車載LANメッセージの出現頻度と変化量を利用したリアルタイムデータ圧縮方式

上村孔明[1] 井上博之[2] 大平修慈[3] 石田賢治[2]

概要：車載LANに流れる情報をクラウドにリアルタイムにアップロードすることで，車両単体また他の車両との連携により車載LANデータの活用や異常検出等が可能となる．しかしながら，車載LANのトラフィックを常時記録した場合のデータ量は膨大であり，可能な限り圧縮したい．本研究では，車載LANデータ中におけるメッセージの出現頻度と変化量に着目した圧縮方式を適用することで車載LANデータのリアルタイムな圧縮が可能であることを示す．提案方式では，車載LANで一般的に使用される通信プロトコルであるCANの構成要素である，CAN IDとペイロードの出現頻度に着目し，ハフマン符号化を行うことでデータ量の圧縮を図った．また，車載LANデータにはCANメッセージが送信された時刻であるタイムスタンプについては，メッセージの前後での相対的時間差をランレングス符号化を行うことで圧縮を試みた．その結果，従来方式に対して更に半分程度のデータサイズへ圧縮が可能であることを確認した．

A Real-time Data Compression Method Using Frequency of Occurrence and Variation in In-vehicle LAN Messages

Koumei Uemura[1] Hiroyuki Inoue[2] Shuji Ohira[3] Kenji Ishida[2]

1. はじめに

近年，インターネットに常時接続された車両であるコネクテッドカーが実用化され始め，例えば，車両から得られるセンサ情報や制御情報をクラウドにリアルタイムにアップロードすることで，車載LANデータの活用や異常検知等が可能となる[1]．また，複数の車両からの情報を監視することで，外部からの不正アクセスおよび異常の検知も可能となる[2]．車種によって異なるが車載トラフィックは概ね1時間あたり40Mbyte程度のデータ量となり，業務用車両で1日あたり8時間の走行を想定すると1ヶ月で約10Gbyteものデータ量になってしまう．車載LANでは，一般的に通信プロトコルであるCAN（Controller Area Network）[3]が採用されており，このうち車載ログの解析に必要な情報はCANメッセージ中のCAN IDとペイロード，および送信時刻であるタイムスタンプとなる．加えて，クラウドにデータを送信するデータロガーおよび圧縮器は，組込みシステムである車載器に組み込む際にマシンパワーが要求されることが想定される．

本研究では，CAN IDとペイロード，タイムスタンプの特性を考慮し圧縮することで，車載トラフィックのデータ量を削減できることを示す．CAN IDとペイロードの値は出現頻度とデータのフォーマットが事前に分かっているので，出現確率に基づくハフマン符号化を適用し，タイムスタンプは，メッセージの前後での相対的時間差をランレングス符号化を行うことで圧縮を試みた．なお，CAN IDとしては11ビットと29ビットのものがあるが，今回はCAN IDが11ビットで，ペイロード長が最大8バイトのCANの標準フレームフォーマットのメッセージを対象として検討を行った．その結果，従来の手法に比べて約半分となる，4分の1から5分の1のデータサイズにまで圧縮できることが分かった．

1 広島市立大学情報科学部
Faculty of Information Sciences, Hiroshima City University
2 広島市立大学大学院情報科学研究科
Graduate School of Information Sciences, Hiroshima City University
3 奈良先端科学技術大学院大学先端科学技術研究科
Graduate School of Science and Technology, Nara Institute of Science and Technology

2. 車載LANデータの特徴と関連研究

2.1 CAN(Controller Area Network)

CANとは車載LANで一般的に使用される通信プロトコルであり，共有バス上にCANメッセージをブロードキャストすることで通信を行う．CANメッセージは，CAN ID，DLC，ペイロード等から構成され，図1にそのフォーマットを示す．

2.1.1 CAN ID

11bitのCAN IDの場合は，16進数で0x000から0x7FFまでの範囲をとる．CAN IDは例えば0x00AのCAN IDは車速を示し，0x00BのCAN IDはハンドルの舵角を示すといったように，当該CANメッセージの役割を示す場合が多い．また，同一の車両メーカのものであっても，車種Aでは車速を0x00Aで表すが，車種Bでは0x00Cで表すといったように，車種によっても役割が異なる場合もある．

