NTMobile における移動透過性の実現

佐藤 洸輔 †*，鈴木 秀和 †，内藤 克浩 ‡，渡邊 晃 † († 名城大学，‡ 愛知工業大学)

Realization Method of IP Mobility in NTMobile

Kosuke Sato†, Hidekazu Suzuki†, Naito Katsuhiro‡, Akira Watanabe† (†Meijo University, ‡Aichi Institute of Technology)

1 はじめに

スマートフォンやタブレット端末の普及により，手軽にインターネット接続が可能となった．モバイル端末においては，電波帯域の逼迫により，通信中でも携帯網のトラフィックを任意の Wi-Fi ネットワークにオフロードしたいという要求が出てきている．IP 通信では，一般にネットワークを切り替えると IP アドレスが変わるため，通信が継続できない．そこで，このような場合でも通信を継続できる移動透過性が求められている．NTMobile（Network Traversal with Mobility)[1][2] は移動透過性を実現できる有用な手段である．本稿では NTMobile における移動透過性の実現方法について報告する．

2 NTMobile の概要

NTMobile は，移動を前提とした処理を行う NTM 端末と，Direction Coordinator (DC) により構成される．DC は，NTM 端末の管理，仮想 IP アドレスの管理及び通信経路の管理などを行う．NTMobile では端末の移動にともなう実 IP アドレスの変化を隠蔽するために，アプリケーションはノード識別子である仮想 IP アドレスを用いて通信を行う．このとき，使用する仮想 IP アドレスは DC より配布される．NTM 端末は仮想 IP アドレスによる IP パケットを，位置識別子である実 IP アドレスを用いてカプセル化することによりトンネル通信を行う．そのため，移動して実 IP アドレスが変化してもアプリケーションは同一仮想 IP アドレスを継続して利用可能であり，移動透過性を実現することが可能である．

3 移動透過処理の実際

NTMobile は，エンド端末が NAT 配下に存在していても，エンド端末間で直接通信ができる．Fig.1 に，NTMobile の移動透過性の実現方法を示す． 通信開始側 NTM 端末を MN，通信相手側の NTM 端末を CN とする．MN の移動前実 IP アドレスを RIP1，仮想 IP アドレスを VIP1，CN の実 IP アドレスを RIP2，仮想 IP アドレスを VIP2，MN の移動後実 IP アドレスを RIP3 とする．Fig.1 は，MN 側から通信を開始し，MN が通信中に移動した場合の例である．MN，CN ともに NAT 配下に存在する場合が多いが，Fig.1 では NAT の記述は省略してある．DC はグローバル空間に設置されている必要がある．DC には MN，CN の実 IP アドレス，仮想 IP アドレス，NAT の IP アドレス，ノード ID などのアドレス情報がすでに登録されているものとする．

NTMobile では，NTM 端末間の通信識別子として，Path ID を用いる．Path ID は，MN と CN の仮想アドレスペアから生成する．MN と CN は Path ID を検索キーとしたトンネルテーブルを持つ．トンネルテーブルにはカプセル化に使う実 IP アドレスが格納されている．移動時は，トンネルテーブルを書き換える必要があるが，このとき，確実な排他制御を行う必要がある．

Fig.1 において，MN は通信開始時に Path ID を生成し，DC に対して，CN との通信に係る経路指示要求を送信する．DC は MN と CN の位置を事前に把握しているため，MN と CN に経路

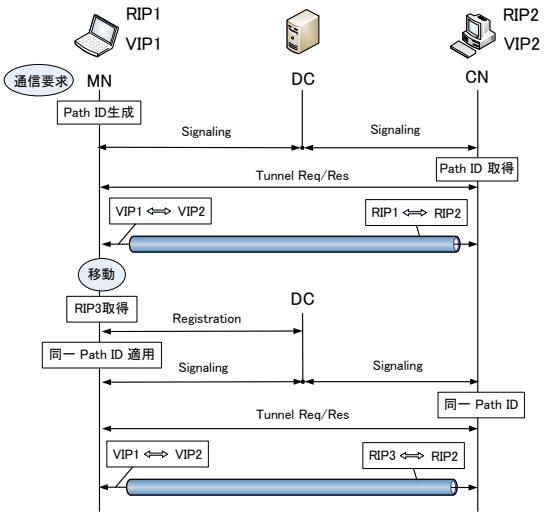


Fig. 1 Sequence of IP Mobility in NTMobile

指示ができる．経路指示時に Path ID を CN へ送り，PathID を共有する．以上の動作を Fig.1 では，Signaling と記載している．MN と CN は DC の指示に従ってトンネル経路を生成する．トンネル生成は NAT 配下から開始する必要がある．MN と CN が両方とも NAT 配下の場合には，第三の装置 Relay Server(RS) を経由する必要があるが，Fig.1 では簡単のため RS を省略している．トンネル経路生成後の MN と CN 間の通信はすべてカプセル化される．すなわち，仮想 IP アドレス VIP1 ⇔ VIP2 として生成された通信パケットが実 IP アドレス RIP1 ⇔ RIP2 でカプセル化される．

MN が通信中に移動すると，実ネットワークに設置されている DHCP サーバから RIP3 を取得する．MN は自身のインタフェースの IP アドレスを監視しており，IP アドレスの変化を検出した場合，移動したものと判断する．MN は，新たな IP アドレスを DC に Registration として通知する．次に，移動前と同一の Path ID を用いて，通信開始時と全く同じ Signaling 処理および，トンネル経路生成を実行する．これにより，RIP3 ⇔ RIP2 による新しいトンネル経路が生成される．ここで，カプセル内のパケットは VIP1 ⇔ VIP2 のまま変化しないため上位アプリケーションは実 IP アドレスが変化したことに気づくことなく通信は継続される．

4 まとめ

本稿では，NTMobile における移動透過性の実現方法について示した．NTMobile では，仮想 IP アドレスによる IP パケットを実 IP アドレスを用いてカプセル化通信を行うことにより移動透過性を実現している．

文 献

[1] 鈴木．他：情報処理学会論文誌 Vol.54, No.1, pp.367–379, 2013.
[2] 上酔尾．他：情報処理学会論文誌 Vol.54, NO.10, pp.2288-2299, 2013

NTMobileにおける移動透過性の実現

佐藤 洸輔†, 鈴木 秀和†, 内藤 克浩ǂ, 渡邊　晃†
†名城大学　理工学部 情報工学科
ǂ愛知工業大学　情報科学部

Watanabe Lab.

研究背景

- スマートフォンやタブレット端末の普及
  - ➢電波帯域が逼迫
    - Wi-Fiへのオフロード
- Wi-Fi, LTE, 3Gなど接続先が変化
  - ➢IPアドレスが変化
    - 通信が切断

移動透過性の必要性

NTMobileの概要

- *NTMobile（Network Traversal with Mobility）
  - 通信接続性と移動透過性を同時に実現する技術

- 通信接続性の実現
  - 経路指示装置の導入
    - NAT配下の端末に対して通信開始が可能
    - IPv4 – IPv6間通信が可能

- 移動透過性の実現
  - 位置に依存しないIPアドレス（仮想IPアドレス）を導入
    - IPアドレスの変化をアプリケーションから隠蔽

*鈴木 秀和.他 “NTMobileにおける通信接続性の確立手法と実装” 情報処理学会論文誌, Vol.54, No.1pp-367-379(2013).

NTMobileの課題

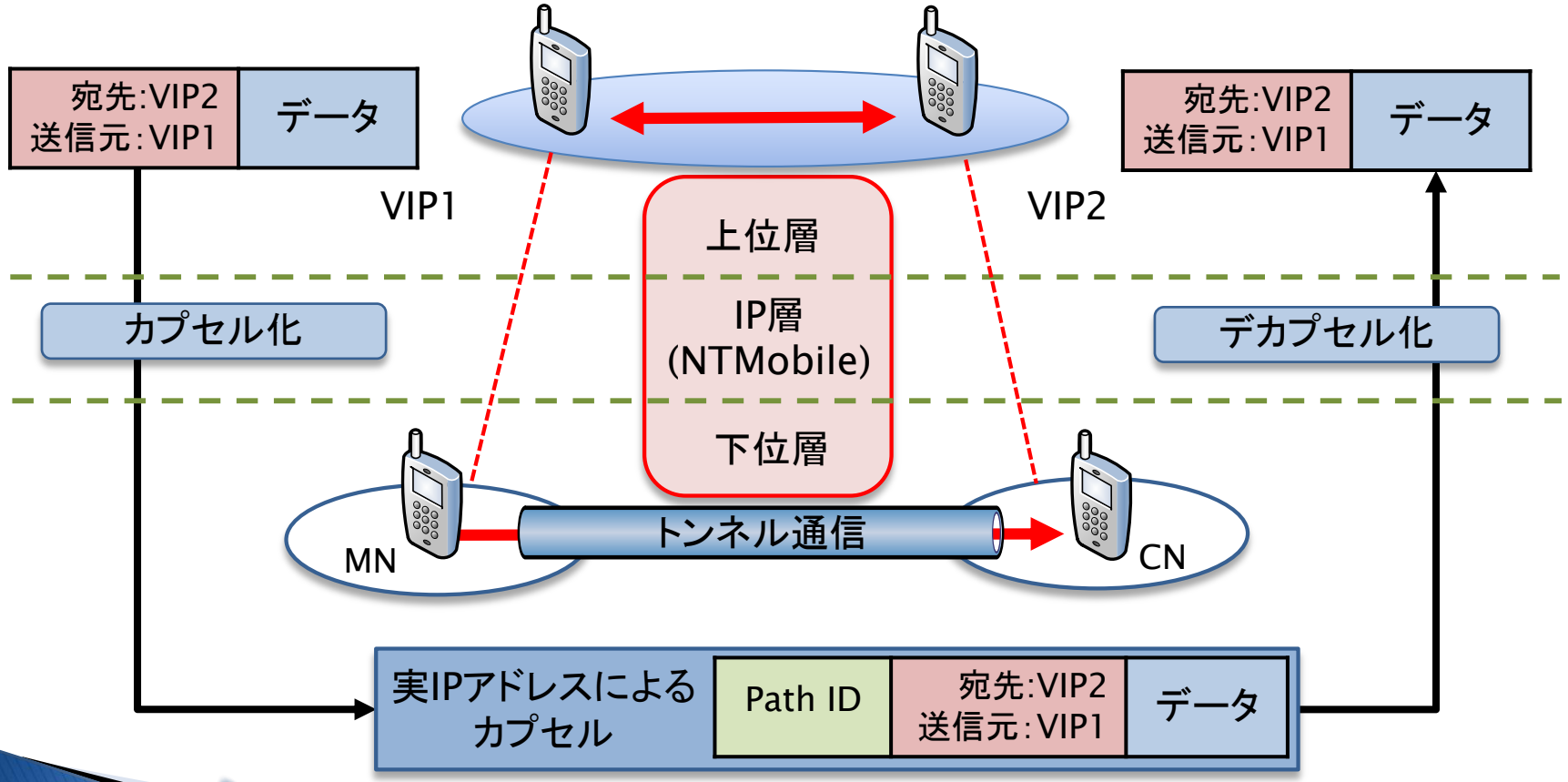
- NTMobileの基本機能は移動処理も含めカーネル実装版により確認済み
  - スマートフォンではルート権限が必要

NTMobile機能をアプリケーションで実装

NTMobile自体の基本シーケンスの見直しが必要

NTMobileの通信

- アプリケーションは仮想IPアドレスを認識
  - 実IPアドレスの変化をアプリケーションから隠蔽



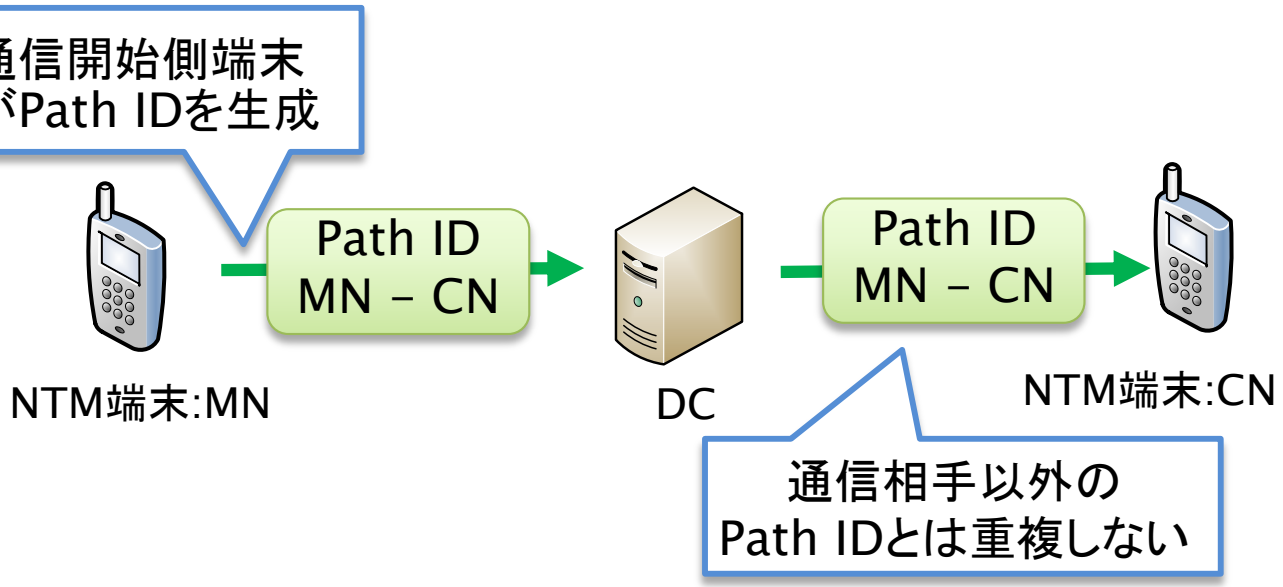
MN(Mobile Node)
CN(Correspondent Node)

