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

140441100 稲垣 智
渡邊研究室

1. はじめに

LAN 内での通信を前提とすると端末間の通信に制約がほとんどないため, 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生じ, 開発に時間を要する. 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. NTMobile（Network Traversal with Mobilty）[1] 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 NTMobile は既存のアプリケーションをそのまま利用できないという課題がある.

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するシステムの実現方法を提案する.

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 NTMfw と呼ぶ通信ライブラリを利用することにより実現できる. アプリケーションは NTMfw を適宜呼び出すことにより, 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ある.

3. 提案方式

提案方式では, NTMfw の機能を TUN/TAP インタフェースを用いた 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, 一般のアプリケーション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. この仕組みは VPN 通信のためのもので, 既存のアプリケーションを変更することなくパケットのカプセル化を実現できるため, NTMobile のカプセル化通信に利用できる. 提案手法の TUN アプリではフックしたパケットに 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.

TUN アプリ起動時に NTMobile の登録処理を行い, 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, 及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.

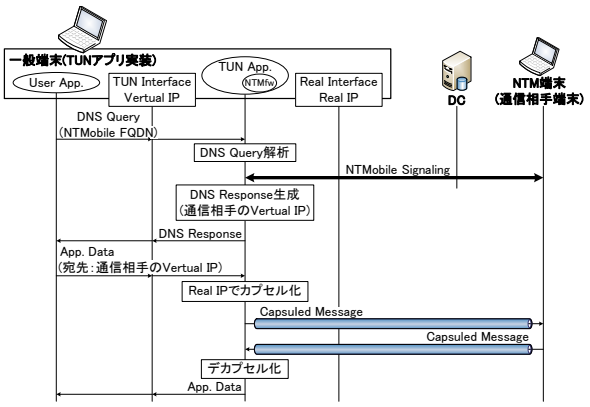


図 1: 提案方式における動作シーケンス

図 1 に提案方式における動作シーケンスを示す. ユーザアプリが通信相手の FQDN を指定することで 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へ返信する. 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宛先が仮想アドレスとなるため, 全て TUN インタフェースを通じて TUN アプリが受け取る. TUN アプリは受け取ったパケットに NTMfw による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.

4. 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.

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 アプリを提案した. また提案手法を Linux 上に実装し, 通信接続性を確立できることを確認した.

参考文献

[1] 上醉尾一真ほか：情報処理学会論文誌, Vol.54, No.10, pp.2288-2299 (2013).

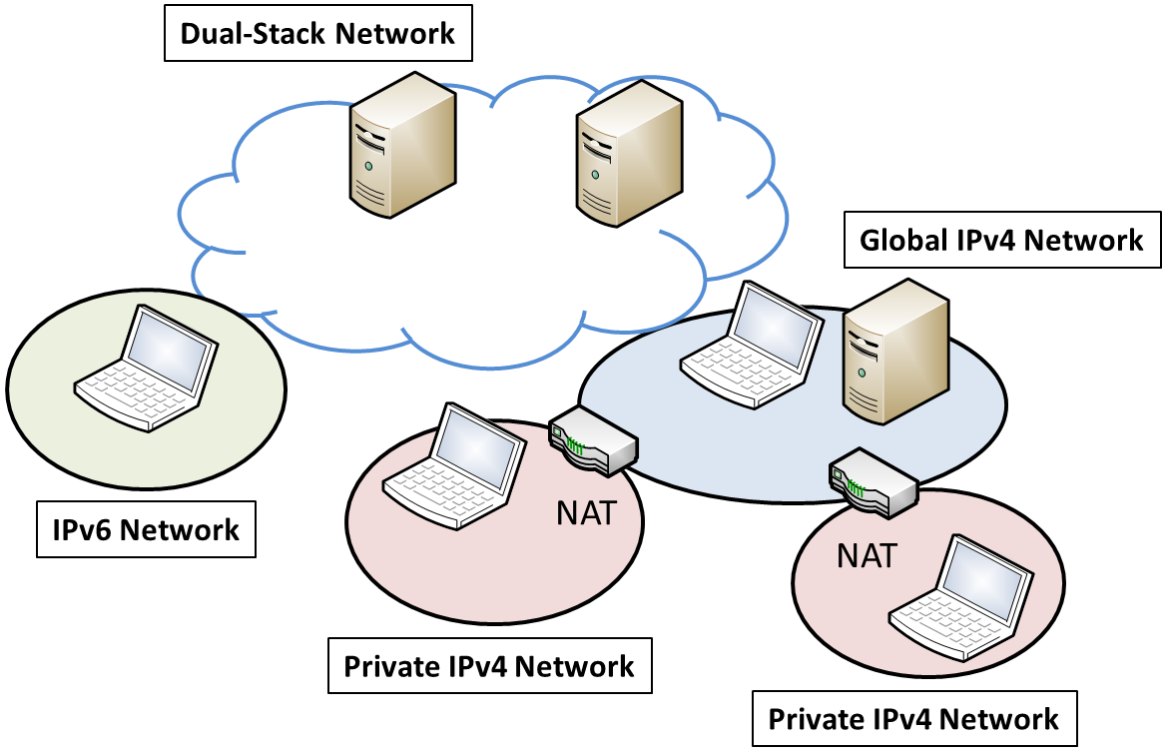
LAN内通信システムをインターネット上で利用可能にするTUNアプリの提案と実装

渡邊研究室
140441100
稲垣 智

Watanabe Lab.

