

無線LANにおけるパソコンのリクエスト／レスポンス速度の向上

2年4組 尾崎 大晟 2年4組 上甲勇之介
2年4組 二宮滉太郎 2年4組 兵頭 周
2年4組 平岡 義将
指導者 二宮 正司・田中 善久

1 課題設定の理由

パソコン等の情報機器における通信は、情報の一塊であるパケットを送受信することで行っている。無線LANについても同様であるため、本校の無線LAN環境下で情報機器を利用する際、情報のリクエスト（要求）からレスポンス（応答）までの時間を短縮させることができるのではないかと考えた。そこで、複雑なプログラムを用いることなく、日常的に活用できるという条件で、回線利用を円滑化できると考えられる方法を選定したいと考えた。その方法の有効性について検証し、また、それによって生じるデメリットについても調べる。その結果を根拠にして、パソコンのリクエスト／レスポンス速度を向上させる方法を、本校生徒に周知させるという目的で本課題を設定した。

2 原理

(1) コマンドプロンプト

Windowsに標準搭載されている簡易命令実行プログラム。適当なコマンド（命令）を促すことでそれに対応したプログラムが起動する。

(2) pingコード（図1）

コマンドプロンプトで起動することのできるプログラムの一つ。適当なルーター、サイト、IPアドレスを入力すると、一度に4つのパケットを同一の目的地に送信し、それぞれのパケットの受信までに要した時間をms（ミリ秒）

でかえす。そこで、その4つのパケットをPacket1、Packet2、Packet3、Packet4と呼ぶこととする。

(3) チューニングレベルの変更（図2）

コマンドプロンプトで起動することのできるプログラムの一つ。受信データ量の上限を変更する機能がある。

```
C:\windows\system32>ping google.co.jp

google.co.jp [72.14.203.104]に ping を送信しています 32 バイトのデータ:
72.14.203.104 からの応答: バイト数 =32 時間 =46ms TTL=51
72.14.203.104 からの応答: バイト数 =32 時間 =42ms TTL=51
72.14.203.104 からの応答: バイト数 =32 時間 =43ms TTL=51
72.14.203.104 からの応答: バイト数 =32 時間 =42ms TTL=51

72.14.203.104 の ping 統計:
    パケット数: 送信 = 4, 受信 = 4, 損失 = 0 (0%の損失),
    ラウンドトリップの概算時間 (ミリ秒):
        最小 = 42ms, 最大 = 46ms, 平均 = 43ms
```

