QOS AND SECURITY IN ENERGY-HARVESTING WIRELESS SENSOR NETWORKS

Antonio Vincenzo Taddeo, Marcello Mura and Alberto Ferrante
ALaRI, Faculty of Informatics, University of Lugano, via G. Buffi 13, Lugano, Switzerland

Keywords: Security, priority, Wireless sensor networks, Quality of services, Energy harvesting.

Abstract: Wireless sensor networks are composed of small nodes that might be used for a variety of purposes. Nodes communicate together through a wireless connection that might be subject to different attacks when the network is placed in hostile environments. Furthermore, the nodes are usually equipped with very small batteries providing limited battery life, therefore limited power consumption is of utmost importance for nodes. This is in clear opposition with the requirement of providing security to communications as security might be very expensive from the power consumption stand point. Energy harvesting methods can be used to recharge batteries, but, in most of the cases the recharge profile cannot be known in advance. Therefore, nodes might face periods of time in which no recharge is available and the battery level is low.
In this paper we introduce an optimization mechanism that allows the system to change the communication security settings at runtime with the goal of improving node lifetime, yet providing a suitable security level. The optimization mechanism further improves energy consumption by putting in place a quality of service mechanism: when energy is scarce, the system tends to send only essential packets. As shown by the simulations presented in this paper, this mechanism optimizes the energy consumption among different recharges.

1 INTRODUCTION

Wireless Sensor Networks (WSN) are composed of a large number of nodes with sensing, processing, and data communication capabilities. Recent improvements in microelectronic technologies provide the ability to create small and inexpensive nodes with limited power consumption. In a typical application scenario, nodes are provided with a local power source which is usually limited and non-replaceable; thus, power consumption requirements represent the utmost constraint for most WSN nodes (Akyildiz et al., 2002). Typically, sensor nodes proper communicate over limited distances with a "sink" - usually acting as gateway to some remote system - which may have less stringent power requirements as it might be equipped with better power sources.

Recently, in (Alippi et al., 2008) it was observed that, in real usage scenarios in which WSN are used for monitoring the environment, nodes cannot be powered just by using small batteries. Energy Harvesting becomes an essential feature in this as well as in other contexts where small batteries cannot provide the required power for the required time. Different techniques have been explored to harvest energy from the environment (Roundy et al., 2004); so far, solar energy has been the most exploited energy harvesting resource as the technology is sufficiently mature to provide a suitable quantity of energy for wireless applications. Though, the use of solar energy introduces a further level of uncertainty in the amount of energy available to the system. Energy consumption is, in any WSN, partially deterministic and partially non-deterministic (e.g. due to random components of the wireless protocol and asynchronous alarms). The recharge rate given by a solar source superposes a natural deterministic feature (i.e. day/night, seasons) to a dominant non-deterministic stochastic nature (weather conditions) as well noted in (Moser et al., 2007a). In particular, the length of the periods in which the insulation is insufficient to guarantee nodes operation is not known. For this reasons, defining strategies to optimize network operations in such conditions is of primary importance research field.

Another fundamental factor influencing the design and performance of sensor networks concerns security of their communications. Security services such as authentication, confidentiality, and availability becomes especially critical for WSNs operating in hostile environments and managing potentially sensible

Vincenzo Taddeo A., Mura M. and Ferrante A. (2010).
QOS AND SECURITY IN ENERGY-HARVESTING WIRELESS SENSOR NETWORKS.
In *Proceedings of the International Conference on Security and Cryptography*, pages 241-250
DOI: 10.5220/0002982202410250

data. Generally, in order to enforce communication security, additional computational resources are required (Chandramouli et al., 2006): typically, packet header sizes are incremented, thus inducing a significant increase in the energy spent for communications (Mura et al., 2008). Traditional security solutions are designed by using ad-hoc approaches which offer specific protection against certain attacks. However, they rely on the assumption that the operative environment is well-known and essentially static. Moreover, some of these technologies have not been specifically developed for embedded systems; in many cases, their adoption in the pervasive world would be impossible due to high hardware resources requirements (Ferrante et al., 2005).

In practice, when designing secure systems, the worst case scenario is considered: the system has to guarantee adequate protection against the strongest possible security attacks. This is generally in contrast with the typical requirements of resource-constrained devices: mobility, flexibility, real-time configuration, open and dynamic operative environment (Keeratiwintakorn and Krishnamurthy, 2006; Großschädl et al., 2007). In the work presented in this paper we adopted a new approach to security by relating it to the current system conditions. The best possible security solution, specified within a range of possible choices, is chosen dynamically to optimize both security and lifetime of the system. As discussed in (Chigan et al., 2005; Lighfoot et al., 2007), evaluating run-time the trade-off between security and consumed energy is not straightforward.

