

# draft-ietf-lpwan-ipv6-static-context-hc-21

Authors:

Laurent Toutain <Laurent.Toutain@imt-atlantique.fr>

Carles Gomez <carlesgo@entel.upc.edu>

Ana Minaburo <ana@ackl.io>

Dominique Barthel <dominique.barthel@orange.com>

Juan Carlos Zuniga <JuanCarlos.Zuniga@sigfox.com>

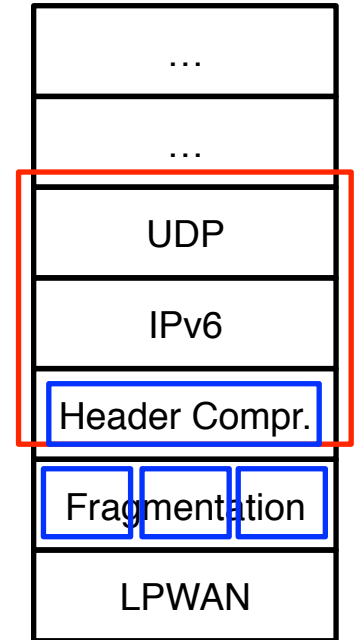
# Presentation agenda

- What is this draft about?
- What has happened since IETF104?
- IETF105-Hackathon report
- What is coming up next?
- Changes since -18

# What is this draft about?

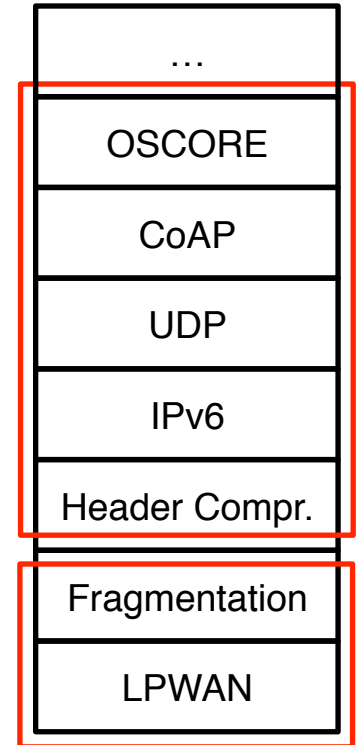
# 3 deliverables in one draft

- Spec. of a Header Compression engine (**Section 7**)
  - Generic engine, uses Static Context (→ SCHC)
- Specification of a fragmentation protocol (**Section 8**)
  - Has 3 different “modes” described in this draft
  - The different modes address different requirements
- Spec. of simple UDP/IPv6 compression (**Section 10**)
  - Using this SCHC engine



# Other related drafts

- Canonical representation of context
- Apply SCHC compression to upper-layer protocols
  - For exemple, CoAP/UDP/IPv6
- Apply SCHC fragmentation to underlying networks
  - For exemple, Sigfox, LoRaWAN®



# What has happened since IETF104?

# What has happened since IETF104?

- *IoT Dir preliminary review* by Carsten, March 4<sup>th</sup>
  - Discussed potential changes with Carsten at IETF104
- Fixed Ack-on-Error description following implementors comments
- Received *AD Review* by Suresh on July 2<sup>nd</sup>
- Implemented changes in response to these reviews
- Published -19 on July 4<sup>th</sup> , entered IETF Last Call July 5<sup>th</sup>
- Completed *IETF Last Call* July 19<sup>th</sup>, no comment
- Published -20 (July 22<sup>nd</sup>), -21 (July 23<sup>rd</sup>)
- Currently in “*Waiting for (AD) write-up*” status

# IETF I05-Hackathon report



# SCHC Hackathon at IETF105



10 team members:

- Ana (first-time hackathoner)
- Cédric
- Diego (remote from Chile)
- Dominique
- Ivaylo
- Laurent
- Matthieu
- Pascal
- Sergio (remote from Spain)
- Shoichi (remote from Japan)



<https://github.com/openschc>

# Work done at IETF I05-Hackathon

- Made OpenSCHC ***easier to use***
  - Cleaned up Sphinx documentation
  - Added tutorial, for easier start
- Improved ***functionalities***
  - Merged several development branches
    - More fragmentation functionalities
    - Compression and Fragmentation now integrated
  - Added some OAM functionalities
  - Wrote plan for Fragmentation random testing

# What is coming up next?

# What is coming up next?

- Work our way to RFC
  - IESG Evaluation
- Communicate/educate about SCHC
- Carry on the OpenSCHC development
- Evaluate performance, applicability

# Current uses of SCHC

- Drafts
  - OSCORE/CoAP/UDP/IPv6 compression with SCHC,
  - SCHC over Sigfox, SCHC over LoRaWAN<sup>®</sup>, SCHC over NB-IoT
- Demos
  - CoAP/UDP/IPv6 compression over LoRaWAN<sup>®</sup>
  - CoAP compression over (IP-enabled) LTE-m
  - SSH over LoRaWAN<sup>®</sup>
- Under evaluation
  - DLMS/UDP/IPv6/LoRaWAN, at LoRa Alliance<sup>®</sup>
  - IPSec ESP compression
- Implementations
  - OpenSCHC, Acklio, Universidad de Chile, RIOT (tbc, expressed intention)
- Scientific papers

# Changes since -18

# Functional

- Specifies behavior of Field Position = 0
- Subsumes *compute-checksum* and *compute-length* into *compute-\**
- Improves specification of compression of
  - IPv6 Traffic Class field (ECN bits)
  - Downlink IPv6 Hop Limit field
  - UDP Length field
- Specifies that UDP checksum field must be verified before elision
- Fixes mistake in description of Ack-on-Error algorithm

# Security Considerations

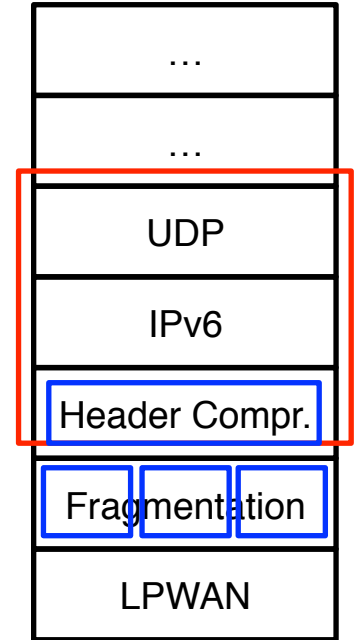
- Major rewrite, more focused
- Expansion of frames with Decompression
- New protocol introduced with Fragmentation
  - Opportunity for loops
- UDP checksum elision



# Editorial

- Reshuffling of Introduction
- Improved description of Field Position
- Added definition of scope for RuleID
- Improved description of compression algorithm
- Replaces MIC with Reassembly Check Sequence
- Fixes compression example in Appendix A
- Fixed a few RFC 2119 language abuses

Thank you for your attention



# (( LPWAN ))

