

コンシューマ・システム論文

自動運転向け車内ネットワークシステムにおける データ伝送方式の開発

木谷 光博^{1,a)} 片岡 幹雄¹ 寺岡 秀敏¹

受付日 2016年2月28日, 採録日 2016年7月7日

概要: 自動運転を実現するために車載ネットワークの広帯域化が求められており, 自動車への Ethernet の適用が検討されている. 車載 Ethernet の導入にあたっては, Ethernet と既存の CAN ベースの車内ネットワークを接続するセントラルゲートウェイ (CGW) と呼ばれるゲートウェイ機能が重要になる. 本研究では, CGW を導入した自動運転向け車載ネットワークアーキテクチャにおける課題を明確化する. 次に, 明確化した課題に基づいて, CAN と Ethernet 間をまたがったデータ通信経路制御方式, およびセンサデータの時間同期方式を提案する. 最後に, これらの方式を組み込んだ CGW 試作機を開発し, 方式の動作および有効性を評価することで, 自動運転システムへの適用を検証する.

キーワード: 車内ネットワーク, 車載イーサネット, セントラルゲートウェイ, 自動運転

Development of Data Communication Method In-vehicle Network for Automated Driving

MITSUHIRO KITANI^{1,a)} MIKIO KATAOKA¹ HIDETOSHI TERAOKA¹

Received: February 28, 2016, Accepted: July 7, 2016

Abstract: In recent years, the data traffic of in-vehicle network is increasing. In order to solve the problem, the introduction of automotive Ethernet has been studied. In the network architecture with Central Gateway (CGW) for interconnecting Ethernet and existing CAN based network, the challenges related to communication for automated driving are studied. Based on the studied challenges, the data communication path control method and the time synchronization method of a plurality of sensor data are proposed across between CAN and Ethernet network. CGW prototype which is implemented these methods are developed. These methods are evaluated the operation and effectiveness to verify the application of the automated driving system.

Keywords: in-vehicle network, automotive ethernet, central gateway, automated driving

1. はじめに

近年, ドライバの安全性・快適性向上, 環境負荷低減を目的として, 安全運転支援システム (ADAS: Advanced Driving Assist System) が広く普及し, ADAS を高度化した自動運転に関する技術開発が活発化している.

自動運転では, ドライバの代わりにシステムが走行環境認識および危険判断を行うため, カメラやレーダ等の各種センサが多数搭載される. このため, 自動運転システムにおいては通信量が飛躍的に増大し, センサや電子制御装置 (ECU: Electronic Control Unit) 間を接続する一部の伝送路は広帯域化が必要とされる. 本要求への対応として, 車載 Ethernet の導入が検討されており, 一部の車種では商用化されている [1]. また, 自動運転システムでは, センサからの情報を基に車両の走行軌道・動作制御情報を生成し, アクチュエータへ指令を出す, という一連の処理を複数の ECU が協調・連携して行う必要があるため, ネット

¹ 株式会社日立製作所研究開発グループシステムイノベーションセンター

Hitachi Ltd. Research & Development Group, Center for Technology Innovation - System Engineering, Yokohama, Kanagawa 244-0817, Japan

a) mitsuhiro.kitani.jd@hitachi.com

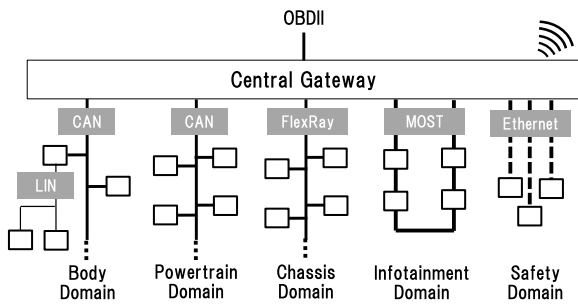


図 1 CGW 型ネットワークアーキテクチャ
Fig. 1 CGW network architecture.

トワークシステムがより複雑化していく。これに対応するため、「パワートレイン」「シャシー」, 「ボディ」, 「情報」, 「安全」といった、より密接な連携が必要となる機能ドメインの単位ごとに CAN (Controller Area Network) 等のサブ・ネットワークを構成し、セントラルゲートウェイ (CGW: Central GateWay) と呼ばれるゲートウェイ (GW: GateWay) を介してサブ・ネットワーク間を相互に接続する CGW 型ネットワークアーキテクチャが提案されている (図 1) [2].

CAN は、車内 LAN として最も普及しており、様々な機能ドメインで広く使われている。通信速度は 500 [kbps], 250 [kbps], 125 [kbps], 83.3 [kbps], 33.3 [kbps] 等が普及しており、ネットワークトポロジはバス型で、いずれのノード (ECU) からも自由に通信を開始できるマルチマスタ方式である。複数のノードで同時に通信開始イベントが発生した場合、各ノードの信号を用いて調停を行い通信の衝突を回避する。CAN の ECU 間で共有すべき情報は、各 ECU から周期的にバスへ送信され、バス上の ECU はデータ受信後に受信要否を判定するフィルタ処理を行うことで、情報のやりとりを行う仕様が一般的である。CAN メッセージのペイロード (データフィールド) は 0~8 [Byte] である。

また、近年急速に自動車のセキュリティ強化の重要性が高まっており、そのために CGW を導入することが検討されている [3]. ここでは、CGW はセキュリティゲートウェイとして、車両外部 (OBD (On-Board Diagnostics) インタフェースや無線通信モジュール) からのアクセスと車両内ネットワークを分離し、車両内の自動運転等に関連する各種機能を保護する役割を担う。

そこで本研究では、CGW アーキテクチャにおいて自動運転を実現するためのデータ伝送方式の検討を行い、システム試作により提案方式の評価を行う。

2. 先行研究

車載ネットワークの GW で使用される CAN から Ethernet へのプロトコル変換アルゴリズムが提案されている [4]. 提案アルゴリズムでは、複数の CAN メッセージを任意の大きさのバッファやタイマ値によってバッファリングした

