IoT Nodes: System-Level View

2

Pascal Urard and Mališa Vučinić

In this chapter, we detail the key elements of the wireless sensor network nodes architectures. We also review the most important tradeoffs to make in order to maximize the system energy efficiency, keeping the cost of the solution under control. We also review the pairing and registration operations and detail the security requirements as well as the impact of security on the energy efficiency and the cost of the solution.

2.1 Architecture of IoT Nodes

There are multiple possible architectures of IoT nodes. Depending on the mission profile, the use-case conditions, the pairing conditions to setup the network, topology of the network, and for sure the application use-cases, then the architecture shall be optimized in a direction or another. However, some elements are common to the possible architectures:

- One or several processors, typically MCUs (microcontroller units) like STM32, based on ARM CortexM solutions
- a communication unit (i.e.,: radio for wireless sensor networks),
- one or several sensors or actuators
- a battery: many possibilities, from alkaline to long-life Lithium based

The current market trend is to enlarge battery life by increasing the global solution energy efficiency. One of the research directions is to get rid of the disposable batteries and create an energy-autonomous solution at a reasonable price thanks to small energy harvesters. In the case of such an autonomous node the system would also embed:

- An energy harvester enabling energy harvesting from the environment: photovoltaic cell or vibration harvester, or even a Seebeck effect harvester enabling to create some energy from a delta of temperature between two materials, or between a hot surface and a radiator.
- an energy storage, usually a rechargeable battery or a supercap
- a power management unit, typically taking care of adapting harvester voltage value to the battery, and managing battery charge/discharge, plus eventually energy distribution to the system (Fig. 2.1).

In the following, we take as an example a wireless sensors network end-device node. This node is collecting data from the environment (sensor) or act on the environment (actuator). The so-called ‘environment’ is a general term that can represent multiple domains. The sensing

P. Urard (✉) • M. Vučinić
STMicroelectronics, Crolles, France
e-mail: urard.pascal@gmail.com

M. Alioto (ed.), *Enabling the Internet of Things*, DOI 10.1007/978-3-319-51482-6_2

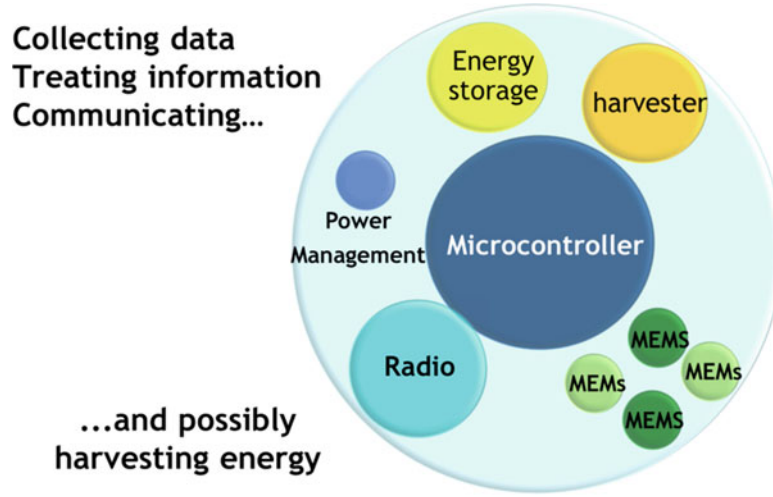

Fig. 2.1 Basic blocks and functions

function is for sure adapted to each environment. In the case of:

- a motor: sensing T° or vibration, . . .
- a room: sensing T°, pressure, humidity, light, gas (CO or CO_2 or any chemicals), noise, presence of roommate, . . .
- a forest: sensing humidity, T°, fire, light, gases, chemicals, noise, and even the presence of bugs,. . .

The first task is to precisely define the use cases and the targeted cost of the solution. This enables to select what kind of sensors have to be used, as well as the amount of local computing in the node (by opposition to raw-data sending through the network) and the kind of MCU needed. Then the global architecture often respects the following tradeoffs.

- In order to keep a low-cost solution, the application processor can manage directly sensors and actuators:
- Sensor power supply: in the vast majority of the low-power sensors, sensor power supply can be directly connected to one of the MCU digital General Purpose Input/Output (GPIO) pins.
- Actuator power supply: in some other case, when the load need in terms of current is higher than the MCU capability (tens of mA), the actuator can be connected to the power source through a switch or a relay, managed by a GPIO. This is typically the case of an actuator like a motor, or in the case of some high-current consumption sensors.
- The digital sensors interfaces can be directly connected to the I2C or SPI buses of the application processor
- The analog sensors can be connected directly to analog-to-digital IOs. Some MCUs like STM32 can share this ADC between several analog inputs, enabling to reuse this important component to manage several analog sensors.
- In the case a sensor is used as a wake-up function, then the usually available "wake-up" output of the sensor should be connected to the processor interrupt controller to turn on the MCU
- In the case a sensor needs some specific analog voltage values, MCUs also embed a digital-to-analog converter (DAC).

Concerning MCU choice, modern low-power MCU solutions like STM32L family embed a built-in DCDC converter enabling to drastically reduce the power consumption of the processor itself. In the case such a processor is used, the sensors and eventually the radios powered by this processor will also benefit of this power reduction. Let's take the example of a 3 V battery-operated sensing system embedding an MCU and a low-power sensor. In the case the MCU has an

eDCDC (i.e.,: embedded DCDC, which is the case of the STM32L151), then a 1.2 V sensor that consumes ~5 mA under 1.2 V will only draw 2 mA from the 3 V battery if powered through the MCU.

In many cases, the radio or the radio subsystem has its own DCDC or can also take advantage of the eDCDC of the main processor.

Radio subsystem can rely on a module or on separated components. Module solution is easier to integrate but also more expensive. In recent generations of MCUs (e.g., NXP/Freescale), the radio subsystem covering Bluetooth-low-energy and 802.15.4 is integrated in the main processor.

One of the first choices to be done is the system partitioning at processor level. Several possibilities exist at SoC level as shown in Fig. 2.2.

Depending on the use case, the chosen architecture of the MCU should be more or less powerful. One of the key requirements of a low-power solution is to not wake-up the processor(s) for nothing. Some system analysis with the different tradeoffs shall be done, taking into account the different wake-up of the MCU(s), the power-consumption during these active periods, and also the cost of waking-up the processors. Choosing a powerful MCU to manage at the same time application and communication may be lower cost than two processors, but also require to turn-on regularly a large power-consuming processor. On the other hand, keeping an application processor for system and application management and an additional network co-processor for the radio and protocol management enables to have a quite robust solution where both processors can be turned on or off independently. It also enables to set the optimization of each subsystem independently from the other (i.e.,: choose for each function the most adequate solution in terms of computing capacity, without affecting the other functions).

Having only one processor is more complex because depending on radio protocol, radio actions may be higher priority than application, interrupting an on-going sensing in some cases (so we have to define priorities and possibly require software-level concurrency contol).

The battery voltage choice is also a key parameter of the system line-up, as the battery voltage is heavily affecting the global architecture of the node. If the battery voltage is higher than the energy harvester output voltage, then a DCDC boost is needed to charge de battery. This component will enable to target high charging voltage batteries like LiPO. Another important parameter in the choice of the battery is the internal resistance value. Minimizing this resistance enables to increase system power efficiency. A high internal resistance (10s or 100s of ohms) will lead to an important voltage drop when the system is running, and may require raising the input voltage value when the harvester is charging the battery. Some batteries

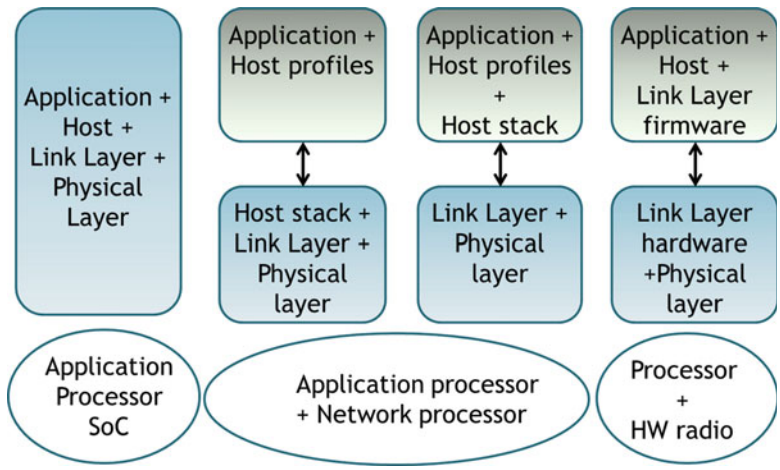

Fig. 2.2 System level radio partitioning

must be charged at a fixed given voltage. In this case, the battery current charge decreases when the battery charge reaches its maximum capacity. In other case, the battery must be charged in current, always keeping the current bellow a maximum given in order to avoid any damage to the battery. In this case, the PMU shall adapt the impedance of the harvester to the battery, and battery voltage (V_{batt}) increases with the battery charge, up to a certain point where the charge has to be stopped to preserve the battery. Continuing the charge may damage it.

