# 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

大平 修慈‡ 金森 健人† 井上 博之†* 石田 賢治†

**概要**：車載 LAN には様々なセンサ情報や制御情報が流れており，それらの情報をクラウドのような広域網とつながるサーバやストレージ上にリアルタイムに伝送し，蓄積および分析するような車載 LAN トラフィック監視システムを用いることで，車両の運行情報の収集や運転評価による保険料への反映等の新たなサービスが実現可能となる．また，複数の車両からの情報をリアルタイムに収集し監視することで，車載 LAN や ECU に対する外部からの不正アクセスや異常の検知が可能となる．ところで，車載 LAN から得られるデータは 1 時間あたり約 40M バイトにもなり，単純に伝送するには伝送路の負担が大きい．先行研究では，車載 LAN のデータを一定期間溜めて圧縮したり，データそのものを間引いたりして送信する方式があるが，全てのデータをリアルタイムに受信することができないという課題がある．本論文では，車載 LAN トラフィック監視システム向けに，車載 LAN の主要なプロトコルである CAN のメッセージの特徴に着目し，全ての車載 LAN データをリアルタイムにデータ圧縮する方式を検討する．組込み機器向けに軽量なアルゴリズムにより効率的に可逆圧縮でき，リアルタイムにメッセージを伝送できるようなアルゴリズムを提案する．複数の車種の実車トラフィックに提案方式を適用したところ，通信のデータ量を約 50〜70%削減でき，圧縮に要する CPU 時間は十分短いという結果が得られた．

**キーワード**：車載 LAN，データ圧縮，CAN

# Real-time Data Compression Method for the In-vehicle LAN Traffic Monitoring System

Shuji Oohira‡ Kento Kanamori† Hiroyuki Inoue†* Kenji Ishida†

***Abstract***: In a typical in-vehicle LAN, the information that flows is the control information and that from various sensors. The in-vehicle LAN traffic monitoring system transmits and analyzes such information in real time by using servers and storage such as a cloud connected with a Wide Area Network (WAN). It is possible to develop new services in this system such as collection of information about the vehicle and reflection on the insurance fee by evaluation of driving. Additionally, it is possible to detect an unauthorized or abnormal access to the in-vehicle LAN or Electronic Control Unit (ECU) by collecting and monitoring the information of several vehicles in real time. However, the data obtained from the in-vehicle LAN is of a magnitude of 40 Megabytes per hour, which is a burden on the communication. The previous research has a method of sending data by compressing it for a certain period of time or by thinning it. However, there exists a problem of the impossibility to receive the entire data in real time. In this paper, we propose a real-time data compression scheme focusing on the characteristics of the Controller Area Network (CAN), which is the main protocol of an in-vehicle LAN used in an in-vehicle LAN traffic monitoring system. Further, we studied algorithms that can transmit messages in real time with lightweight algorithms for embedded devices and efficient lossless compression. Application of the proposed method to actual vehicle traffic consisting of multiple vehicle types showed that the amount of communication data can be reduced by about 60 to 70% and the CPU consumption is sufficiently short time for the compression.

***Keywords***: in-vehicle LAN, data compression, CAN

## 1. はじめに

車載 LAN には様々なセンサ情報や制御情報が流れており，それらの情報をクラウドのような広域網とつながるサーバやストレージ上にリアルタイムに伝送し，蓄積および分析するような車載 LAN トラフィック監視システムを用いることで，車両の運行情報の収集や運転評価による保険料への反映等の新たなサービスが実現可能となる[1]．また，複数の車両からの情報をリアルタイムに収集し監視することで，車載 LAN や ECU に対する外部からの不正アクセスや異常の検知が可能となる[2]．

車載 LAN のデータ全てをクラウドに伝送するには，車種に依存するが概ね 1 台につき 1 時間あたり 40Mbyte のトラフィックとなり，1 日当たり 8 時間の運行で 1 ヶ月あたり約 10Gbyte ものトラフィックとなってしまう．先行研究[3]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

