

# NTMobile アダプタの実現方式の検討

尾久 史弥<sup>†1</sup> 納堂 博史<sup>†2</sup> 鈴木 秀和<sup>†2</sup> 内藤 克浩<sup>†3</sup> 渡邊 晃<sup>†2</sup>  
<sup>†1</sup> 名城大学理工学部 <sup>†2</sup> 名城大学大学院理工学研究科 <sup>†3</sup> 愛知工業大学情報科学部

## 1 はじめに

現状のネットワークは、グローバル IP アドレスの枯渇から、NAT を利用してプライベートアドレスによるネットワークを構築するのが一般的である。しかし、グローバル側の端末からプライベート側の端末に対して通信を開始できない問題がある (NAT 越え問題)。

この問題を解決するための技術として、筆者らは NT-Mobile を提案している [1]。NTMobile は、NTMobile framework (NTMfw) と呼ばれるアプリケーションをエンド端末に組み込むことにより、NAT の有無に関わらず双方向通信を実現することができる。しかし、組み込み型家電や安定性を重視するサーバなどは新たな機能の組み込みができない場合がある。

そこで一般端末に隣接設置し、一般端末の通信を NT-Mobile 通信に変換するアダプタ (以下 NTMA) の実現方式について検討した。一般端末と NTMA との間の通信は、RAW ソケットで実現し、NTMA と NTM 端末との間の通信は、NTMfw を改造することにより実現する。本稿では、NTMA をイニシエータ側の一般端末に設置した場合の動作について確認したので記述する。

## 2 NTMobile

NTMobile は、NTMfw を実装した端末 (NTM 端末)、NTM 端末の実 IP アドレスと仮想 IP アドレスの管理、及び通信経路を指示する DC (Direction Coordinator) によって構成される。NTM 端末は、起動時に DC に対して実 IP アドレスを登録すると共に、位置に依存しない仮想 IP アドレスの割り当てを受ける。NTM 端末のアプリケーションは、仮想 IP アドレスに基づいて通信を行う。実際の通信は、実 IP アドレスで全パケットをカプセル化し、UDP トンネルによる通信を行う。DC が適切な通信経路を両 NTM 端末に指示することで、通信経路上に NAT が存在しても常に双方向通信が実現できる。

## 3 NTMfw

NTMfw は、アプリケーション層で動作する通信ライブラリであり、上位アプリケーションに対して、NT-

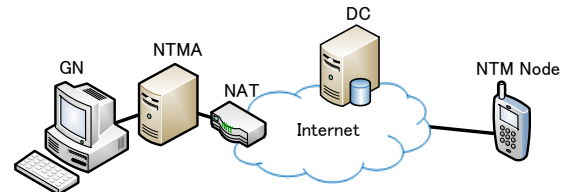


図1 提案システムのネットワーク構成

Mobile 通信のソケットインターフェースを提供する。NTMfw において NTMobile 通信を実現する方法は以下の通りである。ユーザがデータを送信する場合は、ユーザデータを NTMfw が lwIP (lightweight IP) の機能によって、送信元/宛先アドレスが仮想 IP アドレスの IP パケットを生成し、さらに NTM ヘッダを追加する。このパケットをデータとみなして、Linux が提供する UDP ソケットによりカプセル化して送信する。また、ユーザがデータを受信する場合は、受信したパケットを Linux が提供する UDP ソケットによりデカプセル化した後、NTMfw によって NTM ヘッダを除去し、さらに lwIP によってユーザデータを上位アプリケーションに渡す。

## 4 NTMA

### 4.1 ネットワーク構成

図1に提案システムのネットワーク構成を示す。NTMA は物理 NIC を2枚用意して、一方は NAT 配下の一般端末 GN とブリッジ接続し、他方はインターネット側とブリッジ接続する。DC と GN の通信相手の NTM 端末は、インターネット上に設置する。NTM 端末と NTMA は、起動時に DC に対して実 IP アドレスの登録および仮想 IP アドレスの割り当てを受ける。NTM 端末と NTMA は、DC に対して定期的に Keep Alive を行い、いつでも DC からの経路指示を受けることができる。GN の IP アドレスには、NTMA が DC から割り当てられた仮想 IP アドレスを設定する。

### 4.2 動作シーケンス

図2に提案システムの通信シーケンスを示す。通信開始時に、GN は相手 NTM 端末の FQDN を指定して DNS クエリを送信する。DNS クエリを受け取った NTMA は、NTMfw の機能を利用して、NTMobile シグナリング処理により、DC の指示に従って NTMA と

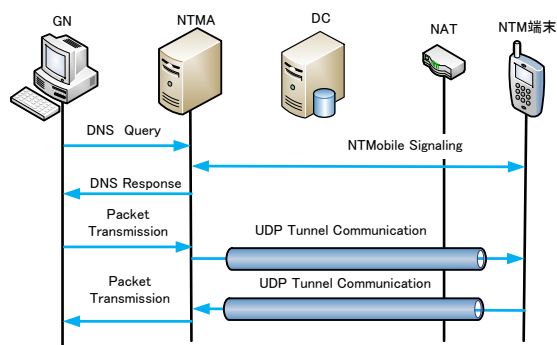


図2 提案システムの通信シーケンス

NTM 端末との間でトンネル構築を行う。トンネル構築終了後、NTMA は相手 NTM 端末の仮想 IP アドレスを取得するので、このアドレスを DNS レスポンスにのせて GN へ送信する。次に GN は、相手 NTM 端末の仮想 IP アドレス宛てにパケットを送信する。このパケットを受信した NTMA は、NTMfw の機能を利用して NTMA と相手 NTM 端末の実 IP アドレスでカプセル化してデータパケットを送信する。相手 NTM 端末からパケットが返信されてきた場合は、パケットを受信した NTMA が NTMfw によりデカプセル化を行い一般通信のパケットに変換してから、GN へ送信する。

## 5 実装

### 5.1 モジュール構成

図3に NTMA(イニシエータ側)のモジュール構成を示す。NTMA と相手 NTM 端末との間の通信は、NTMfw が提供する NTMobile 通信で実現する。また、NTMA と GN との間の通信は、RAW ソケットで実現する。Init モジュールは、DC に対して実 IP アドレスの登録を行うとともに、仮想 IP アドレスの割り当てを受ける。また、Init モジュールは、DHCP サーバの機能を包含し、GN からの IP アドレス要求に対して上記仮想 IP アドレスを割り当てる。イニシエータモジュールは、GN から送信されたパケットを RAW ソケットで受信する。受信したパケットが DNS クエリであった場合、相手 NTM 端末との間で NTMobile シグナリングによりトンネル構築処理を行う。また、トンネル構築終了後に、DNS クエリの返信を GN に送信する。GN 側から受信したパケットが UDP/TCP パケットであった場合は、改造した NTMfw (R-NTMfw : Remodeled-NTMfw) の API を介してトンネル通信でパケットを送信する。GN は、NTM 端末の仮想 IP アドレスをターゲットとする ARP リクエストを送信することがあるので、NTMA は自らの MAC アドレスで ARP リプライを返信する。

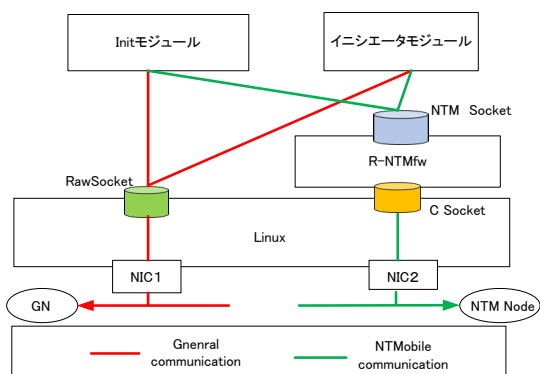


図3 NTMA(イニシエータ側)のモジュール構成

### 5.2 R-NTMfw

NTMfw は lwIP によりユーザデータから仮想 IP アドレスによるパケットを生成する機能があるが、NTMA は、GN が送信する IP パケットを RAW ソケットでそのまま受信する。そのため、NTMfw 内部で仮想 IP パケットを生成する必要がない。また、相手 NTM 端末から受信したパケットは、仮想 IP アドレスのパケットを抽出した後、IP ヘッダを除去せずに、そのまま GN へ中継すればよい。IP パケットの生成と削除は、NTMfw 内の lwIP で行われていたため、R-NTMfw では lwIP の処理をスキップするように改造した。また、NTMfw は、カーネルが認識する全ての NIC に対して NTMobile の処理を行うように実装されているため、R-NTMfw では NTMobile 通信を行わない NIC を認識しないように変更した。

### 5.3 動作検証

提案方式の動作検証をするにあたって、Init モジュールとイニシエータモジュールの中継処理部を C 言語で作成した。検証方法は、NTMA と NTM 端末間でのトンネル構築処理はあらかじめ終了させた。また、GN の IP アドレスとして NTMA が DC から割り当てられた仮想 IP アドレスを静的に設定した。この状態で GN から NTM 端末の仮想 IP アドレス宛てにパケットを送信した。NTMA は、このパケットを NTMobile 通信に変換して送信し、NTM 端末で受信することを確認した。逆に、NTM 端末から送信したパケットを NTMA が一般パケットに変換し、GN がデータを受信することを確認した。

## 6 まとめ

本稿では、NTMobile を実装できない一般端末のために、一般端末に隣接設置して NTMobile 通信を代行する NTMA を提案した。また、試作により NTMA が動作できることを確認した。今後、NTMA のレスポンス側の機能について検討していく。

#### 参考文献

- [1] 上醉尾一真ほか：情報処理学会論文誌，Vol54，No10，  
pp. 2288–2299(2013).

# NTMobileアダプタの実現方式の検討

尾久 史弥<sup>†</sup> 納堂 博史<sup>†</sup> 鈴木 秀和<sup>†</sup> 内藤 克浩<sup>‡</sup> 渡邊 晃<sup>†</sup>

<sup>†</sup>名城大学 理工学部

<sup>‡</sup>愛知工業大学 情報科学部



# 研究背景 -現状のネットワーク-

## ◆ 現在の主流はIPv4ネットワーク

### ➤ IPv4グローバルアドレスの枯渇問題

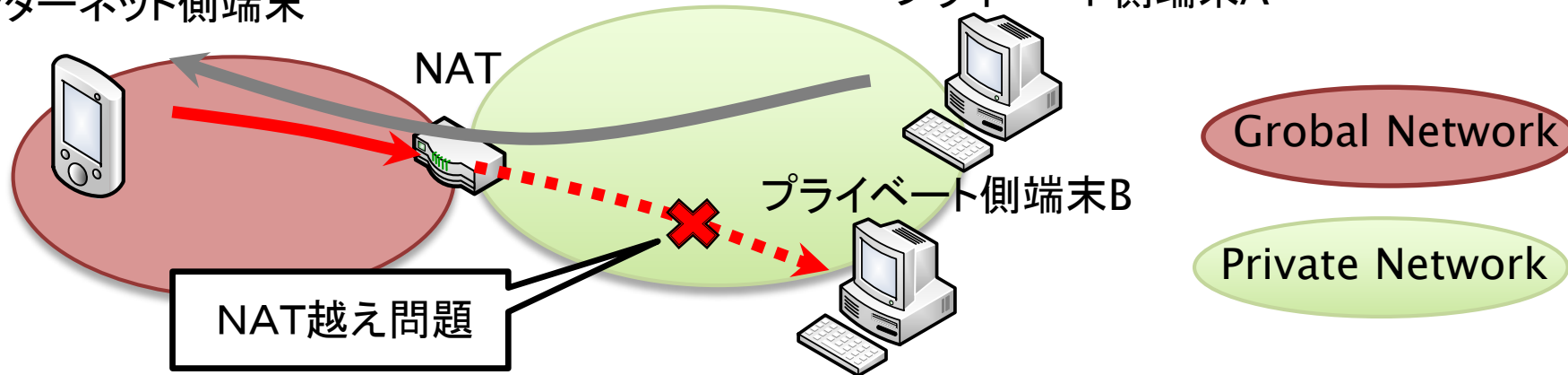
- NATを利用したプライベートネットワークの構築が一般的

## ◆ NAT越え問題

### ➤ インターネット側端末から通信を開始できない

インターネット側端末

プライベート側端末A

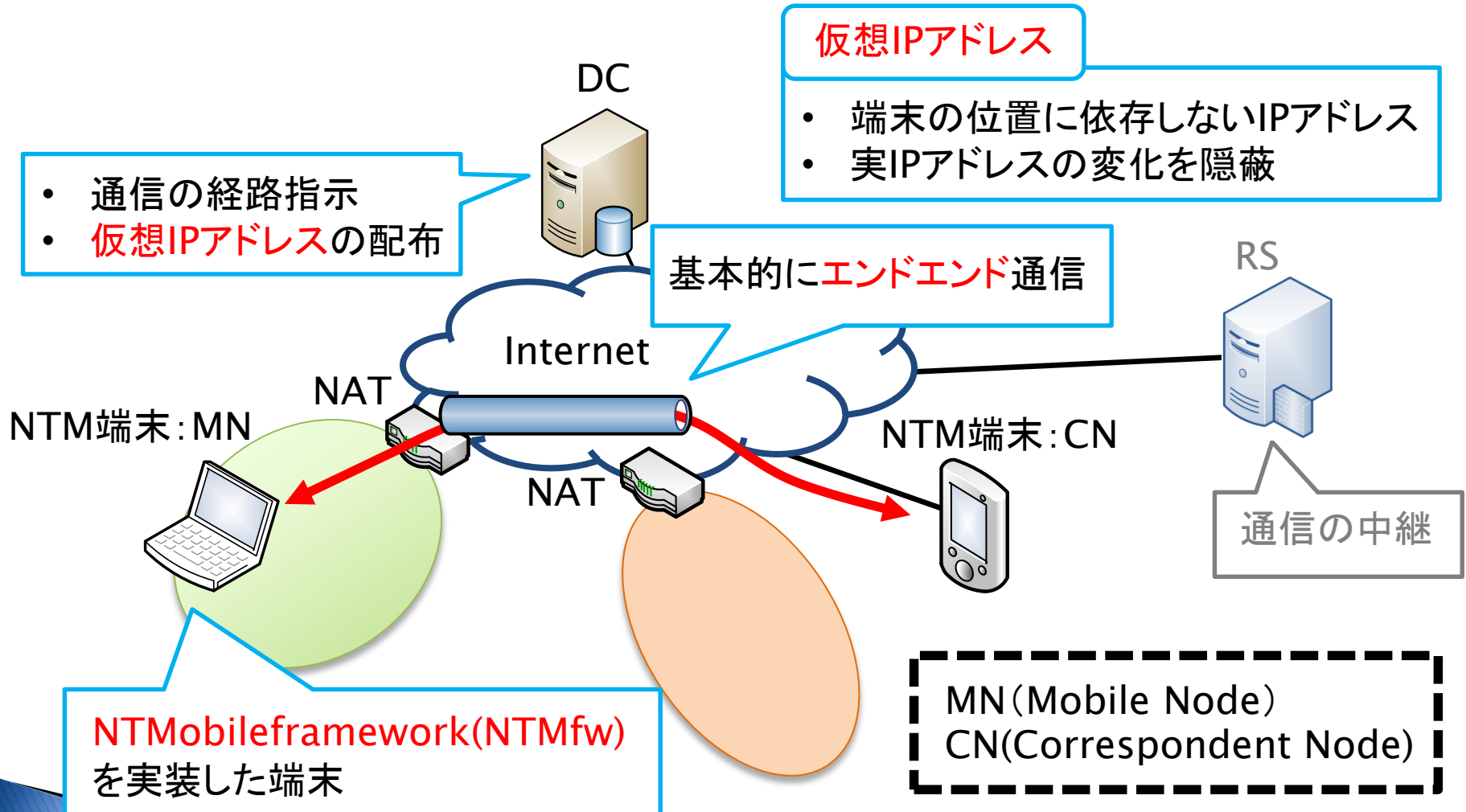


NAT越え問題を解決する技術

\*NTMobile(Network Traversal with Mobility)が提案されている

# NTMobile(Network Traversal with Mobility)

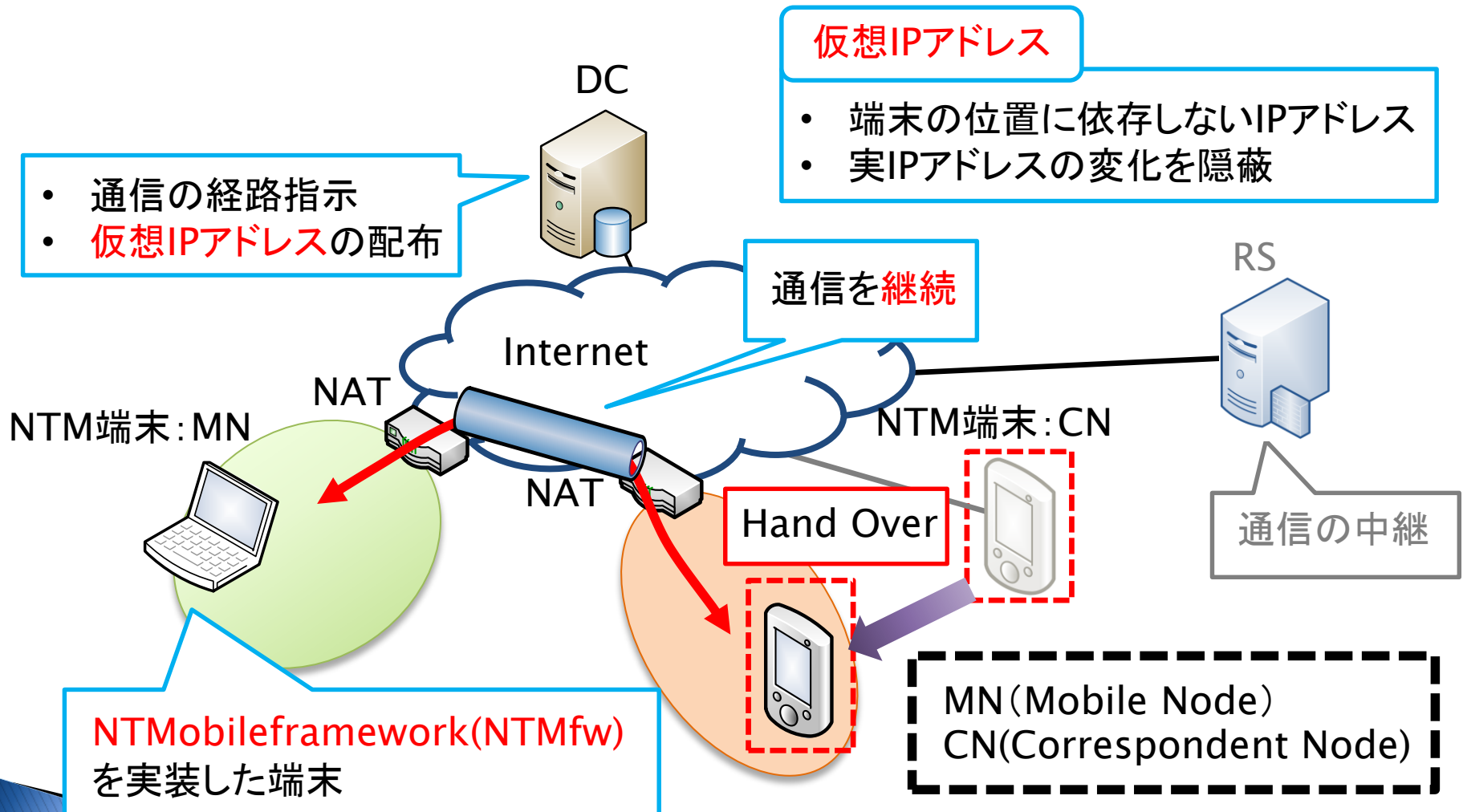
- 移動透過性の実現と**NAT越え問題の解決**を同時に実現する技術
- NTM端末, DC(Direction Coordinator), RS(Relay Server)で構成





