service is paradoxically more reliable. It is

because “acknowledgement shower” is triggered by all the recipients as a response to the acknowledged message reception in the latter, and the network congestion might happen.

2.7. Physical Layer properties

In order not to limit LON networks to any particular communications medium, Physical Layer is not specified in LonTalk/EIA-709.1 protocol, but is defined by any number of transceivers that are connected to Neuron Chips. Thus, the transceiver manufacturers decide about the data encoding, channel bit rate, the maximum number of nodes per segment, the network range, etc. Multiple Physical Layer protocols and data encoding are used in LON networks. For wireless transmission, the FSK modulation is used.

A variety of transceivers for wireless communication are available on the market. Several versions operate in 400-470 MHz and 900 MHz bands. For example, Control Network Solutions Ltd provides low power LON radio transceivers with the range up to 200 meters at 418 MHz/433 MHz, and with 19.5 kbit/s bit rate. For applications needing high power radio, Motorola Remote LonWorks Telemetry Units (RTLUs) radio offers 4800 bit/s, and the range up to 15 km depending on antenna heights and gains.

3. Performance evaluation of LonTalk MAC protocol

The predictive CSMA protocol determines the most important performance measures of LonWorks systems, i.e. the channel throughput and the channel access delay. The main objective of LON network design is to optimize the network performance. The present section is devoted to the performance evaluation of the MAC protocol, and the reliability of message delivery.

3.1. Definitions of performance measures

Intuitively, the basic measures evaluating the random access protocol performance are the probabilities that a certain packet transmission is successful $p_{succ}(n)$, or unsuccessful $p_{coll}(n)$ when a number of n nodes try to transmit. Both probabilities are the relative frequencies that the successful/unsuccessful transmission occurs. However, the more common and more useful metrics for the network performance evaluation are the channel *throughput* and *collision rate*, respectively.

The *throughput* is defined as the fraction of time used for successful transmissions of packets in a channel [14], and might be expressed as a percentage of the *channel bit rate* (sometimes such a definition is assigned to the *channel utilization*, and the throughput is measured in bits or bytes per second). The *collision rate* is defined as the fraction of time during that transmitted packets are involved in collisions [14].

The difference between both pairs of measures, that is, probabilities $p_{succ}(n)$, $p_{coll}(n)$ and throughput/collision rate, consists in that the former describes *which fraction of packet cycles* is successful/unsuccessful, and the latter specifies *which fraction of channel bandwidth* is dedicated to successful/unsuccessful transmissions. Therefore, the definitions of throughput and collision rate take into account distinct lengths of successful and unsuccessful packet cycles (as follows from the probability calculus, in average the latter is a bit longer than the former) and a fraction of bandwidth,

wasted during the channel competition (i.e. interpacket gap β_1 and the number of time slots β_2 before the start of a transmission).

The other principal measure of sensor networks is a delay in accessing the channel. Due to randomization, the transmission cannot start immediately in the persistent CSMA scheme, even if a channel is idle. Since the maximum access delay in CSMA is generally unbounded, the *mean access delay* is chosen usually to evaluate the network access latency. The *mean access delay* defined as the average time from the instant the node starts trying to send a packet until the beginning of its successful transmission [14]. The channel access delay consists of the following components:

- deferring transmission, when the channel is busy (as detected by carrier sense hardware),
- delaying transmission by the fixed interval called *interpacket gap* (β_1) following a transmission on the network to make it possible for all the nodes to sense an idle channel,
- deferring transmission for the random delay (from 0 to $1007 \beta_2$ time slots) to reduce the probability of packet collision,
- if a packet is involved in one or more collisions, it may be deferred before any of the transmission attempts.

3.2. Simulation model of LonWorks network

Now we present the key characteristics of the predictive p -CSMA protocol, that is, the channel throughput, the collision rate, and the mean access delay against the offered traffic for the selected number of nodes. The reported results have been obtained by the simulation of LON network. The following simulation model has been presupposed.

