OVERVIEW OF VSOMIP & COMMONIP

Samira Meziany & Soukaina El Frikh

August 2024

1 Introduction

SOME/IP is an abbreviation for "Scalable service-Oriented middlewarE over IP". This middleware was designed for typical automotive use cases and for being compatible with AUTOSAR (at least on the wire-format level).

2 SOME/IP Specification

2.1 On-Wire Format

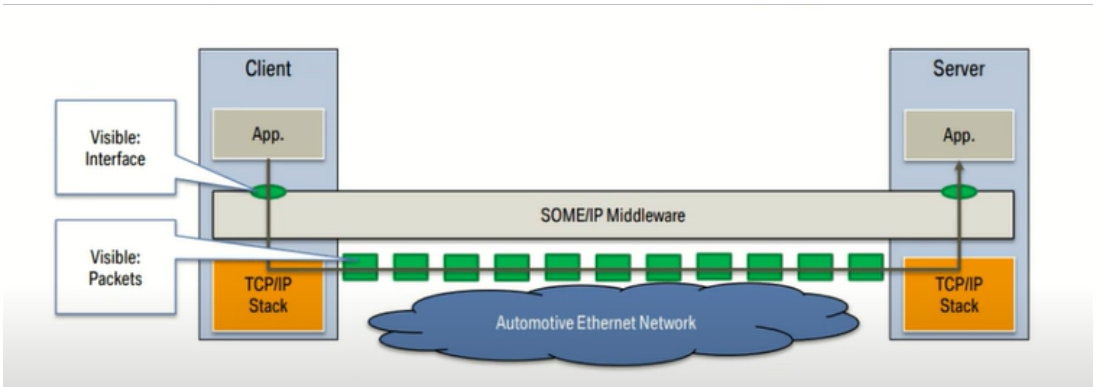

Figure 1: SOMEIP

The on-wire format of SOME/IP includes the following components:

- **Service ID**: Unique identifier for each service.
- **Method ID**: 0-32767 for methods, 32768-65535 for events.
- **Length**: Length of payload in bytes (includes 8 additional bytes for other IDs).
- **Client ID**: Unique identifier for the calling client inside the ECU; must be unique in the overall vehicle.
- **Session ID**: Identifier for session handling; must be incremented for each call.
- **Protocol Version**: 0x01.
- **Interface Version**: Major version of the service interface.

- **Message Type**:
  - REQUEST (0x00): A request expecting a response (even void).
  - REQUEST_NO_RETURN (0x01): A fire-and-forget request.
  - NOTIFICATION (0x02): A request of a notification/event callback expecting no response.
  - RESPONSE (0x80): The response message.
  - REQUEST_ACK (0x40)
  - NOTIFICATION_ACK (0x42)
  - ERROR (0x81)
  - RESPONSE_ACK (0xC0)
  - ERROR_ACK (0xC1)
  - UNKNOWN (0xFF)
- **Return Code**:
  - E_OK (0x00): No error occurred.
  - E_NOT_OK (0x01): An unspecified error occurred.
  - E_WRONG_INTERFACE_VERSION (0x08): Interface version mismatch.
  - E_MALFORMED_MESSAGE (0x09): Deserialization error, payload cannot be deserialized.
  - E_WRONG_MESSAGE_TYPE (0x0A): Unexpected message type received.
  - E_UNKNOWN_SERVICE (0x02)
  - E_UNKNOWN_METHOD (0x03)
  - E_NOT_READY (0x04)
  - E_NOT_REACHABLE (0x05)
  - E_TIMEOUT (0x06)
  - E_WRONG_PROTOCOL_VERSION (0x07)
  - E_UNKNOWN (0xFF)

SOME/IP messages consist of two parts: header and payload.

2.2 Protocol

The SOME/IP protocol includes:

- Transport bindings: UDP and TCP.
- Basic communication patterns: publish/subscribe and request/response.

2.3 Service Discovery

The SOME/IP Service Discovery mechanism is used to locate service instances, detect if service instances are running, and implement the Publish/Subscribe handling.

