Implementation and Evaluation of the Enhanced Header Compression (IPHC) for 6LoWPAN

Alessandro Ludovici, Anna Calveras, Marisa Catalan, Carles Gómez,
and Josep Paradells

Wireless Networks Group (WNG), Universitat Politècnica de Catalunya, C/Jordi Girona,
1-3, Mòdul C3 - Campus Nord. 08034 Barcelona, Spain
anna.calveras@entel.upc.edu

Abstract. 6LoWPAN defines how to carry IPv6 packets over IEEE 802.15.4 low power wireless or sensor networks. Limited bandwidth, memory and energy resources require a careful application of IPv6 in a LoWPAN. The IEEE 802.15.4 standard defines a maximum frame size of 127 bytes that decreases to 102 bytes considering the header overhead. A further reduction is due to the security, network and transport protocols header overhead that, in case of IPv6 and UDP, leave only 33 bytes for application data. A compression algorithm is necessary in order to reduce the overhead and save space in data payload. This paper describes and compares the proposed IPv6 header compression mechanisms for 6LoWPAN environments.

Keywords: 6lowpan, IPv6, header compression, sensor network, IEEE 802.15.4, blip.

1 Introduction

6LoWPAN is defined as a protocol to enable IPv6 packets to be carried on top of Low Power Wireless Personal Area Networks (LoWPANs) [1]. LoWPANs are composed of devices compatible with the IEEE 802.15.4 standard.

The aim is to develop personal networks, mainly sensor based, that can be integrated to the existing well-known network infrastructure by reusing mature and wide-used technologies. IPv6 has been chosen as network protocol because its characteristics fit to the problematic that characterizes LoWPAN environments such as the large number of nodes to address and stateless address auto-configuration.

1.1 IEEE 802.15.4

The IEEE 802.15.4 standard [2] defines protocols and interconnections of devices via radio communication in a Low Rate Wireless Personal Area Network (LR-WPAN). It follows the OSI reference model and specifies the physical and the Medium Access Control (MAC) sublayer of the data link layer. The main characteristics of these LR-WPANs include: (1) data rates of 250 kbps, 100 kbps, 40 kbps and 20 kbps; (2) IEEE 16-bit short or 64-bit extended address; (3) Low power consumption.

M. Oliver and S. Sallent (Eds.): EUNICE 2009, LNCS 5733, pp. 168–177, 2009.

IEEE 802.15.4 devices are classified into Full Function Devices (FFD) and Reduced Function Devices (RFD). The FFD operates as a PAN coordinator and border router. Two important features of 802.15.4 are its self-healing and self-organizing properties. This means that nodes are able to detect the presence of other nodes and organize themselves in a network, and they can detect and recover from faults.

There exist four different frame types: (1) beacon frame, (2) data frame, (3) acknowledgment frame, (4) MAC command frame. The maximum frame size defined in IEEE 802.15.4 is fixed to 127 bytes, of which 25 bytes are reserved for frame overhead. This leaves 102 bytes for payload.

1.2 6LoWPAN Architecture

In order to transport IPv6 packets over 802.15.4 links it is required, as specified in [3], to provide an adaptation layer below the network layer (Fig.1). It is demanded in order to comply with the minimum MTU required by IPv6 that is fixed to 1280 bytes.

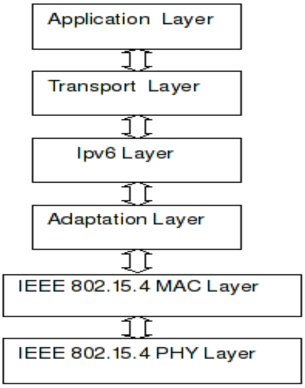

Fig. 1. 6LoWPAN protocol stack

The packet is prefixed by LoWPAN encapsulation headers that, as defined in [3], include the presence of a one byte IPv6 Dispatch header and the definition of the following header fields and their ordering constraints. The two leftmost bits are settled to 01 or 00 indicating if there is a 6LoWPAN frame or not. The remaining 6 bits can define up to 64 different dispatch header types. However, only 5 dispatch header types are defined in [3].

