Adaptive Collision Avoidance through Implicit Acknowledgments in Wireless Sensor Networks

Daniele Messina, Marco Ortolani and Giuseppe Lo Re
Dept. of Computer Engineering, University of Palermo
Viale delle Scienze, ed. 6 – 90128 Palermo, Italy
{dmessina, ortolani, lore}@unipa.it

***Abstract*—The large number of nodes, typical of many sensor network deployments and the well-known hidden terminal problem make collision avoidance an essential goal for the actual employment of Wireless Sensor Network (WSN) technology. Collision avoidance is traditionally dealt with at the MAC Layer and plenty of different solutions have been proposed, which however have encountered limited diffusion because of their incompatibility with commonly available devices.**

As part of our work on the GEOGRID project, we developed an approach to collision avoidance which is designed to work over a standard MAC Layer, namely the IEEE 802.15.4 MAC and is based on application-controlled delays of packet transmission times. The proposed scheme, which we present in two variants, is simple, decentralized and scalable.

***Index Terms*—Wireless Sensor Networks, Collision Avoidance, Implicit Acknowledgment.**

I. INTRODUCTION AND BACKGROUND

Wireless Sensor Networks (WSNs) consist of small battery-operated low-cost nodes which collect information from the environment and communicate through wireless links. WSNs have specific characteristics which differentiate them from wireless ad-hoc networks, including typically larger network sizes, limited energy and different traffic characteristics and requirements [1]. Application scenarios, where many hundreds of devices are deployed, pose great scalability and manageability challenges. Energy sources are generally considered not renewable, hence, in order to extend the network lifetime, both hardware and protocols design have been primarily concerned with energy efficiency. Since the transceiver consumes a significant amount of energy, a considerable research effort has been directed to the design of energy-efficient communication strategies, and an important role is played by MAC protocols, which provide schemes for multiple access to the wireless medium. MAC protocols for WSNs, in particular, are required to address the fundamental problem of collision avoidance while coping with a large number of competing stations and severe hidden terminal issues.

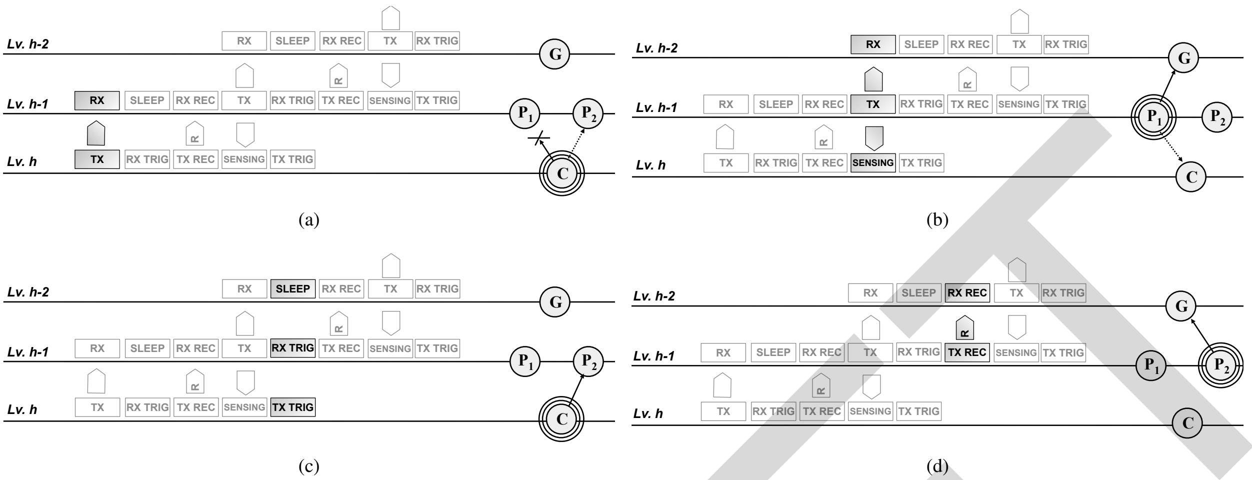
A general classification of sensor network MAC protocols makes a distinction between random (or non deterministic) protocols and scheduled ones [2]. The former are less complex and can be fully distributed, hence they are generally more scalable; low complexity and the absence of shared information, or 'state', also reduce memory and processing requirements as well as control overhead. Most non deterministic protocols are modeled after CSMA/CA and exploit the information that is directly available through the node radio, therefore being able to avoid collisions only at the sender side. The introduction of RTS/CTS control packets and virtual carrier sensing has been proposed to specifically address the hidden terminal problem, however such approach is not general, as it is based on the assumption of symmetric links and cannot be applied to the case of broadcast transmissions. Scheduled MAC protocols organize nodes for transmitting according to a common schedule and provide the capability of reducing energy waste due to collisions, *overhearing* and *idle-listening*, at the cost of higher complexity, state information distribution, and synchronization overhead. Schedule maintenance is complicated by node mobility and failures, network segmentation and incomplete information available at the nodes.

