

卒業論文 2010年度(平成22年)

ZigBee ネットワーク上の効率的な HTTP 転送手法

慶應義塾大学 環境情報学部

学籍番号:70748511

山田 真弘

ZigBee ネットワーク上の効率的な HTTP 転送手法

近年, Twitter 等に代表されるように XML (Extensible Markup Language) や HTTP (HyperText Transfer Protocol) 等のいわゆる Web 技術を利用しているアプリケーションが増加している. また, 最近では Web 技術を用いてセンサーから実空間情報を直接取得し共有するシステムが多く提案されている. しかし, ZigBee などに代表される無線センサーネットワーク環境を用いる場合, これらは省電力, 低帯域, 高遅延という特性を持っており, そのような中で XML や HTTP を用いた REST 型インターフェイスを作成, 利用する時, センサーネットワーク内で情報の輻輳が発生してしまい, 安定した通信が行えなくなってしまう. これに対し, HTTP と同様の応答が定義され, かつ 1 パケットあたりのデータ量を短縮出来る CoAP (Constrained Application Protocol) と呼ばれる通信プロトコルの標準化が IETF で進行している. これはセンサーネットワーク上の通信を想定したプロトコルであり, 非同期通信を想定しているため遅延に対する耐性を持っている. 本研究では, まず CoAP 型の通信を実際に ZigBee ネットワーク上で動作させ, 遅延とパケット損失を測定した. その結果, ZigBee ネットワークでマルチホップする場合には, パケット損失が生じる可能性があり, CoAP が提案する UDP を用いた通信だけではアプリケーションでの再送が必要となることを明らかにした. そこで, 本研究では CoAP で定義された簡略化パケットを ZigBee の再送制御機能である APS_ACK と組み合わせる CoAP/ZigBee を提案し, 効果を定量評価した. CoAP/ZigBee を用いて XML と HTTP を用いた時と同様の通信をセンサーネットワーク上で行った時, 通信の遅延を約 40% 程低減し, 通信エラーを抑制出来ることを示した.

キーワード

1. センサーネットワーク, 2. ZigBee, 3. Constrained Application Protocol

慶應義塾大学 環境情報学部

山田 真弘

An efficient HTTP transport over ZigBee network.

Because of the worldwide availability of web server and browser, HTTP based applications are widely used in these days. Even sensor data are collected using HTTP transport. But when a wireless sensor network, such as ZigBee, is involved in a web service system, we need to consider the delay and packet loss in the wireless sensor network. Straightforward application of a web system over ZigBee entails a significant performance degradation because HTTP messages are relatively large and redundant. CoAP (Constrained Application Protocol) is proposed in IETF to solve such problems. CoAP is a compressed HTTP over UDP. This thesis evaluates the fundamental characteristics of HTTP and CoAP transports to find that the large amount of data in HTTP transport causes packet errors in ZigBee network. CoAP is effective to reduce the data amount. But it was also shown that the complete elimination of packet loss in CoAP communications is difficult. Accordingly, this thesis proposes CoAP/ZigBee, where CoAP/UDP is transported using ZigBee with application layer acknowledgement "APS_ACK". An experiment reveals that we can suppress packet delay by 40% while preserving sufficient throughput 33kbps by using CoAP/ZigBee.

Keywords :

1. Sensor NetWork., 2. ZigBee, 3. Constrained Application Protocol

Keio University , Faculty of Environment and Information Study

Masahiro YAMADA

目次

第1章 序論	1
1.1 はじめに	1
1.2 論文構成	1
第2章 関連プロジェクト・技術	2
2.1 RACOW プロジェクト	2
2.1.1 RACOW プロジェクトが目指す物	2
2.1.2 本研究の役割	3
2.2 ZigBee	3
2.2.1 低消費電力	3
2.2.2 Personal Area Network の構成	3
2.2.3 各ノードの役割	4
2.2.4 ツリー型とメッシュ型	4
2.3 本章のまとめ	5
第3章 既存手法が抱える問題	6
3.1 HTTP と XML を用いた通信の信頼性	6
3.1.1 実験環境	6
3.1.2 実験	7
3.1.3 結果	8
3.1.4 考察	11
3.1.4.1 HTTP の通信シーケンス	11
3.1.4.2 TCP が付加する情報	12
3.1.4.3 HTTP が付加する情報	12
3.1.5 問題点	13
3.2 Constrained Application Protocol	13
3.2.1 特徴	13
3.2.2 CoAP パケット構造	14

3.2.3	パケットヘッダの簡略化	14
3.2.4	CoAP の信頼性	14
3.2.4.1	実験	14
3.2.4.2	構成	15
3.2.4.3	計測結果	16
3.2.5	考察	20
3.3	本章のまとめ	21
第 4 章	CoAP/ZigBee の提案	22
4.1	提案手法	22
4.2	APS_ACK	22
4.2.1	APS_ACK によるデメリット	24
4.3	本章のまとめ	24
第 5 章	CoAP/ZigBee の評価	25
5.1	評価方法	25
5.1.1	評価手法	25
5.1.2	構成	25
5.1.3	結果	27
5.1.4	考察	30
5.2	本章のまとめ	30
第 6 章	結論	31
6.1	本研究のまとめ	31
6.2	今後の課題と展望	31
6.2.1	CoAP Gateway	32
	謝辞	34

目次

2.1	RACOW プロジェクトの構成図	2
2.2	ZigBee ツリー型 PAN 構成図	4
2.3	ZigBee メッシュ型 PAN 構成図	5
2.4	ZigBee スター型 PAN 構成図	5
3.1	Digi XBee モジュールと XBee プログラミングボード	7
3.2	ZigBee リンク構成	8
3.3	HTTP 通信平均応答時間	9
3.4	HTTP 通信エラーレート	9
3.5	応答時間ヒストグラム	10
3.6	応答時間累積グラフ	10
3.7	HTTP 通信シーケンス	11
3.8	CoAP パケット構造	14
3.9	CoAP で圧縮された HTTP リクエストパケットヘッダ	15
3.10	CoAP で圧縮された HTTP 応答パケットヘッダ	15
3.11	実験構成図	16
3.12	CoAP 平均通信時間	17
3.13	CoAP 通信エラーレート	18
3.14	平均通信時間	19
3.15	通信エラーレート	19
3.16	CoAP シーケンス図	20
4.1	APS フレームフォーマット	23
4.2	APS_ACK を用いない通信シーケンス	23
4.3	APS_ACK を用いた通信シーケンス	23
4.4	APS_ACK を用いた再送制御シーケンス	24
5.1	XBee 通信シーケンス	26
5.2	実験構成図	26

5.3	APS_ACKの有無による応答時間ヒストグラム	28
5.4	通信スループット	29
5.5	エラーレート	29
6.1	CoAP Gateway	32

表 目 次

3.1	実験に使用した PC とソフトウェア	6
3.2	XBee モジュール設定詳細	7
3.3	実験に使用した機材とソフトウェア環境	16
3.4	XBee モジュール設定詳細	17
5.1	実験に使用した機材とソフトウェア環境	27
5.2	XBee モジュール設定詳細	27

第1章 序論

1.1 はじめに

近年のアプリケーションは Twitter[1] 等に代表されるように XML(Extended Markup Language) や HTTP[2](Hyper Text Transfer Protocol) を用いた REST 型インターフェイスが用いられることが多い。そのため, patchube[3] のようなセンサーから直接実空間情報を収集し共有するシステム等に関しても同様の技術を用いた提案が多くなされている。

しかし, このようなシステムを ZigBee[4] に代表されるような省電力, 高遅延, 低帯域な無線センサーネットワーク上で構築する場合, XML や HTTP をそのまま用いると問題が生じる。これは HTTP が TCP[5](Transmission Control Protocol) や IP[6](Internet Protocol) を前提としていることや, XML タグ表記が冗長なためであると考えられる。無線センサーネットワークで用いられるプロトコルのペイロードサイズは非常に小さく, また, 帯域も広く確保できない。そのような環境の中で1パケットあたりのデータサイズが大きい通信を行おうとするとパケットの断片化が多く発生する。これにより, パケットがエラーに遭遇する機会が増加してしまう為, 安定した通信が行えなくなってしまう。本研究では, それらの問題を解決し, センサーネットワーク内での安定した通信が行えるようになることを目的とし, 通信の効率化を目指す。

1.2 論文構成

本論文は全6章で構成される。第2章では, 本研究に関連する技術, プロジェクトについて整理し, 本研究の立ち位置を明確にする。第3章では, XML と HTTP, 及び CoAP をメッセージプロトコルとして用いた際の問題点について実験を交えて整理し, 解決すべき問題点を明確にする。第4章では, 提案手法である CoAP/ZigBee, 及びその中で用いる要素技術である ZigBee APS-ACK に関して述べる。第5章では, 提案手法を用いて実環境で通信を行い, 通信品質に関して評価を行う。第6章では, 本研究の結論と今後の展望を述べる。

第2章 関連プロジェクト・技術

本章では本研究に関連するプロジェクト, 及び関連技術について整理する.

2.1 RACOW プロジェクト

本研究は RACOW プロジェクト [7] に関連して行われているものである. RACOW とは RFID Auto-Commissioning Open system with WiMAX の略称であり, 総務省平成 21 年度第 2 次補正予算「ネットワーク統合制御システム標準化等推進事業」における WiMAX を利用したデータ収集システムによる環境負荷低減の実証を行うプロジェクトである.

2.1.1 RACOW プロジェクトが目指す物

RACOW プロジェクトが目指すものは, 機器やデバイスに関連するエネルギーサービス, 情報サービスを誰もが自由に追加し, 消費者が状況に応じて取捨選択できるようにするシステムである. また, RACOW プロジェクトでのセンサーネットワークにおいては省電力短距離通信規格である ZigBee を選定している.

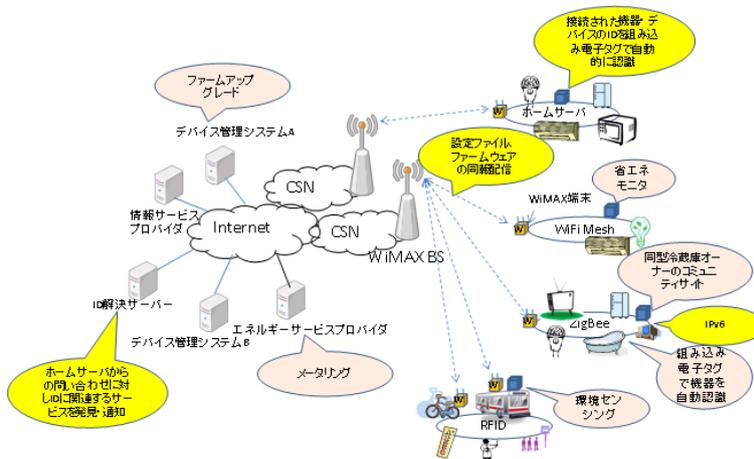


図 2.1: RACOW プロジェクトの構成図

2.1.2 本研究の役割

本研究は,RACOW プロジェクト内の家庭内センサーネットワークに着目し,センサーネットワーク内での通信の効率化に関して研究を行う.

2.2 ZigBee

本研究では前述の家庭内センサーネットワークを実現するための通信技術として ZigBee[4]を使用している. ZigBee は前項で述べた RACOW プロジェクトで利用する通信技術としての要求を満たす, 次のような特徴を持つ.

