

# 仮想化ネットワークインターフェースを用いたトリガ駆動に基づく Wi-Fi アクセスポイントの動的負荷分散方式

河田 真宏<sup>1</sup> 玉井 森彦<sup>1</sup> 安本 慶一<sup>1</sup>

**概要:** 本稿では、無線 LAN 環境において、仮想化された無線ネットワークインターフェース (WNIC) を用いることで、ハンドオーバーによるオーバーヘッドを可能な限り小さくし、トラフィックの変化に対して動的にアソシエーションを更新することで、アクセスポイント (AP) 間の負荷分散を実現する方式を提案する。大規模な無線 LAN 網における AP 間の負荷分散方式は、これまでに多数提案されてきた。しかし既存方式では、トポロジの更新が頻繁に行なわれるとハンドオーバーによるオーバーヘッドで通信品質の劣化を招き、また更新周期が長いとトラフィックの変動への追従が不十分になるという問題がある。本研究では、端末が装備する WNIC の仮想化を行い、複数の AP と同時に接続可能にすることで、AP 間のハンドオーバーによるオーバーヘッドを可能な限り小さくする。その上で、無線 LAN 網のトラフィックの変化を継続的にモニタリングし、その結果を利用して適切なタイミングでトポロジの更新を行う。ns-3 を用いたシミュレーションにより、提案方式における集約スループット、および端末間スループットの公平性について、受信電波強度が最大の AP に接続する方式 (レガシ方式) などとの比較を行った。その結果、レガシ方式と比べると、集約スループットは約 59% 増加し、また Jain's Fairness Index を用いた公平性については、レガシ方式と比べ約 28% 増加することが分かった。

## Trigger-based Dynamic Load Balancing among Multiple Wi-Fi Access Points Using Virtualized Network Interfaces

MASAHIRO KAWADA<sup>1</sup> MORIHIKO TAMAI<sup>1</sup> KEIICHI YASUMOTO<sup>1</sup>

**Abstract:** In this paper, we propose a method, which dynamically adapts association topology to the traffic variation, for dynamic load balancing among multiple access points (APs) in wireless LANs (WLANs), while keeping the handover overhead negligible using virtualized wireless network interfaces (WNICs). A number of load balancing methods for large scale WLANs have been proposed. However, in most of the existing methods, a difficulty in determining timings of topology updates arises from the fact that the frequent updates of topology incur large handover overhead that deteriorates communication quality, while the longer update periods cause insufficient adaptation to the traffic variation. In our proposed method, to make the handover overhead as small as possible, the WNIC on a station is virtualized and connected to multiple APs at the same time. In addition, a centralized controller, which is located at a wired network, monitors the traffic variation of WLAN continuously, and triggers topology updates at appropriate timings based on the monitored results. We conducted simulation-based evaluation experiments with ns-3 where we measured aggregated throughput and fairness of throughput among stations by comparing our method with other methods such as RSSI-based AP selection method (legacy method). From the experiments, we observed that the aggregated throughput and the Jain's fairness index of our method are about 59% and 28% higher than that of the legacy method, respectively.

## 1. はじめに

無線 LAN ネットワーク (WLAN) の普及に伴い、オフィス、大学、公共施設など多くの場所で大規模な WLAN が敷設されるようになった。WLAN では、多くのデバイス (ノート PC, スマートフォン, ゲーム機, タブレット等) が各アクセスポイント (AP) の限られた資源を共有する。このとき、各端末と各 AP 間の接続関係 (トポロジ) について、特定の AP に多数の端末が接続したり、またはトラフィックが集中することで、AP 間で負荷の不均衡が生じる場合がある。このような負荷の偏りは、特定の端末のスループットが極端に低下したり、特定の AP の資源が有効活用されないという問題を引き起こす。

WLAN における AP 間の負荷分散方式について、盛んに研究が行なわれている [1-9]。多くの既存研究では、トポロジの更新を、新規端末が WLAN に参加する際か、もしくは一定の周期に従って行う。しかし、WLAN のトラフィックは、様々な要因 (各端末の位置や使用帯域量など) により刻一刻と変化する。そのため多くの既存方式では、AP 間の負荷に偏りが生じて、それに対して長時間対応できないままになってしまうという問題が生じる。

周期的にトポロジを更新する場合、周期が長ければトラフィックの変動に対し追従が遅れてしまう。一方、トポロジを更新するためには、端末の接続先 AP を切り替える必要があり、それには制御メッセージの交換などで数百ミリ秒のオーダの遅延が発生する [10]。そのため、周期を短くするとオーバーヘッドが大きくなり、スループットの低下をまねく可能性がある。

この問題を解決するため、本稿では、無線ネットワークインタフェース (WNIC) の仮想化方式を利用した、トリガ駆動に基づく AP 間の負荷分散方式を提案する。提案方式では、トポロジの更新に伴うオーバーヘッドをできるだけ小さくするため、端末の物理 WNIC を仮想化して複数の WNIC に見せかけることで、端末を複数の AP へ同時接続させ、AP を切り替える際の制御メッセージの交換を不要にする。その上で、有線ネットワーク上のコントローラにより WLAN のモニタリングを行い、AP 間の負荷に偏りが生じたと判断された場合には、トポロジを即座に更新して AP 間の負荷を均等化する。

ns-3 シミュレータを用いて、提案方式と既存方式との性能を比較した。その結果、提案方式は、RSSI (Received Signal Strength Indication) が最大の AP へ接続するレガシ方式に比べ、集約スループットを最大で約 59% 向上でき、また、端末間の公平性の指標である Jain's Fairness Index については、最大で約 28% 向上できることを確認した。

