

車載用プロトコルスタック： 仕様作成からテスト、量産までの手順



*本資料はTechnica Engineering社Technical Fellowを務めるDr. Lars Völkerの講演「車載用プロトコルスタック：仕様作成からSOP行程の解説」(2020年12月)を元にガイロジック株式会社が執筆を行いました。

目次

1.	<u>はじめに</u>	3
2.	<u>プロトコル スタックのデザイン</u>	3
2.1.	<u>重要なルール</u>	3
2.2.	<u>推奨プロトコル</u>	3
2.3.	<u>プロトコル スタックのサブセット</u>	4
2.4.	<u>セキュリティ</u>	5
2.5.	<u>SOME/IPとシリアル化</u>	6
2.6.	<u>SOME/IP-SDのタイミング</u>	8
2.7.	<u>CAN over Ethernet</u>	8
3.	<u>エンジニアリング要件への提案</u>	9
4.	<u>スタックのテスト</u>	9
5.	<u>データの記録</u>	10
5.1.	<u>通信データ記録時の留意点</u>	10
5.2.	<u>TECMP</u>	12
6.	<u>まとめ</u>	12
図 1.	<u>OSI参照モデルとプロトコル</u>	4
図 2.	<u>データの処理性能と特徴</u>	7
図 3.	<u>SOME/IP-SD タイミング例</u>	8
図 4.	<u>ロガーシステム構成例</u>	11

1. はじめに

このホワイトペーパーで、私が10年以上車載Ethernetに携わる中で色々なお客様から得た知見について解説していきたいと思います。まずプロトコルスタックのデザインに欠かせない情報を整理し、次にエンジニアリング要件についての提案をします。そのあとで、スタックのテストに関する推奨を述べ、最後に通信データの記録に関する参考情報をお伝えします。

2. プロトコルスタックのデザイン

2.1. 重要なルール

まずプロトコルスタックについてですが、過去数年の経験について基づいて得たプロトコルスタックを設計するなら守るべき簡単かつ重要なルールがあります。

第一に「可能な限り標準になっているプロトコルスタックを再利用する」ということです。すでに標準化されているものを再利用することで品質が向上し、テスト・検証が容易になります。第二に「プロトコルの数を制限する」ということです。同じ適用範囲のプロトコルを複数使わず、1つのプロトコルで統一します。人的リソースには限りがあり、仕様書を書く作業やサポート、エラー発生時の対応のために多種のプロトコルの専門家を常に備えておくのは容易ではありません。プロトコルを少なくすることは品質の良い製品を作るのに欠かせないということを意識しておくといよいでしょう。

そして3つ目は「他のOEMが過去に何をしてきたかを学ぶ」ということです。他のOEMで使用されているということは、それらはすでに他の誰かがテストしているということです。これらを意識しておくことでより早く量産を開始できます。

2.2. 推奨プロトコル

次にプロトコルの推奨事項を、OSI参照モデルを用いて解説します。図1：OSI参照モデルとプロトコルは7つの層からなる通常のOSI参照モデルにプロトコルスタックを図示しています。一番下の1層では、いくつかのプロトコルを実現するための異なる物理規格に関連するプロトコルが書かれています。

100Mbpsの100BASE-TXプロトコルはOBDでも使用されます。ECU内のプログラムやデータを高速で書き換えたい場合、100BASE-TXを応用しますが、その時は100Mbps対応のOBDコネクタも必要になります。100BASE-T1はIEEE 802.3で規定された標準規格で100BASE-TXと同じプロトコルを用いることができます。IEEE 802.3には1000BASE-T1もあり、これらの100/1000BASE-T1はツイストペアのケーブルで車載グレードの通信を可能にします。Multi-Gigに関しては、標準化作業はほぼ完了しており、10Gbpsを出すことができます。

2層では通常のMACレイヤーが使用できますし、VLANを用いることで仮想ネットワークを利用できるようになります。これもIEEE 802.1qで標準化されています。またIEEE 802.1Qavに対応させると、遅延が許されない非常に重要な制御データを扱うことも可能です。

