

CANの高速化に関する研究

倉地 亮^{†1} 宮下 之宏^{†2} 高田 広章^{†1}

現在、自動車内の制御系ネットワークにおいて、事実上の標準として使われている CAN(Control Area Network) プロトコルは、現在の制御システムにおいてもその通信容量は限界に達しているとされる。CAN はメッセージ送信時、1 ビットずつ同期させることで送信権の調停や ACK の返却を行ってきたが、高速化の限界は、伝搬遅延の影響によりそれらが成立しないことにある。本研究では、高速化への限界と課題について整理し、トポロジの変更及び衝突検知アルゴリズムを用いることで、CAN 高速化への手法を提案する。

Research on high-speed CAN(Control Area Network) for improved bandwidth

RYO KURACHI,^{†1} YUKIHIRO MIYASHITA^{†2} and HIROAKI TAKADA^{†1}

Today, the data capacity in automotive control network has reached the limits of CAN(maximum data rates for 1 megabits per second). Speed of CAN is affected by the propagation delay, because of using bit synchronous transmission. For increasing data transmission speed, we must change arbitration process and ACK process. We propose the 10 megabits per second CAN architecture for improved bandwidth by using a new collision detection algorithm and topology change.

1. はじめに

近年、自動車制御システムの急速な電子化に伴い、一台の車に搭載される ECU(Electronic Control Unit) の数も急激に増加し、これまで事実上の標準として採用されてきた CAN (Control Area Network) プロトコル¹⁾ で流せる通信容量も限界に達しているとされる。そのため、次世代向け通信規格として FlexRay(最大通信速度 10Mbps) などが提案され、現在注目を集めている。²⁾⁻³⁾

FlexRay では X-by-Wire へ向けた様々な機能が付加されており、信頼性が高い反面、物理的コストが高いことなどから、導入にあたっては必要に迫られた部分的なサブネットワークからすすめられていくであろうとされる。一方、制御システム全体に FlexRay を適用するためには、CAN で設計されていたアプリケーションを再構築する必要があるため、設計資産の流用や信頼性の担保という点からも大きな転換コストが必要になることが課題とされる。特に自動車のようなコスト制約が厳しく、複雑な制御システムでは、従来の

アプリケーションを生かしたより高速な通信方式が必要となる。

本研究では事実上の標準として採用されてきた CAN をベースに構成されるアプリケーションを生かしたより高速な通信方式として 10MbpsCAN 方式を提案する。

本論文では、第 2 章で研究対象となる車載 LAN システムの概要ならびに、CAN 高速化の課題を整理し、第 3 章では問題点を考慮した上での 10Mbps CAN 通信方式のアルゴリズムの説明を行う。第 4 章では 10Mbps CAN 通信方式の実装について説明し、考察する。最後に第 5 章で本論のまとめとする。

2. 車載 LAN システム

2.1 車載制御システム

これまで、自動車制御システムでは様々な要求に対応するため、制御の中心が機械式制御から電子制御へと徐々に移行され、CAN を中心にそのネットワークが構成されてきた。そして現在では FlexRay などの次世代プロトコルの規格により、ブレーキやステアリングなどの非常に高い信頼性が求められる制御においても、X-By-Wire と呼ばれる電子制御システムへの移行が実現されようとしている。³⁾ しかし、FlexRay は物理的なコストが高いことなどから、現在 CAN で実現されているシステム全体を移行するのではなく、

^{†1} 名古屋大学 大学院情報科学研究科 附属組込みシステム研究センター

Center for Embedded Computing Systems, Nagoya University

^{†2} 株式会社 オートネットワーク技術研究所
AutoNetworks Technologies, Ltd.

今後必要とされるサブネットワークから段階的に導入され、複数のプロトコルを中継するゲートウェイなどの装置を用いて、それぞれのサブネットワーク間を接続したシステム構成になると想定できる。

2.2 CAN

CAN ではフレーム単位のメッセージで送受信を行い、一つのフレームで送信できる最大データ長は8バイトである。それぞれのフレームは11ビットまたは28ビットの固定長IDをもつ。メッセージの最後尾には送信ユニットが計算したCRCが付与されており、受信ユニットでは自ら計算したCRCと受信したCRC Fieldの値が一致した場合のみ、正常に受信できたと判断しACKを送信することでメッセージの送受信が成立したとする。その後、End of Frame(EOF)3ビットとInter Frame Space(IFS)7ビットのインターバル期間待った後、次のメッセージの送信が可能になる。尚、CANでは1ビットをドミナントとリセッシブの2値で表現している。⁴⁾

