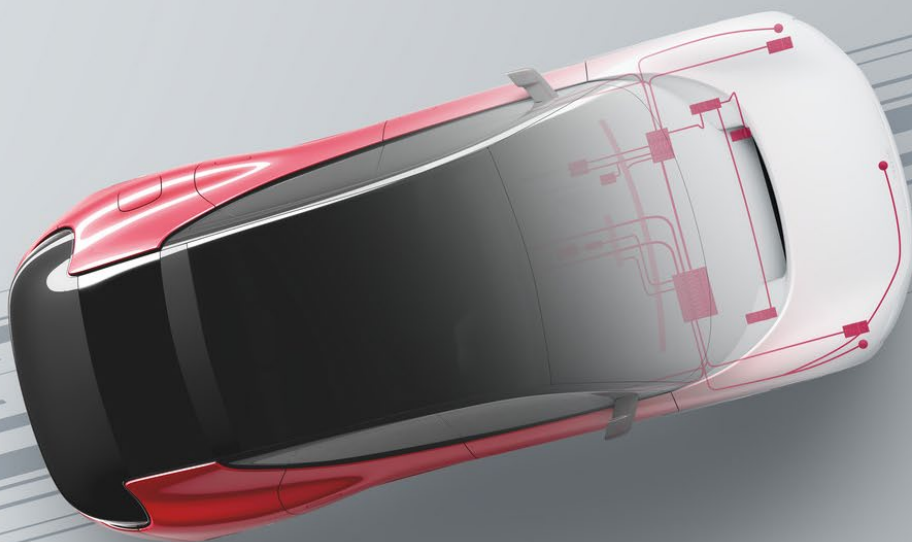


はじめての SOME/IP

ビギナー向け資料「はじめてシリーズ」



目次

はじめに.....	4
対象者	4
1. 近年の車載通信の動向.....	5
2. SOA とは.....	7
2.1 SOA の概要	7
2.1.1 SOA の車載通信への導入イメージ	8
2.2 SOA のキーワード.....	8
3. SOME/IP とは	9
3.1 SOME/IP の概要	9
3.2 SOME/IP の用途、適用範囲	10
3.3 SOME/IP のキーワード	10
4. SOME/IP プロトコル	13
4.1 SOME/IP-SD とは	13
4.1.1 SOME/IP-SD の例.....	13
4.1.2 SOME/IP-SD の処理	14
4.1.3 SOME/IP-SD の動作フェーズ	16
4.2 Remote Procedure Call とは	18
4.2.1 Remote Procedure Call の例	18
4.2.2 Remote Procedure Call の処理.....	19
4.3 SOME/IP のメッセージフォーマット.....	20
4.3.1 SOME/IP-SD のメッセージフォーマット	22
4.4 ペイロードのフォーマット	24
4.4.1 データ種類	24
4.4.2 動的なシリアライゼーション	25
5. SOME/IP-TP とは	26
5.1 SOME/IP-TP のヘッダー構造	26
5.2 SOME/IP-TP の動作.....	28
あとながき	29
6. お問い合わせ窓口	30

発行元:ベクター・ジャパン株式会社

ベクター・ジャパン株式会社の書面による許可なしに、本書の内容を転載、複製、複写することを禁じます。
記述されている内容や製品画面の表示名称は予告なく変更されることがあります。

はじめに

近年では車載 Ethernet の導入が進み、車載アーキテクチャが変化していくに伴い、SOME/IP プロトコルの採用が進んでいます。本稿では、SOME/IP が採用されている背景から、SOME/IP を使用するメリット、その動作に至るまで、SOME/IP の概要を紹介します。

対象者

本稿は、以下の内容を理解している方を対象にしています。

- > 車載通信の基本を理解している方
- > 車載 Ethernet の基本を理解している方
- > TCP/IP の基本を理解している方

上記内容を確認したい方のため、CAN や車載 Ethernet などの概要を紹介した「はじめてシリーズ」を用意しています。ベクターの Web サイト「はじめてシリーズ」よりダウンロードできます。

<https://www.vector.com/jp/ja/know-how/vj-beginners/>

- > はじめての CAN/CAN FD
- > はじめての車載 Ethernet

1. 近年の車載通信の動向

近年は、車載通信に Ethernet を採用する流れが加速しています。その背景には CASE があります。

近年の車両は自動運転(ADAS)を実現するため、センサーや LiDAR を多数搭載し、その高精細化が進んでいます。自動運転を実現するための機械学習など、ソフトウェアの高性能化も進み、膨大なデータを取得し、通信するための高速な車載ネットワークが求められています。さらには Car2X などのコネクティビティ(Connectivity)に代表される、車外デバイスとの接続やバックエンドサービスなどの AUTOSAR に準拠しないサービスとの接続が求められてきています。

