A Token-based MAC Protocol for Linear Sensor Networks

El Hadji Malick NDOYE, Frédérique JACQUET, Michel MISSON, Ibrahima NIANG

1 Clermont Université / LIMOS CNRS - Complexe scientifique des Cézeaux, 63172 Aubière cedex, France
2 Laboratoire d’Informatique, Université Cheikh Anta Diop de Dakar (UCAD), B. P. 5005 Dakar-Fann, Sénégal
E-mail: ndoye@sancy.univ-bpclermont.fr, jacquet@sancy.univ-bpclermont.fr, misson@sancy.univ-bpclermont.fr, iniang@ucad.sn

Received: 4 June 2015 / Accepted: 26 June 2015 / Published: 30 June 2015

Abstract: A wireless sensor network is a large number of sensor nodes deployed in a fixed or random manner over a wide area for environmental monitoring applications. Wireless sensors communicate via wireless links and are powered by batteries. They collect and provide information to the base station usually called sink. The information collected is generally physical, chemical or biological nature. For some of these applications, as pipeline or road monitoring, wireless sensor nodes have to be deployed in a linear manner. We refer to these WSNs as Linear Sensor Networks (LSNs). Due to specificity of LSNs, MAC protocol designed for WSNs, as contention or TDMA based protocols, are often not suitable. Furthermore, wireless node deployment can provide a certain form of redundancy to prevent link or node failures. In this paper, we propose a token based MAC protocol for linear sensor networks in order to improve the network performance. We evaluate the effect of the redundancy on the throughput and on the end-to-end delay. Copyright © 2015 IFSA Publishing, S. L.

Keywords: Wireless sensor network, Linear topology, Throughput, MAC protocol, Token passing, End-to-end delay.

1. Introduction

In recent years, sensor networks are presented as a solution to the need to observe, to analyze and control physical phenomena on hostile wide area. This is an ad hoc network of a large number of micro-sensors which collect and transmit environmental data autonomously for example. The position of the nodes is not necessarily predetermined and can be randomly dispersed in a geographic area corresponding to the field of interest for the phenomenon being observed. Therefore, they are essential to many environmental and scientific applications and are increasingly used in the environment and the industry especially with the latest developments in the field of wireless technologies.

In some cases, the monitored infrastructures make the deployment of sensor nodes linearly on one or more lines. We see such deployment in applications of pipeline monitoring (water, gas, etc.), road monitoring, trains, railways, etc. This type of wireless network is called wireless linear sensor network [1].
Linear Sensor networks are characterized by a limited neighborhood and limited routing possibilities as shown in Fig. 1. Indeed, in a linear sensor network, each node has some neighbors at left (left neighbors) or right (right neighbors). The numbers of neighbors or redundancy are defined by the radio range of the sensor node and the distance between two adjacent nodes. A linear network is R-Redundant if a node has R left neighbors and R right neighbors. These specific characteristics of linear sensor network can have negative effects on the network performances as throughput or end-to-end delay if the access mechanism does not take them into account their specificities. For example, in the case of linear networks where a node has exactly two neighbors (on the left and on the right), a default radio link or the hidden terminal problem can lead to a dysfunction or a considerable energy consumption.

![Fig. 1. Example of LSN.](image)

Therefore it is important that the MAC protocol implements appropriate mechanisms that take into account the neighborhood between the sensor nodes, linearity and effective energy management in order to increase network lifetime and to improve its performance. MAC protocols based on contention or TDMA are not often adapted to linear networks because they do not manage the specific characteristics of this topology. Consequently, collisions generated respectively by the contention and the synchronization may be more problematic in the case of the linear sensor. In this paper, we propose a MAC protocol that takes advantage of the specific characteristics of the LSNs such as the linearity, the redundancy, etc. Our protocol is based on token which gives a node the permission to access to the transmission channel. We evaluate the protocol in terms of throughput, end-to-end delay to show the impact of redundancy factor R according to the deployment of the nodes.