As mentioned before, IPv6 allows stateless address auto-configuration. This property allows hosts to generate their own address combining locally available information together with the one advertised by routers. The host generates the interface identifier

```
+---------------+-------------+---------+
| IPv6 Dispatch | IPv6 Header | Payload |
+---------------+-------------+---------+
```

Fig. 2. LoWPAN encapsulated IPv6 datagram

while the router provides the subnetwork prefix associated with a link. The interface identifier is defined with a length of 64 bits [5]. Thus, there is no problem if the PAN uses 64 bits IEEE 802.15.4 extended addresses but a modification is needed when using 16 bit IEEE 802.15.4 short addresses. The modification consists of adding a 48 bits pseudo address to the 16 bits interface identifier in order to obtain the required length of 64 bits. The pseudo address is formed as follows:

PAN ID (16-bit): zero bits (16): IEEE 16-bit short address

Considering an IEEE 16-bit short address equal to “64” (hex) and PAN ID equal to ”10” (hex) we obtain the following pseudo address:

00:10:00:00:00:64

2 Related Work on IPv6 Header Compression in LoWPAN

IP Header Compression can be defined as “*the process of compressing excess protocol headers before transmitting them on a link and uncompressing them to their original state on reception at the other end of the link*” [4]. Compression is possible since the information carried in the packet is redundant. The redundancy may be present because we are sending packets belonging to the same flow and so the information contained in the headers is repeated several times, or because it is already present in other protocol headers in the packet.

Traditionally, the header compression is performed over a link between two nodes called compressor and decompressor. Moreover, there is the concept of flow context,

```
 0       4       8       12      16      20      24      28   31
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Version| Traffic Class |          Flow Label                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Payload Length        |  Next Header  |   Hop Limit   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                                                               +
|                                                               |
+                         Source Address                        +
|                                                               |
+                                                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                                                               +
|                                                               |
+                      Destination Address                      +
|                                                               |
+                                                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Fig. 3. 40 bytes IPv6 Header

which is a "*collection of information about field values and change patterns of field values in the packet header*" [4]. As just mentioned, IP header compression is usually a hop by hop compression. In a sensor network, this compression approach has high cost in terms of power consumption, indeed at each hop the IP header should be decompressed and re-compressed by the devices. Therefore, this approach might not fit with the constraints of 6LoWPAN networks. In addition to the increased processing operation at each node and the consequent increase of the needed power, there is also the problem of the maintenance of the context due to limited memory in sensor devices.

2.1 LOWPAN_HC1

The first specification of IPv6 header compression for LoWPAN has appeared in [3], and it is specified as LOWPAN_HC1. Considering the IPv6 header as shown in Fig.3, the common case for 6LoWPAN communications can be listed as:

- IP Version: it is 6 for all packets
- Traffic class and flow label: they are zero
- Payload length: it can be inferred from layer 2 or from the "datagram_size" field in the case we have a fragmented packet.
- Next header: it can be UDP, TCP or ICMP, so using 2 bits suffices.
- Source and Destination address: they are link-local (that is, the IPv6 interface identifier can be inferred from source and destination address present in layer 2).

All these fields can be compressed to 1 byte. As mentioned in [3], it is mandatory not to compress the hop limit field, which always needs to be carried inline. So the resulting compressed header would have a size of 2 bytes instead of the 40 bytes of the uncompressed header as seen in Fig. 4.

```
  0   1   2   3   4   5   6   7   8   9   0   1   2   3   4   5
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|  LOWPAN_HC1 dispatch header   |  SA   |  DA   |TF |   NH  |HC2|
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

Fig. 4. 2 bytes encoding LOWPAN_HC1 format

LOWPAN_HC1 is only applied to link-local addresses. In consequence, it would not be possible to compress global addresses. The compression of global addresses would save 32 bytes of link-layer MTU. Moreover, a communication with global addresses would give full capabilities of the IPv6 protocol adoption to a LoWPAN, such as end-to-end communication across different LoWPANs and external IP networks.

To solve this problem, an IETF Internet draft [6], LOWPAN_HC1g, has been published, specifying a method for compressing global addresses. The LOWPAN_HC1g compression came from the fact that "*To support compression of global unicast address, LOWPAN_HC1g assumes that a PAN is assigned on compressible 64-bit global IP prefix. When either the source or destination address matches the* *compressible IP prefix, it can be elided*" [6]. LOWPAN_HC1g does not substitute LOWPAN_HC1, but it extends its applicability.

The compression of global addresses would be useful to gain bytes in the packet to send user data. In order to achieve this, an alternative header compression scheme has been developed under the name of LOWPAN_IPHC [7]. In this paper we focus and implement this one, which is presented in the following section.