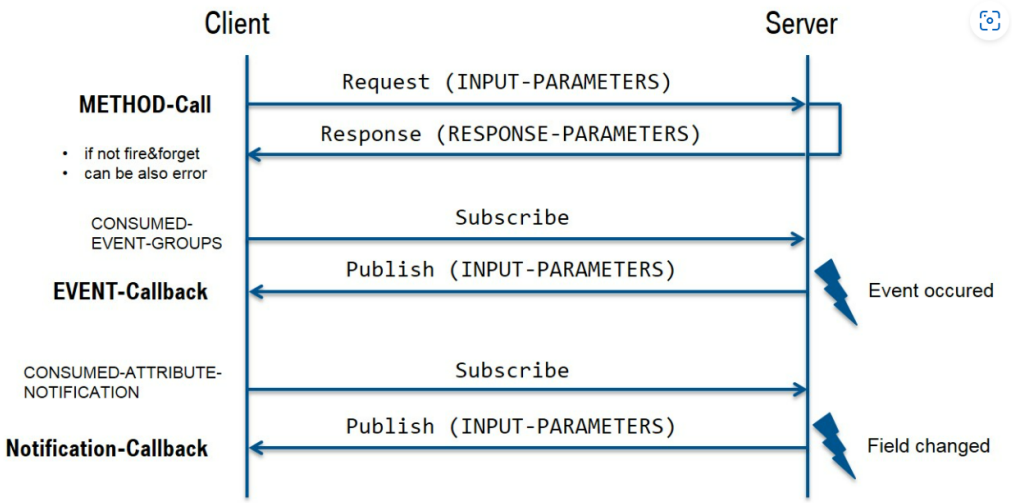

Figure 2: Basic SOME/IP communication pattern

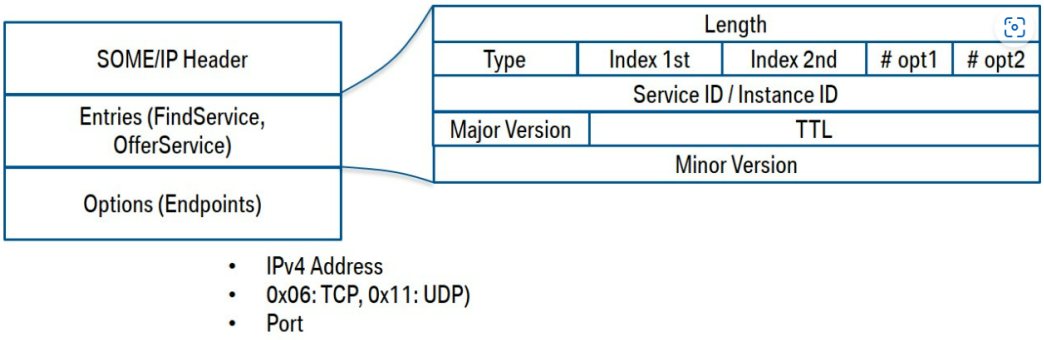

Figure 3: Structure of a SOME/IP SD message

3 vsomeip

The vsomeip is the basic structure of the COVESA implementation of SOME/IP.

3.1 Communication Framework

The provided image illustrates the communication framework of vsomeip, highlighting both external and internal communication mechanisms. Vsomeip facilitates communication between devices via defined communication endpoints that specify the transport protocol (TCP or UDP) and associated parameters, which are configurable in a vsomeip configuration file (JSON format, as detailed in the vsomeip user guide). Internally, interprocess communication is managed through local endpoints implemented using Unix domain sockets via the Boost.Asio library. This internal communication bypasses a central component (such as a D-Bus daemon), ensuring high-speed performance.