‡ 広島市立大学 情報科学部
Department of Information Sciences, Hiroshima City University
† 広島市立大学大学院 情報科学研究科
Graduate School of Information Sciences, Hiroshima City University
* 重要生活機器連携セキュリティ協議会（CCDS）研究開発センター
Connected Consumer Device Security Council (CCDS)

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．以上より，車載 LAN データを削減することなく，リアルタイムに送信できる圧縮方式の検討が必要である．

車載 LAN で一般的に使用されている通信プロトコルである CAN（Controller Area Network）[4]では，メッセージ中で分析が必要な部分としては CAN ID とペイロードとなり，これらの特徴に着目することでリアルタイム送信が可能でかつ効率的に可逆圧縮できる可能性がある．また，クラウドにデータを送信するための車載器は CPU やメモリの性能が比較的限定されており，クラウド側は多数のデータを受信し伸張することから，圧縮および伸張方式の軽量化が課題となる．本研究では，車載 LAN トラフィック監視システム向けに，車載 LAN の主要なプロトコルである CAN のメッセージの特徴に着目し，全ての車載 LAN データをリアルタイムにデータ圧縮する方式を検討する．組込み機器向けに軽量なアルゴリズムにより効率的に可逆圧縮でき，リアルタイムにメッセージを伝送できるようなアルゴリズムを提案する．

## 2. 車載 LAN データの特徴と圧縮方式

### 2.1 CAN(Controller Area Network)

車載 LAN で一般的に使用されている通信プロトコルである CAN のデータフレームにおいて，データ内容や送信ノードの識別や通信調停の優先順位を決める CAN ID と送信データを格納するペイロードの 2 つが車の状況に深くかかわっている．CAN メッセージのフォーマットを図 1 に示す．

一般的な車載 LAN で使用されている CAN ID は 11bit につき 16 進数で 0x0～0x7FF の範囲をとり，データ内容や送信ノードを識別するために使用される他に，通信調停の優先順位を決定することも担っている．CAN が採用している CSMA/CA 方式では，バスが使用中には他ノードから送信できないが，同時に複数のノードからデータが送信されてしまう場合もある．このような場合に通信調停が発生し，CAN ID の値が小さい方が優先される．一般に車速やエンジン回転数等の自動車の制御に関わる重要なメッセージは優先度を高いため，0x0 に近い値が設定される．さらに，CAN ID は車種や，メーカー毎に異なるという特徴がある．例えば，車速を表す CAN ID は，国産ハイブリッド車（以降，車種 X）の場合 0x0B4 であるが，国産 ADAS（先進運転支援システム）機能搭載車（以降，車種 Y）の場合 0x0D1 となっている．

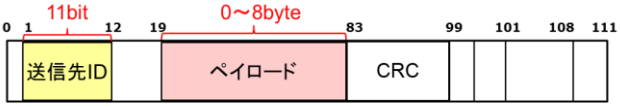



図 1　CAN メッセージのフォーマット

ペイロードの長さは，DLC（Data Length Code）と呼ばれるペイロードが何バイトのデータであるかを表すビットフィールドによって決定される．ペイロードは最大 64bit，すなわち，8byte なので DLC の設定範囲は 0～8 となる．ペイロード 64bit に，CAN ID 毎に実際の運転状況を表す車速，ハンドル，ブレーキ等の様々な状態だけでなく，車両の制御状態や各種センサやアクチュエータの情報が格納される．

