The Standardization Efforts at the IETF for the Internet of Things

Georgios Papadopoulos

IMT Atlantique
Dépt. Systèmes réseaux, cybersécurité et droit du numérique
2, rue de la Châtaigneraie
CS 17607
35576 Cesson Sévigné Cedex
Téléphone : +33 (0)2 99 12 70 00
Télécopie : +33 (0)2 51 85 81 99
URL : **www.imt-atlantique.fr**

Collection des rapports de recherche d'IMT Atlantique
RR-2010003-SC

Georgios Z. PAPADOPOULOS

Pôle de Recherche Avancée en Communications

The Standardization Efforts at the IETF for the Internet of Things

Rapport prospectif 2018

Date d'édition : 9 juin 2018
Version : 1.6

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Sommaire

Liste des figures

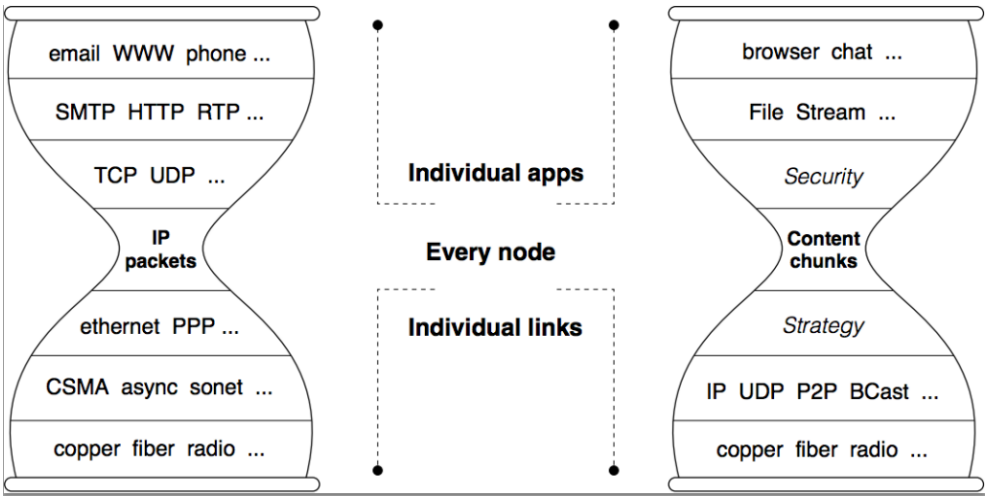


FIGURE 1 – The “thin waist" illustration of the internet.

1. Introduction

During the last years we have experienced the emergence of a new paradigm called the Internet of Things (IoT) in which smart, uniquely identifiable and connected objects (e.g., machines, sensors, actuators) construct a network of things. Those things can communicate between themselves or across existing network infrastructure such as the Internet. They can be deployed nearly anywhere, at homes, hospitals, factories, cities, even within human bodies.

Among the previously presented use-cases, Industry 4.0 is an emerging domain of application for the Internet of Things (IoT), with goals to reduce the management cost and to contribute to the automation of the Operational Technology (OT) found in production chains in factories [1, 2]. Cost reduction can be achieved, in particular, by replacing the existing cables with a wireless medium, as long as an appropriate level of service for critical applications can still be guaranteed at all times. To that aim, the network must exhibit deterministic performance in terms of network reliability and timely delivery [3, 2]. More precisely, an industrial communication framework must provide several nines of reliability in data delivery. For instance, several consecutive losses in an industrial automation control loop are sufficient to stop a production chain. Moreover, it should guarantee a worst case latency for a data packet across the network. This latency must be known in advance, and remain constant throughout the lifetime of the associated path. In order to replace wires, a wireless network should exhibit a high delivery ratio with an ultra-low jitter, regardless of transient variations in the wireless medium and of the network congestion.

The Internet Protocol (IP) can be considered as the “thin waist" of the Internet, see Fig. 1. Indeed, it is the key enabler of the Internet’s explosive evolution and growth during the last 40 years. Thanks to widespread usage of IP version 6 (IPv6) [4], more and more constrained devices are getting connected to the Internet, which eventually converged into a new paradigm called the IoT [5]. There has been essential interest in designing and deploying IoT-based applications, with business sectors ranging from smart grid [6] to industrial IoT [7], where low cost and easily deployed IoT devices can provide significant benefits.

The IoT, also referred to as Low-power and Lossy Network (LLN), is usually composed of hundreds of small, uniquely identifiable and limited in memory capacity objects. Typically, these devices are employed with low-power and lossy communication technologies. In Fig. 2 an example of a 6TiSCH network is illustrated, which comprises a number of sensor and actuator devices that are connected to the 6LoWPAN Border Router (i.e., root) in a multi-hop manner.

At the Internet Engineering Task Force (IETF), a number of Working Groups (WG) have been established to define a set of protocols for various layers of the LLN protocol stack to mitigate any potential issues. CoRE WG defined the web transfer protocol, CoAP [8], ROLL WG specified the routing protocol, RPL [9], 6LoWPAN WG [10] defined the adaptation layer, while 6TiSCH WG [11] focuses on enabling IPv6 over the

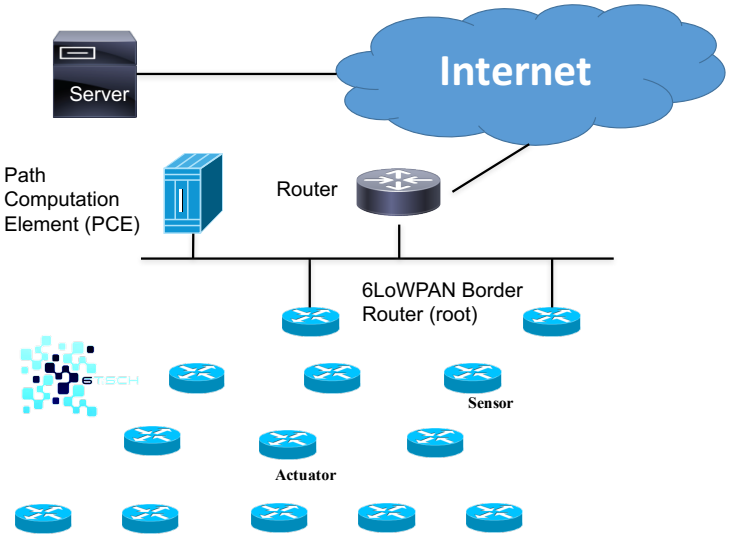


FIGURE 2 – An example of a 6TiSCH Network.

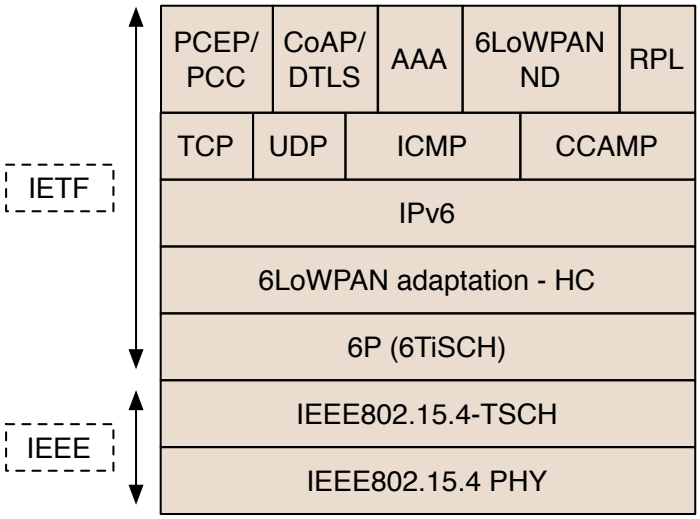


FIGURE 3 – LLN protocol stack.

IEEE802.15.4-TSCH standard [12]. In Fig. 3, an LLN-based stack is depicted. Note that the first low layers are standardized at the Institute of Electrical and Electronics Engineers (IEEE), while the upper layers at the IETF. Within this report the majority of the layers along with their key-protocols will be presented, with special focus given at the IETF layers.

The report is organized as follows. In Section 2, the Medium Access Control layer is described and, more specifically, the IEEE 802.15.4-2015 standard. Then, in Section 3, the 6TiSCH WG is presented along with its fundamental specifications. In Section 4, the 6lo WG is presented, as well as its set of protocols related with the adaptation layer. Then, in Section 5, the ROLL WG is introduced where the well-adopted RPL standard is presented. Section 6 presents the ongoing efforts at the IETF related with the deterministic wireless networks. Furthermore, Section 7 details the key-activities at the CoRE WG. Finally, Section 8 concludes the report.