図1 コマンドプロンプト画面（pingコード）

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\student>netsh interface tcp set global autotuninglevel = experimental
```

図2 コマンドプロンプト画面（チューニングレベルの変更）

3 仮説

(1) コマンドプロンプトを用いてチューニングレベルを変更し、受信データ量の上限を引き上げることで大幅な通信速度の向上が望める。

(2) チューニングレベルを「experimental」に変更することでシステムは不安定になる。そのほかに電力の負担がPCにかかる。

4 実験方法

(1) 実験1の方法

ア 概要

PCのリクエスト/レスポンス速度を向上させる方法について検証するために、コマンドプロンプトのpingコードによるパケット送受信の時間計測を行う。

イ 使用機器

- ・同一機種タブレットPC

PC名：SSH001、SSH002、SSH003（ハードディスク内のデータ量はほぼ一定）

- ・アクセスポイント：ポケットWi-Fi端末（以下APと表記）

ウ 実験条件（図3）

- ① 床から45cmの高さにPCとAP方向に5.0m離す。実験中、PCとAPの間には障害となり得るものや中継機等は一切置かない。
- ② 各PCについて実験を7回ずつ行う。次の表1のとおり、2つの時間計測を行い、[試行1]での計測時間は、[試行2]の計測時間と比較するための基準値とする。

エ チューニングレベルの変更の手順

- ① コマンドプロンプトを開く。
- ② 「netsh interface tcp set global autotuninglevel = *****」を入力する。表2のようなチューニングレベルを「*****」の部分に入力でき、今回は「experimental」で試行した。

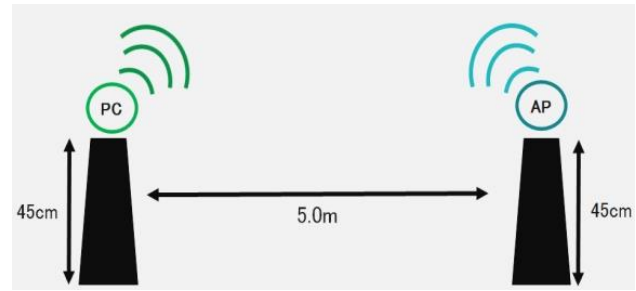


図3 実験1の方法（模式図）

表1 PCのリクエスト/レスポンス速度を向上させるための試行リスト

[試行1]	デフォルト状態 ^{※1} での時間計測
[試行2]	チューニングレベル ^{※2} を変更した状態での時間計測

※1 PCやAPで何の設定変更も実行されていない状態。標準設定の状態。

※2 CPのnetshコードでチューニングレベル（一度に受信できるパケットデータ量の上限の設定）を変更した状態。今回はそのレベルを高くして試行する。

表2 チューニングレベルリスト

disabled	normal	restricted	experimental
無効	規定値	拡大	極端に拡大

(2) 実験2の方法

ア 概要

実験1においてチューニングレベルを変更し、受信データ量を増やすことは、電力を大幅に消費してしまうと考え、そこで電力チェッカーによる消費電力量を計測する。

イ 使用機器

- ・同一機種タブレットPC

PC名：SSH001、SSH002、SSH003（ハードディスク内のデータ量はほぼ一定）

- ・PCの充電アダプタ
- ・電力チェッカー

ウ 実験条件

- ① チューニングレベルを「experimental」に設定したPCを用いてインターネットブラウザを開く。
- ② 完全にサイトが開き終わったタイミングの電力消費量を計測する。

5 実験結果

実験1の結果を、表3～表5に示したのち、表6にまとめた。実験2の結果を表7にまとめた。

表3 PC001

デフォルト値 PC001 (ms)							
	1回目	2回目	3回目	4回目	5回目	6回目	7回目
Packet1	67	55	76	68	101	64	60
Packet2	72	73	62	69	84	93	107
Packet3	63	77	66	74	71	84	90
Packet4	72	72	70	74	65	72	70
Max	72	77	76	74	101	93	107
Min	63	55	62	68	65	64	60
Ave	67	69	68	71	80	78	81
チューニングレベル変更値 PC001 (ms)							
	1回目	2回目	3回目	4回目	5回目	6回目	7回目
Packet1	26	26	27	26	27	28	27
Packet2	29	25	27	26	26	26	27
Packet3	27	23	30	25	27	26	26
Packet4	27	24	26	30	26	27	26
Max	29	26	30	30	27	28	27
Min	26	23	26	25	26	26	26
Ave	27	24	27	26	26	26	26

表4 PC002

デフォルト値 PC002 (ms)							
	1回目	2回目	3回目	4回目	5回目	6回目	7回目
Packet1	51	141	73	64	71	76	75
Packet2	437	79	63	148	50	50	91
Packet3	218	98	128	72	60	69	108
Packet4	57	93	62	78	66	53	86
Max	437	141	128	148	71	76	108
Min	51	79	62	64	50	50	75
Ave	189	102	81	93	61	62	90
チューニングレベル変更値 PC002 (ms)							
