社団法人　情報処理学会　研究報告
IPSJ SIG Technical Report

2005－MBL－35（6）
2005－ITS －23（6）
2005／11／17

モバイルPANにおける隠れ端末を考慮した動的アドレス割り当てプロトコルの評価

四條雅博[†]　田中希世子[††]　鈴木偉元[††]
石川憲洋[††]　石原進[‡]

筆者らは，移動した先々でユーザの持つ携帯端末から周りに存在する周辺機器のサービス利用を可能とするMobile Personal Area Network（mPAN）を提案している．このmPANを実現するには，携帯端末と周辺機器の通信を可能とするために，各ノードに対してIPアドレスを割り当てネットワークを構築する必要がある．そこで筆者らはmPANにおけるアドレス割り当て方法を提案してきた．mPANの想定環境として携帯端末と周辺機器間の通信を1ホップ通信のみに限定している．このような1ホップ内のアドレス唯一性を提供するアドレス割り当てプロトコルとして，IETF Zeroconf WGによるAutoIPがある．しかし，2ホップ先のノードとのアドレス衝突が起きていた場合，その中間ノードにおいてアドレス衝突を起こしたノード間の区別が付けられないという問題がある．

筆者らの提案してきたアドレス割り当てプロトコルでは，周辺機器が1ホップ内のノードのIPアドレスとMACアドレスの対応を定期的にブロードキャストすることで，周辺機器の周りに存在するノードでは自身から周辺機器を介した2ホップ先の隠れ端末とのアドレス衝突を検出できるようにしている．本稿では，本プロトコルのプロトタイプ実装を行い，実環境における動作確認より正常に動作したことを確認した．

Evaluation of dynamic address assignment protocol considering hidden terminals for mobile PAN

Masahiro Shijo ,[†] Kiyoko Tanaka,[††] Hideharu Suzuki,[††] Norihiro Ishikawa [††]
and Susumu Ishihara[‡]

We have proposed Mobile Personal Area Network (mPAN) that enables mobile device's user to exploit the service from peripheral devices in the moving place. In mPAN, all nodes must be join to a network maintaining their IP addresses to communicate each other. So we have proposed a protocol assigns IP address to nodes in mPAN. In mPAN, communication is limited assumes 1 hop between a mobile device and a peripheral device. AutoIP, an IP address assignment protocol developed in IETF Zeroconf WG, is assuming uniqueness of IP addresses within 1 hop area. However in case that a node causes address collisions with hidden terminals in 2 hop away, the intermediate node between them can not distinguish each nodes.

In our proposed protocol, a node is able to detect address collisions with hidden nodes by receiving 1 hop neighbor address lists from its neighbor nodes. The list contains the binds of a IP address and a MAC address of the neighbor of the sender of the list. We implemented the proposed protocol, and confirmed that it worked properly in a wireless network include hidden terminals.

1. はじめに

近年，Linux OSやSymbian OSを搭載した携帯電話の普及に見られるように，携帯電話等の携帯端末の高機能化が進んでいる．また広域無線通信とは別に，BluetoothやIrDA，ZigBeeといった短距離無線通信技術も進歩してきた．一方で，現在における広域無線はメールの送受信，ホームページの閲覧に，短距離無線通信はPCとのデータ同期，ヘッドセットとの無線接続というようにそれぞれ比較的単純な形態での利用しかされて来なかった．

先に挙げたBluetoothのように，従来のPANはローカルに閉じたネットワークであり，一般にPANを構成する機器やデバイスは固定されたものが想定されていた．しかし，ユビキタスネットワークとして期待されているような，いつでも・どこでも・何にでも接続される環境を実現するためには，ユーザの状況や目的に合わせて目の前にある様々な機器やデバイスを随時動的に，連携・利用可能とすることが不可欠である．ユーザの嗜好や状況に適応させた柔軟なサービスを提供するためには，ネットワーク上に保存されたユーザプロファイル等の情報とローカル通信によって近くに存在する周辺機器から得られる情報とを連携させるため，外部ネットワークアクセスを用いて情報を取得する必要がある．このような背景があり，広域・短距離無線通信技術を融合した新たなサービス創出が期待されている．

† 静岡大学大学院理工学研究科
Graduate School of Science and Engineering, Shizuoka University
†† （株）NTTドコモ
NTT DoCoMo, Inc.
‡ 静岡大学工学部
Faculty of Engineering, Shizuoka University

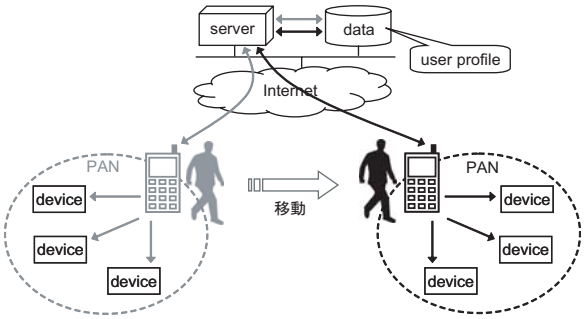
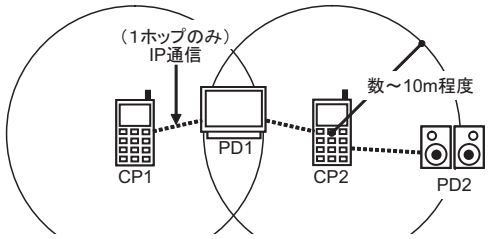