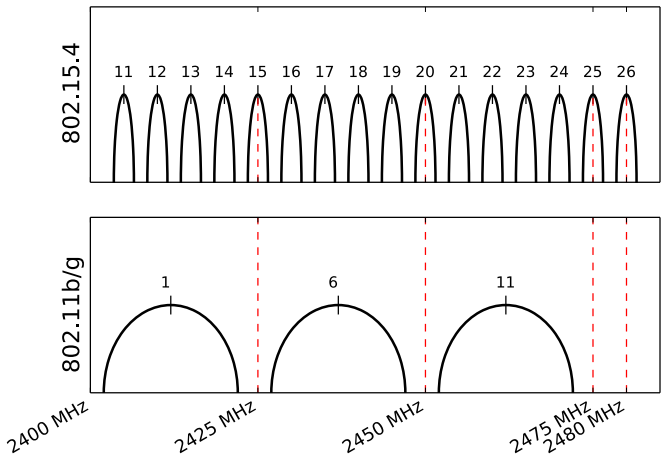


FIGURE 4 – Interfering radio channels : IEEE 802.15.4 and IEEE 802.11 : 1, 6 and 11 are the three non-overlapping and broadly used radio channels for the IEEE 802.11 technology [13].

2. IEEE Std. 802.15.4-2015 TSCH

In 2016 the IEEE 802.15.4-2015 standard [12] was published to offer a certain quality of service for deterministic industrial-type applications. Among the operating modes defined in this standard, Time-Slotted Channel Hopping (TSCH) is a Medium Access Control (MAC) protocol for low-power and reliable networking solutions in LLNs. It comes with 16 radio channels operating at 2.4GHz, where each channel has a bandwidth of 2MHz and a channel separation of 5MHz (see Figure 4). Under TSCH, the communication among the nodes is coordinated by a scheduler. These schedules can be adjusted to the industrial network's requirements and its topology and guarantee collision-free communications.

At its core, TSCH uses Time Division Multiple Access (TDMA) and Frequency Hopping Spread Spectrum (FHSS) techniques to achieve high network reliability, reduce energy consumption and mitigate multi-path fading and the impact of external interference. In a TSCH network, the nodes continuously re-synchronize on a periodic slotframe with their neighbors, based on Enhanced Beacons (EB) packets. The time is sliced into timeslots of equal length, sufficient enough to transmit a data packet and to receive an acknowledgement. If the acknowledgment is not received within the timeslot, the retransmission of the packet will be delayed to one of the following timeslots. At each timeslot, each node is aware if it has to stay "awake" (i.e.,radio ON) in order to transmit or receive a packet, or to "sleep" (radio OFF) to save energy. A set of timeslots constructs a slotframe that repeats perpetually ; according to the standard it consists of 101 timeslots but it is configurable. Furthermore, the timeslots are identified by an Absolute Sequence Number (ASN) counter that increments as time elapses ; the ASN actually counts the number of timeslots since the establishment of the TSCH network. All nodes in the network are aware of the current ASN value.

To define a TSCH schedule, for each radio link a collection of timeslots and channel offsets is assigned, called its cells. A channel offset is a "virtual channel" that is translated into a physical radio channel that is going to be employed for communication. The translation is carried out by a FHSS algorithm :

$$\texttt{frequency} = F(\texttt{ASN} + \texttt{channelOffset}) \,\%\, \texttt{nFreq} \tag{1}$$

where `nFreq` is the number of available physical channels (e.g.,16 when using IEEE802.15.4-compliant radios at 2.4 GHz with all channels in use) [14]. F is a look-up table function that translates the result from the operation to actual radio channel (i.e., from 11th to 26th in 2.4 GHz band). In Fig. 5, a TSCH schedule is depicted.

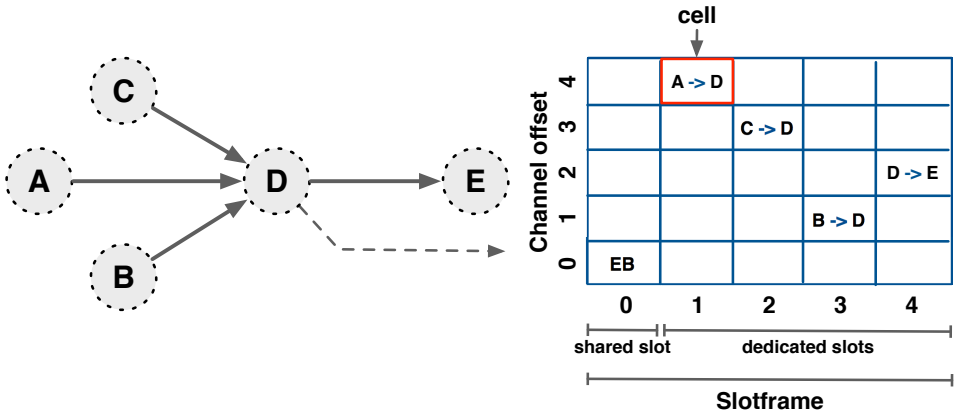


FIGURE 5 – An example of TSCH scheduling for node D. A → D stands for *'A transmits to D'*, while EB cells are used for broadcast and advertisement frames.

3. 6TiSCH WG

As previously was stated, IEEE Std. 802.15.4-TSCH is the emerging standard for industrial automation and process control LLNs. As a result, TSCH is a direct competitor of WirelessHART [15] and ISA100.11a [16]. Therefore it is essential to specify IPv6 over TSCH. IPv6 over the TSCH mode of IEEE 802.15.4e (6TiSCH) WG is a key-enabler of the standardization of IPv6 in industrial environment, as well as the convergence of Operational Technology (OT) with Information Technology (IT).

Note that IEEE802.15.4 only specifies the Layer 2, i.e., link-layer mechanisms. Thus, it does not standardize how the network communication schedule is built and matched to the traffic requirements of the network. The 6TiSCH WG aims at defining protocols to bind IPv6 (i.e.,6LoWPAN) and reservation based MAC layer (i.e.,IEEE 802.15.4-TSCH).

3.1. 6TiSCH Architecture

The 6TiSCH architecture [17] will define the packets belonging to a deterministic IPv6 flow are routed over the multi-hop network within jitter and latency budgets.

3.1.1. Route Computation

Path computation can be achieved either in a centralized manner, e.g.,a node asks a Path Computation Element (PCE) for new cells to use. The PCE is located either on the backbone or farther in the IPv6 network over a backhaul. Alternatively, the route computation can be done in a distributed manner (e.g.,Scheduling Function decides how many cells to allocate based on the local measures).

3.1.2. Track

6TiSCH introduces the concept of *track* [17]. A track corresponds to dedicated radio resources, along with a multi-hop path. More precisely, a set of cells (a *bundle*) is reserved for each hop. Different tracks use a different set of cells (i.e.,different timeslots and/or channels). Thus, by selecting different cells for different data flow, 6TiSCH may provide traffic isolation. A track forwarding scheme is in this case applied : when 6P receives a frame to forward, it automatically finds the outgoing bundle associated with the incoming cells.

In Fig. 6, the flow from A to the border router R, via node B, will be assigned to track 1. Besides, the same node (e.g., B) may forward an additional flow, for instance from C, using a different track (i.e.,2). Since each cell is associated with one single track, we have a kind of label switching : when a packet is received during a cell labelled with a given track, the mote can forward the packet only during a cell with the same track label.

3.1.3. Chunk

Furthermore, 6TiSCH introduces the concept of *chunk* to operate such spectrum distribution for a whole group of cells at a time. Each node is able to separate the scheduling matrix in non overlapping chunks [18]. Then, the chunks are allocated (in a centralized or distributed manner) to different nodes, so that each of them can pick in its churn when it has to allocate new cells.

3.2. 6TiSCH Minimal

6TiSCH minimal scheduling uses a single slotframe, where its size is announced in the EB. There is only one scheduled cell in the slotframe, while the remaining cells are unscheduled. This cell can be scheduled at any timeslot and channel offset within the slotframe. Typically, it is reserved at the beginning of the slotframe to exchange control packets, i.e.,in EB is announced the location of this cell. For instance, Enhanced Beacons (EBs) are transmitted during this cell so that the neighbors may associate with the existing network. Note that the EB are transmitted at the link-layer in broadcast mode and, thus, they are not acknowledged. Furthermore, each device can use the scheduled cell to transmit or receive frames.

