

IPv6におけるMobile PPCの実現と評価

083430024 寺澤 圭史

渡邊研究室

1. はじめに

モバイル端末や公衆無線環境の普及に伴い、移動しながら通信を行いたいという要求が高まっている。しかし、IP ネットワークでは、通信中に端末がネットワークを移動することにより IP アドレスが変化すると通信が切断してしまう。この課題を解決するための機能を移動透過性と呼び、我々はこれまでエンドエンドで移動透過性を実現する通信プロトコルとして Mobile PPC (Mobile Peer to Peer Communication)[1] を提案し、IPv4 での実装を完了している。

一方、IPv4 グローバルアドレスの枯渇により今後 IPv6 の普及が必須となる。移動透過性を IPv6 ネットワークで実現する場合においても、エンドエンド通信が維持できることが望ましい。そこで本稿では Mobile PPC の原理と特徴を活かしたまま、IPv6 ネットワークでも移動透過性を可能とする Mobile PPCv6 (以後、MPPCv6) の実装と性能測定を行ったので報告する。MPPCv6 の実現にあたり、特にハンドオーバー処理を工夫し、他方式に比べ高速な切替を可能とした。

ハンドオーバー処理は、L2 ハンドオーバー、移動検知、アドレス生成、重複アドレスチェック、L3 ハンドオーバー、通信再開に分かれており、本稿では移動検知と重複アドレスチェックについて詳細な検討を行った。移動検知とは、ネットワーク層とデータリンク層の機能がそれぞれ独立しているため、L2 ハンドオーバーを上位層でどのように検知するかという方法である。本稿ではアプリケーションからデータリンク層の状態を常時監視する方式を用いることとした。重複アドレスチェック (DAD; Duplicated Address Detection) は取得した IP アドレスが他の端末と重複していないことを確認する機能で、一般にタイマにより検出するため時間がかかる。これを解決するため、本稿ではカーネルを改造してよりタイマに頼らない方式を考案した。それに伴い MPPC のネゴシエーションも一部改善を行った。

2. Mobile PPCv6 の概要

本稿で用いる記号を以下のように定義する。

- A; 端末の IP アドレス
- $A \rightarrow B$; A から B への通信
- $A \leftrightarrow B$; A と B 間の通信
- $A \Leftarrow B$; A から B, または B から A へのアドレス変換

Mobile PPC はエンド端末だけで移動透過性を実現する通信プロトコルである。以下に示す Mobile PPC の原理は、IPv4, IPv6 どちらにも適用できる。通信開始時における通信相手の IP アドレスは DDNS (Dynamic Domain Name System) を使用するして解決できることを前提とする。

IP ネットワークでは通信識別子として IP アドレス/ポート番号が用いられる。そのため、移動して IP アドレスが変化すると、アプリケーションに対してアドレスの変化を隠す必要がある。MIPv6 ではカーネルでのトンネルやモビリティヘッダを用いてこれを実現していたが、MPPC ではカーネル内に作成したアドレス変換テーブルを用いて IP アドレスの変化を隠す。

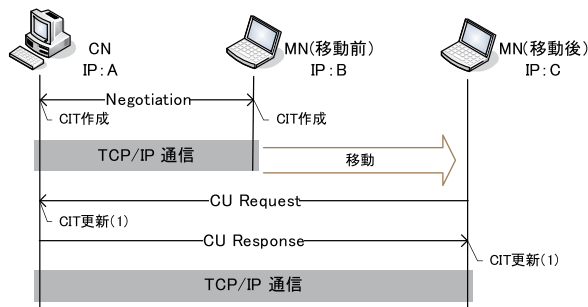


図 1: Mobile PPC のシーケンス

図 1 に MobilePPC のシーケンスを示す。通信開始に先立ち、Diffie-Hellman (以下 DH) 鍵交換を用いて共通鍵を共有する [2]。これは移動通知の相手認証のために用いられる。ネゴシエーション完了後、MN と CN には CIT (Connection ID Table) が IP 層に生成される。この時 CIT には通信開始時 (移動前) のコネクション識別子 CID (Connection ID) が記録されている。通信中に MN がネットワークを移動すると、新しい IP アドレス C を取得し、CU (CIT Update) ネゴシエーションを開始する。CU ネゴシエーションでは移動後の CID を直接通知し合うことでお互いの CIT に移動後の CID を追加する処理を行う。まず、MN は移動後の IP アドレス C を通知するために CU Request を CN に送信する。CU Request には共通鍵より生成した MAC (Message Authentication Code) が付加されており、盗聴やハイジャックを防ぐことができる。CN は CU Request の内容を認証後、自らの CIT を

$$CIT: A \leftrightarrow \{B \Leftrightarrow C\} \quad (1)$$

のように更新する。次に、CN は MN に対して CU Response を送信する。MN は CU Response を認証後、(1) と同様に自らの CIT を更新する。

CU ネゴシエーション完了後の通信パケットはすべて CIT に従ってアドレス変換を行うことにより、移動透過性を実現する。

3. Mobile PPCv6 の実現

3.1 移動検知機能

図 2 のシーケンスは L3 ハンドオーバーに相当する。L3 のハンドオーバーが行われる前には必ず L2 のハンドオーバーが発生する。L2 と L3 は一般に独立しており、L3 で L2 のハンドオーバーが発生したかどうかを知ることはできない。そのため、従来技術では L2 ハンドオーバーが完了した後にルータから定期的送信される RA に依存していた。RA (Router Advertisement) の送信間隔はルータの標準設定では 3~10 秒のランダム値であるため、L2 ハンドオーバー完了後から数秒の遅延が必ず発生する。そこで、データリンク層とネットワーク層の機能制御、すなわち、L2 ハンドオーバーの発生をネットワーク層に通知する機構が必須である。MPPCv6 ではアプリケーションからデータリンク層の状態を監視し、

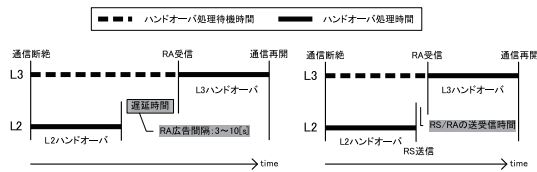


図 2: L2 ハンドオーバーと L3 ハンドオーバー間の流れ (左:移動検知なし, 右:移動検知あり)

L2 ハンドオーバーの発生をネットワーク層に伝えることでネットワーク層の機能を制御する。

図 2 に L2 と L3 のハンドオーバーの流れを示す。図 2 の左は移動検知機能がない場合であるが、L2 ハンドオーバーと L3 ハンドオーバーの間に大きな遅延時間が発生する。図 2 の右は、L3 ハンドオーバーのトリガとなる RA を端末側から要求することで遅延時間をなくす。すなわち、L2 ハンドオーバーの完了を短い間隔で監視し、L3 ハンドオーバーにトリガを与える。具体的には、ルータからの RA を要請するための RS (Router Solicitation) を送信する。この機能によってルータから広告される RA の送信間隔に依存することなく高速な移動検知を可能とした。

3.2 EDAD

端末が移動した後、DAD 処理が完了するまでは通信ができない状態となるため、ハンドオーバーに大きな影響を与える。DAD 処理ではアドレス生成後、NS (Neighbor Solicitation) パケットに生成した IP アドレスを記述してすべての近隣端末にマルチキャストする。一定時間内に NA (Neighbor Advertisement) の応答がなければ、アドレスは重複していないと判断する。タイマは一般に 1 秒であり、この間に通信はできない。その後、端末からルータに対して NA パケットを送信することで、IP アドレスと MAC アドレスの対応関係を示す NC (Neighbor Cache) のエントリを追加する。このエントリが作成されることによりルータから端末へのパケットを転送することが可能になる。

DAD による遅延時間を短縮するために、MPPCv6 では EDAD (Escaping DAD) を考案した。具体的に制御を行うのは、カーネル内の DAD 完了を示すフラグと、NA パケットのフラグである。MN は新しい IP アドレスを生成したあと、ただちに DAD 完了を示すフラグを立てる。この制御により端末は DAD が完了したと認識し、ルータに素早く NA を送信することが可能となる。次に、NA パケットには、ルータに既存の NC エントリ存在した場合に上書きするかを選択できるフラグが定義されており、上書きしないことを選択する。この NA をルータへ送信することで、ルータに既に NC エントリが存在しなかった場合にのみ新エントリを作成させることができる。

以上のように、EDAD では端末の DAD 処理に対して独自の制御を追加することで、処理時間を最小限に抑えることを可能にする。アドレスが重複した場合でも、他端末に影響を与えることはない。

3.3 CU ネゴシエーションの拡張

DAD 回避方式の実装に伴い CU ネゴシエーションの拡張も行った。既存の MPPC では、CU Request を CN が受け取るとただちに CIT を更新してしまう。MN のアドレスが他端末 ON (Other Node) のアドレスと重複していた場合、MN 宛のパケットがすべて ON に届けられることになる。この状態で CN がパケットを送信すると、ON は予期しないパケットを受け取ることになり、ON の動作や通信を妨害する可能性がある。この課題を解決するために CU