Furthermore, in some scenarios, nodes might require the ability to guarantee the delivery of critical data packets even in presence of scarce resources. For this purpose, a network Quality of Service (QoS) mechanism may be implemented in the nodes. In our case, the scarce resource is not network bandwidth as in the large majority of cases in which QoS is adopted. Instead, in this context, the scarce resource is energy: when available energy is low, only essential packets should be delivered to preserve the system main functionalities, in case a solar recharge is not available. To this goal, we associate with each packet a priority level and propose a run-time mechanism to manage security and QoS inside WSN nodes equipped with solar panels. The mechanism proposed in our approach provides the unique ability to optimize the trade-off between consumed power and security while providing support for QoS. In our approach, security is being adapted to the energy conditions of the system. At the same time, high-priority packets are processed faster and with higher security.

In Section 2, the related works are presented and the innovation of our work is highlighted. We introduce the main parameters that we considered in the design of our solution and provide details regarding the trade-off mechanism in Section 3. A case study to prove the validity of our approach as well as the simulations results are outlined in Section 4.

2 RELATED WORK

This study targets the problem of deciding which level of service should be guaranteed by a WSN under energy recharging conditions. Various task scheduling policies suitable for energy harvesting WSNs nodes were studied in (Moser et al., 2007a; Moser et al., 2007b); in (Moser et al., 2008) algorithms for maximizing a function of merit of the devices are proposed. Appropriate voltage/frequency levels selection depending on the available energy is studied in (Liu et al., 2008) and in (Liu et al., 2009). Adaptivity by means of setting different reliability levels depending on the available energy is presented in (Wang et al., 2009). Most current studies discuss this subject with the classical approach used for scheduling tasks on a microprocessor: packets are considered as tasks and their schedulability is evaluated by substituting energy to CPU time.

We modify this classical approach by introducing a QoS management mechanism (Sean Convery, 2004) similar to the ones commonly used in conventional networks. Packets might be subdivided into different categories, each one with a different “importance”. More critical packets (i.e., the ones classified with high priority) are prioritized over the others, thus guaranteeing to them an higher probability of being delivered even in presence of scarce resources (i.e., a low battery level).

Concerning security aspects, the problem of optimizing resources used for security, yet providing an adequate level of protection, is an hot topic at the moment (Ravi et al., 2004). In particular, the trade-off between energy and performance requirements of security solutions is of utmost relevance for embedded systems (Chandramouli et al., 2006). Each adopted security solution should be a good compromise between factors that are conflicting in nature such as, for example, power consumption and performances. This optimization is a complex task, especially when performed at run-time (Chigan et al., 2005; Lighfoot et al., 2007). With respect the classical approaches for securing WSNs, our solution is able to dynamically adapt the security settings based on current node energy conditions and according to specific security requirements. Furthermore, it provides the highest se-

curity for high-priority packets.

3 SECURITY AND QOS MANAGEMENT

In this paper we considered a WSN in which nodes periodically send packets to a sink destination. Packets to be delivered may have security requirements that, depending on the operative context, the system might or might not be able to satisfy. For example, the battery level might be too low to use certain security settings. Therefore, in order to assure a high number of packets transmitted, a change in the security provided has to be applied. Moreover, delivery of critical packets has to be guaranteed: a priority value is used to mark each packet and to provide such a QoS. Priorities should be carefully assigned to packets at system configuration time: packets that are essential for the network, for example, should be associated to highest priority levels as low-priority packets may be delayed indefinitely.

Our adaptation mechanism manages QoS and changes the security settings dynamically to maximize the trade-off between the number of packets sent (with priority precedence) and their security. Such optimizations require to estimate the energy consumption connected with the security processing and the transmission of packets. The security and QoS management process that we propose is based on a set of optimization strategies. The optimizations to be applied depend on a set of parameters related both to the system status and to the characteristics of data packets to be transmitted.

In this section we first describe the considered parameters; then we introduce the optimization process that we have designed.

3.1 Parameters

The system parameter considered is the available energy $E_{available}$; it includes the contribution of the energy recharge E_{harv} due to the harvesting system.

The characteristics of data packet considered are both related to intrinsic packet characteristics (e.g., their length) and to associated parameters such as their importance (expressed as a priority level) and their required security.

3.1.1 Energy Model

Energy consumption data can be collected only after the corresponding activity happens. Therefore, we introduce a prediction model that gives an indication of the foreseen power costs for packet transmission. The actual consumption is collected after corresponding packets have been sent and is used to have a more precise estimate of available energy.

In most networking protocols a boundary on the amount of time and energy necessary to ensure that a given packet is transmitted does not exist. For example, by using a CSMA algorithm, there may be cases in which the access to the channel cannot be obtained due to channel jamming. For this reason, an average energy for transmitting the packet should be considered instead of the worst case one. The penalty for transmitting multiple times a packet because of errors on the channel is very strong. For this reason our average takes into consideration a number of retrials that depends on the state of the network.

