Evolution of the Internet Architecture for LoWPAN

L. Toutain, A. Pelov, N. Montavont, A. Lampropulos
Institut Mines Télécom; Télécom Bretagne; IRISA
Cesson Sévigné 35576 Cedex - France
Email: firstname.lastname@telecom-bretagne.eu

M. Heusse
Grenoble Institute of Technology,
CNRS UMR 5217, Grenoble, France
Email: martin.heusse@imag.fr

1. INTRODUCTION

The Internet is built on the end-to-end connectivity principle where a packet sent by a source node reaches the destination mostly unchanged (to the sole exception of the hop limit in IPv6, in fact). IPv6 reaffirms this approach as it does not allow intermediate routers to modify the packets, and they are even supposed to ignore any option (except the hop-by-hop option, obviously). This approach makes sense in networks for which the data rate follows an ever increasing trend: the idea is to limit the complexity of the treatment at intermediate routers.

In this perspective, LoWPANs[1] are a game changer: those networks target low power consumption and low cost, so that they provide –relatively– low data rates and small frame sizes. This requires some kind of adaptation mechanism at the edge of the LoWPAN. This challenges the consensus approach to networking but it could also be an enabler. In other words, the Internet interconnection model has been successful until nowadays since, even if the network is composed of different kinds of links with various speeds, error rates or delays, they are globally quite homogeneous or at least evolve in parallel. The arrival of new network types and associated constraints such as Wireless Sensor Networks (WSN), Intelligent Transportation Systems (ITS) and even Delay Tolerant Networks (DTN) challenged the interconnection architecture and revealed its limitations.

This paper proposes an evolution of the Internet architecture where the interconnection dogma can be weaker and allows some changes in the way packets are processed and network is managed. We will show some of the limitations of the current design of IPv6 especially regarding extension management and propose an innovative approach where packet formats may transform depending on the nature of the network. We will use LoWPANs as an example and show the possible development of this paradigm.

The proposed approach of evolving the Internet architecture has been implemented in an WSN urban network and has been deployed in a real-world scenario, courtesy of the ANR project ARESA2.

The remainder of the paper is organized as follows. Section 2 presents the currently proposed IETF architecture and protocols for LoWPAN, while in Section 3 we discuss its limitations and the necessary packet and address format modifications. Section 4 presents our new architecture for LoWPANs.

2. IETF ARCHITECTURE

One of the main goals of LoWPANs is to minimize energy consumption and be very cheap to allow massive deployment everywhere in home, industries, cities, etc. This generally leads to the usage of low power radio technologies and as a consequence – small frame size. One of the reasons to introduce IP in such environment is to move from a vertical approach where a separate network is required for every individual application, to a horizontal approach where a single network is able to carry the data for all applications. Furthermore, IP hides Layer 2 characteristics and provides an abstraction, thus making the applications de-correlated from the way information is carried from node to node. However, IP introduces some extra complexity in terms of code footprint and management traffic that could be incompatible with the constrained nature of some of the environments. IPv6 is a good candidate for LoWPAN networks since the large address size can easily handle the vast number of nodes expected to be deployed even in the more optimistic scenarios. In addition, IPv6 possesses address auto-configuration properties that are indispensable in such environments.

Several IETF working groups work on the adaptation of the IPv6 protocol stack to constrained devices:

- IPv6 adaptation layer – 6LoWPAN protocol [7, 5] (Layer 3). It is used to adapt the IPv6 protocol to the characteristics of a constrained Layer 2 technology. Initially, 6LoWPAN was aimed exclusively at adapting IPv6 for IEEE 802.15.4 networks, but recently the scope of its application has been expanded and there are proposals for running 6LoWPAN over multiple types of low-

This work was partially funded by the ANR VERSO ARESA2 project.