3.3. 6top Protocol (6P)

6top is a logical link control plane between the IP layer and the IEEE 802.15.4-TSCH layer. The 6top Protocol (6P) is the 6TiSCH Operation sublayer and runs one or more 6top Scheduling Functions (SFs). 6P enables nodes to communicate with their neighbors in a distributed schedule management. For instance, in the 6TiSCH network illustrated in Fig. 6, a node “A" could send a 6P request to its “parent" node “B". 6TiSCH makes a distinction between the protocol which defines how to negotiate the cells (i.e.,6P [19]) and the algorithm deciding how many cells to allocate in the schedule (e.g.,SF0 [20]). The solution is very flexible since any scheduling algorithm may be practically implemented : a new Scheduling Function has just to be defined and interfaced with 6P.

3.4. Scheduling Functions (SFs)

The Scheduling Function (SF) defines where in the schedule to add or delete a TSCH cell to a neighbor. To do so, SF relies on 6P to negotiate the cells that will be allocated or removed. Depending on the application requirements different SFs could be defined. For instance, some networks require a high level of determinism and reliability while others minimum energy consumption. Note that the devices in a 6TiSCH network may run multiple SFs simultaneously. In each 6P message there should always be the SFID field to allow a node to trigger the appropriate SF.

4. 6lo WG

The IPv6 over Networks of Resource-constrained Nodes (6lo) WG is the successor of the concluded 6LoWPAN WG. The 6LoWPAN WG was replaced by the 6lo WG, which focuses on facilitating IPv6 connectivity over a wider range of radio technologies, such as Bluetooth Low Energy (BLE) RFC 7668 [21] and Z-Wave RFC 7428 [22], that use a base 6LoWPAN stack (RFC 4944 [10], RFC 6282 [23], RFC 6775 [24]) for IPv6 low-power adaptation, stateless header compression and Neighbor Discovery Optimization for reduced multicast and reliable communication.

The IEEE 802.15.4 technology comes with number of limitations, among which are limited communication range and a Maximum Transmission Unit (MTU) of 127 *bytes*. Considering that IPv6 comes with a MTU of 1280 *byte* [4], it would be impossible to transmit an IPv6 datagram over IEEE 802.15.4. Therefore, the IETF 6LoWPAN WG was chartered to fulfill the IPv6 requirements and, thus, enable IPv6 packet transmissions over LLNs [25]. The 6LoWPAN WG specified an intermediate layer between layers two and three, called the IPv6 adaptation layer [10]. The 6LoWPAN adaptation layer defines compression, fragmentation, reassembling and forwarding mechanisms for IPv6 datagrams that do not fit in the MTU of 127 *bytes*.

The compression algorithm is necessary to reduce the 40 *bytes* IPv6 header. Then, fragmentation and reassembly are required to split the large IPv6 datagrams into multiple short fragments. Finally, RFC 4944

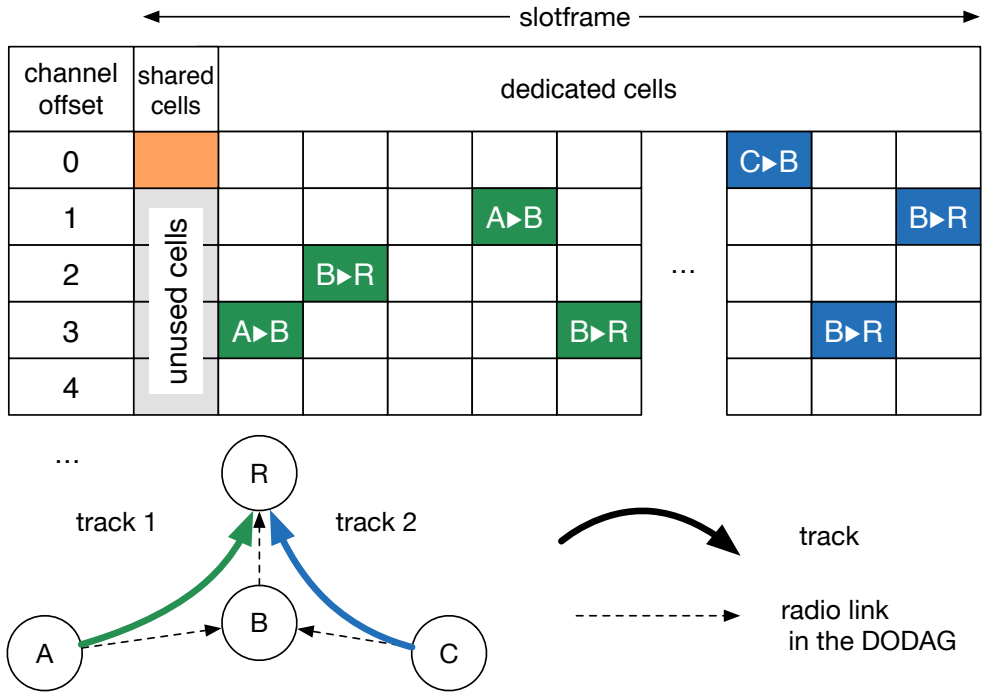


FIGURE 6 – Schedule in a 6TiSCH network, using two different tracks for traffic isolation.

provides hop-by-hop forwarding solutions, where at each hop a node reassembles and fragments again the entire datagram before transmitting to the next hop along the path, which is not the most beneficial for a forwarder. Even though RFC 6282 [23] updates RFC 4944, it does not redefine the 6LoWPAN fragment forwarding method.

4.1. Header Compression

One of the works that the 6LoWPAN WG carried out was related with header compression. Indeed, the adaptation layer standardized in [10] defines header compression schemes that can be employed to compress IPv6, UDP, TCP and ICMPv6 headers.

RFC 4944 comes with two Header Compression (HD) schemes. HC1 is defined for stateless header compression of IPv6 packets. Indeed, based on HC1, the protocol reduces the size by removing fields such as Version, Traffic Class, Flow label. To do so, HC1 uses a combination of the following inputs :

- the low order 64 bits of an IPv6 address (the link-local address) can be the device's MAC address.
- the 802.15.4 frame carries these MAC addresses.
- a number of the fields in the IPv6 header are static.

As a result, by employing the previously stated facts, the HC1 protocol may compress the 40 bytes IPv6 header down to 2 bytes, i.e., including the HC Header byte, in an IPv6 link-local communication (i.e., a direct single-hop communication). Furthermore, RFC 4944 also defines HC2 compression for transport layer compression, i.e.,UDP, TCP and ICMP.

Finally, RFC 6282 [23] specifies two new compression schemes, i.e.,LOWPAN_IPHC and LOWPAN_NHC. Based on shared states, the LOWPAN_IPHC protocol compresses the unique local, global and multicast IPv6 addresses. LOWPAN_IPHC employs a 13-bit encoding field for compression, the last 5 bits of the dispatch byte and an additional byte, and an extra 8 bits to store context information, when necessary. Thus, when considering multi-hop scenarios, where a packet is transported through multiple hops, the LOWPAN_IPHC protocol can compress the 40 bytes IPv6 header down to 7 bytes.

As was previously stated, HC2 can only compress UDP, TCP and ICMPv6 headers. Therefore, RFC 6282 defines the LOWPAN_NHC scheme, that comes with a variable length next header identifier which could be used for future next header compression methods.

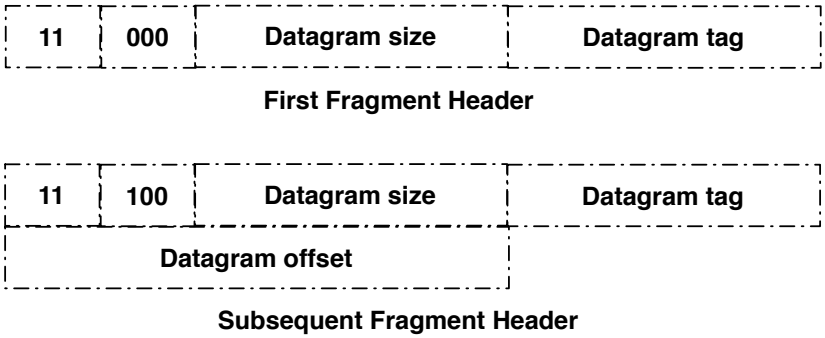


FIGURE 7 – The Fragmentation Header consists of 4 bytes for the first fragment and 5 bytes for subsequent fragments.

4.2. Fragmentation

As it is specified in [10], the fragmentation procedure takes place only when an IPv6 datagram does not fit within a single IEEE 802.15.4 frame. Indeed, if the IPv6 data packet does fit, then it will be transmitted unfragmented and, thus, the LoWPAN encapsulation will not contain the fragmentation header. On the contrary, the transmitter node splits the datagram into multiple link fragments when it does not fit in 127 *bytes*, the maximum physical layer frame size. The length of each link fragment is defined in multiples of eight bytes. To enable the fragmentation and reassembly mechanisms, the fragmentation header comes with the following fields :

- The *datagram size* to identify the size of the IPv6 datagram.
- The *datagram tag* to identify all fragments of a single datagram.
- The *datagram offset* to identify the location of the received fragment.