### 2.2 実車における CAN メッセージの分析

CAN メッセージの特徴から以下の 2 つを取り上げ，さらに実車トラフィックにてそれらの特徴を解析した．

(1) 送信先を決定する CAN ID にはどの車種においても CAN ID によって出現頻度に偏りがある．

(2) 各 CAN ID のペイロードの変化量は，0～数十 bit となり，常時変わらない bit が存在する．

解析した実車トラフィックは車種 X および車種 Y ともに同じ走行コースを走った 1 分間の走行時のログ（以降，テストデータ）である．また，CAN トラフィックは制御に関するデータが大部分で統計的な頻度は変わらないため，テストデータは 5 分のデータのうち 1 分間を抽出したものとした．車種 X と車種 Y のテストデータの基本データおよび解析した結果を表 1 にまとめた．

まず，CAN ID よって出現頻度に偏りがあることについては，図 2 に示す車種 X の 1 分間の走行中ログについてのグラフのように CAN ID が 0 に近い値であるほどに出現頻度が増加することがわかる．これは，速度やエンジン回転数等のメッセージ周期の短い制御に関わる重要な CAN ID は，優先度が高くなる 0x0 に近い CAN ID を割り振られるからである．一般に，0x0 に近い CAN ID は一定時間あたりの出現頻度は多いが，0x7FF に近い CAN ID ほど一定時間あたりの出現頻度は少ない．

次に，各 CAN ID の 1 メッセージ毎のペイロードの変化量を観測した．CAN アナライザにてパケットキャプチャした CAN ID が 0x0B4 のメッセージを図 4 に示す．この例では，点線の四角で囲んだ高々3byte のみが頻繁に変化していることが確認できる．表 1 に示す通り，ペイロードの前後でメッセージに変化がないビットの総データ量は，テスト

表 1　解析したテストデータの統計データ

| | 車種 X | 車種 Y |
|---|---|---|
| 総メッセージ数 | 52,413 | 96,294 |
| 総 CAN ID 数 | 87 | 50 |
| CAN ID とペイロードの総データ量 | 3,558,367[bit] | 6,881,392[bit] |
| CAN ID の総データ量 | 576,543[bit] | 1,058,728[bit] |
| ペイロードの総データ量 | 2,981,824[bit] | 5,822,664[bit] |
| 前後で変化がないペイロードの総データ量 | 2,763,715[bit] | 5,520,774[bit] |
| 平均ペイロード長 | 56.9[bit] | 60.5[bit] |