The traffic source consists of a certain number of nodes which collectively form an independent Poisson process with the mean packet generation rate of λ packets/s. The total network traffic load is divided between an appropriate number of nodes so each node is loaded with the same input traffic. Since we use the wide range of traffic rate, such a traffic model encompasses either the stochastically distributed light load (if the interarrival times are long), or the congestion (when the interarrival times are very short). The latter corresponds to a huge long-lasting burst of events.

All the nodes are provided with infinite output buffer lengths. The channel is noise-free, i.e. Physical Layer does not introduce errors. The network consists of a single segment that by definition does not contain store and forward repeaters. The simulation is performed for 12 byte packet length, and the channel bit rate 78 kb/s. The preamble and End of Message (EOM) code violation at the end of each packet is not considered. β_1 is set to 4 bits duration time, and β_2 , to 2 bits, respectively. Network Layer throughput taking into account the waste of bandwidth due to collisions is calculated. Note that such a throughput definition takes into account the successful transmission of all the packets (i.e. not only messages, but also acknowledgements and retransmissions of corrupted packets).

3.3. Simulation results for unicast transactions

The simulation results of the predictive p -CSMA performance measures are presented first for a load scenario where all the transactions are unicast regardless of message service. We denote this scenario as the *load scenario 1*.

If none of the packets is addressed using multicast addressing and the acknowledged service in the channel with no collision detection, the predictive part of the algorithm is deactivated. It is because the resultant change of backlog counter is never greater than zero in each unicast packet cycle: a possible increment by one during an acknowledged message transmission is cancelled by periodical backlog decrement at each packet cycle (see the backlog counting rules in Section 2.5). Thus, a load scenario using exclusively unicast transactions reduces the predictive p -CSMA to a pure p -persistent algorithm, where $p=1/16=0.0625$, because a contention window is constant and consists of 16 slots. Moreover, under light traffic load the predictive CSMA operates closely to 0.0625-persistent CSMA regardless of the message service type: even occasional multicast messages do not increase considerably the long-term mean randomizing window over 16 slots.

Figures 4, 5 and 6 show simulation results of the channel throughput, collision rate, and the mean access delay for unicast transactions versus the offered load for different number of nodes.

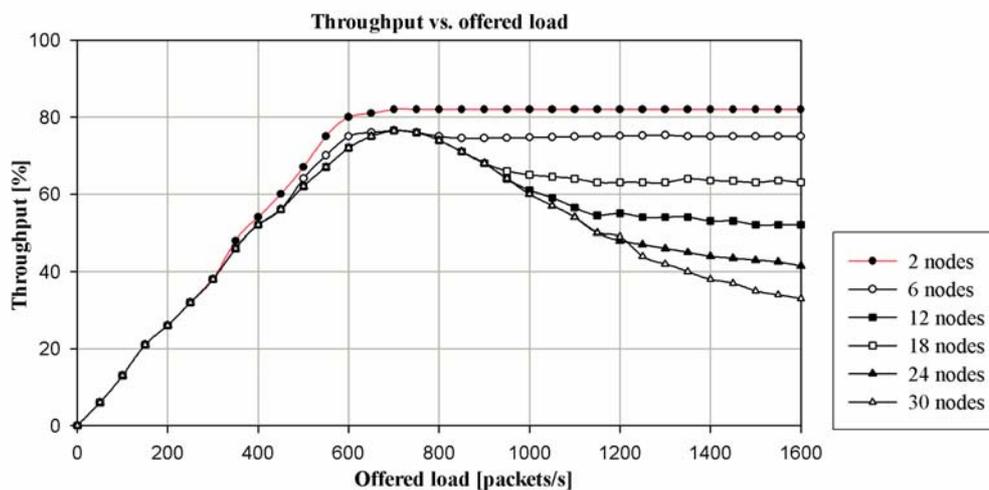


Figure 4. Throughput versus offered load for different number of nodes (load scenario 1).

