

“Wi-Fiを利用した行動解析”のためのシステム開発

大阪産業大学 デザイン工学部 情報システム学科
情報教育システム研究室

15H021 角谷卓哉

“Wi-Fi を利用した行動解析”のためのシステム開発

15H021 角谷卓哉

1 はじめに

大阪産業大学(以下、本学という)は敷地が東西に長い長方形である。そのため講義の開講場所によっては敷地の端から端まで移動するため、次の授業の開講に間に合わないことがある。なお本学はその西側の敷地を中央キャンパス、東側の敷地を東キャンパスとそれぞれ呼称する。

我々は講義開講場所間の移動距離が長いことによって、学生の講義受講に支障が発生している可能性を考えた。

2 目的

本研究の目的は、本学内における学生の移動を把握し移動時間を算出することで、学生の受講環境を明らかにするシステムを開発することである。学生の移動を把握する方法は、スマートフォンなど Wi-Fi 通信機能を搭載した機器(以下、モバイルデバイスという)が、アクセスポイント^{*1}と接続する際の通信を計測する。計測した通信から人の移動を把握する手法を利用する [1]。

3 行動解析手法について

モバイルデバイスはアクセスポイントと接続する際に Probe request(以下、PR という)を発信する。PR には発信元の MAC アドレスが含まれている。この MAC アドレスで個人を判別する。本研究では Raspberry Pi と Wireshark で、PR を計測するための機器(以下、計測機器という)を作成し、本学内の 11 箇所に設置した。計測期間は約 1 ヶ月である。計測した PR で学生の移動を把握し移動時間を算出することで、学生の受講環境を明らかにする。

4 結果と考察

計測機器を本学内に設置することで多数の MAC アドレスを計測することができた。本学の学生及び教職員(大阪桐蔭高校職員並びに本学の非常勤講師含む)の合計は約 1 万人である。約 1 ヶ月間設置した場所の中で、最も多く計測できた MAC アドレスの数は約 65 万個である。1 日に計測できた MAC アドレスは約 2 万個である。1 人が 2,3 個のモバイルデバイスを持つことはあるため、計測できた MAC アドレスの数は妥当であると考ええる。

5 まとめ

本研究の目的は、学生の移動を把握し移動時間を算出することで、学生の受講環境を明らかにするシステムを開発することである。学生の移動を把握することは達成できたが、計測した PR の管理面に課題がある。計測機器が万が一盗難にあった場合、計測した PR を悪用されるおそれがある。また本学内には様々な人が行き交うため、計測した MAC アドレスが全て学生であるとは言えない。今後の課題として、計測機器の盗難対策、学生とそれ以外の人と分ける方法、と考える。

参考文献

- [1] 望月祐洋, 上善恒雄, 西田純二, 中野秀男, 西尾信彦ほか. Wi-fi パケットセンサを利用した匿名人流解析システムの構築. 研究報告ユビキタスコンピューティングシステム (UBI), Vol. 2014, No. 45, pp. 1-8, 2014.

^{*1} アクセスポイントとは、スマートフォンなどをネットワークに接続するための機器である。

目次

1	はじめに	1
2	目的	2
3	行動解析手法について	3
3.1	設置場所	3
4	データ収集システム	6
4.1	Probe Request について	6
4.2	計測機器	9
4.3	設置環境	11
4.4	システム構成	12
4.5	匿名化	18
5	結果と考察	19
5.1	計測結果	19
5.2	考察	21
5.3	プログラムの評価	21
6	結論	23
6.1	今後の課題	23
付録 A	ソースコード	26
A.1	ソースコード.1	26
A.2	ソースコード.2	26
A.3	ソースコード.3	27
A.4	ソースコード.4	27
A.5	ソースコード.5	28

1 はじめに

大阪産業大学(以下、本学という)はその敷地を東高野街道で区切られている。本学はその西側の敷地を中央キャンパス、中央キャンパスの南側の敷地を南キャンパス、東側の敷地を東キャンパスとそれぞれ呼称している。本学は敷地が東西に長い長方形である。講義の開講場所によっては敷地の端から端まで移動する。そのため移動に時間がかかり、次の講義の開講に間に合わないことがある。

我々は講義間の移動距離が長いことによって、学生の講義受講に支障をきたしている可能性を考えた。そこで本研究では学生の移動を把握する方法から、本学内における学生の移動を把握し、受講環境の実態調査を行うためのシステムを開発した。

第2章では本研究の目的について述べる。第3章では本研究の行動解析手法について述べる。第4章ではデータ収集システムについて述べる。第5章では計測の結果をその考察とともに述べる。第6章では研究の成果とともに今後の課題についてまとめる。

2 目的

本研究の目的は、本学内における学生の移動を把握し移動時間を算出することで、学生の受講環境の実態を明らかにするシステムを開発することである。学生の移動を把握する方法は、スマートフォンなど Wi-Fi 通信機能を搭載した機器 (以下、モバイルデバイスという) が、アクセスポイント^{*1}(以下、AP という) と接続する際の通信を計測する。計測した通信から学生の移動を把握する手法を利用する。

^{*1} アクセスポイントとは、スマートフォンなどをネットワークに接続するための機器である。

3 行動解析手法について

本研究では本学内における学生の移動を把握するために、モバイルデバイスが AP と接続する際に (または接続している状態でも) 発する Probe Request^{*2}(以下、PR という) を計測する。計測した PR から学生の移動を把握する手法 (以下、PR 人流把握手法という) を用いる。PR には発信したモバイルデバイスの MAC アドレス^{*3}が含まれる。1 人につきモバイルデバイスを 1 つ所持すると仮定した場合、MAC アドレス=モバイルデバイス=個人が成り立つ。先に述べた MAC アドレス=個人の考えから、PR を計測したときのタイムスタンプ^{*4}を辿ることで学生の移動を把握する。

本研究では、PR を計測するための機器 (以下、計測機器という) を作成し、本学内の 11 箇所で人通りが多いと予想した場所に設置した。また計測期間は 4 週間^{*5}である。設置した計測機器で本学内における学生の移動を把握し、移動時間を算出することで受講環境の実態を明らかにする。

3.1 設置場所

計測機器の設置場所と名前を表 1 と図 1 にそれぞれ示す。

表 1 計測機器設置場所の名前の一覧

設置場所の名前	
中央キャンパス	5 号館
	7 号館
	9 号館
	16 号館
	図書館
	本館
東キャンパス	3 号館
	4 号館
	8 号館
	15 号館
	クリスタルテラス

*2 Probe Request については 4.1 で解説する。

*3 MAC アドレスとはネットワーク機器に一意に割り当てられている 12 桁の 16 進数で表されたアドレスである。

*4 時刻情報のことである。

*5 2018 年 11 月 21 日 (水)-2018 年 12 月 18 日 (火)