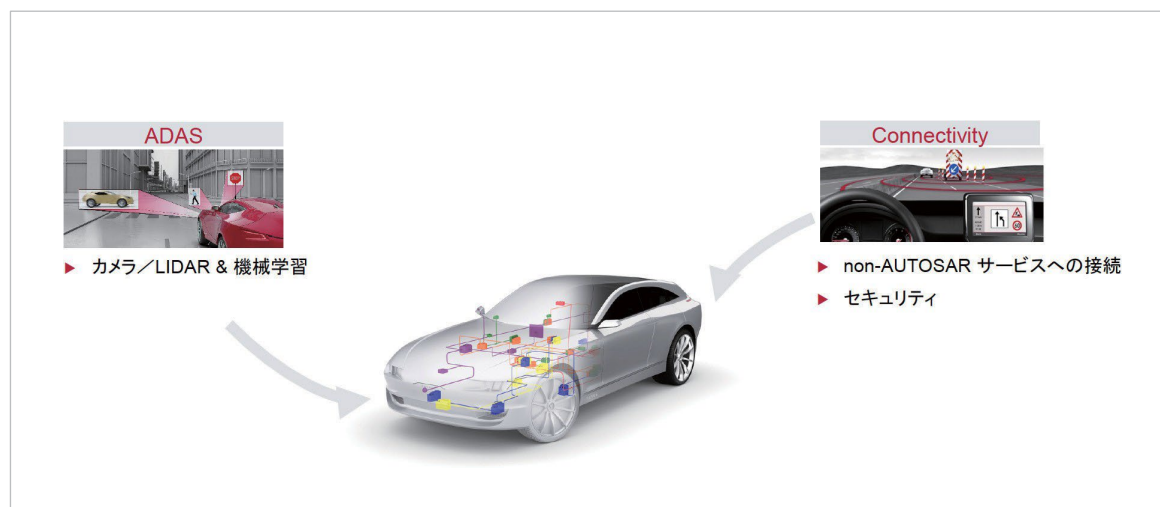


図 1: 近年の車載通信のトレンド

CASE の影響は、車両の EE アーキテクチャにも変化を与えています。

最近では、欧州の OEM を中心に、以下のようなゾーンアーキテクチャが検討されています。

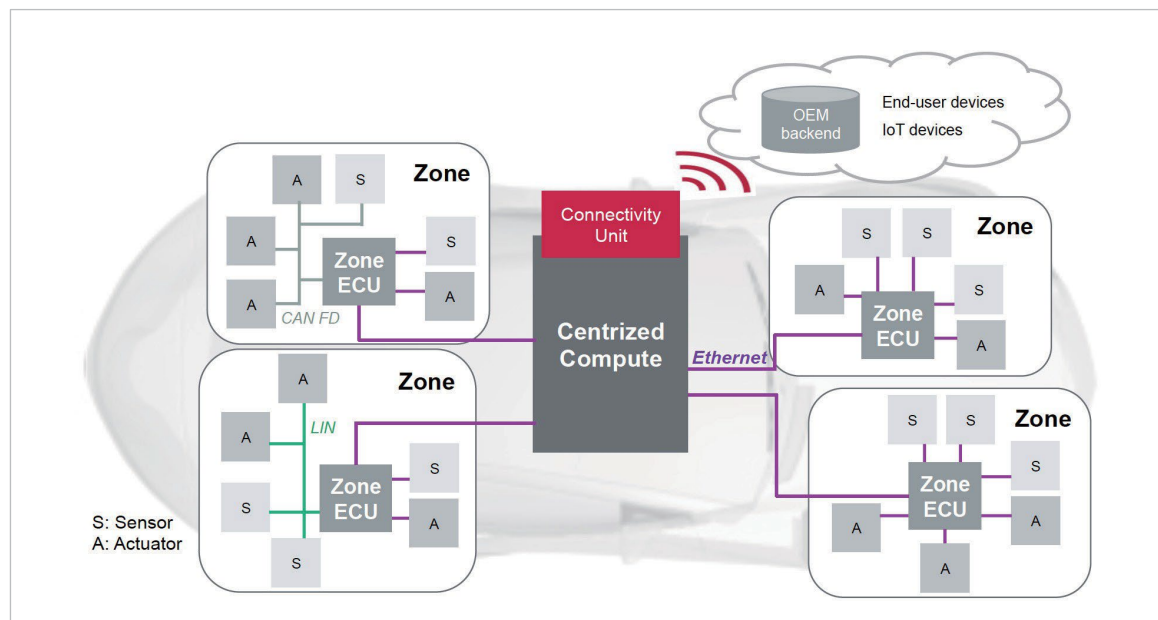


図 2: 将来の車両の EE アーキテクチャ『ゾーンアーキテクチャ』

ゾーンアーキテクチャは、車両をいくつかのゾーンに分割します。それぞれのゾーンには、センサーやアクチュエーターと、ドメインコントロールと Gateway 機能を兼ねた ECU (Zone ECU) を配置します。Zone ECU が中央集中管理用の HPC (High Performance Computer) と通信するような構成です。

ネットワークの複雑化を回避するために ECU は Zone ECU に削減・統合され、数少ない Zone ECU と HPC がさまざまな機能を持つようになります。

今後は外部のネットワークと接続し、車両出荷後にも機能更新・アップグレードできるような機能が追加され、「自動車がスマートフォン化していく」と言われています。

現在の車載ネットワークでは、アプリケーションの機能は ECU に強く紐づけられています。この機能を ECU 外から使うのは容易ではありません。将来的にクルマがスマートフォン化していく中、新しくダウンロードしたアプリが、クルマのネットワークにあるさまざまな機能を見つけ出し、それを自由に使える仕組みが求められています。

このような背景から、SOA などの新しい設計手法の導入が進んでいます。

2. SOA とは

車載通信や EE アーキテクチャの要求の変化に伴い、車載ネットワークに SOA の概念を導入する流れが進んでいます。本章では、SOA の概要について解説します。

2.1 SOA の概要

SOA (Service Oriented Architecture : サービス指向アーキテクチャ) とは、大規模なコンピューターシステムを構築する際の考え方・手法です。SOA は自動車に特化したものではなく、一般的なコンピューターシステムで使われてきた概念です。SOA 自体は概念であり、明確な仕様や定義などは存在しません。

SOA では、アプリケーションの機能や処理を「サービス」と呼ばれる単位に細分化・部品化して、サービスをアプリケーションの外部から使用可能な状態にします。システムを構築・追加する際には、部品化したサービス(機能)を組み合わせることでシステムを実装します。SOA はアプリケーションの機能が「サービス」という形で独立しているため、機能を再利用しやすいのが利点です。新しいシステムの導入や機能更新など、さまざまな要求にも素早く対応できます。

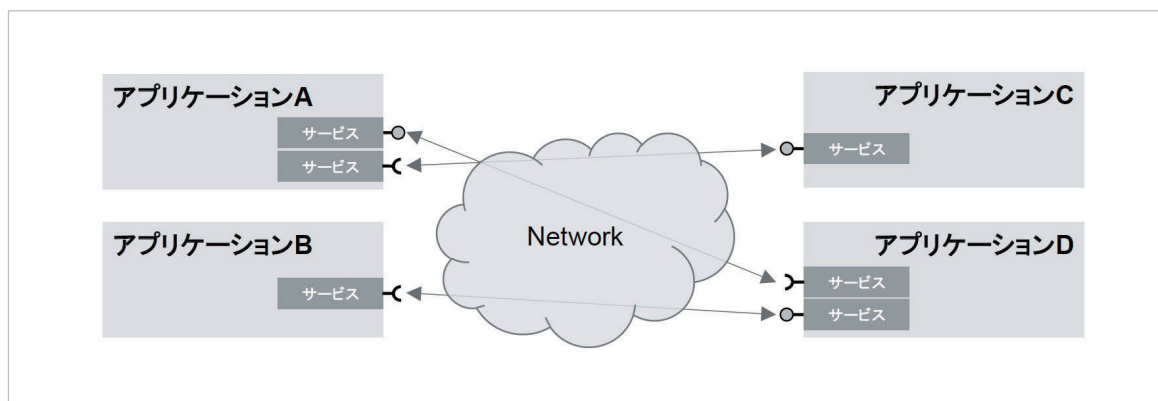


図 3: SOA のイメージ

【NOTE】

SOA の基本概念である「機能の部品化」という考え方は、古くから存在していました。ソフトウェア開発におけるサブルーチンや共通ライブラリ、オブジェクト指向プログラミング、分散オブジェクト(コンポーネント)などの手法を拡大したものが SOA と言えます。

2.1.1 SOA の車載通信への導入イメージ

以下のような運転支援系のシステムで、統合制御ユニットの経路計算サービスを更新し、更新した機能をハイパフォーマンスなデバイスで実行したい場合を考えます。

SOA では、経路計算サービスに関連しないサービスはそのままに、経路計算サービスに関連するサービスだけを更新して対応できます。経路計算サービスを HPC(High Performance Computing)等のハイパフォーマンスなデバイスに移行して経路計算サービスを実行できるため、ECU ハードウェアに依存せずにシステムを更新できます。

