

# IP ネットワークをフラット化する通信アダプターの提案と実装

尾久史弥<sup>†1</sup> 鈴木秀和<sup>†1</sup> 内藤克浩<sup>†2</sup> 渡邊晃<sup>†1</sup>  
<sup>†1</sup> 名城大学大学院理工学研究科 <sup>†2</sup> 愛知工業大学情報科学部

## 1 はじめに

IP インフラは隅々まで普及し、ほとんどのシステムが IP ネットワークを前提として構築されている。しかし、IP ネットワークには、NAT 越え問題、IPv4/IPv6 非互換性、移動透過性、セキュリティなどに係る様々な課題がある。これらの課題を除去し、実ネットワークの制約に左右されないフラットなネットワークを構築できると有用である。

フラットなネットワークを構築するための技術として、DSMIPv6(Dual Stack IPv6)、HIP(Host Identity Protocol)、NTMobile(Network Traversal with Mobility)[1] がある。DSMIPv6 と HIP は、それぞれ固有の課題を抱えるとともに、カーネル空間に実装するのが前提であるため、スマートフォンのような端末への適用が困難である。

NTMobile は、上記のような課題は無く、スマートフォンにも適用できるため、システムを構築する上で有用な技術である。しかし、組み込み機器やアプリケーションサーバーのようにプログラムの改造が一切できない通信装置が存在する。そこで、これらの通信装置に対しても、NTMobile の利用を可能とするアダプターの提案と実装を行った。

## 2 既存技術の概要と課題

### 2.1 DSMIPv6

DSMIPv6 は、MobileIPv6 を IPv4/IPv6 混在環境に拡張した技術である。HA(Home Agent) を IPv4/IPv6 デュアルスタックネットワーク上に設置し、移動端末 MN(Mobile Node) との間でトンネルを構築することにより、あらゆる通信環境での通信接続性と移動透過性を実現することができる。しかし、移動端末がグローバル IPv4 アドレスを消費するという課題があり、グローバル IPv4 アドレスが枯渇した今日においては、現実的な方式ではない。また、カーネルにて本機能をサポートするのが前提であり、スマートフォンでは利用できない。

### 2.2 HIP

HIP は、ネットワーク層とトランスポート層の間に、新たに HIP 層を定義し、端末識別子として HI(Host Identifier) を用いるのが特徴で、既存の NAT 越え技術やセキュリティ技術を組み合わせて、通信接続性と移動透過性を実現する。しかし、NAT が存在する環境では移動透過性の実現のためのオーバーヘッドが大きい。また、カーネルを改造する必要があるため、スマートフォンでは適用できない。

## 3 NTMobile

NTMobile は、システム内で重複しない仮想 IP アドレス (VIP : Virtual IP Address) を導入し、全ての通信を実 IP アドレス (RIP : Real IP Address) でカプセル化する。NTMobile を利用するには、NTMobile framework(NTMfw)[2] と呼ぶライブラリを利用してアプリケーションを開発する (以下、NTMapp と呼ぶ)。

NTMobile は、NTMapp をインストールした NTM 端末と、NTM 端末の VIP や位置情報等を管理し、UDP トンネルの構築指示を出す DC(Direction coordinator) で構成される。通信開始時の名前解決をトリガとし、DC と NTM 端末の間でシグナリングを実行して最適なトンネル経路を構築する。以後の全ての通信は、VIP に基づいたパケットを RIP のパケットでカプセル化することにより通信を行う。

NTMobile は、DSMIPv6 や HIP が抱えるような課題がなく、スマートフォンでも実現できる。ただし、組み込み機器やアプリケーションサーバーなどは、プログラムの改造ができないため、NTMobile を利用できない課題がある。

## 4 提案システム

### 4.1 アダプタの構成

本稿は、プログラムの改造が一切できない通信装置とネットワークの間に直列に設置することにより、NTMobile 通信を可能とする NTMA(NTMobile Adaptor) を提案する。NTMA は、イニシエータ側 (NTMA<sub>ini</sub>)、レスポнда側 (NTMA<sub>res</sub>) のどちら側でも利用できる。

NTMA は NIC を 2 枚用意し、一方を一般端末 GN(General Node) に、他方を IP ネットワークに接続する。以後イニシエータ側の GN を GN1、レスポнда側の GN を GN2 とする。

---

Proposal and Implementation of Communication Adapter that makes IP Network Flat

Fumiya Ogyu<sup>†1</sup>, Hidekazu Suzuki<sup>†2</sup>, Katsuhiro Naito<sup>†2</sup> and Akira Watanabe

<sup>†1</sup> Graduate School of Science, Meijo University

<sup>†2</sup> Faculty of Information Science, Aichi Institute of Technology

## 4.2 動作シーケンス

図1に提案システムの動作シーケンスと通信パケットのアドレス遷移を示す。通信開始前に、NTMA<sub>ini</sub>とNTMA<sub>res</sub>は、DCからVIPの割り当てを受け、GN1とGN2には、適切なIPアドレスを設定しておく。また、図1では、説明の簡略化のためにNATの存在及びNTNMobileシグナリングの詳細を省略している。

通信開始時にGN1は、NTMA<sub>res</sub>のFQDNをターゲットとするDNSクエリを送信する。NTMA<sub>ini</sub>は、DNSクエリを受信後、NTNMobileシグナリングを実行し、DCの指示の基でNTMA<sub>ini</sub>とNTMA<sub>res</sub>間でトンネル構築をする。トンネル構築後にNTMA<sub>ini</sub>は、NTMA<sub>res</sub>のVIP<sub>res</sub>をDNSレスポンスに載せてGN1に返す。GN1は、NTMA<sub>res</sub>のVIP<sub>res</sub>を通信相手と認識してパケットを送信する。NTMA<sub>ini</sub>は、送信元RIP<sub>GN1</sub>をVIP<sub>ini</sub>に変換する。さらに、NTMA<sub>ini</sub>とNTMA<sub>res</sub>のRIPでカプセル化してNTMA<sub>res</sub>へ送信する。NTMA<sub>res</sub>はパケットを受信後して、デカプセル化する。その後、VIP<sub>res</sub>をRIP<sub>GN2</sub>に変換して、GN2へパケットを中継する。GN2からの応答は、上記と逆の手順で中継する。

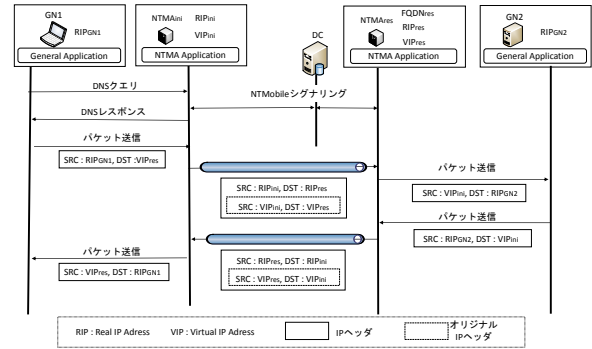


図1 NTMAの通信シーケンスとパケットのアドレス遷移

## 5 実装

図2にNTMAのモジュール構成図を示す。NIC0は、ネットワーク側の端末との通信に使用し、NTMfwによりNTNMobileのパケットを扱う。NIC1はGNとの通信に使用し、RAWソケットによりパケットをIPヘッダから受信して、データとして扱う。

NTMfwは、NTNMobileシグナリングをGNに代わって実行する。また、通信パケットに対しては、NTNMobile通信を識別するNTMヘッダの生成/除去を行い、Linuxカーネルにて、実IPパケットのカプセル/デカプセル化処理を行う。NTMappは、NTMfwに対して初期化処理(Init)を指示する。GNからのDNSクエリに対して、NTMfwにシグナリングを指示し、その結果をDNSレスポンスとしてGNに送信する。さらに通信パケットに対して、RAWソケットとNTMソケットを経由した中継処理をする。

## 6 評価

### 6.1 動作検証

NTMA<sub>ini</sub>をRaspberry Pi上で実現し、NTMA<sub>res</sub>をLINUX PC上に実装した。NTMA<sub>ini</sub>とNTMA<sub>res</sub>をそれぞれNAT配下に設置し、GN1とGN2に接続した。両GNは、一般のUDP/TCP通信を行うアプリケーションを使用して動作検証を行い、NATが介在するネットワークにおいても自由に通信を行えることを確認した。

### 6.2 性能評価

性能評価は、動作検証と同一条件で、Raspberry Piに実装したNTMAを用いてパケットの中継時間を測定し

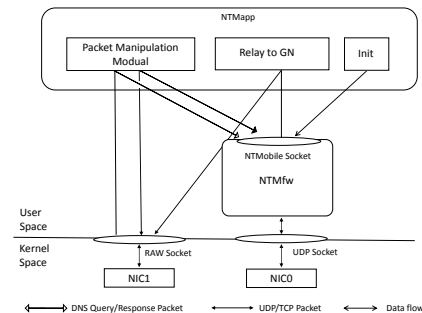


図2 NTMAのモジュール構成

表1 NTMAのパケット中継時間

区分	時間 (us)
GN から IP ネットワーク	505
IP ネットワークから GN	697

た。表1にNTMAがパケットの中継に要した時間を示す。GNからIPネットワーク、及びIPネットワークからGNへパケットを中継するのに要する時間は、それぞれ505マイクロ秒、697マイクロ秒であった。この時間には、NTNMobileによる暗号化/復号化処理も含まれる。この結果により、アダプタが介在することによる伝送遅延は、実用上問題ないことが分かった。

## 7 まとめ

本稿では、プログラムの改造が一切できない通信装置に対して、NTNMobileによるフラットなネットワーク上で通信を行うことができるNTMAを提案した。実装及び性能評価の結果、NTMAによる伝送遅延は少なく、実用上問題ないことを確認した。

### 参考文献

- [1] 上酔尾一真ほか: 情報処理学会論文誌, Vol54, No10, pp. 2288–2299(2013).
- [2] 納堂博史ほか: 実用化に向けたNTNMobileフレームワークの実装と評価, 信学技報, Vol.116, No.509, pp.281–288 (2017).

# IPネットワークをフラット化する 通信アダプターの提案

尾久 史弥<sup>(1)</sup> 鈴木 秀和<sup>(1)</sup> 内藤 克浩<sup>(2)</sup> 渡邊 晃<sup>(1)</sup>

(1)名城大学 理工学研究科, (2)愛知工業大学 情報科学部

# 研究背景

IPネットワークの現状

IPインフラが普及し、多くのシステムがIP通信を利用

# 研究背景

IPネットワークの現状

IPインフラが普及し、多くのシステムがIP通信を利用

IPネットワークの課題

NAT越え問題 IPv4/IPv6の非互換性 移動透過性 セキュリティ

# 研究背景

## IPネットワークの現状

IPインフラが普及し、多くのシステムがIP通信を利用

## IPネットワークの課題

NAT越え問題 IPv4/IPv6の非互換性 移動透過性 セキュリティ

## 解決策

通信プロトコルレベルで課題を  
解決できると有用

## 技術例

- (1) HIP (Host Identity Protocol)
- (2) DSMIPv6 (Dual Stack Mobile IPv6)
- (3) NTMobile (Network Traversal with Mobility)

(1) H. Soliman: Mobile IPv6 Support for Dual Stack Hosts and Routers, RFC5555, IETF (2009).

(2) R. Moskowitz, et al.: Host Identity Protocol Version 2 (HIPv2), RFC7401, Updated by RFC8002, IETF (2015).

(3) 納堂博史, ほか: NTMobileの実用化に向けた統合的枠組の検討, 情報処理学会研究報告, Vol.2015-MBL-77, No.20, pp.1-8(2015).

# 既存技術の概要と課題

DSMIPv6 <sup>※</sup>	概要	IPv4/IPv6混在環境において移動透過性を提供する Mobile IPv4とMobileIPv6を統合した技術
	課題	Mobile IPv4の課題を引き継ぐ 移動端末にIPv4グローバルを割り当てる必要がある
HIP <sup>※</sup>	概要	エンドポイント間でのセキュリティプロトコル
	課題	通信開始時に経路探索のオーバーヘッドが大きい

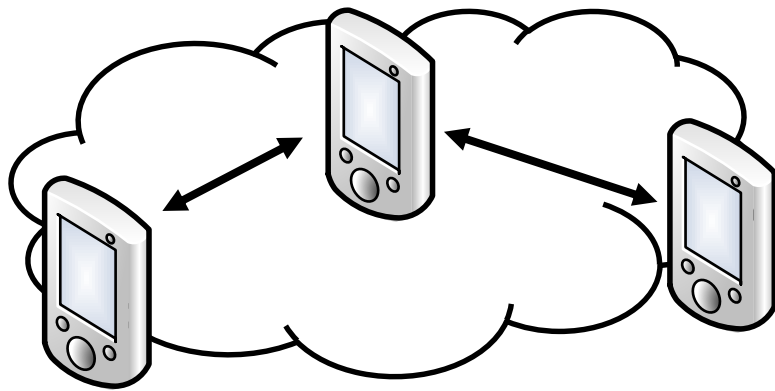
## 共通の課題

カーネル空間に改造が必要

# NTMobileについて

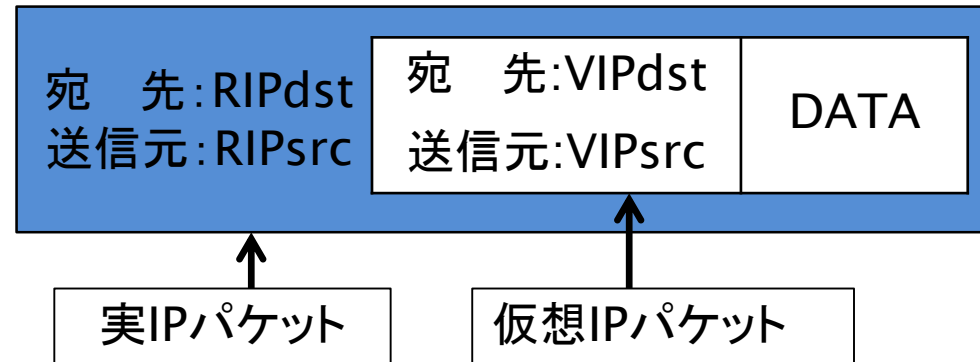
## 特徴

DSMIPv6やHIPなどが抱える課題が**存在しない**  
アプリケーションレイヤで**利用可能**



仮想IPネットワーク

## NTMobileパケットフォーマット





# NTMobileの実装方法(NTMfw)

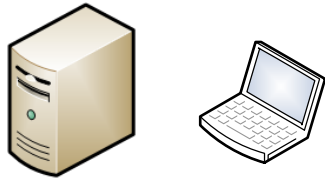
NTMobile framework(NTMfw)と呼ばれる通信ライブラリを利用  
アプリケーション開発者は, NTMソケットAPIを利用する

## NTMobileアプリケーション

```
ntmfw_getaddrinfo(...);  
    ntmfw_socket(...);  
        ntmfw_sendto(...);  
            ntmfw_recvfrom(...);
```

# NTMobile

NTM端末



NTMfw を実装した端末

DC (Direction Coordinator)



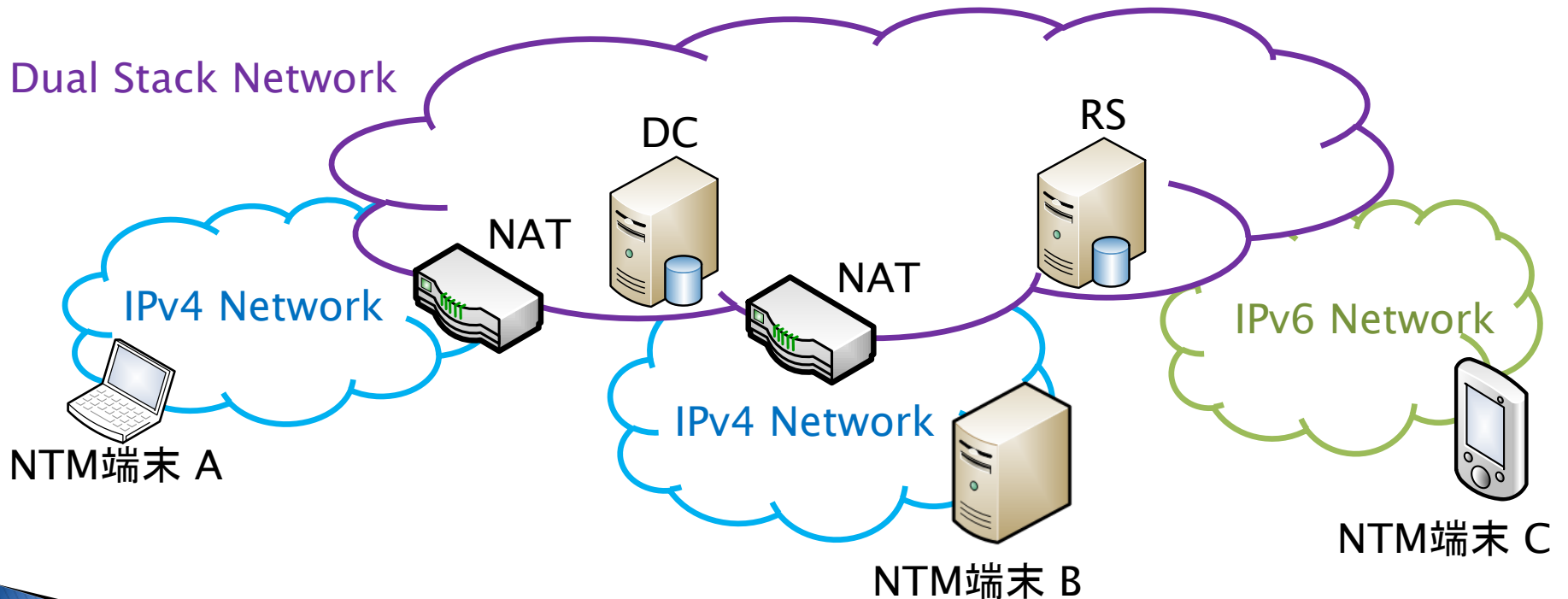
仮想IPアドレスの配布,  
通信経路の指示をする

RS (Relay Server)