仮想IPアドレス管理方法

- 通信の識別
  - NTMobileの通信は通信識別子Path IDのみで識別することが可能

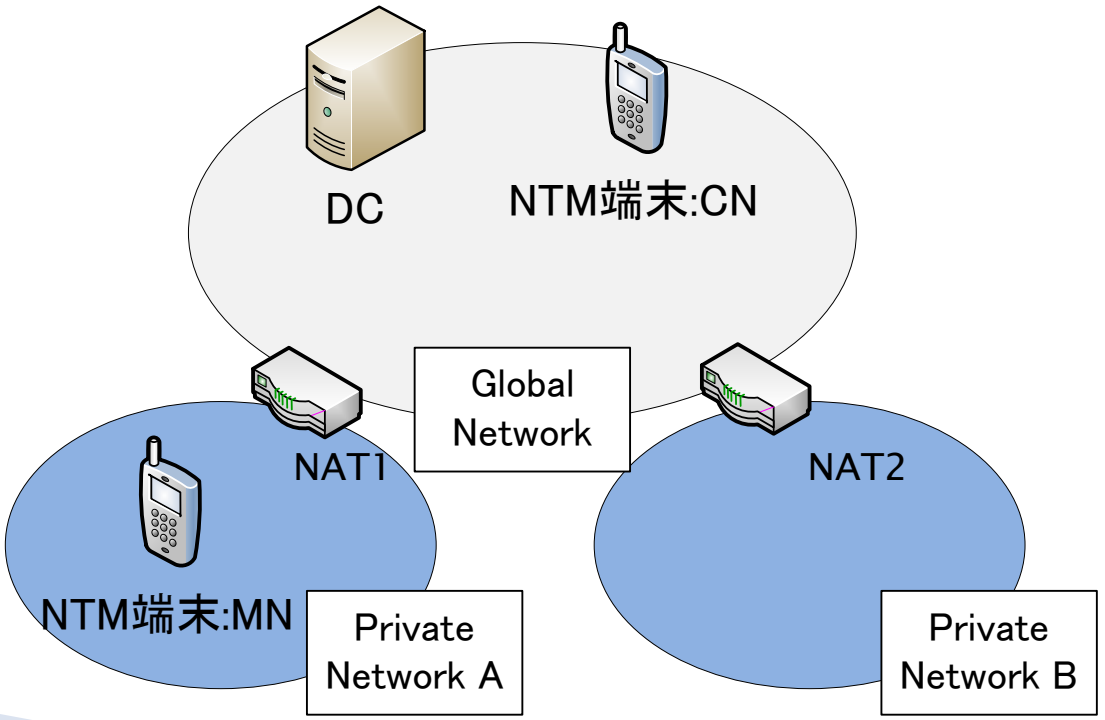
Path ID

NTMobileの通信を一意に識別する通信識別子（128bit）



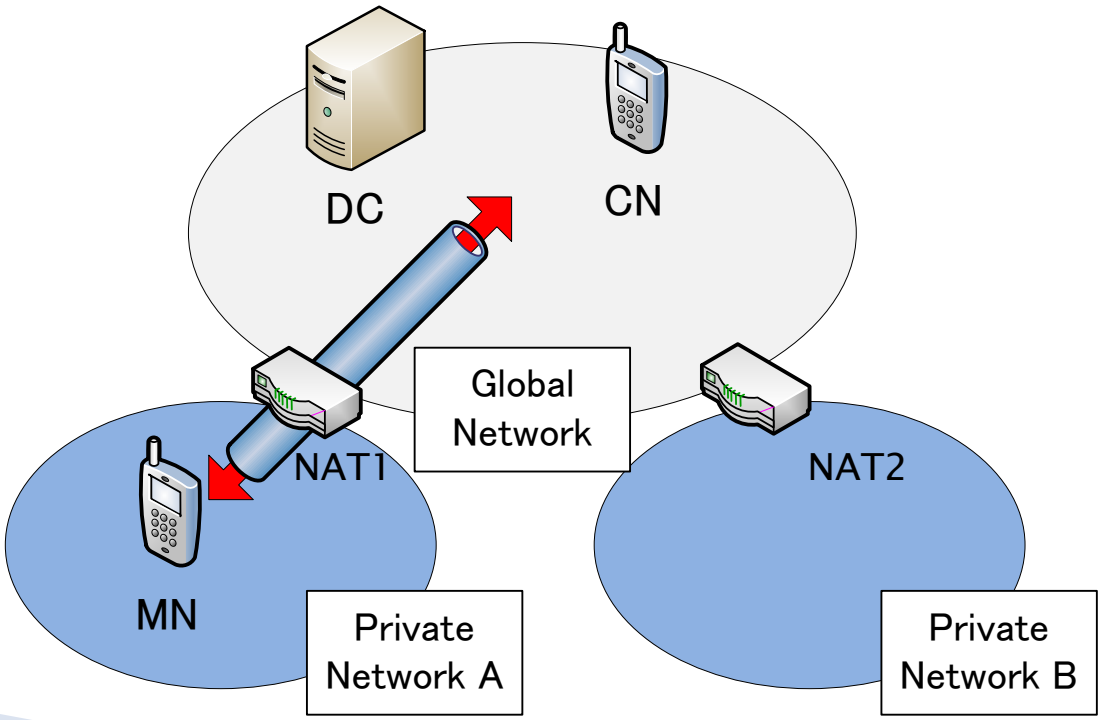
NTMobileの構成

- NTM端末
  - NTMobileを実装した端末
- DC(Direction Coordinator)
  - 仮想IPアドレスの管理
  - 通信経路指示



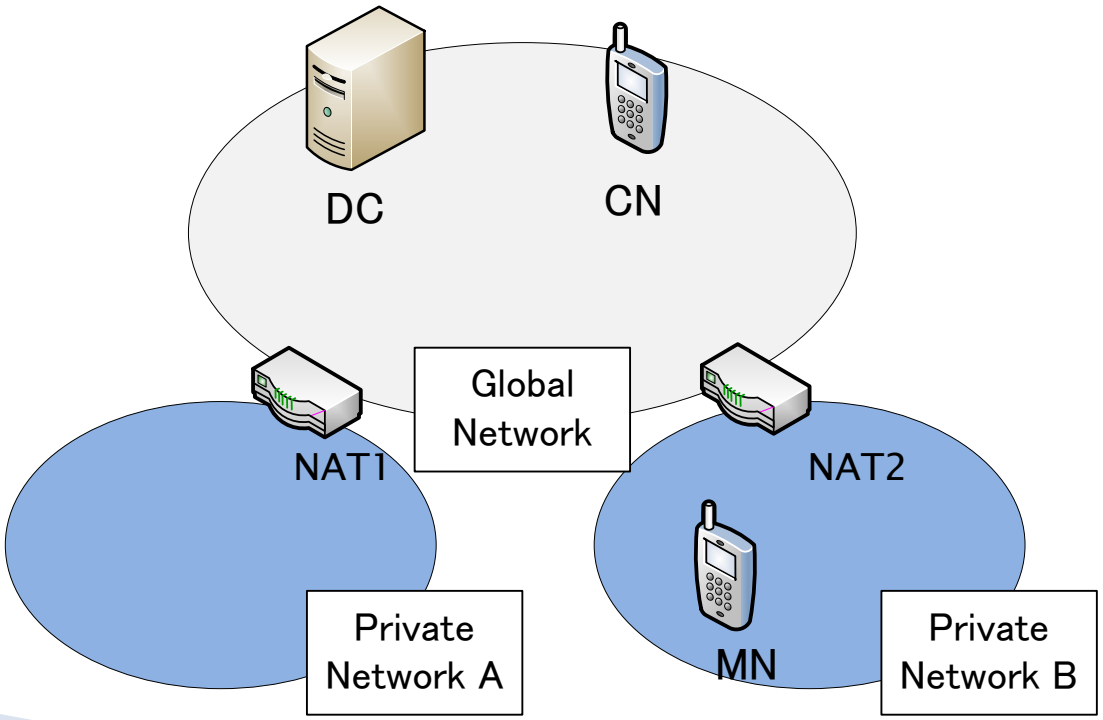
移動透過性の実現

- 移動透過性の実現
  - 通信に仮想IPアドレス(Virtual IP address)を使用
  - 全通信を実IPアドレス(Real IP address)でUDPカプセル化
  - DCが通信端末間の
    トンネル構築を指示



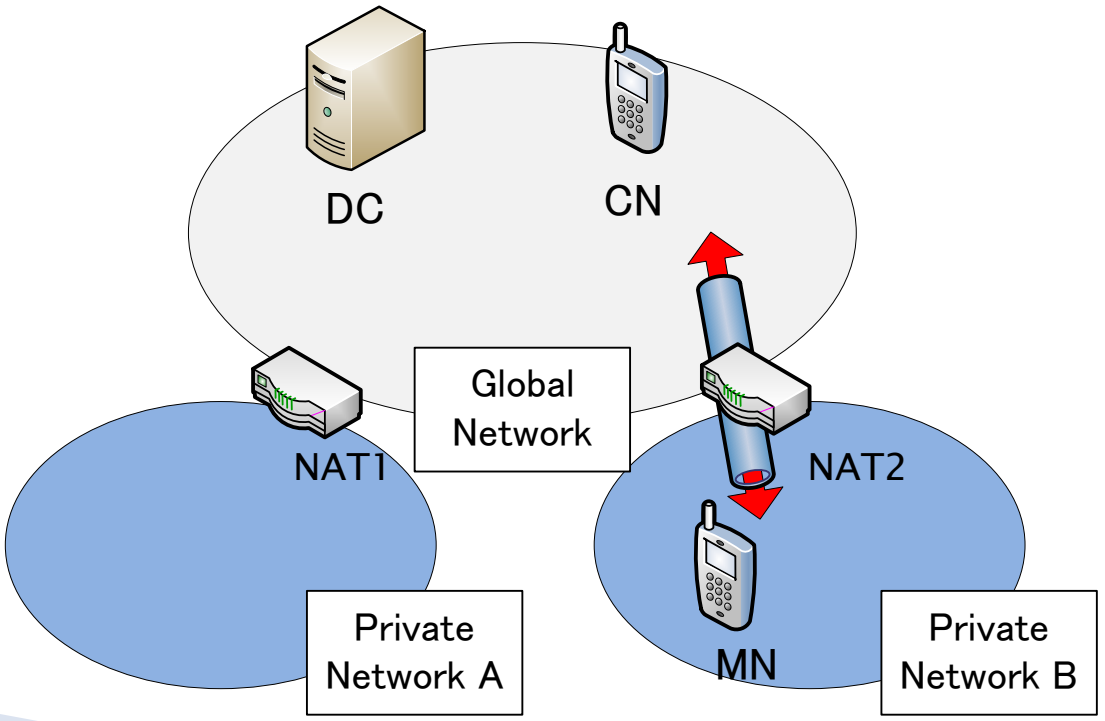
移動透過性の実現

- 移動透過性の実現
  - 通信に仮想IPアドレス(Virtual IP address)を使用
  - 全通信を実IPアドレス(Real IP address)でUDPカプセル化
  - DCが通信端末間の
    トンネル構築を指示

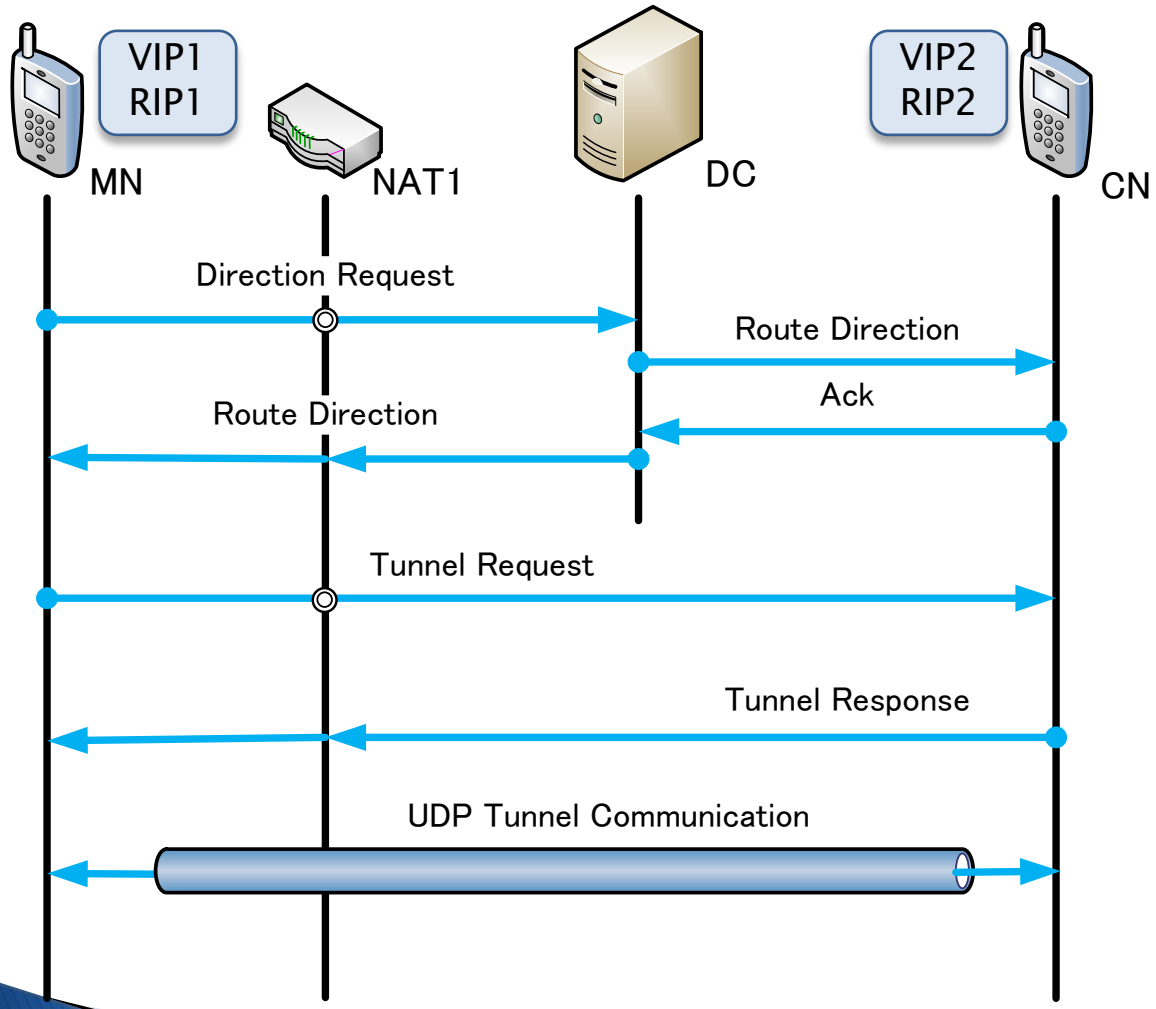


移動透過性の実現

- 移動透過性の実現
  - 通信に仮想IPアドレス(Virtual IP address)を使用
  - 全通信を実IPアドレス(Real IP address)でUDPカプセル化
  - DCが通信端末間の
    トンネル構築を指示