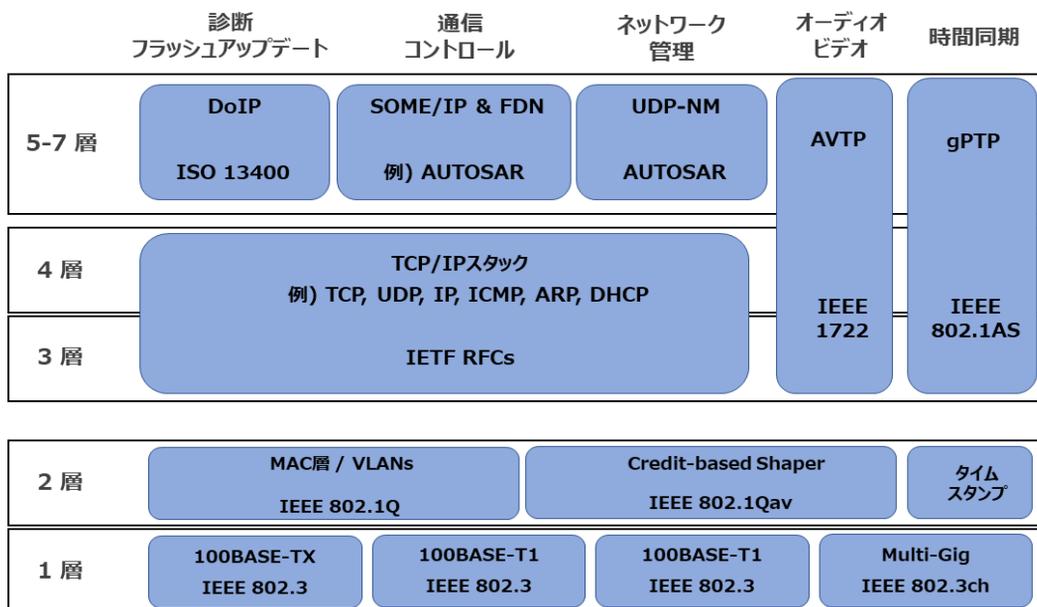


図1. OSI参照モデルとプロトコル

もう1つの特徴はタイムスタンプです。このサブセットとして、gPTPと呼ばれるIEEE 802.1ASとして標準化された時間同期のプロトコルがあります。これは自動車業界でもよく知られ、実際にOEMで採用されています。

3層以上では、車内でオーディオやビデオをストリーミングするために必要なAVTP (Audio Video Transport Protocol) があり、IEEE 1722として標準化されています。

3層、4層は基本的にはLinuxやWindowsなどのすべてのオペレーティングシステムの典型的なソフトウェアベースのプロトコルのようなものです。そしてこの図ではTCP/IPスタックがあり基本的にはIPv4がベースになります。IPv4は少し古く聞こえるかもしれませんが、実装の品質はもちろん、長年の実績により堅牢になっています。ここで使われるプロトコルには、例えばTCP、UDP、IP、ICMP、ARP、DHCPなどがあります。これらはIETF (Internet Engineering Task Force) で標準化されていますので、各プロトコルのRFC (Request For Comments)を読むことをお勧めします。

5層から7層にはアプリケーションとインターネットの世界ですので、ここで詳細な話を

することはできませんが、3つのプロトコルが書かれています。DoIP (Diagnostics over IP) はIPを用いた診断プロトコルのことでISO 13400で標準化されています。データの伝送やECUの診断、ECUのフラッシュに対応しています。UDP-NM (UDP Network Management)というものがありますが、これはその装置がどのような機能を持っているのか知るためのプロトコルで、例えばスリープ状態に遷移できる機能の有無などを知ることができます。このプロトコルはAUTOSARで指定されており、EthernetではUDP (User Datagram Protocol) と呼ばれるプロトコルを用います。

SOME/IPは自動車のために生まれた、サービス指向のプロトコルで、サービスの制御やデータ通信を行います。これについては「2.5 SOME/IPとシリアル化」以降で解説します。

2.3. プロトコルスタックのサブセット

プロトコルスタックのサブセットについて、2つのトピックがあります。

10Base-T1S

現在、車載Ethernetの新しい規格が発表されていますが、これはより高速ではなく10Mbpsの10BASE-T1Sと呼ばれている10Mbpsで半2重通信を行う規格です。リンクとスイッチで構成されるEthernetのネットワークと異なり、複数のECUが1つのバスに接続して、ラウンドロビンのような調停を行うネットワーク構成です。すべてのユースケースに最適なソリューションではないかもしれませんが、特にローエンドでは既存のネットワークとのギャップを縮め、CANなどの置き換えに適しています。

