

LPWA を用いたネットワーク機器のアウトオブバンド管理のための通信量削減手法

田邊 広大¹ 北形 元² 長谷川 剛³

概要: ネットワーク機器の管理において、ネットワーク管理者は機器が具備する対話型操作機能を用いて障害解決や設定の変更を行う。障害発生時においても確実にネットワーク経由で機器を遠隔操作するためには、専用の管理用ネットワークを構築する必要があるが、設備コストが課題となる。ここで、長距離通信が可能な LPWA を用いることで安価に管理ネットワークを構築できれば、その利点は大きい。しかし、LPWA の通信速度は遅いため、対話型操作に使用することは難しい。そこで本稿では、ネットワーク機器のアウトオブバンド管理における通信量を削減するために、対話型操作コマンドに含まれる単語の出現確率に基づくコマンド系列の符号化手法を提案する。提案手法を用いて圧縮を行うプログラムを試作し、これを用いた実験を通して、提案手法の有効性を評価する。

1. はじめに

学校やオフィスビル、あるいは商業施設等の構内ネットワークの管理において、ネットワーク管理者は、ネットワークの障害発生時や、経路や VLAN 等の設定変更の必要が生じた際に、ネットワーク機器が具備する対話型操作機能を用いて、障害解決や設定の変更を行う [1]。そのため的手段として、ルータやスイッチ等のネットワーク機器には、シリアル通信や telnet, SSH (Secure Shell) などを用いたネットワーク経由で、機器を遠隔操作する機能が備わっている。

ネットワーク経由で機器を遠隔操作するために必要となる管理用ネットワークの構築方法の違いから、ネットワーク管理はインバンド管理とアウトオブバンド管理の 2 種類に分けられる。インバンド管理は VLAN 等のレイヤ 2 仮想化機能を用いて、提供するネットワークの物理的なケーブルやスイッチに相乗りする形で、管理用ネットワークを構築する。インバンド管理では物理的な管理用ネットワークを新たに構築する必要がないため、設備コストがかからない利点がある。しかし、障害が発生した場合や設定に人為的なミスがあった場合、提供するネットワークのユーザプレーン通信が不通になる際に、管理用ネットワークも不

通となる。

一方、アウトオブバンド管理は、提供するネットワークとは別の管理用ネットワークを用いる。アウトオブバンド管理では、提供するネットワークが不通となった場合でも管理用ネットワークが通じていれば管理可能なため、頑健な管理ネットワークを構築できるという利点がある。しかし管理用ネットワークを新たに構築するためのコストがかかる。

アウトオブバンド管理のために構築するネットワークの選択肢として、イーサネットケーブル等を用いた有線自営網が考えられる。有線自営網は、ケーブルを敷設するため工事が必要となり、構築コストが高い。他の方法として、セルラ網や Wi-Fi, LPWA 等の無線ネットワークを用いることが考えられる。セルラ網を用いる場合、導入時の通信業者との契約の手間や運用時の利用料金が必要であるが、Wi-Fi や LPWA (Low Power Wide Area) 等の自営無線網を用いる場合、それらが不要であるという利点がある。Wi-Fi はセルラ網と比較して通信距離が短いため、1 ホップで届かない場合は中継用の機器を導入する必要があり、構築コストが高くなる。一方、LPWA は Wi-Fi に比べ通信距離が長い場合、中継用の機器の数を減らすことができ、構築コストを抑えられる可能性がある。

しかし、LPWA を用いてネットワーク機器の管理をする際には、通信速度が遅いことが問題となる。例えば国内で運用可能な LoRa モジュールである株式会社アイ・ツー製の LRA1 [2] の場合、通信距離を優先する設定にした場合の通信速度の理論値は 91.6 bps である。この通信速度は

¹ 東北大学 大学院情報科学研究科
Graduate School of Information Sciences, Tohoku University
² 盛岡大学
Morioka University
³ 東北大学 電気通信研究所
Research Institute of Electrical Communication, Tohoku University

