情報処理学会論文誌 Vol.53 No.8 1962–1975 (Aug. 2012)

最大遅れ時間解析による スケーラブルCANプロトコルの性能評価

倉地 亮[1,a)] 高田 広章[1] 西村 政信[2] 堀端 啓史[2]

受付日 2011年11月30日, 採録日 2012年5月12日

概要：現在，様々な機能を実現するために，車載ネットワークに流れるデータ量は急激に増加しており，広く採用されている Controller Area Network（CAN）では回線容量が不足している．しかしながら，CAN の最大伝送速度はバス上での送信権の調停や ACK 応答メカニズムにより制約されており，最大伝送速度を向上させるためにはプロトコルの改良が必要となる．そこで，我々は CAN を改良したスケーラブル CAN プロトコルを提案している．本論では，スケーラブル CAN プロトコルの最大遅れ時間解析手法を提案し，スケーラブル CAN プロトコルが CAN よりスループットを向上させたことにより，十分なメッセージ量を転送できる能力があることを示す．

キーワード：Controller Area Network，CAN，車載ネットワーク，Rate Monotonic Analysis

Performance Analysis of Scalable CAN with Worst-case Response Time Analysis

Ryo Kurachi[1,a)] Hiroaki Takada[1] Masanobu Nishimura[2] Satoshi Horihata[2]

Received: November 30, 2011, Accepted: May 12, 2012

Abstract: The Controller Area Network (CAN) is widely employed in in-vehicle networks to install new functions. However, CAN has significant restrictions on upper bound of maximum speed due to its bit-wise arbitration and acknowledgement sequences. Therefore, we have proposed a new high-speed protocol "Scalable CAN" to improve upper bound speed of CAN. In this paper, we propose the worst-case response time analysis for "Scalable CAN". According to our analysis, the Scalable CAN has achieved much higher throughput performance than CAN, and it has the ability to transfer a sufficient amount of message.

Keywords: Controller Area Network, CAN, In-vehicle network, Rate Monotonic Analysis

1. はじめに

現代の自動車では，多くの Electronic Control Unit（ECU）が車載ネットワークを介して大量のデータを交換することにより，様々な制御や機能を実現している．車載ネットワークの中でも特に厳しいリアルタイム性が要求される制御系ネットワークでは Controller Area Network（CAN）[1] が広く使われている．CAN は自動車制御のようなリアルタイムシステムのための通信ネットワークに適したプロトコルであるが，その最大伝送速度が 1 Mbps であるために X-by-Wire [2] のような将来の車載ネットワークシステムに対する要求を満たしているとはいえない．

そこで，これまでに CAN プロトコルを改良することにより高速化を実現する様々な研究が進められている [3], [4], [5]．しかしながら，これらの研究で提案される CAN を改良したプロトコルは物理的に 1 対 1 接続を採用するため，バス接続が主流である CAN を置き換えるにはゲートウェイなどの追加のハードウェアが必要になりコストが高くなるこ

[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, Nagoya, Aichi 464–8601, Japan
[2] 株式会社オートネットワーク技術研究所
AutoNetworks Technologies, Ltd., Yokkaichi, Mie 510–8503, Japan
[a)] kurachi@nces.is.nagoya-u.ac.jp

とや，配線量が増えるために車重が増加することが問題である．そのため，我々はバス接続を可能とするスケーラブル CAN プロトコルを提案している [6]．

本論では，スケーラブル CAN プロトコルの伝送能力を評価するために，スケーラブル CAN の最大遅れ時間解析手法を提案する．次に，最大遅れ時間解析手法を用いて CAN との性能比較を実施し，スケーラブル CAN が CAN よりも高いスループットを実現したことにより，十分なメッセージ量が送信可能であることを示す．

本論の構成は，2 章でスケーラブル CAN プロトコルの概要を説明し，3 章で解析の前提や定義，関連研究を説明する．続く 4 章で，スケーラブル CAN メッセージの最悪状況を示し，5 章で最大遅れ時間解析手法を導出する．6 章では最大遅れ時間解析を用いてスケーラブル CAN の伝送能力を評価し，7 章で本論をまとめる．

2. スケーラブル CAN

本章では，我々が提案しているスケーラブル CAN プロトコルについて説明する．なお，本論では性能評価に必要となる機能に着目して説明するため，その他のプロトコルの詳細は文献 [6] を参照されたい．

2.1 プロトコルの概要

スケーラブル CAN プロトコルは，最大伝送速度を 1 Mbps とする CAN プロトコルのバス上での送信権の調停や ACK 応答メカニズムを改良することにより，その最大伝送速度を改善することを目的とするプロトコルであり，CAN 同様，OSI 参照モデルにおける Logical Link Control（LLC）および Media Access Control（MAC）に位置づけられるプロトコルである．また，スケーラブル CAN は CAN から送信権の調停方法と ACK 応答メカニズムを改良しているため，バス上では CAN プロトコルとの互換性を持たない独自のバスプロトコルである．

このスケーラブル CAN ではソフトウェアからみて CAN コントローラと同じレジスタセットを持つ通信コントローラを用意するため，CAN のために作られたソフトウェアの移植が容易となる．FlexRay [7] などの他の次世代車載 LAN プロトコルと比較する場合，CAN からスケーラブル CAN へはアプリケーションが容易に移植できるため，導入時のソフトウェアの開発規模が抑えられる点が優位である．このため，もしユーザが CAN の回線容量に不満がある場合には，**図 1** に示すように，通信コントローラやトランシーバなどのハードウェアを交換することで容易に帯域拡張が可能となる．さらに，FlexRay ではプロトコルで ACK をサポートしていないために，既存する CAN アプリケーションを FlexRay 上に移植する場合には，定常的に再送を行うことで送達を保証する必要があり，CAN メッセージの送信周期を見直す必要がある．具体的には，CAN で設定された送信周期内に 2 回あるいは 3 回送信するなどして，再送を保証する必要があり，実質的な回線容量が 1/2 倍や 1/3 倍になる懸念がある．一方，スケーラブル CAN ではプロトコルで ACK をサポートしているため，このような見直しは必要なく CAN と同じアプリケーションを使用できる点でも優位である．

2.2 巡回型スケジューラ

スケーラブル CAN プロトコルは，CAN のようなバス上でのフレーム単位での送信権の調停は行わず，各 ECU にあらかじめ割り当てられた順番で送信権が巡回するスケジューリングを採用している．このため，各 ECU はタイムスロットと呼ばれる送信可能な時間が周期的に割り当てられ，その時間においてのみ送信を行うことができる．

本プロトコルでは各タイムスロットには 1 つのフレームのみを送信することを前提としており，各 ECU は自身が送信するタイムスロットを通信前に静的に設定されるものとする．

バスの起動時，まず各 ECU はネットワークの各 ECU が保持するタイムスロット時間の同期処理を行う．この同期

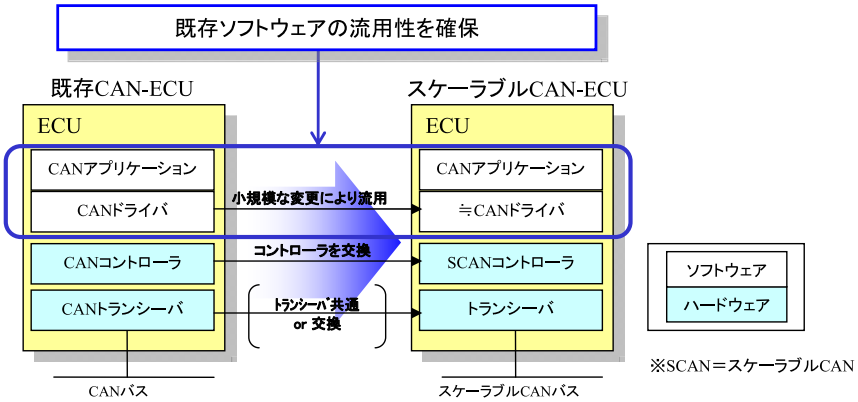


図 1 スケーラブル CAN のコンセプトと CAN からの移行イメージ
Fig. 1 Concept and migration image from CAN to Scalable CAN.