A common drawback of MAC-based approaches, which prevents their widespread adoption, is the incompatibility with existing devices. The increasing interest in the recent IEEE 802.15.4 standard for WSNs and the diffusion of IEEE 802.15.4-based devices has motivated our research towards a different approach to the collision avoidance problem.

In this work we propose a collision avoidance technique, which we developed as part of our work on the GeoGrid project, aimed at improving the collision avoidance functionality provided by the IEEE 802.15.4 MAC protocol. This technique will be implemented in the hardware/software framework also developed by our research group, for supporting a distributed, WSN-based system for the acquisition of marine environment data. Full compatibility with standard IEEE 802.15.4 devices, requiring no modification either to the MAC Layer, or to the PHY layer, was a project requirement.

Our technique exploits the periodic nature of traffic, typical of many applications of WSNs, in order to adaptively set up a global schedule of packet transmissions and minimize collisions. The schedule is controlled by the Application Layer, through the introduction of proper delays when passing packets from the application to the MAC Layer, for the transmission over the wireless channel.

In our reference scenario the network uses a tree-based topology, rooted at the data collecting center (the base station, BS). Nodes synchronize to a global communication schedule,



(a) (b) (c) (d)

Fig. 1. The data forwarding process in presence of transmission errors: a) transmission attempt and caching, b) forwarding and implicit ack, c) retransmission request, d) retransmission.

which spans over a time duration called *epoch* and repeats periodically. During one epoch, each node performs a sensor reading, aggregates data received from children nodes, and transmits the collected information within a single packet. Implicit acknowledgments are used to detect transmission failures and to start a recovery procedure. The feedback provided by implicit acknowledgments is exploited by our application, in order to adaptively adjust the adopted transmission delay.

We devised two decision algorithms: the first one changes the adopted delay when the amount of consecutive transmission failures exceeds a configurable threshold; while the second one relies on a slightly more sophisticated and flexible filtering that operates on the past few transmission results. The proposed algorithms converge to a steady global schedule of transmissions in a totally decentralized manner. This results in good scalability and proves to be effective in addressing the hidden terminal problem. Moreover the use of implicit acknowledgments, as the mechanism to detect successful transmissions, allows nodes to acquire locally relevant information while obviating the need of control packets or location information.

Despite their diversity, many WSN applications exhibit periodic traffic generation and use the same communication model, known as *convergecast*, where multiple sources generate data that are to be forwarded toward a collecting entity. Hence, the approach proposed here can be virtually applied to optimize performance to a wide variety of scenarios.

We describe the considered reference application in Section II, while Section III describes the details of the proposed collision avoidance mechanisms. Finally, Section IV presents our conclusions.

II. AN APPROACH TO DATA GATHERING WITH IMPLICIT ACKNOWLEDGMENTS AND RETRANSMISSIONS

In [3] we have presented a data gathering framework for monitoring applications in medium and large scale WSNs. The protocol we proposed operates at the Application Layer and is designed to work on top of a cluster-tree beacon-enabled IEEE 802.15.4 network. The data gathering protocol is optimized for the converge-cast traffic pattern and uses a tree-based topology, rooted at the base station. As packets traverse multiple hops, flowing from levels farther from the BS towards the center of the network, data are aggregated at each step in order to minimize transmission time and improve energy efficiency. Nodes synchronize to a periodic global schedule, made up of *phases* for specific activities such as transmission, reception and sleep, as shown in Figure 1. Each phase is implemented by combining several adjacent IEEE 802.15.4 superframes.