2.1.2 DLC(Data Length Code)

DLCは4bitで構成されており，0から8の範囲をとる．DLCはペイロードが何バイト長のデータであるかを表す．

2.1.3 ペイロード

ペイロードは0から8バイトで構成されており，CAN IDに対応したデータが格納される．CAN ID 0x00Aでは6バイトとなるが，CAN ID 0x00Bでは8バイトとなるといったように，ペイロードの長さや内容はCAN IDや車種によって異なっている．また，ペイロードのデータの格納場所はCAN IDによって異なり，例えばCAN ID 0x00Aの1から6バイト目には余り変化がない静的なデータが格納され，7バイト目には車速の上位ビットが，8バイト目には車速の下位ビットが格納されるという場合もある．

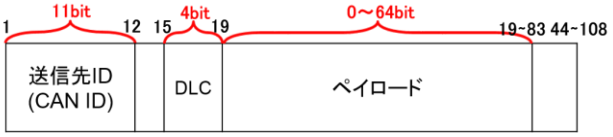

図 1 CAN メッセージのフォーマット

なお，一般的に，CAN IDやペイロードにどのような値を割り当てるかは，メーカや車種毎の独自仕様となっている．

2.2 タイムスタンプ

車載LANのCANでは，高速CANと呼ばれる通信速度が500kbpsのものが多くの車種で使用されている．CANメッセージは少なくとも約40ビットの長さを持ち，500kbpsでは1ビットは2μ秒に相当するため，CANメッセージの送信時刻を表すタイムスタンプの粒度は1μ秒単位とする．

2.3 関連研究

関連研究[4]では，クラウドに上げる車載LANデータを一定期間蓄積し圧縮アルゴリズム適用したり，異常検知を観測するまで送信データそのものを削減したりすることによるデータ量削減方式が提案されている．しかし，クラウドでの異常検知はデータ数の増加に伴い精度を向上させることができるため[2]，送信データそのものを削減したり，一定期間車載LANデータを蓄積したりすれば，車両に対する攻撃に迅速に対応できなくなるという問題が生じる．すなわち，車載LANデータを削減することなく，リアルタイムにデータを送信する必要がある．関連研究[5]では，1台の車両が複数の車両のデータを路側へ送信することで，通信するデータ量の削減を図っているが，車群形成の際のエラーによって複数の車両のデータを送出することができない．関連研究[6][7]では，プローブデータの圧縮方式の評価を行っているが，車載LAN上の全てのデータを送信せずに，プローブデータのみを5分から10分間隔で圧縮し，送信する方式であり，リアルタイム性がない．

CANメッセージそのものを圧縮する研究として，関連研究[8][9]等がある．関連研究[8][9]では，同一のCAN IDのメッセージについて前後でペイロード全体の差分を取った結果が0の場合”0”の，0以外の場合”1”のビットをヘッダーとして付与する．そして，ヘッダーが0であるメッセージを送信しないことで圧縮を実現している．また関連研究[10]では，同一のCAN IDのメッセージで差分を取った時に，ペイロードの各バイトのビット長に応じて異なるヘッダーを設けて，差分を取った結果ペイロードが0となるメッセージを送信しない処理を行うことで圧縮している．しかし，これらの方式は差分を取ったペイロードが0となる場合についてのみ圧縮されており，CAN自体の特性が活かされていない．

筆者らが以前に発表した方式[11]（以下，従来方式）では，同一CAN IDのCANメッセージの前後でペイロードの値の差分を取り，ランレングス符号化を施すとともに，CAN IDの出現頻度に着目して，CAN IDにハフマン符号化を施す圧縮方式を提案している．本研究では，従来方式に加えて，ペイロードの値の出現頻度に着目し，ペイロードにもハフマン符号化等の圧縮を施すことで更なる圧縮率の向上を検討する．

3. CANに着目したデータ圧縮方式

3.1 概要

従来方式では，同一CAN IDのCANメッセージの前後では，データの変化量が小さいという車載LANの特性を利用して，CANメッセージの前後で差分をとり，符号化を施すことで可逆圧縮を実現していた．本研究では，ある国産の車種Xと車種Yから得られた車載ログを用いて実験を行った．また，出現頻度が特に大きいCAN IDのCANメッセージについて，ペイロードの各バイトに対してハフマン符号化を施すことで，圧縮率の向上を図った．