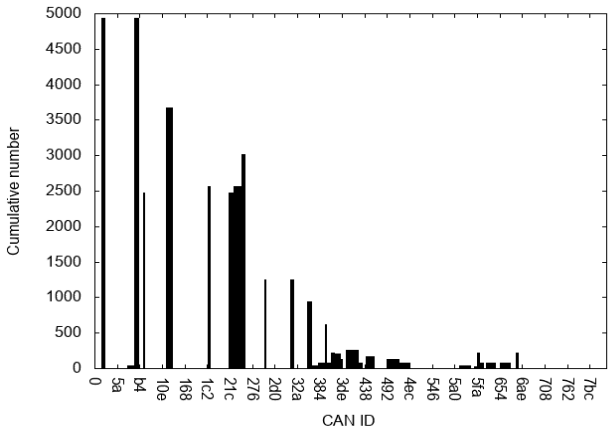



図 2　車種 X 走行時 1 分間の CAN ID の出現頻度

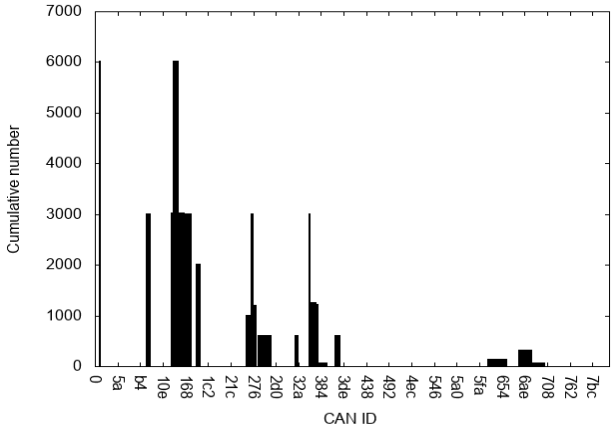



図 3　車種 Y 走行時 1 分間の CAN ID の出現頻度

| Time (abs/rel) | Tx | Er | Description | ArbId/Header | Len | DataBytes |
|---|---|---|---|---|---|---|
| | | | | 0B4 | | |
| 24.834 ms | | | HS CAN $B4 | B4 | 8 | 00 00 00 00 50 12 8D AB |
| 24.998 ms | | | HS CAN $B4 | B4 | 8 | 00 00 00 00 54 12 83 A5 |
| 24.170 ms | | | HS CAN $B4 | B4 | 8 | 00 00 00 00 5C 12 8A B4 |
| 25.830 ms | | | HS CAN $B4 | B4 | 8 | 00 00 00 00 64 12 80 B2 |
| 25.003 ms | | | HS CAN $B4 | B4 | 8 | 00 00 00 00 68 12 7E B4 |
| 24.996 ms | | | HS CAN $B4 | B4 | 8 | 00 00 00 00 70 12 7B B9 |
| 24.998 ms | | | HS CAN $B4 | B4 | 8 | 00 00 00 00 78 12 6D B3 |
| 25.172 ms | | | HS CAN $B4 | B4 | 8 | 00 00 00 00 7C 12 72 BC |
| 24.827 ms | | | HS CAN $B4 | B4 | 8 | 00 00 00 00 84 12 73 C5 |
| 24.354 ms | | | HS CAN $B4 | B4 | 8 | 00 00 00 00 8C 12 6B C5 |

図 4　車種 X の車速メッセージのペイロードの変化

| Time (abs/rel) | Tx | Er | Description | ArbId/Header | Len | DataBytes |
|---|---|---|---|---|---|---|
| | | | | 0D1 | | |
| 19.824 ms | | | HS CAN $D1 | D1 | 4 | 95 00 00 04 |
| 19.998 ms | | | HS CAN $D1 | D1 | 4 | 94 00 00 84 |
| 19.999 ms | | | HS CAN $D1 | D1 | 4 | 94 00 00 84 |
| 19.998 ms | | | HS CAN $D1 | D1 | 4 | 93 00 00 04 |
| 20.197 ms | | | HS CAN $D1 | D1 | 4 | 92 00 00 84 |
| 19.803 ms | | | HS CAN $D1 | D1 | 4 | 92 00 00 84 |
| 20.000 ms | | | HS CAN $D1 | D1 | 4 | 91 00 00 84 |
| 20.001 ms | | | HS CAN $D1 | D1 | 4 | 91 00 00 84 |
| 20.004 ms | | | HS CAN $D1 | D1 | 4 | 90 00 00 04 |
| 20.227 ms | | | HS CAN $D1 | D1 | 4 | 90 00 00 04 |

図 5　車種 Y の車速メッセージのペイロードの変化

データのペイロードの総データ量と比べ近い値である．すなわち，同一 CAN ID の時間的に隣接するメッセージの間ではペイロードの変化は少ないと言える．すなわち，CAN ID ごとにビット単位の差分をとることで情報量を削減できるので，それをランレングス符号化することで，平均符号長を短くできる可能性がある．

また，図 3 および図 5 に示すように，上記 2 つの CAN ID とペイロードに関する特徴は車種 Y ついても同様に確認された．以上の分析により，CAN ID には出現頻度に偏りがあり，出現する頻度が高い CAN ID には短いビット列を，頻度が低い CAN ID には長いビット列を割り当てる．すなわち，ハフマン符号化を CAN ID に適用することで，平均符号長を短くできる可能性がある．