処理中はアプリケーションからのデータ送信は行えず，同期用のフレームのみがネットワーク上に送信される．なお，同期アルゴリズムの詳細は文献 [8] を参照されたい．そのうえで，同期が完了した各 ECU は，自身の送信タイムスロットが回ってくるときに送信要求されたメッセージがある場合には，その送信要求されたメッセージのうち最も優先度の高いメッセージをそのタイムスロットに送信する．一方，送信要求されたメッセージがない場合にも，ACK フレームと呼ばれる最小長のフレームを送信することで，各 ECU 間のリンクを継続させる方法をとる．なお，もし ECU が故障しあるタイムスロットにフレームが送信されない場合には，その他の ECU はそのタイムスロットへ遷移後フレームが送信開始されるまでの上限時間内にフレームの受信開始を検出できないと，そのタイムスロットをアイドルスロットと見なし次のタイムスロットへと遷移することで，バス上での無効な時間を最小としつつ巡回を続けることができる．このため，各タイムスロット長は送信するフレームの長さにより伸縮するものである．

また，1 サイクル内に存在するすべてのタイムスロットの合計数を $turn$ と呼び，あるタイムスロットを θ_i（$1 \leq i \leq turn$）で表す．各タイムスロットは小さい番号順（$\theta_1 \to \theta_2 \to \cdots \to \theta_{turn}$）に実行されるものとする．前述するように各タイムスロット長はそのタイムスロットに送信されるフレーム長により伸縮するため，1 サイクル長（1 サイクル時間）も各タイムスロット長に応じて伸縮するものとする．また，$turn$ 番目のタイムスロットの次は 1 番目のタイムスロットに戻ることで，バスの起動中は各タイムスロットがつねに巡回し続けるラウンドロビン型のプロトコルとなる．

2.3 送信メッセージの確定タイミング

送信メッセージの確定タイミングは，次に遷移するタイムスロットが自身の送信タイムスロットである場合にそのタイムスロットの開始時刻において確定されるものとし，以降ではこの時刻を送信開始時刻と呼ぶ．

3. 解析の前提と定義，関連研究

3.1 解析の前提

本論では，バスの起動時に行われる各 ECU 間のタイムスロットの同期処理については扱わず，同期完了後の最大遅れ時間解析について議論する．最大遅れ時間とは，ある解析対象となるメッセージが送信要求されてからバス上に送信完了するまでの最大遅延時間のことを指す．また，バスに接続される各 ECU の故障時は考えないものとする．さらに，メッセージの送信要求に対するジッタは発生せず，送信要求はあらかじめ決められた時刻に必ず発生するものとする．なお，これらの前提は，比較対象となる CAN メッセージの最大遅れ時間解析と同じ条件で比較するための前提となる．

また，スケーラブル CAN では，CAN 同様，各メッセージは最大データ長を 8 バイトとする小さなフレーム単位で通信を行うため，バスに送信を開始すると送信完了まで送信動作を継続する仕様である．このため，本解析においては，各メッセージはノンプリエンプティブとして扱う．なお，以降では，対象メッセージを送信する ECU を自 ECU と呼び，それ以外の ECU を他 ECU と呼ぶものとする．

3.2 関連研究

CAN メッセージの最大遅れ時間解析手法とは，リアルタイムスケジューリング理論である Rate Monotonic Analysis（RMA）に基づいて，CAN メッセージをノンプリエンプティブな優先度ベーススケジューリングとして扱うことにより，その最大遅れ時間を導出するための手法である．

最初の CAN メッセージの最大遅れ時間の解析手法として，Tindell らは基本となるオフセットが付かない CAN メッセージの解析手法を提案し [9]，様々な研究が行われた [10], [11], [12], [13]．その後，各 CAN メッセージにオフセットと呼ばれる初回送信時までの待ち時間を持たせ，各 ECU から送信される CAN メッセージを分散させることで，リアルタイム性を保証しながら CAN ネットワークの回線使用率を向上させる取り組みがなされており [14], [15]，これらのオフセット付き CAN メッセージの最大遅れ時間の解析手法については，飯山らがマルチフレームタスクのモデルを適用した手法を提案した [16]．また，Lei らは安全な解析手法として，飯山らとは異なる解析手法を提案している [17]．

本論で提案する解析手法は，文献 [16] で用いられる解析手法を改良し，ラウンドロビン型のネットワークであるスケーラブル CAN に適用したものである．

3.3 メッセージセット

メッセージセットは，m 個の ECU（ECU_1, $ECU_2, \ldots, ECU_m$）で構成されるものとし，各 ECU は複数の送信メッセージを持つものとする．ある送信メッセージ τ_i は，メッセージの優先度 i，そのメッセージの最大送信時間 C_i，送信周期 T_i，初回送信時のオフセット O_i の 4 つ組み (i, C_i, T_i, O_i) で表される．なお，CAN とスケーラブル CAN ではフレームの構成が異なるため，最大送信時間 C_i の導出方法は付録 A.1 を参照されたい．

3.4 オフセットとメッセージモデル

各メッセージの送信要求は，初回送信時に限りオフセットと呼ばれる各 ECU の起動時刻からの一定の待ち時間後に実行されるが，以降は周期的に実行される．

1 つの ECU が送信するすべてのメッセージは同じタイマを用いて送信要求を行うため，同一 ECU から送信され

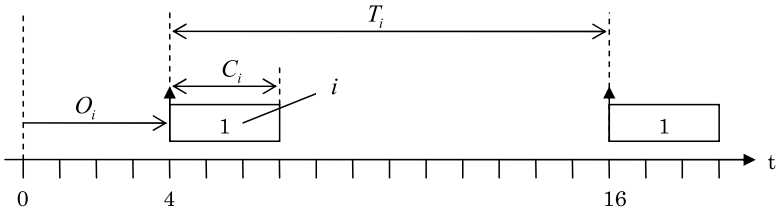


図 2 メッセージモデル
Fig. 2 Message model.

るメッセージの送信要求時刻は同期しているものとして扱うことができる．一方，異なる ECU が送信するメッセージは，各 ECU で独立するタイマを用いて送信要求され，各タイマを同期させる機構はないため，同期していないものとして扱う．つまり，同一 ECU 内から送信するメッセージは，オフセットにより送信要求時刻を分散させることで送信要求の衝突を回避し，送信までの待ち時間を減らすことが可能だが，他 ECU からは最大となる送信要求状況でメッセージの送信が阻害される可能性がある．

本論で扱うメッセージモデルは，あるメッセージ $\tau_1 = (1, 3, 12, 4)$ である場合，**図 2** のように図示するものとする．

4. メッセージの最悪状況

本章では，スケーラブル CAN におけるメッセージの最大遅れ時間を解析するために，どのような状況においてメッセージの遅れ時間が最悪となるかを示す．以降では，まず，解析対象となるメッセージが他 ECU の送信メッセージから受ける影響のみに限定した場合の最悪状況を示した後，自 ECU の送信メッセージから受ける影響も含めた議論を行う．

4.1 メッセージの滞留時間

あるメッセージが通信コントローラ内の送信メールボックスに格納されてから，その送信完了するまでの時間をメッセージの滞留時間と呼ぶ．このため，メッセージの滞留時間には，そのメッセージが他 ECU の送信メッセージから受ける影響を含むが，自 ECU から送信される他のメッセージの影響は含まれない．以下では，メッセージの滞留時間の上限を与える定理を示し，その証明を行う．

定理 1（メッセージの滞留時間） あるメッセージの滞留時間は，そのメッセージが送信メールボックスに格納されると同時に，自 ECU に与えられる送信タイムスロットの送信開始時刻を過ぎ，次の自 ECU の送信タイムスロットまでの間に存在するすべての他 ECU が送信するタイムスロットにおいて，最大となるメッセージの送信が発生する状況である．

証明 1 $turn = n$，つまり n スロットを 1 サイクルとして巡回するネットワークにおいて，各 ECU に 1 つずつ送信タイムスロットが割り当てられているものとする．このとき，解析対象となるメッセージ τ_i が送信されるタイムスロット θ_n のある開始時刻を 0，そのタイムスロットの終了時刻を t_0 とすると，その初回送信タイムスロット θ_n の送信時間は t_0 となる．また，次サイクルの送信タイムスロット θ_n の開始時刻 t' は，初回送信タイムスロット θ_n の送信時間 t_0 とそれ以降の他の ECU が送信するタイムスロット θ_1 から θ_{n-1} までの各最大送信時間を足すことにより，次式から導出できる．

$$t' = t_0 + \sum_{j=1}^{n-1} C_j \tag{1}$$

なお，C_j は各タイムスロットにて送信されるメッセージが最大送信時間を使いきる場合のタイムスロット時間を表し，各タイムスロットにおいて最大長となるメッセージが送信される場合，t' は最大となるサイクル時間を表す．
対象メッセージ τ_i がオフセット O_i を用いて，時刻 0 より後（$0 < t < t'$）に送信要求される場合，式 (1) で与えられる t' よりも短い時間で次サイクルの θ_n に送信開始することができ，メッセージ τ_i の滞留時間は t' より短くなる．このため，最長となる遅れ時間が発生するのは，$O_i = 0$ のときに送信要求される場合と仮定できる．
また，もしメッセージ τ_i が時刻 0 よりも前に送信要求される場合には，τ_i は時刻 t_0 に終了するタイムスロットもしくはそれ以前に存在するタイムスロットで送信する機会が与えられることになり，いずれにしてもメッセージ τ_i は最大となるサイクル時間 t' よりも短い待ち時間で次の自送信タイムスロットが与えられ送信が開始されることになる．それゆえ，自 ECU の送信タイムスロットの送信開始時刻に送信要求される場合がクリティカルインスタントとなる．なお，クリティカルインスタントとは，メッセージが送信要求されてから送信完了するまでの時間が最も長くなる状況を指す．