The rest of the paper is outlined as follows: in Section 2 we present a state of the art on the MAC protocols used in linear networks. Section 3 gives the hypothesis and the network topology; Section 4 presents our proposal token based MAC protocol. In this section we show the principles of the token generation by explaining the role of temporal information related to the token. We study the FIFO behavior in order to introduce the clipping effect phenomena in Section 5. The in Section 6, we analyze the concept of redundancy and shows the impact of R factor on the throughput. We present our simulation results in Section 7. Finally, we end this paper with a conclusion and perspectives in Section 8.

2. State of Art

We find linear sensor networks in the pipeline monitoring applications [2-4]. In this case, sensor networks are used to detect the toxic products contained in those pipelines (gas, chemicals, radioactive elements, oil, etc.) or leakage. Another application of the linear topology wireless sensor networks is the monitoring of mines [5-6] or volcanoes [7]. In this case, the sensor nodes are particularly responsible for monitoring environmental temperature changes to detect abnormalities in a mine. LSNs are also found in the monitoring of trains, rails [8-9] for detecting a malfunction of hardware components such as wagons, the quality of railways, etc. These wireless networks with linear topology are also used in some monitoring applications of bridges or roads.

The protocols used in wireless sensor networks are often not suited to linear networks due to the fact that they do not take into account the specific characteristics of this type of network. These protocols are mainly based on the contention or on a strict synchronization. In the case of contention-based MAC protocols such as CSMA /CA with or without RTS/CTS [10], 802.15.4 [11], SMAC [12] or TMAC [13] collisions generated during the contention periods due to the hidden terminal problem can degrade the network performance. For strict synchronization based protocols such as TDMA, FLAMA, TRAMA [14] the clock drifts can cause considerable packet losses.

Sensor networks are often designed for convergecast traffic needed for applications using data aggregation at one or several collecting point. In this type of network, routing from hop to hop is an effective way to route data to the sink. However, this technique leads to an overload of the FIFO queues for the nodes closer to the sink. Indeed, each sensor node must aggregate its local production and data from its neighbors. This phenomenon is shown in [15] where the authors demonstrated, through an analytical study, the traffic accumulation of nodes closer to the sink.

These zones of accumulation also called congestion zones degrade the network performances. Some disadvantages that can be observed are:

- An overload of the FIFO of node within the congestion areas, inducing delays in the forwarding process and a risk of frame dropping due to FIFO overflow;
- An increasing of the number of busy medium status which is returned by channel sensing (CCA operation);
- An increasing of the rate of collisions due to the hidden terminal problem.

These two last phenomena cause losses of frames (collision or dropping). In the literature, this specific behavior has been identified early for solutions based on linear deployment of a wireless infrastructure made of 802.11 Access Points [16]. For WSN applications, many researches focus on the propriety...
of MAC protocol in order to avoid congestion and thus improve the throughput or end-to-end delay. In recent years, a lot of researches focus on token based protocols due to the limitation of classic MAC protocols (contention and TDMA). Nevertheless, most of these proposed protocols do not match in linear sensor network. TKN-TWN (ToKeN TWiNs) described in [17] is a high throughput data collection solution exploiting the advantages of TDMA. It eases the scheduling burden by using two tokens to arbitrate transmission activities. Access to the medium is organized by a centralized token passing mechanism. The sink node generates two types of tokens which are passed to two different top-subtrees, and a multi-channel approach is used to avoid interference. In [18] an implementation and an evaluation of TKN-TWN in term of throughput are presented. Authors show that in a binary-tree-formed network TKN-TWN outperforms the collection data protocol [19] in terms of throughput. This proposition was designed for WSNs deployed according to a tree topology and it loses a part of its interest for linear topology. In [20] authors present an Enhanced Dynamic Token Protocol (EDTP) using TPQ as described in WDTP [21]. They show that in underwater acoustic WSN, EDTP performs better throughput than TDMA. In [22] authors describe a Data Filtration-Aware MAC protocol for wireless Sensor Networks (DF-MAC). It is a token-based MAC protocol based on TR-MAC. In this contribution, nodes manage a logical ring topology and data transmission is performed by using a token. It was shown that DF-MAC provides a better throughput than TR-MAC. Unfortunately, DF-MAC is designed for clustered network and cannot be relevant for LSNs. The Token Bus Based MAC protocol (TBB-MAC) proposed in [23] aims to reduce energy waste by minimizing the amount of redundant communications and to improve end-to-end delay but is designed for clustered WSN.

