

LAN内通信システムをインターネット上で利用可能にする TUNアプリの提案と実装

稲垣 智^{†1} 尾久 史弥^{†2} 鈴木 秀和^{†1} 内藤 克浩^{†3} 渡邊 晃^{†1}

^{†1} 名城大学理工学部 ^{†2} 名城大学大学院理工学研究科 ^{†3} 愛知工業大学情報科学部

1 はじめに

LAN内での通信を前提とすると柔軟なアプリケーション開発を行うことができる。しかしインターネット上での通信を考慮すると NAT 越え問題や移動透過性等の様々な問題を考慮する必要がある。これらの問題を解決し、インターネットをあたかも大きな LAN として扱うことができると有用である。DSMIPv6[1] や NTMobile (Network Traversal with Mobility) [2] はこのような目的のために開発された技術である。しかし DSMIPv6 はグローバルアドレスを大量に消費し、カーネルの改造が必要という課題がある。また、NTMobile は既存アプリケーションをそのまま利用できないという課題を抱えている。

そこで本稿では様々な OS に標準実装されている TUN/TAP インタフェースを用いて、LAN 内通信システムをインターネット上でそのまま利用可能にし、かつカーネルやアプリケーションの改造を必要としないシステムの実現方法を提案する。

2 DSMIPv6 と NTMobile

2.1 DSMIPv6

DSMIPv6 は不変的なアドレスとして HoA (Home Address) を端末に割り当て、移動先ネットワークで取得する CoA (Care of Address) によってカプセル化して通信を行う。上位アプリケーションは HoA を用いて通信するため、NAT 越えや移動に係る CoA の変化をアプリケーションから隠蔽することができる。しかし HA (Home Agent) をグローバル空間に設置する必要があることから、モバイル端末ごとにグローバルアドレスが必要になる。また、DSMIPv6 はカーネル空間に実装することが前提であることから、スマートフォンでの利用はできない。

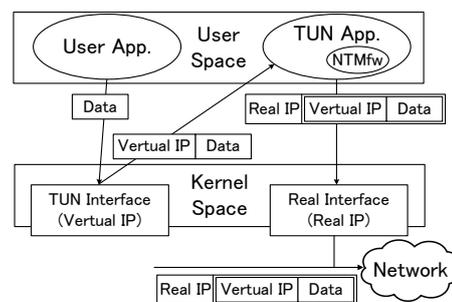


図1 提案方式におけるパケットカプセル化の様子

2.2 NTMobile

NTMobile は NAT 越え、IPv4/IPv6 間通信、移動透過性を同時に実現する技術であり、本提案のベースとなる技術である。NTMobile は端末の不変的なアドレスとして仮想アドレスを用いる。仮想アドレスは端末の立ち上げ時に DC (Direction Coordinator) により割り当てられる。NTMobile は仮想アドレスにより生成された通信パケットをすべて実アドレスでカプセル化するという特徴がある。また DC がエンド端末に最適な通信経路を指示し、どのような通信環境においても双方向の通信接続性を保証する。

NTMobile はこれらの機能を NTMobile framework ライブラリ (NTMfw) [3] と呼ばれるアプリケーションライブラリとして提供している。アプリケーションに NTMfw を組み込むことにより NTMobile の機能を利用できる。しかし、一般通信とソケットインタフェースが異なるため、既存のアプリケーションをそのまま使用することができないという課題が存在している。

3 提案方式

提案方式では、NTMfw の機能を TUN アプリケーションとして実現し、既存のアプリケーションをそのまま使えるようにする。TUN/TAP インタフェースは、トンネル通信を実現するためのもので、一般のアプリケーション (ユーザアプリ) により生成されたパケットをネットワークに送信する直前にフックし、ユーザ空間のアプリケーションへ渡す仕組みである。TUN アプリではフックしたパケットに対し、NTMfw を用いて NTMobile 用パケットを生成することで、ユーザアプリに一切手を加えることなく NTMobile 機能を実現する。図1に提案方

Proposal and Implementation of TUN Application that Makes LAN Communication System Available on the Internet

Satoru Inagaki^{†1}, Fumiya Ogyu^{†2}, Hidekazu Suzuki^{†1}, Katsuhiro Naito^{†3} and Akira Watanabe^{†1}

^{†1} Faculty of Science and Technology, Meijo University

^{†2} Graduate School of Science and Technology, Meijo University

^{†3} Faculty of Information Science, Aichi Institute of Technology

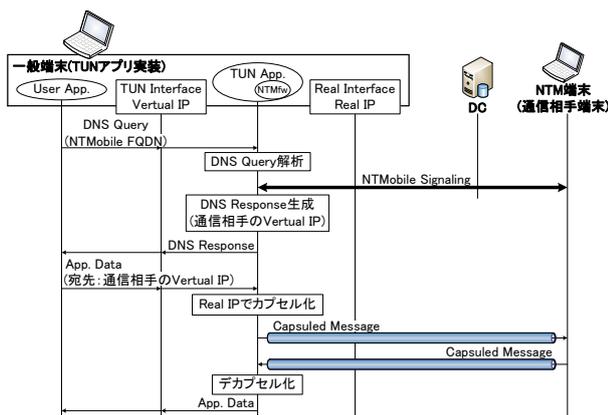


図2 提案方式における通信確立時の動作シーケンス

式におけるカプセル化の様子を示す。カプセル化処理はユーザ空間のTUNアプリとカーネルで分担して行うことができ、カーネルの改造も不要である。

ユーザアプリ起動からNTMobile通信実現までの流れは以下の通りである。まず、TUNアプリがNTMobileの登録処理を実行し、DCから仮想IPアドレスを取得する。このときTUNアプリはTUNインタフェースを作成し、ここに取得した仮想IPアドレスを割り当てる。また、DNSクエリがTUNインタフェースへ渡されるようルーティングテーブルの設定を変更する。これによりDNSクエリ、および仮想アドレス宛のパケットは全てTUNインタフェースを通じてTUNアプリへ渡されることになる。

図2に通信確立時の動作シーケンスを示す。ユーザアプリが通信相手のFQDNを指定することにより、DNSクエリが送信される。DNSクエリはTUNインタフェースを通じてTUNアプリへ渡される。DNSクエリを受信したTUNアプリは相手FQDNの解析を行う。NTMobile固有のFQDNが指定されていた場合は、NTMfwによりNTMobileシグナリング処理を実行してトンネル経路を生成するとともに、通信相手の仮想IPアドレスを取得する。取得した通信相手の仮想IPアドレスをDNS応答に記載し、TUNインタフェースを通じてユーザアプリに返信する。その後ユーザアプリは通信相手の仮想IPアドレス宛にデータを送信する。これらのパケットはすべてTUNインタフェースを通じてTUNアプリが受け取る。受け取ったパケットにNTMfwがヘッダ付与や暗号化等の処理を行い、実インタフェースを通じてカプセル化した後、通信相手に送信する。通信相手から受信したパケットは上記と逆の手順により、TUNインタフェースを通じてユーザアプリへ渡される。

DNSクエリの宛先名が一般の通信端末であった場合は、クエリをそのままローソケット経由で物理ネットワークに中継する。本方式によると、複数のNTMobile

表1 既存技術と提案方式の比較

	項目(1)	項目(2)
DSMIPv6	×	○
NTMfw	○	×
提案方式	○	○

対応のアプリケーションと、複数の一般通信用アプリケーションが同時に通信を行える。

4 実装・評価

4.1 実装・動作検証

TUNアプリをLINUX上で実装し動作検証を行った。検証方法は2台の提案方式による端末をVMにて準備し、NATを経由した通信を実行した。この状態で双方の端末上でLAN内通信システム対応のアプリケーションを動作させると、NATが混在する環境でも双方向の通信接続性を確立できることを確認した。また一般アプリとTUNアプリを利用したアプリが複数同時通信できることを確認した。

4.2 評価

表1に既存技術と提案方式の比較を示す。評価項目は以下の通りとした。

- (1) カーネルの改造を必要としないか
- (2) 既存アプリを使用できるか

DSMIPv6ではカーネルの改造が必要であるが既存アプリケーションをそのまま利用できる。NTMfwはカーネルの改造が不要であるが、既存のアプリケーションがそのまま使用できないという課題がある。提案方式ではカーネルの改造を必要とすることなく、さらに既存のアプリケーションもそのまま利用することができる。

5 まとめ

本稿では、TUN/TAPインタフェースを用いてLAN内通信システムをインターネット上で利用可能にするTUNアプリの提案及び実装を行った。提案方式によりカーネルの改造やアプリケーションの改造を必要とすることなくLAN内で実現した通信システムをインターネット上でそのまま利用することが可能になった。今後は提案方式の性能評価を行う予定である。

参考文献

- [1] Soliman, H.: Mobile IPv6 Support for Dual Stack Hosts and Routers, RFC 5555, IETF (2009).
- [2] 上酔尾一真ほか: IPv4/IPv6混在環境で移動透過性を実現するNTMobileの実装と評価, 情報学論, Vol.54, No.10, pp.2288-2299 (2013).
- [3] 納堂博史ほか: 実用化に向けたNTMobile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).