## 2. 関連研究

従来用いられてきた RSSI が最大の AP へ接続する方式では、端末の位置の偏りにより一部の AP へ接続が集中するという問題が生じる。そのため以下で述べるように、AP 間の負荷分散を実現する方式が多数考案された。

Fukuda ら [1] は MLT (Maximizing Local Throughput) と呼ばれる AP 選択方式を提案した。この方式では、各端末が、近隣の AP から受信する probe response および beacon フレームから AP の RSSI と接続端末数を取得し、それに基づき各 AP の期待スループットを求め、その値が最大の AP へ接続する。また Yen ら [2] は、SNMP を用いて WLAN をモニタリングし、その結果に基づきトポロジの更新を行う方式を提案した。有線ネットワーク上のサーバが、各 AP の情報を格納した MIB-II オブジェクトを監視し、端末は必要に応じてサーバから AP の情報を取得し接続先 AP を決定する。さらに Bejerano ら [3] は、AP 間の負荷分散に加え、端末間の公平性についても考慮した方式を提案した。

上記以外にも様々な WLAN 環境を考慮した研究がある。文献 [4] では、beacon の送信電力を制御し、インフラ側で各 AP のカバレッジを調整することで、端末が自律的に接続先 AP を変更するのを促す方式が提案されている。文献 [5] では、端末間で均等に帯域を共有するために、Gibbs Sampler に基づく AP 選択とチャンネル割り当て方式が提案されている。文献 [6] では、多数の端末が同時に WLAN に参加して引き起こされる Flash Crowd 問題を解決するために、キューを利用した AP の接続端末数の管理方式が提案されている。文献 [7] では、マルチレート環境下において、自律分散的に各端末の公平性を均等化するための方式が提案されている。文献 [8] では、期待スループットの予測に主眼を置き、その結果を AP 選択に反映させる方式が提案されている。文献 [9] では、IEEE 802.11n に関して、IEEE 802.11a/b/g の端末が存在する場合の影響について調査し、異種の端末が混在する環境下で最適なトポロジを計算する方式が提案されている。また、エンタープライズ向けの商用 AP 製品のベンダである Aruba Networks 社では、端末が新規参加する際、接続を試みた先の AP が接続要求を受け入れるかどうかについて、チャンネル利用率に関する閾値によってその決定を行う機能を導入している [11]。これら既存研究では、ある時点で得られる WLAN の情報から、その時点における最適なトポロジを計算するための方式に焦点が当てられている。一方本研究では、時間の経過に伴い変化するトラフィックに対し、適切なタイミングでトポロジを更新するための方式を提案することを目的としている点で、既存研究と異なる。

<sup>1</sup> 奈良先端科学技術大学院大学  
Nara Institute of Science and Technology

### 3. 接続先 AP の切り替えにより生じるオーバーヘッドとその解決策

トラフィックの変化にすばやく追従するためには、頻りにトポロジを更新することが求められるが、それを実現するためには、端末の接続先 AP の切り替えによるオーバーヘッドを削減する必要がある。本章では、まず、IEEE 802.11 におけるハンドオーバー手順とオーバーヘッドの原因について説明する。次に、そのオーバーヘッドを削減するための、WNIC の仮想化方式について述べる。

#### 3.1 IEEE 802.11 におけるハンドオーバーのオーバーヘッド

IEEE 802.11 でのハンドオーバーには、いくつかの制御メッセージの交換を必要とする。まず、端末は現在の AP に対する接続を解除するためのメッセージを交換する。次にチャンネルスキャンを行い、近隣の AP を探索する。これには、端末が probe request を送信し、AP がそれに probe response で答えるアクティブスキャンか、もしくは AP が定期送信する beacon を端末が受信するパッシブスキャンが用いられる。チャンネルスキャンは、全ての利用可能チャンネルに渡って行われるため、大きなオーバーヘッドとなる。その後、端末は切り替え先 AP に対し認証と接続のためのメッセージの交換を行う。

Mishra ら [10] により、上記の手順で要する時間について 802.11b での調査が行われており、最大で 500 ミリ秒程度必要であったと報告されている。このオーバーヘッドを削減するため、文献 [12, 13] では、高速ハンドオーバー方式が提案されているが、認証や再接続処理に伴うオーバーヘッドまで削減することは難しい。

別のアプローチとして、複数の WNIC を用いる Multi-Scan と呼ばれる方式が Brik ら [14] により提案されている。この方式では、端末は 2 つの物理 WNIC を搭載し、一方をプライマリ、もう一方をセカンダリとして用いる。プライマリが接続する AP とはデータの送受信を行う。一方セカンダリでは、別の接続先 AP の候補をバックグラウンドで探索する。もし条件のよい新たな AP を発見した場合、セカンダリが新規 AP と接続した後、プライマリとセカンダリの役割を交換する。これにより、AP の切り替えに伴う制御メッセージの交換が不要となる。しかし、物理 WNIC を複数搭載することによりコストや消費電力が増加するため、現実的には利用が困難である。