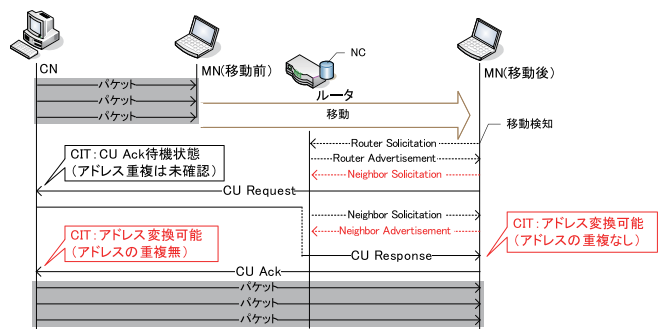


図 3: 拡張した CU ネゴシエーション。

Ack メッセージを新たに定義して、CU ネゴシエーションを 3WAY に拡張する。

図 3 に拡張した CU ネゴシエーションのシーケンスを示す。移動した MN が CU Request パケットを送信し、CN はそれを受信すると通常通りに CIT を更新するが、この時 CIT の状態を CU Ack 待機状態に遷移させる。CN は CU Response パケットを MN に送信し、MN は CIT を更新する。最後に、MN は CN に CU Ack パケットを送信し、それを受けた CN は CIT をアドレス変換可能な状態に遷移させる。もし、CU Ack が MN から応答されない場合は MN のアドレスが重複したものと判断し、パケットを送信しない。以上のように CIT の状態遷移のタイミングをずらすことにより、EDAD を用いても他の端末への影響はない。

4. 実装および性能評価

本稿で述べた MPPCv6 および追加機能を FreeBSD7.0 に実装を行い、ハンドオーバー実験を行った。カーネルの IPv6 スタックに MPPCv6 メインモジュールの実装を行い、その中に EDAD を制御する機能を追加した。アプリケーションには移動検知などのモジュールの実装を行った。

実験環境には、3 台のルータと 2 台の無線 AP を用いて IPv6 無線ネットワークを構築してハンドオーバーを行った。AP と MN 間の接続では無線規格は 802.11a を用いた。トラフィック生成ツールである iperf で通信を行い、通信中に AP を切り替える。ハンドオーバーは予めワンタッチで無線の切替えるスクリプトを作成しておいた。

上記実験環境においてハンドオーバーの処理時間を測定したところ、移動検知および EDAD の効果で、ハンドオーバー時間が約 6~7 秒から平均 3 秒に短縮された。

5. まとめ

本稿では IPv4 で有効性が示されていた MPPC を IPv6 に対応させた MPPCv6 の実現について報告した。IPv6 における移動透過性の課題となる移動検知や DAD 処理に独自機能を実装することにより対応した。

今回の実装で IPv4 と IPv6 の両者での Mobile PPC が実現されたため、今後は IPv4 と IPv6 が混在するネットワーク環境でも対応可能な Mobile PPC の検討と実装を行っていきたい。

[1] 竹内元規, 鈴木秀和, 渡邊 晃: エンドエンドで移動透過性を実現する Mobile PPC の提案と実装, 情報処理学会論文誌, Vol. 47, No. 12, pp. 3244?3257 (2006).

[2] 瀬下正樹, 渡邊 晃: Mobile PPC における認証方式の実装, DICOMO2006 シンポジウム論文集, Vol. 2006, No. 6, pp. 809?812 (2006).

IPv6におけるMobile PPCの 実現と評価

名城大学

寺澤 圭史 鈴木 秀和 渡邊 晃

研究背景

▶ 移動透過性

- 通信中に端末がネットワークを跨る移動を行うと、通信が切断してしまう課題を解決.

▶ IPv6ネットワーク

- IPv4アドレスの枯渇
- 現在, IPv6の普及が始まっている
- IPv6ネットワークではエンドエンド通信が望ましい

IPv6ネットワークにおいて
エンドツーエンドの移動透過性を実現する

移動透過性

- ▶ 通信中のIPアドレスが変化することは想定外
 - IPアドレスが変化する端末に通信を開始できない
 - IPアドレスの変化により上位層が通信を識別できなくなる
 - 移動端末にパケットが正しくルーティングできない
- ▶ IPv4
 - Mobile IP, **Mobile PPC**
- ▶ IPv6
 - Mobile IPv6, MAT, LIN6

Mobile PPCでIPv6の移動透過性を実現

用語定義

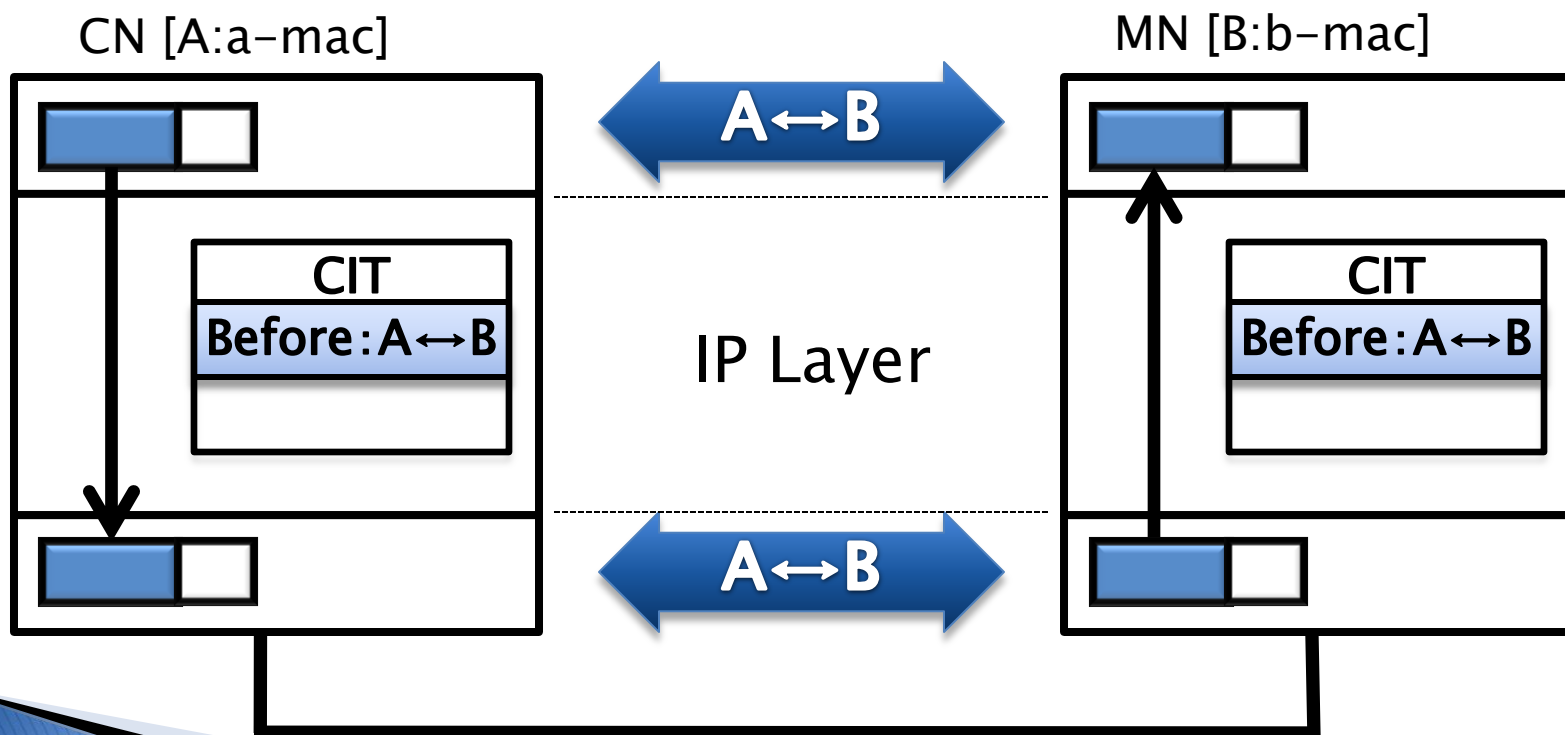
▶ 以後, 下記のような表記で示す

- Node[A:a-mac] 端末[IPアドレス:MACアドレス]
- $X \leftrightarrow Y$ XとYの通信
- $M \Rightarrow N$ 状態やIPアドレスの変化

- RS Router Solicitation (ルータ広告要求)
- RA Router Advertisement (ルータ広告)
- NS Neighbor Solicitation (近隣探索)
- NA Neighbor Advertisement (近隣応答)

