Hindawi Publishing Corporation
Journal of Computer Networks and Communications
Volume 2012, Article ID 863521, 10 pages
doi:10.1155/2012/863521

Research Article

Delay-Tolerant, Low-Power Protocols for Large Security-Critical Wireless Sensor Networks

Claudio S. Malavenda,[1,2] F. Menichelli,[2] and M. Olivieri[2]

[1] *Large Systems BU, SELEX Sistemi Integrati, 00131 Rome, Italy*
[2] *Department of Information Engineering, Electronics and Telecommunications, Sapienza University of Rome, via Eudossiana 18, 00184 Rome, Italy*

Correspondence should be addressed to M. Olivieri, olivieri@diet.uniroma1.it

Received 1 August 2012; Accepted 23 October 2012

Academic Editor: Bruno Neri

This paper reports the analysis, implementation, and experimental testing of a delay-tolerant and energy-aware protocol for a wireless sensor node, oriented to security applications. The solution proposed takes advantages from different domains considering as a guideline the low power consumption and facing the problems of seamless and lossy connectivity offered by the wireless medium along with very limited resources offered by a wireless network node. The paper is organized as follows: first we give an overview on delay-tolerant wireless sensor networking (DTN); then we perform a simulation-based comparative analysis of state-of-the-art DTN approaches and illustrate the improvement offered by the proposed protocol; finally we present experimental data gathered from the implementation of the proposed protocol on a proprietary hardware node.

1. Introduction

In recent years, wireless sensor networks (WSN) research has grown exponentially spreading through several fields of science, from circuit design to algorithm design, antenna design, and protocol design. The main constraints that a generic WSN node has to deal with can be summarized by its limited computing resources and its energy consumption requirements. While the computing resources and corresponding consumed energy tend to grow with silicon technology improvements, available energy budget does not advance very fast with battery technology or can even be bounded in other cases (i.e., energy scavenged from the environment). Power management must therefore be taken into account at every level of the design of any WSN.

In security-critical applications, the deployment of large networks faces—among others—the implications of delay variability on the correct operation of security algorithms. This paper illustrates the results of an industrial work on the analysis, optimization, implementation, and experimental testing of a dedicated protocol featuring delay tolerance and energy efficiency for large WSNs in the security application domain.

This paper is organized as follows: in Section 2 we present an overview on wireless sensor networking with particular regard to delay-tolerant networking (DTN) and specifically to the DTN logical link control (LLC) layer, with the aim of stating general and direct hints for the protocol design. Section 3 illustrates a dedicated DTN simulation framework and presents simulation results on existing widely used protocols compared with the newly proposed protocol. Section 4 presents the test methodology and the experimental results on a working application of the new protocol implemented on a hardware sensor node architecture used in security market.

2. Overview on DTN Design

2.1. WSN Protocol Stack General Issues. A WSN is a dynamic, self-configuring network composed of interconnected, battery-powered embedded systems. The main characteristics of these kinds of systems are scalability, self-organization, self-configuration, adaptation, exception-free operation, and communication failure tolerance [1]. All these requirements have to be implemented in an embedded device (node)

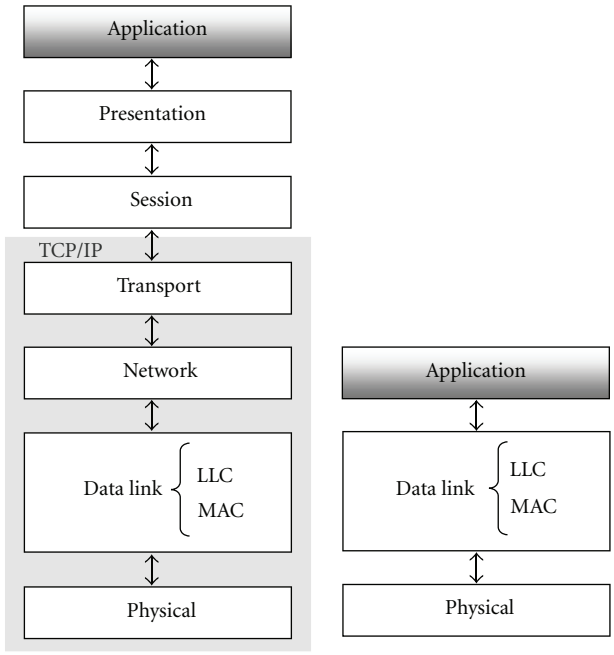

Figure 1: Comparison of a common OSI model and a reduced one for WSN application.

that typically has limited energy budget, computing power, storage capacity, transmission range, and bandwidth.

As formerly investigated in several works for embedded systems communications [2, 3], an important aspect for achieving the above goal is to reduce the protocol stack of a common OSI model in order to have a faster computation and smaller number and size of packets to transmit. Figure 1 shows the difference between a typical OSI model (left side) and an adapted one (right side) for WSN systems.

A shorter stack is a simplification from the point of view of network design and computation load within the node, but complicates the software development of services offered to applications. In fact, a shorter stack implies that applications directly drive layers close to the physical one and requires much more complex workarounds to achieve "high layer-like" functionalities. Such complexity is usually hidden to the application developer in common OSI-based network stacks (e.g., TPC/IP networks) where lower layers are seen as black boxes. As a consequence of such limiting factor in application development, the huge production of WSN protocols in the last years has often adopted the classical approach ignoring the optimization of the stack at lower layers (shorter stack) and often producing a heavy weighted protocol stack that does not fit with the most common operating constraints for WSN [4]. On the contrary, the proposed approach fully adopts a shorter stack approach.