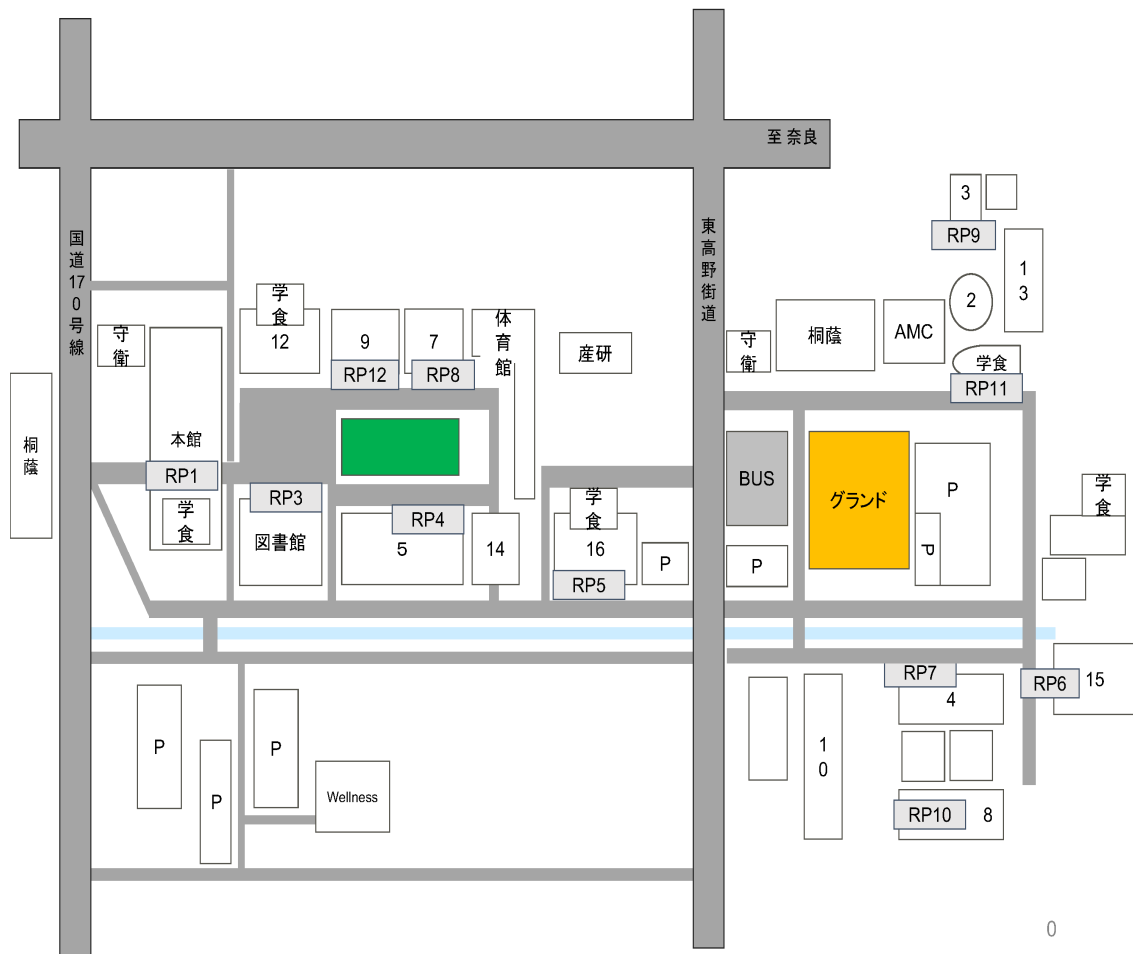


図1 計測機器を設置した場所を示した本学の簡易マップである。図中のRPと数字は計測機器の設置場所を示す。設置した計測機器は全部で11機である。図中の四角に割り振られている番号は建物の名前である。一部の建物の名前は省略している。一例で5が割り振られている建物は5号館である。図中右上にある学食はクリスタルテラスという名前である。Pはパーキング(駐輪場)である。Wellnessは本学の総合フィットネス健康施設である。図中の東高野街道から左側が中央キャンパスである。中央キャンパスの下側に3つのPとWellnessがある。この一帯が南キャンパスである。東高野街道から右側が東キャンパスである。

基本的に設置場所は学生の手が届きづらく、また設置場所で業務を行う人の監視の目が届きやすい学科事務室に設置した。学科事務室以外では、図書館内の受付カウンターの内側などに設置した。計測機器を設置したときの写真を図2と図3にそれぞれ示す。

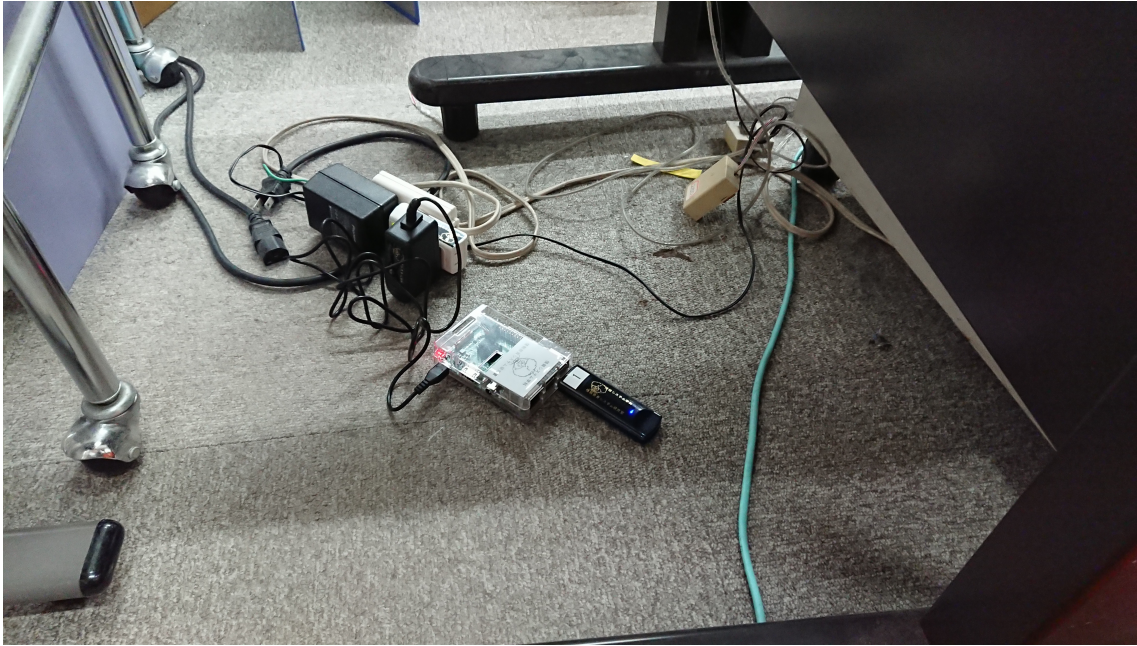


図 2 計測機器設置場所の 1 つである本館での設置時の写真である。図中に写っているプラスチックケースは Raspberry Pi^{*6}である。その Raspberry Pi に USB 接続されている機器は無線 LAN アダプタである。図中のテーブルタップには、Raspberry Pi の AC アダプタが接続されている。AC アダプタを含め、上述した機器を纏めて計測機器とする。計測機器設置場所のコンセントの差し込み口を埋めないために、電源タップを持ち込み使用している。



図 3 計測機器設置場所の 1 つである 7 号館での設置時の写真である。図中に写っているプラスチックケースは Raspberry Pi である。その Raspberry Pi に USB 接続されている機器は無線 LAN アダプタである。図中左側の電源タップに接続されているのが、Raspberry Pi の AC アダプタである。AC アダプタを含め、上述した機器を纏めて計測機器とする。計測機器設置場所のコンセントの差し込み口を埋めないために、電源タップを持ち込み使用している。

*6 詳細は 4.2 に記述する。

4 データ収集システム

ここでは本研究で計測する PR を解説したのちに、計測機器の仕様とシステムについて記述する。

4.1 Probe Request について

PR はモバイルデバイスが AP に接続するときに発信する。モバイルデバイスと AP が接続する際に、2 つのパターンがある。1 つはパッシブスキャン。もう 1 つはアクティブスキャンである。PR はアクティブスキャンを行う際に、モバイルデバイスから発信される。

4.1.1 パッシブスキャン

パッシブスキャンは AP がブロードキャスト*7するビーコンフレーム*8をモバイルデバイスが受信し、モバイルデバイスが把握する AP の ESSID*9情報と合致した場合のみ、接続のためのやりとりを開始する。パッシブスキャンを行う際のモバイルデバイスと AP のやりとりの簡易図を図 4 に示す。

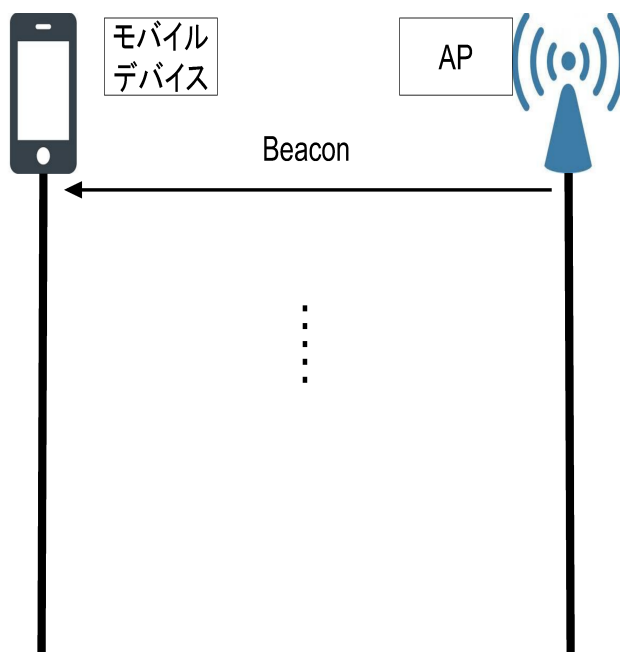


図 4 パッシブスキャン時の AP とモバイルデバイスのやりとりの簡易図である。図中左側のアイコンはモバイルデバイス、右側のアイコンは AP である。AP は一定時間ごとにビーコンフレームにより自身の ESSID 情報をブロードキャストする。ビーコンフレームを受け取ったモバイルデバイスは ESSID が合致する AP を検索する。合致していれば接続を開始する。この接続するまでの一連の流れがパッシブスキャンである。