Even if the token based protocols presented above improve performance of classic or clustered WSNs, they are not designed for LSNs and they do not take advantage of the specificities of such a topology. The design of a MAC protocol for a LSN must consider a trade-off between the specificities of this kind of networks: topology, low density, small processing power, energy limitations, etc.

We propose a MAC protocol based on a slotted access method needing a soft synchronization and taking advantage of linearity of the deployment of nodes. The use of a token to propagate the right to transmit is a way to avoid the constraints of a strict and global synchronization.

Before sending data frames, a current node has to wait the reception of a token from one of its neighbors. When this token reception occurs, this current node becomes Token Holder for a given amount of time SD (Shuttle Duration). Then this node can transmit uplink traffic toward the sink or downlink traffic toward the Allocator (Fig. 1.), details will be given in the next paragraph. The token frame contains information as the token generation period, the sleep and wakeup calendar. We evaluate our protocol in terms of throughput and end-to-end delay in order to show the impact of redundancy of linear network sensor on its behavior.

3. Hypothesis and Network Topology

We outline now the details about the density of LSNs we are studying, the timing of MAC protocol we propose and the traffic model we use in this paper. In the following, we consider a one-dimensional LSN of N nodes uniformly placed over a length-L with equal distance d between two adjacent nodes. The kind of LSN we are going to use is depending on the density of the topology. If the radio range p and the distance d between nodes are such that d < p < 2d, each node has two neighbors (one to its left and one to its right). We are dealing with a strictly LSN or a 1-Redundant LSN. When p > 2d, each node has at least 4 neighbors (two or more to its left and two or more to its right). For such topology, if a link is broken or a node is out of order it is still possible to forward the frames by skipping the defective device. This kind of LSN is called R-Redundant where R is the number of neighbors in each direction (right and left) as shown in Fig. 2. In the following, we focus on this kind of redundant LSNs, and we refer to them as 2-Redundant LSNs or 3-Redundant LSNs according to their density. This contribution is based on the strong following hypothesis: for an R-Redundant topology, the relation $R \cdot d < p < (R + 1) \cdot d$ is stable, that to say the length of the radio links of such a topology is constant and uniform all over the LSN.

![Fig. 2. Connectivity of an R-Redundant LSN.](image)

Each node has its own ID. In this paper we suppose the following points:
1) The node located at left end of the LSN is Node 1 and is the Allocator providing tokens,
2) The node at the right side of Node 1 is Node I+1,
3) The node located at the right end of the LSN is the Sink.

The routing (or forwarding) scheme works as follows: each node transmits data frames to its one-hop or R-hop neighbor according to the LSN topology. For many reasons such as
1) Retransmission credit exceeded (occurs when a node has transmitted five times the same data frame without receiving any acknowledgment),
2) Queue overload (occurs when the queue of the node is requested to keep too many packets), packets may not be received its destination.

We define the aggregate throughput as the number of packets received by the sink per unit of time. Throughput increases as the offered load provided by nodes (packets locally produced) increases. When the offered load reaches a certain threshold, the throughput does not increase any more. Sometimes, it can even start to decrease. If we denote by \( t \) a period of time including several token periods, if \( n \) is the number of packets reaching during \( t \), we can define the aggregate throughput of the LSN as:

\[
S = \frac{n \times \text{PacketSize}}{t}
\]  

(1)

We also define the end-to-end delay as the time between the packet creation at the source, and its successful delivery at the sink.

