

データ圧縮技術 SCHC を用いた Bluetooth Mesh のペイロード拡張 Payload extension on Bluetooth Mesh by SCHC: Static Context Header Compression

向本 将規[†]

Masaki Mukamoto

[†] 株式会社東芝 研究開発センター Toshiba R&D Center

1. はじめに

近年、ビルや工場などに設置された装置の保守作業において無線システムの適用が進んでいる。そのような装置は高所やピットなど、人のアクセスが容易ではない場所に設置されていることがあり、現地での保守作業が困難であった。無線システム適用により、簡易の点検であれば保守員が各装置の設置場所に入りこむことなく、スマートフォンやタブレットを使って保守作業ができるようになる。これにより保守業務の効率化が期待できる。

ビルや工場などに設置される装置には電池駆動のものも存在し、コスト制約が厳しいものもある。そのため、保守作業の効率化のために導入できる無線システムは省電力であり低コストであることが求められる。また、スマートフォンなどとの連携を考えれば、スマートフォンに標準装備されている無線方式が望ましい。

Bluetooth Mesh[1] はこれらの要件を満たす。さらに Bluetooth Mesh はマルチホップネットワークを構成できるため、保守対象装置と保守員の作業場所の間に中継器を設置することで通信の安定性を確保できる。Bluetooth Mesh の課題は転送効率の低さである。Bluetooth Mesh は 1 つのパケットで送信できるアプリケーションデータは最大 11 バイトと小さい。ひとつのパケットに収まらないデータは分割され複数のパケットで送受信されるが、やり取りするパケット数が増えるとその分、消費電力も大きくなってしまふ。

そこで本稿では、Bluetooth Mesh のヘッダ圧縮方式を提案する。提案方式は IETF (Internet Engineering Task Force) で標準化された Static Context Header Compression (SCHC) [2] を Bluetooth Mesh 向けに応用し、独自の圧縮ルールを設計した。また、提案方式を M5Stack Core2 に実装し、通信性能評価を行う。

本稿は以下の構成である。2 章で関連研究を紹介し、3 章で Bluetooth Mesh の概要を説明する。4 章で提案方式について述べる。5 章には提案方式の実装と共に評価実験結果について議論する。6 章で本稿をまとめる。

2. 関連研究

パケットあたりのアプリケーションデータを大きくするアプローチには大きくふたつある。ひとつはパケットの構造を拡張するアプローチである。もうひとつは既存のパケットの構造のまま、パケット内のオーバーヘッドを削減し、その分をアプリケーションデータの領域に割り当てるアプローチである。

Bluetooth Mesh のパケット構造を拡張するアプローチとしては Pérez-Díaz-De-Cerio らの提案が挙げられる[3]。Pérez-Díaz-De-Cerio らは Bluetooth 5 のパケット構造を Bluetooth Mesh に適用することで 530 バイトのデータをひ

とつのパケットで送信できるようにした。Bluetooth 5 のパケット構造適用を Bluetooth Mesh スタックの下位レイヤにあたるネットワーク層とベアラ層の間で行うことで、上位レイヤのアプリケーションとの互換性を保っている。Nordic 社の Instaburst[4]も同様のアプローチで、ベアラ層でパケット構造の拡張を行う。これにより、ひとつのパケットで 498 バイトのアプリケーションデータを送信できるようになる。このようにパケット構造そのものを拡張することでより大きなデータをひとつのパケットで送信できるようになるが、ハードウェアに近い下位レイヤでの改変が必要であり、普及に課題がある。

パケット内のオーバーヘッドを削減するアプローチとしてはパケットのヘッダ圧縮が典型的な手法として挙げられる。IETF では RFC 7668[5]として BLE 上で IP パケットをやり取りする際のヘッダ圧縮方式が開発された。具体的な圧縮処理は LOWPAN_IPHC[6]である。IETF では LPWA (Low Power Wide Area network) 上で IP をやり取りする際のヘッダ圧縮方式として SCHC[2]も開発された。SCHC はルールベースの圧縮方式で LOWPAN_IPHC よりも圧縮効率が高い。なお、IETF で提案されている IP パケットのヘッダ圧縮処理は通信スタック内で実装可能なものであり、導入障壁が低い特徴も持つ。