*7 不特定多数の端末に対して同じ情報を送ることである。

*8 AP から 100 ミリ秒ごとに発信される。これには利用できる (占有されていない) チャンネルの情報、AP の ESSID の情報が含まれる。

*9 Extended Service Set Identifier の略で、無線 LAN のネットワークで使われる識別子のこと。アクセスポイントの識別子である SSID を、複数のアクセスポイントを設置したネットワークでも使えるように拡張したものである。

4.1.2 アクティブスキャン

アクティブスキャンはモバイルデバイスがビーコンフレームを一定時間受信できなかったときに、モバイルデバイスからデバイス自身の把握する AP の ESSID 情報を PR で発信する。また自身の把握する AP の ESSID だけでなく、ブロードキャストアドレスで発信する場合がある。それを受け取った AP は AP 自身の ESSID 情報と合致した場合のみ、接続のためのやりとりを開始する。アクティブスキャンを行う際のモバイルデバイスと AP のやりとりの簡易図を図 5 に示す。

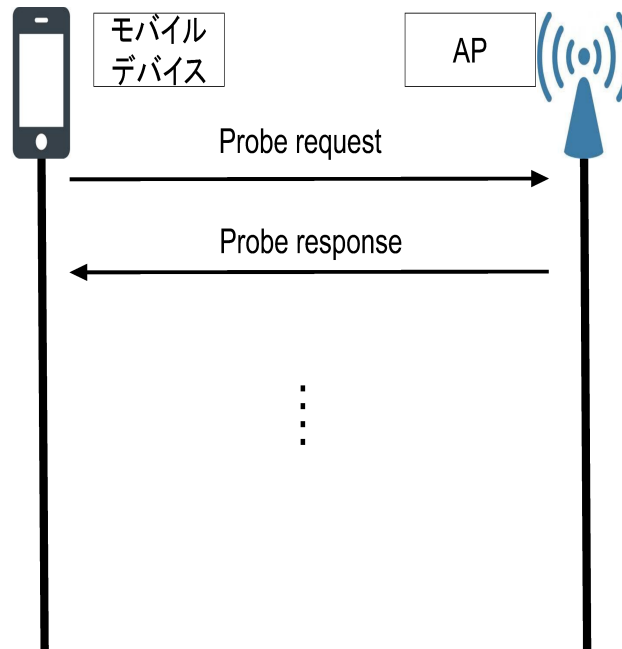


図 5 アクティブスキャン時の AP とモバイルデバイスのやりとりの簡易図である。図中左側のアイコンはモバイルデバイス、右側のアイコンは AP である。モバイルデバイスは一定時間ビーコンフレームを受信できなかった場合、自身の把握する AP の ESSID 情報 (ブロードキャストアドレスで発信する場合もある。) をブロードキャストする。これが PR である。次に AP はモバイルデバイスから発信された PR を受信後、PR に含まれる ESSID が AP 自身の ESSID と同じ場合、PR に応答する。これが Probe response である。モバイルデバイスと AP が互いに Probe でやりとりをした後に接続を開始する。この接続するまでの一連の流れがアクティブスキャンである。

4.1.3 Probe Request の詳細

無線 LAN のフレームフォーマットにおいて、PHY^{*10}フレームのデータの中の MAC フレームのヘッダ内にフレームコントロールフィールドがある。フレームコントロールフィールドには 2byte のデータが格納される。その 2byte 内の各 bit の組合せで MAC フレームの種類が変わる。このフレームコントロールフィールドには 3 種類のフレームがある。マネージメント (管理) フレーム、コントロール (制御) フレーム、データフレームである。その中の 1 つであるマネージメントフレームの中に、PR^{*11}が存在する。フレームコントロール内の Frame Type の値が [00]^{*12}かつ、Frame Subtype の値が [0100]^{*13}であれば、MAC フレームはマネージメントフレームの PR になる。無線 LAN のフレームフォーマットの簡易図を図 6 に示す。

*10 Physical の略である。

*11 他にも Beacon, Probe response, Association request が存在する。

*12 00: マネージメント、01: コントロール、10: データの値が入る。

*13 このフィールドには 4bit の値が入る。

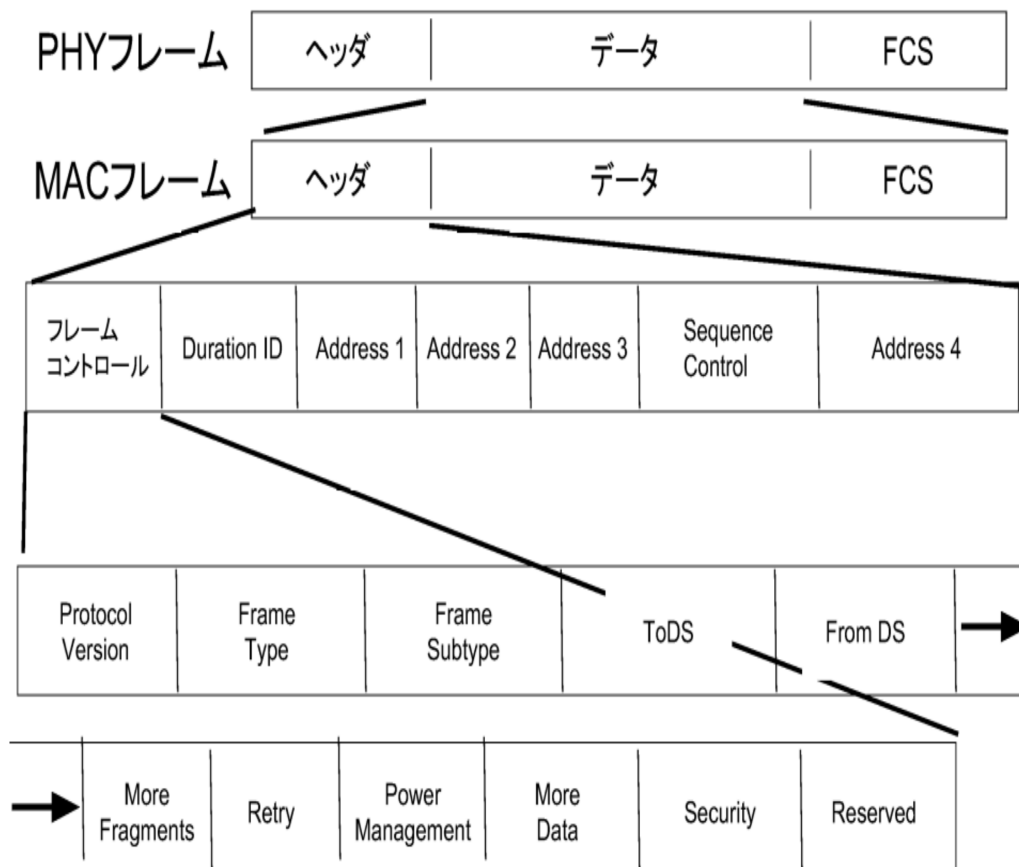


図 6 MAC フレームのヘッダ内部における各フィールドの名称及びその内訳を表している。MAC フレームのヘッダ内にあるフレームコントロールフィールド。このフィールド内の Frame Type が [00] かつ Frame Subtype が [0100] であれば、PR になる。

4.1.4 Probe Request の発信間隔

モバイルデバイスから発信される PR の発信間隔は Android,iOS,Windows といった OS でそれぞれ異なる。そのため一概にモバイルデバイスからの PR の発信間隔は記述できない [1]。参考に Xperia XZs からの PR の発信間隔を表 2 に示す。

表 2 Xperia XZs からの Probe Request の発信間隔表

モバイルデバイス名 (OS)	モバイルデバイスの表示画面	Wi-Fi 接続時	Wi-Fi 未接続時
Xperia XZs(Android 8.0)	ホーム	40 秒	5 秒-60 秒
	Wi-Fi 設定	10 秒	

4.2 計測機器

計測機器の仕様を表 3 に示す。

表 3 計測機器仕様一覧

コマンドライン	tshark ver 2.2.6
ハードウェア	Raspberry Pi 3 Model B
無線 LAN アダプタ	WLI-UC-AG300N
OS	Ubuntu MATE 16.04.4 LTS(Xenial Xerus)
SD カード容量	64GByte
その他付属パーツ	RTC

以降、Raspberry Pi のことを RP、Ubuntu MATE のことを MATE とそれぞれ略す。実際に計測機器を設置した際の写真を図 7 に示す。



図 7 計測機器の写真である。図中に写っているプラスチックケースが RP である。その計測機器に USB 接続しているのは無線 LAN アダプタである。電源からの距離と AC アダプターのコードの長さの関係上、設置場所から落ちる可能性があるため、養生テープで固定している。