3.2 CAN IDにおける圧縮

CAN IDにおける圧縮は，従来方式と同一の手法を採用した．従来方式では，CAN IDの出現頻度の偏りに着目して，CAN IDにハフマン符号化を施すことでデータの削減を実現している．ハフマン符号化は，出現頻度の大きい文字には短いビット列を，出現頻度の小さい文字には長いビット列を割り当てることで，メッセージ全体のビット量の削減を図る符号化であり[12]，本研究ではCAN IDにビット列を割り当てることでデータ量の削減を図っている．図2および図3に，本研究で使用した車種Xおよび車種Yから得られた車載ログのCAN ID毎の出現頻度を示す．図2および図3からわかるように，CAN IDの出現頻度は均一でなく，偏りが存在する．ここでハフマン符号化は出現頻度の大きいCAN IDに短いビット列を与え，出現頻度の大きいCAN IDに長いビット列を与える．これにより，出現頻度の最も大きいCAN IDについて車種X，車種Yともに3bitのビット列で表現可能とな

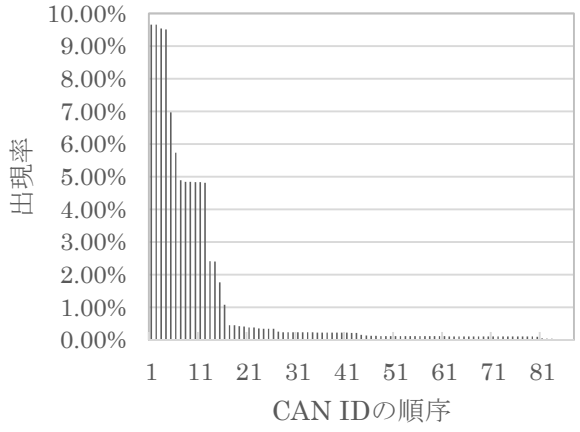

図 2 車種 X における CAN ID 毎の出現率

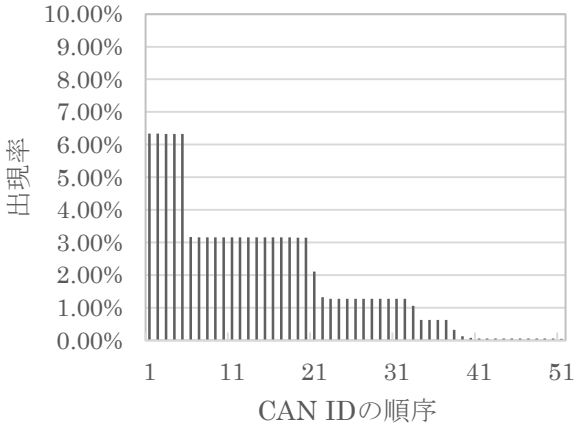

図 3 車種 Y における CAN ID 毎の出現率

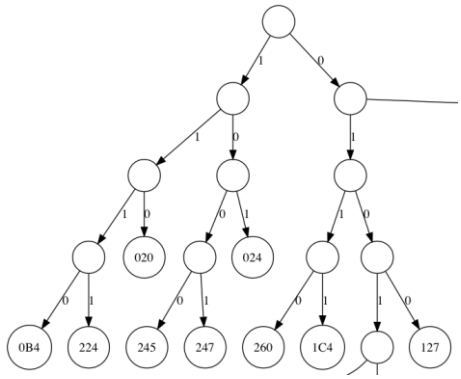

図 4　車種 X における CAN ID のハフマン木の一部

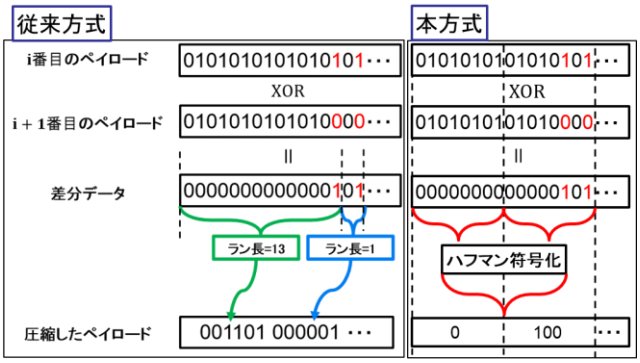

図 5　従来方式と提案方式の違い

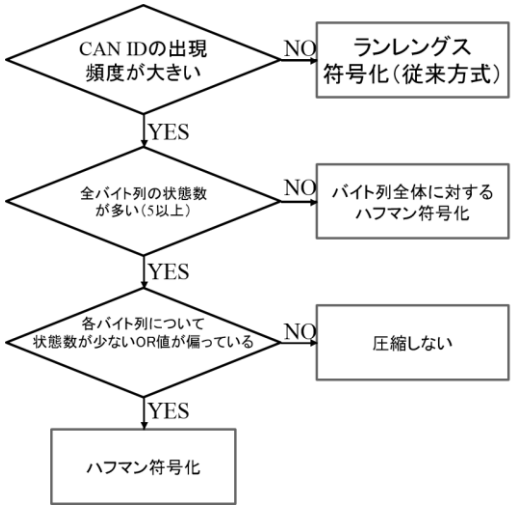

図 6　CAN メッセージに対する圧縮手順

り，11bitから大きくデータ量が削減される．図4に車種XにおけるCAN IDのハフマン木の一部を示す．図4からわかるように，例えばCAN ID 0x020はビット列110によって表される．

3.3 ペイロードにおける圧縮