4. Our Proposal: Token Based MAC Protocol

4.1. Access Control

The token generation governing the access to the medium is initiated by the node of the other end opposite to the sink called Allocator. Each produced token circulates from node to node until it reaches the sink. The token contains temporal information about the transmission or waiting periods of data to or from the sink. In the following, the path followed by a token is always A, B, C, ..., Sink. Each node which receives a token is allowed to transmit during a given amount of time called shuttle duration. So, a node has two major states: it is either “token holder”, or “waiting for a token”. In token holder state, the node can transmit differently data frames during the shuttle duration, according to three time intervals.

1) \( T_1 \): during this amount of time the node transmit data frames to its neighbor toward the Allocator. This traffic is called downlink traffic and comes from the sink.

2) \( T_2 \): during this amount of time the node transmit data to the sink. This traffic is called uplink traffic.

3) At the beginning of period \( T_2 \), the node being token holder passes the token to its neighbor at its right side and reaches the waiting for a token state. When there is no pending downlink traffic, a token holder node uses also \( T_1 \) for uplink traffic. When a node is in the waiting token state, it can either listen for uplink or for downlink packets coming from the sink. After that, this state can be used by the node to set its radio stage to sleeping mode to save energy. The temporal pattern of the activity of a current node is given on Fig. 3.

![Fig. 3. Temporal pattern of token holder node.](image)

4.2. Periodicity of the Token Production

In this section, we define the minimal distance in term of nodes between two consecutive tokens in order to have several tokens circulating in the network at the same time. This distance must be calculated according to the transmission range of each node and on the possibility of having downlink traffic. In Fig. 4(a), it is shown that in a strictly LSN without downlink traffic, the distance between two token holders is two nodes (three hops). In the case of strictly LSN with downlink traffic as shown in Fig. 4(b), the distance is set to three nodes (4 hops). So, in the case of strictly LSN the size of the cluster (CluSize) is set to three hops without downlink traffic and four hops with downlink traffic.

So if a reverse traffic is possible, a node is token holder at most only a quarter of its time. During three quarters of its time, it has to queue the traffic locally produced. During the remaining quarter of its time, the node has to queue (in an interleaved manner) the local traffic in addition to the traffic forwarded by the previous node of the LSN. If the radio range allows the possibility to exchange with its two-hop neighbors, the spatial reuse becomes less efficient, as it requires an increase in the distance between two nodes being simultaneously in the token holding state.

As shown in Fig. 4(c) and Fig. 4(d), for a 2-Redundant LSN, two token holder nodes have to be separated by at least four nodes when the traffic is only for the sink, and by six nodes when a reverse traffic is possible. The size of the linear cluster (CluSize) is respectively 4 and 7. The minimal distance between two successive token holder nodes has a strong impact on the network performance and on the token production activity of the allocator. For a given token holding time \( (T_1 + T_2 + T_3) \) as defined in Fig. 3), the minimal period of token production is given by:
\[ T_{\text{Token}} \text{ (min)} = \text{ShuDur} \times \text{CluSize}. \]

(a) 1-Redundant LSN without downlink traffic

(b) 1-Redundant LSN with downlink traffic

(c) 2-Redundant LSN without downlink traffic

(d) 2-Redundant LSN with downlink traffic

Fig. 4. Distance between token holders.

The part of the LSN corresponding to nodes (A, B, C and D) can be considered as a cluster moving from node to node at each expiration of the shuttle duration as presented in Fig. 5.

Fig. 5. Linear cluster shifting.

5. The FIFO Queue Behavior Analysis

In this section we analyze the impact of the token process on the FIFO queue behavior. We perform our analysis on a 1-Redundant LSN with downlink traffic enable. The model we use for a current node is given in Fig. 6. A queue for packets is attached to each node. In the following, we show the impact of the FIFO size and of the shuttle duration length on the end-to-end delay of packets delivered to the sink and on the number of packets reaching it.