ネットワーク機器の多くが採用しているシリアル通信の標準的な速度である 9,600 bps の 1/100 以下である。このことから、LPWA を用いて対話型操作における通信データを送受信することは難しいと言える。

そこで本研究では、LPWA のような低速な通信手段を用いてネットワーク機器の対話型操作をするために、対話型操作におけるデータを圧縮する手法を提案する。提案手法における圧縮の対象は対話型操作におけるコマンドの単語列である。提案手法は、情報源アルファベットのシンボルを単語単位にすること、符号表を単語毎に持つこと、及びハフマン符号化を適用することから構成される。また、提案する圧縮手法を用いて圧縮を行うモジュールを試作して、従来の圧縮手法と圧縮率^{*1}の観点から比較し、評価する。

2. 関連研究

Jessica S. Velasco らは、[3]において、海洋気象ブイのセンサデータを LoRa を用いて送信する際に、ハフマン符号化を用いてデータを圧縮することを提案している。[3]では2つの方法でデータを圧縮している。1つ目はデータ名や単位などの定数を 16 進数値にあらかじめ割り当てておき、圧縮の際に置き換える方法である。2つ目は、センシングデータをデジタル信号処理のためのハフマン符号化を経て値を割り当てた 16 進数値に変換する方法である。後者の手法による圧縮率は最大で 40%であり、全ての手法を用いた場合の圧縮率は最大で 94.4%であった。

Pramoth Pongpunpurt らは、[4]において、センサデータを送信するための LoRa 開発ボードを実装し、実装したボードを用いて、データ転送時にハフマン符号化を用いることの効果を評価している。センサデータの転送方式として、従来の ASCII 形式とハフマン符号を用いた方式を検討し、送信データサイズの観点で比較している。ハフマン符号化では、事前に記録された 100 種類のデータセットを用いてハフマン符号を計算する。ハフマン符号を用いた方式は圧縮率 39.01%を実現した。

D. I. Săcăleanu らは、[5]において、無線センサネットワークにおいて LoRaWAN プロトコルを用いる際にデータ伝送時間が大幅に長くなり、エネルギー消費量が増加することを指摘し、センサノードにおいてデータ圧縮を用いてエネルギー効率を向上させる手法を提案している。具体的には、センサが取得したデータを送信する際、現在の取得値と前回の取得値の差分を符号化することでデータを圧縮している。

既存研究の多くはセンサデータの圧縮を目的としている

*1 本稿では、圧縮率を以下の式 (1) で定義する。

$$\frac{A - B}{A} \times 100\% \quad (1)$$

式 (1) における A は圧縮前のデータサイズ、 B は圧縮後のデータサイズである。

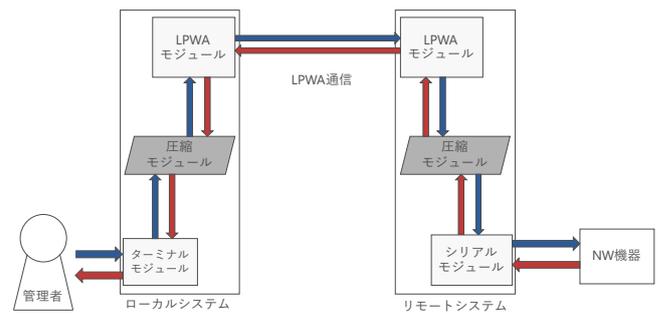


図1 提案手法を用いたネットワーク機器のアウトオブバンド管理の概要

が、本研究はネットワーク機器の対話型操作の際に発生するデータの圧縮を目的としている。センサデータを文字列として送信する際、データ名や単位などは固定であり、計測値などの逐次変化するものを除くと出現する単語の種類は少ない。しかし、対話型操作においては出現する単語の種類が多いため、同様の手法で高い圧縮率を達成することは難しい。また、機器の対話型操作のコマンドは決められた単語から構成されているため、文字単位ではなく単語単位にグループ化してからハフマン符号化をすることにより、効率的に圧縮できる可能性がある。