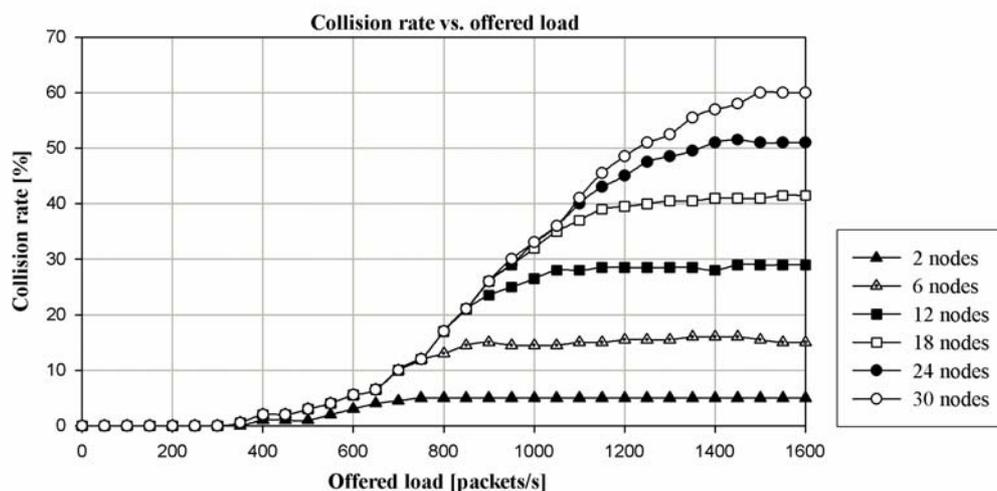


Figure 5. Collision rate versus offered load for different number of nodes (load scenario 1).

Three types of the relationship between the performance measures of the predictive p -CSMA and the offered load appear. If the channel is *lightly* loaded, the throughput *grows* with the offered load. Next, if the input traffic increases after achieving its maximum value, the throughput starts to *decrease* for a *medium* load range. Finally, the throughput *remains* constant regardless of the traffic rate if the channel is *heavily* loaded. The load range, when the throughput does not increase with the growing offered load is called *congestion region*.

According to this classification, the network stays under light traffic load if *only one* node competes for the channel in the average packet cycle. Then, the channel is able to transport successfully almost all the messages sent by transmitting nodes, and the channel throughput against the traffic load can be approximated by the linear function. The channel throughput reaches its maximum for a "knee" point, at which the offered load approximately matches the channel bit rate (78 kb/s). The further increase of the offered load causes the throughput to decay due to frequent collisions since *more than one* node has the message to send in the average packet cycle. We call it as the *medium load range*. Finally, the network reaches the *saturation* if output queues in *all* the nodes are occupied. It corresponds to heavy steady-state load condition when the throughput and the collision rate reach its minimum and maximum, respectively. At all loads above this point the throughput/collision rate remains constant, only the packet queues in nodes become longer. This is the range of the *worst-case protocol performance*.

As expected, the greater number of transmitting nodes, the greater probability of collision and, as a result, the lower realized throughput. For example, the throughput is less than 45% for 24 transmitting nodes and less than 35% for 30 nodes (Fig. 4). Thus, the network performance is not satisfactory for a large number of nodes in case of overload.

Fig. 6 shows the mean access delay versus the offered load. For light offered load the mean access delay remains low and nearly constant. If the network begins to saturate, the mean access delay rapidly grows since packets have to wait in a queue to get the channel.