The distinguishing feature of this framework is that communication reliability is not provided through MAC-Layer acknowledgments and retransmissions, but relies on a strategy of caching, implicit acknowledgments, and Application-Layer retransmissions. Implicit acknowledgment is a general term indicating the capability of a node, in a broadcast communication environment, of listening to the data forwarded by upstream nodes and inferring the correct reception of its own data. In our framework, this feature is provided by the adopted aggregation functions, introduced in [4], which allow to recognize the presence of the original component data into a digest.

Figure 1 shows an example of the multi-hop communication process. Assume that node C is located at the tree level h, and has chosen node P_1 as its preferred parent; node P_2 is also within hearing distance from C. After transmitting a packet p_{data} during its `TX` phase (see Figure 1(a)), node C keeps its radio on during the `SENSING` phase in order to overhear the transmission of packet p_{digest} by its parent node P_1 (Figure 1(b)). By analyzing the digest contained in p_{digest}, node C determines that p_{data} was not correctly received, i.e. it gets a negative implicit acknowledgment. Meanwhile, node P_2 may also have received and cached packet p_{data}, so upon detection of the transmission failure, node C triggers a retransmission (Figure 1(c)); this forces node P_2 to create a new digest containing p_{data} and to forward the newly generated packet upstream toward the BS during its `TX_REC` phase (Figure 1(d)).

III. COLLISION AVOIDANCE

When dealing with cluster-tree beacon-enabled IEEE 802.15.4 sensor networks, two different collision avoidance problems are to be addressed. On one side, while the IEEE 802.15.4 specifications include a description of a cluster-tree topology which is supposed to operate in beacon-enabled mode, no details of an actual implementation are provided and the critical problem of network-wide synchronized beaconing is not discussed. Beacons are transmitted periodically and without any backoff algorithm, thus any practical implementation must ensure that nodes transmit their beacons avoiding systematic collisions. On the other side, the slotted CSMA/CA algorithm used by the IEEE 802.15.4 MAC protocol for data packets does not perform well for large scale sensor networks [5] and its performance is heavily impaired by hidden terminal issues, as it has been discussed by many authors in the recent past and as we directly observed through our own simulations.

A. Beacon Frame Collision Avoidance

The phase scheme described in Section II relies on beacon-based synchronization and the current phase is determined based on the received Beacon Sequence Numbers (BSNs). Any intermediate node receives and transmits beacons. Nodes initiate beaconing at the beginning of the `RX` phase and stop at the end of the `RX_TRIG` phase. All nodes must be able to receive beacons from their parents at least once in order to synchronize to the phases schedule. However, it is desirable that beacon reception rate does not fall below a minimum threshold in order to cope with potential clock drifts. Since the IEEE 802.15.4 MAC protocol does not use a backoff algorithm when transmitting beacons, a wrong scheduling of beacon transmission times may result in persistent collisions. Unless additional intervention is provided, some nodes may not be able to identify the current phase; they will not participate to the data gathering process, thus decreasing the overall reliability of the network. The beacon frame collision problem has been addressed as Request for Comments in the Task Group 15.4b [6] and more recently discussed in [7], where the authors proposed a scheduling based on Time Division. However, this approach is centralized and requires knowledge of node locations, thus it does not meet our goals of low complexity and decentralization.

In our system we suggest a simpler approach and we implemented two mechanisms to prevent beacons from colliding systematically. First, a node will defer the transmission of its beacons by selecting a random delay D_b relative to the reception of the beacons from its parent. D_b is determined upon the first beacon reception and it is not modified during the lifetime of the network. Secondly, since it is still possible that some nodes select delays that lead to collisions, nodes use an additional random offset $d_b \in [-T, +T]$, which changes at each epoch. The entire scheme is shown in Figure 2, where we set T as equal to the duration of the MAC backoff interval. In this way, the first beacons of two consecutive activity periods are not strictly one epoch duration away. The combined effect of the two above mechanisms allows to achieve a near one hundred percent synchronization of the nodes to the phases schedule.

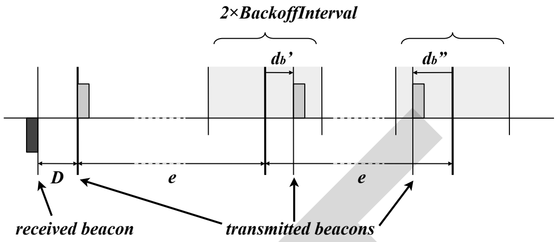


Fig. 2. Beaconing timing.