後、1つの Ethernet パケットとしてまとめて送信する方式や、優先度の高い CAN メッセージは、GW 内で待機せずに Ethernet へ変換して送出する方式の検討をしている。これらの方式については、プロトコルヘッダのオーバーヘッド、メッセージの平均遅延、GW の CPU 負荷率の各項目が FPGA ベースの GW 試作機で評価され、ネットワーク要件に応じて適切なアルゴリズムを選択する必要があると結論付けられている。また、定期送信される CAN メッセージの送信周期をデッドラインとして GW 内での CAN メッセージの待機時間を調整する別の方式も提案されている [5]. 本提案方式は、OMNeT++ [6] ベースのシミュレーション環境で評価され、デッドラインの制約を満たしつつ、従来手法に比べてネットワーク負荷を低減できるとまとめている。ほかにも、工場等に用いられる産業分野のネットワークをターゲットとし、フィールドバス (CAN) と Ethernet を相互接続する GW におけるプロトコル変換方式が提案されている。具体的には、Ethernet から CAN へのプロトコル変換時に、CAN バス上のトラフィックを平滑化する目的で、CAN メッセージを GW 内でバッファリングする方式が提案され、シミュレーション評価によって、トラフィック平準化を達成できる一方で、最大遅延が大きくなる場合があると結論付けられている [7].

これらの研究は、いずれも複数の CAN バスと単一の Ethernet 機器と接続された GW におけるプロトコル変換アルゴリズムに関するものである。すなわち、CAN メッセージから Ethernet パケットへ変換する際の CAN メッセージの多重化と、Ethernet パケットから CAN メッセージへ変換する際の Ethernet パケットの多重分離化方式が提案され、シミュレーションや FPGA による評価が行われている。一方で自動運転システムは複数チャンネルの CAN バスと、複数の Ethernet 機器で構成されるため、前述のパケット変換方式に加えて、アドレッシングや伝送フォーマット等のシステム全体を考慮したネットワーク設計や経路制御方式が重要となる。

そこで本研究では、これらの先行研究をふまえて、自動運転対応の CGW 型ネットワークシステムにおいて、ネットワーク設計と CGW でのデータ通信経路制御方式の検討を行い、GW 用途を想定して開発された車載マイコンを用いたシステムの試作評価によって方式の有効性を確認した。

3. 自動運転ネットワーク設計と課題

本研究で対象とする自動運転対応の車載ネットワークシステムを図 2 に示す。対象システムは、通信帯域の増大が必要となる自動運転関連の安全系ドメインをデータ通信速度が 100 [Mbps] の車載 Ethernet (以下、Ethernet) で構成し、自動運転以外の基本機能は従来の CAN (データ通信速度は 500 [kbps]) で構成する。安全系ドメインは、前方の状況を取得するステレオカメラ ECU (SC), 周囲の状況

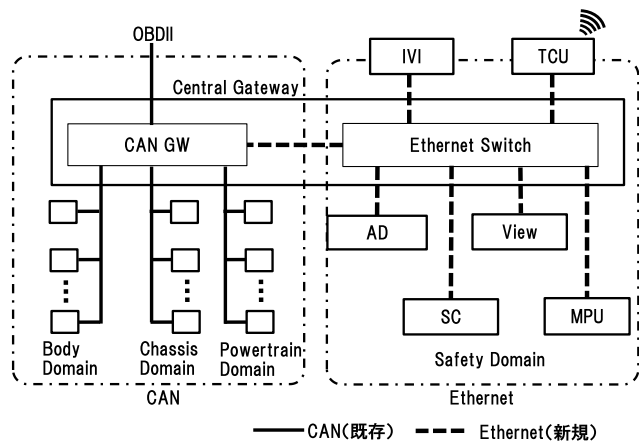


図 2 研究対象システム
Fig. 2 Study of the system.

を取得するビューカメラ ECU (View), 自車位置の推定と自動運転用高精度地図の管理を行う ECU (MPU), SC や View, MPU からの情報を基に自動運転システム全体を制御する自動運転 ECU (AD), 車載通信モジュール (TCU), 車載情報通信システム (IVI) から構成する. これらの ECU 群は CGW の Ethernet スイッチを介して接続され, ネットワーク内の ECU 間で通信する場合は, Ethernet スイッチがパケットの中継処理を行う. Ethernet で送受信されるデータの例として, 各カメラ ECU から自動運転 ECU (AD) に送信される外界の認識情報や高精度ロケータ/地図から自動運転 ECU に送信される地図情報がある. 一方, CAN はバスごとに CGW 内の CAN GW に接続され, バスをまたいでデータ通信する場合は CAN GW でメッセージのルーティング処理を行う.

Ethernet と CAN をまたがった通信は, CGW の CAN GW でデータ通信経路制御とプロトコル変換を行う. CAN から Ethernet に送信されるデータの例として, 高精度ロケータ等が位置情報の推定に利用する速度情報がある. また, Ethernet から CAN に送信されるデータの例として, 制動命令やステア舵角等の運動制御指示がある.

以下, 本章では, 本システムのネットワーク設計の検討と自動運転を実現するうえでの課題を述べる.

3.1 自動運転ネットワーク設計

ネットワーク設計のため, Ethernet 内のデータ伝送フォーマットを検討する. 検討の観点としては, 経路制御機能, 優先制御機能, 伝送性能, 柔軟性の 4 点を考慮した. 伝送フォーマットとして考えられる Ethernet フレームと IP パケットの 2 つを比較した結果を表 1 に示す.