### 2.3 関連研究

関連研究[3]では，監視データ量の削減手法として，車の挙動に関するデータのみ送信，異常検知ができる最大の間隔でデータそのものを削減する，一定期間のデータに対し 7zip 圧縮を適用している．また，車載 LAN 内で異常が見られた時のみフルログに 7zip 圧縮を適用している．これらの方式では，一定期間データを溜めて圧縮アルゴリズムを適用するためリアルタイムにメッセージを送信できない．リアルタイムにメッセージを送信できることで，トラフィック監視システムがリアルタイムに異常を検知可能となる．また，全データを送信することで運転評価にデータを用いる際にも全データを用いなければ精密な分析ができない．

関連研究[5][6]では，位置情報，前方車との距離，速度，気温といったプローブデータの圧縮方式と，車載器とスマートフォンを接続したシステムを実装し評価を行っている．車載 LAN に接続することを前提としているが，車載 LAN 上の全データを送信せず，プローブデータのみを送信し，かつ 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．

関連研究[7][8]では，CAN 上の帯域を削減する専用の DR アルゴリズムの改善を提案している．CAN 上におけるデータ量削減において CAN ID を変更して通信することは不可能であることから，CAN ID のデータ量には着目しておらず，圧縮も行っていない．

以上のことから，従来方式では一定期間のデータに対し圧縮を適用した後に送信するため，リアルタイムにメッセージを送信できない問題があり，CAN ID のデータ量を削減することでデータ量の削減が見込める．本研究では，先ほど述べた CAN メッセージの特徴に着目し，CAN に適した可逆圧縮方式を提案し，圧縮率やリアルタイム性に関する評価を行う．

## 3. 提案方式

CAN ID とペイロードの特徴を分析した結果から，それぞれ異なった可逆圧縮方式を適用するほうが良いことが分かった．本章では，それぞれの圧縮方式と CAN メッセー

ジ全体のフォーマットについて詳細を提案する．

### 3.1 車載 LAN 向け圧縮方式の要件

CAN メッセージ向けの可逆圧縮方式の要件をまとめると以下のようになる．

(1) 可逆圧縮
(2) リアルタイムに通信可能な処理時間
(3) 車載器のような制限された性能でも実装可能な方式

### 3.2 CAN ID におけるデータ圧縮

2.2 節で述べたように，CAN ID には出現頻度の偏りがある．この特徴に着目すると，よく出現する CAN ID には短いビット列を，あまり出現しない CAN ID には長いビット列を割り当てる，すなわち，ハフマン符号化によって情報源符号化することにより全体的なデータ量が削減できることが期待できる．

ハフマン符号化を行うために，各 CAN ID の出現頻度を総メッセージ数で割った値をその ID の出現確率とし，CAN ID とその出現確率によりハフマン木を生成する．例えば，車種 X の CAN ID の出現頻度は図 2 のようになり，これをハフマン木とすると図 6 のようになる．メッセージ周期の短い速度に関する CAN ID 0x0B4 や，エンジン回転数に関