研究背景

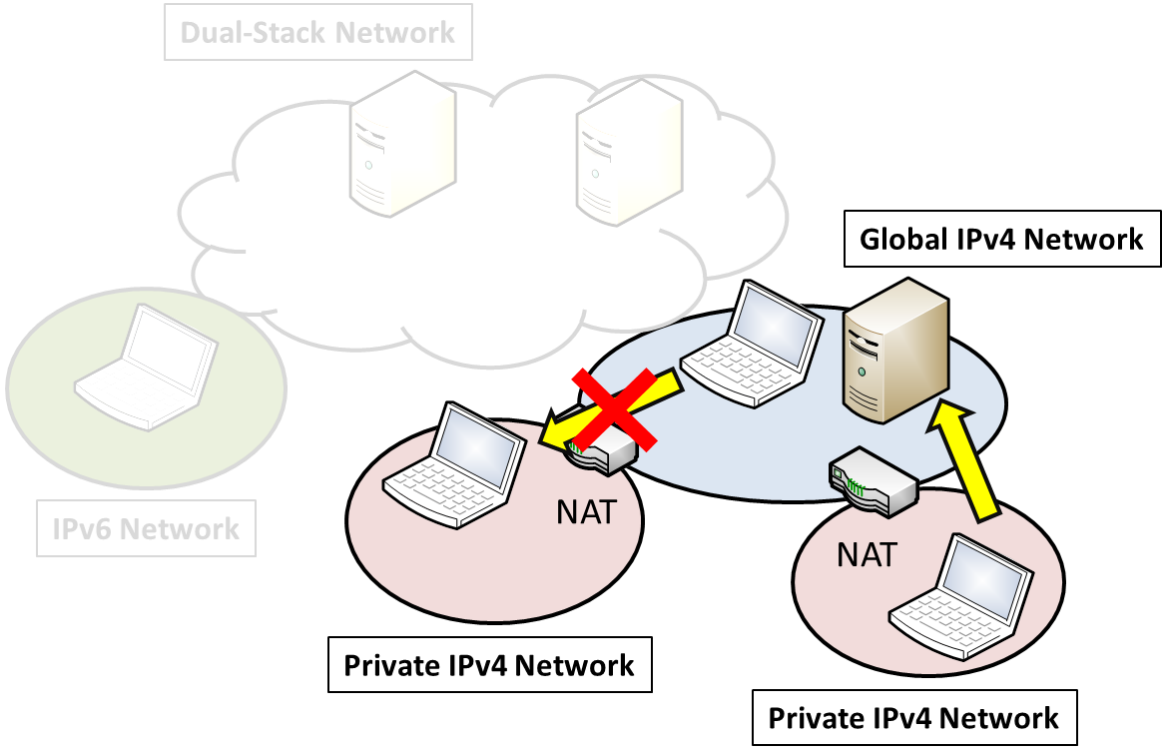
- NAT越え問題
- IPv4–IPv6間での通信不可問題
- 移動透過性の課題



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

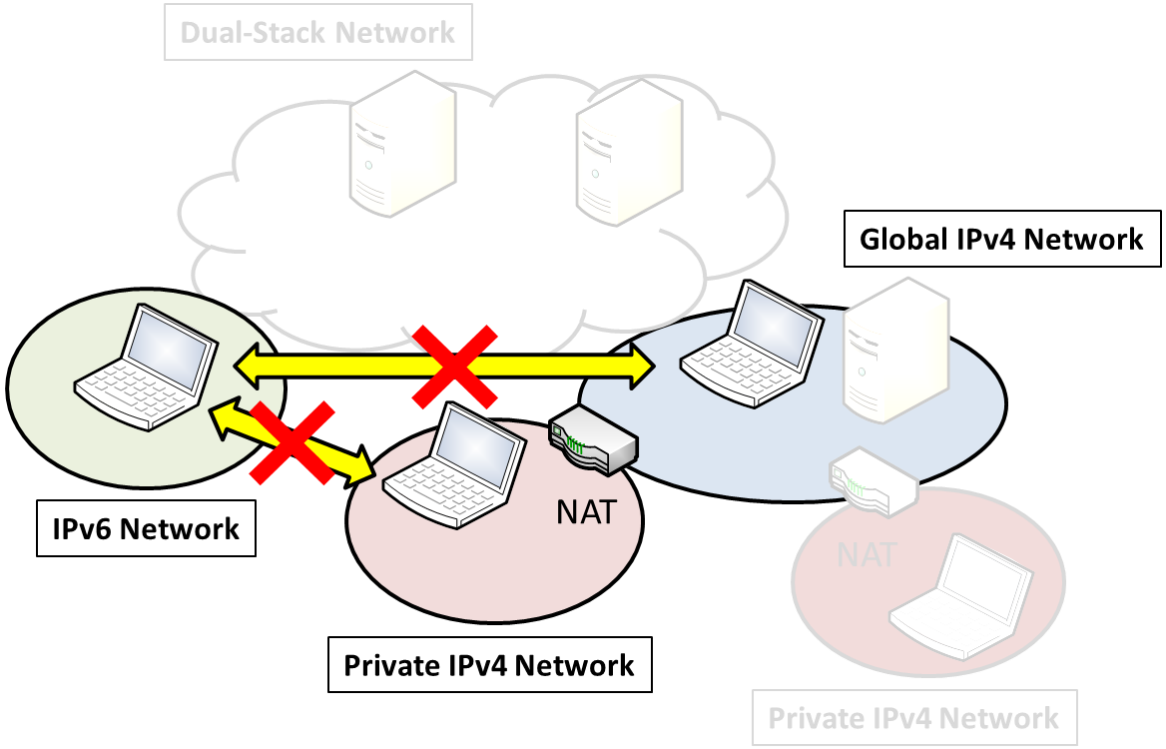
- NAT越え問題
- IPv4–IPv6間での通信不可問題
- 移動透過性の課題