4.2 自 ECU から送信されるメッセージの影響

あるメッセージの滞留時間を求めるために，そのメッセージが送信要求されるとすぐに送信メールボックスに格納されるという初期条件を用いている．しかしながら，この条件は，自 ECU から送信されるメッセージが 1 つしか

存在しない場合には成立するが，自 ECU から複数のメッセージが送信要求される場合には，その対象メッセージよりも先に送信要求されたメッセージ（先行メッセージ）や，後に送信要求されるメッセージ（後続メッセージ）により送信が遅延させられる可能性がある．この状況を扱うために，*busy period* [18] の概念を導入する．より具体的には，先行メッセージあるいは後続メッセージに対象メッセージの送信が遅延させられる状況を**表 1** のメッセージセットを用いて説明する．

表 1 の例は，各 ECU に 1 つずつ送信タイムスロットが割り当てられているものとし，$ECU_1(\theta_1) \rightarrow ECU_2(\theta_2) \rightarrow ECU_3(\theta_3) \rightarrow ECU_1(\theta_1) \rightarrow \ldots$ のように巡回される $turn = 3$ のネットワークである．また，各送信タイムスロットにおいて ACK フレームが送信される場合には，$C_i = 1$ とする．以降では，対象メッセージを τ_3 とする場合の遅れ時間について議論する．

表 1 メッセージセット例
Table 1 Example of a message set.

ECU_i	τ_i	i	C_i	T_i	O_i	θ_l
ECU_1	τ_1	1	1	25	0	θ_1
	τ_2	2	2	25	5	
	τ_3	3	3	25	15	
ECU_2	τ_4	4	4	25	0	θ_2
	τ_5	5	5	25	7	
ECU_3	τ_6	6	6	25	0	θ_3

4.2.1 先行メッセージにより送信が遅延させられる例

まず，対象メッセージ τ_3 の先行メッセージである τ_2 の送信要求時刻から始まる状況が τ_3 の遅れ時間に影響を与える場合を説明する．先行メッセージ τ_2 の送信要求時刻 5 において，最大の滞留時間が発生する場合，まず最初の自 ECU の送信タイムスロット θ_1 で ACK フレームが送信されるものとし，式 (1) の $t_0 = 1$ となる．次に，ECU_2 は τ_5，ECU_3 は τ_6 の送信要求が先行メッセージ τ_2 の送信要求時刻に発生し，続くタイムスロット θ_2 と θ_3 に τ_5 と τ_6 が送信されるものとすると，τ_3 の送信は τ_5 の送信時間 $C_5 = 5$ と τ_6 の送信時間 $C_6 = 6$ だけ阻害されることになる．この結果，τ_2 の送信要求時刻 5 から，$t = 1 + 5 + 6 = 12$ だけ経過した時刻 17 に，自身の送信タイムスロットである θ_1 の開始時刻が回ってくる．このとき，対象メッセージ τ_3 の送信要求時刻 15 を超えるため，先行メッセージ τ_2 が対象メッセージ τ_3 の送信に影響を与えることになる．この場合の ECU_1 の送信状況と各スロットの送信状況を**図 3** に示す．以降の送信状況は，図 3 で示されるように，τ_2 が送信された後，θ_2 と θ_3 ではそれぞれ ACK フレームが送信され，τ_3 は時刻 21 で送信を開始し時刻 24 で送信完了する．この結果，τ_3 の遅れ時間は送信完了時刻 24 から送信要求時刻 15 を引いた 9 となる．

一方，**図 4** は図 3 の送信状況と比べ，θ_2 に送信する

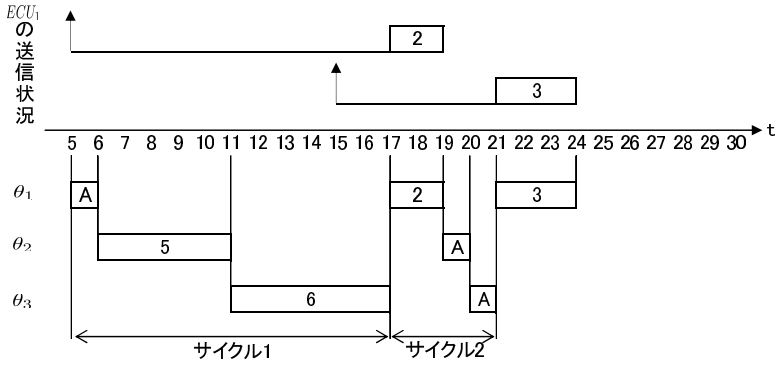


図 3 先行メッセージ τ_2 により送信が遅延させられる例 1
Fig. 3 Example 1 for interference from pre-requested message τ_2.

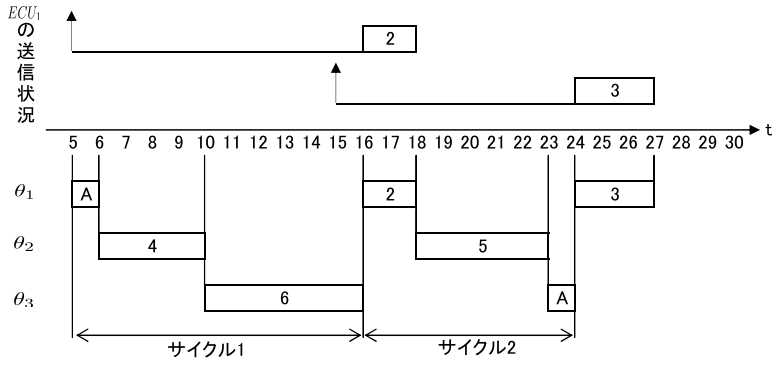


図 4 先行メッセージ τ_2 により送信が遅延させられる例 2
Fig. 4 Example 2 for interference from pre-requested message τ_2.

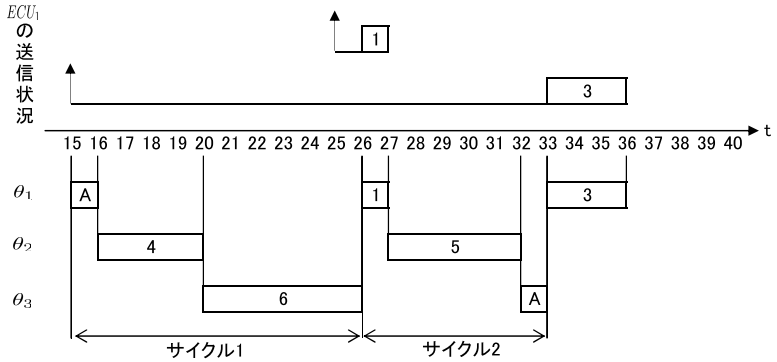


図 5 後続メッセージ τ_1 により送信が遅延させられる例
Fig. 5 Example for interference from post-requested message τ_1.

ECU_2 の送信状況が異なる場合であり，ECU_2 は τ_4 の送信要求時刻 0 が先行メッセージ τ_2 の送信要求時刻に発生する状況を表している．

この状況では，最初の θ_2 は τ_4，次の θ_2 は τ_5 が送信されるため，図 3 よりも遅れ時間が大きくなる．このときの τ_3 の遅れ時間は，送信完了時刻 27 から送信要求時刻 15 を引いた 12 となる．この結果，図 3 よりも図 4 の方が τ_3 の遅れ時間がより長くなる状況であるため，τ_3 にとっては図 4 がより悪い状況といえる．

4.2.2 後続メッセージにより送信が遅延させられる例

後続メッセージにより送信が遅延させられる状況とは，対象メッセージの送信が遅れ，後続の優先度の高いメッセージの送信要求時刻を超える場合に，後続の優先度の高いメッセージが対象メッセージより先に送信する状況を指す．この具体例について，表 1 の例を用いて説明する．τ_3 の送信要求時刻にて最大の滞留時間が発生する場合，**図 5** に示すように後続の τ_1 の送信要求時刻である時刻 25 を超えるため，τ_3 よりも優先度の高い τ_1 の送信が先に行われ，τ_3 の送信は次の送信タイムスロットまで遅延される．この結果，このときの τ_3 の遅れ時間は，送信完了時刻 36 から送信要求時刻 15 を引いた 21 となる．