図5に従来方式と提案方式のペイロードにおける圧縮方式の違いを示す．従来方式および提案方式では，同一CAN IDのCANメッセージについて，前後で差分をとることで差分データを生成する．差分データはCANメッセージの性質上，0の連続するデータとなることから，1が出現するまでの0の個数を記憶することで，元のデータを一意に復号することができるランレングス符号化[13]を利用してデータの削減を行った．提案方式では，従来方式に加えて，出現頻度の大きいCAN IDのCANメッセージについて，ペイロードの各バイトに対してハフマン符号化を施すことでデータ量の削減を図った．例えば，車種XのCAN ID 0x224について，6バイト目に出現する差分データの値の大半が0であり，出現するデータは全部で32種類しかないことから，6バイト目にハフマン符号化を適用することで，データ量の削減が期待できる．

表 1　車種 X について CAN ID 毎のフォーマット

ID	FLAG	D1	D2	D3	D4	D5	D6	D7	D8
0AA		○		○		○		○	
020	○	○	○	○	○	○			
025		○	○	○	○	○		○	
024		○	○	○		○		○	
127		○	○	○	○	○	○		
260	○	○	○	○	○	○	○	○	○
1C4	○	○		○	○	○	○	○	
247	○	○	○	○	○	○			
245	○	○	○	○	○	○			
224	○	○	○	○	○	○	○	○	○
0B4		○	○	○	○	○	○		
230	○	○	○	○	○	○	○	○	○

表 2　車種 Y における CAN ID 毎のフォーマット

ID	FLAG	D1	D2	D3	D4	D5	D6	D7	D8
141	○	○	○	○	○	○	○	○	○
140		○	○	○	○	○	○	○	○
149	○	○	○	○	○	○	○	○	○
148		○	○	○	○	○	○	○	○
002		○	○	○	○	○	○	○	○
144	○	○	○	○	○	○	○	○	○
164	○	○	○	○	○	○	○	○	○
15A	○	○	○	○	○	○	○	○	○
152	○	○	○	○	○	○	○	○	○
150	○	○	○						
0D4	○		○		○		○		○
0D3		○	○	○	○	○	○	○	
0D2	○	○	○	○	○	○	○	○	○
0D1	○	○	○	○	○				
0D0		○	○	○	○	○	○	○	○
156	○	○	○	○	○	○	○	○	○
371		○	○		○	○	○	○	○
280	○	○	○	○	○	○	○	○	○
7E8		○	○	○	○	○	○	○	○
7E0		○	○	○	○	○	○	○	○