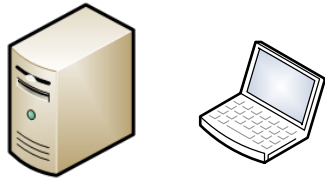
直接通信できない場合の  
通信中継装置

Dual Stack Network



# NTMobile

NTM端末



NTMfw を実装した端末

DC (Direction Coordinator)



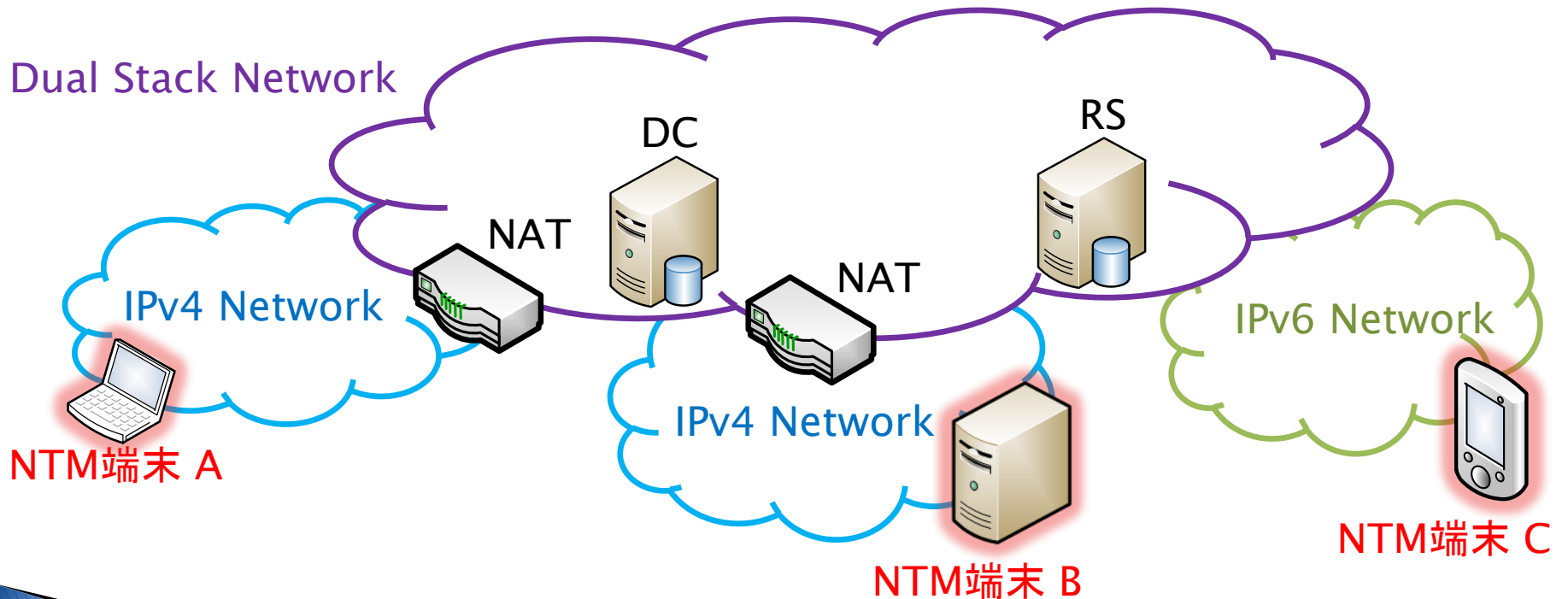
仮想IPアドレスの配布,  
通信経路の指示をする

RS (Relay Server)



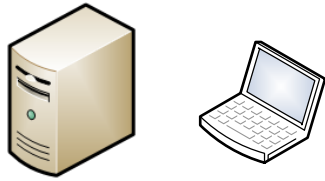
直接通信できない場合の  
通信中継装置

Dual Stack Network



# NTMobile

NTM端末



NTMfw を実装した端末

DC (Direction Coordinator)



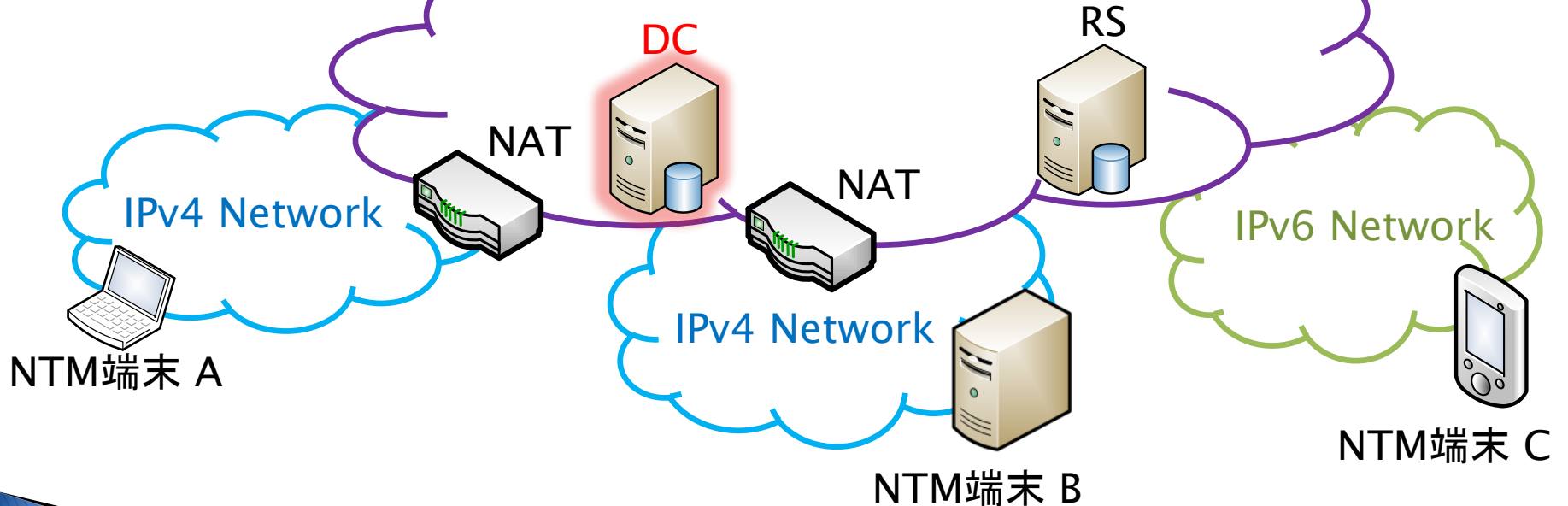
仮想IPアドレスの配布,  
通信経路の指示をする

RS (Relay Server)



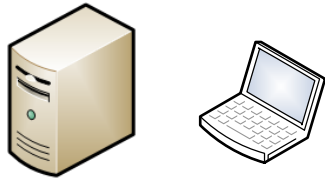
直接通信できない場合の  
通信中継装置

Dual Stack Network



# NTMobile

NTM端末



NTMfw を実装した端末

DC (Direction Coordinator)



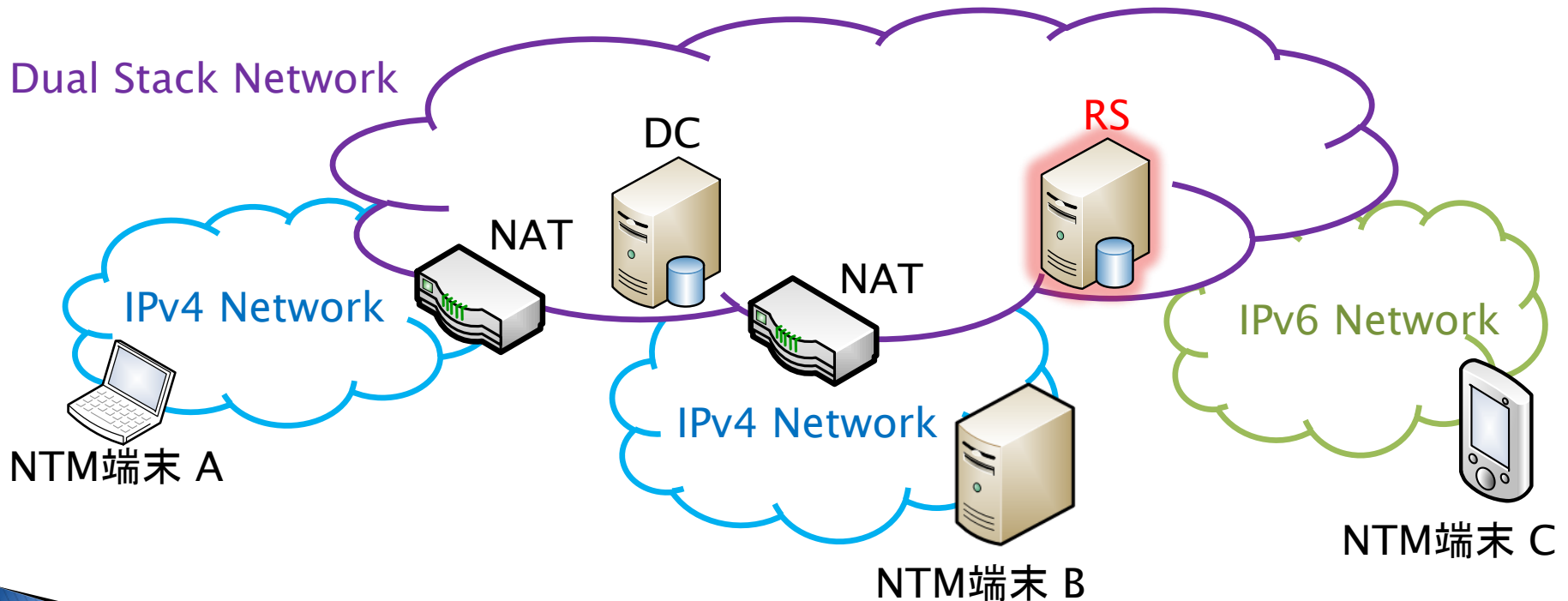
仮想IPアドレスの配布,  
通信経路の指示をする

RS (Relay Server)



直接通信できない場合の  
通信中継装置

Dual Stack Network

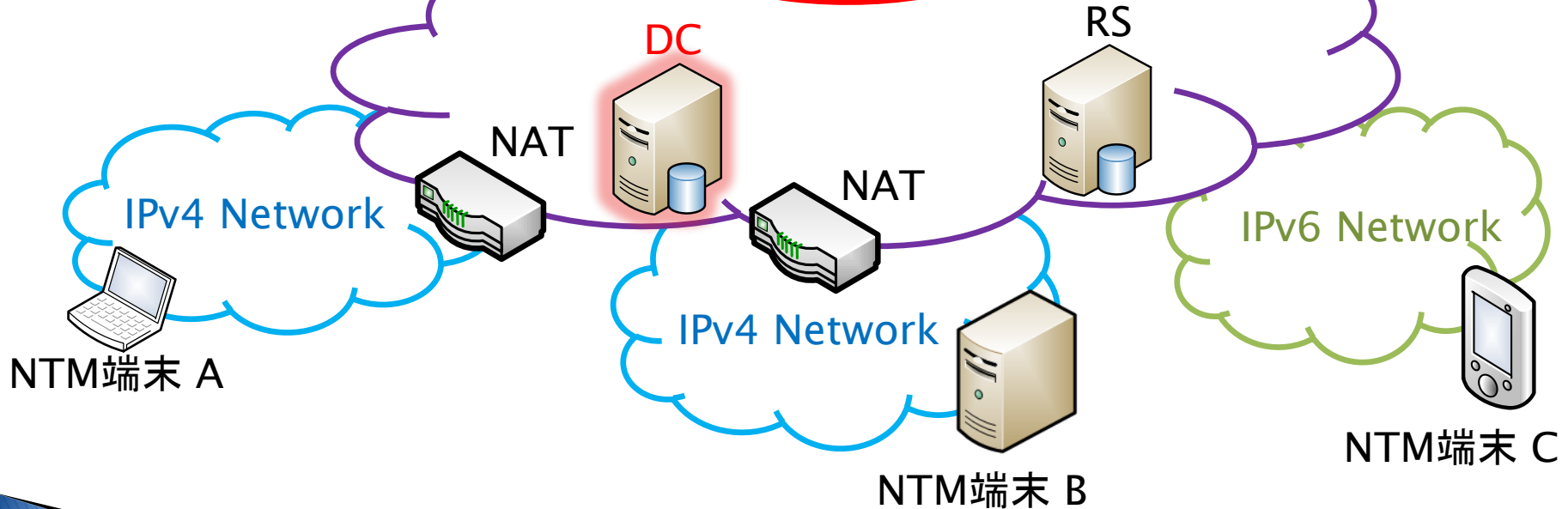


# NTMobile

IPネットワーク上に  
仮想IPネットワークを構築

Virtual IP Network  
(Overlay Network)

Dual Stack Network

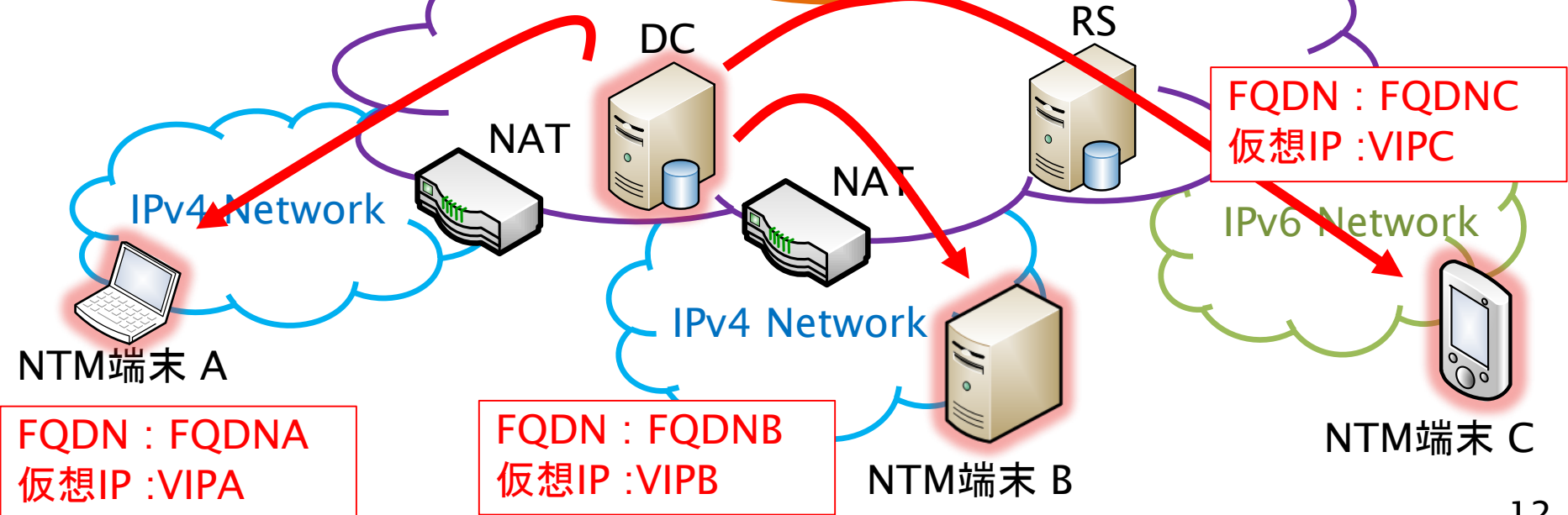


# NTMobile

NTM端末に仮想IPアドレスとFQDNを割り当てる

Virtual IP Network  
(Overlay Network)

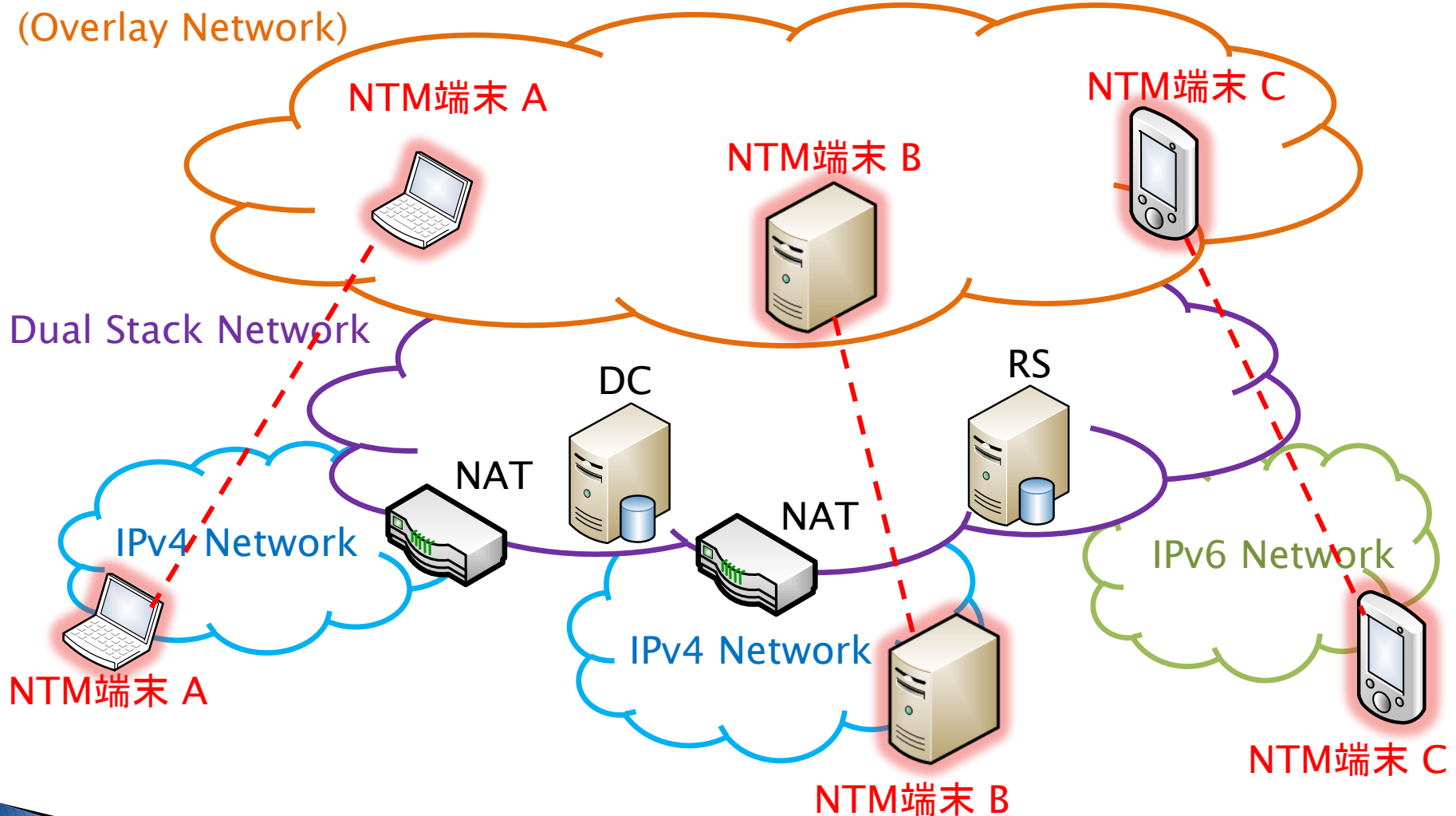
Dual Stack Network



# NTMobile

仮想IPネットワークに  
NTM端末をマッピング

Virtual IP Network  
(Overlay Network)