TSN

もう1つはTSN (Time-Sensitive Networking) です。このTSNの中にもいくつかのトピックがあります。

□ 802.1CBは信頼性を向上させるためのフレームの複製と排除に関するものです。同じフレームを複数のリンク上に転送し、複数の経路をたどって目的のECUに到達できるようにします。多くの場合は必要ないかもしれませんが、システムレベルの冗長性確保には重要です。

□ 802.1Qbuというのは興味深いものです。例えば、自分が非常に優先度の高いメッセージを持っていて、直前に別のフレームが送信を開始したケースを想定してください。通常は、先に送信されたフレームが送り終わるまで待つことになり、このフレームが非常に大きなものであれば待ち時間が長くなります。これはヘッドオブラインブロッキング (Head-of-Line Blocking) と呼ばれる問題として知られています。802.1Qbuでは送信中のフレームの特定ス

ロットで後のフレーム送信が可能になり、前のフレームを追い越すことができるようになっていきます。

□ 802.1Qbvはスケジュールされたトラフィックのための拡張です。特定のフレームが特定のタイムスロットで送信される場合に有効です。例えば、複数のスイッチが同期している時間を制御し、最大の遅延計算等を行うことができます。

□ 802.1Qch、これはそのネットワークのトポロジーに関係なく時間確定の計算簡単に行い、フレーム到達までの時間確定を助けるものです。

□ 802.1QciはTSN機能の拡張で最も重要なものだと思います。これは受信ポートで監視を行い、不要なトラフィックの流入を防ぐものです。これによりネットワークスイッチなどが不正な攻撃を受けて不要な大量のデータ転送を行わないようにすることができます。ネットワークの世界ではすでに多くのスイッチが搭載しています。

すべてのTSN機能において一番重要なものはユースケースに大きく依存します。これは業界全体で大きな議論になっています。今後どの機能が採用されるのか、新しい自動車用の仕様やプロファイル策定が行われるのかは今後も議論が続くでしょう。

2.4. セキュリティ

プロトコルスタックに関する今の大きな話題はセキュリティです。プロトコルスタックに入れたいセキュリティ機能や、セキュリティ機能に関連した機能もあるでしょう。

ここで最も重要なのはネットワークセキュリティだと考えます。メッセージを操作されないようにしたい、盗聴されないようにしたい、誰からもメッセージの中身を見られない、と安心できるようにする必要があります。

推奨するネットワークセキュリティ機能にIEEE 802.1AEとして標準化されているMACsecがあります。これは標準として加えたいと思います。推奨する理由は単純で、MACsecはユニキャスト、マルチキャスト、ブロードキャストのすべてのメッセージをラインスピードで保護することができるからです。これによりネットワーク全体に非常に強力なセキュリティ基盤をもつことが簡単に実現できます。

さらにバックエンドの接続にはSSL/TSL、IPsecとVPNプロトコルも推奨します。自動車とインフラ、IPサーバーとの間のチャンネルも保護する必要があるからです。もちろん低速のバスシステムを含み、時にはEthernetメッセージも保護することができます。オンボード通信の標準セキュリティ規格SecOCはメッセージを改ざんから保護する機能を提供しています。

その次にスタックに追加すべきセキュリティ関連の機能はアクセス制御です。特にEthernetスイッチに接続しているデバイスにはMACsecと連携している802.1Xをベースにしたアクセス制御を実装すべきです。SOME/IPでは一部のポリシーを執行するために必要となります。サービスを提供する相手は常にチェックされていることに留意する必要があります。ECUは特定の相手と特定のサービスしか接続できませんので、ユーザーがPCを用いてECUに接続しても、そのサービスに接続可能とは限りません。これは、割とよくある問題です。

あとはファイヤウォール、パケットフィルタリングというEthernetのネットワークでは一般的

な機能があります。一般的には3層、4層で利用されていますが、適切な半導体を使用していれば2層でも可能です。スイッチにとっても非常に強力な重要な機能ですので、機能設定だけでなく十二分に検討すべきものです。