3. 提案手法

本章では、本研究の提案であるネットワーク機器の対話型操作のデータを圧縮する手法について述べる。

3.1 提案手法に基づくネットワーク機器のアウトオブバンド管理

図1に、提案手法を用いたネットワーク機器のアウトオブバンド管理の構成を示す。図中の矢印は対話型操作のデータの流れを表しており、青色の矢印がコマンドの流れ、赤色の矢印がその応答の流れである。管理者はターミナルモジュールを通じて対話型操作を行う。圧縮モジュールは提案手法に基づいて圧縮・復元を行うモジュールであり、対話型操作のデータはこのモジュールによって圧縮された状態で LPWA 通信を通して送受信される。管理者側にはローカルシステムを、ネットワークの機器側にはリモートシステムを配置し、それぞれ圧縮モジュールを組み込む。リモートシステムは LPWA 通信とネットワーク機器との通信を中継する役割も担う。

本稿では、管理者が入力するコマンドの圧縮手法を提案する。

3.2 圧縮手法

3.2.1 情報源アルファベット

ネットワーク機器の管理における対話型操作では、管理者が操作内容に対応するコマンドを入力し、機器はそのコマンドを解釈して指示された操作を行う。入力されるコマ

ンドは機器に搭載された管理用ソフトウェアが規定する文法に従って構成される。この時、コマンドを構成する単語の数は有限であるため、符号化における情報源アルファベットに含まれるシンボルとして文字を用いるのではなく、単語を用いることで、符号化の効率化が期待できる。そこで提案する圧縮手法では、情報源アルファベットに含まれるシンボルはコマンドに含まれる単語とする。

3.2.2 符号の構成単位

対話型操作におけるコマンドは、単語列の一部が確定すると、次に出現する単語の候補が限られる。一般に、符号化する候補が少ないと、符号化した際の符号長が短くなることから、あるコマンド中の単語に対する符号を割り当てる際、それよりも前に入力されている単語列ごとに符号を規定することにより、符号長が短くなることが期待される。

提案手法においては、コマンドの入力系列を考慮するために、図2に示すように、機器が受理可能なコマンドを木構造で表現し、圧縮モジュールに保持する。図中、Cmdはコマンドに含まれる単語、Probは親ノードからの遷移確率、Codeは割り当てられる符号をそれぞれ表している。コマンドにおける単語同士の繋がりは辺で表し、rootノードを親ノードに持つノードはコマンドの1単語目であることを表している。この木のことを以降、コマンドモデルと呼ぶ。

3.2.3 ハフマン符号化

ネットワーク機器の運用においては、機器の設置場所や運用方針、運用者の癖などに起因して、用いられるコマンドの使用頻度には偏りがあることが想定される。そのため、符号語の割り当てにハフマン符号を用いることで、効率的に符号化できると期待される。提案する圧縮手法では、コマンドモデルの各ノードに、親ノードからの遷移確率、すなわち、その単語が入力される確率を保持し、その確率に基づいてハフマン符号化を行う。

図2を用いてハフマン符号化の適用方法の例を示す。rootノードを親に持つaccess-profile, clear, connect, showのノードが保持するProbは1単語目の単語の出現確率である。一方、親ノードがaccess-profileであるignore-sanity-check, merge, replace, <Enter>はコマンドの1単語目がaccess-profileであり、ノードが保持するProbはaccess-profileに続く2単語目の単語の出現確率である。提案手法において、各ノードごとに異なる符号表を持たせるために、各ノードの子ノードに対応する符号を持たせる。

コマンドの単語列に対応する符号を取得する際は、コマンドの単語列に従ってコマンドモデルを走査する。

3.2.4 圧縮の手順

実際の運用における圧縮の手順を説明する。まず、運用を開始するための事前準備として、管理者側・機器側の圧縮モジュールに機器のコマンドモデルを持たせる。コマンドモデルでは、ノードの遷移確率を各ノードが保持し、こ