B. Data Frame Collision Avoidance

Simulation results discussed in [3] have shown how performance of IEEE 802.15.4 networks is heavily affected by the number of nodes and their density, with more populated networks experiencing larger amounts of collisions and, in the end, poorer reliability. Rather, typical applications of WSNs demand for scalability, thus some technique aiming at reducing the contention level on the wireless channel is needed.

The IEEE 802.15.4 MAC protocol uses binary exponential backoff with a collision window equal to $[0, 2^{BE}-1]$. The minimum value of the backoff exponent (BE) is controlled by the configurable parameter $macMinBE$. If $macMinBE = 0$, collision avoidance is disabled during the first iteration of the algorithm and the MAC protocol makes an attempt of transmitting outgoing packets immediately after the reception from the Application Layer. The transmission is deferred only if the channel assessment returns a busy state. Although a certain probability of many deferrals exists, these settings give the Application Layer more control on the timing of the transmissions and partially delegate to it the responsibility of avoiding collisions. Thus, the first step of our technique is to set $macMinBE = 0$.

The second step consists of the introduction of a random backoff delay D_T between the beginning of the `TX` phase and the time when the Application Layer passes the packet to the MAC Layer for transmission.

Finally, the last step is the adoption of a set of rules which determine whether to keep the current D_T, or to adopt a new random one.

In the most basic solution, D_T is determined independently for each packet, similarly to the proposal in [8]. We refer to this simple scheme as Algorithm Random. More elaborated algorithms can be devised that use the collected information concerning past successes and failures. In the following we introduce two different decision algorithms: the first one based on the count of consecutive transmission failures and the second one based on a FIR filter which processes a configurable number of past transmission results.

Algorithm 1 Consecutive Failures.

```
procedure INITIALIZE
    $TxFailCount \leftarrow 0$;
    $D_{T,old} \leftarrow random(0, MaxDelaySlots - 1)$;
end procedure

procedure SENDDATA                                        ▷ Phase TX
    if $TxFailCount <$ MAX_TX_FAIL then
        $D_T \leftarrow D_{T,old}$;
    else
        $D_T \leftarrow random(0, MaxDelaySlots - 1)$;
        $TxFailCount \leftarrow 0$;
    end if
    $D_{T,old} \leftarrow D_T$;
    if $level \geq 2$ then
        $TxFailCount \leftarrow TxFailCount + 1$;
    end if
    wait for $D_T$ MAC backoff slots;
    pass the packet to the MAC Layer;
end procedure

procedure RECVDATAFROMPARENT                              ▷ Phase SENSING
    if packet includes data transmitted during phase TX then
        $TxFailCount \leftarrow 0$;
    end if
end procedure

$level$ 1 $only$ :
procedure RECVTRIGGERFROMCHILD($child\_ID$)               ▷ Phase RX_TRIG
    if data from node child_ID transmitted during phase TX then
        $TxFailCount \leftarrow TxFailCount + 1$;
    end if
end procedure
```

Fig. 3. Pseudocode describing delay assignment through Algorithm 1.

Algorithm 2 Weighted Average.

```
procedure INITIALIZE
    $tx\_\vec{res} \leftarrow \vec{0}$;
    $D_{T,old} \leftarrow random(0, MaxDelaySlots - 1)$;
end procedure

procedure SENDDATA                                        ▷ Phase TX
    if $\sum_{i=1,\ldots,n} tx\_res \times tx\_w <$ TX_FAIL_THR then
        $D_T \leftarrow D_{T,old}$;
    else
        $D_T \leftarrow random(0, MaxDelaySlots - 1)$;
        reset $tx\_res$ to all zeros;
    end if
    $D_{T,old} \leftarrow D_T$;
    $rightShift(tx\_res)$;
    if $level \geq 2$ then
        $tx\_res[0] \leftarrow 1$;
    end if
    wait for $D_T$ MAC backoff slots;
    pass the packet to the MAC Layer;
end procedure

procedure RECVDATAFROMPARENT                              ▷ Phase SENSING
    if packet includes data transmitted during phase TX then
        $tx\_res[0] \leftarrow 0$;
    end if
end procedure

$level$ 1 $only$ :
procedure RECVTRIGGERFROMCHILD($child\_ID$)               ▷ Phase RX_TRIG
    if data from node child_ID transmitted during phase TX then
        $tx\_res[0] \leftarrow 1$;
    end if
end procedure
```

Fig. 4. Pseudocode describing delay assignment through Algorithm 2.