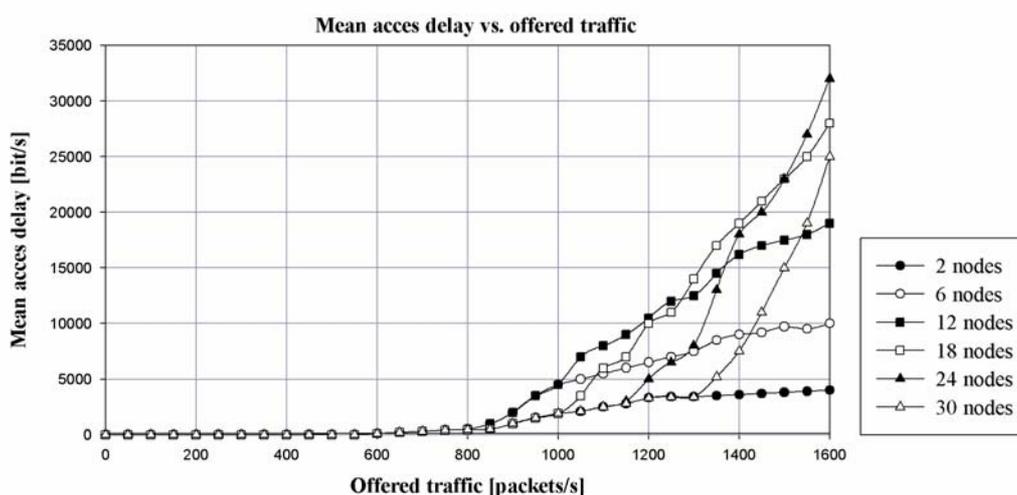


Figure 6. Mean access delay versus offered load for different number of nodes (load scenario 1).

The shape of the predictive CSMA characteristics for the *general case load* scenario resembles the shape of appropriate measures for unicast traffic. The difference between them is more quantitative

than qualitative, and in fact consists in distinct protocol behaviour for multicast transactions and heavy load. Under light traffic load, as it was stated, the predictive CSMA approaches 0.0625-persistent CSMA regardless of the message service. Thus, in order to recognize complete protocol characteristics for a particular load scenario, it is enough to evaluate its performance for the network *saturation* conditions.

3.4. Saturation status performance

Saturation status of the network, defined as the situation when all the nodes stay under heavy traffic and have packets to send, constitutes the critical point of the predictive *p*-CSMA characteristics. The saturation condition represents the largest possible load offered to the network by a given number of nodes, and the worst-case network performance versus the offered traffic. As it was mentioned, the event-triggered application architecture is prone to *event shower*, i.e. traffic including a lot of redundant event information often triggered by a single physical event. Such a situation can occur for example when a fire is detected in a building and a number of redundant temperature and smoke sensors begin reporting the event. The load in event shower situation can be close to the saturation conditions.

3.5. Load scenario case study

Let us consider the general case load scenario, where any combination of the message service (unacknowledged/acknowledged) and transaction types (unicast/multicast) distributed in any proportion among the input traffic is possible. We denote this scenario as the *load scenario 2*. Suppose the *input traffic* (i.e. traffic of original messages) generated into the network by a number of n nodes consists of the following parts:

- 20% of messages are unacknowledged,
- 30% of messages are the unicast using the acknowledged service,
- 30% of messages are the acknowledged multicast to 2 recipients,
- 20% of messages are the acknowledged multicast to 3 recipients.

The acknowledged messages trigger off acknowledgements. A successfully transmitted unicast message generates a single acknowledgement, and a multicast one, multiple acknowledgements, respectively. The total traffic in the channel is a *superposition* of the traffic of original messages and acknowledgements.

In the idealized channel (i.e. noise-free and collision-free) without collision detection, the size of the competition window does *not* depend on the number of contending nodes, but on the *structure of the input traffic* generated to the network, that is, the proportion between unicast/multicast transactions and unacknowledged/acknowledged message service. The maximum probability is assigned to the minimum size of the window (16 slots), and the probabilities for higher backlog values are lower. For the saturation conditions the probabilities decrease exponentially [7]. It is visible as linear plot for the logarithmic scale of a vertical axis (Fig. 7). However, the distribution shown in Fig. 7 approximates well the real network behavior for small number of contending nodes, since the real channel operates closely to the idealized channel if the collision rate is low (see Section 2.5). To be precise, the real

distribution of the competition window approaches in saturation status the exponential distribution asymptotically for low collision rate.