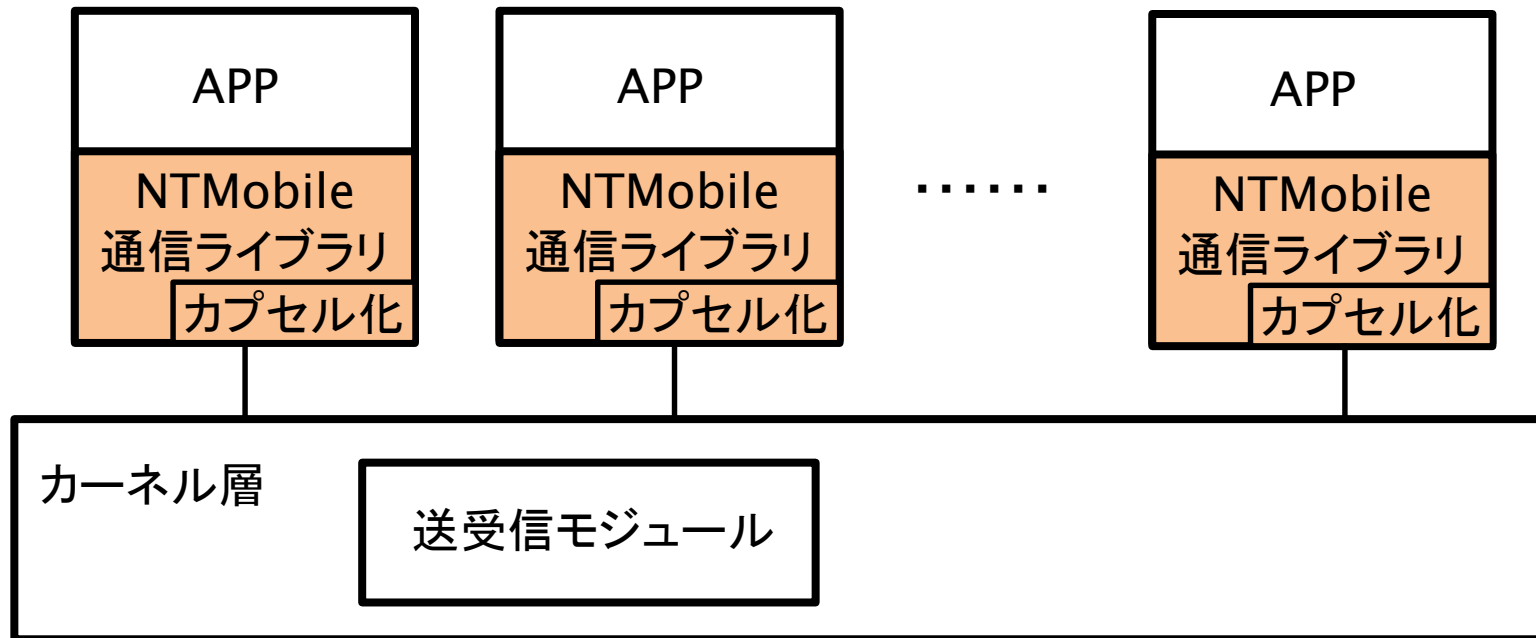
# NTMobile(Network Traversal with Mobility)

- 移動透過性の実現とNAT越え問題の解決を同時に実現する技術
- NTM端末, DC(Direction Coordinator), RS(Relay Server)で構成



# NTMobile framework(NTMfw)

- 上位アプリケーションに対してNTMobile機能を提供
  - NTMobile通信用のAPIを提供



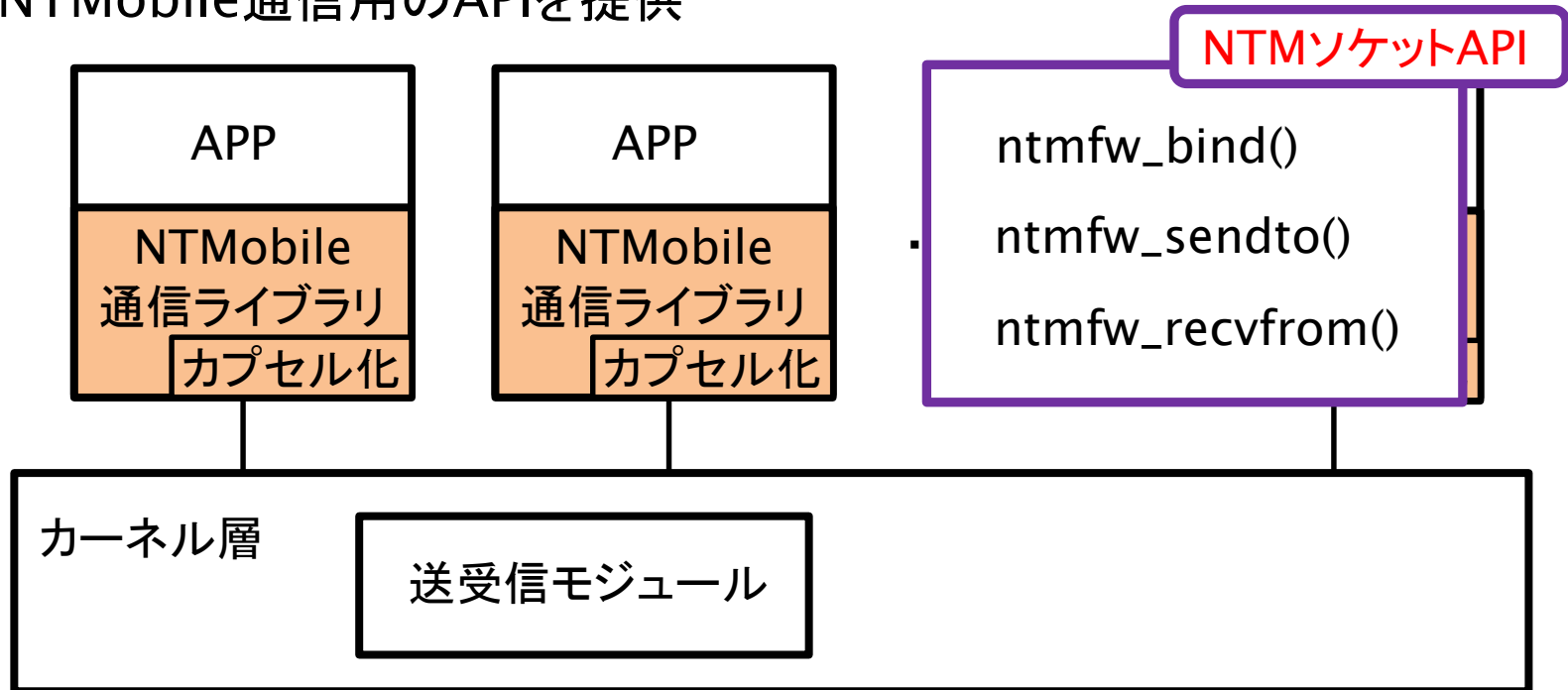
|       | 通信ライブラリ   | アップデート | ソフトウェア | 対応可能OS                 |
|-------|-----------|--------|--------|------------------------|
| NTMfw | アプリケーション層 | 必要なし   | 変更必要あり | Linux,<br>Android, iOS |

**\*Linuxでは、動作確認済み**



# NTMobile framework(NTMfw)

- 上位アプリケーションに対してNTMobile機能を提供
  - NTMobile通信用のAPIを提供

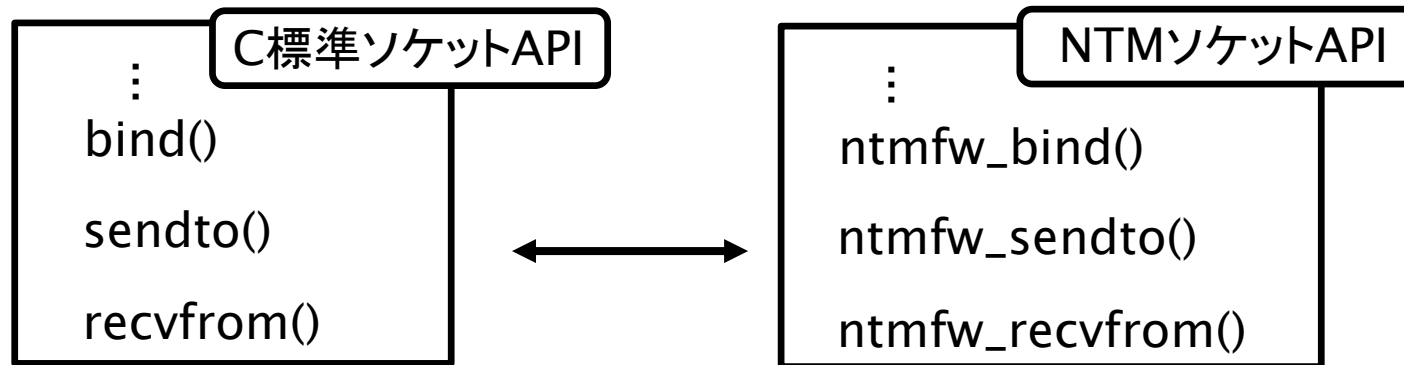


|       | 通信ライブラリ   | アップデート | ソフトウェア | 対応可能OS                 |
|-------|-----------|--------|--------|------------------------|
| NTMfw | アプリケーション層 | 必要なし   | 変更必要あり | Linux,<br>Android, iOS |

**\*Linuxでは、動作確認済み**

# NTMfwの課題

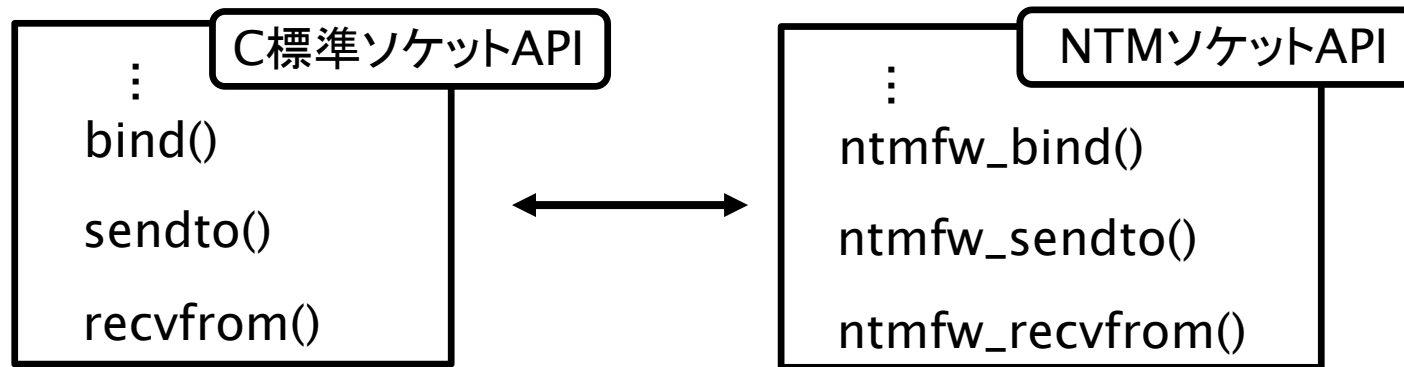
- NTMobileはLinuxで動作が確認されている
  - ✗ Linux以外のOSは未対応
- NTMfw(NTMobile framework)を組み込めない



- ✗ 組み込み型の家電
  - 工場出荷後はプログラムを書き換えられない
- ✗ サーバー
  - 安定性を重視することから、新しい機能の追加が許可されない

# NTMfwの課題

- NTMobileはLinuxで動作が確認されている
  - ✗ Linux以外のOSは未対応
- NTMfw(NTMobile framework)を組み込めない

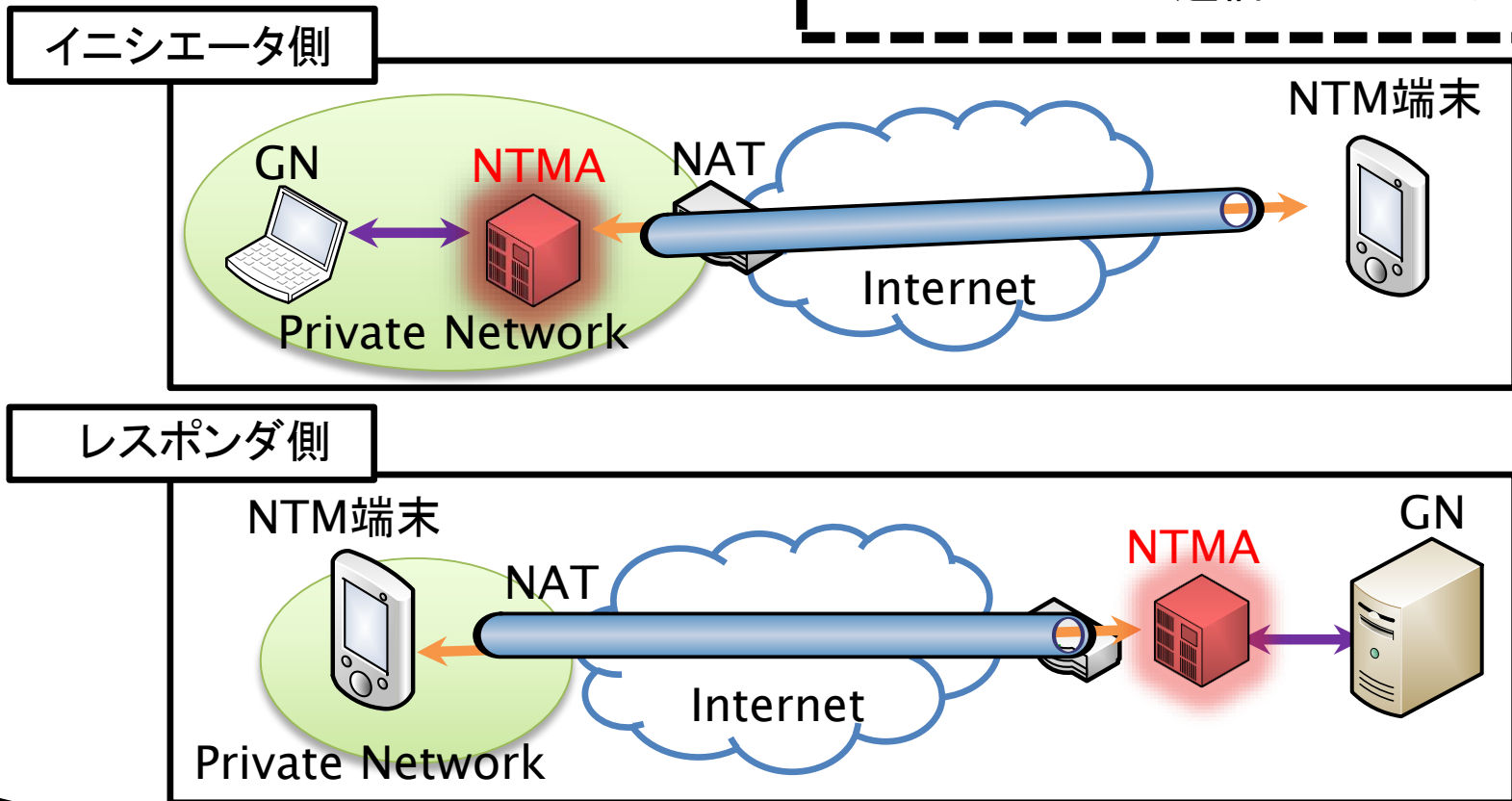


✗ 組み込み型の家電

NTMobileアダプタ(NTMA)の提案

# 研究目的

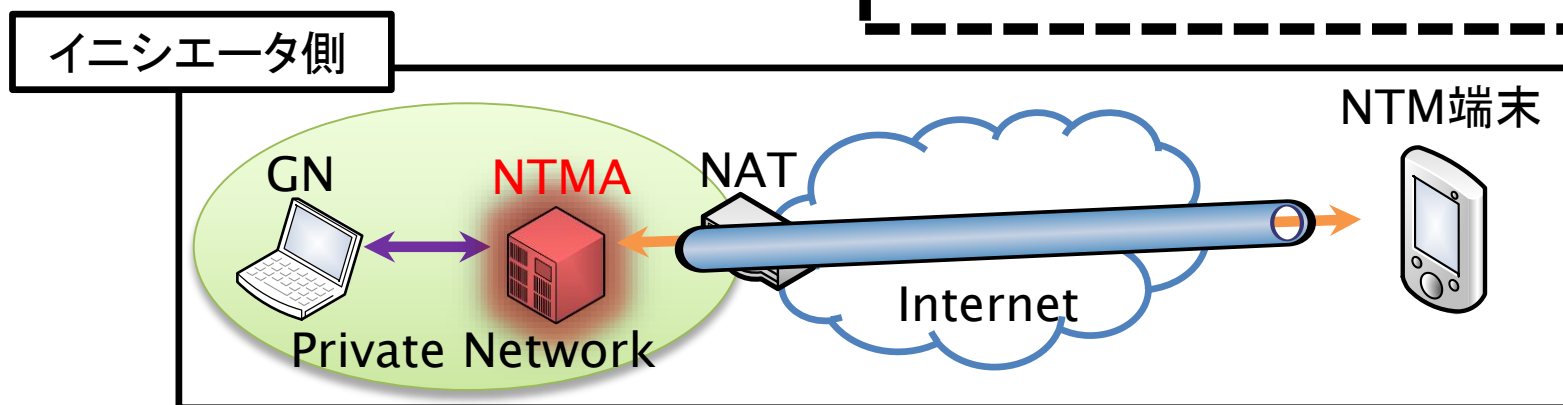
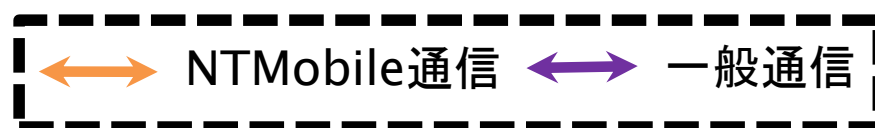
- 一般端末(GN:General Node)に**NTMA**を隣接設置
  - GNの通信をNTMobile通信に変換
  - GNのプログラムに手を加えない



# 研究目的 (イニシエータNTMA)

## ■ イニシエータNTMA

- 通信開始側(イニシエータ)のGNに設置
- GNの通信をNTMobile通信に変換



### GNの想定環境

- ✓ Windows PC
- ✓ NTMfw未実装端末

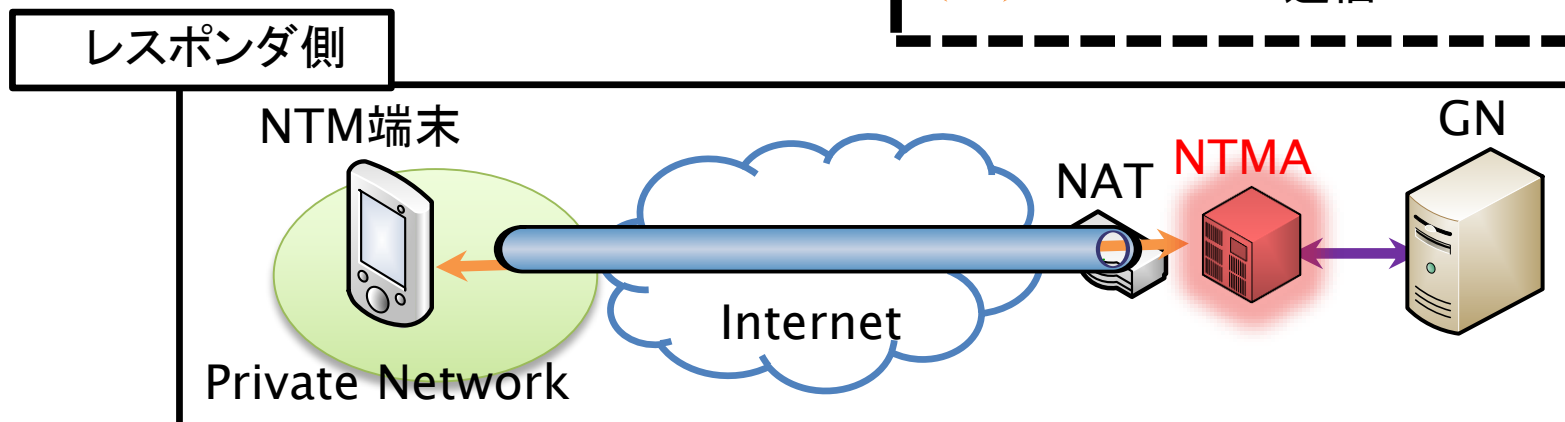
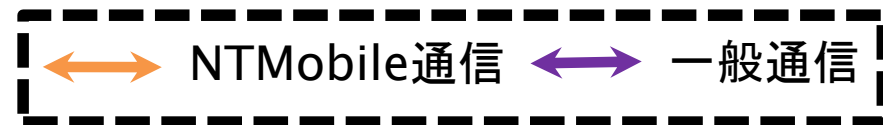
### NTMobileの課題