4.2.1 Probe Request から得られる人に関する情報

計測機器で PR を計測した結果、pcap^{*14}データが得られる。以下に計測できた PR で人^{*15}の位置に関わる項目を示す。また本研究では RSSI^{*16}の情報を使用しない。

- MAC アドレス
- RSSI

RSSI が人の位置に関わる理由を図 8 に示す。

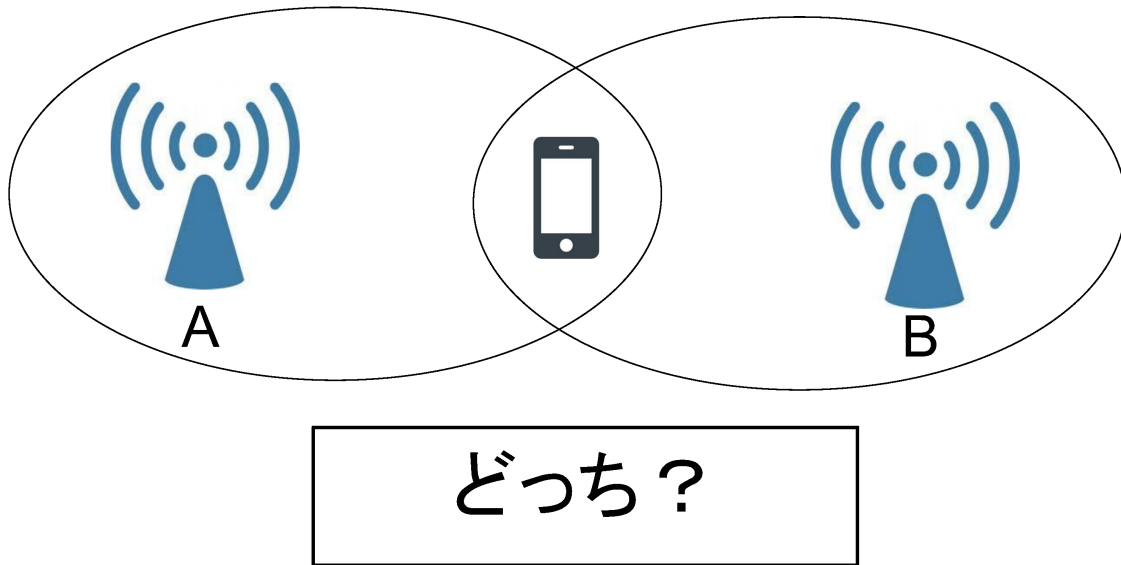


図 8 RSSI が人の位置に関わる理由を簡潔に示した図である。仮に計測地点 A と B があり、その両方で PR が計測できてしまった場合、その人はどちらにいたのか分からない。この状況だと B 地点の RSSI が強かった場合、その人は B 地点の近くにいたといえる。このような状況でも、ある程度はどちらの地点の近くにいるかどうかは RSSI で判別できる。以上が、RSSI が人の位置に関わる理由である。

RSSI の情報を使用しない理由については以下の通りである。

- 設置場所が離れており、図 8 の状況になりづらい。そのため RSSI を使う必要性がない。
- RSSI の情報はあくまで目安であり、信頼性が高くない。

4.2.2 計測機器の Probe Request の検知範囲

以下の環境で計測テストを行ない、計測機器の PR の検知範囲を調査した。

- 廊下で障害物なし
水平方向に約 38m で計測できた。
- 室内で針金入りガラス扉 2 枚
水平方向に屋外に設置したモバイルデバイスを約 18m で計測できた。

^{*14} pcap とはネットワーク上に流れるパケットを、キャプチャするためのファイル形式である。

^{*15} 本研究における人とは学生である。

^{*16} 信号強度。モバイルデバイスは接続できる AP が 2 箇所同時に存在する場合、各 AP から発信されるビーコンの RSSI の強度がより強い AP に接続しようとする。

4.3 設置環境

ここでは計測機器を設置するにあたって、本研究で重要視した項目について記述する。

4.3.1 空調の有無

本研究では計測機器を設置するにあたり空調設備が備わっている学科事務室などの室内に設置した。空調の有無を重要視した理由は RP の CPU の過熱によるフリーズである [2]。

4.3.2 電源の有無

本研究では長期的な計測を考え、計測機器は電源が十分にある学科事務室などに設置した。先行研究では計測機器を複数種類用意し、設置場所の環境に合わせて計測機器のハードウェアや OS を変えている [3]。なお先行研究では計測機器のことを AMP センサ^{*17}と記載している [3]。その AMP センサの基本構成は以下である。

- Wi-Fi アダプタ
- コンピュータ
- バッテリ

以上から AMP センサ設置の際にバッテリーを使用していたと考えられる。また用意した AMP センサのハードウェアの中に RP の名前がある。AMP センサの基本構成の画像から、バッテリーは RP のために使用していたと考えられる。RP の消費電力は少ないため可能ではあるが、本研究では長期間 (4 週間) の計測を想定しているため、実際の計測に耐えられるバッテリーは高価であると思われる。それを本研究で計測機器の個数分用意し、計測機器と共に設置すると盗難の可能性が高いと考えた。

^{*17} AMP センサとは、Anonymous MacAddress Probe センサの略である。

4.4 システム構成

計測機器のシステム構成を図 9 に示す。

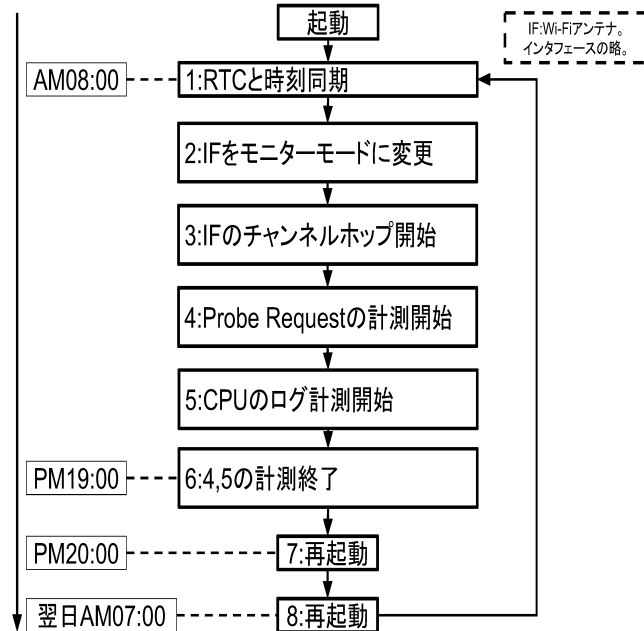


図 9 計測機器のフロー図である。この図における番号 1 は 4.4.2、番号 2,3 は 4.4.3、番号 4 は 4.4.4、番号 5 は 4.4.5、番号 7,8 は 4.4.6、番号 2-5 のプログラムの起動を担当するのは 4.4.1 に記述する。

4.4.1 プログラムの定間隔実行

本研究では `crontab` コマンドを使用し後述するプログラムを定刻及び定間隔^{*18}で実行をさせている。

この `crontab` コマンドは各ユーザーの設定ファイルに記述された時間とコマンドを実行するコマンドである。設定ファイルに記述する方法は `(sudo) crontab -e` を実行する。この `crontab` コマンドは設定ファイルに記述されたコマンドを上から順に実行する。PR は無線 LAN アダプタをモニターモード^{*19}にしなければ計測できないため、必ず一番最初に `monitor.sh`^{*20}を起動するように記述しなければならない。 `sudo crontab -e` で root ユーザーの設定ファイルを編集できる。root ユーザーの設定ファイルで記述されたコマンドは root 権限で実行される。本研究では root ユーザーの設定ファイルでプログラムを実行する。また時間を指定する箇所に `@reboot` を記述することで、指定したコマンドを起動時に一度だけ実行させる。定刻実行させる理由は以下の通りである。

- 設置場所の室温に依るが、RP を 24 時間計測させるとフリーズする可能性がある
- 大凡ではあるが AM08:00JST^{*21}以降でなければ本学内に学生がいない
- 大凡ではあるが PM19:00JST 以降は本学内に学生がいない

定間隔実行させる理由は、万が一 `@reboot` による起動時実行が失敗したときのためである。また本学内の各場所に設置した計測機器の動作条件を統一するために、AM07:00 と PM20:00 に計測機器を再起動させる。 `crontab` のソースコードは A.5 に記載している。