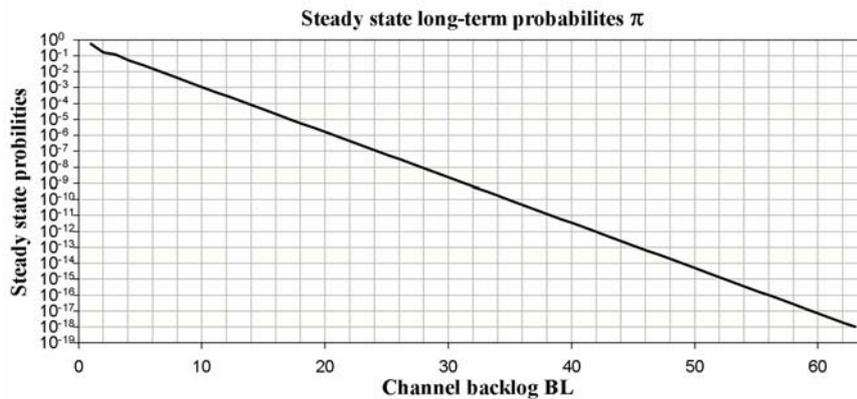


Figure 7. Distribution of steady-state long-term probabilities for backlog stages 1, ..., 63 in the idealized channel (noise-free and collision-free) for saturation conditions (load scenario 2). The logarithmic scale is set in the vertical axis.

Table 1 (a, b) shows simulation results of the saturation performance measures of the predictive p -CSMA for unicast traffic (load scenario 1), and for a particular case study of channel load (load scenario 2). The measures $p'_{succ}(n)$ and $p'_{coll}(n)$ represent which fraction of transmission attempts is successful or unsuccessful, respectively. Note that $p'_{succ}(n)$ and $p'_{coll}(n)$ that are experimentally evaluated, approach the probabilities of successful $p_{succ}(n)$ and unsuccessful $p_{coll}(n)$ transmission if the number of observed packet cycles is very large.

Table 1. Saturation performance of predictive p -CSMA for the scenario 1 (a) and the scenario 2 (b).

(a)				
n	p_{succ} [%]	p_{coll} [%]	Throughput [%]	Collision rate [%]
2	93,89	6,11	81	5,3
4	88,08	11,92	79	11
8	86,12	23,88	70	22
10	71,13	28,87	66	27
20	48,32	51,68	45	48

(b)				
n	p_{succ} [%]	p_{coll} [%]	Throughput [%]	Collision rate [%]
2	94,82	5,18	79	4,3
4	89,93	10,07	78	8,8
8	81,05	18,95	73	17
10	75,89	24,11	69	22
20	57,78	42,22	54	39

The appropriate formulas for calculation of $p_{succ}(n)$ and $p_{coll}(n)$ can be found in [9]. Unlike for $p'_{succ}(n)$ and $p'_{coll}(n)$, the sum of throughput and collision rate is not necessarily equal to one.

However, $p'_{succ}(n)$ and $p'_{coll}(n)$ can be used as the approximation for a simplified evaluation of the throughput and collision rate if the number of transmitting nodes is large.

The results for both load scenarios are similar for low number of contending nodes. However, the network is more robust in case of congestion if the load scenario includes multicast transactions (see Table 1b). For a network consisting of 20 nodes, the saturation throughput for pure unicast messages (load scenario 1) is nearly 20% lower than for the load scenario 2. The presented results exemplify the efficiency of expected load evaluation embedded in backlog counting. The expected load in the load scenario 1 is evaluated to zero because of the lack of acknowledgements. On the other hand, the expected load in the load scenario 2 is a significant part of the total channel load since acknowledgements make up 60% of all the packets in the channel.

The other observation relates to the upper bound on the achievable throughput. Note that even if the percentage of collision is low (e.g. 5%), the throughput equals to about 80% only. It is because for a lightly loaded channel the randomization contributes more to the waste of channel bandwidth than collisions.

3.6. End-to-end reliability of message delivery

In CSMA-based schemes without collision resolution there is no guarantee for successful message delivery. Therefore, *message retransmissions* are used to improve transaction reliability. By default, the number of retries should be large enough to ensure that message delivery is successfully completed with desired probability, e.g. greater than 95%. Let us evaluate the reliability of message delivery in particular message services.