図 1　mPAN の構成



アドレス衝突を認めない	CP1-PD1, CP1-CP2, PD1-CP2, CP2-PD2, PD1-PD2間
アドレス衝突を認める	CP1-PD2間

図 2　mPAN の想定環境

そこで筆者らは広域・短距離無線通信技術を融合した新たな PAN 利用形態として Mobile Personal Area Network (mPAN) を提案している．mPAN とは，携帯端末が周りに存在する周辺機器と PAN を構築し，移動先でもサービス起動，異なる複数の周辺機器を用いることで以前起動していたサービスの継続，再起動等が可能なサービスモビリティを実現するためのコントロールポイントとなるネットワークサービスである．

mPAN を実現するためには，ローカルに存在する各ノード間の通信を可能とするために携帯端末，周辺機器へアドレスを割り当てることで，ネットワークを構築する必要がある．現在のネットワークでは，ノードに対するアドレス割り当てに Dynamic Host Configuration Protocol[1] (DHCP) サーバのようなアドレス割り当てサーバが広く用いられている．mPAN では，DHCP サーバの有無にかかわらず，周辺機器をユーザのニーズに合わせて設置可能であるべきと考える．筆者らは mPAN の環境に適した固定サーバが存在しない mPAN 向けの携帯端末と周辺機器間のアドレス割り当て手法を提案している．本稿では本プロトコルの実装と評価について述べる．

そこで本稿では，これまでに提案してきた mPAN における携帯端末とローカルに存在する周辺機器間のアドレス割り当て方法[9] を紹介し，評価を行う．

以降，第 2 章で mPAN の説明を行い，第 3 章で関連研究，第 4 章でアドレス割り当て方法について述べる．第 5 章でアドレス設定時におけるアドレス衝突の発生確率について述べ，第 6 章で本プロトコルのプロトタイプ実装と動作検証を，第 6 章でまとめを行う．

2. mPAN

2.1 mPAN 概要

mPAN とは，携帯電話等の携帯端末（Control Point: CP）がユーザの周辺に存在する周辺デバイス（Peripheral Device: PD）と PAN を構築し，ユーザが移動した先々でもサービスを起動したり，複数周辺デバイス間で起動したサービスを継続することができるサービスモビリティを実現するためのコントロールポイントとなるネットワークサービスである[2]．PD はリッチな入出力機能を持ち，ユーザの持つ CP からのサービス要求に対し，CP に代わって自身の入出力機能を利用することで，CP に対しサービスを提供する．**図 1** に mPAN の構成例を示す．ユーザの持つ CP がサービスを実行するために必要となる PD の選択や接続の制御を行うことで PAN を構築し，サービスに応じて PAN 内のローカル通信と CP からの外部ネットワークアクセスを利用する．CP がユーザの移動に伴って変化する PD との間で随時 PAN 構築を制御し，その時々の PAN における必要なサービスの起動や，PAN の変化によらないサービスの継続といったサービスモビリティを実現する．

この mPAN では，サービスシナリオによって，次のような 3 つの形態に分けられる．

(1)　**PAN 内ローカル通信**
　　例）ローカルに存在する PD から CP へのダイレクトな情報取得．

(2)　**PAN-外部ネットワーク通信**
　　例）ローカルで取得した情報をトリガとして外部ネットワークにアクセスし，外部サーバとの連携によるサービスの自動実行．

(3)　**複数 PAN 間通信**
　　例）自分の CP が構築した PAN のみでなく，外部ネットワークを介し，家族や友人等の CP が構築した他の PAN との接続による複数 PAN 間連携サービス．

以上の様な mPAN 実現の技術課題の 1 つに，ユーザの持つ携帯端末とローカルに存在するノード間の通信を可能とするための携帯端末，周辺デバイスへのユニークなアドレスの割り当てがある．

2.2 想 定 環 境

mPAN では，以下の環境を想定している（**図 2**）．

- CP，PD はローカル通信を行うための短距離無線インタフェースを保持する．
- 短距離無線を利用した CP と PD の通信可能範囲は数 m ～ 10m 程度とする．
- CP-PD 間通信は 1 ホップ通信に限定する．
- 通信は CP-PD 間でのみ行われ，PD どうしでの通信は行われないものとする．ただし，アドレス衝突処理にあたってはその限りではない．
- PD が関与しない CP-CP 間通信は行われないものと

する．ただし，アドレス衝突処理に当たってはその限りではない．

- アドレス割り当てを行う中央サーバは存在しない．
- CP を介した PAN 外通信，mPAN で用いるアプリケーションの汎用性を考慮し，PAN 内の通信プロトコルに IP を使用する．さらに，現行インターネットとの互換性を考え，IPv4 を用いる．

2.3 目標とする PAN

mPAN における PAN の目標形態，目標特性を以下に示す．

- **隠れ端末があっても，アドレスの衝突がなく，通信を維持可能**

  mPAN では CP-PD 間の 1 ホップ通信のみを扱うため，各ノードは自身の通信可能範囲内でアドレス衝突がなければ隣接ノードとの通信が可能である．しかし，図 2 のように PD1 を介してお互いに直接通信不可能な CP1-CP2 間（隠れ端末間）でアドレス衝突が起きていた場合，PD1 では CP1-CP2 間の区別が付けられない．さらには，隠れ端末間のアドレス衝突により，ARP 汚染やアプリケーションの誤作動を招きかねない．したがって mPAN では，**図 3** のように CP1 から利用することのない 2 ホップ先の隠れ端末（PD2）ともアドレス衝突を回避し PD1 との通信を維持できるべきでり，同時に，mPAN 環境においては自身から 2 ホップ内においてアドレス衝突検出を行えばよいことになる．