- ✓ NTMfwが未実装
- ✗ Windowsは未対応
- ✗ NTMfwが未実装の端末

# 研究目的 (レスポндаNTMA)

## ■ レスポндаNTMA

- 通信開始される側(レスポнда)のGNに設置
- NTMobile通信を一般通信に変換



### GNの想定環境

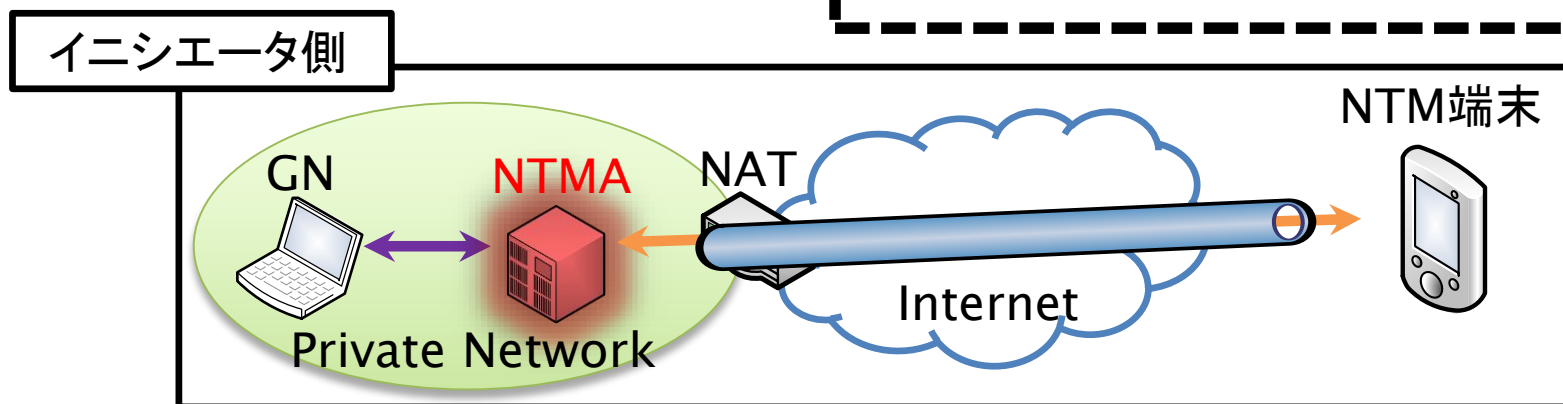
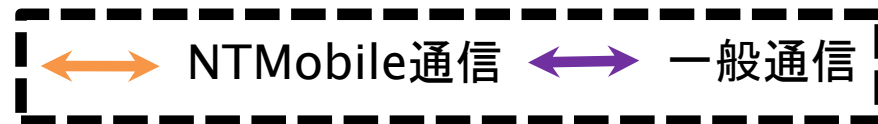
- ✓ サーバー
- ✓ 組み込み型家電

### NTMobileの課題

- ✓ NTMfwを組み込めない
- ✗ 新しい機能の追加が許可されない
- ✗ 工場出荷後はプログラムを書き換えられない



# 研究目的（イニシエータNTMAの発表）



## GNの想定環境

- ✓ Windows PC
- ✓ NTMfw未実装端末

## NTMobileの課題

- ✓ NTMfwが未実装
- ✗ Windowsは未対応
- ✗ NTMfwが未実装の端末

イニシエータ側NTMAの動作について確認

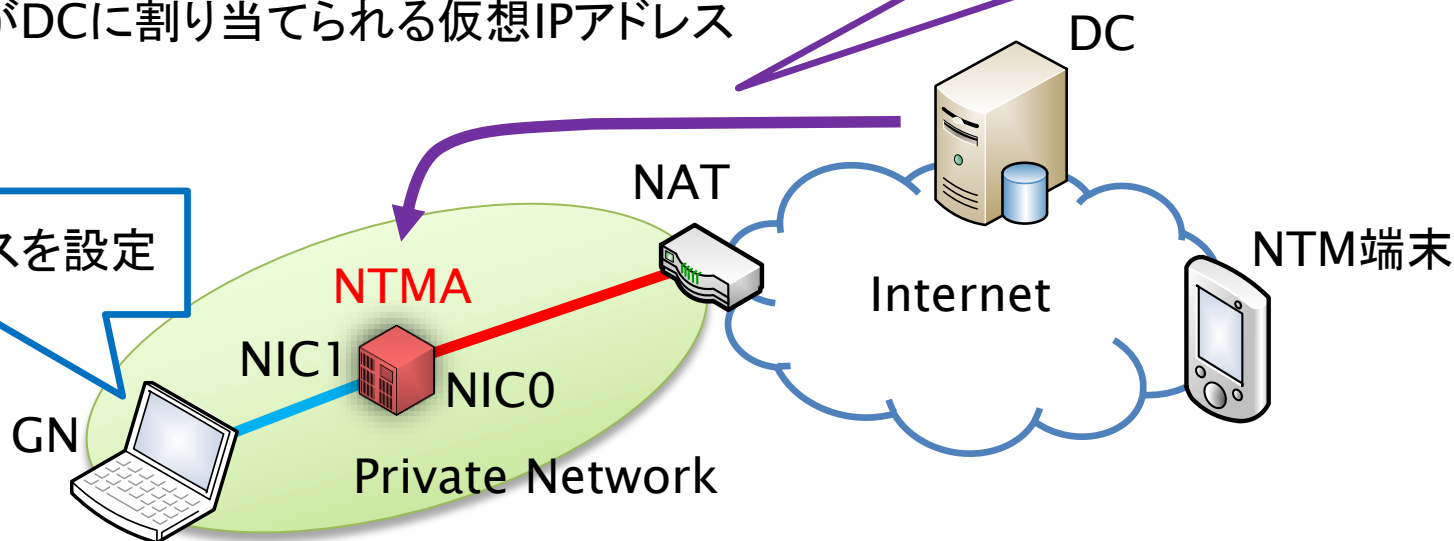
# 提案方式の構成

- ◆ NICを2枚用意
  - NIC0をインターネット側にブリッジ接続
  - NIC1をGNにブリッジ接続
- ◆ GNに仮想IPアドレスを設定
  - NTMAがDCに割り当てられる仮想IPアドレス