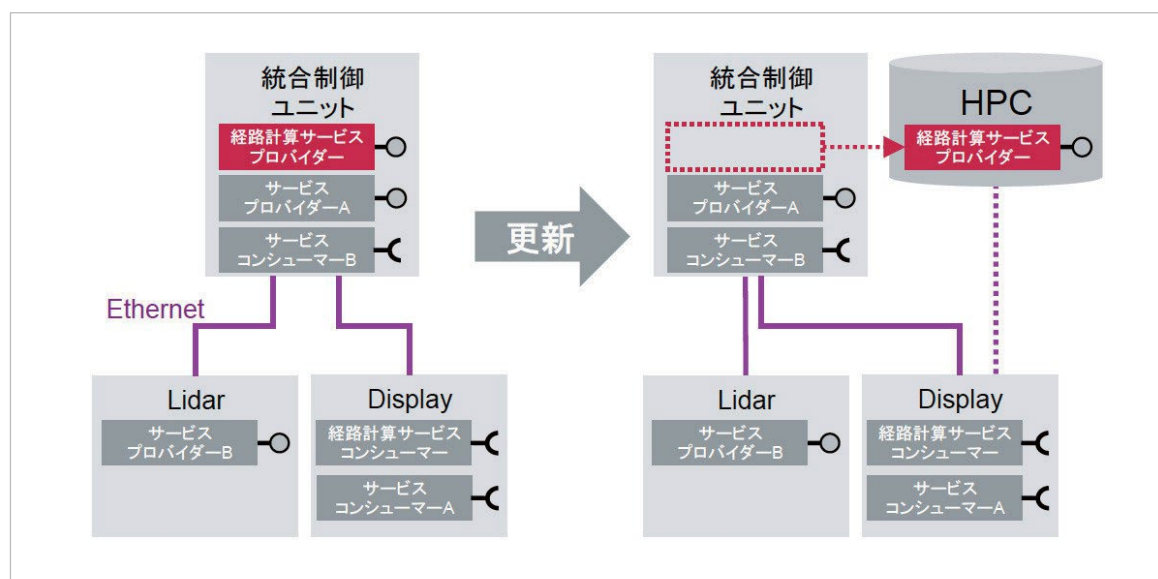


図 4:SOA のシステム更新イメージ

2.2 SOA のキーワード

本節では、SOA を理解するうえでのキーワードを紹介します。

> サービス

SOA におけるサービスとは、ソフトウェア・アプリケーションの機能を細分化・部品化して、外部からも使用できる状態にしたものです。一般的には、複数のアプリケーションにまたがって共有される機能(ソフトウェア部品)を指します。

どこまで機能を細分化・部品化するかといった明確な基準はありません。サービスの粒度は、ソフトウェア・アプリケーションの設計次第です。

3. SOME/IP とは

「ネットワーク上のサービスを組み合わせて使用する」などの SOA の概念を車載ネットワークに取り入れたものが、サービス指向の車載通信プロトコルである SOME/IP および SOME/IP-SD です。本章では SOME/IP の概要を紹介します。

3.1 SOME/IP の概要

SOME/IP (Scalable service- Oriented MiddlewarE over IP) は、Ethernet をベースにした通信プロトコルです。TCP もしくは UDP 上で動作し (SOME/IP-SD は UDP 上のみ動作します)、OSI モデルでは、以下のようにネットワーク層以上の上位層に分類されます。また、AUTOSAR、GENEVI などのさまざまなプラットフォームに対応しています。

SOME/IP では、カメラ機能やステアリング制御機能等のアプリケーションの機能・処理を「サービス」単位で定義します。サービスを使用する時は、ネットワーク上にあるサービスを検索し、検知したサービスを組み合わせてシステムを構築します。データは共通の Interface を経由して交換します。

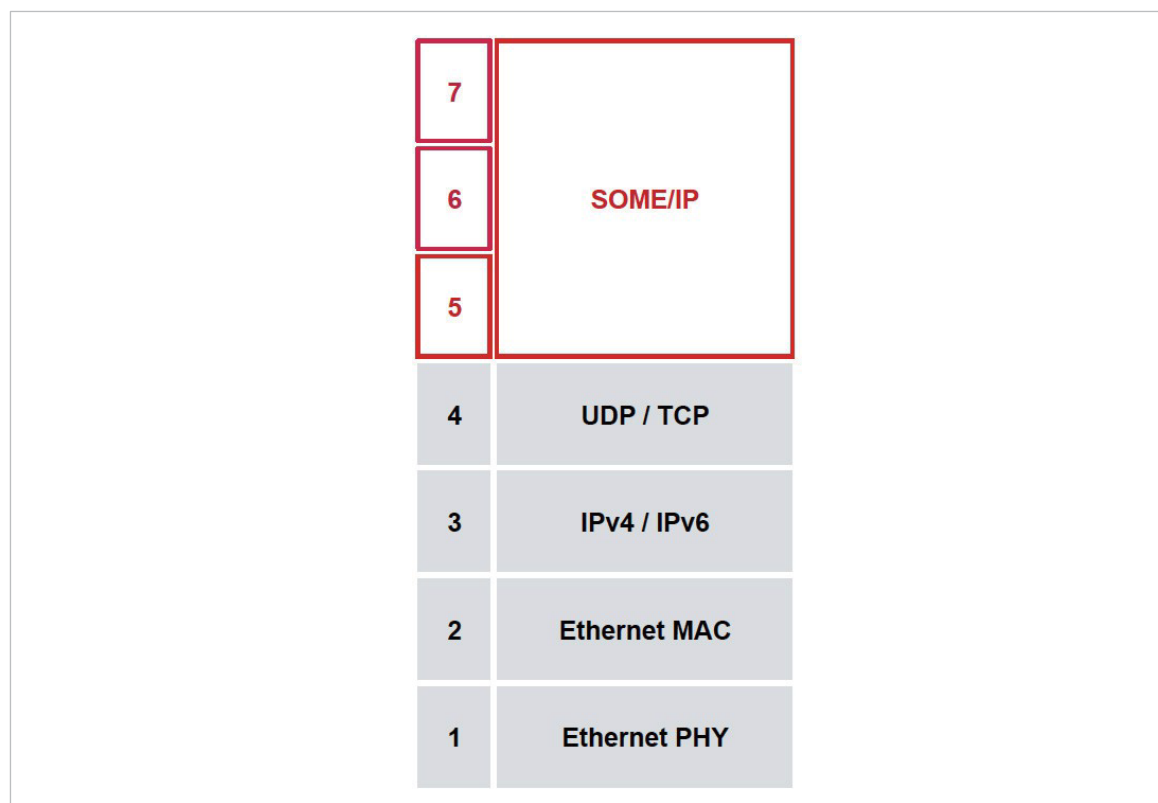


図 5: SOME/IP の OSI レイヤー構造

3.2 SOME/IP の用途、適用範囲

2022 年現在ではインフォテインメントシステムや ADAS の周辺カメラシステムなどに使用されています。欧州 OEM の車両を中心に採用されており、今後も採用が広がっていくと考えられます。

3.3 SOME/IP のキーワード

本節では SOME/IP を理解するうえでのキーワードを紹介します。

> SOME/IP のサービス

SOME/IP におけるサービスは、カメラサービスやステアリング制御サービス等、抽象的な機能の単位を指します。サービスは 1 つ以上のサービス IF を持ち、固有の ID (Service ID) で識別します。