- **通信中にノードのアドレスが変わっても通信の復旧が可能**

  ノードが他ノードとのトランスポート層コネクションを確立していた場合など，アドレス衝突によるアドレス変更が行われることで，コネクションが破壊され通信は途絶えてしまう．mPAN ではこのようなアドレス変更前後における通信の継続が可能であることを目指す．

- **アドレス割り当てに伴う CP の電力消費を抑制する**

  CP は携帯電話のような常に起動された端末を想定しており，mPAN を利用することで CP の電力を極端に消費してしまっては CP の使い勝手が悪くなってしまう．したがってアドレス割り当て作業においても CP の電力消費を抑制することを目指す．一方で，本稿において PD は CP に比べて安定した電源供給が得られることを仮定する．

3. 関 連 研 究

これまでに IP ネットワークにおける DHCP 等の中央サーバが不要な，様々な自動設定プロトコルが提案，標準化されてきた．

例えば，Cheshire らの AutoIP[3] や Thomson による IPv6 Stateless Address Autoconfiguration(IPv6 SAA)[5] では，リンク内においてユニークなリンクローカルアドレスの設定が可能となっている．これらプロトコルのアドレス衝突検出方法では，設定したいリンクローカルアドレスに対し AutoIP では ARP パケットを，IPv6 SAA では ICMPv6 パケットをブロードキャストし，それに対する応答があるか否かでアドレス衝突を判断する．しかし，これら ARP パケットや ICMPv6 パケットは直接通信不可能なノードへは到達しないことから，AutoIP，IPv6 SAA では隠れ端末とのアドレス衝突検出を行うことができない．加えて，AutoIP，IPv6 SAA では IEEE EUI-48 MAC アドレスをグローバルユニークであると仮定しているが，実際には重複した MAC アドレスを保持した NIC が出荷されていた例があり，IEEE EUI-48 MAC アドレスをグローバルユニークであると考えることには問題がある．

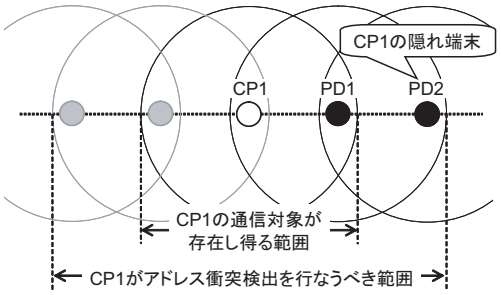


図 3 アドレス衝突検出の範囲

Prakash らの提案する MANETconf[6] では，マルチホップ通信環境を想定したモバイルアドホックネットワーク（MANET[7]）内においてユニークなアドレス設定を可能としている．この MANETconf では既に MANET に参加し，MANET 内で使用済みのアドレスをリストとして保持するノード（initiator）が新たに参加するノード（requester）に対してアドレスを割り当てる．

同様に MANET におけるアドレス割り当てを可能としたプロトコルとして，Weniger の PACMAN[8] がある．PACMAN ではアドホックネットワークで使用されるリンクステートパケット等の情報を利用して，アドレス割り当て・アドレス衝突検出が行なわれる．さらにアドレス変更時において，アドレス変更前の旧アドレスを用いたパケットをアドレス変更後の新アドレスを用いて IP カプセリングすることで，アドレス変更前の通信をアドレス変更後においても維持可能としている．

mPAN では，自身から 2 ホップ先までのアドレス唯一性が確保されればよく，PACMAN や MANETconf に見られるようなマルチホップ通信可能な全てのノードとのアドレス唯一性を確保するプロトコルはアドレス資源の無駄となる．また，アドレス空間をより多くのノードで共有することでアドレス衝突の確率が高まることにもつながる．mPAN では 1 ホップ通信が基本となるため，PACMAN のようなルーティングプロトコルパケットを利用したアドレス割り当て・アドレス衝突検出方法は直接用いることができない．

4. mPAN におけるアドレス割り当て

筆者らはこれまでに文献 9) で，mPAN における隠れ端末を考慮した動的アドレス割り当てプロトコルを提案した．本章では，この携帯端末（CP）と周辺デバイス（PD）間のローカルな範囲における隠れ端末を考慮したアドレス割り当てプロトコルの概要を述べる．

前述のように本稿では IPv4 の利用を前提とする．IPv4 環境においては，AutoIP[3] が RFC 化されており，この AutoIP を拡張することで本アドレス割り当てプロトコルを実現する．

4.1 基 本 方 針

mPAN では，PD は安定した電力供給が見込まれることを想定しているのに対し，CP は携帯電話のような常に起動された端末を想定しておりパワーが限られるという，CP と PD 間の非対称性が存在する．この非対称性を利用し，隠れ端末を含めた 2 ホップ内のノードとのアドレス衝突検出を可能とするために，PD のみ自身から 1 ホップで通信可能なノードの IP アドレス，MAC アドレスの対応表（1 ホップ近隣アドレスリスト）を定期的にブロードキャストさせる．一方で，CP はデータ送信量を抑制するため 1 ホップ近隣アドレスリストのブロードキャストは行わず，周辺の PD と通信するときのみ，定期的に自身の IP アドレスに対する ARP Request をブロードキャストする．したがって CP は PD よりも，最大で 1 ホップ近隣アドレスリスト分だけのデータ送信量が少なくなる．