We know that the traffic in the case of LSN is managed in a store-and-forward manner. Let us consider the case of Fig. 4(b), a node has only to queue the traffic locally produced during three quarters of its time corresponding to the sleeping state. During the remaining quarter of its time, a node has to superpose local traffic queuing and forwarding activities: the node is successively in reception and transmission states. So, it has to queue the local traffic in addition to the traffic forwarded by the previous node of the LSN.

The content of the queue of a current node can be divided in several parts according to the packets arrival order.

At the bottom of the FIFO, some packets eventually remain if the size of the previously used shuttle was too short to completely empty the queue the last time the node was token holder. Then, at the top of the FIFO, the packets forwarded by a neighboring node being token holder (uplink and downlink traffic) are interleaved with the local production. Thus, in such a network, the traffic is accumulated node after node and saturation appears when the FIFO of the current node is not able to queue the traffic forwarded by the previous nodes. Nevertheless, uplink traffic is always queued at the top of the FIFO and is exposed to the queue overload effect. Packets coming from nodes far from the sink are exposed several times to dropping when the LSN is overloaded.

This is what we call the clipping effect [24]. Packets generated by the allocator or by the first nodes of the linear network are exposed to be systematically dropped before arriving to the sink, when the offered load of the network is too large for the capacity of the FIFO nodes for a given \( T_{\text{Token}} \) and a given shuttle duration (ShuDur).

6. Throughput Analysis

The creation and the forwarding of a new token by the Allocator, allows the traffic of the nodes to be drained in a multi-hop manner. Each time a node is token holder, it can transmit

1) First the traffic towards the Allocator (downlink traffic),

\[ \text{Fig. 6. Node model.} \]
And during the remaining holding time, the traffic having the sink as destination (uplink traffic). Our choice to allow uplink and downlink traffic has an impact on the frequency of the token production. The distance between two consecutive nodes being token holder, expressed in number of hops, depends on the range of the radio link and on the distance between two consecutive nodes. We can notice that the case of a 1-Redundant linear network with downlink traffic, it is easy to show (Fig. 4(b)) that the distance between two nodes being token holder is equal to four hops. That is to say 3 times the radio range expressed in number of hops plus 1. For a 2-Redundant LSN (Fig. 4(d)), this distance increases to 7 hops (3 times the radio range plus 1) and for a 3-Redundant LSN this distance reaches 10 hops (3 times the range plus 1). Fig. 7 is used to introduce how the token passing is managed and how the path of data packets takes advantage of redundancy. In these figures, dashed arrows indicate the path systematically followed by the tokens: $A \rightarrow B \rightarrow C \rightarrow \ldots$ Sink, blue arrows indicate the links which are simultaneously active at a given point in time.

Fig. 7(a) is for the 2-Redundant case, the traffic generated by node A is transmitted directly to node C to be concatenated with the traffic locally generated. This traffic is then forwarded to node E and so on. In such a network it can be seen that there are two branches (A, C, E, G) and (B, D, F, H) which converge towards the sink [25].

Fig. 7(b) is for the 3-Redundant case. The traffic generated by node A is transmitted directly to node D to be concatenated with the traffic locally generated. This traffic is then transmitted to node G and so on. In such a network, three branches converging towards the sink S, can be identified: (A, D, G, J), (B, E, H, K) and (C, F, I, L). This way of spreading the LSN traffic over these three branches has an impact on performance in terms of throughput.

Let us consider Fig. 7(b) when node A and node K are token holder. Node A is allowed to transmit data packets to D, and then its token to B. Node K will directly transmit its data packets to the sink and its token to L. We can notice two important things:

- The sink can receive data packets every time one of its neighbors is token holder. So for a 3-Redundant LSN, the same token gives 3 opportunities to the sink to receive data packets. Each token allows the sink S of Fig. 7(b) to receive consecutively from J, K and L.
- The FIFO of each node being a neighbor of the sink must be large enough to store and forward the traffic of its branch. So for a 3-Redundant LSN, the evaluation of the FIFO size of a sink neighbor node is only governed by the traffic of its branch. The FIFO size of node K of Fig. 7(b) is supposed to be larger enough to concatenate the traffic of nodes of the branch (A, D, G, J).