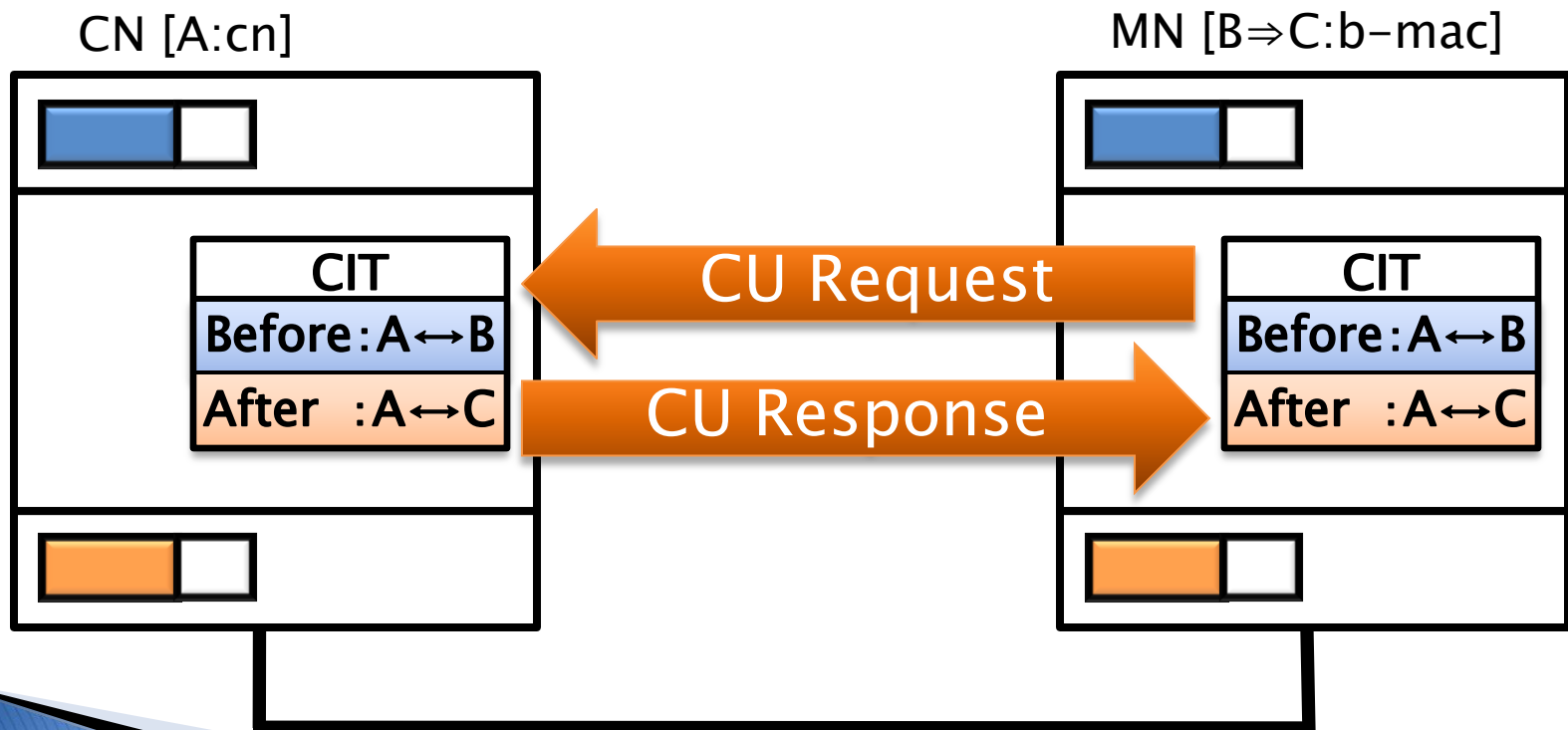
Mobile PPCv6 (移動前)

- ▶ DDNSを用いて通信相手のIPアドレスを解決
- ▶ アドレス変換テーブルの生成
 - CIT (Connection ID Table)



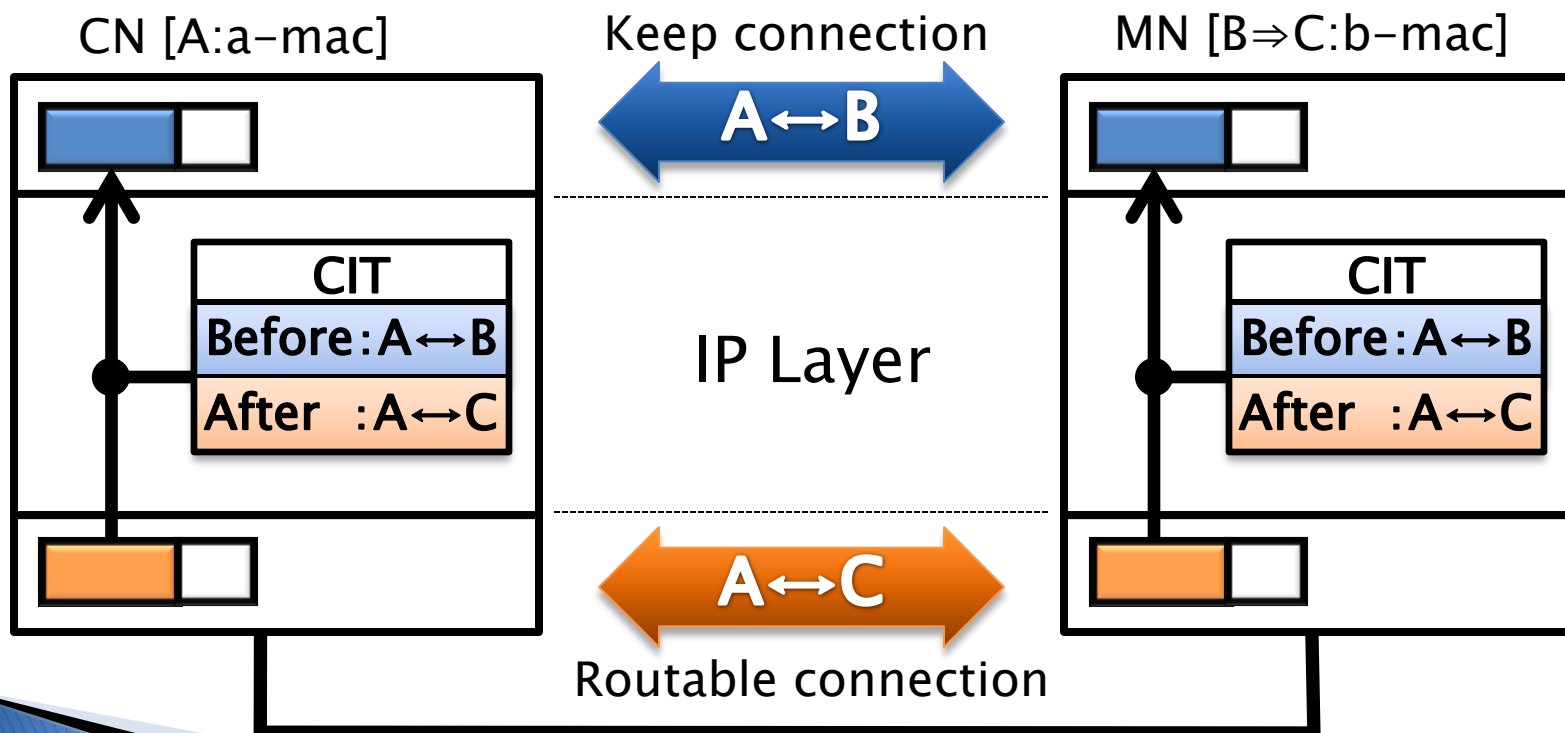
Mobile PPCv6 (移動時)

- ▶ CU Negotiation (CIT Update Negotiation)
 - 移動後のIPアドレスを通知して, CITを更新する
 - CU Request/CU Responseの一往復のネゴシエーション

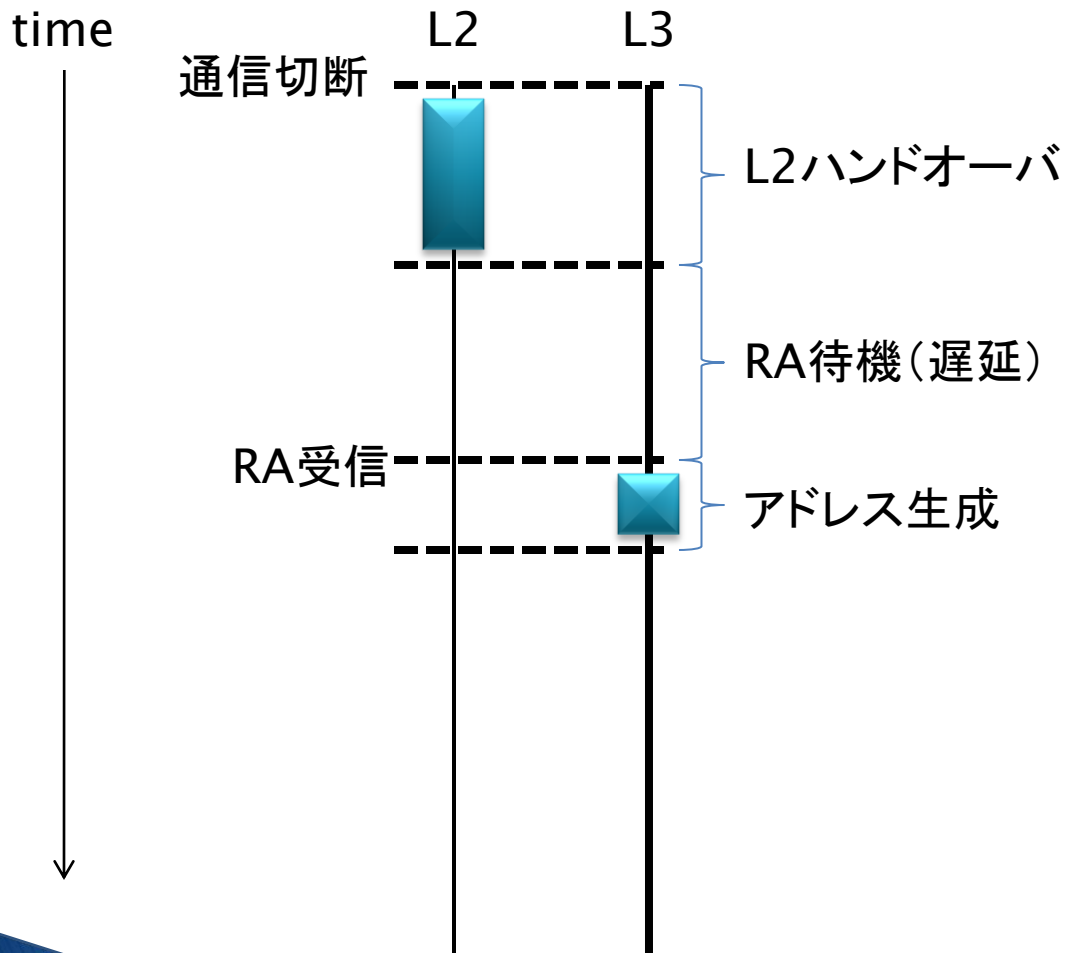


Mobile PPCv6 (移動後)