また，車種YのCAN ID 0x0D1のCANメッセージなどは全てのバイトが0である割合が大きく，バイト毎ではなくバイト列全体に対してハフマン符号化を施す．同様にCAN ID 0x15AやCAN ID 0x164のCANメッセージのように，バイト列全体の状態が3状態で表されるようなCAN IDのCANメッセージに対しても，バイト列全体に対してハフマン符号化を施す．出現頻度の大きいCAN IDのCANメッセージについて，各バイト列の出現する値の組み合わせの数（以下，状態数）が多い場合，つまりハフマン符号化が有効でないと推測されるバイト列に対しては，圧縮処理を行わず，8bitのままで送信する．ただし，状態数が多い場合においても出現頻度に偏りがある，即ち出現した値の内，90%以上が0であったような場合は，明らかにハフマン符号化が有効であるため，ハフマン符号化の処理を施す．その他，出現頻度の小さいCAN IDのCANメッセージに対しては，従来方式のランレングス符号化を施す．図6に，CANメッセージに対する圧縮手順の詳細を示す．表1および表2に車種Xおよび車種Y

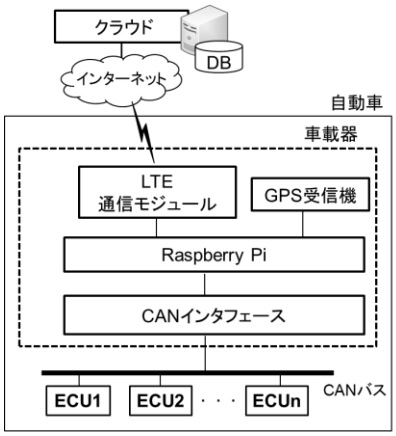

図 7 評価実験の構成

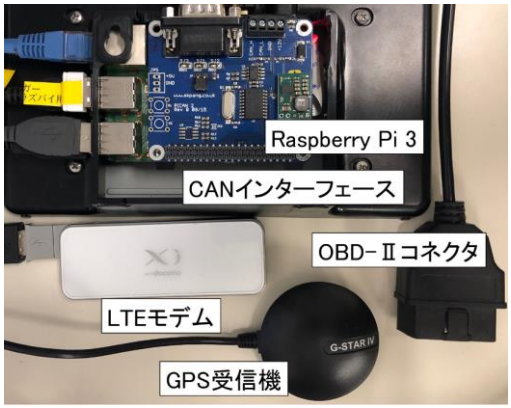

図 8 評価に用いた機材の外観

表3 Raspberry Pi 3の仕様

OS	Linux（Raspbian）
CPU	Broadcom BCM2837 （1.2GHz ARM Cortex-A53 Quad core）
RAM	1GB （LPDDR2）
Storage	16GB（MicroSD card）

においてCAN ID毎のCANメッセージのフォーマットを示す．丸印はハフマン符号化を行うことを表し，Dは各バイト列について，FLAGはバイト列全体に対するハフマン符号化を行うか否かを表す．

3.4 タイムスタンプにおける圧縮

あるCANバスにおいて，車両の動作時は，CANメッセージは1秒間に数十個～数百メッセージが送信されことから，CANメッセージのタイムスタンプは，前のメッセージの送信時刻との算術的な差を取ることで先頭から0の連続したデータとなる．よって，初めに出現する1までの0の個数を表し，以降のデータは符号化を行わず送信するといった，ランレングス符号化を1回のみ適用する方式を採用することで，効率的な可逆圧縮が可能となる．この場合，ラン長を表すのに必要なビット数は，取得するタイムスタンプの時間の上限に依存する．

4. 評価と考察

4.1 実験環境

図7に本研究の実験環境の構成を示す．図7の示すように，CANバス上のCANメッセージをリアルタイムに取得し，Raspberry Pi 3上で圧縮を施すことで，車載器を再現した．その際，Raspberry Pi 3がCANメッセージを取得した時刻をタイムスタンプとして付与した．また，コネクテッドカーを想定し，圧縮したデータを同一ネットワーク内に設置してあるサーバ上に送信し、サーバ側で復号を行うことで，圧縮から復号までの一連の流れを確かめた．図8に実際に使用した機材の外観を示す．表3に使用したRaspberry Pi 3の仕様を示す．