NIC: Network Interface Card  
✓ LANに接続するための装置

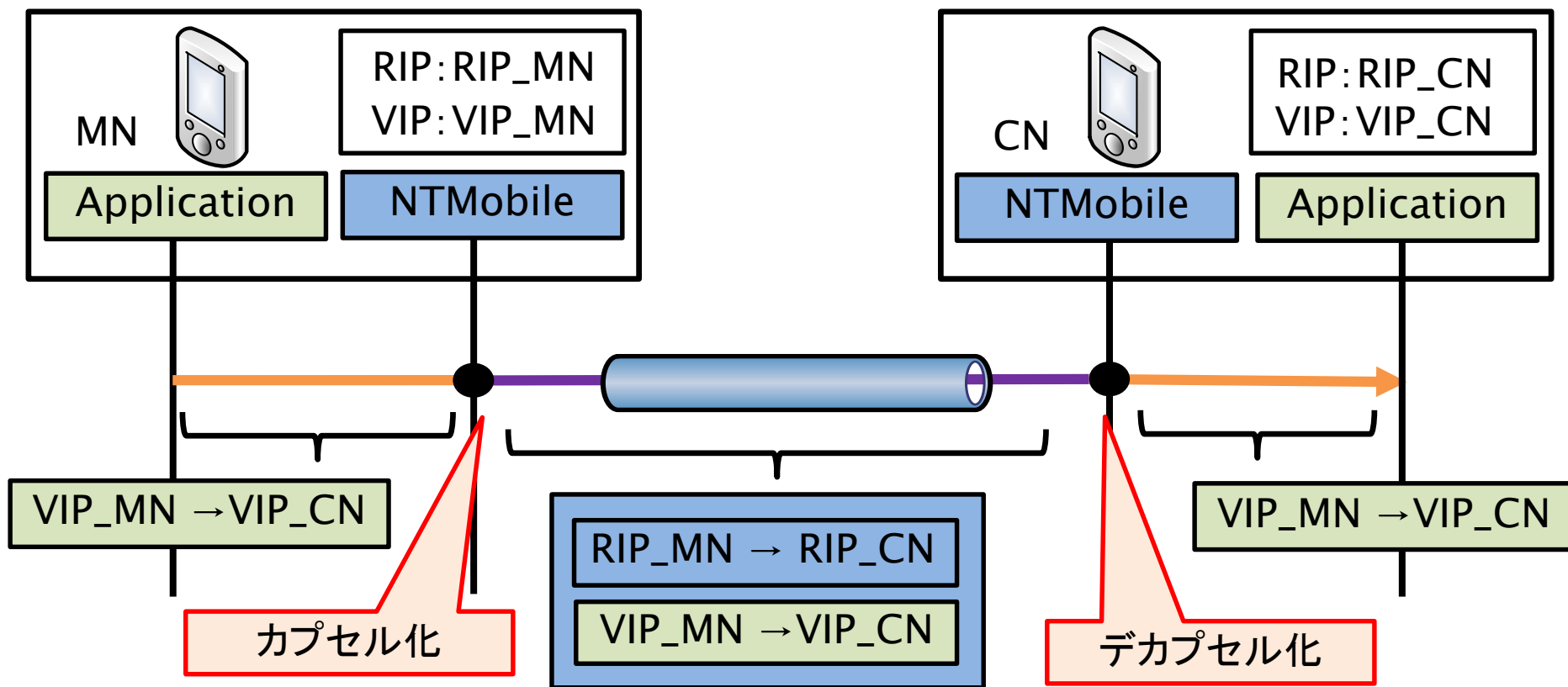
仮想IPアドレスの割り当て

仮想IPアドレスを設定



— イーサネット0 — イーサネット1

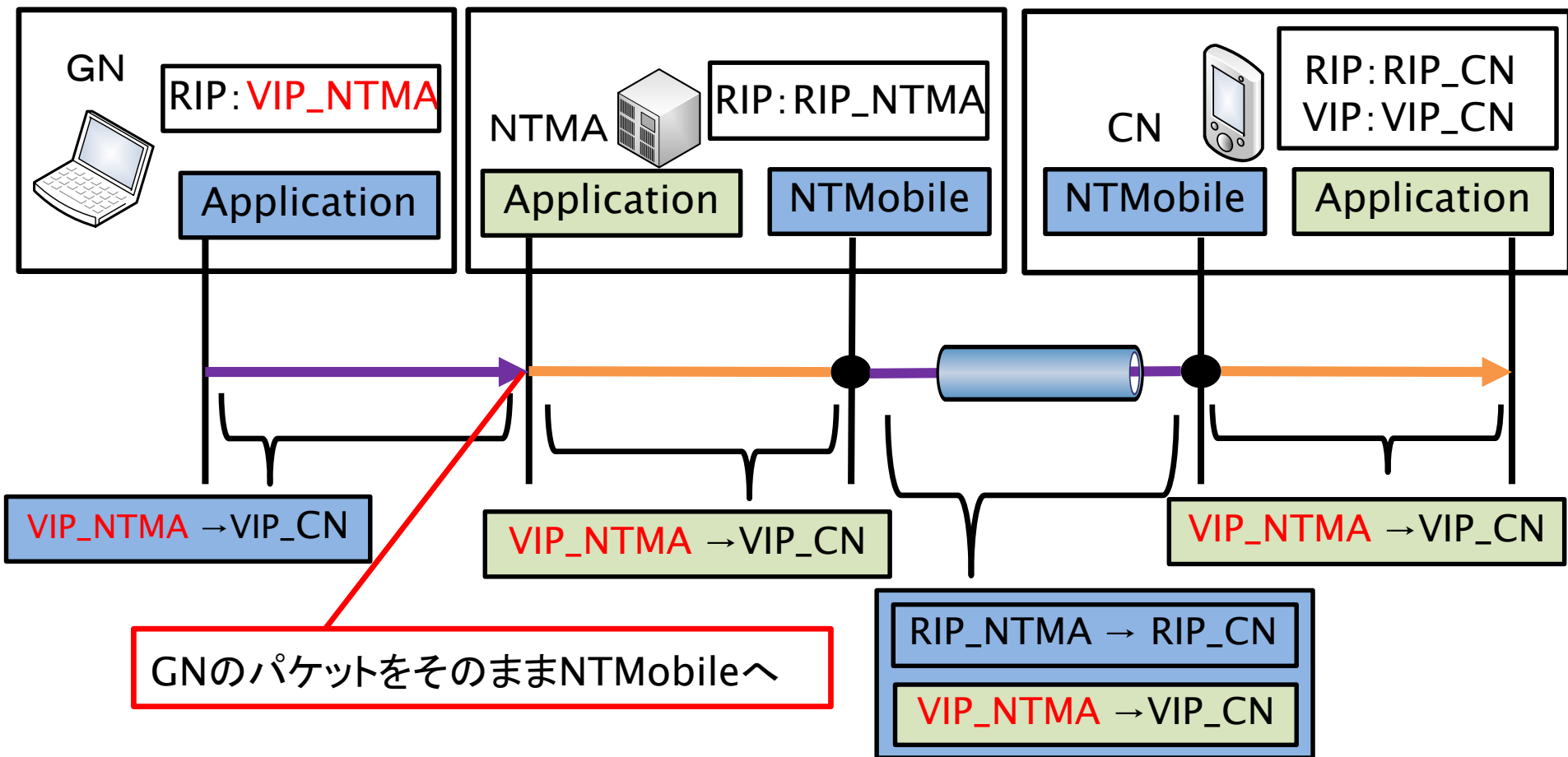
# NTMobile -通信時-



RIP: 端末の実IPアドレス  
 VIP: 端末の仮想IPアドレス

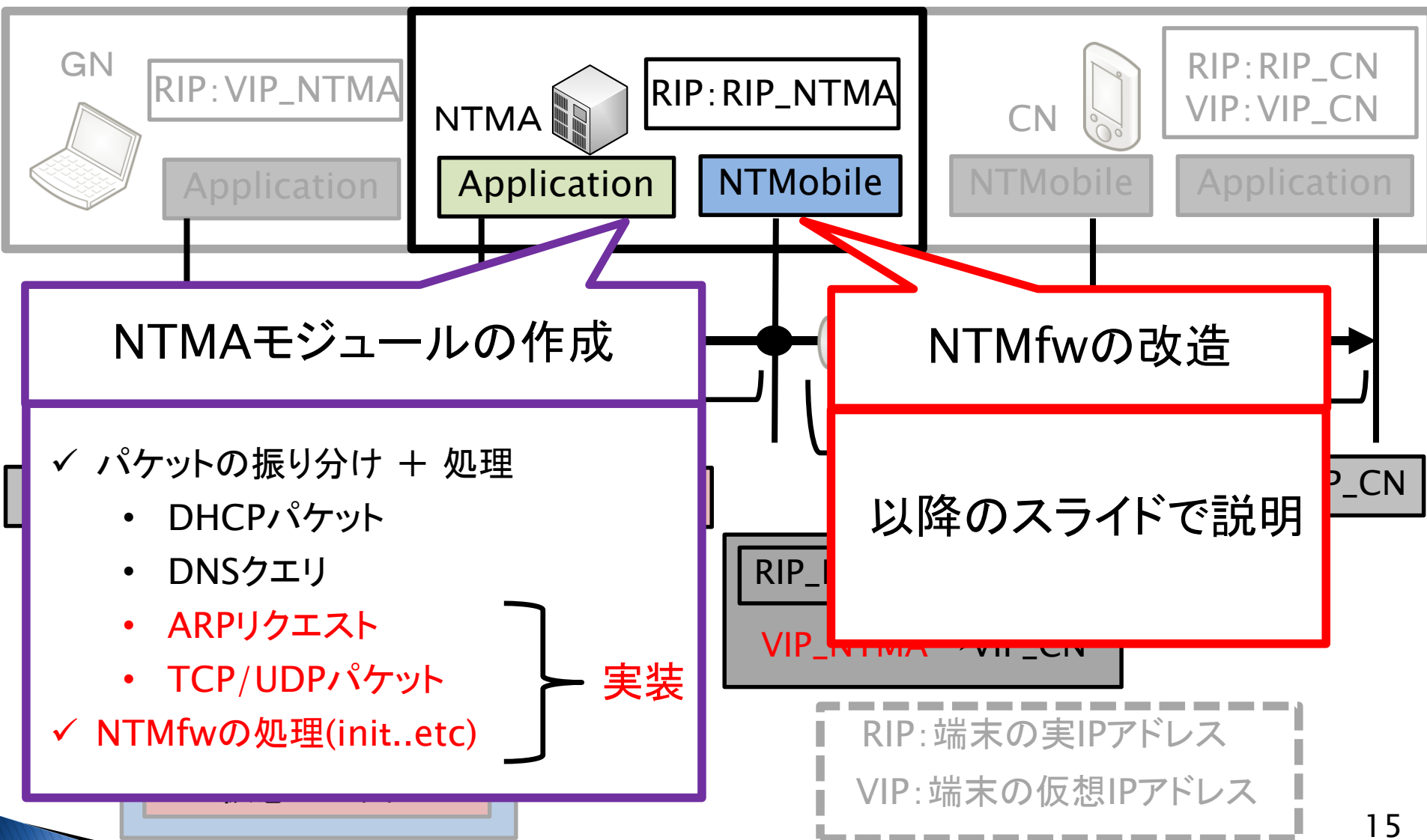
Data flow  
 Packet flow

# 提案方式 -通信時-



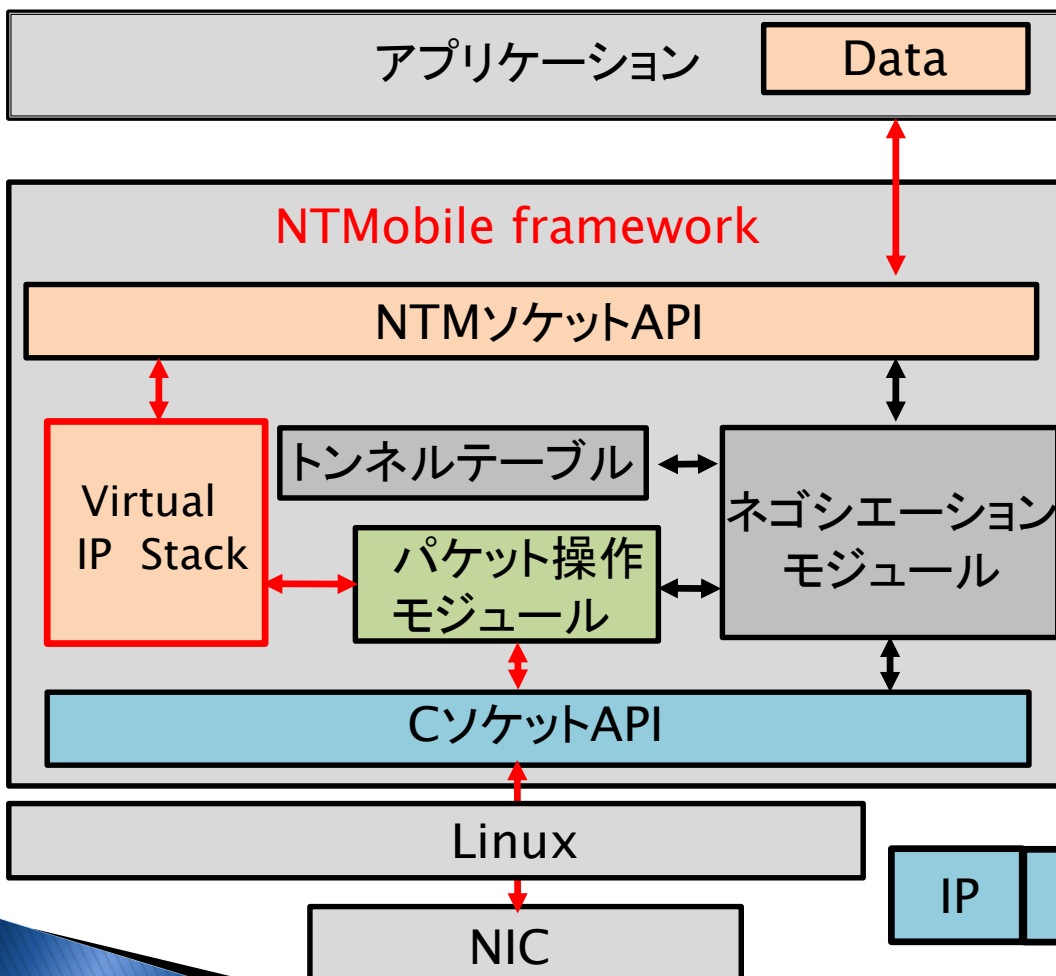
GNのパケットをそのままNTMobileへ

# 提案方式 -通信時-



# NTMfw (NTMobile framework)

- ◆ アプリケーション層で動作する通信ライブラリ
  - 上位アプリケーションにNTMobile機能を提供



## ● NTMソケットAPI

- 上位アプリケーションにNTMソケットAPIを提供

```
ntmfw_sedto()..etc
```

## ● Virtual IP Stack

- 仮想IPパケットの生成



## ● パケット操作モジュール

- 暗号化・MAC付与



## ● CソケットAPI

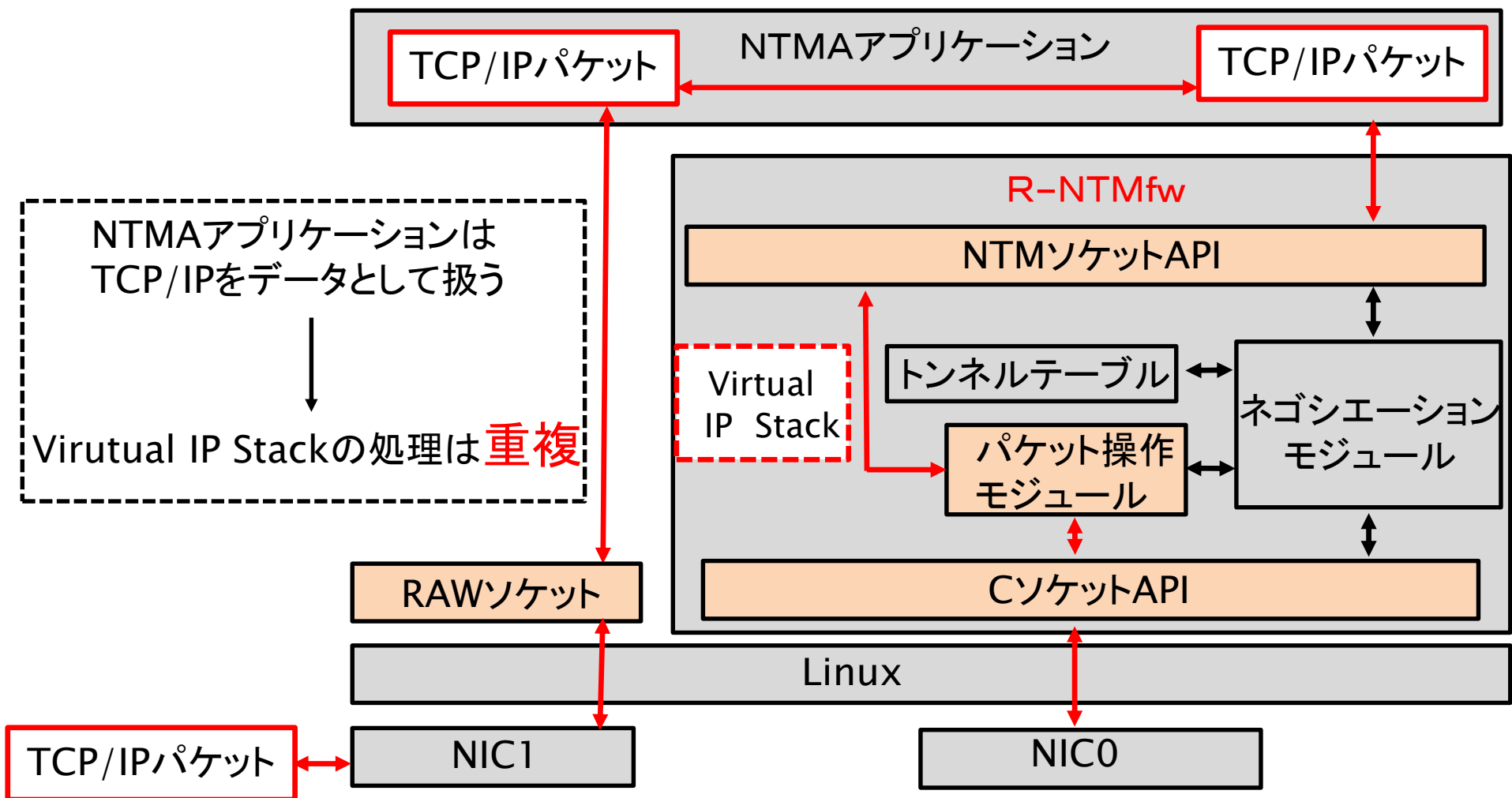
- Linuxが提供するCソケット



# R-NTMfw(Remodeled NTMfw)

## ◆ R-NTMfw

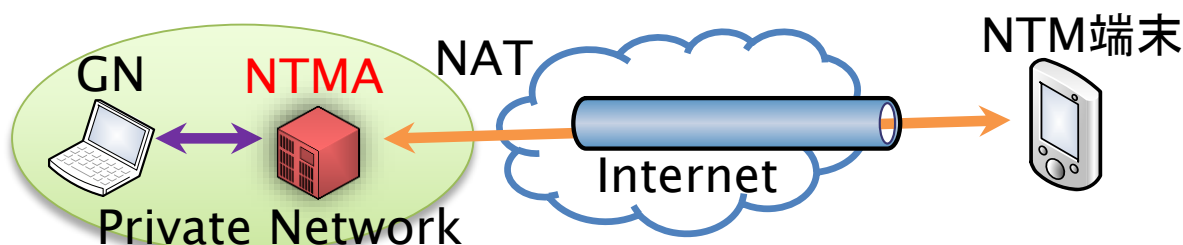
### ➤ NTMfwを改造したアプリケーション



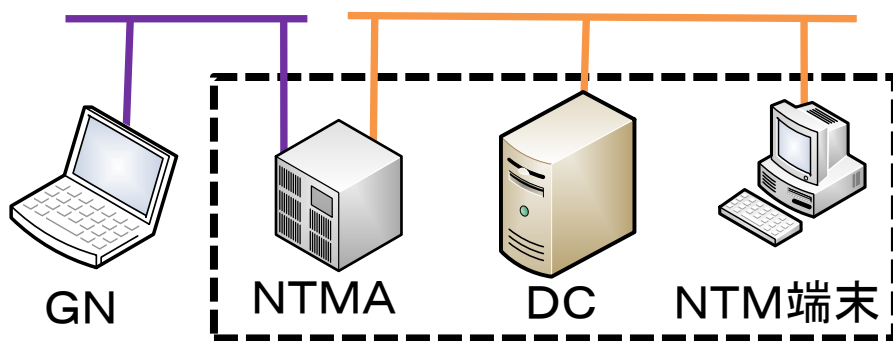
※RAWソケット 生のパケットをダイレクトに送受信するソケット

# 動作検証

- ◆ 提案方式をLinux上に実装
  - 一般通信をNTMobile通信に変換する動作を確認



- ◆ 測定環境
  - 仮想マシン上にNTMAを実装
  - 仮想マシン上にNTM端末,DCを構築
  - Windows実機のGN



| NTMA(Virtual Machine) |                           |
|-----------------------|---------------------------|
| OS                    | Ubuntu 14.04              |
| CPU                   | Intel Corei7-930(2.80GHz) |
| Memory                | 2GB                       |

|             |             |
|-------------|-------------|
| — Ethernet0 | — Ethernet1 |
|-------------|-------------|



# 性能評価

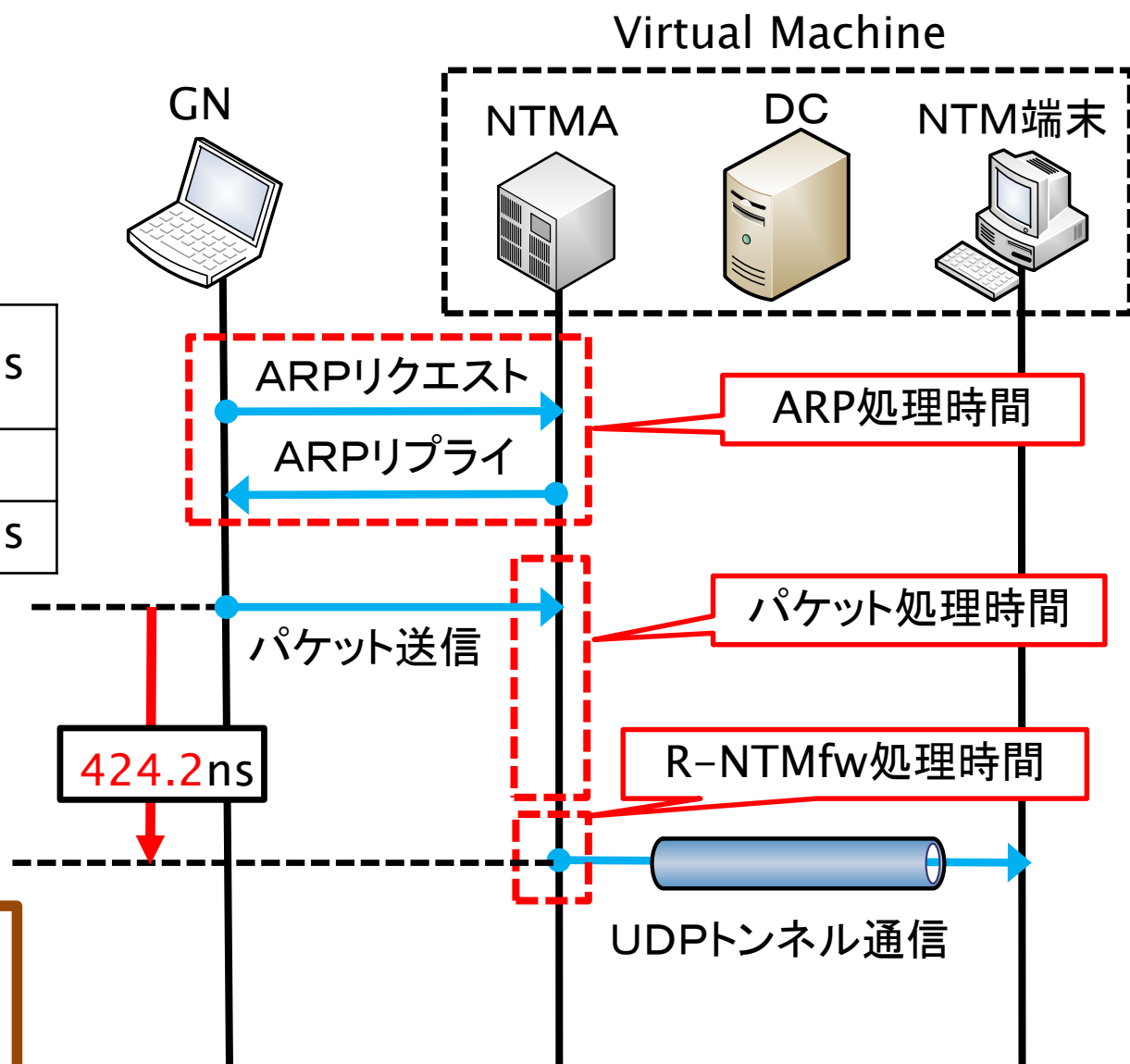
10回の平均値

|                  |         |
|------------------|---------|
| パケット中継に<br>要する時間 | 424.2ns |
| パケット処理時間         | 15.5ns  |
| R-NTMfw処理時間      | 408.7ns |

|         |         |
|---------|---------|
| ARP処理時間 | 485.7ns |
|---------|---------|