シーケンス（通信開始時）



Path ID 生成

通信識別子

DCへ経路指示要求

CNへ経路指示

CNがPath ID を取得

MNへ経路指示

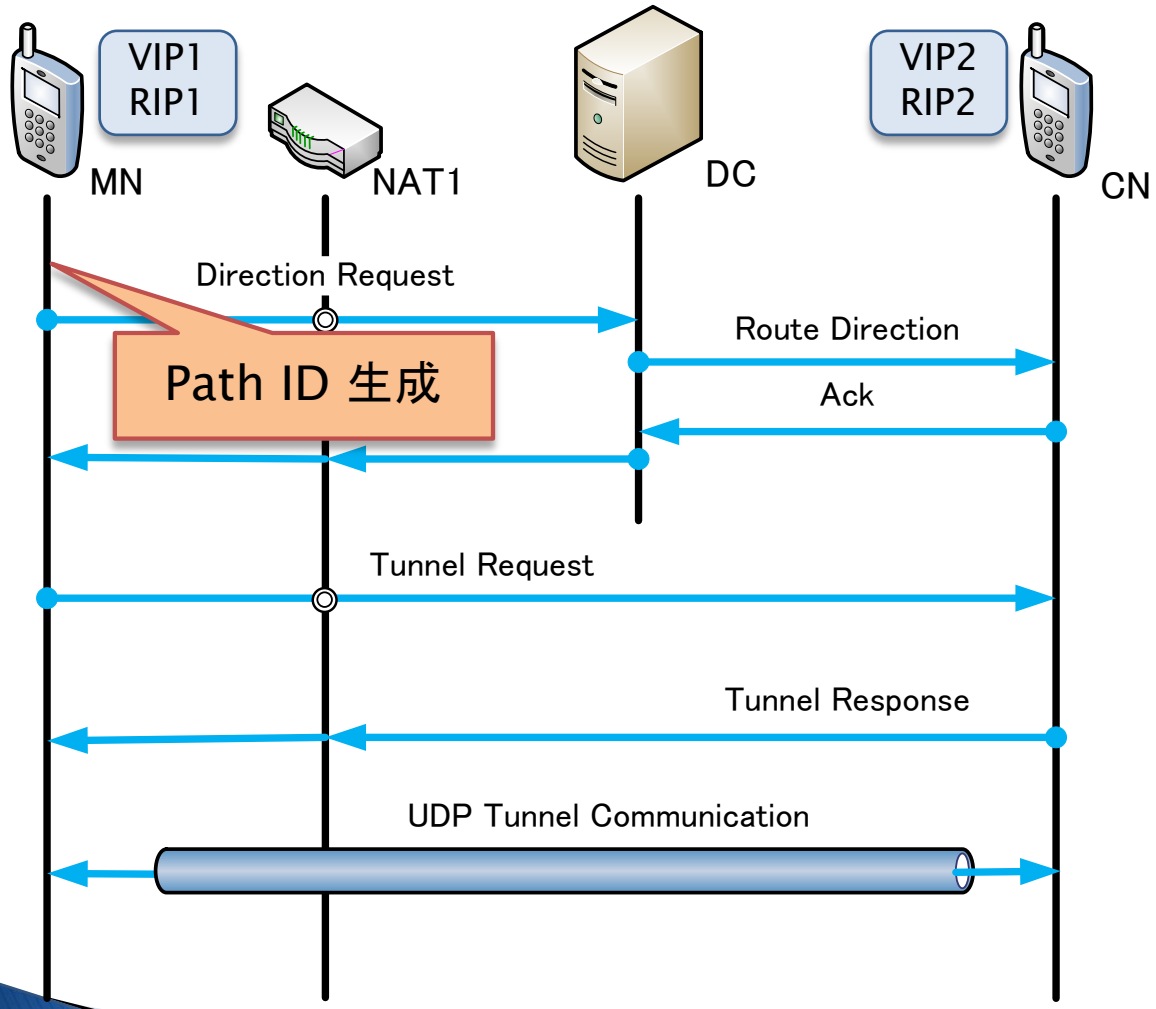
トンネル要求

トンネル応答

トンネル通信

RIP1 ⇔ RIP2
VIP1 ⇔ VIP2

シーケンス（通信開始時）



Path ID 生成

通信識別子

DCへ経路指示要求

CNへ経路指示

CNがPath ID を取得

MNへ経路指示

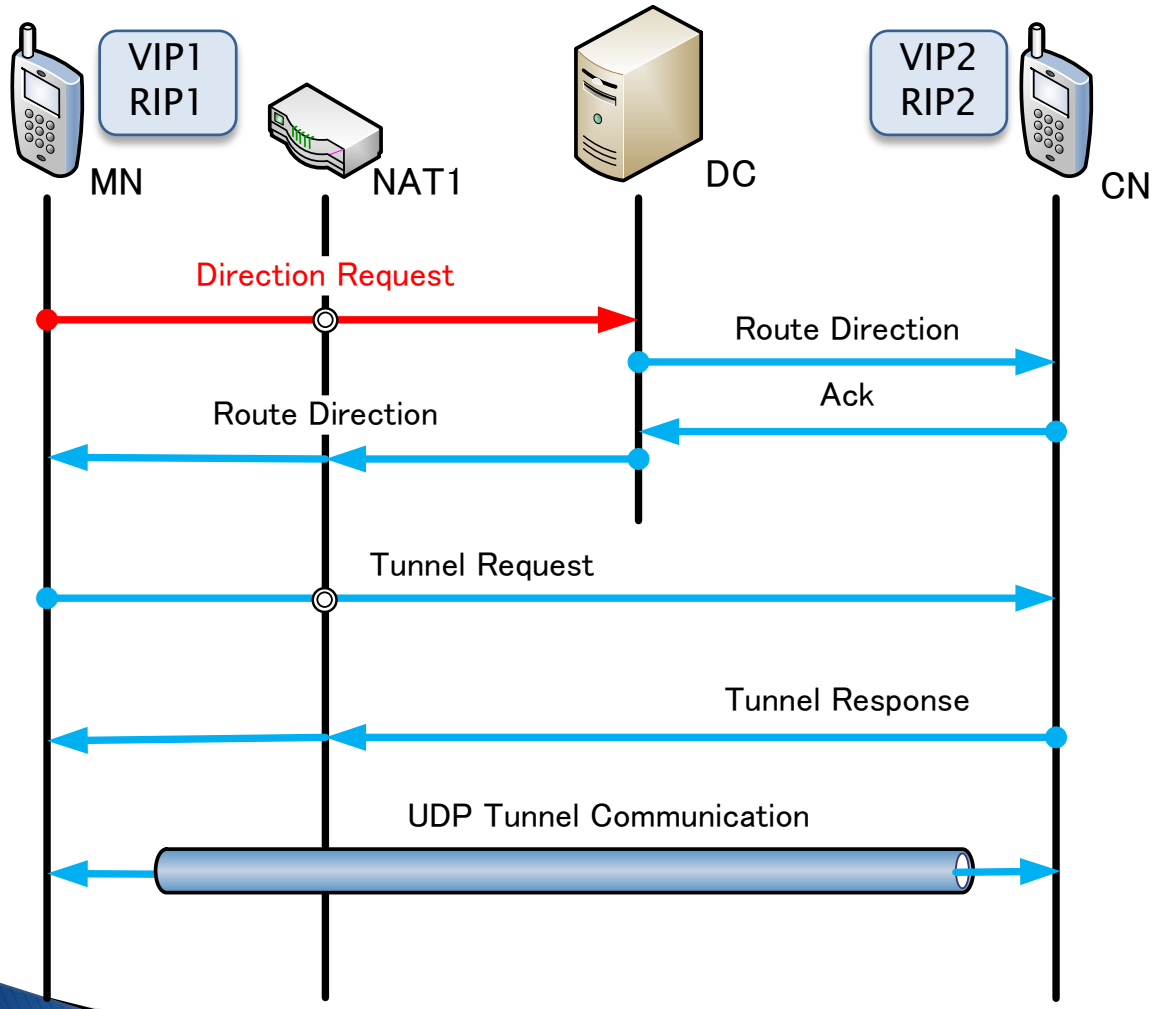
トンネル要求

トンネル応答

トンネル通信

RIP1 ⇔ RIP2
VIP1 ⇔ VIP2

シーケンス（通信開始時）



Path ID 生成

通信識別子

DCへ経路指示要求

CNへ経路指示

CNがPath ID を取得

MNへ経路指示

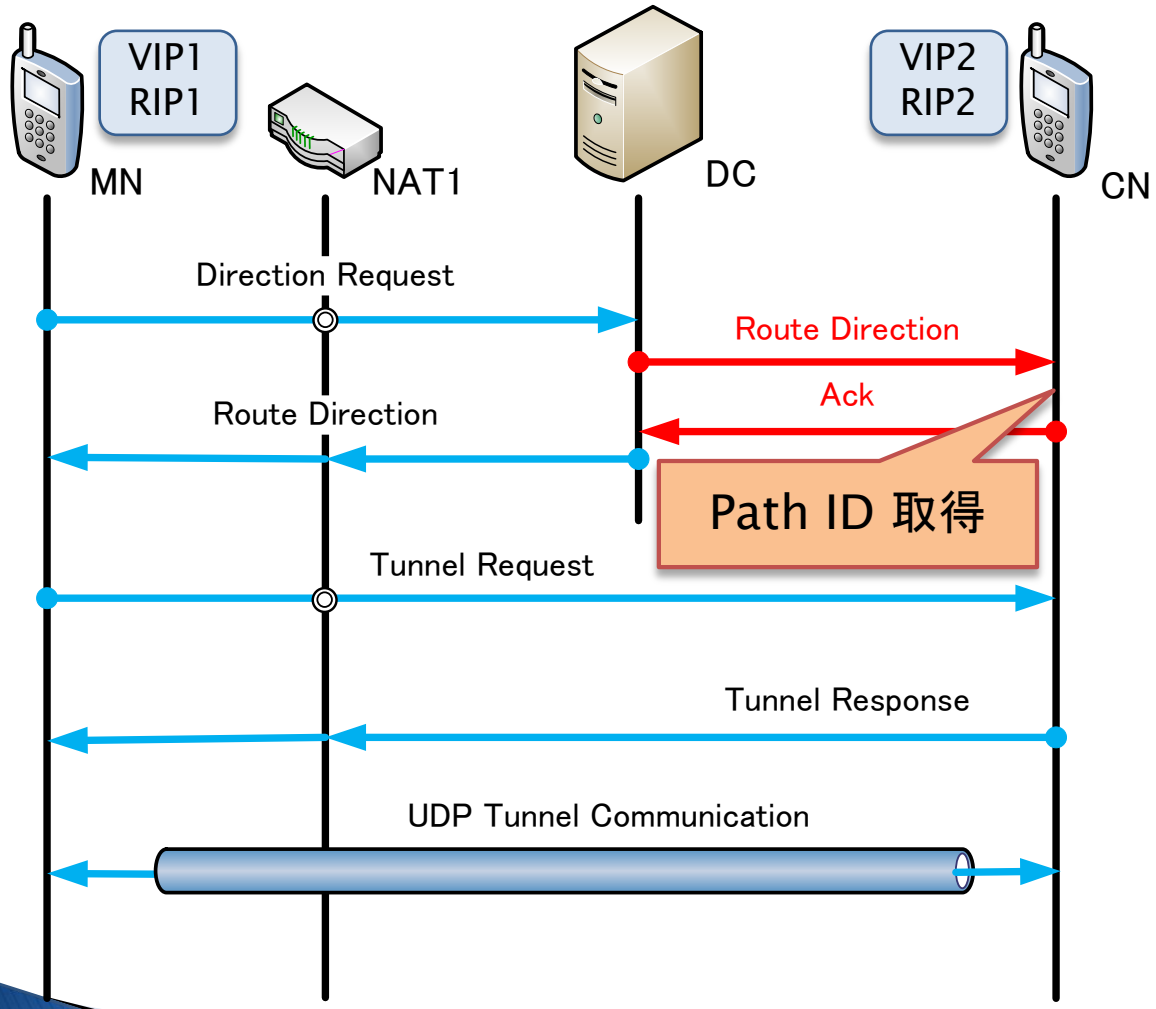
トンネル要求

トンネル応答

トンネル通信

RIP1 ⇔ RIP2
VIP1 ⇔ VIP2

シーケンス（通信開始時）



Path ID 生成

通信識別子

DCへ経路指示要求

CNへ経路指示

CNがPath ID を取得

MNへ経路指示

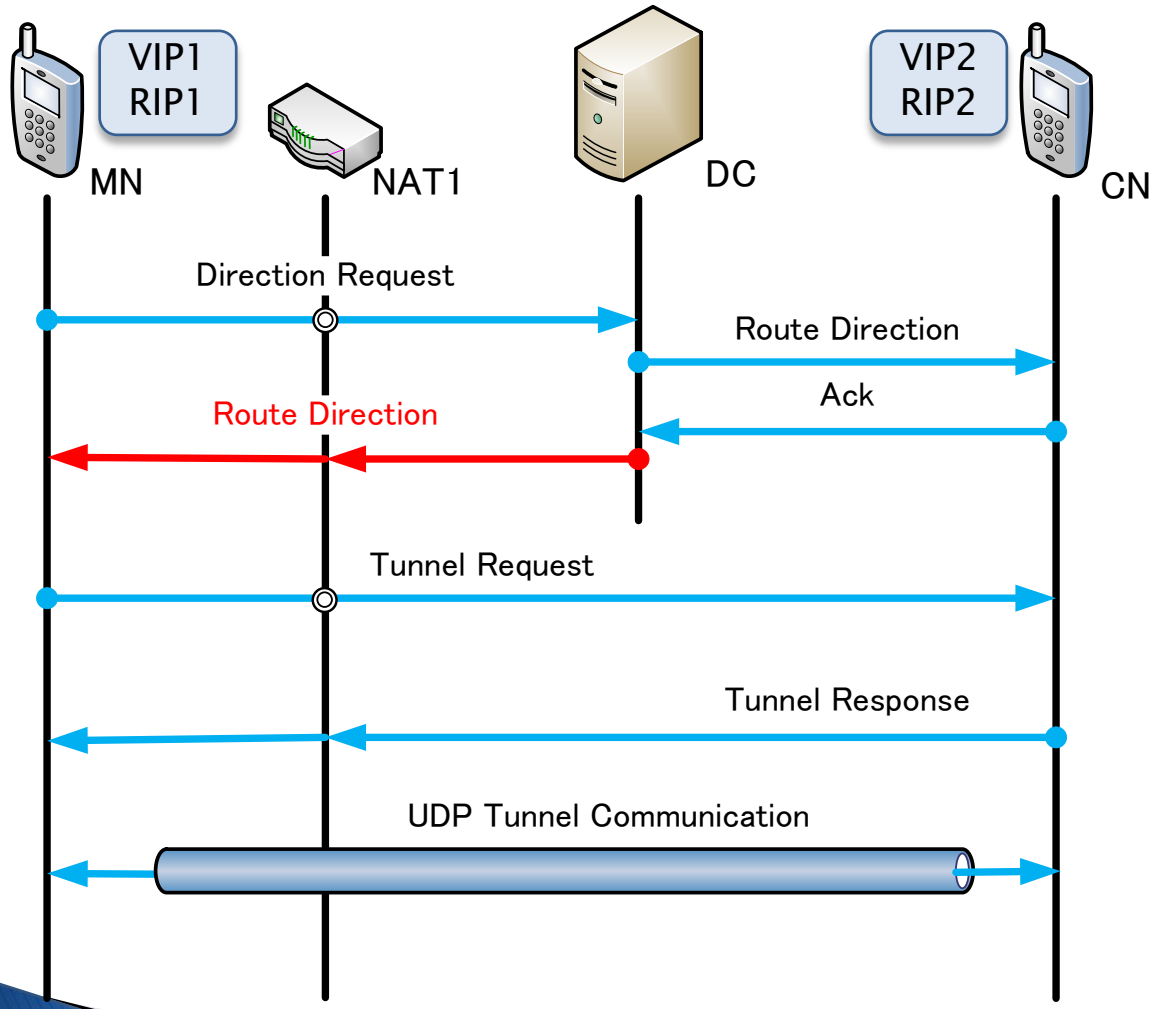