- 低消費電力
- マルチホップネットワーク
- ISM バンドである 2.4GHz 帯を利用している端末もあり, それらを国内で利用する場合は機器が技術基準適合認定を得ていれば無線局免許は不要

2.2.1 低消費電力

ZigBee は消費電力が極めて低く出来る通信プロトコルである. ノードのスリープ状態が仕様で定められており, 通信を行わない場合はほとんど電力を消費しない. スリープ状態のノードが宛先のデータはネットワークを構成するノードがキャッシュするため, スリープ状態のノードは待機受信を行なわない. しかしそれゆえに, 低速かつ短距離での通信に限定されており, 通信速度は理論値で 250Kbps である. このような特徴をもつ ZigBee は 物理レイヤー, 論理レイヤーは IEEE 802.15.4[8] で標準化され, それ以上のレイヤーは ZigBee Alliance[4] により仕様が策定されている. そのため, 安価な製品が存在し, 同様に安価に製造されるセンサ群等を用いたシステムで利用する通信技術として有用である.

2.2.2 Personal Area Network の構成

ZigBee により構築されたネットワークは Personal Area Network(PAN) と呼ばれ, 以下の 3 種類のノードで構成されている.

- Coordinator

- Router
- End Device

2.2.3 各ノードの役割

各ノードはそれぞれ以下の役割を持って PAN ネットワークを構築している。

- ”Coordinator”は”Router”としての機能を持つと同時に ネットワーク全体の制御を行う。
- ”Router”はマルチホップネットワークを構築し、データの中継を行う。
- ”End Device”は”Router”と”Coordinator”に依存して通信を行う。

2.2.4 ツリー型とメッシュ型

ZigBee は、ツリー型のネットワーク、メッシュ型のネットワーク、スター型のネットワークを構築することができる。本研究は、ツリー型のネットワークを使用するという前提で研究を行う。

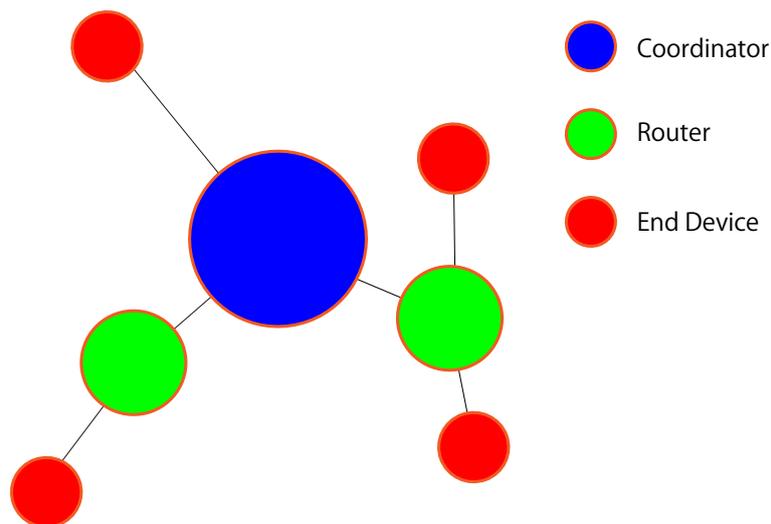


図 2.2: ZigBee ツリー型 PAN 構成図

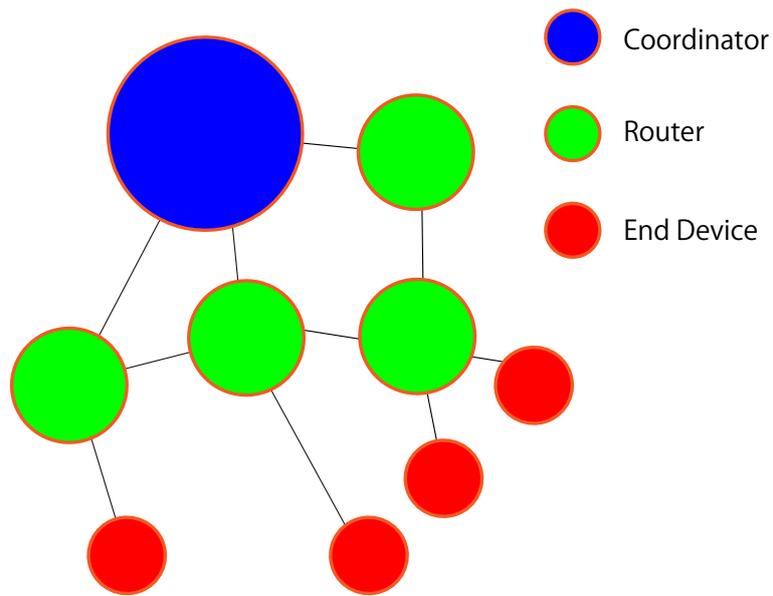


図 2.3: ZigBee メッシュ型 PAN 構成図

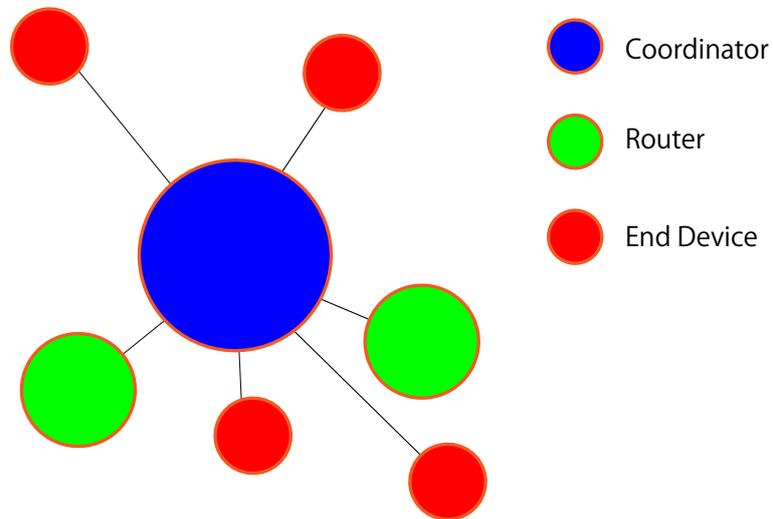


図 2.4: ZigBee スター型 PAN 構成図

2.3 本章のまとめ

本章では, 本研究に関連するプロジェクト及び, 関連技術に関して整理を行い, 本研究が着目している点を明確にした. 次章では, 無線センサーネットワーク上で XML や HTTP を用いて情報通信を行う際の問題点について実験を交えながら述べる.

第3章 既存手法が抱える問題

本章では、XML と HTTP を用いた通信及びセンサーネットワーク上で利用した場合の問題点について実験を交えて述べ、本研究が解決しなければならない点を明確にする。

3.1 HTTP と XML を用いた通信の信頼性

RACOW プロジェクトの家庭内センサーネットワークでは ZigBee を用いているが、前章で述べたように ZigBee は転送速度が低く、かつ通信可能範囲が短いため、通信が安定しにくい。そのような環境で、今日のアプリケーションインターフェイスとして一般化している XML と HTTP を用いてセンサーネットワーク上で通信を行った場合、どのような状態になるか実験を行った。

3.1.1 実験環境

実験環境はクライアント PC とサーバー PC 間を ZigBee 端末である Digi XBee[11] を用いてシリアル通信している状態にし、次に Point to Point Protocol[12](PPP) を用いて IP アドレスを両 PC に割り振った。そしてクライアントとサーバーにそれぞれソフトウェアを実行させ以下の実験を行った。なお、実験に用いた機材やソフトウェア、設定項目を以下に示す。

	機材名	動作ソフトウェア
Server	Fujitsu MG50S	Apache 2.2.3
Client	Lenovo ThinkPad T510	ruby 1.8.7 patchlevel 72 [i386-cygwin]

表 3.1: 実験に使用した PC とソフトウェア

3.1. HTTP と XML を用いた通信の信頼性

	Server	Client
Firmware type	End Device	Coordinator
利用モジュール	Digi XBee Series2	
使用周波数	2445MHz(19Ch)	
使用PANID	1222	
通信ボーレート	115200bps	

表 3.2: XBee モジュール設定詳細

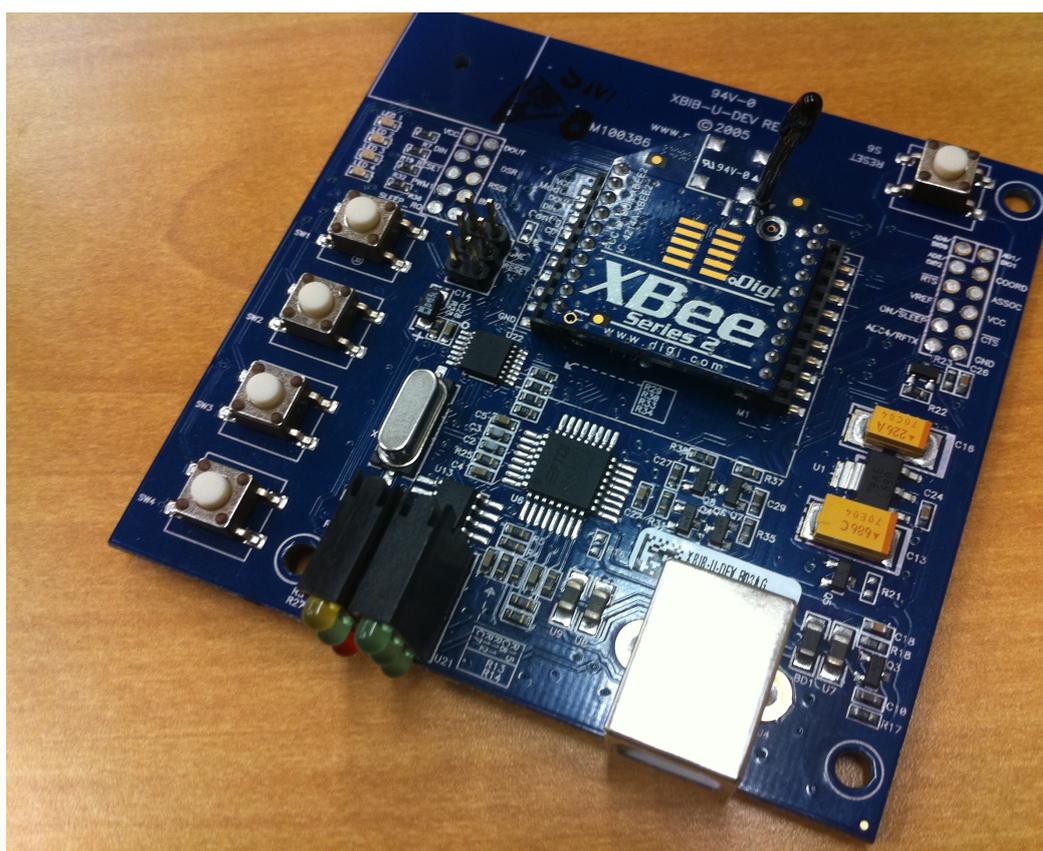


図 3.1: Digi XBee モジュールと XBee プログラミングボード

3.1.2 実験

上記実験環境を整えた上でクライアント側でテスト用の Ruby スクリプトを動作させ, HTTP リクエストをサーバー側に対して 100 回連続試行し, サーバーから HTTP の応答が送信されるまでにかかった時間を計測する. また, ZigBee の通信を中継する Router ノードを交え, マルチホップ環境にした場合の応答時間に関しても計測を行い, 評価する. なお, 下図に

ZigBee のリンク構成, 使用機材と環境, XBee の設定を示す.