We model the energy needed to process a packet as:

$$E_{packet} = E_{tx} + E_{errors} \quad (1)$$

where, E_{tx} is the energy required to transmit a packet assuming that the channel is always free. Instead, E_{errors} takes into account the energy spent for multiple executions of the CSMA algorithm that are necessary when the channel is found to be busy or when packets need to be retransmitted due to collisions. E_{tx} is composed of: the energy consumed by the CSMA algorithm, E_{csma}; the energy spent for sending the packet, considered as the sum of the energies necessary to send the header of the packet (i.e. E_{header}), the possible increase in the overhead of the packet due to security (i.e. E_{ovhsec}) and the energy necessary for sending the payload $E_{payload}$. That is:

$$E_{tx} = \left(E_{csma} + E_{header} + E_{ovhsec} + E_{payload}\right) \quad (2)$$

In Table 2 is reported the security overhead considered (Mura et al., 2007). E_{header}, as well as E_{ovhsec} and $E_{payload}$ depend on the size of the corresponding parts of the packet and on the energy per byte e_b (i.e. transmission power/throughput in bytes). Our methodology does not involve any modification of the header different than that modeled by the security component. As a matter of fact, depending on the architectural choices for the node platform, there may also be a relevant contribution to energy budget due to execution of encryption/authentication algorithms as shown in (Mura et al., 2008). In this paper, we refer to node platforms that process security through HW co-processors in which such contribution is negligible.

Following the analysis of the 802.15.4 algorithm done in (Mura et al., 2007) the other contributions can be refined as:

$$E_{csma} = mode * (E_{idle2rx} + E_{cca}) + E_{twait} \quad (3)$$

Therefore, E_{csma} can be decomposed into the sum of the energies to switch the radio from the idle to the receiving mode ($E_{idle2rx}$) and the energy consumed while receiving the CCA (E_{cca}). These operations are executed once in beaconless mode ($mode = 1$) and twice when beacons are used ($mode = 2$). E_{twait} is the energy consumed, in idle state, while waiting the random time before performing the CCA; we used the worst case value for this parameter.

E_{errors} is composed of two factors:

$$E_{errors} = E_{busy} + E_{re\_tx} = \quad (4)$$

$$= \frac{\left(K_{busy} * E_{csma} + N_{re\_tx} * E_{tx}\right)}{M_{sent}} \quad (5)$$

the first is the energy spent for executing multiple times the CSMA procedure when the CCA reports that the channel is busy, hence it equals K_{busy} times the energy of a CSMA algorithm. The latter is the energy consumed for retransmitting the entire packet due to channel collisions, that is N_{re_tx} times E_{tx}. Values are meant by the number of packets successfully sent M_{sent} in a transmission time slot. Values of these parameters, K_{busy}, N_{re_tx} and M_{sent} has to be computed based on the applicative scenario. They can assume a constant value, if the channel behavior is quite static; or can be computed at run-time, for example, by monitoring the channel conditions and updating proper counters.

Every time a packet is processed and transmitted, the corresponding energy consumption E_{packet} is drawn from the available energy $E_{available}$.

Finally, we introduce an *energy constraint* E_{frame} which is a discrete time derivative of the current energy consumption $E_{available}$:

$$E_{frame} = E_{available}(t) - E_{available}(t-1) \quad (6)$$

This constraint is used to assures that, for each transmission time slot, the variation of the energy required to process a certain number of packets is below the specified threshold E_{frame}. Thus, it is a hard constraint in the maximum energy consumption allowed in a transmission slot. The value for E_{frame} can be either constant, (and, therefore, assigned statically at design time) or computed dynamically as a function of the energy currently available. In general, the value assumed by this parameter should be tuned according to the considered scenario.

3.1.2 Packet Characteristics

In our system each packet, with a given payload size, is labeled with an identifier $P_{\#}$, it is associated to security requirements as well as to a priority level. These pieces of information are only used by the optimization mechanism and they do not cause any network overhead as they are local parameters. An example is shown in Table 1.

Table 1: Packet schema.

Packet #	102	*SecReq*	
Size	Priority	SecSuite	ActiveSuite
[*bytes*]	[0,3]	(8*bit*)	[0,7]
`50`	`3`	`[1,0,1,0,1,1,0]`	`5`

Network messages can have different importance, ranging from essential information, either for network maintenance or application scope, to low-significance information that can be dropped with limited impact. A *priority* ρ can have values between 1 and 4 (the higher the number, the higher the priority). As mentioned earlier, when energy is scarce, our system will favour high-priority packets. Low priority packets will be delayed indefinitely or discarded if the memory becomes full. The aim of this approach is to preserve as much energy as possible to send necessary packets. Non-essential packets might be sent later when harvesting allows the system to recover the battery level.

Security requirements associated to packets are a central parameter in our trade-off mechanism. The security level chosen for each packet impacts both on the associated consumed energy and on the associated computational and network latencies. A secure packet, due to longer packet header and security processing, consumes more energy and takes more time to be managed by the network then a non-secure one (Mura et al., 2008). While some packets have mandatory security requirements, others may have softer requirements. Therefore security of such packets can be decided at runtime depending on the current status of the system.

