

# GHC

Carsten Bormann, IETF89 (London, 2014-03-05)

# Status

- draft-bormann-6lowpan-ghc-00.txt 2010-10-18
  - reviewed by many (→ reductions in complexity)
  - technically completed in 6LoWPAN WG
- Presented at 6lo IETF88 (Vancouver, 2013-11-05)
- draft-ietf-6lo-ghc-00.txt 2013-12-18

# 6CLO option:

- Capability indication for ND
    - 48 bits can be allocated (GHC needs 1)
    - in a pinch, the field can be extended
  - IANA requirements in -00: “RFC required”
  - “IETF review”?
  - Experimentation bits?
- (RFC 5226)

# Editorial:

- expand section 2
- terms and variables
- target buffer vs. destination buffer vs. final result
- static dictionary vs. predefined dictionary
  - explain choices for static dictionary [Hmm]
- References (RFC 2460 section, LZSS)
- one of the examples could expand what the precise actions have to be to get from the initial byte sequence to the result [→ Hmhhh]

**Ship it.**

# 6LoWPAN-GHC

- ▶ Generic compression of remaining headers and header-like payloads
- ▶ draft-ietf-6lo-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