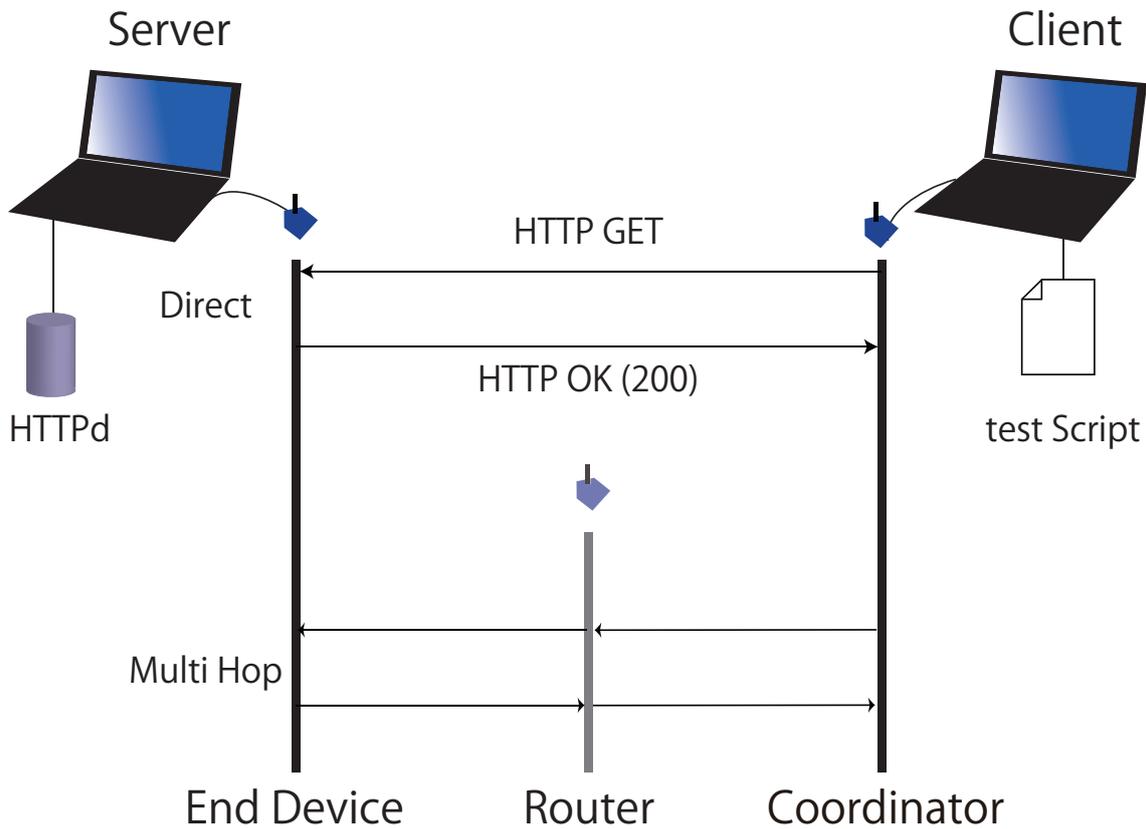


図 3.2: ZigBee リンク構成

3.1.3 結果

以上の実験を行った結果のグラフ及び, 上記実験の応答時間のヒストグラム, 累積をグラフにしたものを図 3.3-3.6 に示す.

3.1. HTTP と XML を用いた通信の信頼性

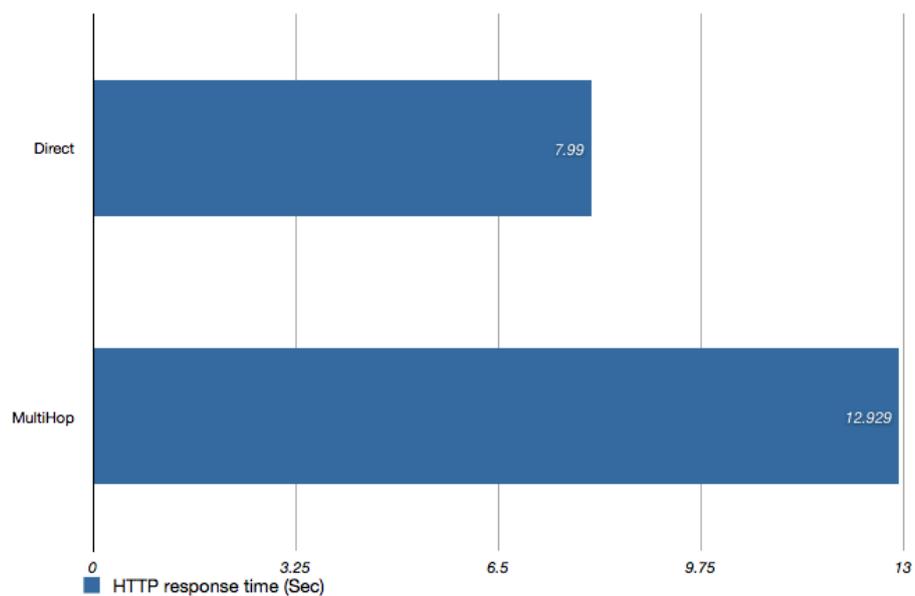


図 3.3: HTTP 通信平均応答時間

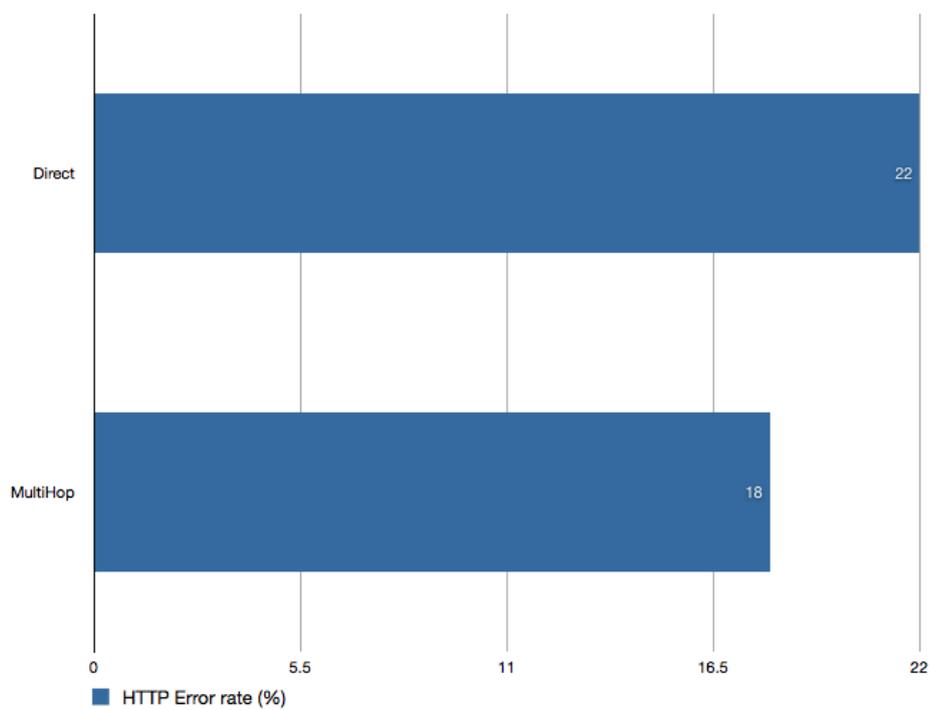


図 3.4: HTTP 通信エラーレート

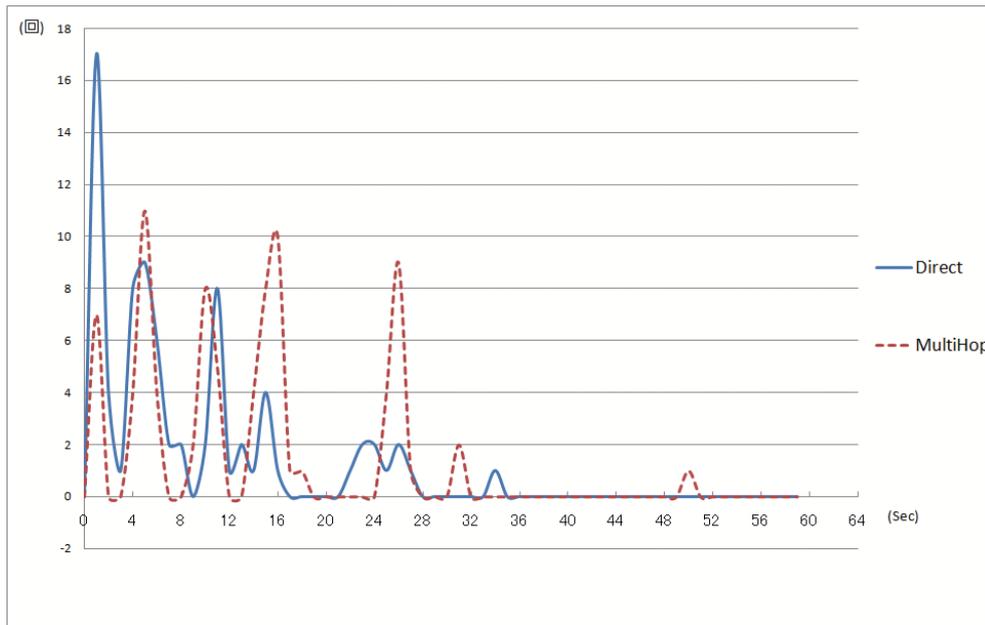


図 3.5: 応答時間ヒストグラム

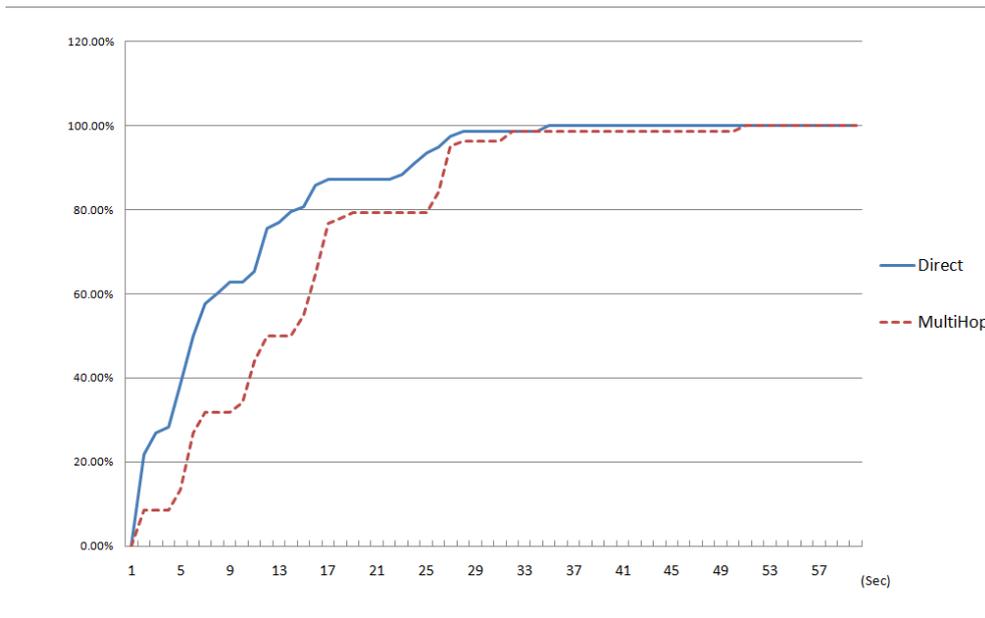


図 3.6: 応答時間累積グラフ

3.1.4 考察

以上の結果より, マルチホップ環境では, 通信にかかる時間の上昇が確認できる. また, 通信エラーレートがマルチホップ環境下では低下しているが, これは通信遅延が多く発生したものの, タイムアウトまでの時間内で取まったことに起因していることがヒストグラムより判断できる. このような結果が生まれる原因としては HTTP が TCP (Transfer Control Protocol)[5] を用いており, 1 つのデータ転送に必要なパケット数が多いことと TCP 通信及び HTTP 通信が付加するパケットヘッダによって通信が冗長になることが考えられる.