The two previous points will be used to evaluate the throughput capacity of a LSN according to the redundancy factor R. For R equals from 1 to 3, we want to estimate the optimal number of packets delivered to the sink for a given bit rate. The word optimal is used here to point out the fact the FIFO size is adjusted to the shuttle duration to maximize the throughput and to avoid the loss of data packets. For this evaluation we chose to deal with medium sized data packets and we suppose that all the nodes have the same FIFO size for generality purpose. Each neighbor of the sink is supposed to be token holder during just enough time to empty its FIFO. A direct consequence of this hypothesis is that the size of the FIFO equals the maximum number of data packets a token holder can sent before passing the token. In other words, the capacity of the shuttle is equal to the size of the node FIFO.

Let SC (Shuttle Capacity) be the number of packets a shuttle can carry.

For a 1-Redundant LSN, the sink receives some packets for each time period of 4 Shuttle Durations (SD). The delivery rate of data packets to the sink is:

$$\text{Throughput}(1) = \frac{1 \times SC}{4 \times SD}$$

(2)

For a 2-Redundant network and for each token, the sink receives S packets from its one hop neighbor but also S packets from its two hop neighbor. For such a case two consecutive tokens are separated by 7 Shuttle Durations. The delivery rate of data packets to the sink is:

$$\text{Throughput}(2) = \frac{2 \times SC}{7 \times SD}$$

(3)

For a 3-Redundant LSN we have so:

$$\text{Throughput}(3) = \frac{3 \times SC}{10 \times SD}$$

(4)
For an R-Redundant LSN the theoretical throughput is given by:

$$\text{Throughput}(R) = \frac{R \times SC}{(3 \times R + 1) \times SD}$$  \hspace{1cm} (5)

$\frac{SC}{SD}$ is common ratio independent of R, but depending on the value of parameters of physical and MAC layer used for the LSN. The implementation following this study will be based on the physical layer of 802.15.4, it is why this evaluation will be done using a bit rate of 250 Kbps. The throughput evaluation is based on 100 byte data-frames; the length of the data-frames has not a significant impact on the throughput. The use of short frames reduces slightly the throughput introducing more overhead but it improves the end-to-end delay.

Simulations allow estimating the average time between the transmissions of two consecutive acknowledged frames. This time is useful to estimate the capacity of the shuttle that is to say the number of packets a node is able to send while being token holder.

If each node is token holder during 250 ms (SD) and if transmitting period is 4.5 ms, the shuttle capacity (SC) is about 55 packets. So, $\frac{SC}{SD}$ is equal to 44 Kbps.

$\frac{R}{3 \times R + 1}$ is an increasing function. It starts at 0.25 and converges to 0.33. The optimum throughput obtained for a 3-Redundant LSN is 12/10 higher than the one of a strictly linear network, so this function can be considered as a redundancy gain.

This throughput increase has no impact on the FIFO size of the nodes. Traffic of a branch of such a network should be calculated so that the node of this branch that is neighbor of the sink contains at most SC packets when it becomes token holder. In this theoretical evaluation the impact of factor R on the delivery packet ratio, the time needed for the token passing mechanism was neglected. This choice has no real impact if each node remains enough time token holder in order to send a significant number of data frames.

In the following we show how simulations have been carried out in order to validate this theoretical approach. Our simulations results confirm the expected gain in terms of throughput.