In addition to the generic framework of computation-optimized shorter stack, the following main characteristics of WSN protocols that differ from a common TCP/IP network have been addressed in the new protocol.

(i) *Intermittent connectivity:* a connection path among nodes does not always exist, and available links are time varying. So the network could be partitioned in several and different parts during its life.

(ii) *Relatively long and variable delay:* propagation delay among network nodes is relevant. Delay is not fixed and can vary according to network traffic and link quality. This condition tends to cause failure in protocols that are based on quick data/ack return.

(iii) *Lossy link:* the end-to-end communication suffers a high error rate due to several physical causes. Packets are frequently lost in hop-to-hop connection.

2.2. Asynchronous Networking. Taking in mind the above starting point, a *synchronous* MAC [5], either slotted or frame based, could be hardly suitable because of the synchronization needed among nodes. This result comes from years of experimentation and protocol testing during the development of the proposed protocol. In fact, synchronous MAC needs the successful communication among nodes of periodic packets that synchronize neighbors for subsequent transmissions. Each sensor node would start this communication with a delay according to a fixed cycle started with the shared time-synchronization event.

Conversely, *asynchronous* MACs [5] do not impose restrictions on when a sleep/active cycle is taking place. Neighbors therefore do not need to coordinate their cycles and consequently wake up independently of each other. This avoids the overheads and bookkeeping associated with running a time synchronization protocol and a global scheduler, as in a synchronous MAC, at the expense of requiring the sending nodes to arrange a rendezvous with the intended receiver whenever it wakes up. As a consequent drawback, asynchronous protocols suffer of congestion problems when the density of active nodes becomes high due to the intrinsic nature of its relaying mechanism. In fact, the number of neighbors becomes a pointer to discover potential congestion in the network. However, asynchronous MAC remains the preferred way in our application context where the reliability of the medium and of communication timing cannot be continuously known (DTN application context).

2.3. Delay-Tolerant Networking. DTN responds to the need to deliver messages in networks characterized by probable lack of end-to-end connection paths, either proactively available [6] or reactively established with conventional routing protocols. Thus, these networks must operate without the assumption that there is a permanent connection or instantaneous end-to-end paths between the source and the destination node.

This is quite common in those WSNs where disconnections among nodes occur dynamically. The main causes of node disconnections can be attributed to *mobility of nodes* and *sparse network*. In the first case, the assumption is that a WSN node has mobile capabilities, and its movement can lead to lack of connectivity when the node moves out of the radio range of any of its neighbours. The sparse network case may occur even when a WSN comprises only static nodes due to node malfunction, battery discharge, change in node's functional state, or node switch to sleep mode following

a duty cycle different from its neighbours. The resulting distribution of nodes creates holes in network topology. In our target application context, both mobility of modes and sparse network condition must be assumed.

The solutions to this issue are usually some elaboration of the basic *store* and *forward* scheme. In this direction, the concept of *Data Mule* [7], as a specialized vision for the general DTN case, is sometimes introduced in mobile networks. A Data Mule is a mobile WSN node with high data storage capability, high throughput, and ability to move in order to establish connection among unconnected islands or networks.

As shown in Figure 2, the Data Mule collects messages incoming from a network island, when it is in proximity to that island. If an incoming message is addressed to a node of the network that is within the Data Mule's transmission range, the Data Mule will forward the message. Otherwise, the Data Mule stores the message and physically moves towards the destination node's network island to start forwarding the stored messages.

As for the routing layer, typical routing protocols for WSNs are divided in reactive and proactive ones [6]. Proactive routing sets up predefined paths from all source nodes towards all possible destination nodes before starting to route data messages. Reactive routing establishes a connected end-to-end path on demand, when a generated message needs to be routed from its source towards a destination node. In a typical DTN network application, a path typically cannot be preestablished, so that reactive routing is the mandatory choice.

2.4. Overhead Sources in Delay-Tolerant Networking. For the power efficiency of a WSN protocol, a critical aspect is the minimization of bytes/packets transmitted in the network for its correct operation, in order to minimize the energy spent in transmission. As a result, a primary design criterion is the overhead of communications exchanged for protocol specific purposes and of other energy consuming operations. In the specific context of DTNs, the main sources of overhead that must be addressed and minimized are as follows.

(i) Idle Listening Overhead. The time spent listening to the medium and receiving nothing. While communications are usually a quite rare event, the receiving radio must be kept on every time a packet could be incoming; otherwise it would miss some of the messages being sent to it. This is the main source of energy waste as typical radios consume much more energy in receive mode (even when no data is arriving) than in sleep mode. In asynchronous protocols, the idle listening can be computed with the receiving time-window that occurs each WOR period over the effective receiving periods that catch radio packets.

(ii) Overhearing Overhead. The nature of the wireless medium implies broadcast communication among neighbor nodes, so that all neighbors of the destination node will receive the same packet. Overhearing these messages is a waste of energy: the node spends energy to receive a packet that is not addressed to it. This source of overhead becomes problematic in dense networks. These kinds of deployments are common, for instance, when sensing range is smaller than communication range so that a high number of nodes are inside the communication range, in order to cover the smaller sensing range.

(iii) Collision-Related Overhead. When a packet collision occurs, usually it implies the retransmission of the collided packet and a waste of energy. In this respect, *traffic fluctuations* in WSN where packets are generated just in case of an event to report can cause a peak of transmission load, network congestion, and frequent retransmissions. Also back-off period calculated with random generators can still produce contentions, because collisions can still occur between the carrier sense time and the effective transmission. The protocol overhead usually uses the RTS/CTS handshake to implement collision avoidance, but it is considered prohibitive in comparison to the small, 32-byte WSN payloads leaving the hidden-terminal problem unaddressed.

