Agenda

- LPWANs characteristics
- Header Compression at IETF
- SCHC compression
- SCHC fragmentation
- Applying SCHC
LPWANs characteristics

- See RFC8376
- Very low datarates
  - Down to 100 bits per second
- Small payloads
  - Down to a dozen of bytes
- High transmission cost
  - Battery lifetime, time-on-air
- Star topology, asymmetric links

Header Compression at IETF

- Van Jacobson (RFC1144, Feb 1990)
  - Custom TCP/IPv4 intraflow header compression for slow links
- ROHC (2001-2013)
  - Rule-based, dynamic context with feedback
- 6LoWPAN (RFC6282, Sept 2011)
  - Stateless IPv6 header compression for constrained mesh networks
- SCHC (RFC8724, Apr 2020)
  - Stateless, rule-based, static context compression for extremely constrained links
  - Also specifies fragmentation (3 different modes)
SCHC compression basic principles

Uncompressed header + data

Card 1
Header description

Rule
context
Rule ID
Residue
Card # + compressed header + data

Uncompressed header + data

Card 1
Header description

Protocol parser
SCHC compression

- Match packet header against list of fields in candidate Rule
  - Matching operator
- If match, implement Compression Action
- On Decompressor side, retrieve Rule by ID and rebuild header based on Decompression Action and residue
# UDP/IPv6 compression example

<table>
<thead>
<tr>
<th>FID</th>
<th>FL</th>
<th>FP</th>
<th>DI</th>
<th>TV</th>
<th>MO</th>
<th>CDA</th>
<th>Sent [bits]</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPv6 Version</td>
<td>4</td>
<td>1</td>
<td>Bi</td>
<td>6</td>
<td>ignore</td>
<td>not-sent</td>
<td></td>
</tr>
<tr>
<td>IPv6 Diffserv</td>
<td>8</td>
<td>1</td>
<td>Bi</td>
<td>0</td>
<td>equal</td>
<td>not-sent</td>
<td></td>
</tr>
<tr>
<td>IPv6 Flow Label</td>
<td>20</td>
<td>1</td>
<td>Bi</td>
<td>0</td>
<td>equal</td>
<td>not-sent</td>
<td></td>
</tr>
<tr>
<td>IPv6 Length</td>
<td>16</td>
<td>1</td>
<td>Bi</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IPv6 Next Header</td>
<td>8</td>
<td>1</td>
<td>Bi</td>
<td>17</td>
<td>ignore</td>
<td>compute-*</td>
<td></td>
</tr>
<tr>
<td>IPv6 Hop Limit</td>
<td>8</td>
<td>1</td>
<td>Bi</td>
<td>255</td>
<td>ignore</td>
<td>not-sent</td>
<td></td>
</tr>
<tr>
<td>IPv6 DevPrefix</td>
<td>64</td>
<td>1</td>
<td>Bi</td>
<td>[alpha/64, fe80::/64]</td>
<td>match-</td>
<td>mapping-sent</td>
<td>1</td>
</tr>
<tr>
<td>IPv6 DevIID</td>
<td>64</td>
<td>1</td>
<td>Bi</td>
<td></td>
<td>ignore</td>
<td>DevIID</td>
<td></td>
</tr>
<tr>
<td>IPv6 AppPrefix</td>
<td>64</td>
<td>1</td>
<td>Bi</td>
<td>[beta/64, alpha/64, fe80::/64]</td>
<td>match-</td>
<td>mapping-sent</td>
<td>2</td>
</tr>
<tr>
<td>IPv6 AppIID</td>
<td>64</td>
<td>1</td>
<td>Bi</td>
<td>::1000</td>
<td>equal</td>
<td>not-sent</td>
<td></td>
</tr>
<tr>
<td>UDP DevPort</td>
<td>16</td>
<td>1</td>
<td>Bi</td>
<td>8720</td>
<td>MSB(12)</td>
<td>LSB</td>
<td>4</td>
</tr>
<tr>
<td>UDP AppPort</td>
<td>16</td>
<td>1</td>
<td>Bi</td>
<td>8720</td>
<td>MSB(12)</td>
<td>LSB</td>
<td>4</td>
</tr>
<tr>
<td>UDP Length</td>
<td>16</td>
<td>1</td>
<td>Bi</td>
<td></td>
<td>ignore</td>
<td>compute-*</td>
<td></td>
</tr>
<tr>
<td>UDP checksum</td>
<td>16</td>
<td>1</td>
<td>Bi</td>
<td></td>
<td>ignore</td>
<td>compute-*</td>
<td></td>
</tr>
</tbody>
</table>
### SCHC fragmentation

- **No-ACK**
  - Last fragment contains a Reassembly Check Sequence (RCS)
  - Receiver checks RCS to detect fragment losses

- **ACK-Always**
  - Fragments are grouped into windows
  - Each window is ACK’ed with bitmap of fragments
  - Missing fragments are retransmitted
  - Final window ACK also includes RCS status

- **ACK-on-Error**
  - Packets are divided into tiles and windows
  - Last window is ACK’ed
  - Extra ACKs sent only for windows with missing fragments
  - Supports MTU change during retransmission
Applying SCHC

• SCHC (RFC8724) is a generic mechanism
  – first applied between IP and link layer,
  – can also be used elsewhere in the stack

• Upcoming RFCs will
  – pick options and parameters
    • to fit underlying layers characteristics
  – specify SCHC rules, extensions, architecture
    • to support more upper-layers
    • e.g., OSCORE/CoAP two-layer compression
Thank you!