Static Context Header Compression (SCHC) for the Constrained Application Protocol (CoAP)

draft-ietf-schc-8824-update-03

Marco Tiloca, RISE
Laurent Toutain, IMT Atlantique
Ivan Martinez, Nokia Bell Labs
Ana Minaburo

IETF 121 meeting – Dublin – November 8th, 2024

Recap

- **Document goal: obsolete RFC 8824**

- **Document scope: same as RFC 8824, plus:**
  - Clarified and amended text originally from RFC 8824, also based on filed Errata
  - SCHC compression of CoAP options
    - Clarified both in general and for specific CoAP options
    - Defined for the Hop-Limit Option and for other recent options
    - Updated for the OSCORE Option (extended by some CoRE documents)
  - Clarified SCHC compression for the CoAP payload marker
  - Spelled out SCHC compression in the presence of proxies, with examples provided
  - New IANA registry “SCHC Compression of CoAP Fields” for seamless extensibility
  - YANG data model extending the one defined in RFC 9363

Outline

- **Updates since v -02**
- **Open point: adding a function length for a field of the OSCORE Option**
- **Comments from Quentin – Thanks!**
  - https://mailarchive.ietf.org/arch/msg/schc/080P7Wv5sbjUBezGnl_WitoQCYg/
- **Next steps**

Updates since v -02

- **Various editorial improvements and fixes**
  - Rephrasing and uniform formulations across sections
  - Example rules: consistent TV for the “CoAP Version” field: s/01/1
  - Example rules: added unit (B/b) for variable-length fields, where CDA is “value-sent” or “LSB”

- **Split content about compressing “Token Length” (TKL) and “Token” fields**
  - Now two different Sections 4.3 and 4.6, respectively
  - Now the different Sections 4.x are as many as the CoAP header fields, in the same order

- **SCHC compression of the “Token Length” (TKL) field**
  - Originally admitting only (MO=equal, CDA=not-sent) or (MO=ignore and CDA=value-sent)
  - Now admitting also (MO=match-mapping, CDA=mapping-sent)

Updates since v -02

- **Disambiguated example Rule for eliding the Uri-Path Option (Section 8.1, Table 3)**
- **Marco and Laurent discussed this offline**
  - The 1 in FP already refers to the first occurrence of the Uri-Path Option
  - One might think that the rule requires a match with a Uri-Path Option that has exactly the value "1st element of the path", like in a CoAP request targeting *coap://example.com/1st%20element%20of%20the%20path*

OLD

CoAP Uri-Path	var	1	Dw	1st element of the path	equal	not-sent

- **Conclusion: the old example is ambiguous**
  - Avoid the generic value representation
  - For clarity, use an explicit value in TV, among double quotes like in other examples

NEW

CoAP Uri-Path	var	1	Dw	"status"	equal	not-sent

- **In general, maybe SCHC can benefit of a dedicated, catch-all wildcard to use in TV**

Updates since v -02

- **Found and fixed a mistake in the examples of OSCORE outer compression**

- **Per the example Rules to compress CoAP requests …**
  - The Descriptors for the fields “CoAP OSCORE_piv” and “CoAP OSCORE_kid” specify …
  - FL=var, MO=MSB, CDA=LSB

- **Per Section 7.4.2 of RFC 8724**
  - *If the field is … of variable length, then applying the CDA to compress this field may result in a value ... of variable size (e.g., value-sent or LSB). …, the residue for that field is the bits that result from applying the CDA to the field, preceded with the size of the value.*

- **Three example figures were missing the preceded size of the value**
  - Section 8.3, Figure 15 – Simple example inherited from Section 7.3 of RFC 8824
    - The error was also present in RFC 8824, Section 7.3, Figure 14
  - Section 10.2, Figures 30 and 32 – Example when using a CoAP proxy

Updates since v -02

- **Section 12.1 – Security considerations for the YANG module in Appendix A**
  - Security considerations are due, per https://wiki.ietf.org/group/ops/yang-security-guidelines

- **The YANG data model in Appendix A is just extending the one from RFC 9363**
  - Just more elements corresponding to additional/updated CoAP options
  - It looks like nothing more or special needs to be added compared to RFC 9363

- **Current content of Section 12.1**
  - *The YANG data model defined in Appendix A extends the ietf-schc module defined in [RFC9363]. Therefore, all the security considerations compiled in Section 8 of [RFC9363] apply to the resulting, extended YANG data model as well.*

Good enough?
Anything more to add?