(iv) Protocol Overhead. All headers/footers and control packets are overhead, that is, a waste of energy in front of zero data information transmitted. The minimization of these fields/packets type is the scope of a good WSN design.

The optimization of these parameters has driven the design of the proposed protocol, tested in Section 4.1.

2.5. Performance Metrics. Performance metrics are not easy to define in WSN due to its unique properties. Common metrics used in wireless communication, like *fairness* and *throughput*, might not be meaningful because WSN nodes can cooperate and because raw data transmission is a rare application in WSN.

We used the following metrics to measure protocol performances in both simulator and implementation, whose results are reported in Section 3.2.

(i) Latency. Time delay between the message transmission from the source node and the first arrival of the message to the destination node.

(ii) Delivery Ratio. Ratio of the number of successfully delivered data packets over the number of packets generated by source-nodes.

(iii) Overhead. Number of redundant packet copies that are disseminated in the network and the extra control packets exchanged for protocol specific purposes.

We note, for completeness, that also another metric can be defined.

(iv) Network Efficiency. The sum of all packet copies generated by all of the relaying nodes (including the source node) in order to deliver one packet (other definitions can be application dependent).

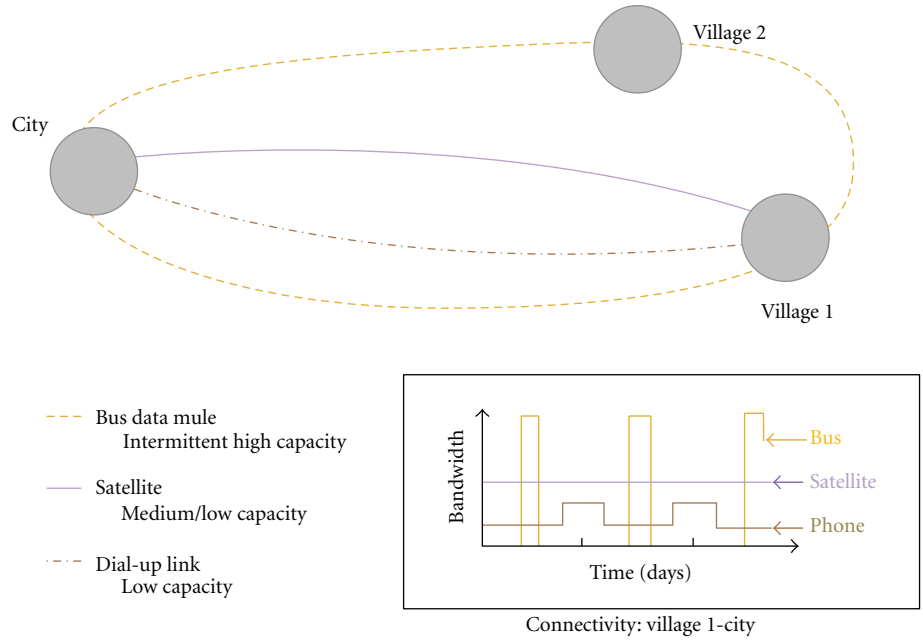

Figure 2: Comparison of common bandwidth versus time data exchange in intermittent island and medium/low capacity but connected island [8].

However, the definition of *network efficiency* is quite variable and it is usually related to a specific application. That is why it will not be used as a comparison in Section 3.2.

3. Simulation-Based Analysis of Existing DTN Protocols

3.1. Protocols under Analysis. The most widely used DTN protocols reported in the literature [7, 9–12] are listed below:

(i) Direct Diffusion,
(ii) First Contact,
(iii) Epidemic,
(iv) Fuzzy Spray,
(v) PRoPHET,
(vi) MaxProp,
(vii) Spray and Wait (and variants),
(viii) Scar,
(ix) FAD,
(x) Rapid.

The above protocols can be classified according to the map in Figure 3. The gray cell represents typical characteristic of a DTN protocol. The lower part of the map inherits the characteristics at the highest level. In the following, a brief description of each characteristic is listed in the map.

Single Transmission. A packet is transmitted in broadcast and just once after its creation.

Multitransmission. A packet can be transmitted more than once from the same node.

Replication. A packet can be relayed from a receiving node. This is the first step for multihop communication.

Queue Management. From this level, the management of relays starts. In this case, the relay of the packet is accomplished according to a queue that can be managed on the sender node in several ways. For instance, a simple management can be a FIFO queue, but parameters on node energy are taken into account.

Delivery Probability. According to the specific protocol, every packet is associated with a probability that can, for instance, be linked to the destination of the packet, or according to the routed path. If the probability associated to the incoming packet is greater than a certain percentage, the packet is relayed or not.

Limited Copies in Network. This characteristic limits the number of copies that can simultaneously coexist in the network. Protocols that implement this characteristic vary on the rules adopted to limit copies in the network.

We performed a comparative analysis of the above protocols on a commercially available simulator [10], in order to have a basis on which we can build the mechanisms that could lead to an optimization of the network in the target application context.