3.6.1. UNACK/UNACK_RPT service reliability

The number of retries in UNACK_RPT service is selectable and constant in each transaction even if earlier attempts succeed. Assume that the number of m repetitions is configured. Since the probability that all $(m+1)$ transmission attempts (i.e. an initial transmission and follow-on m repetitions) are involved in collisions equals to $(1-p_{succ})^{m+1}$, then the probability $p_{succ}^{(m)}$ that at least one of the transmission attempts using UNACK_RPT service is successful equals:

$$p_{succ}^{(m)} = 1 - (1 - p_{succ})^{m+1}, \quad (2)$$

where p_{succ} is the probability of a successful transmission in a single attempt.

UNACK service is modelled by the equation (2) for $m = 0$, that is, if there are no repetitions of an initial transmission attempt.

For example, if $p_{succ} = 0.7$, then the probability $p_{succ}^{(2)}$ that the UNACK_RPT transaction succeeds in 3 attempts equals 0.9997. Typically, the probability that UNACK_RPT transaction fails with 3 delivery attempts amounts to a fraction of one percent even in the overloaded network.

3.7.2. ACK/REQ-RESP service reliability

Using ACK or REQ-RESP service, the delivery attempts are repeated until all the acknowledgement(s) are received by a sender. The maximum number of retries is selectable. Usually from 2 to 5 retries are configured. Assume the probability that a single attempt succeeds is p_{succ} and is

constant during the transaction time. The probability $p_{succ}^{(m)}$ that a message delivery within a group of i members succeeds at most in $(m+1)$ attempts (i.e. an initial transmission and follow-on m retries), is described by a *geometric distribution* and expressed by the following formula:

$$p_{succ}^{(m)} = p_{succ}^i \sum_{l=0}^m (1 - p_{succ})^l (p_{succ})^{m-l+1} \quad (3)$$

Note that the formula (3) describes for $i=1$ the reliability of unicast transactions.

4. Optimization of energy consumption in wireless LON

The well-known disadvantage of the contention-based MAC protocols, which the predictive CSMA exemplifies, is ineffective energy management. It is because nodes listen all the time the channel, receive unnecessary traffic, including control packets and collide due to the lack of coordination in accessing the network. The major sources of energy waste are: collisions, overhearing, control packet overhead, idle listening [17,18,19]. As it was stated the energy consumption is not a primary objective of LonWorks platform design since the batteries powering devices are in general replaceable in contrast to dense wireless sensor networks. However, it is worth to outline the main techniques of energy management optimization in the wireless LON systems.

4.1. Collision avoidance

Collisions are the inherent phenomenon associated with the CSMA-based family and it is impossible to avoid them completely. Energy consumption is increased due to retransmissions of corrupted packets. Collision is the first source of energy waste in the wireless sensor networks. Collisions are frequent if many nodes share a common channel in a network segment (i.e. subsystem that does not contain store-and-forward routers). A typical LonWorks segment consists of up to 64 nodes. All the packets sent by a particular node in a segment are 'visible' by the other nodes within this segment. In order to reduce the number of collisions without decreasing the input traffic, LON network has to be split into multiple segments. LON technology offers intelligent routers like *configured routers* or *learning routers* to set appropriate segments [3]. The effective way of collision minimization is also to reduce traffic by using *send-on-delta concept* instead of *periodic updates* [8].

4.2. Overhearing minimization

Overhearing consists in that a node picks up packets that are destined for other nodes. In the CSMA-based protocols any packet sent by a transmitting node is received by all the nodes in the segment, even though only a single node or a set of nodes are the intended receivers. It is because each node has to "unpack" the received packet in order to check its address. Overhearing is a significant source of energy waste when the traffic in the channel is heavy and the number of nodes in the network is large. Thus, the same solution as suggested for collision avoidance is proposed for overhearing minimization. This is a network segmentation using configured routers or learning routers that can decrease the number of traffic sources sharing a common channel and the total load in each network segment.

4.3. Control packet overhead

Control packet overhead results in sending and receiving control packets which consume energy, but do not carry application data. The only control packets used in LonWorks networks are *acknowledgements*. Since collisions are not detected in wireless transmission, the *acknowledged service* is a default type of reliable operation for LON-based wireless networks. To request an acknowledgement is the only way the sender can make sure that a message has been successfully received by recipient(s). As it was stated, the multicast acknowledged service activates the collision avoidance of the predictive *p*-CSMA MAC protocol. However, the prediction is designed to deal with the traffic of acknowledgements itself, and is not intended to predict the traffic of original messages.