There are some key advices that could save time to design a performant solution:

- The battery voltage range should be chosen taking into account the energy harvester output voltage. Minimizing the number of voltage conversions enables better system efficiency.
- The harvester has to be chosen to meet the requirements of your use cases: typical/minimal light and output charging voltage to minimize the need for complex PMU. A PV cell can easily and at low cost be designed on purpose in many places worldwide.
- In the case of unknown harvesting conditions or risk of non-respected use-cases, an MPPT (maximum power point tracking) or pseudo-MPPT power management unit can be chosen to adapt to any situation. This option has a higher cost.
- A quick estimation of the battery efficiency is given by the following ratio: V_{batt} when using the system/V_{batt} when charging. The higher the internal resistance of the battery, the lower is this ratio.

2.2 Requirements for IoT Nodes

Depending on the usage and the targeted market one radio protocol will have to be chosen to enable interoperability. For industrial market purpose, it is admitted that sub-gigahertz radios are the ones to target. For home-automation and building automation, it can be both. The market seems to push 2.4 GHz at home level: NEST-labs, acquired by Google and promoting Thread, seem to rely primarily on Bluetooth low-energy and 802.15.4 @ 2.4GHz (web:), but Zwave and at a lower scale the energy-harvested solution Enocean still offer sub-gigahertz solutions. At local or regional area, we see upcoming solutions like Lora or Sigfox enabling long (resp. very long) distance of transmission at the cost of a low or super-low data rate. There are plenty of applications that could take advantage of such solutions in the coming years.

We can divide these solutions in two topologies (see Fig. 2.3): the solutions enabling star network topology such as Wifi, Lora, Sigfox, or even GSM or LTE, and the ones enabling

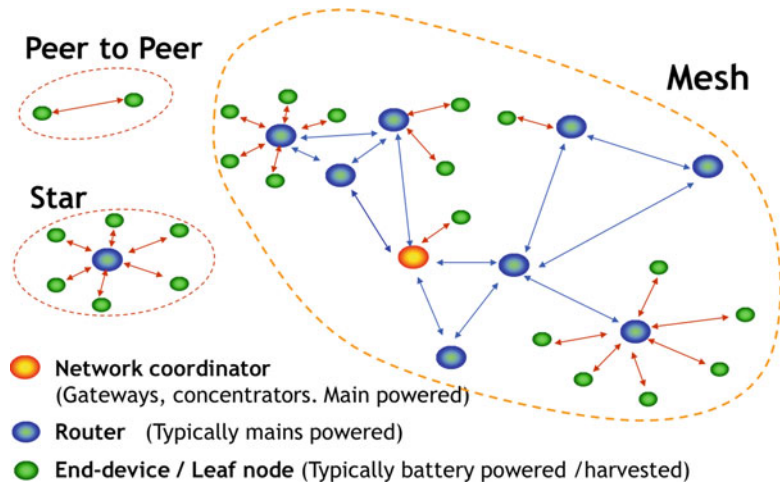

Fig. 2.3 Network topologies and definitions

Mesh networks such as 802.15.4x including Zigbee & Thread, Zwave, Enocean,. . .

We discuss in the next section the impact of topology on the power consumption.

2.2.1 Power

In order to target long battery life duration or energy harvesting compatibility, there should be a ratio of at least x500 between the system sleeping time and its functional time.

- Processor in terms of current is in the range of mA (e.g., 5–10 mA for CortexM3 running at 12 MHz under a 3 V supply, which results in 15 to 30 mW power) when in activity, but only around 1uA in deep sleep mode (3 μW). It should logically evolve in the coming years, thanks to Moore's law and additional low power techniques to achieve a division by 3 of this power consumption, reaching 5 to 10 mW for the same function.
- Energy cost of wake-up and go-to-sleep has to be taken into account in the tradeoff, as it is not negligible. Processors usually offer several kinds of sleep modes (light, medium, deep). Each of them can have some interest, depending on your application and context. Consider that you have to empty a large on-board capacitance to make the system sleep; the price-of-wake-up may be higher than putting the system in an intermediate sleep mode where this capacitance remains charged. So it is important to take this parameter into account, and not put the system in some deeper sleep mode when the power budget is finally larger than staying in some lighter sleep mode for a given time.
- Sensors have to be turned off as much as possible (i.e.,: between two sensing phases), or at least forced to low-power mode if they have some, especially when they have the mission to wake-up the system.
- Radio has to be mainly off.

We can consider two families of systems:

- System with regular wake-up
- System with wake-up on purpose (e.g., alarm).

In the vast majority of modern sensors, this function is natively embedded in the sensors, providing a mode with activity watch. So the design can directly use this feature, usually called *low-power mode*.

In order to successfully design an ultra-low-power solution, one should think energy harvesting and duty-cycling. See Fig. 2.4.

For sure, every component of the node has to be carefully chosen to minimize power. However, in the first place, the radio choice and radio protocol are one of the major contributors in power consumption.

Minimizing power by minimizing protocol overhead is an excellent starting point. However as we will see in the next sections this may go against interoperability. The first adopted systems in the past used to propose proprietary optimized solutions for radios and non-IP protocols. However, the future seems to be standards-based and IP-based. The Moore's law helps to gain in power efficiency: the former generations of radios used to consume 20 mA @ 3 V to send a frame at 0dBm (1 mW in the air). In 2015, the best solutions on the market consume bellow 6 mA for the same service, and the trend is still to reduce further, towards 3 mA in the 2 to 4 coming years.

Once the radio power is under control, the leakage has to be addressed. Let's imagine a node working during 20 ms every 2 min (mean schedule in real operating mode) with a mean power of 30 mW (10 mA under 3 V). It would consume around 600 μJ per active cycle. If during the inactive period, the total current (board level leakage + low-power-watch-dogs + counters) is around 2 μA (6 μW), then the total leakage would be 720 μJ per inactive cycle. The Pareto of the power consumption would start by the leakage! So the leakage can really become your number one issue to meet your total power budget. In the case your objective is to build an energy-harvested platform, the global requirements should lead you to consider every leakage above 30 nA in stop mode, and any way to save 5 μJ or more in active mode.

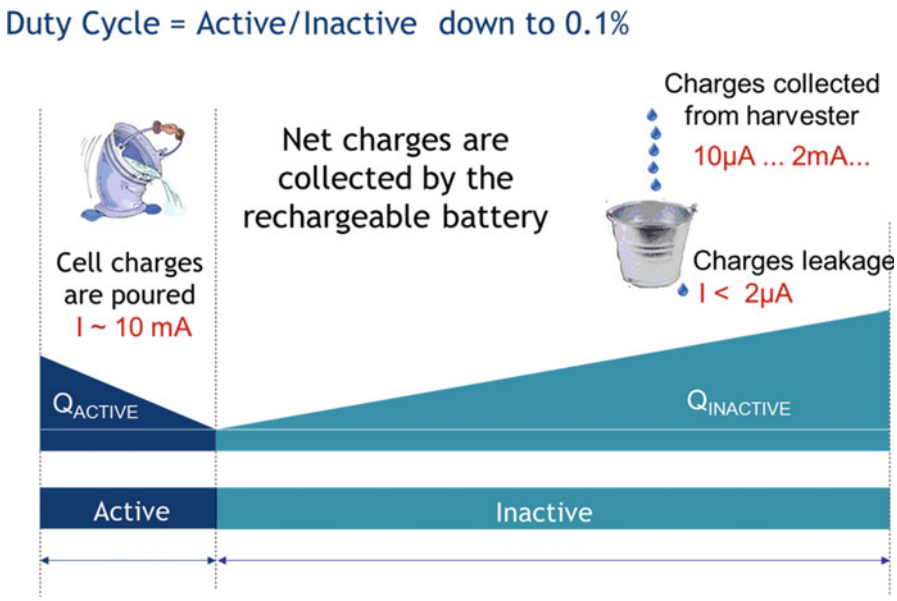

Fig. 2.4 Heavy duty cycling enables Ultra-Low-Power solutions

2.2.2 Cost

Radio protocol has a direct impact on the global cost, because of its huge impact on the power efficiency. Also, factors like whether the upper protocol is IPv6 or not-IP, if security is enabled or not will directly impact useful data rate, and so the global energy per useful bit transmitted, leading to expend the battery size or reduce the lifetime of the node operating in harsh conditions where the eventual energy harvester would not be able to harvest anything (i.e.,: dark for a PV cell).

The battery choice, as seen in the previous section will heavily impact the cost of the node.

Choosing components with high power consumption would lead to choose a large battery, like the ones of the cellphones. Even if the price in volume may seem reasonable, it shall impose to add a specific power-management unit, enabling to charge this battery up to 4.2 V. Such a high voltage will prevent to connect directly the MCU to the battery: MCU maximum operating voltage is usually around 2.6 V.