#### 3.2 WNIC の仮想化方式

物理的に複数の WNIC を搭載するのと同様の効果を単一の WNIC で実現するため、近年、WNIC の仮想化方式が提案されている [15-18]。Chandra ら [15] は、1 つの WNIC を仮想化し、物理 WNIC 上でステーションモードと AP モードを同時に稼働する方式を提案した。この方式は、現

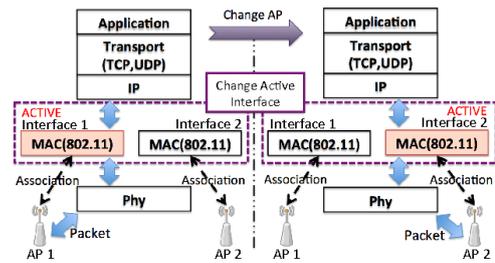


図 1 仮想化 WNIC を用いた AP 切り替えの様子

在 Windows 7 に搭載されている [19]。Kandula ら [16] は、仮想化 WNIC を用いて、複数の AP と同時接続を行ない、近隣 AP の有線側のバックホール帯域を集約する方式を提案している。

図 1 に、端末のネットワークスタック上での仮想化 WNIC の動作の様子を示す。各仮想化 WNIC は、各々異なる AP に対し接続を完了し、既に通信が可能状態となっている。MAC 層では、各 AP 毎に個別に状態管理を行う必要があるため、各仮想化 WNIC に対応して個別の MAC 層が割り当てられる。物理 WNIC は一つだけであるため、ある時点で通信可能な AP は一つのみである。現在通信を行っている AP へ接続する仮想化 WNIC をアクティブインタフェースと呼び、アクティブインタフェースの接続先 AP をアクティブ AP と呼ぶ。また、アクティブインタフェース以外の仮想化 WNIC をアイドルインタフェースと呼ぶ。各アイドルインタフェースは AP へ接続済みであるため、アクティブインタフェースの切り替えに際して、3.1 節で述べたオーバーヘッドなしにアクティブ AP の切り替えが行える。なお文献 [18] によると、チャンネル変更を必要とする場合でも、アクティブインタフェースの切り替えは、3ms 程度で実現可能であると報告されている。

本研究では、仮想化 WNIC の実現についてはこれら既存方式をそのまま用いて実現するものとする。

### 4. AP 間の動的負荷分散方式

本章では、トラフィックの変化にすばやく追従してトポロジの更新を行うための、トリガ駆動に基づく負荷分散方式について述べる。

#### 4.1 WLAN のモニタリングとトポロジの更新のための制御メッセージ

WLAN においてトポロジを管理する方法は、大きく集中管理型と自律分散型の 2 つに分類できる。自律分散型では、各端末が自律的に接続先 AP の選択を行う。一方集中管理型では、有線ネットワーク上のコントローラによってトポロジを集中的に管理するため、ピンポン効果（端末が 2 つの AP 間で接続と切断を繰り返す）のような端末と AP 間の協調ミスの発生を防ぎやすい。提案方式では集中管理型によりトポロジを管理する。

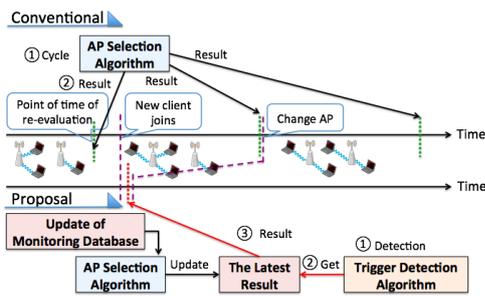


図 2 トリガ駆動に基づくトポロジの更新方式（上部）と周期的なトポロジの更新方式（下部）との違い

提案方式では、以下のメッセージを用いて WLAN のモニタリングを行う。(1) 端末ステータスアナウンスメッセージは、端末の情報を AP に送信するためのメッセージであり、各仮想化 WNIC の接続先 AP の MAC アドレスと RSSI、および端末の MAC アドレスが含まれる。(2) AP ステータスアナウンスメッセージは、AP とその AP に接続している端末の情報をコントローラへ送信するためのメッセージであり、端末ステータスアナウンスメッセージから得られる情報と、AP の MAC アドレス、および AP に接続する全端末の伝送レートと帯域使用量が含まれる。

これらのメッセージは、端末から AP、または AP からコントローラへ定期的に送信される。コントローラは、これらの情報をデータベース（以下、モニタリングデータベースと呼ぶ）に格納し、4.2 節で述べる方法に従い、トポロジの更新が必要かどうかを判断する。トポロジの更新が必要な場合、アクティブ AP の切り替えを行う各端末に対し、AP 切替リクエストメッセージが送信される。それを受信した端末は、アクティブ AP を切り替え、AP 切替レスポンスメッセージで応答する。

#### 4.2 トリガ駆動に基づくトポロジの更新

図 2 の上部に、既存方式（トポロジの更新を周期的に行う方式）でのトポロジの更新タイミングを示す。既存方式では、端末の参加に伴い AP 間の負荷に偏りが生じた場合、次の更新タイミングまで待たなければトポロジが更新されず、その間端末のスループットが低下する可能性がある。

一方提案方式では、図 2 の下部に示すような構成に従い、トリガ駆動に基づくトポロジの更新を実現する。提案方式では、前節で述べたモニタリングの結果に基づき常にトラフィックの変化を監視しておき、トラフィックに一定量の変化が生じた場合、すぐにトポロジの更新を実行する。トラフィックの変化に対し、トポロジの更新を発動するためのアルゴリズムをトリガアルゴリズムと呼ぶ。