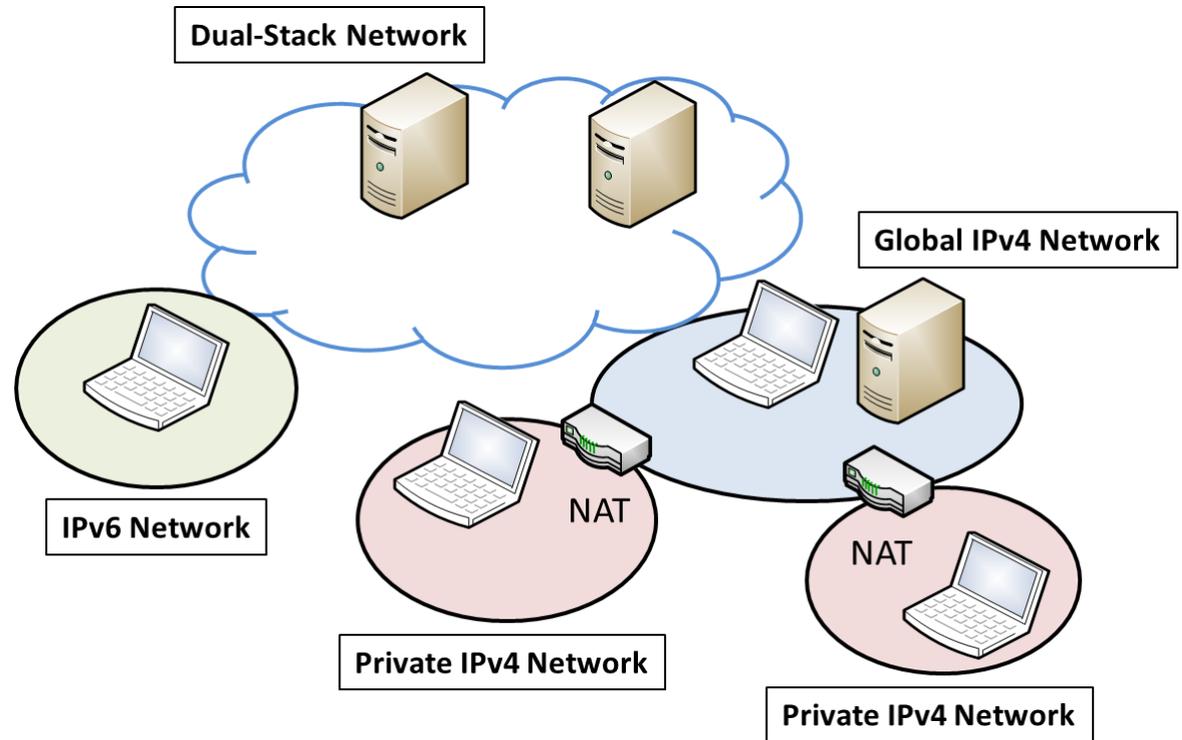
LAN内通信システムをインターネット上で 利用可能にするTUNアプリの提案と実装

稲垣 智† 尾久 史弥† 鈴木 秀和† 内藤 克浩‡ 渡邊 晃†
†名城大学
‡愛知工業大学



研究背景

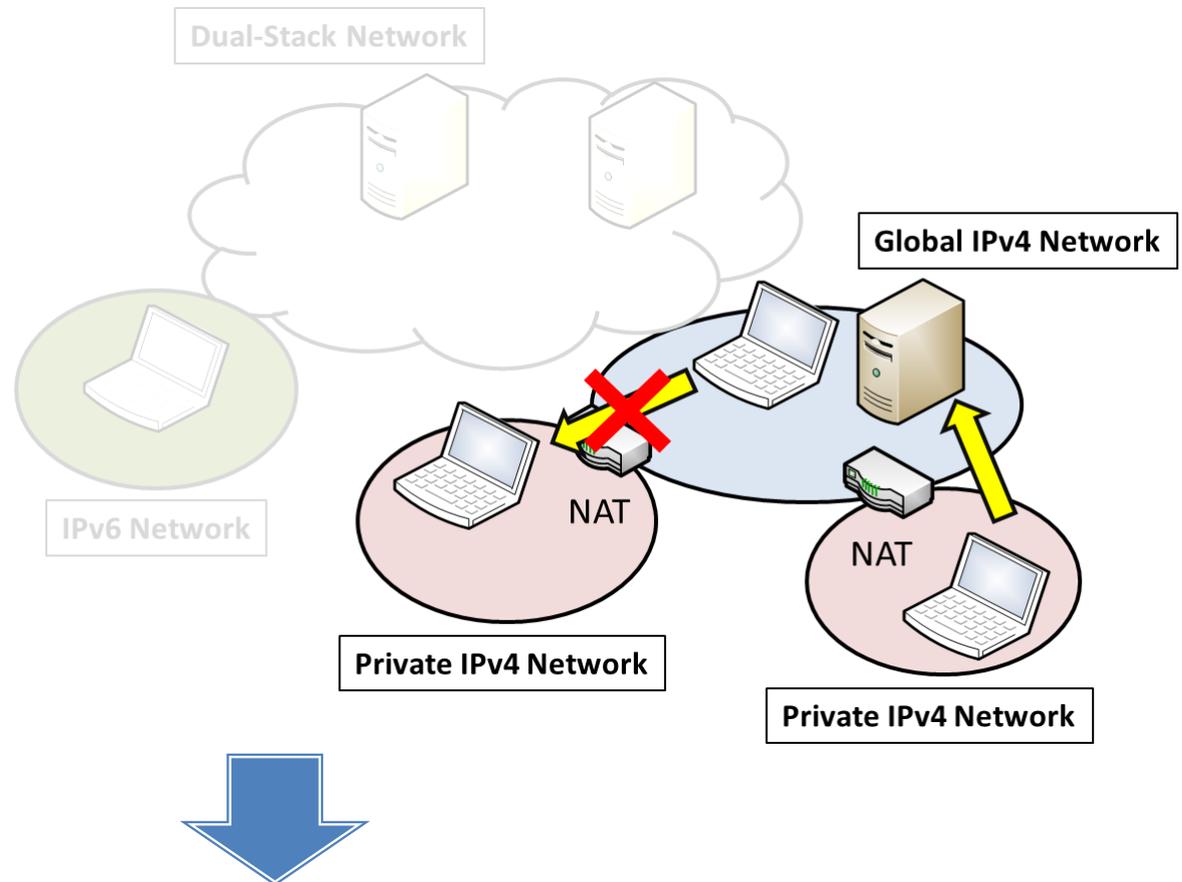
- ▶ NAT越え問題
- ▶ IPv4-IPv6間での通信不可問題
- ▶ 移動透過性の課題



上記の問題を解決してネットワークをフラット化し、インターネットをあたかも**大きなLAN**として扱えると有用

研究背景

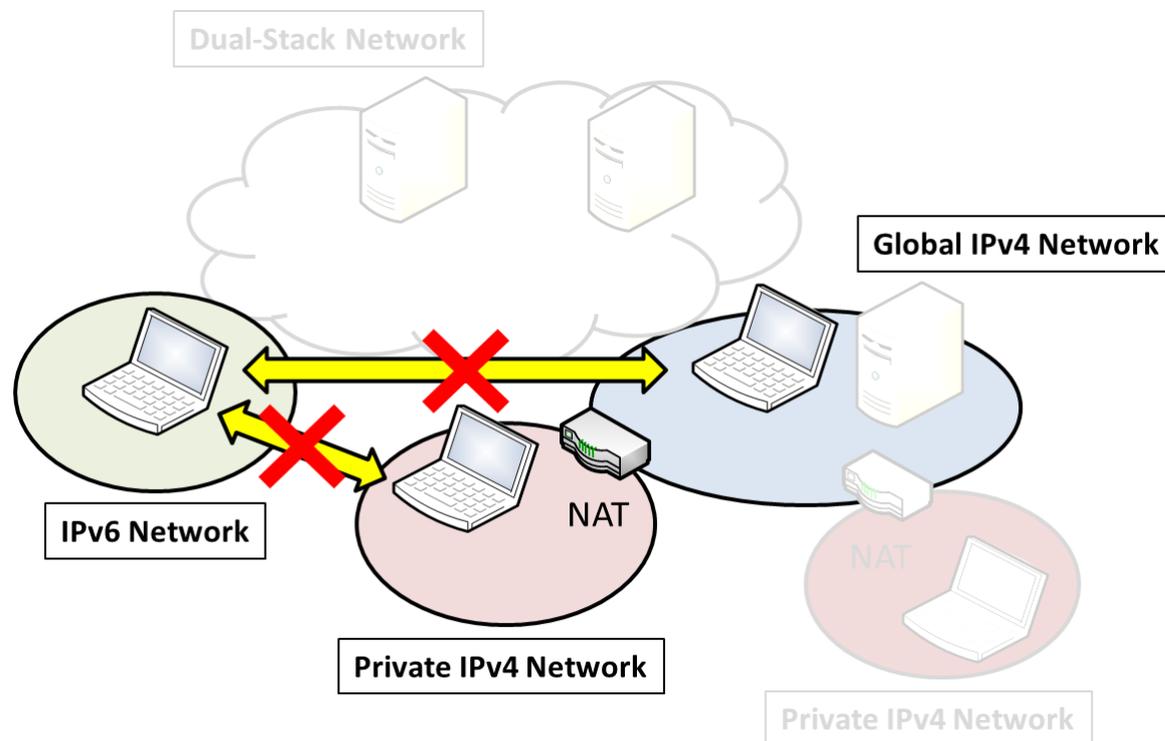
- ▶ NAT越え問題
- ▶ IPv4-IPv6間での通信不可問題
- ▶ 移動透過性の課題



上記の問題を解決してネットワークをフラット化し、インターネットをあたかも**大きなLAN**として扱えると有用

研究背景

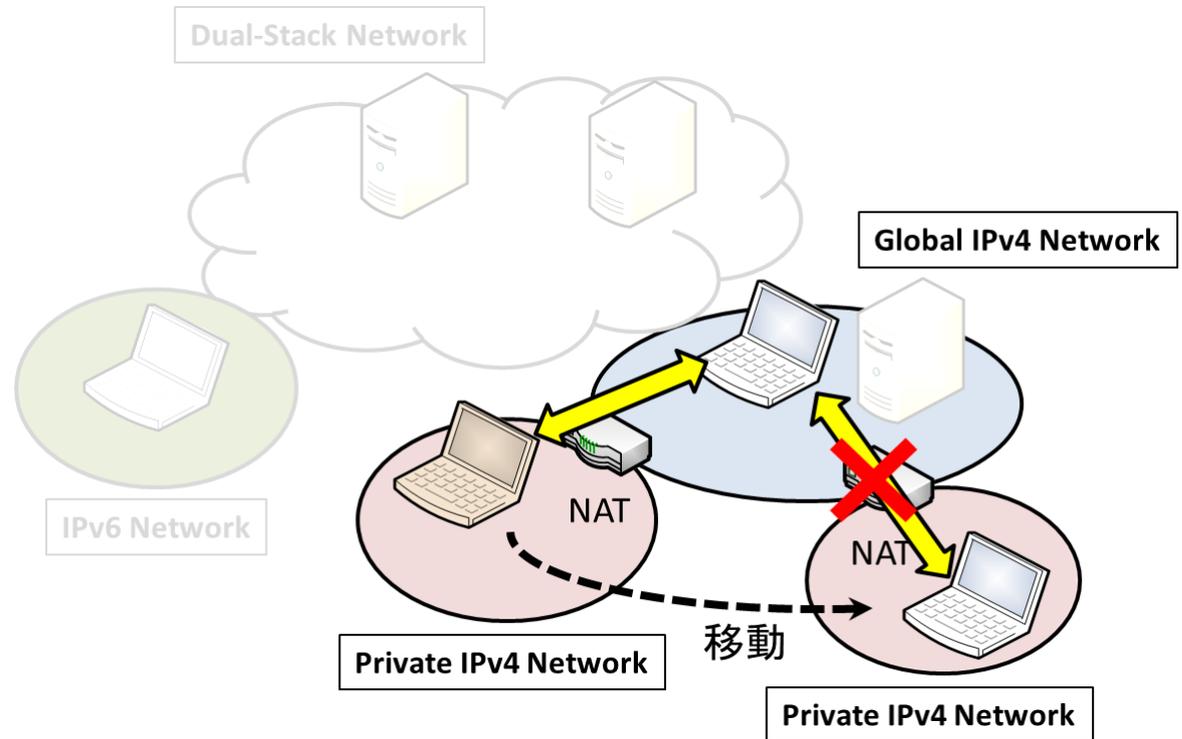
- ▶ NAT越え問題
- ▶ IPv4-IPv6間での通信不可問題
- ▶ 移動透過性の課題