[1] Low Power Networks, typically Wireless Sensor Networks, but can include a broader spectrum of wired and wireless technologies providing limited throughput.

power technologies (e.g. Bluetooth, DECT). The main limitation in all aforementioned L2 technologies comes from the necessity to limit the energy consumption (and cost), which leads to high error rates and consequently low frame sizes (e.g. IEEE 802.15.4 limits frame size to 127 bytes). It is therefore impossible to use IPv6 directly over these L2 technologies, as the frame size limitation is incompatible with the specification of IPv6. 6LoWPAN solves this issue by implementing a fragmentation mechanism (and adding a fragmentation header), which splits the (potentially) large IPv6 packets into smaller frames which can be transmitted over the LoWPAN – transparently for the IPv6 layer. However, fragmentation can lead to significant performance penalty and should be limited. In order to avoid fragmentation as much as possible, 6LoWPAN provides IPv6 header compression. The 40 bytes of IPv6's uncompressed header may be compressed down to 4 bytes in the best cases, but generally the compressed header size will be around 20 bytes.

- Specialized routing protocol – RPL [10] (based on Distance Vector). The main characteristic is to optimize communication with a sink, which can be either a router to the Internet or an HTTP/CoAP gateway. RPL defines an abstract metric called Rank to build a Direction-oriented Directed Acyclic Graph (DoDAG). How the rank is computed from the metrics is defined in the "Objective Functions" documents, which allow the specification of various routing policies independently of the protocol itself. Several DoDAG instances may coexist in a single WSN to solve different constraints (minimize energy, minimize delays, . . .). RPL defines a Hop-by-Hop extension to detect routing loops and allows for a node to specify the routing behavior.
- REST compatible architecture, implemented via CoAP – a protocol close to HTTP but much lighter. On top of CoAP, we may find some profiles defined for various types of services, for instance ZigBee Smart Energy 2.0.

The current usage of this adaptation layer still follows the interconnection paradigm, even if in some cases it reaches its limits.

3. LIMIT OF THE IETF MODEL

3.1 Limit of the architecture

From an architectural point of view, the compression done at the 6LoWPAN level is quite different from other compression mechanisms already defined by the IETF such as Van Jacobson PPP compression [6] or even RoHC [2]. The latter schemes were designed for point-to-point links where the packet is uncompressed before being processed by the IP layer. In a 6LoWPAN network, a packet using the route-over mode is forwarded from node to node until it reaches its destination. In some implementations, such as Contiki[2], every node uncompresses the 6LoWPAN packet before processing it and compresses it again before forwarding it. This behavior is sub-optimal since two buffers are needed to process a packet (one for the uncompressed packet and another for the compressed one). In very constrained environments such as LoWPANs, node memory is limited. A direct processing of the compressed header, if possible, would be more efficient, since less memory would be required, resulting in a shorter processing times.

In its current definition 6LoWPAN serves as an intermediate packet format, since it is necessary to go back to the standard IPv6 packet format before processing it (i.e., uncompress the header). The heart of our proposal is to circumvent the IPv6 format altogether and use the 6LoWPAN header instead. In other words, we propose to use 6LoWPAN as the unique header format for LoWPANs.

To understand the concept, a parallel can be established with Layer 2 IEEE 802 architectures. IEEE 802.3 defines a frame format used to transport information on Ethernet networks. But this format is also the common format used to bridge over other technologies. For instance, in Metropolitan Ethernet [1] Provider Backbone Bridges (PBB) add or suppress information in the frame header such as service identifier, MAC addresses of PBB, or classes for Quality of Service.

IEEE 802.11 defines a completely different framing, containing more information than IEEE 802.3 in order to manage the channel access method (CSMA/CA) or to enable infrastructure-mode addressing. When bridging to an Ethernet network, the access points extract the needed information from an 802.11 frame to build an Ethernet frame and vice-versa. Instead of having a single protocol covering all cases, we have different protocols, each one adapted to a particular environment. However, this is totally transparent to hosts systems users which see them as an Ethernet network. Interconnection is eased by the architecture design such as address format and universal identifiers: since the Ethernet frame format contains only the minimal information needed to process a frame, this information is always found in other L2 technologies frames.

Indeed, IPv6 and Ethernet headers look very similar in their nature, as they contain the minimal information for the network to forward the packet to the destination (i.e. source address, destination address and upper layer protocol).

One could argue that IPv6 extensions are designed to carry extra information not found in the IPv6 header. But in fact extensions are global and do not fit the scheme we just described. RFC 2460 [3], specifying IPv6, states: "*With one exception, extension headers are not examined or processed by any node along a packet's delivery path, until the packet reaches the node (or each of the set of nodes, in the case of multicast) identified in the Destination Address field of the IPv6 header*".