CP は 1 ホップ近隣アドレスリストをブロードキャストしないことから，CP の隣接ノードは CP を介した 2 ホップ先のノードのアドレスを知ることはできない．CP が自身の 1 ホップ内に存在するノード間のアドレス衝突を検出した場合には，アドレス衝突を起こした片方のノードの MAC アドレスに対してアドレス衝突報告メッセージを送信し，アドレス変更させることで対応する．

CP，PD の各ノードは，隣接ノードからの ARP パケットや隣接する PD から受信した 1 ホップ近隣アドレスリストより取得した情報を用いて，自身から 2 ホップ内のノードのアドレス情報（IP アドレスと MAC アドレスの対応）を 2 ホップ近隣アドレスリストとしてローカルで管理する．アドレス衝突に伴うアドレス変更時には，この 2 ホップ近隣アドレスに含まれないアドレスを選択することで，あらかじめアドレス衝突を回避したアドレス設定を可能とする．

なお，PD による 1 ホップ近隣アドレスリスト，CP による PD との通信の際の ARP Request，各ノードが保持する 2 ホップ近隣アドレスリスト，アドレス衝突報告メッセージは，本プロトコルにおいて AutoIP に対して新たに追加したものである．

4.2 初期アドレス設定

初期のアドレス設定を行う際は，基本的に AutoIP の手順に従う（**図 4**）．

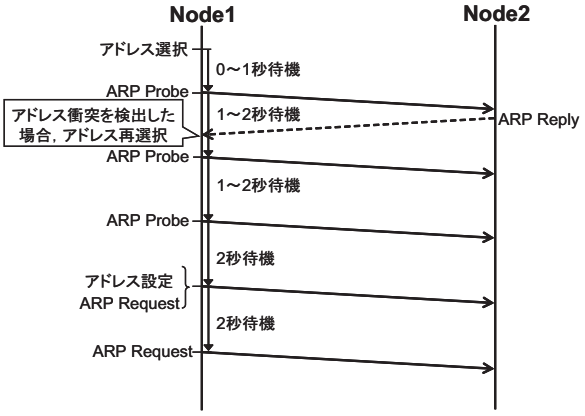


図 4 アドレス設定時シーケンス

まず，169.254/16 アドレス空間からランダムに仮のアドレス（仮アドレス）を選択する．この仮アドレス生成においては，以前使用していたリンクローカルアドレスを優先的に選択する．仮アドレスを選択後，2 ホップ内におけるアドレス唯一性を確認するため，送信元 IP アドレスに未指定アドレス（0.0.0.0）をセットした ARP Request（ARP Probe）を仮アドレス宛にブロードキャストする．ARP Probe 後，これに対する ARP Reply もしくはアドレス衝突報告メッセージを受信しなければ，アドレス衝突は起きていないと判断し，自身の NIC に仮アドレスを設定する．アドレス衝突が検出された場合は，新たな仮アドレスを選択する．

アドレス設定後，自身のアドレスを隣接ノードの ARP テーブルに反映させるため，設定したアドレスに対する ARP Request（ARP Announcement）をブロードキャストする．アドレス衝突報告メッセージに関しては後で説明する．

アドレス衝突に伴うアドレス変更時には，自身の 2 ホップ近隣アドレスリストに含まれていないアドレスを選択することで，あらかじめアドレス衝突を避けたアドレス設定を可能とする．一方，ノード起動直後などの初期アドレス設定時には，自身の 2 ホップ近隣アドレスリストを作成するために隣接ノードからの ARP パケットや隣接 PD からの 1 ホップ近隣アドレスリストの受信を待つことなくアドレス選択を行う．これは，隣接ノードからの ARP パケットや PD からの 1 ホップ近隣アドレスリストを待つのに時間を要することや，アドレス空間の大きさを考えた場合に，2 ホップ内のノードのアドレス情報を取得せずともアドレス衝突を起こす確率は十分に低いためである．初期アドレス設定時におけるアドレス衝突の確率に関しては第 5 章で検証する．

4.3 アドレス衝突検出

アドレス衝突検出は，CP，PD の各ノードにおいて，隣接ノードからの ARP パケットや PD からの 1 ホップ近隣アドレスリストを利用して行う．アドレス衝突は自身と自身から 1 ホップ内の他のノード間，自身の 1 ホップ

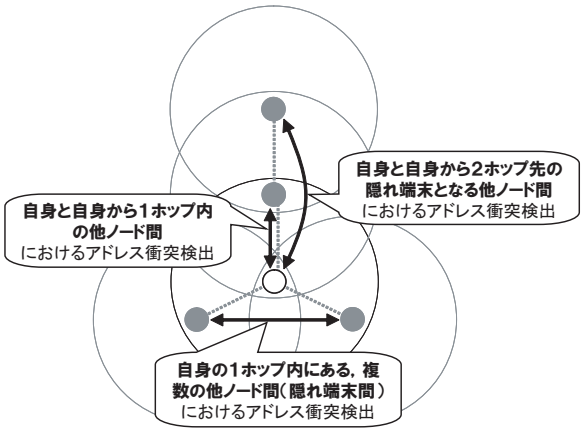


図 5　アドレス衝突検出パターン

内にある，複数の他ノード間，もしくは自身と自身から 2 ホップ先の隠れ端末となる他ノード間（**図 5**）の 3 パターンに分けられる．それぞれのアドレス衝突検出方法を次に示す．