上記の問題を解決してネットワークをフラット化し、インターネットをあたかも**大きなLAN**として扱えると有用

研究背景

- ▶ NAT越え問題
- ▶ IPv4-IPv6間での通信不可問題
- ▶ 移動透過性の課題



上記の問題を解決してネットワークをフラット化し、インターネットをあたかも**大きなLAN**として扱えると有用

既存技術

- ▶ DSMIPv6 (Dual Stack Mobile IP version 6)*1

- ▶ NTMobile (Network Traversal with Mobility)*2

*1 RFC5555, 2009

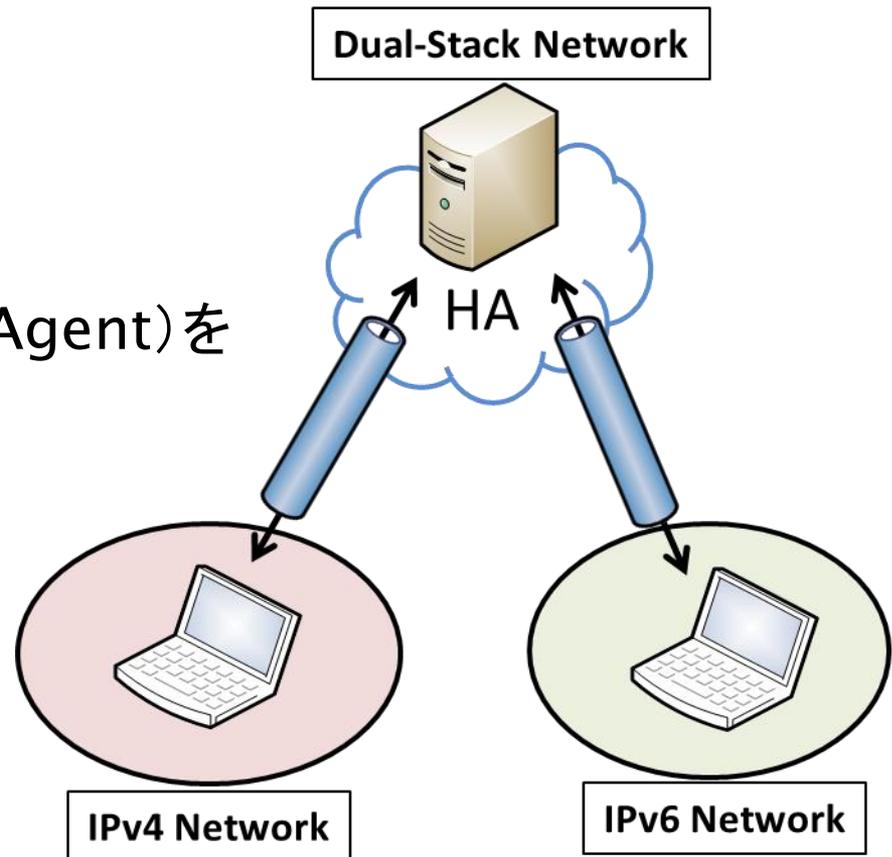
*2 納堂ほか: 信学技報, Vol.116, No.10, pp.2288-2299, 2017

DSMIPv6

▶ 様々な課題

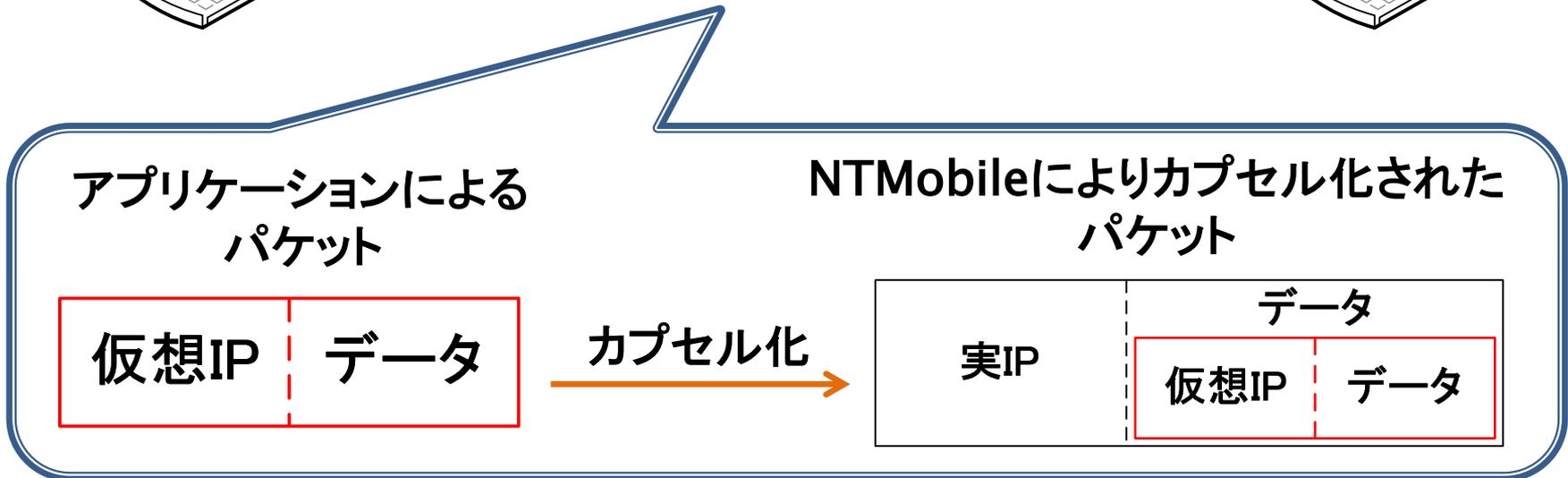
- 移動端末毎に IPv4 グローバルアドレスが必要
- IPv4 環境では必ず HA (Home Agent) を経由した冗長経路となる

**カーネル空間への実装が必要
普及が進まない**



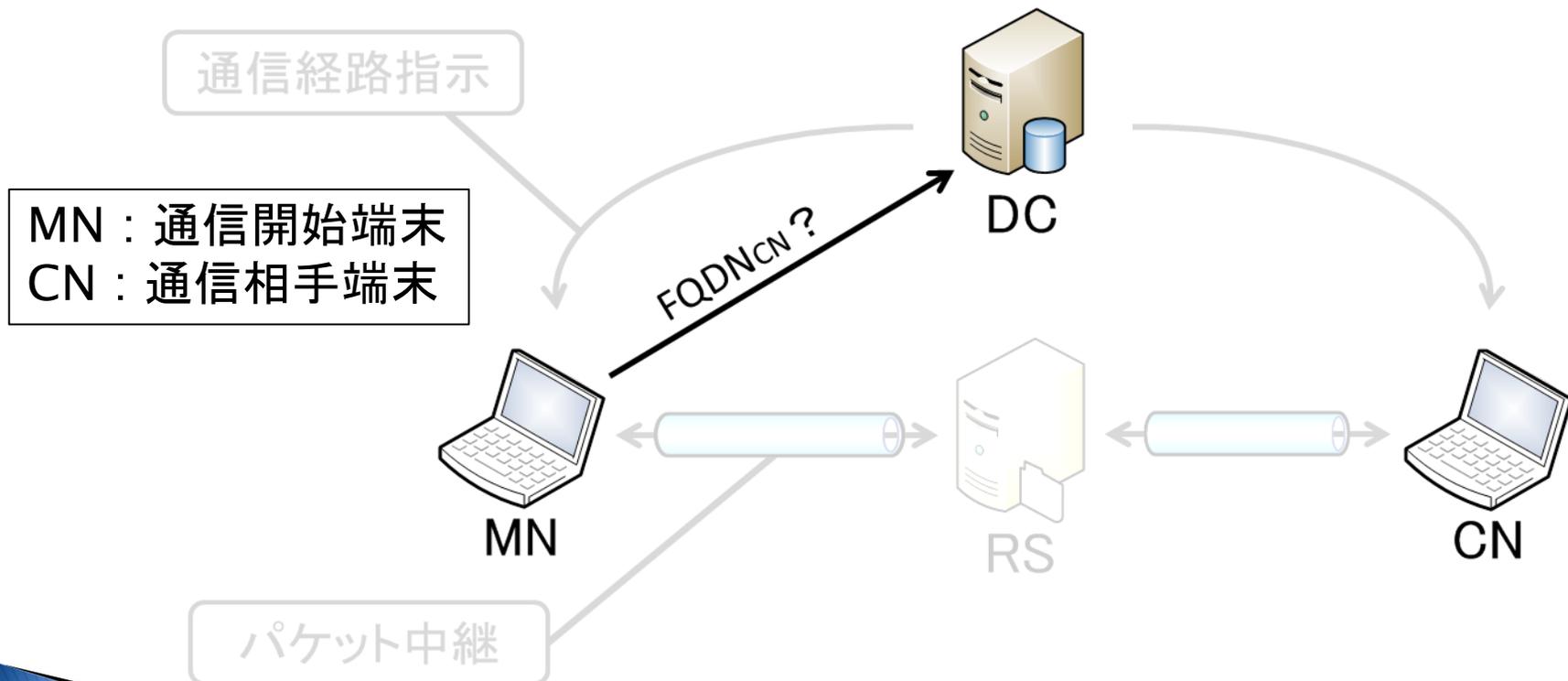
NTMobile (1 / 3)