For instance, to allow forwarding on a specific DoDAG instance, the IPv6 Hop-by-Hop extension has to be used. Unfortunately, following RFC 2460 rule, this extension cannot be removed when leaving the LoWPAN network. In this case, each and every backbone router has to process this extension, which will in the least degrade their per-

[2] Open source OS and networking stack. The de-facto standard for 6LoWPAN networking.

formance. The RPL specification suggests tunneling these extensions into another IPv6 packet; extensions are inclosed into a tunnel IPv6 header coming from or going to the border router. This way, the 6LoWPAN border router can remove the header with the extensions. Some other possibilities have been discussed during the standardization efforts, such as storing the instance in the Flow Label field, but they have not been proved to be without unwanted consequences (*e.g.* flow label reuse can lead to incompatibilities if the field is used for other purposes by an application).

3.2 Limit of the IPv6 address

Apart from the packet format evolution, the addressing scheme plays a major role in wireless multi-hop network to efficiently locate a destination node. IP networks rely on a hierarchical approach with addresses of a fixed length. An IP address is divided into two parts; the first part is a hierarchical prefix that determines the destination physical network (also called *link*). The second part is a flat identifier, called Interface ID, that is used to identify a device on a given link. In IPv6, both parts are 64 bits long.

IPv6 allows a node to build its own address with the Neighbor Discovery Protocol (NDP), a mechanism referred as auto-configuration [11]. The IPv6 prefix is obtained through Router Advertisement messages sent by the on-link router. The Interface ID is computed locally by the node, e.g., from a random number or the interface MAC address. A node only needs to concatenate the IPv6 prefix and the Interface ID, and checks whether this address is unique on the link by sending a multicast Neighbor Solicitation message. In LoWPAN, the Interface ID must be obtained from the MAC address and the prefix may be learned through NDP. RFC6775 [9] optimizes NDP by drastically reducing the multicast traffic in LoWPAN. Instead of broadcasting requests in the LoWPAN, RFC6775 defines a unicast exchange between a each node and the border router (the router connecting the LoWPAN to the outside world). Thus the border router acts as a data base and helps in performing Duplicate Address Detection.

A LoWPAN may be attached to several IPv6 providers, for instance in the case of a urban network, a LoWPAN may be connected to a city backbone network and a 3G backup network. Following the IPv6 rules, nodes should have one prefix per provider. It is well-known that this configuration does not work as is, since providers implement ingress filtering. If a packet reaches a border router with a source address that belongs to a different provider, the packet will be rejected because the prefix does not match. Many works have been done to solve this problem for the general case, e.g., MIP6, Shim6 [4].

Regarding LoWPAN, this optimized version of NDP still produces a large amount of traffic for node configuration. Multicast traffic is not needed anymore, but we fall back in the same problem that has been highlighted in multi-homed IPv6 networks where nodes are unable to select the appropriate source address. As we will see, our approach proposes to remove the notion of global address from the LoWPAN, thus avoiding the problem of the prefix advertisement, and source address selection.

4. PROPOSED ARCHITECTURE

We propose a new paradigm for the Internet, in this case applied to LoWPAN, where IPv6 is viewed only as the standard interface between heterogeneous networks. In some special networks, such as a LoWPAN, specific protocols are used to answer the particular issues of the networks. While the specific protocols to a network are different from IPv6, they are still designed to be easily converted into IPv6. In our case, the conversion is made by the border router, which interconnects standard IPv6 on one side, and a domain-specific protocol on the other side. For example, the domain-specific constraints in 6LoWPAN include low link capacity, instability, and the limited resources in term of energy and node power / memory.

As described earlier, there exist several IPv6 adaptations for LoWPAN (6LoWPAN, RFC6775, etc.). However, two main problems remain unaddressed: there is still a large amount of traffic generated necessary to configure a unique address, and the 6LoWPAN packets require a lot of processing.