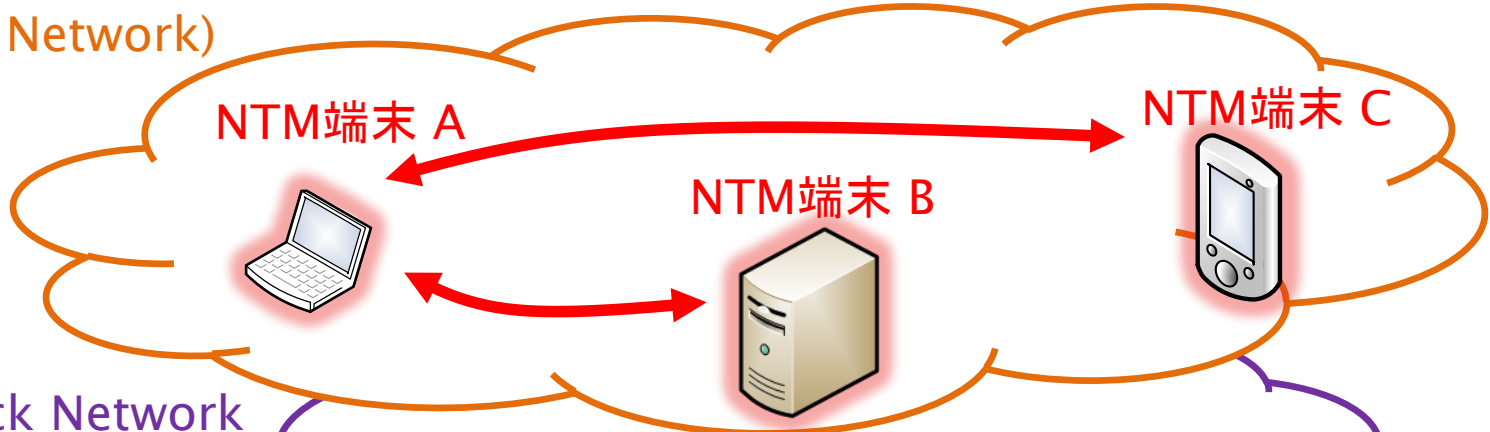




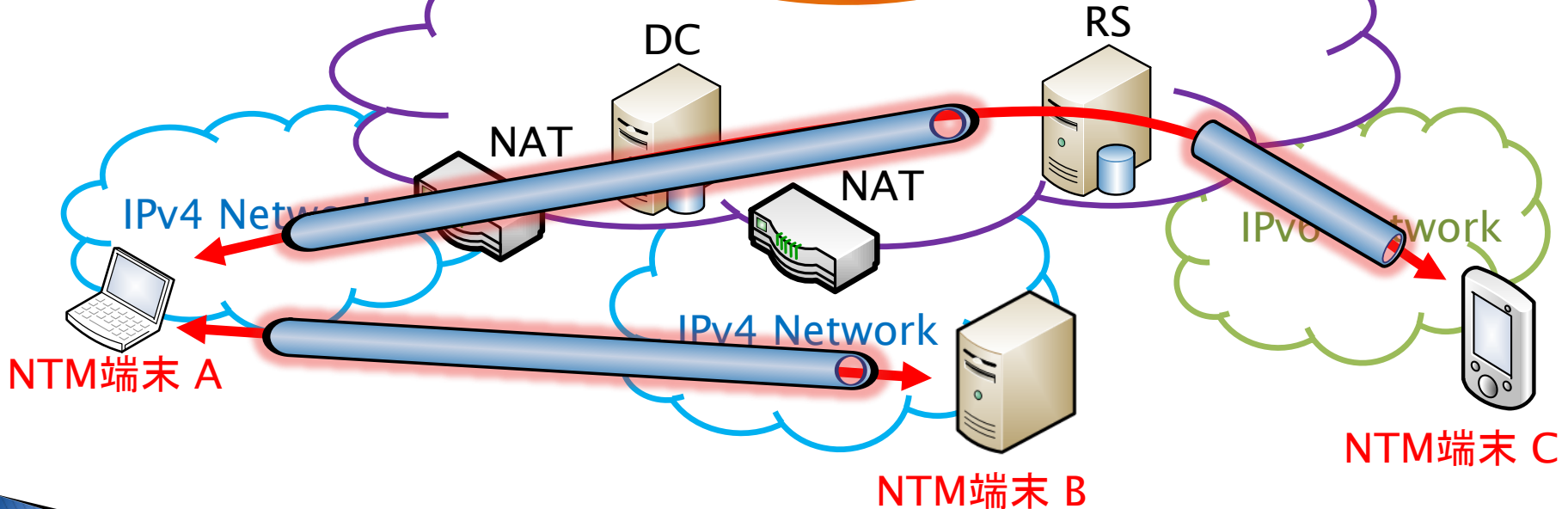
# NTMobile

NTMobileプロトコルにより  
両エンド端末間でエンドツーエンド通信

Virtual IP Network  
(Overlay Network)



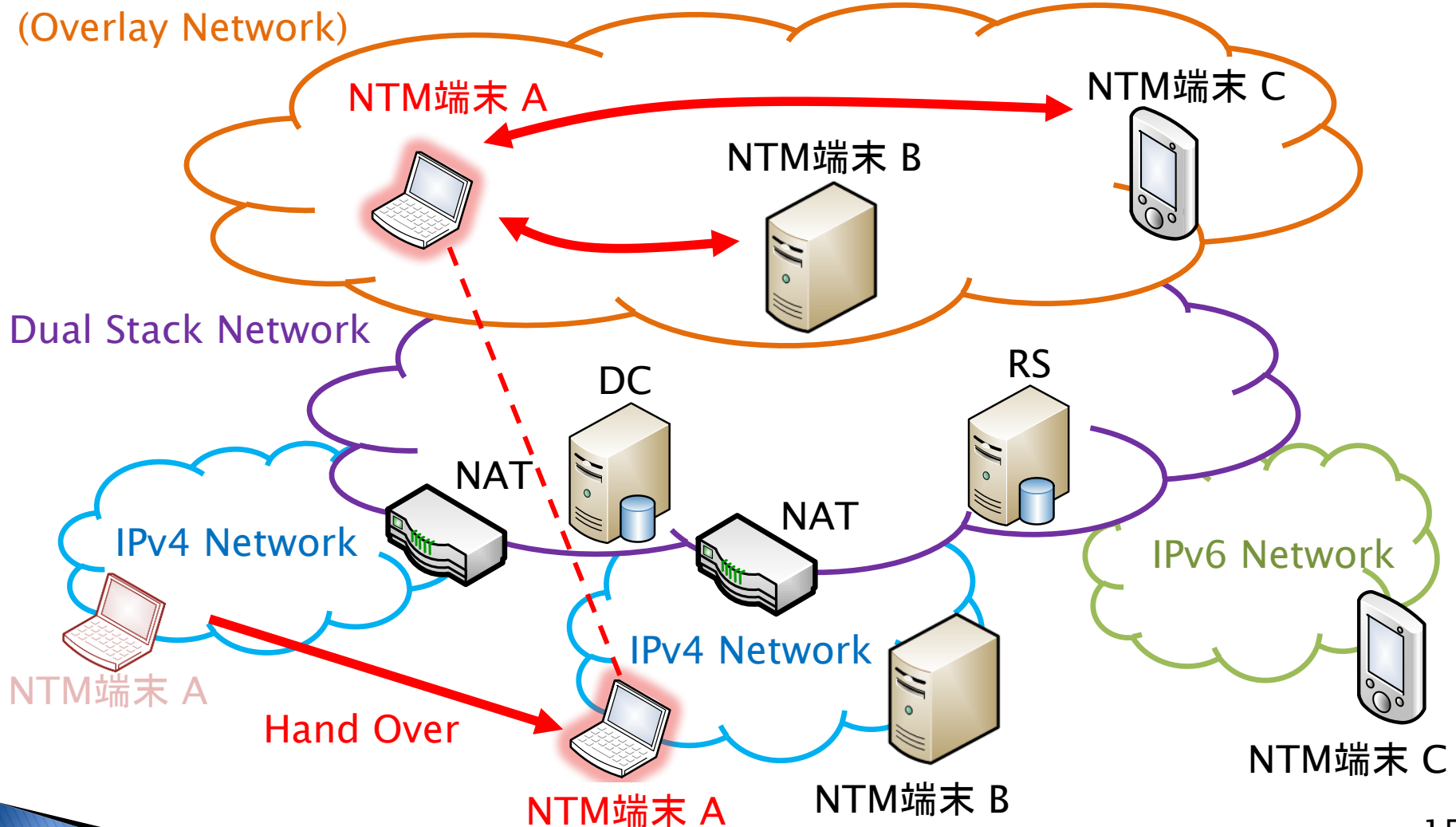
Dual Stack Network



# NTMobile

移動端末を仮想IPネットワークに再度マッピングすることで通信を継続

Virtual IP Network (Overlay Network)



# 既存通信装置への要求

既存の機器がNTMobile通信を使いたい

## アプリケーションの改造が許可されない通信装置の存在

アプリケーションサーバー  
組み込み型機器

アプリケーションの改造が許可されない  
プログラムの書き換えが不可能

# 既存通信装置への要求

既存の機器がNTMobile通信を使いたい

アプリケーションの改造が許可されない通信装置の存在

アプリケーションサーバー

アプリケーションの改造が許可されない

組み込み型機器

プログラムの書き換えが不可能

## 研究目的

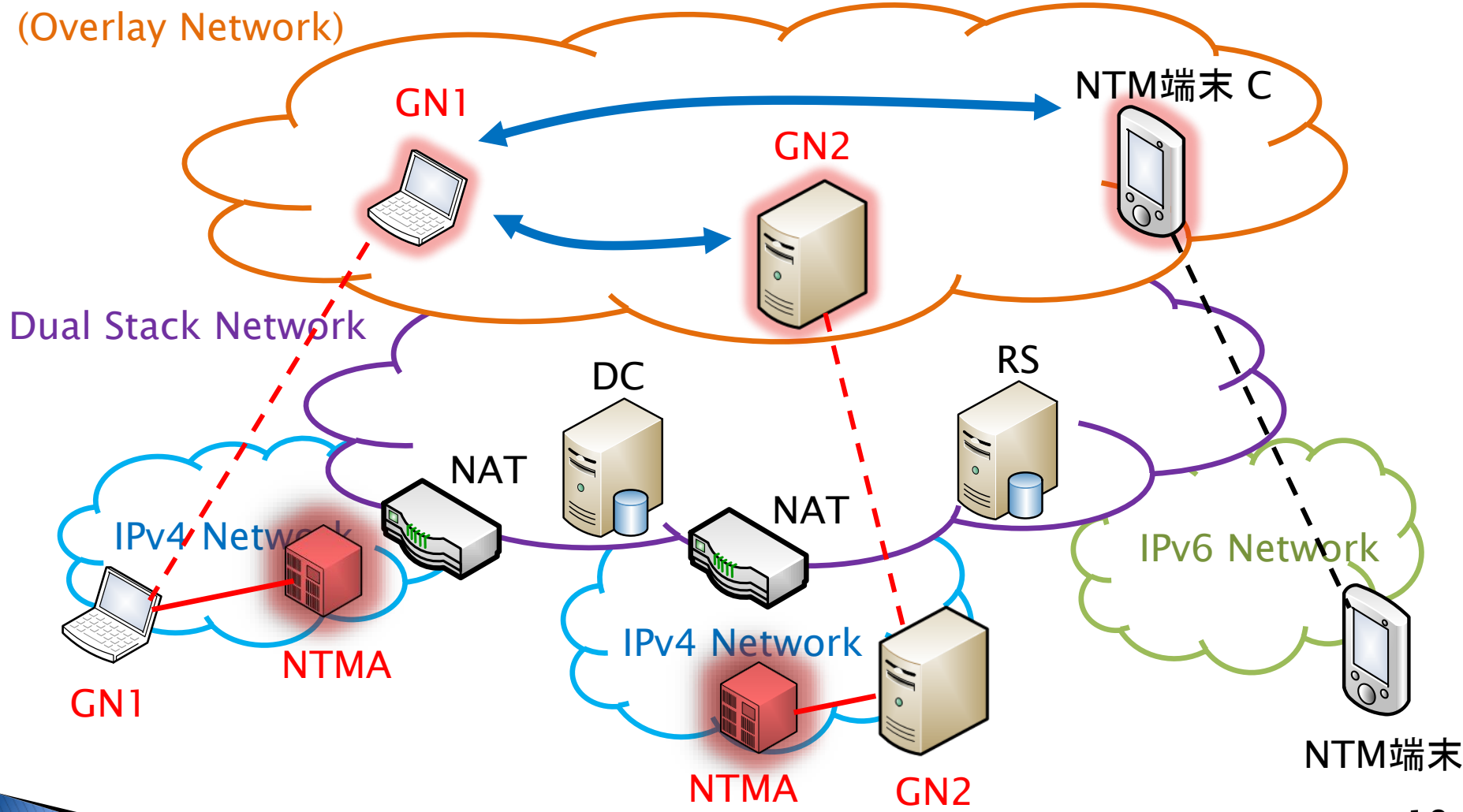
NTMobileを利用できない通信装置(GN : General Node)でも  
サポートを可能にできると有用

**NTMobile Adaptor(NTMA)の提案**

# NTMAへの要求

NTMAとGNを直接つなげることで  
GNを仮想IPネットワークにマッピング

Virtual IP Network  
(Overlay Network)