We defined a *security requirements*, *SecReq*, as composed of:

$$SecReq = \langle SecSuite; ActiveSuite \rangle \quad (7)$$

SecSuites is a list of security suites that may be used for the particular packet. *SecSuites* is a bitstream in which every bit corresponds to a different security suite supported by the transmission protocol, the bit is set to 1 if it is possible to use that suite for the particular packet. For example, if we refer to the security suites of the IEEE 802.15.4 wireless standard (Sastry and Wagner, 2004), as reported in Table 2, we need 8 bits to model the entire set.

ActiveSuite identifies the security suite to be used for processing the given packet. If a change in the security provided is required, the *ActiveSuite* is updated to the new security suite ID. Changes in *ActiveSuite* are allowed within the packet security suites bitstream. Therefore, to have a fixed security suite only one bit should be set in *SecSuites*

Table 2: Security suites supported by IEEE Std 802.15.4 and their packets overhead (Mura et al., 2008).

Security Suites			
SuiteID	Description	Service	Overhead [*byte*]
0	Null	No Security	0
1	AES-CBC-MAC-32	Authentication	9
2	AES-CBC-MAC-64		13
3	AES-CBC-MAC-128		21
4	AES-CTR	Encryption (only)	5
5	AES-CCM-32	Authentication and Encryption	9
6	AES-CCM-64		13
7	AES-CCM-128		21

list. This corresponds to a mandatory security requirement given by, for example, the following tuple: $\langle SecSuite = [0,0,0,1,0,0,0]; ActiveSuite = 4\rangle$. *ActiveSuite* is the index of the 8 bit array of *SecSuite*.

An example of definition of softer packet *SecReq* is shown in Table 1; the packet with label 102 has a priority value 3 and may support different suites for data protection: 0, 2, 4 and 5. The current active suite is the number 5, that is the suite AES-CCM-32 of Table 2. If an adaptation is needed, the system may decide to use a different security suite (e.g. $ActiveSuite = 4$) to save energy (because of lower header size overhead).

3.2 Optimization Process

By using the aforementioned parameters our management algorithm optimizes the security of packets while managing QoS and optimizing energy consumption. An optimization strategy is applied by the optimization mechanism to obtain these results.

3.2.1 Optimization Strategies

Different optimization strategies can be defined for different system conditions. An optimization strategy ($\wp$) can be defined as a composition of actions to be performed in order to meet the energy constraint given by E_{frame}.

We designed the strategies with two goals in mind: to maximize the number of high-priority packets that are delivered and to ensure that security requirements of each packet are satisfied. These strategies are optimized to provide to each packet the safest possible suite among the ones specified in *SecSuite*.

We identify the following actions that can be combined within a strategy $\wp$:

- change the *ActiveSuite* security suite used to protect the considered packet, according to the suites specified in *SecSuite*;
- either drop or delay the packet transmission to the next communication slot, according to the priority of the considered packet;
- limit the number of packets to be sent. Such a limitation can be imposed both on predetermined priorities levels or globally on all packets.

QoS may be provided through one of the well known approaches listed in (Sean Convery, 2004), for example by applying *Weighted Fair Queueing* (WFQ) and *Priority Queueing* (PQ).

By combining the aforementioned actions in a strategy, the system can directly change the energy consumption needed to process the packets that are in the send queue with the goal of satisfying the energy constraint given by E_{frame}.

For example, a possible strategy to meet such a constraint may be as follows:

- low priority packets are delayed to successive communication slots;
- other, less energy hungry, security suites are chosen for the remaining packets in queue;
- other packets may be selected for being delayed.

3.2.2 Optimization Mechanism

An adaptation is required when the amount of energy required to transmit λ packets is above the threshold E_{frame}. The adaptation is performed by enforcing a certain strategy $\wp$. The process has been designed by following the Monitor-Controller-Adapter loop (Derin et al., 2009). In the monitoring phase, the available energy capacity $E_{available}$ and the λ packets to be delivered to a sink node are measured every transmission time slot. The available energy includes the contributions of the harvested energy E_{harv} obtained through solar cells. Depending on the current energy conditions, it exists a given energy constraint E_{frame} to be respected. Moreover, network conditions are monitored by updating the counters of the Equation 4 (N_{ri_tx}, K_{busy} and M_{sent}) at each transmission time slot.

All these monitored information are propagated to the controller module that decides which packets to send and their associated security suites. These information are then propagated to the adapter that organizes packet transmission according to what decided by the controller. Packets are then sent by using conventional network stack.

The controller uses the energy model introduced in Section 3.1.1 to estimate the energy per packet E_{packet}, and the total energy consumption of the candidate packets. This estimation is used to select packets according to a strategy $\wp$.

A further constraint on the candidate packet selection might be imposed by the current network conditions: for example, it can be set an higher bound for the number of packets that can be transmitted.

In Algorithm 1 we show an example of a selection mechanism used within the controller module to select the optimal packets.

