SOME/IP Overview

Scalable service-Oriented Middleware over IP (SOME/IP) is an automotive middleware protocol developed by the AUTOSAR consortium.
It is mainly used for communication between different electronic control units (ECUs) in vehicles over Ethernet.
SOME/IP supports service discovery, event notification, and remote procedure calls (RPC), making it suitable for service-oriented architectures.

1. SOME/IP Packet Structure

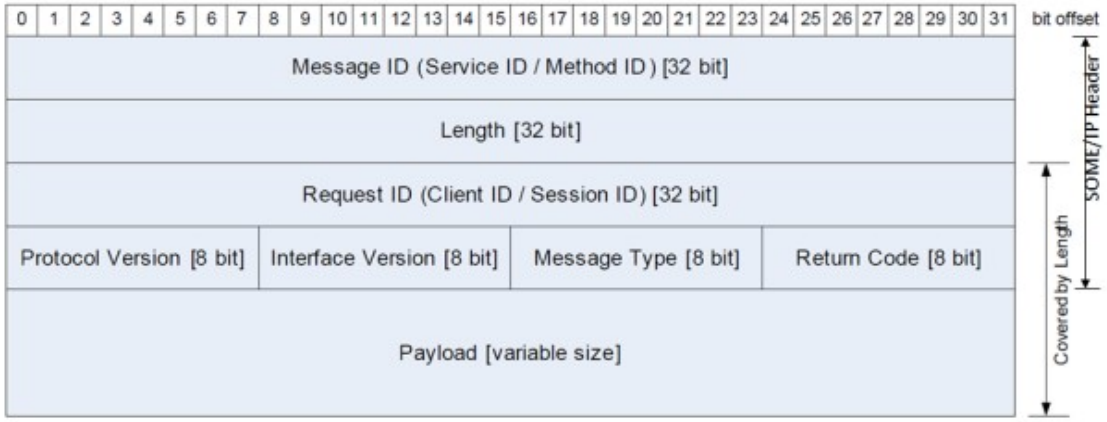

Figure 4.1: SOME/IP Header Format

A SOME/IP message is composed of a header and payload. The header includes metadata used to identify services and methods.

Main SOME/IP header fields:
- **Message ID (32 bits):** Identifies the service and method/event.
- **Length (32 bits):** Indicates the length of the message (excluding the first 8 bytes).

- **Request ID (32 bits):** Combines Client ID and Session ID to match requests and responses.
- **Protocol Version (8 bits):** Version of the SOME/IP protocol (usually 0x01).
- **Interface Version (8 bits):** Version of the service interface.
- **Message Type (8 bits):** Defines message kind (Request, Response, Notification,

etc.).
- **Return Code (8 bits):** Indicates the result status (OK, Not OK, etc.).

2. SOME/IP Methods and Events

- **Methods:** Represent callable operations of a service (like functions). A client sends a request, and the server responds.
- **Events:** Represent asynchronous notifications from a service to subscribed clients.

-**Fields:** A field represents a status and has a valid value. The consumers subscribing for the field instantly after subscription get the field value as an initial event.

field shall be a combination of getter, setter and notification event.

-**Notification Events:**

Notifications describe a general Publish/Subscribe-Concept. Usually the server pub-lishes a service to which a client subscribes. On certain cases the server will send the

client an event, which could be for example an updated value or an event that occurred.

SOME/IP is used only for transporting the updated value and not for the publishing and

subscription mechanisms. These mechanisms are implemented by SOME/IP-SD.

Error Handling: Error handling can be done in the application or the communication layer below. There-

fore SOME/IP supports two different mechanisms:

• Return Codes in the Response Messages of methods

• Explicit Error Messages

Which one of both is used, depends on the configuration.

Strategy for sending notifications:

For different use cases different strategies for sending notifications are possible. The

following examples are common:

• Cyclic update — send an updated value in a fixed interval (e.g. every 100 ms for

safety relevant messages with Alive)

• Update on change — send an update as soon as a "value" changes (e.g. door

open)

• Epsilon change — only send an update when the difference to the last value is

greater than a certain epsilon. This concept may be adaptive, i.e. the prediction is

based on a history; thus, only when the difference between prediction and current

value is greater than epsilon an update is transmitted.

3. SOME/IP Service Discovery (SD)

Service Discovery (SD) is a mechanism in SOME/IP that enables automatic detection of services in a network.
It uses UDP multicast messages to announce availability, offer services, and find requested services.

4. Message Types

- Request (0x00): Client → Server call.
- RequestNoReturn (0x01): One-way call.
- Notification (0x02): Event from Server → Client.
- Response (0x80): Server → Client reply.
- Error (0x81): Indicates an issue with the call.

Message Types

Message Type	Value	Description
REQUEST	0x00	Client requests an action
REQUEST_NO_RETURN	0x01	Request without expecting response
NOTIFICATION	0x02	Event message (asynchronous)
RESPONSE	0x80	Server's reply to a request
ERROR	0x81	Error occurred during request

5. Choosing the transport protocol

SOME/IP supports User Datagram Protocol (UDP) and Transmission Control Protocol(TCP). While UDP is a very lean transport protocol supporting only the most importantfeatures (multiplexing and error detecting using a checksum), TCP adds additionalfeatures for achieving a reliable communication. TCP not only handles bit errors butalso segmentation, loss, duplication, reordering, and network congestion.

Inside a vehicle many applications requires very short timeout to react quickly. These requirements are better met using UDP because the application itself can handle the unlikely event of errors. For example, in use cases with cyclic data it is often the best approach to just wait for the next data transmission instead of trying to repair the last one. The major disadvantage of UDP is that it does not handle segmentation. Hence, only being able to transport smaller chunks of data.

Guideline:

• Use TCP only if very large chunks of data need to be transported (> 1400 Bytes) and no hard latency requirements in the case of errors exists.

• Use UDP if very hard latency requirements (<100ms) in case of errors is needed.

• Use UDP together with SOME/IP-TP if very large chunks of data need to be transported (> 1400 Bytes) and hard latency requirements in the case of errors exists.

• Try using external transport or transfer mechanisms (Network File System, APIX link, 1722, ...) when they are more suited for the use case. In this case SOME/IP can transport a file handle or a comparable identifier. This gives the designer additional freedom (e.g. in regard to caching).

The transport protocol used is specified by the interface specification on a per-message

basis. Methods, Events, and Fields should commonly only use a single transport pro-tocol.

6. Summary Notes

- SOME/IP is the backbone of AUTOSAR Ethernet communication.
- Supports both synchronous (method) and asynchronous (event) messaging.
- Allows dynamic discovery and binding through SOME/IP-SD.
- Provides flexibility, scalability, and interoperability in automotive networks.