# NTMAの実現方法

1. NTMobileの仮想IPパケット生成処理に着目
2. 仮想IPヘッダ生成フェーズをNTMfwの機能から分離して一般端末で代行

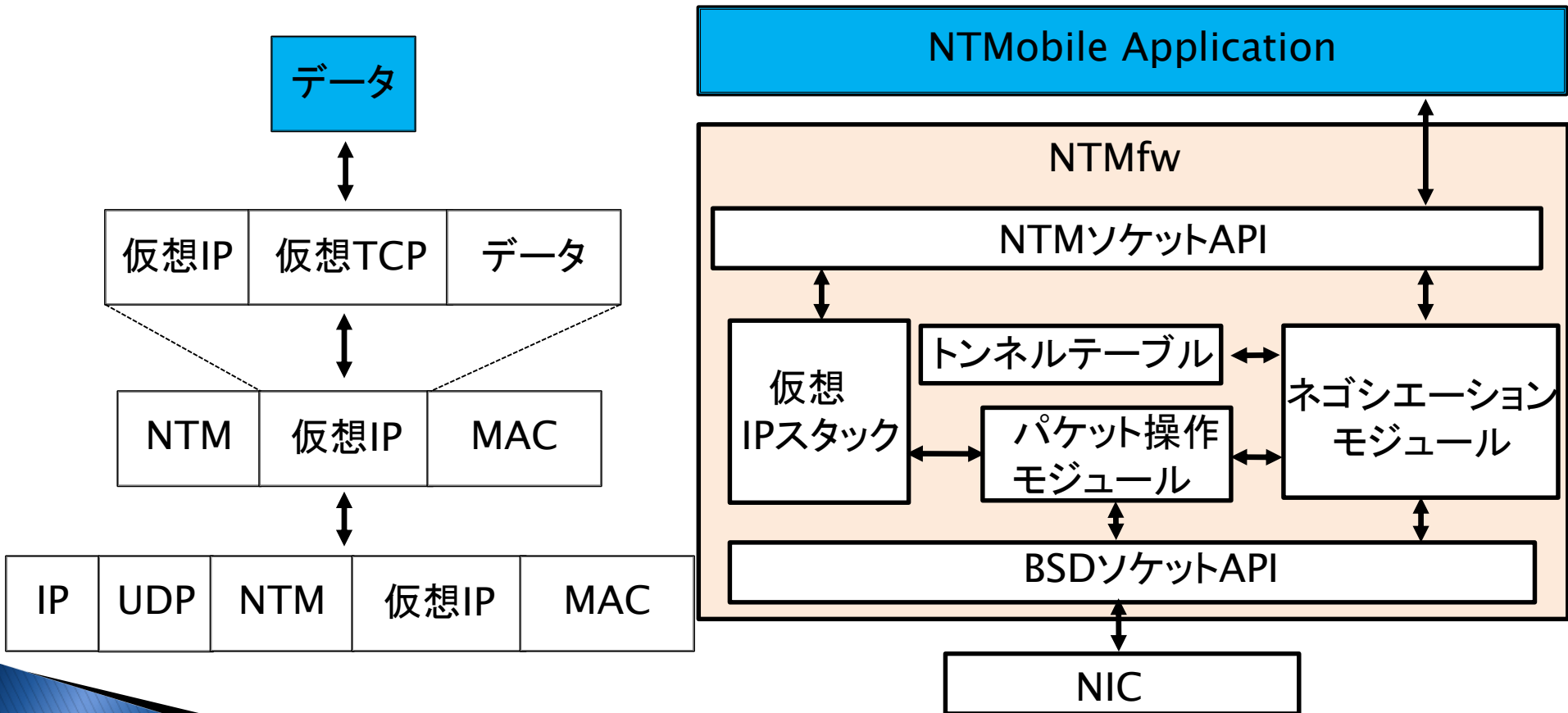
# NTMAの実現方法

1. NTMobileの仮想IPパケット生成処理に着目
2. 仮想IPヘッダ生成フェーズをNTMfwの機能から分離して一般端末で代行

# NTMfwのパケット遷移

## NTMobileアプリケーション

- NTMソケットAPIで記述されたアプリケーション

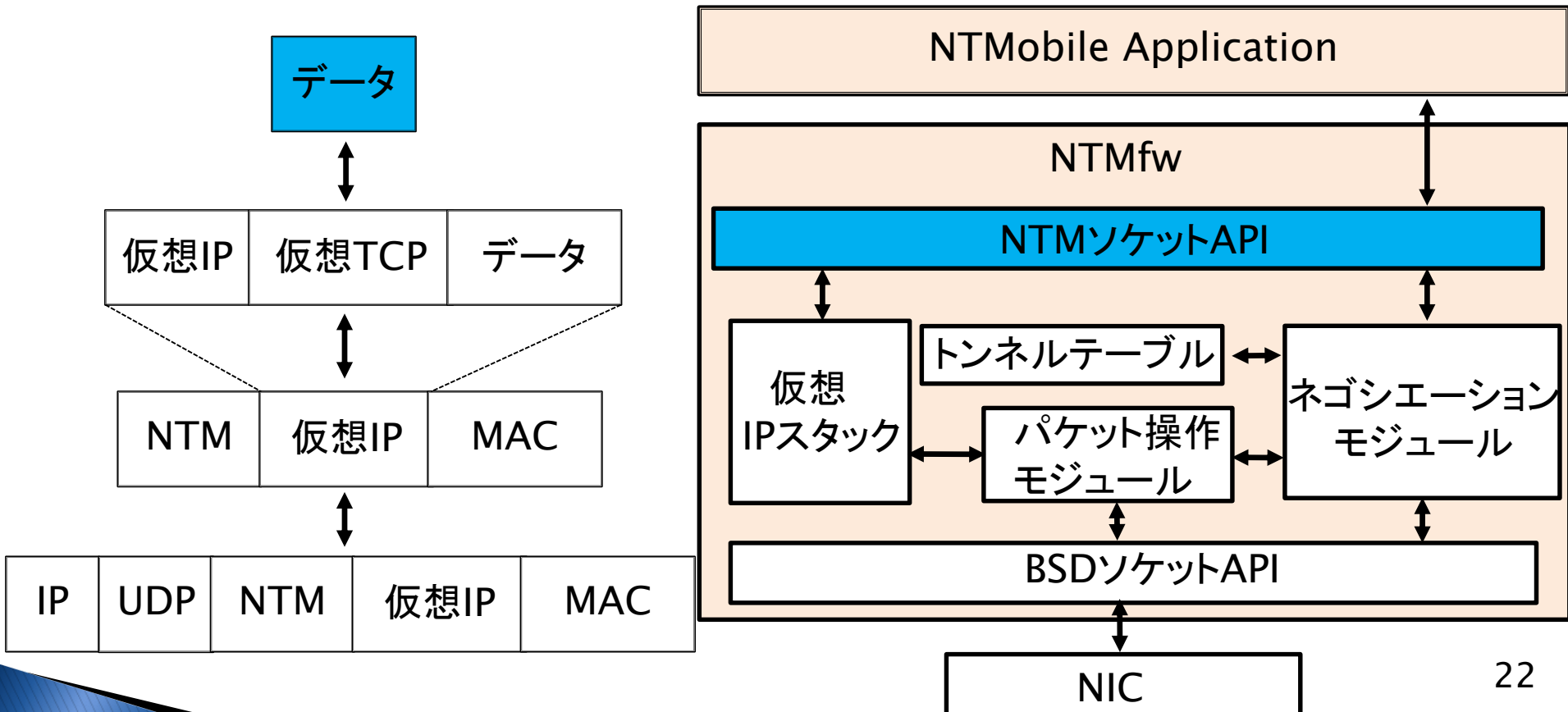




# NTMfwのパケット遷移

## NTMソケットAPI

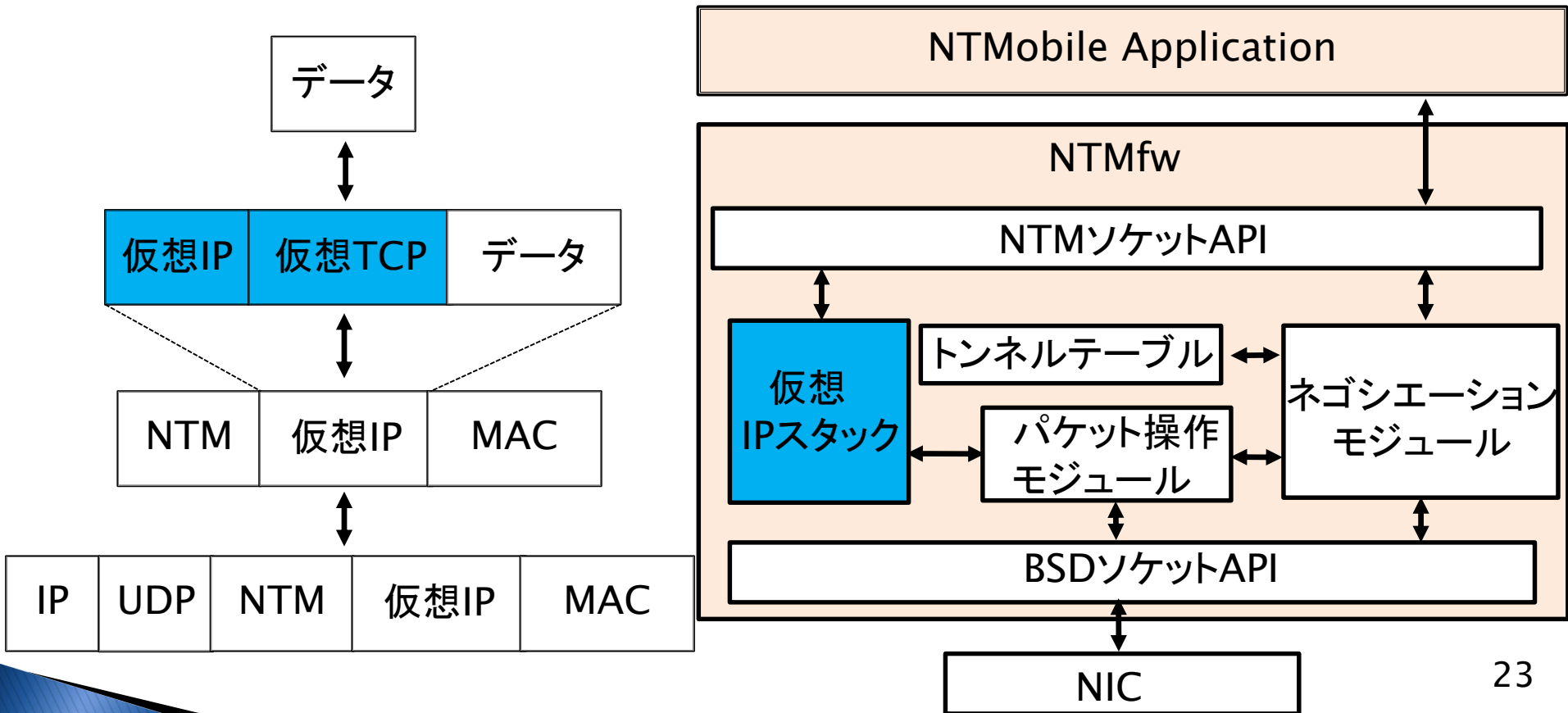
- BSDソケットAPI互換のソケットAPI
- アプリケーションデータを仮想IPスタックへ渡す



# NTMfwのパケット遷移

## 仮想IPスタック

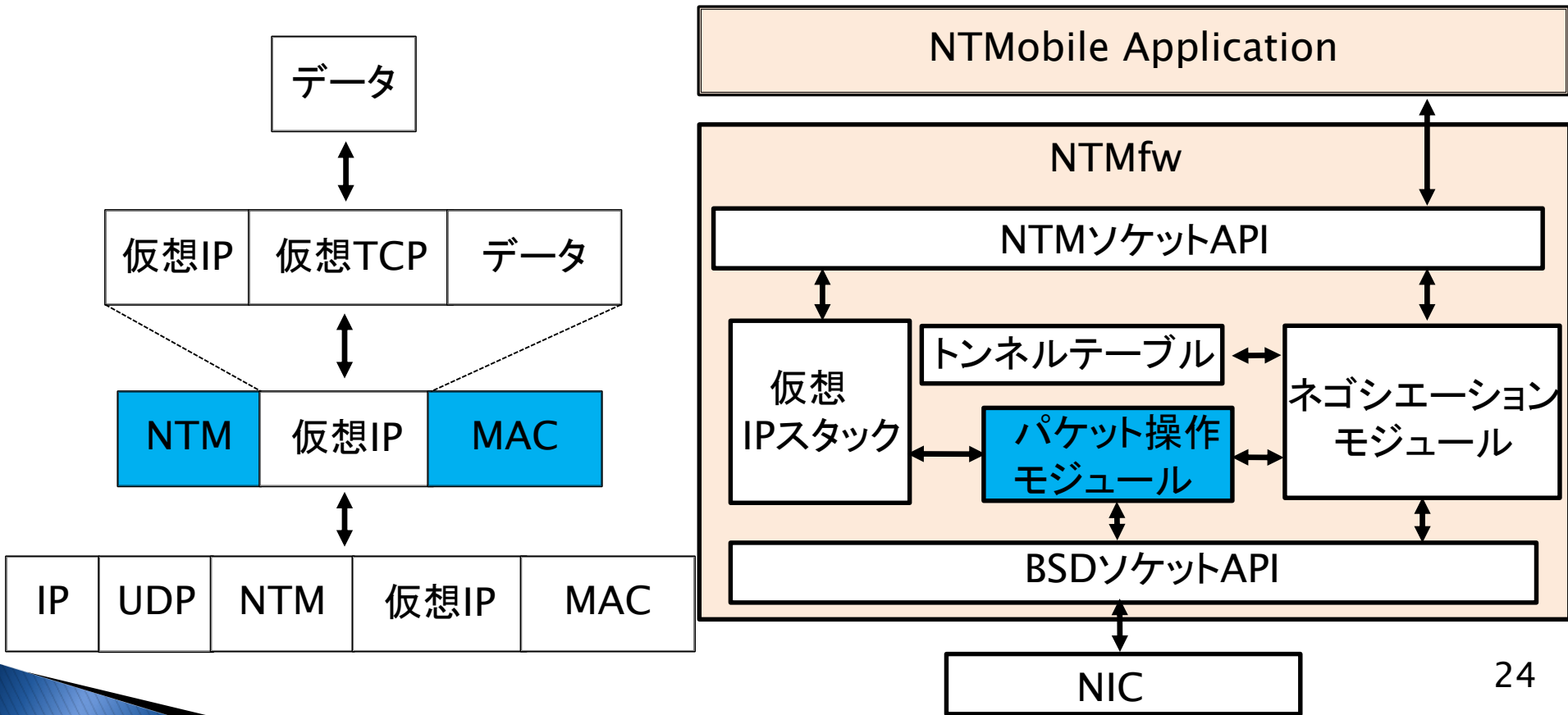
- アプリケーションデータから仮想TCP/IPパケットを生成



# NTMfwのパケット遷移

## パケット操作モジュール

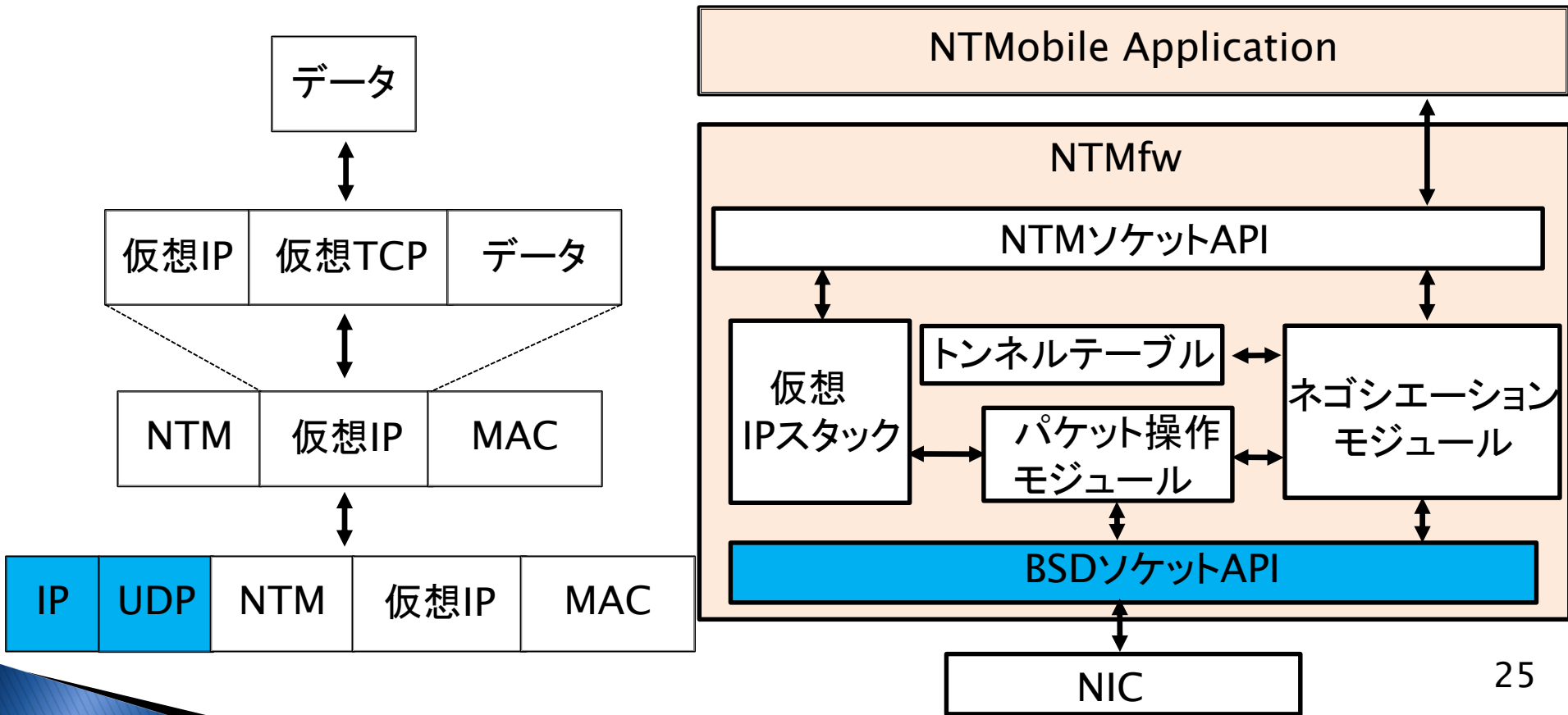
- NTMヘッダの付与
- MAC付与/検証等



# NTMfwのパケット遷移

## BSDソケットAPI

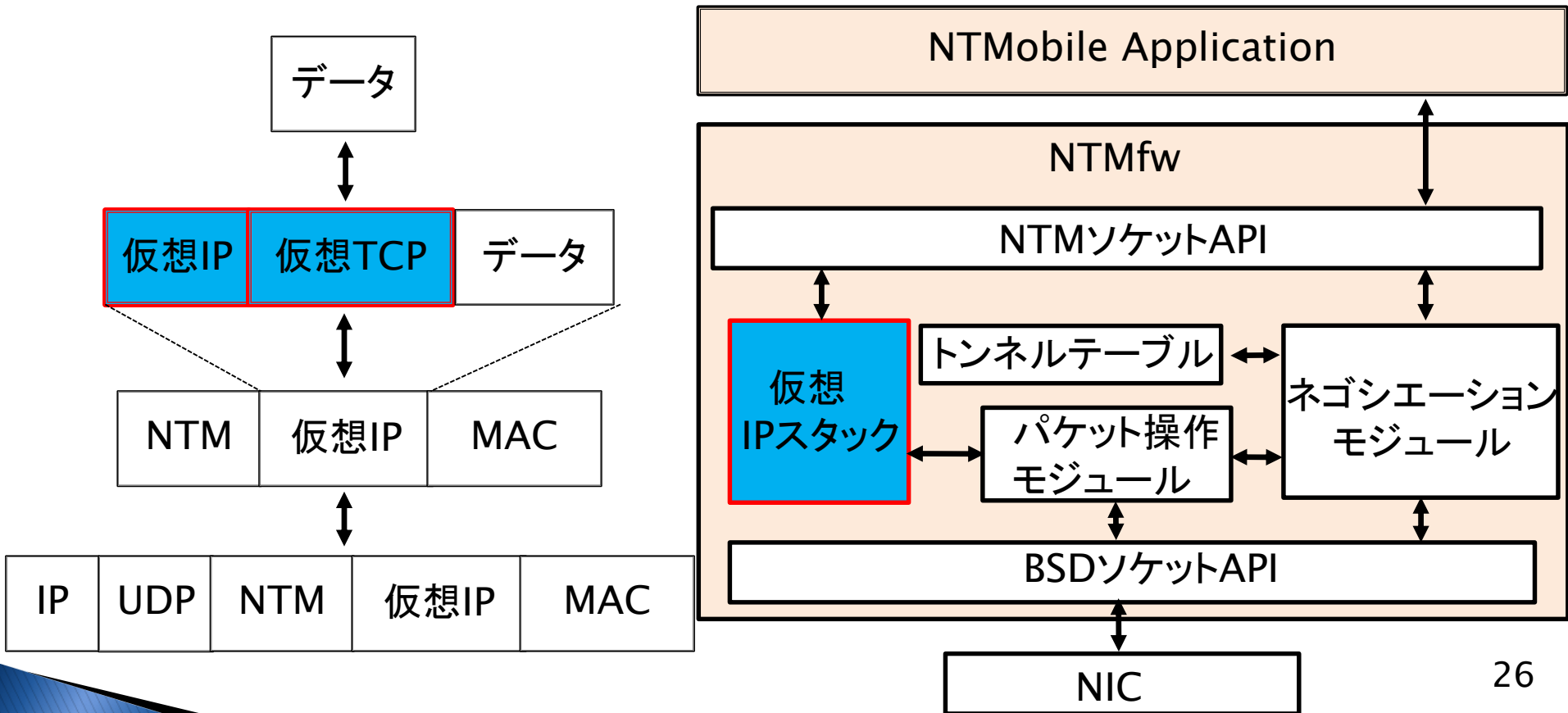
- C言語のソケットAPI
- データパケットの送受信(カプセル化/デカプセル化)