最後にプロトコルスタックでは見られない侵入検知機能IDS (Intrusion Detection System) があります。このためにはスケーラブルな分散ソリューションが必要になります。侵入検知データをネットワークに転送し、その情報を共有することでネットワーク全体を保護したいと思うでしょう。

2.5. SOME/IPとシリアル化

SOME/IPは自動車業界でもっとも使われているEthernet関連のプロトコルで、私自身がほぼすべての仕様を作成しました。なぜOEMはEthernetによるデータ通信のためにSOME/IPを使用するのでしょうか。基本的にSOME/IPは車載アプリケーションに必要な抽象化を行うからです。SOME/IPは自動車向けに作成され、非常に小さな組み込みデバイスから高性能ECUまでカバーできるようにしています。もちろん車載用のあらゆるユースケースをサポートしています。また、サービスディスカバリーを内蔵しているのも非常に重要な点です。これは柔軟性を担保するもので、自動で各ECUの設定が可能です。このように制御された柔軟性をもつことで最適なソリューションを提供できます。そうすることによって優れた自動車のソリューションを得られます。さらにSOME/IPの優れた点としてAUTOSARで標準化している

(AUTOSAR Document Identification No 696)

ことが挙げられます。

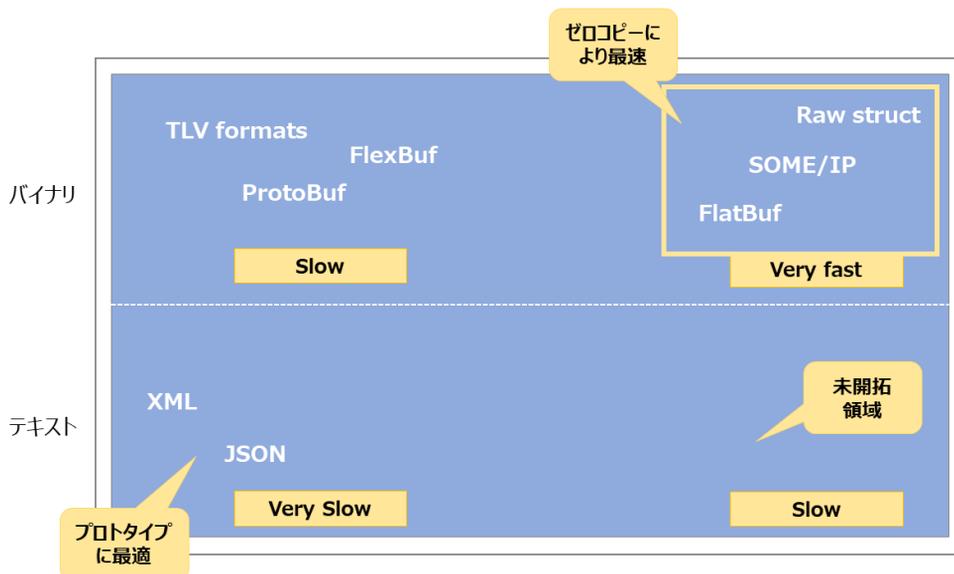


図2. データの処理性能と特徴

また、SOME/IPはAUTOSAR AdaptiveとClassicの両方に対応していますので、デバイスの大小に関わらずあらゆるECUに利用できます。いくつかのIPはライセンスフリーになっていますので、それらを用いれば特許やライセンスの心配もありません。

そしてSOME/IP最大の利点は非常に効率的で高速なデータのシリアル化（直列化）が可能ということです。シリアル化はネットワークを通じてECUがメッセージを取得したり、非常に大きなデータを交換したりする際に重要な要素になります。

「図2 データの処理性能と特徴」はデータ処理の特徴をもとに処理性能の分布を表しています。

通信においてはリンク速度が速くなればなるほど処理しなければいけないメッセージが増えます。ですからシリアル化の処理が、それに追いつく十分な速さであるということは本当に重要です。組込みシステムのリソースは非常に限られていますので、非常に効率的で高速なソリューションが必要とされます。ここでは有名なソリューションをマッピングしました。上の段がバイナリベース、下の段がテキストベース