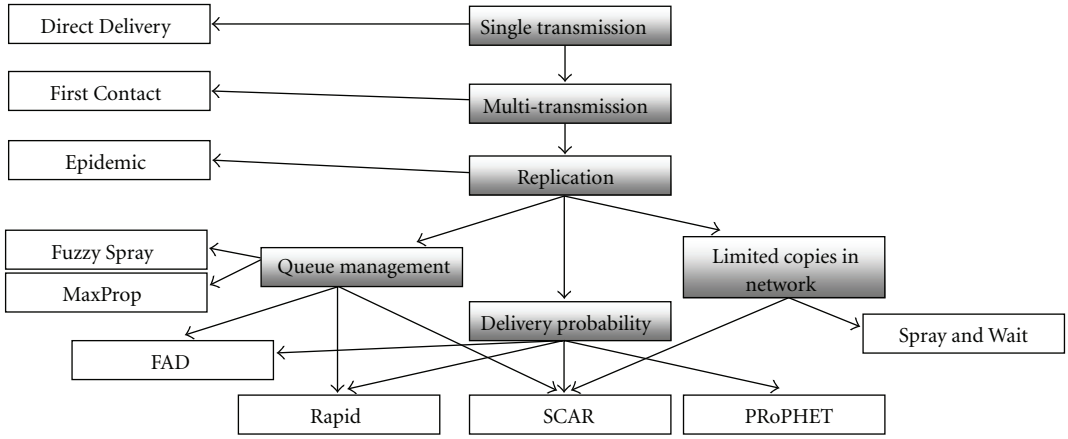

Figure 3: Map of most widely used DTN protocols.

3.2. Simulation Results. The diagrams in Figures 5 and 6 present the results of the comparison based on the previously chosen metrics.

Figure 4 shows latency measures for all tested protocols. In a subsequent analysis, we limit the exploration to a set of the most performing ones, specifically MaxProp, Prophet, and Spray and Wait. PRoPHET is representative of protocols implementing only the data forwarding scheme, Spray and Wait only the controlled replication scheme, and MaxProp both.

It is possible to remark that due to the limited buffer size, PRoPHET significantly suffers from message discarding, while Spray and Wait, by limiting the total number of copies, can in any case achieve good performance.

From Figure 4, we can also note that the selected protocols mark two extremes of a range of latency values, while other protocols are positioned between them according to the scheme implemented. Other protocols having performance outside this range are considered out of interest.

Figures 5 and 6 present the performances of the selected protocols, regarding delivery ratio and overhead, respectively.

Considering the trade-off between performance and power consumption, the Spray and Wait protocol comes out to be the one with the lowest overhead while maintaining average results on delivery ratio and delay, in the target application domain. As a consequence of the analysis, the newly developed protocol has been an optimization of Spray and Wait.

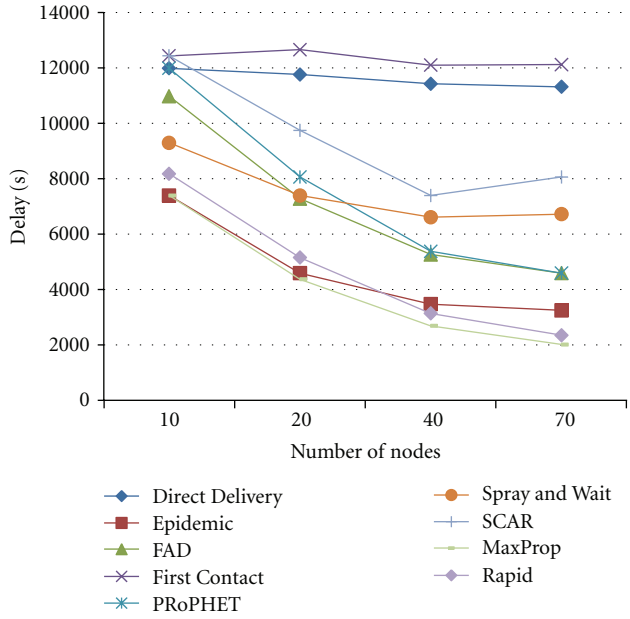

Figure 4: Latency result comparison.

4. New Protocol Simulation and Experimental Testing

4.1. Analysis on a Dedicated State-Accurate Simulator for DTN Protocols. In order to have a deeper control on the developed protocol, with state-level accuracy, and in order to have a better energy model, a custom simulator framework for DTN protocols has been developed. OMNET++ 4.2 [13] has been chosen as a starting framework. The simulator has been layered over the basic OMNET API, without any other add-on installed.

The simulator aims at modeling, with state-level accuracy, the hardware of a WSN node with particular regards to the radio and microcontroller states, in order to produce accurate results on their power consumption. It has been designed in order to provide a dynamic positioning of WSN nodes over a simulated area.

Connections among nodes are dynamically established according to physical parameter relative to each single node, which is modeled with a particular antenna gain and receive sensitivity. Working frequency is used to model the communication range achievable from each node according to the mutual position of the nodes.

Figure 7 shows a test topology used to verify the reliability of the simulator. The graphical rendering of OMNET++

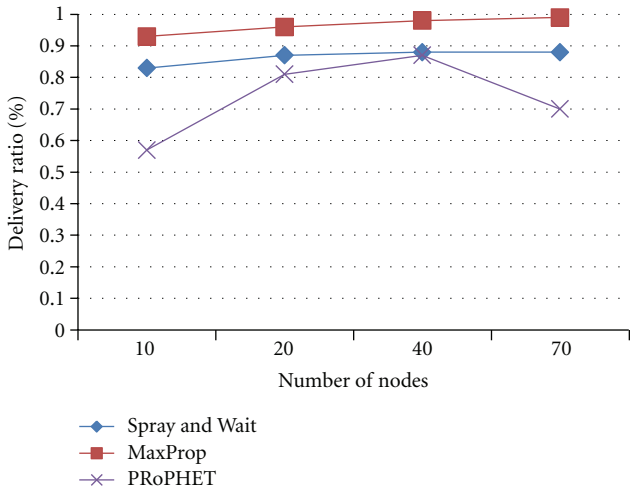

Figure 5: Delivery ratio result comparison.