提案方式では、AP 選択アルゴリズムとして実際にどの方式を採用するかについて、無線 LAN 網の管理者が、ネットワークの特性やポリシーに従い、2 章で述べたような様々な方式から自由に選べるようにするため、トリガアル

ゴリズムと AP 選択アルゴリズムがそれぞれ独立に並行して動作するというアーキテクチャを採用した。AP 選択アルゴリズムでは、モニタリングデータベースの更新があった際に、逐次最新のモニタリング結果に基づき最適なトポロジを計算し、その結果を保持しておく。一方トリガアルゴリズムでは、モニタリングの結果に基づき、一定量のトラフィックの変化があった際に、AP 選択アルゴリズムの最新の出力結果を参照し、そのトポロジをネットワークへ反映させる。

トリガアルゴリズムでは、トラフィックの変化による AP の負荷への影響をとらえるため、各 AP と通信する端末の使用帯域と、AP が端末との通信において使用する伝送レートに着目し、それらの値を次のように利用してトリガを発動させる。

対象の無線 LAN 網における端末の数を  $N$  とし、AP の数を  $M$  とする。ある AP  $j$  (ただし、 $0 \leq j \leq M$ ) をアクティブ AP とする端末の数を  $N_j$  とする。AP  $j$  で計測される端末  $i$  (ただし、 $0 \leq i \leq N_j$ ) の使用帯域を  $TrafficRate(j, i)$  とする。また端末  $i$  が、AP  $j$  と通信する際に使用する伝送レートを  $TxRate(j, i)$  とする。さらに、AP  $j$  をアクティブ AP とする全端末の内、 $j$  との通信で使用されている伝送レートの最小値を  $TxRate_{min}(j)$  とする。トリガアルゴリズムでは、以下の 2 つのスコアを各 AP  $j$  に対し計算する。

$$S(j) = \sum_{i=1}^{N_j} \frac{TrafficRate(j, i)}{TxRate(j, i)}, \quad (1)$$

$$S_{min}(j) = \sum_{i=1}^{N_j} \frac{TrafficRate(j, i)}{TxRate_{min}(j)}. \quad (2)$$

式 (1) は、AP  $j$  をアクティブ AP とする全ての端末の帯域使用率の合計を表す。低い伝送レートを用いる端末が、高い伝送レートを持つ端末に対して悪影響を与える [20] ことを考慮し、低い伝送レートで通信する端末は、AP に対しより大きな負荷を与えるものとする。また、AP  $j$  が提供できる総帯域量は、 $TxRate_{min}(j)$  の値が小さくなるにつれ減少する。そのため式 (2) では、 $TxRate_{min}(j)$  を用いて計算した場合の帯域使用率により、AP  $j$  の負荷を表す。

以上のスコアを用いて、ネットワーク状況の変化を検知するため、現在のスコアと前回計測したスコアを比較する。前回トリガが発動した際に計測された  $S(j)$  を  $S^{prev}(j)$ 、同様に  $S_{min}(j)$  を  $S_{min}^{prev}(j)$  と表記する。

もし  $S_{min}^{prev}(j) < S(j)$  となれば、AP  $j$  の負荷が増加したと判断してトリガを発動させる。一方で  $S^{prev}(j) > S_{min}(j)$  となれば、AP  $j$  の負荷が減少したと判断して、この場合もトリガを発動させる。このアルゴリズムを Algorithm1 に示す。

### Algorithm 1 トリガアルゴリズム

Require: 対象の無線 LAN 網に存在する AP の数を  $M (M > 1)$  とする

```

1: while true do
2:   if モニタリングデータベースが更新された then
3:     for  $j = 1$  to  $M$  do
4:        $S(j)$  と  $S_{min}(j)$  を計算
5:       if  $S_{min}^{prev}(j) < S(j)$  or  $S_{min}^{prev}(j) > S_{min}(j)$  then
6:          $S_{min}^{prev}(j) \leftarrow S(j)$ ,  $S_{min}^{prev}(j) \leftarrow S_{min}(j)$ 
7:         AP 選択アルゴリズムの最新結果を参照
8:         トポロジの更新に伴い, アクティブインタフェースの切り替えを行う必要のある各端末に対し, AP 切り替えリクエストメッセージを送信
9:       break
10:    end if
11:  end for
12: end if
13: end while

```

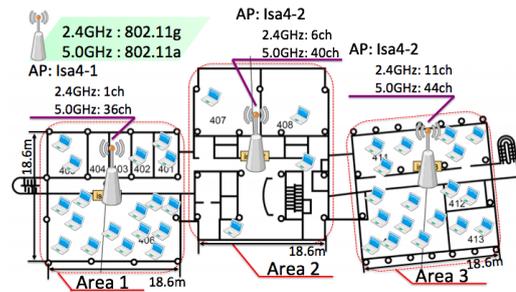


図 4 想定フィールドと各 AP および端末の配置と設定

$Candidate_k$  に切り替える．この操作を，ボトルネック AP の帯域使用率の改善がなくなるまで繰り返し，その最終結果を AP 選択アルゴリズムの最新の解として出力する．