4.3 最悪状況の決定

自 ECU が送信するすべてのメッセージの送信周期の LCM までに存在する全送信要求時刻から最大遅れ時間を導出すれば，必ずその中に最大遅れ時間が含まれている．しかしながら，あるメッセージを対象メッセージとして着目する場合には，*busy period* の定義から前述する先行メッセージと後続メッセージが対象メッセージに影響を与えうる範囲のみを調べつくせばよいことになる．このため，あるメッセージを対象メッセージとする場合には，まずその送信要求時刻から最大遅れ時間を導出し，以降，対象メッセージの送信要求時刻に影響を与えうる先行メッセージの送信要求時刻に遡り，その各時刻から最大遅れ時間を計算することにより，*busy period* に存在するすべての最大遅れ時間の候補を導出することができる．その結果，すべての最大遅れ時間の候補の中で最も大きい最大遅れ時間が対象メッセージの最大遅れ時間となり，その状況が最悪状況と決定できる．

5. 解析アルゴリズム

本章では，まず CAN とスケーラブル CAN の解析手法の違いについて説明したうえで，他 ECU から送信されるメッセージ群が対象メッセージに与える最大送信要求時間の導出方法を定義し，対象メッセージの最大遅れ時間を解析するアルゴリズムを導出する．

5.1 CAN とスケーラブル CAN の解析手法の違い

CAN とスケーラブル CAN ではそのスケジューリング方式の違いから，ある ECU が他の ECU に存在する対象メッセージを邪魔する時間の導出方法が異なる．

CAN の解析では，Maximum Interference Function（MIF）[19] を用いて，ある ECU が送信するある優先度以上の全送信メッセージが，他の ECU に存在するより優先度の低いメッセージの送信を同時に最大で邪魔する時間を導出する [16]．しかしながら，スケーラブル CAN の解析では各 ECU に与えられた送信タイムスロットで 1 メッセージずつしか送信できないために，MIF をそのまま適用することはできない．

より具体的には，ある送信タイムスロットの開始時刻は，初回送信タイムスロット以降そのタイムスロットが開始されるまでに存在する各タイムスロットの長さに依存しており，各タイムスロットの最大時間を積み上げて導出する必要がある．このため，ある送信タイムスロットの開始時刻は他の送信タイムスロットの送信状況に影響されるため，MIF のように一意に定まるものではない．さらに，ある送信タイムスロットを N 回目の自送信タイムスロットとする場合，過去の $N-1$ 回目の自送信タイムスロットまでに送信されたメッセージが M 個存在すると仮定すると，ある送信タイムスロットの開始時刻において送信要求される $M+1$ 番目のメッセージが与える最大影響時間を導出する必要がある．しかしながら，MIF ではある時刻までに送信

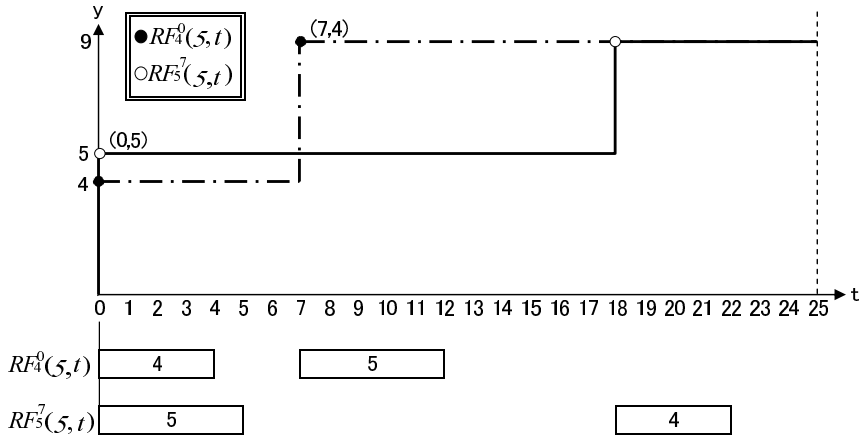


図 6 ECU_2 の MRF
Fig. 6 MRF in ECU_2.

要求される最大メッセージ数を考慮した最大影響時間の導出方法は検討されていない．

このため，我々は従来の手法である MIF を拡張し，ある時刻までに送信要求される最大メッセージ数を考慮した最大影響時間を導出するための手法として，メッセージ数を限定した Maximum Request Function（MRF）を提案する．

このメッセージ数を限定した MRF の効果は，ある時刻までに送信要求される最大メッセージ数を考慮した最大影響時間を導出することができるために，ある送信タイムスロットに送信されるメッセージの最大送信時間を導出することができる．より具体的には，$N-1$ 回目の送信タイムスロットの開始時刻における M 個のメッセージからの最大影響時間と N 回目の送信タイムスロットの開始時刻における $M+1$ 個のメッセージからの最大影響時間の差分から，N 回目の送信タイムスロットにおいて送信しうる $M+1$ 番目のメッセージの最大送信時間を導出する方法である．

5.2 Maximum Request Function

本節では，メッセージ数を限定した MRF のベースとなる MRF の導出方法の概要と表記方法を説明したうえで，具体例を用いて説明する．なお，MRF は MIF と同様の導出方法ではあるが，各メッセージの送信要求時刻において，その送信時間を累積する時間関数として表される点が MIF とは異なる点に注意されたい．

MRF の導出手順は，まず，ある ECU に存在する対象メッセージよりも優先度の高いメッセージ群の送信周期の最小公倍数（LCM）までに存在するすべての送信要求時刻からの送信シーケンスを表す Request Function（RF）を作成する．次に，作成されたすべての RF を重ね合わせ，各時刻において最も送信時間が大きくなる状況が MRF となる．

ここで，RF とは，優先度 i 以上のあるメッセージ τ_j の送信要求時刻 st から開始される送信メッセージ群の送信要求状況を表す累積的な時間関数であり，ある時刻 t におけるその累積的なメッセージの送信時間を $RF_j^{st}(i,t)$ で表すものとする．また，ある時刻 t における ECU_n の優先度 i 以上の MRF は，各 RF を重ね合わせたときの最も大きい送信時間を持つ線分をつなぐことで導出でき，グラフの横軸を時刻 t，縦軸を累積的な送信時間 $y = MRF_j^{st}(i,t)$ とすると，各凸点の座標 (t_k, y_k) の配列とその周期 T_{LCM} を用いて，以下の式 (2) で表すものとする．

$$MRF_n(i,t)=\{(t_1,y_1),(t_2,y_2),\ldots,(t_o,y_o),T_{LCM}\} \quad (2)$$

次に，表 1 に存在する ECU_2 の τ_4 と τ_5 を用いて，MRF の導出手順と表記方法をより具体的に説明する．前述する導出手順から，まず，ECU_2 の τ_4 と τ_5 の送信周期の LCM は 25 であり，その中に存在するすべての送信要求時刻は τ_4 の時刻 0 と τ_5 の送信要求時刻 7 の 2 つ存在する．この 2 つの送信要求時刻から始まる RF は，**図 6** の下図に示す送信シーケンス $RF_4^0(5,t)$ と $RF_5^7(5,t)$ であり，その各時刻における送信時間を図 6 の上図のグラフで表現する．なお，図 6 の横軸は各送信シーケンスの開始時刻からの相対的な経過時間を表し，上図の縦軸は送信要求されたメッセージによる累積的な送信時間を表すものである．この結果，式 (2) より，図 6 の MRF は $MRF_2(5,t)=\{(0,5),(7,4),25\}$ と表記できる．

また，提案手法であるメッセージ数を限定した MRF を導出するために用いられるメッセージ数を限定した RF は，その開始時刻以降に存在する送信要求メッセージ k 個に限定された累積的な時間関数 $RF_j^{st}(i,k,t)$ で表すものとする．

ある ECU_n の優先度 i 以上の送信要求メッセージ k 個に限定された MRF を $MRF_n(i,k,t)$ と表し，各凸点の座標 (t_k, y_k) の配列およびその周期性はないため周期を 0 として，以下の式 (3) のように表すものとする．

$$MRF_n(i,k,t)=\{(t_1,y_1),(t_2,y_2),\ldots,(t_o,y_o),0\} \quad (3)$$

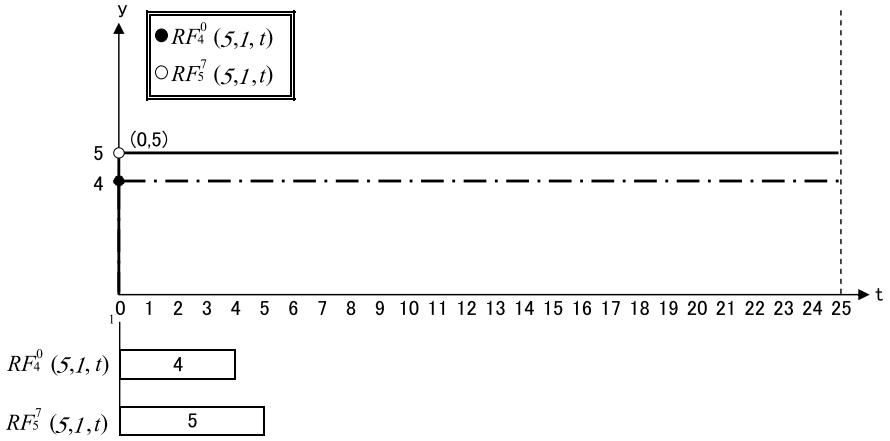


図 7 ECU_2 に存在する 1 メッセージ分の MRF
Fig. 7 MRF for a message in ECU_2.