です。そして左側が自己記述的なソリューションで右側が非記述的なソリューションです。テキストベースのプロトコルとしてXMLやJSONがありますが、これらのものはご存じの方も多いでしょう。これらの特徴は、テキストベースの構造化されたファイルが対象で自己記述的ですので、それを多少知らなくても中身を理解しやすいということです。ここで重要なのは、これらはプロトタイプにとってはとても良いソリューションだということです。しかし、これらの実行には非常に時間がかかります。そんなに早く処理できるものではありません。上段左にあるのが記述的なバイナリベースのもので、ご存じの通り、バイナリ形式の方がテキストよりもよいパフォーマンスを得ることができます。ここでは典型的なバイナリ形式のものが書かれています。そして多くのものがTLV(type-length-value)のような形式を持っています。ProtoBufもTLVに似たアプローチです。これらのプロトコルは一般的に柔軟ですが、そのために少し遅くなっています。これらが遅くなっているのはすべてに自己記述性があるからです。

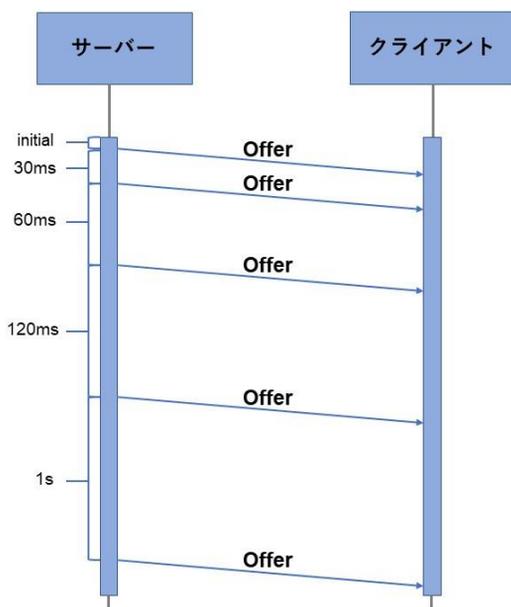


図3. SOME/IP-SDタイミング例

変数	設定値例
Repetition Base Delay	30ミリ秒
Repetitions Max	3ミリ秒
Main Cycle	1秒
TTL	3秒

そして上段右に実現したい非記述的なバイナリプロトコルがあります。最も高速な生の構造体を持ったデータがありますが、これはシリアル化されていません。ECUの中でのみ高速にデータ移動が可能です。SOME/IPは速度をできるだけそこに近づけました。高速ネットワーク通信で利用するために必要な高速シリアル化を実現するためにゼロコピーのアプローチを採用しました。コピーにはIO毎に時間がかかります。性能の最適化のためにSOME/IPではコピー処理を排除しました。

2.6. SOME/IP-SDのタイミング

SOME/IPにはSOME/IP-SDというサービス ディスカバリの機能があります。これは3つのフェーズで成り立っています。

1つ目はInitial wait Phaseで、サービスの準備をしている段階です。送信前に複数のサービスをまとめて1つのビッグメッセージを構築します。2つ目がRepetition (反復) Phaseです。これは高速同期のためのもので、スタートアップを早くしつつ、ネットワーク上で必要なサービスの検索とサービス提供に関する情報交換を行います。

その後がmain phaseでシステムの安定化を図ります。より少ない頻度で通信を行い、システムが安定化しているかを確認するものです。これらの段階で重要なのはタイミングの選択です。タイミングによってスタートアップのパフォーマンス、エラー発生時の挙動が変わってきます。また、その後のデータ分析に関しても、重要なデータが取れるかどうかが決まります。もちろん、経験も非常に大切です。

「図 3 SOME/IP-SD タイミング例」は、タイミングを表す1つの例です。この図は標準に基づいているので、他の人たちのプレゼンテーションを使っています。

2.7. CAN over Ethernet

SOME/IPと並行してCAN、LIN、FlexRay等のメッセージをEthernet上で転送することはよりシンプルで好ましいコミュニケーションになります。こういった既存のメッセージをどのようにゲートウェイで送るのかはアーキテクチャ デザインによって変わってきます。

フレキシブル デジタル ネットワークと呼んでいるTechnica Engineering社のソリューションで採用している簡潔なルールをお伝えします。

1つ目のルールは、できるだけ複雑なゲートウェイ戦略を避けるということです。できるだけシンプルにしておく、これによりパフォーマンスを上げ遅延を下げることができます。メッセージにオペレーションがあればあるほど時間がかかってしまいます。

