VECTOR

SOME/IP in a Nutshell

A Middleware for Service Oriented Architecture in Modern Vehicles

V0.1 | 2019-07-09

Agenda

1. Overview

2. Header Format

3. Serialization

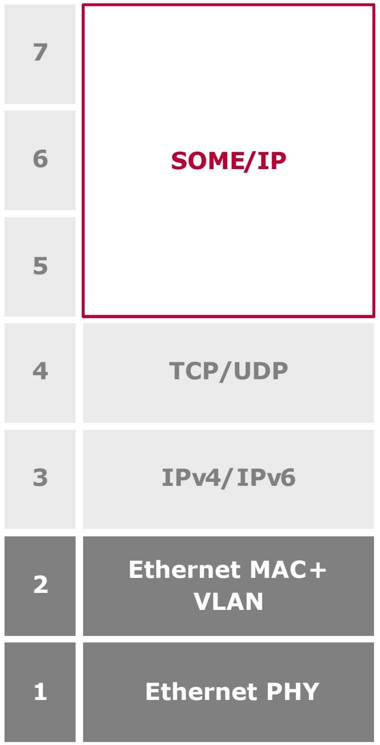
4. Service Discovery

What is SOME/IP?

- Stands for Scalable service-Oriented Middleware over IP
- Standardize:
  - Header format
  - Payload serialization rules
  - Service discovery mechanism
- Designed to be compatible with AUTOSAR, GENEVI and other plaftforms
- Supported by ARXML and FIBEX databases as well as Franca IDL

- Service-oriented control/command in-vehicle communication:
  - Support event notifications (similar to CAN, LIN or FlexRay)
  - Support Remote Procedure Calls (request/response similar to MOST)

- Rely on UDP and TCP sockets for peer-to-peer communication:
  - UDP: connectionless and unreliable (unicast/multicast)
  - TCP: connection-oriented and reliable (unicast only)



7	SOME/IP
6	
5	
4	TCP/UDP
3	IPv4/IPv6
2	Ethernet MAC+ VLAN
1	Ethernet PHY

What is a Service ?

- A service is a versioned interface‵s contract
- It communicates via kind of digital standardized formular
- A server can instantiate a service interface and provide a impleme
- Clients consume servers‵ service instances
- A service instance is uniquely identified by its location (IP Address
- Different servers can provide differente instances of the same serv
- A server can provide several instances of the same service interfac different location
- A client can consume several instances of the same service interfa
- Serialized message content is in direct relation to the service interf

Formular ID: **0x00020002**

Version: 2

Total length: 23 bytes

Name length: 8 bytes

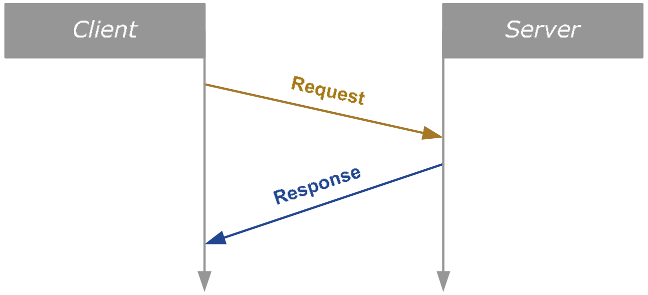
Name: John (wchar*)

Surname length: 3 bytes

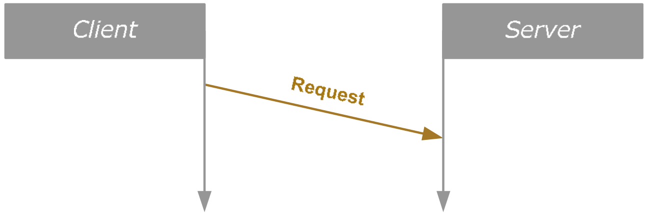
Surname: Doe (char*)

Age: 22 (uint32)

Communication Patterns (1/2)



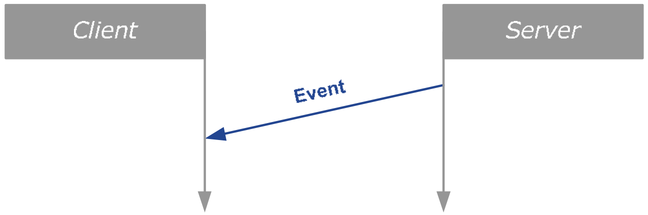
- Request/Response (Method):
  - Request: message which calls a method on a server
  - Response: message which returns the result of a method call