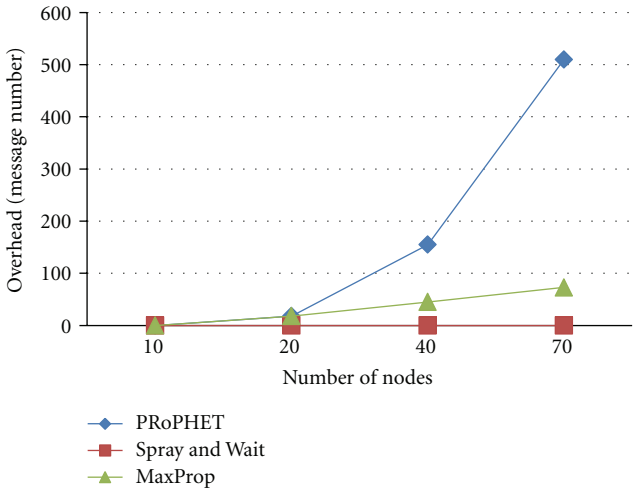

Figure 6: Communication overhead result comparison.

shows the topology of fixed nodes disposed on a virtual field. Each position of the virtual field is mapped with a coordinate reference in a 3D virtual space. In this way it is possible to map the mutual distance between nodes.

A configuration file describes the physical characteristics of each node joining the WSN with the possibility of inheriting standard ones, in the case that no particular physical parameters have been specified for a node.

The first use of the simulator has been done to verify timing on packet delivery and model packet exchanging among nodes with a first version of the selected protocol, in order to validate the simulator with known results and acquire more data on the simulated network.

As it is possible to see in Figure 9 that it never occurs that a node starts transmitting while another one in its visibility range is yet in transmission phase. Moreover, it is possible to see the packet relay period of 1 second when no collisions occurs which correctly model the protocol used.

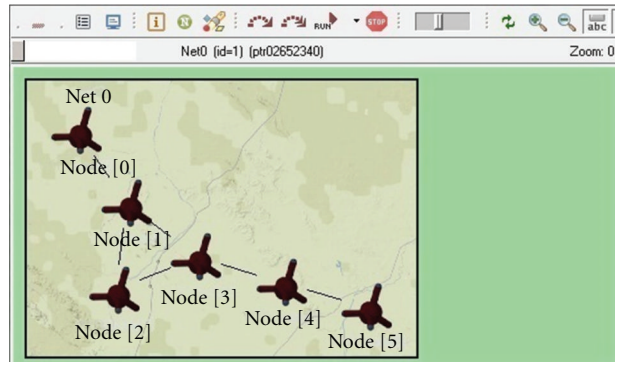

Figure 7: Network topology—A screenshot with fixed nodes.

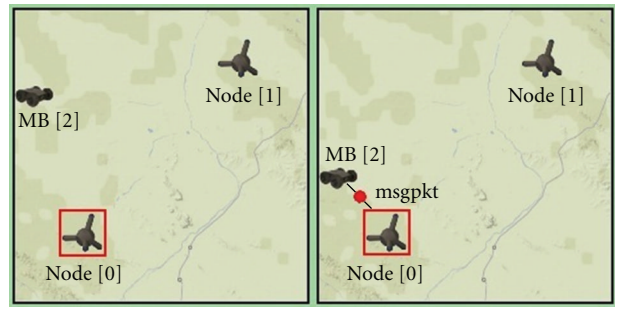

Figure 8: Network topology—mobile UGV and fixed nodes.

Since the protocol is a DTN one and well fits for communication among mobile nodes, a mobile node modeling feature has been developed and introduced in the framework as well (Figure 8).

4.2. Hardware Test Session No. 1. The testing of the protocol implemented in a real commercial hardware WSN node has been divided into different set of testing sessions.

The first session deals with node power consumption, by analyzing the duty cycle and power consumed during different transmission phases. All testbeds have been set up in the WSN laboratory of SELEX Sistemi Integrati (formerly ElsagDatamat) in an air-conditioned environment at 25°C.

4.2.1. Testbed Setup. The testbed is settled up with a single MasterZone [8] node.

The node has been programmed in order to configure its transceiver in Wake-on-Radio status: the radio goes in reception mode for a short period (15 ms) and after that stays in sleep state for 800 ms.

Sporadically, the node performs a transmission. In this configuration, it is possible to monitor the consumption of the node during its reception and transmission phase.

The measurement of the current consumed by the node is performed with a current probe in order to produce a temporal log of measures and distinguish the power consumed between each phase.

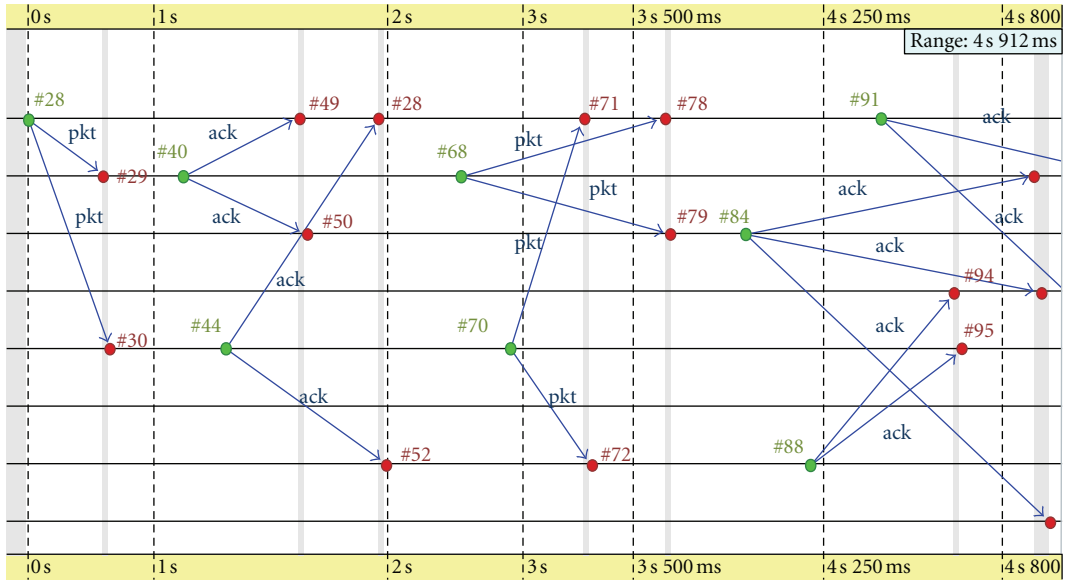

Figure 9: Network timing monitoring view.