	1回目	2回目	3回目	4回目	5回目	6回目	7回目
Packet1	229	324	33	24	21	20	20
Packet2	605	22	34	22	21	20	20
Packet3	損失	21	29	21	21	21	20
Packet4	618	20	損失	21	25	22	24
Max	618	324	34	24	25	22	24
Min	229	20	29	21	21	20	20
Ave	484	96	32	22	22	21	21

表5 PC003

デフォルト値 PC003(ms)							
	1回目	2回目	3回目	4回目	5回目	6回目	7回目
Packet1	61	67	57	113	57	136	57
Packet2	51	61	53	65	127	74	126
Packet3	56	68	65	97	102	73	95
Packet4	72	48	87	79	損失	67	82
Max	72	68	87	113	127	136	126
Min	51	48	53	65	57	73	57
Ave	60	61	65	88	95	87	89
チューニングレベル変更値 PC003(ms)							
	1回目	2回目	3回目	4回目	5回目	6回目	7回目
Packet1	22	91	24	154	26	31	29
Packet2	26	損失	54	26	26	27	26
Packet3	25	62	43	26	26	41	31
Packet4	25	76	22	26	26	26	27
Max	26	91	54	154	26	41	31
Min	22	62	22	26	26	26	26
Ave	24	76	35	58	26	31	28

表6 各PCのデフォルト値、チューニングレベル変更値における応答時間（平均値）

PC名	デフォルト値	→	チューニングレベル変更値
PC001	77.9ms	→	36.7ms
PC002	96.9ms	→	99.6ms
PC003	73.4ms	→	36.4ms

表7 各PCのデフォルト値、チューニングレベル変更値における消費電力

PC名	デフォルト値	→	チューニングレベル変更値
PC001	47.2W	→	46.8W
PC002	46.1W	→	46.3W
PC003	48.5W	→	46.3W

6 考察

(1) 実験1の考察

ア チューニングレベルを「experimental」に変更した場合、表6より、PC001とPC003は大幅にリクエスト/レスポンス速度が速くなった。

イ PC002は、1回目で大幅にリクエスト/レスポンス速度が落ちてしまったため、最後に平均値を出しても意味がなくなった。チューニングレベルを変更した場合、システムが不安定になる場合が不定期にある。

(2) 実験2の考察

コマンドプロンプトでチューニングレベルを「experimental」に変更したが、表7より、ほとんど消費電力は変化しなかった。チューニングレベルの変更は消費電力には影響を与えない。

7 まとめと今後の課題

(1) 各実験のまとめ

ア 実験1

コマンドプロンプトのチューニングレベルを「experimental」に設定し、PC001とPC003では大幅にリクエスト/レスポンス速度を向上させることができた。しかし、PC002においては、1回目の時間計測で484msと出てしまった。それは、チューニングレベルを変更したことによって、システムが不安定になってしまった結果であると考えられる。不定期に起こる損失もシステムが不安定になることによるものだと考える。

一般的に、チューニングレベル「experimental」は、規定値よりも大幅に受信データ量を増やしてしまうと言われている。よって、システムが不安定になり、頻繁に回線が落ちるが、今回の実験結果のように大幅に通信速度を向上させることが可能とも言える。安定して回線を利用するためには、チューニングレベルを「restricted」に設定することが求められる。

イ 実験2

チューニングレベルの変更は、消費電力には影響を与えない。

(2) 今後の課題

ア チューニングレベル「experimental」よりも「restricted」のほうが安定した通信ができるが、リクエスト/レスポンス速度は遅くなる。そこで、「restricted」での正確なリクエスト/レスポンス通信速度を、今回の実験と同様の方法で計測する。

イ 実験2では、チューニングレベルを「experimental」に変更すると、電力を多く消費するというデメリットがあると思って調べてみた。チューニングレベルを変更しても、CPUやメモリの処理に変化はないのか、アクセスポイント側に与える影響はないのか調べてみたい。

参考文献

- ・「Manage Engine」 <http://www.atmarkit.co.jp/ait/articles/0012/01/news002.html>
- ・「Windowsの便利な使い方」 <http://www.k-tanaka.net/cmd/ping.php>
- ・「Tech net」 [https://technet.microsoft.com/ja-jp/library/cc732509\(v=ws.10\).aspx](https://technet.microsoft.com/ja-jp/library/cc732509(v=ws.10).aspx)