7. Simulations and Results

7.1. Simulation Parameters

We perform simulations on NS2 simulator to show first the impact of the clipping effect on the end-to-end delay and the impact of the redundancy on the throughput. We suppose in the following that all the nodes of the LSN have the same type of queue managed in a first in first out (FIFO) manner. The capacity of this queue can be expressed by the number of packets (FiFoSize) it can contain. The existence of a channel for a downlink traffic is an interesting capability allowing the possibility of changing the period of token generation for example, but, in the following we suppose that the number of packets targeted to the sink represents the dominant traffic. The size of the reverse traffic is ignored in our simulations. The propagation model is Two-Ray Ground model and we suppose also that all the packets have the same size.

### Table 1. Simulations parameters.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1: time for downlink traffic</td>
<td>10 ms</td>
</tr>
<tr>
<td>T2: time for uplink traffic</td>
<td>240 ms</td>
</tr>
<tr>
<td>Token size</td>
<td>11 Bytes</td>
</tr>
<tr>
<td>Frame size</td>
<td>100 Bytes</td>
</tr>
<tr>
<td>Number of repetitions</td>
<td>50</td>
</tr>
<tr>
<td>Physical Layer</td>
<td>802.15.4</td>
</tr>
<tr>
<td>FIFO size</td>
<td>50-60 packets</td>
</tr>
<tr>
<td>Transmission Power</td>
<td>-5 dBm</td>
</tr>
<tr>
<td>Distance between two nodes in 1-Redundant LSN</td>
<td>90 m</td>
</tr>
<tr>
<td>Distance between two nodes in 2-Redundant LSN</td>
<td>45 m</td>
</tr>
<tr>
<td>LSN offered load</td>
<td>[10-100] Kbps</td>
</tr>
<tr>
<td>Simulation start time</td>
<td>[0-1] s</td>
</tr>
<tr>
<td>Simulation end time</td>
<td>[99-100] s</td>
</tr>
<tr>
<td>Shuttle duration</td>
<td>[50-400] ms</td>
</tr>
</tbody>
</table>

7.2. Results and Discussion

Fig. 8 and Fig. 9 present the comparison of end-to-delay for nodes 1, 8, 14 and 15 according to global offered load and LSN density (strictly and 2-LSN). We focus on these nodes for many reasons. Node 1 is the first node of the topology, and does not receive traffic from any node. Thus, node 1 only considers local traffic. Node 14 and 15 are the nodes closer to the sink: they are the nodes with the largest accumulated traffic. Finally, node 8 is a current node at the center of the topology: it receives traffic from others nodes, in addition to its own local traffic production. We show that of course end-to-end delay of a given packet increases as the number of hops increase, that for the two types of LSN. Nevertheless, for each node, the end-to-end delay is constant if the offered load remains under the saturation load (50 Kbps for a 1-Redundant LSN and 40 Kbps for a 2-Redundant LSN). Below this stage, there is no clipping effect due to the low load of the network. Beyond this value, the delay increases slowly for nodes closer to the sink and more quickly for others. The impact of the clipping effect is particularly clear on the two curves for high offered load. Due to the clipping effect, packets retransmissions are frequent and thus increase the delay. In [24], authors explain the impact of the clipping effect on the delivery ratio. Details given about the FIFO in such a LSN confirm the results related to the end-to-end delay.
Let us consider the case of packets that survive the clipping effect and that arrive to the sink when the offered load is high. Thus, when the content of a FIFO is bigger than the capacity of the shuttle, some packets have to stay in the same FIFO, for a time longer than one token period. They have to wait for the next token before being forwarded. As the token production depends on the factor R, packets have to spend more time in the FIFO before being forwarded in the 2-Redundant LSN than in the strictly LSN. This mechanism could be improved by adding a priority policy to the queue management. Frames coming from the farthest nodes should have highest priorities.

The queue management could be improved by using a priority policy to privilege traffic from the farthest nodes or by aggregating packets to increase performance of the kind of LSNs in an energy saving manner.