- ▶ 仮想IPアドレスと呼ぶ変化しないアドレスを端末に割り当てる
- ▶ アプリケーションは仮想IPアドレスに基づいた通信を行う
→ **移動通信の実現**



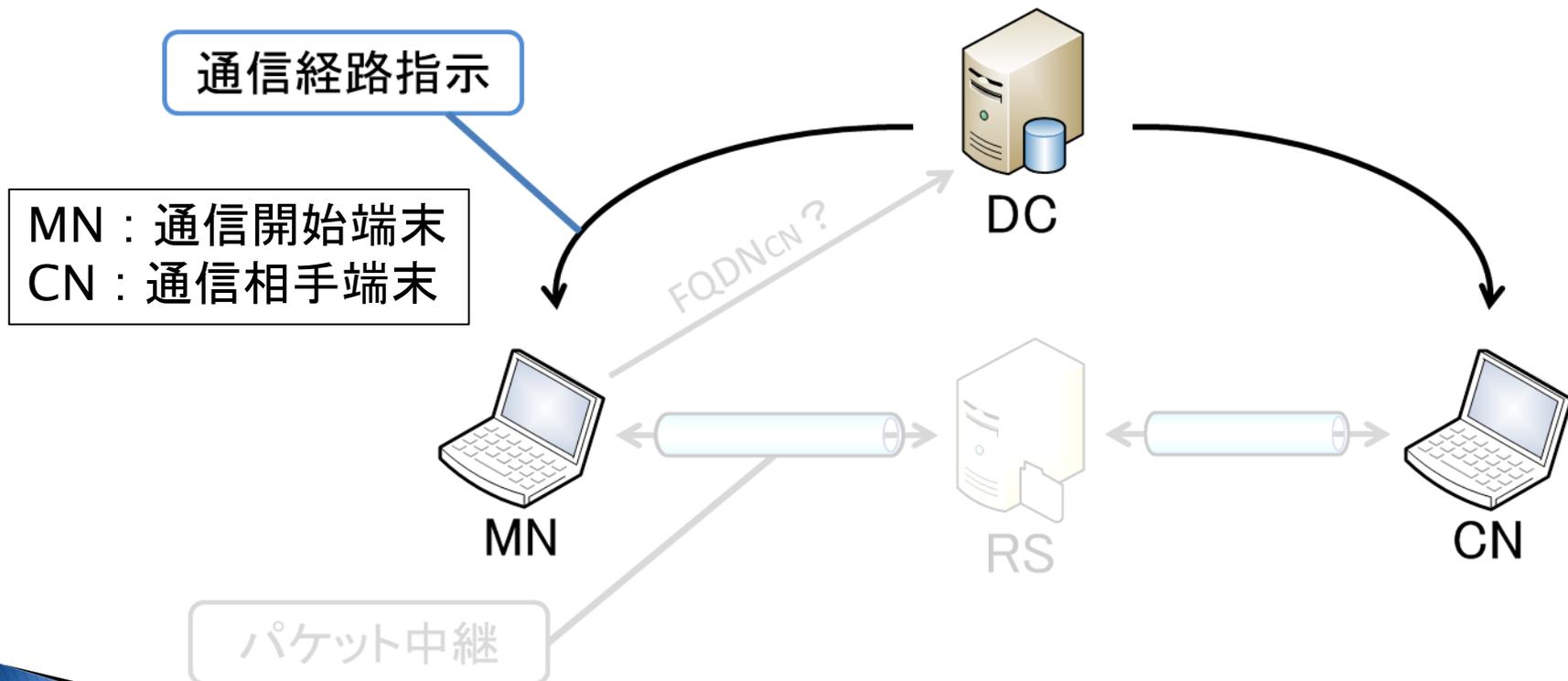
NTMobile (2/3)

- ▶ DC (Direction Coordinator) が最適な通信経路を指示
 - DNSの問い合わせをトリガとして通信経路構築
- ▶ 直接通信が不可能な場合はRS (Relay Server) が通信を中継
 - NAT越え, 異なるバージョン間通信の実現



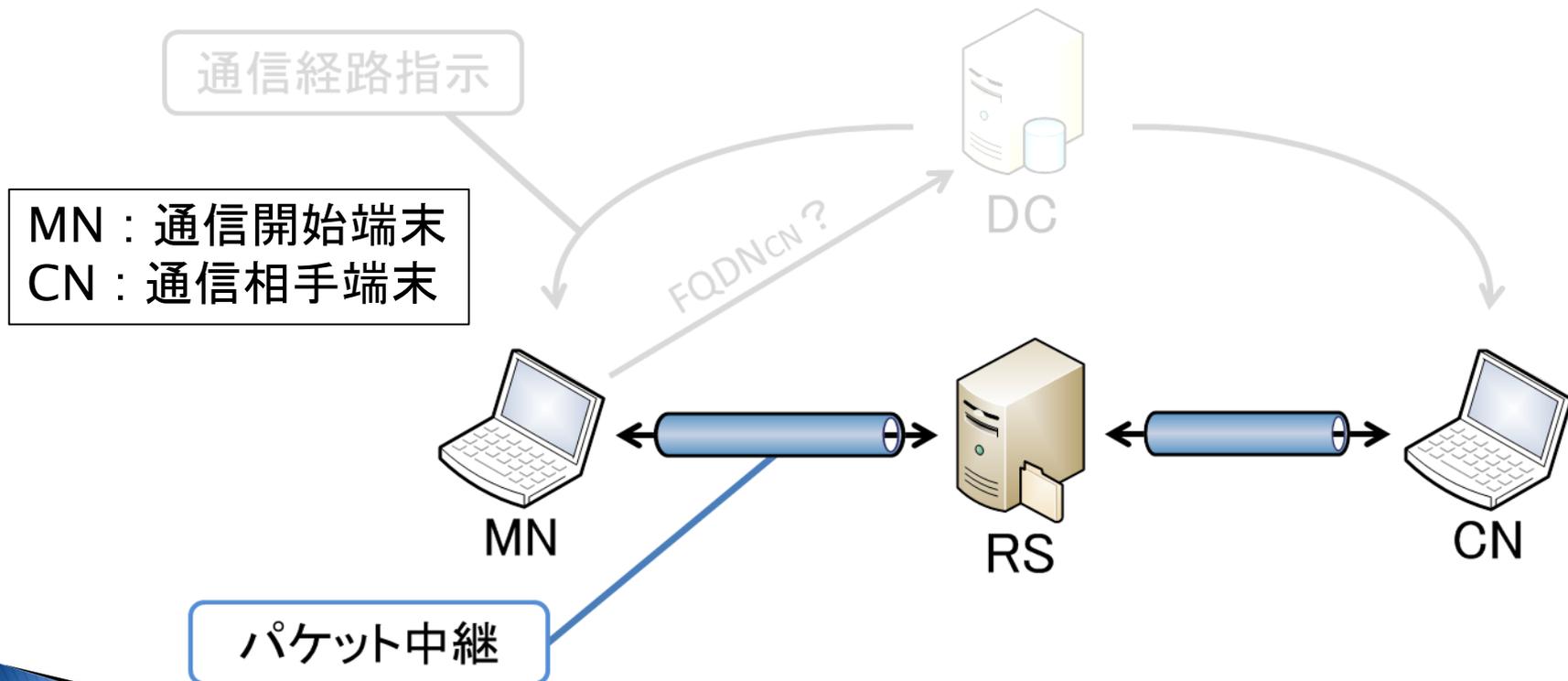
NTMobile (2/3)

- ▶ DC (Direction Coordinator) が最適な通信経路を指示
 - DNSの問い合わせをトリガとして通信経路構築
- ▶ 直接通信が不可能な場合はRS (Relay Server) が通信を中継
 - NAT越え, 異なるバージョン間通信の実現



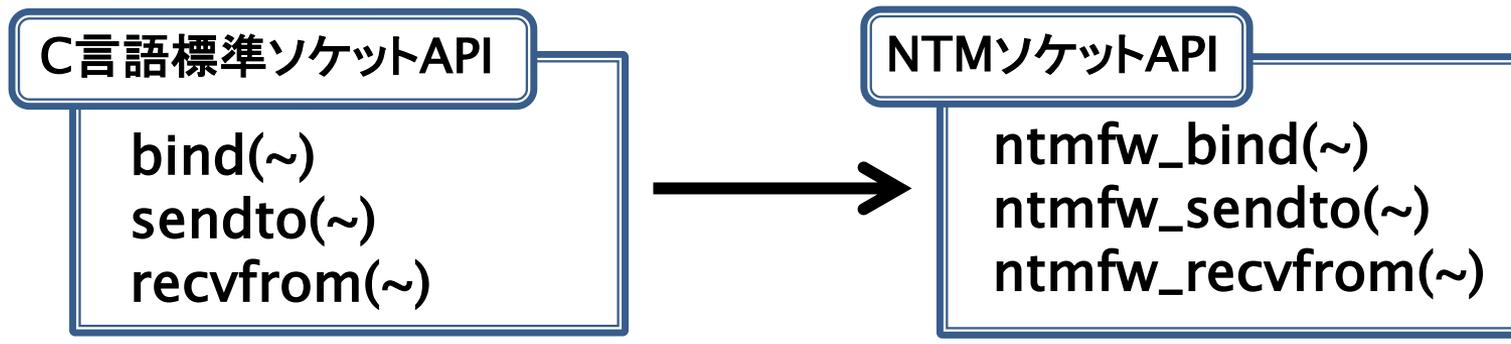
NTMobile (2/3)

- ▶ DC (Direction Coordinator) が最適な通信経路を指示
 - DNSの問い合わせをトリガとして通信経路構築
- ▶ 直接通信が不可能な場合はRS (Relay Server) が通信を中継
 - NAT越え, 異なるバージョン間通信の実現



NTMobile (3 / 3)

- ▶ NTMobileはこれらの機能をNTMfw (NTMobile Framework) と呼ぶアプリケーションライブラリとして提供
→ **カーネル空間への実装を必要としない**



しかし...