*18 定刻実行させる時刻は AM08:00 である。定間隔実行は 1 分毎である。

*19 モニターモードについては 4.4.8 で記述する。

*20 `monitor.sh` の詳細は 4.4.3 に記述する。

*21 Japan Standard Time の略である。

4.4.2 計測機器の時刻情報

本研究の計測機器はネットワークに接続せずに計測する。RTC^{*22}はハードウェアクロックである。RP には RTC が搭載されていない。そのため RP は電源が OFF の間は時刻情報が変化しない。各設置場所の計測機器の時刻に誤差がある場合、正確な学生の移動を把握できない。これを解決するために RP 専用の RTC を取り付けた。

この RTC を RP に実装させるためには、以下の作業を順に実行する必要がある。

1. RP にインストールした MATE の `/pi.boot/config.txt` に以下を記述する。記述する場所は最下行にする。
`dtparam=i2c_arm=on`
`dtoverlay=i2c-rtc,ds3231`
2. RP にインストールした MATE の `/etc/rc.local` の最下行に以下のコマンドを記述する。
`sudo hwclock -s`
3. RP にインストールした MATE の `/lib/udev/hwclock-set` ファイルで以下の 3 行をコメント化する。
`if [-e /run/systemd/system] ; then`
`exit 0`
`fi`
4. RP にインストールした MATE の GUI などネットワークに接続し以下のコマンドを順に実行する。
`sudo update-rc.d -f fake-hwclock remove`
`sudo ntpdate -u ntp.jst.mfeed.ad.jp`

4.4.3 無線 LAN アダプタのモード変更プログラム

このプログラム (`monitor.sh`) は RP に接続した無線 LAN アダプタのインタフェースモードをモニターモードに変更し、チャンネルホップ^{*23}するためのプログラムである。`monitor.sh` は Channel Hopping Shell Script のソースコード [4] を基に本研究用に書き換えた。

`monitor.sh` は初回起動時に `/tmp` に LOCK ファイルを作成し、自身の多重起動を阻止する。`crontab` で 1 分毎に `monitor.sh` を起動させるため、この仕組みは必要である。LOCK ファイルとは、ファイルロック機構によって作成されるファイルである。ファイルロック機構は、コンピュータのファイルへのアクセスを一時的に 1 人のユーザーや 1 つのプロセスに制限する機構である。次は無線 LAN アダプタをモニターモードにするために、`ifconfig` で無線 LAN アダプタをダウンさせる。`iwconfig` で無線 LAN アダプタをモニターモードに変更し、`ifconfig` で無線 LAN アダプタをアップさせる。最後は、モニターモードにした無線 LAN アダプタをチャンネルホップする。`while` 文の条件式は、AM08:00 から PM19:00 の間は処理をループさせる。`while` 文の内部処理は無線 LAN アダプタを 2.4GHz 帯でチャンネルホップする。ホップする間隔は 0.1s である。

`monitor.sh` のソースコードは A.1 に記載している。`monitor.sh` を実行したときの画面を図 10 に示す。なお図 10 は、手動で実行した場合に表示される。`crontab` コマンドによるバックグラウンド実行の場合は表示されない。

*22 リアルタイムクロックの略である。コンピュータに内蔵された、電池などで常に動作し続ける時計のことである。

*23 チャンネルを切り換えることである。

```

Switching to channel 1 07:02:38:241912857
Switching to channel 2 07:02:38:388315798
Switching to channel 3 07:02:38:521577745
Switching to channel 4 07:02:38:656242533
Switching to channel 5 07:02:38:788828393
Switching to channel 6 07:02:38:923436679
Switching to channel 7 07:02:39:057875335
Switching to channel 8 07:02:39:190366982
Switching to channel 9 07:02:39:322428263
Switching to channel 10 07:02:39:453956428
Switching to channel 11 07:02:39:583035516
Switching to channel 12 07:02:39:714430537
Switching to channel 13 07:02:39:848896100
Switching to channel 14 07:02:39:980191045
Switching to channel 1 07:02:40:106234449
Switching to channel 2 07:02:40:236749657
Switching to channel 3 07:02:40:368896650
Switching to channel 4 07:02:40:504018876
Switching to channel 5 07:02:40:637276985
Switching to channel 6 07:02:40:771001678
Switching to channel 7 07:02:40:904086498
Switching to channel 8 07:02:41:037173998
Switching to channel 9 07:02:41:170411369
Switching to channel 10 07:02:41:302404104
Switching to channel 11 07:02:41:435915444
Switching to channel 12 07:02:41:568118274
Switching to channel 13 07:02:41:699522714
Switching to channel 14 07:02:41:833891982
Switching to channel 1 07:02:41:966996337
Switching to channel 2 07:02:42:093468182
Switching to channel 3 07:02:42:227309996
Switching to channel 4 07:02:42:359614565
Switching to channel 5 07:02:42:493336685
Switching to channel 6 07:02:42:625970305
Switching to channel 7 07:02:42:760516661
:

```

図 10 monitor.sh の手動実行結果を示したものである。図中左側の Switching to channel と数字は切り替えているチャンネルを示している。図中右側はチャンネルを切り替えた時刻情報を示している。時刻情報を見ると、約 0.1 秒毎にチャンネルが切り替わっているのが分かる。

4.4.4 PR 計測プログラム

このプログラム (roop_tshark.sh) は、PR を計測するためのプログラムである。LOCK ファイルについては 4.4.3 で記述しているため省く。while 文の条件式も同様である。roop_tshark.sh で使用する tshark コマンドのオプションの一部を表 4 に示す。

表 4 tshark コマンドの一部のオプション一覧

オプション	説明
-D	インタフェースの一覧を表示する。その一覧に表示されるインタフェースには番号が割り振られる。
-i number	-D で表示された数字をこのオプションで指定することでインタフェースを指定できる。
-a duration:sec	tshark を起動してから何秒後に終了させるかを指定できる。
-f	計測するパケットの種類を指定できる。本研究では PR を指定する。
-w	保存する際のファイル名を指定できる。

roop_tshark.sh のソースコードは A.3 に記載している。

4.4.5 CPU 使用率記録プログラム

このプログラム (roop_cpu.sh) は万が一、RP がフリーズした場合の原因究明のために、RP の CPU 使用率を記録するために使用する。LOCK ファイル及び、while 文については省く。roop_cpu.sh で使用する sar コマンドのオプションの一部を表 5 に示す。

表 5 sar コマンドの一部のオプション一覧

オプション	説明
-u	CPU に関するレポートの表示項目を制御できる。
-P ALL	全ての CPU をレポートの対象にする。また数字を指定することで、指定秒数毎にレポートを更新する。一例で、60 であれば 1 分毎に更新する。
-e	sar コマンドを指定した時刻で終了させる。

roop_cpu.sh のソースコードは A.2 に記載している。roop_cpu.sh 実行後に作成されるデータの内容を図 11 に示す。

```
Linux 4.4.38-v7+ (hoge-desktop)      2018年07月23日  _arnv7l_      (4 CPU)
09時48分05秒    CPU    %user    %nice    %system    %iowait    %steal    %idle
09時48分15秒    all    10.70    0.00    3.06    49.90    0.00    36.34
09時48分15秒    0     11.30    0.00    3.16    20.57    0.00    64.97
09時48分15秒    1     19.18    0.00    2.81    40.16    0.00    37.85
09時48分15秒    2     7.41     0.00    2.94    51.88    0.00    37.77
09時48分15秒    3     4.84     0.00    3.33    86.78    0.00    5.05

09時48分15秒    CPU    %user    %nice    %system    %iowait    %steal    %idle
09時48分25秒    all    5.89     0.00    5.38    5.94    0.00    82.80
09時48分25秒    0     10.74    0.00    8.47    12.71    0.00    68.08
09時48分25秒    1     4.86     0.00    4.66    1.32    0.00    89.17
09時48分25秒    2     4.08     0.00    3.88    1.94    0.00    90.10
09時48分25秒    3     3.95     0.00    4.55    7.89    0.00    83.60

09時48分25秒    CPU    %user    %nice    %system    %iowait    %steal    %idle
09時48分35秒    all    4.31     0.00    4.24    0.03    0.00    91.42
~
~
~
~
(END)
```

図 11 roop_cpu.sh 実行後、作成されるデータの内容を示したものである。10 秒毎に CPU のログが記録されているのが分かる。図中右端の %idle は CPU の idle 状態を % で示している。一例として、%idle が 36% なら CPU の使用率は 64% である。

4.4.6 計測機器の再起動プログラム

このプログラム (reboot.sh) は計測機器を再起動するためのプログラムである。消費したメモリを開放するために使用する。reboot.sh のソースコードは A.4 に記載している。