# NTMfwのパケット遷移

## 仮想IPスタック

- アプリケーションデータから仮想TCP/IPパケットを生成



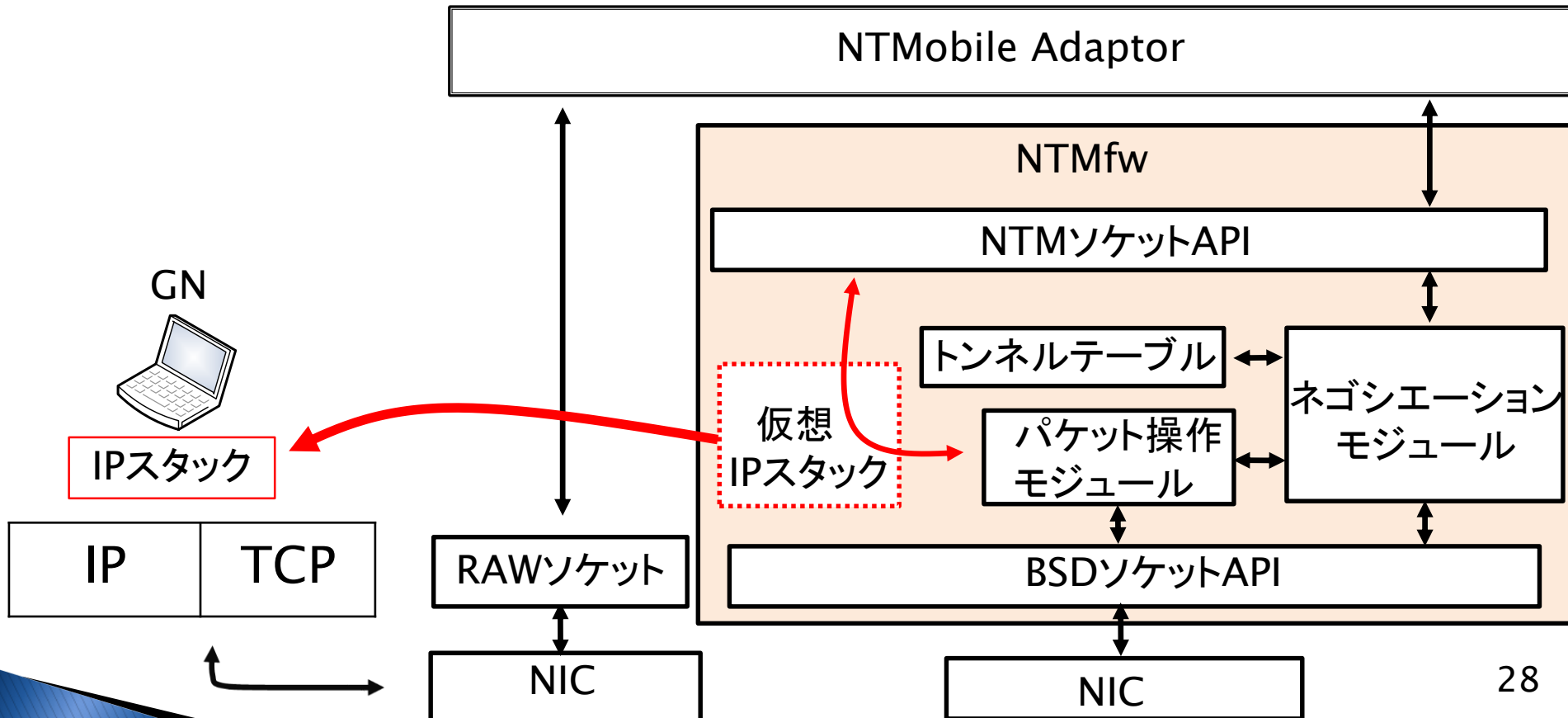
# NTMAの実現方法

1. NTMobileの仮想IPパケット生成処理に着目
2. 仮想IPヘッダ生成フェーズをNTMfwの機能から分離して一般端末で代行

# NTMAのモジュール構成

## NTMfwの変更点

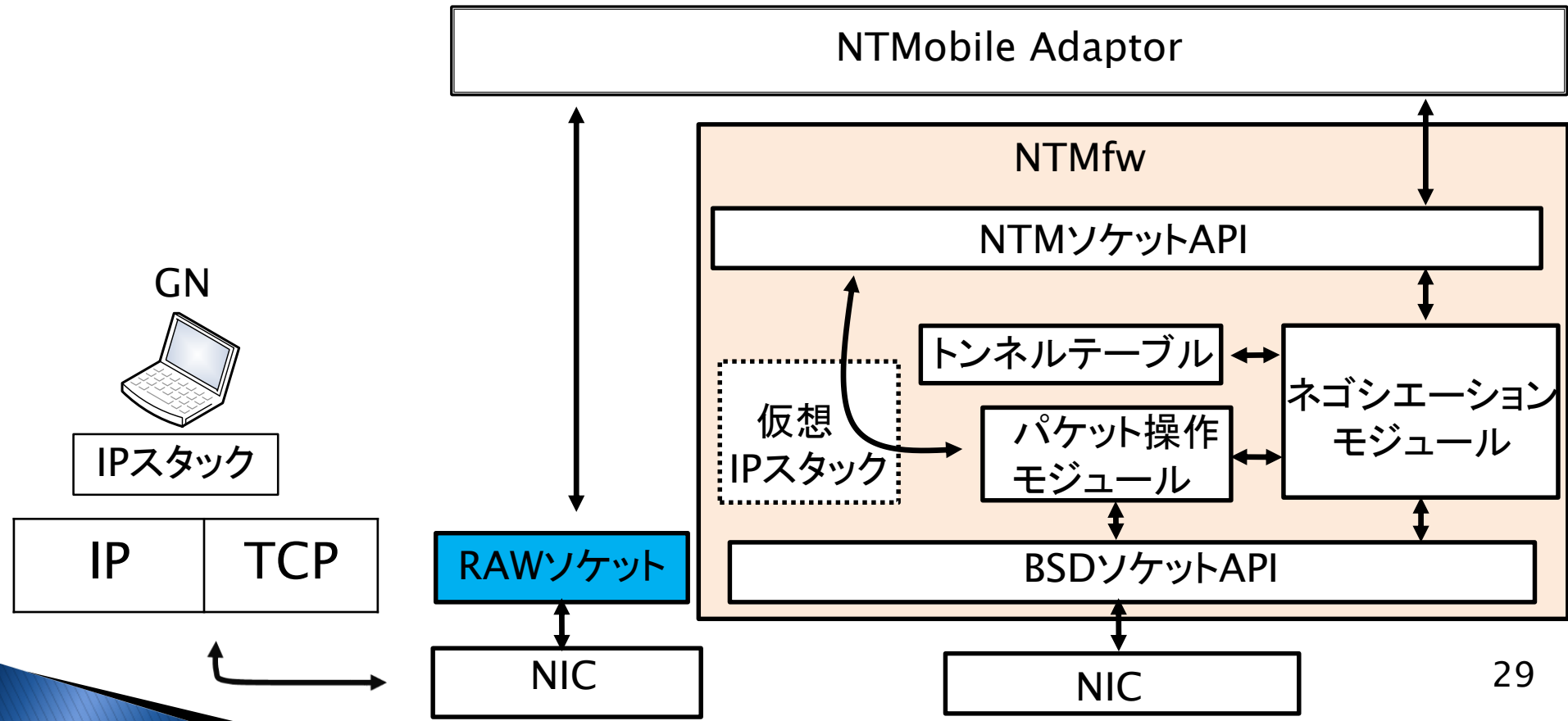
- NTMfwから仮想IPスタックの処理をスキップ
- 仮想IPスタックの処理をGNで代行



# NTMAのモジュール構成

## RAWソケット

- 生のパケットをダイレクトに送受信
- 自インターフェース宛以外のパケットも送受信できるように設定

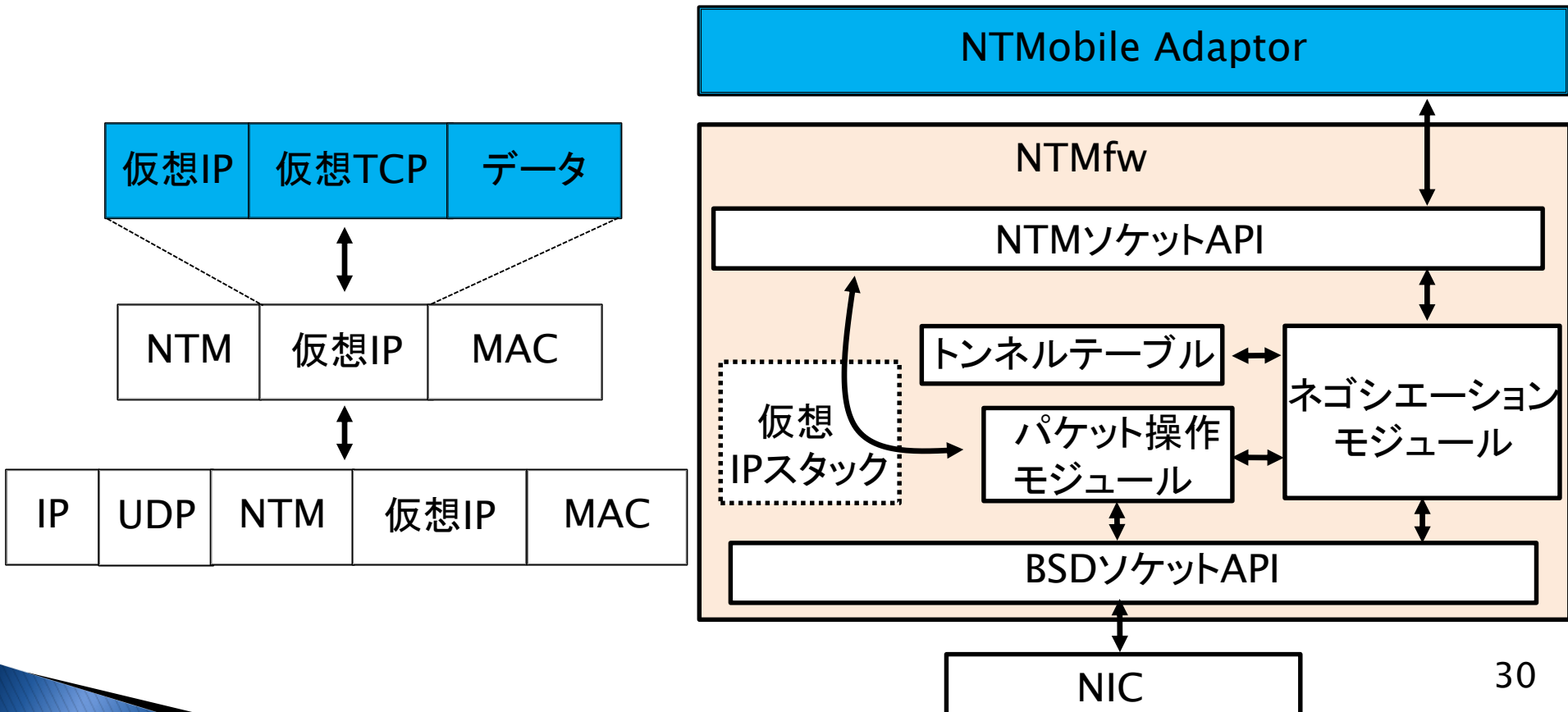




# NTMAのパケット遷移

## NTMAアプリケーション

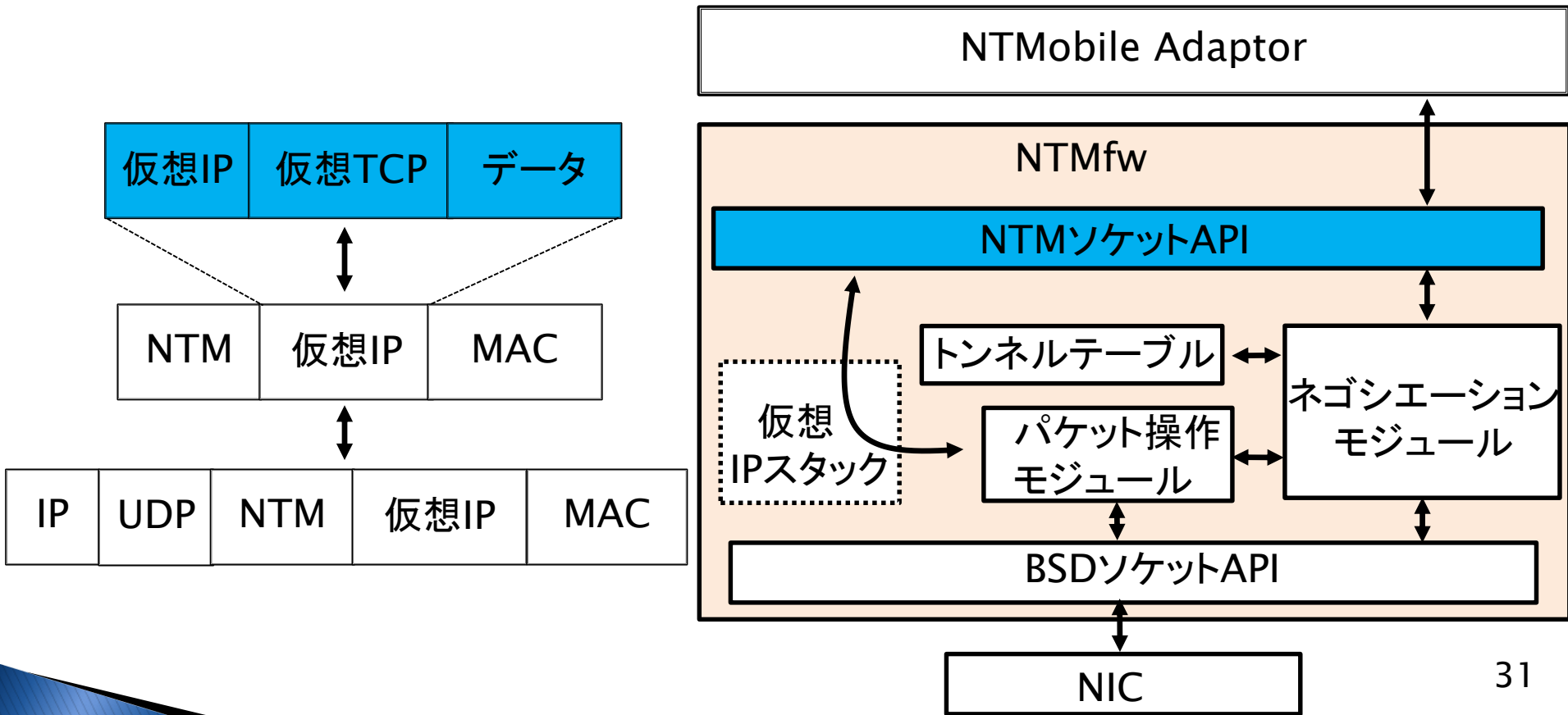
- GNとの通信は, RAWソケットで送受信
- NTM端末との通信は, 仮想IPスタックを**スキップ**するNTMソケットAPIで送受信



# NTMAのパケット遷移

## NTMfwソケットAPI

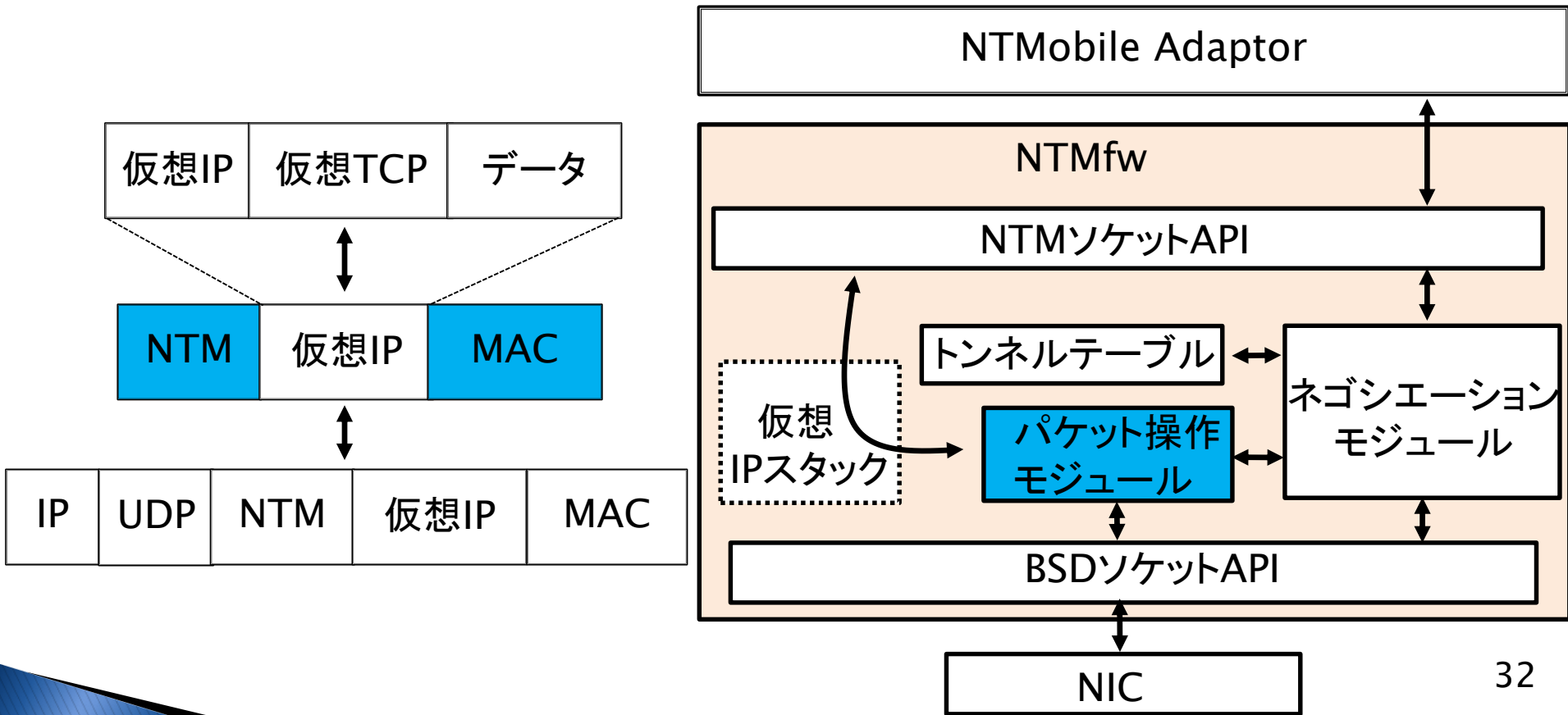
- 仮想IPスタックの処理をスキップするNTMfwのAPI
- パケット操作モジュールへ処理を渡す