図 3 にこのアルゴリズムの適用例を示す．3つの AP が存在する無線 LAN 網を考える．AP  $b$  の使用率は  $y_b = 1/6$  であり，AP  $b$  がボトルネック AP である．ここで，端末 2 と端末 3 は共に AP  $b$  をアクティブ AP としている．このとき，端末 3 に注目すると，AP  $c$  について高い RSSI 値 (-70 dBm) を観測しているため，端末 3 を切り替え候補とする．端末 3 のアクティブ AP を  $c$  に切り替えたと仮定して，各 AP の使用率を再計算する．その結果，AP  $c$  の使用率は  $y_c = 1/8$ ，AP  $b$  の使用率は  $y_b = 1/48$  となり，ボトルネック AP が  $b$  から  $c$  へ変更される．また，これ以上ボトルネック AP の使用率が改善できるケースは存在しないため，ここでアルゴリズムが終了する．

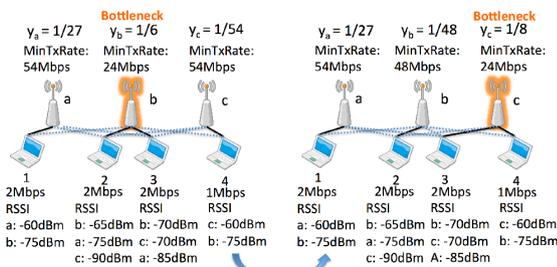


図 3 AP 選択アルゴリズムの実行例．ボトルネック AP  $b$  を検出し，それをアクティブ AP とする端末 3 を，別の AP  $c$  へ切り替える

### 4.3 AP 選択アルゴリズム

提案方式が採用するアーキテクチャでは，無線 LAN 網の管理者のポリシーなどに従い，様々な AP 選択アルゴリズムを導入できる．本稿ではその典型例として，全 AP の中で帯域の使用率が最大の AP を検出し，その AP の使用率を最小化するアルゴリズムを導入する．

AP  $j$  の帯域使用率を  $y_j = S_{min}(j)$  と定義する．この値が最も高い AP をボトルネック AP と呼ぶ．ボトルネック AP をアクティブ AP とする各端末  $i$  において， $i$  のアイドルインタフェースを介して接続する AP の集合を求め，その集合の中で最も高い RSSI 値である AP を選択する．このとき，端末  $i$  が選択した AP を  $Candidate_i$  と表記し，端末  $i$  の  $Candidate_i$  に対する RSSI 値を  $RSSI_i$  と表記する．ここで，ボトルネック AP をアクティブ AP とする全端末の内， $k = \arg \max_i RSSI_i$  となる端末  $k$  に注目する． $k$  のアクティブ AP を， $Candidate_k$  に切り替えたときのみした場合の各 AP の帯域使用率を計算し，ボトルネック AP が別の AP に変更される場合，すなわち，切り替え後のボトルネック AP の帯域使用率が，切り替え前のボトルネック AP の帯域使用率を下回る場合， $k$  のアクティブ AP を

## 5. 性能評価

本章では，提案方式の評価のため，集約スループットと端末間のスループットの公平性に関して，各クライアントが最も高い RSSI の AP へ接続する方式（レガシ方式）と，MLT 方式 [1] との計算機シミュレーションによる比較を行った結果について述べる．

### 5.1 シミュレーションの設定

実験では，シミュレータとして ns-3 [21] を用いた．シミュレータでは，IEEE 802.11 MAC の動作を再現し，WEP や WPA，または IEEE 802.1X のようなセキュリティメカニズムについては考慮しないものとした．

シミュレーションの対象フィールドとして，図 4 に示すように奈良先端科学技術大学院大学の情報科学研究科 A 棟 4F を想定した．対象フィールドでは，エリア 1 およびエリア 3 が研究室，エリア 2 が共有スペースである．

エリア 1 および 3 には，端末を各々 18 台ランダムに配置し，エリア 2 には 4 台の端末をランダムに配置した．エリア 1 および 3 は研究室であり，エリア 2 よりも多めに端末が存在するものとした．各 AP の動作モード，およびチャネルは，図 4 に示すように 802.11a/g を混在して利用する環境を想定して設定した．シミュレーションの各種パ

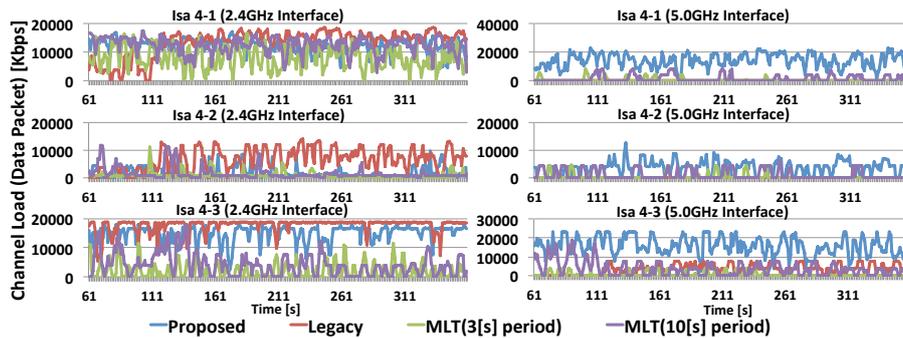


図 5 各 AP の負荷の時間変化

表 1 シミュレーションの設定

シミュレーション時間	360 秒 (ウォームアップ期間 60 秒)
シミュレーション回数	5 回
トランスポート層	UDP
MAC 層	IEEE 802.11a/g
Offered Load	4 Mbps
パケットサイズ	1024 Byte
レート制御	ARF
RTS/CTS	OFF
送信電力強度	16dBm
フェージングモデル	レイリー分布
パスロスモデル	壁による減衰を考慮した 物理層シミュレータ