トンネル要求

トンネル応答

トンネル通信

RIP1 ⇔ RIP2
VIP1 ⇔ VIP2

シーケンス（通信開始時）



Path ID 生成

通信識別子

DCへ経路指示要求

CNへ経路指示

CNがPath ID を取得

MNへ経路指示

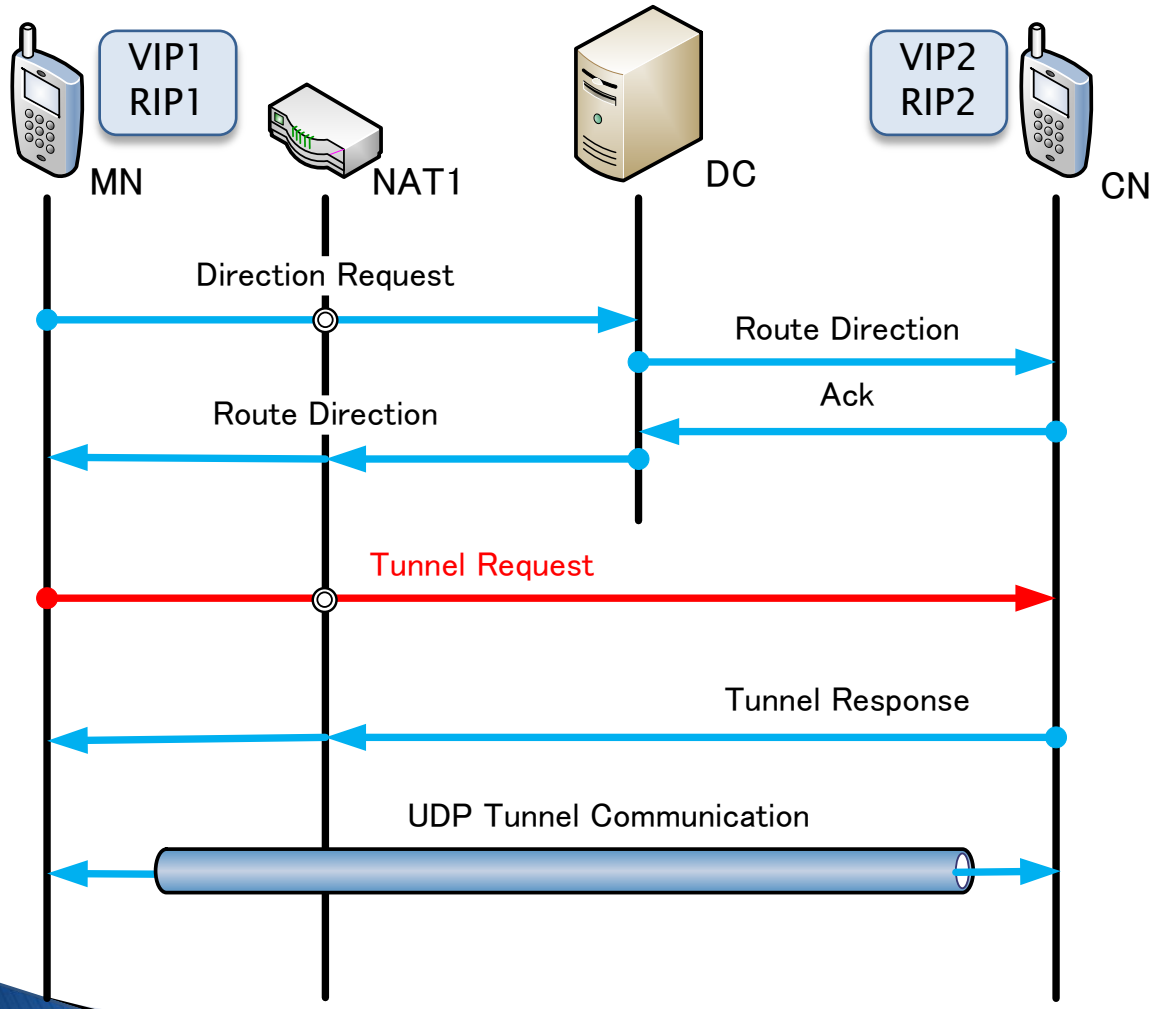
トンネル要求

トンネル応答

トンネル通信

RIP1 ⇔ RIP2
VIP1 ⇔ VIP2

シーケンス（通信開始時）



Path ID 生成

通信識別子

DCへ経路指示要求

CNへ経路指示

CNがPath ID を取得

MNへ経路指示

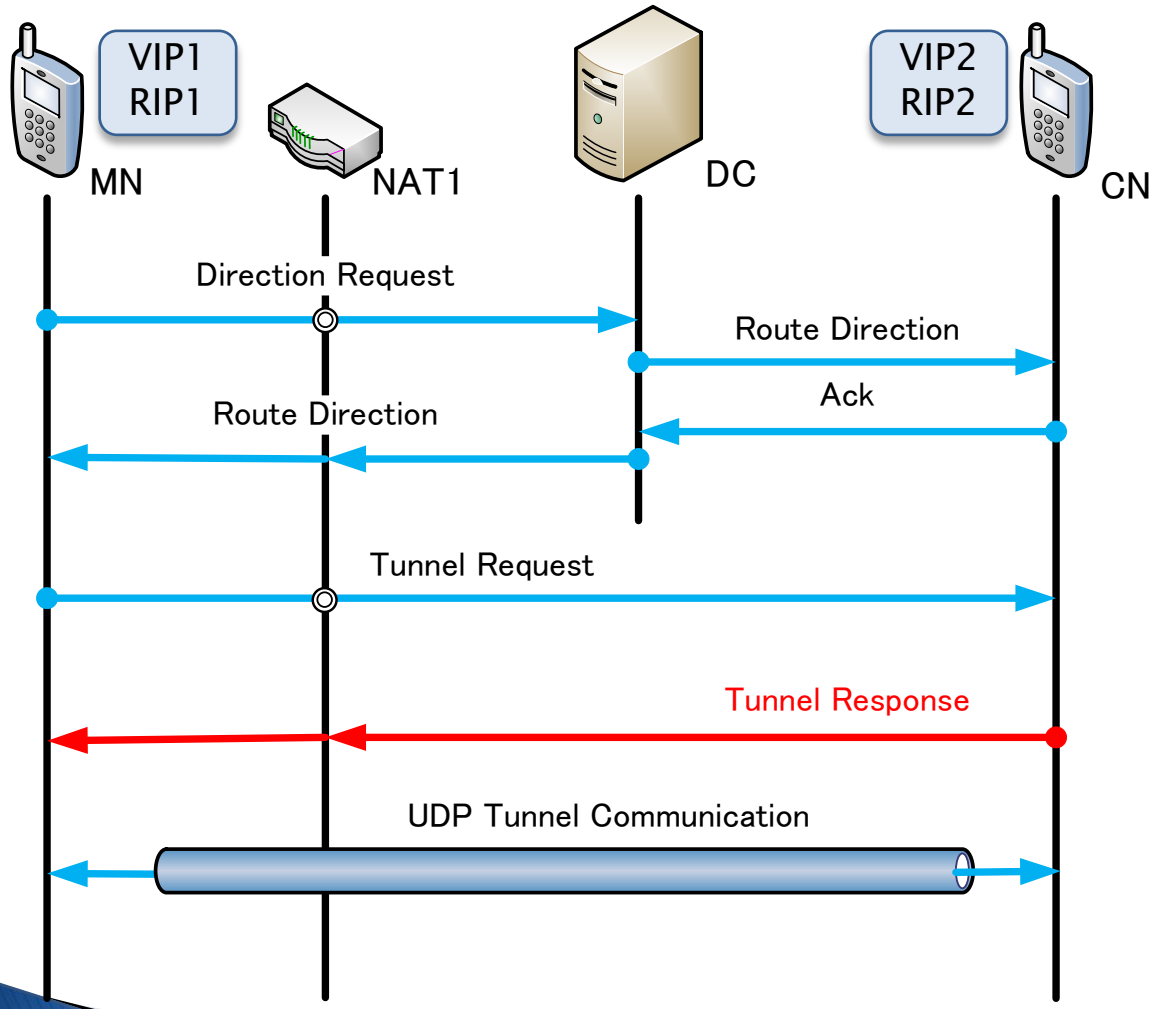
トンネル要求

トンネル応答

トンネル通信

RIP1 ⇔ RIP2
VIP1 ⇔ VIP2

シーケンス（通信開始時）



Path ID 生成

通信識別子

DCへ経路指示要求

CNへ経路指示

CNがPath ID を取得

MNへ経路指示

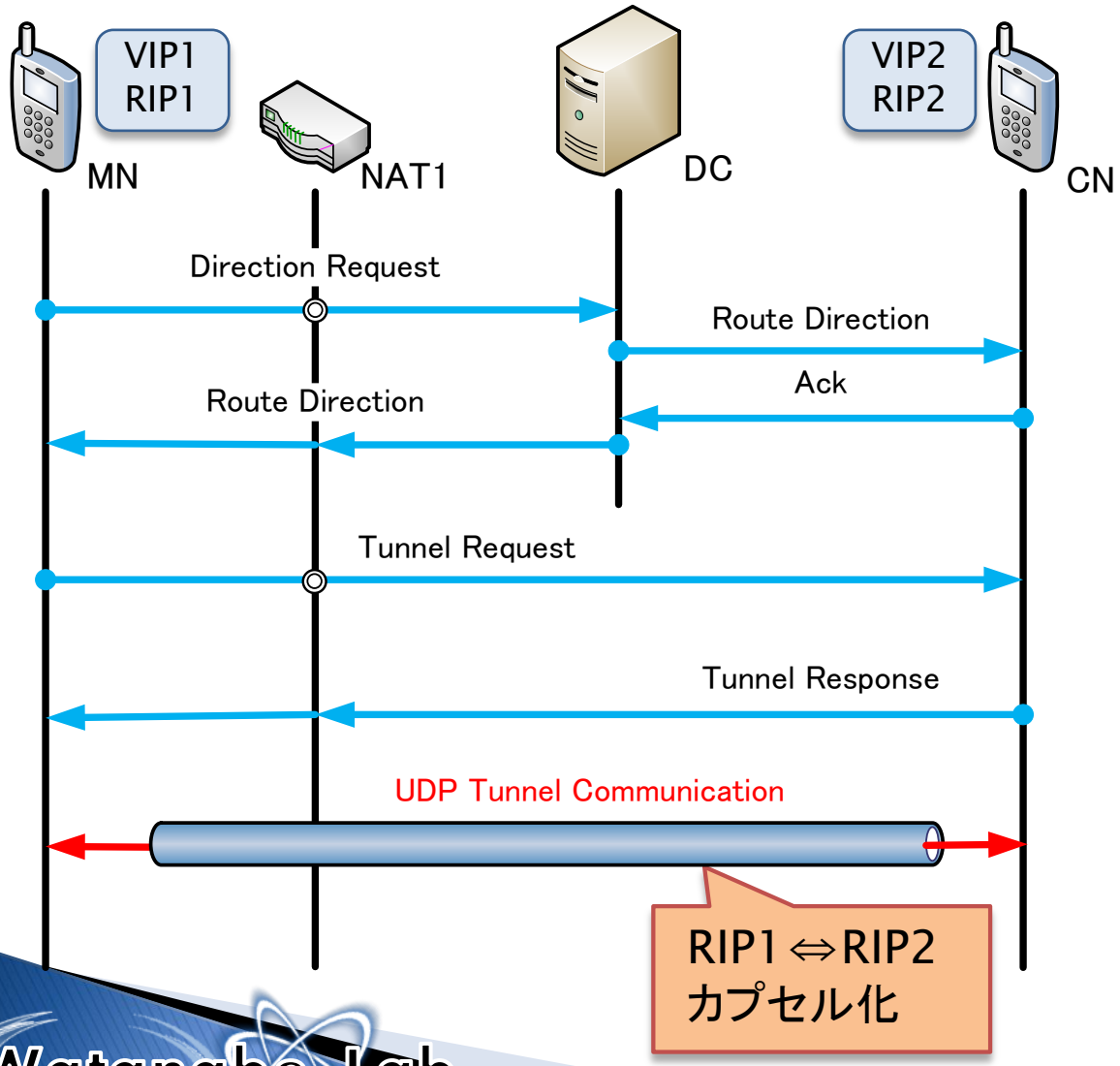
トンネル要求

トンネル応答

トンネル通信

RIP1 ⇔ RIP2
VIP1 ⇔ VIP2

シーケンス（通信開始時）



Path ID 生成

通信識別子

DCへ経路指示要求

CNへ経路指示

CNがPath ID を取得

MNへ経路指示