5.3 スケーラブル CAN における他 ECU が最大で送信要求する時間の導出方法

スケーラブル CAN の各タイムスロットでは 1 つのメッセージのみしか送信できないため，各タイムスロットの最大送信時間を導出するために，メッセージ数を限定した MRF を用いる．そこで本節では，メッセージ数を限定した MRF の導出方法と，この手法を用いてある ECU がある送信タイムスロットに送信するメッセージの最大送信時間の導出方法を具体例を用いて説明する．そのうえで，ある対象メッセージの最大遅れ時間を導出する方法についても説明する．

まず，ある ECU が与える最大影響時間の導出方法の前提となるメッセージ数を限定した MRF の導出方法を説明する．この導出方法の具体例として，表 1 に存在する ECU_2 の τ_4 と τ_5 を用いて説明する．この ECU_2 における 1 つのメッセージの MRF を導出する場合，各 RF の送信シーケンスを 1 メッセージに限定することで，1 つのメッセージから送信要求される最大時間を導出する．より具体的には，**図 7** に示すように，各 RF の送信シーケンスにおいて先頭の 1 メッセージのみに限定して導出する．その結果，この 1 メッセージ分に限定された MRF には周期性はないため，周期を 0 とし，$MRF_2(5,1,t) = \{(0,5),0\}$ と表すものとする．同様に 2 メッセージ分の MRF は，図 6 と同様の送信シーケンスをとるが，その周期を 0 とし，$MRF_2(5,2,t) = \{(0,5),(7,4),0\}$ と表される．以降，同様の方法を用いて，各 RF の開始時刻以降に存在する k 個分の送信メッセージのシーケンスを作成することで，メッセージ数 k 個に限定した MRF を導出する．

また，ある ECU がある送信タイムスロットに送信するメッセージの最大送信時間は，以下の手順で導出する．ある ECU を ECU_n と仮定する場合，まず，解析の開始時において，ECU_n の送信メッセージ数 $sendcnt_n$ と ECU_n の最後の送信タイムスロットにおける累積的な送信時間 $rftime_n$ を 0 に初期化する．次に，初期化以降の ECU_n の初回送信タイムスロットの開始時刻 t_1 において，送信メッセージ中の最低優先度を i として，その累積的な送信時間 $y_{sendcnt_n+1} = MRF_n(i, sendcnt_n + 1, t_1)$ を計算し，もし $y_{sendcnt_n+1}$ と $rftime_n$ が異なる（$y_{sendcnt_n+1} > rftime_n$）場合には，現時刻までの累積的な送信時間 $y_{sendcnt_n+1}$ と $rftime_n$ との差分をそのタイムスロットの送信時間とする．一方，$y_{sendcnt_n+1}$ と $rftime_n$ が同じ値である場合には，前回の自送信タイムスロット以降に送信要求されたメッセージが存在しないものと判断し，ACK フレームが送信されるものとする．最後に，次のタイムスロットの計算に進む前に，$rftime_n$ に $y_{sendcnt_n+1}$ の値を保持し，もしこのタイムスロットで ACK フレーム以外の新たなメッセージの送信が行われた場合には，$sendcnt_n$ を 1 つインクリメントする．以降，このような手順を繰り返すことにより，各 ECU の送信タイムスロットの開始時刻において新たな送信メッセージが存在するかどうかを判定し，各送信タイムスロットにおける最大送信時間を導出する．

最後に，4.2.2 項の具体例を用いて，ECU_2 の送信メッセージ τ_4 と τ_5 が対象メッセージ τ_3 の送信を最大で邪魔する時間に着目しながら，メッセージ τ_3 の遅れ時間を導出する方法を説明する．この対象メッセージ τ_3 の送信要求時刻 15 で最悪状況が発生する場合，まず解析の開始時に ECU_2 の送信メッセージ数 $sendcnt_2$ と ECU_2 の累積的な送信時間 $rftime_2$ を 0 に初期化する．次に，初回送信タイムスロット θ_1 に ACK フレーム（$t_0 = 1$）が送信された後，続く θ_2 の開始時刻 16 では $y_{sendcnt_2+1} = MRF_2(5, sendcnt_2 + 1, t)$ より $sendcnt_2 = 0$ を代入した $y_1 = MRF_2(5,1,t)$（図 7 参照）を用いて，$t = 1$ だけ経過したときの値 $y_1 = 5$ を得る．このとき，$rftime_2 = 0$ であるため，$y_1 = 5$ との差分となる時間 5 のメッセージが最大で送信されるものとする．

また，以降の送信タイムスロットの送信状況は**図 8** に示す状況となるため，送信タイムスロット θ_2 の 2 回目の送信タイムスロット（図 8 より，時刻 28）における送信メッセージは，$sendcnt_2 = 1$ より，2 メッセージ分の MRF

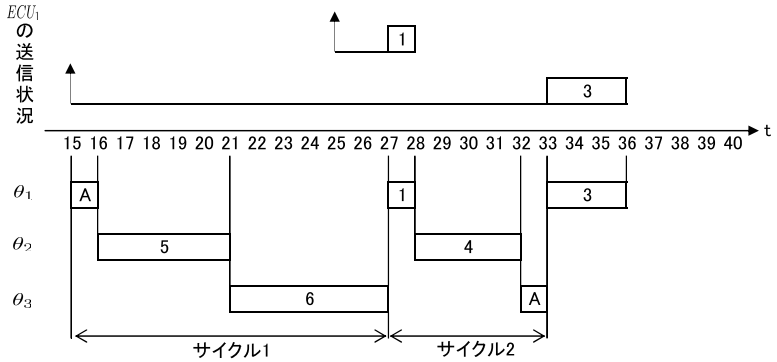


図 8 提案手法の具体例
Fig. 8 Example for our proposed analysis.

である $MRF_2(5,2,t)$ を用いて導出する．より具体的には，この送信タイムスロットの開始時刻 28 は，最悪状況である時刻 15 から 13 だけ経過した時刻であるため，図 6 より $y_2 = MRF_2(5,2,13) = 9$ を得る．また，このときの $rftime_2$ は 5 であるため，$y_2 = 9$ と $rftime_2 = 5$ との差分 4 を最大長とするメッセージの送信が行われるものと導出できる．

このような計算を各送信タイムスロットの開始時刻で繰り返した結果，図 8 に示す送信状況となり，この状況における対象メッセージ τ_3 の遅れ時間はその送信完了時刻 36 から送信要求時刻 15 を引いた 21 となる．この解析結果より，本解析手法で解析される最悪状況を示した図 8 は，実際に発生しうる最悪状況を示した図 5 と θ_2 における送信メッセージ τ_4 と τ_5 の送信順序が異なるものの，導出される最大遅れ時間は同じ結果となる．

5.4 最大遅れ時間解析アルゴリズム

前述した計算方法を用い，ある 1 つのスケーラブル CAN メッセージの最大遅れ時間解析アルゴリズムを以下に示す．

(1) 対象となるメッセージを送信する自 ECU に割り当てられるある送信タイムスロットを初回送信タイムスロットとする．
(2) 対象メッセージの初回送信要求時刻をクリティカルインスタントの候補とする．
(3) 初回送信タイムスロットで，CAN の解析におけるブロッキング時間 [21] と同様に，1 つのメッセージに送信を邪魔されるものとし，自 ECU から送信される最も送信時間が長いメッセージの送信時間 t_0 に邪魔されるものとする．
(4) 次のタイムスロット以降，そのタイムスロットが他 ECU が送信する場合は 5.3 節で提案する方法により，そのタイムスロットに送信する ECU の MRF を用いて最大送信時間を導出し，遅れ時間に加算する．一方，そのタイムスロットが自 ECU の送信タイムスロットである場合，その時刻までに送信要求されたメッセージのうち最も優先度の高いメッセージの送信時間が遅れ時間に加算される．このとき，対象メッセージが送信されるまで本手順 (4) を繰り返し，最大遅れ時間の候補を導出する．
(5) 以降，自 ECU の全送信メッセージの送信周期の LCM 内に存在する対象メッセージとそれに影響を与える先行メッセージの各送信要求時刻をクリティカルインスタントの候補とし，手順 (3) に戻りすべての場合の最大遅れ時間の候補を求める．
(6) 自 ECU に複数の送信タイムスロットがあり，いまだ初回送信タイムスロットとして解析されていないものがある場合，それを初回送信タイムスロットとし，手順 (2) に戻り，全クリティカルインスタントの候補に対し解析を行う．その結果，すべての最大遅れ時間の候補のうち，最も大きい遅れ時間を持つ候補を最悪状況とし，その遅れ時間を最大遅れ時間とする．