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

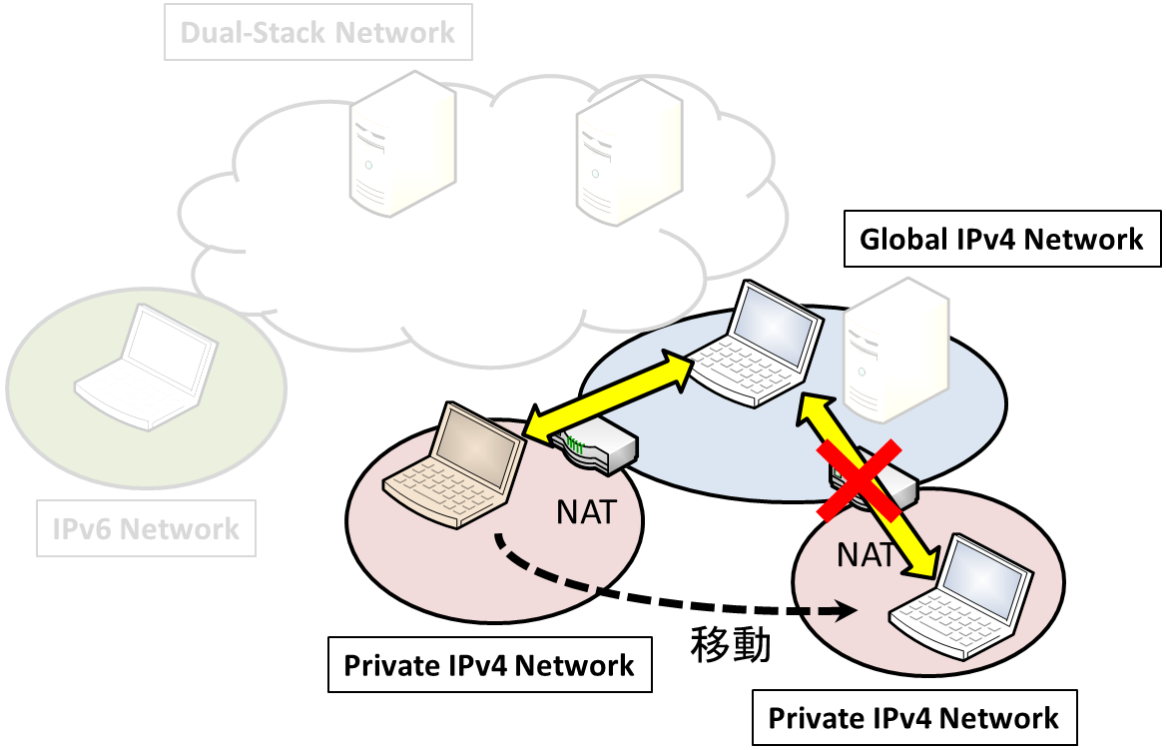
- NAT越え問題
- IPv4–IPv6間での通信不可問題
- 移動透過性の課題



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

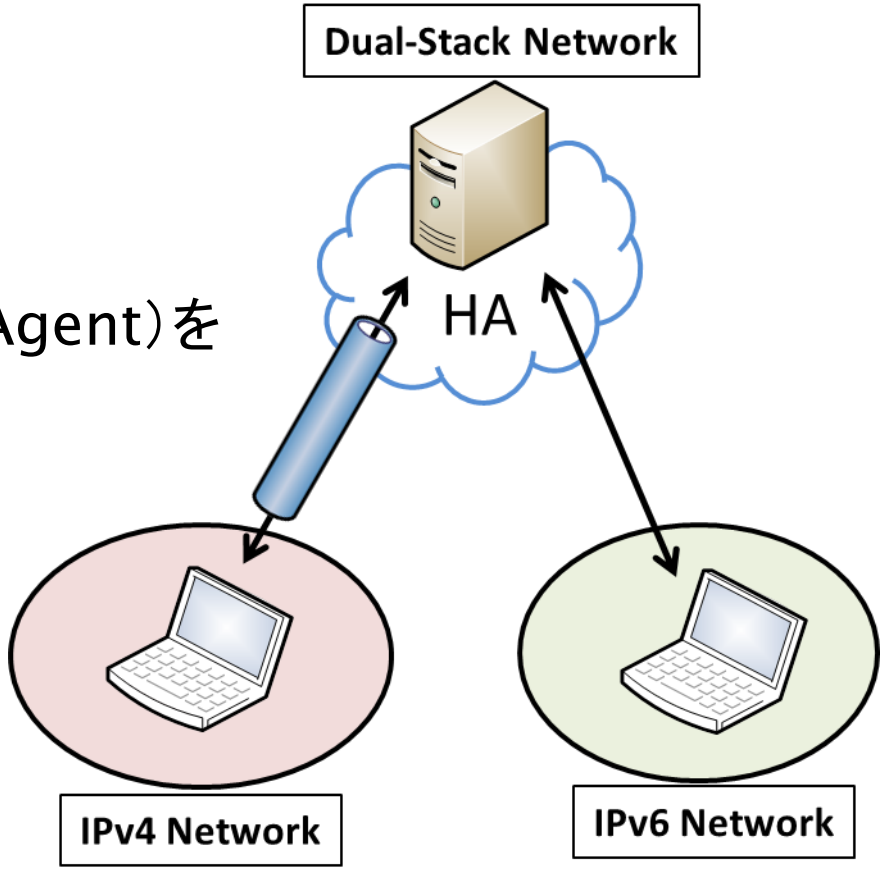
- NTMobile（Network Traversal with Mobility）*2

*1 RFC5555, 2009
*2 納堂ほか：信学技報, Vol.116, No.10, pp.2288-2299, 2017

DSMIPv6

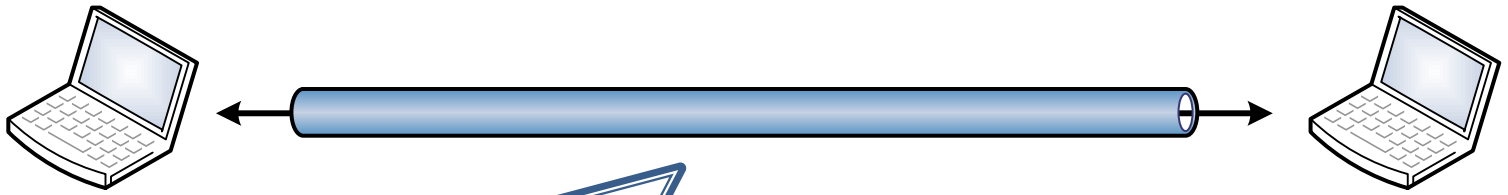
- 様々な課題
  - 移動端末毎に
    IPv4グローバルアドレスが必要
  - IPv4環境では必ずHA(Home Agent)を
    経由した冗長経路となる