2つ目のルールは、ソフトウェア ソケットも標準的なもの、例えばAUTOSAR準拠のものを使うことです。解りやすい例で言えば、CANやFlexRayのデータをEthernetでトンネリングして転送します。Ethernetのような大きなIDと長いデータ領域があれば既存のCANやFlexRayを運ぶのには十分です。CAN-FDをEthernetに変換して再びCAN-FDに戻すのに遅延は2ミリ秒未満で実現できます。

3. エンジニアリング要件への提案

エンジニアリング要件に関する典型的な問題は仕様を理解するのが非常に大変だということです。

そこで、エンジニアリング要件に関して3つの提案があります。

1つ目の提案は、仕様ですべてのゴールを記述するのをやめるということです。仕様は単に製品の規格を明記するものですから、この仕様を理解できないと適切なシステムを構築できません。またはバグなしで製品を作れないということになります。仕様に関する最大の問題はこれです。

2つ目の提案は、標準規格を採用するということです。これも仕様書の問題としてよくあることです。OEMはたくさんの仕様をもっていてどんどん長くなってしまふということです。あるジョブの仕様が数千ページになるということも起こりえます。それを理解するのは容易で

はありません。要件が多ければ多いほど中々それを達成できません。標準規格の採用で、こういった問題を減らすことは可能だと思います。例えばAUTOSARベースの仕様を使うのであれば、その動作ができる用にコンフィグレーション可能かどうかを確認する必要があります。また、動作の狙いを説明することも大事です。それは仕様の中に説明を入れるということです。例えばチャプターの冒頭に説明、導入をいれるとか、またはエクセルの表で追加情報を提供するような方法もあります。そしてできるだけ標準規格を採用して標準を守っていくということです。例えば、独自のプロトコルを使って説明するよりも、ここではSOME/IPを使いたい、UDPを使いたいと言う方が簡単です。

そして3つ目の提案は、仕様や規格、システムデザインのプレゼンテーションやトレーニングを用意するという事です。Tier1やTier2がよりよく理解するためのものですが、これができれば最終的により高品質の製品につながります。

4. スタックのテスト

次にスタックのテストについて解説します。多くのOEMは十分にスタックをテストしていません。Ethernetは自動車業界にとって新しく複雑なものなので、これは理解できます。たくさん標準、たくさんの要件があっても見えないバグもあります。Ethernetテストの戦略は非常に難しいということになります。もちろんテストにはエキスパートが必要ですが、十分なエキスパートは存在しないわけです。このような状況でよくある間違いはなにかというとTier1がスタックテストをやってくれるとあって、すべてを任せてしまうということです。

Tier1がたとえEthernetのことを熟知していたとしても、OEMの要件を理解していない、何をつくりたいか解らないといった状況で、どうすれば実装上の問題をみつけることができるのでしょうか？多分無理だと思います。

私たちはサードパーティのツールを使って、より安定したテストを行うことで製品を見直すよう推奨しています。

次にEthernet ECUの統合テストについて提案します。通信スタックは各ECUに分散しています。実際には複数のスタックを持つECUや複数のECUがあるので、1つのECUテストでは検出できない問題があります。どのように対処すべきでしょうか。

すべてのECUを自動車に搭載してはじめて問題を発見するというのでは少し遅すぎます。私たちはもっと早くEthernet ECUの統合テストを実施することを推奨しています。Ethernetのネットワーク統合をできるだけ早くやるということです。ECUを自動車に搭載するまで待つのではなく、できるだけ早く統合テストを行ってデータを記録し、分析することで問題は発見されます。例えばスタートアップやシャットダウンの問題が見つかった場合、自動車に搭載してから見つかるよりも遥かにいいわけです。

統合テストを早期に行うことでEthernetユニットがより速く安定化し、それが製品全体の品質を高めます。

5. データの記録

5.1. 通信データ記録時の留意点

おわかりかと思いますがEthernet通信を記録するのは、CANと比べるとはるかに難しくなります。もちろんCANはより遅いわけですが、それだけの違いではありません。重要なのはEthernetの場合、データは移動するリンクにだ