3. Bluetooth Mesh 概要

Bluetooth Mesh は BLE 4.0 で追加された機能の一つである。通常の Bluetooth はスター型のネットワークを構成し、1 台のセントラルノードが 1 台以上のペリフェラルノードそれぞれと通信する。一方 Bluetooth Mesh ではマルチホップネットワークを構成し、ネットワーク内のノードはすべて等価であり相互通信する。パケットの送信ノードと受信ノードが直接の通信範囲内に存在しない場合、中間に位置するノードが当該パケットをパケットリレー形式で中継する。

このパケットリレー形式の中継はパケットのフラグディングにより実現されている。パケットのヘッダには TTL (Time To Live) フィールドがあり、送信ノードは TTL フィールドに 1 以上の値を設定してパケットをブロードキャストする。パケットを受信したノードはそれが自身宛でなければ中継ノードとして動作し、TTL フィールドの値を 1 だけ減じ、その結果が 1 以上であれば当該パケットをブロードキャストで再転送する。このブロードキャストによる再転送が繰り返されることで、パケットはその宛先ノードに届けられる。なお、中継ノードがパケットを再転送した後、同一パケットを他のノードから受信した場合は当該パケットを破棄する。これにより不必要な転送は抑制される。

図 1 に Bluetooth Mesh のパケット構成を示す。Bluetooth Mesh のアプリケーションペイロードは最大で 11 バイトである。送信データが 11 バイトを超える場合は、データが

分割され、複数のパケットで送信される。例えば 15 バイトのデータを送る場合、送信ノードからふたつのパケットが送信され、それぞれがネットワーク内の各ノードによりブロードキャストで再転送される。

Bluetooth Mesh の課題はそのペイロードの小ささにある。前述の通りアプリケーションペイロードは 11 バイトで、適用できるアプリケーションに大きな制約がある。アプリケーションデータをセグメント分割することでより大きなサイズのデータも送信できるが、Bluetooth Mesh はフラグディング通信を行うため多数のパケットを送信すると複製によりネットワークにパケットが爆発的に増加し、ネットワーク全体の性能が低下するおそれがある。

4. 提案手法

そこで、本稿では Bluetooth Mesh のヘッダ圧縮方式を提案する。ヘッダ圧縮によりアプリケーションペイロード領域が拡張され、従来よりも大きなデータをパケット分割無しで送信できるようになる。

提案方式では IETF で標準化された SCHC[2]を応用する。SCHCは LPWA (Low Power Wide Area network) 向けに開発された IP (Internet Protocol) ヘッダの圧縮方式で、ルールと呼ばれる圧縮規則に従ってヘッダの圧縮と展開を行う。この特徴から、あらかじめルールを送受信ノード間で共有しておく必要はあるものの、高いヘッダ圧縮効率を実現できる。SCHCはIPの圧縮を想定して開発された方式だが、ルールの記述方法など、SCHCの基本部分は汎用性が高く、他の通信プロトコルにも適用できる。

SCHCの特徴はヘッダ内のフィールドごとに圧縮要否の制御や圧縮方法の指示ができる点である。表 1 に各ヘッダフィールドに対する SCHC の圧縮ルールの設定項目の一部を示す。

表 1 SCHC のルール設定項目例

項目	説明
FID	Field ID。フィールド名。
FL	Field Length。対象フィールドのビット長。
TV	Target Value。実際のフィールド値との比較で圧縮要否判断する際に使用する値。
MO	Matching Operator 実際のフィールド値と TV の比較方法の指定。
CDA	Compression/Decompression Action。圧縮・展開方法の指定。