SOF	ID	Control	Data	CRC	ACK	EOF
-----	----	---------	------	-----	-----	-----

図1 CANメッセージフレーム概要

CANのメッセージ送信手順として、バスが空いている場合、全てのユニットが送信を開始することができるマルチマスタ方式が採用されている。

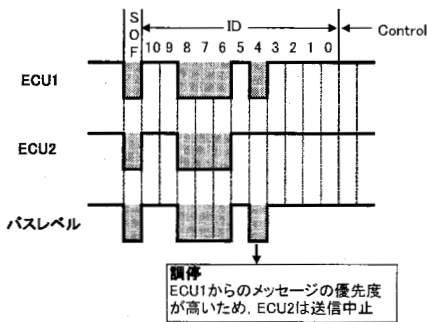


図2 アービトレーション (送信権の調停)

複数のユニットが同時に送信を開始した場合には各メッセージは衝突するが、IDを1ビットずつ同期しながらぶつけることで、調停に負けたユニットから送信を中断し、最も小さいIDを持つメッセージが優先的に送信されるようアービトレーション(送信権の調停)を行う。CANではアービトレーションにより、必ずひとつのメッセージは送信することを保証している。(図2) また送信権の調停が成立するためには各ユニット間において、ビット単位の同期が成立していることが前提となり、これはACKの返送に対しても同様と

いえる。

2.3 CAN高速化の課題

CANの高速化を検討する上で、通信路上における伝搬遅延の影響を考慮する必要がある。下図は現状のCANにおける伝搬遅延の影響と高速化した場合の伝搬遅延の影響を模式化したものである。(図3, 図4)ここでは伝搬遅延時間を τ 、1BTを1ビット送信するのにかかる時間とし、ECU1からECU2へメッセージが送受信された場合を表現している。

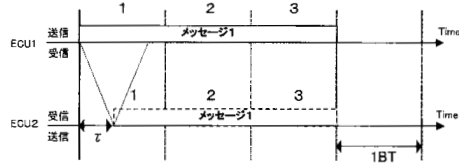


図3 伝搬遅延に対する説明 ($\tau < 1BT$)



図4 伝搬遅延に対する説明 ($\tau > 1BT$)

CANにおける通信速度の限界はアービトレーション(送信権の調停)とACKの返却方法が成立しないことにある。CANでは送信ユニットと受信ユニットが伝搬遅延時間 $\tau < 1BT$ で同期しながら、送信権の調停やACKの送受信を行うことで通信が成立するが、CANハーネスを例にとると、1メートルあたり5(ns)程度の伝搬遅延が発生するため、現状の通信方式のまま通信速度を上げた場合、極端に伝送路を短くしない限りは送信ユニットと受信ユニットの間で伝搬遅延時間 $\tau > 1BT$ が発生し、通信が成立しなくなるといえる。(図4) このため、伝搬遅延の問題を解決しない限り、従来のCAN通信方式を高速化させることはできないといえる。

また、自動車内の制御ネットワークでは物理的なコストや配策スペース低減などから多くはバストポロジが採用されてきた。しかし、電気的な負荷などから通信バスにつながるECUの数は通信速度によって制限されるため、通信速度を上げるためには一つのバスにつながるECUの数を減らさざるを得ない。結果、高速な通信を採用する場合にはバストポロジからトポロジの変更を行う必要があるといえる。

本論では、これらの問題をCAN高速化の限界と定

義する。

3. 解決のための提案

3.1 前提条件

本方式では CAN 高速化への実現可能性を追求するため、以下の前提条件を定めた。

- 高速化するにつれ、増大する伝搬遅延に対応可能な通信方式
- アプリケーションからみて、CAN と同じインタフェースを有するコントローラであること
- 全二重ではなく、半二重であること