# NTMAのパケット遷移

## パケット操作モジュール

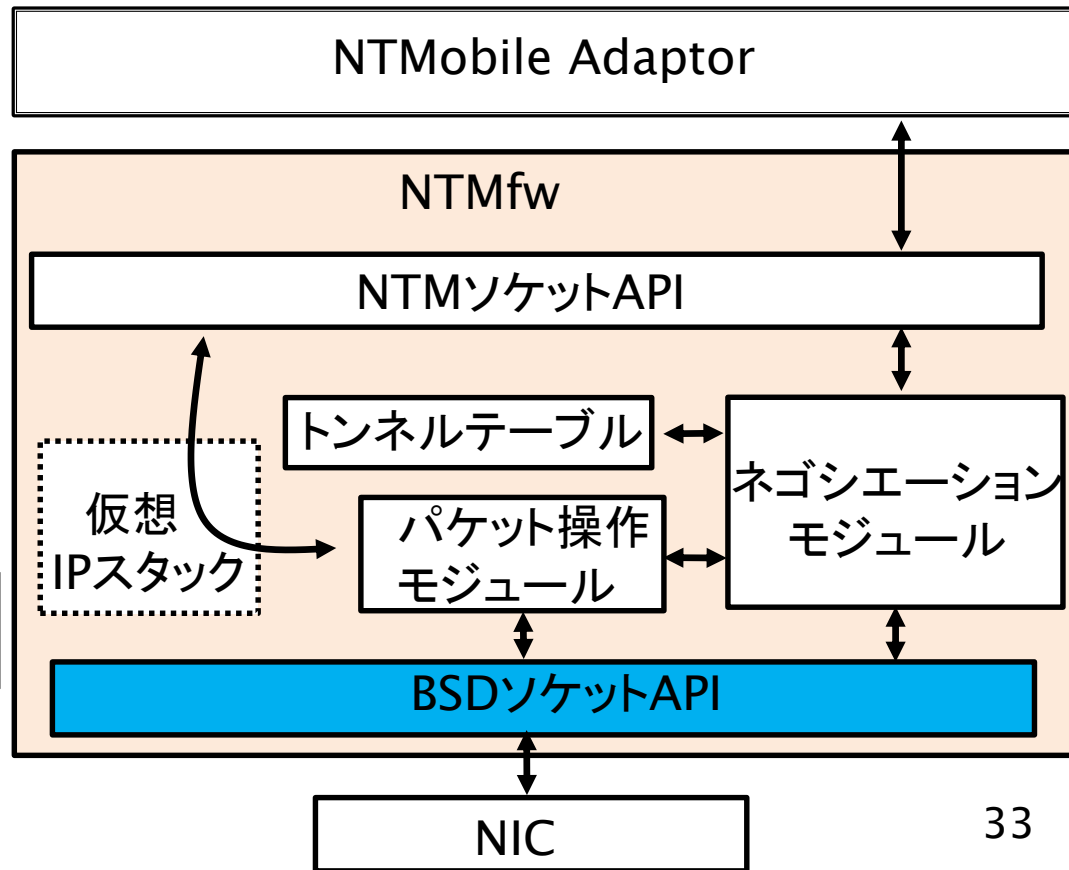
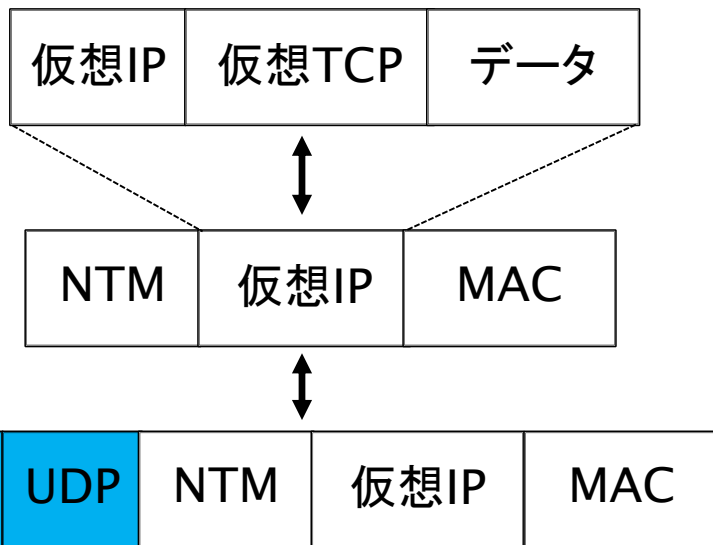
- NTMfwと同様の処理