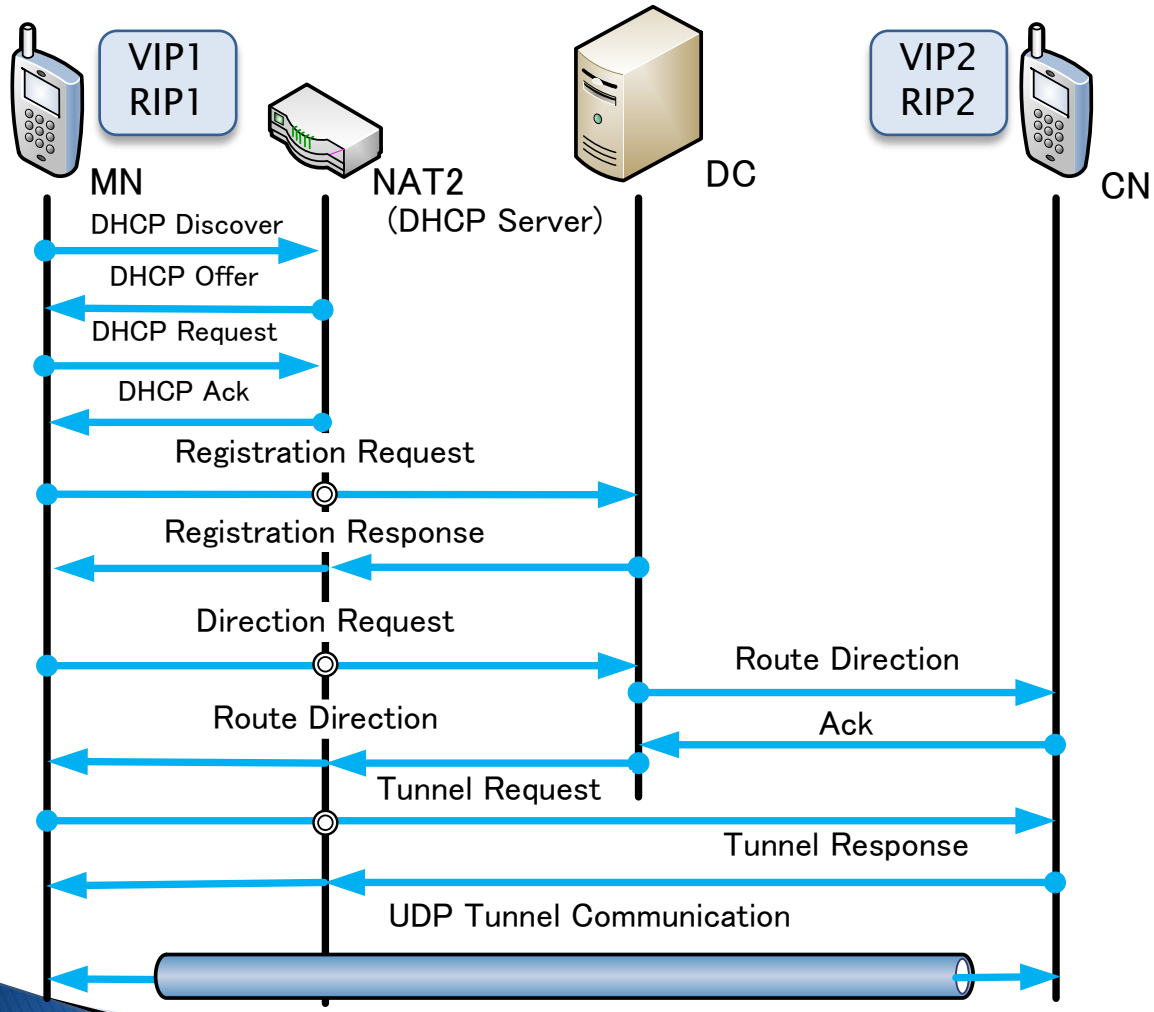
トンネル要求

トンネル応答

トンネル通信

RIP1⇔RIP2
VIP1⇔VIP2

シーケンス（移動時）



アドレス取得

RIP3取得

登録処理

DCへ変化後
IPアドレスを登録

同一Path ID使用

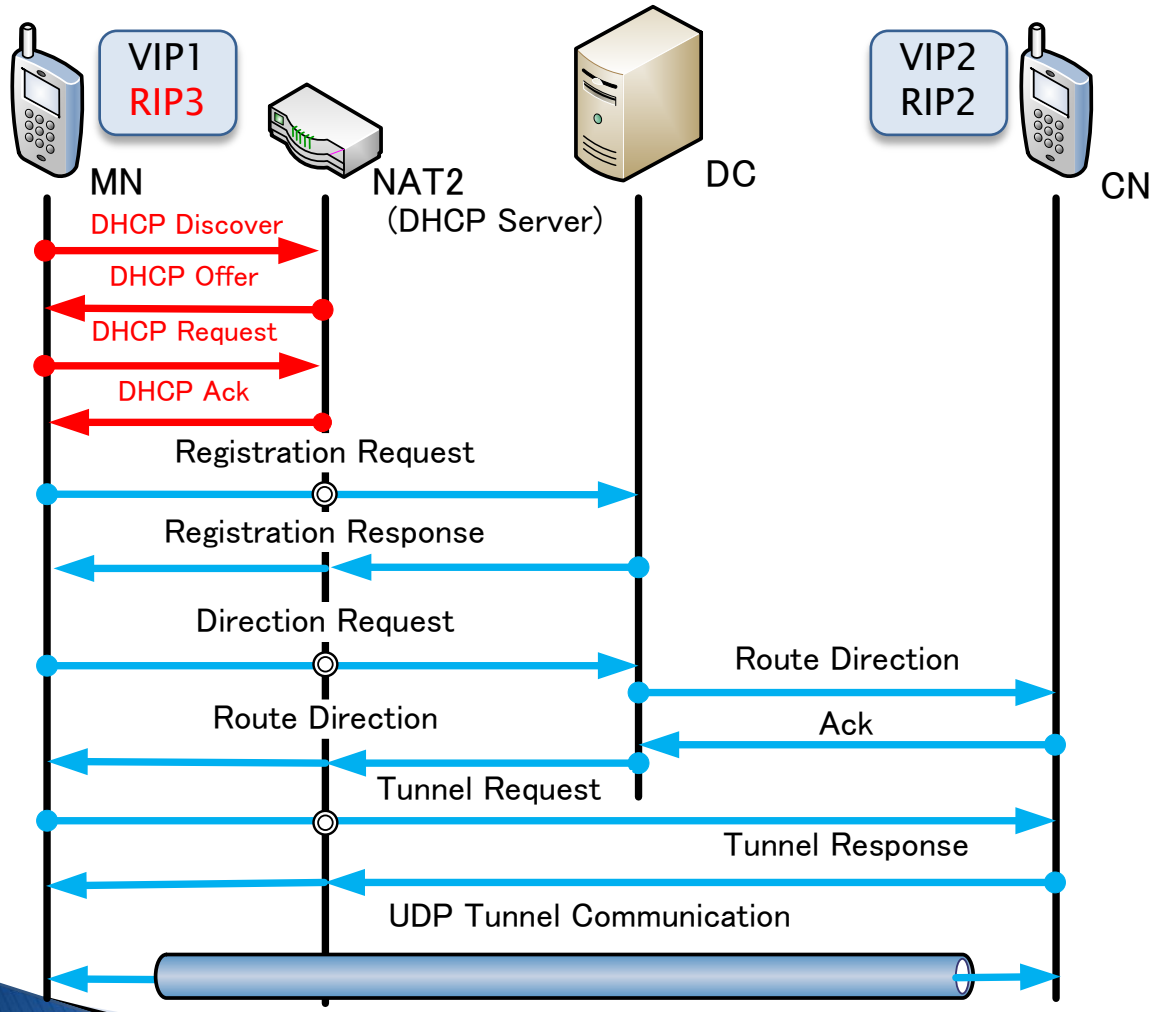
DCへ経路指示要求

CNへ経路指示

同一Path ID使用

MNへ経路指示

シーケンス（移動時）



アドレス取得

RIP3取得

登録処理

DCへ変化後
IPアドレスを登録

同一Path ID使用

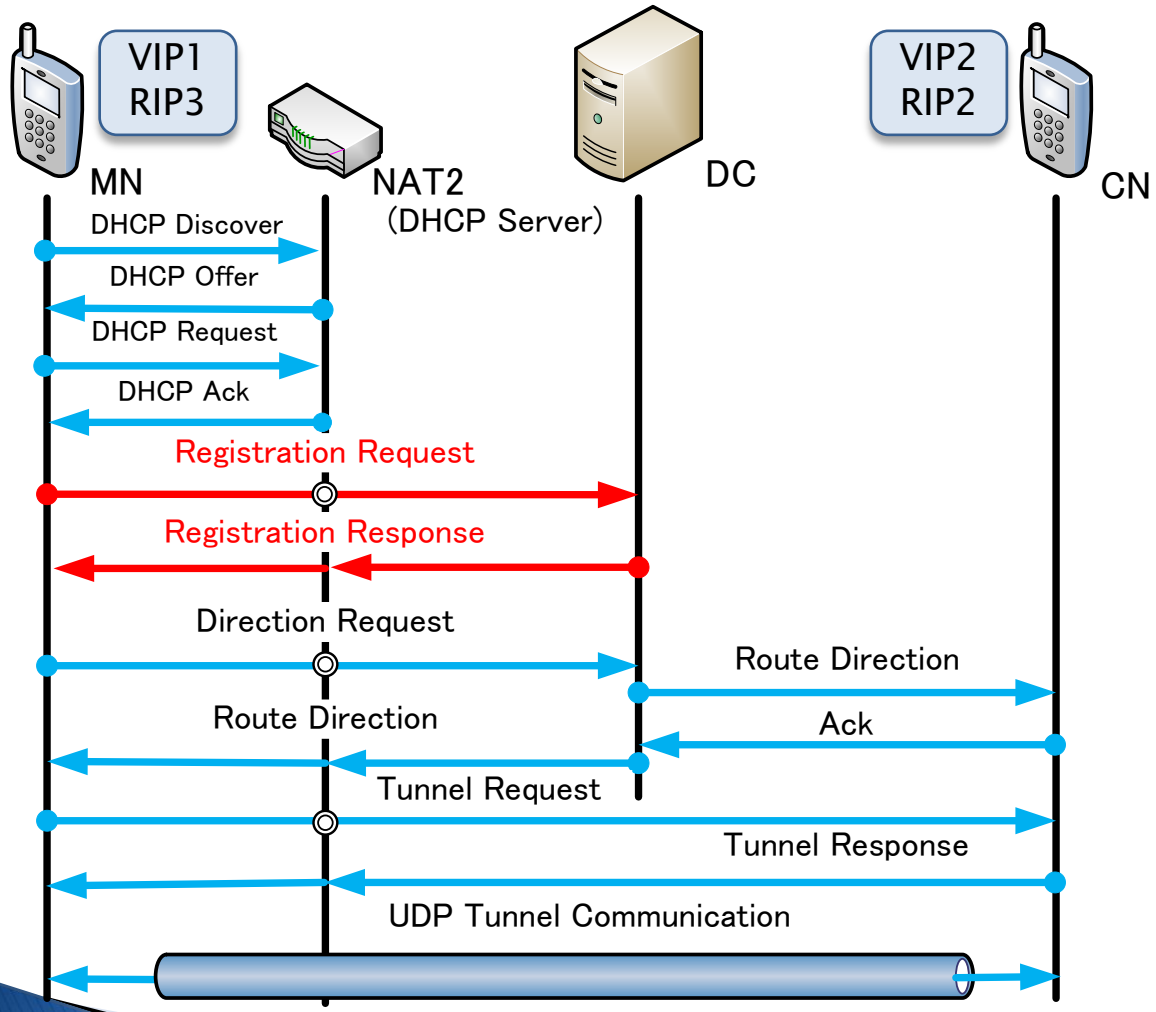
DCへ経路指示要求

CNへ経路指示

同一Path ID使用

MNへ経路指示

シーケンス（移動時）



アドレス取得

RIP3取得

登録処理

DCへ変化後
IPアドレスを登録

同一Path ID使用

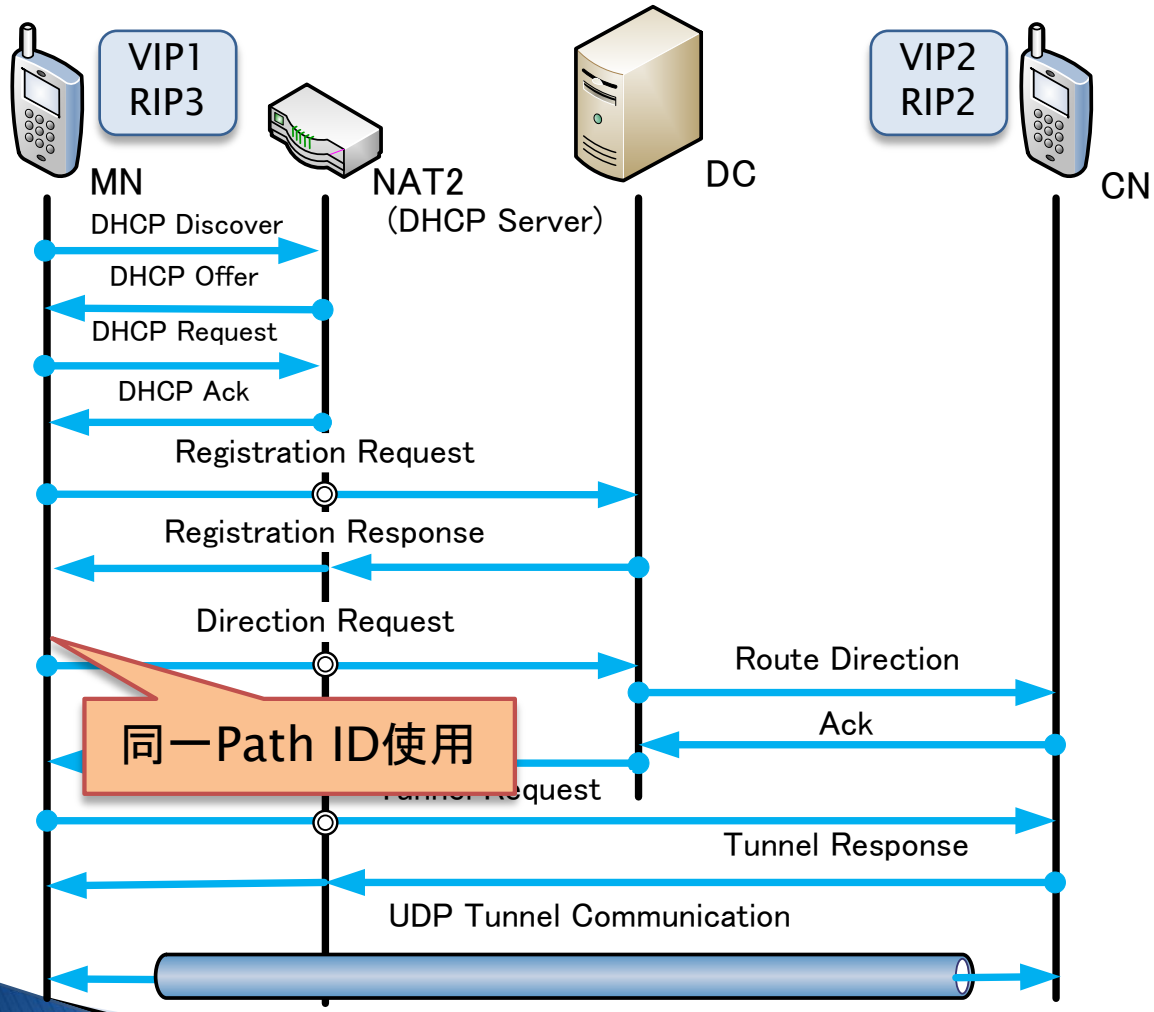
DCへ経路指示要求

CNへ経路指示

同一Path ID使用

MNへ経路指示

シーケンス（移動時）



アドレス取得

RIP3取得

登録処理

DCへ変化後
IPアドレスを登録

同一Path ID使用

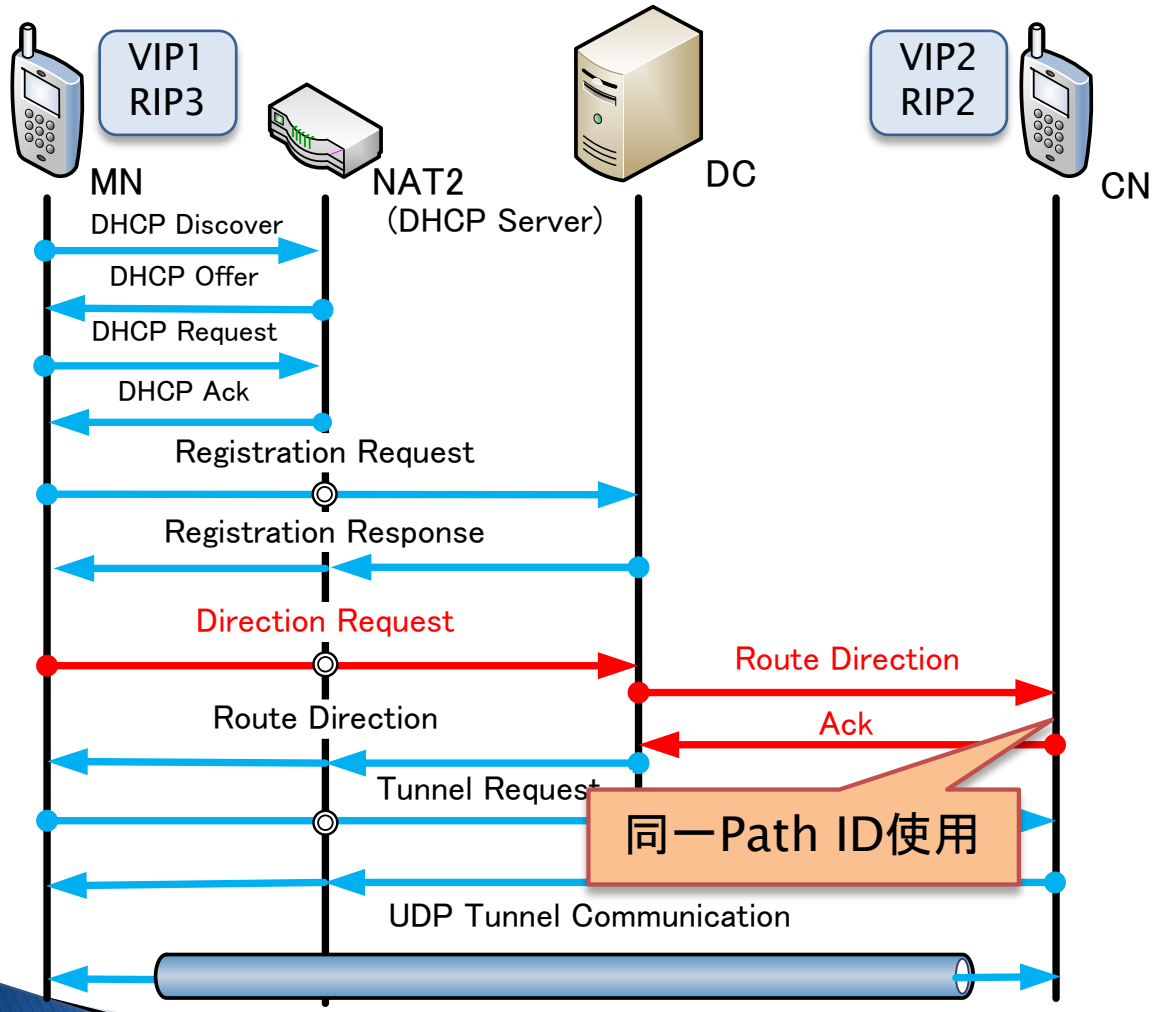