C. Algorithm 1: Consecutive Failures

Algorithm 1 is described by the pseudocode in Figure 3. The algorithm uses a counter for keeping track of consecutive transmission failures ($TxFailCount$) and a configurable threshold (`MAX_TX_FAIL`).

During initialization the counter is set to a null value and an initial random value for the backoff delay is determined ($D_{T,old}$).

When the `TX` phase begins, $TxFailCount$ is compared with `MAX_TX_FAIL`. If $TxFailCount <$ `MAX_TX_FAIL`, the old delay ($D_{T,old}$) is adopted as D_T. Otherwise, a new random D_T is selected. In this case the counter is also reset, because we want to test the goodness of the new D_T. Before the transmission, the adopted delay is stored and $TxFailCount$ incremented, i.e. a failure is assumed by default.

When, during the `SENSING` phase, a node overhears the packet forwarded by its parent, $TxFailCount$ is reset in case of successful implicit acknowledgment. Note that, if a node does not sense any packet, this counts as a failure and $TxFailCount$ is not changed.

Nodes belonging to level 1, i.e. direct children of the BS, cannot use implicit acknowledgment to detect failures, as the BS does not forward packets. Hence, the algorithm for these nodes is slightly different. Namely, every new transmission is initially assumed successful, and $TxFailCount$ is not incremented. A failure is inferred when a child node, whose data have been received and forwarded, requests a retransmission. In fact, since data originating from the complaining node have been transmitted, the reason for the retransmission request is likely to be a collision of the transmitted packet.

A similar algorithm, with a few exceptions for level 1 nodes, is used for the transmission of recovery packets during the `TX_REC` phase.

D. Algorithm 2: Weighted Average

The approach of Algorithm 1 can be further refined so as to enable nodes to detect bad values of the delay D_T more efficiently. Since the backoff algorithm performed by the MAC Layer still introduces a certain variability in the exact instants of transmission, we can expect that the same settings, in terms of adopted delays, may lead to different results in terms of collisions from one epoch to another. Hence situations could arise such that collisions occur at an intolerable rate, but never exceed the configured thresholds. Under these circumstances nodes do not change the adopted delays and the network keeps running with poor reliability. Of course lower thresholds could be configured to prevent such situations, but this would introduce higher potential instability, with nodes unable to find a steady schedule. To overcome these shortcomings, we considered a slightly different technique which is based on a weighted average over the last transmission results.

Algorithm 2 for transmissions during the `TX` phase, uses a binary vector ($tx\_\vec{res}$) which records the last n transmission results, with 1 indicating a failure and 0 for a success. A configurable set of real weights is stored in vector $tx\_\vec{w}$ and a threshold (`TX_FAIL_THR`) is used. The algorithm is described by the pseudocode in Figure 4.

An initial backoff value ($D_{T,old}$) is selected at initialization and the vector of results ($tx\_\vec{res}$) is filled with zeros.

During the `TX` phase, a weighted average of the values in $tx\_\vec{res}$, computed using the weights in $tx\_\vec{w}$, is compared with `TX_FAIL_THR`. In analogy with Algorithm 1, when the computed quantity is smaller than the threshold, $D_{T,old}$ is reused, otherwise, a new D_T is selected and the vector of results is reset. Afterwards, the adopted delay is stored in $D_{T,old}$ and $tx\_\vec{res}$ is right-shifted, in order to make room for the new transmission result. Since a transmission failure is assumed by default, we set $tx\_res[0] = 1$.

A node receiving a positive implicit acknowledgment during the `SENSING` phase, resets $tx\_res[0]$; while $tx\_res[0]$ remains 1 in case of negative acknowledgment or in the absence of a received packet.

The above reasoning about level 1 nodes also applies to Algorithm 2, which uses retransmission requests in order to infer collisions and set $tx\_res[0] = 1$. Again, the algorithm for packets transmitted during the `TX_REC` phase is very similar.

IV. CONCLUSIONS

The present work described an approach to collision avoidance for IEEE 802.15.4 cluster-tree networks operating in beacon-enabled mode. Our scheme exploits the periodic and synchronized exchange of data packets to setup an adaptive schedule of transmission times, which are controlled by the application by shifting the delivery of packets to the MAC Layer. We discussed two different algorithms which determine the transmission delay to be applied, based on a heuristic evaluation over recent transmission successes and failures. The adopted rules let each node manage only its own transmission times, while the whole network converges to a steady schedule of transmissions, which minimizes collisions and makes efficient use of the radio. Unlike most scheduled protocols, the proposed technique is totally distributed and it is a good candidate for dynamic environments and large scale networks. The use of the standard IEEE 802.15.4 MAC makes our approach feasible for implementation on actual devices.