The cost of not-optimizing the global power consumption would lead to a poor battery life for some or even all the nodes of the network. This issue was not a major problem few years ago, but it is more and more reported by final customers that the battery budget per year can be a show-stopper after a first trial. Imagine you have to change every year heavy-duty long-duration lithium-AA batteries in 10 devices. The cost after 5y shall be higher than the system itself. The pain to change the batteries every year in a more-than-20 devices network very quickly becomes upsetting or too expensive if a specialist has to do it. We can easily understand why it is requested to optimize the solutions, still maintaining the cost low:

- by choosing an optimal system partitioning
- by optimizing global power efficiency
- by minimizing additional components and voltage conversions.

Let's take as an example the final cost of an energy-harvested node in volume, the GreenNet node V2.1 as shown in Fig. 2.5 (Urard et al. 2015). It was targeting a Bill-Of-Material (BOM) in high volume below 12 USD. This R&D project has been an ST-internal demonstrator to understand the WSN challenges and

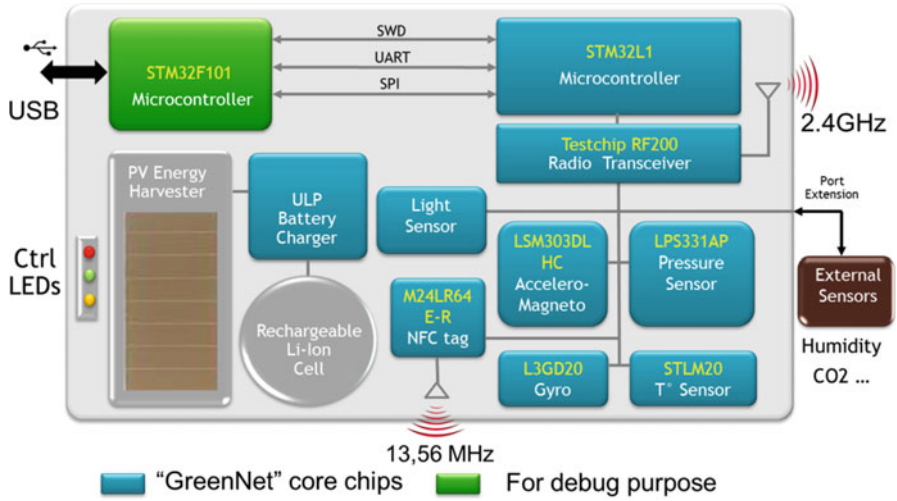

Fig. 2.5 GreenNet V2.1: Energy Harvester secured IPv6 Node content

requirements, regarding system energy efficiency on the hardware side, and secured-IPv6 power-optimized solution compatible with energy harvesting on the protocol side, in cooperation with the Laboratoire d'Informatique de Grenoble (LIG). GreenNet is a successful demonstrator but has not been commercialized. Some of the key elements are detailed in Chap. 17.

2.2.3 Interoperability

Wireless Sensor Networks have suffered for years from the fragmentation of the radio offer: too many different standards, not enough interoperability. Zigbee is the typical example where 4 main "application profiles" were offered as defensive layers, one for each of the markets: Smart Energy, Home Automation, Building Automation, and Lighting. Thread, pushed by Google through Nest-labs, has pushed further by proposing a single profile for all the use-cases. IPv6 is a definitely a trend: ensuring interoperability by offering an IPv6 solution. Nowadays, it seems also as one of the preferred ways to adopt a secured solution. IPv6 security is constantly evolving, so progressing at low cost can only come from solutions that can be shared among the standards.

Figure 2.6. shows the key elements of various 802.15.4 solutions and highlights in red some of the most interesting added values to increase the quality of service or the energy efficiency of the network. Please note the similarity between those solutions. Most of them are using 6LoWPAN for IPv6 interoperability. The two on the right (circled in red) are the latest ones, with a larger adoption of 802.15.4e, now part of 802.15.4-2015 specification. However, in some particular cases detailed in the coming chapters, there is a room for improvement and some part of the 802.15.4-beaconed may in the future be reused to further improve the energy efficiency of 802.15.4e standard.

802.15.4x solutions aren't the only WSN protocols to adopt IPv6: as an example, 802.15.1 (Bluetooth) compliancy to IPv6 has been developed in 2015 and should be available during 2016.

2.2.4 Security

Few years ago (2013), while the author was demonstrating GreenNet in Paris, audience was not convinced, at the time, about the need of secured connections for WSN. In only 2 years, the number of hacks worldwide, the need for more secured solutions at all levels, especially

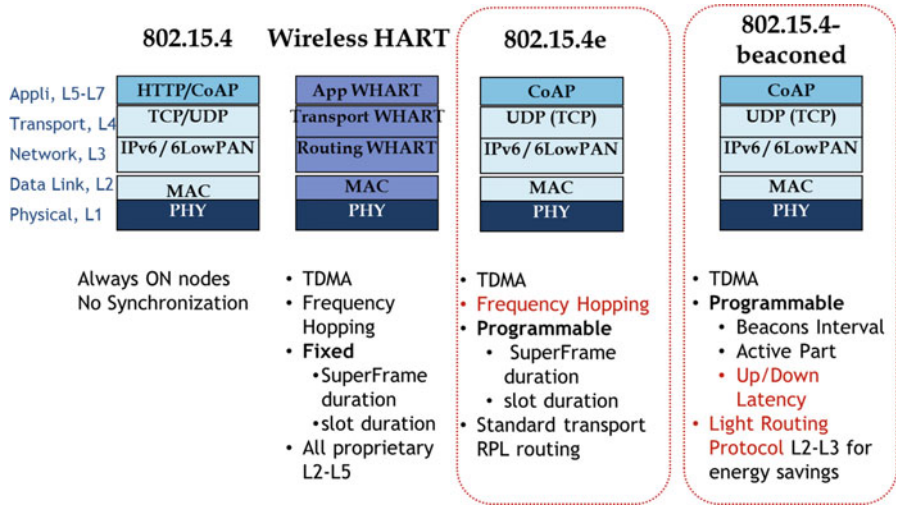

Fig. 2.6 2.4 GHz 802.15.4 enlarged protocols family

private data, and the fact that the interesting information for hackers is maybe not the one you think about, made people change their mind. Nowadays, no question any more: security is a must.

We present these hereafter an overview of typical threats and attacks in WSNs. Some more details can be found in (Vučinić et al. 2015). Security of a system can be studied within a given model. Internet protocols typically consider the traditional Dolev-Yao model (Dolev and Yao 1982) where the attacker has full control over the network.

More precisely, the attacker can:

- Intercept messages,
- Modify messages,
- Block messages,
- Generate and insert new messages

It is important to understand that cryptographic algorithms are considered "perfect" and the attacker can decrypt/forge a message only if he possesses the corresponding key. In the networking context, "message" corresponds to a Protocol Data Unit (PDU) of an abstraction layer under study. For instance, if we consider security solution at the link layer (radio protocol), message corresponds to a radio frame.

Traditionally, there are two typical classes of attacks:

- Passive attacks: Such as eavesdropping and traffic analysis, where the attacker gains knowledge on ongoing communication by passive means. For instance, if messages are sent in clear, attacker is able to read full message content. If network messages are encrypted, attacker may still be able to infer some information by studying communication patterns, timing, or message length.
- Active attacks: Attacker actively participates in the communication by re-playing old messages (replay attack), modifying messages and playing Man in the Middle (MITM), pretending to be another entity in order to gain unauthorized access to a resource and similar. A particular class of active attacks are Denial of Service (DoS) attacks, where the attacker's ultimate goal is to disrupt the availability of a network service, such as the alarm notification, typically by exhausting physical resources (memory, energy, bandwidth) on the target node.

An important point to note is that the Dolev-Yao model typically considered in protocol design is a formal model that does not take into account physical compromise of a node. Therefore, research around WSNs (Atakli et al. 2008; Chan et al. 2003; Karlof and Wagner 2002;

Rezvani et al. 2015; Vempaty et al. 2012) has often taken into account a more powerful, Byzantine attacker (Awerbuch et al. 2004). In such scenarios, attacker has access to local cryptographic material on the node and we cannot rely on cryptographic techniques to prevent attacks (Rezvani et al. 2015). Indeed, Byzantine attacker can compromise a set of nodes and through them inject false data that passes all cryptographic checks.

Becher et al. (2006) conclude that physical compromise of a device in order to extract keying material and obtain full control over it, as assumed by the Byzantine model, is not as easy as often perceived in WSN literature. It requires costly equipment, expert knowledge on hardware and hard determination of the attacker. An interesting observation of this study is that such attacks often require that a node be removed from the network for a non-trivial amount of time making detection of unusual activity via neighbor discovery protocols a simpler approach than specialized Byzantine-tolerant schemes. Common sense practices, such as disabling JTAG port or Bootstrap Loader (BSL) once the product is deployed, go a long way towards making attacks in the field more difficult.

We do recognize that in many IoT deployments, devices will be physically available to the general public and as such, system designers should take into account the threat of a physical compromise and extraction of the keying material. We emphasize that final IoT products should either have hardware- level or software-level protection against physical tampering, i.e., tamper-resistant packages or schemes to detect unauthorized access to the hardware (Becher et al. 2006; Liu et al. 2007).