以下のように、サービスはサービス IF、Method、Event、Field で構成されます。

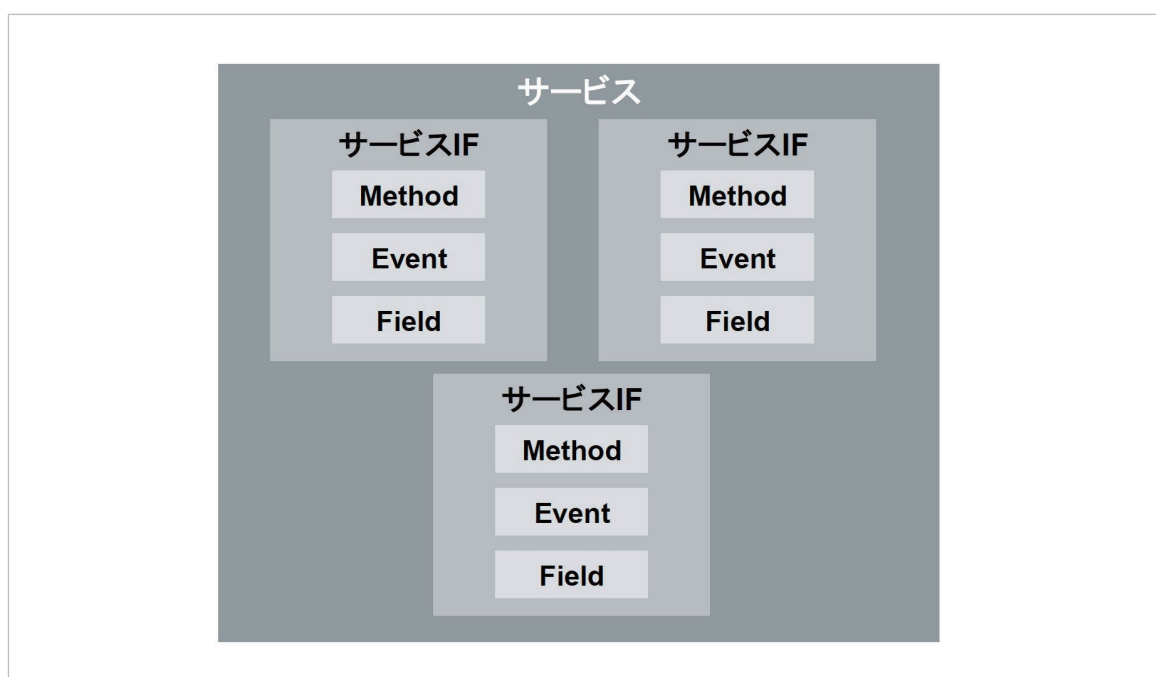


図 6: サービスとサービスの構成要素の関係

サービスを構成する要素を紹介します。

> サービス IF

サービス IF (インターフェイス) は、サービスを持つ機能にアクセスするためのインターフェイスです。サービス IF はネットワーク上で共通の仕様を持ちます。サービスの API 仕様とも言えます。

> Method

クライアントから呼び出されるサーバー上の関数を Method (メソッド) と呼びます。メソッドは固有の ID (Method ID) で識別します。

> Event

データが変更された時や周期的な呼び出し時など、何らかのアクションが発生した時に発生する処理と、そのデータを Event (イベント) と呼びます。

> Field

Field (フィールド) とはサービスが提供するデータとプロパティです。Field の種類を以下に示します。

- ✓ Getter
Getter とは Field の値を取得する処理です。
- ✓ Setter
Setter とは Field の値を変更する処理です。
- ✓ Field Notification
Field Notification とは Field の値が変更された時に発生する処理です。

【NOTE】Field と Event の違い

Field は、プロバイダ側で属性値として常に保持しているデータです。Field Notification では、この属性値を通知します。

Event は、通常はプロバイダ側で値を保持しません。Event の送信では、イベント発生などの瞬間的な値を通知します。

> サービスインスタンス

サービスをネットワーク等で実際に使用するには、インスタンス化※する必要があります。サービス IF をインスタンス化し、使用可能にしたものをサービスインスタンスと呼びます。サービスインスタンスはソフトウェアコンポーネントに実装され、ネットワーク上で 1 つのみ存在します。サービスインスタンスは固有の ID (Instance ID) で識別します。サービスとサービスインスタンスなどの関係を以下に示します。

※インスタンス化とは、サービスなどの抽象的なものを実装(実体化)し、ネットワーク上などの環境で使用可能な状態にする処理です。

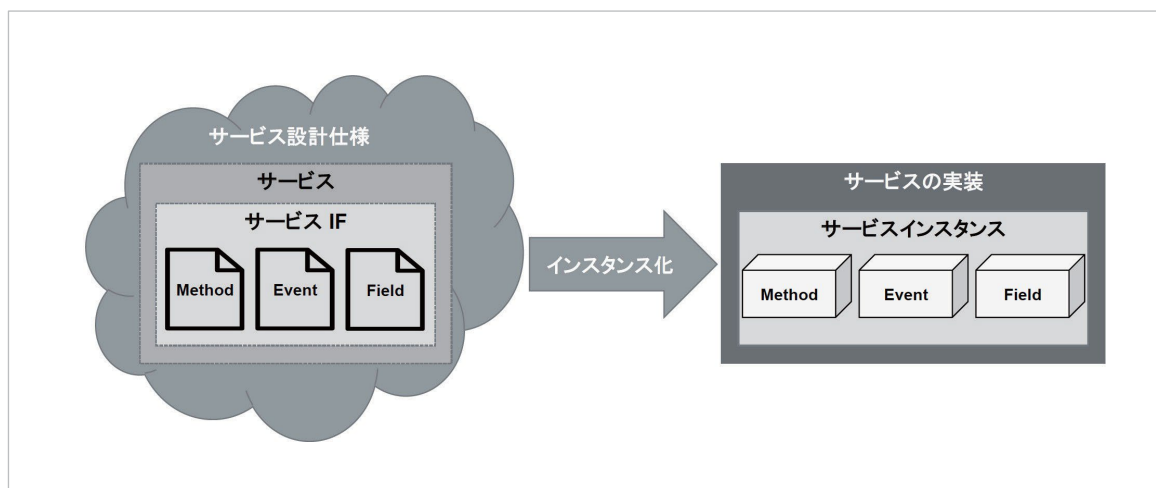


図 7: サービスとサービスインスタンスの関係

4. SOME/IP プロトコル

本章では、SOME/IP プロトコルの動作について説明します。SOME/IP の通信仕様は主に 2 種類に分けられます。

1. SOME/IP-SD
2. Remote Procedure Call

それぞれの概要を説明します。

4.1 SOME/IP-SD とは

SOME/IP-SD(Service Discovery)とは、ネットワーク上にあるサービスを検索して使用する処理です。SOME/IP-SD を使用すると、必要な時に必要なサービスを検索して使用するため、ネットワーク上のサービスを動的に参照／使用できます。ハードウェアに依存せず機能を開発／追加する事も可能です。

【NOTE】

SOME/IP-SD は UDP 上のみで動作します。

4.1.1 SOME/IP-SD の例

SOME/IP-SD を使用して、後付けした後方カメラサービスを使用する場合を考えます。

- ① ECU は、ネットワーク上で後方カメラのサービスを検索します。
- ② 後方カメラサービスを持つ後方カメラは、サービスがある事をネットワーク上へ通知します。ECU は通知を受け取り、後方カメラサービスを使用できるようになります。