前述のように SCHC によるヘッダ圧縮ではパケットの送受信ノード間であらかじめルールを共有しなければならない。本方式では Bluetooth Mesh の中継ノードも SCHC の圧縮・展開処理をするため、中継ノードを含むすべてのノードに対しルールを設定することになる。

表 2 に提案方式の圧縮ルールを示す。

表 2 提案方式の Bluetooth Mesh の圧縮ルール

FID	FL	TV	MO	CDA
IVI	1	-	ignore	value-sent
NIC	7	-	ignore	value-sent
CTL	1	0	equal	not-sent
TTL	7	-	ignore	value-sent
Seq	24	0x00	equal	LSB
Src Addr	16	x2	equal	not-sent
Dst Addr	16	x3	equal	not-sent
SEG	1	0	equal	not-sent
AKF	1	1	ignore	not-sent
AID	6	x4	equal	not-sent
OpCode	8	x5	equal	not-sent
Parameters	-	-	ignore	value-sent
Net MIC	32	-	ignore	value-sent
Trans MIC	32	-	ignore	value-sent

CTL は当パケットがネットワーク制御パケットかどうかを示すフィールドである。本ルールはアプリケーションデータを送信するためのもののため、当ルールを使用しているデータは一律でアプリケーションデータに限定することができ、当フィールドを 0 として事前共有できる。そのため当該フィールドはパケットから削除可能といえる。TTL は送信者によって自由に値を定めることができる。そのため正確な値をあらかじめ把握することは難しいため、当該フィールドは圧縮せずそのまま送信するものとする。OpCode および Parameters が Bluetooth Mesh におけるアプリケーションデータとなる。OpCode はアプリケーションの種別を示す ID で、例えば照明の状態取得なら xx、制御命令なら yy といった形でアプリケーションの内容と動作によって ID が定められている。Bluetooth Mesh を用いて単一のアプリケーションを動作させる場合、もしくはアプリケーション毎に複数のルール表を作成する場合はルール表において OpCode を一意に定めることができる。そのため当該フィールドはパケットから省略することができる。