- **自身と自身から 1 ホップ内の他ノード間**

  自身のアドレス情報（IP アドレス，MAC アドレスの対応）と，ARP パケットの送信元アドレス情報もしくは PD からの 1 ホップ近隣アドレスリストに含まれるアドレス情報を比較することでアドレス衝突を検出する．MAC アドレスが異なり，かつ IP アドレスが同じものを検出した場合，自身と 1 ホップ内のノード間でアドレス衝突が起きたと判断する．
- **自身の 1 ホップ内にある，複数の他ノード間**

  自身が保持する 2 ホップ近隣アドレスリストのアドレス情報と，ARP パケットの送信元アドレス情報もしくは PD からの 1 ホップ近隣アドレスリストに含まれるアドレス情報を比較することでアドレス衝突を検出する．MAC アドレスが異なり，かつ IP アドレスが同じものを検出した場合，自身の 1 ホップ内の他ノード間でアドレス衝突が起きたと判断する．
- **自身と自身から 2 ホップ先の隠れ端末となる他ノード間**

  隣接端末からアドレス衝突報告メッセージを受信した場合，自身と自身から 2 ホップ先の隠れ端末となる他ノード間においてアドレス衝突が起きたと判断する．

MAC アドレスが重複していた場合の対応策として，PD は起動時にあらかじめ 48 ビット乱数を生成する．PD はこの 48 ビット乱数を 1 ホップ近隣アドレスリストに付加してブロードキャストする．こうすることで，重複した MAC アドレスを保持した PD が存在した場合にも，それら PD から 1 ホップ近隣アドレスリストを受信したノードにおいて MAC アドレスの衝突や IP アドレスの衝突を検出できるようにしている．複数の PD 間において MAC アドレスが同じで 48 ビット乱数が異なっていた場合，これら PD 間で MAC アドレスの衝突が起きていると判断し，MAC アドレスの代わりに 48 ビット乱数を用いることで，PD の識別を行う．

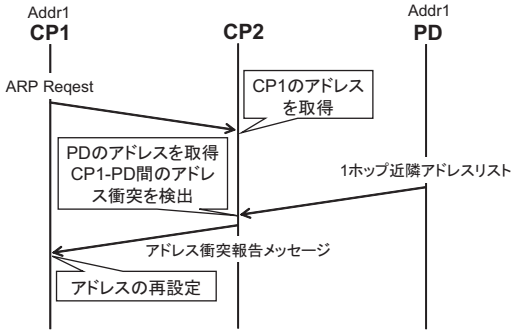


図 6　隠れ端末間アドレス衝突検出時シーケンス例

4.4　アドレス衝突解決

アドレス衝突解決の方法は，アドレス衝突検出同様，3 パターンに分けられる．

- **自身と自身から 1 ホップ内の他ノード間**

  自身と自身の 1 ホップ内の他ノード間でアドレス衝突を検出した場合（図 5），すぐに自身はアドレス変更を行う．この際，自身の 2 ホップ近隣アドレスリストに含まれないアドレスを選択することで，あらかじめ 2 ホップ内で衝突を回避したアドレス選択が可能となる．
- **自身の 1 ホップ内にある，複数の他ノード間**

  自身の 1 ホップ内にある複数の他ノード間のアドレス衝突を検出した場合（図 5），アドレス衝突を起こした片方のノードの MAC アドレスに対してアドレス衝突報告メッセージをデータ・リンク層で送信する．このとき，PD は複数 CP に接続された状態が考えられることから，優先的に CP に対してアドレス衝突報告メッセージを送信し，アドレス変更させる（**図 6**）．
- **自身と自身から 2 ホップ先の隠れ端末となる他ノード間**

  自身と自身から 2 ホップ先の隠れ端末となる他ノード間のアドレス衝突を検出した場合，つまりデータ・リンク層においてアドレス衝突報告メッセージを受信した場合，すぐに自身のアドレス変更を行う．

なお，ノードが他ノードと通信している際にアドレス衝突に伴うアドレス変更が行われた場合，アドレスが変わることで通信が途絶えてしまう．これを避けるため PAC-MAN 同様に，アドレス変更前の旧アドレスを用いたパケットを変更後の新アドレスを用いて IP カプセリングすることで，アドレス変更前の通信をアドレス変更後においても維持可能とする．

4.5　リンク状態管理

CP，PD の各ノードはそれぞれローカルで保持する 2 ホップ近隣アドレスリストの各アドレス情報に対して有効期限を設定する．受信した ARP パケットや 1 ホップ近隣アドレスリストからノードの存在を検出するごとに，

各アドレス情報の有効期限を更新させる．これにより，2 ホップ近隣アドレスリストの情報を最新の状態に保つ．

5. アドレス衝突の発生確率

本プロトコルでは，アドレス変更時におけるアドレス選択は自身の 2 ホップ近隣アドレスリストに含まれていないアドレスを選択することで，あらかじめアドレス衝突を回避したアドレス選択が可能となっている．しかし，ノード起動直後等の初期アドレスの設定時には 169.254/16 アドレス空間からランダムにアドレスを選択する．そこで，mPAN での初期アドレス選択時におけるアドレス衝突の発生確率を検証する．

本プロトコルでは AutoIP と同じアドレス空間を用いている．この 169.254/16 アドレス空間では，169.254.0.x , 169.254.255.x のアドレスは IANA により別の使用目的で確保されている．したがって，本プロトコルで使用できるアドレスは 169.254.1.0 〜 169.254.254.255 の 65024 個のアドレスとなる．