6. 評価

本評価では，まず提案するスケーラブル CAN の最大遅れ時間解析手法の評価として，全数探索プログラムを用いて提案手法の妥当性の評価を行う．そのうえで，CAN とスケーラブル CAN プロトコルの伝送能力の違いを最大遅れ時間解析を用いて比較評価する．

6.1 対象と前提

本評価では既存する CAN バス上の ECU がスケーラブル CAN へと移行する場合を想定し，実際の車両で使われているメッセージセットから送信メッセージの負荷率の合計が大きい 6 つの ECU を抽出し，それら 6 つの ECU が 1 つのバス上に配置されたメッセージセットを用いる．なお，送信メッセージの負荷率とは，ある送信メッセージ τ_i の最大送信時間 C_i をその送信周期 T_i で割り百分率で表した値である．この抽出されたメッセージセットは全 48 メッセージ，CAN500 kbps におけるバス負荷率は約 45%であり，このメッセージセットのメッセージ量を 1 倍量と呼ぶ．また，このメッセージセットは，いくつか同値の送信周期が与えられているメッセージが存在するものの，基本的に

表 2 提案する解析手法と全数探索の計算時間の比較
Table 2 Comparison of computation times of our proposed analysis and the simulation.

	提案手法	全数探索
3ECU に限定したメッセージセット	4.562 秒	568.504 秒
1 倍量のメッセージセット	6.530 秒	82,827.355 秒（約 1 日）

表 3 解析手法と全数探索の誤差の比較
Table 3 Comparison of accuracy of our proposed analysis and the simulation.

	誤差のあるメッセージ数	誤差の平均	誤差の最大
3ECU に限定したメッセージセット	14 個	0.099%	0.320%
1 倍量のメッセージセット	47 個	0.539%	2.200%

は送信周期が短い順に高い優先度が与えられているという特徴を持つ．

以降の評価では，この 1 倍量のメッセージセットを X 倍量したものを用いており，10 倍量のメッセージセットと呼ぶ場合には，この基準となる 1 倍量のメッセージセットのメッセージ数を 10 倍したものである．

本評価環境として，CPU：Intel® Core2 Quad 2.83 GHz，メモリ：3.25 GB の PC を用いた．

6.2 提案手法の妥当性評価

本提案手法では，対象メッセージの送信を阻害する他 ECU から送信されるメッセージ群の送信状況を MRF を用いて抽象化することで，計算量を減らしつつその最大送信要求時間を導出する方法を提案した．しかしながら，この抽象化は，文献 [20] で指摘されるのと同様に計算量を抑えることができる一方，実際に発生する最大遅れ時間よりも解析結果が大きくなる場合が存在する．そこで，我々は本提案手法の妥当性を評価するために，スケーラブル CAN の正確な最大遅れ時間を求める全数探索プログラムを開発し，その結果と比較することで本提案手法の計算時間と計算誤差について評価する．

本評価のために開発した全数探索プログラムは，すべての ECU に存在するすべてのメッセージの送信要求時刻から始まる状況に対し各メッセージの遅れ時間を計算することで，全メッセージの正確な最大遅れ時間を導出するプログラムである．このため，全数探索プログラムでは各メッセージの正確な最大遅れ時間が導出できるものの，計算時間が膨大となることが懸念される．そこで，本評価では，抽出された 1 倍量のメッセージセットと，1 倍量のメッセージセットから送信メッセージの負荷率の合計の大きい 3 つの ECU のみに限定したメッセージセット（以降では，3ECU に限定したメッセージセットと呼ぶ）の 2 つを用いて評価する．この 3ECU に限定したメッセージセットは全 34 メッセージ，CAN500 kbps におけるバス負荷率は約 40%となる．なお，本評価では，各 ECU に 1 つずつ送信タイムスロットが割り当てられるものとし，伝送速度 500 kbps の場合について評価する．

6.2.1 計算時間の比較

提案手法と全数探索に要する計算時間の比較結果は**表 2** に示すとおりであり，提案手法では，全数探索と比べ十分に短い計算時間で解析できることを示している．特に，1 倍量のメッセージセットを用いた場合には全数探索に要する時間が約 1 日かかることから，ECU 数やメッセージ数が増えた場合には，提案手法を用いることで計算時間を低減させることができる．

6.2.2 計算誤差の比較

提案手法と全数探索から得られる最大遅れ時間の計算誤差の評価結果は**表 3** に示すとおりであり，誤差があるメッセージ数とその誤差の平均および最も誤差が大きいメッセージの誤差（誤差の最大）を示している．なお，誤差の平均と誤差の最大は，提案手法と全数探索から得られた最大遅れ時間の差をそのメッセージの送信周期で割り，百分率で表したものである．この理由は，車載ネットワークの分野では各メッセージの送信周期がデッドラインとして与えられることが多いため，そのデッドラインに対してどの程度の誤差が存在するのかを示すために本指標を用いるものとした．

本評価結果から，3ECU に限定したメッセージセットでは，34 メッセージ中 14 メッセージに誤差が存在し，残る 20 メッセージは正確な値が得られた．この誤差が存在する 14 メッセージの誤差の平均は 0.099%であり，このうち誤差の最大は 0.320%であった．この結果，このメッセージセットでは，解析上の誤差は 1%未満であり，ほとんど正確な値が得られたといえる．また，1 倍量のメッセージセットにおいては，48 メッセージ中 47 メッセージに誤差が存在し，その誤差の平均は 0.539%，誤差の最大は 2.200%であった．スケーラブル CAN の転送速度 500 kbps，ECU6 個，メッセージ数 48 個のメッセージセット（バス負荷率約 45%程度）を用いた場合において，最大遅れ時間の誤差が最大で約 2.200%程度悲観的ではあるものの，この誤差の最大は 1 つのメッセージの最大送信時間内に収まっている．

表 4 本評価で用いるスロット割当ての設定内容
Table 4 Settings of assigned time-slot for Scalable CAN.

	(A) 方式 各 1 スロット式	(B) 方式 ドント式	(C) 方式 最短送信周期のメッセージ数式
ECU1	1	1,7,11,15	1,7,11,15,19
ECU2	2	2,8,12,16	2,8,12,16
ECU3	3	3,9,13	3,9,13,17,20,22
ECU4	4	4	4
ECU5	5	5	5
ECU6	6	6,10,14	6,10,14,18,21
合計 *	6 スロット	16 スロット	22 スロット

* 1 サイクルあたりのスロット数を表す．

6.3 CAN とスケーラブル CAN の比較

6.3.1 比較方法

本評価では，前述するスケーラブル CAN の最大遅れ時間解析手法を用いて，CAN とスケーラブル CAN プロトコルの伝送能力の違いを比較する．本評価では，まず，伝送効率の違いを比較するため，伝送速度とメッセージ量が同じ場合の比較を行う．次に，スケーラブル CAN では CAN と比べて最大伝送速度が改善されることから，伝送速度に応じたデータ量を送信した場合の比較を行う．なお，本論で比較対象として用いる CAN メッセージの最大遅れ時間解析手法は，文献 [16] で提案されたオフセット付き CAN メッセージの最大遅れ時間の解析手法に対し，スケーラブル CAN と同等の解析条件とするため，ECU 内の優先度逆転の考慮を削除したうえで，文献 [21] の問題を考慮した解析方法を用いる．

本評価では各メッセージにオフセットを付与する場合と付与しない場合 (つまり，オフセットが 0 の場合) の 2 パターンのメッセージセットを用いて評価を行う．なお，本評価で使用するオフセットの決定方法は，文献 [22] の手法を用いる．

次に，評価指標は全メッセージの最大遅延率の平均とする．最大遅延率とは，最大遅れ時間解析から得られる最大遅れ時間をそのメッセージの送信周期で割り百分率で表した値である．最大遅延率を用いる理由は，各メッセージを評価するには最大遅れ時間の方が適切ではあるが，各メッセージの送信周期はメッセージごとに様々な値に設定されているため，メッセージセット全体を評価するためには，全メッセージの最大遅延率の平均を用いる方が適切と考えるためである．また，もう 1 つの理由としては，実際にメッセージセットを決定する際には，少なくともどのメッセージも最大遅れ時間が送信周期を超えないように設計されるため，すべてのメッセージが送信周期を超えていないことを確認するためでもある．