DCへ経路指示要求

CNへ経路指示

同一Path ID使用

MNへ経路指示

シーケンス（移動時）



アドレス取得

RIP3取得

登録処理

DCへ変化後
IPアドレスを登録

同一Path ID使用

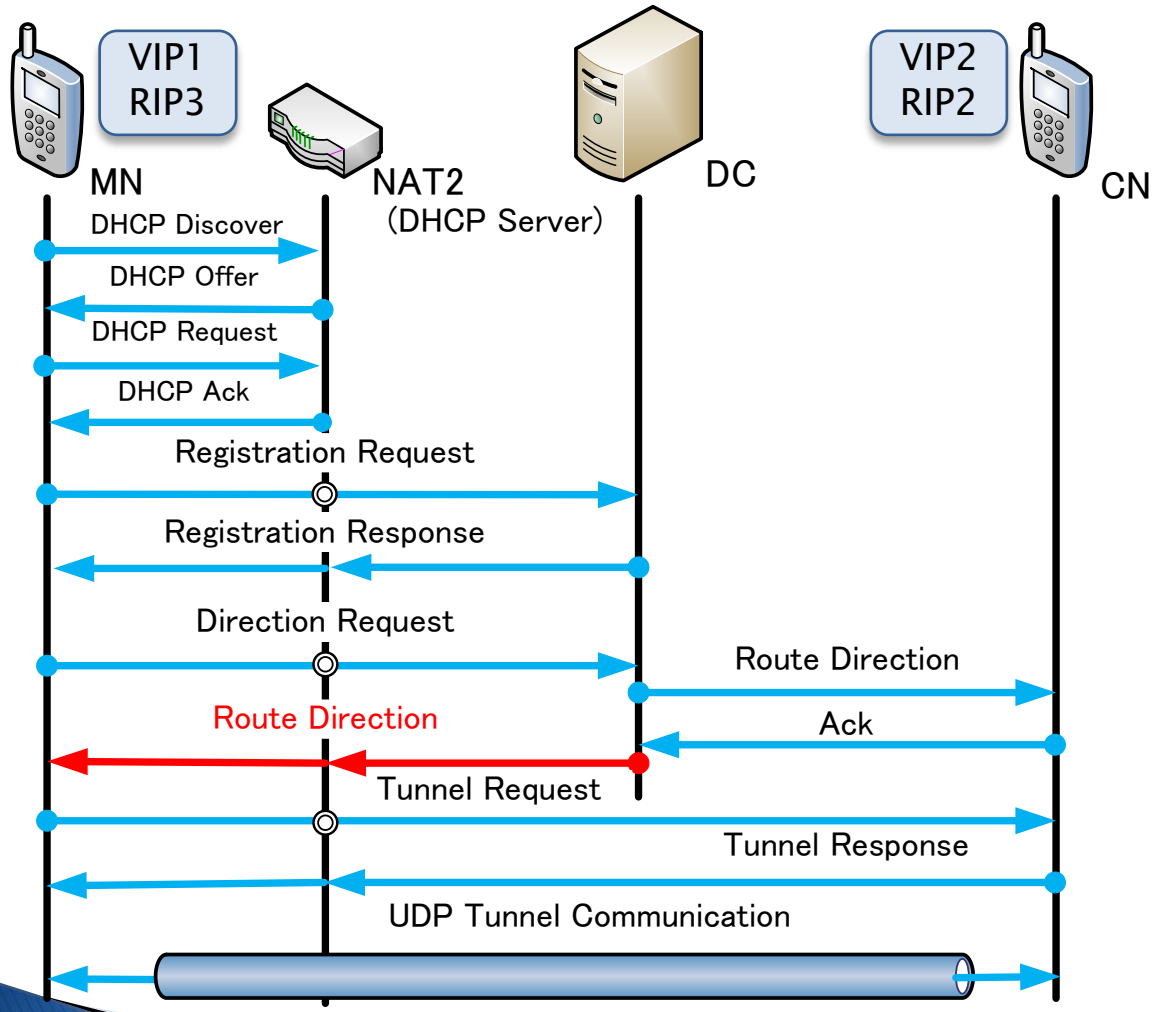
DCへ経路指示要求

CNへ経路指示

同一Path ID使用

MNへ経路指示

シーケンス（移動時）



アドレス取得

RIP3取得

登録処理

DCへ変化後
IPアドレスを登録

同一Path ID使用

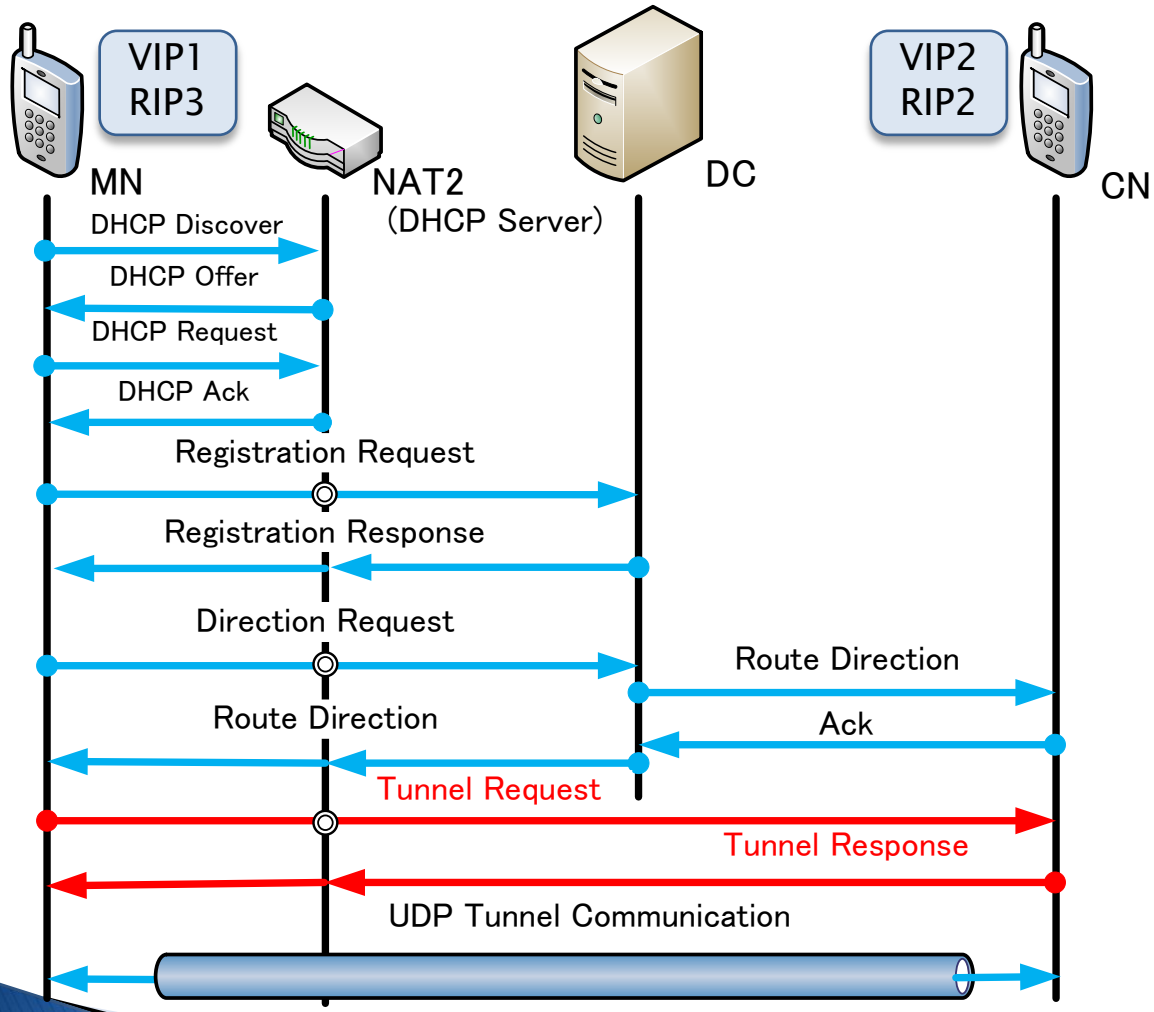
DCへ経路指示要求

CNへ経路指示

同一Path ID使用

MNへ経路指示

シーケンス（移動時）



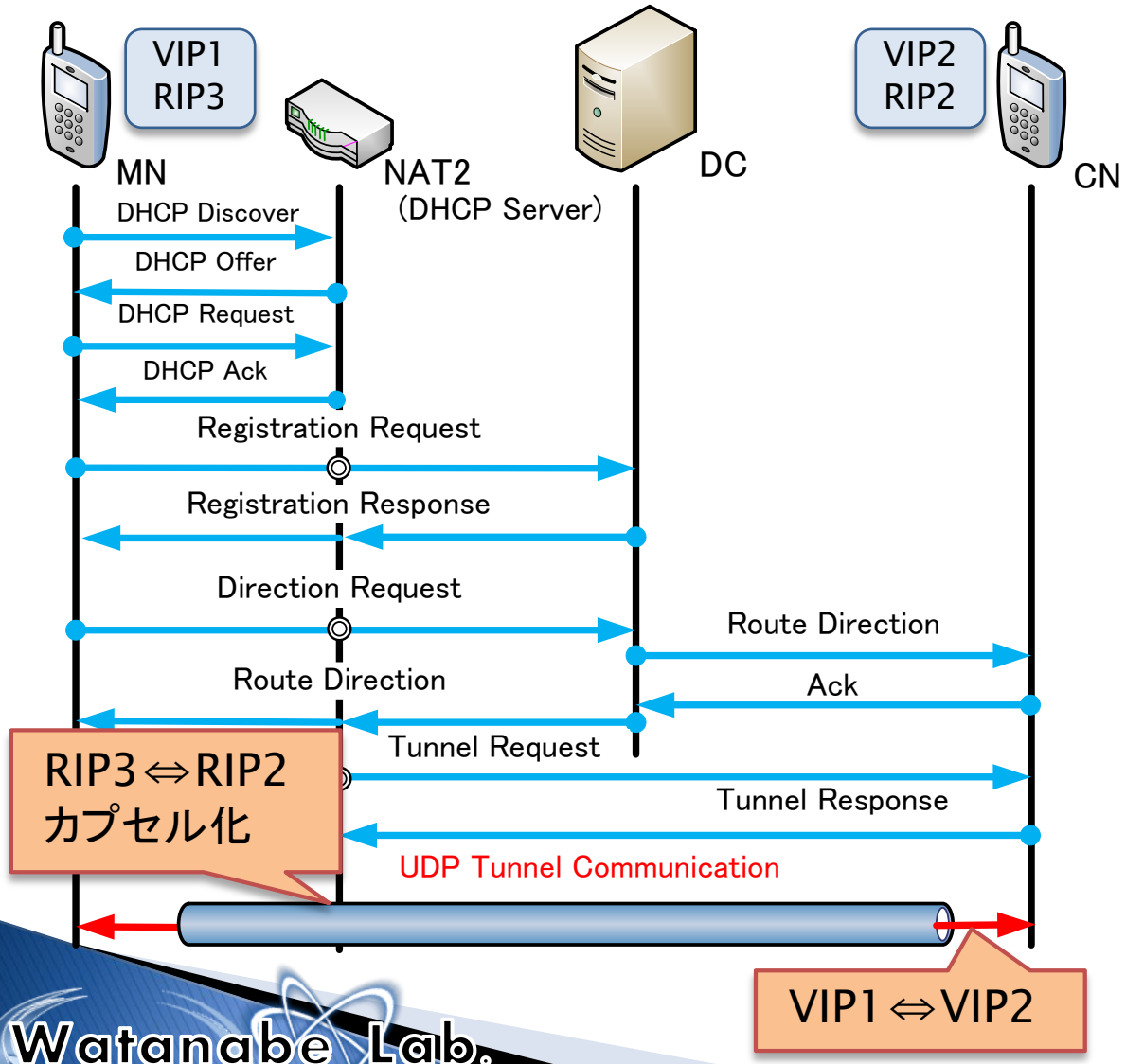
トンネル要求

トンネル応答

トンネル再構築

- RIP3⇔RIP2で実アドレストンネル再構築
- カプセル内パケット VIP1⇔VIP2
- アプリケーションは実IPアドレスの変化に気づかない

シーケンス（移動時）



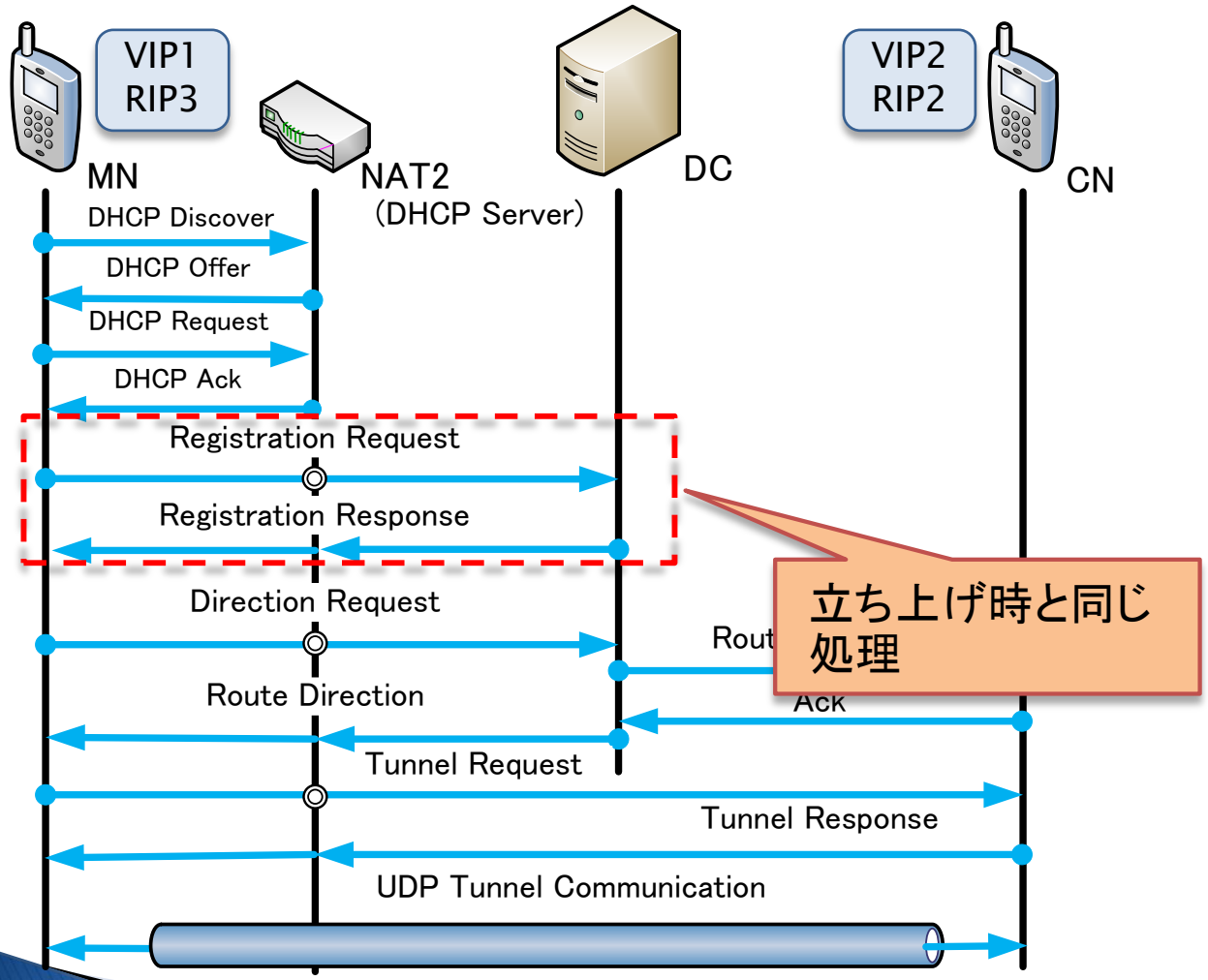
トンネル要求

トンネル応答

トンネル再構築

- RIP3⇔RIP2で実アドレストンネル再構築
- カプセル内パケット VIP1⇔VIP2