In Fig. 7, the layout of 6LoWPAN headers are illustrated.

4.3. Reassembly

The receiving node, once it receives the first fragment, it initiates the reassembly procedure to construct the actual IPv6 packet. In order to achieve this, the receiver checks the *datagram tag* field to identify the fragments that belong to a given IPv6 data packet, while it checks the *dataset offset* to identify the offset (i.e., location) of the received fragment within the original datagram. The size of the original unfragmented data packet as well as the size of the reassembly buffer can be identified by the *datagram size* field. Once a node receives a fragment with a certain *datagram tag* value, it starts a reassembly timer. This timeout value must be configured to 60 seconds at maximum, which is the timeout in the IPv6 reassembly procedure [4]. When it expires, if the datagram has not been reconstructed, the received fragments are discarded, while the reassembly procedure is cancelled.

4.4. Fragment Forwarding Schemes in 6LoWPAN

Furthermore, 6LoWPAN comes with two Fragment Forwarding (FF) mechanism, ***mesh under*** and ***route over***. The first scheme is operating at the adaptation layer, while the later at the network layer.

4.4.1. Mesh under

In mesh-under operation, the network (i.e., IP) layer does not proceed with any IP routing. In fact, the 6LoWPAN adaptation layer executes the routing and forwarding over the mesh network. Indeed, to transmit a datagram to a certain destination, the EUI 64 bit address or the 16 bit short address is employed. Thus, in order to forward the received frame, the 6LoWPAN layer employs the mesh header as well as the link layer source and destination addressees that are included in the IEEE 802.15.4 header. Therefore, it is not necessary in fact to unpack the IPv6 header

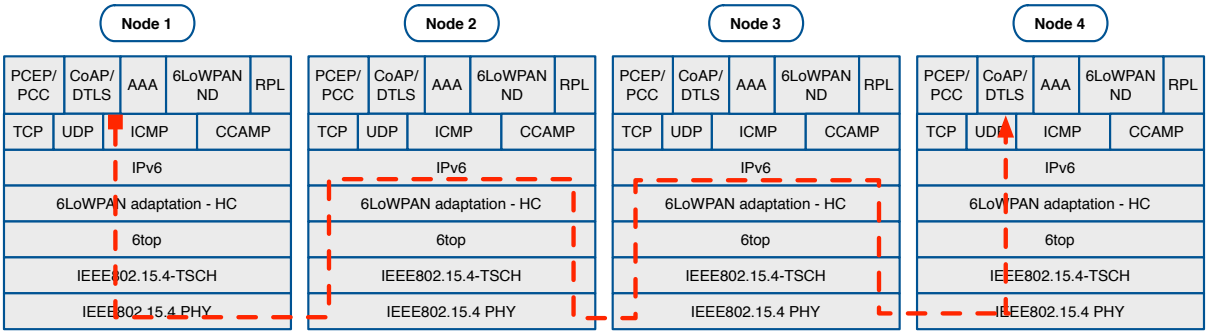


FIGURE 8 – L3 forwarding : IPv6 forwards the fragments over multiple hops.

4.4.2. Route Over

In the route-over scheme, all routing and forwarding operations are performed in the network layer. Considering that IEEE 802.15.4 [12] is operating over mesh networks, often a routing protocol is operating at Layer 3. For example, the well known IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) [9] is one of the most adopted protocol for routing packets in a multi-hop network. Each node can act as a relay for others. Thus, since the frames are routed at the network layer in [10], it is straightforward that the 6LoWPAN adaptation layer processes the received fragments at each hop to reassembly the original IPv6 data packet and then to fragment again, before forwarding to the next hop as it is depicted in Fig. 8. In order to achieve this, in addition to the hop-by-hop source and destination, the link-layer addresses of the transmitter and the final receiver should be included.

4.5. LLN Minimal Fragment Forwarding

Recently, in the 6lo WG, a design team was established to tackle the previously listed problems from [10]. They published a new Internet Draft that proposes a Virtual Reassembly Buffer (VRB) technique which reduces the end-to-end delay while it improves the reliability in route-over forwarding [26]. The VRB method allows 6LoWPAN fragments to be delivered over multiple hops without requiring actual fragmentation or reassembly at each hop.

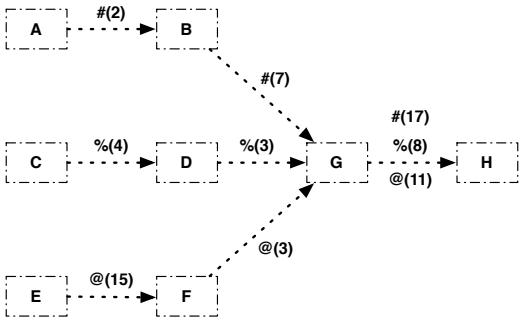
4.5.1. Fragment Forwarding

At its core, VRB allows a node to immediately forward a fragment that it receives, without reassembling the complete packet first. Originally, this concept was introduced in [27], where once a node obtained all required information about the data, it may retransmit it in the form of a forwarded fragment. To do so, it is necessary that all fragments are transmitted with the same outgoing address and datagram tag, otherwise the final receiver will not be able to proceed with a full reassembly and, thus, it would discard the received fragments.

Each node in the network maintains a VRB table, similarly to a switching table with a maximum pre-allocation memory which is implementation dependent. In the beginning, all VRB tables are empty. For instance, in Fig. 9, when node G receives the first fragment from node B with datagram $tag = 7$, it analyzes the content of the fragment to find out the IPv6 destination address. If it is not the final destination, it identifies the next hop node to forward the fragment. To do so, it builds an entry in the VRB table with the following 4 fields :

- link-layer address of the previous hop
- locally unique datagram tag of the received fragment
- link-layer address of the next hop
- locally unique datagram tag for the transmitting fragment

Thus, the following fragments that match the “incoming" columns, of the receiving node VRB table, are forwarded based on the “outgoing" columns. As a result, based on virtual reassembly buffer at each node, the packets are virtually reassembled and fragmented, without actually reserving a memory.



incoming		outgoing	
L2 src	tag	L2 dest	tag
B	7	F	17
D	3	F	8
F	3	F	11
empty			

Node G's VRB table

FIGURE 9 – VRB table of node G : #(2), %(4) and @(15) are fragments from packets coming from nodes A, C and E, with datagram_tag configured to 2, 4 and 15, respectively.

Finally, once the last fragment is forwarded to the next hop, the node may clear the entry in its VRB table. However, if the last fragment is never received, the node may set a timer maximum of 60 seconds, as it is defined in [10], and the VRB table entry may be cleared after the expiration of the timeout period.

5. ROLL WG

5.1. Routing Over Low power and Lossy networks (roll)

The ROLL Working Group is working on standardizing routing solutions for connected smart buildings and smart city networks. The WG aims for high network reliability in very large networks (several thousands of nodes) and harsh environments (interfering technologies). Note that ROLL focuses on IPv6 only.

The ROLL WG continues to work on routing issues for LLNs and on maintaining and improving the already developed and standardized RPL protocol. Moreover, the WG will particularly focus on enabling deterministic networks and on routing security issues. Finally, ROLL works closely with the working groups from other areas related with embedded nodes and constrained networks, such as 6lo, 6TiSCH, and CoRE.

5.2. RFC 6550 : RPL

In order to extend the network beyond the radio coverage of one node, a mesh technology enables a node to act as a relay for others, but, beyond one hop, it will require a protocol for routing packets throughout the network. Note that the constrained devices with limited memory and processing resources, can be interconnected by a variety of links, such as IEEE 802.15.4, Bluetooth, Low Power WiFi or Power Line Communication (PLC) links. LLNs are transitioning to an end-to-end IP-based solution to allow interoperability across the networks. To enable and manage such interconnection, ROLL WG standardized the IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) [9], which is one of the most adopted routing protocols for the IoT. The summary of the RPL terminology is illustrated in Fig. fig :RPLterminology.

5.2.1. Proactive routing protocol

In proactive routing protocols, routes are built *a priori* and, as a result, all nodes in a network are aware of the routes to any destination at any time. Thus, a node may transmit a data packet to any destination with no delay, since all routes are stored in the routing tables. However, periodic routing-related control packets need to be transmitted to maintain the routing table updated. Furthermore, to control the network overload, the periodicity at which these control packets must be accurately defined.