6.3.2 スケーラブル CAN の送信タイムスロット割当て方法

本評価では，以下の 3 種類の送信タイムスロットの割当て方法を用いた．なお，具体的なタイムスロットの配置については**表 4** に示すとおりである．

(A) 各 1 スロット式 各 ECU あたり 1 スロットずつ割り当てる方法であり，1 サイクルあたりのスロット数は ECU 数と同様，6 スロットとなる．

(B) ドント式 1 サイクルあたりのスロット数を 16 とし，各 ECU に 1 スロットずつ割り当てたうえで，残りの 10 スロットは各 ECU の送信メッセージの負荷率の合計に応じて分配する方法をとる．

(C) 最短送信周期のメッセージ数式 メッセージセットの中で送信周期の短いメッセージの最大遅延率が大きくなることが予想できる．このため，メッセージセット中の最短周期となる送信メッセージの数に応じて，各 ECU にスロットを付与する．なお，1 サイクルあたりのスロット数は 22 スロットとなる．

6.3.3 CAN とスケーラブル CAN の比較結果

ここで，オフセットを付けない場合と付けた場合の評価結果をそれぞれ**表 5** と**表 6** に示す．

表 5 の結果より，オフセットのない場合には，各プロトコルで伝送速度が同じ場合を比較すると，スケーラブル CAN の方が巡回するためのオーバヘッドやフレームの拡張のため，自明なことではあるが，CAN と比べて相対的に最大遅延率の平均は長くなる．一方，スケーラブル CAN で伝送速度に応じてメッセージ量を増やした場合の比較として，スケーラブル CAN の (B) 方式に着目すると，伝送速度 500 kbps，メッセージ量を 1 倍量とした場合の最大遅延率の平均は 23.16% であるが，伝送速度 5 Mbps，メッセージ量を 10 倍量とした場合には 17.72% に改善される．これは，スケーラブル CAN の伝送速度が上昇したことにより，初回送信タイムスロットにおける邪魔時間 t_0 および次の自送信タイムスロットが回ってくるまでの時間が減少した結果，メッセージセット全体の最大遅延率が改善されたことを表している．また，CAN とスケーラブル CAN を比較する場合，CAN の伝送速度を 500 kbps，メッセージ量を 1 倍量とする場合の最大遅延率の平均 11.78% とスケーラブル CAN の伝送速度を 5 Mbps，メッセージ量を 10 倍量，スロット割当て方法を (B) 方式とする場合の最大遅延

表 5 オフセットがない場合の従来 CAN とスケーラブル CAN の最大遅延率の平均の比較
Table 5 Comparisons of the worst-case response delay avarage for CAN messages and Scalable CAN messaes without offsets.

伝送速度	メッセージ量	オフセット	CAN	スケーラブル CAN		
				(A) 方式	(B) 方式	(C) 方式
500 kbps	1 倍量	なし	11.78%	23.49%	23.16%	24.62%
5 Mbps	10 倍量	なし	10.39%*	19.48%	17.72%	18.73%
5 Mbps	9 倍量	なし	-	17.55%	15.97%	16.90%
5 Mbps	8 倍量	なし	-	15.61%	14.21%	15.07%
5 Mbps	7 倍量	なし	-	13.68%	12.47%	13.22%
5 Mbps	6 倍量	なし	-	11.76%	10.75%	11.39%

* 実際には CAN を 5 Mbps で実行することはできないが，仮に 5 Mbps で実行できるものと仮定した場合の参考値である．

表 6 オフセットがある場合の従来 CAN との最大遅延率の平均の比較
Table 6 Comparisons of the worst-case response delay avarage for CAN messages and Scalable CAN messaes with offsets.

伝送速度	メッセージ量	オフセット	CAN	スケーラブル CAN		
				(A) 方式	(B) 方式	(C) 方式
500 kbps	1 倍量	あり	6.01%	9.37%	10.91%	11.28%
5 Mbps	10 倍量	あり	0.93%*	1.06%	1.18%	1.30%

* 実際には CAN を 5 Mbps で実行することはできないが，仮に 5 Mbps で実行できるものと仮定した場合の参考値である．

率の平均 17.72% を比較した結果，スケーラブル CAN の最大遅延率の平均は CAN と比べて約 6% 程度の遅延の増加に収められる．

また，伝送速度 5 Mbps のスケーラブル CAN が伝送速度 500 kbps の CAN の遅延率（表 5 中の 11.78%）に収めるために，メッセージ量を 10 倍量から 9 倍量，8 倍量と徐々に減らした場合を解析した．この結果，表 5 に示すとおり，スケーラブル CAN ではメッセージ量が 6 倍量のときに CAN の最大遅延率の平均以下となった．このため，本評価に用いたメッセージセットを用いる場合，CAN の時間制約を守りつつスケーラブル CAN へ移行するには，伝送速度 5 Mbps のスケーラブル CAN ではメッセージ量を 6 倍量まで増加させることができるといえる．

次に，オフセットを付与した場合の結果を表 6 に示す．表 6 の結果より，オフセットを付与した場合でもオフセットを付与しない場合と同様に，同一条件下においては，スケーラブル CAN よりも CAN の方が最大遅延率の平均が低く良い結果となる．しかしながら，スケーラブル CAN の伝送速度を 5 Mbps，メッセージ量を 10 倍量，スロット割当て方法を (A) 方式とする場合の最大遅延率の平均は 1.06% であり，CAN の 500 kbps の結果である 6.01% よりも良い結果となる．これは，スケーラブル CAN では伝送速度が上がったことにより 1 つのメッセージから送信を阻害される時間が減少したうえで，各メッセージがオフセットにより分散されるため，最大遅延率が改善できることを示しており，スケーラブル CAN により高速化した場合のメリットといえる．

6.4 オフセットとスロット割当て方法の有効性

スケーラブル CAN におけるオフセットとスロット割当て方法の有効性について考察する．

まず，オフセットの有効性については，表 5 と表 6 のオフセット有無の結果を比較すると，すべての条件においてオフセットを付与した方が最大遅延率が低減できており，オフセットの効果が確認できた．

次に，スロット割当て方法の有効性については，表 5 および表 6 の結果より，多少の差はあるものの，実験に使用した 3 種類のスロット割当て方法についてどの方法が良いとはいえない．このため，最適なスロットの割当て方法については今後の課題とする．

6.5 10 倍量のメッセージセットを用いたときの解析時間

スケーラブル CAN の 5 Mbps 時における解析手法の計算時間について考察する．6.2.1 項では，500 kbps 時の全数探索と提案手法の計算時間と計算誤差の比較を実施した．しかしながら，実際の使用環境である 5 Mbps 時にはメッセージ数が 10 倍量に増加されるため，提案手法においてもどの程度計算時間が必要になるのか懸念される．このため，本節では 5 Mbps 時における提案手法の計算時間を**表 7** に示す．なお，この結果は各 ECU にタイムスロットを 1 つずつ割り当てた (A) 方式での結果のみを示している．

表 2 と表 7 の結果を比較すると，メッセージ量が 10 倍になったことにより，解析にかかる時間が 10 倍以上になっている．これは，解析対象となるメッセージ数が増えたこと，および解析に使用するメッセージ数を限定した MRF

表 7 5 Mbps 時における提案する解析手法の計算時間
Table 7 Computation times of our proposed analysis in 5 Mbps.

	オフセットなし	オフセットあり
10 倍量のメッセージセット	2,954.113 秒	3,623.648 秒

の数が増えたことに起因していると考えられる．また，表 7 において，オフセットなしよりもオフセットありの方が計算時間が増加しているのは，オフセットにより各メッセージの送信要求時刻が分散されたため，解析対象となるメッセージのクリティカルインスタントの候補が増えたことに起因すると考えられる．

6.6 FlexRay との性能比較

スケーラブル CAN と FlexRay との性能比較について考察すると，どちらも同等の転送速度で動作することが可能であり，CAN の置き換えを考える場合においては FlexRay のダイナミックセグメントを使用する方法があげられる．FlexRay のダイナミックセグメントではサイクル時間を守る範囲内で与えられた送信タイムスロットにメッセージの送信を行うことができる．しかしながら，FlexRay ではサイクルをまたぐメッセージの送信は行えないため，サイクル境界付近では通信路上の無駄なアイドル時間が増えるものと考えられる．このため，定性的ではあるが，スケーラブル CAN と FlexRay を同一転送速度で実行する場合には，メッセージセットには依存するものの，ほとんど同程度の伝送路効率があると考えられる．なお，この定量的な比較評価については今後の課題とする．

7. まとめ

車載ネットワークの標準である CAN の最大伝送速度が 1 Mbps に制約されることから，我々は新しい高速な車載ネットワーク向けプロトコルとしてスケーラブル CAN を提案している．本論では，スケーラブル CAN の性能評価として，最大遅れ時間解析手法を提案し，CAN との伝送能力の違いを比較し評価した．本論の評価結果より，スケーラブル CAN では CAN に比べてオーバヘッドが存在するために同一条件下においては遅延率が増加するものの，遅延率を抑えつつ十分なメッセージ量の送信が可能であることを示した．特に，スケーラブル CAN は CAN では実現できない伝送速度で動作できるため，これまでに CAN では扱うことができなかったメッセージ量を扱える点が有効である．また，オフセットを付与したメッセージセットを使用する場合には，現状の伝送速度 500 kbps の CAN よりも低い遅延率で回線容量に応じたメッセージ量を送信できることからも，スケーラブル CAN の有効性を示している．

今後の課題として，本論では最適なオフセットとスロットの割当て方法については議論しなかったため，検討し議論する必要がある．また，スケーラブル CAN の今後の課題としては，FPGA 上に通信コントローラを実装し評価を進めており，特に耐ノイズ性能の評価を通じて，通信バス上のビットエラー発生時にリカバリ動作が正しく振る舞うことを検証する必要がある．さらに実用化に向けては，実車環境下における評価が必要となるため，これらが今後の課題である．また，スケーラブル CAN と FlexRay の伝送能力の比較評価についても今後の課題とする．

参考文献

[1] International Organization for Standardization, Road vehicles: Controller area network (CAN). Part1: Data link layer and physical signaling, ISO IS11898-1 (2003).