2.2.4.1 Wireless Network Threats

- Physical Jamming. The most basic attempt to disrupt the network service is the attack on physical resources—the radio channel. Attacker can generate high-power signal that will interfere at different receivers in the network and increase the error rate, possibly completely disrupting wireless operation (Law et al. 2005; Li et al. 2007; Raymond and Midkiff 2008). This DoS attack is often called jamming and is mostly a concern in military scenarios. Common defense is channel hopping that increases the bandwidth attacker has to jam, which can require a substantial power supply and thus make the attack less practical. Also, network-level redundancy can help in order to route around the jammed area.
- Traffic Injection. Injecting false traffic in the network can have multiple consequences. Firstly, it is possible to affect network applications, e.g., by introducing a bogus temperature reading to trigger the Heating, Ventilation, and Air Conditioning (HVAC) system, or even to directly control an actuator, such as the pressure regulating valve in the industrial automation system. Similarly, one can obtain full control over the network by forging network maintenance packets (Karlof and Wagner 2002), e.g., beacons, and corrupting neighbor tables of the nodes. Secondly, attacker can launch DoS attacks by generating significant traffic loads that can cause network collapse in terms of depleted energy due to, e.g., multihop forwarding. First-level protection against such attacks is link-layer security—network nodes should not accept any radio frame other than those secured with link-level keys they possess locally. At the application level, access rights should be properly configured in order to limit the damage if one of the nodes in the network is compromised. For instance, node measuring the temperature should not be allowed to issue pressure valve regulating commands. Second-level defense is common sense programming—if some of the network nodes gets compromised and starts injecting cryptographically-valid traffic, one should locally check the rate at which it is forwarding packets or performing local operations instead of blindly following the protocol.
- Attacks on Join Protocol. Link-layer security protects the wireless network in "steady" state, when all the nodes have joined and have been provisioned with necessary keying material. Before we admit a new node in the

network, it is necessary to perform some checks. Join protocols are technology-specific but some common points exist:
  - The joining node may initiate the join protocol multiple hops away from the gateway
  - Several messages may need to be exchanged between the joining node and the gateway before the "admittance" decision can be made. This necessitates that intermediate nodes in the mesh forward the messages that may come from a rogue joining node (attacker), which opens up the possibility of DoS attacks. Although this threat can never be fully neutralized, a common strategy is to minimize the damage a potential attacker can do. As such, one may ensure that joining messages do not instill state information in the network and can control the rate at which intermediate nodes forward join protocol messages.
- Attacks on Routing Protocol. Due to their distributed nature, WSNs are prone to attacks that involve an attacker that can for example:
  - Selectively forward messages if it is within the network, or jam radio transmissions and cause collisions from the outside.
  - Advertise false routes in order to attract the surrounding traffic and create a sinkhole.
  - Present multiple false identities to other nodes in order to reduce effectiveness of fault-tolerant schemes.
  - Create radio "tunnels", so called "wormholes", between two distant parts of the network in order to appear closer to the gateway and create a sinkhole at the other end of the tunnel. Such attacks can only partly be neutralized by using link-layer security in order to reject radio frames coming from the outside. When an attacker is inside the network, i.e., a compromised node, defense requires careful design of the routing protocol that takes security into account from the beginning (Karlof and Wagner 2002).
- Privacy issues. Sensor and actuator networks that make part of our daily life bring along various privacy issues. While management of data collected by these networks in itself represents a privacy concern, we focus on information that may leak to an outsider. Obviously, data confidentiality at the link layer (protected radio frames) is the first step to improve user's privacy. In many IoT scenarios, however, radio communication alone suffices to reveal some information about the user. For example, a presence sensor may initiate radio communication when a person enters a building (Tschofenig et al. 2015) or a light switch may indicate that the state has been toggled by emitting a radio frame. Typical defense would involve injecting dummy traffic in the network but that may not always be feasible due to the local energy constraints.

2.2.4.2 Countermeasures

Threats described in above section are typically fought using security mechanisms at 2 levels: Level 2 (L2, security between radio neighbors) for hop-to-hop security or link-layer security and Level 5 for end-to-end security (L5, security between an IoT node and e.g., smartphone). Each level of protection needs to meet 4 fundamental security goals (Fig. 2.7):

1. Confidentiality: ensured by using encryption.
2. Integrity: we want the message to arrive safely, and not loose parts of it. Ensured by appending a checksum at the end of a message.
3. Authentication: Am I talking to the right node and is the message I received coming from the right node? Ensured by an authentication protocol and by using secure checksums, computed using a secret key.
4. Availability: Achieving the guaranteed level of operational performance even in presence of Denial of Service attacks (DoS) is a non-trivial task for a system designer. In terms of link-layer security, message filters and access control lists implemented in hardware allow certain degree of confidence but are alone not enough due to possible jamming attacks. Radio technologies that use frequency hopping (802.15.4e and BLE) help in preventing networks to be stuck on a single

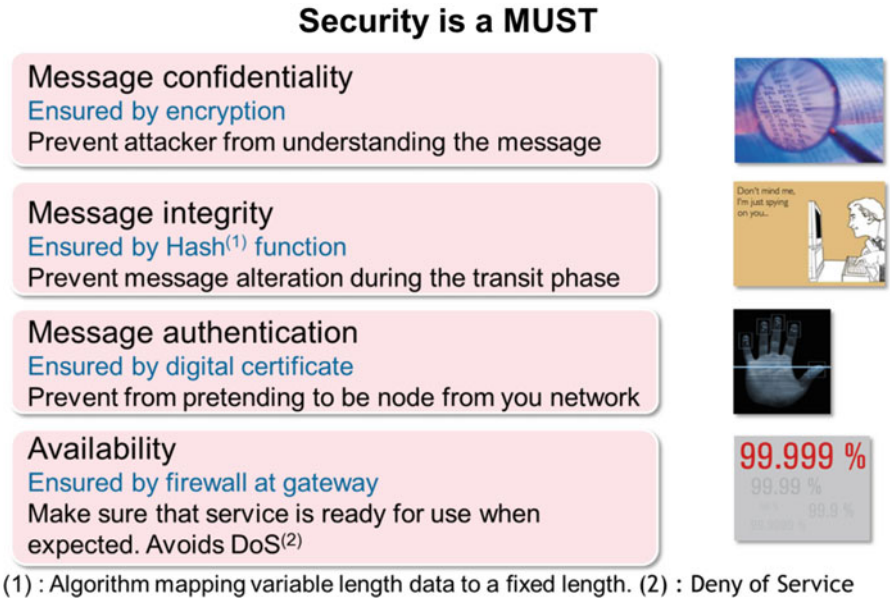

Fig. 2.7 The four fundamental security goals of a secured solution

radio channel and make it more difficult to the attacker to disrupt the service. However, availability is a system-level goal and thus must not be treated only at the node level, but also at the gateway level thanks to:

(a) The gateway firewall
(b) The decoupling of fast Internet world (HTTPS) and the slow one (CoAP over DTLS).

Attacks on the routing protocol are, from the point of view of confidentiality and authentication, defended using end-to-end security mechanisms, where even the on-path attacker is not able to modify or read the data, as it is not in possession of the end-to-end keys.

2.3 Power-Related Challenges and Design Tradeoffs

Chapter 2 of Varga's PhD thesis (Varga 2015) presents an overview of the latest technologies used in IoT as well a presentation of the synchronized and unsynchronized operation mode. These are presented hereafter.

2.3.1 Node Availability and Duty-Cycled Operation

As for the system, the radio shall sleep as long as it makes sense. This is different than "as long as possible" for several reasons. First reason: the need for synchronization in radio protocols. In this trade-off, the topology plays a major role.

- Case of star network topologies: we can consider the central node (concentrator) as always-on like on Wi-Fi or Sigfox or Lora.
- Case of Mesh or extended Star topology (extended star enables each node to have one additional peer attached to it). In this case there are multiple solutions:
  - Either the routers are always or mainly listening (e.g., Enocean, Zwave, Zigbee, 802.15.4-by-default) in order to receive information from the other nodes of the mesh and transmit them to the next node
  - Either both emitter and receiver are mainly off. This is the trend of the new radio standards. Bluetooth-low-Energy (BLE), 802.15.4e, 802.15.4-beaconned-option standards are proposing mainly off

solutions with synchronized wake-ups. This enables better energy efficiency versus older standards. We can say the system has slots: an active period where the radio transmits or listens to the RF activity (in case of message for the node), and an inactive period where the radio is turned off. By extension we say the system is slotted.

Slotted systems have however a constraint to solve: the need for synchronization between emitters and receivers. As there is no always-listening node over the air, a de-synchronized network would have a very poor quality of service, unless the active slots represent a very high percentage of the time, reducing by this way the efficiency and the interest of such solutions regarding previously existing ones.

As a conclusion: in order to maximize energy efficiency, new systems have the requirement to be mainly-off so they are synchronized in order to transmit the data in the shortest time for both nodes, produce the relevant acknowledge (emitter get the feedback that the data has been received) and go back to sleep mode as soon as possible.