4.4.7 使用コマンド一覧

計測機器で PR の計測に使用したコマンド名を以下に示す。

- tshark
GUI で長期間計測した場合 RP がフリーズする可能性があるため、WS の CLI である tshark を使用し、RP に負荷が掛らないようにした。
- sar
数回ほど実験で短期稼働 (3 日間) させた計測機器の内の 1 つがフリーズした。フリーズする原因調査のために、フリーズする直前の CPU 使用率を把握するために使用する。
- iwconfig
無線 LAN アダプタをモニターモードにするために使用する。
- ifconfig
無線 LAN アダプタをモニターモードにする前に、無線 LAN アダプタを up,down させるために使用する。
- crontab
上述するプログラムの定刻及び定間隔の起動を促すために使用する。
- hwclock
RP に接続した RTC で、RP のシステムクロックを合わせるために使用する。
- date
特定の時間帯の間に、プログラムの処理を動かすための条件式に使用する。

4.4.8 モニターモードについて

モニターモードとは、NIC^{*24}が使用できる動作モード^{*25}の 1 つである。ただし全ての NIC がモニターモードに対応するとは限らない。また OS によっては NIC がモニターモードに対応していても変更できないことがある。モニターモードに変更した NIC は常に 1 つのチャンネルでしか、ネットワーク上に流れるパケットをキャプチャできない。このため本研究では無線 LAN アダプタをモニターモードに変更後は、チャンネルホッピングをしている。

PR を計測するためには、NIC をモニターモードにしなければならない。NIC をモニターモードにするための方法として、Linux 系 OS などで無料配布されているパッケージ airclack-ng の中にある airmon-ng コマンドを使う方法がある。airmon-ng コマンドの実行後、NIC の動作モードがモニターモードに変更される。ただし、このコマンドを実行後は NIC のインタフェース名が変わるので、モニターモードに変更後の NIC のインタフェース名を指定するときは注意すべきである。airmon-ng を実行する前と後での無線 LAN アダプタの名前及びモードの変化の結果を図 12,13 にそれぞれ示す。

^{*24} ネットワークインタフェースカードの略である。コンピュータネットワーク内でコンピュータ間の通信を行うために使用されるハードウェアの 1 つである。

^{*25} 他には Master,Managed,Ad hoc,Repeater が存在する。

```
wlx4ce676cfcf6e IEEE 802.11 ESSID:off/any
Mode:Managed Access Point: Not-Associated Tx-Power=20 dBm
Retry short long limit:2 RTS thr:off Fragment thr:off
Power Management:off
```

図 12 airmon-ng コマンド実行前に、iwconfig コマンドを実行した結果を示している。インタフェース名 wlx4ce676cfcf6e が無線 LAN アダプタである。動作モードが Managed になっていることが分かる。

```
wlan0mon IEEE 802.11 Mode:Monitor Tx-Power=20 dBm
Retry short long limit:2 RTS thr:off Fragment thr:off
Power Management:off
```

図 13 airmon-ng コマンド実行後に、iwconfig コマンドを実行した結果を示している。図 12 ではインタフェース名が wlx4ce676cfcf6e だったが、こちらでは wlan0mon に変わっていることが分かる。図 12 では動作モードが Managed だったが、こちらでは動作モードが Monitor に変わっていることが分かる。

補足で、aircrack-ng パッケージの中に airodump-ng コマンドがある。このコマンドは AP とその AP に接続されたクライアント*²⁶のリストを表示する。次に airodump-ng コマンド実行後に表示されるフィールド名について数個記述する。

- BSSID
AP の MAC アドレスを表示する。
- PWR
NIC によって報告される信号レベルを表示する。RSSI である。
- Beacons
AP から送信されるビーコンの数を表示する。
- CH
ビーコンパケットから取得できるチャンネル番号を表示する。
- ESSID
ワイヤレスネットワーク名を表示する。
- Probes
クライアントが探索した ESSID を表示する。現在接続されていないクライアントが接続しようとしているネットワークを表示する。

airodump-ng コマンドではタイムスタンプが取得できない。学生の動きを把握するためには、最低限 MAC アドレスとタイムスタンプが必要である。そのため、本研究では airodump-ng コマンドは使用しない。

*²⁶ 本研究ではモバイルデバイスである。

5 結果と考察

5.1 計測結果

計測機器を4週間設置した。この期間中に得られたMACアドレスの合計数を場所毎に示したグラフを図15に示す。16号館及び5号館で得られたMACアドレスの合計数を日付毎に示したグラフを図16,17にそれぞれ示す。

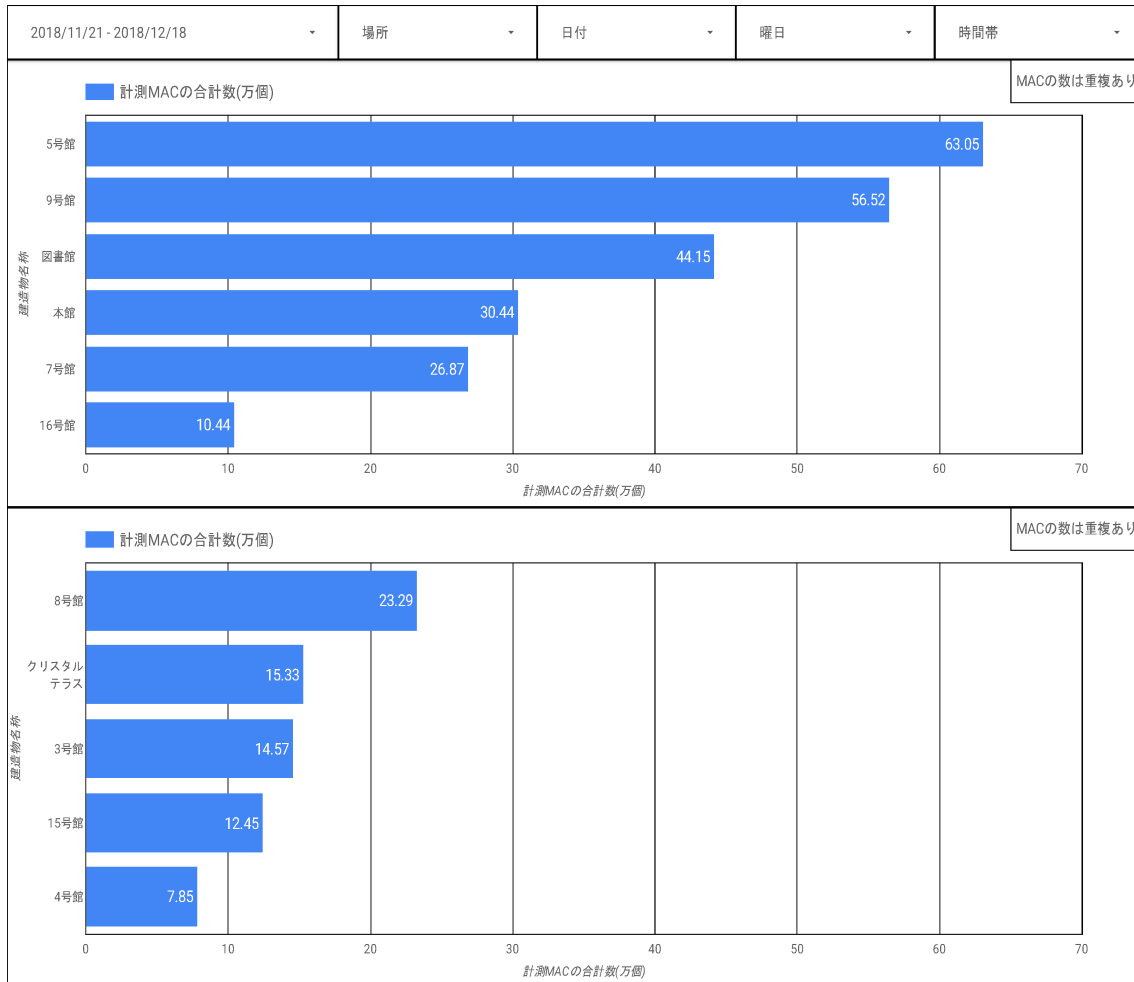


図15 全設置箇所計測できたMACアドレスの合計数を示す。図中の中央の線から上側は中央キャンパス内で得られたMACアドレスを場所ごとに示している。また中央の線から下側は東キャンパスで得られたMACアドレスを場所ごとに示している。このグラフから中央キャンパス内でMACアドレスが多く取れているのが分かる。なお上下のグラフのMACアドレスの数はそれぞれ重複がある。