424.2ns

一般通信をNTMobile通信に  
変換する時間は**僅か**



# まとめ

## ◆ NTMobileアダプタの実現方式の検討

- 一般端末にNTMAを隣接設置
  - 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

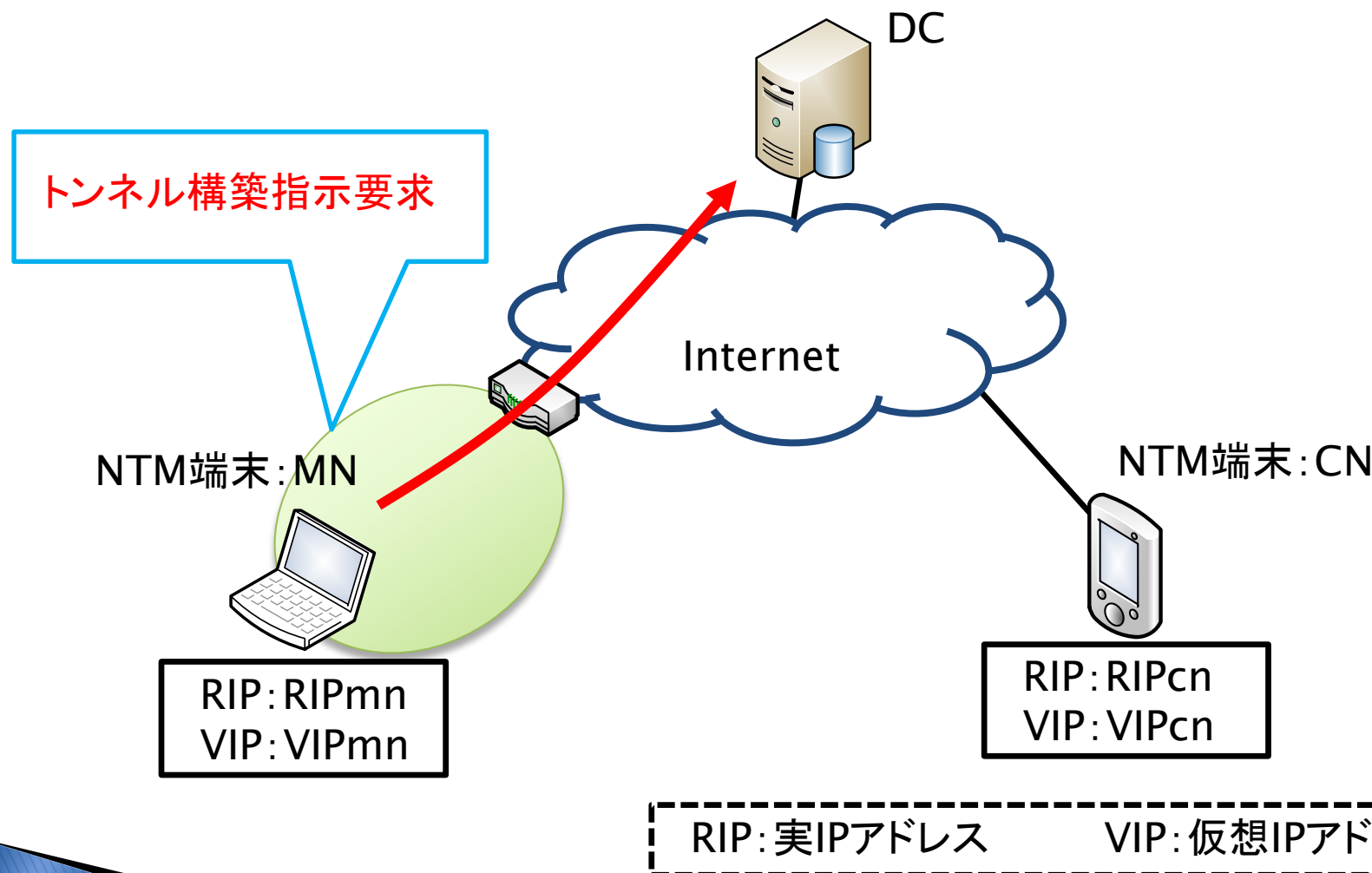
## ◆ 今後の方針

- イニシエータ側NTMAの実装の完了
- レスポнда側NTMAの機能について検討・実装
- NTMAの機能拡張に向けて検討

# 付録

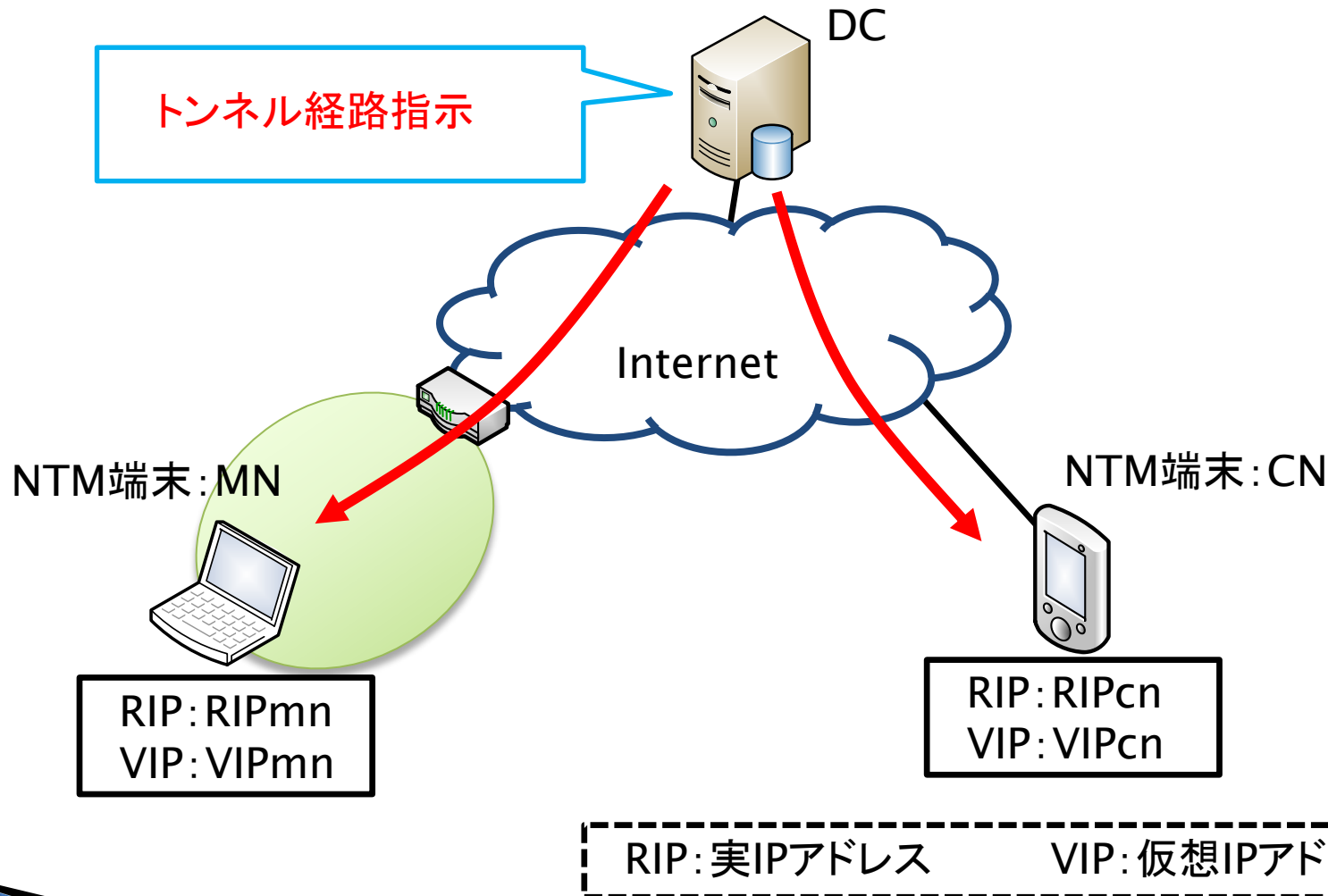
# NTMobileシグナリング

- NTM端末が通信開始時に行う処理
  - 両NTM端末間でトンネル構築処理を行う



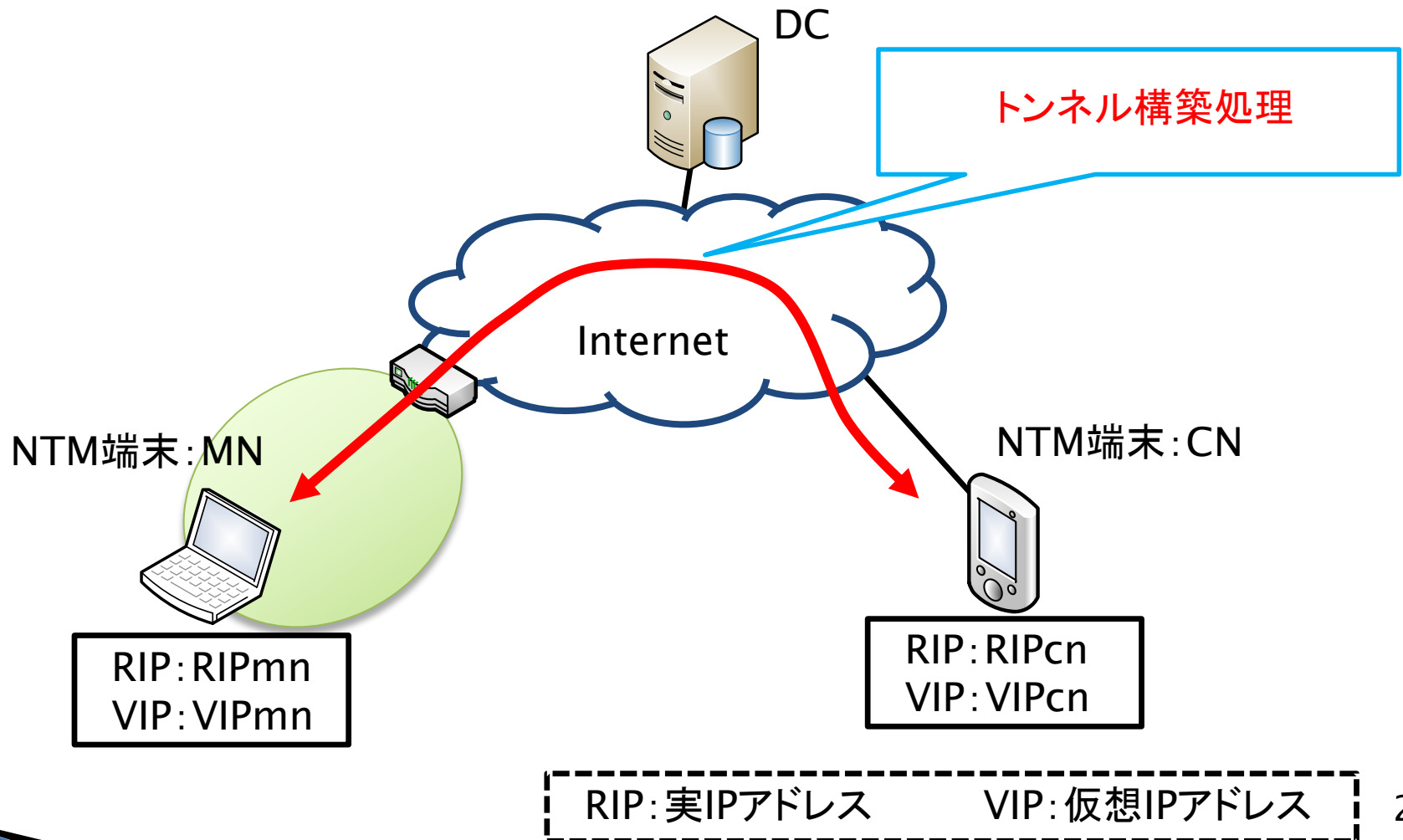
# NTMobileシグナリング

- NTM端末が通信開始時に行う処理
  - 両NTM端末間でトンネル構築処理を行う



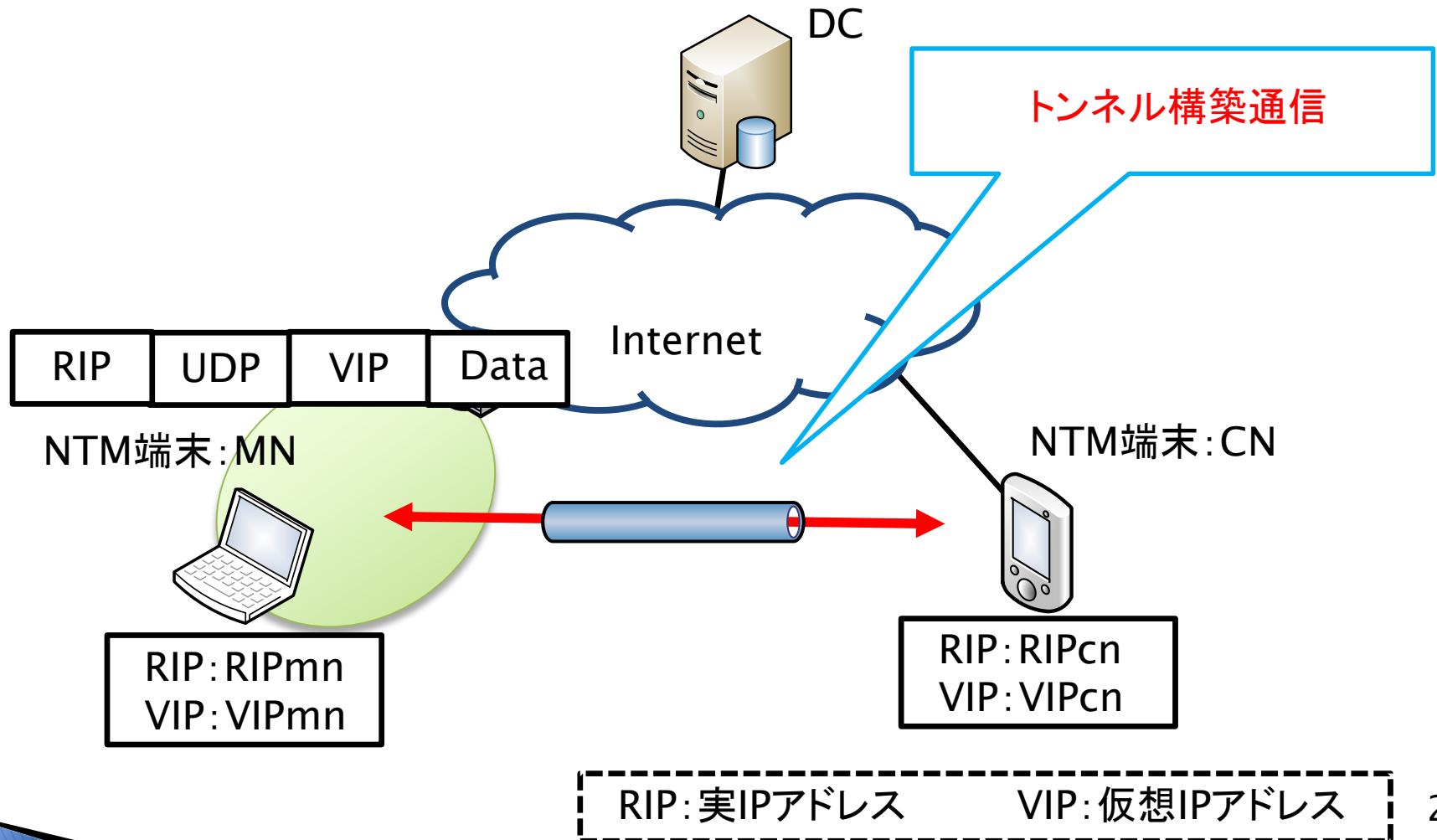
# NTMobileシグナリング

- NTM端末が通信開始時に行う処理
  - 両NTM端末間でトンネル構築処理を行う