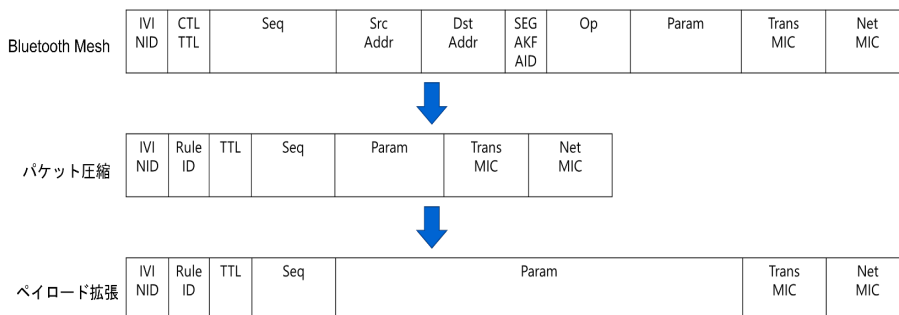


図 1 提案手法によるパケット圧縮およびペイロード拡張の流れ

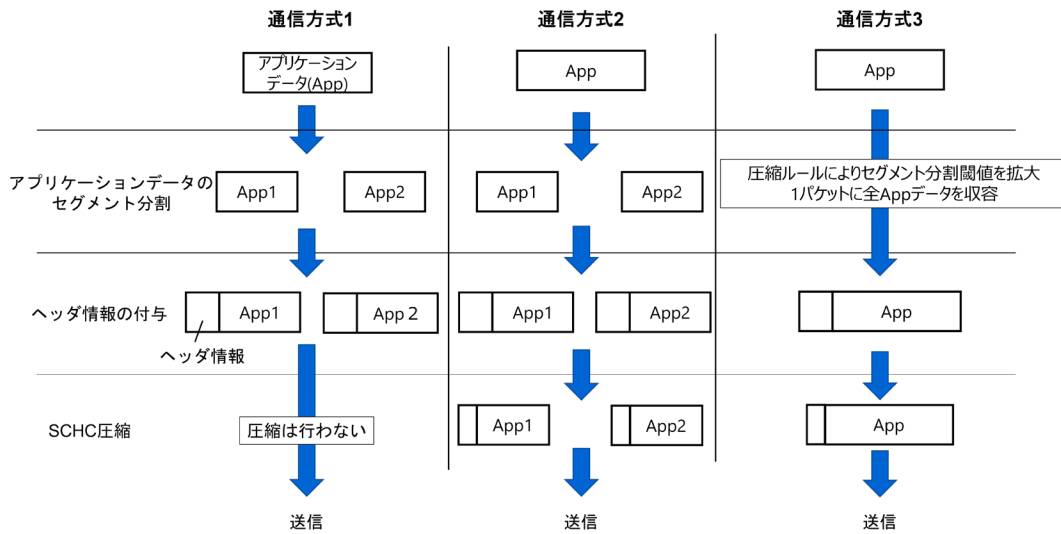


図2 評価対象とする各種通信方式概要

図1に提案手法を適用した際の packets 構造の変化を示す。上記ルールを適用することで計8バイトのフィールドを削除することができる。代わりに、各ヘッダには packets に適用すべきルールを示す1バイトのルールIDフィールドを付与する。結果7バイト分ヘッダサイズを削減でき、その分、アプリケーションペイロードサイズを11バイトから18バイトに拡張させることができる。

5. 評価

提案方式SCHCの圧縮・展開処理やSCHCの圧縮ルールをM5Stack Core2[7]のBluetooth Meshスタックに実装する¹。ファームウェアの実装はESP32[8]向け開発環境ESP-IDF[9]を用いて行った。ESP-IDFはESP32の無線機能に関する実装を備えており、Bluetooth Meshのプロトコルスタックもこちらを利用する¹。無線機間の距離は15~30cm程度とし、これは1ホップで十分にBluetooth通信ができる距離である。無線機の数数は2台または4台とし、それぞれの場合に対して後述する評価実験を行う。4台の場合もいずれの無線機対も同様の距離にあり、完全グラフのようなトポロジーを形成する。最初にアプリケーションデータ送信は特定の1台から行い、その他の無線機のうち1台を宛先としてデータ送信を行う。宛先とならない残りの2台の無線機は中継機となり、当該データを受信した際にフラッディング通信による転送を行う。

5.1 評価緒言

本評価では以下の性能を比較する。

- 通信方式1：SCHC非適用のBluetooth Mesh
- 通信方式2：SCHC圧縮のみ行うBluetooth Mesh

¹ M5Stack Core2のファームウェアのビルドにはPlatformIO 6.1.6を使用。変更したのはESP-IDF v3.2.0のBluetoothスタック実装である。

- 通信方式3：SCHC圧縮かつペイロード拡張を行うBluetooth Mesh

図2に各通信方式の動作概要を示す。通信方式1は従来のBluetooth Meshの動作で、アプリケーションデータを1 packets で送信できるサイズにセグメント分割した上で各種ヘッダを付与して送信する。通信方式2, 3はどちらもSCHCを適用して packets 圧縮を実施するが、方式2は圧縮のみを行いアプリケーションペイロードの最大値は11バイトを維持する。方式3はセグメント分割時に圧縮ルールを参照することで1 packets あたりのアプリケーションペイロードの最大値を18バイトに拡大し、それに基づいてセグメント分割を行う。これにより、他の方式ではセグメント分割を要する packets について、分割を行わず一つの packets に収容できたり分割数を減らしたりできる。

これらの通信方式において同一のアプリケーションデータを10秒おきに100回送信し、送信にかかる平均時間およびアプリケーション送信成功率を評価する。アプリケーションデータは11バイトおよび15バイトのものを使用する。前者の packets は全ての通信方式において単一 packets で送信でき、後者の packets は通信方式1, 2ではセグメント分割を要するが方式3では1つの packets に収容できるサイズである。