- ▶ 移動後の通信では、常にアドレス変換を行う
 - 上位層には移動前のIPアドレス, 下位には移動後のアドレス

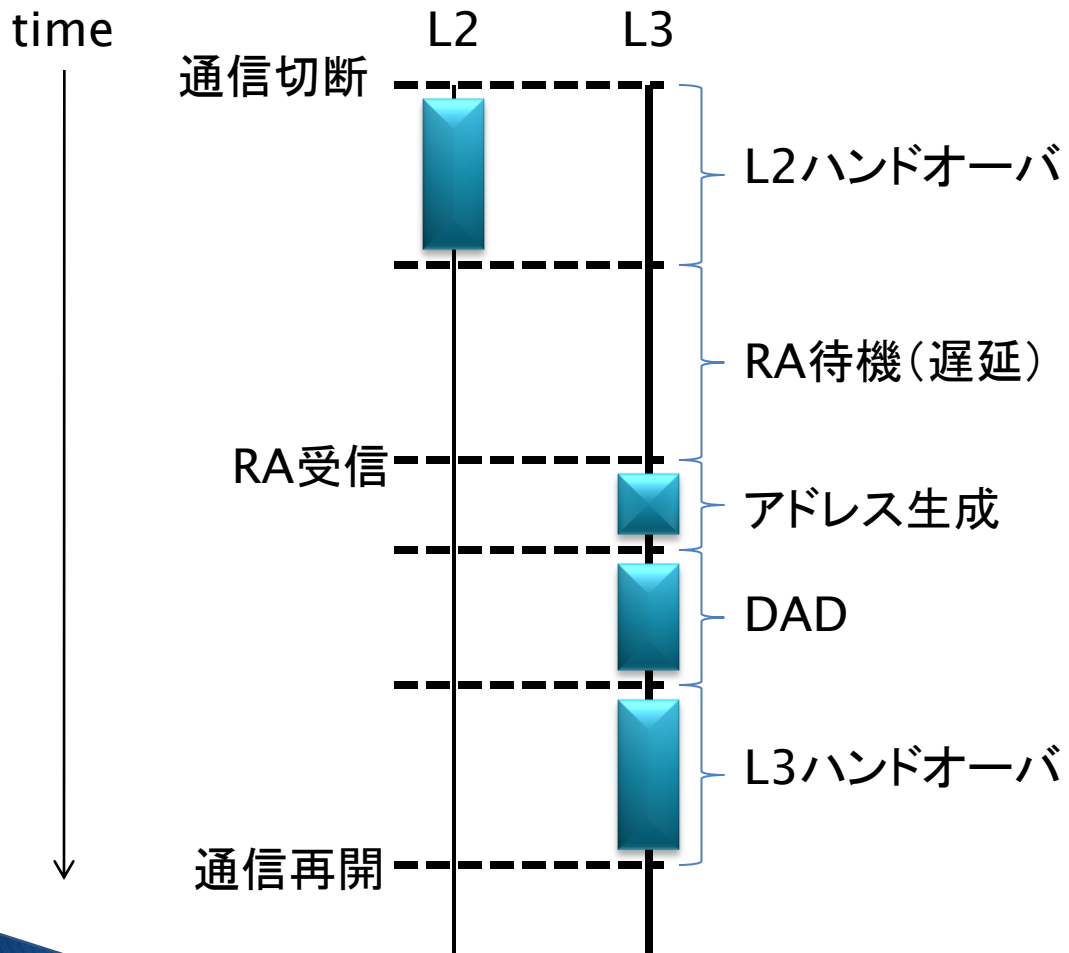


ハンドオーバー



- ▶ L2ハンドオーバー
 - 無線APの切替
- ▶ RA待機時間
 - L2処理とL3処理間の遅延時間
 - 通常3～10秒
- ▶ アドレス生成
 - ルータからのRAによりアドレス生成

ハンドオーバー



▶ DAD

(Duplicated Address Detection)

- 重複アドレスチェック
- 通常, 約2秒

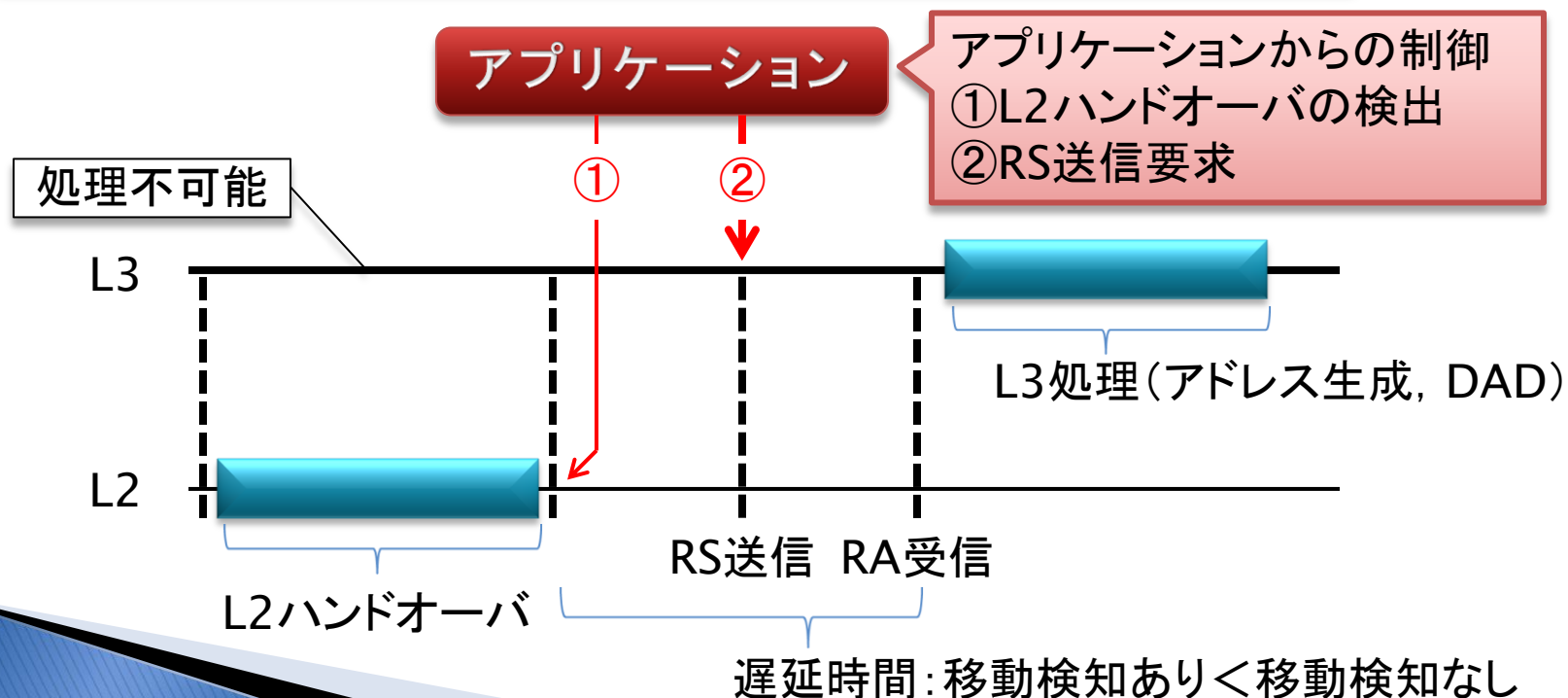
▶ L3ハンドオーバー

- 移動透過性
 - Mobile IPv6
 - MAT
 - LIN6
 - Mobile PPCv6

移動検知時間の短縮

- ▶ アプリケーションから制御を行う
 - L2ハンドオーバー完了後, RSを送信するように制御

移動検知により遅延時間を短縮



DADとは

- ▶ NDP (Neighbor Discovery Protocol) の機能の一つ
 - ▶ アドレス生成直後に行われる
 - ▶ アドレスが重複していないかを検出
 - ローカルリンク内へNSを送信する
 - 一定時間※1内にNAの応答がなければ、アドレス重複なし
- ※1 通常は約1秒
- ▶ NC (Neighbor Cache)
 - IPアドレスとMACアドレスの対応関係を示すテーブル
 - ルータにNCエントリがなければ、ルータから転送されない