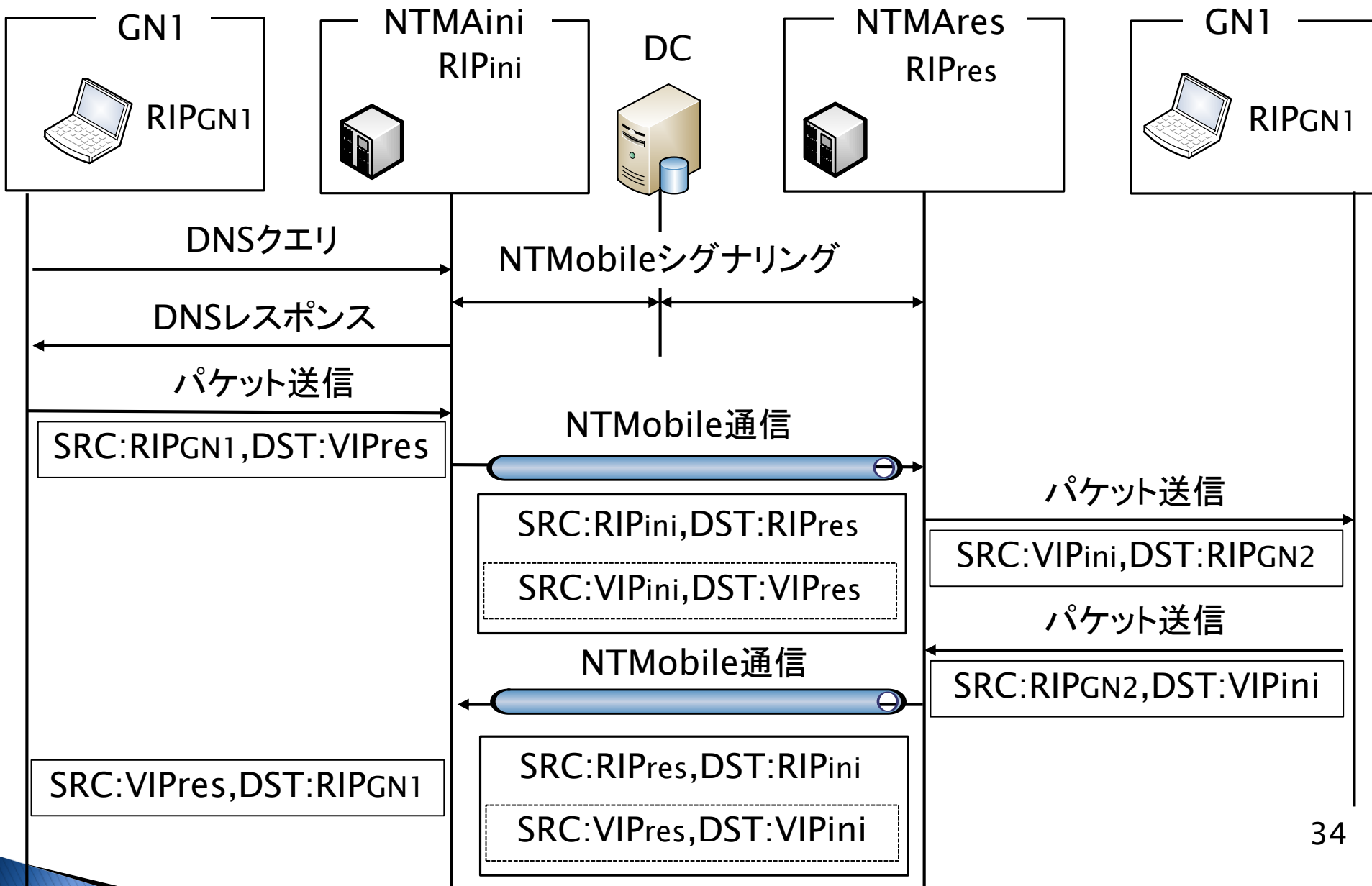
# NTMAのパケット遷移

## BSDソケットAPI

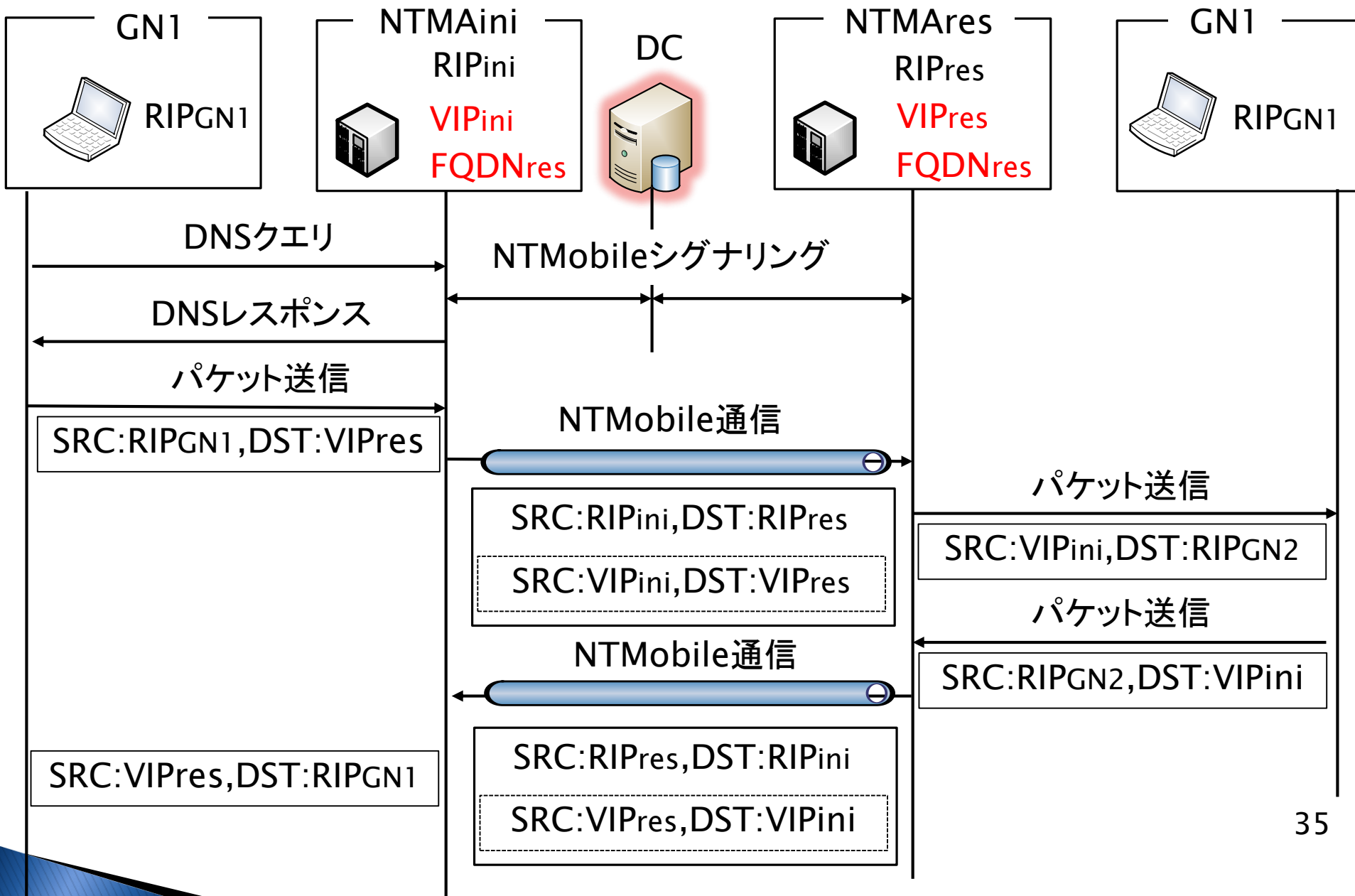
- NTMfwと同様の処理



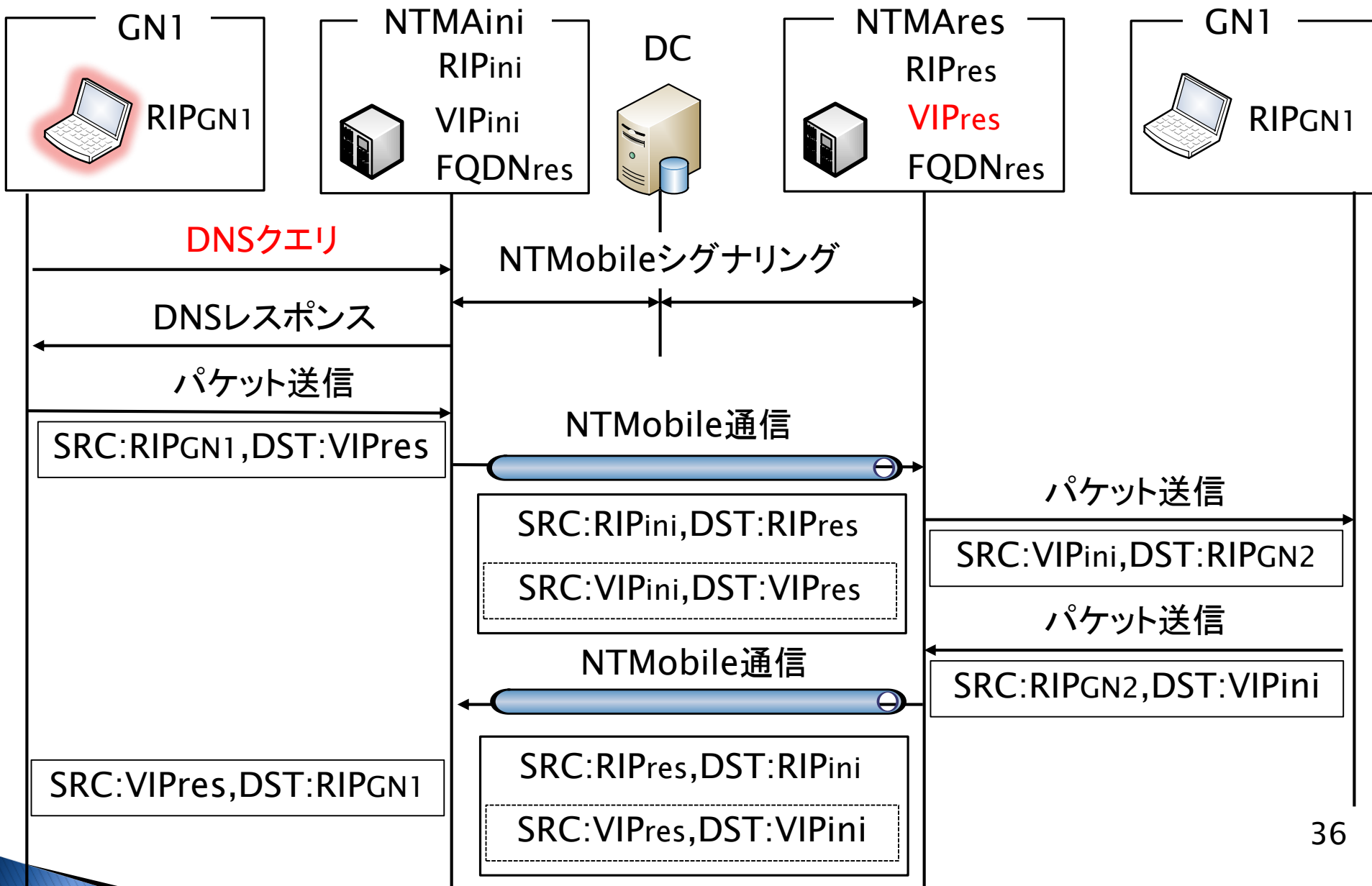
# NTMAの動作シーケンス



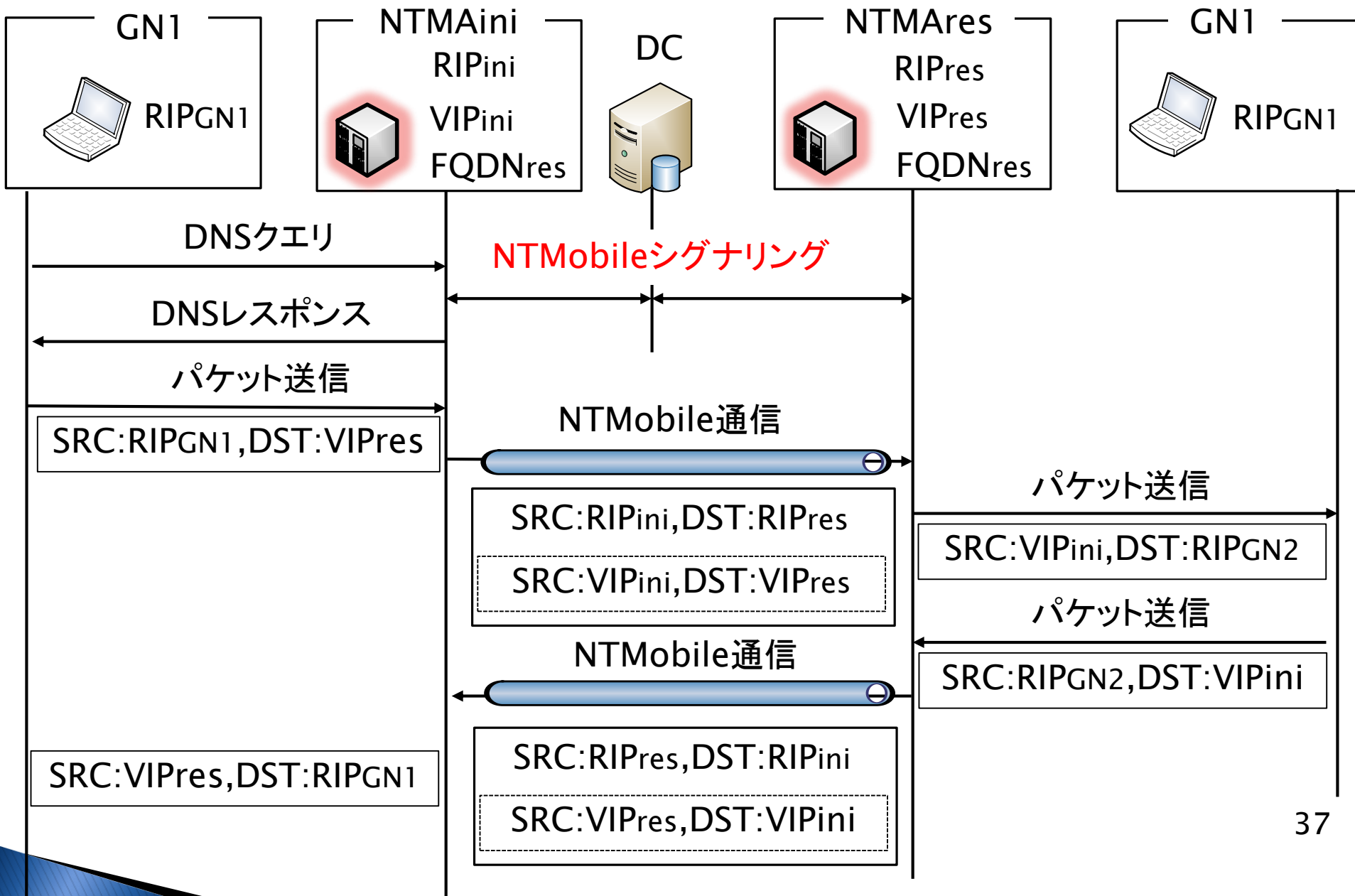
# NTMAの動作シーケンス



# NTMAの動作シーケンス

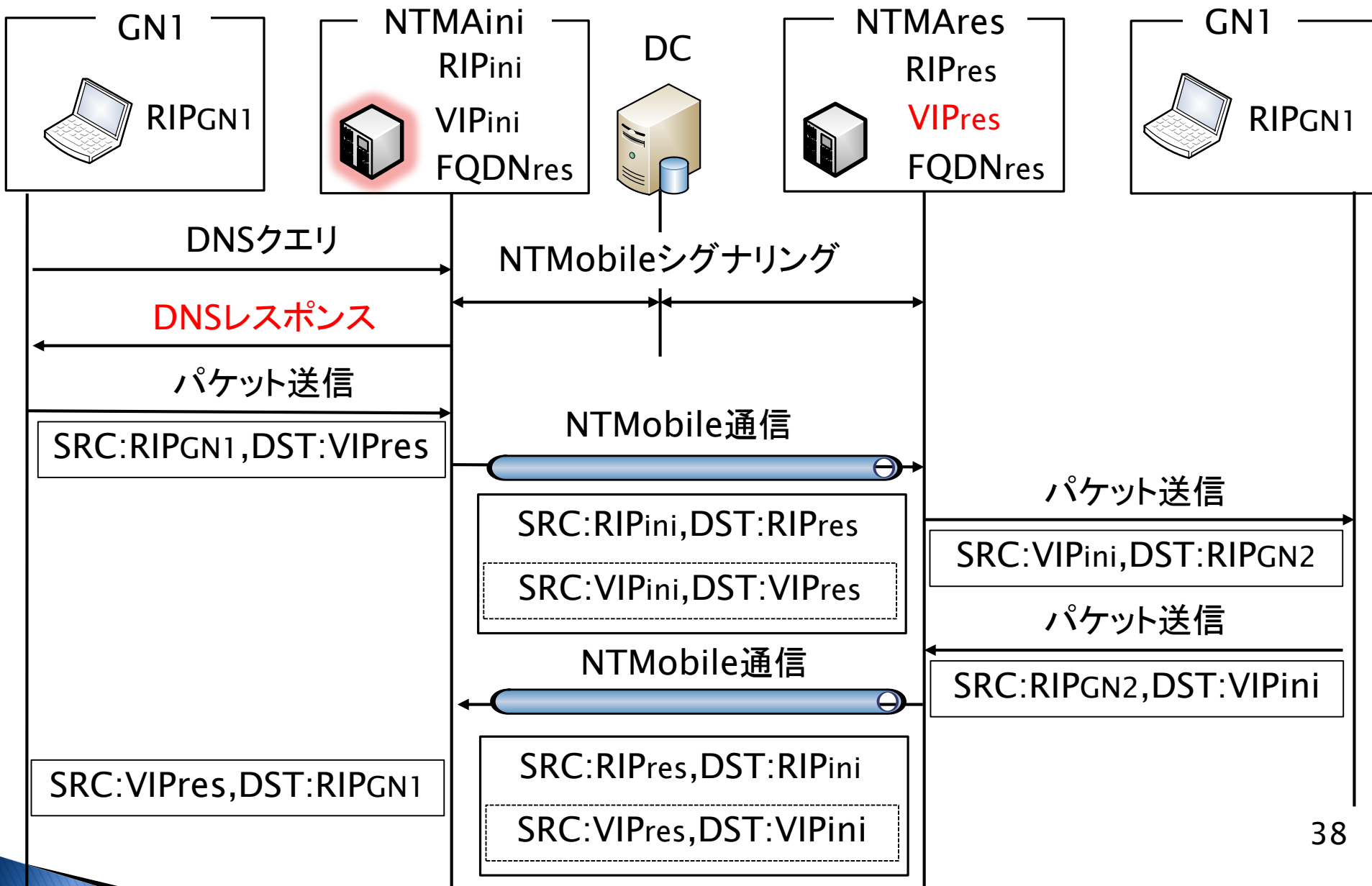


# NTMAの動作シーケンス

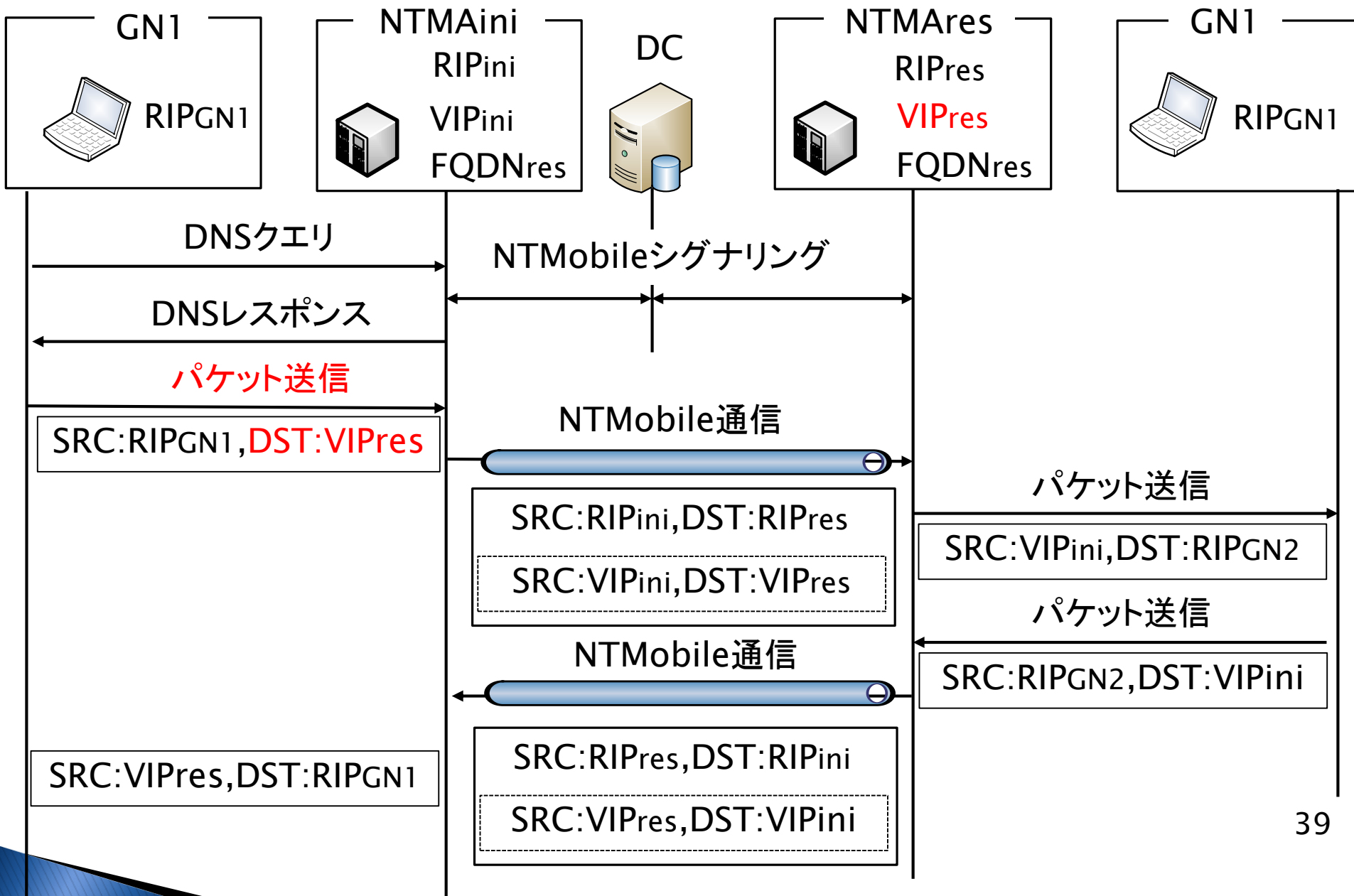




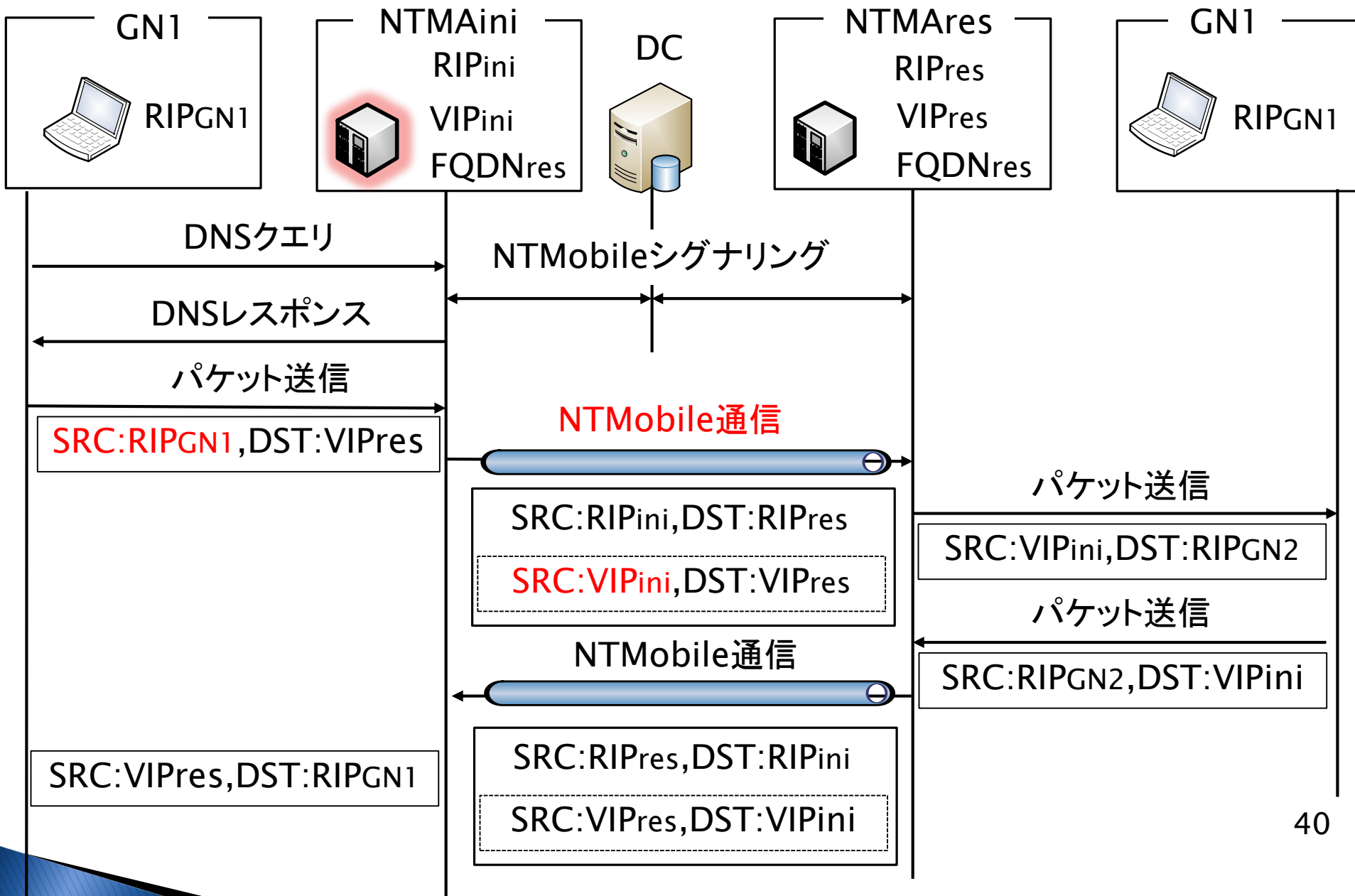
# NTMAの動作シーケンス



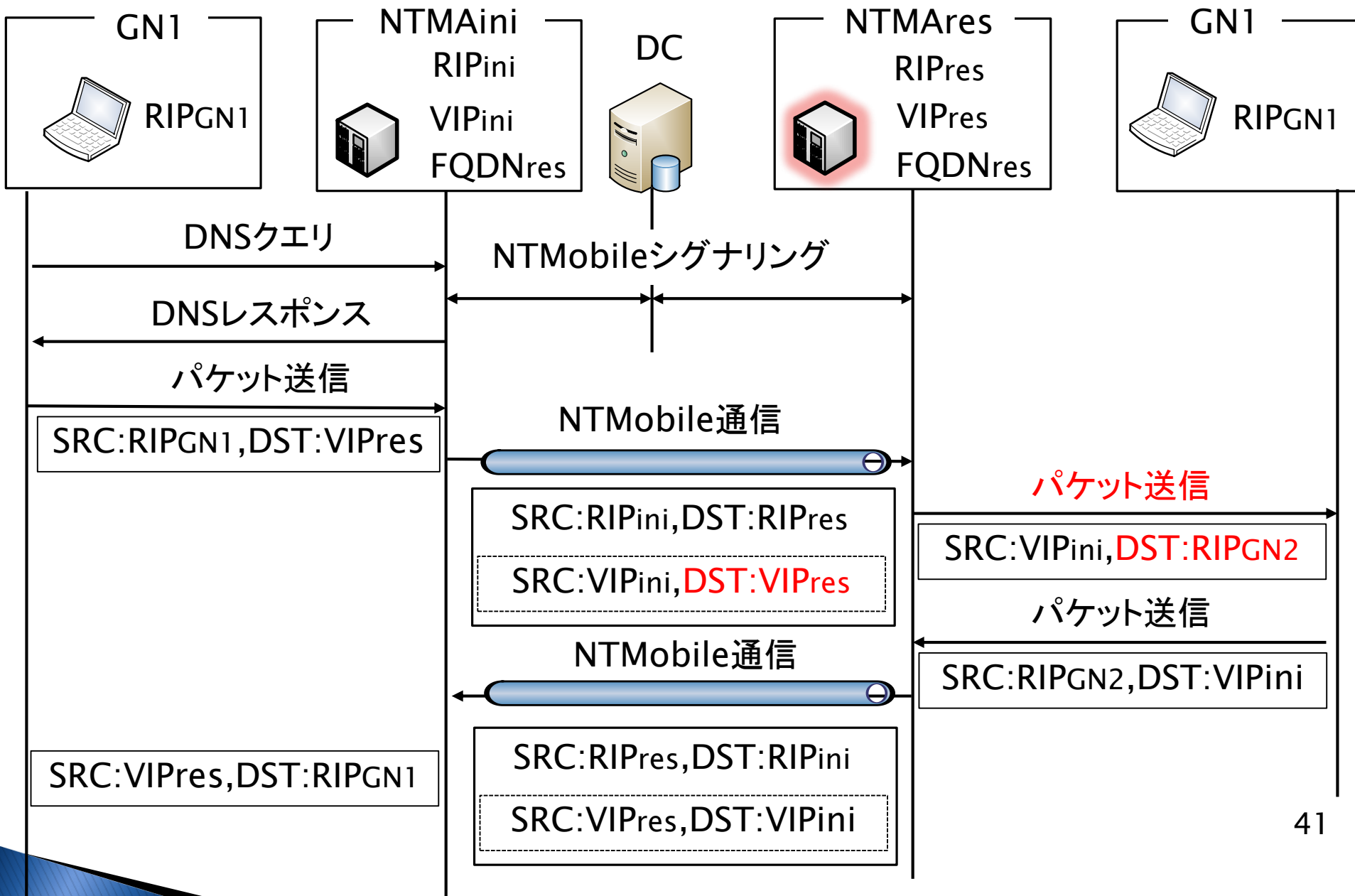
# NTMAの動作シーケンス



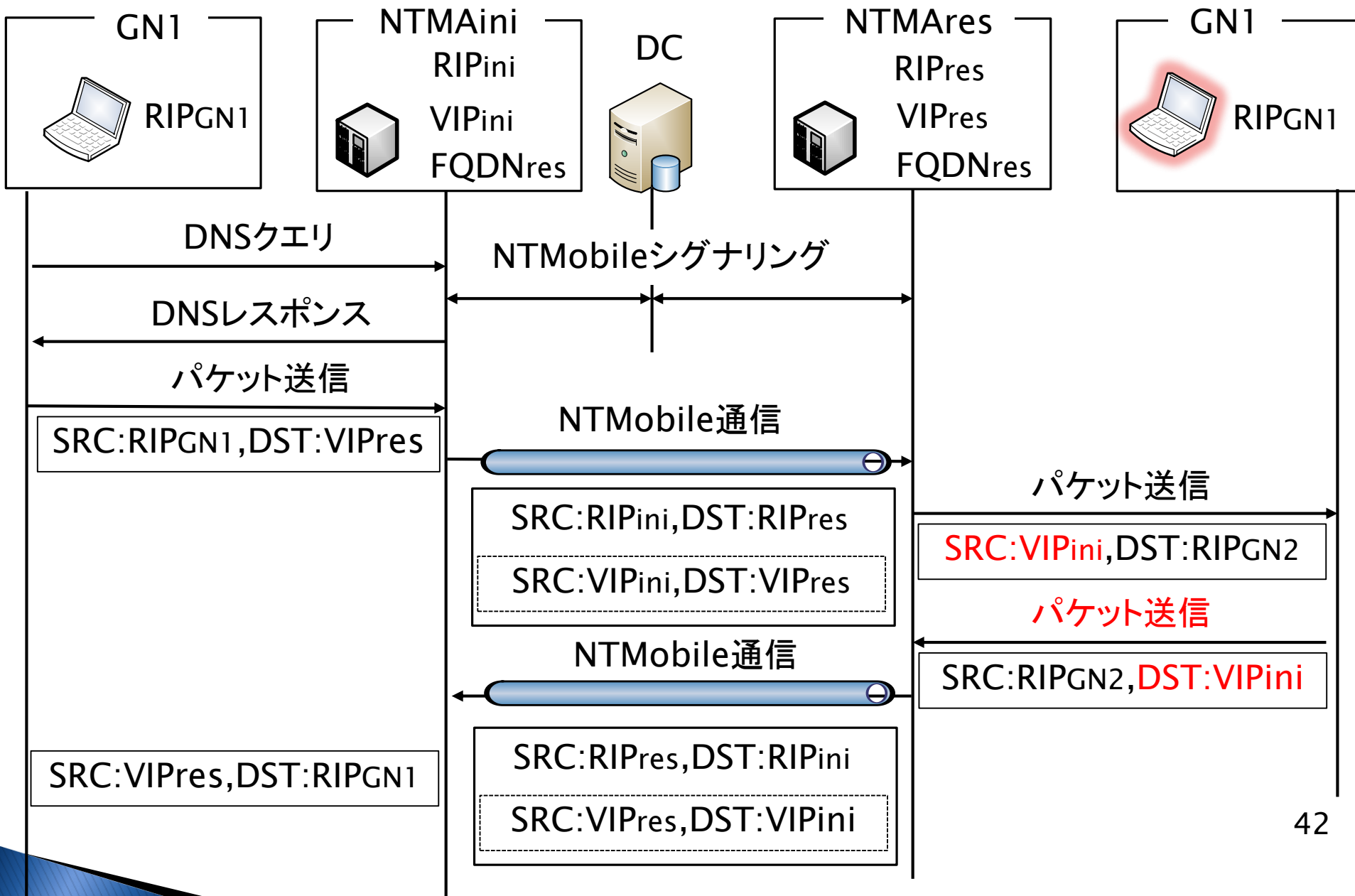
# NTMAの動作シーケンス



# NTMAの動作シーケンス

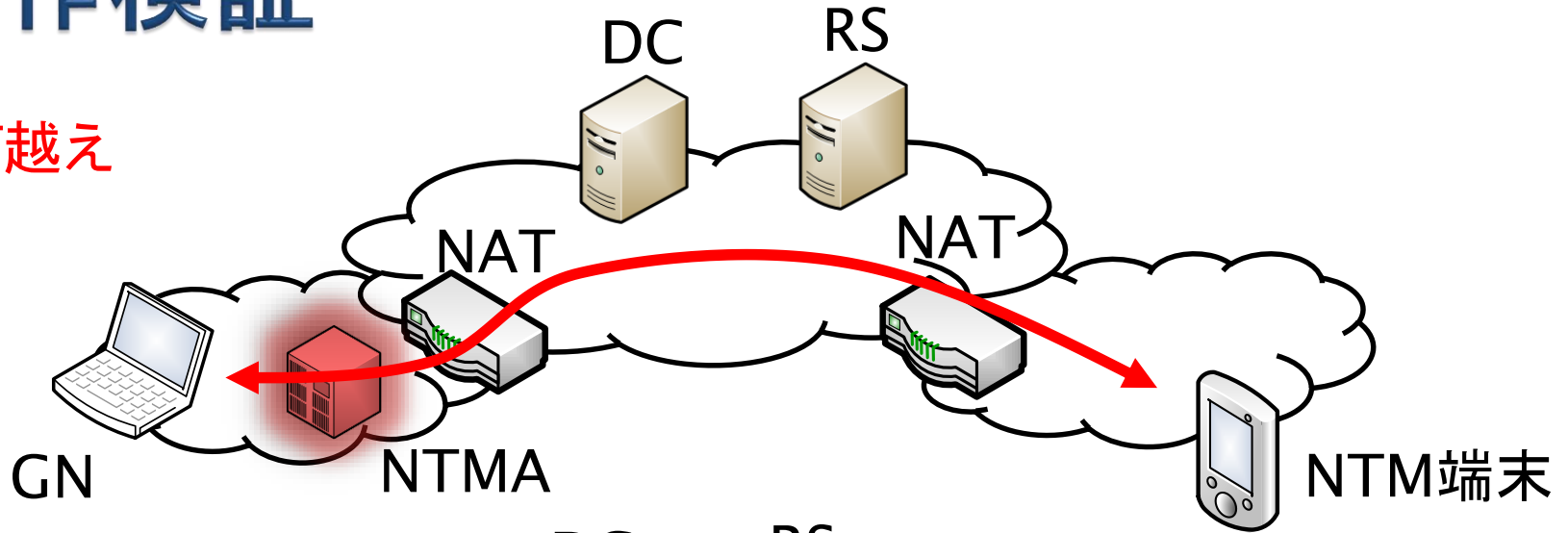