There are two ways to maintain the synchronization of the network.

- Either there is enough and regular communications: the transmitted data or the first-level acknowledge can carry the synchronization information: this is the case of the 802.15.4e: when two nodes communicate together, the receiver sends to the emitter a low-level acknowledge embedding the synchronization data (e.g., timing corrections for the next wake-up). This solution requires a one-to-one communication: a specific rendezvous between the two nodes. This is possible in the 802.15.4e standard in which a specific time slot and frequency channel is attributed by the router (or the master node) to each of the nodes that needs to communicate with him.
- Either the network synchronization is maintained globally thanks to the usage of beacons: the master node sends a beacon in broadcast, all the surrounding nodes needing to communicate with him can get by decoding this beacon, the time of the next available beacon, the time slot when to emit if they need and even the fact that they have some information to request to the master node in the case they have: Bluetooth low-energy, 802.15.4 beaconed-option are working this way.

All these possibilities enable to have more or less energy efficient protocols see for instance Romaniello (2015). In Fig. 2.8, we can consider that green bubble could enable energy-harvested nodes in some specific conditions.

- In the first category, we can fit mains-powered devices. We can also consider part of this category the networks where all the devices have been declared as routers. This is typically the case when all regular devices offer routing capability to enable mesh network: Zwave & Zigbee.
- ZigbeePro-GreenPower, Enocean, would fit in the second category, enabling end-device to be super-low-power and energy-harvested at the cost of a mains-powered, always-listening router device. However, bi-directionality is not granted, preventing low-cost battery-operated actuators or IPv6 secured link.
- BLE and 802.15.4e and 802.15.4-beaconned would be categorized on the third category (row), enabling bidirectional IPv6 and secured networks, at the cost of a need for synchronization through beacons or regular data exchanges.
- 802.15.4-beaconned could also be categorized in the fourth category in some specific conditions (low-latency network), enabling energy-harvested routers, as demonstrated in the GreenNet project.

2.3.2 Activity Profile and Power Modes

One of the goals of an embedded code software designer in charge of providing a solution for an autonomous wireless sensor network is to minimize the amount of energy spent in the nodes.

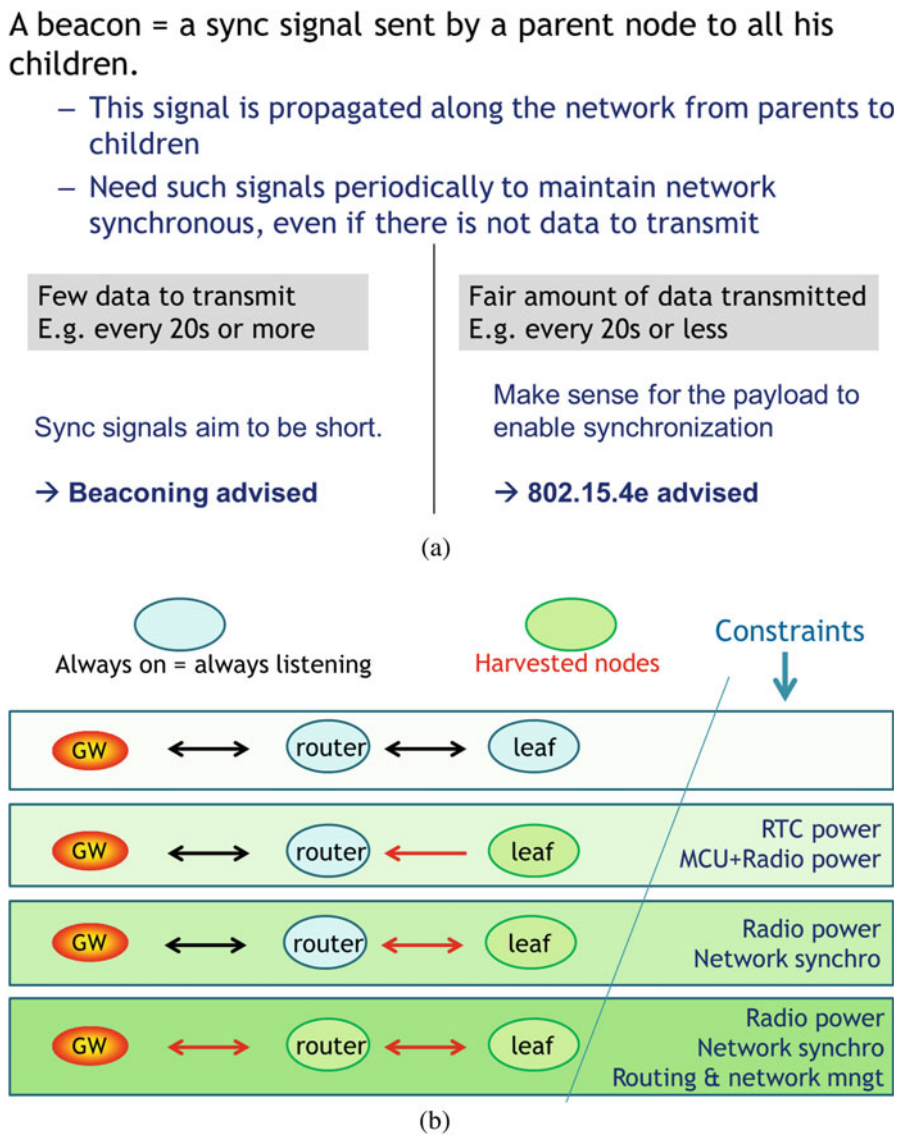

Fig. 2.8 (**a**) 802.15.4-beaconned option versus 802.15.4e; (**b**) network topology impacting the power (*red arrows* mean ultra-low-power protocols, *bidirectional arrows* mean bidirectional communication)

In the case of a sensor node operating regularly, there are few main ways to achieve this goal:

- Maximizing the deep-sleep periods of time. In deep-sleep mode, the application processor is programmed at the lowest level of power, still maintaining its stack in retention mode (or equivalent) in the System RAM and the logic glue, and turning off the NVM. In some case having the local low-speed oscillator running for the next RF rendezvous, or having some sensors in low-power mode, able to wake up the node in the case the environment parameter (e.g., temperature, vibration) goes beyond a programmed limit. In this case, the node would register this kind of alarm and send it to the gateway as soon as the next communication slot is available. So on one hand, the amount of time spent in deep-sleep

mode shall be as long as possible. However, waking-up from deep-sleep requires some energy, so on the other hand, the number of time we put the system in deep-sleep has to be minimized. One unique long period between each sensor measurement is an ideal case and answers to both constraints.

- When the duration of the sleeping period is not sufficient (e.g., less than 2 ms in the case of the GreenNet node) some intermediate stop mode may be used: reducing the power by a fair amount, still consuming more than the deep-sleep, but at a lower penalty cost for wake-up.
- Turn the radio off as often and as soon as possible.
- The sensors have to be turned to their lowest power mode when in run mode and to be switched off asap.

In the case of an alarm node able to operate anytime, like a light-switch, the amount of energy required by the node is directly proportional to the latency you would accept to wait in order to operate the action. Same principle for an actuator: the amount of energy consumed to enable actuation control is directly linked to the actuator latency. E.g.: in the case of a light switch, we would consider only the amount of energy to be able to command the switch, not the energy consumed by the light. The total latency would then be: switch latency to send the command + network latency + actuator latency to receive and execute the order.

In the case the alarm node is operating without any battery but with a capacitor coupled to a pulsed-energy harvester generating 100 uJ or less, then the only way to operate is to have an always-on router, quite-always listening, that will immediately take the message and relay it inside the network.

2.4 Cost-Related Challenges and Design Tradeoffs

Is it better to design a single node with multiple programmable sensors you can activate over the air, or it is worse putting only one sensor per node? The answer has a direct impact on the cost of the solution. It depends on the cost of maintenance versus the cost of fabrication. Only a detailed cost study taking into account the forecasted volumes, price per node in each case, and the cost of SW maintenance of several applications versus 1 unique application can give you a precise answer.

2.4.1 Impact of Power on Cost of IoT Node and Concentrator

The most expensive part of a sensor node can easily become the battery. Having a low-cost coin battery is a plus as its price in volume is around 1USD (2032 LiMn). However, it is limited to 200 mAh (announced) and the usable part would only be around 40mAh (absolute limit) in the case you want to preserve the battery from short life duration.

What would be the maximum activity rate of a node using such a battery? We first consider it is acceptable to cycle by 0.5% per day on such a battery which means 1mAh (i.e., 2.6 Coulombs) per day. In other words, the mean current consumed by such a node would have to be lower than 41.6 μA if no energy is harvested during 24 h. Any energy harvested would enable more activity. For example a 6 h harvesting duration per day with a 20 cm^2 PV cell would enable the same system to have a mean current of 55 μA. This would correspond to an ultra-low-power system operating every 20–30 s, which is not so bad. If the components are consuming more, or if the communication needs to occur more often, then the battery may have to be upgraded to a larger capacity, from another type, more expensive, with sometimes more complex charging protocols.