DADについて

▶ アドレスの状態

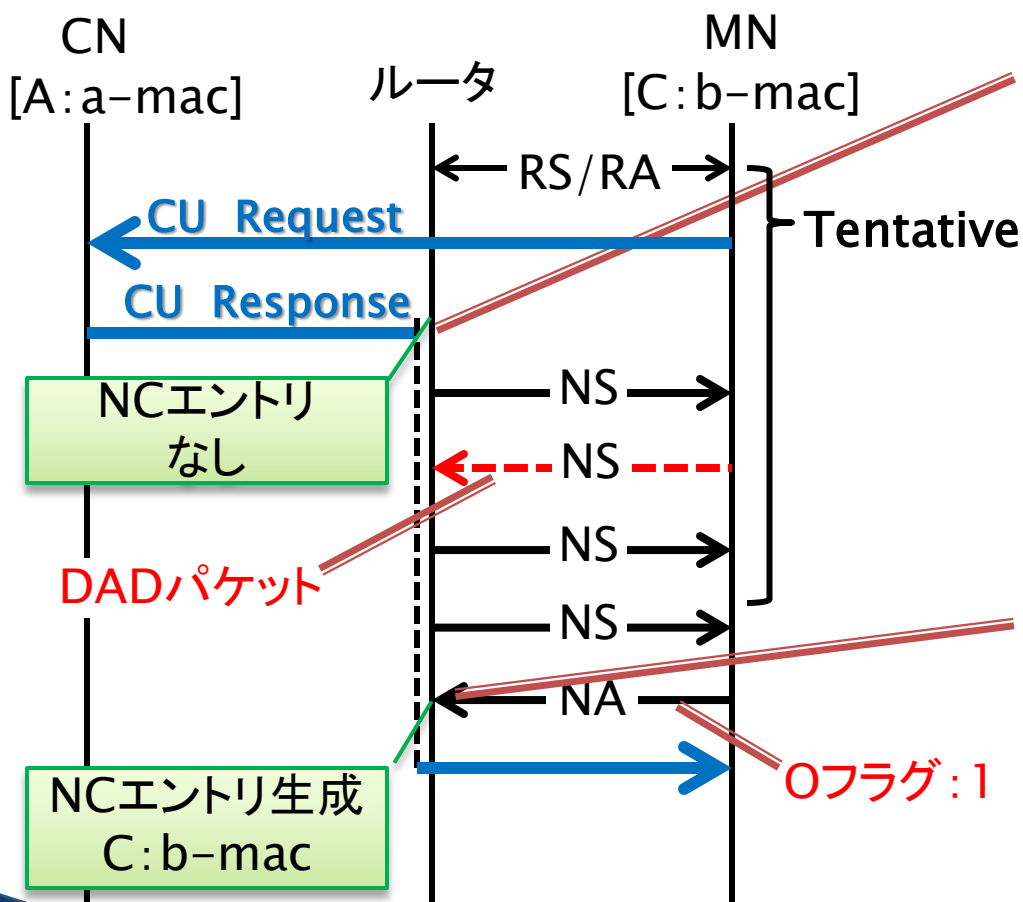
- Valid: 通常通り使用可能な状態(DAD完了後)
- Tentative: アドレス生成直後の状態(DAD処理中)
 - Tentativeの状態では, NAを応答できない

▶ Oフラグ(OVERRIDE)

- 1: NCエントリの上書きを強制
- 0: NCエントリを上書きしない

既存のNCエントリがない場合は, 新規作成を行う

一般的なDADシーケンス



➤ NCエントリの作成が必要
✓ NS/NAでエントリ生成

➤ DADによる遅延
✓ DAD待機時間内では
ルーターにNCエントリが
生成されない
⇒ TentativeではNAが
応答できない

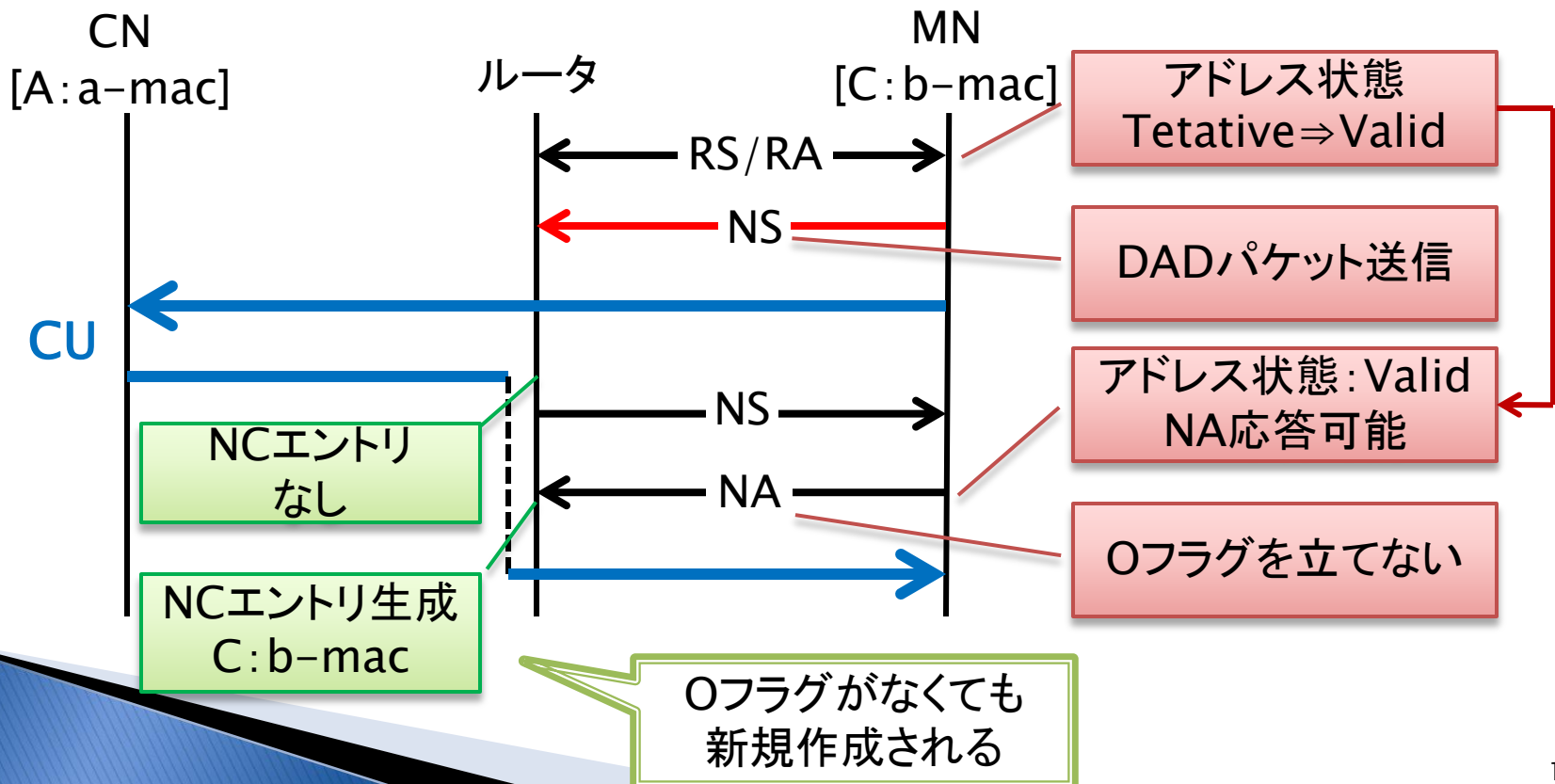
EDAD (Escaping DAD)

- ▶ DADによる処理時間を短縮
 - ルータのNCキャッシュを素早く作成
 - アドレスの状態を制御 (Tentative ⇒ Valid)
- ▶ 他端末に影響を与えない
 - NAのO(OVERRIDE)フラグを制御する.
 - EDAD処理中はOフラグを立てないようにする
 - 既存のNCキャッシュが必ず優先される