4.1 Architecture

The core of our proposal is to use an *implicit prefix* `(::/64)` for all nodes in a given LoWPAN. This simple *architectural* constraint has important consequences addressing the issues with the classical 6LoWPAN mentioned earlier. Prefixes are actually only useful outside the LoWPAN domain to locate a network, but inside the LoWPA the routing can be done only with the identifier[3]. At the edge, the border router plays the role of a NPTv6 router [12] by changing the implicit prefix (::/64) to the global prefix assigned to the LoWPAN. In this way, we reduce the signaling inside the LoWPAN to the neighbor advertisement message needed for RPL. Note that we still guarantee end-to-end communication between any node on the Internet and the sensors.

The implicit prefixing also addresses the multi-homing case, where a LoWPAN is attached to two or more IPv6 connectivity providers. Indeed, as the prefix is added at the border router, the problem of choosing the source IPv6 address becomes trivial – with extremely low complexity of the solution.

Finally, it is important to note that our proposal is compatible with the standards as specified by the IETF. Most notably, this proposal does not require any change to the 6LoWPAN header compression scheme.

4.2 Examples

4.2.1 Urban topology

Let us analyze the common urban topology scenario shown in Figure 1, where a LoWPAN is connected to two different providers. The first provider is using the prefix $\alpha :: /64$, while the second one is using $\beta :: /64$. For uplink traffic originated from the LoWPAN to the Internet the border router can select the appropriate prefix for the source address (i.e., α or β) to avoid ingress filtering. However, as the prefix of the source address will change, the L4 checksum (typically UDP) will be invalidated as it was computed on the ::/64 prefix. We circumvent this limitation by using NPTv6, as it

[3] Since in route-over mode, a routing table already contains /128 entries, there is no increase of the routing table size

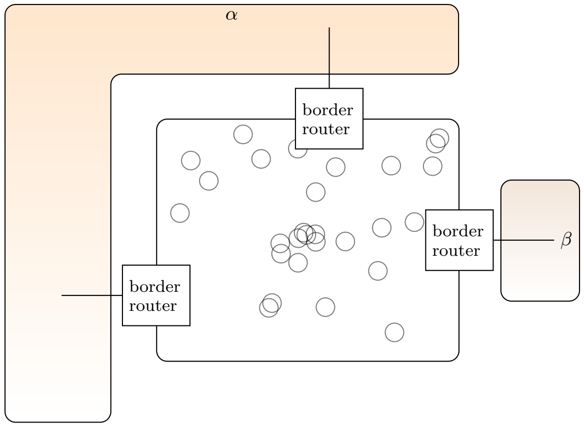

Figure 1: Multi-homed architecture

makes the address translation L4-checksum neutral. It does so by adjusting the Interface ID (IID) value in such a way as to to compensate for the change in the prefix. In theory, this will not change the node identification, since instead of the IID, the invariant value *IID-checksum(prefix)* is used by the destination to find the original sensor ID.

4.2.2 GSE routing

This approach is close to a mechanism proposed at the beginning of the IPv6 standardization called GSE [8] for *G*lobal, *S*ite, *E*nd System, which constitute the various parts of an IPv6 address. G and S represent the prefix and E the identifier. In GSE, the main idea is that the Global part of the source address is not written by the sender, but is inserted by the edge router when the packet leaves the customer's network. On one hand, GSE enables multihoming: the source address always reflects the provider used to send the traffic to the destination. In case of failure, the interior routing protocol changes the default route and the next packets are sent through another provider and the source address will be automatically changed to the new provider prefix. On the other hand, GSE changes the way flows must be identified at Layer 4, since the G and S parts of the address are subject to changes. The main constraint was to only use the E part of the address to identify the flows at Layer 4.

In the early IPv6 addressing schemes, the E part of the address was always derived from the MAC address of the interface. MAC addresses are supposed to be unique, but there is no absolute guarantee that all card manufacturers follow the strict rules prescribed by the IEEE. Therefore, if two different end-systems with the same MAC addresses located at two different sites contact the same server, the server will not be able to differentiate the two connections. Nowadays, this problem would be worse, since the perception of addressing as evolved, and the Interface ID can be manually assigned or randomly drawn. Due to the impact on the Layer 4 architecture, GSE was never adopted, and currently the full address is used to identify flows and global unicity is required.