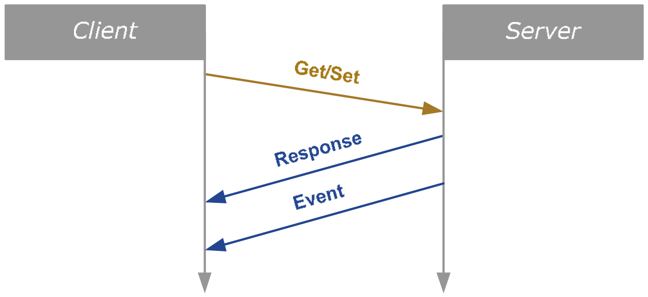
- Fire&Forget (Method):
  - Request: message which calls a method on a server
  - No response is returned

Overview

Communication Patterns (2/2)



- Event:
  - Server notifies previously subscribed clients, when something happens
  - A notification message is sent to the clients each time the event occurs even if the event is the same
  - Does not have a status, initial value or lifetime



- Field:
  - Has a status with an initial value and a lifetime
  - Get: method to read the current field value
  - Set: method to write the current field value
  - Response: message containing the current field value
  - Event: server notifies previously subscribed clients only when field value changed

Agenda

1. Overview

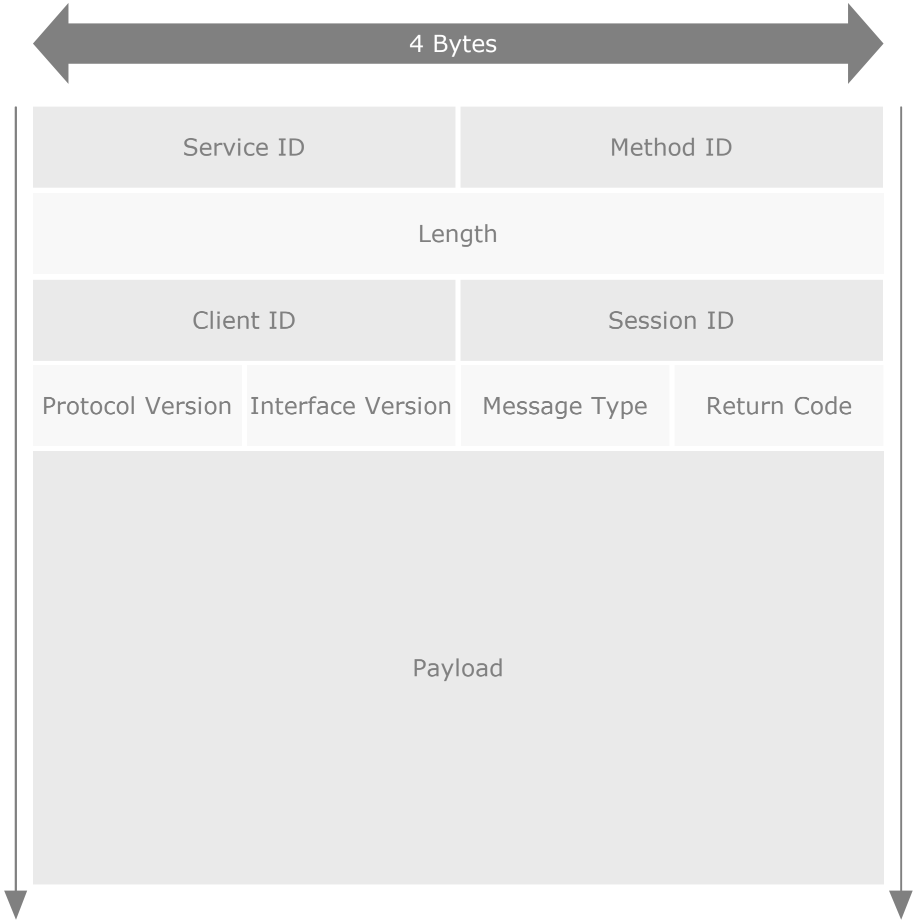
2. Header Format

3. Serialization

4. Service Discovery

Header Format

SOME/IP Message



4 Bytes

Service ID | Method ID

Length

Client ID | Session ID

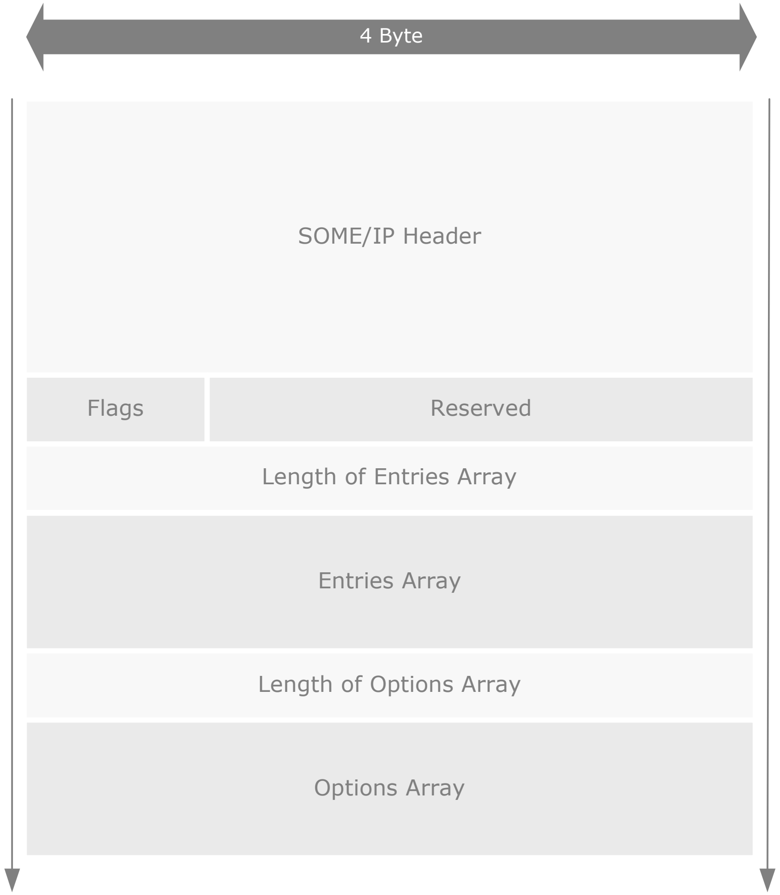
Protocol Version | Interface Version | Message Type | Return Code

Payload

- SOME/IP header is 16 bytes long:
  - 1st 4 bytes contains message identifier
  - 2nd 4 bytes contains message length
  - 3rd 4 bytes contains request identifier
  - 12th byte contains SOME/IP protocol version
  - 13th byte contains service interface major version
  - 14th byte contains message type, ex:
    - Notification
    - Request
    - Response
    - Request_no_return
  - 15th byte contains possible message error code
- SOME/IP Payload:
  - Contains content of a serialized method, event or field

Header Format

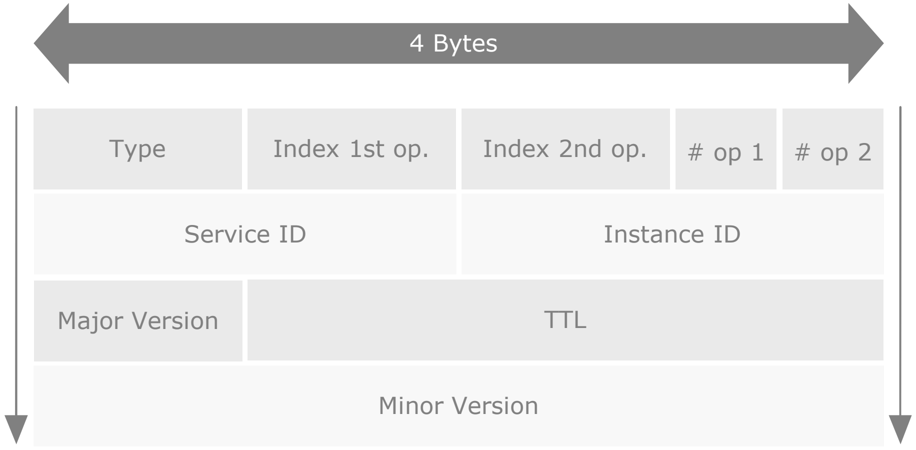
SOME/IP-SD Message



- SOME/IP header:
  - Same content as the normal SOME/IP header
  - Reserved message ID for SD is 0xFFFF8100
- SOME/IP-SD header:
  - Flags: used for ECU reboot detection
  - Length of entries array: tranports number of SD entries
  - Entries array contains entries of type Find, Offer, Subscribe, Subscribe ACK, Stop Offer, etc.
  - Length of options array: transports number of SD options referenced by entries
  - Options array contains options of type IPv4/IPv6 Endpoint, IPv4/IPv6 Multicast, Configuration, Protection, Load balancing, etc.