2.2 LOWPAN_IPHC

LOWPAN_IPHC [7] is the third proposed IPv6 header compression scheme. Currently, it is at its fourth update referred as LOWPAN_IPHC-04. Hereafter, LOWPAN_IPHC refers to the fourth update. It has been thought as an improvement of LOWPAN_HC1. In particular, it extends the applicability of header compression to support communication to nodes internal and external to LoWPANs (that is global address), multicast communication and both mesh-under and route-over configurations. Global IPv6 address compression is based on shared states within contexts. In contrast with LOWPAN_HC1, in the proposed LOWPAN_IPHC it is not mandatory to carry inline the hop limit field. A mechanism is specified to compress traffic and flow label in case they are not null fields. LOWPAN_IPHC uses five of the rightmost bits of the dispatch type (bits 3 to 7 in Fig. 5) in order to specify compressed fields of IPv6 header that are not related with the address compression. The dispatch header is followed by the LOWPAN_IPHC header that defines how source and destination addresses are compressed. An additional byte is present when communicating with global address; it is called Context Identifier Extension (CID). The four leftmost bits specify the context for source address. The remaining four rightmost bits specify the context used for destination address. Using context based compression, we could compress up to 16 network prefixes and save 60 bits of payload when communicating with external 6LoWPAN networks.

As reported in [7], LOWPAN_IPHC can compress the IPv6 header down to two octets (the dispatch octet and the LOWPAN_IPHC encoding) with link-local communication as seen in Fig. 5. When routing over multiple IP hops, LOWPAN_IPHC can compress the IPv6 header down to 7 octets (2-octets dispatch/LOWPAN_IPHC, 1-octet Hop Limit, 2-octet Source Address, and 2-octet Destination Address).

```
  0   1   2   3   4   5   6   7   8   9   0   1   2   3   4   5
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| 0 | 1 | 1 |  TF   |NH | HLIM  |CID|SAC|  SAM  | M |DAC|  DAM  |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

Fig. 5. LOWPAN_IPHC Encoding