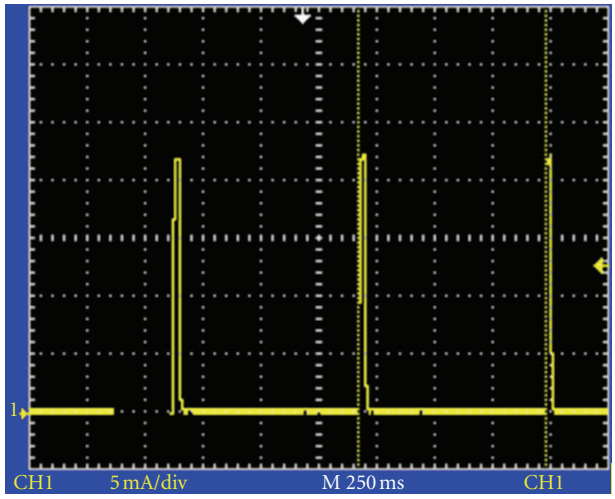

Figure 10: Receiver WOR period—power consumption test result.

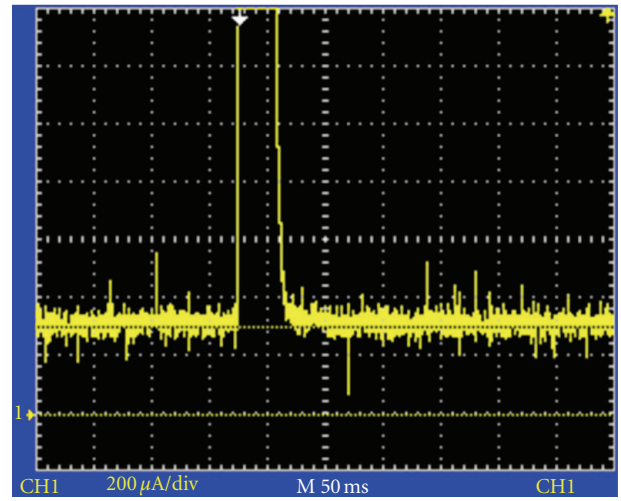

Figure 11: Sleep power consumption test result.

The log obtained has been splitted and reported in Figures 10 and 11 in order to focus on each particular Rx and Tx phases.

4.2.2. Results and Analysis. The first measure concerns the WOR timing. As from Figure 10, every 800 ms the power consumed by the node shows a high step due to the state change from "idle/sleep" to "receive."

Figure 11 shows a detail of the power consumption trace where we can observe a background consumption of 200 uA in sleep mode and a raising peak of 22 mA in active receive mode.

Figure 12 illustrates the corresponding test for a transmission phase. We can see a first phase of 30 ms with a power consumption of 30 mA for the CSMA/CA phase at the beginning of the transmission phase, and a 900 ms transmission phase with a 23 mA of power consumption at −15 dB of Tx power.

It is possible to observe that the transmission phase has a bounce in energy consumption. It is due to the fast change of states in the transmitter radio (from idle to transmitter). Results obtained in this test comply with the expected results.

4.3. Hardware Test Session No. 2. The second set of tests has been set up using a single node. This test deals with the correct functioning of the radio of the node.

4.3.1. Testbed Setup. The target measures in this test aim at the detection of the sensitivity of the node radio receiver

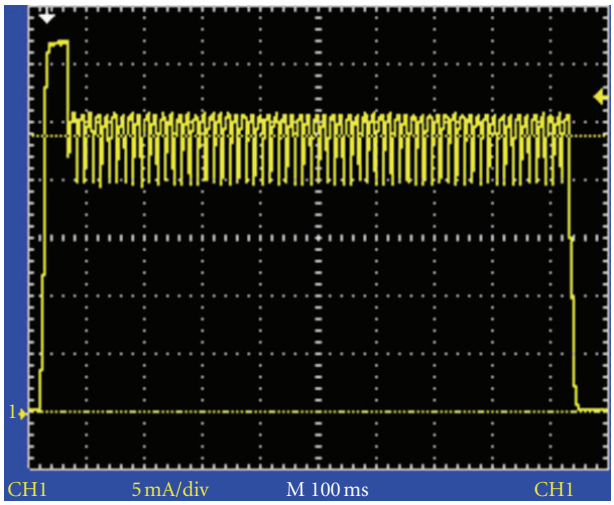

Figure 12: Transmitter power consumption test result.

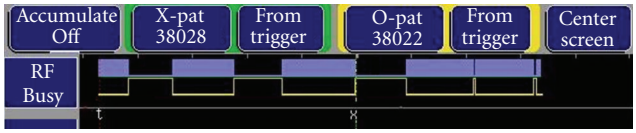

Figure 13: Sensitivity measurement.

and confirm the correct functioning of the CSMA strategy adopted.