Open point

- **The length of the Token field is computed by the function length ‘tkl’**
  - No confusion with the Token Length (TKL) field
  - Avoided SCHC-encoded size of the compressed Token field on the wire
- **The same rationale has been used for the new fields of the OSCORE Option**
  - Function length ‘osc.x.m’ and ‘osc.y.w’ to compute the size of the subfields “CoAP OSCORE_nonce” and “CoAP OSCORE_oldnonce”, respectively
- **The same rationale is <u>not</u> used for the “CoAP OSCORE_piv” field of the OSCORE Option**
- **A new function length ‘osc.piv’ can be easily defined**
  - It returns the value encoded by the 3 rightmost bits of the first flag-byte in the OSCORE Option
  - Avoid confusion with those 3 bits; save 4 bits of SCHC-encoded size on the wire
- **Planned addition for the next version of the draft**

Comments from Quentin (1/4)

- **(Minor) Concerns on the "tkl" Function [1]**

  The draft specifies that the CoAP Token Field MUST NOT be sent as variable-size to avoid ambiguity with the Token Length field. I would appreciate clarification on this issue. While I understand that a "compute-" function for the Token field length optimizes the size of the residue, the rationale behind this restriction is unclear.*

- **The rationale consists exactly of the two points above**
  - (A) Avoid ambiguity between the “Token Length” (TKL) field and the size the “Token” field
  - (B) Avoid to send on the wire the size of the compressed “Token” field

- **(A) is mentioned both in RFC 8824 and in this document; (B) is not mentioned in either**

- **Proposal: state (B) explicitly in Section 4.6 “CoAP Token Field”**

[1] https://mailarchive.ietf.org/arch/msg/schc/080P7Wv5sbjUBezGnl_WitoQCYg/

Comments from Quentin (2/4)

- **Handling CoAP Options [1]**

The new draft requires interpreting CoAP "raw" option fields and applying specific logic based on Option Types.

Note: This construct was defined in RFC 8824 and is not changed in this document

This raises a concern about future compatibility: if new CoAP Options are introduced, we may face several challenges:

1. Document Updates: We would need to revise the draft to accommodate new options.

- **We thought of that and defined the new IANA registry "SCHC Compression of CoAP Fields"**
  - Goal: collect a simple list of defined ways to do SCHC compression of CoAP fields
  - Rationale: enable a strong decoupling of two orthogonal and independent tasks:
    - Definition or amendments of CoAP header fields, especially of CoAP options
    - Definition or amendments of SCHC Compression for those fields

- **No need to further update this document because of new/updated CoAP options**

 [1] https://mailarchive.ietf.org/arch/msg/schc/080P7Wv5sbjUBezGnl_WitoQCYg/

Comments from Quentin (3/4)

› **Handling CoAP Options [1]**

2. Data Model Updates: Communication between SCHC endpoints requires both to use the same model, necessitating versioning to track capabilities (e.g., model v1.X). This could complicate interactions, especially in diverse environments.

› **Yes. In general, two SCHC endpoints have to be on the same page to work correctly**

– The new IANA registry “SCHC Compression of CoAP Fields” helps out w.r.t. CoAP fields

3. Software/Firmware Updates: If endpoints are operated by different parties, discovering supported definitions may require additional control traffic.

› **Yes, but it can happen as part of the Rule distribution/establishment process**
(more in the next slide)

 [1] https://mailarchive.ietf.org/arch/msg/schc/080P7Wv5sbjUBezGnl_WitoQCYg/

Comments from Quentin (4/4)

- **Alternative Proposal [1]**

Last year, we proposed an alternative approach for handling CoAP Options by specifying the compression using CoAP.OptionDelta, CoAP.OptionLength, and CoAP.OptionValue fields (https://mailarchive.ietf.org/arch/msg/schc/N8lNqnJqwLRQecOxZ8sYoM8GUuq/) While this method has a drawback of larger compression rule sizes (three field descriptors per option), it may mitigate the complexities associated with updating data models and implementations when new CoAP Options are introduced.

Quentin

- **Pros**
  - Straightforward workflow for supporting (new) protocols in SCHC and (new) CoAP options
  - Consistent with how SCHC treats other protocols, i.e., relation between fields and Field Descriptors
  - More future-proof: new options don't need to update the YANG data model or SCHC parsers
- **Cons**
  - Small penalty in residue size
  - Rules have a longer list of Field Descriptors

Authors

- **Comments (Ana)**
  - It's implementation specific. You can do it, while missing abstraction. We don't need 3 FIDs per field.
  - Compression has to be aware of what exists and is standardized. It has to be flexible to extensions/updates.
  - Rules may have to be updated anyway, for other reasons.
- **Cons (Laurent)**
  - Not an appropriate format to exchange Rules using the data model. 3 Field Descriptors per field are not needed.
- **Cons (Marco)**
  - Option values may need (FL=var, CDA=ignore, CDA=value-sent)
    - CoAP.OptionLenght is sent as-is (always 4 bits)
    - CoAP.OptionLengthExtended has variable length and is sent as-is (prepended by 4-bit size, or using new function length)

Next steps

- **Propose new function length ‘osc.piv’**
  - To better compress the Partial IV subfield of the OSCORE Option
- **There can be a change to the OSCORE Option and its compression**
  - Due to a possible change in *draft-ietf-core-oscore-key-update* extending the option
  - That would mean removing “CoAP OSCORE_y” and “CoAP OSCORE_oldnonce”
- **Synch with content of other documents**
  - E.g., *draft-ietf-schc-architecture* and *draft-ietf-core-oscore-capable-proxies*
- **Administrivia**
  - Sources are now hosted at https://github.com/marco-tiloca-sics/draft-ietf-schc-8824-update
  - Can we instead have a repo in the LPWAN Github organization? https://github.com/lp-wan

Thank you!

Comments/questions?

https://github.com/marco-tiloca-sics/draft-ietf-schc-8824-update