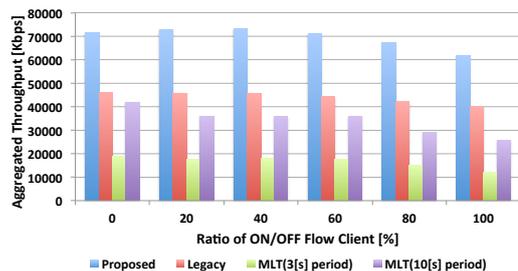


図 6 集約スループット

ラメタの設定を表 1 に示す。なおパスロスモデルの構築には、壁による減衰の影響を考慮するため、各地点のパスロスデータを Scenargie [22] に搭載されている物理シミュレータによって求め、そのデータを利用した。

トラフィックフローのモデルとして、常にトラフィックが流れるモデルと、断続的にトラフィックが流れる ON/OFF フローモデルの 2 つを用いた。ON/OFF フローモデルを利用する端末を ON/OFF フロークライアントと呼ぶ、トラフィックが流れる ON 期間は 5 秒、一方 OFF 期間は平均 5 秒の指数分布に従い生成した。ON/OFF フローモデルは、断続的にトラフィックを発生させる Web ブラウジングによるトラフィックをモデル化することを意図している。提案方式では、ネットワーク全体を管理するコントローラが必要なため、それを有線ネットワーク上に設置した。

提案方式に対する比較方式として、レガシ方式と MLT 方式 [1] を用いた。レガシ方式では、RSSI の最も大きい AP へ接続した後、その AP を使用し続ける。MLT 方式では、各 AP の接続端末数と RSSI に基づく期待スループットを求め、その値が最大の AP へ接続する。MLT 方式では、トポロジの更新周期として、3 秒と 10 秒を用いた。

提案方式では、AP 選択アルゴリズムとして 4.3 節で述べたものを使用した。

## 5.2 集約スループットと公平性の評価

ON/OFF フロークライアントの数を 0 から 100%まで変化させた時の提案方式の性能を、レガシ方式と MLT 方式と比較した。

表 2 ON/OFF フロークライアントの割合を変えた時の端末の平均スループット [Kbps]

方式	0%	20%	40%	60%	80%	100%
Proposed	1807	1835	1841	1761	1685	1545
Legacy	1144	1135	1139	1089	1048	1003
MLT [3 秒周期]	481	447	469	457	388	306
MLT [10 秒周期]	1053	925	920	926	752	665

### 5.2.1 集約スループットと各 AP の負荷の時間変化

図 5 は、各 AP の WNIC で計測されたスループットの時間変化を示している。レガシ方式、及び MLT 方式では、2.4GHz 帯を使用する割合が高い。特にレガシ方式では、RSSI が最大の AP へ接続するため、電波強度の減衰が大きい 5.0GHz 帯は使用されにくいことが分かる。提案方式および MLT 方式は、5.0GHz 帯を利用しようとするが、MLT 方式では全体的にスループットが小さいことが分かる。これは、MLT 方式がチャネルスキューニングのオーバーヘッド、及びマルチレートの影響を考慮していないためであると考えられる。

図 6 は、各方式での集約スループット (全 AP のスループットの合計) を比較している。この結果より、提案方式はレガシ方式と比べ、集約スループットを最大で約 59%向上できたことが分かる。また MLT 方式は、前述した影響により集約スループットがレガシ方式よりも劣るという結果となった。その影響は、トポロジの更新周期を短くする (3 秒) ことで、さらに顕著となった。

表 2 は、端末の平均スループットを示している。この結果より、提案手法は他の手法に比べ平均スループットにつ

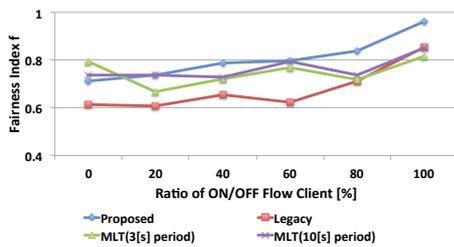


図 7 Jain's Fairness Index

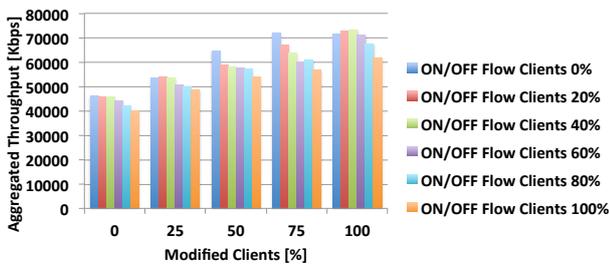


図 8 改良済み端末とレガシ端末が混在する場合の集約スループット

いても向上できることが分かる。

### 5.2.2 Jain's Fairness Index に基づく公平性

公平性の指標として、Jain's Fairness Index [23] を用いた。Fairness Index は次式で表される。

$$f = \frac{(\sum_{i=1}^n x_i)^2}{n \sum_{i=1}^n x_i^2} \quad (1 \leq i \leq n). \quad (3)$$