3.3 Security Considerations

Aim of our framework is to provide an adaptive mechanism to satisfy system energy constraints, yet providing adequate protection to the applications. The use of multiple cryptographic algorithms may lower the security of applications. In particular, when weakest algorithms are used the transmitted data are more exposed to attacks. Though, our approach tries to provide a reasonable solution for those situations in which device constraints would not give the possibility to provide any (or little) security to applications.

Aim of our self-adaptation mechanism is to provide the highest possible security level in any instant of time. Degradation of security is only performed if the energy constraints of the system cannot be satisfied. In this particular context, lowering the security of the communication increases the potential of attacks only for a limited quantity of data (i.e., just the ones that are being transmitted in those periods of time).

Possible attacks in which the system can be forced to decrease the security level of applications may also affect our security degradation method. For example, multiple communications requests can be used to drain the battery, thus forcing the system to degrade security. Though, we should consider that our framework performs adaptation of security level based on application security requirements. Therefore, applications are always guaranteed to have the minimum level of protection that they require.

Similar attacks can be used to cause a denial of service by totally draining the battery. Our management of security allows the system to stand longer to these attacks, even though it does not provide a specific protection for them. The best defense against this kinds of attacks would be to have an intrusion detection system installed on the nodes. Though, running such a system, even if simplified, would be too expensive in terms of computational power, memory, and consumed energy.

4 EXPERIMENTAL RESULTS

To evaluate the optimization method described in this paper we considered a case study and simulated it. The case study represents a simple yet realistic case and allows evaluating performance of our optimization algorithm.

In this section we provide a description of the considered case study; we then describe the simulations we have performed and we show the results we have obtained.

4.1 Case Study

We consider as a case study a wireless sensor network composed of 7 nodes. The optimization algorithm is applied at node level, thus, its efficiency does not depend on the number of nodes. Each node is equipped with a digital camera, and it acquires an image every 30s and sends it to the central point. Every picture is composed of 160 packets of 90 bytes. Though, it should be noticed that the actual dimension of packets depends on the security suite adopted. The application is data intensive for a WSN, having a global application throughput of about 35kbps. This is a classical scenario in many application fields (e.g. surveillance, traffic monitoring, environment monitoring).

Information are usually uniformly spread in pictures. For this reason we divided each picture into five segments. Data of each segment are subdivided into network packets. All the packets belonging to the same segment share the same priority level. Priorities are assigned to each segment by following an uniform distribution. Moreover, each packet may then have different security requirements depending on its importance. Security requirements of packets (i.e., *SecSuite* bitstream) are taken from a binomial distribution. Consumption of the packets to be delivered is estimated according to the model specified in Section 3.1.1. In particular, by considering data collected during previous simulations, we estimate a value of $K_{busy} = 64$, $N_{re\_tx} = 32$ and $M_{sent} = 160$ packets for the parameters of Equation 4.

In this scenario, we considered the IEEE 802.15.4 protocol in Beaconed Mode (i.e., $mode = 2$ in Equation 3). We simulated a star topology, therefore every communication passes through the coordinator. We assume that the coordinator is connected to a power line; the other nodes have limited energy capacity and they are able to harvest solar energy. Considering the fact that charge/discharge cycles are harmful for chemical batteries, we suppose to use a supercapacitor instead of the battery. While this improves time-invariance of the power section, the energy con-

tained in the ultra-capacitor is about one order of magnitude less than the one of a commercial battery. We considered our power section made of a small solar cell that produces at peak about 300*mA*, and a super-capacitor of 310*Farad* with slightly more than 400.000*mAs* available. This correspond, for example, to the adoption of a Maxwell BCAP0350 E270 T9 SuperCap. The consumption of the digital camera was considered to be 25*mAs* per picture. E_{frame} has been considered to be a constant value of 3.5*mAs* plus a percentage (0.09%) of the total energy capacity, meant by the number of frames. This value has been chosen to provide a system lifetime, between battery recharge, of roughly 4 days.

Considering the above scenario, in conditions of peak solar power, the super-capacitor is fully recharged in less than half an hour.

4.2 Simulations

The case study discussed above has been simulated to verify the effectiveness of our approach. Results show that, while our method does not allow for an absolute decrease of consumed energy, it allows for an optimization of the trade-off among consumed energy, importance of the packets sent, and their security level.

In our case study we adopted the optimization algorithm shown in Algorithm 1. We assumed an approach to QoS similar to Priority Queueing (PQ). With PQ, higher-priority packets are transmitted before lower-priority ones, thus guaranteeing that an higher number of high-priority packets are delivered even in conditions of scarce energy. We considered the following optimization strategy $\wp$: by default, the active security suite for all packets is the $SuiteID = \#0$ (i.e. *NULL*); when the energy consumption of the high-priority packets is below the threshold, their security is increased according to packet security requirements. After the security upgrade, the remaining packets in queue are analyzed in order to consume the residual energy. Instead, if the energy consumption of the high-priority packets is above threshold, the most energy-hungry packets are removed from the list of candidates.