**アプリケーションの実装を変更する必要があるため、
既存のアプリケーションをそのまま使用できない**

既存技術の課題

- ▶ DSMIPv6 (Dual Stack Mobile IP version 6)
 - **カーネル空間への実装が必要となり、普及が進まない**
 - その他にも技術的な課題が多い
- ▶ NTMobile (Network Traversal with Mobility)
 - **アプリケーションの実装を変更する必要があり、既存のアプリケーションをそのまま使用できない**



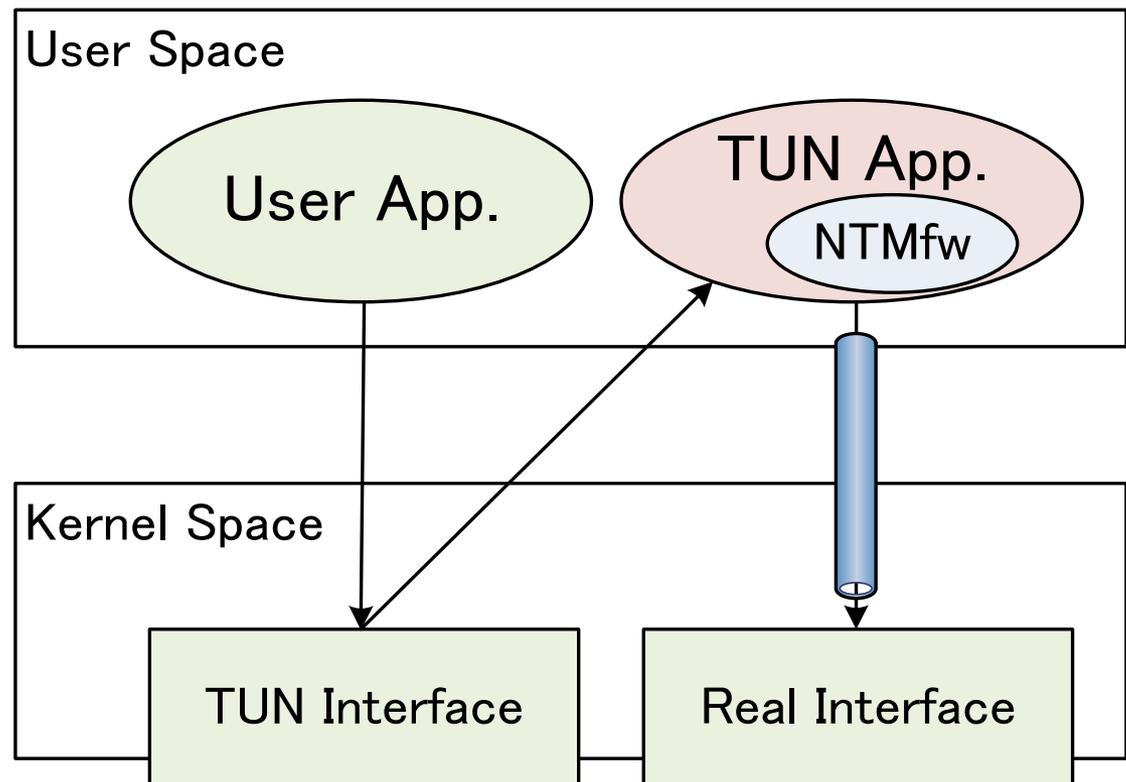
提案方式

カーネル空間への実装を必要とせず、既存のアプリケーションをそのまま使用できるシステム

提案方式 (1/2)

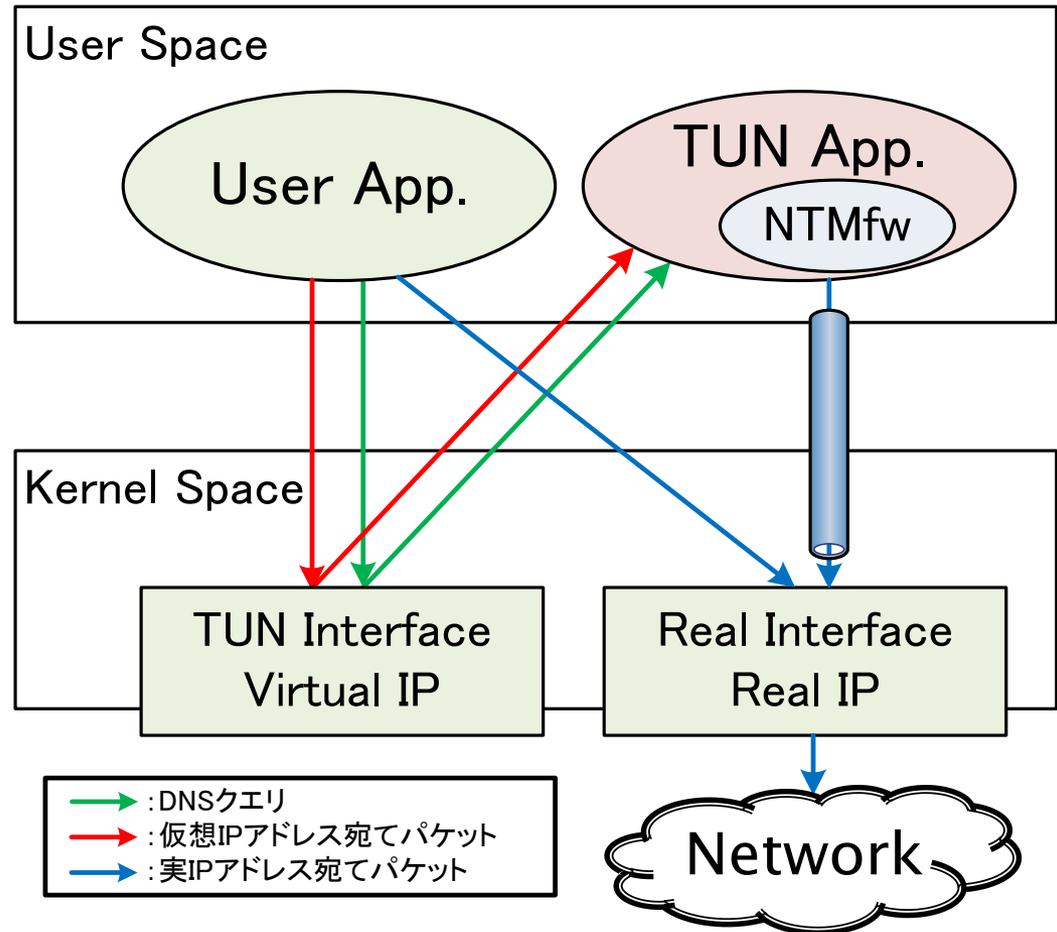
- ▶ NTMfwの機能をTUNインタフェースを利用したTUNアプリケーションとして実現

- ▶ TUNインタフェース
 - 送信パケットをユーザ空間のアプリへ渡す仕組み

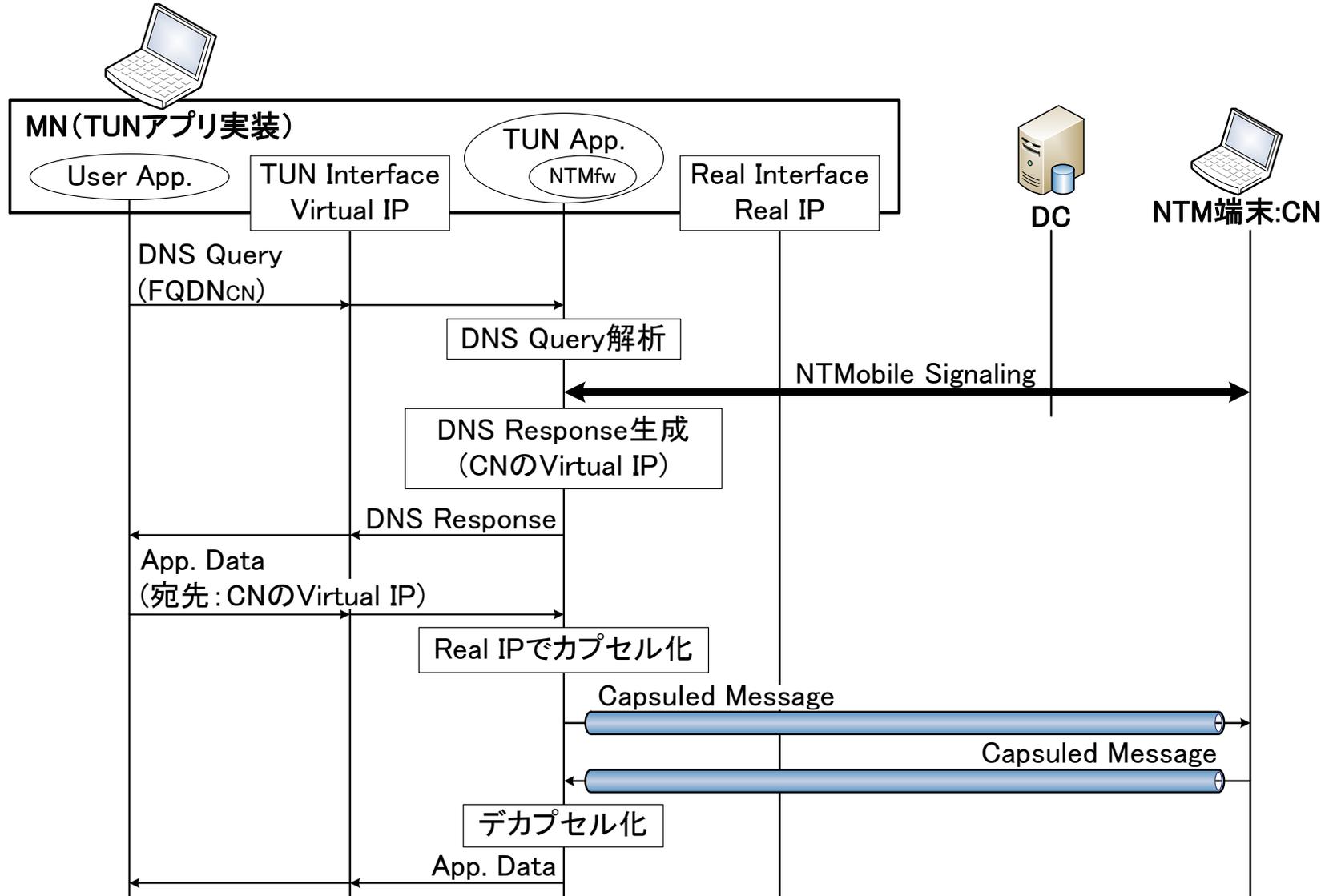


提案方式 (2/2)