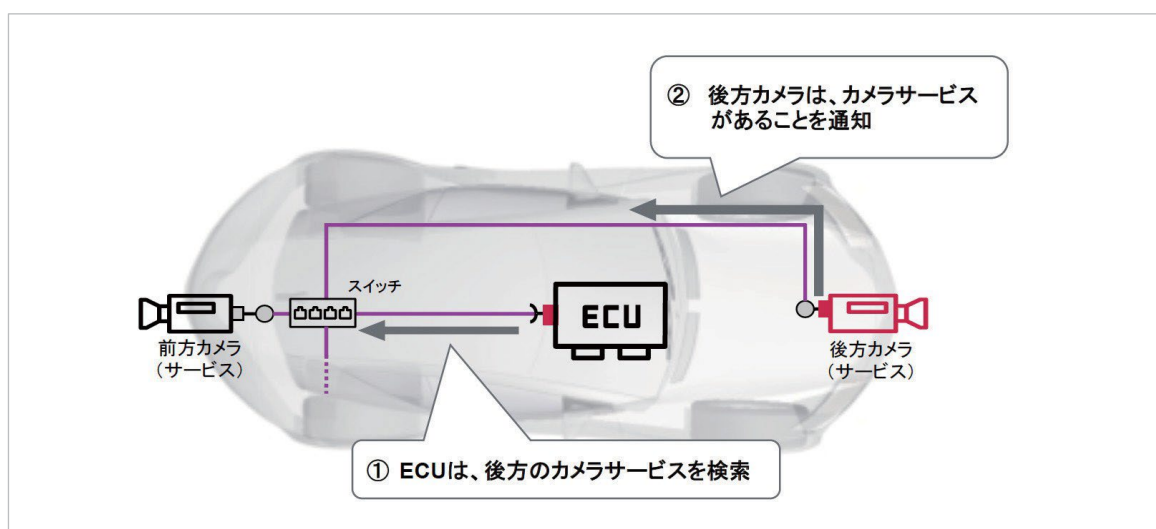


図 8: SOME/IP-SD の動作イメージ

4.1.2 SOME/IP-SD の処理

ネットワーク上の ECU が、SOME/IP-SD で他の ECU のサービスを検索して使用する場合があります。この時、サービスを使用する側をクライアント、サービスを提供する側をサーバーと呼びます。

クライアント-サーバー間の SOME/IP-SD 通信シーケンスの例を以下に示します。

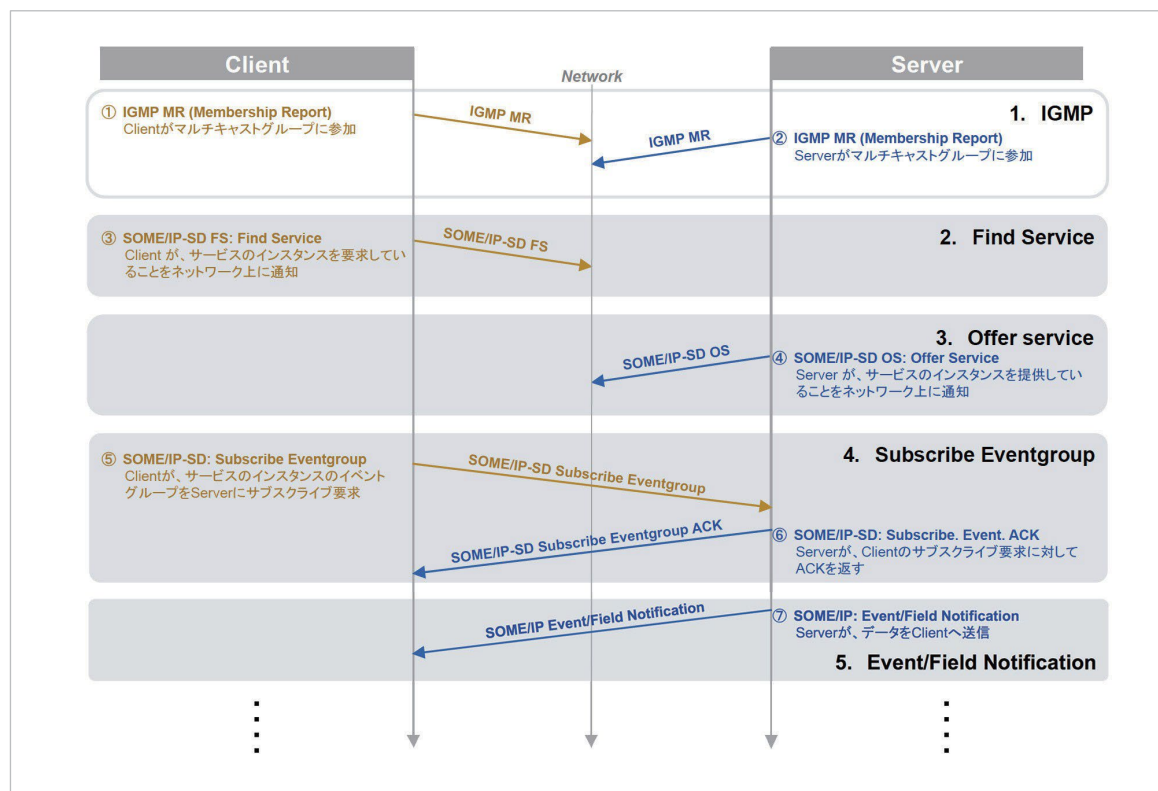


図 9: クライアント-サーバー間の SOME/IP-SD 通信シーケンスの例

それぞれのシーケンスでの動作を説明します。

1. IGMP

クライアント・サーバーそれぞれがネットワーク上に参加するため、IGMP を送信します。
(IGMP は固有の通信プロトコルです。SOME/IP や SOME/IP-SD に含まれるものではありません。)

2. Find service

クライアントは、サービスを検索している事をネットワーク上に通知します。

3. Offer service

サーバーは、自分が提供できるサービスをネットワーク上に通知します。

4. Subscribe Eventgroup

クライアントは、ネットワーク上の Offer Service を受信してサービスを使用できる事を確認し、サブスクライブ(購読)したいイベントグループをサーバーへリクエストします。サーバーはクライアントからのリクエストを受け取り、承認します。

これでクライアントはイベントグループを購読できる状態になります。

5. Event/Field Notification

サーバーは、値をクライアントへ通知します。Event Notification と Field Notification の 2 種類の方法で値を送信します。

A) Event Notification

周期的や、値の変更・更新時など、何らかのイベントが発生した時にデータを送信します。

ADAS カメラの交通標識認識系の情報の通知などに使用されます。

B) Field Notification

Field の値が変更された時にデータを送信します。

ADAS カメラの転送レート情報の通知などに使用されます。

4.1.3 SOME/IP-SD の動作フェーズ

SOME/IP-SD には以下 4 つの動作フェーズが存在します。

1. **Down Phase**
2. **Initial Wait Phase**
3. **Repetition Phase**
4. **Main Phase**

クライアントの場合・サーバーの場合それぞれにおいて、各フェーズの動作が異なります。それぞれの動作を説明します。

クライアントの場合

クライアントの場合の SOME/IP-SD の動作フェーズを説明します。

1. **Down Phase**

Down Phase はソケットがクローズし、SOME/IP 通信をしていない状態です。

2. **Initial Wait Phase**

Down Phase 中にサービスインスタンスが起動すると Initial Wait Phase へ移行します。

Initial Wait Phase に移行後、ランダム時間※だけ待機します。ランダム時間が経過後、Find service を送信して Repetition Phase へ移行します。

Initial Wait Phase で Offer service を受信すると Main Phase へ移行します。

3. **Repetition Phase**

Repetition Phase で Offer service を受信すると Main Phase へ移行します。

4. **Main Phase**

Main Phase では既に Offer service を受信した後であるため、Find service は送信しません。

サーバーの場合

サーバーの場合の SOME/IP-SD の動作フェーズを説明します。

1. Down Phase

Down Phase はソケットがクローズし、SOME/IP 通信をしていない状態です。

2. Initial Wait Phase

Down Phase 中にサービスインスタンスが起動すると Initial Wait Phase へ移行します。

Initial Wait Phase に移行後、ランダム時間※だけ待機します。ランダム時間が経過後、Offer service を送信して Repetition Phase へ移行します。