4.2 評価実験と結果

評価実験は，車種Xから得られた5分間の走行中のログと車種Yから得られた5分間の走行中のログをCANバス上にそれぞれ再生することで行った．表4と表5にそれぞれ車種X,Yにおいて，タイムスタンプ，CAN ID，ペイロード，DLCからなるCANメッセージを対象に行った各圧縮方式による圧縮率を示す．圧縮率は元のデータ量に対する圧縮後のデータ量を表している．図9，10にそれぞれ車種X，Yにおける圧縮の前後でのデータ量の比較を示す．なお，3.2節で述べたようにCAN IDの圧縮方式は従来方式と同じであり，圧縮率に差はない．タイムスタンプとCANメッセージを合わせた全体の圧縮率は，従来方式と比べて約半分に向上した．

4.2.1 DLCについて

図9，10では，見かけ上DLCが圧縮されているように見えるが，従来方式の示すDLCとは“ラン長の塊がいくつあるか”を示すフィールドであり，ペイロードが何バイト長であるかを表すといった本来のDLCの役割と同一ではない．また提案方式では，CAN ID毎のペイロード長はデコーダ（復号器）に記述されることを想定しており，圧縮されたCANメッセージのフォーマットにDLCを含まず，出現頻度の低いCAN IDのCANメッセージを従来方式で圧縮していることから，見かけ上DLCのデータ量が減っている．

4.2.2 ペイロードの圧縮率の比較と評価

表4，5および図9，10に示すように，ペイロードの値全体あるいは各バイト列にハフマン符号化を施した結果，車種Xおよび車種Yについて，従来方式と比べてさらに約半分のデータサイズへ圧縮が可能であった．車種Xと車種Yでの違いは，ペイロ

表4 車種Xにおける各方式の圧縮率

	タイムスタンプ	CAN ID	ペイロード	全体
従来方式		19.4%	51.0%	46.7%
提案方式	42.2%	19.4%	26.1%	24.4%

表5 車種Yにおける各方式の圧縮率

	タイムスタンプ	CAN ID	ペイロード	全体
従来方式		20.9%	40.9%	40.4%
提案方式	40.3%	20.9%	18.1%	19.5%

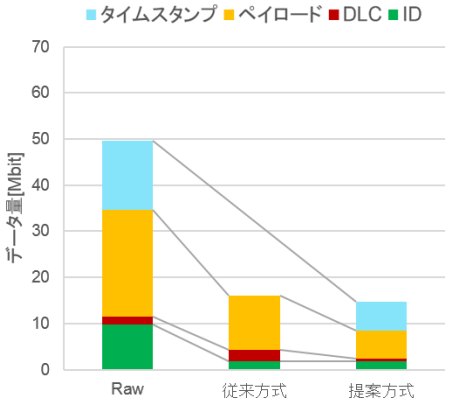

図 9　車種 X における各方式のデータ量の比較

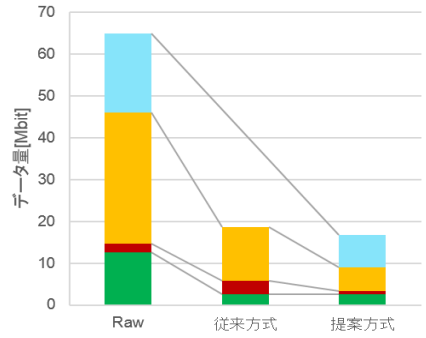

図 10　車種 Y における各方式のデータ量の比較

ードの値の変化が車種Yでは少なかったためと考えられる.

4.2.3 タイムスタンプの圧縮率の評価

評価実験では，CANのデータレートが500kbpsであることを考慮し，タイムスタンプを25時間未満までを1μ秒単位で取得可能とした．つまり，一意に復号が可能であることを想定すると，0μ秒から86399999999μ秒に対応する必要があり，

$$log_2(86400000000) \fallingdotseq 36.3$$

より，1つのタイムスタンプあたり37bitを必要とする前提で圧縮率を評価した．ここで，前後のタイムスタンプ同士で算術的な差を取った結果が全て0だった場合，0が37桁続くデータとなることから，ラン長は6bitで表す.