端末への実装のみで処理時間を短縮

EDADにおける処理と流れ

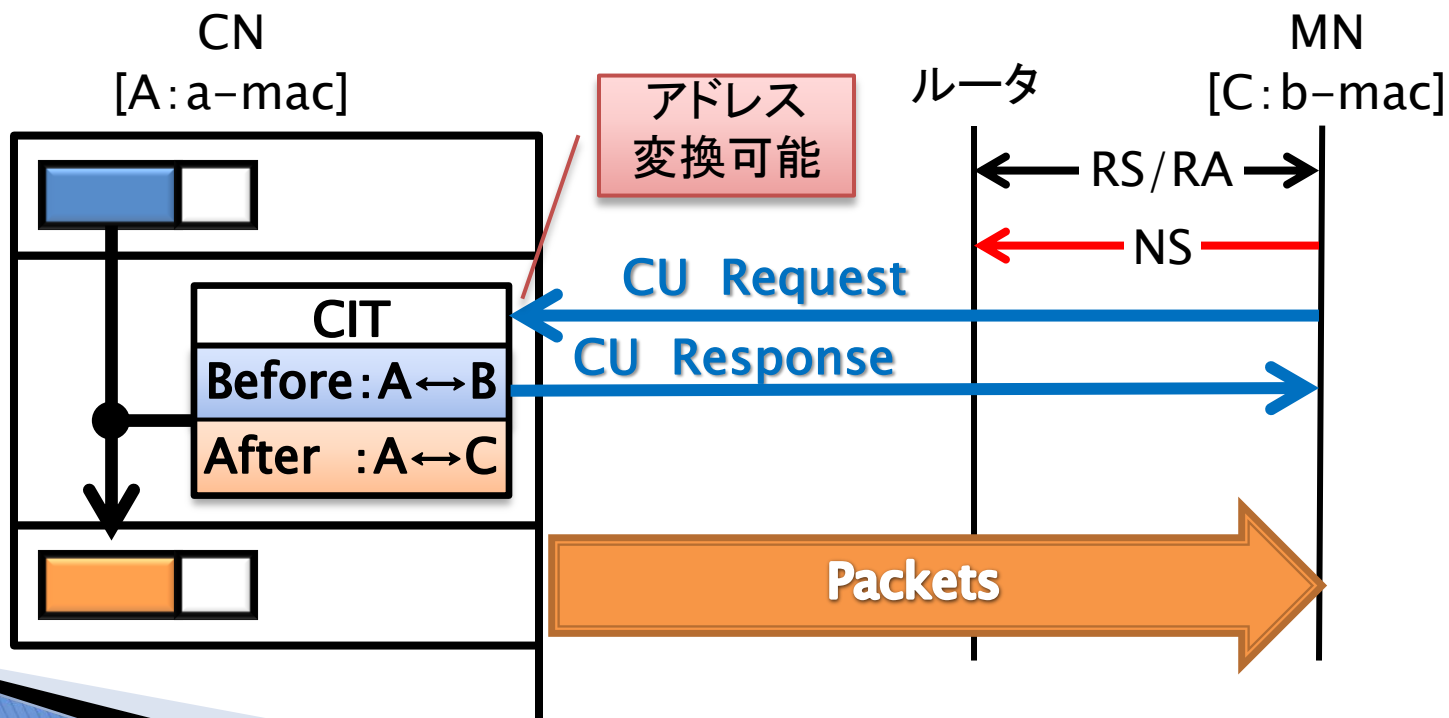
- アドレス状態の制御 (Tentative ⇒ Valid)
- オリジナルでDADパケットを送信
- NAパケットのOフラグの書き換え



CUネゴシエーションの見直し

▶ 現状のCUネゴシエーション

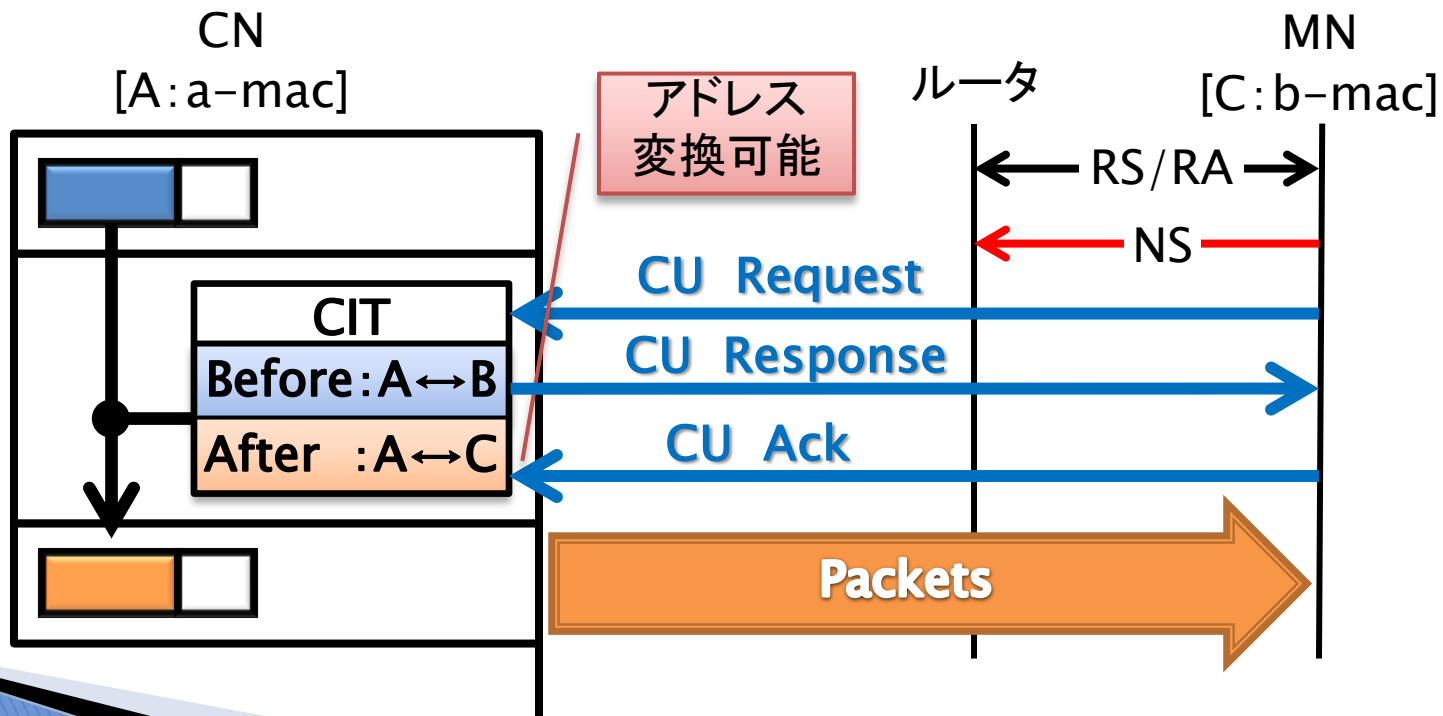
同じアドレスを持つ端末に影響を与える可能性がある！



CUネゴシエーションの改善

▶ 1.5往復のシーケンス

CU Ack受信まで通信再開しない

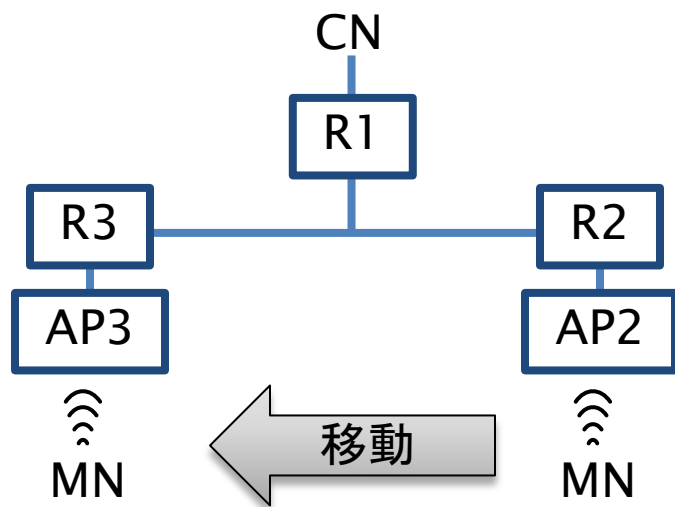


実装

- ▶ FreeBSD 7.0-RELEASEに実装
- ▶ カーネル
 - Mobile PPCv6モジュール
 - EDADモジュールの追加
- ▶ アプリケーション
 - MPPCデーモン
 - 移動検知
 - ルーティングテーブル監視