The Routing Protocol for LLNs (RPL) [9] is today the main protocol in the proactive family of routing protocols chosen in Low power and Lossy Network (LLN). It is actually a distance vector routing protocol specified by the ROLL WG [28]. RPL is defined as Link-layer agnostic, so it can operate over Wireless or Power Line Communication (PLC) networks for example.

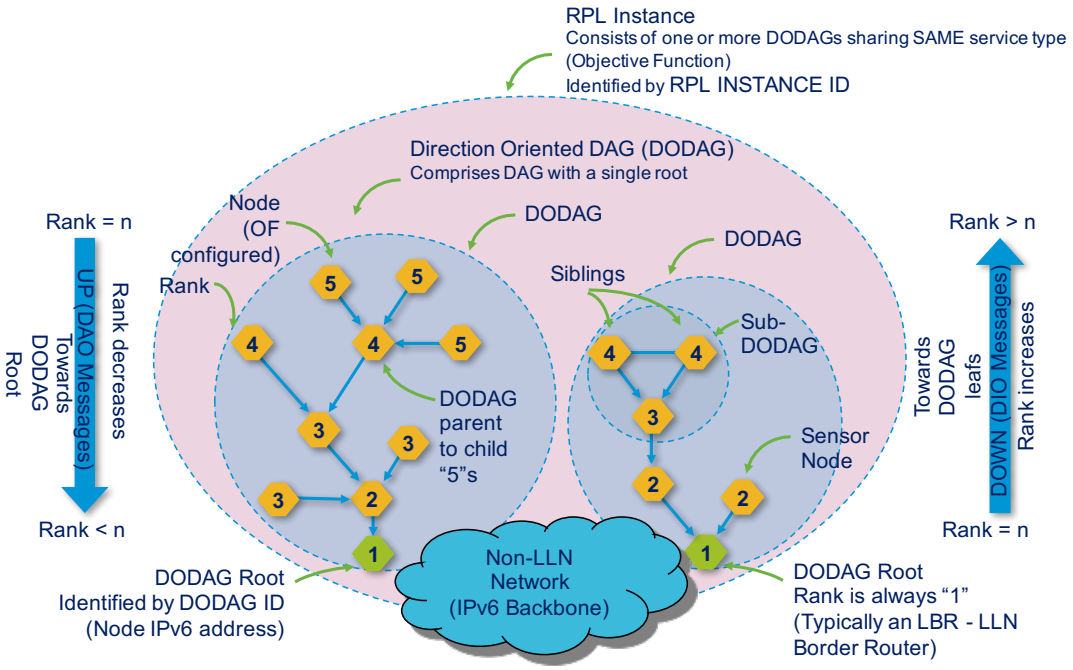


FIGURE 10 – Summary of RPL terminology.

5.2.2. Topology management under RPL

In a LLN, the topology is not predefined and, thus, RPL is in charge of discovering and carefully selecting nodes in order to construct optimal routes. The topology is organized based on a Directed Acyclic Graphs (DAG), a graph where the connections between nodes have a direction and a "non-circular" property. Based on the "acyclic" nature of the DAG, the graph comprises at least one root, a node with no outgoing edge. In Figure 11 (a), a DAG composed of ten nodes and three DAG roots is illustrated. To construct a routing topology, RPL employs an extension of DAG : the Destination Oriented DAG (DODAG) which is similar to DAG with single DAG root. To this aim, RPL assigns to each device participating in the routing a *rank*, i.e.,a metric which denotes the virtual distance from the root. In a smart grid scenario, the root of a RPL network could be the data concentrator that gathers the metering information. Figure 11 (b) depicts a DODAG topology that consists of eight nodes with one root.

To establish and maintain routes, RPL uses three different types of ICMPv6 control packets :

- DAG Information Object (DIO)
- DAG Information Solicitation (DIS)
- Destination Advertisement Object (DAO)

The upward route construction, the one used between smart meters and the core network, is managed by transmitting DIO messages in multicast. DIO messages contain information that allows discovering the set of parents, calculating its own rank. When a node receives a DIO message, it computes a new rank, and compares it with its current rank. Then, if the newly calculated rank is smaller than the current, it will add the transmitter’s address in its set of potential parents. From this set of parents, a node chooses its preferred parent as the node from which its rank will be minimal. The rank contained in the DIO message is the rank of the node sending the DIO message and determines the relative position of a node in the DODAG. The rank is computed by the objective function using routing metrics and its purpose is to avoid loops. The downward route construction, which is optional in RPL, is managed by the DAO messages to propagate information about the destination in the upward direction. To construct the downward routes, there are two modes : Storing and Non-Storing. Finally, DIS control packets are utilized to solicit a DIO message from a RPL node.

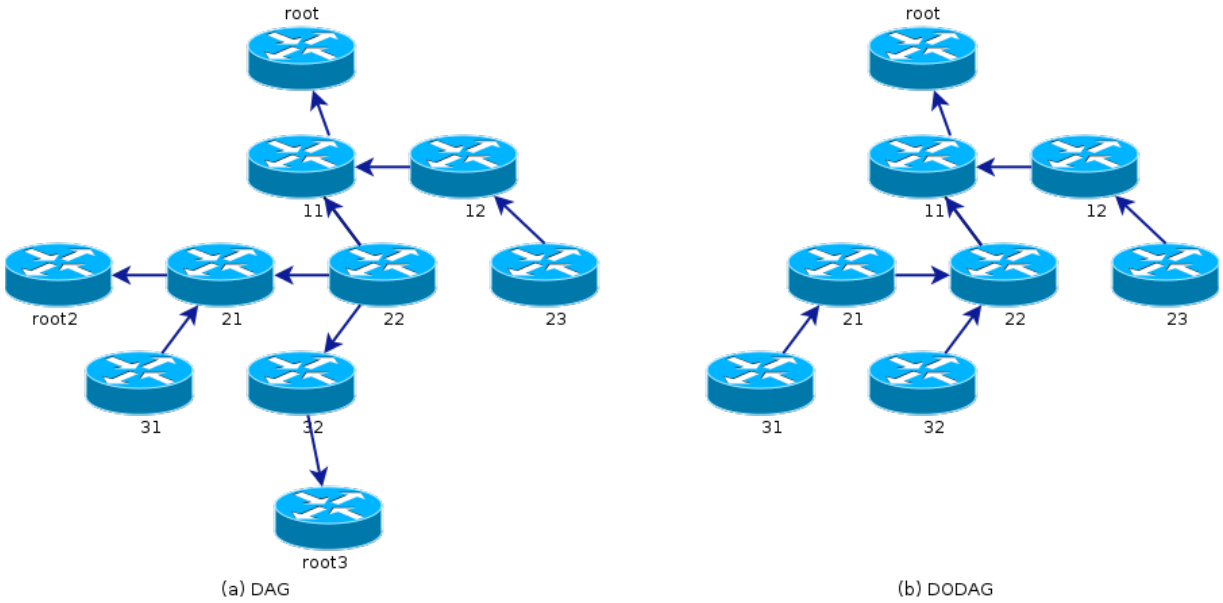


FIGURE 11 – Example of a DAG and a DODAG.

5.2.3. Routing table maintenance under RPL

As previously stated, DIO messages are periodically transmitted to build and maintain the RPL DODAG. However, if the network is stable, the DIO message frequency is decreased to reduce the overhead of signaling messages. On the contrary, if the condition of the network is not stable, more DIO messages have to be transmitted. This timing function is called the Trickle timer [29]. If a received DIO message does not imply any change on the receiver in terms of rank, parent set or preferred parent, the DIO is considered consistent. As long as consistent messages are received, the interval between DIO messages is exponentially doubled to reduce the overhead of periodic messages. Conversely, when the network is not stable and DIO messages are inconsistent with the known topology, more DIS and DIO messages are needed to update the node routing tables. Messages such as multicast DIS without a solicited information option or DIO messages containing infinite rank are considered inconsistent, and cause the trickle timer to reset, and the interval time is set to its minimum value. The Trickle algorithm allows to be reactive in case of a change or failure in the network while minimizing the overhead when the network is stable.

For the downward route construction, a DelayDAO is sent to govern the emission of the DAO messages. At each transmission of a DAO message, a random interval is chosen before the actual transmission.

5.2.4. Routing strategy : metrics and constraints

A metric in RPL is a quantitative value, and it is used to evaluate the path cost. Vasseur et al. [30] define two kinds of metrics that can be used for path calculation :

- The **link metric** that concerns the link's attributes e.g., Link Quality Level (LQL), Expected Transmission Count (ETX), latency, throughput.
- The **node metric** that takes into account the Node State and Attribute (NSA) such as energy (remaining energy, power source) or min-hop (number of hops to the root).