従来方式ではタイムスタンプは考慮していなかったが，今回の方式により，表4,5および図9,10より．車種X，Yともに圧縮を行わない状態と比較してデータ量が約4割に圧縮できた.

4.3　リアルタイム性の評価

リアルタイム性の評価では，CANの1メッセージの到着する平均時間内で1メッセージの圧縮を終える必要がある．車種Xは約860メッセージ/秒，車種Yは約1500メッセージ/秒であり，1秒当たり約2000メッセージを越えることはなかったため，1メッセージの圧縮に要する時間の最大値を0.5msとする．前述の5分間分の走行ログを用いて，車種Xと車種Yのそれぞれで圧縮および送信に要した時間を表6に示す.

表6より，1メッセージあたりの圧縮に8μsから20μs程度の時間を要している．CANメッセージの送信間隔に比べて十分小さいことから，リアルタイムに圧縮処理を行うことが可能であるといえる.

表6　5分間のログの圧縮に要した時間

	メッセージ数	合計処理時間 [s]	平均処理時間 [μs]
車種 X	254698	2.22	8.72
車種 Y	473115	9.34	19.7

また，圧縮中のRaspberry PiのCPU使用率は，おおよそ26から30%程度であり，組み込み機器である車載器においても実装可能な範囲に留まっている.

4.4　考察

メッセージの圧縮に関しては，従来方式と比較しても，ペイロード部分の圧縮率が大きく向上したことによりメッセージ全体の圧縮率も2倍程度に向上し，車種Xで24.4%に，車種Yで19.5%となった．すなわち元の車載LANのデータを4分の1から5分の1に圧縮でき，クラウドにデータを送信する際の通信帯域を大きく減らすことが可能となる．なお，本方式では車種毎のCAN IDとバイト列の出現頻度に基づき，特定の圧縮方式を適用することで圧縮率の向上を実現しているため，車種によって圧縮率は異なる．自動車は組込みシステムにつき，CANメッセージの出現頻度に基づくリストを事前に作成することは可能であり，従来方式では同一の圧縮方式を適用していたペイロードの圧縮の部分で大きな改善が可能になったと考えられる.

また，本方式ではハフマン符号化を適用している以上，ハフマン符号化のリスト(以下，ハフマンリスト)にないCAN IDおよびペイロードの値は符号化することができない．ハフマンリストに無いCAN IDのメッセージを取得した場合はハフマンリストを動的に作成し直す方法が考えられるが，ハフマンリストを変更するとエンコーダとデコーダで同期を取る必要があり，変更の都度その履歴を保存しておく必要があるため，復号の手間を考えると現実的ではない．つまり，ハフマンリストにないCAN IDのメッセージを受信した場合は，圧縮を施さず生のデータのまま送信する方式が有効である．ハフマンリストに無いペイロードの値を取得した場合は，従来方式のランレングス符号化を施す方法が考えられるが，バイト列毎にハフマン符号化およびランレングス符号化を適用すると復号に手間がかかると同時に特定のバイト列にランレングス符号化を施す関係上，従来方式に比べてビット長が短くなり，効率的な圧縮が施されないことが考えられる．またCAN IDの場合と同様に動的にハフマンリストを更新する方式は現実的ではない．つまり，ハフマンリストにないペイロードの値を受信した場合は，圧縮を施さない方式を適用する．ハフマンリストにないペイロードの出現頻度は非常に低いため，全体の圧縮率に及ぼす影響は小さい.

タイムスタンプの圧縮について，この圧縮率は計測可能とする時間の最大値と分解能によって決まるラン長の最大値に依存している．4.2.3節で述べたように，評価では25時間未満までを1μ秒単位で取得可能としたが，4.3節で述べたように実際はCANメッセージは1秒間に何百と送信されており，送信時間の差を取るには1~2秒程度に対応するだけで実用上は十分だと

考えられる.