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

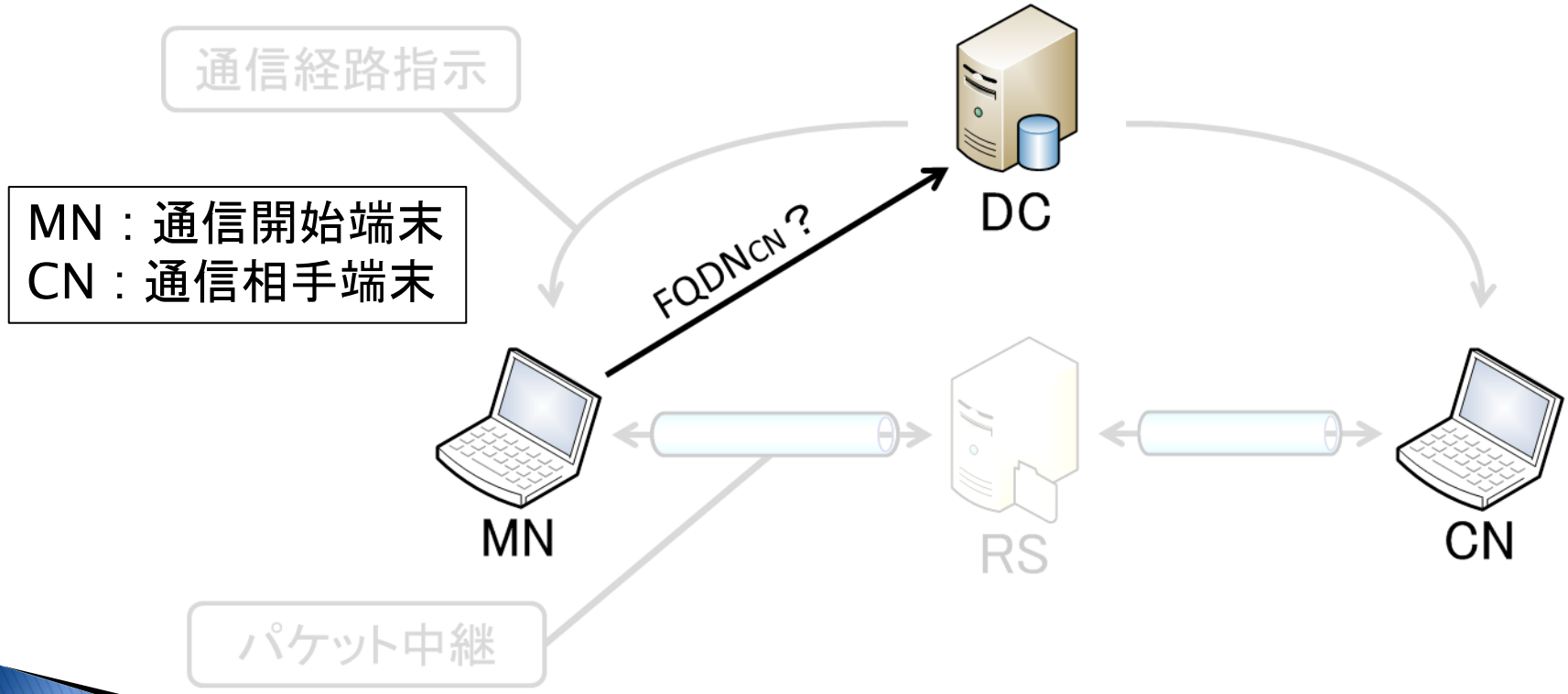


アプリケーションによる
パケット

NTMobileによりカプセル化された
パケット

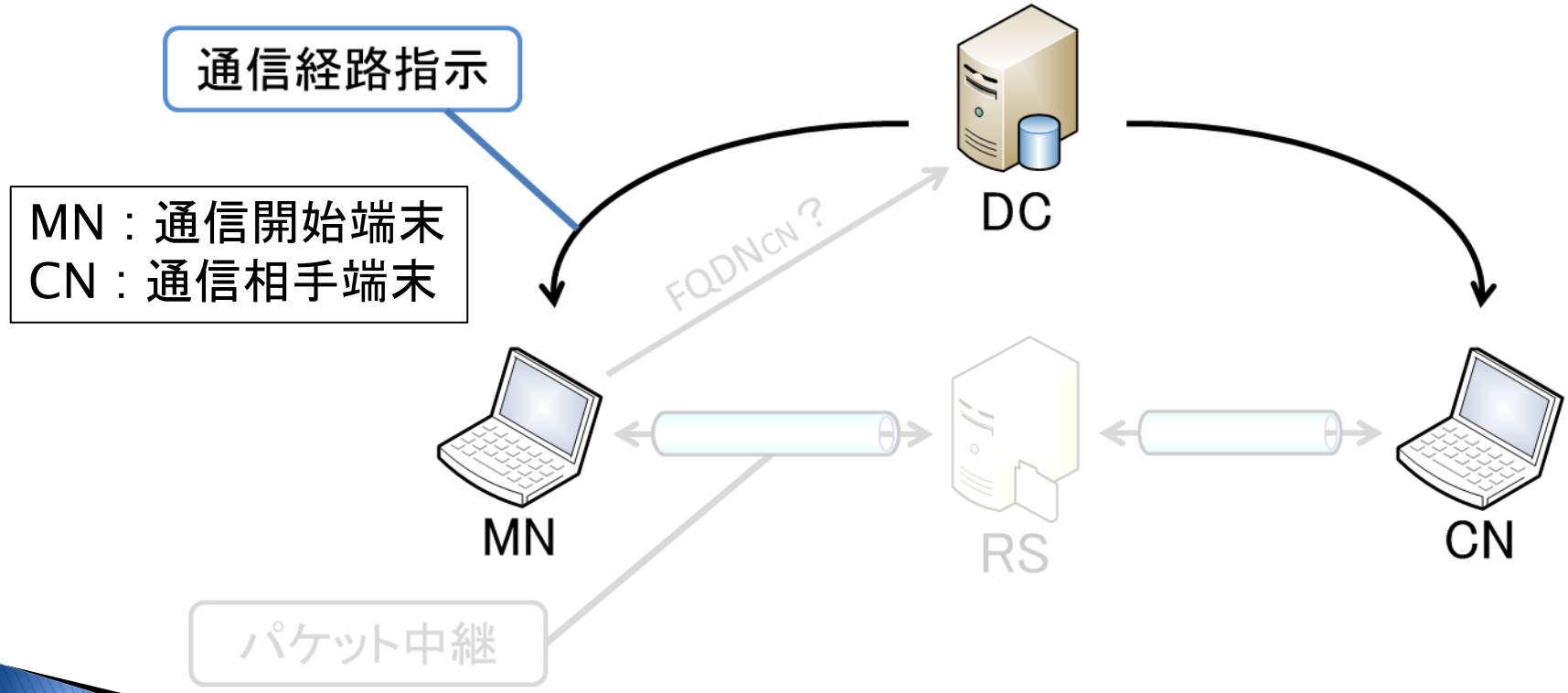
NTMobile (2/3)

