

WAPL のアーキテクチャとハンドオーバーの実現方式

山崎 浩司[†] 小島崇広[‡] 市川 祥平[‡] 竹尾 大輔[‡] 渡邊 晃[†]

名城大学理工学部[†] 名城大学大学院理工学研究科[‡]

1. はじめに

インターネットを利用したサービスの増加、通信端末のモバイル化、日用品化に伴い、場所や時間に制約されず、低コストでブロードバンド環境を提供できる無線 LAN への需要が高まっている。無線 LAN エリアの拡大のためには、アクセスポイント (AP) の整備が不可欠である。しかし、現在 AP 間は有線で接続されているのが一般的であり、AP の設置には多大な時間と費用を要するのが現状である。そこで我々は、AP 間通信を無線化することにより、これらの問題を解決する WAPL (Wireless Access Point Link) を提案している [1][2]。本稿では、WAPL で使用する AP (以下 WAP) のアーキテクチャと、実装についての報告を行う。また、WAPL における端末の移動(ハンドオーバー)の方式を検討したので 報告する。

2. WAPL とは

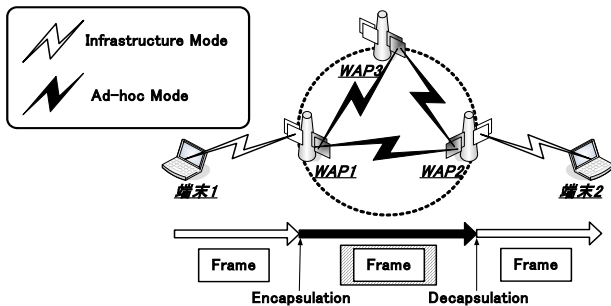


図 1. WAPL の概要

WAPL の概要を図 1 に示す。WAP は無線インタフェースを二つ持つ。一つは、端末とインフラストラクチャモードで通信を行う。もう一つは WAP どうしで、アドホックモードにより通信を行う。WAP 間の通信に必要なルーティングテーブルは、MANET (Mobile Ad-hoc Network) [3]のルーティングプロトコルにより生成される。WAP を適切に設置すれば、WAP 間で通信環境を自動的に生成できるので、無線 LAN エリアを容易に構築することが可能となる。端末は WAPL のサービス範囲内を自由に参加、移動、離脱することができる。端末から送信されたフレームは、最寄の WAP によりカプセル化される。カプセル化されたフレームは、相手端末が属する WAP までマルチホップで転送される。カプセル化されたフレームを受信した WAP は、デカプセル化をして、配下の端末に

フレームを送信する。WAP は、カプセル化を実現するために、リンクテーブルを保持する。このテーブルは、端末の MAC アドレスと、端末が所属する WAP のアドホック側インタフェースの IP アドレスを、対応づけて管理したもので、通信に先立つ ARP をトリガとして必要に応じて生成される。このようにして WAP 間通信はイーサネットをエミュレートする。よって、端末は WAPL の存在を意識せずに、通信を行うことが可能である。

3. リンクテーブルの生成

図 2 にリンクテーブルの生成シーケンスを示す。端末 1 が端末 2 と通信を開始するとき、MAC アドレス解決のために、端末 1 から端末 2 に対して必ず ARP Request を送信する。WAP1 は端末 1 からの ARP Request を受信すると、ブロードキャストアドレスでカプセル化をして他の WAP に転送する。WAP2, WAP3 は ARP Request を受信すると、配下の端末に転送すると同時に、端末 1 と WAP1 の関連を示すリンクテーブルを生成する。WAP2 は、配下に存在する端末 2 からの ARP Reply を受け取ると、リンクテーブルを参照し、WAP1 のアドレスでカプセル化をして WAP1 へ転送する。上記 ARP Reply を受信した WAP1 は、カプセル化を解いて端末 1 に転送すると同時に、端末 2 と WAP2 の関連を示すリンクテーブルを生成する。その後の端末 1 と端末 2 の通信は、生成されたリンクテーブルを参照して行われる。ここで、WAP3 に生成されたリンクテーブルは一定時間以上参照されないため、自動的に消去される。

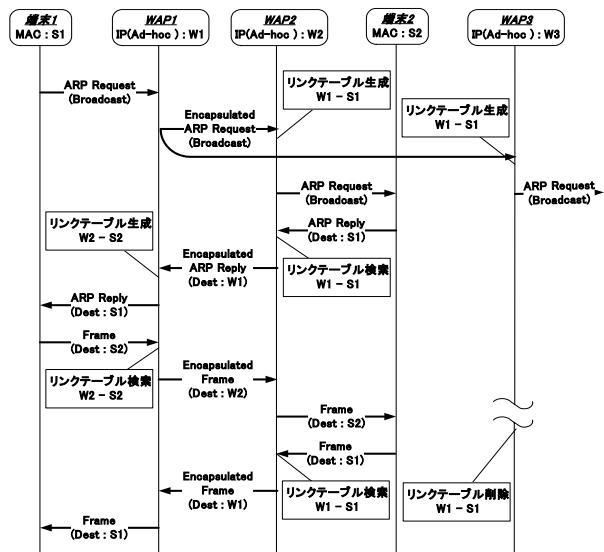


図 2. リンクテーブルの生成シーケンス

Realization of WAPL Architecture and its Handover

[†]Koji Yamazaki and Akira Watanabe

Faculty of Science and Technology, Meijo University

[‡]Takahiro Kojima, Shouhei Ichikawa, and Daisuke Takeo

Grad School of Science and Technology, Meijo University

4. WAPの実装

図3にWAPの実装を示す。WAPは端末に対してAPの機能を提供するAP機能(APF)と、イーサネットをエミュレートするカプセル化機能(CAPF)から構成される。APFはBUFFALO社製のAP(WLA-G54)をそのまま利用した。CAPFはOSにFreeBSD 5.4-RELEASEを適用したノートPC(Epson endeavor NT350)に、カーネルを改造する形で実装した。アドホック側のインタフェースはPCに内蔵された無線インタフェースを利用した。CAPF部のEthernet側のインタフェースは、APから送られてくるパケットを全て処理するために、プロミスクラスモードに設定した。WAPLでは、MANETのルーチングプロトコルには一切手を加えず、システムに合わせて最適なルーチングプロトコルを選択することが可能である。今回の実装では、ルーチングプロトコルにプロアクティブ型のOLSR[4]を採用し、WAPを用いた2ホップ通信の動作確認を行った。