5.2 評価結果

2台で評価を行った際の平均送信時間および packets 送信成功率を図3, 4に示す。送信時間を見ると通信方式1および方式2は15バイトのアプリケーションデータ送信において packets の送信時間が11バイトデータ送信のおおよそ1.5~2倍となっている。これは当該データがセグメント分割により2つのフラグメントに分割されたため、それにより packets 2つ分の伝送時間がかかっていることが大きな要因と考えられる。一方ペイロード拡張を実施した通信方式3は送信時間に大きな変化は見られない。これはデータ圧縮を行って確保した領域をアプリケーションペイロードとして用いることで1つの packets で伝送を行うことができたためと考えられる。

次にパケット送信成功率を見ると、ペイロード拡張を行わない通信方式 1, 2 において 15 バイトデータの方が送信成功率が高くなっている。これは ESP-IDF の Bluetooth Mesh 実装が原因と考えられる。当実装ではセグメント分割を行う際、全フラグメントを複数回送信する仕様になっている。各フラグメントについて、複数回送信されるうち 1 つでも受信することができれば当該フラグメントは受信成功となるため通信方式 1, 2 は成功率が上昇していると考えられる。一方で通信方式 3 の 15 バイト送信はペイロード拡張によりセグメント分割を生じないため、他の通信方式のような成功率上昇が発生していないと考えられる。

次に 4 台で評価した際の送信時間および送信成功率を図 5, 6 に示す。図 5 を図 3 と比較すると、大半の場合においては 2 台の場合とほぼ変わらない値を示しているが、通信方式 1 で 11 バイト送信を行う場合において 4 台で評価した場合の送信成功率が上昇している。これは周囲の中継機によるフラッディング通信によって直接受信できなかったパケットも受信できたためであると考えられる。また、いずれの結果でも機器台数を 4 台に増やしたことによる悪影響は見受けられない。少なくとも 4 台のうち 1 台がアプリケーション送信を行う程度のネットワークでは、フラッディング通信による悪影響は顕在化しないと考えられる。

6. おわりに

本稿では、IP ヘッダ圧縮技術 SCHC を Bluetooth Mesh に応用し Bluetooth Mesh のアプリケーションペイロードを拡張する方法を提案し、評価結果から提案手法が送信時間の面で効果的なことを確認した。本手法を適用することで 11 バイトを超えるアプリケーションデータを伝送する際に送信されるパケット数が減少するため、このようなデータを伝送する環境においてネットワーク内の通信機数が多くフラッディング通信が頻繁に行われたり、高頻度にデータ通信がなされたりする場合には効用が見込まれる。

一方で本稿における検証では高々 2 つのフラグメントに分割される程度の長さのパケットでの評価に留まっており、提案手法が大きく影響する環境での評価は行っていない。ノード数やアプリケーションデータの送信頻度、データサイズ等の規模などを大きくした際にどのようなふるまいを示すかの評価については今後の課題としたい。

参考文献

- [1] Mesh Profile Bluetooth Specification V1.0.1, Bluetooth SIG, Kirkland, WA, USA. 2019
- [2] RFC 8724 SCHC: Generic Framework for Static Context Header Compression and Fragmentation <https://www.rfc-editor.org/rfc/rfc8724>
- [3] D. Pérez-Díaz-De-Cerio et al, "Speeding up Bluetooth Mesh", IEEE Access, vol. 9, pp. 93267-93284, 2021.
- [4] Nordic Semiconductor. (2020). Instaburst: Nordic Semiconductor Info-center. Accessed: May 17, 2021. [Online]. Available: https://infocenter.nordicsemi.com/index.jsp?topic=%2Fcom.nordic.infocenter.meshsdk.v5.0.0%2Fgroup__INSTABURST.html
- [5] RFC 7688 IPv6 over BLUETOOTH(R) Low Energy <https://www.rfc-editor.org/rfc/rfc7668.html>
- [6] RFC 6282 Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks <https://www.rfc-editor.org/rfc/rfc6282.html>
- [7] M5Stack Core2 <https://shop.m5stack.com/products/m5stack-core2-esp32-iot-development-kit>
- [8] ESP32 <https://www.espressif.com/en/products/socs/esp32>
- [9] ESP-IDF <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/>