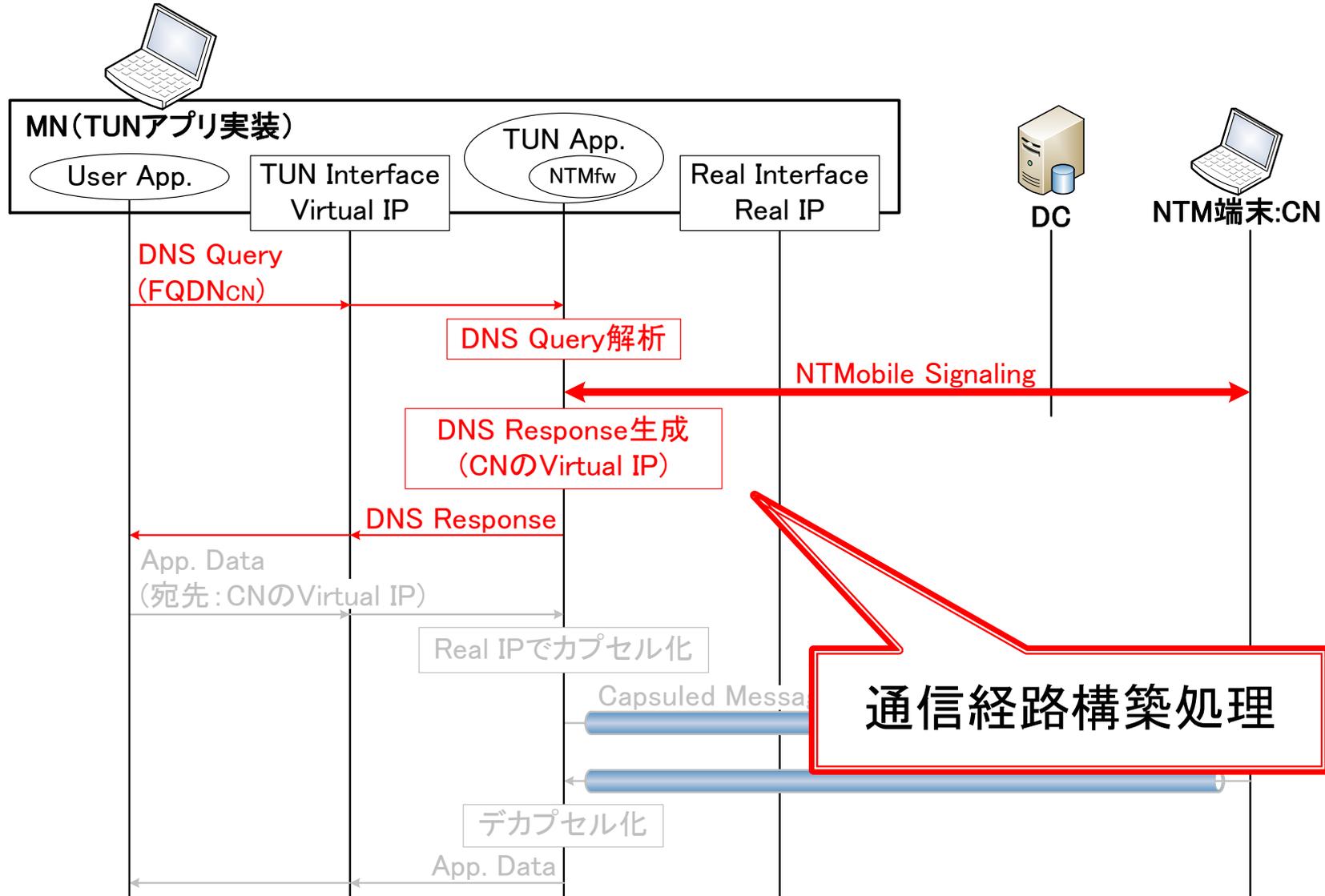
- ▶ TUNインタフェースに仮想IPアドレスを割り当て
- ▶ DNSクエリがTUNインタフェースを経由するようルーティングの設定



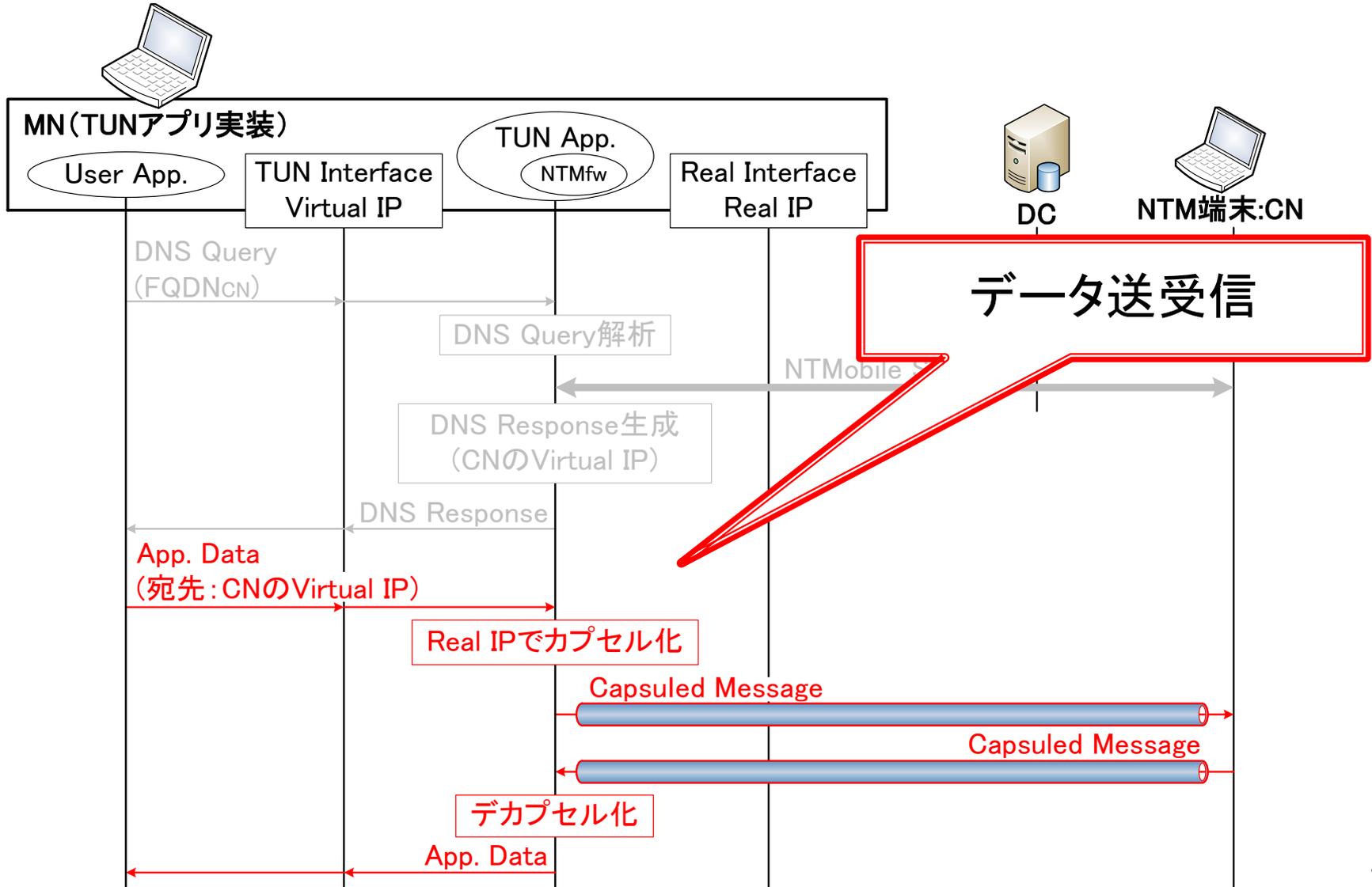
提案方式の動作シーケンス



提案方式の動作シーケンス



提案方式の動作シーケンス

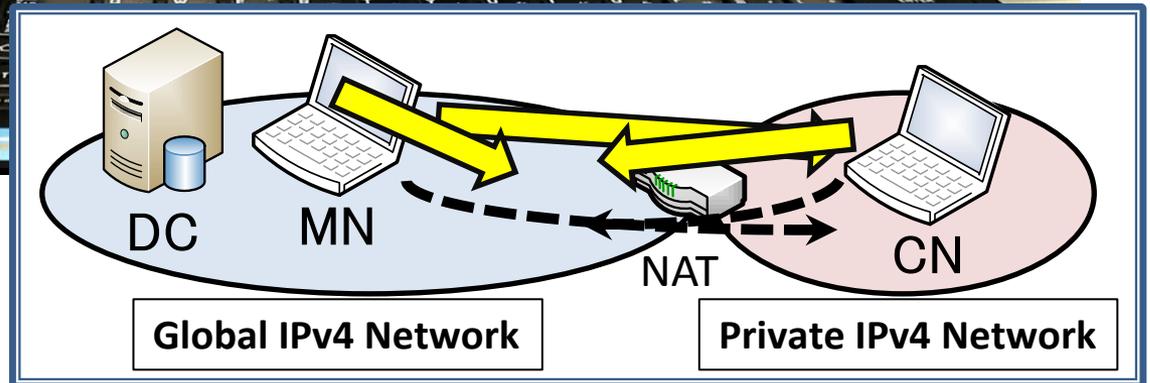


実装

- ▶ 提案方式の一部をLinux上に実装
 - DNSクエリのルーティングは静的に設定

- ▶ 動作検証
 - NAT越え, 移動通信ができることを確認
 - 次スライドにて実演

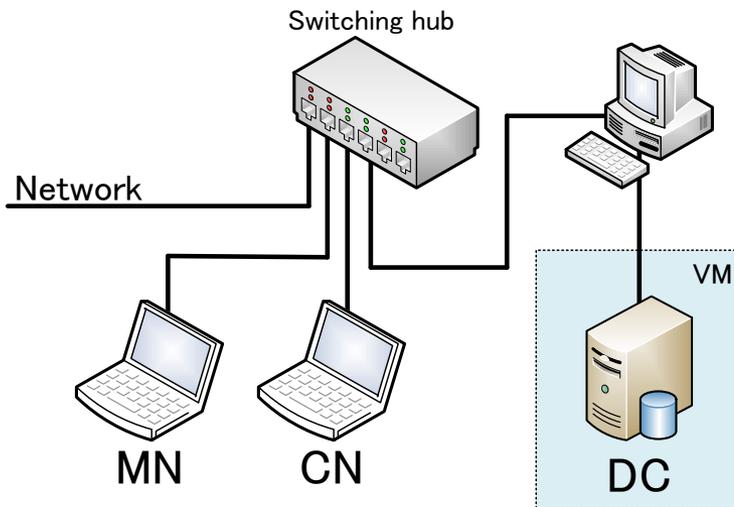
動作検証



性能測定 (1 / 4)

▶ 測定環境

- Linux実機を2台用意(MN, CN)し, 提案方式を実装
- 仮想マシン上にDCを構築



MN, CNの仕様	
OS	Ubuntu 14.04 32bit
CPU	Intel Core i5-2520M 2.50GHz
Memory	1944360 kB