Initial Wait Phase で受信した Find service はすべて無視します。

3. Repetition Phase

Repetition Phase では、あらかじめ設定した周期・回数で Offer service を周期送信します。設定した回数まで Offer service を送信した後、Main Phase へ移行します。

Initial Wait Phase で受信した Find service はすべて無視します。

4. Main Phase

Main Phase では Offer service を周期的に送信します。

【NOTE】ランダム時間だけ待機する目的

もしネットワークですべての ECU やサービスインスタンスが同時に起動した場合、ネットワーク上に SOME/IP-SD が大量に流れて通信負荷が増大します。ネットワークの負荷軽減のため、Initial Wait Phase で「ランダム時間だけ待機する」処理が実装されています。

すべてのサービスインスタンスが同時に起動しても、それぞれのサービスインスタンスがランダムな時間だけ待機した後に Find Service/Offer Service を送信することで、ネットワークの負荷を分散できます。このランダム時間は、あらかじめ設定した最小値と最大値の間から無作為に選ばれます。

クライアント・サーバーそれぞれの動作フェーズの例を以下に示します。

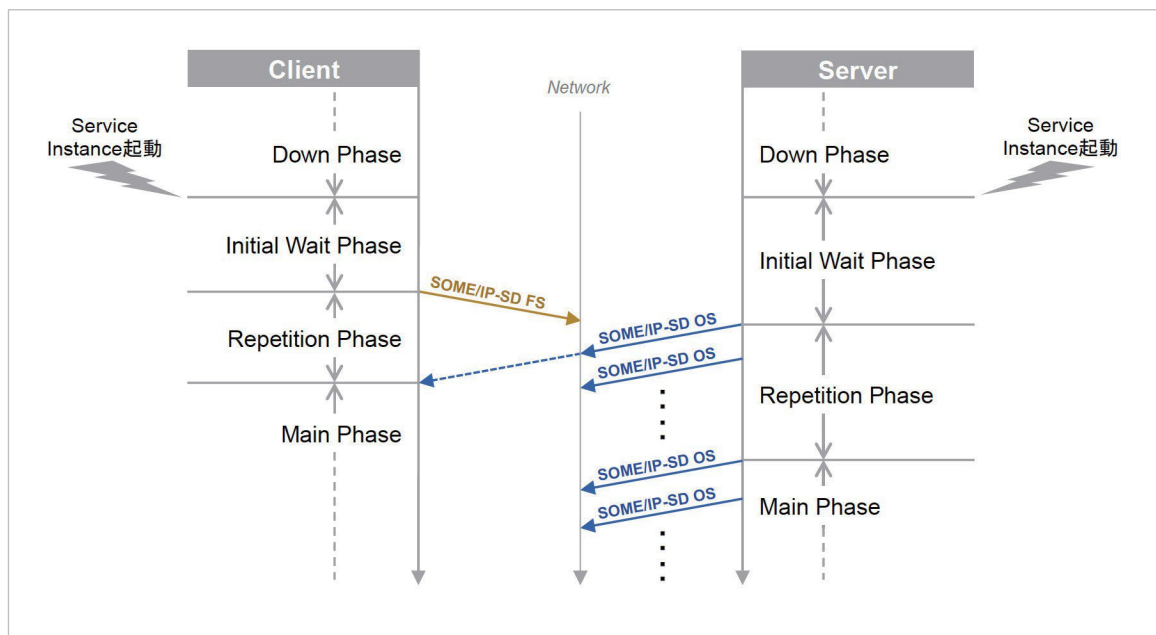


図 10: クライアント・サーバーそれぞれの動作フェーズの例

4.2 Remote Procedure Call とは

Remote Procedure Call (RPC) とは、処理の実行を他のソフトウェアに依頼する処理です。

Remote Procedure Call を使用すると、高負荷な処理を遠隔のバックエンドサーバーや HPC (High Performance Computing) などに依頼できるため、自分自身の処理負荷を軽減できます。機能を簡単に更新／使用できる事も利点です。

4.2.1 Remote Procedure Call の例

Remote Procedure Call を使用して、ヘッドユニットが目的地へのルートを計算する場合を考えます。

- ① ヘッドユニットは、現在位置と目的地の情報を、ルート計算サービスを持つ HPC (High Performance Computer) へ送信します。
- ② HPC は、受け取った現在位置と目的地の情報からルートを計算し、結果のみをヘッドユニットに返します。結果を受け取ったヘッドユニットは、自分自身でルート計算処理を実行する事なく、結果を取得・使用できます。

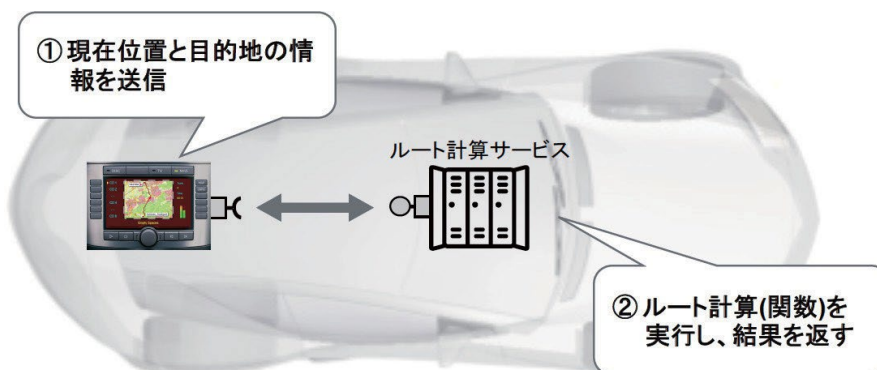


図 11: Remote Procedure Call の動作イメージ

4.2.2 Remote Procedure Call の処理

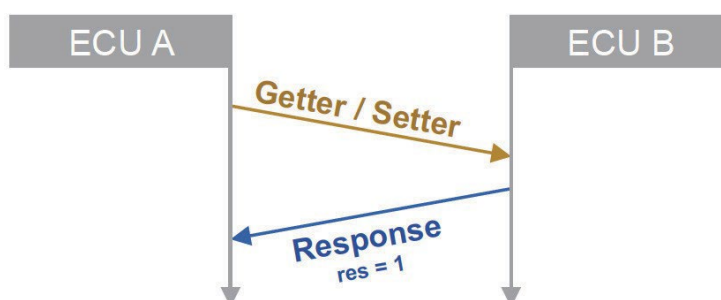
Remote Procedure Call の処理は Getter / Setter、Method call、Fire & forget の 3 つに分けられます。

それぞれの処理内容を説明します。

> Getter / Setter

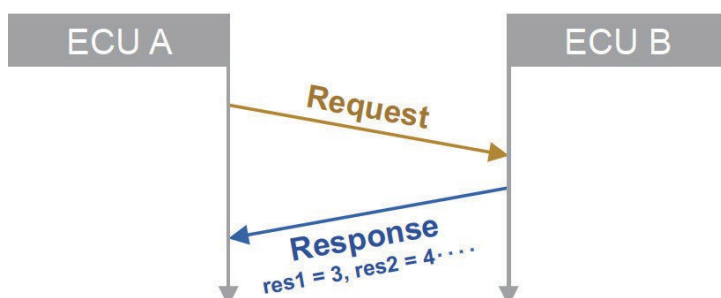
Getter / Setter は Field に関する処理を依頼し、その結果を受け取る処理です。Getter は Field の値を取得し、Setter は Field の値を設定する処理です。

ADAS カメラの転送レートの取得／変更処理などで使用されます。



> Method call

Method call は処理を依頼し、処理結果(戻り値)を受け取る処理です。ナビゲーションのルート演算処理などで使用されます。



> Fire & forget

Fire & forget は処理の依頼のみを行う処理です。結果は受け取りません。カーエアコンのスタート／ストップ処理などで使用されます。



4.3 SOME/IP のメッセージフォーマット

本節では、SOME/IP の各メッセージのフォーマットを紹介します。

SOME/IP のヘッダー長は 16byte です。SOME/IP のメッセージフォーマットを以下に示します。

byte	1	2	3	4
0	Service ID		Method ID	
4	Length			
8	Client ID		Session ID	
12	Protocol Version	InterfaceVersion	Message Type	Return Code
16	Payload			
..				