の確率を用いてハフマン符号を計算し、符号を割り当てる。

運用中に管理者がコマンドを入力すると、ローカルシステムの圧縮モジュールが保持するコマンドモデルを走査して対応する符号を取得する。この時、入力されたコマンドを履歴として保持し、コマンドモデルの各ノードが持つ遷移確率の更新のために用いる。さらに、遷移確率の更新に応じて、定期的に符号表を更新する。これにより、機器の運用時間が経過するにつれて、機器の特性や運用者の癖を考慮した効率的な符号化が行われるようになる。

4. 性能評価

4.1 実験環境

4.1.1 使用機器

ネットワーク機器として、Cisco社製のネットワークスイッチであるCatalyst 2960-S (OS: C2960S-UNIVERSALK9-M Version 15.0 (2) SE)を用いた。評価において試作した圧縮モジュールをWindows 10が稼働するPC上で動作させた。試作した圧縮モジュールはPythonで記述した。

4.1.2 コマンドモデルの取得

本実験で用いたスイッチのCLI (Command Line Interface) では、コマンド列の入力途中で“?”を入力することにより、その次に受理可能な単語の候補が出力される。その様子を図3に示す。この機能を利用し、コマンドモデルを取得した。

しかし、上述の手法では、完全なコマンドモデルを取得するには至らなかった。主な原因を以下に示す。

- コマンドには、その候補数が無限、あるいは有限だが非常に多くなる、IPアドレス等の引数が含まれる。そのため、全ての候補をコマンドモデルに含めるとコマンドモデルのサイズが発散する。
- コマンドはオプションの指定を含む場合があるが、オプションの指定順は任意であるため、全ての場合を考慮するとコマンドモデルのサイズが発散する。
- コマンドの出力結果の表示方法を正規表現で指定することができる場合、正規表現の記述パターンは有限ではないため、全ての場合を考慮するとコマンドモデルのサイズが発散する。
- 一部のコマンドの入力において、“?”を入力した際の出力形式が、他と異なる場合がある。

IPアドレス等の候補が無数にあるものは、図3に示す候補の出力においてHostnameやwordといった単語が用いられるため、コマンドモデルにおいてはそれらの単語を用いた。オプションを指定する単語、正規表現を指定する単語、及び、候補の出力形式が変わる単語に関しては、それらの単語が出現した際に、そのコマンドに対するコマンドモデルの生成を終了した。