- 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



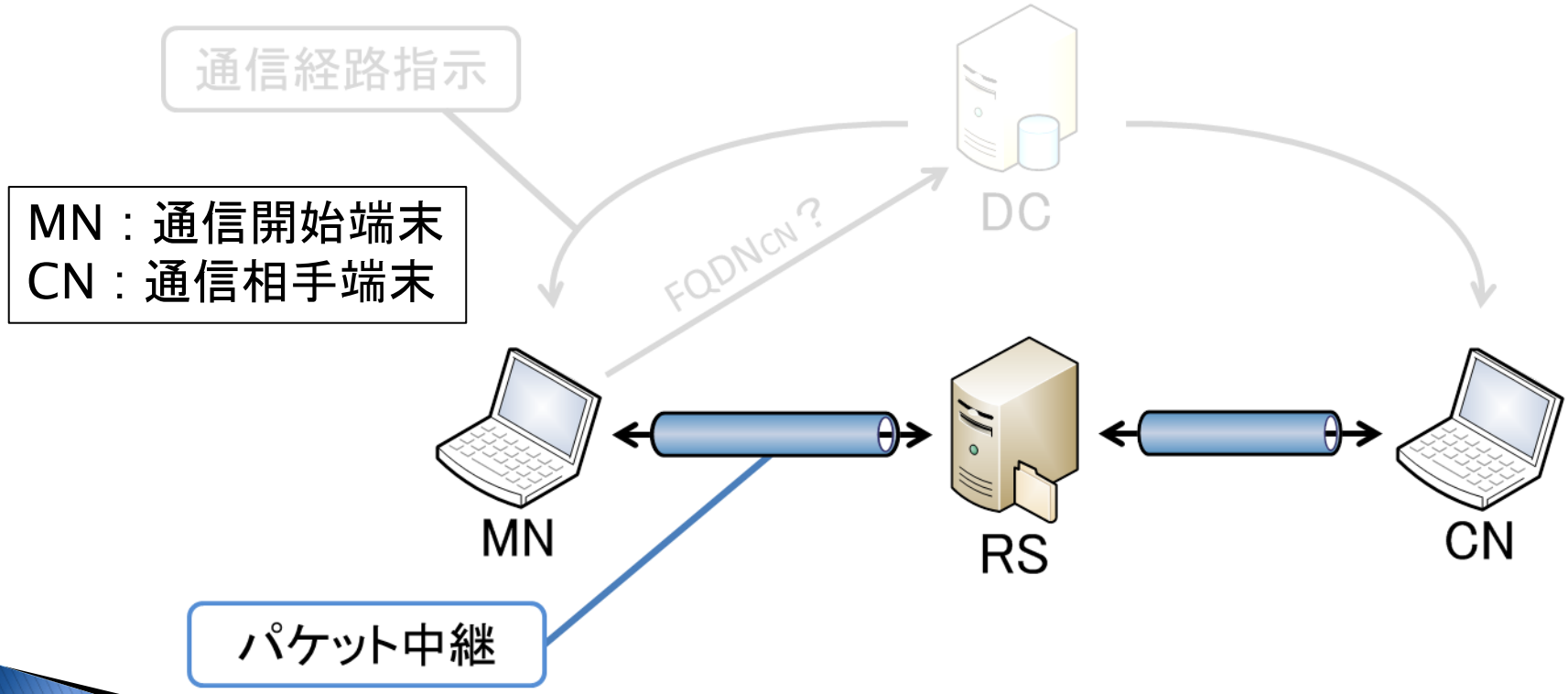
NTMobile (2/3)

- 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



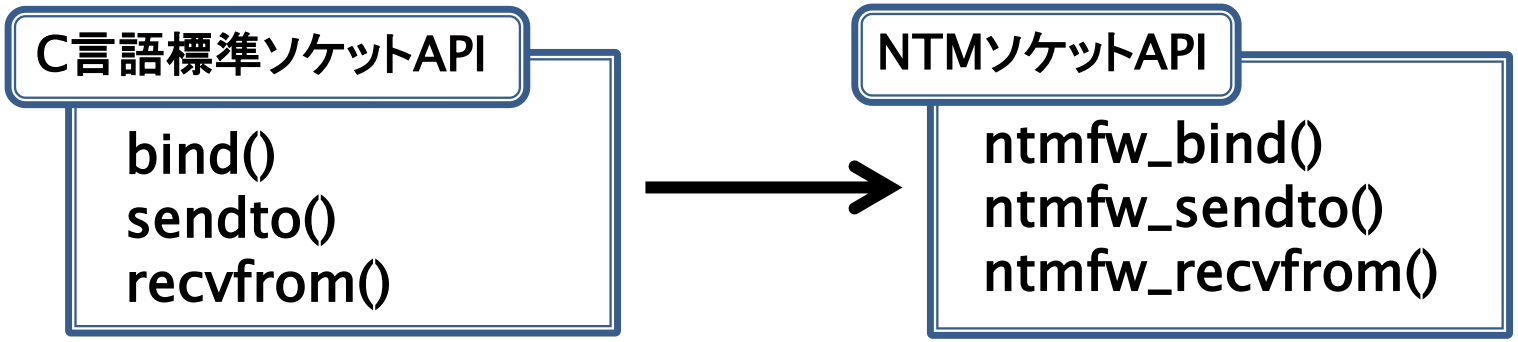
NTMobile (2/3)