通信路上における伝搬遅延時間の影響は伝送路を極端に短くすることが無い限り発生する。今後、10Mbps 以上の通信速度を実現するためには、高速化するにつれ増大する伝搬遅延に耐える方式であることとした。そして CAN と親和性が高いとする理由として、CAN で設計されたアプリケーションからみて、従来の CAN コントローラと同じインタフェースを有するコントローラであり、現状のシステムを移行するのに容易であることを前提とした。また、現在の車載制御システムに適用する場合、ハーネスの増加は物理的なスペースの逼迫やコスト、車重の増加につながるため、従来のとおり半二重であることを前提とした。

3.2 Point-to-Point 接続

これまで ECU の多くはバス型ネットワークにつながれていたが、今後、次世代の通信規格により通信速度が高速化するにつれ、物理的な制約から、アクティブスタートポロジなどの Point-to-Point 接続 (P-P 接続) へと変遷していくと想定できる。FlexRay では、アクティブカプラーという中継機を用いて ECU と P-P 接続をすることにより、アクティブスタートポロジを構成し高速化を実現している。FlexRay におけるアクティブカプラーは電気信号の中継に使うリピータと同等であり、従来の CAN でいうゲートウェイを用いて ECU と P-P 接続を行うことと物理的には同様であるといえる。このため、本方式では ECU とゲートウェイの P-P 接続により高速化を実現するものとする。(図 5)

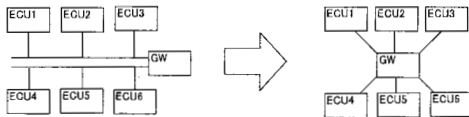


図 5 CAN の P-P 接続

一方、ECU とゲートウェイの P-P 接続により、これまで CAN で問題とされていた高トラフィック時の

送信時間保証が実現可能になる。従来の CAN では ID による送信権の調停により、優先度の低い ID をもつメッセージの送信がいつまで経っても実行されず、以降のメッセージも送信できない問題が存在した。⁵⁾ そこで本方式では P-P 接続により、メッセージ送信と応答を交互に行い、更に応答には送信データの付与を可能にすることで、高トラフィック時でもメッセージの送信時間保証を行うことを可能にする方式とした。

3.3 衝突検知アルゴリズム

高速化し伝搬遅延の影響が顕在化することにより、従来の送信権の調停は成立しないため、一方が先に送信したメッセージを他方が受信し始めるまでの間にメッセージを送信し始めると、互いのメッセージが衝突する場合がある。そのため、バスアイドル時に送信を開始したメッセージでも衝突は起こりうるとの条件から、衝突後の再送では必ず送受信が成立するアルゴリズムが必要となる。衝突検知後は ECU とゲートウェイの P-P 接続であることから、再送処理の優先権は先に ECU から送信されたメッセージを持つゲートウェイを優先とした。また、再送を含む送受信が成立して以降、送信したいメッセージが存在し続ける間は必ず交互に送受信を行うことで通信トラフィックが高い場合には衝突を回避する方式とした。

下図は本方式で提案する衝突検知アルゴリズムである。(図 6) ここで表すそれぞれの記号はドミナントビット D 、リセツピット R 、EOF 時間 EOF 、IFS 時間 IFS 、伝搬遅延時間 τ 、送信側と受信側のクロック誤差を α と定義した。

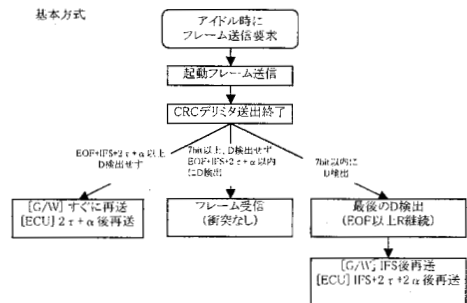


図 6 衝突検知アルゴリズム

各ユニットはバスアイドル時にフレームの送信要求が到着すると、衝突が起こりうる起動フレームを送信開始し、CRC デリミタの送信が完了した時点に受信応答確認を開始する。このとき、送信したフレームが衝突したかどうかについては送信完了後 7 ビット以内にドミナントビットを受信したかどうかで判断する。まず CAN ではフレームの先頭である SOF がドミナントで定義されており、CRC デリミタはリセツピット