- アプリケーションは実IPアドレスの変化に気づかない

シーケンス（移動時）



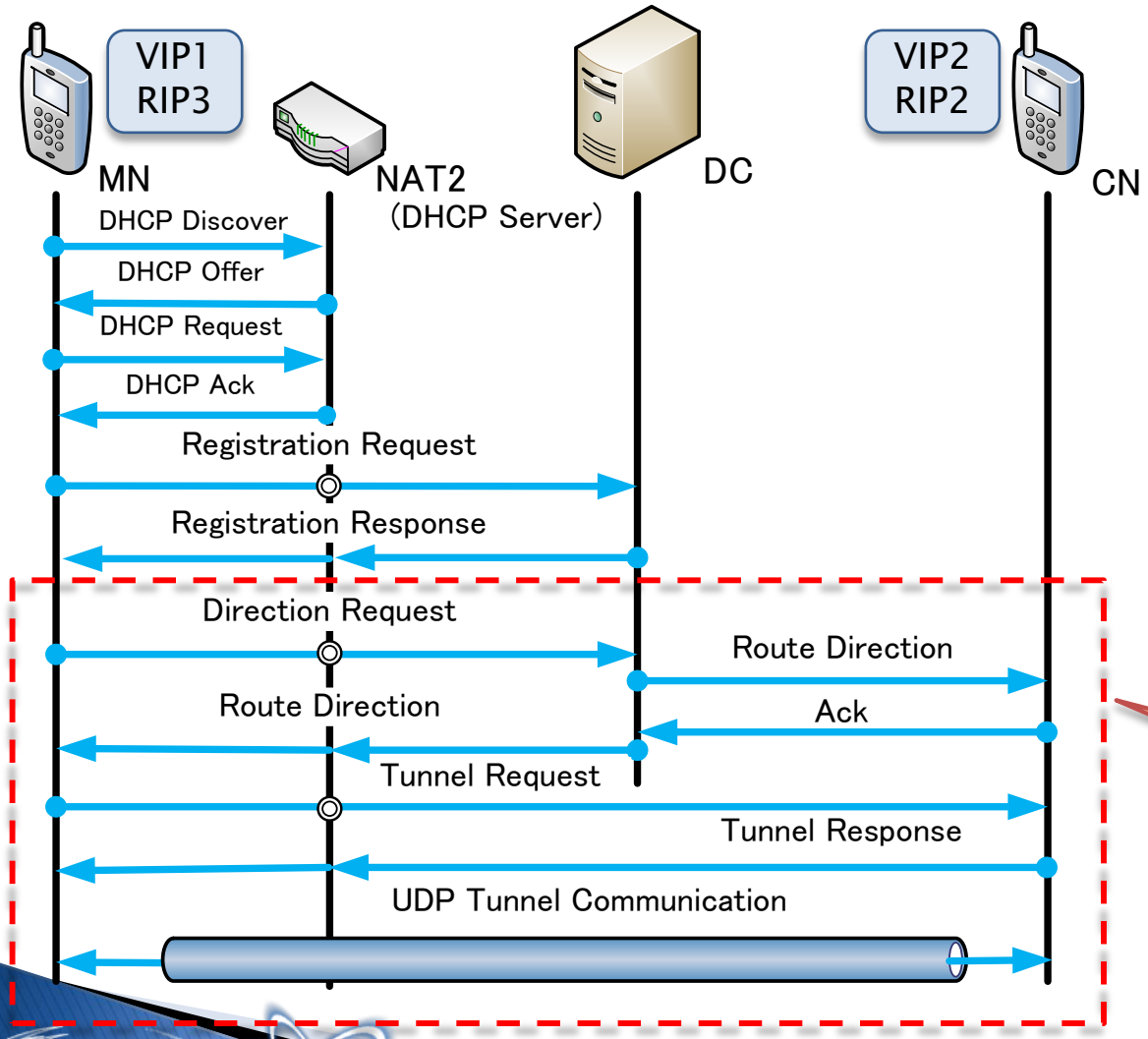
トンネル要求

トンネル応答

トンネル再構築

- RIP3⇔RIP2で実アドレストンネル再構築
- カプセル内パケット VIP1⇔VIP2
- アプリケーションは実IPアドレスの変化に気づかない

シーケンス（移動時）



トンネル要求

トンネル応答

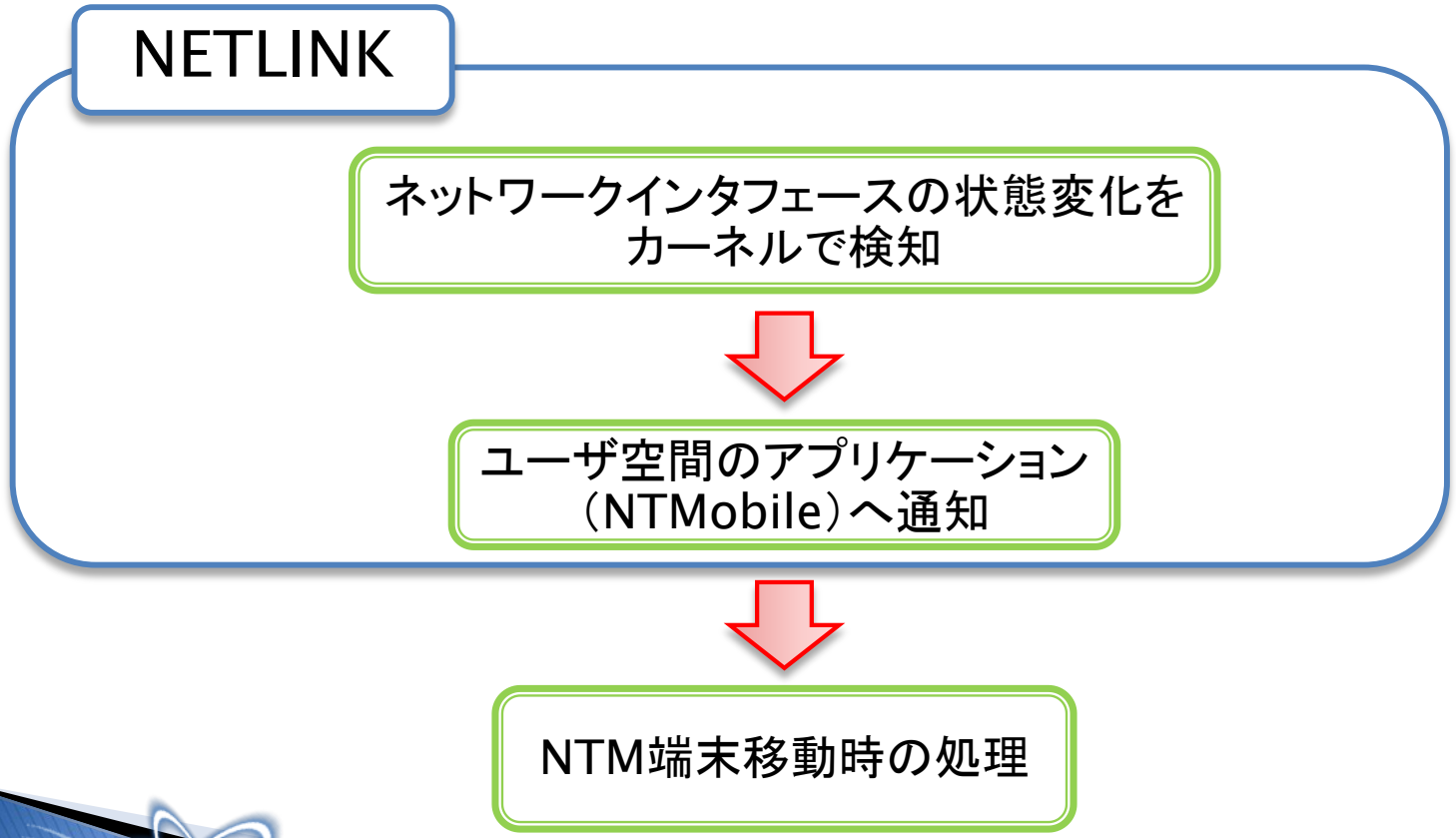
トンネル再構築

- RIP3⇔RIP2で実アドレストンネル再構築
- カプセル内パケット VIP1⇔VIP2
- アプリケーションは実IPアドレスの変化に気づかない

通信開始時と同じ処理

移動検出

- 通信端末の移動検出に NETLINK を用いる



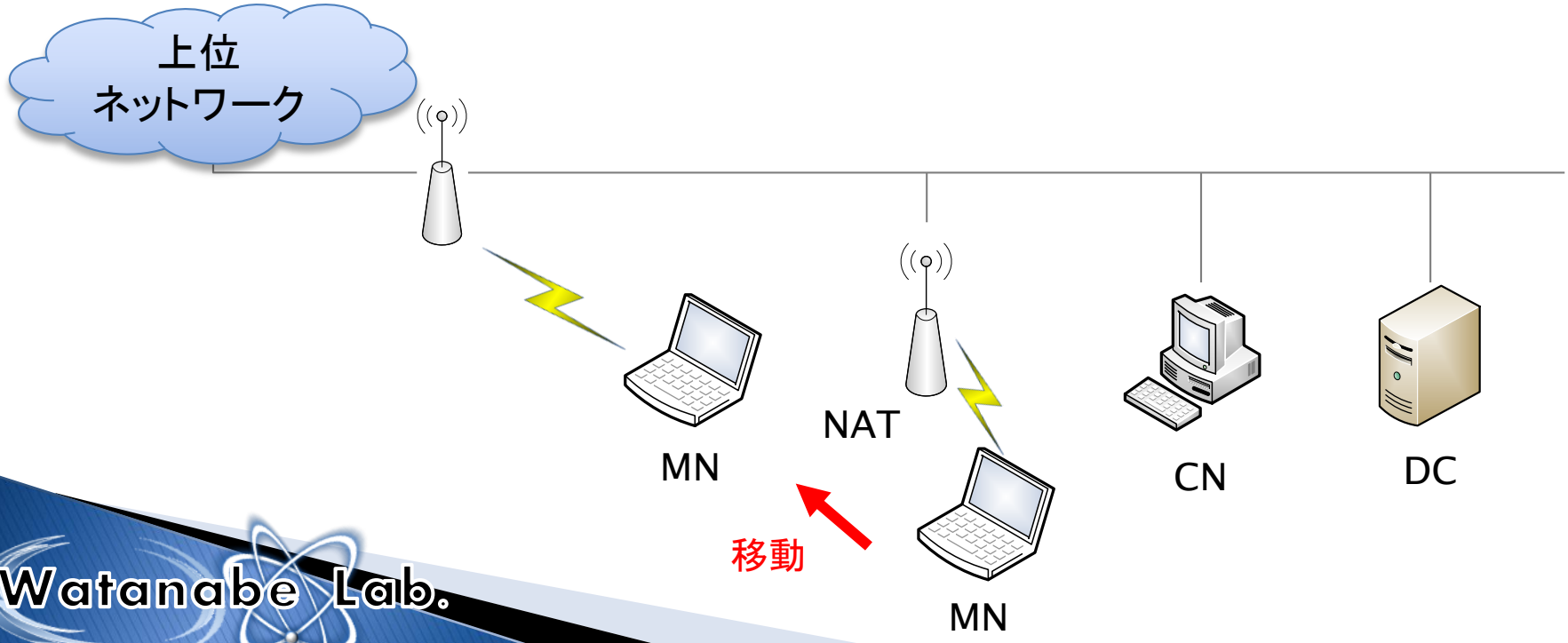
性能評価

- 装置の仕様

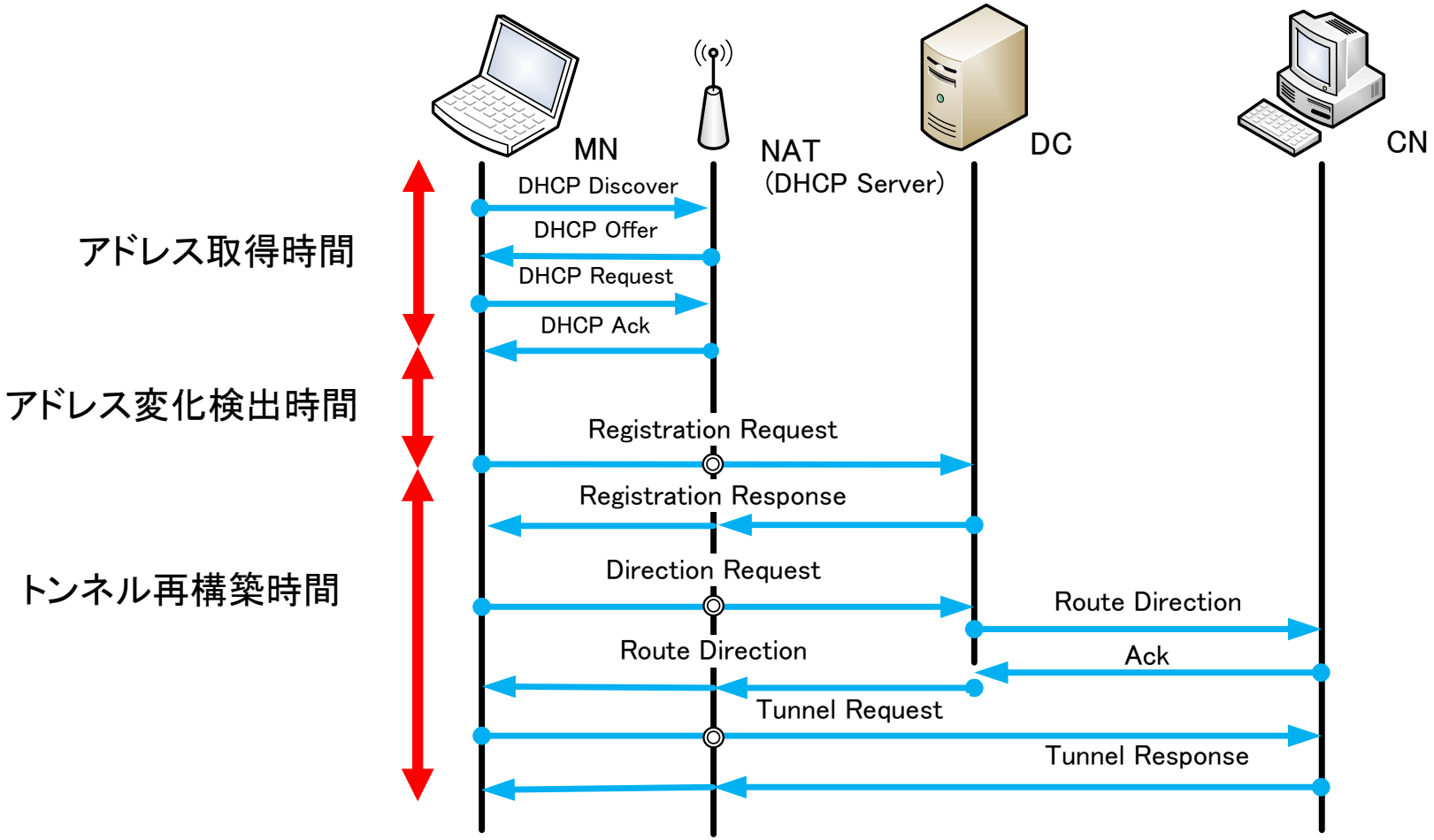
Device		OS	CPU	Memory
MN		Ubuntu 14.04	Intel Core i5-3320M 2.60GHz	4GB
CN		Ubuntu14.04	Intel Core i7-2600 3.40GHz	8GB