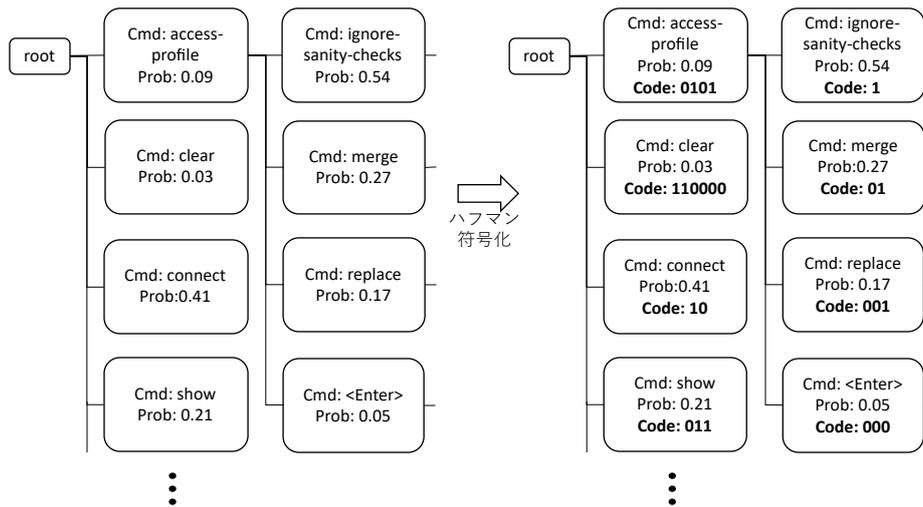


図 2 機器が受理可能なコマンドの木構造による表現と符号化の様子

```

Switch?
Exec commands:
access-enable Create a temporary Access-List entry
access-profile Apply user-profile to interface
clear Reset functions
connect Open a terminal connection
crypto Encryption related commands.
disable Turn off privileged commands
disconnect Disconnect an existing network connection
dswexec Mode-independent "dswexec" prefix support
enable Turn on privileged commands
exit Exit from the EXEC
help Description of the interactive help system
lock Lock the terminal
login Log in as a particular user
logout Exit from the EXEC
name-connection Name an existing network connection
ping Send echo messages
rcommand Run command on remote switch
release Release a resource
renew Renew a resource
resume Resume an active network connection
set Set system parameter (not config)
show Show running system information
ssh Open a secure shell client connection
sysstat Display information about terminal lines
tclquit Quit Tool Command Language shell
telnet Open a telnet connection
terminal Set terminal line parameters
traceroute Trace route to destination
tunnel Open a tunnel connection
where List active connections
    
```

図 3 CLIで“?”を入力した際に提示される受理可能な単語の候補とその説明

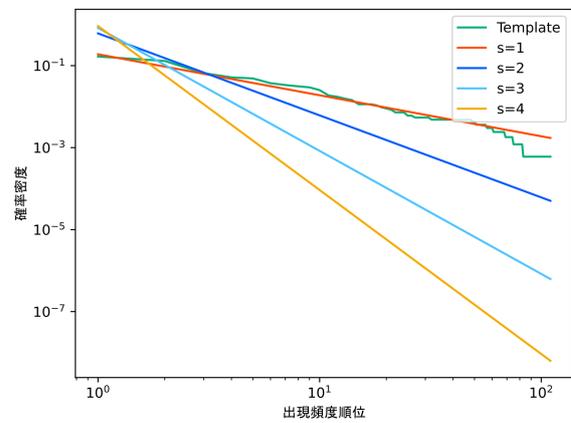


図 4 コマンドの先頭の単語の出現確率分布

4.2 単語の出現確率の決定

提案手法におけるハフマン符号化の効果を評価をするためには、コマンドモデルにおけるノード間の遷移確率に相当する、単語の出現確率を与える必要がある。実際のネットワーク機器の運用においては、過去に入力されたコマンド履歴を用いて遷移確率を決定する。本章における実験では、各ノードの遷移確率が Zipf 則 [6] に従うことを仮定して出現確率を与える。

この根拠として、Cisco が公開しているネットワーク機器の設定において入力されるコマンド列の例 [7] を用いて得られた単語の出現確率の分布と Zipf 則に基づく分布を比べた結果を示す。図 4 に、各コマンドの先頭に表れる単語の出現確率の分布と、Zipf 則におけるパラメータ s を 1, 2, 3, 4 に設定した場合の確率密度関数をプロットした結果を示す。図 5 には、コマンドに含まれる全ての単語の出現確率の分布と、Zipf 則の確率密度関数をプロットした結果を示す。これらの結果から、ネットワーク機器の対話型操作のコマンドにおける単語の出現確率分布は、 $s=1$ の Zipf