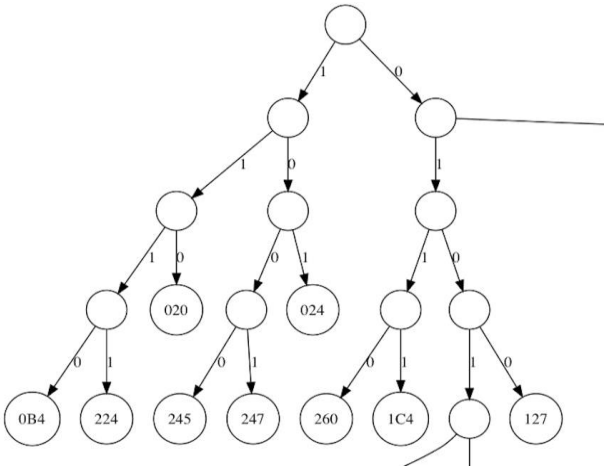



図 6　CAN ID のハフマン木の一部（車種 X）

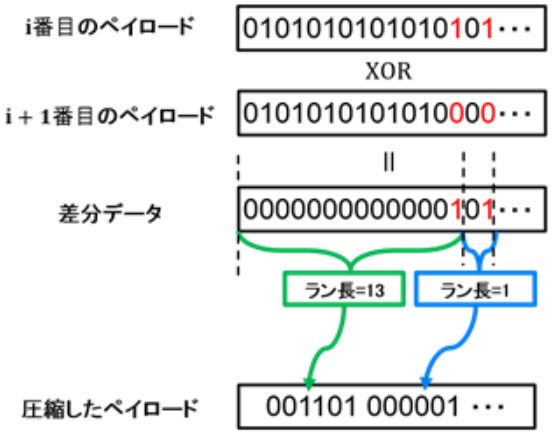



図 7　ペイロードにおける圧縮方式

する CAN ID 0x1C4 はそれぞれ，ビット列 1110 と 0110 のように符号化され 11bit から 4bit にまでデータ削減されている．

### 3.3 ペイロードにおけるデータ圧縮

2.2 節で述べたように，各 CAN ID の 1 メッセージ毎のペイロードの変化量は，0〜数十 bit となる．これより，前後のメッセージで XOR をビット単位で演算することで前後のメッセージで変化しているビットが抽出でき，この変化量を送信できれば大幅にペイロードのデータ量が削減可能である．図 7 にペイロードにおける圧縮方式の流れを示す．また，変化したビットを抽出したデータを以下では差分データと呼ぶ．差分データは，0 が頻繁に連続するデータとなることが明らかであるため，連続したビットの長さをラン長として送信するランレングス符号化を用いてデータ圧縮できると考えられる．これより，まずペイロードの変化量を得るために前データとの XOR の演算結果を差分データとして取得する．また，得られた差分データは CAN のペイロードの最大 64bit の場合でランレングス符号化を考えると，ラン長は 0〜63 の 64 通りとなる．よって，6bit あればすべての組み合わせに対応でき，固定長の 6bit で格納することでランレングス符号化を行う．

図 7 に示した方式（以降，方式 A）の場合，圧縮後のペイロードに格納されるランの長さを表すビット長が 6bit のため，CAN のペイロードの最大 64bit がすべて変化するときにはペイロードが$6 \times 64 = 384$bitとなってデータ量が非常に大きくなってしまう．これを解消するために，ペイロードの初期のビットが 0 なら 0 の連続を，1 なら 1 の連続をラン長とし，その後は 0 の連続と 1 の連続とを交互に反転して数え符号化する方式（以降，方式 B）も考えられる．この 2 つの符号化を比較し考察する．

さらに，車が走行時のログであるにも関わらず，同一 CAN ID のメッセージの前後で変化するビットが存在しない，すなわち，XOR 演算の結果が全て 0 である差分データが，車種 X では 50.6%，車種 Y では 26.3%，1 分間のログに含まれていることが観測された．同一 CAN ID のメッセージの前後で変化がない場合，ペイロードに何もデータを格納しない特別な規則を設ける．この特別な規則と各方式 A, B を組み合わせた方式を以降方式 A+, B+と呼ぶ．

### 3.4 圧縮された CAN メッセージのフォーマット

3.2 節と 3.3 節の検討結果から，圧縮された CAN メッセージのフォーマットを検討する．

CAN ID のハフマン符号化は，送信側と受信側で同一のハフマン木を持っていなければならない．CAN ID は車種ごとに異なるため，どの CAN ID の出現頻度が高いかは車種ごとに異なる．また，非周期なメッセージも存在するため，送信側と受信側でハフマン木を一定期間毎動的に生成することで最適なハフマン木を常に維持することが可能となる．ハフマン木の送信側と受信側の同期する方法として，