The measures are accomplished by means of a logic state analyzer linked to a control IO of the node under test. This pin is directly controlled by the microcontroller and reports the status of the radio channel in use (i.e., if the radio channel is busy or free, according to a predefined threshold on received power). The threshold has been set to the minimum value available: in this way the pin will take a low-logic state when the minimal energy is detected in the received channel.

The antenna plug of the node has been connected directly through a coaxial cable (50 Ohm, SMA connector) to a RF generator which provides a radio signal directly injected in the reception circuits of the node. The direct connection from the RF generator avoids errors in the measurement that could be introduced by a free air link.

4.3.2. Results and Analysis. The results collected prove a −90 dBm sensitivity of the node. In fact, going below this threshold causes the pin that monitors the status of the air to bounce independently from the actual injection of RF.

In Figure 13, it is possible to see the correct response obtained from the node when −90 dBm of RF power is injected.

The first line monitors the transmission of RF from the generator. The absence of glitch in the white signal means a good clear channel measurement. Going below this power, the white line starts to bounce: the node cannot really discern a free channel from a busy one. This test attests the sensitivity of the node at −90 dBm.

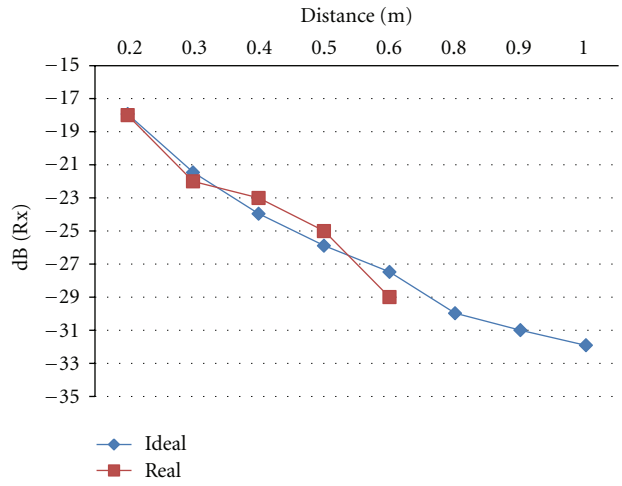

Figure 14: Measurements achieved versus expected ones.

4.4. Hardware Test Session No. 3. The third set of tests has been set up on a two node network: the target is the measure of the distance achievable by a point-to-point transmission.

4.4.1. Testbed Setup. This testbed is set up with two MasterZone nodes [8] suitably programmed.

The first one has been configured to periodically transmit a packet. In this test environment, the content of the packet is not important, but just the fact that it is received or not by the second node, since we are going to measure physical values related to RF transmission.

The second node is configured to remain in reception state, read the RSSI level of received packets, and translate it in dBm values. This translation has been tuned in advance using reference values from datasheets. The node sends the data to a PC via an RS232 serial connection, where they are timestamped and logged.

4.4.2. Results and Analysis. Figure 14 shows a plot of actual measurements towards ideal values. The ideal values (in blue) depict the expected dBm power at the receiver according to a Free Path Loss law with a Tx power of 5 dBm and an antenna gain of −6 dBm at a working frequency of 420 MHz.

As we can see from the figure, the mapping of ideal values and the real ones is quite 1 : 1 with a few dBm difference.

Assuming that the measurements follow this trend, the threshold level of −90 dBm may be reached at 800 m distance between the transmitter and the receiver. More tests should be conducted with greater distances between nodes to confirm the trend with distances next to the maximum one achievable.

4.5. Test Session No. 4. The fourth session has been set up on a multihop testbed and the target parameter has been the measure of the delay. In this testbed, we have a source node, a relaying node, and a sink node. All nodes are visible to each other. The test aimed at verifying a simple relay functioning.

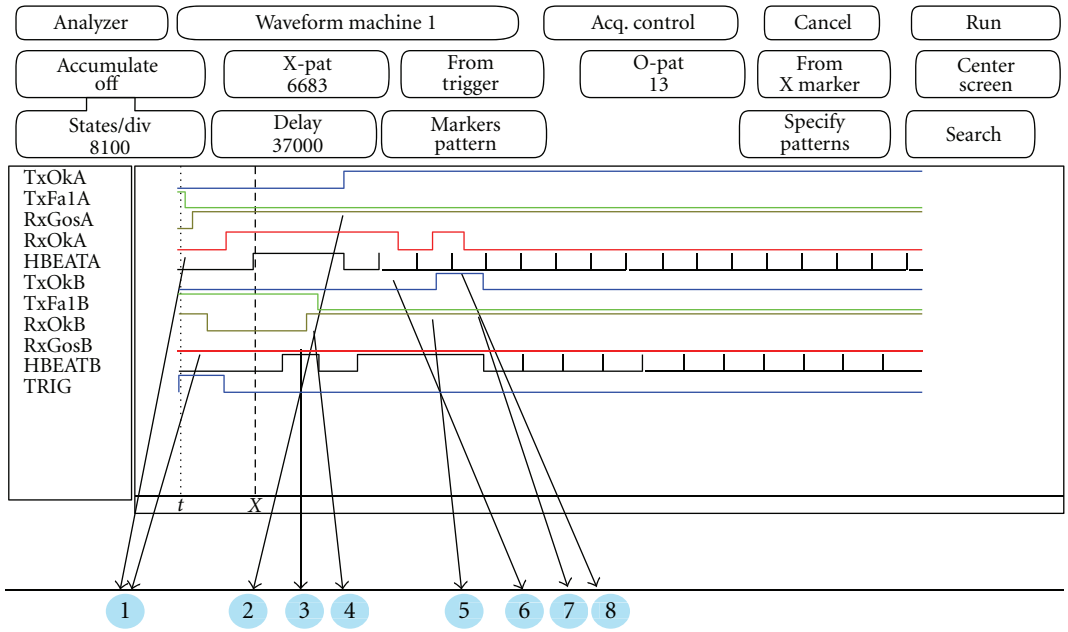

Figure 15: Multihop signal test.