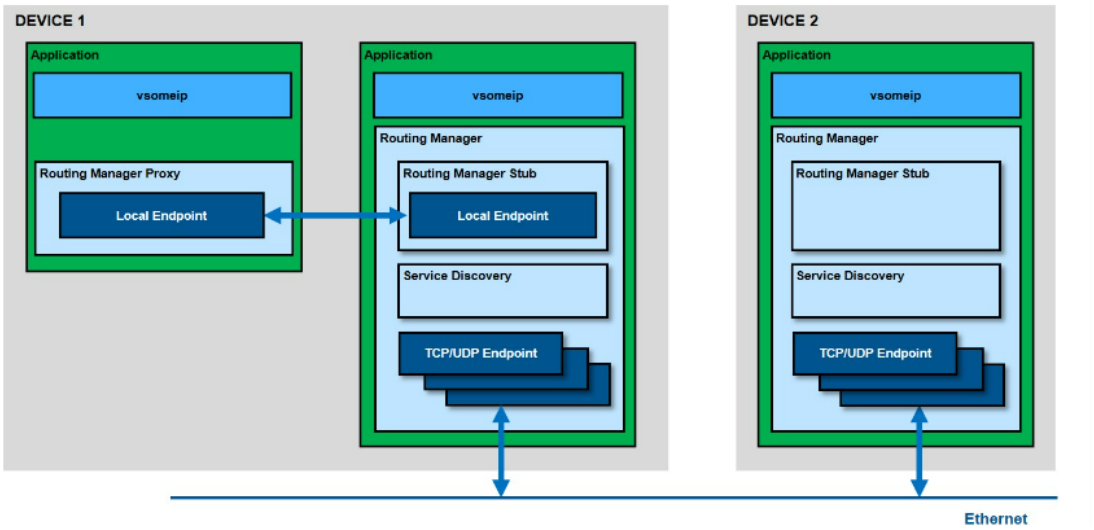

Figure 4: VSOMIP Communication

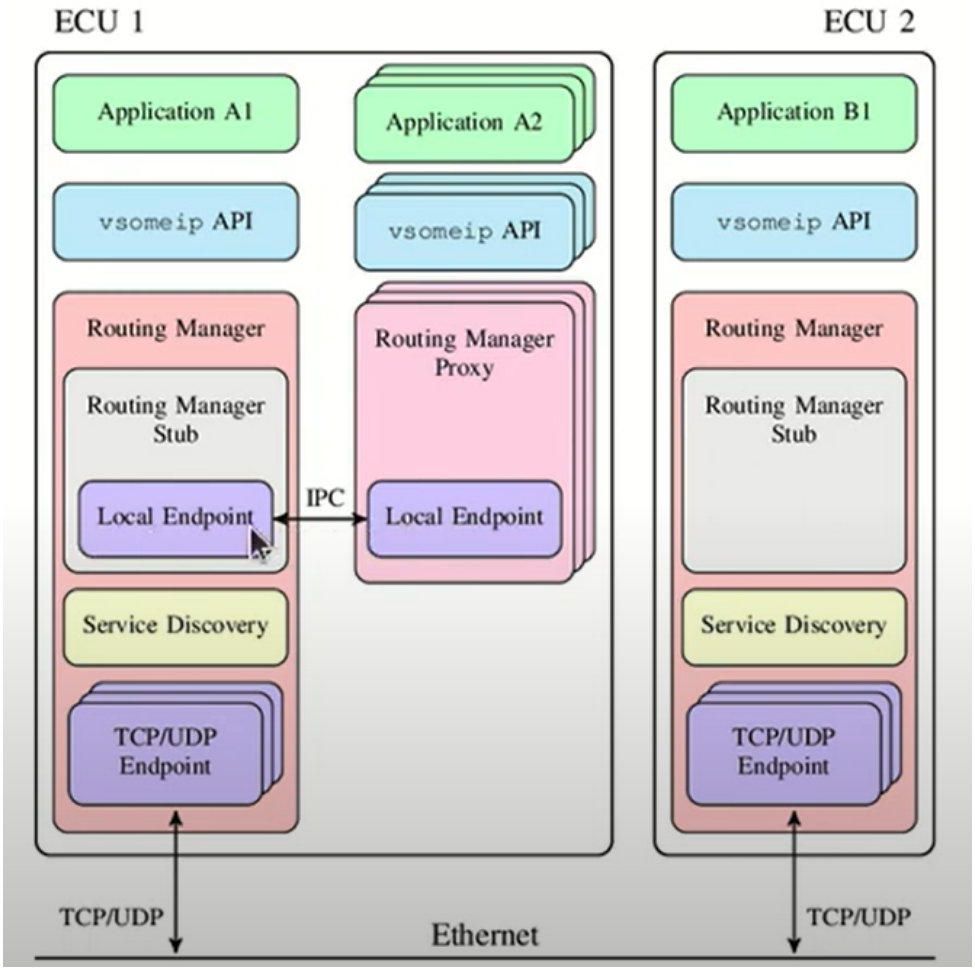

Figure 5: Communication framework of vsomeip

3.2 Important Note

Vsomeip does not implement the serialization of data structures! This is covered by the SOME/IP binding of CommonAPI. Vsomeip just covers the SOME/IP protocol and the Service Discovery.

4 Conclusion

This brief overview provides a foundational understanding of SOME/IP and vsomeip. Further details and examples are available in `https://github.com/COVESA/vsomeip/wiki/vsomeip-in-10-minutes`