RPL supports also constraint-based routing where the constraint may be applied on both link and nodes. If a link or a node does not satisfy a constraint, it is discarded from the parent set.

This constraint is used to include or eliminate a link or a node that not meet a specific criteria. For instance, the objective function will not choose a path that traverses a node that is battery-powered or a link with low ETX. RPL objective function could combine metrics and constraints to compute the best path.

5.2.5. Path computation under RPL

To compute the optimal path, the objective function plays a major role in RPL protocol. To this aim, the two following algorithms have to be defined :

- the computation of the node's rank according to one or several metrics
- the parent selection operation according to metrics and constraints

Two objective functions have been defined by the ROLL working group : Objective Function Zero (OF0) and Minimum Rank with Hysteresis Objective Function (MRHOF) that are presented next.

The Objective function zero

The OF0 [31] works by computing the rank based on the addition of a scalar, representing the link properties to the rank of the preferred parent. The scalar value is normalized between 1 and 9 for expressing the link properties with 1 for excellent, and 9 for very poor. Note that any kind of metric could be used for the scalar value. This objective function allows for finding the closest grounded root (a root that offers connectivity to the application goal) by selecting a preferred parent and a backup successor if available. The rank computation is given by the algorithm below :

$$R(N) = R(P) + rank\_increase \tag{2}$$

$$rank\_increase = ((Rf * Sp + Sr) * MinHopRankIncrease) \tag{3}$$

where :

- R(P) is the preferred parent's rank
- Sp (the step_of_rank), Sr (stretch_of_rank) and Rf (rank_factor) are respectively the expression of the link properties normalized between 1 and 9, the maximum augmentation to the step_of_rank of a preferred parent to allow the selection of an additional feasible successor and a value used to increase the importance of the link properties.
- MinHopRankIncrease is a multiplying factor that plays a major role in the rank computation by reflecting the impact of the metric on the rank increase. The default value is **256** as it is described in[9].

OF0 parent selection is governed by several rules (see Section 4.2.1 of [31]), but the most important is that the selected parent must be the one that causes the lesser resulting rank for the node. This selected parent become the "preferred" parent.

The Minimum Rank Hysteresis Objective Function

MRHOF [32] optimizes the path to the root that minimizes a defined metric. However, it avoids to change this path frequently. Light metric variations cause changes in the network that are decreased by introducing an hysteresis. MRHOF works with additive metrics and introduces the path cost for the rank computation, that specifies the property of the path to the root regarding the employed metric. The path cost is calculated by the sum of the path cost advertised by the parent and the link metric cost to the parent.

The rank computation for MRHOF is given by the algorithm below :

$$path_{cost} = parent_{path\_cost} + link\_cost \tag{4}$$

$$rank = func(path_{cost}) \tag{5}$$

where :

- $parent_{path\_cost}$ is advertised by the parent and represents the path cost of the parent.
- link_cost is the cost associated with the parent's link regarding to the selected metric.

MRHOF parent selection is governed by an hysteresis function given by the equation below where $P1_{path\_cost}$ and $P2_{path\_cost}$ being respectively the path cost to Parent 1 and Parent 2. PP is the selected parent designated as Preferred Parent. P1 is the current best parent and P2 is a candidate parent.

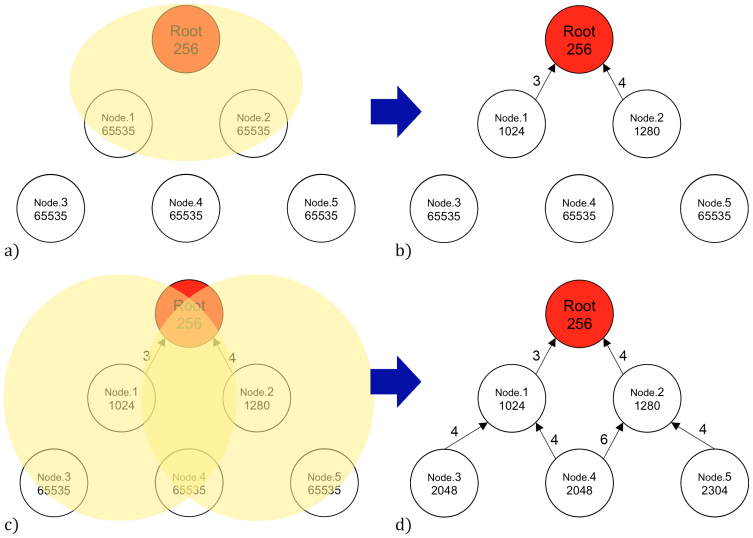


FIGURE 12 – Example of an Upward route construction with RPL.

$$PP = \begin{cases} \text{P2} & if \quad P1_{\text{path\_cost}} + Threshold > P2_{\text{path\_cost}} \\ \text{P1} & else \end{cases} \tag{6}$$

where *Threshold* is the hysteresis function, i.e., the minimum difference between the cost of the path through the preferred parent and the cost path of a candidate parent to trigger the selection of a new preferred parent. This objective function allows for selection of the route towards the root with the lowest path cost, e.g., minimum hop counts if the hop-count metric is used.

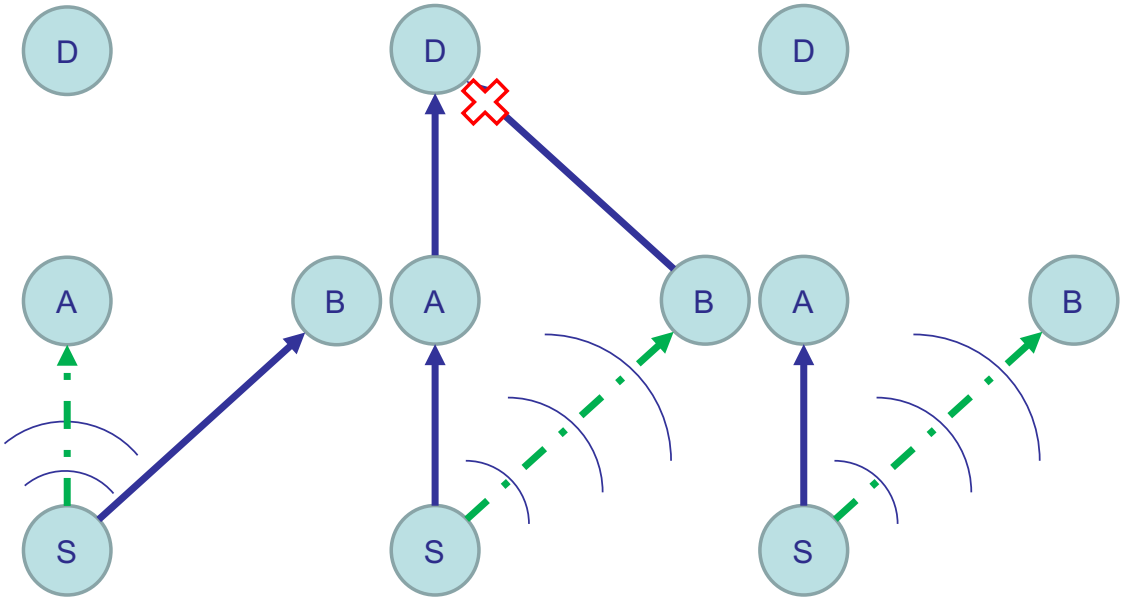
5.2.6. Summary of the RPL DODAG construction

Figure 12 shows an example of the upward route construction using hop-count metric. Once the trickle timer is expired, RPL root will broadcast a DIO message, containing its rank. Nodes in the coverage area of the root (i.e., yellow circles) will receive the DIO message and process it. If the DIO message had been corrupted, it would have been discarded. Since the root is the sink of the network, nodes 1 and 2 can not be closer to the root so they will add the root as their preferred parent and compute their rank. To test if a candidate neighbor is eligible to be a preferred parent, a node will verify if the rank contained in the received DIO message added to a RPL parametric value (min_hop_rank_increase) is less than its rank. Then node 1 and 2 will broadcast their own DIO message with their new computed rank. Note that since the root has a smaller rank than the one advertised in nodes 1 and 2 DIO messages, nodes 1 and 2 will not be considered as potential parents for the root. It is worth mentioning that ranks shown under node names in this example depend on the objective function and values shown beside edges represent the link quality (i.e., ETX). The arrows between nodes represent the upward route and when a node installed at least one of them, it is considered to have joined the DODAG. It has to be noted that a node may either stay silent and wait for a DIO message or it may send a DIS message during the initialization process.