Except the energy waste problem, the traffic of acknowledgements increases the collision rate and the mean access delay. Thus, in fact acknowledgements degrade the *global reliability* of packet delivery. Both problems escalate if broadcast addressing is used since the effect of “acknowledgement shower” occurs. Summing up, the control packet overhead is a price for back information about a result of a particular transaction completion, and acknowledgements pose a fundamental tradeoff between *end-to-end* and *global reliability*.

4.4. Reduction idle listening in sleep mode

Idle listening happens when the node is sensing the channel to receive possible data. Idle listening contributes significantly to waste of energy when there is no data to send during a long time.

The basic technique to improve energy consumption due to idle listening is to put the node into *sleep mode* if it is not engaged in data processing. LonWorks technology provides the programming instructions that allow a node to go to sleep and to wake up if specified events occur. Each node sleeps autonomously calling `sleep()` function [5]. The node is usually put to sleep mode in two-step process. In the first step all outstanding outgoing and incoming messages are flushed. In the second step the node goes to sleep when the flush completes. The node can wake up when some activity in local environment (on I/O port), or in remote environment (on the communication channel) is detected. However, the node can be configured to ignore incoming messages, e.g. when a sensor node is programmed as a source of messages only, and is not interested in message receiving. Use of the sleep mode is especially efficient if the node does not interact with other nodes and only reports rare events in its local environment.

5. Conclusions

In the paper, a discussion on how to optimize LonWorks/EIA-709 sensor networking technology for wireless applications, is presented. The typical wireless LonWorks applications are listed. Since the main objective of LON network design is to optimize the network performance rather than to minimize the energy consumption, the MAC protocol and transaction control used in LON networks is analysed in detail. As follows from the analysis of CSMA-based LonTalk MAC protocol performance, collision avoidance is activated only by multicast acknowledged transaction. For unicast load scenario, the predictive *p*-CSMA reduces to pure 0.0625-persistent CSMA.

The acknowledged service improves the *end-to-end reliability*, but degrades the *global reliability* at the same time since introduces the control packet overhead. As a matter of fact, the acknowledged service activates the collision avoidance mechanism, however, this mechanism is designed to deal with the traffic of acknowledgements itself, and is not intended to predict the traffic of original messages. Summing up, the only benefit of using the acknowledged service is the possibility to notify the sender that a particular transaction succeeds. It is not possible to do it in case of the unacknowledged service even if the probability of the successful completion of the unacknowledged (repeated) transaction is greater. However, in general the acknowledged service is a default type of message service in random access networks. The wireless LON channel should not be overloaded in a normal operation with the mean number of transmitting sources not exceeding a few nodes.

Finally, LON technology is based on the approach different from that presented in dense wireless sensor networking. However, both approaches supplement each other in a complementary way and are dedicated to different applications.

Acknowledgements

The authors would to thank Teresa Krajewska for manuscript revision.

References and Notes

1. *LonTalk Protocol Specification*, Version 3.0; Echelon Corporation 1995.
2. *LonWorks Technology Device Data*, Rev. 4; Motorola 1997.
3. *LonWorks Router User's Guide*, Rev. 3; Echelon Corporation, 1995.
<http://www.echelon.com/support/documentation/manuals/routers/078-0018-01D.pdf>
4. *Layer 7 LonMark Interoperability Guidelines*, Ver. 3.2; LonMark Interoperability Association, 2002. <http://www.lonmark.com/press/download/LYR732.pdf>
5. *Neuron C Reference Guide*; Echelon Corporation, 1995.
<http://www.echelon.com/support/documentation/manuals/devtools/078-0140-02E.pdf>
6. Miskowicz, M.; Sapor, M.; Latawiec W.; Zych, M. Performance analysis of predictive p -persistent CSMA protocol for control networks. *Proceedings of IEEE International Workshop on Factory Communication Systems WFCS'2002*, Vasterås, Sweden, **2002**, 249-56.
7. Miskowicz, M. Saturation performance of predictive p -CSMA without collision detection. *Proceedings of IEEE International Conference on Emerging Technologies and Factory Automation ETFA'2005*, Catania, Italy, **2005**, 93-100.
8. Miskowicz, M. Send-on-delta concept: an event-based sensor data reporting strategy. *Sensors*, Special Issue: Wireless Sensor Networks and Platforms, **2006**, 6, 49-63.
9. Buchholz, P.; Plonnigs, J. Analytical analysis of access-schemes of the CSMA type. *Proceedings of IEEE International Workshop on Factory Communication Systems WFCS'2004*, Wien, Austria, **2004**, 127-136.
<http://web.inf.tu-dresden.de/~jp14/publications/BuPl04WFCSAnalysis.pdf>
10. Dietrich, D.; Loy, D.; Schweinzer, H.-J. *Open control networks: LonWorks/EIA 709 technology*, Kluwer Academic Publishers: Boston, 2001.