Jain's Fairness Index は、 $f$  の値が 1 に近いほど公平性が高いことを示す。

この指標の測定結果を図 7 に示す。提案方式は、ON/OFF フロークライアントが 0% の時を除いた全ての場合において、他の方式と比べ同等以上の公平性を得られることが分かる。特にレガシ方式と比較して、最大で約 28% の公平性の向上を達成し、ON/OFF フロークライアントが 100% の場合、 $f = 0.9617$  という高い公平性を得た。MLT 方式は、公平性について考慮されている方式であるため、ON/OFF フロークライアントの割合によらず、比較的良好な公平性を保っている。しかし、MLT 方式では、パケットサイズが固定長であるなど、比較的規則的なトラフィックを対象としており、ON/OFF フロークライアント数の増加による不規則なトラフィックの割合の増加につれ、提案方式との差が大きくなることが分かる。

### 5.3 レガシ端末が混在する場合の評価

提案方式では、端末の無線 LAN の制御ソフトウェアに対する改良を行う必要があるが、実用上全ての端末を改良できるとは限らない。本節では、提案方式を利用する端末（改良済み端末）とレガシ方式を利用する端末（レガシ端末）とが混在する場合に、スループットと公平性がどの程度低下するかについて調査を行った結果について述べる。

本実験では、改良済み端末の割合を 0 から 100% まで変

表 3 改良済み端末の割合を変えた際の端末の平均スループット [Kbps]

ON/OFF	端末	改良済み端末の割合				
		0%	25%	50%	75%	100%
0%	All	1153	1337	1617	1803	1788
	Legacy	1153	1144	1451	1696	-
	Modified	-	1917	1784	1838	1788
20%	All	1148	1350	1473	1673	1824
	Legacy	1148	1137	1188	1315	-
	Modified	-	1988	1758	1793	1824
40%	All	1148	1337	1456	1593	1835
	Legacy	1148	1127	1283	1328	-
	Modified	-	1966	1627	1682	1835
60%	All	1106	1267	1438	1500	1779
	Legacy	1106	1088	1234	1248	-
	Modified	-	1805	1643	1583	1779
80%	All	1053	1245	1428	1525	1689
	Legacy	1053	1075	1187	1219	-
	Modified	-	1758	1669	1627	1689
100%	All	1003	1221	1345	1423	1547
	Legacy	1003	1094	1226	1287	-
	Modified	-	1600	1465	1469	1547

化させて計測を行った。

#### 5.3.1 レガシ端末が混在する場合の集約スループット

図 8 に、各改良済み端末の割合で達成される集約スループットを示す。また、表 3 に端末の平均スループットを示す。表 3 では、レガシ端末と改良済み端末の全てに対する平均スループットを All、レガシ端末のみに対しては Legacy、改良済み端末のみに対しては Modified と示している。端末全体で見た場合、集約スループット、平均スループットともに改良済み端末の割合が 100% に近づくにつれ増加する傾向にあることが分かる。また、改良済み端末はレガシ端末に比べ、平均スループットを向上できることが分かる。これより、改良済み端末はスループットの観点から、改良による恩恵を十分に受けることができると言える。さらに、レガシ端末のみで見た場合でも、改良済み端末の割合が増加するにつれ、平均スループットが増加することが分かる。

#### 5.3.2 レガシ端末が混在する場合の Fairness Index

図 9 に、端末全体で見た場合、レガシ端末のみで見た場合、および改良済み端末のみで見た場合の各々における、端末間のスループットに対する Jain's Fairness Index の値を示す。端末全体で見た場合、改良済み端末の割合が増加するにつれ、端末間の公平性が緩やかに増加する傾向にあることが分かる。また、レガシ端末のみで見た場合でも、改良済み端末の増加に伴い公平性が増加傾向にあることが分かる。

以上の結果より、レガシ端末と改良済み端末が混在する場合でも、どちらか一方の性能を犠牲にすることなく、改良済み端末の割合が増加するにつれ全体の性能を向上させることが分かる。

## 6. おわりに

本稿では、AP 間の動的負荷分散方式を提案した。提案

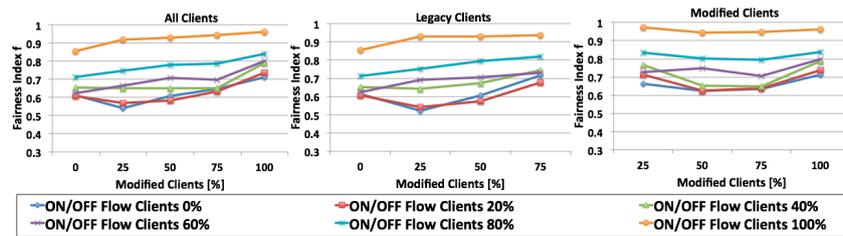


図 9 改良済み端末とレガシ端末が混在する場合の Fairness Index

方式では、仮想化 WNIC による複数 AP への同時接続を可能にすることで、AP 間の切り替えのオーバーヘッドをできるだけ小さくし、その上で、無線 LAN 環境のモニタリング結果に基づき適切なタイミングでトポロジの更新を行い、トラフィックの変化にすばやく追従して AP 間の負荷を均等化する。

ns-3 シミュレータを用いた実験により、提案方式は MLT 方式やレガシ方式と比べ、集約スループットと端末間の公平性について高い性能が得られることを確認した。集約スループットについては、レガシ方式と比べて最大で約 59% 向上でき、Jain's Fairness Index については、レガシ方式と比べて最大で約 28% 向上できた。さらに、改良済み端末とレガシ端末が混在する場合においても、レガシ端末の性能に大きな影響を与えることなく、かつ改良済み端末の増加とともに端末全体の性能を向上できることが分かった。

今後の課題として、提案方式を実端末へ実装し、テストベッド上で評価を行うことが考えられる。

#### 謝辞

本研究の一部は、JSPS 科研費 23700099 の助成のもと行った。

#### 参考文献

[1] Y. Fukuda, T. Abe and Y. Oie "Decentralized Access Point Selection Architecture for Wireless LANS - Deployability and Robustness," *Proc. IEEE VTC*, Vol. 2, pp.1103-1107, 2004.