- 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

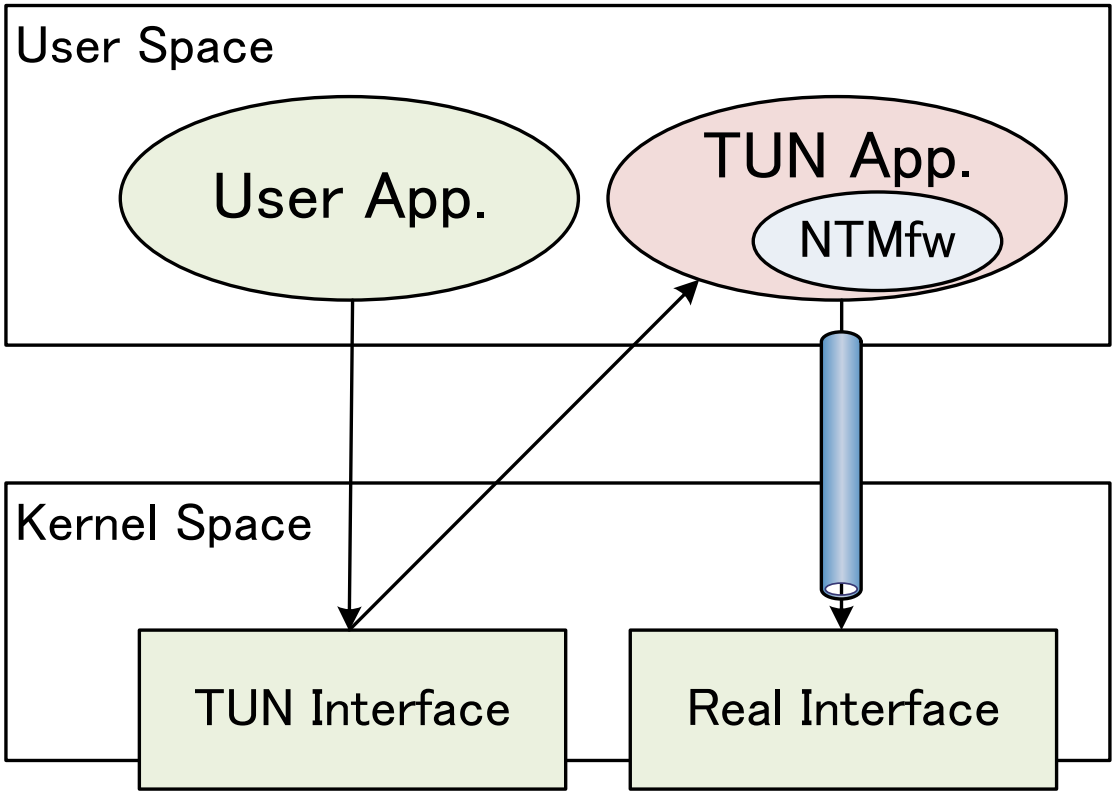
- NTMobile（Network Traversal with Mobility）
  - アプリケーションの実装を変更する必要があり，
    既存のアプリケーションをそのまま使用できない

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**

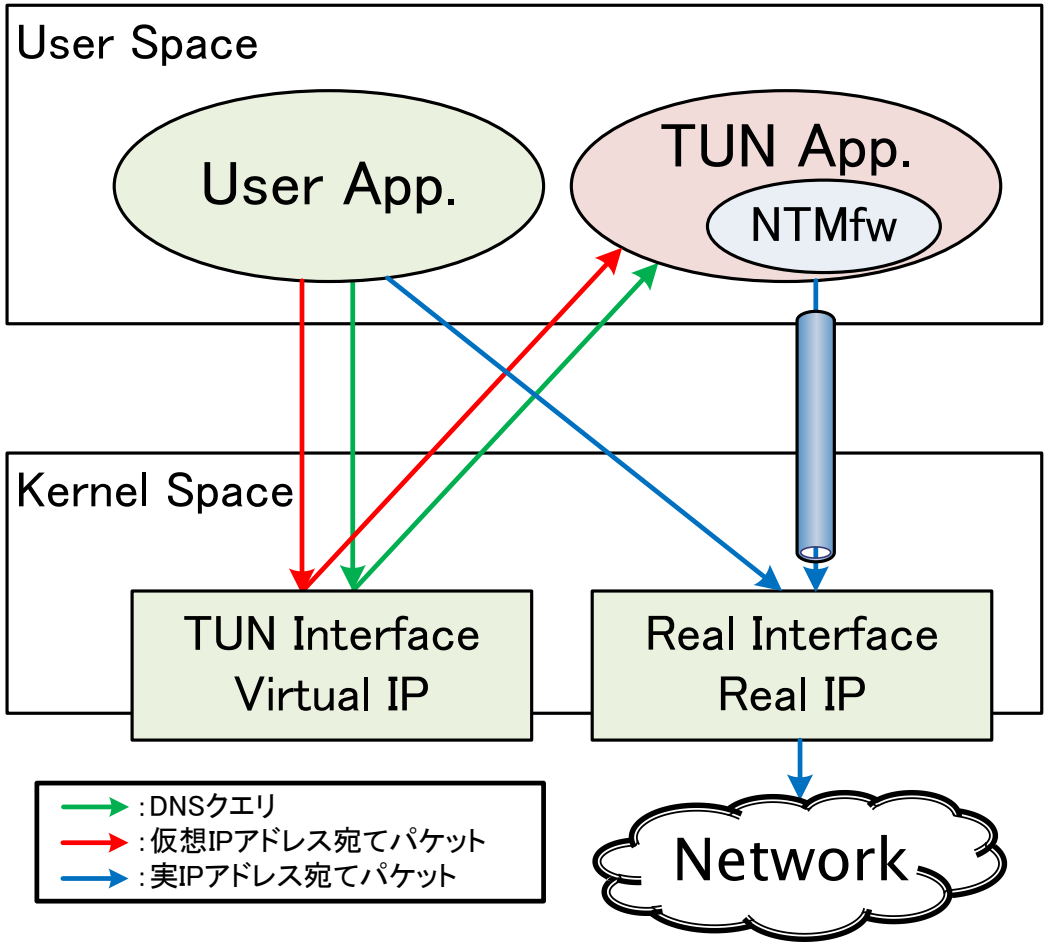
提案方式 (1/2)