経路制御については, Ethernet パケットの場合は VLAN ID か MAC アドレスを, IP パケットの場合は IP アドレスを用いる方法が考えられ, いずれの方式も目的を達成することができる. CAN との相互通信を考慮した場合でも, 両方式ともに経路制御用の識別情報として使用できる. 優先

表 1 伝送フォーマットの比較

Table 1 Comparison of the transmission format.

| 比較項目 | Ethernet ベース | IP ベース |
|------|------------------------|----------------|
| 経路制御 | ○ VLAN ID, MAC アドレス | ○ IP アドレス |
| 優先制御 | ○ VLANTag(Priority) | ○ IP TOS |
| 伝送性能 | ○ | ○ |
| 柔軟性 | × | ○ UDP/TCP 等 |

制御は, Ethernet パケットの場合は, VLAN タグ内のプライオリティ情報を, IP パケットの場合は, IP ヘッダのサービスタイプ (TOS : Type Of Service) フィールド情報を用いて同等の優先度制御を実現できる. 伝送性能については, ペイロードの実効通信帯域を比較検討する. 本システムにおける Ethernet の通信データ量は, Ethernet 内を送受信する外界の認識情報や地図情報等の各データの通信仕様から机上評価したところ合計 10 [Mbps] 以下になる見通しである. そのため 10 [Mbps] を帯域要件として定義する. 外界の認識情報はカメラ画像データではなく, 各カメラ ECU で画像認識処理によって抽出されたオブジェクト情報のみを Ethernet で送受信することを想定する. Ethernet の実効通信帯域 (理論値) は, パケット長を 64~1518 [Byte] とすると, Ethernet パケットでは 54~97 [Mbps], IP パケット (UDP) では 21~95 [Mbps] となる. よって, Ethernet と IP の両パケットともに, 帯域要件の 10 [Mbps] を満足する見通しである. 柔軟性に関しては, 伝送するデータの通信要件から検討を行った. 自動運転システムでは, センサからの周辺認識情報等のリアルタイム性を要求するものや, 自動運転用の高精度地図情報等のリアルタイム性は要求されないがデータの欠落が許容されないものがある等, 様々な通信要件に対応する必要がある. これに対応するためには, Ethernet 上にそれぞれの要件に対応した仕組みを独自開発するよりも, 民生分野を中心に広く普及している UDP/TCP を適用するほうが効率的であると考えられる. さらに, 自動運転システムでは, 無線を用いた ECU ソフトウェア更新による機能拡張や, 複数の車両間における高精度地図作成のための情報共有等のサービスが検討されており, これらの車外システムとの連携も考慮する必要がある. この観点でも IP ベースの伝送フォーマットの方が適用性に優れると考える.

以上の検討結果から, 柔軟性の観点で優位である IP パケットを Ethernet のデータ伝送フォーマットとして用いる. 本システムの各 Ethernet I/F に対して図 3 のように IP アドレス割当て (アドレッシング) を行い, データ通信を実現する.

Ethernet におけるデータ通信の実現方法を示す. データを送信する ECU は, データの宛先 ECU の IP アドレスを IP ヘッダの Destination Address フィールドに設定して

IP パケットを送信する。Ethernet スイッチは、パケット受信後、IP ヘッダの Destination Address フィールドを解釈し、IP アドレス情報に応じてあらかじめ設定したスイッチの Ethernet 出力ポートからパケットを送出する。本処理はレイヤ 3 (L3) スイッチとして動作させることで実現する。データを受信した ECU は、IP パケットヘッダの Destination Address フィールドの IP アドレス情報に基づいて、受信対象のパケットであるかどうかを判定する受信フィルタ処理を通じ、パケット廃棄か受信処理のいずれかを実行する。

3.2 自動運転ネットワークシステムの課題

本ネットワークシステムにおいて、自動運転を実現するための課題を以下に示す。

(1) Ethernet から CAN への通信経路制御方法の確立

Ethernet 内の自動運転 ECU で、センシング情報と自動運転用の高精度地図データと自車両の走行位置情報を基に、ステアリング、ブレーキ、トランスミッション等を操作するための車両運動制御に関わる情報を生成して、CAN へ送信する。よって、Ethernet 内の ECU から CAN 内の ECU へデータを伝送するための通信経路制御方法の確立が必要である。

(2) CAN から Ethernet (複数宛先) への通信経路制御方法の確立

CAN 内の各 ECU 間で共有されている自車両速度や自車両旋回角度情報等の車両制御情報を、ドライバへ提示したり、自動運転制御へフィードバックしたりするため、IVI や自動運転 ECU 等 Ethernet 内の複数の ECU においても共有しなければならない。Ethernet 内の単一 ECU への CAN メッセージの送信は、CAN ID から ECU の IP アドレスへの変換テーブルを用いて伝送する。一方、Ethernet 内の複数 ECU への送信は、CAN メッセージを複製する ECU の候補が複数考えられるため、伝送方法の策定が課

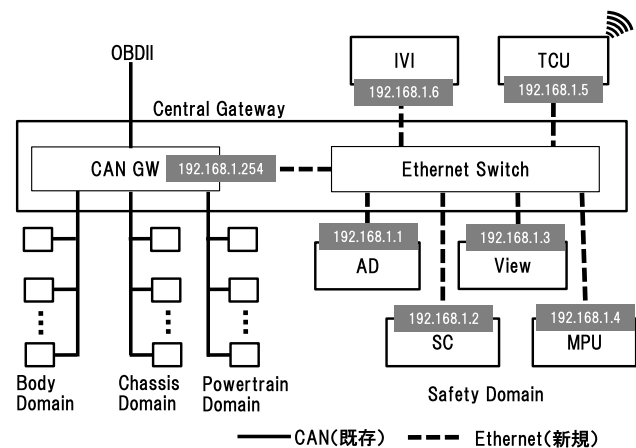


図 3 研究対象システムへの IP アドレスリング
Fig. 3 IP addressing for the target system.