本方式では，評価の対象外としたが，実用的にはGPS等から得られた車両の位置情報も車載ログに含まれていると考えられる．位置情報は主に緯度，経度，高度から構成されている．車両の移動を考えると，緯度，経度，高度は極端に変化するとは考えられず，明らかにそれぞれ前後で差分を取ることで0が連続するデータとなることが想定される．つまり，位置情報の圧縮の際にはタイムスタンプの圧縮と同様にランレングス符号化を施す方式が効果的だと考えられる．また，位置情報の送信周期は例えば1秒のように大きく，1秒間に1000個前後存在するCANメッセージと比較すると全体のデータ量はわずかであることから，全体の圧縮率への影響は小さくなる．

5. おわりに

インターネットに接続された車両であるコネクテッドカーを想定して，車載LANメッセージをクラウドに伝送する際に適用するリアルタイムな圧縮方式に関してタイムスタンプを含めて検討した．従来方式に加えて，出現頻度の大きいCAN IDのCANメッセージに対してペイロードの各バイト列またはバイト列全体に対してハフマン符号化を施すことで，従来方式に対して約半分となる4分の1から5分の1のデータサイズに圧縮できることがわかった．

今後の課題として，インターネットを介してクラウド上にデータを送信した場合における実行時間計測の実験による，より実用に即したリアルタイム性の評価がある．また，出現頻度が比較的小さいCAN IDのメッセージにも同様にハフマン符号化によるデータ量削減の考察を行うことで，圧縮率の更なる向上を図る．

謝辞

本研究の一部は，JSPS科研費18K11299により行われた．ここに記して謝意を表す．

参考文献

[1] 江崎貴也，金森健人，鶴田智大，手柴瑞基，井上博之: 全車載LANデータをクラウドサービスで安全に利用するためのシステムの試作，第9回地域間インタークラウドワークショップ（RICC），pp.1-6，Mar. 2016.

[2] 芳賀智之，氏家良浩，鶴見淳一，岸川剛，前田学，松島秀樹，安西潤: クラウドを利用した車載ネットワーク向け統計的異常検知システムの提案，SCIS 2016，pp.1-8， Jan. 2016.

[3] International Organization for Standardization: Road vehicles － Controller area network (CAN) －, Part 1: Data link layer and physical signaling, ISO 11898-1:2015, 2015.

[4] 佐々木崇光，高橋良太，鶴見淳一，岸川剛，芳賀智之，松島秀樹: 車載向けセキュリティシステムの運用コストを削減する監視データ量削減方式，SCIS2017，pp.1-7，Jan. 2017.

[5] 成田干城，朝倉啓充，屋代智之，重野寛，岡田謙一: 車群形成を用いた路車間通信量の削減方法，情報処理学会研究報告高度交通システム，vol.60(2004-ITS-017)，pp.25-32，2004.

[6] 清原良三，伊藤一彦，斎藤正史，小塚宏: テレマティクスサービス向け情報圧縮方式，情報処理学会研究報告，vol.2011-MBL-60，no.16，pp.1-8，2011.

[7] 中瀬裕多，日江井太郎，清原良三: 車載スマートフォンにおけるプローブデータ圧縮方式の評価，情報処理学会第75回全国大会，pp.123-124，2013.

[8] Yu-jing WU, Jin-Gyun Chung, Myung Hoon Sunwo: Design and Implementation of CAN Data Compression Algorithm, Proceedings of 2014 IEEE International Symposium on Circuits and Systems, pp.582-585, Jun. 2014.

[9] Yu-jing WU, Jin-Gyun CHUNG: Efficient controller area network data compression for automobile applications, Front Inform Technol Electron Eng 2015, vol.16, pp.70-78, Jan. 2015.

[10] Supriya Kelkar, Raj Kamal: Boundary of Fifteen Compression Algorithm for Controller Area Network Based Automotive Applications, Proceedings of 2014 International Conference on Circuits, Systems, Communication and Information Technology Applications, pp.162-167, Apr. 2014.

[11] 大平修慈，金森健人，井上博之，石田賢治: 車載LANトラフィック監視システム向けのリアルタイムデータ圧縮方式，CSS2017，pp.1469-1474，2017.

[12] A. Huffman: A method for the construction of minimum-redundancy codes，Proceedings of the I.R.E，pp. 1098-1102，Sept. 1952.

[13] Robinson. A. H，Cherry. C: Results of a prototype television bandwidth compression scheme，Proceedings of the IEEE，IEEE. 55 (3)， pp.356–364，1967.