5.1 初期アドレス設定時における 2 ホップ内でのアドレス衝突発生確率

ノードが密度 N_d $[/m^2]$ で 2 次元平面上に一様に分布しており全てのノードが既にアドレスを保持している場合を想定する．この環境において，新たなノード（新規ノード）が，初期アドレスを設定した際に自身の 2 ホップ内においてアドレス衝突を起こす確率を考える．アドレス数を A_n，各ノードの通信可能範囲を半径 R とすると，新規ノードの 2 ホップ内に存在するノード数（N）は $N = 4\pi N_d R^2$ となり，それぞれのノードが互いに 2 ホップ内に存在するとは限らないことから，新規ノードの 2 ホップ内では $4\pi N_d R^2$ 以下のアドレスが既に使用されていることになる．また，新規ノードから $2R$ 内に存在するノードまで必ずしも 2 ホップで到達可能とは限らない．以上から，新規ノードが初期アドレス設定時にアドレス衝突を起こす確率 $P_1(N_d, R)$ は

$$P_1(N_d, R) \leq P_1'(N_d, R) = \frac{4\pi N_d R^2}{A_n} \qquad (1)$$

となる．

（1）より，ノード密度 $N_d = 0.1$ $[/m^2]$，通信可能半径 $R = 10$ [m] のとき，新規ノードがアドレス衝突を起こす確率 P_1 は $P_1(1.0, 10) \leq P_1'(0.1, 10) = 0.00193258$（0.2% 以下）となる．つまり，2 ホップ内に 125 台のノードが存在した場合のアドレス衝突の発生確率が 0.2% 以下ということになる．

5.2 初期アドレス設定時における隠れ端末とのアドレス衝突発生確率

次に，初期アドレス設定時における隠れ端末とのアドレス衝突の発生確率について計算する．新規ノードの 1 ホップ内のノード数（N_1）は $N_1 = \pi N_d R^2$，2 ホップ先に存在するノード数（N_2）は $N_2 = N - N_1 = 3\pi N_d R^2$ となる．ここで，新規ノードの 2 ホップ先に存在するノードどうしで互いに 2 ホップ内に存在するとは限らないことから，新規ノードの 2 ホップ先では $3\pi N_d R^2$ 個以下のアドレスが既に使用されていることになる．また，新規ノードから $2R$ 内に存在するノードが必ずしも 2 ホップで到達可能とは限らない．以上から，新規ノードが初期アドレス設定時において隠れ端末とアドレス衝突を起こす確率 $P_2(N_d, R)$ は

$$P_2(N_d, R) \leq P_2'(N_d, R) = \frac{3\pi N_d R^2}{A_n} \qquad (2)$$

となる．

（2）より，ノード密度 $N_d = 0.1$ $[/m^2]$，通信可能半径 $R = 10$ [m] のとき，アドレス衝突の発生確率 P_2 は $P_2(0.1, 10) \leq P_2'(0.1, 10) = 0.00144943$（0.15% 以下）となる．また，(1)（2）を比較すると新規ノードの 2 ホップ内におけるアドレス衝突において，その 3/4 近くが 2 ホップ先の隠れ端末とのアドレス衝突により発生することが分かる．したがって，本プロトコルは，隠れ端末とのアドレス衝突検出を可能とした点において有効であるといえる．$P_1'(N_d, R)$，$P_2'(N_d, R)$ の変化を**図 7** に示す．

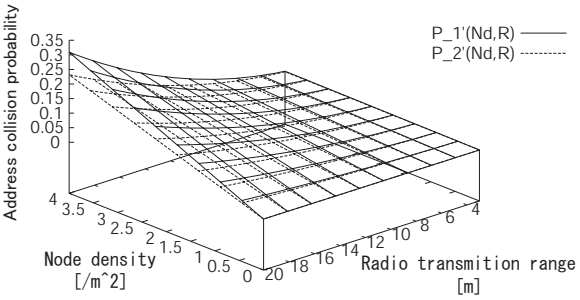


図 7 初期アドレス設定時のアドレス衝突発生確率

6. 実　　装

6.1 実 装 概 要

IETF Zeroconf WG で公開されている AutoIP のプログラム（Simple IPv4 LL[10]）を元にして，ノート PC（Linux kernel ver. 2.4.27）上にプロトタイプの実装を行った．CP による定期的な ARP Request , PD による定期的な 1 ホップ近隣アドレスリスト，各ノードにおけるアドレス検出時のアドレス衝突報告メッセージをデータ・リンク層の RAW パケットで送受信できるようにした．また，CP , PD の各ノードで 2 ホップ近隣アドレスリストを保持させ，アドレス設定時はこの 2 ホップ近隣アドレスリストに含まれていないアドレスを選択させるようにした．

本プロトコルでは AutoIP と同じアドレス空間を利用している．したがって AutoIP ネットワーク中で本プロトコルの動作するノードを起動させた場合アドレス衝突が発生し得る．AutoIP では ARP パケットを用いたアドレス衝突検出のみサポートしていることから，AutoIP の動作するノードと本プロトコルの動作するノード間では 1 ホッ

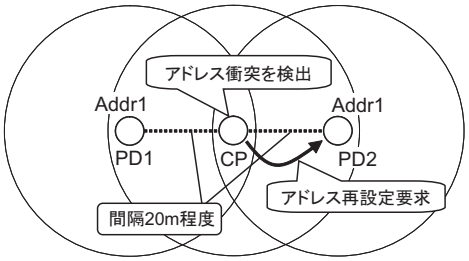


図 8 隠れ端末間アドレス衝突の評価環境

プ内のアドレス衝突検出のみ機能する．本プロトコルにおいて新たに作成した 1 ホップ近隣アドレスリスト，アドレス衝突報告メッセージは，データ・リンク層のプロトコルを ARP とは別に新たに作成したため，AutoIP ノードではこれらメッセージは破棄され AutoIP ネットワークの動作に悪影響を及ぼすことはない．