# NTMobileの実際の通信

- UDPTunnel通信
  - 実IPパケットで仮想IPパケットをカプセル化



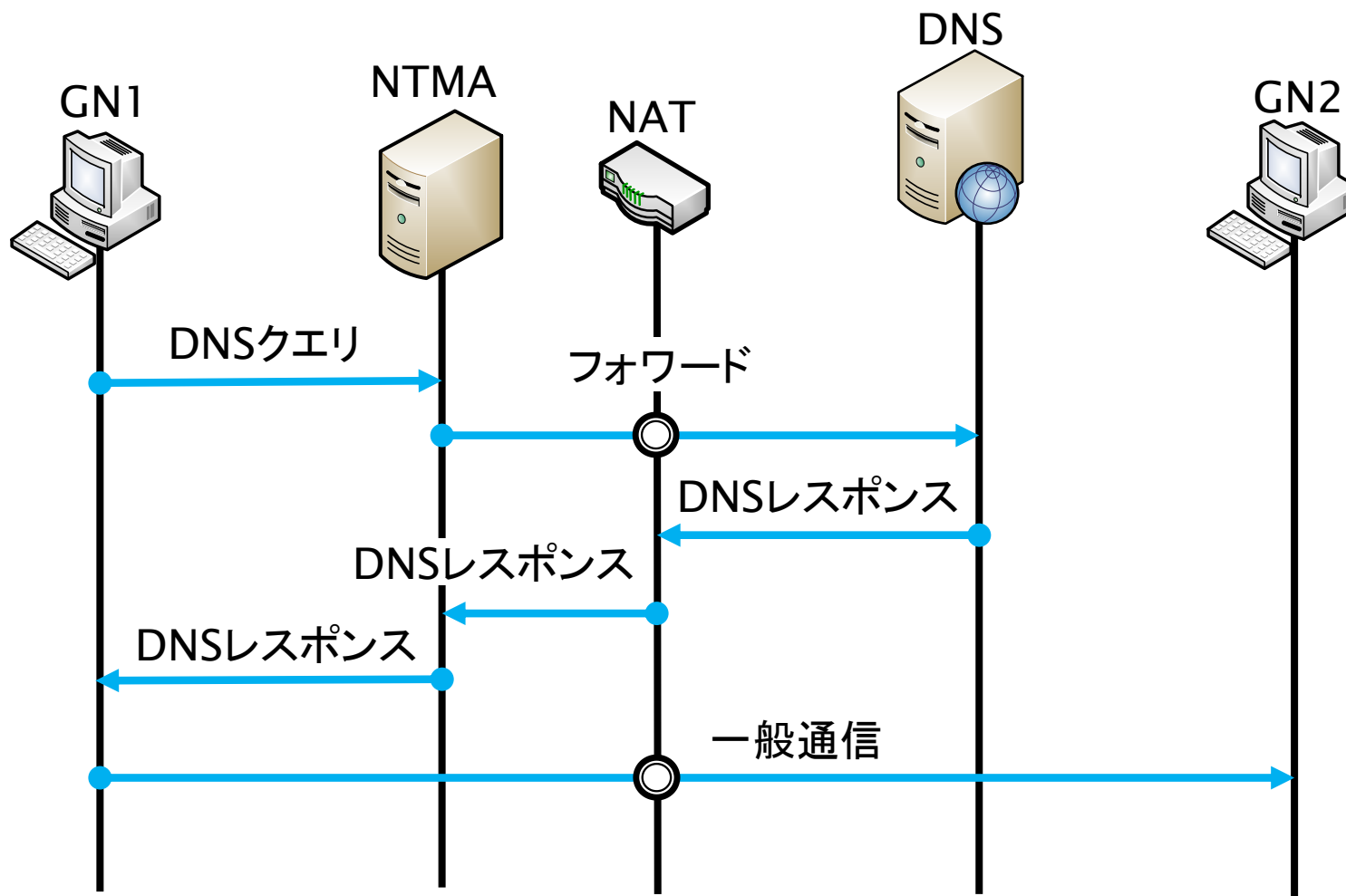
# 通信相手がGNの場合

## ■ Bind

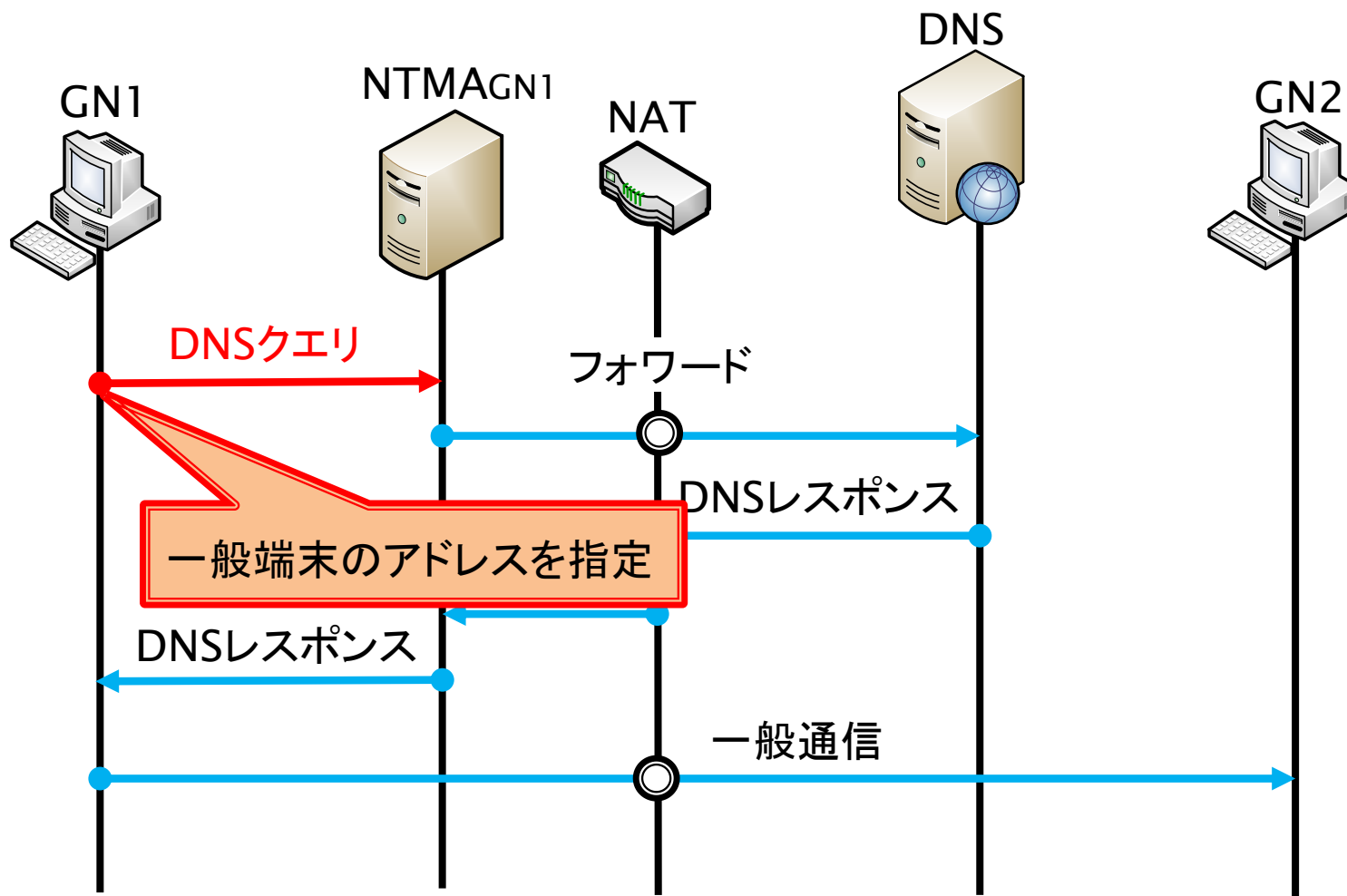
- DNSクエリのフォワード
- Bindのインストール/設定のみで実現可能



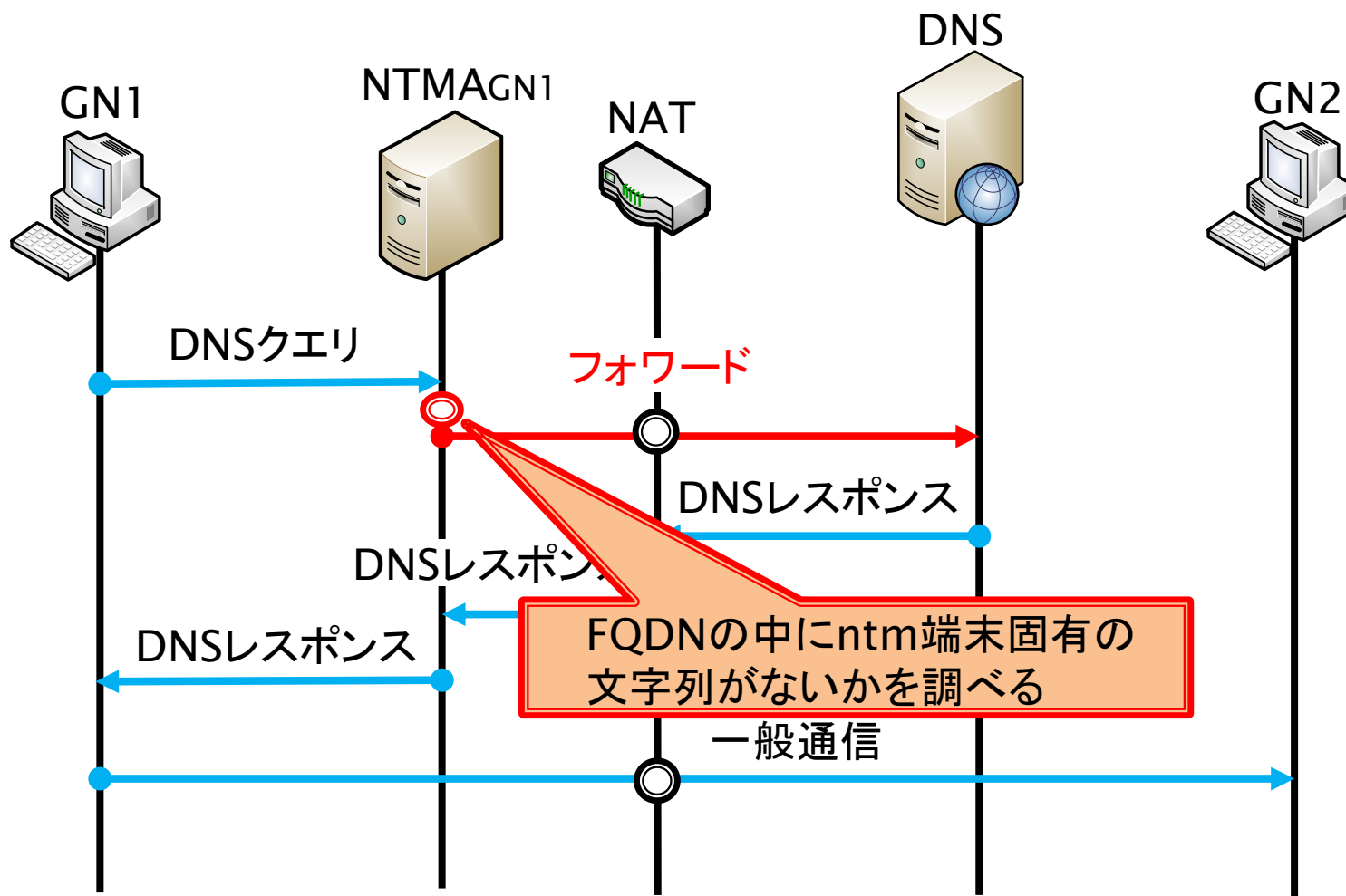
# 提案方式 -動作シーケンス-



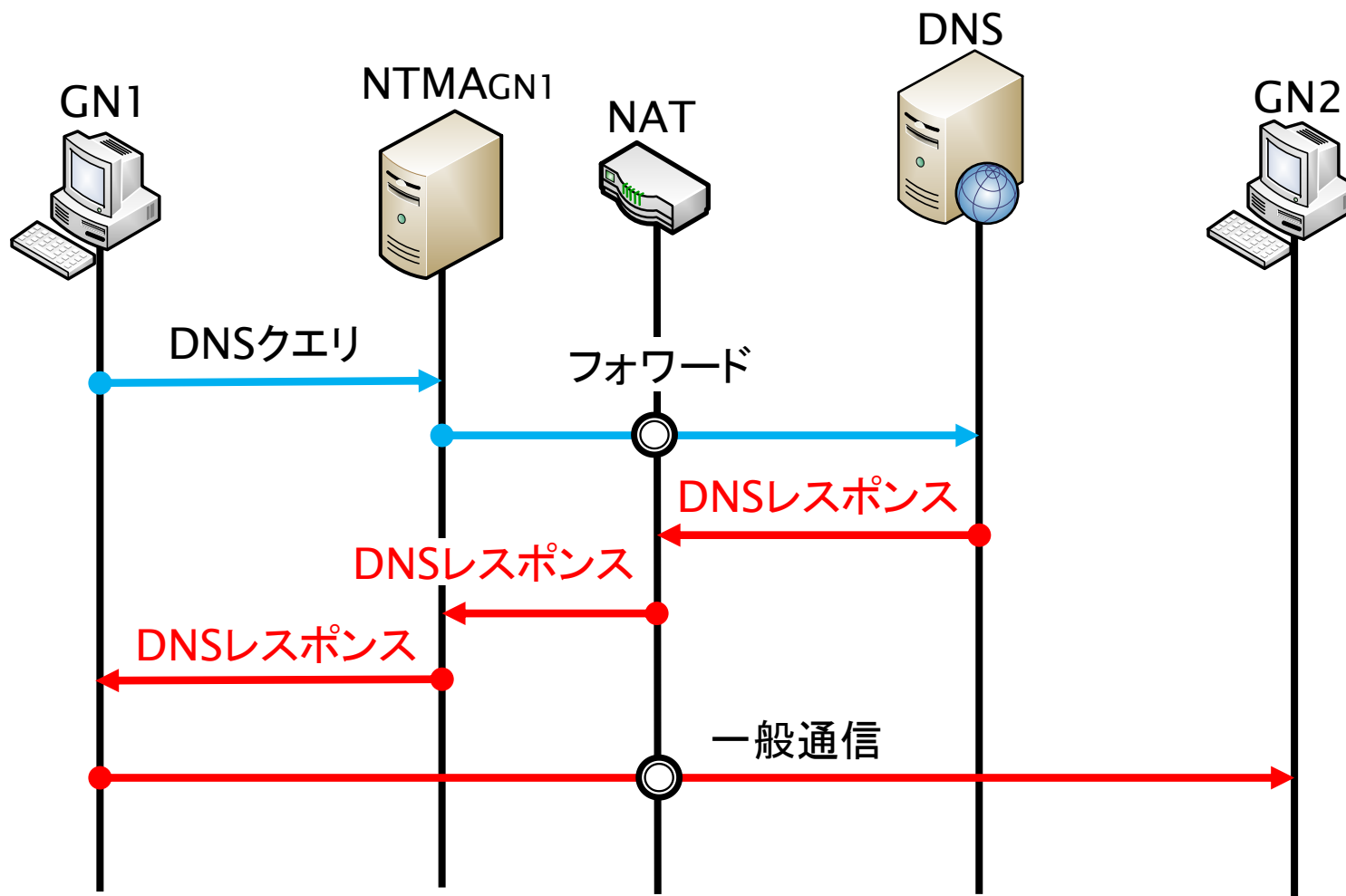
# 提案方式 -動作シーケンス-



# 提案方式 -動作シーケンス-

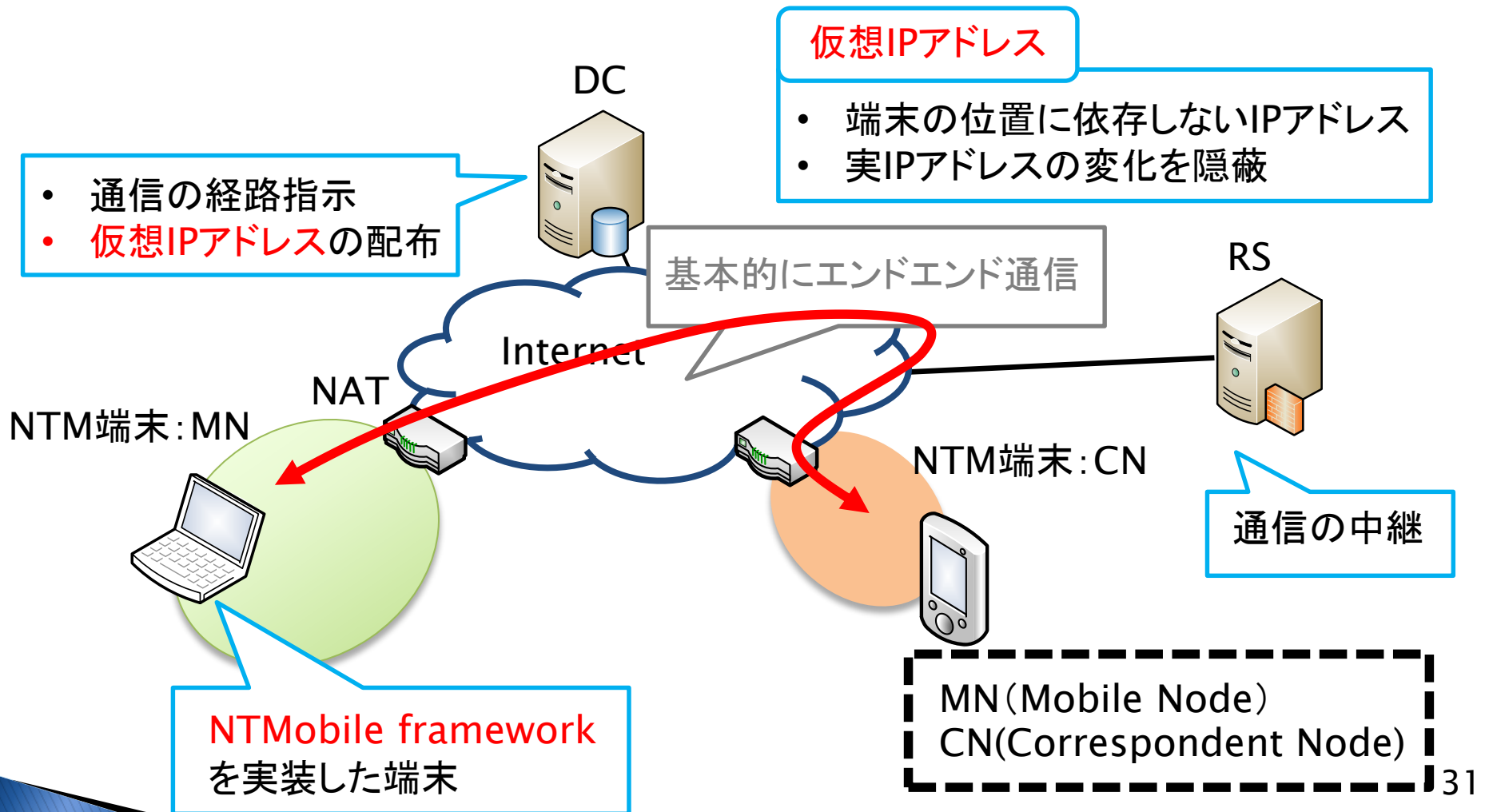


# 提案方式 -動作シーケンス-

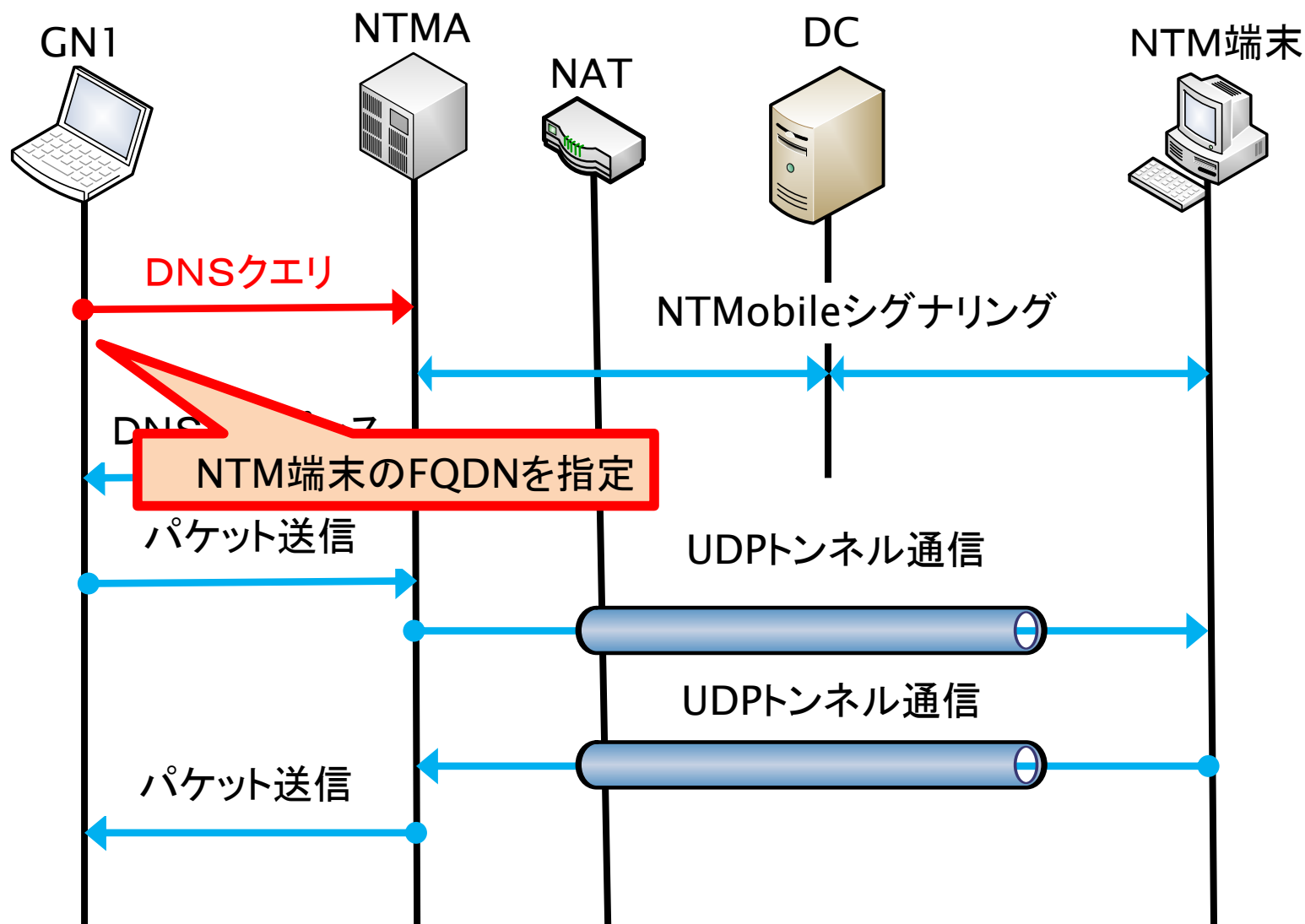


# NTMobile(Network Traversal with Mobility)

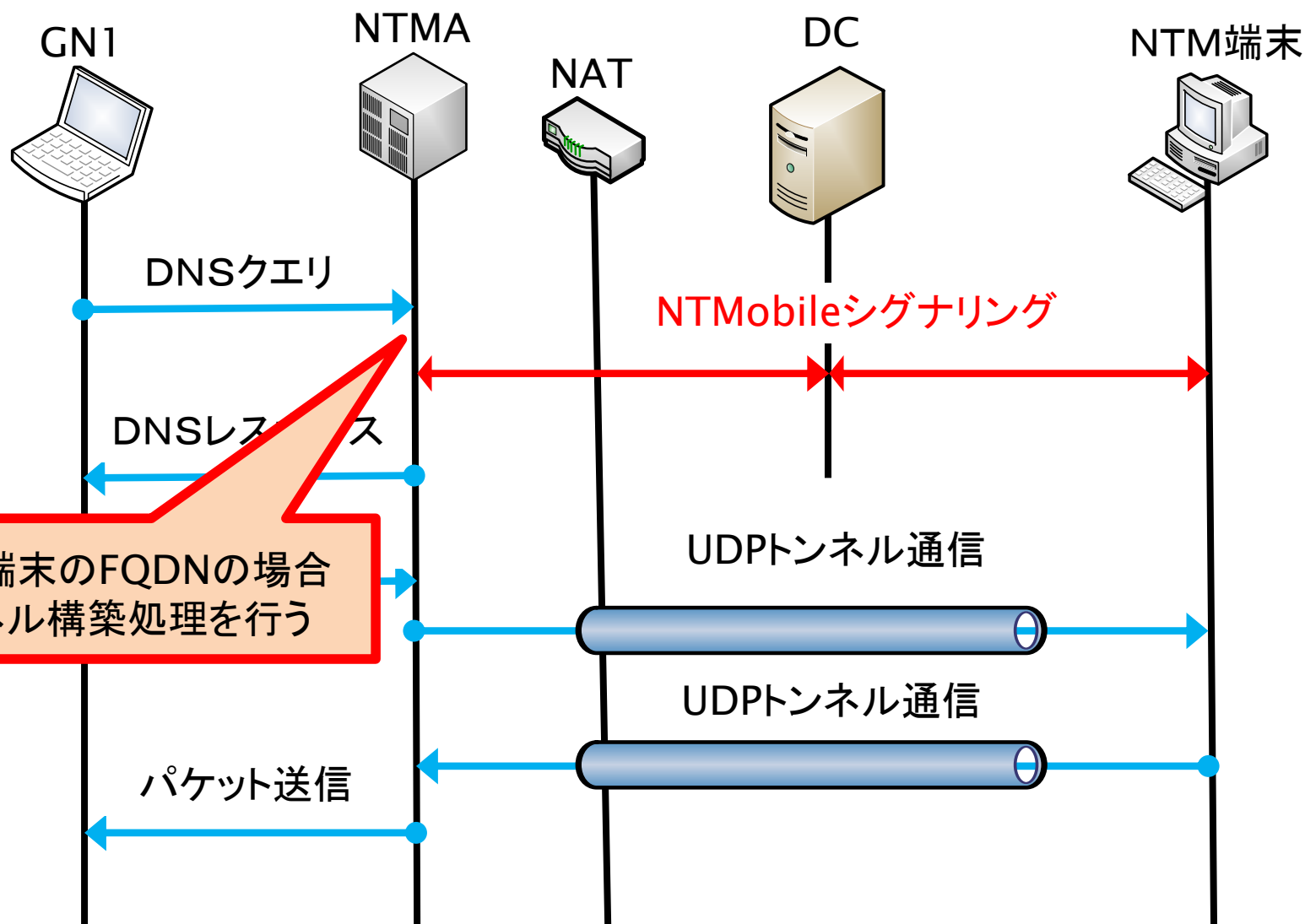
- NAT越え問題を解決する技術
- NTM端末, DC(Direction Coordinator), RS(Relay Server)で構成