題となる。すなわち、CAN から Ethernet (複数宛先) への通信経路制御方法の確立が必要である。

(3) センサデータの時間同期方式の確立

自動運転 ECU では、Ethernet および CAN 内の複数のセンサからの情報を統合 (センサフュージョン [8]) して判断することでセンシング精度を高めている。センサフュージョン処理では、各種センサ・ECU からの情報を統一の座標系に変換して表現する必要がある。各センサのセンシング周期の相違、通信遅延、送信側 ECU 内での処理遅延等が原因で、自動運転 ECU で処理する入力データの同期がとれていないことがある。その場合、実際の位置と異なる場所に物体を検出する、同一の物体を別のものと判定する等、車両の軌道計画や走行制御に大きな影響を与える可能性がある。そこで、Ethernet と CAN 内の ECU から送信される複数のセンサデータを、時間情報を考慮し同期して処理するための方式の確立が必要である。

4. 課題に対する方式検討

(1) Ethernet から CAN への通信経路制御方法の確立

Ethernet から CAN への通信経路制御の方法として、ECU は送信先に CAN GW の Ethernet I/F の IP アドレスを指定してデータ送信を行い、CAN GW において、CAN メッセージの種別を表す CAN ID と送信すべきバス情報を対応付けたルーティングテーブルを持つことで、IP Datagram 内の CAN ID から通信経路を決定する方法が考えられる (図 4 (a))。

本方法の場合、ルーティングテーブルのエントリ数は、CAN ID 数に依存するため、CAN ID 数が多いほどテーブルに使用するメモリサイズや、テーブルの検索処理時間が増大し問題となる。そこで、ルーティングテーブルのエントリ数を抑制するため、CAN バスごとに IP アドレ

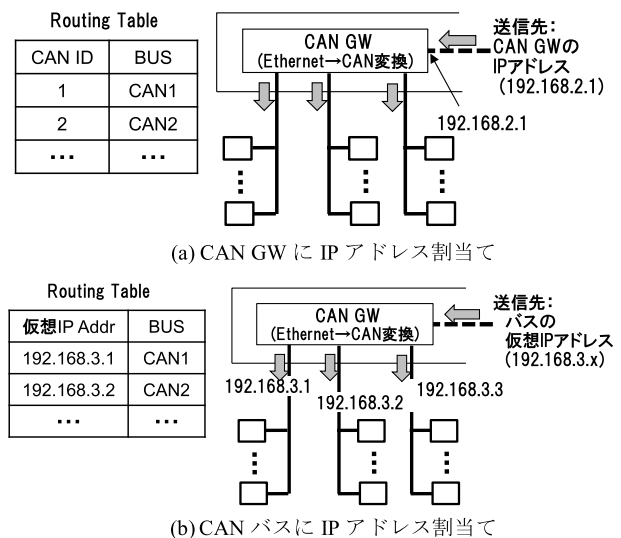


図 4 Ethernet-CAN 経路制御方法
Fig. 4 Ethernet-CAN routing control method.

表 2 CAN へのアドレッシング方法比較

Table 2 Addressing method comparison to the CAN network.

| IP アドレス割当て | 方式(a) CAN GW | 方式(b) CAN バス |
|----------------------------|-------------------|-----------------|
| Ethernet スイッチで 必要なエントリ数 | 1 (CAN GW 数) | 3 (CAN バス数) |
| CAN GW で必要な エントリ数 | 100 (CAN ID 数) | 3 (CAN バス数) |

スを割り当てる方法を提案する (図 4(b)). CAN バスは Ethernet I/F を持たないため, 仮想的に IP アドレス (仮想 IP アドレス) を割り当てる. 送信元 ECU は, CAN バスの仮想 IP アドレスを指定してデータ送信を行い, CAN GW において, バスごとの仮想 IP アドレスと送信すべきバス情報を対応付けたルーティングテーブルを持つことで, IP Datagram 内の CAN ID を用いることなく通信経路を決定することが可能になる.

上述した2つの方法 ((a) CAN GW の IP アドレス割り当て, (b) CAN バスごとの IP アドレス割り当て) について, ルーティングテーブルのエントリ数を比較した結果を表 2 に示す.

Ethernet から CAN への転送が必要な CAN ID 数は 100, CAN GW に接続される CAN バス数は 3 を想定する. Ethernet スイッチのルーティングテーブルのエントリ数を比較すると, 方式 (a) が 1 であるのに対して, 方式 (b) は 3 である. Ethernet スイッチはデバイス内でルーティングテーブル用の記憶領域はあらかじめ静的に確保されているため, Ethernet スイッチについて両方式に優劣はない. CAN GW のルーティングテーブルのエントリ数を比較すると, 方式 (a) が 100 であるのに対して, 方式 (b) は 3 である. CAN GW のルーティングテーブル用のメモリは Ethernet スイッチと異なり, 設計時に動的に確保するため, 方式 (a) に比べて方式 (b) のほうが確保すべきメモリサイズを小さくできる. テーブル探索の処理性能に関しては, CAN GW は CPU で探索処理を行うため, 方式 (a) に比べて方式 (b) のほうが処理時間を小さくできる.

以上, CAN GW に接続されるバスごとに仮想 IP アドレスを割り当てる提案方式は, CAN GW に IP アドレスを用いた経路制御と比べて, ルーティングテーブルに要するメモリサイズとテーブル探索処理時間を小さくしつつ, Ethernet から CAN への通信経路制御を実現する.

(2) CAN から Ethernet (複数宛先) への通信経路制御方法 確立

CAN から複数の宛先へデータ送信する方法として, IP ネットワークにおいて標準化されている Broadcast, Multicast, Unicast の 3 つの通信方法の適用を検討する. 検討の観点として, 設計・実装効率と各 ECU の処理負荷を考慮した. 設計・実装効率については, 初期設計・設計変更時・実装の 3 点を考慮した. また, 処理負荷については,

表 3 CAN から Ethernet (複数宛先) への経路制御方式比較

Table 3 Routing control method comparison from CAN to Ethernet (multiple destination).

| 送信方法 | | 方式 (i) Broadcast | 方式 (ii) Multicast | 方式 (iii) Unicast |
|----------------|------------|---------------------|----------------------|---------------------|
| 設計 実装 効率 | 設計 | ○ | × | △ |
| | 実装 | ○ | △ | × |
| 負荷 | 送信側 ECU/GW | ○ | ○ | × |
| | 受信側 ECU | × | ○ | ○ |
| | ネットワーク | △ | ○ | ○/× |

送信側 ECU, 受信側 ECU, ネットワークの負荷をそれぞれ考慮した. これらの観点で各方式を比較した結果を表 3 に示す.