4.3 Protocol adaptation

The restrictions which rendered GSE inapplicable to the Internet do not stand in a LoWPAN. For example, NDP [9] makes the assumption that the MAC address from which the Interface ID is derived is unique. In our approach, where we do not use a prefix on the LoWPAN, a traffic between a pair of nodes inside the LoWPAN will be identified by the Interface ID. But if the traffic is between a sensor node and a node outside of the LoWPAN, the prefix added by the border router makes the full address unique.

In addition to the addressing scheme proposed, we suggest to change the packet format. We propose to use an independent header format for the LoWPAN that does not require IPv6 compliance, nor need to follow the IPv6 rules. However, we provide a format that is easily convertible into a standard IPv6 header, since a border router will have to process many packets coming in and out of the LoWPAN. We saw previously that in 6LoWPAN, a tunnel need to be set up between a sensor node and the border router to include a hop-by-hop extension and to allow the border router to remove it when the packet leaves the LoWPAN. We simplify this by adding a destination option to the 6LoWPAN header. This destination option can be transported up to nodes outside the LoWPAN if needed, or removed / added by the border router. By allowing this, we avoid using a tunnel between a sensor and the border router, and thus we have a single 6LoWPAN compressed header in the LoWPAN. In the ANR ARESA2 project we keep the destination option up to the destination in the Internet to carry the information allowing the destination to choose routing parameters in LoWPAN for its traffic.

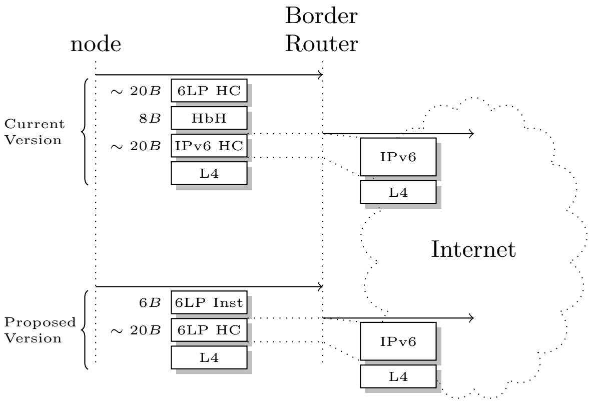

Figure 2: header compression

5. IMPLEMENTATION

In order to evaluate the proposed architecture, we developed and deployed a testbed which includes a full implementation of the architectural and protocol modifications described in Section 4. The testbed consists of extremely limited in terms of processing and memory devices based on the Arduino platform – Arduino UNO. We developed a networking stack capable of running on these devices, called *pico IPv6 stack*, based on Contiki's micro IPv6 stack.

Arduino UNO's flash module size is 32KB and the SRAM size is 2KB. For wireless communications we use XBee modules providing a hardware implementation of the IEEE 802.15.4 protocol stack. These modules can be easily attached to Arduino devices and interact via an open source API. Our *pico*

IPv6 stack was successfully validated for compatibility with regular Contiki nodes. The code is available open source on our Github [4].

Additionally, we ported the complete micro IPv6 stack from Contiki to the Arduino platform [5]. However, a more powerful device is needed in this case (such as the Arduino MEGA). This version of Arduino has a 256KB flash memory and a 8KB SRAM, working at 16 MHz clock speed. Arduino MEGA and UNO devices + XBee modules interact forming a LoWPAN where the former play the intermediate router role and the latter play the leaf role.

When comparing both stacks, we obtain the following values regarding code and data memory segments:

Figure 1 compares implementation of the micro-IPv6 (on Arduino MEGA) and pico-IPv6. The stack size of micro IPv6 renders it unfit to work on Arduino UNO (or comparable). Only pico IPv6 satisfies the constraints.

		OS	Xbee	Stack
micro IPv6	code	4500 B	136 B	31500 B
	data	1460 B	134 B	2 400 B
pico IPv6	code	2590 B	136 B	12710 B
	data	384 B	134 B	790 B

Table 1: Comparison between micro and pico IPv6

6. CONCLUSION

In this paper we presented the current trends to providing a standard protocol stack for sensor networks (also called LoWPAN). At a time where all communication networks adopt IP, the LoWPAN IP architecture is still under development at the IETF. Basically, the main goal is to provide end-to-end connectivity between a sensor node and any other node on the Internet, while addressing the specific problem inherent to a LoWPAN, such as the lossy radio links, the energy constraints and the low node resources in terms of computation and memory.