Our current research efforts are focused on the design of self-configuring algorithms which automatically choose thresholds based on network conditions.

REFERENCES

[1] D. Estrin, L. Girod, G. Pottie, and M. Srivastava, "Instrumenting the world with wireless sensor networks," in *Proc. of Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP 2001)*, Salt Lake City, Utah, May 2001.

[2] I. Kurtis Kredo and P. Mohapatra, "Medium access control in wireless sensor networks," *Comput. Netw.*, vol. 51, no. 4, pp. 961–994, 2007.

[3] D. Messina, M. Ortolani, and G. Lo Re, "Achieving robustness through caching and retransmissions in IEEE 802.15.4-based WSNs," in *Proceedings of 16th International Conference on Computer Communications and Networks (ICCCN 2007)*, 2007, pp. 1117–1122.

[4] S. Nath, P. Gibbons, S. Seshan, and Z. Anderson, "Synopsis diffusion for robust aggregation in sensor networks," in *Proc. ACM SenSys*, Nagoya, Japan, Jan. 2004, pp. 290–294.

[5] A. Koubaa, M. Alves, and E. Tovar, "A comprehensive simulation study of slotted CSMA/CA for IEEE 802.15.4 wireless sensor networks," *Factory Communication Systems, 2006 IEEE International Workshop on*, pp. 183–192, June 27, 2006.

[6] *IEEE 802.15 WPAN Task Group 4b*. http://grouper.ieee.org/groups/802/15/pub/TG4b.html.

[7] A. Koubaa, A. Cunha, and M. Alves, "A time division beacon scheduling mechanism for IEEE 802.15.4/Zigbee cluster-tree wireless sensor networks," *Real-Time Systems, 2007. ECRTS '07. 19th Euromicro Conference on*, pp. 125–135, 4-6 July 2007.

[8] I. Chatzigiannakis, A. Kinalis, and S. Nikoletseas, "Wireless sensor networks protocols for efficient collision avoidance in multi-path data propagation," in *PE-WASUN '04: Proceedings of the 1st ACM international workshop on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks*. New York, NY, USA: ACM, 2004, pp. 8–16.

Daniele Messina received his Laurea degree in Electronic Engineering in2004 from the University of Palermo, and he is currently a PhD candidate at the Department of Computer Engineering of the same university. He has been a research collaborator for projects in the area of Wideband Wireless Communications, where he worked on the design of MAC protocols and Radio Resource Allocation Algorithms. Since he joined the Network Research Group at the Department of Computer Engineering, in January 2006, his activity has been focused on Wireless Sensor Networks, including ZigBee-IEEE 802.15.4 Wireless Personal Area Networks.

Marco Ortolani received his Laurea degree in Computer Engineering in 2000 from the University of Palermo, Italy, and his Ph.D. degree in Electronic, Computer, and Telecommunications Engineering from the same university in 2004. Since June 2005 he is the holder of a Research Grant on "Optimization and security in Wireless Sensor Networks" and he joined the Network Research Group at the Dept. of Computer Engineering at the University of Palermo. He has participated to several research projects at a local and national level, regarding the development of robust data gathering protocols for WSNs and the application of Wireless Sensor Networks to structural, and agricultural monitoring. His current research activities regard distributed data analysis and networking.

Giuseppe Lo Re received his Laurea degree in Computer Science at the University of Pisa, Italy, and the Ph.D. degree in Electronic, Computer, and Telecommunications Engineering at the University of Palermo, Italy, in 1990 and in 1999, respectively. In 1991, he joined the Italian National Research Council, where he achieved the Senior Researcher position. In 1994 he was a Visiting Scientist at the IBM European Networking Center, Heidelberg, Germany, and in 1999 at the International Computer Science Institute, Berkeley USA. In December 2004, he joined University of Palermo, where he currently is an Associate Professor of Computer Engineering and responsible of the Networking Laboratory at the Department of Computer Engineering. He is author of several papers published in international journals and conference proceedings, in the area of networking. His current research interests are in the area of computer networks, wireless sensor networks, and distributed systems.