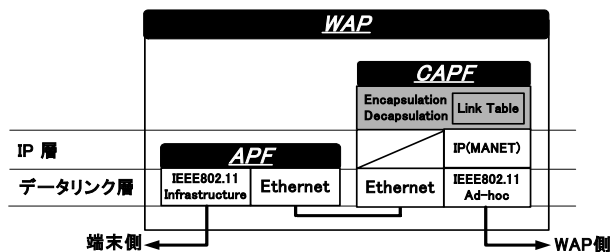


図3. WAPの実装

5. WAPLのハンドオーバー

WAP間通信がEthernetをエミュレートしている点を活かし、WAPLのハンドオーバーについて検討した。図4に通常のEthernetと市販のAPの組み合わせシステムにおいて、端末1が通信中にAP1からAP2へハンドオーバーしたときのシーケンスを示す。APはBUFFALO社製(WLA-G54)を使用し、端末はWindows XPを使用した。端末1はAP1、端末2はAP2とそれぞれインフラストラクチャモードでアソシエートしている。各APのチャンネルと、ESSID(Extended Service Set Identification)は同一とした。また、有線のパケットの解析にはEthereal(Version 0.10.13)、無線のパケットの解析にはAirPacMon(Version 1.00)を使用した。端末1がAP1のサービスエリア内から、AP2のサービスエリア内に移動すると、端末1とAP1のアソシエーションが切断される。無線レイヤで端末1とAP2がアソシエートすると、AP2は端末1が再参加してきたことを他のAPへ伝えるために、LLCとCISCOWL-L2をマルチキャストする。このときハンドオーバーに使用されるパケットは、APのベンダにより多少異なるが、同一ベンダ同士であれば、各AP内のテーブルが即座に書き換えられ、効率的なハンドオーバーが実現される。WAP間はEthernetをエミュレートしているので、WAP配下の端末が他のWAP配下に移動したときには、上記シーケンスがそのままWAP間を流れる。WAP内で使用するAPF部を同一ベンダAPで統一すれば効率的なハンドオーバーが可能となる。また、端末はアソシエートが一度切断されると、保持しているARPキャッシュテーブルをクリアする。そのため、端末1が再参加後に、端末2との通信を再開するために、必ずARP Requestが送信される。このARPを用いてWAPの

リンクテーブルを生成し直すことができる。このようにしてWAPLの環境下においてもハンドオーバーを実現し、通信を継続することが可能となる。

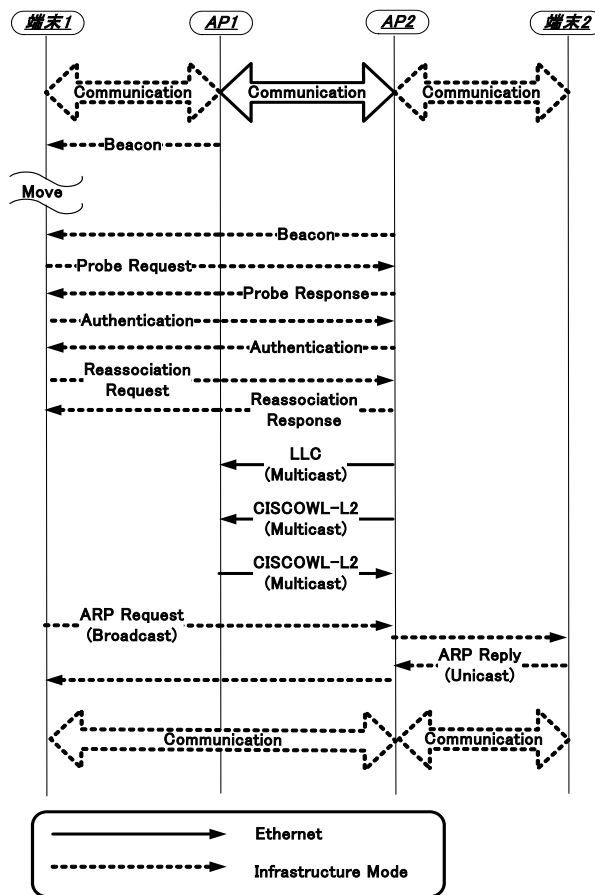


図4. ハンドオーバーのシーケンス

5.むすび

WAPのアーキテクチャと実装方式、およびWAP間のハンドオーバーの実現方式について検討した。今回の実装方式でWAP間でのハンドオーバーは容易に実現できることを確認した。今後は、実機を用いたWAPLの性能評価を行う。

参考文献

- [1] 市川祥平, 渡邊晃, アクセスポイントの無線化を実現するWAPLの方式, DICOMO2005
- [2] 小島崇広, 市川祥平, 渡邊晃, 無線アクセスポイントリンク“WAPL”の立上げ方式, DICOMO2005
- [3] <http://www.ietf.org/html.charters/manet-charter.html>
- [4] T.Clausen P.jacquet, “Optimized Link State Routing Protocol”(OLSR) REC3626 Oct.2003