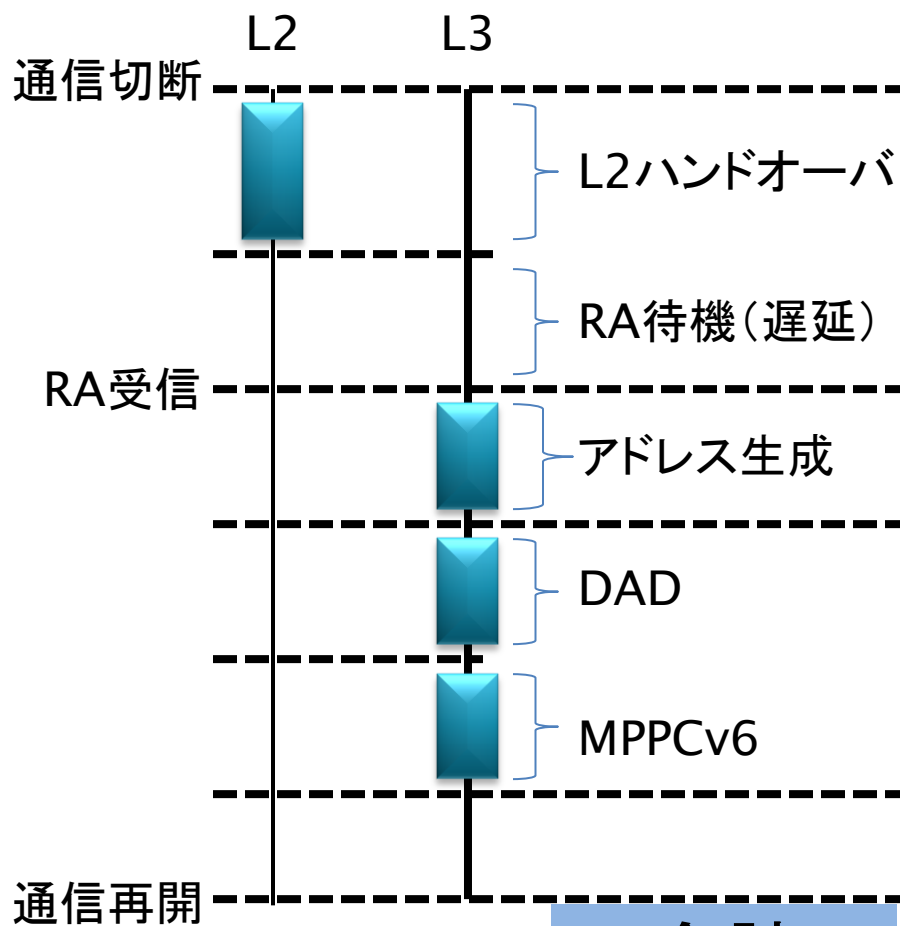
ハンドオーバー処理時間の測定

- ▶ 無線ハンドオーバーにおける実験構成(左図)
 - R: ルータ, AP=アクセスポイント
 - 無線規格 802.11a
- ▶ 各装置における仕様(右表)



	MN	CN
CPU	Core DuI U2500 1.2GHz	Pentium 4 3.0GHz
Memory	1014MB	512MB
NIC	Inte 3945ABG	100Base-TX
OS	FreeBSD 7.0	FreeBSD 7.0

ハンドオーバー処理の測定結果



	移動検知あり EDAD	移動検知なし 通常のDAD
	1.931 [s]	1.931 + 1~5 [s]
	276 [ms]	276 [ms]
	96.5 [ms]	2.005 [ms]
	708 [ms]	708 [ms]
合計	3.004 [s]	4.92 + 1~5 [s]

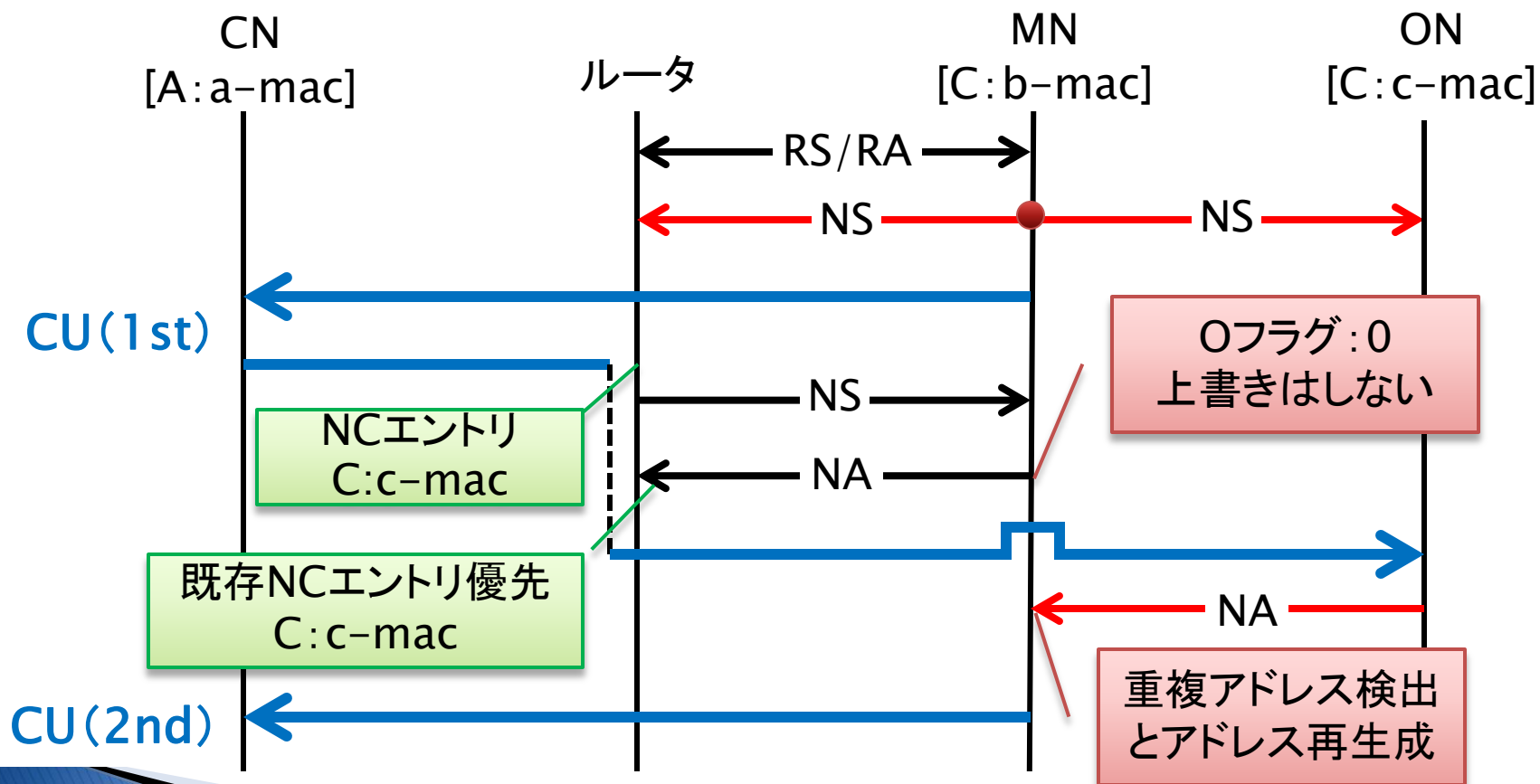
まとめ

- ▶ IPv6ネットワークにおける移動透過性の実現
 - Mobile PPCをIPv6へ対応
 - Mobile PPCv6に新たな機能の追加
- ▶ Mobile PPCv6の実装
 - IPv6ネットワークでの動作確認
 - ハンドオーバー処理時間の測定
 - 追加機能の性能測定
- ▶ 今後の課題
 - IPv4とIPv6ネットワークを跨った場合の移動透過性実現

付録

DAD回避方式(アドレス重複時)

- ▶ アドレス重複端末への影響はない



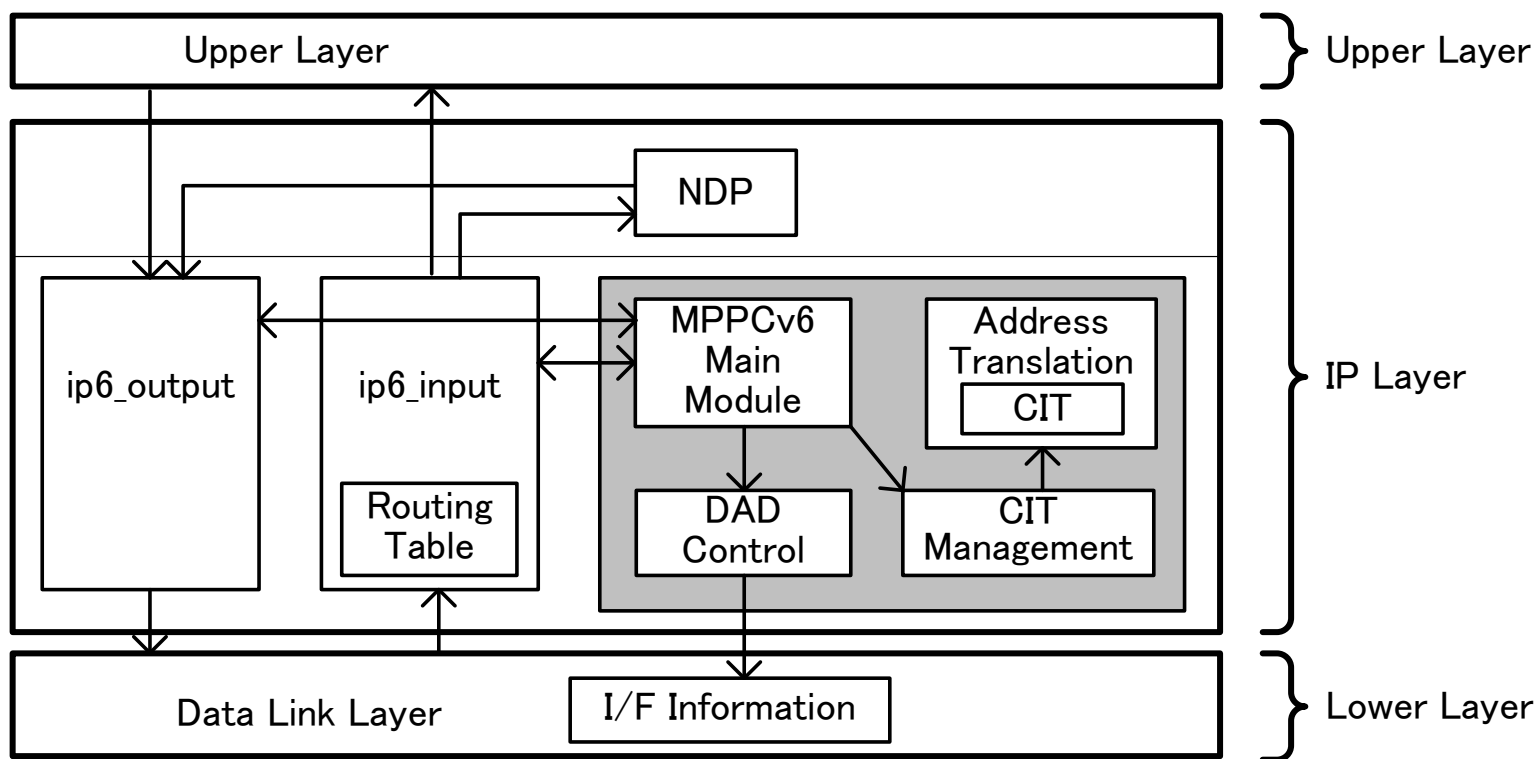
測定結果

- ▶ CUネゴシエーションとDAD処理
 - CU Request送信～NS受信
 - NS受信～NA送信(DAD処理)
 - NA送信～CU Response受信

