

June 9th, 2026

VOICI

A minimal Multiplexer for SCHC

Presenters: Q. Lampin

Interim meeting SCHC #9



Agenda

- Problem Statement
- Requirements
- Gap Analysis
- VOICI in a Nutshell
- Integration in the SCHC framework
- References & WG feedback

TL;DR

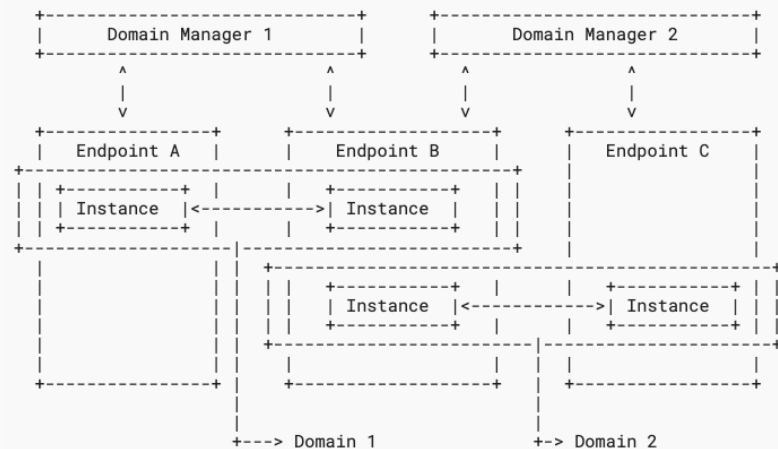
- A simple, yet comprehensive Architecture.
 - Fits the LPWAN use-cases
 - Extends to SCHC use-cases

Problem Statement

Datagrams are routed to the appropriate Instance by the Dispatcher using the Discriminator and admission rules based on information provided in the Instance Configuration. The Dispatcher is a single point of decision for packet forwarding within the Endpoint.

In some deployments, the Discriminator is derived entirely from lower-layer context (e.g., a specific PPP link, an IPv6 address, or a UDP port). If external context is insufficient or unavailable, the Dispatcher may need an explicit Discriminator.

For example, Datagrams can be encapsulated in a light transport protocol whose header contains a Session, Context, or Instance identifier, and can provide additional services such as integrity checking (CRC).



from draft-ietf-schc-architecture

Requirements (driven by SCHC)

Session identification: A mechanism to **distinguish Sessions** and route Datagrams to the correct processing handler (for example, a SCHC Instance). The identifier (Session ID) is locally significant to the link.

Layer independence: The encapsulation **MUST** operate over any link layer that carries compressed traffic, whether identified by an EtherType, IP Protocol Number, or UDP port (draft-ietf-schc-protocol-numbers-06). □ **SCHC over Foo***

Original EtherType/port recovery (optional): A mechanism to **carry the original EtherType, UDP port number or IP Next Header** when the carrier uses the SCHC EtherType or SCHC UDP port. This is needed when the payload is decompressed so that the receiver can restore the original framing layer after decompression.

Integrity protection (optional): A mechanism to **detect the corruption of the Datagram**, including the Session ID and the compressed residue.

Requirements (extended)

Content identification: A mechanism to [identify how the Datagram payload is encoded when the link carries Datagrams from multiple mechanisms](#) (for example, SCHC, uncompressed). This allows the receiver to dispatch the Datagram to the correct decompressor without inspecting its content.

Possible uses:

- Uncompressed traffic (alternative to no-compression rule)
- Evolutions of SCHC
- Concurrent Data Models.
- GHC, RoHC, third-party compression mechanisms

Gap Analysis

Mechanism	Multiplexing	Integrity	Overhead	Link Coverage
Ethertype	No	No	0 bytes	IEEE 802 only
MPLS	Yes	No	4+ bytes	Ethernet, IP
IP Protocol Num	No	No	0 bytes	IP only
UDP Protocol Num	No	Yes	8 bytes	over UDP only
UDP Port	Yes	Yes	8 bytes	over UDP only
QUIC	Yes	Yes	32+ bytes	over QUIC only

Table 1: Comparison of multiplexing mechanisms

8

VOICI, In a Nutshell

Content Identifiers

CI	Content Mechanism
0	Unprocessed / raw -- Datagram requiring no reconstruction; used for minimalistic multiplexing only
1	SCHC -- standard SCHC compressed residue
2-31	Reserved for future mechanisms

Table 2: Initial CI assignments

VOICI, In a Nutshell

Minimal Header

2 bytes

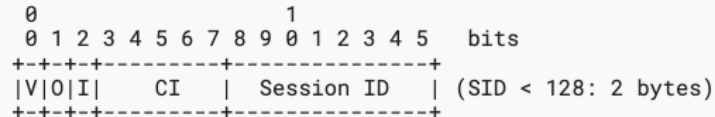
V (1 bit): :=0

O (1 bit): :=0

I (1 bit): :=0

CI (5 bits): 00001 (SCHC)

Session ID (8 bits): < 128



VOICI, In a Nutshell

Header configurations and sizes

V (1 bit): Version, 0 for current.

O (1 bit): Original EtherType/Port carried if set.

I (1 bit): Integrity flag. CRC-16 field if set.

CI (5 bits): Content Identifier for the datagram

Session ID (variable length, LEB128): Identifies the logical session that owns this Datagram.

CRC (16 bits, optional): CRC-16/CCITT-FALSE over the flag byte (V-O-I-CI), the Session ID, the Original EtherType/Port field (if O=1), and the entire Datagram payload.

Original EtherType/Port (1-2 bytes, optional): Carries the EtherType, IP Next Header, UDP port number that was replaced by VOICI identifier.

Configuration	V	O	I	SID < 128	SID >= 128
Session ID only	0	0	0	2 B	3 B
+ CRC	0	0	1	4 B	5 B
+ Orig. EtherType/UDP Port	0	1	0	4 B	5 B
+ Orig. IP Next Header	0	1	0	3 B	4 B
All fields	0	1	1	5-6 B	6-7 B

Table 3: VOICI header size summary

Integration in the SCHC Framework

draft-ietf-schc-protocol-numbers requests an [Internet Protocol Number](#), an [Ethertype](#) assignment, a [CCSDS \(Consultative Committee for Space Data Systems\) Encapsulation Number](#), and [well-known ports](#) for SCHC.

If multiplexing is required, use VOICI for multiplexing within SCHC.

References & Working Group Feedback

Document available at

<https://datatracker.ietf.org/doc/draft-lampin-voici/>

First impressions? Did I miss something or should we do differently?

Thank you for you attention