3.1.4.1 HTTP の通信シーケンス

HTTP は以下のシーケンスで通信される.

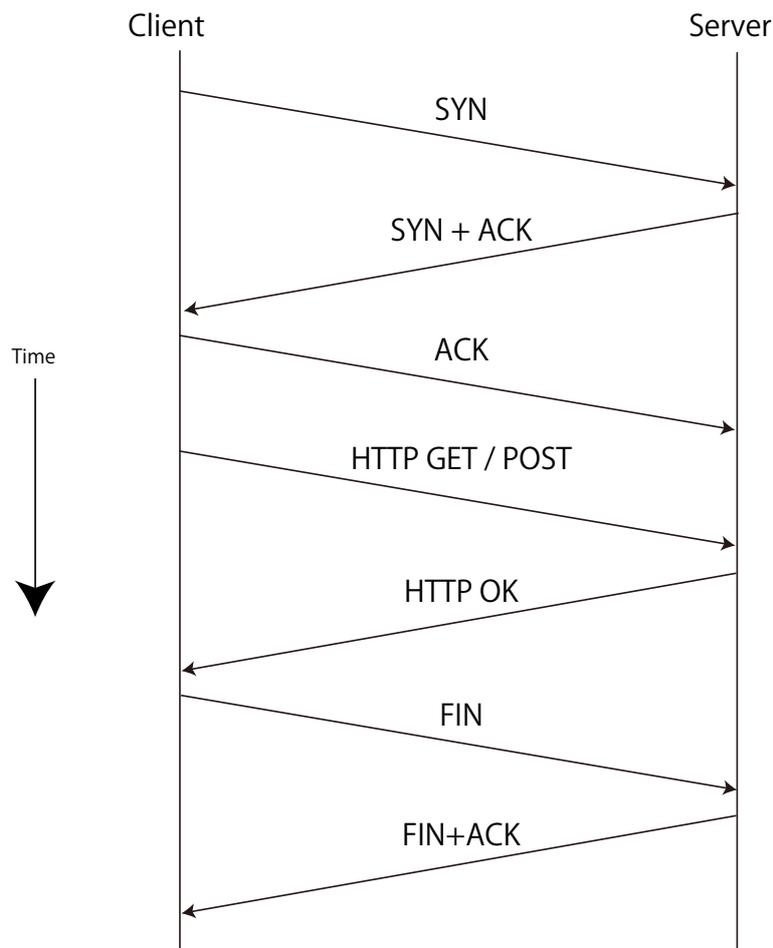


図 3.7: HTTP 通信シーケンス

HTTP は TCP を用いているため, Three-way Handshake を行いクライアント, サーバー間の接続を確立してから本来のデータ通信を行う. また, 通信終了に関しても

サーバー側からの Acknowledgement を待つ。このため、エラーなく通信が行えた場合は 7 パケット分のデータが送信されるが、エラーが発生した場合パケットの再送が発生するため、ZigBee 伝送路で帯域の輻輳が発生する。

3.1.4.2 TCP が付加する情報

TCP が付加する情報はオプションを除くと以下のものがある。これらは 20 バイトのデータ長を IP パケットに対して付加する。

- 送信元ポート情報
- 宛先ポート情報
- シーケンス情報
- 通信状態
- Acknowledgement 番号情報
- ウィンドウサイズ
- チェックサム

3.1.4.3 HTTP が付加する情報

HTTP がヘッダとして付加する情報には以下のような物などがある。状況によってこれらのサイズや項目は変動するが、本実験での HTTP リクエストパケットでは約 50 バイト、応答パケットでは約 300 バイトの大きさを持っていた。

- リクエストメソッドの種類
- HTTP のバージョン情報
- 時間
- サーバー情報
- 最終変更日時
- E-tag 情報
- コンテンツの種類

3.1.5 問題点

ZigBee のペイロードに利用出来る領域は非常に小さい。これは最大 127 バイトの ZigBee のパケットサイズから ZigBee MAC ヘッダ, ZigBee Network ヘッダ, ZigBee APS ヘッダ, ZigBee FCS それぞれの大きさをそれぞれ差し引くことで求められる。計算すると、ペイロードとして利用出来る領域は最大で 100 バイトしかないため, HTTP 通信を用いてデータを送受信するためにはいくつもの ZigBee パケットを送受信しなければならない。そのため, ZigBee パケットが 1 つパケットロスしただけで通信が成立しなくなってしまう恐れがある。さらに, TCP での再送制御などが働き, 通信を維持しようとパケットを送り直そうとするため, 結果 ZigBee の伝送路が輻輳してしまい, 通信のエラーレートの上昇や応答時間の遅延などが発生してしまう。その上, これらをマルチホップ環境におき, パケットの中継を行うと, 各ノード間で同様の事象が発生し, 通信状態が悪化してしまっているのではないかと考える。

3.2 Constrained Application Protocol

前項で挙げた問題を解決するため, IETF では CoAP (Constrained Application Protocol) [9] をメッセージプロトコルとして XML と HTTP に代わり利用することが提案され, 標準化が進んでいる。

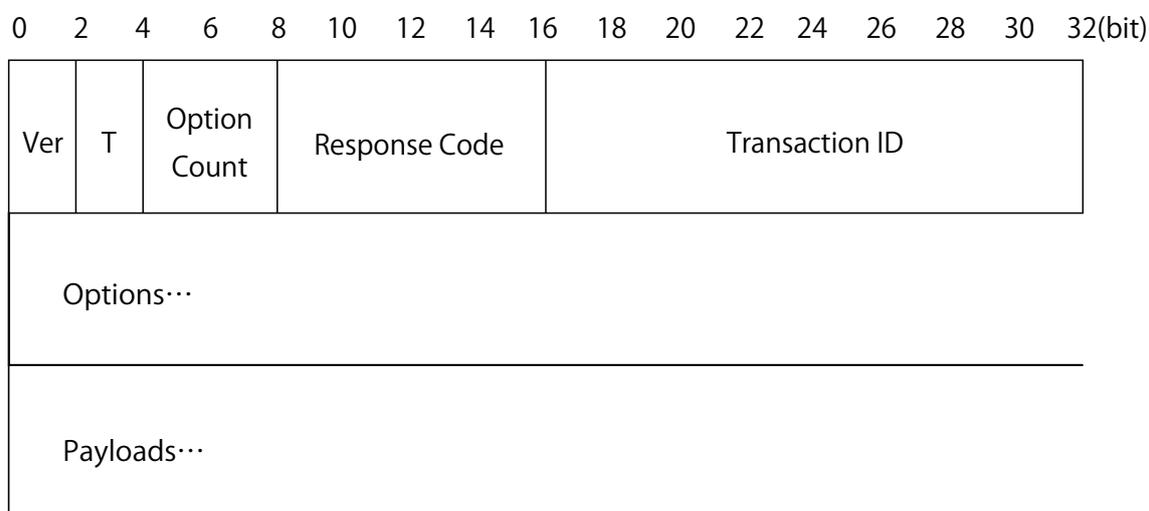
3.2.1 特徴

CoAP は Internet Draft (draft-ietf-core-coap) として現在策定作業中の通信プロトコルである。本プロトコルは低速通信, 及びマシンリソースが望めない組み込み型デバイスを用いたセンサーネットワーク上での通信を想定したプロトコルである。CoAP の特徴としては以下の点が挙げられる。

- 非同期通信のサポート
- パケットヘッダの簡略化
- HTTP と同様の応答マッピングを持っている
- UDP (User Datagram Protocol) [10] を利用

3.2.2 CoAP パケット構造

CoAP のパケット構造は以下のようにになっている。パケットごとに TransactionID が割り振られており、これを用いて非同期通信を可能にしている。



Ver:CoAP Protocol Version

T: Transaction Type

図 3.8: CoAP パケット構造

3.2.3 パケットヘッダの簡略化

前項で実験した HTTP 通信のリクエストパケットに内包されるヘッダのサイズはオプション部分を除き約 50 バイトであったが、CoAP では同様の内容を保持したままで 4 バイトまで減らすことができる。さらに応答パケットも、実験時は約 300 バイト程度大きさのヘッダサイズを最小で 4 バイトまで減らすことが可能になっている。なお、先述したの実験のパケットヘッダはそれぞれ図 3.9,3.10 のように表せる。

3.2.4 CoAP の信頼性

CoAP は前項で挙げたように通信の簡略化、通信量の削減を可能にするが、仕様として UDP を用いているため、通信経路上でデータを損失してしまう可能性が考えられる。それを明らかにするべく、以下の実験を行った。

3.2. CONSTRAINED APPLICATION PROTOCOL

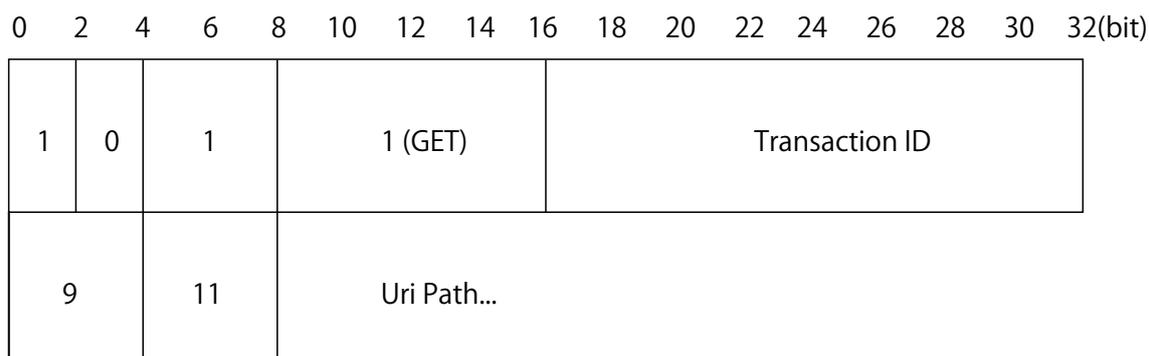
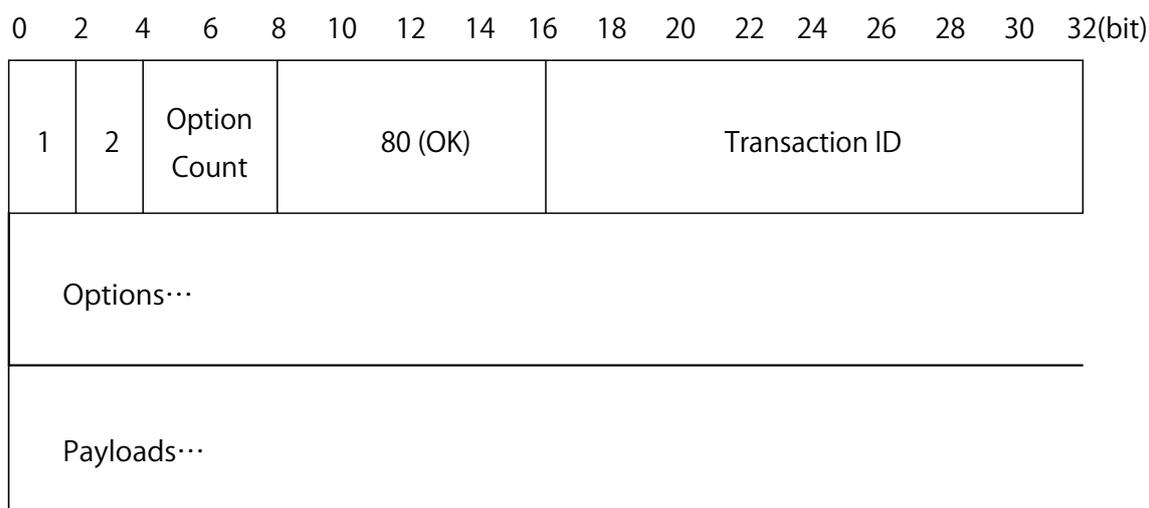


図 3.9: CoAP で圧縮された HTTP リクエストパケットヘッダ



Ver:CoAP Protocol Version

T: Transaction Type

図 3.10: CoAP で圧縮された HTTP 応答パケットヘッダ

3.2.4.1 実験

前項で述べた実験と同様に Digi XBee モジュールを利用し, サーバー側 PC とクライアント側 PC 間に Point to Point Protocol を用いて IP アドレスを割り振り, Ruby を用いて実装した CoAP クライアントとサーバーによる CoAP 通信を連続で 100 回発生させ, 応答までの時間とエラーレートを計測した. また, ZigBee の通信路に ZigBee Router を介したマルチホップ構成にした場合の計測も行った.