初期設計を比較した場合, Broadcast 方式は宛先にかかわらず, 送信先に Broadcast アドレスを指定すればよいため設計・実装が容易であり, 送信先が追加削除された場合でも柔軟に対応できる. Multicast 方式は, 宛先に応じた Multicast アドレス設計, および CGW にルーティングの実装が必要となる. Unicast 方式は, 送信側 ECU もしくは CGW において, 複数の送信先に応じたパケットの複製処理の実装が必要となる. 送信先が追加削除された場合, Multicast 方式と Unicast 方式はともに設計・実装の修正が必要となり, Broadcast 方式に比べてその点では劣る. 送信側の負荷については, Broadcast 方式と Multicast 方式は Ethernet スイッチのパケット複製機能を用いてハードウェア処理可能である. Unicast 方式は, 送信側 ECU または GW の CPU において宛先の数だけパケット複製処理が必要になるため, CPU 負荷や処理性能の面で他方式に比べて劣る. 受信側の負荷については, Broadcast 方式の場合, 受信不要なデータを含めすべていったん受信し, そのデータが必要かどうかのフィルタリング処理が必要なため処理負荷が高い. Multicast 方式と Unicast 方式の場合, フィルタリング処理は必須ではない. ネットワーク負荷については, Broadcast 方式は全 Ethernet 伝送路に必ずデータ送信されるため, Ethernet の使用効率が悪い. Multicast 方式は, CAN, Ethernet どちらのネットワークも必要最低限のデータしか流れないため使用効率が良い. Unicast 方式は, 送信側 ECU が複数の宛先に対してデータをそれぞれ複製して送る場合, Broadcast 方式や Multicast 方式に比べて送信側 ECU から CGW の区間で複製されたデータが余計に流れるため使用効率が悪い. 一方, CAN GW でデータを複製して送る場合の使用効率は Multicast 方式と同等である.

以上の比較結果より, 設計が複雑になるが, 各 ECU の処理負荷やネットワーク負荷の観点で最も優れる Multicast 方式を使って, CAN から Ethernet 内の複数 ECU への通信経路制御を行う.

Multicast 方式は, IGMP (Internet Group Management

Protocol) を用いた一般的な IP Multicast を使用せず独自のものをを用いる。車載ネットワークは、通信キャリアやデータセンタ等情報分野のものと異なり、運用開始(車両販売)後の変更を想定せずあらかじめ設計したネットワークを永続的に使用するという特徴がある。本方式はその特徴に基づき、3.1 節で述べた Ethernet でのデータ通信方法によって実現する。データを送信する ECU は、IP ヘッダの Destination Address フィールドに送信先 ECU に応じた Multicast アドレスを設定する。Ethernet スイッチは、IP ヘッダの Destination Address フィールドを解釈し、Multicast アドレス情報に応じてあらかじめ設定したスイッチの Ethernet 出力ポートからパケットを送出する。データを受信した ECU は、IP パケットヘッダの Destination Address フィールドの Multicast アドレス情報に基づいて、受信対象のパケットであるかどうかの受信フィルタ処理を行う。

(3) センサデータの時間同期方法確立

複数のセンサデータを同期する方式として、Ethernet AVB (Audio Video Bridging) がある。AVB は、リアルタイム性が要求される A/V ストリームを伝送ベストエフォート型の Ethernet 上で伝送するためのプロトコルであり、複数の端末間を 1 [μs] 以内の誤差で同期することを保証している [9]。本自動運転システムでは、仮に 2 [ms] の同期ずれが発生したとき、自車両と他車両の速度差が 100 [km/h] である場合の他車両の検知位置のずれは 56 [mm] となる。本検知位置のずれは自車の走行軌道計画へは影響を与えないと想定する。よって、本研究の対象システムでは AVB 相当の高精度な同期は必ずしも必要ないと考え、コストや CPU 処理負荷の面で優位な UDP/IP ベースの簡易な時間同期方式を検討する。

自動運転 ECU でデータの取得時刻を考慮したフュージョン処理を実施するために、センサデータには各 ECU での入力時点での時間情報 (TS : timestamp) を付与する。このためには、付与する時間情報がネットワーク内で同期されている必要がある。一般的にネットワークシステムで時刻を同期する場合、各機器はクロックマスタとなる機器から基準時刻を取得し、それに合わせて自機器の時刻を補正する。本システムにおいては、時間情報を利用してフュージョン処理を行う自動運転 ECU をクロックマスタとして、各 ECU に基準時刻情報 (BT : Base Time パケット) を配信する簡易的な方法を検討する。本方式は簡易的に実現できる一方、(i) 自動運転 ECU から各 ECU までの伝送遅延により基準時刻がずれてしまう可能性があること、(ii) クロック偏差により、基準時刻受信からの経過時間によりずれが大きくなること、といった課題がある。ここでは、以下 2 つの課題に対して解決方法を示す。

(i) BT パケットの伝送遅延による時間同期誤差

自動運転 ECU から BT パケットを受信する ECU まで

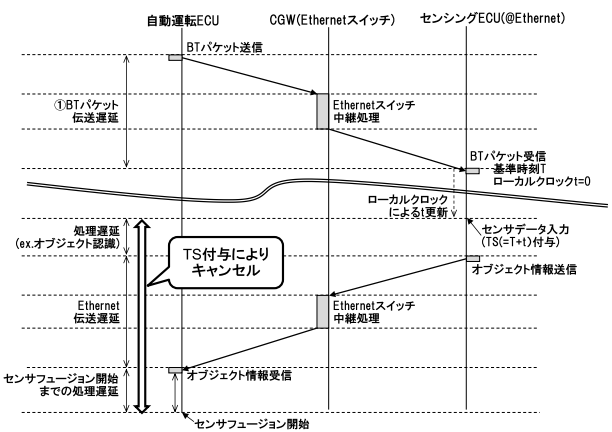


図 5 時間同期方式の処理フロー (Ethernet)

Fig. 5 Processing sequence of time synchronization scheme (Ethernet).