送信側が符号化テーブルを送信する方法，受信側からハフマン木を更新するパケットを送信する方法等が考えられる．このようなハフマン木を同期するプロトコルに関しては今後検討していく必要がある．ペイロードについては，絶対的なデータと差分データを区別するために Abs or Diff という 1bit のフラグを用いる．例えば，Abs or Diff = 0 のとき，ペイロードは絶対的なデータ，Abs or Diff = 1 のとき，ペイロードは差分データとなる．また，DLC は差分として出てくる 1 の値のビット数の総和を表すことから $2^6 = 64$ で 6bit 必要となる．

圧縮された CAN メッセージのフォーマットの図 8 に示す．

## 4. 評価と考察

評価は車種 X と車種 Y における，それぞれの 1 分間の走行時ログをテストデータとして用いて，圧縮率と圧縮に要する CPU 時間の計測を行った．

### 4.1 実験環境

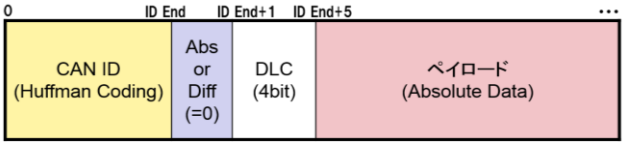
車載器は組込みシステムでありコストや CPU やメモリ等の資源が限られており，類似のハードウェアを持つ機器として，提案方式の評価には Raspberry Pi 3 を使用する．使用した Raspberry Pi 3 の仕様を表 2 に示す．

### 4.2 圧縮率の比較

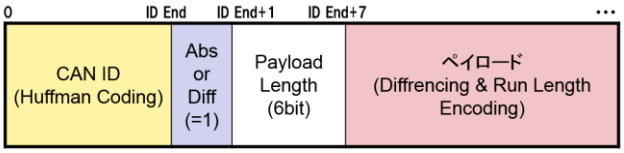
3 章で説明した CAN ID とペイロードのデータ圧縮方式をテストデータに適用し，圧縮率を評価する．

表 3 に，車種 X と車種 Y について，CAN ID をそれぞれハフマン符号化した結果を示す，車種 X の場合は圧縮率 42.4%，車種 Y の場合は圧縮率 45.5%となった．

ペイロードについて，0 の連続をランとして数えるランレングス符号化の方式 A をテストデータに適用した結果を表 4 に示す，車種 X の場合は圧縮率 52.6%，車種 Y の場合は圧縮率 40.2%となった．次に，0 と 1 のランを交互に数えるランレングス符号化の方式 B をテストデータに適用したところ，車種 X の場合は圧縮率 53.5%，車種 Y の場合は圧縮率 42.8%となった．方式 A と方式 B を比べると，方式 A の方がテストデータに対し効率的に圧縮できている．ペイロードのうち毎回変化する部分は一部であり，差分には多くの 0 が含まれる．そのため，方式 A の圧縮率が高くなったと考えられる．




(a) ペイロードが絶対的なデータの場合のフォーマット




(b) ペイロードが差分データの場合のフォーマット

図 8 圧縮された CAN メッセージのフォーマット

表 2 Raspberry Pi 3 の仕様

| | |
|---|---|
| OS | Linux（Raspbian） |
| CPU | ARM Cortex-A53 1.2GHz |
| RAM | 1Gbyte |
| Storage | 16GByte（MicroSD card） |

3.3 節でも述べた通り，車が走行時であっても CAN メッセージの中では変化量がないメッセージが多数存在することに着目して，変化量がない場合，すなわち，差分データが全て 0 の場合，ペイロードにはデータなしという特殊な規則を追加する．この規則と方式 A を組み合わせた方式 A+は，車種 X では，圧縮率 42.9%，車種 Y では，圧縮率 30.8%となり，車種 X，車種 Y ともに方式 A と比べ，約 10%近く圧縮率を改善できた．

CAN ID のハフマン符号化，最も圧縮率の高かったペイロードの圧縮方式を合わせ，DLC なども含めると，メッセージ全体の圧縮率は，表 5 に示すように，車種 X で 48.3%，車種 Y で 35.2%となり，およそ 50～70%のデータ量が削減できた．