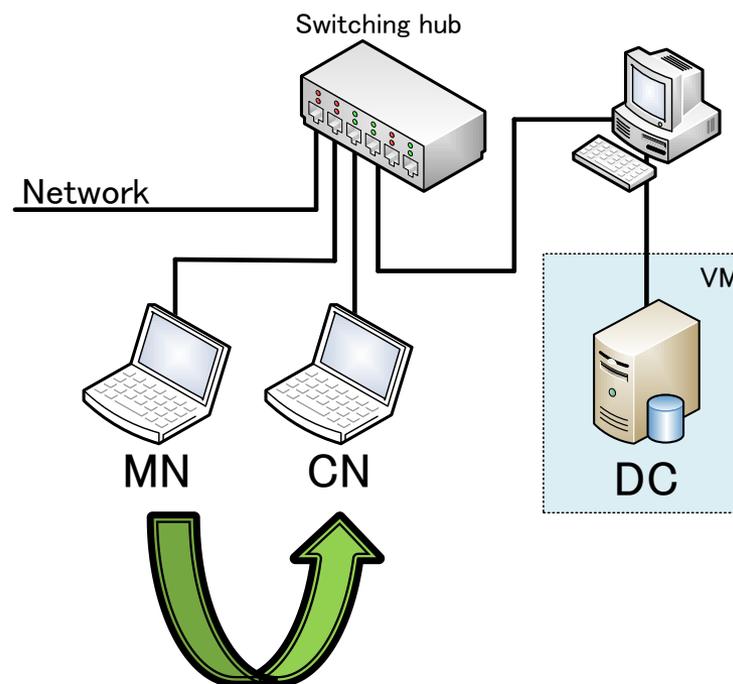
性能測定 (2/4)

▶ スループットの測定

- MNからCNへ向けて1400バイトのUDPパケットを送信
- 10秒間の測定を10回ずつ行い, 平均を算出

通常の通信	TUNアプリ 経由での通信
95.27Mbps	4.73Mbps

スループットの低下

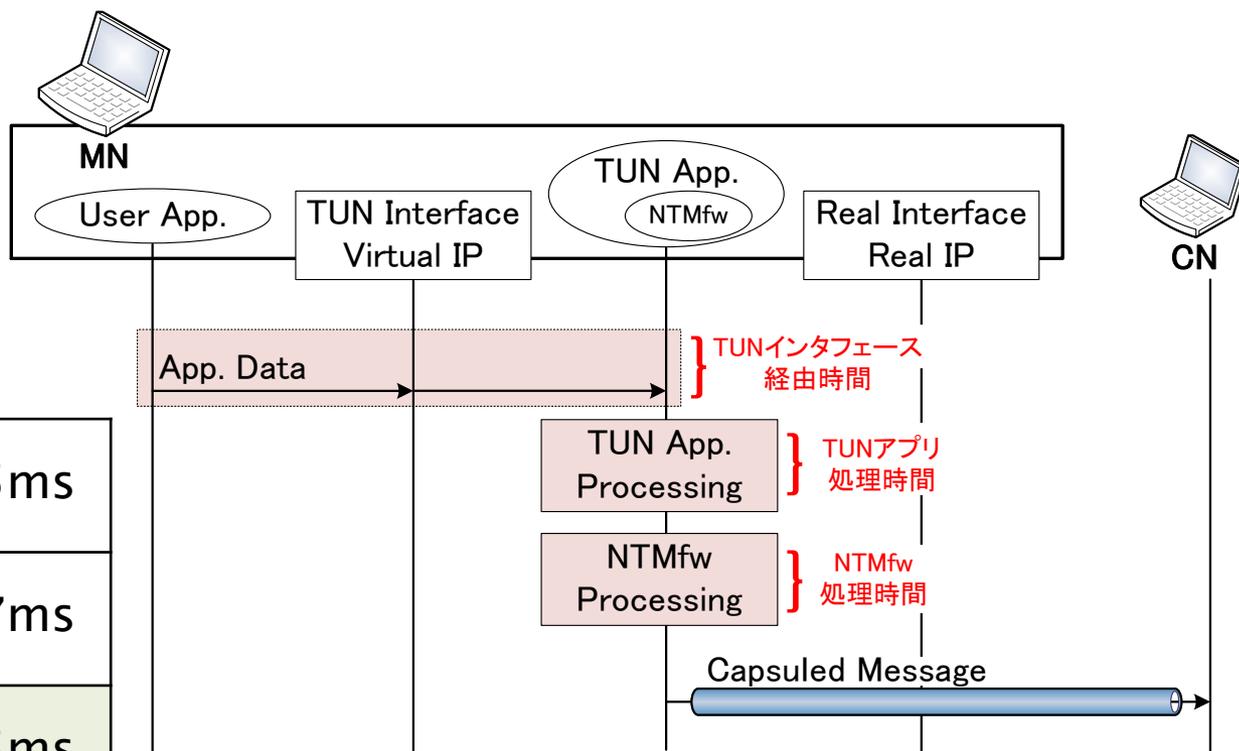


性能測定 (3/4)

▶ パケット送信側における処理時間の測定

- 1400バイトのUDPパケットにて測定
- 10回の平均値

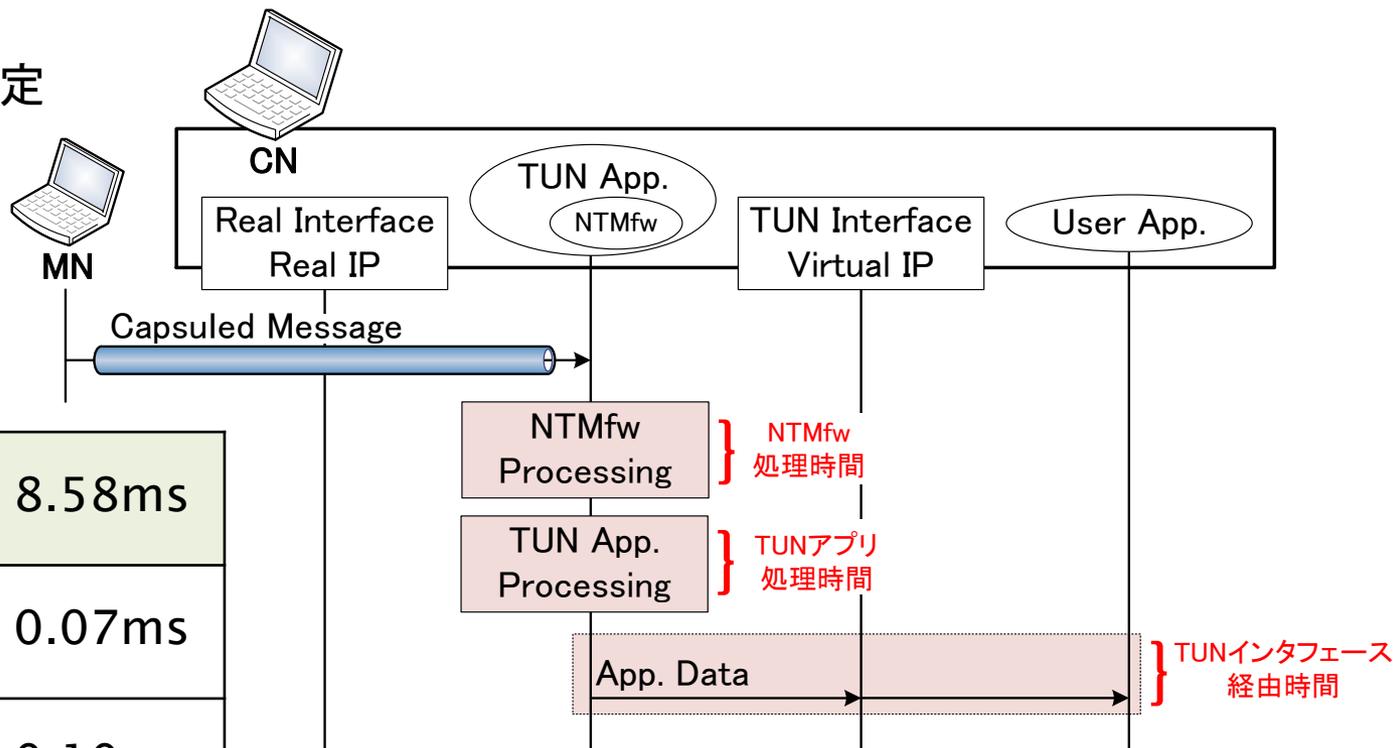
TUNインタフェース 経由時間	0.15ms
TUNアプリ 処理時間	0.07ms
NTMfw 処理時間	8.35ms



性能測定 (4/4)

▶ パケット受信側における処理時間の測定

- 1400バイトのUDPパケットにて測定
- 10回の平均値



NTMfw 処理時間	8.58ms
TUNアプリ 処理時間	0.07ms
TUNインタフェース 経由時間	0.10ms

考察

- ▶ 提案方式を使用した場合のスループットは4.73Mbps
 - 一般的な動画データや画像データ等のやり取りは快適に行える*
 - 大きなデータのやり取りでなければ利用可能
 - ▶ 送信側，受信側ともNTMfwの処理にもっとも時間を要した
 - NTMfwの暗号化・復号処理に時間を要している
 - 暗号化技術は現在のネットワークに必要不可欠
- 
- 今後は既存の暗号化プロトコルと性能比較を行う