3.2.4.2 構成

実験構成, ZigBee のリンク構成, 用いた機材とソフトウェア, XBee の設定は表 3.3, 3.4 の通

りである.

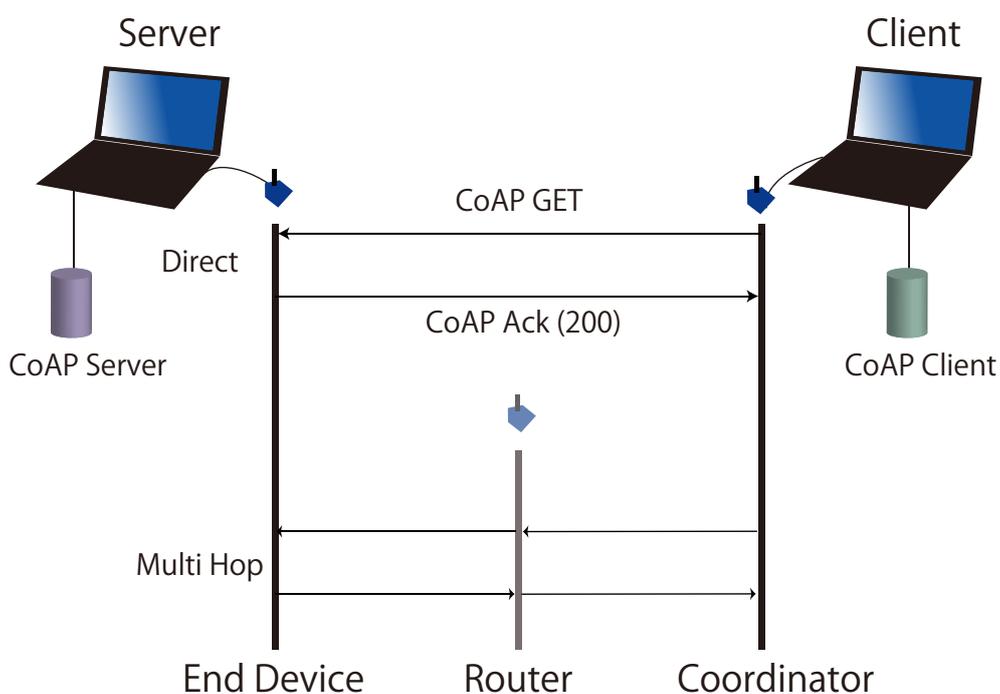


図 3.11: 実験構成図

	機材名	使用ソフトウェア,Version
Server	Fujitsu MG50S	ruby 1.8.6 patchlevel 111 [i386-cygwin]
Client	Lenovo ThinkPad T510	ruby 1.8.7 patchlevel 72 [i386-cygwin]

表 3.3: 実験に使用した機材とソフトウェア環境

3.2.4.3 計測結果

計測結果を図 3.12-のグラフに示す.

3.2. CONSTRAINED APPLICATION PROTOCOL

	Server	Client
Firmware type	End Device	Coordinator
利用モジュール	Digi XBee Series2	
使用周波数	2445MHz(19Ch)	
使用 PANID	1222	
通信ボーレート	115200bps	

表 3.4: XBee モジュール設定詳細

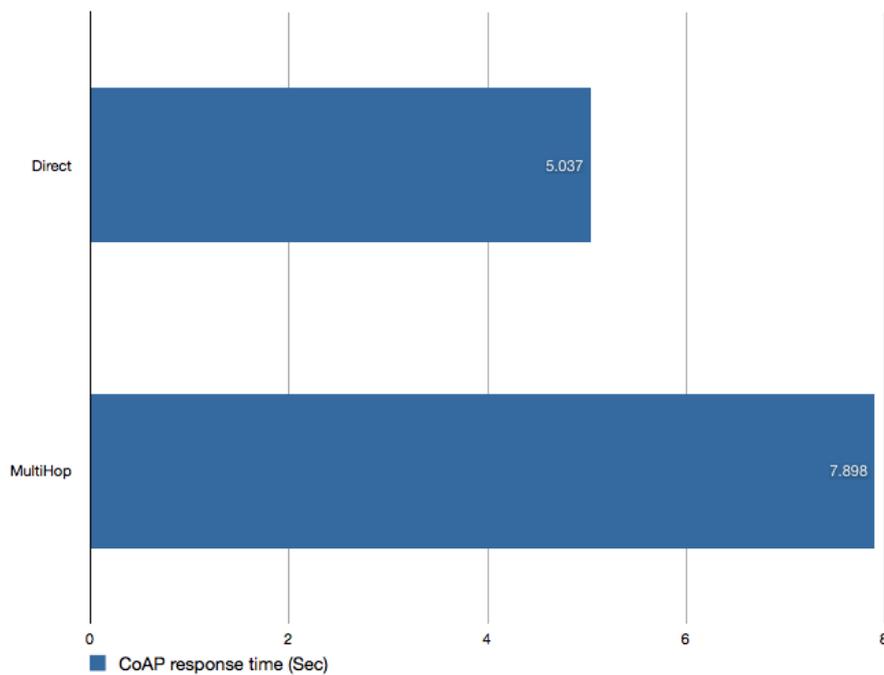


図 3.12: CoAP 平均通信時間

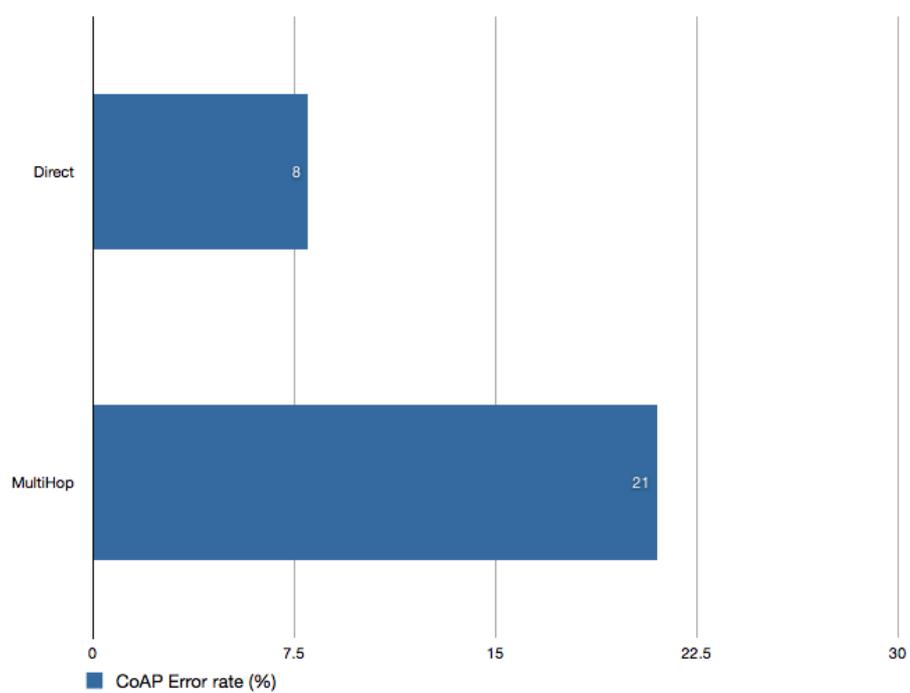


図 3.13: CoAP 通信エラーレート

3.2. CONSTRAINED APPLICATION PROTOCOL

また,HTTP 通信時の結果と比較すると次のグラフのようになる.

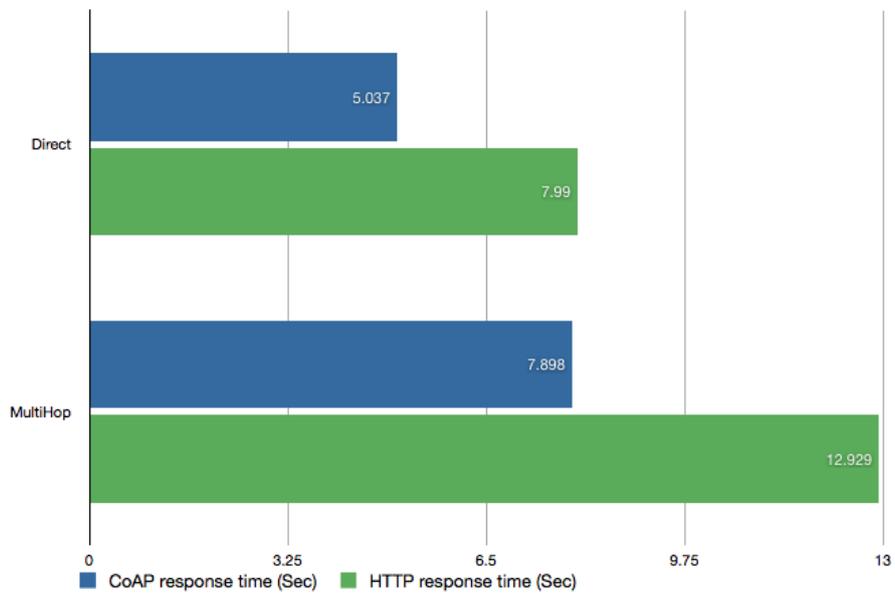


図 3.14: 平均通信時間

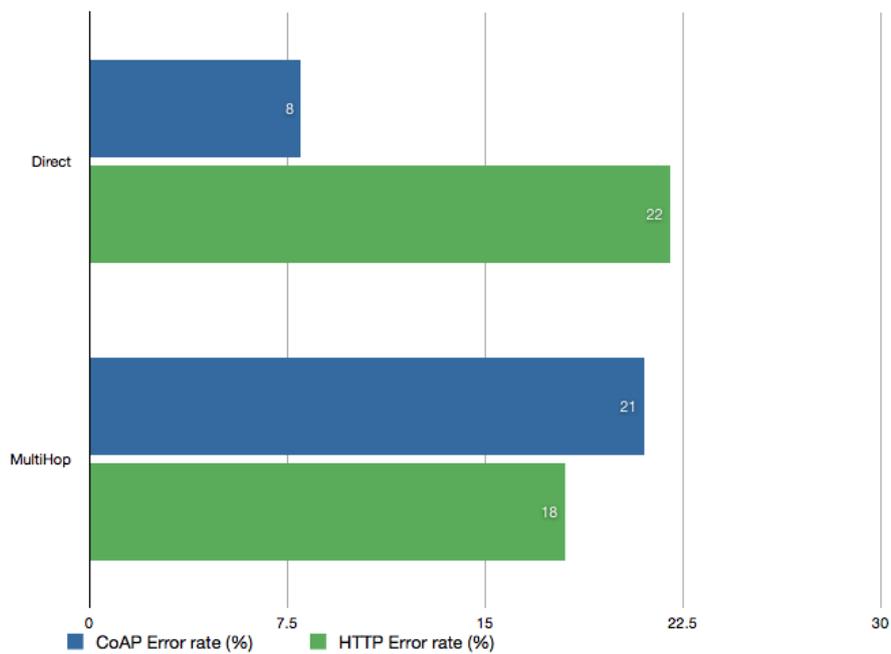


図 3.15: 通信エラーレート

3.2.5 考察

上記の計測結果より,CoAP を用いたときは XML と HTTP を用いた通信よりも遅延を 40%程度短くし,さらに,直接通信環境下ではエラーレートを 8%まで削減出来ることを示せた. これは CoAP によるパケットサイズ圧縮や通信シーケンスの簡略化により,ZigBee のパケット断片化回数を低減させることが出来たためではないかと考えられる.

