

Cross-Layer Issues in Constrained Node/Networks

Informal get-together

2011-11-17 1740-1940 (Room 102) @IETF82

What we want to achieve

- Exchange ideas
- Identify potential areas where work is needed
- Identify interested people

Rules of the game

- This is an IETF hallway meeting
 - “note well” applies
- This is not a WG meeting, BOF, of any kind
- Notetakers?
 - <http://grenache.tools.ietf.org:9001/p/notes-ietf-82-cross-layer> = bit.ly/rKlhbj
- Let's just hijack the 6lowpan@jabber.ietf.org

0 – influence of new PHY/MACs

- 802.15.4g, 802.15.4e
- Bluetooth-LE, DECT-ULE
- Adaptation layer issues (fragmentation...)
- Integration of security?

I – 6LoWPAN GHC vs. various users

- ROLL/RPL
- mostly-text applications (e.g., senml+json, XML in CoAP)

Header Compression

- Pretty much essential for IPv6 over CN/N
- Usually focuses on IP header itself
 - This is not ROHC-style complexity...
- Can be used to compress nested headers
- At some point we reach the payload data

6LoWPAN-GHC

- ▶ Generic compression of remaining headers and header-like payloads: ICMPv6, ND, RPL; DHCP; ...
- ▶ draft-bormann-6lowpan-ghc: simple LZ77 based on **bytecode**
 - **single-page** specification: simple
 - **stateless** (but can use 6LoWPAN-HC context)
- ▶ provides modest compression factors between 1.65 and 1.85 on realistic examples
- ▶ fits in 6LoWPAN-HC's NHC

code byte	Action	Argument
0kkkkkkk	Append k = 0b0kkkkkkk bytes of data in the bytecode argument (k < 96)	The k bytes of data
0110iiii	Append all bytes (possibly filling an incomplete byte with zero bits) from Context i	
0111iiii	Append 8 bytes from Context i; i.e., the context value truncated/extended to 8 bytes, and then append 0000 00FF FE00 (i.e., 14 bytes total)	
1000nnnn	Append 0b0000nnnn+2 bytes of zeroes	
1001nnnn	reserved	
101nssss	sa += 0b0ssss000, na += 0b0000n000	
11nnkkkk	n = na+0b00000nnn+2; s = 0b00000kkk+sa+n; append n bytes from previously output bytes, starting s bytes to the left of the current output pointer; set sa = 0, na = 0	

Example: ND Neighbor Solicitation

▶ Payload:

```

87 00 a7 68 00 00 00 00 fe 80 00 00 00 00 00 00
02 1c da ff fe 00 30 23 01 01 3b d3 00 00 00 00
1f 02 00 00 00 00 00 06 00 1c da ff fe 00 20 24

```

Pseudoheader:

```

20 02 0d b8 00 00 00 00 00 00 00 00 ff fe 00 3b d3
fe 80 00 00 00 00 00 00 02 1c da ff fe 00 30 23
00 00 00 30 00 00 00 3a

```

copy: 04 87 00 a7 68

4 nulls: 82

ref(32): fe 80 00 00 00 00 00 00 02 1c da ff fe 00 30 23

-> ref 101nssss 1 2/11nnkkk 6 0: b2 f0

copy: 04 01 01 3b d3

4 nulls: 82

copy: 02 1f 02

5 nulls: 83

copy: 02 06 00

ref(24): 1c da ff fe 00 -> ref 101nssss 0 2/11nnkkk 3 3: a2 db

copy: 02 20 24

Compressed:

```

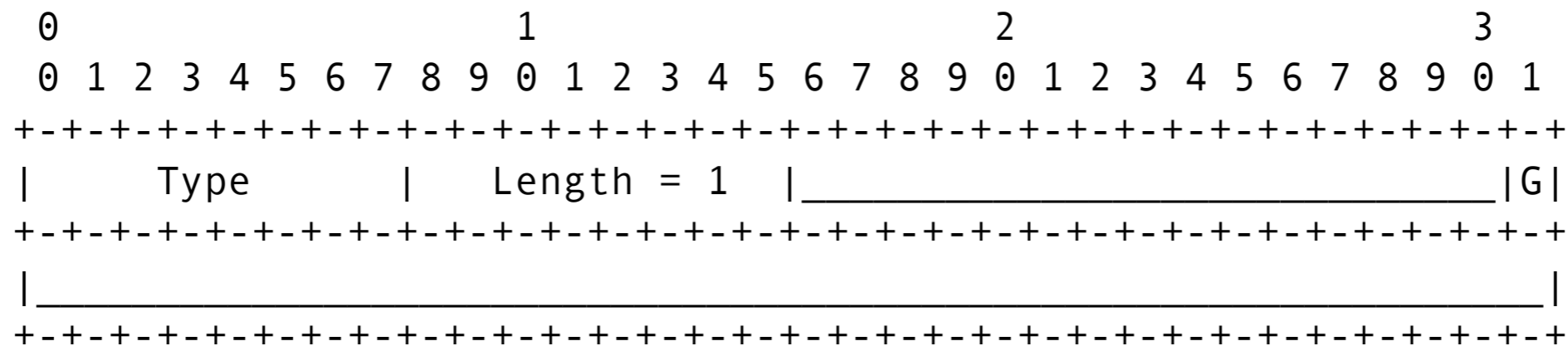
04 87 00 a7 68 82 b2 f0 04 01 01 3b d3 82 02 1f
02 83 02 06 00 a2 db 02 20 24

```

Was 48 bytes; compressed to 26 bytes, compression factor 1.85

Capability Indication (new in -02)

- ▶ How does a node know another node speaks GHC?
- ▶ Add 6LoWPAN Capability Indication (6CIO) option in ND:



- ▶ Typically only needed on initial RS
 - implicit indication takes it from there
- ▶ Option could be used for future other capability indications

Who needs this?

- ▶ ROLL/RPL
- ▶ what other “header-like” protocols?
- ▶ various text-based payloads – JSON, XML, ...?
- ▶ Is GHC needed at all?
The right approach?

Nibblecode

- ▶ GHC optimizes redundancy by repetition (~LZ77)
 - e.g., one senml payload: was 204 bytes; compressed to 146 bytes, compression factor 1.40
- ▶ GHC is not a Huffman coder (does not optimize frequent bytes)
 - Text-based payloads would benefit from that
- ▶ Nibblecode: an attempt to optimize ASCII without full Huffman
 - aeinorst in 4 bits, ASCII in 8, exceptions in 12 bits
 - Appendix B.2 of GHC
- ▶ **needed?**
 - (and is aeinorst the right set of characters? [{ : , " }] maybe?)

2 – cross-layer discovery

- Do we need to do another layer of discovery at every layer?
- cross-layer hinting such as ABRO for CSDS
- anything to learn from HOMENET

3 – LWIG

- What else do we want to document here?
- APIs?

4 – industrial applications

- ISA100.11a works fine
 - but is a very special environment
- is there a need for a middle ground?

5 – management of CN/N

- IETF: was SNMP + MIB only
- now IETF has netconf + yang, ipfix, ...
- could even use CoAP for some of this

- What do we need?
- How do we get it?