処理内容	DAD回避方式	通常のDAD
CU Request送信～NS受信	95.21 [ms]	91.55 [ms]
NS受信～NA送信	0.303 [ms]	1.9606 [s]
NA送信～CU Response受信	0.443 [ms]	0.756 [ms]
合計	95.79 [ms]	2.005 [s]

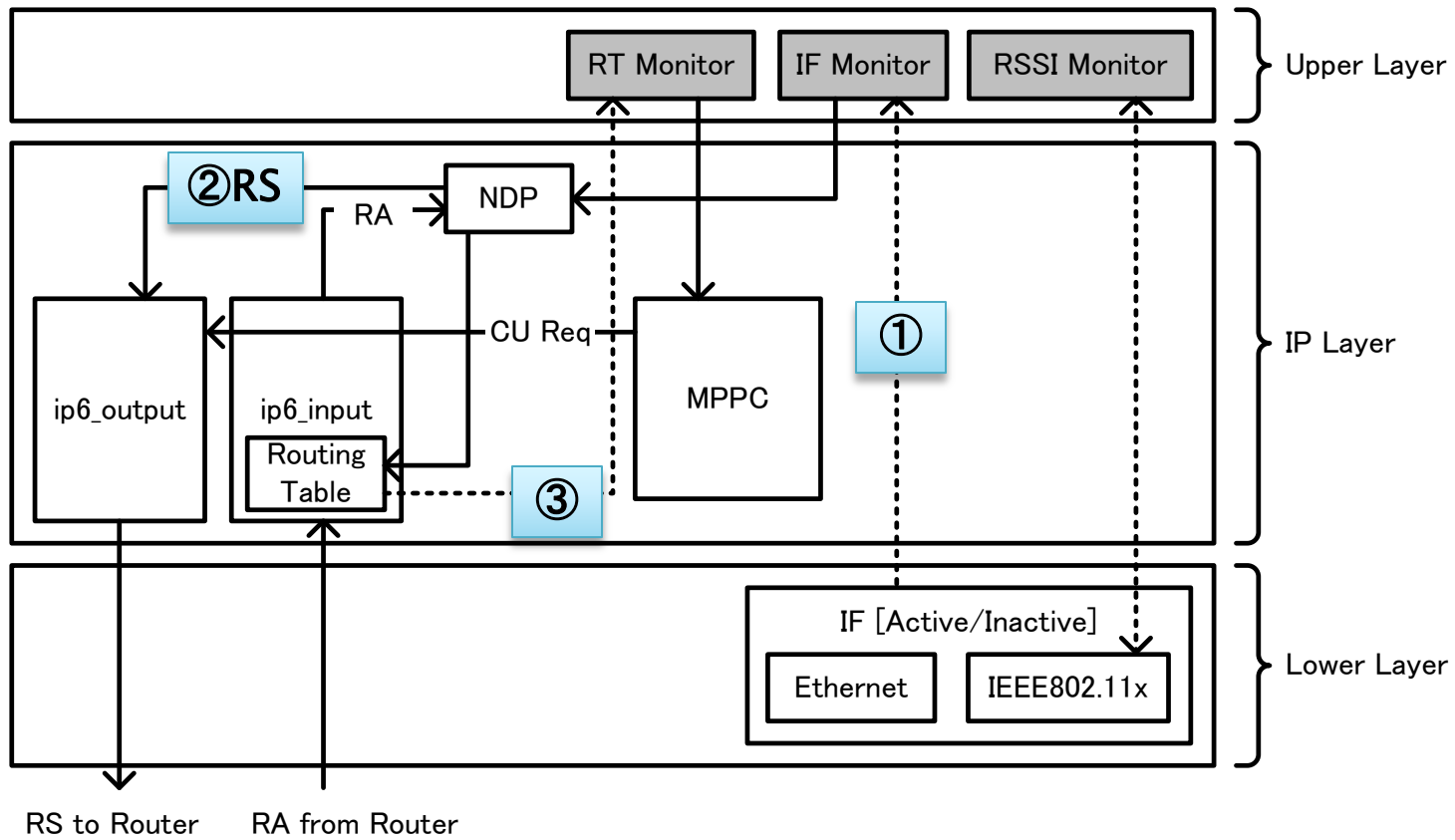
実装(カーネル)

▶ カーネルにおけるモジュール構成

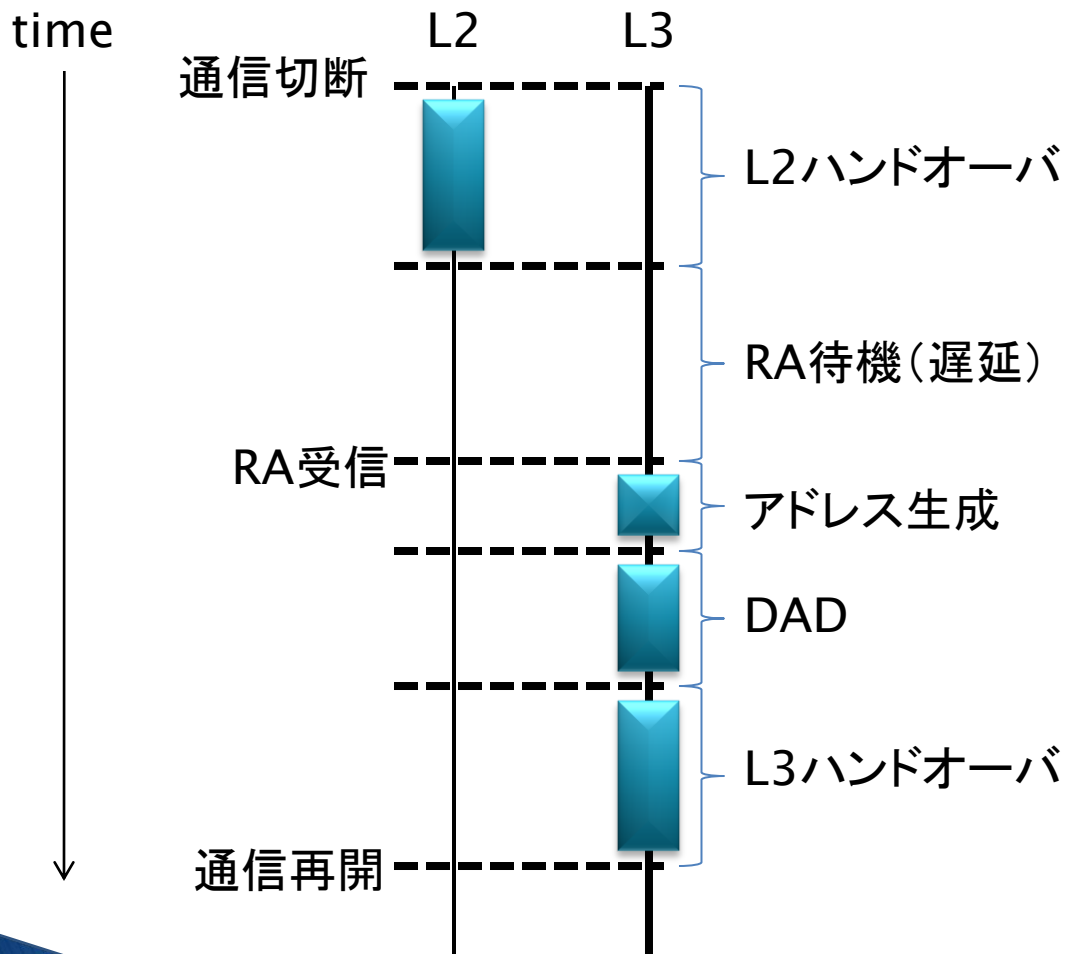


実装 (アプリケーション)

▶ アプリケーションにおけるモジュール構成



移動透過性



- ▶ L2ハンドオーバー
 - 無線APの切替
- ▶ RA待機時間
 - L2処理とL3処理間の遅延時間

移動検知

- ▶ アドレス生成
 - ルータからRAによりアドレス生成

Optimistic DAD

- ▶ DAD完了まで、NS/NAのやり取りをする