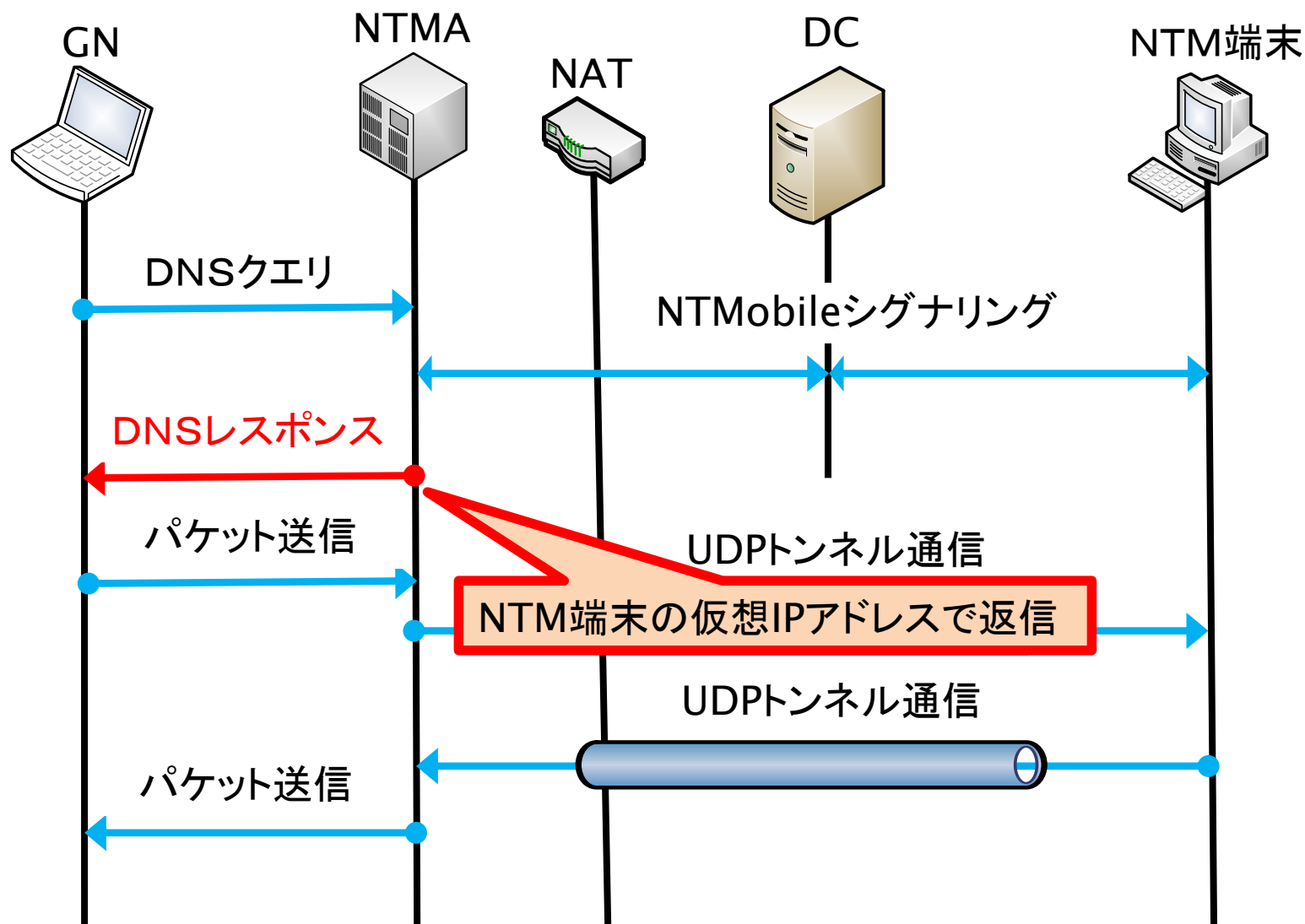
# 提案方式 -動作シーケンス-



# 提案方式 -動作シーケンス-

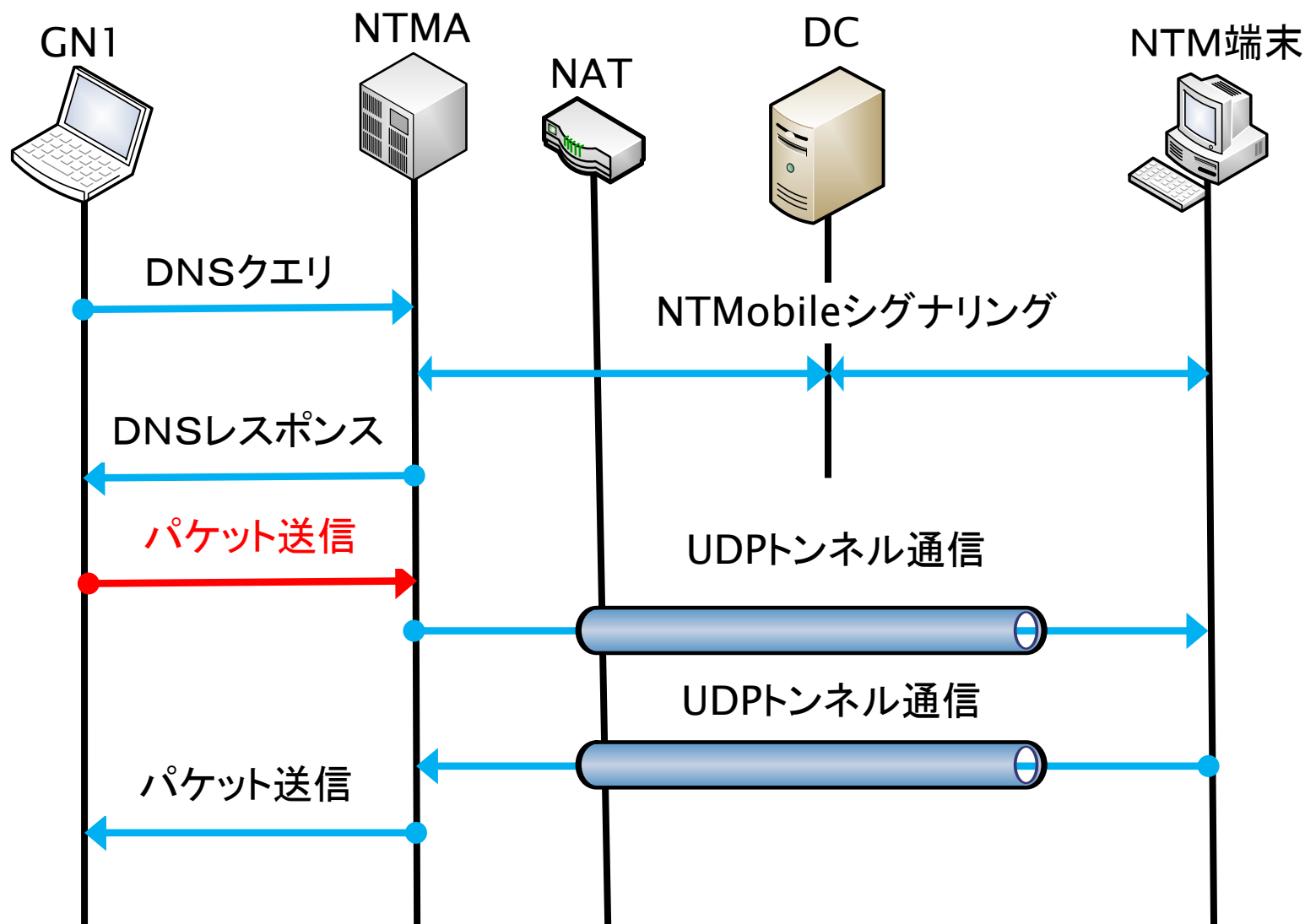


# 提案方式 -動作シーケンス-

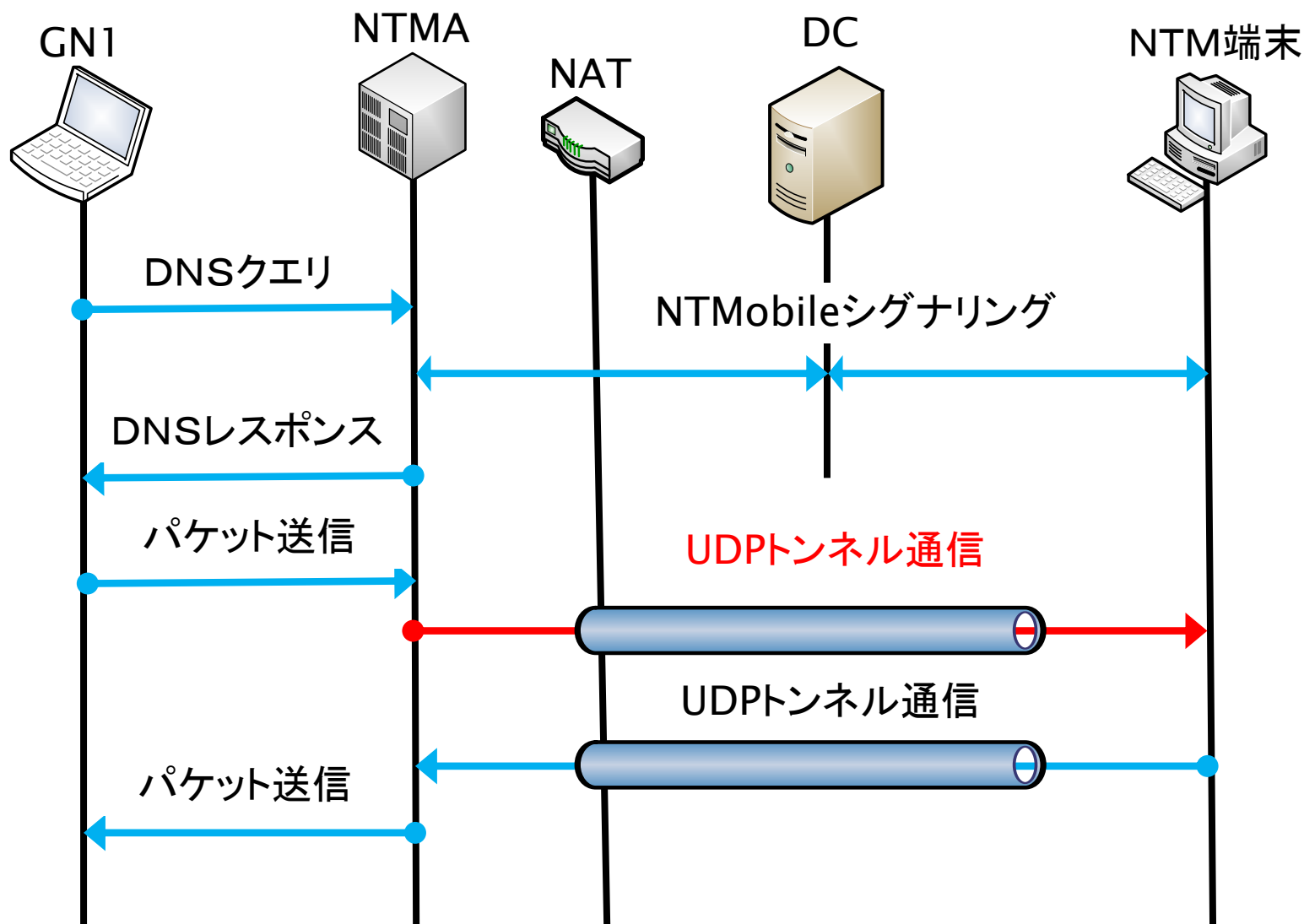




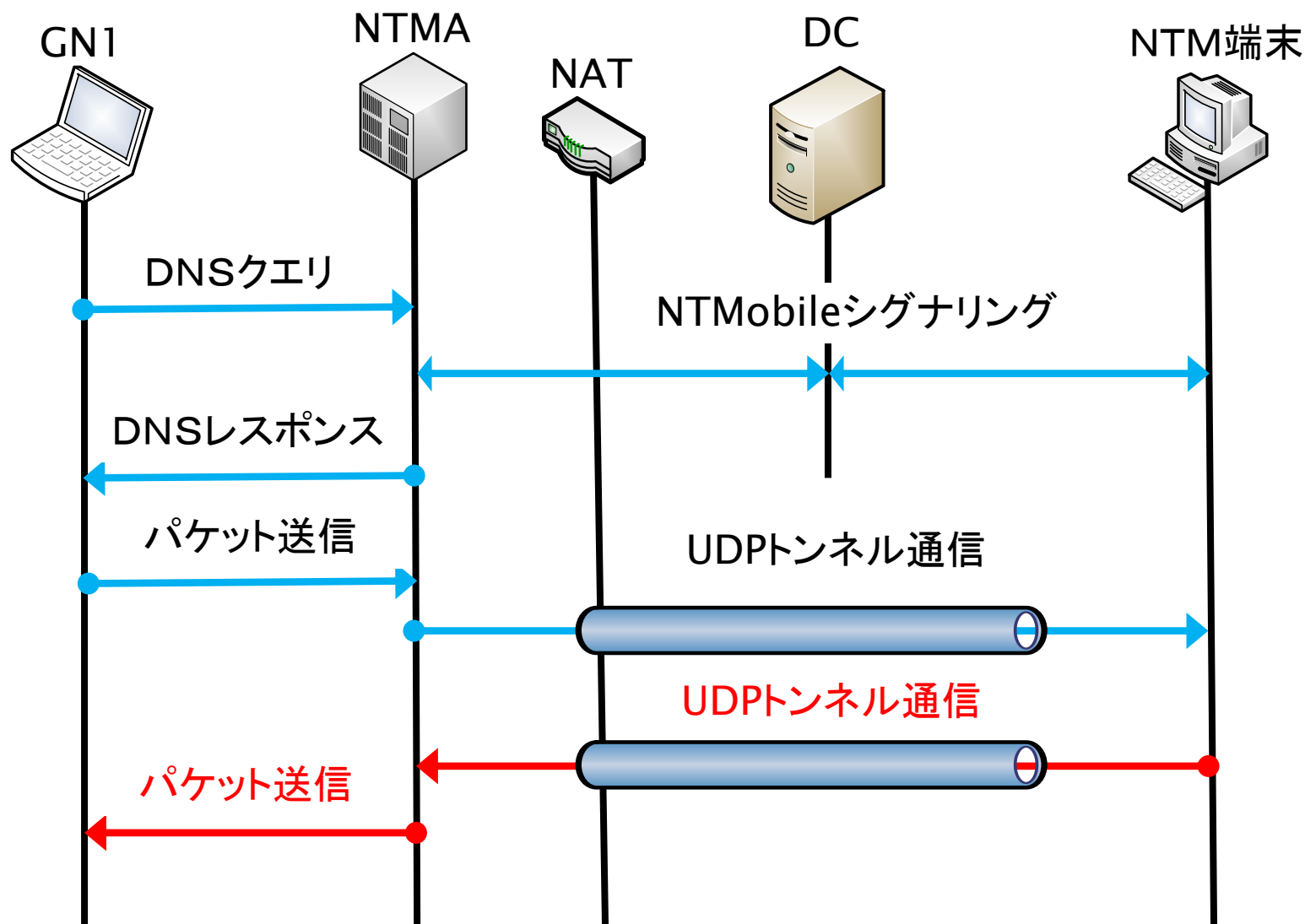
# 提案方式 -動作シーケンス-



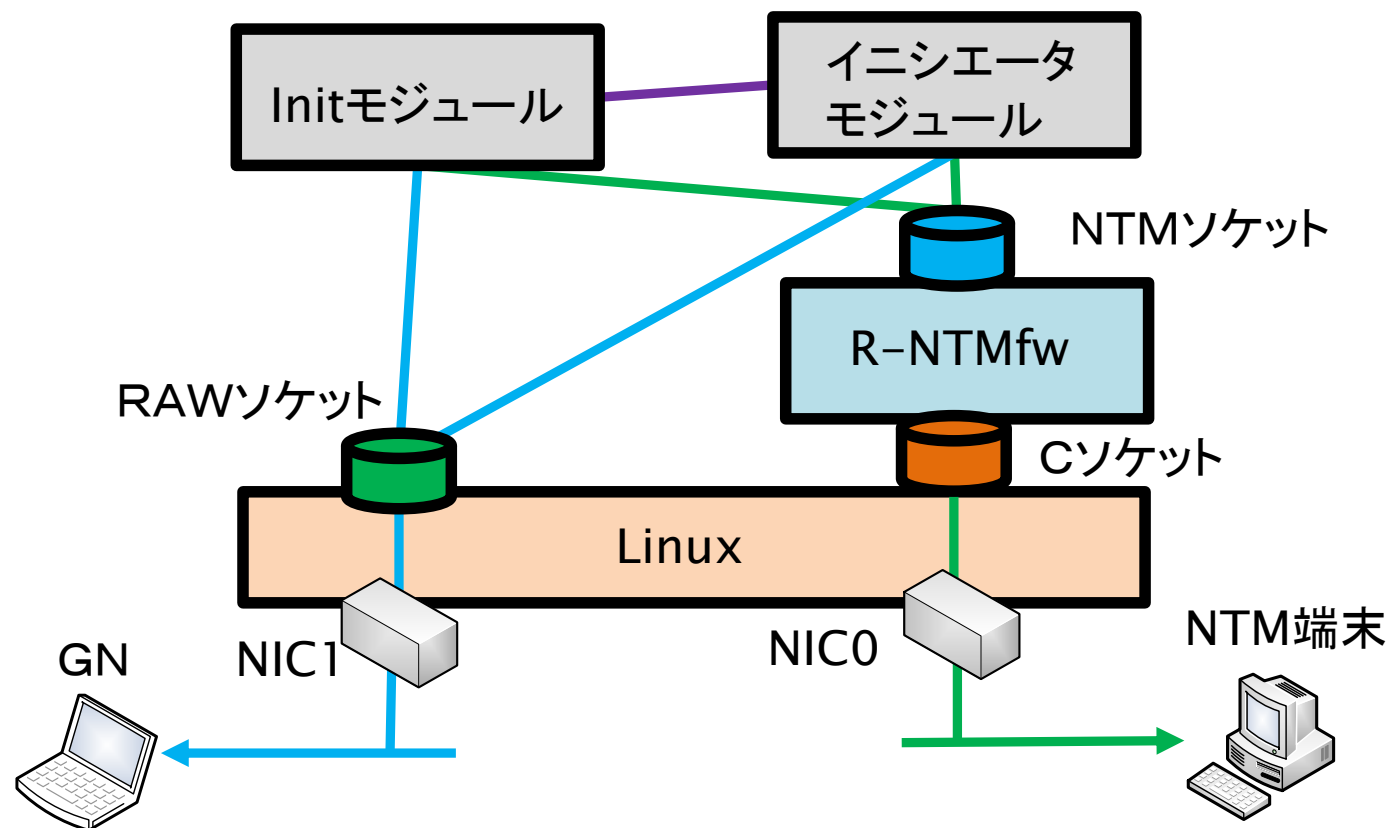
# 提案方式 -動作シーケンス-



# 提案方式 -動作シーケンス-

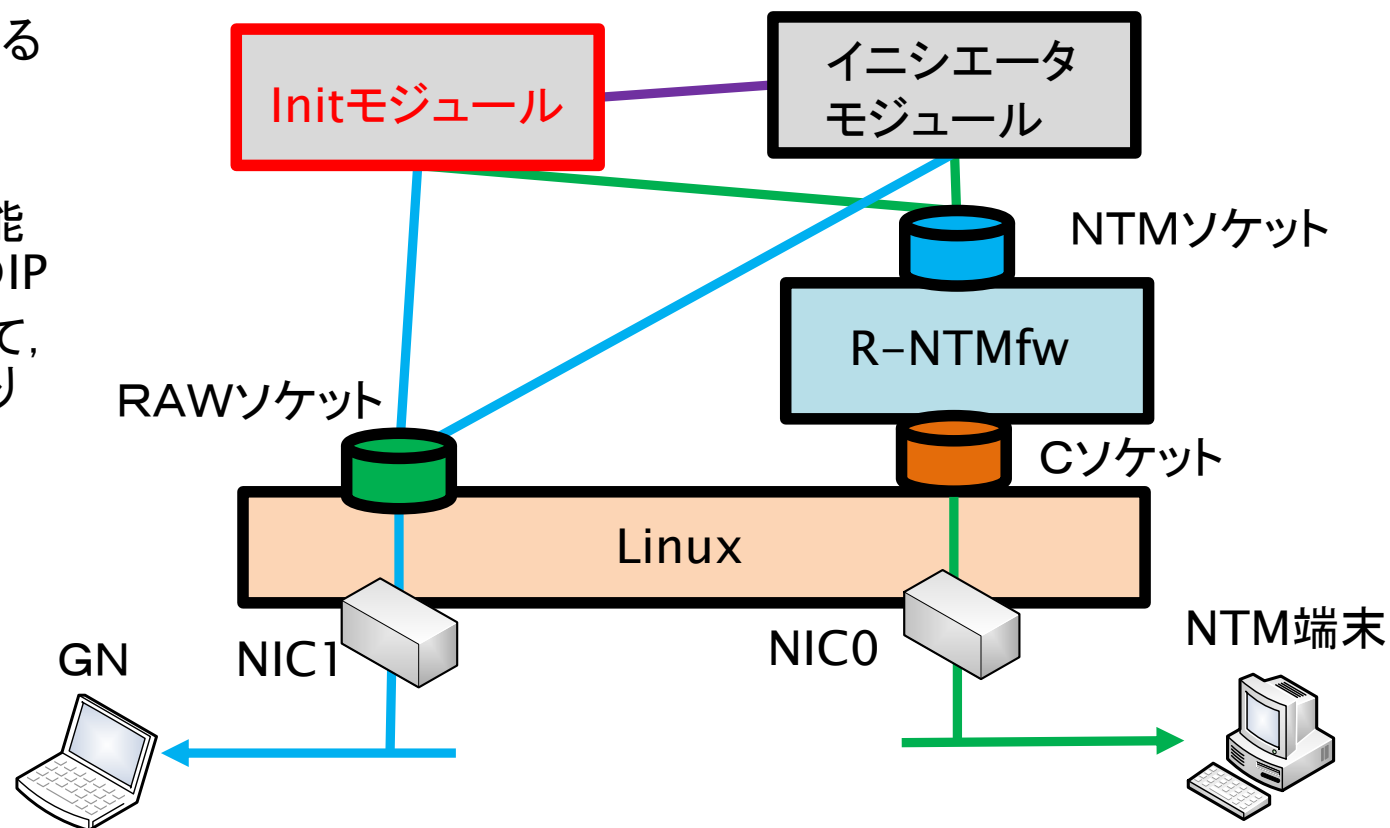


# NTMAのモジュール構成



# NTMAのモジュール構成

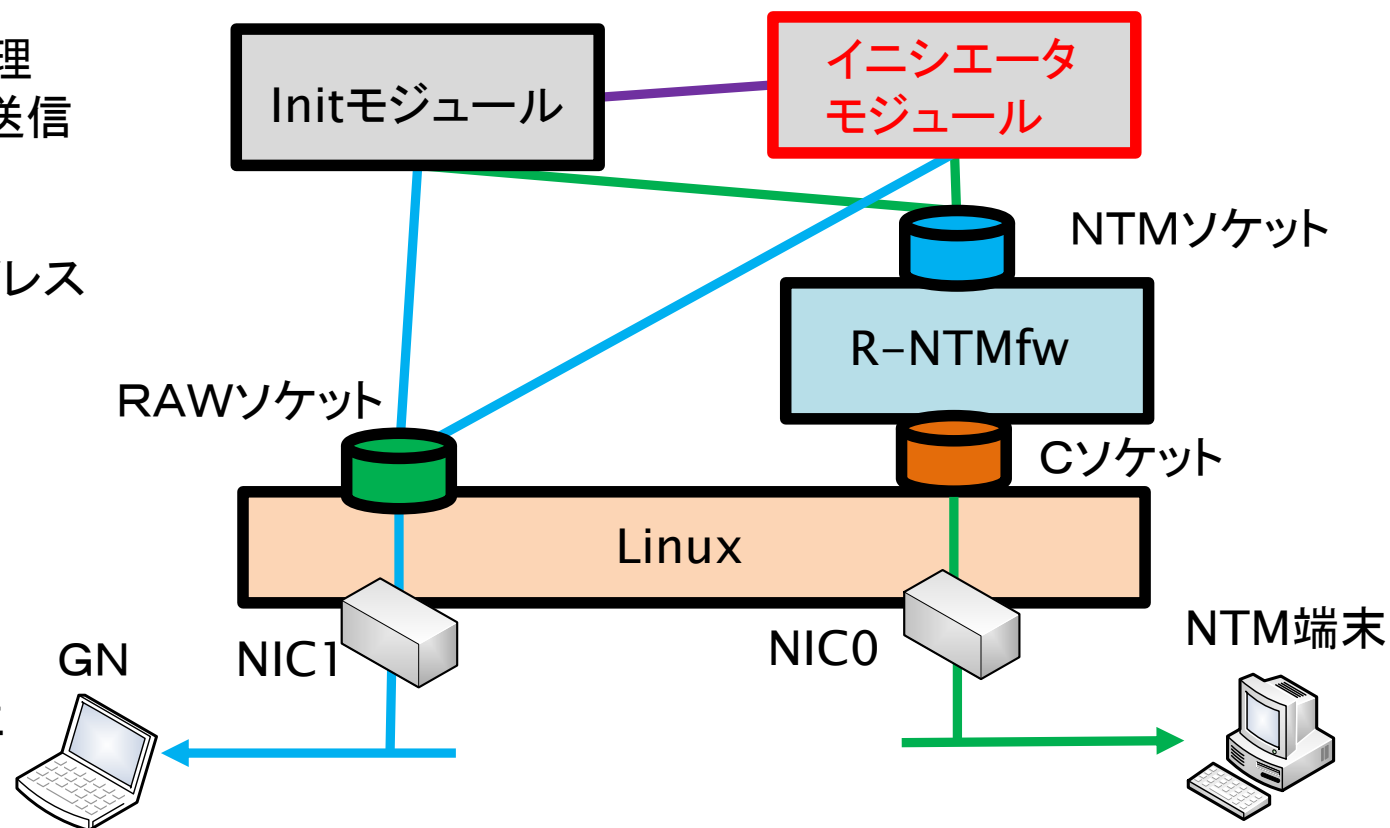
- DCから仮想IPアドレスの割り当てを受ける
- DHCPサーバの機能を包含し、GNからのIPアドレス要求に対して、仮想IPアドレスを割り当てる.



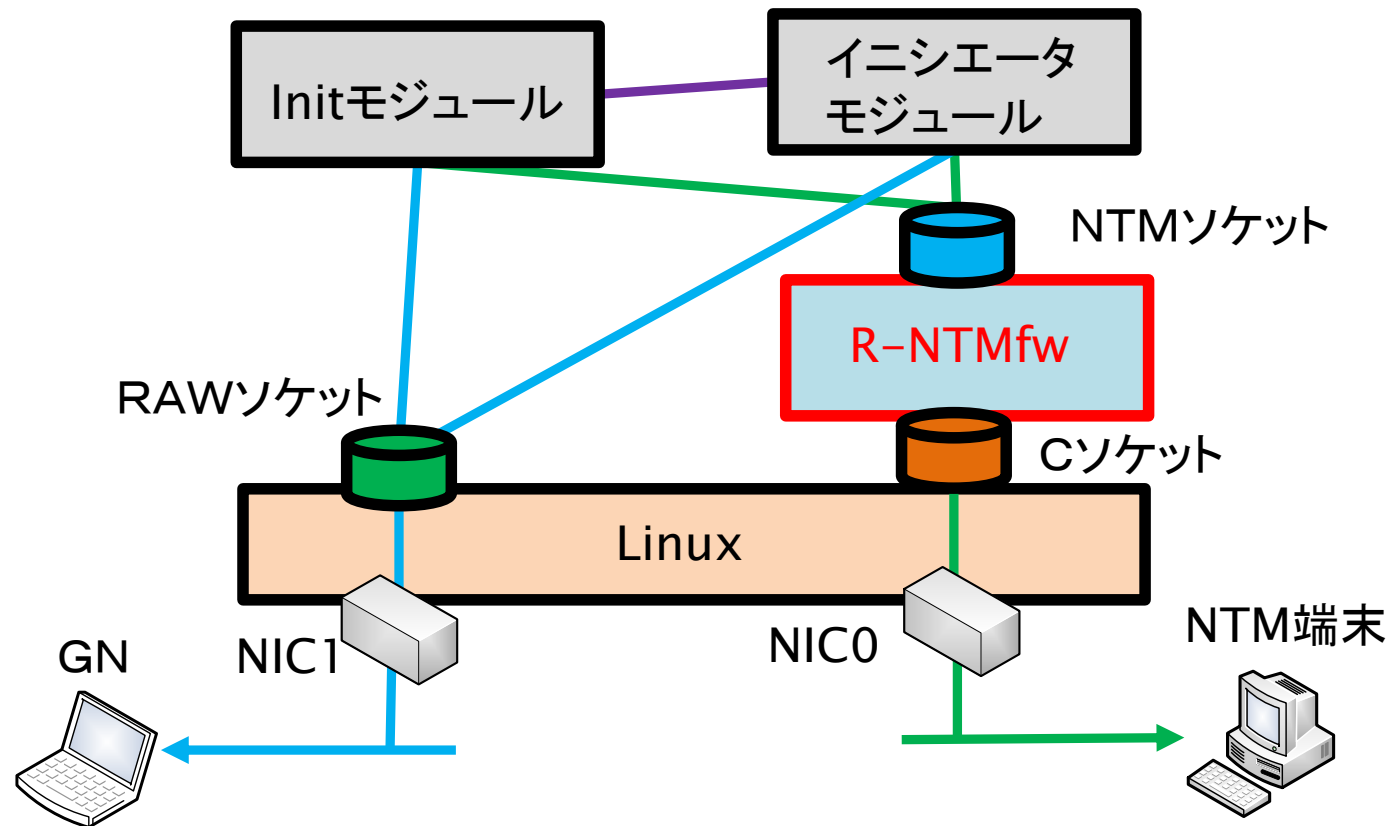
# NTMAのモジュール構成

GNから送信されるパケットを  