```
  0   1   2   3   4   5   6   7
+---+---+---+---+---+---+---+---+
|      SCI      |      DCI      |
+---+---+---+---+---+---+---+---+
```

Fig. 6. CID octet

3 Implementation of IPv6 Header Compression over IEEE 802.15.4 Networks

3.1 Protocol Stack

Presently there are not LOWPAN_IPHC public implementations to our best knowledge. Hence, we have developed the compression and decompression routine focusing on the integration with b6lowpan protocol stack, which is presented in the next section, and reusing functions already provided in it.

The software component has been developed on TinyOS 2.1, which is an open-source operating system designed for wireless embedded sensor networks. The implementation of 6LoWPAN functionalities have been developed and implemented by the Berkeley Wireless Embedded Systems (WEBS) [8]. It has been released as TinyOS contribution and initially named b6loWPAN. Currently it is at its fourth version and has changed the name to Berkeley IP implementation for low-power networks (*blip*). When we started implementing the header compression, b6loWPAN was at first release so we have kept working on this version. From now on we will refer to it as *blip.*

It uses LOWPAN_HC1 header compression and includes IPv6 neighbor discovery, default route selection, point-to-point routing and network programming support. Standard tools like ping6, tracert6, and nc6 can be used to interact with and troubleshoot a network of 6loWPAN devices. Pc-side code is written using the standard BSD sockets API (or any other kernel-provided networking interface).

The *blip* implementation of header compression has been substituted by our implementation of LOWPAN_IPHC IPv6 Header compression.

3.2 Hardware Platform

The hardware platform used is the Crossbow's TelosB mote. It is an open source, low-power wireless sensor module. TelosB motes have a 16-bit RISC MCU at 8 MHz and 16 registers. The platform offers 10 kB of RAM, 48kB of flash memory and 16 kB of EEPROM. Requiring at least 1.8 V, it draws 1.8 mA in the active mode and 5.1 µA in the sleep mode. The MCU has an internal voltage reference and a temperature sensor. Further sensors available on the platform are a visible light sensor (Hamamatsu S1087), a visible to IR light sensor (Hamamatsu S1087-01) and a combined humidity and temperature sensor (Sensirion SHT11).

3.3 Environment and Measurements

A performance analysis has been done taking into account sensor memory usage, sensor energy consumption, average throughput of packet transmission within the sensor network and average Round-Trip delay time. The network topology (Fig. 7) is composed by three nodes:

1. IPBaseStation: it is the “border router” and acts as a bridge between the serial and radio links; it is the destination node.
2. Relay Node: it acts as a relay node.
3. Sensor Node: it transmits UDP packets to the IPBaseStation; it is the source node.

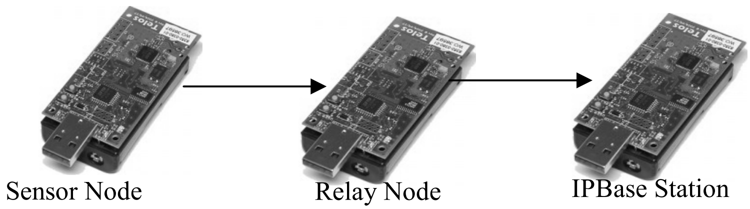

Fig. 7. Network topology

RTT has been measured in a single-hop network topology using the ping6 command included in b6lowpan.

Power consumption analysis has been done at the "relay node" since it is where both, the decompression and compression functionalities were carried out, apart from forwarding (i.e. each time a packet reaches this node it has to decompress, compress and forward the packet). The device used for these measures is the Agilent Technologies DC Power Analyzer N6705A.

All the tests have been done on three different cases of compression: (1) LOWPAN_IPHC, (2) LOWPAN_HC1, (3) No compression.

Performance analysis has been done on communications using global addresses. In the case of LOWPAN_IPHC, the global address has been compressed down to 16 bits.

4 Results

Fig.8 shows the average throughput (in KB/sec) for the three cases listed above. The IP payload ranges from 5 to 70 bytes length. For each payload value, 10 throughput measurements have been done. The final result is the mean value of them. The compressed header reaches a size of 31 bytes for LOWPAN_IPHC, 58 for LOWPAN_HC1 and 62 for the non-compressed headers. The non-compressed header carries all the IPv6 header fields in-line, except the payload field.

In terms of throughput, LOWPAN_IPHC outperforms the others because the bytes of MAC payload used to carry the compressed headers are halved with respect to LOWPAN_HC1. Throughput increases by 39.77% for 70 bytes of payload, which is the maximum payload admitted by LOWPAN_IPHC without the need of fragmentation. Considering the maximum data payload (44 bytes) allowed by LOWPAN_HC1 without packet fragmentation, we obtain a throughput improvement of 25% with LOWPAN_IPHC compression.

The behavior of LOWPAN_HC1 compared with the no compression case needs a brief explanation. Although the UDP header is present in the packet, it is not declared in the next header field of IPv6. Instead of it, an hop-by-hop extension header named source routing header is specified. It is a non-standard header used in *blip* for source routing. In that way we have to carry in-line 8 bits of next header field. This means that, considering the architecture of the stack, the benefit of using one or another compression algorithm depends strongly on how the address fields are compressed more than the other IPv6 fields.