で定義されているが、本方式ではフレームの最後とみなすCRCデリミタをドミナントへ変更する。更にフレーム中に同一レベルが5ビット連続した場合、次のビットをスタップビットとして1ビットの反転データを挿入するビットスタップルールがあり、必ず6ビットに1ビットはドミナントビットが存在することを利用する。このため、最低でも6ビットにクロック誤差などを考慮した1ビットを含む7ビットを監視すれば、衝突検出期間としては適当であると判断し条件を設定した。また衝突検知後の処理は相手の送信が終了する、すなわちパスがアイドルになるまで受信処理を継続し、ゲートウェイとECUに非対称な一定の再送待ち時間を規定することで、ゲートウェイからの再送処理を優先させている。

また一方で、互いに衝突が検知できなかった場合にも、ゲートウェイからの再送により通信が復帰できるよう再送期間を設定した。

本方式により想定される衝突検知は、以下の3通りが挙げられる。

- (1) どちらかが衝突を検知した場合
- (2) どちらも衝突検知した場合
- (3) どちらも衝突検知できない場合

以下にそれぞれの動作について説明する。

3.3.1 どちらかが衝突を検知した場合

どちらかが衝突を検知した場合として、ゲートウェイが衝突を検知した場合を例に説明する。(図7)

ここで用いる記号は図5と同様であり、メッセージフレーム送信時間を C_m とする。

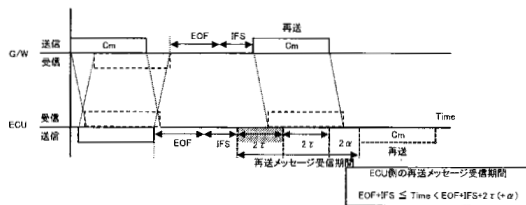


図7 ゲートウェイが衝突検知

まず、ゲートウェイ (GW) が先にメッセージの送信を開始し、ECU が受信を開始するまでの間に送信要求が発生したため、ECU もメッセージ送信を開始する。その後、ゲートウェイが先にCRCデリミタを送信終了し、7ビット以内にドミナントを検知した場合、EOFを検出するまで受信を行い、IFS 待った後にゲートウェイから再送を開始する。一方、ECU 側は送信完了後、 $EOF+IFS < Time < EOF+IFS+2\tau+2\alpha$ 以内に D を検出することでゲートウェイからの再送

フレームのSOFを受信し、その後再送を行なう。以降は各ユニットの送信したいメッセージが無くなるまで、交互通信を実施する。ただし、ECUはゲートウェイからの再送フレームであるSOFを、CRCデリミタ送信完了から $EOF+IFS+4\tau+2\alpha$ 後に受信できない場合はECUから再送を行うものとする。

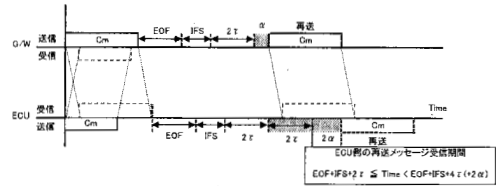


図8 ECUが衝突検知

またECUのみが衝突を検知した場合、ゲートウェイのタイムアウトによる再送を待つようスケジュールすることで、ゲートウェイからの再送を優先的に送信させることが可能となる。(図8)

3.3.2 どちらも衝突検知した場合

$\tau > 1BT$ 以上の伝搬遅延が存在し、ゲートウェイとECUが同じ長さのメッセージを同じタイミングで送信開始した場合、どちらも衝突を検知する可能性がある。(図9) この場合、ゲートウェイとECUのどちらもCRCデリミタ送信完了後に衝突を検知し、ゲートウェイの再送を機に交互通信が開始され、以降メッセージが無くなるまで交互通信を行なう。

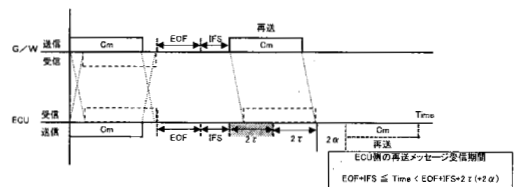


図9 どちらも衝突検知

3.3.3 どちらも衝突検知できない場合

仮に伝搬遅延が $1BT$ 以下である場合 ($\tau < 1BT$)、ゲートウェイとECUが同じ長さのメッセージを同じタイミングで送信開始すると、送信完了後の次のビットには相手のメッセージが伝送路を通過し終わっている可能性があり、どちらも衝突が検知できないと想定できる。(図10) この場合、ゲートウェイとECUの両方とも応答が無いと判断するが、ゲートウェイからの再送を優先させるよう各タイムアウト時間を規定することで再送の成立を可能としている。