SOME/IP メッセージフォーマットの各領域を説明します。

> **Service ID**

サービスの種類を表します。

> **Method ID**

Method の種類を表します。

(Service ID と Method ID を合わせて“Message ID“とする場合もあります。)

> **Length**

Client ID からペイロードまでのメッセージ長をバイト単位で表します。

> **Client ID**

ECU 内の呼び出し元クライアントを識別する識別子です。

ECU は、複数のクライアントからの呼び出しを同じメソッドに区別できます。

> **Session ID**

同じ送信者から発信されたメッセージを区別する識別子です。

(Client ID と Session ID を合わせて“Request ID“とする場合もあります。)

> **Protocol Version**

SOME/IP のプロトコルバージョンを表します。

> **Interface Version**

サービス I/F のメジャーバージョンを表します。

> **Message Type**

SOME/IP のメッセージタイプを表します。値とメッセージタイプ詳細の関係を以下に示します。

No.	値	メッセージタイプ	説明
1	0x00	REQUEST	リクエスト(応答を期待する)
2	0x01	REQUEST_NO_RETURN	Fire & Forget
3	0x02	NOTIFICATION	通知(応答を期待しない)、イベントコールバックの要求
4	0x80	RESPONSE	レスポンスメッセージ
5	0x81	ERROR	エラー

> **Return Code**

リクエストが正常に処理されたかどうかを示す戻り値を表します。

> **SOME/IP Payload**

データ(ペイロード)を表します。

4.3.1 SOME/IP-SD のメッセージフォーマット

SOME/IP-SD ではメッセージフォーマットが変化します。Service ID, Method ID は SOME/IP-SD 専用の値を使用し、Return Code の後に Flags～Options Array が追加されます。

SOME/IP-SD のメッセージフォーマットを以下に示します。

byte	1	2	3	4
0	Service ID		Method ID	
4	Length			
8	Client ID		Session ID	
12	Protocol Version	Interface Version	Message Type	Return Code
16	Flags	Reserved		
20	Length of Entries Array			
24	Entries Array			
..				
..	Length of Options Array			
..	Options Array			
..				

SOME/IP-SD メッセージフォーマットの各領域を説明します。

> **Service ID**

Service ID は、SOME/IP-SD 専用の値として 0xFFFF が使用されます。

> **Method ID**

Method ID は、SOME/IP-SD 専用の値として 0x8100 が使用されます。

> **Flags**

8bit 長の Flags が挿入されます。

> **Reserved**

今後の機能拡張のためのリザーブとして 24bit 長が挿入されます。

> **Length of Entries Array**

Entries Array のデータ長を 32bit で、バイト単位で表します。

> **Entries Array**

SOME/IP-SD の情報を示すデータが可変長で挿入されます。

> **Length of Options Array**

Options Array のデータ長を 32bit で、バイト単位で表します。

> **Options Array**

SOME/IP-SD のオプションを示すデータが可変長で挿入されます。

【NOTE】

SOME/IP-SD の Length、Client ID、Session ID、Protocol Version、Interface Version、Message Type、Return Code は SOME/IP と同様のフォーマットです。

4.4 ペイロードのフォーマット

本節では、SOME/IP のペイロードのフォーマットの概要を紹介します。

4.4.1 データ種類

SOME/IP で使用される基本的なデータの種別を以下に示します。

上記の他に、構造体(structs)、配列(Array)、文字列(Strings)などを使用できます。

No.	Type	説 明	データサイズ [bit]	補 足
1	boolean	TRUE / FALSE	8	FALSE(0)、TRUE(1)
2	uint8	Unsigned Integer	8	—
3	uint16	Unsigned Integer	16	—
4	uint32	Unsigned Integer	32	—
5	uint64	Unsigned Integer	64	—
6	sint8	Signed Integer	8	—
7	sint16	Signed Integer	16	—
8	sint32	Signed Integer	32	—
9	sint64	Signed Integer	64	—
10	float32	Floating point number	32	IEEE 754 binary32 (Single Precision)
11	float64	Floating point number	64	IEEE 754 binary64 (Double Precision)

4.4.2 動的なシリアライゼーション

SOME/IP は動的なデータシリアライゼーションに対応しています。シグナルや PDU は、固定長や可変長のサイズに対応しています。データ構成や長さはランタイムで算出し、必要な領域のみを確保してデータを送信します。

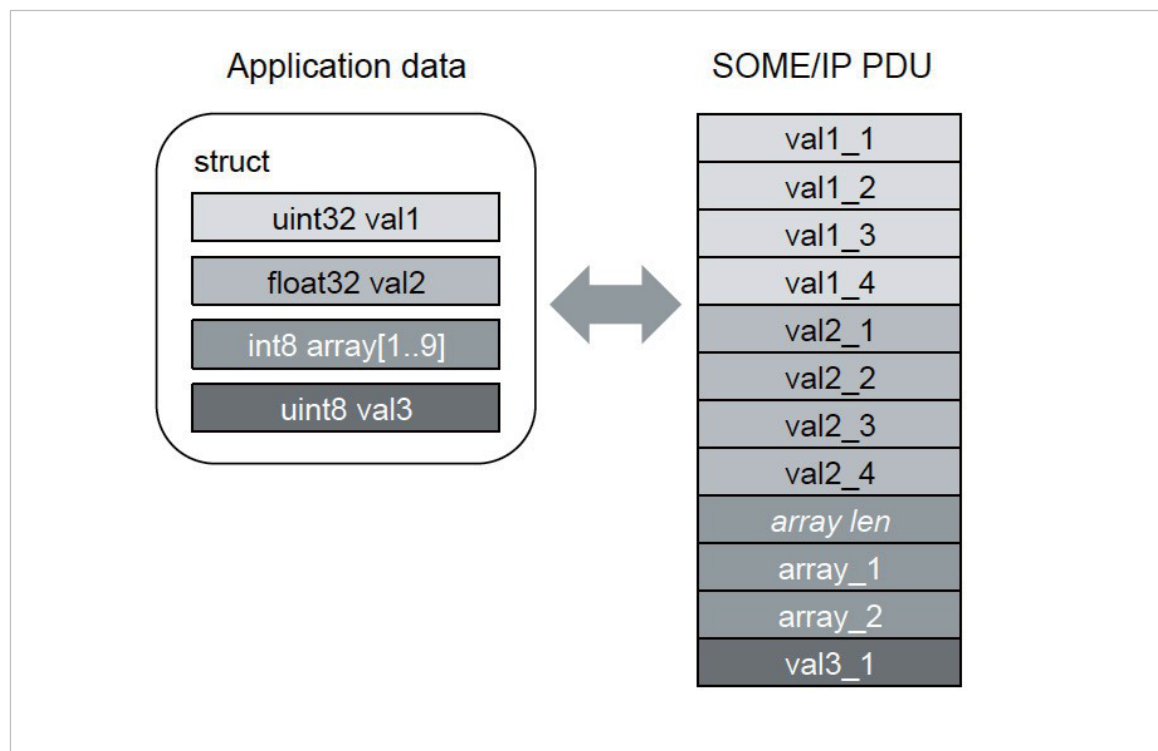


図 12: 動的なシリアライゼーション

5. SOME/IP-TP とは

SOME/IP-TP (Transport Protocol) は SOME/IP の拡張プロトコルです。SOME/IP や SOME/IP-SD で大容量データを送信する場合、SOME/IP-TP を使用します。