Regarding indoor routers: in the general case one doesn't have any choice: routers currently must have a powerful battery or be mains-powered. In few particular use-cases however, it is possible to specify an energy-harvested node that handles the router function. However due to the poor harvesting capabilities, this kind of router would need to be placed very close to an energy source, like a window.

2.4.2 Impact of Protocol on Cost

We have seen in the previous chapters that a trend for interoperability is the use of IPv6 protocol. However, it defers from the usual "fast" internet world (TCP-IP based) in several points. Figure 2.9 shows the equivalence between the WSN "slow" internet world on one hand (low latencies in a WSN are hard to achieve, and are not granted in energy-constrained networks), and "fast" internet world (i.e.,: usual internet). The HTTP protocol is replaced by CoAP (Constrained Application Protocol). The main role of CoAP is to reduce the number and overhead of the exchanged packets. From end device node point of view, CoAP is also interfacing the two internet worlds, protecting the WSN from repeated high-speed requests.

In the IPv6 frame in Fig. 2.9, there is a fairly good reduction of the number of bytes exchanged in a CoAP-based case: from 681 to 111, but over a short IP packet as it is defined in 802.15.4 (127 bytes), the payload is not representing more than 20 to 27 bytes, depending on the options (long or short addresses, security level, . . .). This gives a useful bit efficiency of only 15–20%.

This efficiency would be better with long Internet packet (2047 bytes) like in 802.15.4 g, where these long packets are supported. The IPv6 overhead would remain the same in absolute value, but would be more acceptable with a much longer payload. So there is still room improvement regarding protocol overhead.

In summary, IPv6 heavily impacts the useful data rate. But in order to enable interoperability, as well as implement regularly latest security updates from IETF, it is worth implementing an IPv6 solution, compared to proprietary ones. In another chapter, we will show through the GreenNet demonstrator how efficient such a network can be.

2.5 Pairing and Security

2.5.1 Pairing, Registration, and Installation of IoT Nodes

Pairing is a mandatory phase of a Wireless Sensor Network. The area of pre-paired objects is quite over. Pairing by the final users enables to complete an existing network with new nodes, and customize each network with various set of nodes, even coming from multiple vendors. Technically speaking, pairing enables the node to enter into the network by providing the ID of the network and some network-type specific data like the discovery RF channel to be able to register at each level of the stack, plus some security parameters if the network is secured.

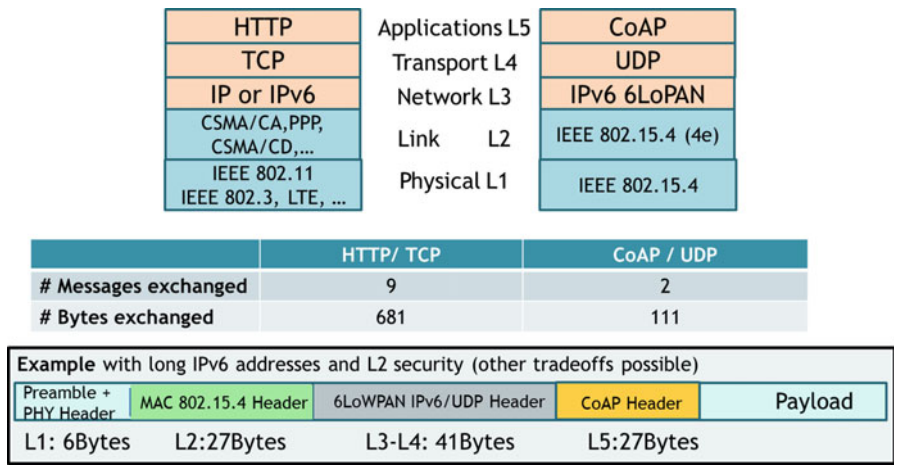

Fig. 2.9 Protocol suites typically used in traditional Internet, versus IP-based 802.15.4 Wireless Sensor Networks

A clear trend to ease secured pairing is to use NFC as an out-of-band (OOB) channel. NFC-forum (http://nfc-forum.org) specifies the way to enable secured pairing for BLE. The same principle could be used for 802.15.4 standard and its extensions. The principle is to exchange handover data using NFC. The negotiated handover protocol, introduced in January 2014 with the 1.3 release of the NFC-forum Connection Handover Specification [http://nfc-forum.org/our-work/specifications-and-application-documents/specifications/nfc-forum-technical-specifications/], enable to use a smartphone to pair securely 2 devices like an IoT node and a Concentrator or a Gateway.

The registration is a phase coming after the pairing, requiring a fair amount of exchanges between a node and the concentrator/gateway. Registration protocol is usually described in the application profile (e.g., Smart Energy Profile—SEP2.0 specified by Zigbee Alliance). An example of SEP2.0 registration for a Temperature sensor is given in Fig. 2.10.

We use the term installation for the physical installation of the node in its *final* working place. A node can always be moved, but regular moves may lead to frequently rebuild the routing tables, which can be energy hungry. This is why we consider ultra-low-power WSN as quasi-static networks.

2.5.2 Impact of Security on Power

Security of WSNs typically relies on Level 2 (MAC) and L5 (CoAP over DTLS) that both provide the four security goals, as seen in Sect. 2.2.4. The impact of L2-802.15.4 security on power is not as bad as often perceived, if we consider the nodes already paired. We measured the impact of link-layer security in terms of energy on the GreenNet demonstrator, using a fully autonomous scenario in harsh environment (e.g., sensor nodes communicating during 31 ms once every 4min20s). The overall cost of IEEE 802.15.4 security in our scenario ranges from 1.94 to 4.18%, depending on the security level. For energy-harvested platforms, such as GreenNet, this result directly corresponds to the requirement that 1.94–4.18% extra energy needs

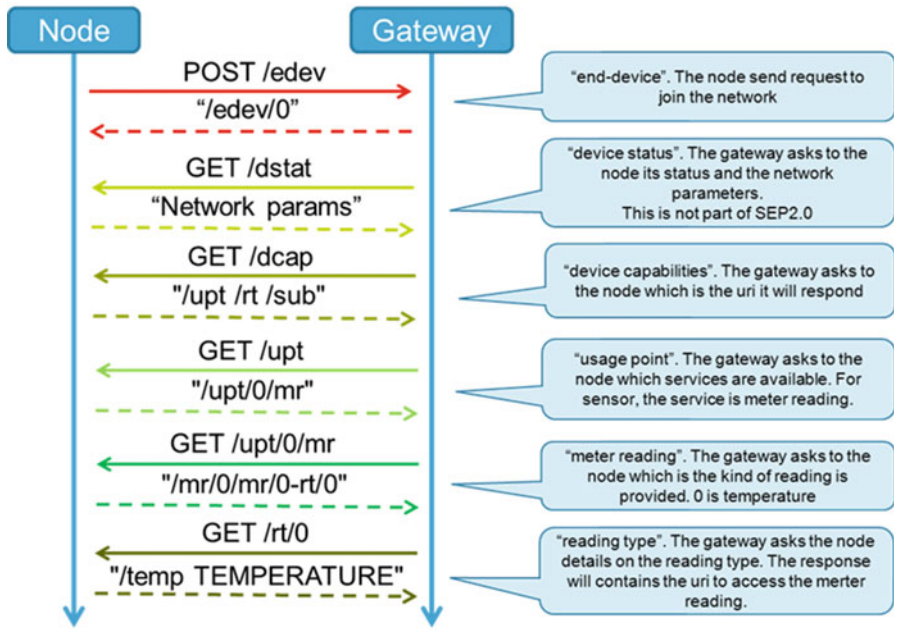

Fig. 2.10 SEP registration procedure example

to be harvested from the environment, in respect to the scenario without security (Vučinić et al. 2015).

L5 end-to-end security impact would be more important, but still smaller than the impact of using IPv6 transmissions. During pairing phase however, the Level 5 security ensured by DTLS implies to exchange a common key. Typically L5 messages shall be exchanged between the gateway and each node, penalizing the energy of the node, and the energy of all the router nodes on the way if the registration is done prior installation.

It is advised to define both pairing and registration protocols as well as installation use-cases early enough, to take their energy impact into account during the architectural definition phase of the nodes. In the case you intend to use energy-harvested or battery-operated routers, we would advise to perform pairing and registration phases close to the gateway, prior to physical installation, to avoid many messages exchanges through the full network.

This exchange of data can take a fair amount of time, up to several minutes in the case of very slow, ultra-low-power networks. This could indirectly raise some energy issues if the physical installation of a given node occurs before the registration is finalized. Effectively, if a node is paired in a place (e.g., nearby the gateway) and installed in another place far from the pairing place (i.e.,: lower in the network tree, at a 'distance' of several hops), the required exchanges to update the routing tables in order to follow a "moving node" inside the network would come on top of the registration exchanges. Some registration exchanges may be lost and would have to be sent several times, leading to burn a fair amount of the energy stored in the nodes.