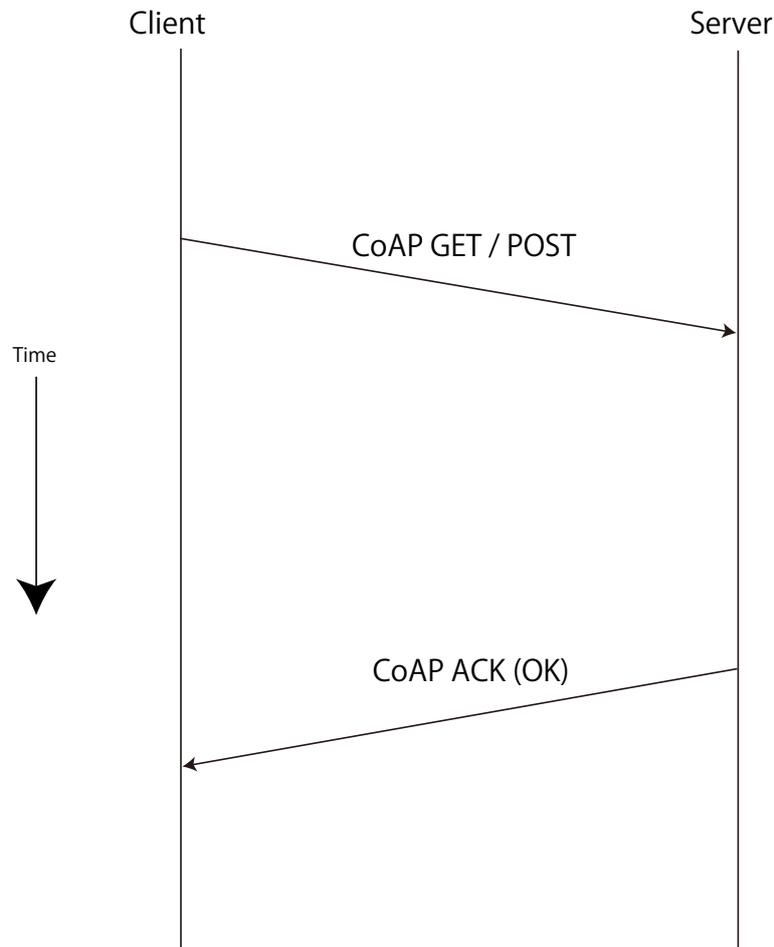


図 3.16: CoAP シーケンス図

また,エラーレートに関しては HTTP 通信のほうがマルチホップ環境下では低下しているが,HTTP は TCP による再送制御がかかるため,遅延は発生するもののエラーレートが低くなっているのではないかと考える. しかし,それぞれの通信を成功させるために最低限必要な IP パケット数は CoAP が 2 パケット,HTTP が 7 パケット必要であることを考慮すると XML と HTTP を用いる場合,パケットの損失機会は CoAP を用いた時より多く発生するのではないかと考察できる. しかし,CoAP は仕様上 UDP を用いているため,通信失

敗時に再送制御等を行うことができずデータを失ってしまうという問題が未だに残ってしまう。これを解決するためにトランスポートレイヤー以下での再送制御が必要である。

3.3 本章のまとめ

本章では既存手法として一般的な XML と HTTP を用いた通信, 及び CoAP を用いた通信をセンサーネットワーク上で用いる際に発生する問題点について実験を交えてまとめた。次章では, 本研究の提案手法と, 提案手法で用いる技術について述べる。

第4章 CoAP/ZigBeeの提案

本章では, 提案手法について述べ, その中で用いている ZigBee APS Acknowledgement について説明する.

4.1 提案手法

前章での実験と考察を踏まえ, 簡略化された通信プロトコルを用いて1データの毎に必要な通信量を抑えることで, センサーネットワーク上の通信遅延の短縮や, 通信品質の向上を図ることが出来る事は示したが, CoAP は UDP を用いて通信しているため, ZigBee 通信路上でデータを失ってしまった場合の回避手段がない. しかし, 前章での HTTP を用いた実験で示したように, 再送制御を TCP 等を用いてトランスポートレイヤー以上で行うと ZigBee のペイロードとして利用出来る領域の一部をプロトコルのヘッダで占有してしまうため, パケットの断片化を誘発することが考えられる. そこで本研究では CoAP と ZigBee の再送制御機能である APS_ACK (Application Support sub-layer Acknowledgement) を組み合わせて通信を行う CoAP/ZigBee を提案する.

4.2 APS_ACK

APS_ACK は ZigBee の機能として提供されている再送制御機能である. ZigBee の仕様 [4] によりユニキャスト通信のみに限定されるが, 通信相手に対してデータの転送が成功したときに APS Acknowledgement パケットを送付するように要求することが出来る. これは ZigBee APS フレームフォーマットの Frame Control オクテット内に定義される Acknowledgement Request フィールドの値を true にすることにより使用することが出来る.

APS_ACK 機能を有効にしているパケットを送信した後, 定められた時間を経過しても APS Acknowledgement パケットを受信出来ない場合, その通信は失敗したものとみなされ, データの再送が行われる.

4.2. APS_ACK

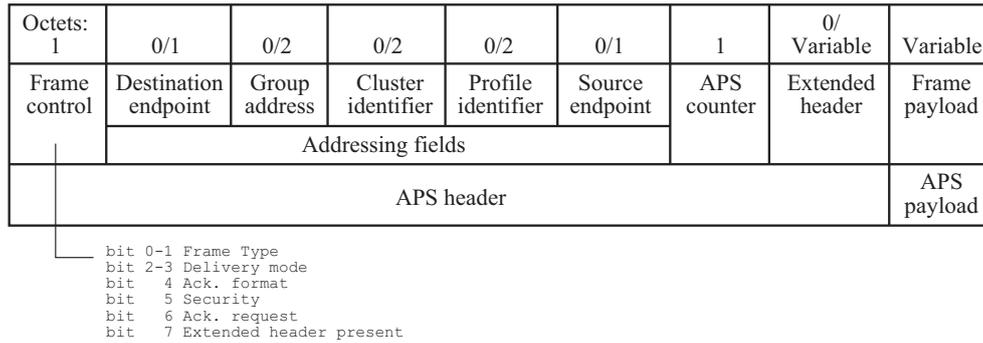


図 4.1: APS フレームフォーマット

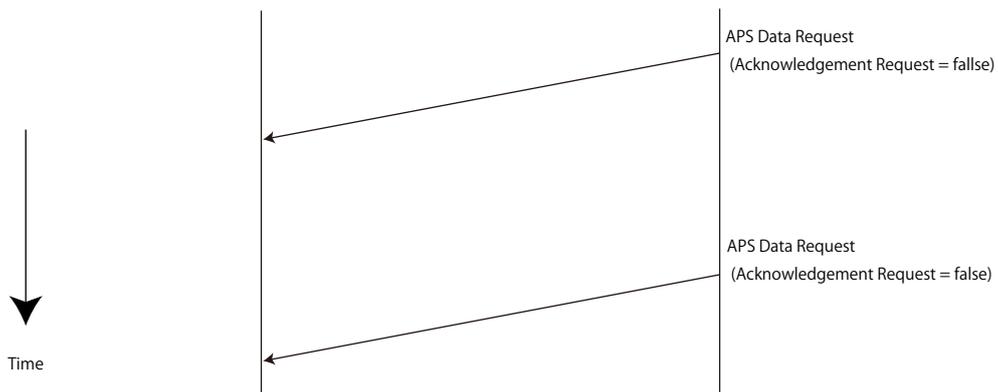


図 4.2: APS_ACK を用いない通信シーケンス

通信失敗と判断されるまでの時間 T は、通信のホップ数 $n_{wkcMaxDepth}$ を用いて、次式で求めることができる。

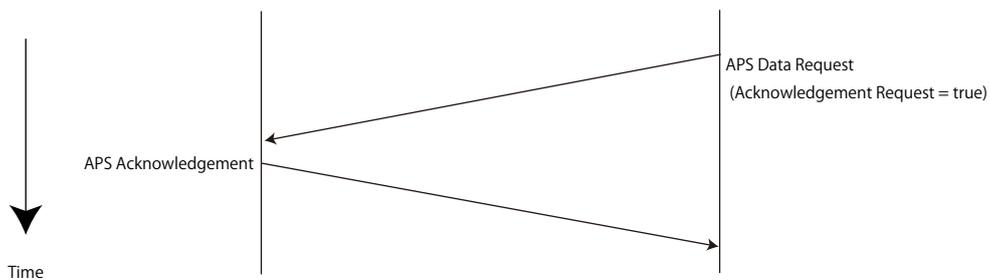


図 4.3: APS_ACK を用いた通信シーケンス

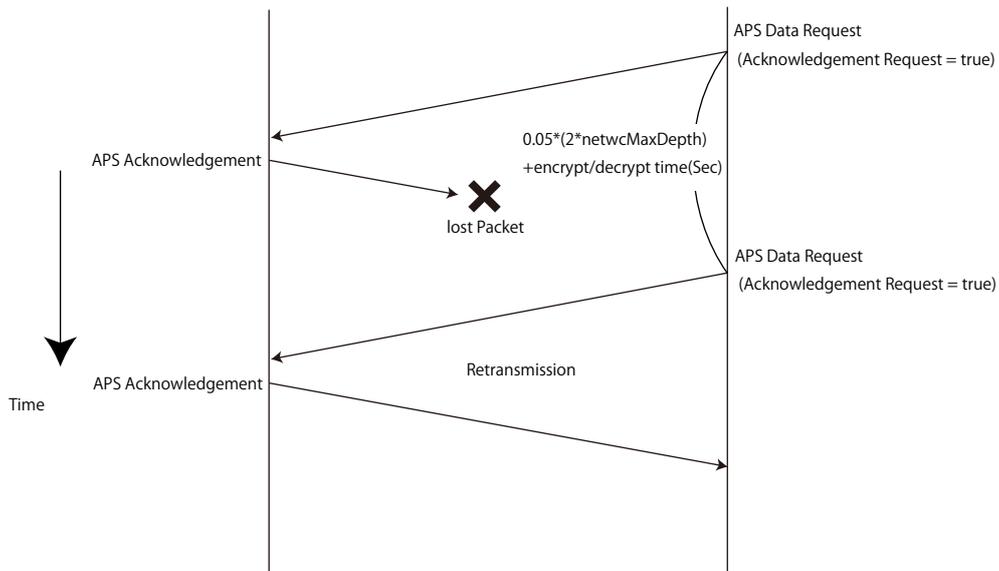


図 4.4: APS_ACK を用いた再送制御シーケンス

$$T = 0.05(2\text{netwcMaxDepth})$$

なお本研究では通信の暗号化を行っていないので、暗号化、復号化による遅延は考慮していない。

4.2.1 APS_ACK によるデメリット

APS_ACK 機能を利用するとパケット送信毎に通信相手からの Acknowledgement を指定時間待つため、通信スループットの低下が予想されるが、先述した CoAP によるパケット及び通信シーケンスの簡略化によりデータ転送の省力化を行いつつ、再送制御を行うことで通信品質のさらなる向上が可能ではないかと考える。

4.3 本章のまとめ

本章では提案手法である CoAP/ZigBee について述べ、その中で用いている技術である ZigBee APS_ACK について説明した。次章では提案手法の実装及び評価について述べる。

第5章 CoAP/ZigBeeの評価

本章では本研究の提案手法である CoAP/ZigBee の評価について述べる.