[2] Nolte, T., Hansson, H. and Bello, L.L.: Automotive communications – past, current and future, *Proc. 10th IEEE International Conference on Emerging Technologies and Factory Automation* (*ETFA '05*), Vol.1, pp.985–992 (2005).

[3] Cena, G. and Valenzano, A.: FastCAN: A high performance enhanced CAN-like network, *IEEE Trans. Ind. Electron.*, Vol.47, No.4, pp.951–963 (2000).

[4] Compton, B.T.: Goldilocks Serial Communication Protocol, SAE World Congress (2008).

[5] 倉地 亮，高田広章，手嶋茂晴，宮下之宏：10 MbpsCAN プロトコルの設計と評価，情報処理学会論文誌，Vol.50, No.11, pp.2643–2653 (2009).

[6] Kurachi, R., Takada, H., Nishimura, M. and Horihata, S.: A New High-Speed Bus Topology LAN Protocol Compatible with CAN, *SAE 2011 World Congress* (2011).

[7] FlexRay Consortium, available from ⟨http://www.flexray.com/⟩.

[8] 倉地 亮，高田広章，西村政信，堀端啓史：スケーラブル CAN プロトコルの設計と評価，自動車技術会論文集，Vol.43, No.2, pp.509–514 (2012).

[9] Tindell, K. and Burns, A.: Guaranteed Message Latencies for Distributed Safety-Critical Hard Real-Time Networks, Technical Report YCS 229, Department of Computer Science, University of York (1994).

[10] Tindell, K., Burns, A. and Wellings, A.: Calculating Controller Area Network (CAN) message response times, *Control Engineering Practice*, Vol.3, no.8, pp.1163–1169 (1995).

[11] Punnekkat, S., Hansson, H. and Norström, C.: Response time analysis under errors for CAN, *Proc. 6th IEEE Real-Time Technology and Applications Symposium* (*RTAS*), Washington DC, IEEE Computer Society Press (2000).

[12] Nolte, T.: Reducing pessimism in CAN response time analysis, Mälardalen University, Mälardalen Real-Time Research Centre, Technical Report MDH-MRTC-51 (2002).

[13] 飯山真一，高田広章：システム構成を考慮した CAN の最大遅れ時間解析手法，情報処理学会論文誌：コンピューティングシステム，Vol.45, No.SIG1(ACS 4), pp.66–76 (2004).

[14] Grenier, M., Goossens, J. and Navet, N.: Near-optimal fixed priority preemptive scheduling of offset free systems, *Proc. 4th International Conference on Network and Systems* (*RTNS 2006*), Poitiers, France (2006).

[15] Grenier, M., Havet, L. and Navet, N.: Pushing the lim-

its of CAN – scheduling frames with offsets provides a major performance boost, *Proc. 4th European Congress Embedded. Real Time Software* (*ERTS 2008*) 2008.
[16] 飯山真一，冨山宏之，高田広章，城戸正利，細谷伊知郎：オフセット付き CAN メッセージの最大遅れ時間解析，情報処理学会論文誌：コンピューティングシステム，Vol.45, No.SIG11(ACS 7), pp.455–464 (2004).
[17] Du, L. and Xu, G.: Worst Case Response time analysis for CAN messages with offsets. *Proc. International Conference on Vehicular Electronics and Safety* (*ICVES 2009*), Pune, India, Nov. 10-12 (2009).
[18] Lehoczky, J., Sha, L. and Ding, Y.: The rate monotonic scheduling algorithm:exact characterization and average case behavior, *Proc. 10th IEEE Real-Time Systems Symposium*, pp.166–171 (1989).
[19] Takada, H. and Sakamura, K.: Schedulability of generalized multiframe task sets under static priority assignment, *Proc. Real-Time Computing Systems and Applications*, pp.80–86 (1997).
[20] 倉地 亮，陳 暘，高田広章：オフセット付き CAN メッセージの正確な最大遅れ時間解析，情報処理学会論文誌，Vol.52, No.8, pp.2431–2440 (2011).
[21] Davis, R.I., Burns, A., Bril, R.J. and Lukkien, J.J.: Controller Area Network (CAN) Schedulability Analysis: Refuted, Revisited and Revised, *Real-Time Systems*, Vol.35, No.3, pp.239–272 (2007).
[22] 陳 暘，倉地 亮，曾 剛，高田広章：CAN メッセージのオフセット決定手法，情報処理学会論文誌，Vol.52, No.7, pp.2245–2255 (2011).

付　　録

A.1 従来 CAN とスケーラブル CAN の最大送信時間の差

従来 CAN とスケーラブル CAN のデータフレームの差は文献 [6] で示したとおりである．このため，従来 CAN とスケーラブル CAN ではフレームの最大送信時間が異なる．

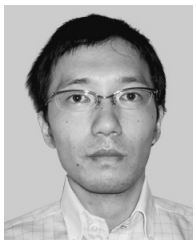
ここで 1 ビットあたりの送信時間を τ_{bit}，データ長（1〜8 バイト）を s_i とすると，CAN メッセージの最大送信時間は以下の式 (A.1) を用いて導出できる．

$$C_i = \left(\left\lfloor \frac{34 + 8s_i - 1}{4} \right\rfloor + 47 + 8s_i \right) \tau_{bit} \qquad \text{(A.1)}$$

一方，スケーラブル CAN の最大送信時間は，メッセージが送信されるタイムスロットの最大時間と見なして解析するため，ACK 情報フィールド長を $acklen$，スロット番号フィールド長を $slotlen$，スロット境界からの送信開始時刻を $acplen$，伝送路上の許容する伝送路遅延ビット数を $delay$ とすると，1 フレームの送信にかかるタイムスロットの最大時間は以下の式 (A.2) を用いて導出する．

$$C_i = \left(\left\lfloor \frac{34 + acklen + slotlen + 8s_i - 1}{4} \right\rfloor + 45 + 8s_i + acklen + slotlen + acplen + delay \right) \tau_{bit} \qquad \text{(A.2)}$$

本論の評価では，$acklen$ を表 4 の 1 サイクルあたりのスロット数で設定し，$acplen = 5$ ビット，$slotlen = 5$ ビット，$delay = 2$ ビットとして計算している．また，ACK フレームの最大送信時間は式 (1) を $s_i = 0$ として導出するものである．



倉地 亮　（正会員）

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研究員．2000 年東京理科大学基礎工学部電子応用工学科卒業後，アイシン・エィ・ダブリュ株式会社にてカーナビゲーションシステムの開発に従事．2007 年東京理科大学専門職大学院総合科学技術経営研究科技術経営専攻修了後，同年より現職．車載ネットワークに関する研究に従事．

高田 広章　（正会員）

名古屋大学大学院情報科学研究科情報システム学専攻教授．1988 年東京大学大学院理学系研究科情報科学専攻修士課程修了．同専攻助手，豊橋技術科学大学情報工学系助教授等を経て，2003 年より現職．2006 年より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長を兼務．リアルタイム OS，リアルタイムスケジューリング理論，組込みシステム開発技術等の研究に従事．オープンソースの ITRON 仕様 OS 等を開発する TOPPERS プロジェクトを主宰．博士（理学）．IEEE，ACM，電子情報通信学会，日本ソフトウェア科学会，自動車技術会各会員．

西村 政信

株式会社オートネットワーク技術研究所ネットワーク研究部．車載ネットワークシステムの開発に従事．

堀端 啓史

株式会社オートネットワーク技術研究所ネットワーク研究部．車載ネットワークシステムの開発に従事．