In the case the network is very slow because very low-energy (e.g., one active mode per minute), it is possible to fix this issues by speeding-up the pairing and registration phases. Fast-exchanges can be used to quickly perform these phases in a few seconds. Then any move of the registered node to its final place inside the network becomes much more energy friendly. This technique has been developed for the GreenNet demonstrator. In this network pairing and registration are done at high rate: up to 32 frames per second if the routers enable it. Once the registration is performed, the node 'slows down' to its normal rate (e.g., 1 communication per minute or less).

2.5.3 Impact of Security, Pairing and Installation on Cost

The bill of material may be affected by the solutions chosen for Pairing and Installation protocols. As seen above, the need for security on top of IPv6 may induce to enlarge the battery capacity and to add some NFC specific chip. Some energy-independent low-cost solutions exist to perform NFC pairing (e.g., Dual-interface I2C-NFC EEPROM). An NFC-specific antenna must also be added on the IoT board. There are some electromagnetic rules to comply with, in order to avoid NFC (12.56 MHz) antenna to interact with the node radio antenna (2.4 GHz or Subgig). You can refer to (AN2866 Application Note) to design a 12.56 MHz customized tag antenna. In the case your IoT node is small enough and isn't compatible with printed NFC antenna size, some discrete coil antenna components are also available from distributors. In some case, the most expensive component may finally be the NFC antenna.

2.6 Battery Lifetime and Examples

Let define what we call battery lifetime: it is the time a node will operate in its normal/typical mode, without replacing the battery.

A first approximation of the battery lifetime (in seconds) can be done by dividing the battery capacity provided by the battery maker (in Coulombs), by the mean power consumption of the node (in Amperes). However battery capacity is usually provided in mAh. 1 mAh means 1 mA delivered during 1 h, which corresponds to 3.6 Coulombs.

Let take a 2000 mAh "AA" battery. It would provide 7200C. If the system consumes only

3.2 uA, then the theoretical battery lifetime would be 2.25E9 seconds which corresponds to about 71 years. This duration does not take into account battery aging or battery auto-discharge (that reaches up to 20% per year in the case of Alkaline batteries), as well as battery usage conditions (temperature, humidity, chemicals, non-continuous discharge, peak current). All these parameters affect the battery lifetime. Long-life non-rechargeable lithium batteries have a much better auto-discharge specification than alkaline ones. However, because of the other aging parameters, it is difficult to predict the exact battery duration. This is why consumer systems are claiming 10-year autonomy but very few makers (if any) really guarantee the battery lifetime.

In the case the node has to use an energy harvester, choosing the right battery capacity is not trivial. We describe hereafter a way to proceed.

The battery capacity provided by the battery maker is obtained when the battery is charged at its maximum capacity then totally discharged with a constant current value which is specified for each kind of battery. This current is usually much smaller than the peak current of a node, but higher than the mean current. The actual battery capacity of your system will depend on the way you use this battery:

- To preserve the battery capacity, it is advised to never charge the battery to 100% of its capacity, but keep a 5–10% margin.
- To preserve battery life, it is required to never discharge completely the battery. Battery cycling (deep discharge) usually reduces the battery capacity; however the effective impact depends on the battery model. Deep cycles (60–80% discharge) have heavy impact compared to light cycles (10% or less). As the harvester is refueling regularly, it is possible to calculate the daily cycle and choose a battery capacity to maintain in typical conditions this cycle around 0.5% of the ideal capacity. In this case the impact on aging can be considered as negligible.
- When operating in a real environment, every day is different. For example during the weekend, there may be less light in the offices and the harvester may not be able to recharge the battery, leading to a weekly cycle, deeper than the daily one see (see Fig. 2.11). You should estimate this weekly cycle and make sure it

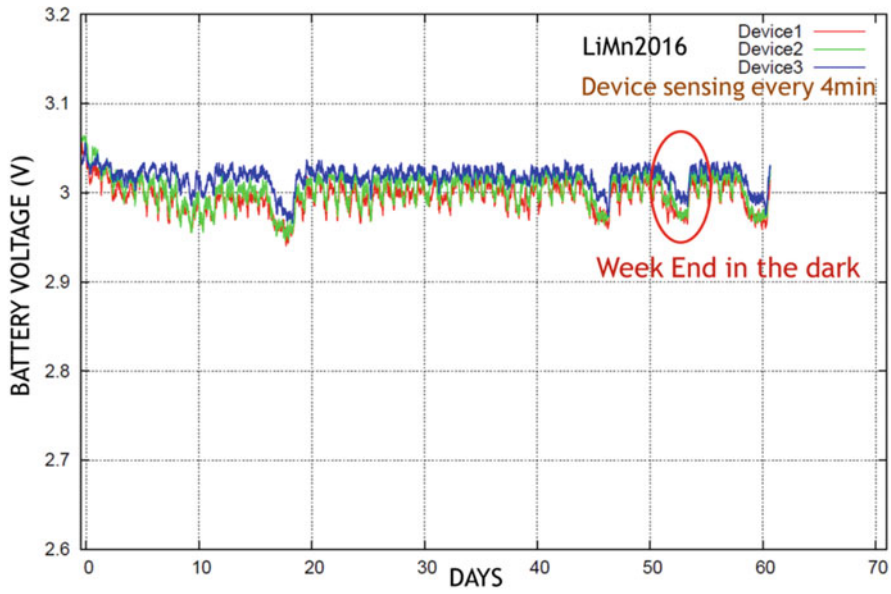

Fig. 2.11 Energy-Harvested GreenNet node rechargeable battery voltage

represents less than 2% of the ideal capacity. In some case you may enlarge the battery size.
- Every week is different, and every month is different. Using the same way, you should estimate monthly cycles be less than 15% and yearly cycles to be less sufficient to target a 10 year life. Values differ for each kind of battery, but a 40% cycle can be considered in many cases as a sufficient value.
- Tradeoff network loss and recovery algorithm to be lower than a monthly cycle in terms of battery discharge
- Take pairing and installation into account as a major cycle

2.7 Global System Power Optimization

The power optimization is of tremendous importance when designing an Ultra-Low-Power node and it is very often a never-ending process. We saw all along the chapter some critical points to keep high the energy efficiency. Here are few additional points to check out, in order to still be in your power budget:

- 15 to 30 nA over-leakage on each of the GPIO seems not so much, but given the number of GPIOs, it may lead to overpass your power budget. Programming GPIOs correctly in the lowest possible energy state for long sleep is not always simple, but very often profitable.
- Tradeoff quartz capacitances and precision to avoid wake-up in advance. You need to compensate the local oscillator error of both the emitter and the receiver by doubling the max possible error. The longer the sleep between two synchronizations, the larger the error.
- In the case of synchronized networks, a large temperature change may lead to the wake-up of the node for synchronization. Regarding energy efficiency in such networks, it is better to wake-up a node to keep synchronization than to recover a loss of synchronization.
- Minimize routing activity: routing protocols are very hungry in terms of transactions, limit broadcast transmissions and network maintenance to its minimal activity
- Tradeoff deep-sleep to save more charges than the amount of charge you loose when you discharge external capacitors. In some cases, some higher leakage sleep mode needing less energy to recover, can be more profitable for the energy efficiency
- Tradeoff higher-level software. Usage of DTLS + CoAP seems a good compromise to ensure an IPv6 network with fairly good energy efficiency.
- Checkout the wires between non-powered debug-purpose on board material. Some pernicious leakage could be hidden there.
- When using multiple sensors, check that sensors are really off even if several sensors were running asynchronously.
- In some case, depending on your node architecture, grouping sensors wake-up and radio wake-up enables to reduce wake-up penalty.

Many of those points concern software impact on hardware. Software impact is effectively huge in term of power efficiency. Ultra-low-energy efficient software programming is not an easy task, and is a new domain for many software engineers.

Figure 2.12 gives an overview of the commercially available nodes in November 2015, and provides a comparison with the results with the GreenNet development. Node names depicted in italic are not commercially available.

2.8 Perspectives and Trends

The current generations of IoT nodes on the market are mainly AAA battery operated, consuming 50–100 mW of power to transmit a message at 0 dBm. Most of them are operating thanks to always-on routers. Few of them are IPv6 compliant, few of them offer a secured solution, few other are energy-harvested but none are able to provide all those features at the same time. In the future, the growing multiplicity of sensors, including image sensors, as well as the needs for lower cost, interoperable and secured

Node	MCU #bits	RAM [kB]	CPU ON [mA/MHz]	CPU sleep [µA]	Tx 0dBm [mA]	Rx [mA]	Har-vested	Batt. Size/type [mAh]
GreenNet	32	32	0.185	0.44	4.9	4.5	Y	25 LiMn
Hikob Azure [H14]	32	16	0.180	0.6	12.8	11.8	Y	2000
SmartMeshIP [WDS13]	32	72	0.176	0.8	5.4	4.5	OPT	2AA
M3OpenNode [FIT15]	32	64	1.138	25	11.6	10.3	N	650 LiPo
OpenMote [O15]	32	32	0.438	0.4	24	20	N	2AAA
WisMote [W15]	16	16	0.312	1.69	25.8	18.5	N	2AAA
TelosB [M04]	16	10	1.8	5.1	19.5	21.8	N	2AA
Waspmote15.4 [WD15]	8	8	1.07	7.2	45	50	OPT	N/A
MICAz [C08]	8	4	1.0	<15	17.4	19.7	N	2AA

Fig. 2.12 GreenNet versus commercially available nodes (Varga et al. 2015)

solutions seem to be paving the way towards the use of secured-IPv6 networking solutions. Easier to maintain than proprietary solutions, IETF compliant, enabling complete interoperability, these solutions seem to be the definitive trend of the WSN.