5.1 評価方法

APS_ACK を用いると通信成立に必要なパケット数が増加し,APS Acknowledgement の受信を待たなければならないため,通信スループットの低下が発生してしまうということは第4章で述べた.本章ではCoAPを用いて通信量の削減を行った環境下でZigBee APS_ACKを用いた通信を行った際の通信エラーレートについて調べ,提案手法であるCoAP/ZigBeeの評価を行う.

5.1.1 評価手法

第3章で行ったCoAPを用いた通信の実験と同様の構成を組み,直接通信環境で発生した通信データをZigBeeパケットスニッファを用いて収集する.XBeeの通信シーケンスは図5.2のようになり,End Device側からの通信はユニキャストを使用するためAPS_ACKを利用できるが,Coordinator側からの通信はブロードキャストとして通信されるためAPS_ACKを利用できない.この性質を用いてZigBeeそれぞれの通信パケットについてパケットスニッファで分析,統計しAPS_ACKを使用した通信と使用していない通信のスループット,通信エラー率をそれぞれ計測する.

5.1.2 構成

実験構成,ZigBeeのリンク構成,用いた機材とソフトウェア,XBeeの設定は表5.1のとおりである.

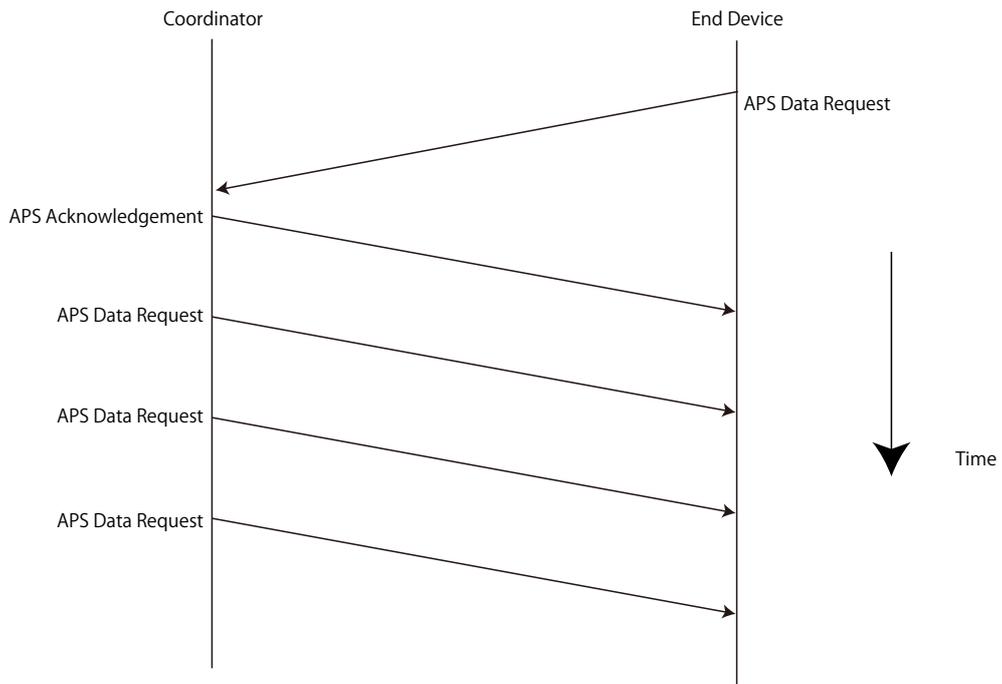


図 5.1: XBee 通信シーケンス

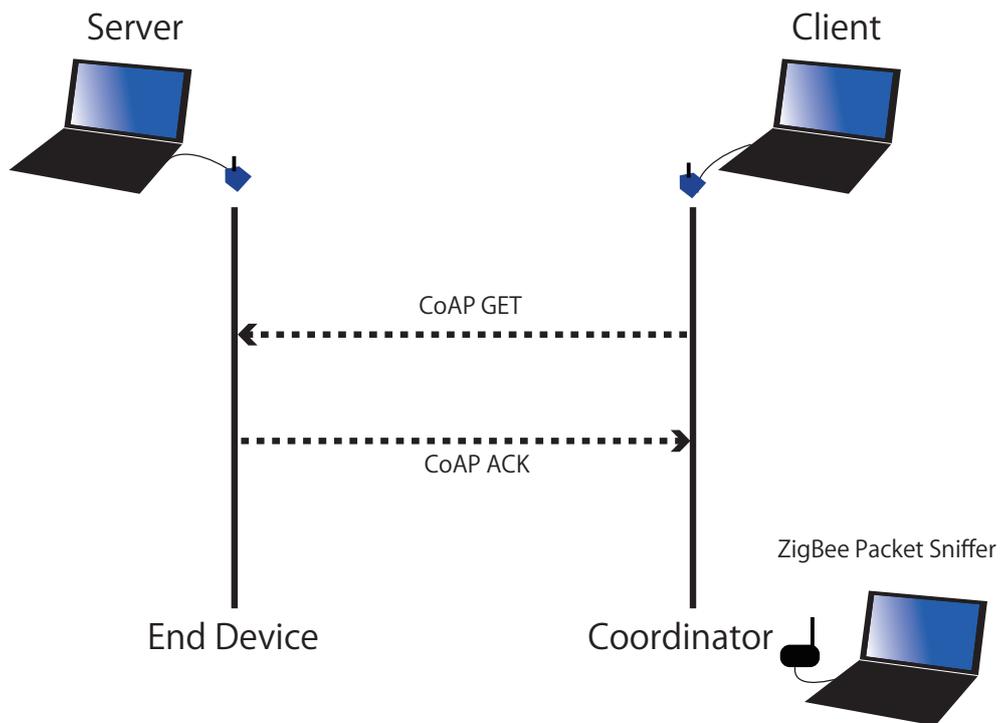


図 5.2: 実験構成図

5.1. 評価方法

	機材名	使用ソフトウェア,Version
Server	Fujitsu MG50S	ruby 1.8.6 patchlevel 111 [i386-cygwin]
Client	Lenovo ThinkPad T510	ruby 1.8.7 patchlevel 72 [i386-cygwin]
PacketSniffer	Daintree Networks 2400E	Daintree Sensor Network Analyzer 2.3.0.8

表 5.1: 実験に使用した機材とソフトウェア環境

	Server	Client
Firmware type	End Device	Coordinator
利用モジュール	Digi XBee Series2	
使用周波数	2445MHz(19Ch)	
使用 PANID	1222	
通信ボーレート	115200bps	

表 5.2: XBee モジュール設定詳細

5.1.3 結果

計測結果は以下のようになった. まず, 収集されたデータよりそれぞれのヒストグラムを算出し, 1 データを送信するまでに必要な時間を求める. ここで求めた時間を T_{ave} と置く.

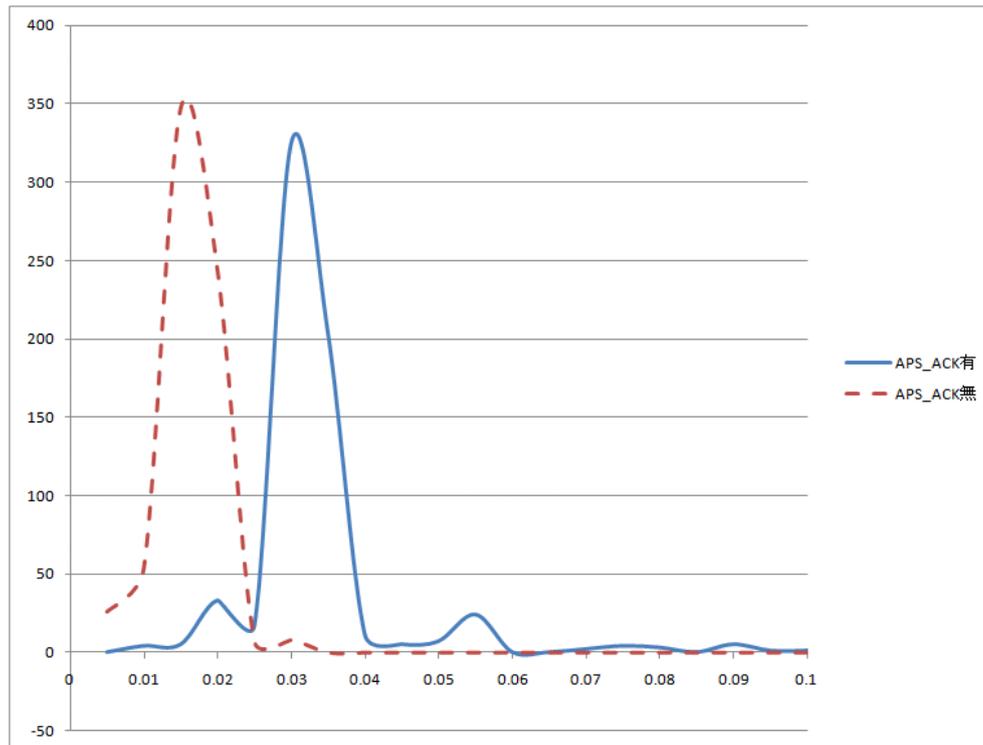


図 5.3: APS_ACK の有無による応答時間ヒストグラム

T_{ave} を用いての通信スループット S は第 3 章で求めた ZigBee の最大パケットサイズを用いて次式で求められる.

$$S = \frac{1024}{8(100(T_{ave}))} (\text{Kbps})$$

これを用いると図 5.4 のグラフを求めることが出来る. なお, 第 3 章で実験を行った XML と HTTP を用いた時の通信に関するスループットも, ZigBee パケットスニッファで取得していたデータを用いて計算し, グラフに加えた.