ペイロードが大きい SOME/IP メッセージは、ペイロードを分割して SOME/IP-TP メッセージとして送信します。この時、分割した SOME/IP-TP メッセージを「セグメント」と呼びます。

5.1 SOME/IP-TP のヘッダー構造

SOME/IP-TP ではメッセージフォーマットが変化します。Message Type が変化し、Return Code と Payload の間に Offset、RES、M が挿入されます。ヘッダーサイズは 20byte になります。

SOME/IP-TP のメッセージフォーマットを以下に示します。

byte	1	2	3	4	
0	Service ID		Method ID		
4	Length				
8	Client ID		Session ID		
12	Protocol Version	Interface Version	Message Type	Return Code	
16	offset			RES	M
20	Payload				
..					

SOME/IP-TP メッセージフォーマットの各領域を説明します。

> **Message Type**

Message Type の 3bit 目に SOME/IP-TP フラグが含まれます。この SOME/IP-TP フラグで SOME/IP-TP を判断します。

> **Offset**

Return Code の後に、28bit 長の Offset が挿入されます。直前までに送信／受信した SOME/IP-TP セグメントのペイロード長を 16byte 単位で設定します。

たとえば Offset が 92 の場合、これまでに $92 * 16 = 1472$ byte 分のデータを送信／受信しています（ヘッダー長は含みません）。

> **RES(Reserve)**

今後の機能拡張のためのリザーブが 3bit 長挿入されます。すべて 0 を設定します。

> **M(More Segments Flag)**

他のセグメントが存在するかを示す、1bit 長の More Segments Flag が挿入されます。

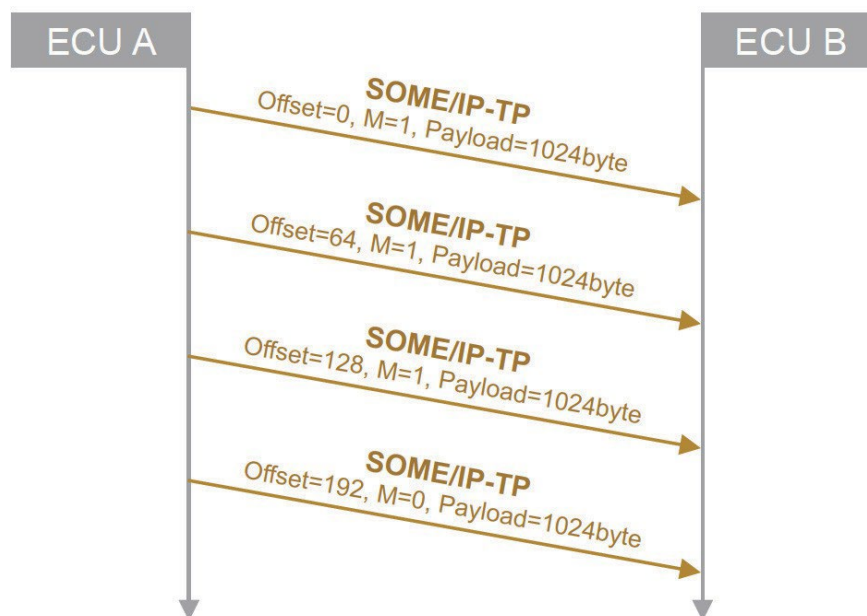
セグメントの最終データである場合は“0”を設定し、他のデータは“1”を設定します。

【NOTE】

SOME/IP-TP の Service ID、Method ID、Length、Client ID、Session ID、Protocol Version、Interface Version、Return Code は SOME/IP と同様のフォーマットです。

5.2 SOME/IP-TP の動作

4096byte のデータを分割し、1024byte のペイロードを持つ SOME/IP-TP セグメント 4 つを送信する場合を考えます。ECU A – ECU B 間において、SOME/IP-TP でセグメント 4 つを送信する場合の例を以下に示します。



最初に Offset = 0、M (More segments flag) = 1、ペイロードサイズを 1024byte とした SOME/IP-TP セグメントを送信します。最初のセグメントのため Offset は 0 となります。未送信データが残っているため M は 1 となります。

次に Offset = 64、M = 1、ペイロードサイズを 1024byte とした SOME/IP-TP セグメントを送信します。この時、1024byte データを送信しているため Offset は $1024 / 16 = 64 (0x40)$ となります。未送信データが残っているため M は 1 となります。

次に Offset = 128、M = 1、ペイロードサイズを 1024byte とした SOME/IP-TP セグメントを送信します。この時、2048byte データを送信しているため Offset は $2048 / 16 = 128 (0x80)$ となります。未送信データが残っているため M は 1 となります。

最後に Offset = 192、M = 0、ペイロードサイズを 1024byte とした SOME/IP-TP セグメントを送信します。この時、3072byte データを送信しているため Offset は $3072 / 16 = 192 (0x120)$ となります。最後のセグメントのため M は 0 となります。

あとがき

ここまで SOME/IP を紹介しましたが、本書で紹介した内容は一部に過ぎません。本書は分かりやすさを重視しており、動作の詳細や例外処理などの説明を省略しています。

SOME/IP の仕様はすべて AUTOSAR から公開されています。SOME/IP の詳細を理解したい方や、最新の仕様を把握したい方は AUTOSAR を参照する事をお勧めします。

本稿で解説した知識が、SOME/IP や車載 Ethernet の開発に携わるエンジニアの方々の一助となれば幸いです。

6. お問い合わせ窓口

技術関連のお問い合わせは、ベクターサポートまでお寄せください。

- ✓ サポートリクエスト(Web サイト経由のサポート問い合わせフォーム)
www.vector.com/jp/ja/support-downloads/support/

【お問い合わせ時のお願い】

お問い合わせの際、サポート開始前に製品のバージョン(例:CANape バージョン xx.x など)、シリアルナンバーなどをお伺いいたします。お手数ではございますが、事前にご確認のうえ、お問い合わせくださいますようお願い申し上げます。

【サポート対応について】

夜間／休日に頂いたお問い合わせは翌営業日の対応となります。また、内容によりご回答までにお時間を頂く場合がございます。なお、不具合が発生した際にご使用されていた関連プログラム一式のコピーを、検証のためご提供いただくようお願いする場合がございます。何卒ご理解ご協力のほど、よろしくお願い申し上げます。

- ✓ お見積もり／保守契約関連のお問い合わせ

営業部 TEL:(東京) 03-4586-1808 E-mail:sales@jp.vector.com



MORE INFORMATION

Visit our website for:

- > News
- > Products
- > Demo software
- > Support
- > Training and workshops
- > Contact addresses

vector.com