6. Deterministic Wireless Networks

Some applications (such as Wireless Industrial IoT) require a robust communications framework that guarantees data packet delivery in a given delay. For example, a periodic process may need to be repeated identically every time. To reach this ambition, the network must not only be reliable but also deterministic.

A deterministic network ensures that the transported data packet will be carried out in a pre-defined and in a tight window of time, whatever the quality of the wireless links and the network congestion. The goal of such network is to exhibit ultra-low jitter performance, i.e., close to 0. IEEE std. 802.15.4 [12]



(a) Packet Replication : Data packet is transmitted to both Default & "Alternate" Parent.

(b) Packet Elimination : node D discards the duplicated packet "previously received packet".

(c) Promiscuous Overhearing : any neighbor of a transmitter may overhear a transmission.

FIGURE 13 – Packet Replication and Elimination operations.

has a provision to provide guarantees for deterministic networks. Time-Slotted Channel Hopping (TSCH) provides a transmission schedule to avoid random access to the medium and channel diversity to fight interference. However, TSCH is prone to retransmissions when the actual transmission was unsuccessful, due to external interference or potential collision and, consequently, it increases the end-to-end delay performance.

Building on the 6TiSCH architecture, the drafts submitted in 6TiSCH [33] and ROLL [34] WGs leverage PRE to improve the Packet Delivery Ratio (PDR), provide a hard bound to the end-to-end latency, and limit jitter. Furthermore, number of Internet Drafts [35], [36] are submitted over several IETF WGs, (e.g., 6TiSCH, ROLL and 6lo), to define set of protocols that enable deterministic behavior in a LLN by employing the multipath strategy.

6.1. Packet Replication and Elimination principles

In a nutshell, PRE consists in establishing several paths in a network to provide redundancy and parallel transmissions to bound the delay to traverse the network. Optionally, promiscuous listening between paths is possible, such that the nodes on one path may overhear transmissions along the other path. Considering the scenario depicted in Fig. 13, two different paths are possible for S to reach D. A simple way to take benefit from this topology could be to use the two independent paths via nodes A and via B. The 6TiSCH PRE may also take advantage to the shared properties of the medium to compensate for the potential loss that is incurred with radio transmissions. For instance, when the source sends to A, B may listen and get a second chance to receive the frame without an additional transmission. Note that B would not have to listen if it already received that particular frame at an earlier time slot.

PRE model can be implemented in both centralized and distributed scheduling approach. In the centralized approach, a scheduler calculates the routes and schedules the communication among the nodes along a circuit such as a Label switched path. In the distributed approach, each node selects its route to the destination. In both cases, a default parent and alternate parent(s) should be selected to set up complex tracks.

In the following subsections, detailed description of all required operations defined by PRE, namely, Packet Replication, Packet Elimination and Promiscuous Overhearing, will be described.

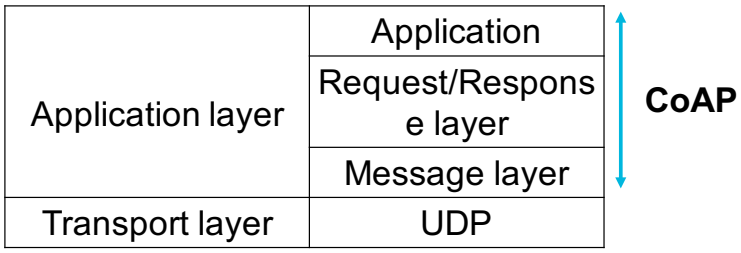


FIGURE 14 – CoAP architecture.

6.2. Packet Replication

The objective of PRE is to offer deterministic networking properties, with high reliability and bounded latency. To achieve this goal, determinism in every level of the forwarding path should be guaranteed. By employing Packet Replication procedure, each node transmits (i.e., replicates) each data packet to both its Default Parent (DP) and Alternative Parent (AP). To do so, each node (i.e., source and intermediate 6TiSCH nodes) transmits the data packet twice in unicast to each parent. For instance, in Fig. 13a, the source 6TiSCH node S is transmitting the packet to both parents, nodes A and B, in two different timeslots within the same TSCH slotframe. Thus, the packet eventually obtains parallel paths to the destination.

6.3. Packet Elimination

The replication operation increases the traffic load in the network, due to packet duplications. Thus, Packet Elimination operation should be applied at each RPL DAG level to reduce the unnecessary traffic. To this aim, once a node receives the first copy of a data packet, it discards the following copies. Because the first copy that reaches a node is the one that counts, and thus will be the only copy that will be forwarded upward, see Fig. 13b.

6.4. Promiscuous Overhearing

Considering that the wireless medium is broadcast by nature, any neighbor of a transmitter may overhear a transmission. By employing the Promiscuous Overhearing operation, DP and AP eventually have more chances to receive the data packets. In 13c, when node S is transmitting to its DP (node A), the AP (node B) may decode this data packet as well. As a result, by employing correlated paths, a node may have multiple opportunities to receive a given data packet. This feature not only enhances the end-to-end reliability but also it reduces the end-to-end delay.

7. CoRE WG

Constrained RESTful Environments (CoRE) WG was established back in 2010. Its primary focus was to define a framework for resource-oriented applications intended to run on constrained IP-based networks. In other words, the main goal was to enable RESTful embedded web-based services, similarly to the traditional web services, while meeting requirements of multicast support, low overhead, and simplicity. As a result, the Constrained Application Protocol (CoAP) was standardized [8], a RESTful web transfer protocol for use with embedded devices and constrained networks. The CoRE WG, instead of going through HTTP [37] compression, specified a subset of the RESTful features, making it interoperable with HTTP.

Unlike HTTP, CoAP employs UDP datagram-oriented transport protocols. CoAP defines two-layer architecture, the *Message layer* to ensure reliable transmission and sequencing over UDP and the *Request/Response layer* to map the requests to responses, see Fig. 14.

7.1. Message layer

The CoAP message layer controls message exchanges over UDP between two devices, as well as to deal with asynchronous interactions with UDP . Each message is tagged with an ID to detect duplicates and for reliability. CoAP comes with four types of messages which are defined in the header :

- **Confirmable (CON) :** messages for which a response is required.
- **Non-confirmable (NON) :** messages for which a response is not required.
- **Acknowledgment (ACK) :** a confirmation message.
- **Reset (RST) :** when a CON message cannot be processed.

7.2. Request/Response layer

The Request/Response layer is employed to send resource operation requests and response data. CoAP request and response semantics are included in CoAP messages. Furthermore, request and response data, such as the Universal Resource Identifier (URI) and payload content-type are sent as CoAP options. Considering that CoAP is running over the non-reliable UDP transport protocol, the messages may either be lost or to arrive out of order or even duplicated. Therefore CoAP comes with reliability mechanisms to tackle the previously mentioned issue, such as duplicate detection, stop-and-wait retransmission reliability and multicast support.

7.3. Summary

Finally, to summarize, the main features defined by CoAP are the following [8] :

- Constrained web protocol specialized to M2M requirements.
- Stateless HTTP mapping through the use of proxies or direct mapping of HTTP interfaces to CoAP.
- UDP transport with application layer reliable unicast and best-effort multicast support.
- Asynchronous message exchanges.
- Low header overhead and parsing complexity.
- Each resource corresponds to URI and Content-type support.
- Simple proxy and caching capabilities.

8. Conclusions

The Industrial Internet of Things (IIoT) is a fairly emerging concept in the wireless communication domain. IP-based communication with the large number of applications developed over the years, and support by open standards, allows 6TiSCH-based technologies to be well positioned in the fast growing IoT market. Its features make the information technology ideal for business use-cases such as smart home automation with sensors and actuators, smart grid with smart metering, smart cities with street and residential lighting and monitoring.

These upcoming LLNs based on 6TiSCH architecture are building on the IEEE Std. 802.15.4-2015 advantages, such as enabling large multi-hop network, reliable communication and low power consumption. The aim of this report has been to provide the communication architecture (over 2.4GHz) for LLNs that are based on embedded and constrained devices. Therefore, this report presents a detailed overview of the IETF community on standardized architecture on LLNs which eventually will replace the proprietary approaches by a transparent end-to-end architecture.

The main WGs that are involved to enable LLNs communication were introduced. Indeed, the 6TiSCH minimal scheduling and the Scheduling Function was presented along with the 6TiSCH WG overview. Next was the 6lo WG, where the header compression and fragment forwarding was discussed. Afterwards, the ROLL WG was presented along with the well-adopted RPL routing protocol for LLNs. In the following section, the ongoing works at the IETF related with the deterministic networks was discussed. Finally, the CoRE WG and its key-features at the application were presented.