RAWソケットで受信

- DNSクエリ
  - トンネル構築処理
  - DNSリプライの送信
- APR
  - 自身のMACアドレス  
でARPリプライ
- UDP/TCPパケット
  - トンネル通信で  
パケットを送信
- DHCP
  - Initモジュールに  
処理を渡す



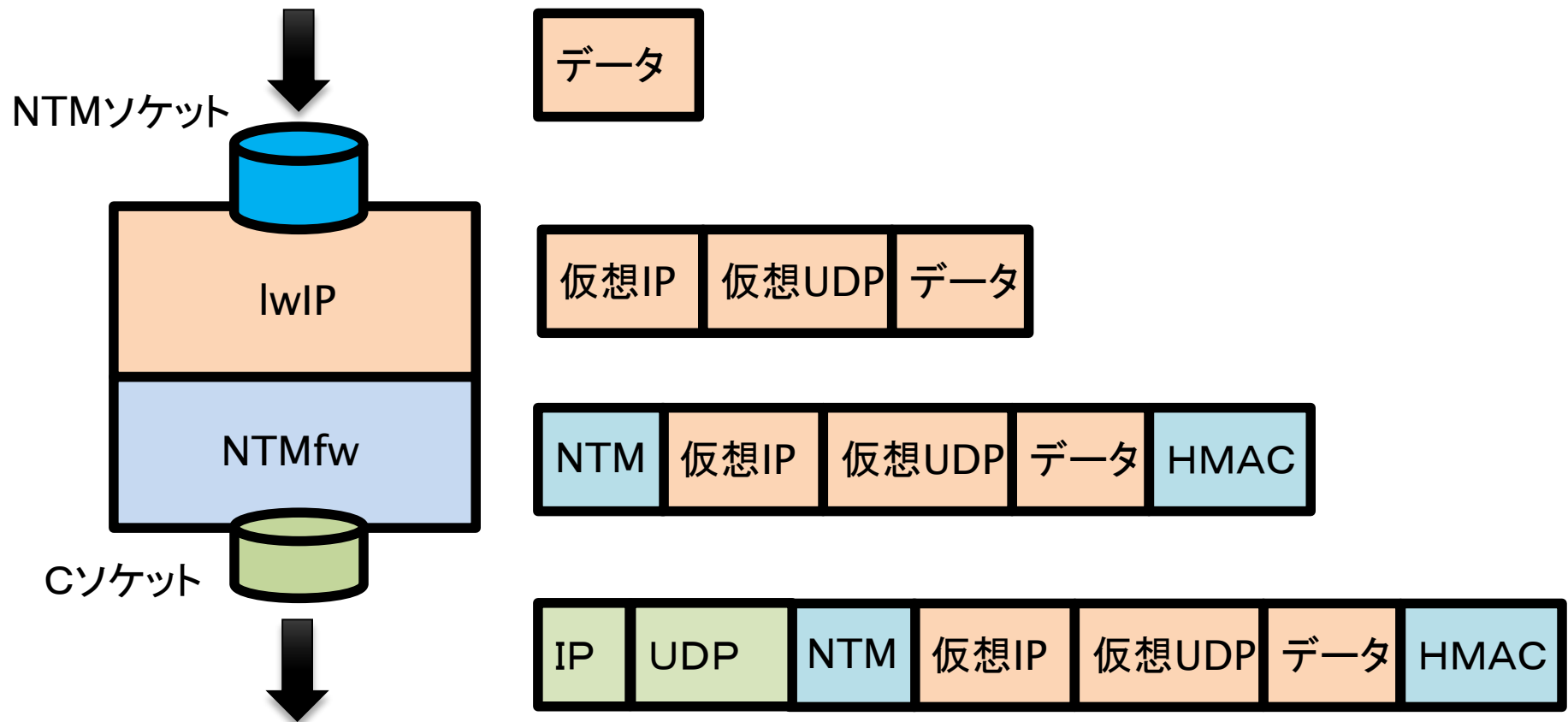
# NTMAのモジュール構成



# R-NTMfw(Remodeled NTMfw)

## ■ NTMfwを改造したアプリケーション

➤ lwIPの処理をスキップ

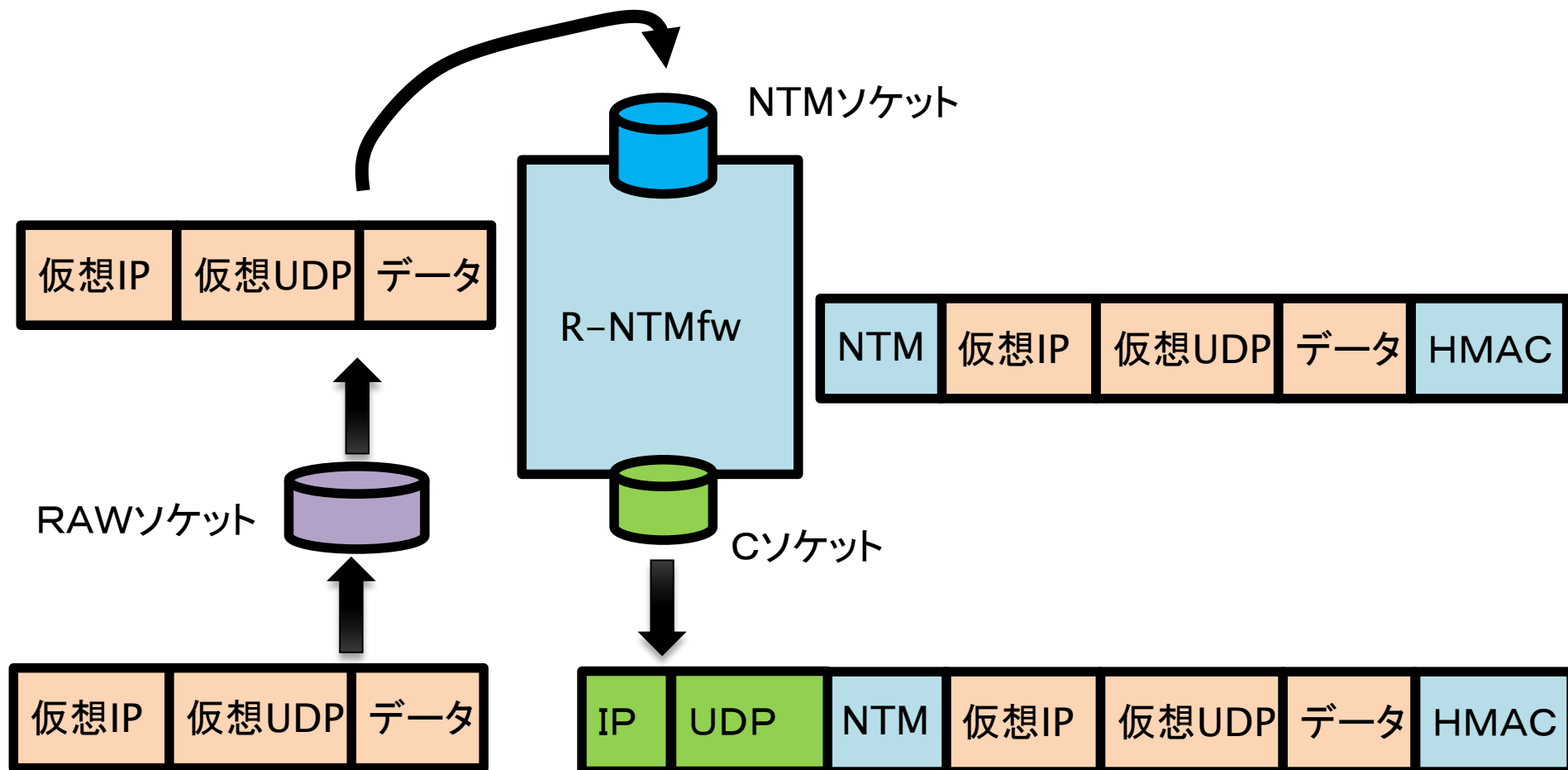




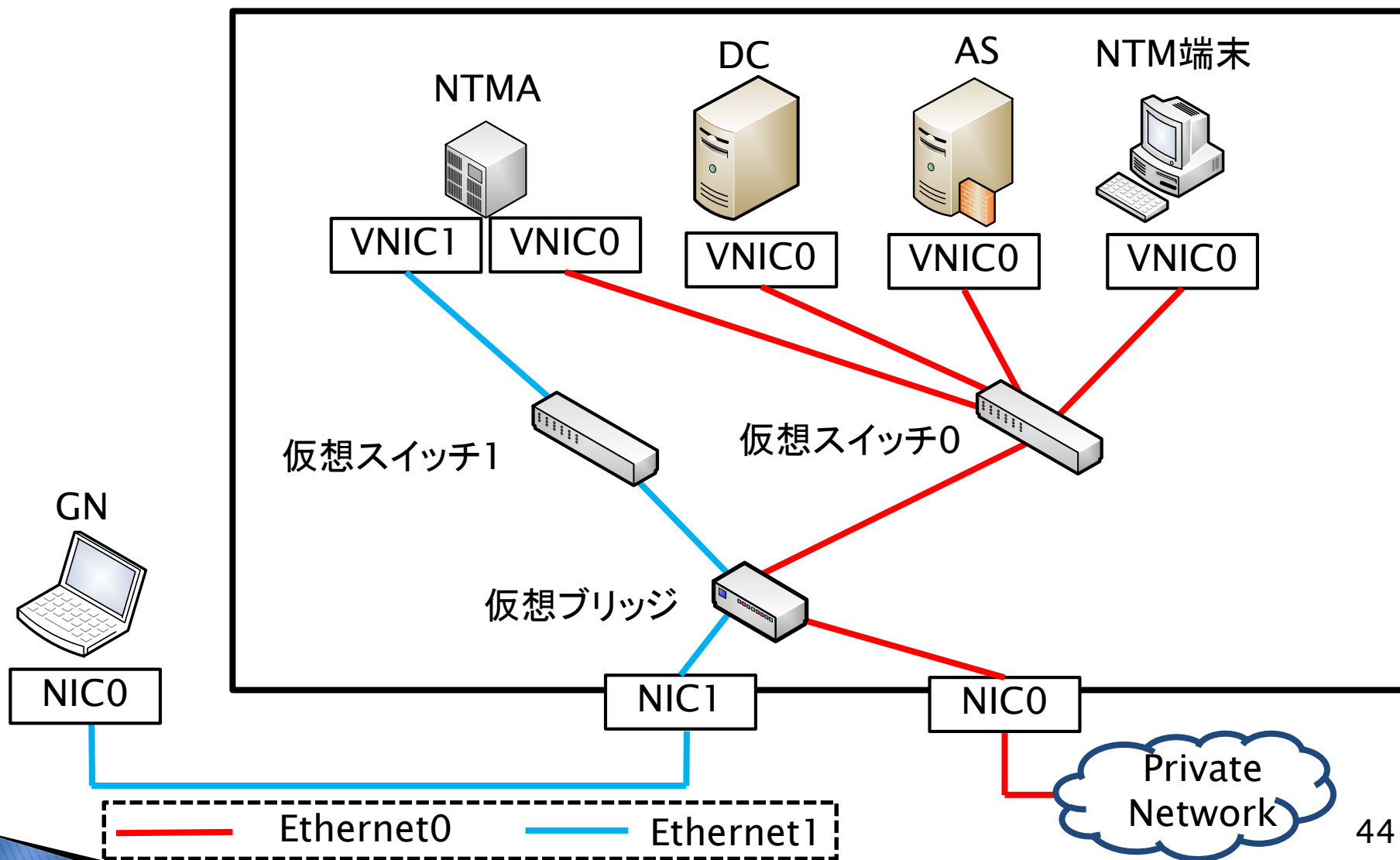
# R-NTMfw(Remodeled NTMfw)

## ■ NTMfwを改造したアプリケーション

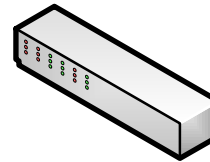
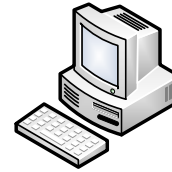
### ➤ lwIPの処理をスキップ



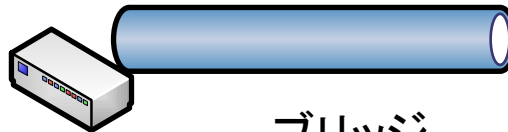
# NICを中心とした構成



# 肥溜め



スイッチ



ブリッジ