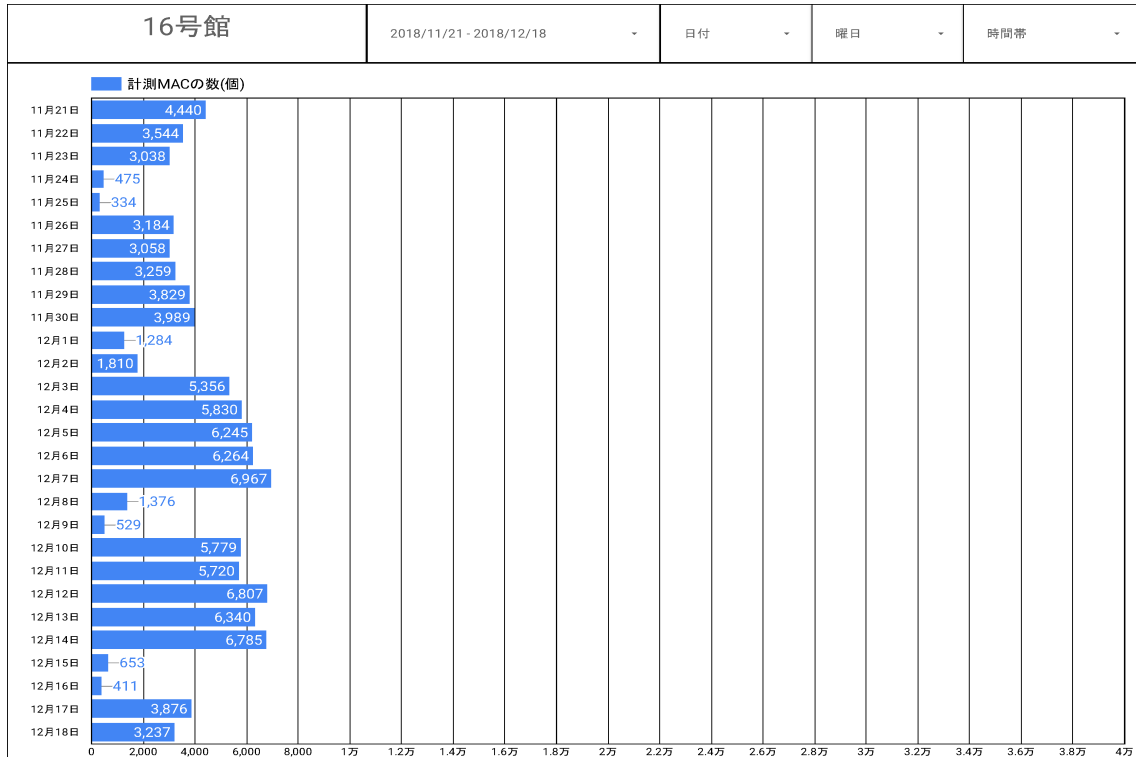


図 16 設置場所の 1 つである 16 号館における得られた MAC アドレスの個数を日付毎に示したグラフである。このグラフの計測できた MAC アドレスの数は重複がある。

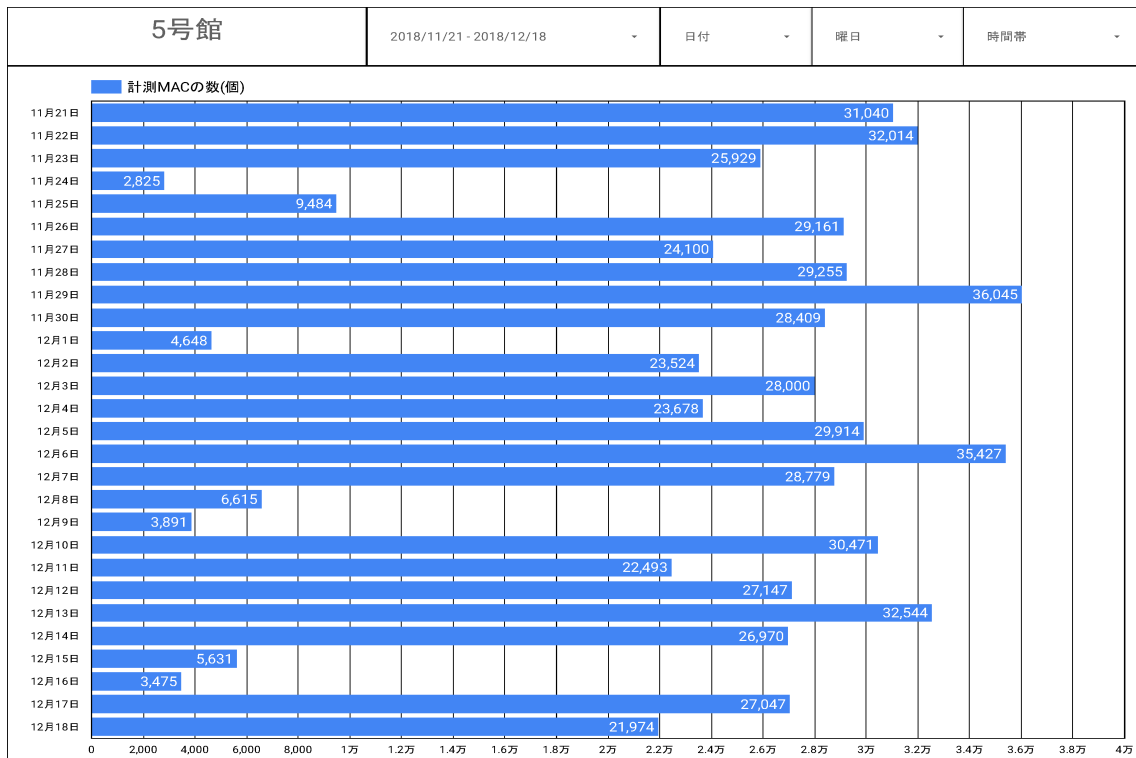


図 17 設置場所の 1 つである 5 号館における得られた MAC アドレスの個数を日付毎に示したグラフである。このグラフの計測できた MAC アドレスの数は重複がある。

5.2 考察

ここでは計測機器によって得られた MAC アドレスの数についての予想及び考察について記述する。

5.2.1 予想

中央キャンパス内で計測できる MAC アドレスの合計数は、東キャンパス内で計測できる MAC アドレスの合計数より多いと予想した。

土曜日曜は、大凡クラブ活動で来ている学生及び土曜に開講される講義受講者しかいないと考える。そのため平日に比べると計測できる MAC アドレスの数は格段に少ないと予想した。

中央キャンパスにある計測機器 6 機の内 4 機は、人通りが最も多いと予想した場所に設置した。人通りが多いと予想した場所は図 1 中の、7 号館、9 号館、5 号館、図書館である。そのため計測できる MAC アドレスの数は、全計測機器の中で特に多いと予想した。

1 人につき 1 つのモバイルデバイスを持っていると仮定する。本学内における学生の人数は、留学生および編転入生含め約 9,000 人である [5](2019 年 1 月現在)。本学の職員は、非常勤講師含め約 900 人である。また本学の付近に大阪桐蔭高校がある。そのため大阪桐蔭高校の職員及び生徒が本学の敷地内を移動することがある。大阪桐蔭高校の職員は、非常勤講師含め約 130 人である。大阪桐蔭高等学校の学生は約 2,000 人である [6](2019 年 1 月現在)。本学の敷地内を移動すると思われる人の合計は約 12,000 人になる。この算出した合計数から、1 日に計測できる設置場所毎の MAC アドレスは 20,000 個前後と予想した。20,000 個前後と予想した理由は、1 人が複数のモバイルデバイスを持つことがあるからである。

5.2.2 計測結果の考察

図 15 から、中央キャンパス内で計測できた MAC アドレスの合計数が、東キャンパス内で計測できた MAC アドレスの合計数より多いのは予想通りである。

図 16,17 から、土曜日曜に計測できた MAC アドレスの合計数が、平日より少ないのも予想通りである。

中央キャンパスにおいて、人通りが最も多いと予想した場所に設置した計測機器 4 機の計測できた MAC アドレスの数は、図 15 を見るかぎり上位を占めているが、7 号館が本館で計測できた MAC アドレスの数より少ないため、予想が外れた。予想が外れた原因は不明である。

図 15 から、5 号館で計測できた MAC アドレスの数は、4 週間で約 630,000 個である。これを 4 で割り、さらに 7 で割ると 1 日で約 20,000 個計測できた計算になる。この計算で考えた場合、計測できた MAC アドレスの数が 9 号館以下の場所は、20,000 個を下回っている。1 日に計測できる設置場所毎の MAC アドレスの数は、5 号館及び 9 号館に限れば約 20,000 個計測できているが、これら以外は 1 日に計測できた MAC アドレスの数は 20,000 個を下回っているため、予想が外れた。