11. Shahnasser, H.; Wang, Q. Controlling industrial devices over TCP/IP by using LonWorks. *Proceedings of IEEE GLOBECOM'1998*, Sydney, Australia, vol. 2, **1998**, 1309-14.
12. Soucek, S.; Schweinzer, H.-J. Considerations on a LonWorks/IP gateway implementation. *Proceedings of International Symposium of Fieldbus Systems and their Applications FeT'1999*, Magdeburg, Germany, **1999**, 291-298.
13. Kleinrock, L.; Tobagi, F. A. Packet switching in radio channels. I. Carrier sense multiple-access modes and their throughput-delay characteristics. *IEEE Transactions on Communications* **1975**, *23*, 1400-1416.
<http://www.cs.ucla.edu/~lk/LK/Bib/PS/paper059.pdf>
14. Yeh, J. Simulation of local computer networks - a case study. *Computer Networks* **1979**, *3*, 401-417.
15. Kopetz, H. *Real-time systems. Design principles for distributed embedded applications*. Kluwer Academic Publishers: Boston, 1997.
16. Whitehouse, K.; Woo, A.; Jiang, F.; Polastre, J.; Culler, D. Exploiting the capture effect for collision detection and recovery. *Proceedings of IEEE Workshop on Embedded Networked Sensors EmNetS'2005*. Sydney, Australia, **2005**.
<http://www.cs.berkeley.edu/~kamin/pubs/whitehouse05emnets.pdf>
17. Ye, W.; Heidemann, J.; Estrin, D. An energy-efficient MAC protocol for wireless sensor networks. *Proceedings of IEEE INFOCOM'2002*, New York, USA, **2002**, 1567-1576.
http://www.isi.edu/~weiye/pub/smac_infocom.pdf
18. Demirkol, I.; Ersoy, C.; Alagöz, F. MAC Protocols for Wireless Sensor Networks: a Survey. Accepted to *IEEE Communications Magazine* (In Press).
http://www.cmpe.boun.edu.tr/~ilker/IlkerDEMIRKOL_COMMAG.pdf
19. Ye, W.; Heidemann, J. Medium access control in wireless sensor networks. Technical Report ISI-TR-580, University of Southern California, 2003.
<http://www.isi.edu/~weiye/pub/isi-tr-580.pdf>
20. Manjeshwar, A.; Agrawal, D. P. TEEN: a routing protocol for enhanced efficiency in wireless sensor networks. *Proceedings of International Parallel Distributed Processing Symposium IPDPS'2001*, San Francisco, USA, **2001**, 189-199.
http://network.chonnam.ac.kr/seminar/Lab_Seminar/Data/00925197.pdf
21. Elson, J.; Estrin, D. An address-free architecture for dynamic sensor networks. Technical Report 00-724, University of Southern California, Computer Science Department, 2000.
<http://citeseer.ist.psu.edu/cache/papers/cs/14766/http:zSzzSzwww.circlemud.orgzSz~jelsonzSzwritingszSzafa-tr.pdf/elson00addressfree.pdf>
22. <http://www.automatedbuildings.com/news/mar01/articles/engenuity/eng.htm>.