Entry Format Type

Service entry

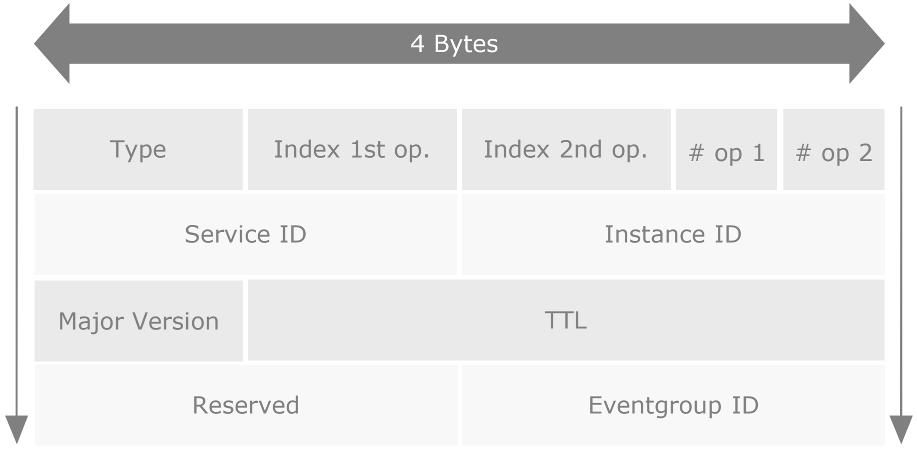


4 Bytes

Type	Index 1st op.	Index 2nd op.	# op 1	# op 2
Service ID		Instance ID		
Major Version	TTL			
Minor Version				

- Used by entry of type:
  - FindService (0x00)
  - OfferService (0x01)

Eventgroup entry



4 Bytes

Type	Index 1st op.	Index 2nd op.	# op 1	# op 2
Service ID		Instance ID		
Major Version	TTL			
Reserved		Eventgroup ID		

- Used by entry of type:
  - Subscribe (0x06)
  - SubscribeAck (0x07)

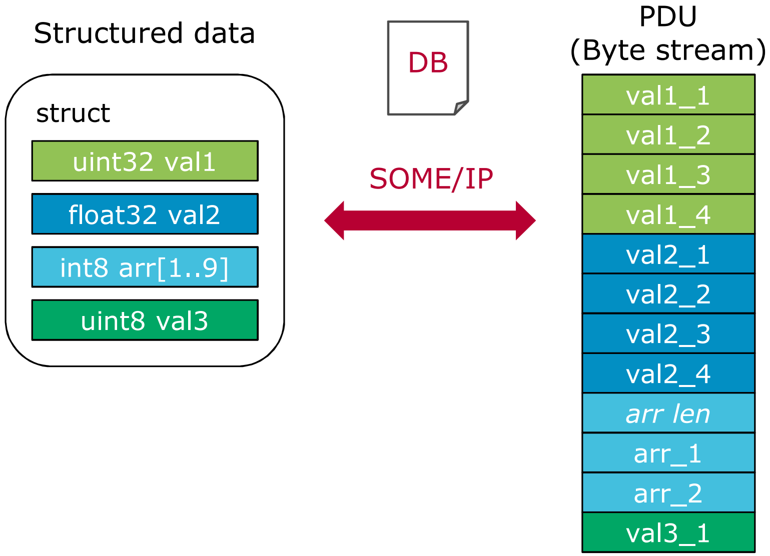
* TTL=0 means „stop“, ex: StopOffer -> Type=0x01 and TTL=0

Agenda

1. Overview
2. Header Format
3. Serialization
4. Service Discovery

SOME/IP Payload

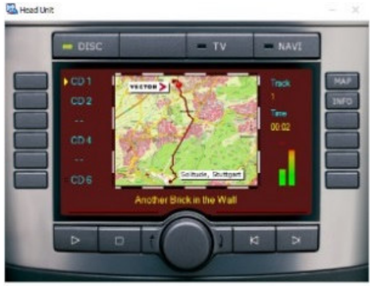
- In CAN or FlexRay signals are "serialized" statically to fit the message layout specified in the database
- Especially in the ADAS domain, there are more and more data elements of dynamic length or optional
  - A static message layout considers the worst case scenario
    - "Empty" signals are always transmitted

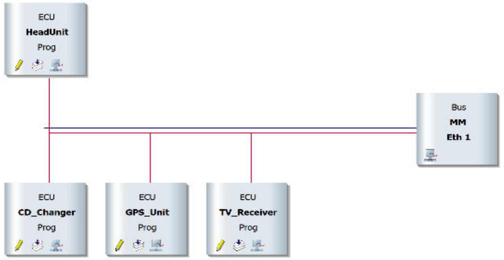


- (De-)Serialization during runtime
  - Signals and messages of dynamic length or optional
  - Don't specify a fix message layout in a database
  - Serialization layout similar to a C struct in memory layout
  - Transmit only the relevant and currently available data elements
    - Save bandwidth
    - Save computing resources

Example

SOME/IP : Simulation of Multimedia in SOME/IP : SOMEIPSimMultimedia.cfg





Setup : Head Unit , CD_Changer , GPS_Unit, TV_receiver

You can operate a **HeadUnit** and a **CD Changer** with the panels on the Trace desktop. With the Head Unit, for example, you select a certain CD and the title to be played. The command is converted to the corresponding bus communication by the assigned CAPL node and the SOME/IP-IL. The same applies to information that the CD changer returns to the head unit.

For more information about Vector
and our products please visit

www.vector.com

Author:
Laygude , Prashant
Vector India