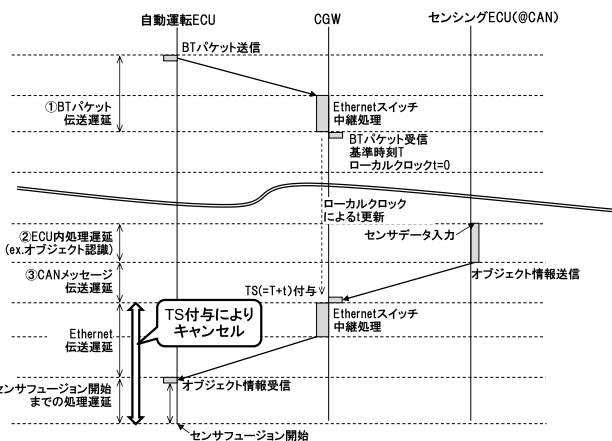


図 6 時間同期方式の処理フロー (CAN)

Fig. 6 Processing sequence of time synchronization scheme (CAN).

の伝送遅延が同期誤差となる。本伝送遅延を最小化するため、BT パケットを唯一の最高優先データとして Ethernet スイッチでキューイング、フォワーディングさせるよう各データの優先度設計を行う。

(ii) TS 算出に用いるクロック偏差による時間同期ずれ

自動運転 ECU と BT パケットを受信する各 ECU のクロック偏差により、時間経過とともに共有した時間情報にずれが生じる場合がある。そこで、BT パケットを周期的に配信することで、時間経過により拡大するクロック精度誤差による同期ずれを補正する。

提案方式の処理シーケンスを図 5、図 6 に示す。Ethernet においては、自動運転 ECU は BT パケットを Ethernet スイッチ経由で同期対象となる複数 ECU へ配信する。それを受信した ECU は自身のローカルな時刻情報を基準時刻へ換算して更新を行う。その後、センサデータパケットにセンシング時点の TS 情報を付与して送信する。自動運転 ECU では TS を基にセンサデータを処理する (図 5)。これにより、センサデータ入力からセンサフュージョン処

理までの処理遅延による時間同期のずれが解消できる。

一方、CAN内の既存ECUでは、修正を加えることなく、BTパケットの受信やセンサデータへのTS情報付与に対応することはできない。そこで、CAN内のECUから送信されるセンサデータはCGW内CAN GWでTS情報を付与してEthernetへ送出することで対応する(図6)。本提案方式によって、Ethernet、およびCANを含めたネットワークシステム全体において、センサデータをセンシングされた時点の情報として処理可能となる。しかしながら、前述のとおりEthernetにおいてはBTパケットの送信遅延(図5④と図6①)、クロック偏差による時間同期のずれが発生する可能性があるため、これを評価する必要がある。さらに、CANにおいては、前記(i)、(ii)の課題に加え、CANバス上のセンシングECU内の処理遅延(図6②)およびCANの伝送遅延(図6③)も同期ずれの要因となりうるため、これを評価する必要がある。

5. 実装と評価

4章で提案したEthernetとCANのネットワークをまたがった通信経路制御方式を実装したCGWを試作し、評価を行った。表4に試作したCGWの主要諸元を、図7に評価システムの概要を示し、以下、評価結果を示す。

(1) EthernetからCANへの通信経路制御方式

CANバスごとに仮想IPアドレスを設定する提案方式(b)によって、EthernetフレームをCANメッセージに変換して所望のCANバスに送信できることを確認した。また、CAN ID数とルーティングテーブルのメモリサイズの関係を図8に示す。また、CAN ID数とテーブル探索時

表4 CGW試作機仕様

Table 4 The Specification of CGW prototype.

| 項目 | 仕様 |
|--------------------|---|
| CPU | 120MHz×2 |
| Memory | RAM:512KB+16MB SRAM, ROM:16MB Flash |
| CAN Interface | 6channel CAN GW |
| Ethernet interface | 6channel Ethernet Switch (BroadR-Reach[10]) |
| OS | AUTOSAR[11]+NonOS |

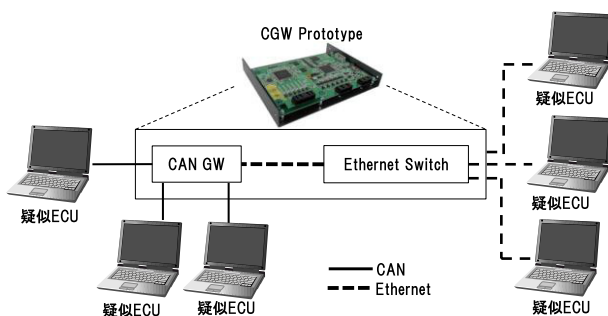


図7 評価システム概要

Fig. 7 Overview of the evaluation system.

間の関係について図9に示す。それぞれについて、(a)と(b)の2つの方式について比較した結果を示した。テーブルの探索は2分探索木を用いた。

テーブル使用メモリサイズは、方式(a)の場合、CAN ID数に対して、線形に増加するが、方式(b)は一定値に抑制することができる。CANメッセージ数=100、CANバス数=3の場合、方式(a)の500[Byte]に対して、方式(b)は15[Byte]に低減できる。テーブル探索処理時間は、方式(a)の場合、CAN ID数に対して増加し、方式(b)は一定値に抑制することができる。CANメッセージ数=100、CANバス数=3の場合、方式(a)の9.8[μs]に対して、方式(b)は2.8[μs]に低減できる。よって、提案方式(b)を用いることで、ルーティングテーブルに要するメモリサイズと探索処理時間を低減できることを確認した。また、