DC	ホストマシン	Windows 10 64bit	Intel Core i7-870 2.93GHz	20GB
	仮想マシン	Ubuntu14.04	1Core	1GB

性能評価

- NTM端末1をNAT配下からDCと同じネットワークへ移動
- トンネル再構築時間を計測
- 全体の処理時間を3つのフェーズに分割

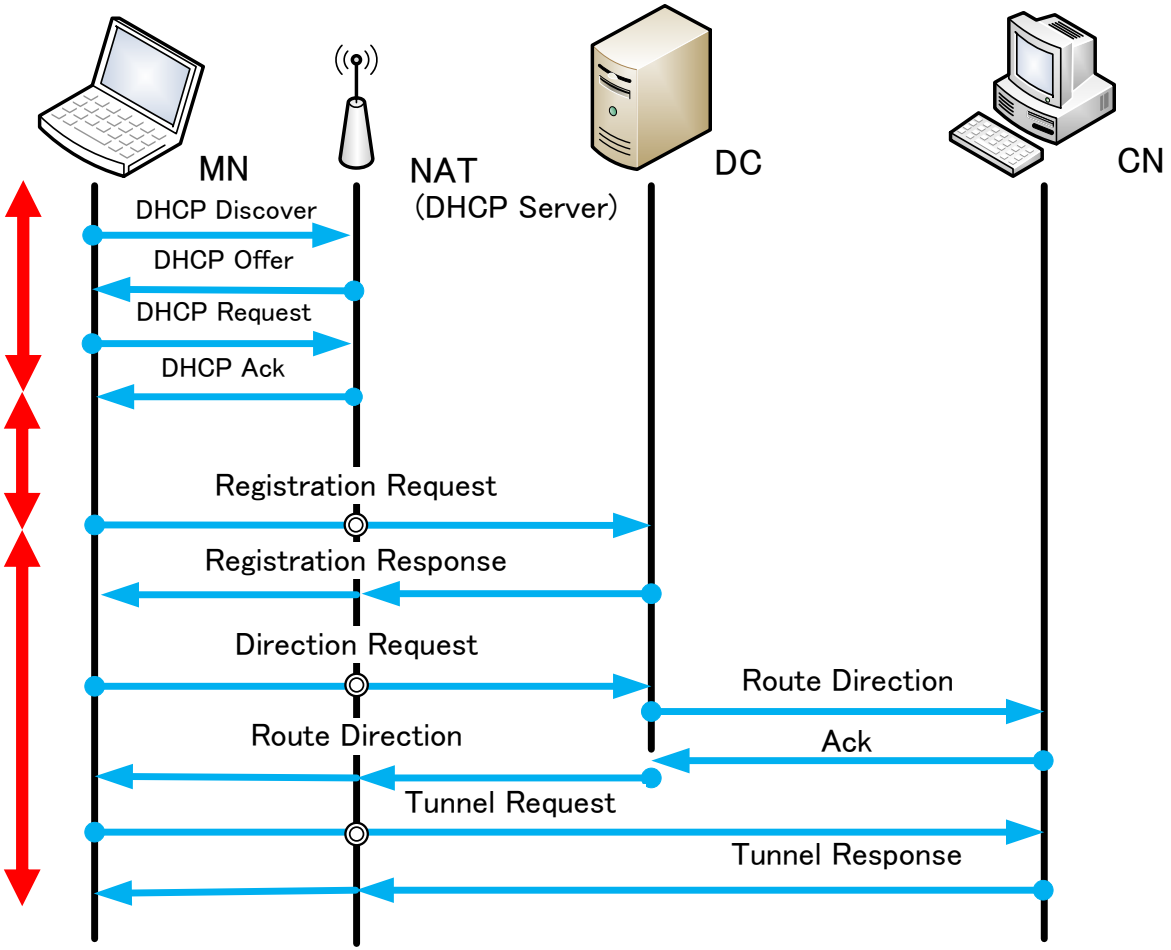


性能評価


MN
NAT
(DHCP Server)
DC
CN
DHCP Discover
DHCP Offer
DHCP Request
DHCP Ack
Registration Request
Registration Response
Direction Request
Route Direction
Route Direction
Ack
Tunnel Request
Tunnel Response
アドレス取得時間
アドレス変化検出時間
トンネル再構築時間

性能評価

区分	処理時間 (ms)
アドレス取得時間	3,889
アドレス変化検出時間	13
トンネル再構築時間	118



まとめ

- NTMobileにおける移動透過性の実現
  - UDPカプセル化通信により実アドレスの変化を隠蔽
  - アドレス変化を監視しトンネル経路切替
  - 通信開始時と同じシーケンス
  - 同一 PathID(通信識別子)を継続して使用

- 今後の予定
  - アドレス変化検出時間の短縮