There are mainly three limitations to adopt secured IPv6: first, the need for interoperability was not strong enough during years in the WSN domain. This has evolved thanks to the Thread initiative. The other two reasons are:

- In some case the cost increase due to additional need for more SRAM in the processor to store and process IPv6 frames.
- In some other case the power consumption reached with the current solutions (MCU + radio + IPv6 secured protocols) which leads to either decrease the lifetime of the battery, or increase the battery capacity, impacting the price of the global solution, preventing the usage of energy harvesting as energy source. In order to use energy harvesting with a plain 802.15.4 secured IPv6, we would need an MCU and digital computing part of the radio able to process 5–10 times more data keeping the same power budget. The new generation of MCUs, using CMOS 40 nm Ultra-Low-Power with embedded non-volatile memories technology should enable to offer 90 nm-like retention current with a dynamic power reduction from *x*2 to x5 on the digital part depending on the needed frequency, thanks to DVFS techniques. This technology will definitely help to fix the power consumption issue, on one hand by keeping power consumption at a fair level, compatible with low-cost batteries, and on the other hand by enabling much more computing capability than the previous technology nodes for the same power budget. One step further will be reached few years after thanks to 28FDSOI-ULP technology, achieving to save an additional *x*2 to x10 dynamic power versus CMOS40 (depending on the targeted frequency), keeping the leakage at a fair level. Once again, cost shall be the limiting factor for adoption and only the needs for additional

features like extended signal processing capability, extended memory storage or even graphical display at node level may justify the need for adoption.

At node level, we start to see NFC-pairing adopted by a fair amount of the solutions reaching the market in 2016. Using NFC for pairing and registration should become a must in a near future.

Among the future evolutions of the radio protocol solutions, two major opportunities are foreseen at the time:

- Bluetooth-low-energy is one of them and seems evident when targeting Personal Area Network communication. BLE should move short term to IPv6 however its poor networking capability and its relatively short distance of communication may not enable to replace WSN-oriented radios (802.15.4 family). In order to solve these issues, BLE will eventually offer real mesh network capability and even some longer transmission range option. The success of these options will depend on how it compares to alternatives such as 802.15.4 family.
- The 802.15.4e evolution for industrial applications, part of the 802.15.4-2015 is enabling the IPv6 frequency hopping multichannel mesh network in 2.4GHz networks. This should increase quality of service in crowded environment.

Both those standards may survive together, and we may see a generalization of the radio combos (e.g., BLE + 802.15.4 in 2.4GHz) each solution enabling to target a different network depending on the required service. As an example: a node could use BLE to transmit some information like an image to a smartphone when it is in the range, but would use 802.15.4 or 4e for infrastructure management. Multi-radio nodes sharing the same antenna for cost reason shall become the standard in a near future.

As an alternative to Mesh WSN enabling to repeat the messages from nodes to nodes to cover long distances, star topology networks could be used in so-called Wide Area Network (WAN). Among the existing solutions, you can find Lora, SigFox and Weightless solutions. All of them operate in the Subgig ISM bands, with reduced data rates (100bit/s to 100 kb/s) compared to 2.4 GHz technologies. Business model should be different in this case, as it requires the usage of an existing infrastructure usually managed by operators, enabling some pay-per-use options. Those solutions will certainly survive in parallel to WSN-based solutions, targeting different needs.

References

AN2866 Application Note, How to design a 12.56 MHz customized tag antenna. http://www.st.com/st-web-ui/static/active/cn/resource/technical/document/application_note/CD00221490.pdf

I.M. Atakli, H. Hu, Y. Chen, W.S. Ku, Z. Su, Malicious node detection in wireless sensor networks using weighted trust evaluation, in *Proceedings of the 2008 Spring Simulation Multiconference*, SpringSim'08 (Society for Computer Simulation International, San Diego, 2008), pp. 836–843

B. Awerbuch, R. Curtmola, D. Holmer, C. Nita-Rotaru, H. Rubens, Mitigating byzantine attacks in ad hoc wireless networks. Department of Computer Science, Johns Hopkins University, Technical Report, Version, 1 (2004)

A. Becher, Z. Benenson, M. Dornseif, Tampering with motes: real-world physical attacks on wireless sensor networks, in *Proceedings of the Third International Conference on Security in Pervasive Computing*, SPC'06 (Springer-Verlag, Berlin, 2006), pp. 104–118

H. Chan, A. Perrig, D. Song, Random key predistribution schemes for sensor networks, in *Proceedings of 2003 Symposium on Security and Privacy 2002* (May 2002), pp. 197–213

D. Dolev, A.C. Yao, On the security of public key protocols. IEEE Trans. Inform. Theory **29**(2), 198–208 (1982)

HIKOB, http://www.hikob.com/wp-content/uploads/2015/06/HIKOB_AZURE_LION_ProductSheet_EN.pdf

C. Karlof, D. Wagner, Secure routing in wireless sensor networks: attacks and countermeasures. Ad hoc Netw. **1**(2), 293–315 (2002)

Y.W. Law, L. van Hoesel, J. Doumen, P. Hartel, P. Havinga, Energy-efficient link-layer jamming attacks against wireless sensor network mac protocols, in *Proceedings of the 3rd ACM Workshop on Security of Ad Hoc and Sensor Networks*, SASN'05 (ACM, New York, 2005), pp. 76–88

M. Li, I. Koutsopoulos, R. Poovendran, Optimal jamming attacks and network defense policies in wireless sensor networks, in *26th IEEE International Conference*

on Computer Communications, INFOCOM 2007 (IEEE, May 2007), pp. 1307–1315

F. Liu, X. Cheng, D. Chen, Insider attacker detection in wireless sensor networks, in *26th IEEE International Conference on Computer Communications*, INFOCOM 2007 (IEEE, May 2007), pp. 1937–1945

M3 Open Node motes, https://www.iot-lab.info/hardware/m3/

MICAz mote, http://www.openautomation.net/uploadsproductos/micaz_datasheet.pdf

OpenMote, http://www.openmote.com/hardware/openmote-cc2538-en.html

D.R. Raymond, S.F. Midkiff, Denial-of-service in wireless sensor networks: attacks and defenses. IEEE Pervasive Comput. **7**(1), 74–81 (2008)

M. Rezvani, A. Ignjatovic, E. Bertino, S. Jha, Secure data aggregation technique for wireless sensor networks in the presence of collusion attacks. IEEE Trans. Dependable Secure Comput. **12**(1), 98–110 (2015)

G. Romaniello, Energy efficient protocols for harvested wireless sensor networks. PhD thesis, Universite de Grenoble, March 2015

TelosB motes, http://www4.ncsu.edu/~kkolla/CSC714/datasheet.pdf

H. Tschofenig, T. Fossati, TLS/DTLS Profiles for the Internet of Things. draft-ietf-dice-profile-14. Work in progress, August 2015

P. Urard, G. Romaniello, A. Banciu, J.C. Grasset, V. Heinrich, M. Boulemnakher, F. Todeschni, L. Damon, R. Guizzetti, L Andre, A. Cathelin, A self-powered IPv6 bidirectional wireless sensor & actuator network for indoor conditions, in *2015 Symposium on VLSI Circuits (VLSI Circuits)* (IEEE, June 2015), pp. C100–C101

L.-O. Varga, G. Romaniello, M. Vucinic, M. Favre, A. Banciu, R. Guizzetti, C. Planat et al., GreenNet: an energy-harvesting IP-enabled wireless sensor network. IEEE Internet Things J **2**(5), 412–426 (2015a)

L.-O. Varga, Multi-hop energy harvesting wireless sensor networks: routing and low duty-cycle link layer. PhD thesis, Grenoble Alps University, December 2015

A. Vempaty, O. Ozdemir, K. Agrawal, H. Chen, P.K. Varshney, Localization in wireless sensor networks: byzantines and mitigation techniques. IEEE Trans. Signal Process. **61**(6), 1495–1508 (2012)

M. Vučinić, Architectures and protocols for secure and energy-efficient integration of wireless sensor networks with the internet of things. PhD thesis, Grenoble Alps University, November 2015

Waspmote, http://www.libelium.com/downloads/documentation/waspmote_datasheet.pdf and http://www.digi.com/pdf/ds_xbeemultipointmodules.pdf

T. Watteyne, L. Doherty, J. Simon, K. Pister, Technical overview of SmartMesh IP. in *Proceedings of IMIS'13* (Washington, DC, 2012)

WisMote, http://wismote.org