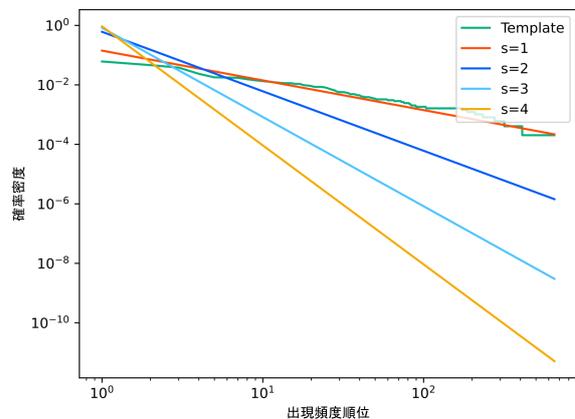


図 5 コマンドに含まれる全ての単語の出現確率分布

則に従うと見なすことができる。よって本実験では、そのようにして遷移確率を決定したコマンドモデルを用いる。

4.3 実験 1: 情報源アルファベットの影響

4.3.1 実験方法

本実験では、ハフマン符号化における情報源アルファベットの違いによる圧縮効率への影響を評価する。比較する手法は以下の2つである。

- 情報源アルファベットのシンボルを1文字単位にする場合
- 情報源アルファベットのシンボルをスペース区切りの単語単位にする場合

なお実験1では、情報源アルファベットの効果の評価を行うために、符号表をコマンドモデル全体で1つだけ持つ環境で評価する。

評価に用いるコマンドモデルにおいて、候補となる単語の出現確率を均等に設定する場合と、候補となる単語の出現確率を Zipf 則の確率分布に従って設定する場合を用いる。評価の指標として、コマンドに含まれる単語の平均符号長を用いる。それぞれの出現確率の設定に対して10,000個のコマンドを生成し、平均符号長を評価する。

4.3.2 実験結果と考察

まず、候補となる単語の出現確率を均等に設定し、コマンドを生成した場合の結果を述べる。情報源アルファベットを1文字単位とした場合は、1文字あたりの平均符号長は4.75 bitsであった。符号化を行わない場合には1文字が8 bitsで符号化されると想定すると、圧縮率は40.6%であった。情報源アルファベットを単語単位とした場合は、1単語あたりの平均符号長は6.08 bitsであった。生成されたコマンド群における1つの単語が含む平均文字数は6.30であったことから、1文字あたりの平均符号長が0.97 bitsであり、圧縮率は87.9%であった。

次に、候補となる単語の出現確率を Zipf 則の確率分布に従って設定し、コマンドを生成した場合の結果を述べる。情報源アルファベットを1文字単位とした場合は、1文字あたりの平均符号長は4.69 bitsであり、圧縮率は41.3%であった。情報源アルファベットを単語単位とした場合は、1単語あたりの平均符号長は5.44 bitsであった。生成されたコマンド群における1つの単語が含む平均文字数は7.69であったことから、1文字あたりの平均符号長が0.71 bitsであり、圧縮率は91.1%であった。

これらの結果から、情報源アルファベットを単語単位にすると圧縮効率が向上することが明らかとなった。さらに、単語の出現確率に偏りがある場合に、圧縮率がさらに向上することがわかった。

4.4 実験 2: 符号の逐次的な更新の効果

4.4.1 実験方法

本実験では、実際に提案手法に基づくシステムを運用することを想定し、符号化のために必要となる単語の出現確率を逐次的に更新した場合における、圧縮率の変化を評価

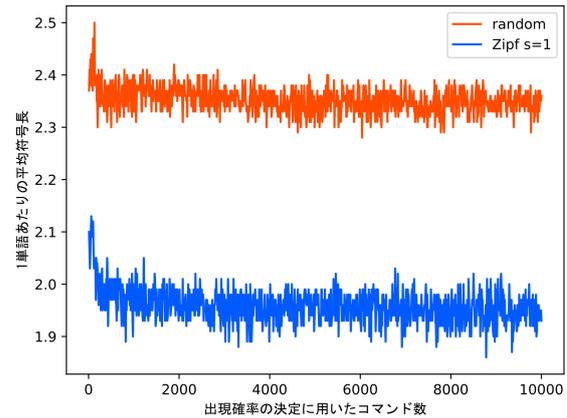


図6 実験2の結果

する。

コマンドモデルの更新、及び符号長の評価を行うための入力コマンドにおける単語の出現確率としては、実験1と同様に、均等に設定する場合と、Zipf 則の確率分布に従って設定する場合を考える。本実験において、符号表をノード毎に持たせる。実験においては、コマンドを10個生成し、それらに含まれる単語を用いて、コマンドモデルにおける各単語の出現確率を更新し、符号を更新する。その後、100個のコマンドを生成して符号化を行い、平均符号長を評価する。この手順を繰り返すことで、単語の出現確率の逐次的更新の影響を評価する。コマンドはコマンドモデルを走査することで生成する。