* <https://mvno-smartphone.com/category6/entry101.html>
<http://mnp-sim.com/89>

まとめ

- ▶ ネットワークの問題を解決するTUNアプリケーションの提案
 - カーネル空間への実装を必要とせず、既存アプリケーションもそのまま使用可能

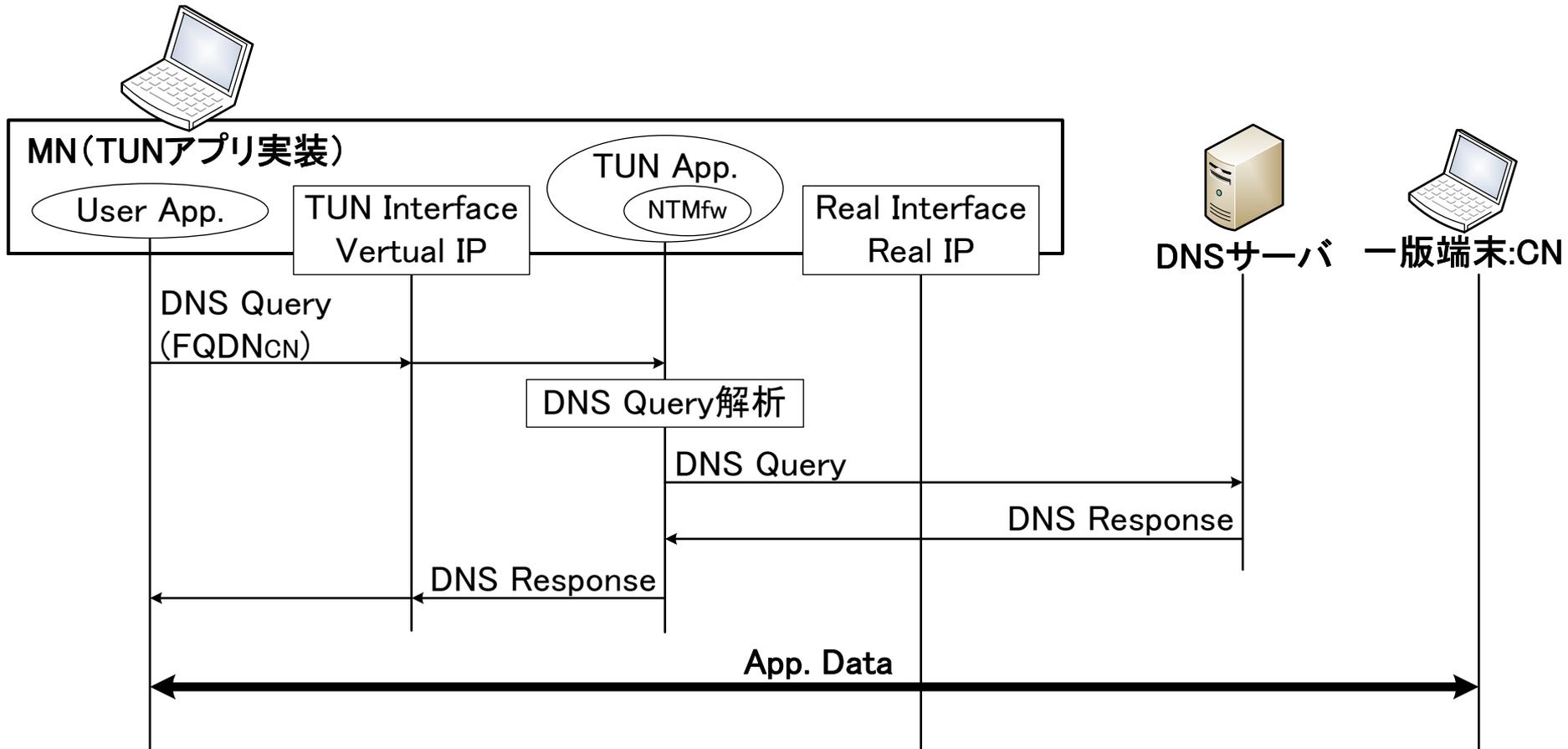
- ▶ 提案方式を実装し、スループットを測定
 - 用途によっては利用可能
 - NTMobileにおける暗号化・復号処理に時間を要している

- ▶ 今後の方針
 - DNSクエリのルーティング設定方法の検討
 - 既存の暗号化プロトコルとのスループットの比較

補足資料

一般端末との通信

- ▶ DNSクエリをそのままDNSサーバに送信



NTMobile実装方式の比較

	NTMfw	アダプタ型	TUN型 (提案方式)	VPNService型
既存アプリの使用	×	○	○	○
物理デバイスの有無	○	×	○	○
複数アプリへの適用	○	○	○	×
管理者権限の必要性の有無	○	○	×	○
多岐OSへの適用	○	○	○	△

DNSクエリのルーティング設定方法

- ▶ DNSサーバ宛てのパケットを全てルーティング

NTMobile

- ▶ 仮想IPv4アドレスは実ネットワークで使用されていないアドレス帯域[198.18.0.0/15]を使用
- ▶ 利用可能なIPv4アドレスは約13万個

TUNとTAP

- ▶ TUNインタフェース
 - IPパケットをフック

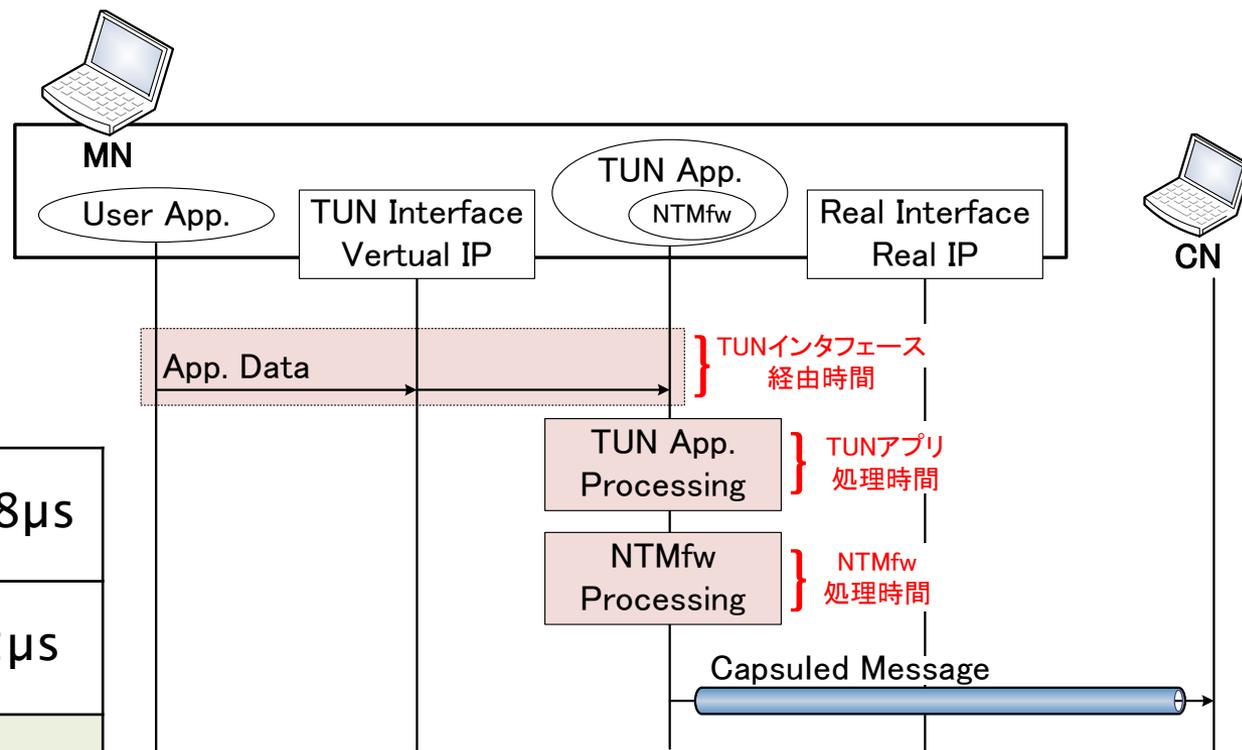
- ▶ TAPインタフェース
 - イーサネットフレームをフック

評価 (2/3)

▶ パケット送信側における処理時間の測定

- 47バイトのUDPパケットにて測定
- 10回の平均値

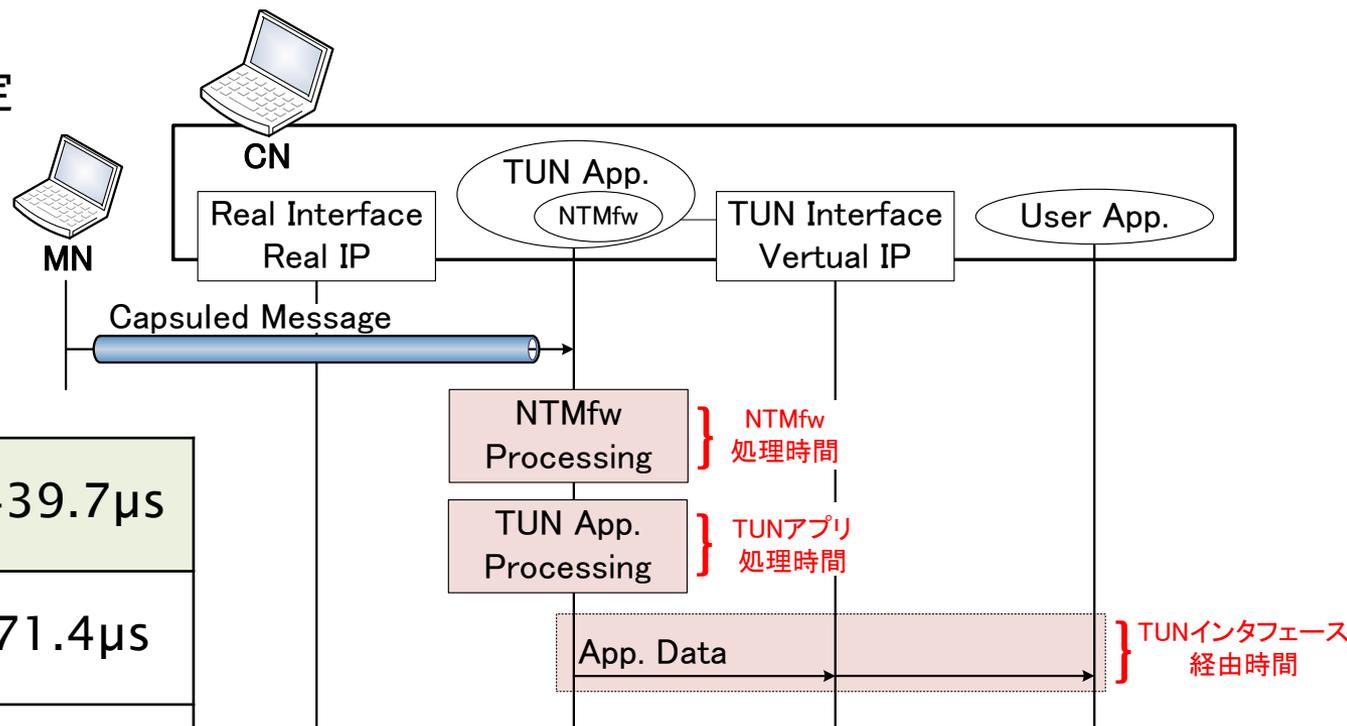
TUNインタフェース 経由時間	152.8 μ s
TUNアプリ 処理時間	69.2 μ s
NTMfw 処理時間	320.3 μ s



評価 (3/3)

▶ パケット受信側における処理時間の測定

- 47バイトのUDPパケットにて測定
- 10回の平均値



NTMfw 処理時間	439.7 μ s
TUNアプリ 処理時間	71.4 μ s
TUNインタフェース 経由時間	119.7 μ s

DSMIPv6

▶ 様々な課題

- 移動端末毎に IPv4 グローバルアドレスが必要
- IPv4 環境では必ず HA (Home Agent) を経由した冗長経路となる

**カーネル空間への実装が必要
普及が進まない**