6.2 隠れ端末間アドレス衝突解決時間の評価

実装したプロトタイプを用い，実環境における隠れ端末間のアドレス衝突解決に要する時間を計測した．

・環　境

評価実験は屋内で行った．CP，および PD としてノート PC を計 3 台使用した．そのうち 2 台を PD として，1 台を CP として動作させた．MAC プロトコルには IEEE 802.11b を用いた．以下にノート PC のスペックと無線 LAN の設定を示す．

- ノート PC スペック
  - IBM Think Pad X30 （1 台）
    - CPU: Mobile Pentium III 1.066GHz
    - Memory: 512MB
    - OS: Linux kernel ver. 2.4.27
  - Panasonic Let’s note R3 （2 台）
    - CPU: Pentium M 1.10GHz
    - Memory: 768MB
    - OS: Linux kernel ver. 2.4.27
- 無線 LAN 設定
  - データ・レート: 11 Mbps
  - 電波強度: 1 mW
  - RTS 閾値: 2312
  - RTS 再送信制限: 16

ノードの配置にあたって，屋内で ping により通信可能範囲を測り，**図 8** のように，PD1-CP 間，CP-PD2 間は直接通信可能で PD1-PD2 間は直接通信不可能な，隠れ端末の存在するトポロジを作成した．これにより，各ノード間の距離は約 20 [m] 程度となった．

・パラメータ

PD による 1 ホップ近隣アドレスリストのブロードキャスト間隔，CP による周辺 PD と通信時における ARP Request のブロードキャスト間隔をそれぞれ 5 [sec] とした．

・方　法

図 8 にノードの配置を示す．まず，ノード 3 台のシステムクロックを同期させておき，図 8 のように配置した．この PD1，PD2 をあらかじめ同じ IP アドレスを設定した状態で起動させておき，その後 PD1-PD2 の中間で CP を起動することで，CP は PD1，PD2 からそれぞれ 1 ホップ近隣アドレスリストを受信し，PD1-PD2 間のアドレス衝突を検出する（**図 9**）．

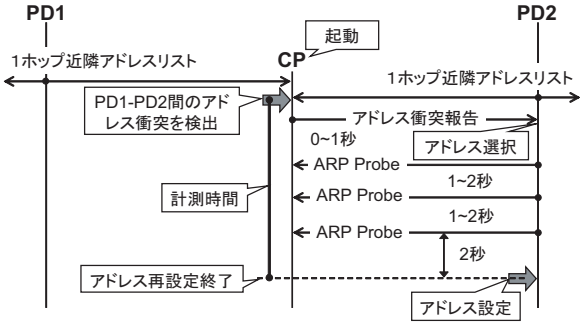


図 9 本プロトコルにおける隠れ端末間アドレス衝突解決時のシーケンス例

CP は PD1-PD2 間のアドレス衝突を検出すると，CP に対して優先的にアドレス衝突報告メッセージを送信し，アドレス再設定を行わせるが，今回の評価環境ではどちらの隠れ端末も PD となることから，どちらか片方の PD に対してアドレス衝突報告メッセージを送信しアドレス設定を行わせることになる．これにより，アドレス衝突報告メッセージを受信した PD がアドレスの再設定を行うことで，アドレス衝突が解決される．この条件下で，CP が 2 台の PD を認識しアドレス衝突を検出した時刻から，PD1-PD2 間のアドレス衝突が解決するまで，つまりアドレス再設定が完了するまでの時刻を測定した．この間の時間をアドレス衝突解決時間と呼ぶことにする．

・結果と考察

実験の結果，本プロトコルを用いた場合，**図 10** のようにアドレス衝突解決に最短 4.93 秒から最長 7.41 秒，平均 5.96 秒を要した．本プロトコルのアドレス設定においては AutoIP 同様，アドレス選択後に 0 〜 1 秒，3 度の ARP Probe 間隔を 1 〜 2 秒，最後の ARP Probe からアドレス設定までに 2 秒という，計 4 〜 7 秒の待機時間が設けられているため（図 9），この待機時間がアドレス衝突解決時間の大半を占めている．

AutoIP のアドレス設定時における各待ち時間は，複数ノードの同時送信によるパケット衝突を避けるためのランダム待機時間や，Ethernet 等の有線環境も想定した ARP の往復遅延時間から決められている．一方，mPAN では通信範囲を数 m 〜 10m 程度とした周辺のノード数がそれほど多くない環境や無線環境を想定していることから，アドレス設定時の各待機時間を AutoIP で想定されるよりも小さく設定することができる．

無線 LAN 1 ホップでの待機時間の見積りをとるため，IEEE 802.11b を利用し，屋内と屋外において ping により 1 ホップの往復遅延時間を計測した．計測には実験で