4.4.2 実験結果と考察

図6に、符号の更新に用いたコマンド数と、1単語あたりの平均符号長の関係を示す。図には、生成したコマンドにおける単語の出現確率を均等に設定する場合 (random) と、Zipf 則の確率分布に従って設定する場合 (Zipf $s=1$) をプロットしている。図より、単語の出現確率によらず、約500個のコマンドを用いて符号を更新することで、平均符号長が収束することがわかる。この結果から、提案手法を用いる場合に必要となるコマンド履歴のサイズが明らかとなった。

単語の出現確率を均等に設定した場合、平均符号長は約2.4 bitsに収束した。この時、1文字あたりの平均符号長は0.40 bitsであり、圧縮率は95.0%である。単語の出現確率を Zipf 則の確率分布に従って設定した場合、平均符号長は約1.95 bitsに収束した。この時、1文字あたりの平均符号長は0.29 bitsであり、圧縮率は96.4%である。

4.5 符号の構成単位が性能に与える影響

単語の出現確率を均等に設定した場合、実験1において行った、符号表を1つしか持たない場合の圧縮率は87.9%であった。一方、実験2で行った、コマンドモデルの各

ノードに符号表を持たせた場合の圧縮率は95.0%であった。また Zipf 則に従って設定した場合においては、符号表を1つだけ持つ場合の圧縮率は91.1%だったが、コマンドモデルの各ノードに符号表を持たせる場合の圧縮率は96.4%であった。これらの結果から、符号表をコマンドモデルの各ノードに持たせることで、圧縮率が高まることが分かった。

5. 結論

5.1 本研究のまとめ

本稿では、LPWAのような低速なネットワークを用いたネットワーク機器のアウトオブバンド管理を実現するために、対話型操作におけるコマンドを圧縮する手法を提案した。提案手法は、コマンド投入において順次入力される単語ごとに符号を規定し、情報源アルファベットをそれらの単語から構成し、ハフマン符号化を適用する。実験によって提案手法を評価した結果、圧縮率が最大で96.4%となることが明らかとなった。

5.2 今後の課題

今後の課題として、コマンドが引数やオプションを含む場合に対応できるように、コマンドモデルの構築方法を拡張することが挙げられる。また、実際のネットワークシステムにおけるコマンド履歴を用いた評価を行いたい。さらに、提案手法を含んだシステム全体の試作を行い、性能評価を行うことも重要である。

参考文献

- [1] 宮本貴朗, 田村武志, 平岡亮司, 松尾英普, 泉正夫, 福永邦雄. 大規模ネットワークにおける VLAN 管理システム. 情報処理学会論文誌, Vol. 41, No. 12, pp. 3234-3244, dec 2000.
- [2] 特定小電力LoRa/FSK無線モジュールハードウェア DATA SHEET LRA1.
<https://www.i2-ele.co.jp/LoRa.html>.
- [3] Jessica S. Velasco, Mariela Credo, Abeille Mae P. Salazar, Shirleen Rence S. De Ocampo, Venice Julliane C. Enriquez, Maria Victoria C. Padilla, Edgar A. Galido, Nilo M. Arago, Glenn C. Virrey, and Lean Karlo S Tolentino. Utilization of huffman coding for weather buoy system using long-range communication. *International Journal of Advanced Technology and Engineering Exploration*, Vol. 8, No. 79, 2021. Available at SSRN: <https://ssrn.com/abstract=3893048>.
- [4] Pramoth Pongpunpurt, Weerawat Khawsuk, and Nikorn Sutthisangiam. Development of huffman code for lora technology. In *Proceedings of 2019 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computing, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCOM/IOP/SCI)*, pp. 1882-1887, 2019.
- [5] D. I. Săcăleanu, R. Popescu, I. P. Manciu, and L. A. Perişoară. Data compression in wireless sensor nodes with lora. In *Proceedings of 2018 10th International Conference on Electronics, Computers and Artificial Intel-*

- ligence (ECAI)*, pp. 1-4, 2018.
- [6] George K. Zipf. *Human Behaviour and the Principle of Least Effort*. Addison-Wesley, 1949.
- [7] Default Configuration Templates.
https://www.cisco.com/en/US/docs/ios/internetwrk_solutions_guides/splob/guides/dial/cam/CAM_3.0/sw_gd/sample.html.