When solar cells are active our optimization algorithm is set to work as if the batteries were fully charged. In this condition our algorithm is at least not worse than any static setting for security. In fact, in this condition, our optimization algorithm will select the highest security algorithm specified for each class of packets. For this reason in the simulations we focused only on a scenario in which there is no solar recharge for a long period of time, for example due to clouds.

Algorithm 1. $sel\_pos = \text{SELECTOR}(E_{frame}, sub\_set)$.

Require: An energy constraint $E_{frame} \geq 0$.
Require: Set of packets (indexes) to analyze, $sub\_set \neq 0$.
Ensure: The optimal set of packets (indexes) to send sel_pos.

```
1:  pri_pos = HIGHPRICANDIDATES(queue, sub_set)
2:  E_packet = GETPACKETSENERGY(queue, pri_pos)
3:  E_residual = E_frame
4:  E_consumption = ∑E_packet
5:  sel_pos = [ ]
6:  if min(E_packet) > E_frame | sel_pos == [ ] then
7:     return sel_pos
8:  else
9:     if E_consumption < E_frame then
10:       sel_pos = pri_pos
11:       E_residual = E_residual − E_consumption
12:       if CHECKSECUPGRADE(E_residual, pri_pos)
          then
13:          E_residual = UPGRADESEC(pri_pos)
14:       end if
15:       sel_trail = SELECTOR(E_residual, TRAIL(pri_pos))
16:       return sel_pos = sel_pos + sel_trail
17:    else
18:       new_pos = REMOVEMAXENERGY(pri_pos)
19:       sel_pos = SELECTOR(E_residual, new_pos)
20:    end if
21:    return sel_pos
22: end if
```

4.2.1 Simulation Environment

The case study has been simulated by using the SystemC network simulator described in (Mura and Sami, 2008). This simulator is capable of emulating node and network operations as well as annotating with μs precision the corresponding power consumption for all the nodes involved. Furthermore, the simulator is capable of managing channel contention and to repeat transmissions when interferences occur.

The simulator is based on an implementation independent model (Mura et al., 2007) that can be later characterized for practical implementations. The characterization includes the substitution of the actual consumption values for the different activities of the node (e.g., reception and transmission). In (Mura et al., 2007) it is shown that the power consumption obtained with the simulator is few percent different from the actual data of consumption gathered from real nodes through an oscilloscope. In order to characterize our implementation independent model, we considered that the devices are equipped with the CC2420 radio with a $0dBm$ transmission power and that they use corresponding power levels.

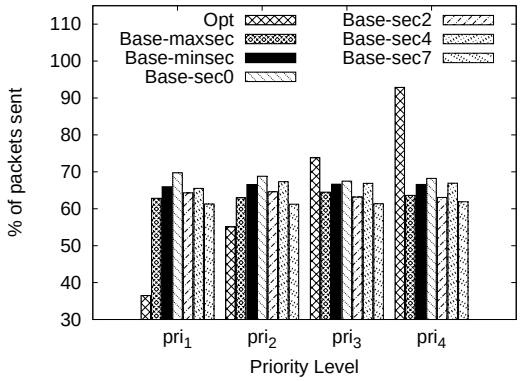

Figure 1: Comparison of average percentage of packets sent for each priority value.

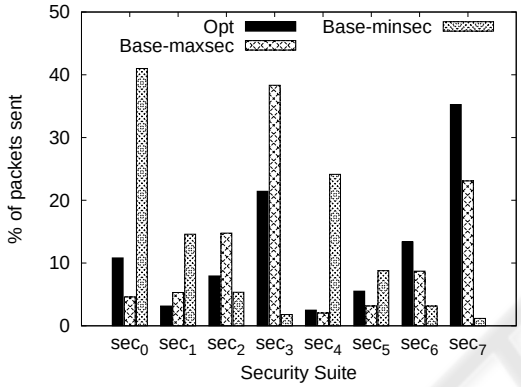

Figure 2: Comparison of average percentage of packets sent for each security suite.

4.2.2 Results

The impact of our optimization method is highlighted by comparing the same scenarios with and without applying the optimizations. In case of no limitations on energy consumption and with no optimization, the system has an operative lifetime of about three days

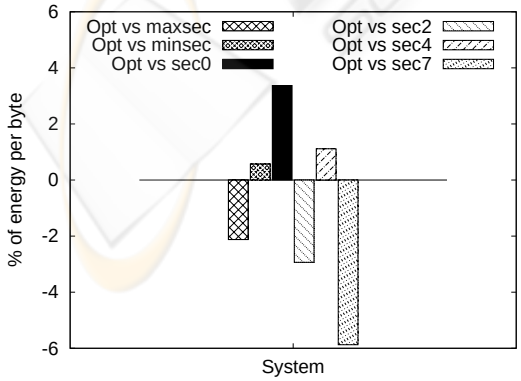

Figure 3: Comparison of percentage variation of energy per byte consumed by the optimal system w.r.t the Base System.

