GHC

Carsten Bormann, IETF88 (Vancouver, 2013-11-05)
Header Compression

- Cost(communication) > Cost(computation)
- IEEE 802.15.4 has small PHY packets
  - gets into fragmentation quickly
- IPv6 headers are large headers
- but: ROHC is complex, doesn’t fit well
RFC 6282: 6LoWPAN Header Compression

- Stateless or based on global state only
- Uses RFC 6775 (6LoWPAN-ND) for state setup
- Compresses IPv6 base header and some extension headers, UDP header
- NHC (next header compression) pluggable
Lots of other headers would like compression

- ICMPv6, also as used in ND, RPL, ...
- DHCP; ...
- DTLS

• Add another RFC 6282-like document for each of them?
  • (Possible outcome for DTLS ➔ DICE)
What can be compressed?

- Internal Redundancy
  - (i.e., within one packet only)
- Often cross-layer
- Clumsy coding
  - Internet protocols often optimized for big-CPU convenience (alignment) and speed
  - (and genuine lack of care for coding)
Generic compression

- Fixing clumsy coding needs bespoke code
- Easier to fix: Redundancy
- Huffman: Compress individual symbols
  - Needs probabilities: hard to do generically
- LZ77: Fix repetition
  - Always wins
6LoWPAN-GHC

- Generic compression of remaining headers and header-like payloads
- draft-bormann-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
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 00 06 00 1c da ff fe 00 20 24
```

Pseudoheader:

```
20 02 0d b8 00 00 00 00 00 00 00 ff fe 00 3b d3
fe 80 00 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 2/11nnnk 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 2/11nnnk 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
Implementing GHC

- Code the packets, run compressor afterwards
  - Finds serendipity opportunities
  - Needs more space
- Integrate into packet coder
  - Exploits planned opportunities only (e.g., IP addresses, known fluff)
  - Fits easily in an LWIG class 1 device
What’s in the spec

- Simple LZ77-style bytecode
  - with run-length encoding for zeroes
- plus simple static dictionary (16 bytes)
- Glue for using this as RFC 6282 NHC
- ND option for capability indication
- Lots of examples based on real pcaps
Status

• 2010 proposal, has since been reviewed by many
  • Main revisions: leaving out features
• Was stuck mainly on 6LoWPAN dissolution
  • Used the time to do a bit of research
  • Eerily supported initial haphazard guesses
• Main angle for any further tweaking:
  Static dictionary
Ship it.