The IETF Working Group 6LoWPAN has defined several protocols and IPv6 adaptation for LoWPAN. It allows the compression of IPv6 headers inside the LoWPAN to reduce the significant overhead caused by large IPv6 addresses. RFC6775 proposes an adaptation of the Neighbor Discovery protocol for node auto-configuration. It transforms border routers to databases for the LoWPAN, in order to centralize duplication address detection to only a few nodes, instead of flooding the entire network.

We proposed a more radical approach, introducing more flexibility to IPv6 management. We proposed not to use any IP prefix in the LoWPAN to avoid all complexity regarding node configuration. We also avoid tunneling between a sensor node and the border router, by allowing the latter to add / remove a destination option. In the ANR ARESA2 project, we implemented and validated our approach on a popular open-source platform – Arduino. Throughout this paper, we made the case that network evolution will not come from the core network, which remains stable, but from the edge (in this case – the LoWPAN). A monolithic and universal protocol such as IPv6 shows its limits in regards to the large variety of networks specificities and behaviors. Different packet formats can be standardized regarding the type of network, where 6LoWPAN can be consider as one of them. Even if the packet format is not the universal factor allowing the global interconnection any more, the scope of the address remains universal. The IPv6 original packet format remains the reference guaranteeing interoperability between various kind of networks or with the core network. We demonstrated that using this approach, a source may generate all the information needed by the core to forward the packets. The principle being – when moving toward the core, some intermediary nodes may add / remove information.

[4] https://github.com/telecombretagne/Arduino-IPv6Stack
[5] http://code.google.com/p/xbee-arduino/

7. REFERENCES

[1] Ieee draft standard for local and metropolitan area networks– virtual bridged local area networks – amendment 6: Provider backbone bridges (draft amendment to ieee std 802.1q -rev). *IEEE Unapproved Draft Std 802.1ah-D4.2, Mar 2008*, 2008.

[2] C. Bormann, C. Burmeister, M. Degermark, H. Fukushima, H. Hannu, L-E. Jonsson, R. Hakenberg, T. Koren, K. Le, Z. Liu, A. Martensson, A. Miyazaki, K. Svanbro, T. Wiebke, T. Yoshimura, and H. Zheng. RObust Header Compression (ROHC): Framework and four profiles: RTP, UDP, ESP, and uncompressed. RFC 3095 (Proposed Standard), July 2001. Updated by RFCs 3759, 4815.

[3] S. Deering and R. Hinden. Internet Protocol, Version 6 (IPv6) Specification. RFC 2460 (Draft Standard), December 1998. Updated by RFCs 5095, 5722, 5871, 6437.

[4] Amine Dhraief and Nicolas Montavont. Toward mobility and multihoming unification : the SHIM6 protocol : a case study. In *WCNC 2008 : IEEE Wireless Communications and Networking Conference*, pages 2840 – 2845, 2008.

[5] J. Hui and P. Thubert. Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks. RFC 6282 (Proposed Standard), September 2011.

[6] V. Jacobson. Compressing TCP/IP Headers for Low-Speed Serial Links. RFC 1144 (Proposed Standard), February 1990.

[7] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler. Transmission of IPv6 Packets over IEEE 802.15.4 Networks. RFC 4944 (Proposed Standard), September 2007. Updated by RFC 6282.

[8] M. O'Dell. GSE: An Alternate Addressing Architecture for IPv6. Internet-Draft draft-ipng-gseaddr-00.txt, IETF Secretariat, February 1997.

[9] Z. Shelby, S. Chakrabarti, E. Nordmark, and C. Bormann. Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs). RFC 6775 (Proposed Standard), November 2012.

[10] Ed. T. Winter, P. Thubert, A. Brandt Ed., J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, JP. Vasseur,

and R. Alexander. RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. RFC 6550 (Proposed Standard), March 2012.

[11] S. Thomson, T. Narten, and T. Jinmei. IPv6 Stateless Address Autoconfiguration. RFC 4862 (Draft Standard), September 2007.

[12] M. Wasserman and F. Baker. IPv6-to-IPv6 Network Prefix Translation. RFC 6296 (Experimental), June 2011.