Références

[1] E. Grossman, C. Gunther, P. Thubert, P. Wetterwald, J. Raymond, J. Korhonen, Y. Kaneko, S. Das, Y. Zha, B. Varga, J. Farkas, F.-J. Goetz, J. Schmitt, X. Vilajosana, T. Mahmoodi, S. Spirou, P. Vizarreta, D. Huang, X. Geng, D. Dujovne, and M. Seewald, "Deterministic networking use cases," Working Draft, IETF Secretariat, Internet-Draft draft-ietf-detnet-use-cases-13, September 2017, https://tools.ietf.org/id/draft-ietf-detnet-use-cases-11.txt. [Online]. Available : https://tools.ietf.org/html/draft-ietf-detnet-use-cases-13

[2] L. D. Xu, W. He, and S. Li, "Internet of things in industries : A survey," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 4, pp. 2233–2243, Nov 2014.

[3] S. Yamamoto, T. Emori, and K. Takai, "Field wireless solution based on isa100. 11a to innovate instrumentation," Yokogawa Technical Report English Edition, Tech. Rep., 2010.

[4] S. Deering and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification," RFC 2460, Dec. 1998.

[5] M. Wollschlaeger, T. Sauter, and J. Jasperneite, "The Future of Industrial Communication : Automation Networks in the Era of the Internet of Things and Industry 4.0," *IEEE Industrial Electronics Magazine*, vol. 11, no. 1, pp. 17–27, March 2017.

[6] H. K. N. P. M. F. K. A. Paventhan, B. D. Darshini and A. Jain, "Experimental evaluation of IETF 6TiSCH in the context of Smart Grid," in *Proceedings of the 2nd IEEE World Forum on Internet of Things (WF-IoT)*, December 2015, pp. 530–535.

[7] X. V. D. Dujovne, T. Watteyne and P. Thubert, "6TiSCH : deterministic IP-enabled industrial internet (of things)," *IEEE Communications Magazine*, vol. 52, no. 12, pp. 36–41, December 2014.

[8] Z. Shelby, K. Hartke, C. Bormann, and B. Frank, "Constrained Application Protocol (CoAP)," IETF CoRE Working Group, Feb. 2011.

[9] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, and A. R., "RPL : IPv6 Routing Protocol for Low-Power and Lossy Networks," RFC 6550, 2012.

[10] G. Montenegro, N. Kushalnagar, and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks," RFC 4944, September 2007.

[11] P. Thubert, T. Watteyne, M. R. Palattella, X. Vilajosana, and Q. Wang, "IETF 6TSCH : Combining IPv6 Connectivity with Industrial Performance," in *Proceedings of the 7th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS)*, July 2013, pp. 541–546.

[12] "IEEE Standard for Low-Rate Wireless Personal Area Networks (LR-WPANs)," IEEE Std 802.15.4-2015 (Revision of IEEE Std 802.15.4-2011), April 2016.

[13] D. Zorbas, G. Z. Papadopoulos, and C. Douligeris, "Local or global radio channel blacklisting for ieee 802.15.4-tsch networks ?" in *Proc. of IEEE ICC*, 2017, pp. 1–6.

[14] T. Watteyne, M. Palattella, and L. Grieco, "Using IEEE 802.15.4e Time-Slotted Channel Hopping (TSCH) in the Internet of Things (IoT) : Problem Statement," RFC 7554, 2015.

[15] "WirelessHART Specification 75 : TDMA Data-Link Layer," *HART Communication Foundation Std., Rev. 1.1,*, vol. HCF SPEC-75, 2008.

[16] "Isa100.11a technology standard."

[17] P. Thubert, "An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4," draft-ietf-6tisch-architecture-14, April 2018.

[18] D. Dujovne, T. Watteyne, X. Vilajosana, and P. Thubert, "6tisch : deterministic ip-enabled industrial internet (of things)," *IEEE Communications Magazine*, vol. 52, no. 12, pp. 36–41, December 2014.

[19] Q. Wang and X. Vilajosana, "6top protocol (6p)," draft-ietf-6tisch-6top-protocol-02, July 2016.

[20] D. Dujovne, L. Grieco, M. Palattella, and N. Accettura, "6TiSCH 6top Scheduling Function Zero (SF0)," draft-dujovne-6tisch-6top-sf0-00, March 2016.

[21] J. Nieminen, T. Savolainen, M. Isomaki, B. Patil, Z. Shelby, and C. Gomez, "IPv6 over BLUETOOTH(R) Low Energy," RFC 7668, October 2015.

[22] A. Brandt and J. Buron, "Transmission of IPv6 Packets over ITU-T G.9959 Networks," RFC 7428, February 2015.

[23] J. Hui and P. Thubert, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks," RFC 6282, September 2011.

[24] Z. Shelby, S. Chakrabarti, E. Nordmark, and C. Bormann, "Neighbor discovery optimization for ipv6 over low-power wireless personal area networks (6lowpans)," IETF RFC 6775, November 2012.

[25] N. Kushalnagar, G. Montenegro, and C. Schumacher, "IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs) : Overview, Assumptions, Problem Statement, and Goals," IETF RFC 4919, August 2007.

[26] T. Watteyne, C. Bormann, and P. Thubert, "LLN Minimal Fragment Forwarding," draft-watteyne-6lo-minimal-fragment-00, February 2018.

[27] Z. Shelby and C. Bormann, *6LoWPAN*. John Wiley & Sons, November 2009.

[28] E. Ancillotti, R. Bruno, and M. Conti, "Reliable data delivery with the IETF routing protocol for low-power and lossy networks," *IEEE Trans. Industrial Informatics*, vol. 10, no. 3, pp. 1864–1877, 2014.

[29] P. Levis, T. Clausen, J. Hui, O. Gnawali, and J. Ko, "The Trickle Algorithm," RFC 6206, March 2011.

[30] J. Vasseur, M. Kim, K. Pister, N. Dejean, and D. Barthel, "Routing metrics used for path calculation in low-power and lossy networks," RFC 6551, March 2012.

[31] P. Thubert, "Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL)," RFC 6552 (Proposed Standard), Internet Engineering Task Force, Mar. 2012.

[32] O. Gnawali and P. Levis, "The Minimum Rank with Hysteresis Objective Function," RFC 6719 (Proposed Standard), Internet Engineering Task Force, Sep. 2012. [Online]. Available : http://www.ietf.org/rfc/rfc6719.txt

[33] G. Papadopoulos, N. Montavont, and P. Thubert, "Exploiting Packet Replication and Elimination in Complex Tracks in 6TiSCH LLNs," IETF, draft-papadopoulos-6tisch-pre-reqs-01, December 2017.

[34] R. Koutsiamanis, G. Papadopoulos, N. Montavont, and P. Thubert, "RPL DAG Metric Container (MC) Node State and Attribute (NSA) object type extension," IETF, draft-koutsiamanis-roll-nsa-extension-01, January 2018.

[35] C. Pu, Y. Wang, H. Wang, Y. Yang, and P. Wang, "Multipath Transmission for 6LoWPAN Networks," IETF, draft-pu-6lo-multipath-transmission-02, March 2018.

[36] L. Thomas, P. Akshay, S. Anamalamudi, S. Anand, M. Hegde, and C. Perkins, "Packet Delivery Deadline time in 6LoWPAN Routing Header," IETF, draft-ietf-6lo-deadline-time-01, March 2018.

[37] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, "Hypertext Transfer Protocol – HTTP/1.1," RFC 2616, June 1999.

OUR WORLDWIDE PARTNERS UNIVERSITIES - DOUBLE DEGREE AGREEMENTS

3 CAMPUS, 1 SITE

IMT Atlantique Bretagne–Pays de la Loire – **http://www.imt-atlantique.fr/**

Campus de Brest
Technopôle Brest-Iroise
CS 83818
29238 Brest Cedex 3
France
T +33 (0)2 29 00 11 11
F +33 (0)2 29 00 10 00

Campus de Nantes
4, rue Alfred Kastler
CS 20722
44307 Nantes Cedex 3
France
T +33 (0)2 51 85 81 00
F +33 (0)2 99 12 70 08

Campus de Rennes
2, rue de la Châtaigneraie
CS 17607
35576 Cesson Sévigné Cedex
France
T +33 (0)2 99 12 70 00
F +33 (0)2 51 85 81 99

Site de Toulouse
10, avenue Édouard Belin
BP 44004
31028 Toulouse Cedex 04
France
T +33 (0)5 61 33 83 65

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Imprimé à IMT Atlantique
Dépôt légal : Février 2017
ISSN : 2556-5060