16 号館は中央キャンパス内にあり、なおかつ計測機器の設置場所は 5 号館と同じく 1 階のため、本来ならば 5 号館と同様の数の MAC アドレスを計測できると考えていた。だが図 16 の 16 号館は、5 号館と比べて計測できた MAC アドレスの数に圧倒的な差がある。16 号館で計測できた MAC アドレスの数が、5 号館で計測できた MAC アドレスの数に比べて少ない原因は、開講される講義の数の差だと考える。5 号館は講義受講のために学生がよく訪れる。学生がよく訪れるために、平日に限り一日毎で約 20000 個の MAC アドレスが計測できたと考える。16 号館は 5 号館と比べれば開講される授業の数の差がある。

5.3 プログラムの評価

作成したプログラムの課題は以下の通りである。

- monitor.sh 及び roop_tshark.sh のソースコードにパスワードを記述する。
万が一計測機器が盗まれた場合、ソースコードに記述したパスワードを見られることで、計測した PR を悪用される可能性があった。パスワードが記述されているプログラムには chmod^{*28} コマンドで root(管理者) ユーザー以外内容の表示をできないようにすべきである。
- reboot.sh の存在
crontab の実行するコマンドを指定する箇所、reboot と書くだけである。reboot.sh ファイルは必要ない。
- crontab のソースコードにミスがある
crontab のソースコードの 4 行目及び 5 行目にミスがある。作成当時は毎日 AM07:00 と PM20:00 に 1 回再起動を実行するよう記述したが、この記述では毎日 PM20:00 の間に 1 分毎に再起動を実行する。5 行目も同様である。これは明らかなミスである。毎日 PM20:00 に 1 回再起動を実行する正しい記述は [0 20 * * * /home/hoge/reboot.sh] である。AM07:00 の場合についても同様である。

^{*28} ファイルの Permission を変更するコマンド。Permission とは、ファイルごとに定義された、読み出し・書き込みなどのアクセスに対する許可情報のことである。

6 結論

本研究では PR 人流把握手法から、モバイルデバイスが発信する PR を計測する計測機器を作成した。作成した計測機器を本学内の 11 箇所に設置することで PR を計測し、計測期間中 (4 週間) の本学の学生の移動を把握した。学生の移動を把握することで、受講環境の実態を明かにするシステムを開発することができた。

6.1 今後の課題

本学内における学生の移動を把握し、移動時間を算出することで、受講環境の実態を明らかにするシステムは完成した。だが計測機器の設置個数が本学内の面積に比べて少ない。そのためおおまかな学生の移動しか分からない。また計測し保存したデータは全て設置場所に赴き、手作業でデータを回収するため非常に手間である。本研究で利用した PR 人流把握手法は、モバイルデバイスが PR を発信しなければ、学生の移動を把握することはできない。モバイルデバイスが PR を発信する間隔は OS によって異なる。そのため計測できなかった MAC アドレスが存在する可能性がある。図 15 で計測できた MAC アドレスの数を示したが、その内の何割が本学の学生なのか不明である。本研究では、計測できた MAC アドレスを学生と判断し、移動時間を算出した。だが本学には本学生以外にも様々な人が行き交う。そのため計測できた MAC アドレスは全て学生とは言えない。

本研究の課題及び改善点を以下に纏める。

- 計測した PR のデータを計測機器内で、定期的にバッチ処理^{*29}で匿名化する仕組みの作成
- バッチ処理で匿名化したデータを自動でクラウドサービスに送信する仕組みの作成
- 計測機器の設置場所及び個数を増やし、より詳細な学生の移動を把握できるようにする
- 計測できなかった MAC アドレスが存在する可能性を下げる
- 計測時に学生と部外者を分ける方法を考える

^{*29} バッチ処理とは、データを一纏めに処理することである

謝辞

本論文執筆及び研究作業等、大垣斉准教授からご指導及びご協力を頂きました。また本研究を進める上で、ご協力頂いた各計測機器設置場所で業務を行われていた本学事務職員の皆様、情報教育システム研究室の皆様、その他見守ってくださった方々に深く感謝いたします。

参考文献

- [1] Pejman Roshan and Jonathan Leary. *802.11 Wireless LAN fundamentals*. Cisco press, 2004.
- [2] Frequency management and thermal control. <https://www.raspberrypi.org/documentation/hardware/raspberrypi/frequency-management.md>. RP の熱制御について記述されている。一定の温度に達すると、周波数を下げて温度上昇値を抑える。
- [3] 森本哲郎, 辻本悠佑, 白浜勝太, 上善恒雄. Wi-fi パケットセンサを用いた人流解析と可視化. データ工学と情報マネジメントに関するフォーラム (DEIM2015) 論文集, 2015.
- [4] Angela Orebaugh, Gilbert Ramirez, and Jay Beale. *Wireshark & Ethereal network protocol analyzer toolkit*. Elsevier, 2006.
- [5] <http://www.osaka-sandai.ac.jp/info/public/archive/data.html>. 大阪産業大学の公式サイト。
- [6] <http://www.osakatoin.ed.jp/about/profile.html>. 大阪桐蔭中学校高等学校の公式サイト。

付録 A ソースコード

A.1 ソースコード.1

ソースコード 1 monitor.sh

```
1 #!/bin/sh
2 pass="*****\n"
3 IFACE=wlx4ce676cfcf6e
4 IEEE80211bg="1_2_3_4_5_6_7_8_9_10_11"
5 IEEE80211bg_intl="$IEEE80211bg_12_13_14"
6 IEEE80211a="36_40_44_48_52_56_60_64_149_153_157_161"
7 IEEE80211bga="$IEEE80211bg_$IEEE80211a"
8 IEEE80211bga_intl="$IEEE80211bg_intl_$IEEE80211a"
9
10 export LOCKFILE=/tmp/LOCK.monitor
11 if [ -e ${LOCKFILE} ]
12 then
13     exit 0
14 fi
15 trap "rm -f $LOCKFILE; exit 0" 0 1 2 3 15
16 touch ${LOCKFILE}
17
18 echo $pass | sudo -S ifconfig $IFACE down
19 sudo iwconfig $IFACE mode monitor
20 sudo ifconfig $IFACE up
21
22 while [ `date +%H%M` -lt 1900 -a `date +%H%M` -ge 0800 ]
23 do
24     for CHAN in $IEEE80211bg_intl ; do
25         echo "Switching to channel $CHAN"
26         sudo iwconfig $IFACE channel $CHAN
27         sleep 0.1s
28     done
29 done
30
31 exit 0
```

A.2 ソースコード.2

ソースコード 2 roop_cpu.sh

```
1 #!/bin/sh
```

```

2
3 export LOCKFILE=/tmp/LOCK.cpu
4 if [ -e ${LOCKFILE} ]
5 then
6     exit 0
7 fi
8 trap "rm_f_${LOCKFILE};_exit_0" 0 1 2 3 15
9 touch ${LOCKFILE}
10
11 while [ `date +%H%M` -lt 1900 -a `date +%H%M` -ge 0800 ]
12 do
13     sar -u -P ALL 10 -e 19:00:00 >> /home/hoge/`date +%Y%m%d_%H%M`_cpu_rp2
14     .txt
15 done
16 exit 0

```

A.3 ソースコード.3

ソースコード 3 roop_tshark.sh

```

1 #!/bin/sh
2
3 pass="***\n"
4
5 export LOCKFILE=/tmp/LOCK.tshark
6 if [ -e ${LOCKFILE} ]
7 then
8     exit 0
9 fi
10 trap "rm_f_${LOCKFILE};_exit_0" 0 1 2 3 15
11 touch ${LOCKFILE}
12
13 while [ `date +%H%M` -lt 1900 -a `date +%H%M` -ge 0800 ]
14 do
15     echo "$pass" | sudo -S tshark -i 4 -a duration:3600 -f "subtype_
16     probereq" -w /home/`date +%Y%m%d_%H%M`_rp2.pcap
17 done
18 exit 0

```

A.4 ソースコード.4

ソースコード 4 reboot.sh

```
1 #!/bin/sh
2
3 reboot
```

A.5 ソースコード.5

ソースコード 5 crontab

```
1 * 8-19 * * * /home/hoge/monitor.sh
2 * 8-19 * * * /home/hoge/roop_tshark.sh
3 * 8-19 * * * /home/hoge/roop_cpu.sh
4 * 20 * * * /home/hoge/reboot.sh
5 * 7 * * * /home/hoge/reboot.sh
```