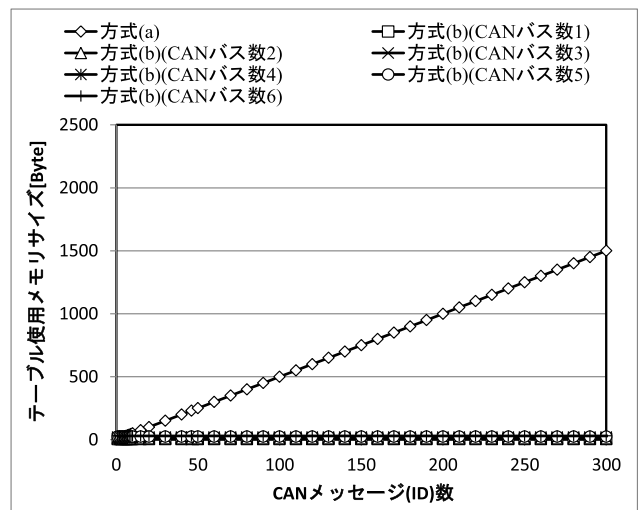


図8 Ethernet-CAN変換テーブル用メモリサイズの比較結果

Fig. 8 The comparison result of the memory size (Ethernet-CAN conversion table).

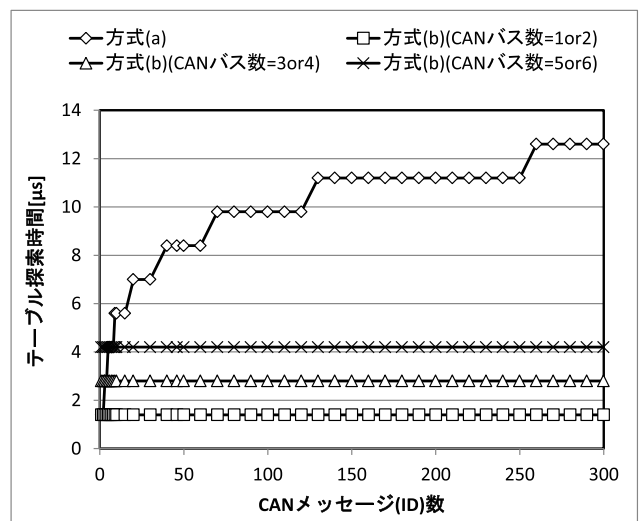


図9 Ethernet-CAN変換テーブル探索時間の比較結果

Fig. 9 The comparison result of the search time (Ethernet-CAN conversion table).

表 5 Multicast アドレス設計 (送信先 ECU ごと)

Table 5 Multicast Address design (Each destination ECU).

| CAN ID | 種別 | データ | Multicast アドレス | 送信先 ECU |
|--------|---------|--------|----------------|------------------------|
| 0x101 | センシング情報 | オブジェクト | 224.0.0.1 | AD, MPU |
| 0x201 | 車両制御情報 | 車速 | 224.0.0.2 | AD, SC, MPU |
| 0x202 | | 舵角 | 224.0.0.3 | AD, SC, MPU, View, IVI |

表 6 Multicast アドレス設計 (データ種別ごと)

Table 6 Multicast address design (Each data Type).

| CAN ID | 種別 | データ | Multicast アドレス | 送信先 ECU |
|--------|---------|--------|----------------|------------------------|
| 0x101 | センシング情報 | オブジェクト | 224.0.0.1 | AD, MPU |
| 0x201 | 車両制御情報 | 車速 | 224.0.0.2 | AD, SC, MPU, View, IVI |
| 0x202 | | 舵角 | | |

Ethernet から CAN へ伝送するための CGW 全体の伝送処理遅延は 66.0~92.2 [μs] であり、車両制御データの伝送要件である 2.5 [ms] [10] を満足する。これにより、本方式を車両制御情報の伝送に使用しても問題ないことを確認でき、Ethernet から CAN への通信経路制御方式を確立した。

(2) CAN から Ethernet (複数宛先) への通信経路制御方法確立

Multicast アドレスに関わる CAN GW のルーティングテーブルの一例を表 5 に示す。本テーブルは、複数 ECU へ送信を行う CAN ID ごとに Multicast アドレスと対応付けられ、その Multicast アドレスは複数の送信先 ECU で構成する宛先グループを一意に指す。この場合、本 CAN ID の数だけエントリが必要となる。また、宛先グループに対する ECU の追加・削除等の設計変更を行う場合、Multicast アドレスが変わることで変更対象以外の ECU 含めた同一宛先グループ内の全 ECU の受信フィルタ処理とルーティングテーブルの修正が必要となり、拡張性の点で課題があることが判明した。そこで、Multicast アドレスが、宛先グループではなく、データ種別を指すようテーブルの変更を行う (表 6)。本変更により、CAN ID の上位ビットがデータ種別を表す仕様の場合は、テーブルのエントリ数を削減することが可能となる。また、宛先グループの設計変更が必要になった場合についても、同一宛先グループ内の設計変更対象外の ECU の修正を不要にできる。表 6 に示すテーブルを CAN GW に、Multicast アドレスと送信先 ECU を対応付けたテーブルを Ethernet スイッチへ実装し、CAN 内の既存の ECU へ修正を加えることなく、CAN から Ethernet の複数 ECU へデータ送信できることを確認した。また、この場合の伝送性能を計測したところ 110.5~158.4 [μs] であった。CAN メッセージにおける最小の周期送信間隔が 10 [ms] であることから、それを CAN から Ethernet ネットワークをまたいだ配信のデッドラインとした場合も問題ないことを確認でき、CAN から Ethernet

表 7 センサデータの時間同期精度評価結果

Table 7 Evaluation result for the time synchronization error of sensor data.

| 区分 | 同期ずれ要因 | 精度誤差 [μs] |
|----------------|----------------|-----------|
| (i) ネットワークシステム | Ethernet 内 | 270~1700 |
| | ①BT パケット伝送遅延 | 20~450 |
| | CAN 内 | 20~150 |
| | ②CAN ECU 内処理遅延 | 250~1250 |
| | ③CAN メッセージ伝送遅延 | 0~1000* |
| (ii) クロック精度誤差 | | 250 |
| | | 0~300** |