けしか流れないということです。すべてのパケットをみるにはすべてのリンクを記録する必要があるため、データの量が多くなります。また、IEEE 802.1AS / PTPの protocols を使った場合は、タイムセンシティブであるということも非常に重要です。

データ記録装置のセットアップに注意しないと干渉が起こってしまい、全体のシステムが使えなくなってしまう。もちろんロギングの問題はEthernetだけではなくありません。お客様がCAN-FDを使用する場合、トポロジーも考える必要があります。単に自動車にロガーを付けて、EthernetもCAN-FDもすべて記録しようとする線長が長くなりすぎます。Ethernet、CAN-FDの問題を解決するにはEthernet用に設計されたロガー（記録装置）とデータの収集を行う装置を分けることです。必要に応じて分散されたデータ収集装置が各プロトコルのデータを集め、それらを他の場所にあるロガーに転送するということです。

DLT (Diagnostics Log and Protocol) やXCP (Universal Calibration Protocol) といった自動車用のプロトコルがあります。これらはECUの内部データを取り出すことができます。この機能がロガーにあるとバグを早く見つけることができますが、残念なことにEthernetでデータ転送を行うと、当然車載ネットワークを通ることになるので、ネットワークに干渉を起こします。特にタイミングが重要なプロトコルである802.1ASはセンシティブです。ロガーをネットワークに追加したことで時差が生じると、データインジェクションというコントロールメッセージが送られます。ユースケースとして、ロガーの有無にかかわらずすべての装置がうまく機能するように調整、確認する必要があります。特にリンクの品質とタイムスタンプは重要です。