# NTMAの動作シーケンス

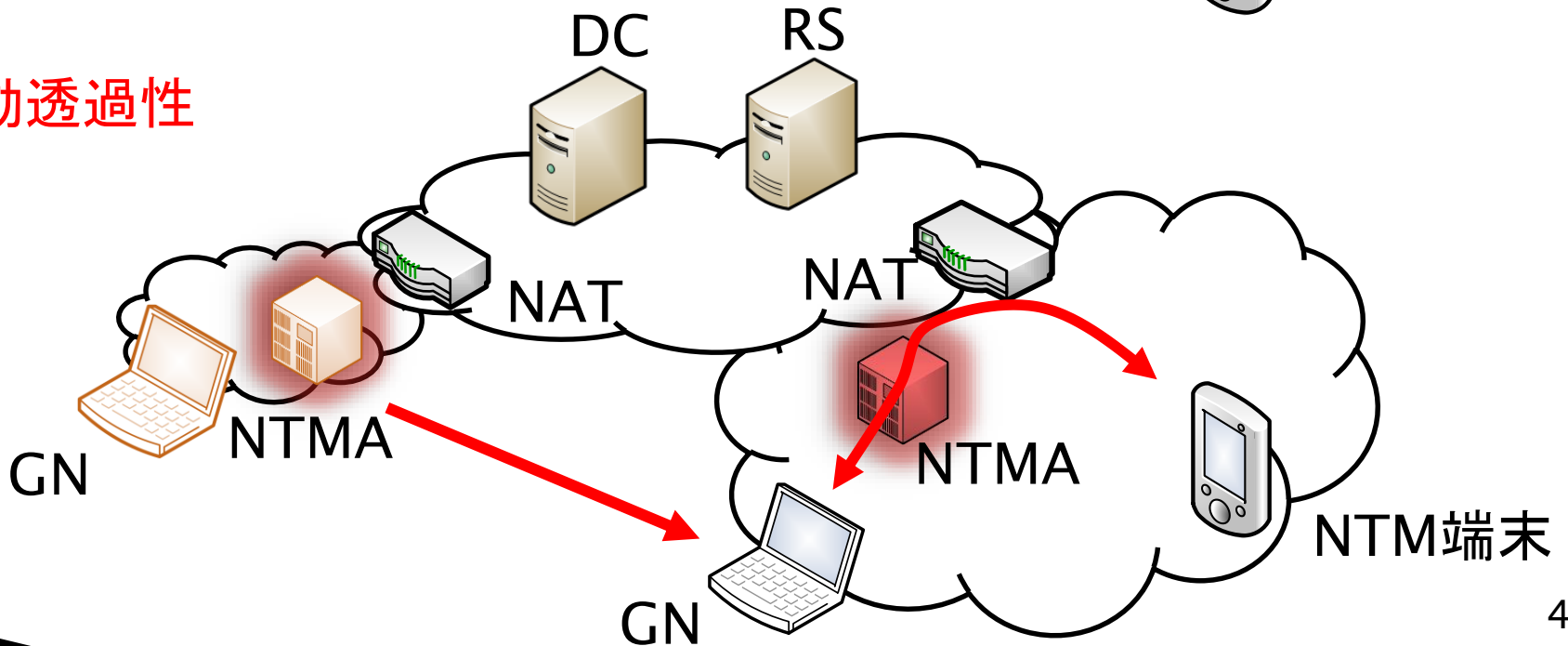


# 動作検証

NAT越え

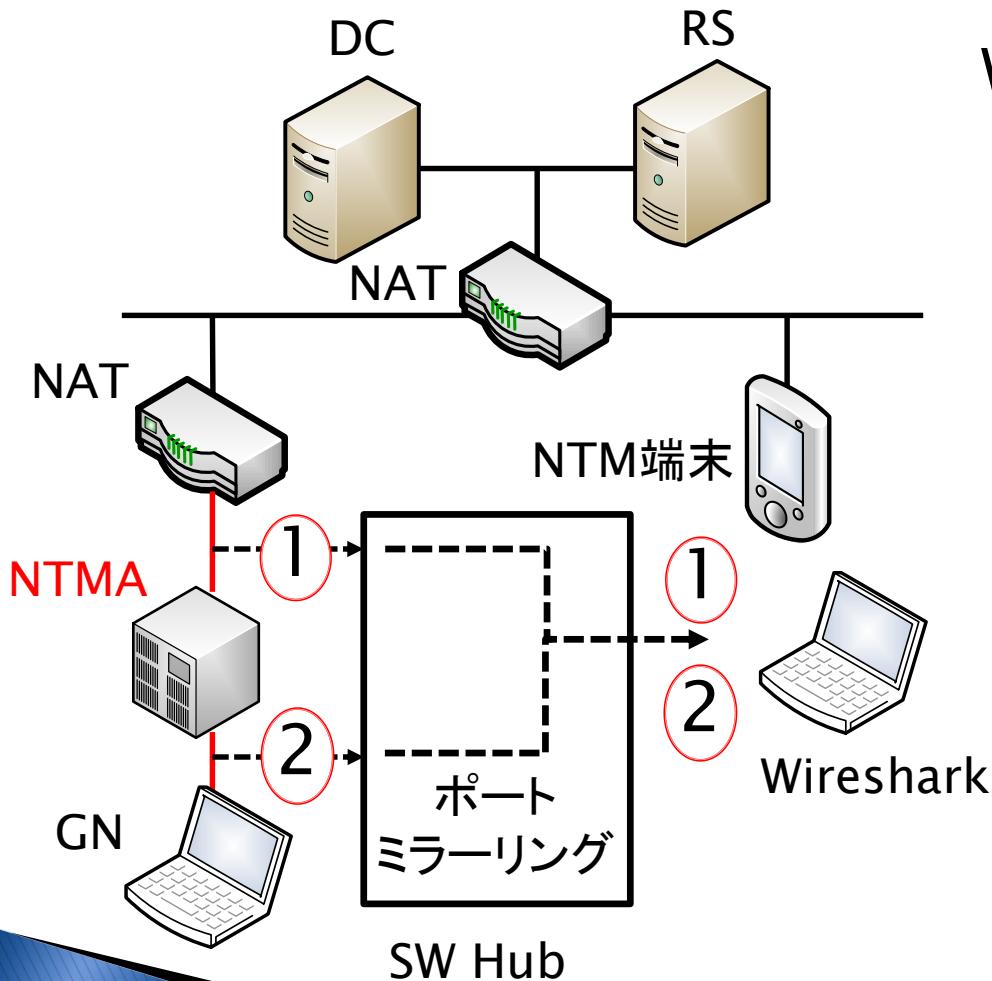


移動透過性



# 評価構成

NTMAの packets 中継に要する時間



Wiresharkで ① ② の時間取得

NTMA  
中継時間 | ① — ② |

評価諸元

マイコン	Raspberry Pi3
OS	Raspbian
Memory	1 GB
CPU	1.2 Ghz
ネットワーク	1000BASE-T

# 性能評価

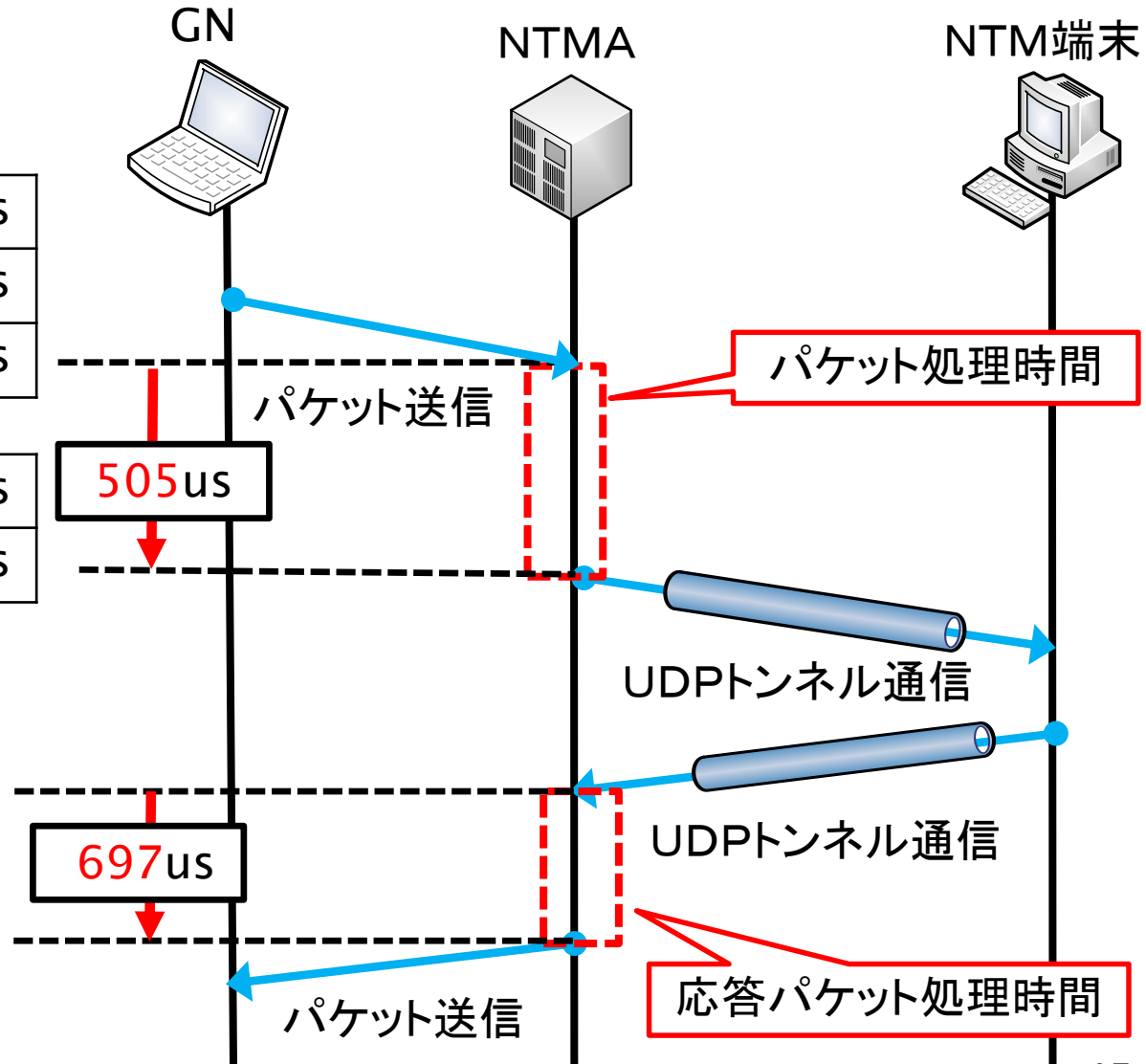
10回の平均値

パケット処理時間	505us
応答パケット処理時間	697us
SW Hub処理時間	358us

パケット処理時間	147us
応答パケット処理時間	339us

\* NTMobileの暗号化  
などの処理時間も含む

SW Hubとの処理時間を  
比較しても非常に高速





# まとめ

- ◆ NTMobileアダプタの実現方式の検討
  - 一般端末にNTMAを設置
    - 一般通信をNTMobile通信に変換
    - 一般端末のプログラムに手を加えない
  
- ◆ 今後の方針
  - NTMAの実装の完了