- 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

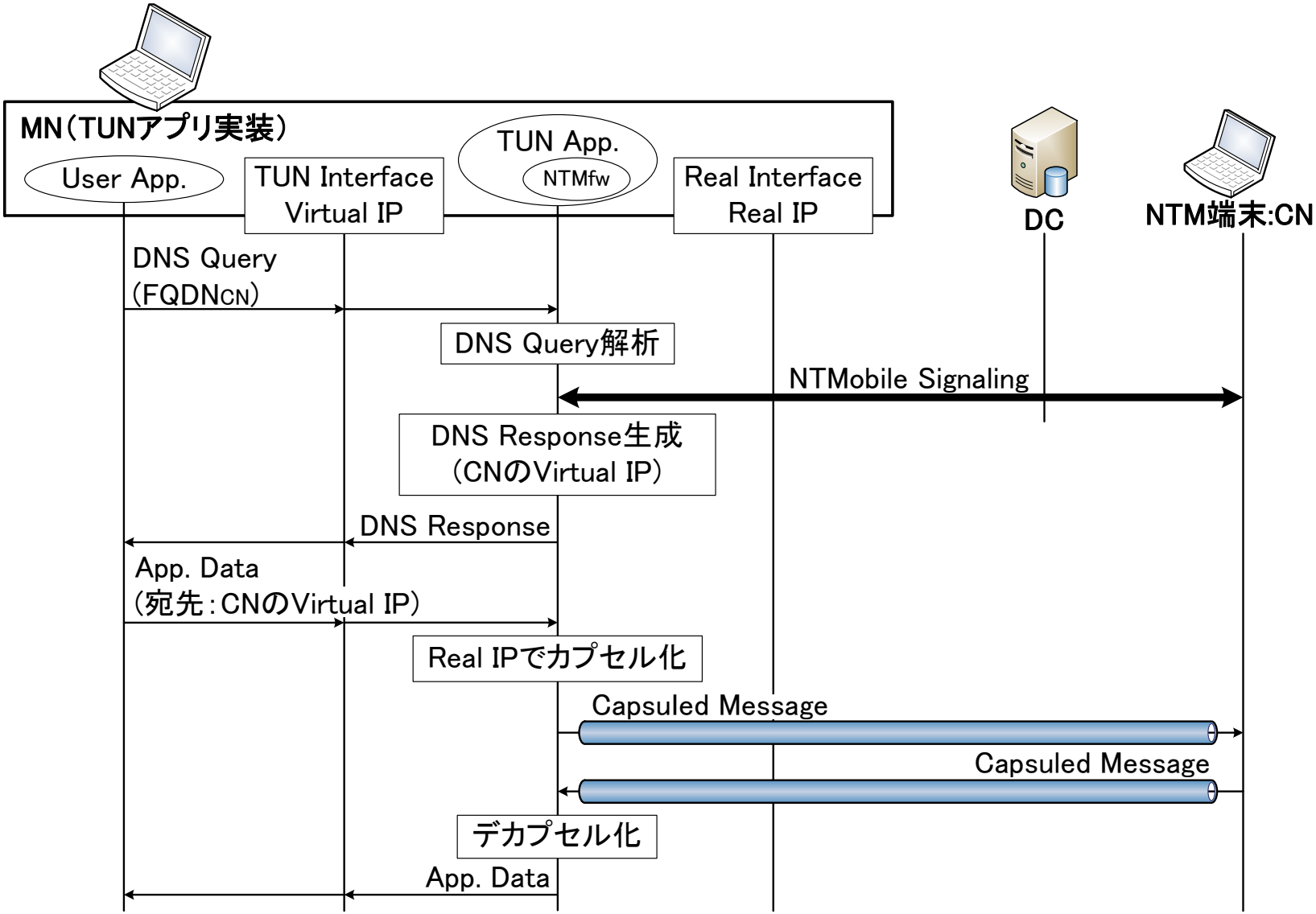


提案方式 (2/2)