### 4.3 CPU 時間による評価

4.2 節で車種 X，車種 Y のテストデータのどちらに対しても，最も圧縮率が高かった CAN ID のハフマン符号化とペイロード圧縮の方式 A+の CPU 時間に関しての評価を行う．

ペイロード圧縮の方式 A+の CPU 時間の 1000 回試行平均は，車種 X は 263[μs]，車種 Y は 278[μs]となった．ここで，CAN の帯域 500kbps でペイロードが 64bit であるようなメッセージが帯域の限界まで送信されることを考えると，単位時間あたり約 4500 メッセージが送信される．これより，それらのメッセージ間隔は，224[μs]となる．CAN の帯域の限界までメッセージを送信したときのメッセージ間隔 224[μs]と，方式 A+にかかる CPU 時間を比べると，方式

表 3 CAN ID 圧縮方式の圧縮率による比較

| | ハフマン符号化 |
|---|---|
| 車種 X | 42.4% |
| 車種 Y | 45.5% |

表 4 各ペイロード圧縮方式の圧縮率による比較

| | 方式 A | 方式 B | 方式 A+ | 方式 B+ |
|---|---|---|---|---|
| 車種 X | 52.6% | 53.5% | 42.9% | 42.1% |
| 車種 Y | 40.2% | 42.8% | 30.8% | 32.9% |

表 5 メッセージ全体の圧縮率

| | ハフマン符号化，方式 A+ |
|---|---|
| 車種 X | 48.3% |
| 車種 Y | 35.2% |

A+にかかるCPU時間の方が大きくなってしまう. しかし, CANはバス型のネットワークであり, 帯域一杯にメッセージを送信すると再送が連続してしまうため，そのような設計は行われない．実際のメッセージ間隔の限界は 224[μs] より大きい．従って，方式 A+にかかる CPU 時間は十分短くリアルタイムに処理できると言える.

CAN ID のハフマン符号化にかかる CPU 時間は，ハフマン木を生成した後は，CAN ID をハフマン符号化 CAN ID に変換するだけであるため，方式 A+にかかる CPU 時間と比べCAN IDのハフマン符号化についてのCPU時間は非常に短いと言える.

以上のCPU時間による評価より, 提案方式にかかるCPU時間は十分短く車載器のようなリソースが限られた環境でもリアルタイムに送信可能である.

## 5. まとめ

車載 LAN データをクラウド上にリアルタイムに伝送する車載 LAN トラフィック監視システムを想定し，車載器とクラウド間の通信路のデータ量を CAN の特徴を利用し削減する方式を提案し，実車データを適用しデータ圧縮率と圧縮に要するCPU時間の評価を行った. 複数の車種の実車トラフィックに提案方式を適用したところ，通信のデータ量を約 50〜70%削減でき, CPU 時間は十分短いという結果が得られた．今後は，提案した圧縮方式に適した CAN のペイロードのバイト順序の最適化や，車載器とクラウド間での詳細なプロトコルの設計，およびシステム全体を実装し動作させることでクラウドでの伸張処理を含むシステム全体の機能確認および評価を行っていく.

**謝辞** 本研究の一部は，広島市立大学特定研究費により行われた．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] 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.
[4] International Organization for Standardization: Road vehicles, controller area network (CAN), Part 1: Data link layer and physical signaling，ISO IS11898-1, 2003.
[5] 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.
[6] 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.
[7] Miucic, Radovan, Syed Masud Mahmud, and Zeljko Popovic: An enhanced data-reduction algorithm for event-triggered networks, IEEE Transactions on vehicular Technology, vol.58.6, pp.2663-2678, 2009.
[8] Supriya Kelkar, Raj Kamal: Boundary of Fifteen Compression algorithm for Controller Area Network based automotive applications, International Conference on Circuits Systems Communication and Information Technology Applications (CSCITA) 2014, pp.162-167, 2014.