*実車の CAN ドメインをキャプチャして取得

**1 秒経過あたりの累積値

(複数宛先) へのデータ通信経路制御方式を策定した。

(3) センサデータの時間同期方式

提案方式によるセンサデータの時間同期精度の評価結果を表 7 に示す。BT パケット伝送遅延 (①) は、図 7 の評価システムで Ethernet スイッチを介した 2 つの ECU 間での UDP パケットの遅延時間を測定した。本パケットはペイロード長を 18 [Byte]、IP ヘッダの TOS 値を最高優先度としたものである。CAN ECU 内処理遅延 (②) は、実車における CAN 内の ECU から定期送信される車両制御関連の特定の CAN メッセージをネットワーク上でキャプチャし、キャプチャ時刻から送信周期に対するジッタを算出した。それをもとに処理ばらつきの幅 (ジッタの最大と最小の差分) を算出して測定結果とした。CAN ECU 内処理遅延は、ECU 内処理に必ず必要となる最小処理時間と、上記処理ばらつきの合計値となる。最小処理時間は一定であることから、CGW で CAN メッセージ受信後に時刻補正することができるため、処理ばらつきを同期精度誤差とした。CAN メッセージ伝送遅延 (③) は、データフィールド長が 8 [Byte] のパケットについて、CAN の通信帯域から理論値として算出した [12]。クロック精度誤差 (ii) は、BT パケットを 1 [s] 間隔で周期配信し、AD と CGW のクロックの周波数精度がともに ±150 [ppm] である場合の時間計測の精度誤差を理論値として算出した。

以上の測定方法により、BT パケットの同報周期を 1 [sec] にすることで、Ethernet の精度誤差は BT パケット伝送遅延 (①) とクロック精度誤差 (ii) の合計から 20~450 [μs] となる。CAN を含めたネットワークシステム全体は、Ethernet の精度誤差に対し、CAN ECU 内処理遅延 (②) と CAN メッセージ伝送遅延 (③) を加えて 270~1700 [μs] となる。同期ずれが 1700 [μs] のとき、自車両と他物体の速度差が 100 [km/h] と仮定した場合の位置の誤差は 47.2 [mm] となる。本誤差は、自車の走行軌道計画には影響を与えないと考えられる。以上の結果から、自動運転システムにおけるセンサデータの簡易的な時間同期方式が適用可能であることを確認できた。

6. おわりに

本研究では、Ethernet に接続された複数の ECU と、複数の CAN バスが接続された CGW 型ネットワークアーキテクチャの車載システムにおいて、自動運転を実現するためのネットワーク設計を行った。設計したネットワーク構成において課題となる、CAN と Ethernet 間をまたがったデータ通信の経路制御方式、およびセンサデータの時間同期方式を検討した。これらの方式を組み込んだ CGW 試作機を開発し、方式の動作および有効性を確認し、自動運転システムへ適用できる見通しを得た。

今後は、実車での試作システムを開発し、実際の自動運転動作環境における提案方式評価を行っていく。

参考文献

- [1] Kirsten, M. and Königseder, T.: *Automotive Ethernet*, Cambridge University Press (2014).
- [2] Triess, B.: Cost Efficient Implementation of High Performance Automotive Networks, *Nikkei Electronics Symposium, In-Vehicle Ethernet to Shape the Future of Automobile* (2013).
- [3] 入手先 (<http://www.sei-automotive.jp/event/hito2014nagoya/pdf/2014n-13.pdf>) (参照 2015-12-17).
- [4] Kern, A., Reinhard, D., Streichert, T. and Teich, J.: Gateway Strategies for Embedding of Automotive CAN-Frames into Ethernet-Packets and Vice Versa, *Architecture of Computing Systems, ARCS2011*, Vol.6566 of Lecture Notes in Computer Science, pp.259-270 (2011).
- [5] 松村 潤 他: 車載ネットワークにおける CAN-Ethernet プロトコル変換アルゴリズム, 情報処理学会, Vol.2013-SLDM-160, No.7 (2013).
- [6] OMNeT++, website, available from (<http://www.omnetpp.org>) (accessed 2015-12-17).
- [7] Nacer, A.A., Runser, K.J., Scharbag, J.L. and Fraboul, C.: Strategies for the Interconnection of CAN buses through an Ethernet Switch, *Industrial Embedded Systems (SIES), 2013 8th IEEE International Symposium*, pp.77-80 (2013).
- [8] Pongsathorn, R. ほか: 単眼カメラとミリ波レーダによる歩行者検出と衝突警報システムの開発, 自動車技術会自動車技術会論文集, Vol.40, No.6, pp.1587-1592 (2009).
- [9] IEEE802.1AS, Timing and Synchronization10) OPEN ALLIANCE, website, available from (<http://www.opensig.org/>) (accessed 2015-12-17).
- [10] Steffen, R.: Design and Realization of an IP-based In-car Network Architecture.
- [11] AUTOSAR, website, available from (<http://www.autosar.org/>) (accessed 2015-12-17).
- [12] Goto, H.: The Latest Trend of In-vehicle Ethernet Network, *Nikkei Electronics Symposium, In-Vehicle Ethernet to Shape the Future of Automobile* (2014).

1 CAN は、Bosch 社の登録商標です。

2 Ethernet は、富士ゼロックス株式会社の登録商標です。

3 BroadR-Reach は、Broadcom Corporation の登録商標です。

4 AUTOSAR は、AUTOSAR GbR の登録商標または商標です。



木谷 光博

1980 年生。2004 年中央大学大学院理工学研究科情報工学専攻博士前期課程修了。同年日立製作所入社。車載システムに関連する研究に従事。



片岡 幹雄

1977 年生。2002 年岡山大学大学院自然科学研究科電子情報システム工学専攻博士前期課程修了。同年日立製作所入社。車載システムに関連する研究に従事。電子情報通信学会会員。



寺岡 秀敏

1976 年生。2002 年京都大学大学院工学研究科修士課程修了。同年日立製作所入社。車載システムに関連する研究に従事。