4.5.1. Testbed Setup. This test bed is set up with two MasterZone nodes [8] suitably programmed (node A and B) and one node interfaced with a PC (TRIG).

All nodes have been configured to test the multihop functionality of the protocol when incoming messages are relayed to neighbor nodes.

The TRIG node transmits "ping" packets under control of the PC. This node will not take part in any other radio handshaking. The "ping" packet received by node A is relayed to node B.

The state analyzer will log all control pin on both nodes in order to catch a clear picture of the handshaking. The test aims at examining if the routing with a minimal set of nodes reflects the expected behavior.

4.5.2. Results and Analysis. Figure 15 reports the result of the test conducted with the configuration just described. The cyan balloons highlight the following communication facts.

(1) Nodes A and B receive ping command from the sink node (A receive twice in the same slot).

(2) A answers to the sink node with a delay of 2.42 sec.

(3) B receives the answer transmitted by A (the signal toggle monitor the end of a transmission).

(4) B tries to forward the ping request issued by the sink node but senses the air occupied.

(5) B forwards the sink request.

(6) A receives the forwarded request from the sink node and filters it because already received.

(7) B transmits the answer from A.

(8) A receives its own answer from B and just drops it.

This handshake reflects the expected behavior.

A logic state analyzer has been connected to the nodes to monitor handshaking occurring between node A and B. Five I/O pins have been configured on each node to monitor events on nodes according to Table 1. The events monitored deal with a successful or failed transmission started from the node, a successful reception, or a reception of a packet yet stored in the reception queue (ghost packet). The signal Hbeat reveals the internal timing of the node.

Table 1: Signal meaning mapping.

Signal name	Meaning
TxOk	Toggle after a successful transmission
TxFal	Toggle after a failed transmission
RxOk	Toggle after the reception of a packet
RxGos	Toggle after the reception of a filtered packet
Hbeat	Low when the node is in sleep mode

5. Conclusions

In this paper, we first presented a comparison between different delay-tolerant protocols for WSN systems. Starting from the definition of the metrics of interest for WSN performance analysis found in the literature, we compared different delay-tolerant protocols.

A wide range of protocols have been investigated through available simulators. After a set of simulation results and comparisons according to the chosen metrics, the most promising one has been selected to develop a new custom protocol.

In order to reach a more accurate control of the simulation and incorporate a wider set of simulation parameters, the simulation platform has been switched to a more versatile one. The code of the custom protocol based on the selected one has been implemented in the new simulation environment. The first simulation results have been collected with fixed and mobile nodes. These tests have confirmed the suitability of the protocol for an actual implementation.

Finally the custom protocol has been ported on a proprietary platform: the correct implementation has been validated through a set of tests on timing, handshaking and power consumption of the developed node, confirming the expected results and paving the way to further subsequent development.

Acknowledgments

This work was supported by SELEX Sistemi Integrati, a Finmeccanica Company. Special thanks are due to to Luca di Donato for his support.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.

[2] B. Maaref, S. Nasri, and P. Sicard, "Communication system for industrial automation," in *Proceedings of the IEEE International Symposium on Industrial Electronics (ISIE'97)*, vol. 3, pp. 1286–1291, July 1997.

[3] W. Hou, S. Hu, R. Li, and M. Fei, "A wireless industrial networks protocol stack with time synchronization and node positioning," in *Proceedings of the IET Conference on Wireless, Mobile and Sensor Networks 2007 (CCWMSN'07)*, pp. 1077–1080, Shanghai, China, December 2007.

[4] S. Jain, K. Fall, and R. Patra, "Routing in a delay tolerant network," in *Proceedings of the ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM'04)*, vol. 34, no. 4, pp. 145–158, New York, NY, USA, September 2004.

[5] C. K. Toh, *Ad Hoc Mobile Wireless Networks: Protocols and Systems*, Prentice Hall, New York, NY, USA, 2001.

[6] "Proactive and reactive routing in wireless sensor networking," http://it.wikipedia.org/wiki/MANET.

[7] M. Demmer, E. Brewer, K. Fall, S. Jain, M. Ho, and R. Patra, "Implementing delay tolerant networking," Intel Corporation, 2004, http://www.dtnrg.org/docs/papers/demmer-irb-tr-04-020.pdf.

[8] http://www.selex-si-uk.com/pdf/Masterzone.pdf.

[9] K. A. Harras, K. C. Almeroth, and E. M. Belding-Royer, "Delay tolerant mobile networks (DTMNs): controlled flooding in sparse mobile networks," in *Proceedings of the 4th IFIP-TC6 International Conference on Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communication Systems (NETWORKING'05)*, pp. 1180–1192, May 2005.

[10] "ONE," simulator web page, http://www.netlab.tkk.fi/tutkimus/dtn/theone/.

[11] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait: an efficient routing scheme for intermittently connected mobile networks," in *Proceedings of the ACM SIGCOMM Workshop on Delay-Tolerant Networking (WDTN'05)*, pp. 252–259, August 2005.

[12] B. Pásztor, M. Musolesi, and C. Mascolo, "Opportunistic mobile sensor data collection with SCAR," in *Proceedings of the IEEE Internatonal Conference on Mobile Adhoc and Sensor Systems (MASS'07)*, pp. 1–12, Pisa, Italy, October 2007.

[13] OMNET++, http://www.omnetpp.org/.