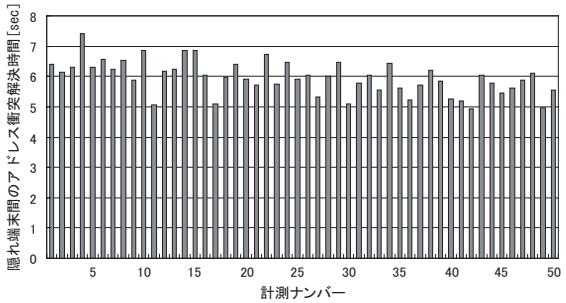


図 10 隠れ端末間アドレス衝突解決時間

用いたノート PC（Panasonic Let's note R3）を 2 台利用し，実験時と同様の無線 LAN の設定において行った．屋内では 2 台のノードの距離を約 20 [m] 程度離して設置した．片方のノードから ping により 206 個の要求パケットを送信させた結果，往復遅延時間は最小 20.077 [ms]，平均 23.474 [ms]，最大 68.561 [ms]，標準偏差 8.161 [ms] となった．屋外では 2 台のノード間隔を約 100 [m] 程度とし，ping により 201 個の要求パケットを送信させた結果，往復遅延時間は最小 20.372 [ms]，平均 28.411[ms]，最大 192.943 [ms]，標準偏差 20.756 [ms] となった．以上より，IEEE 802.11b を用いた環境では 1 ホップの往復遅延時間を最大 200 [ms] とする．

AutoIP では往復遅延時間を最大 1 秒とした場合，ARP Probe 前における他ノードとのパケット衝突回避のための待機時間も同様に最大 1 秒のランダム時間としている．これに習い，仮に mPAN 環境で IEEE 802.11b を用い，ARP の往復遅延時間を最大 200 [ms] とした場合，アドレス選択後に 0 〜 200 [ms]，3 度の ARP Probe 間隔を 200 〜 400 [ms]，最後の ARP Probe からアドレス設定までに 400 [ms] という待機時間を設定できる．その結果アドレス設定に要する時間を最大 1.4 [s] まで短縮できると考えられる．

7. ま と め

本稿では mPAN における隠れ端末を考慮したアドレス割り当てプロトコルを実装し，本プロトコルの検証を行った．

mPAN では，各ノードにおいて隣接ノードおよび 2 ホップ先のノードとのアドレス唯一性が確保できればよく，本プロトコルでは PD に直接通信可能なノードのアドレスをブロードキャストさせることで，CP，PD は 2 ホップ内の割り当て済みアドレスを取得する．この割り当て済みアドレスの情報を利用することで，各ノードはアドレス変更時あらかじめアドレス衝突を回避したアドレス選択をできるようにしている．

ノード密度が 0.1[/m^2] のようなある程度高い環境下では，初期アドレス設定時において隠れ端末と 0.15% 近くの確率でアドレス衝突を起こし得るが，本プロトコルでは隠れ端末とのアドレス衝突を検出し，これを回避できる．

実環境において，プロトタイプ実装により隠れ端末間のアドレス衝突解決時間を計測したところ，その所要時間はアドレスの衝突判定に要する時間が支配的であることが確かめられた．AutoIP のアドレス設定時における待ち時間は，複数ノード間のパケット衝突を避けるためのランダム待機時間や，有線環境も想定した ARP の往復遅延時間から決められている．一方，mPAN では周辺のノード数がそれほど多くない環境や，無線環境を想定していることから ARP の往復遅延時間を含めた待機時間を AutoIP で想定されるよりも小さく見積ることができる．筆者らの測定結果に従い mPAN 環境における ARP の往復遅延時間を 200 [ms] 以下とみなした場合，アドレス設定に要する時間を最大 1.4 [s] まで短縮することが可能であると考えられる．

今後の課題として，アドレス変更時におけるアプリケーションとの連携を考えた支援ソフトウェアの設計実装，アドレス変更時におけるデバイス選択方法との連携を考えた支援ソフトウェアの設計実装等が挙げられる．

参 考 文 献

1) R. Droms: "Dynamic Host Configuration Protocol," *Request for Coments 2131* (1997).
2) 田中 希世子, 鈴木 偉元, 石川 憲洋, 安木 成比古, 石原 進, 峰野 博史, 佐藤 文明, 水野 忠則: "モバイルパーソナルエリアネットワークの提案," 情報学ワークショップ 2004 論文集, pp. 241–245 (2004).
3) S. Cheshire, B. Aboba and E. Guttman: "Dynamic Configuration of IPv4 Link-Lokal Addresses," *Request for Coments 3927* (2005).
4) IETF Zeroconf Werking Group: http://www.ietf.org/html.charters/OLD/zeroconf-charter.html.
5) S. Thomson: "IPv6 Stateless Address Autoconfiguration," *Request for Coments 2462* (2004).
6) S. Nesargi and R. Prakash: "MANETconf: Configuration of Hosts in a Mobile Ad Hoc Network," Proc. IEEE INFOCOM (INFOCOM 2002) (2002).
7) S. Corson and J. Macker: "Mobile Ad hoc Networking(MANET): Routing Protocol Performance Issues and Evaluation Considerations," *Request for Comments 2501* (1999).
8) K. Weniger: "PACMAN: Passive Autoconfiguration for Mobile Ad hoc Networks," IEEE JSAC, Special Issue on Wireless Ad Hoc Networks (2005).
9) 四條 雅博, 田中 希世子, 鈴木 偉元, 石川 憲洋, 石原 進: "モバイル PAN における隠れ端末を考慮した動的アドレス割り当てプロトコル," マルチメディア・分散・協調とモバイル（DICOMO 2005） シンポジウム論文集, pp. 641–644 (2005).
10) Simple IPv4 LL: http://www.zeroconf.org/AVH-IPv4LL.c