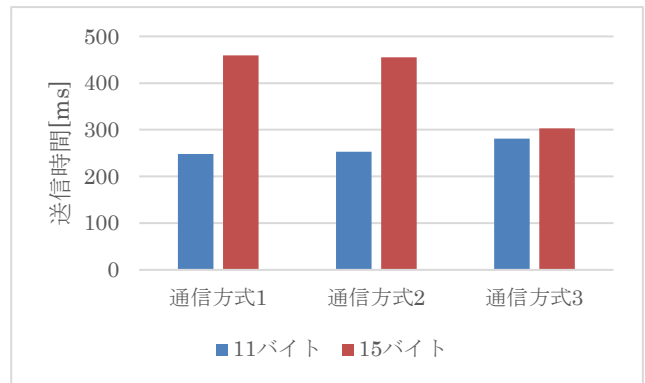


図 3 2 台環境における送信時間評価

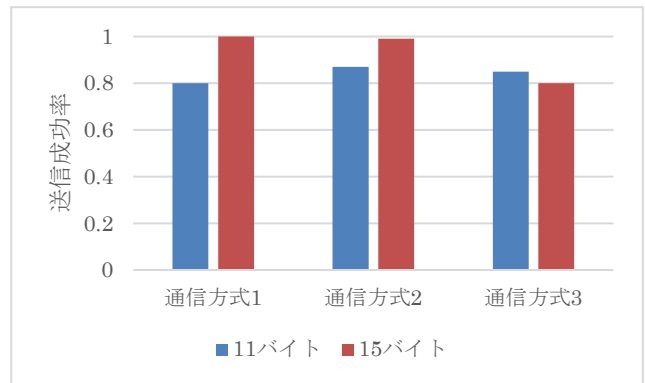


図 4 2 台環境における送信成功率評価

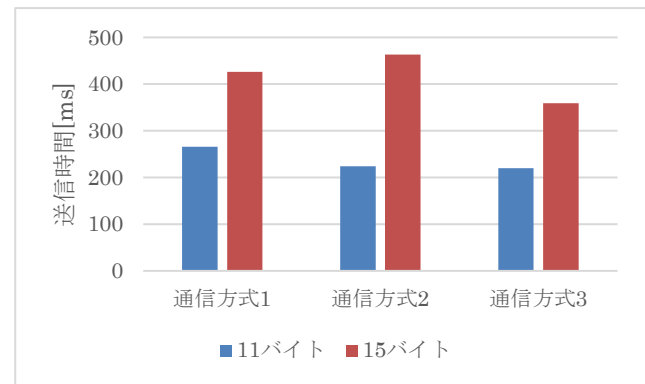


図 5 4 台環境における平均送信時間評価

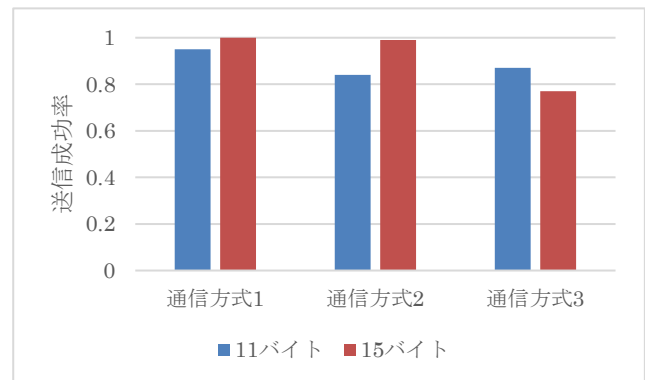


図 6 4 台環境における送信成功率評価