[2] L-H. Yen and T-T. Yeh, "SNMP-Based Approach to Load Distribution in IEEE802.11 Networks," *Proc. IEEE VTC*, Vol. 3, pp. 1196-1200, 2006.

[3] Y. Bejerano, S-J. Han and L. Li, "Fairness and Load Balancing in Wireless LANS Using Association Control," *Proc. ACM Mobicom*, pp. 315-329, 2004.

[4] Y. Bejerano and S. J. Han, "Cell Breathing Technique for Load Balancing in Wireless LANS," *Proc. IEEE Trans. Mobile Computing*, Vol. 8, No. 6, pp.735-749, 2009.

[5] B. Kauffmann, F. Baccelli, A. Chaintreau, V. Mhatre, K.Papagiannaki and C. Diot, "Measurement-Based Self Organization of Interfering 802.11 Wireless Access Networks," *Proc. IEEE Infocom*, pp. 1451-1459, 2007.

[6] A. P. Jardosh, K.Mittal, K. N. Ramachandran, E. M. Belding and K. C.Almeroth, "IQU: Practical Queue-Based User Association Management for WLANs," *Proc. MobiCom*, pp. 158-169, 2006.

[7] H. Gong, K. Nahm and J. W. Kim, "Distributed Fair Access Point Selection for Multi-Rate IEEE 802.11 WLANs,"

*Proc. CCNC*, pp. 528-532, 2008.

[8] K. Sundaresan and K. Papagiannaki, "The Need for Cross-Layer Information in Access Point Selection Algorithms," *Proc. ACM IMC*, pp. 257-262, 2006.

[9] D. Gong and Y. Yang, "AP Association in 802.11n WLANs with Heterogeneous Clients," *Proc. IEEE Infocom*, pp. 1440-1448, 2012.

[10] A. Mishra, M. Shin and W. A. Arbaugh, "An Empirical Analysis of the IEEE 802.11 MAC Layer Handoff Process," *Proc. ACM Sigcomm Computer Communication Review*, Vol. 33, No.2, pp. 93-102, 2003.

[11] Aruba Networks, "Optimizing Aruba WLANs for Roaming," [http://www.arubanetworks.com/pdf/technology/DG\\_Roaming.pdf](http://www.arubanetworks.com/pdf/technology/DG_Roaming.pdf)

[12] I. Ramani and S. Savage, "SyncScan: Practical Fast Handoff for 802.11 Infrastructure Networks," *Proc. Infocom*, Vol.1, pp. 675-684, 2005.

[13] H. Wu, K. Tan Y. Zhang and Q. Zhang. "Proactive Scan: Fast Handoff with Smart Triggers for 802.11 Wireless LAN," *Proc. Infocom*, pp. 749-757, 2007.

[14] V. Brik, A. Mishra and S. Banerjee, "Eliminating Handoff Latencies in 802.11 WLANs Using Multiple Radios: Applications, Experience, and Evaluation," *Proc. ACM IMC*, pp. 27-32, 2005.

[15] R. Chandra, P. Bahl and P. Bahl, "MultiNet: Connecting to Multiple IEEE802.11 Networks Using a Single Wireless Card," *Proc. IEEE Infocom*, Vol. 2, pp. 882-893, 2004.

[16] S. Kandula, K. Lin, T. Badirkhanli and D. Katabi, "Fatvap: Aggregating AP Backhaul Bandwidth," *Proc. 5th USENIX Symposium on Networked Systems Design and Implementation*, pp. 89-104, 2008.

[17] X. Xing, S. Mishra and X. Liu, "ARBOR: Hang Together Rather than Hang Separately in 802.11 Wifi Networks," *Proc. IEEE Infocom*, pp. 1-9, 2010.

[18] A. J. Nicholson, S. Wolchok. and B. D. Noble, "Juggler: Virtual Networks for Fun and Profit," *Proc. IEEE Trans. Mobile Computing*, Vol. 9, No. 1, pp. 31-43, 2010.

[19] Virtual Wi-Fi, <http://research.microsoft.com/en-us/um/redmond/projects/virtualwifi/>

[20] M. Heusse, F. Rousseau, G. Berger-Sabbatel and A. Duda, "Performance Anomaly of 802.11b," *Proc. IEEE Infocom*, 2003.

[21] Network Simulator ns-3, <http://www.nsnam.org/>

[22] Scenargie, Space-Time Engineering, <http://www.spacetime-eng.com/en/index.html>

[23] D. -M. Chiu and R. Jain, "Analysis of The Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks," *Computer Networks and ISDN Systems* vol. 17, No. 1, pp. 1-14, 1989.