Fig. 10, Fig. 11 and Fig. 12 present the performances parameters in a 1-Redundant, 2-Redundant and 3-Redundant LSN in term of throughput in order to show the impact of the redundancy. They evaluate the evolution of the throughput as a function of the global offered load for a given shuttle duration. For each shuttle duration, two phases can be identified: between 8 and 48 Kbps the throughput evolution increases. In this phase, the load of the network is low and thus FIFOs contain a few number of data packets for uplink traffic. The number of packets received by the sink from its direct neighbors per second is low. In this phase, nodes do not use all the period T2 of the shuttle duration for uplink packets transmission. Between 48 and 80 Kbps the throughput evolution is stationary. During this phase the load of the network induces saturation. The FIFOs contain the maximal number of packets that can be sent during the shuttle duration. So, the number of packet received by the sink from its direct neighbor is constant. In this phase, nodes send uplink traffic most of the shuttle duration.

The impact of the redundancy can be shown. The throughput at the sink for each shuttle duration and for a given offered load is higher for the 3-Redundant LSN and followed by the 2-Redundant LSN. For example, let us consider the following case: shuttle duration is equal to 50 ms and global offered load is equal to 80 Kbps, the traffic delivered to the sink is 45.5 Kbps, 35.5 Kbps and for 31.1 Kbps for R equal to 3, 2 and 1 respectively. This confirms our theoretical analysis.
Fig. 12. Throughput comparison by Shuttle duration in a 2-Redundant LSN.

Fig. 13 shows the maximal throughput for a global offered load of 80 Kbps for various values of the shuttle duration. This figure shows that maximal throughput is obtained for shuttle duration of 250 ms. Indeed, direct neighbors from the sink transmit maximal number of packets from their FIFO. They are during in the most time of the shuttle in transmitting state. Contrarily for the shuttle duration of 300 ms, nodes are more often in waiting state due to the fact that FIFOs are empty before the end of the shuttle. That is why the throughput decreases for a shuttle of 400 ms.

Fig. 13. Maximal throughput for a global offered load of 80 Kbps for various shuttle duration. The maximal throughput is achieved with shuttle duration of 250 ms.

The impact of the redundancy can be shown. The maximal throughput is 51.27 Kbps, 46.13 Kbps and 40.37 Kbps for R equal to 3, 2 and 1 respectively.

Fig. 14 is the comparison between theoretical analysis and simulations. The maximal throughput given by simulation is always slightly lower than the theoretical throughput. Some frames must be repeated on the path followed to reach the sink. This is decided after the expiration of the timer used to detect a no-acknowledgement. It is the main reason of this slight difference.

Fig. 14. Throughput comparison between theoretical and simulation analysis of the redundancy gain.

8. Conclusions

Linear Sensor Networks have a large interest for monitoring applications as pipeline, roads, trains. In this paper, we propose a token based MAC protocol to manage the access to the medium. The token contains some temporal information which allows nodes to use the transmission channel for an amount of time. We define an R-Redundant LSN where R is the number of neighbors in each direction for a given node. We study the behavior of LSN in the case of the three topologies corresponding to R equals 1, 2 and 3. We show how the token is propagated from node to node in the network and how the token generation period is calculated in order to avoid packet collisions and limits the effects of the token losses before reaching to the sink. We highlight what we call the clipping effect. This is illustrated in our simulation results where we study the performances of our protocol in terms of end-to-end delay.

We also study the impact of the redundancy on the throughput at the sink.

We show by theoretical and simulation analysis, that more the factor of redundancy R is more great the throughput at the sink is also great. We show also that the redundancy allows nodes to have an equitable distribution of the traffic by dividing the network into branches.

In future works, we plan to use reverse channel in order to master the token production frequency according to the spatial reuse and energy saving constraints. Another way to improve the capacity of such a network is to add a priority policy to FIFO management of nodes by allowing highest priorities to the data frames coming from the farthest nodes. Finally, we plan to use a Log-normal Shadowing model in order to model the path loss due to the complexity of the propagation conditions in a real environment.

References

[1]. Z. Wang, X. Zhao, X. Qian, The application and issues of linear wireless sensor networks, in Proceedings of the International Conference on System Science, Engineering Design and