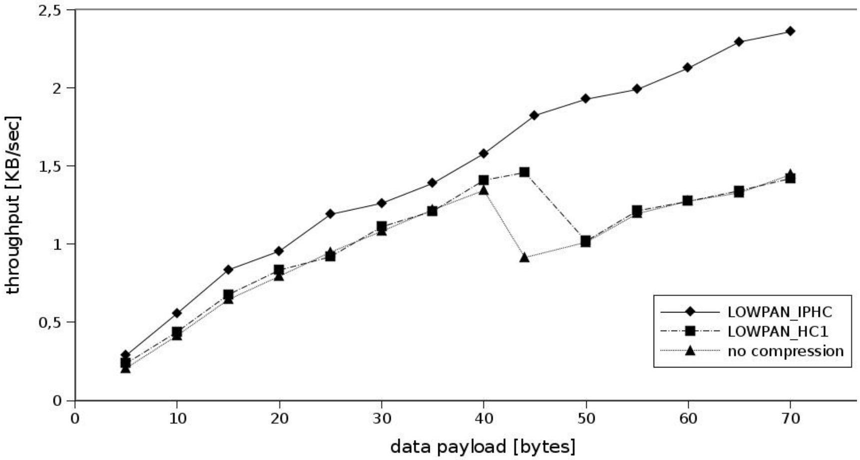

Fig. 8. Throughput obtained for LOWPAN_IPHC, LOWPAN_HC1 and no compression

In terms of energy consumption, we have focused on the effect of compressing IPv6 headers without taking into account the data application payload. Results are shown in Table 1.

Table 1. Energy Consumption

	Consumption (mA)
No compression	19.49
LOWPAN_HC1	19.41
LOWPAN_IPHC	19.27

Table 2. Memory usage

	ROM (bytes)	RAM (bytes)
LOWPAN_HC1	22020	3421
LOWPAN_IPHC	22584	3421

The sample rate has been fixed to 1 ms for a 10 minute test with the Sensor node sending a packet each second and the Base Station replying as soon as the packet arrives. LOWPAN_IPHC shows a better performance also in this case. Battery consumption is lowered 0,72 % between LOWPAN_IPHC and LOWPAN_HC1 and 1,13% between LOWPAN_IPHC and non compression case.

Table 2 compares the memory usage of the basic *blip* installed function that includes header compression LOWPAN_HC1 with the one implemented by LOWPAN_IPHC. LOWPAN_IPHC increases by 564 bytes the occupation of ROM

Table 3. Round Trip Time (RTT) statistics

	Average RTT (ms)	Max RTT (ms)	Min RTT (ms)	Standard deviation (ms)
No compression	171.151	1311.428	87.840	88.397
LOWPAN_HC1	164.560	1192.718	81.323	68.654
LOWPAN_IPHC	79.443	1071.519	63.301	57.741

memory. This reflects the increased complexity of the compression algorithm. Mainly, the use of context based compression makes memory performance worse.

Finally, Table 3 shows the average round-trip delay time obtained from 1000 sent packets. It can be easily appreciated that LOWPAN_IPHC outperforms both no compression and LOWPAN_HC1 cases. These results reflect the throughput performance confirming that the space saved using LOWPAN_IPHC and, in particular, by compressing global addresses steps up the performance in the data transmission. LOWPAN_IPHC decreases RTT by 51.72% respect to LOWPAN_HC1. The average RTT obtained for LOWPAN_IPHC is comparable to others results found in literature [9].

5 Conclusions

In this paper we have presented the header compression mechanisms used to reduce IPv6 headers impact on the performance of 6LoWPAN environments. A first implementation and preliminary results are presented. The obtained results agree with the expected behavior of LOWPAN_IPHC and LOWPAN_HC1.

The main purpose of LOWPAN_IPHC is to offer the performance of a stateful compression in a resource-limited environment such as 6LoWPAN. As we have shown, a stateful compression approach increases the sensor memory usage. However, it outperforms all the other parameters we have taken into account. Moreover, with the refined Traffic and Flow fields compression introduced in LOWPAN_IPHC, the use of mechanisms of congestion control and QoS management on a 6LoWPAN communication would not affect dramatically the overall performance as it could happen with LOWPAN_HC1. This would benefit the application of 6LoWPAN to critical applications (i.e industrial process control, maintenance and surveillance) where there is the need to guarantee the service also in case of network congestion.

Finally, the 6LoWPAN Working Group plans to deprecate LOWPAN_HC1 header compression and push LOWPAN_IPHC [7] forward to become the new header compression standard for 6LoWPAN.

6 Future Work

As future work, the implemented LOWPAN_IPHC compression routine will be adapted to the latest *blip* version. Moreover, it would be useful to study and test possible enhancements of the header compression definition. We plan to compare the benefits of using Context based compression. This will be tested for global addresses when communication happens inside or outside the network.

Acknowledgments. This work has been supported by I2Cat Foundation, FEDER and the Spanish Government through project TEC2006-04504.

References

1. Mulligan, G.: 6LoWPAN Working Group. In: Proceedings of the 4th workshop on Embedded networked sensors, Cork, Ireland, pp. 78–82 (2007) ISBN:978-1-59593-694-3
2. IEEE Std 802.15.4TM -2006 (revision of IEEE Std 802.15.4-2003), IEEE Computer Society (September 2006)
3. Kushalnagar, N., Montenegro, G., Culler, D., Hui, J.: RFC 4944 Transmission of IPv6 Packets over IEEE 802.15.4 Networks. IETF Network Working Group (September 2007)
4. EFFNET AB, An Introduction to IPv6 Header Compression. white paper (February 2004), `http://www.effnet.com/sites/effnet/pdf/uk/Whitepaper_Header_Compression.pdf`
5. Hinden, R., Deering, S.: RFC 4291 - IPv6 Addressing Architecture, IETF Network Working Group (February 2006)
6. Hui, J., Culler, D.: Stateless IPv6 Header Compression for Globally Routable Packets in 6LoWPAN Subnetworks. 6LOWPAN WG draft-hui-6lowpan-hc1g-00 (June 2007)
7. Hui, J., Thubert, P.: Compression Format for IPv6 Datagrams in 6LoWPAN Networks. 6LoWPAN WG draft-ietf-6lowpan-hc-04 (December 2008)
8. Berkeley WEBS, “blip”, `http://smote.cs.berkeley.edu:8000/tracenv/wiki/blip`
9. Hui, J., Culler, D.: IP is Dead, Long Live IP for Wireless Sensor Networks. In: Proceedings of the 6th ACM conference on Embedded network sensor systems. Raleigh, NC, USA, pp. 15–28 (2008) ISBN:978-1-59593-990-6