5.1. 評価方法

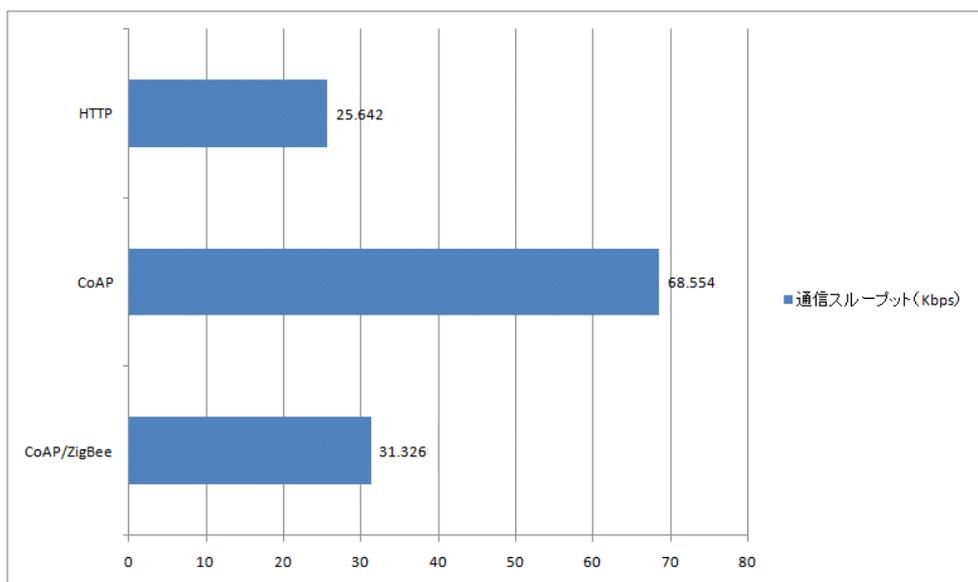


図 5.4: 通信スループット

またエラーレートに関しては図 5.5 のようになった。

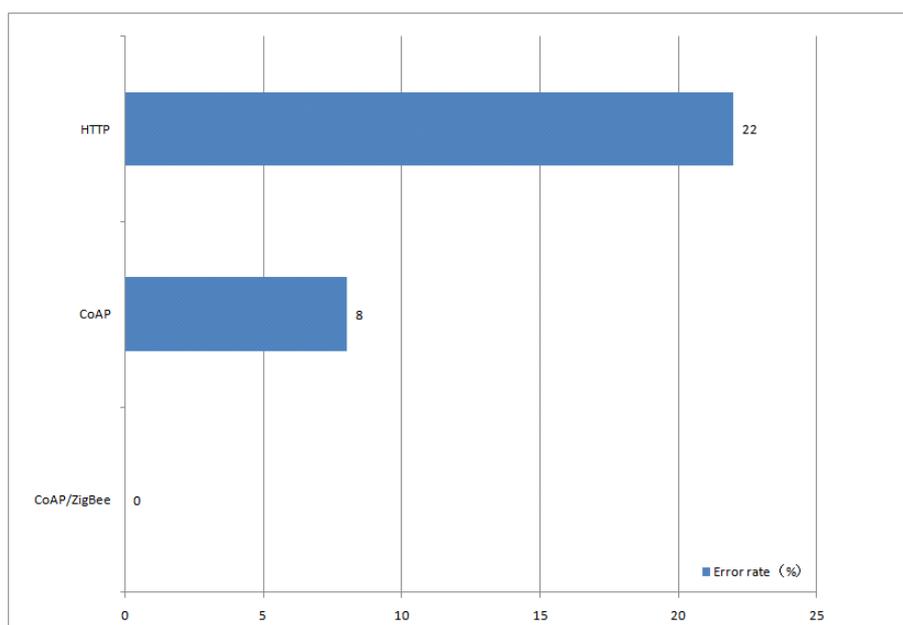


図 5.5: エラーレート

5.1.4 考察

上記グラフより, 提案手法である CoAP/ZigBee の通信スループットは CoAP 単体を用いるよりも約 50% 程度低下することが示されたが, XML と HTTP を用いる通信よりも速いスループットであることを示すことが出来た. また CoAP/ZigBee を用いることで通信エラーレートをさらに下げられるということも示すことが出来た. これは APS_Acknowledgement による再送制御を用いることで, パケットロスしたデータの再送を行うことが出来ている為だと考える.

5.2 本章のまとめ

本章では提案手法である CoAP/ZigBee の評価を行った. XML と HTTP を用いた通信, CoAP のメッセージプロトコルを用いた通信, 提案手法を比較し, 評価結果より, 直接通信間でのエラーレートを抑えることが出来るということを示した. 次章では本論文をまとめ, 今後の課題について述べる.

第6章 結論

本章では本研究のまとめと今後の課題について述べる.

6.1 本研究のまとめ

センサーより実空間情報を取得し,HTTP 等に基づく Web システムを用いて情報を共有するシステムが多く提案されているが,情報の取得に用いる ZigBee 等に代表される無線センサーネットワークは低消費電力,マルチホップ等の特性を持つ一方,遅延やパケット損失が IP ネットワークよりも発生しやすいという特性を持っている.このような無線センサーネットワーク環境上で Web システムをそのまま利用すると,HTTP のデータ量と TCP による再送制御により無線センサーネットワーク内で多くのパケット損失,再送が発生してしまう.

これを回避するために HTTP のデータ量を削減し,かつ UDP を用いて非同期に通信を可能にする CoAP(Constrained Application Protocol) が IETF にて提案,検討されている.そこで,CoAP をメッセージプロトコルとして使用し,HTTP 通信と同様の内容を送受信,比較した所 CoAP を用いた場合,通信の遅延を 30%から 40%程度まで削減することが出来た.これは通信成立までに必要なデータ量とシーケンスを減らすことが出来たためである.しかし,前述したように,無線センサーネットワークではパケット損失が多く発生してしまうため,実験環境では CoAP を用いても 8%の通信損失が発生してしまう.そこで本研究では,ZigBee の APS レイヤーにて再送制御が可能なことに着眼し,CoAP と ZigBee APS_ACK を組み合わせた CoAP/ZigBee を提案した.CoAP/ZigBee を用いると,APS_ACK 受信までの待機時間が発生するため,通信スループットは約 30Kbps まで低下するが,前述の HTTP 通信や CoAP 通信時に避けることが出来ない無線センサーネットワーク上でのパケット損失が避けられることを示した.

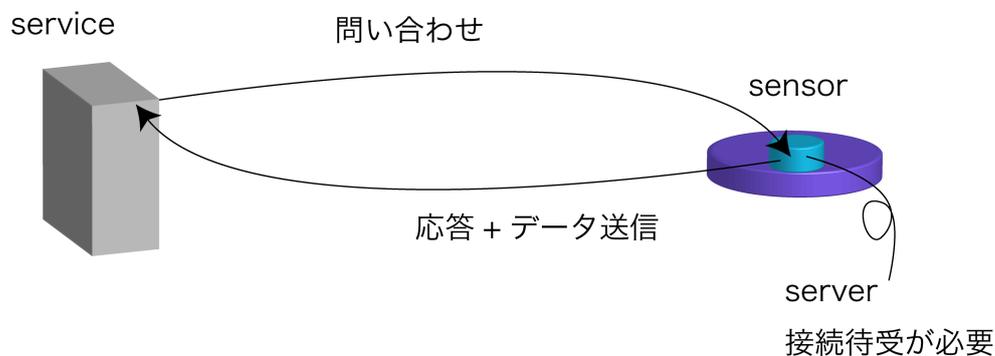
6.2 今後の課題と展望

本節では今後の課題と展望について述べる.

6.2.1 CoAP Gateway

本研究の提案手法で用いている CoAP は 6LowPAN を用いて IPv6 アドレスをセンサーノードに割り振り, 直接データ取得するような形を想定しているが, IPv6 に限定せず IP レイヤーを用いてセンサーに対して直接通信を行う場合, センサーノードに IP レイヤーを理解することができるサーバーを置くことになるため, センサーノードで各種情報のセンシングと同時に通信の待ち受け, 送信を行わなければならない, 通常より多くの電力やリソースを消費してしまう. これらの負荷を低減するために, センサーネットワーク内では ZigBee が本来持つアドレッシング機能を利用, CoAP をメッセージングプロトコルとして利用し, センサーデータを外部から参照する時センサーデータの集約とキャッシングを同時に行うようなゲートウェイがあれば, センサーネットワーク上で流通する通信量のさらなる削減を行うことが出来るのではないかと考えている.

CoAP で想定されるモデル



CoAP ゲートウェイ

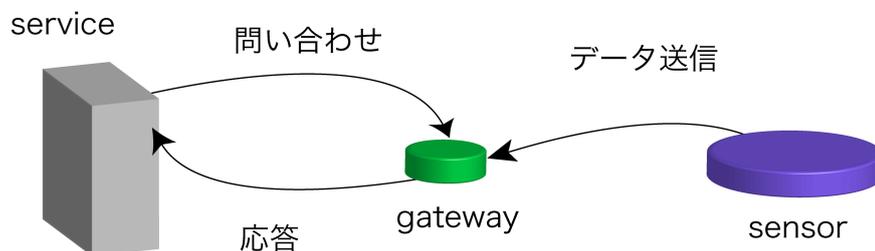


図 6.1: CoAP Gateway

参考文献

- [1] Twitter
<http://www.twitter.com> (2011/2/14 アクセス)
- [2] R. Fielding, J. Gettys, J. Mogul, H. Frystyk and T. Berners-Lee
RFC 2068 Hyper Text Transfer Protocol 1.1 (Jan, 1997)
- [3] patchube
<http://www.patchube.com> (2011/2/14 アクセス)
- [4] ZigBee Alliance. ZigBee Specification, r17 edition, 2008
- [5] J. Postel RFC 793 Transmission Control Protocol (Sep, 1981)
- [6] J. Postel RFC 791 Internet Protocol (Sep, 1981)
- [7] RACOW Project
<http://www.racow.net> (2011/2/14 アクセス)
- [8] IEEE Std 802.15.4-2003 (May, 2003)
- [9] Z. Shelby, B. Frank and D. Sturek
Constrained Application Protocol Internet Draft draft-ietf-core-coap (Oct, 2010)
- [10] J. Postel RFC 768 User Datagram Protocol (Aug, 1980)
- [11] XBee & XBee-PRO ZB ZigBee PRO RF Modules
<http://www.digi-intl.co.jp/digi/wireless/zigbee-mesh/xbee-zb-module.html>
(2011/2/14 アクセス)
- [12] W. Simpson RFC 1661 Point to Point Protocol (Jul, 1994)

謝辞

本論文の作成にあたり、御助言をいただきました慶應義塾大学環境情報学部教授村井純博士、同学部教授中村修博士、同学部准教授楠本博之博士、同学部専任講師重近範行博士、同学部専任講師 Rodney D. Van Meter III 博士、同学部准教授植原啓介博士、同学部教授武田圭史博士に感謝します。

また、常にご指導いただいた環境情報学部准教授 三次仁博士、政策メディア研究科特別専任講師 稲葉達也博士、羽田久一博士、同研究科特別研究助教 鈴木茂哉氏、中根雅文氏に感謝いたします。

学部1年で右も左もわからない中、SING 研究グループでご指導頂いた、政策メディア研究科博士課程 南正樹氏、北陸先端技術大学院大学 小原泰弘博士、株式会社 NTT コミュニケーションズ 尾崎隆明氏、株式会社野村総合研究所 奥村祐介氏、インターネットマルチフィード株式会社 金井 瑛氏 慶應義塾大学政策メディア研究科博士課程 松谷健史氏、空閑洋平氏、同研究科修士課程 波多野敏明氏、上原雄貴氏に感謝いたします。

所属グループが Auto-ID へ変更になり慣れない中で様々なご指導頂いた、パナソニックシステムネットワークス株式会社 徳増理氏、電気通信大学特任助教 川喜田佑介博士、株式会社 IBM 荻阪浩輔氏、株式会社インターネットイニシアティブ 佐藤泰介氏、佐藤龍氏に感謝いたします。

卒業論文執筆で苦楽を共にした、富田千智女史、宮崎圭太氏、広石達也氏、米村茂氏、齋藤俊氏に感謝いたします。

Auto-ID でお世話になりました株式会社日放電子 白石雅彦氏、ホロシェフ ロマン氏、株式会社ブロードバンドタワー 青木伸行氏、電機工業株式会社 松本伸史氏、KDDI 株式会社 神谷尚保氏、慶應義塾大学 SFC 研究所 金仙麗女史、慶應義塾大学政策メディア研究科修士課程、鈴木詩織女史、江村圭吾氏、山口修平氏、慶應義塾大学メディアデザイン研究科修士課程 杉本健一氏、環境情報学部、能島良和氏、佐藤友紀氏、横石 雄大氏、Doan Hoai Nam 氏、五十嵐祐貴氏、総合政策学部 小澤みゆき女史に感謝いたします。

卒論の進捗を見守っていただいた Twitter Friend TimeLine の皆様、友人の皆様、また 徳田・村井・楠本・中村・高汐・重近・バンミーター・植原・三次・中澤・武田合同研究プロジェクトの諸氏皆々様に感謝いたします。

そして、22年間自分をさまざまな方向から支えてくれた家族に感謝いたします。

以上を持って謝辞とさせていただきます。ありがとうございました。