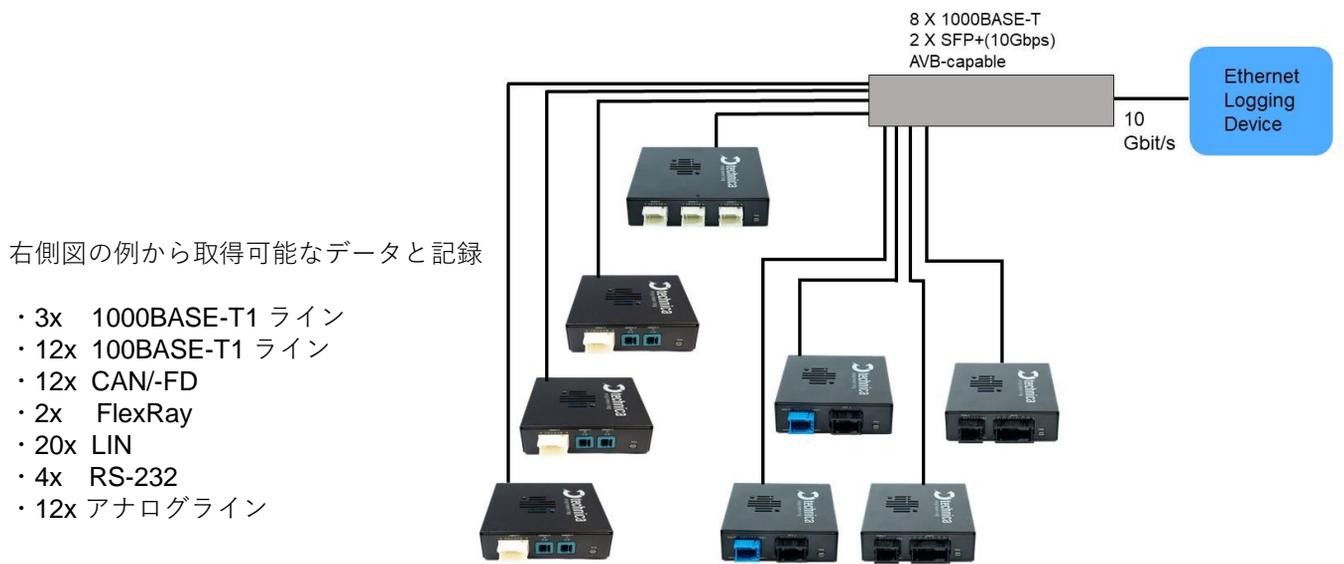


図4. ロガーシステム構成例

問題を追跡したいけれどもデータがうまく記録されずに問題を見つけられないとお客様からよく聞きます。**Ethernet**にはリンクとそのリンクを流れるデータの方向があります。スイッチ間を移動するメッセージにどれくらいの時間がかかるのか、どれくらい1つのスイッチにとどまるのか、**CAN**メッセージが**Ethernet**にルーティングされる時間はどれくらいか、そしてゲートウェイがしっかり機能しているかを知りたい。そういったことを実現するためには各リンクでデータを収集し、そのリンクの場所、データの流れる方向も一緒に記録する必要があります。さらに、**Ethernet**関連の情報だけでなく、正確なタイムスタンプも必要になります。ナノ秒レベルの精度を持ってすべてのデータを同期させる必要があります。すべてのネットワーク上のモジュールが同期したスタンプをもつことが重要です。

リンク障害についてお客様から興味深い話を聞いたことがあります。メッセージロスが発生していたが、それが長い間発見できなかった事があったというのです。例えば**Ethernet**リンクが不安定である場合、原因は**ECU**だけでなく、記録装置やケーブル、コネクタが悪いということがあります。

そしてこのお客様のケースでは、まれにリンクが落ちることがあっても、すぐにリンクアップしていたというわけです。リンクダウンの時間が非常に短時間であったため、周りへの影響が小さく見落としていたというのです。そして問題の発見が遅れてしまったもう1つの原因は、すべてのデータを記録していなかったことです。**Ethernet**のポートのリンクアップ、リンクダウンを含むリンク品質を同時に記録したすべてのリンクのデータがあれば、自動分析をかけて問題を早期に発見できたというのです。それ以降は、テストカーによるフリートテストでも、ポート情報を含む全リンクのデータを記録し、自動分析をかけるようにしたそうです。その結果、問題の早期発見ができ、最終製品である自動車の品質向上に著しく貢献できたと聞きました。

もちろん、**ECU**やテスト装置自体の問題もあり得ます。次に具体的な例を挙げて理解を深めて頂きたいと思います。

「図4 ロガーシステム構成例」に書かれているのは8つのキャプチャーモジュール、**AVB**対応のスイッチとデータロガーです。この構成で以下のラインのデータ取得と記録が可能です。

1000BASE-T1 x3
100BASE-T1 x12
CAN/CAN-FD x12
FlexRay x2
LIN x 20
RS232 x4
アナログ ライン x12

これはあくまでサンプルのセットアップです。一番上のスイッチでキャプチャーモジュールのデータを統合します。そしてそれを1つの10GBのリンクでデータロガーに送ります。PCだとそれほど速く読み込めないかもしれませんが、専用のロガーであれば10GBを問題なく利用できます。

さらに、すべてのキャプチャーモジュールは同期させることが可能で、データの送信にTECMPを使用できます。

5.2. TECMP

ここで少しTECMPについて説明します。

TECMPはTechnically Enhanced Capture Module Protocolのことで、フリーでオープンなプロトコルです。収集された元のデータにリンクアップ、リンクダウンを含むリンク品質やタイムスタンプ情報を含みます。また、収集される元データはEthernetに限らず、CAN、CAN-FD、LIN、FlexRayやアナログ値などにも対応しています。TECMPはオープンソースの実装もあり、2020年10月にアップデートされたWireshark 3.4.0以降ではTECMPが標準でサポートされています。

6.まとめ

□ プロトコル スタックデザイン
スタンダードを再利用し、絞り込むということです。多くのプロトコルを使いすぎではいけません。

□ エンジニアリング要件
Tier1 / Tier2が理解でき、よりよい製品が作られるようにすることが大切です。

□ テスト
サードパーティのテストツールを活用し、統合テストを早期に実現しましょう。

□ データの記録
データの収集装置と記録装置を分けて、柔軟な構成に対応できるようにしましょう。元の通信データだけでなく、リンクの状態や同期したタイムスタンプも非常に重要です。

そして、高品質な製品の量産達成をより速く実現してください。

資料・ Technica Engineering社製品に関するお問い合わせ

日本正規代理店: ガイロジック株式会社
〒180-0004 東京都武蔵野市吉祥寺本町 2-5-11 松栄ビル5階
Tel 0422-26-8211 Fax 0422-26-8212
<https://www.gailogic.co.jp/>
sales@gailogic.co.jp