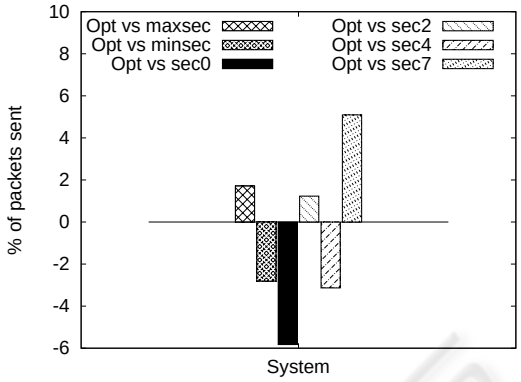

Figure 4: Comparison among the number of packet sent during the simulation time by the optimal system with the ones sent by the non-optimal systems.

when non energy harvesting is possible. When proper constraints on the consumed power are set (E_{frame} of Equation 6), the lifetime in the same conditions is increased to four days. In this paragraph we focus on the comparative analysis of various cases in which energy constraints are set.

Six different configurations have been simulated for comparing different possible situations, one of them in which the optimization methodology was used (*Opt*), and the other five (*Base-**) in which it was not. The last five considered cases are as follows:

- security is the maximum in the *SecSuite* range specified for each packet (*Base-maxsec*);
- security is the minimum in the *SecSuite* range specified for each packet (*Base-minsec*);
- security is at level *0* for all packets (*Base-sec0*);
- security is at level *2* for all packets (*Base-sec2*);
- security is at level *4* for all packets (*Base-sec4*);
- security is at level *7* for all packets (*Base-sec7*);

Figure 1 shows the percentage of packets of each priority level that the node have been able to send (remaining packets are discarded so to respect energy constraints) in the simulation period when different system configuration are considered. The figure shows that, while a non-optimized node does not guarantee any privilege to high-priority packets, the optimized node does. When no QoS is considered, each packet has the same possibility of being discarded (about 30-40% depending on the case considered). When QoS is used, instead, the probability to be delivered for each packet is proportional to its priority. Highest priority packets are delivered more than 90% of the times; lowest priority packets are delivered about 35% of the times. Obviously, the adoption of a QoS mechanism does not change the total number of packets that are sent, it only changes the dis-

tribution of these packets among the different priority levels.

Figure 2 compares the number of packets sent for each security level when the optimized system and the *Base-maxsec* and *Base-minsec* cases are considered. As it can be seen from the figure, the optimized node tends to maximize the number of packets that are processed by using the security suites number *3* and *7* which are to be considered the most secure ones available. Thus, our optimization system tends to guarantee an higher security level to the packets.

Figure 3 compares the energy per byte spent by the optimized system with the energy spent in all the cases in which the optimization is not used. The optimization allows the system to spend less energy per byte then in the *Base-maxsec* case. It also provides the ability of saving energy with respect to the cases in which security level *2* and security level *7* are always selected (*Base-sec2* and *Base-sec7*). The *Base-minsec* case is of course less energy hungry, but, at the same time, it provides, in average, a lower level of security to packets. The same applies to the *Base-sec4* case.

Figure 4 compares the number of packets sent by the optimized system during the simulation period with the ones sent by the non-optimized systems. The optimized system is able to send an highest number of packets with respect to more energy hungry modes. Obviously, less energy hungry modes are able to send more packets before finishing the battery.

Summarizing, the results show how our optimization method provides QoS management while obtaining the highest possible level of security that is compatible with the current system parameters.

It should be noticed that our results report the energy consumption of the whole node, even including the sensing part. It is well known that, in these conditions, the energy spent for transmission is not the predominant part of the total energy consumption. Indeed, the portion of energy devoted to communication (not considering the energy used by the infrastructure) is about 10% of the total consumed energy. For this reason the differences in performances among different cases might appear limited, even if they are not. In fact, they are significant if we focus our analysis only to energy related to data transmission.

5 CONCLUSIONS AND FUTURE WORK

In this paper we proposed a novel adaptation mechanism to deal with secure and priority-based transmission of packets in WSNs. This optimization mechanism allows the system to survive long periods in which the energy harvesting adopted might not be able to provide energy. The algorithm, by performing QoS management, provides the ability to privilege important (i.e., high priority) packets when the energy available is scarce. Security settings are changed dynamically to provide the best security compatible with the current system conditions.

Future work include refinement and extension of the methodology proposed and, in particular, the addition of new capabilities such as, for example, changing dynamically the duty cycle of the network and/or the monitoring period.

ACKNOWLEDGEMENTS

This work was partially supported and funded by the Hasler Foundation under the Project AETHER+ (Grant No. 09083-2043-021). The paper reflects only the view of the authors; the Hasler Foundation is not liable for any use that may be made of the information contained herein.