以上を10MbpsCANの基本方式と定義し、以降ではその実装について述べる。

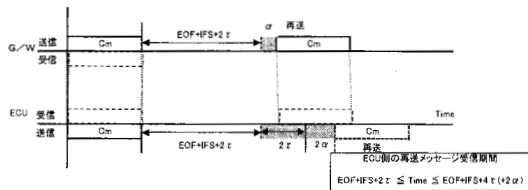


図 10 どちらも衝突検知できない

4. 実 装

実装にあたり、まず最初に C 言語によるソフトウェアシミュレーションを行い、衝突検知アルゴリズムが正しいこと及びコントローラ設計の詳細について検討を行なった。その後、本論で提案している 10MbpsCAN コントローラを Altera 社の FPGA ボードへ実装した。本方式で提案したアルゴリズムを FPGA へ実装する際、10MbpsCAN 方式の状態遷移マシンを設計した。実装したアルゴリズムから生成されたコントローラのブロック図と状態遷移マシンについては下の図のとおりである。(図 11, 図 12)

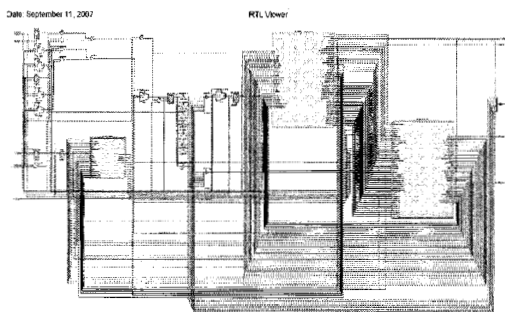


図 11 コントローラのブロック図

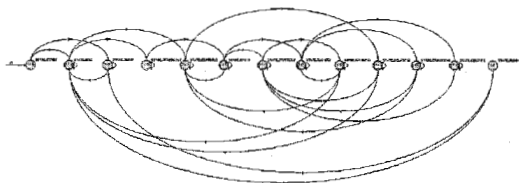


図 12 本方式の状態遷移

また実装したコントローラのロジックエレメント数は次表のとおりとなった。(表 1)

更にホスト CPU と 10MbpsCAN コントローラのインタフェースレジスタは従来の CAN のコントロー

表 1 Flow Summary

	status
Total logic elements	2,505
Total combinational functions	2,505
Dedicated logic registers	1,424
Total registers	1,424
Total pins	19

ラ仕様からゲートウェイと ECU の識別用レジスタ、及び衝突検知アルゴリズムで使用する各規定時間を初期化時に設定するのみに留める事で、従来のアプリケーションとの互換性を保つことを実現した。

5. ま と め

本研究では CAN 高速化の課題を明らかにした上で、CAN 高速化の手法を提案した。

これまで CAN の高速化に対する手法はあまり提案されていなかったが、まず高速化への課題として、伝播遅延の影響が顕在化することによりアービトラージョンや ACK の返却が成立しないことを整理した。その上で従来のアプリケーションを活用可能な高速化手法として、ゲートウェイと ECU の P-P 接続ヘトポロジの変更を行い、非対称な衝突検知アルゴリズムを実装することで高速化への方式を提案した。

今後の課題として、本論で提案する方式の評価が挙げられ、実際のネットワークモデルに適用した場合の妥当性を検証する必要がある。また、更なる高速化を実現可能であることを実証する必要がある。

謝辞 本研究は、株式会社オートネットワーク技術研究所との共同研究であり、本研究を行うにあたり、有益なご助言を頂いた夏目様、山本様には深く感謝いたします。

参 考 文 献

- 1) ISO 11898, Road Vehicles - Interchange of digital information - Controller area Network (CAN) for high speed communication, 1993.
- 2) FlexRay Consortium, <http://www.flexray.com/>.
- 3) Thomas Nolte, Hans Hansson, and Lucia Lo Bello. Automotive communications - past, current and future. Proceedings of 10th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA '05), Vol. 1, pp. 985-992, 2005.
- 4) M. Livani, J. Kaiser, and W. Jia. Scheduling hard and soft real-time communication in the controller area network (can, 1998).
- 5) K.Tindell, A.Burns, A.Wellings, "Calculating Controller Area Network (CAN) Message Response Times," Control Engineering Practice, (1995).