REFERENCES

Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., and Cayirci, E. (2002). Wireless sensor networks: a survey. *Computer Networks*, 38(4):393–422.

Alippi, C., Camplani, R., Galperti, C., and Roveri, M. (2008). Effective design of wsns: From the lab to the real world. In *Sensing Technology, 2008. ICST 2008. 3rd International Conference on*, pages 1–9.

Chandramouli, R., Bapatla, S., Subbalakshmi, K. P., and Uma, R. N. (2006). Battery power-aware encryption. *ACM Trans. Inf. Syst. Secur.*, 9(2):162–180.

Chigan, C., Li, L., and Ye, Y. (2005). Resource-aware self-adaptive security provisioning in mobile ad hoc networks. In *Wireless Communications and Networking Conference, 2005 IEEE*, volume 4, pages 2118–2124Vol.4.

Derin, O., Ferrante, A., and Taddeo, A. V. (2009). Coordinated management of hardware and software self-adaptivity. *Journal of Systems Architecture*, 55(3):170 – 179. Challenges in self-adaptive computing (Selected papers from the Aether-Morpheus 2007 workshop).

Ferrante, A., Piuri, V., and Owen, J. (2005). IPSec Hardware Resource Requirements Evaluation. In *NGI 2005*, Rome, Italy. EuroNGI.

Großschädl, J., Szekely, A., and Tillich, S. (2007). The energy cost of cryptographic key establishment in wireless sensor networks. In *ASIACCS '07: Proceedings of the 2nd ACM symposium on Information, computer and communications security*, pages 380–382, New York, NY, USA. ACM.

Keeratiwintakorn, P. and Krishnamurthy, P. (2006). Energy efficient security services for limited wireless devices. In *Proc. 1st International Symposium on Wireless Pervasive Computing*, pages 1–6.

Lighfoot, L., Ren, J., and Li, T. (2007). An energy efficient link-layer security protocol for wireless sensor networks. In *Electro/Information Technology, 2007 IEEE International Conference on*, pages 233–238.

Liu, S., Qiu, Q., and Wu, Q. (2008). Energy aware dynamic voltage and frequency selection for real-time systems with energy harvesting. In *DATE '08: Proceedings of the conference on Design, automation and test in Europe*, pages 236–241, New York, NY, USA. ACM.

Liu, S., Wu, Q., and Qiu, Q. (2009). An adaptive scheduling and voltage/frequency selection algorithm for real-time energy harvesting systems. In *DAC '09: Proceedings of the Design and Automation Conference*, pages 782–787.

Moser, C., Brunelli, D., Thiele, L., and Benini, L. (2007a). Real-time scheduling for energy harvesting sensor nodes. *Real-Time Syst.*, 37(3):233–260.

Moser, C., Chen, J.-J., and Thiele, L. (2008). Reward maximization for embedded systems with renewable energies. In *Embedded and Real-Time Computing Systems and Applications, 2008. RTCSA '08. 14th IEEE International Conference on*, pages 247–256.

Moser, C., Thiele, L., Brunelli, D., and Benini, L. (2007b). Adaptive power management in energy harvesting systems. In *DATE '07: Proceedings of the conference on Design, automation and test in Europe*, pages 773–778, San Jose, CA, USA. EDA Consortium.

Mura, M., Fabbri, F., and Sami, M. (2008). Modelling the power cost of security in wireless sensor networks: the case of 802.15.4. In *Proceedings of IEEE ICT08*, Saint Petersburg, Russia.

Mura, M., Paolieri, M., Negri, L., Fabri, F., and Sami, M. (2007). Power modeling and power analysis for ieee 802.15.4: a concurrent state machine approach. In *Proceedings of CCNC 2007*, Las Vegas, USA.

Mura, M. and Sami, M. (2008). Code generation from statecharts: Simulation of wireless sensor networks. In *Proceedings of DSD08*, Parma, Italy.

Ravi, S., Raghunathan, A., Kocher, P., and Hattangady, S. (2004). Security in embedded systems: Design challenges. *Trans. on Embedded Computing Sys.*, 3(3):461–491.

Roundy, S., Steingart, D., Frechette, L., Wright, P., and Rabaey, J. (2004). Power sources for wireless sensor networks. In *Wireless Sensor Networks*, volume 2920 of *Lecture Notes in Computer Science*, pages 1–17. Springer Berlin / Heidelberg.

Sastry, N. and Wagner, D. (2004). Security considerations for ieee 802.15.4 networks. In *WiSe '04: Proceedings of the 3rd ACM workshop on Wireless security*, pages 32–42, New York, NY, USA. ACM.

Sean Convery (2004). *Internetworking Technologies Handbook*, chapter 49, pages 49–1 – 49–32. Cisco Press.

Wang, L., Yang, Y., Noh, D. K., Le, H., Abdelzaher, T., and Ward, M. (2009). Adaptsens: An adaptive data collection and storage service for solar-powered sensor networks. In *Proceedings of RTSS 2009*.