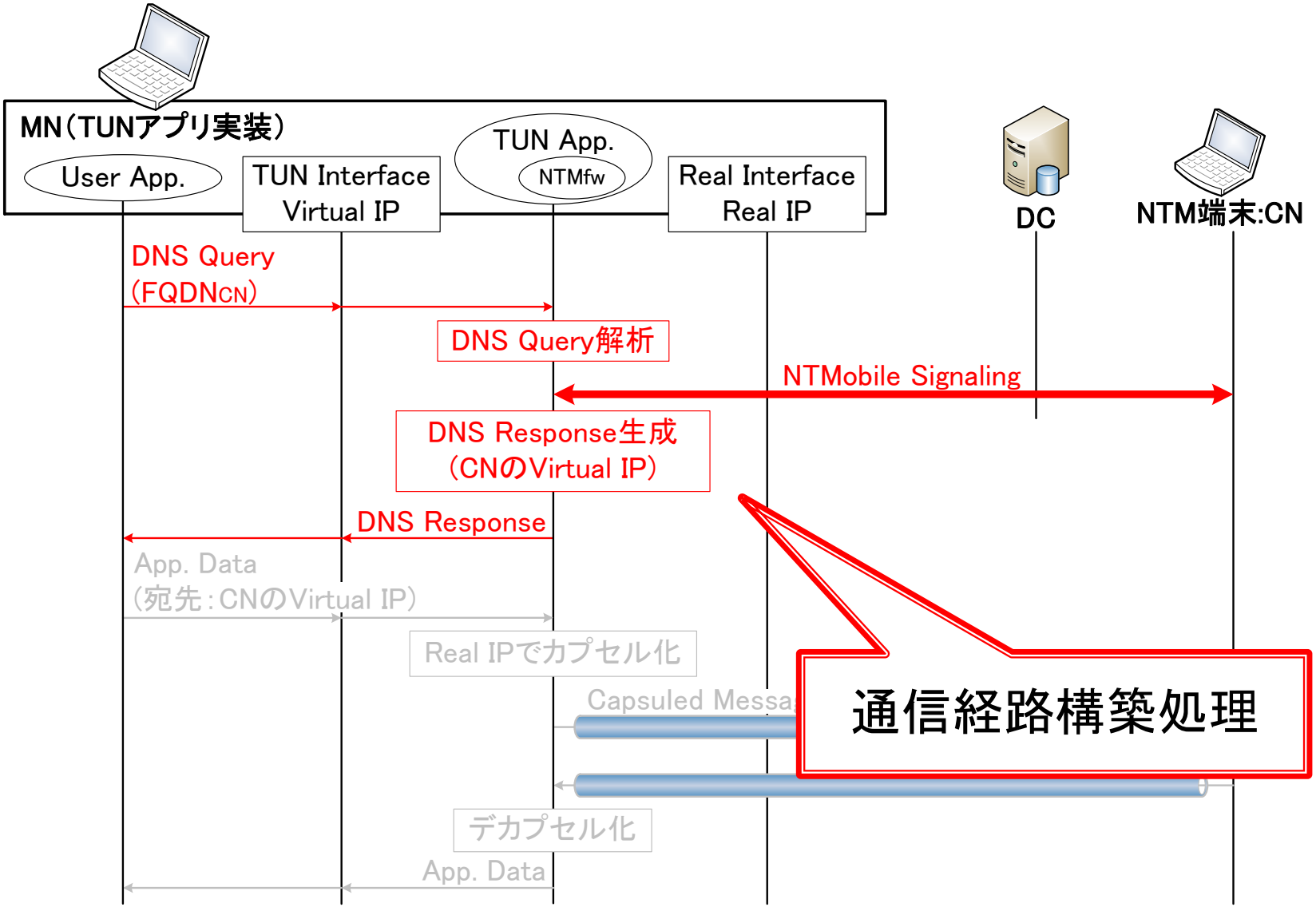
- TUNインタフェースに
  仮想IPアドレスを割り当て
- DNSクエリが
  TUNインタフェースを
  経由するよう
  ルーティングの設定



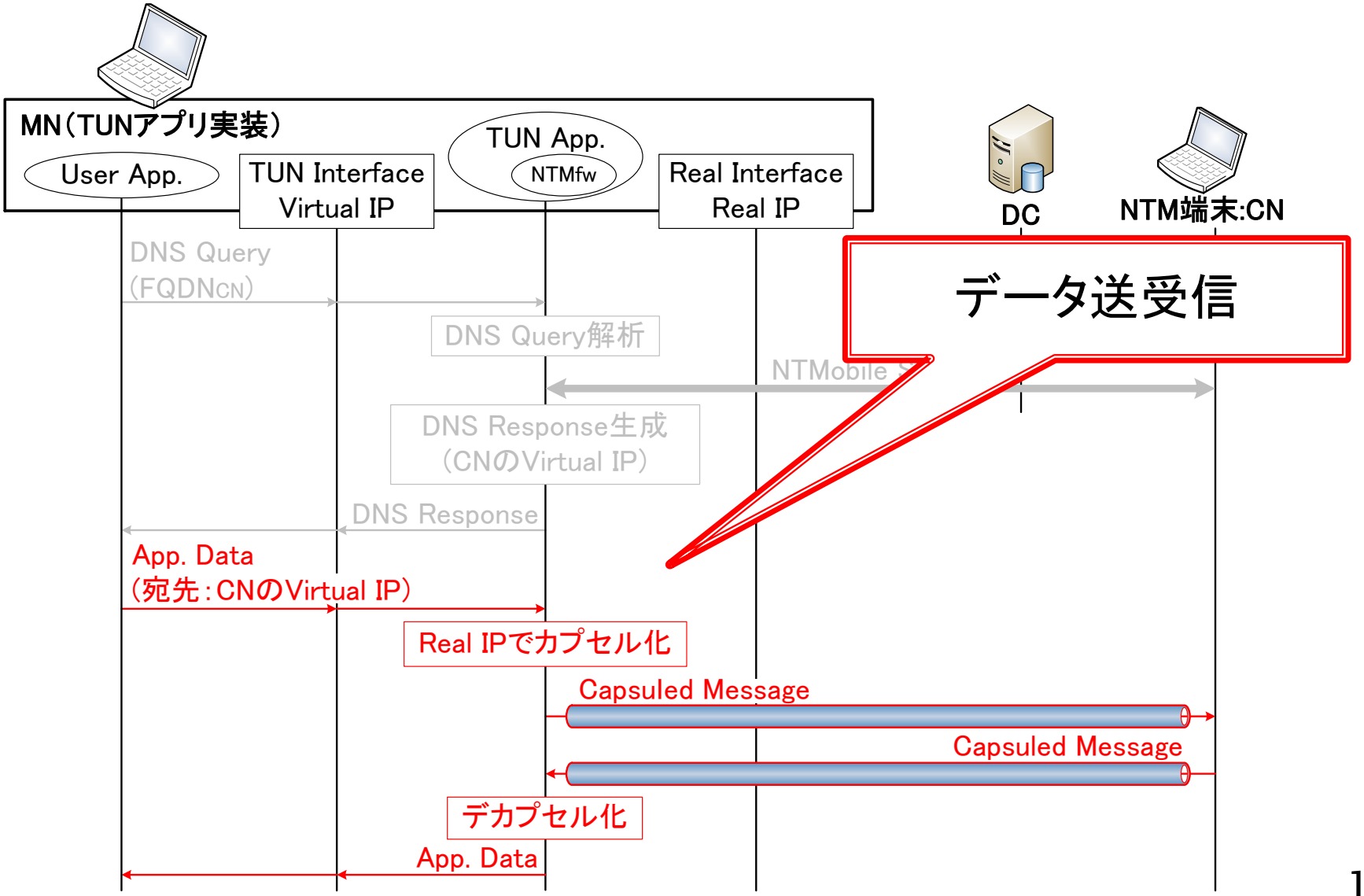
提案方式の動作シーケンス



提案方式の動作シーケンス

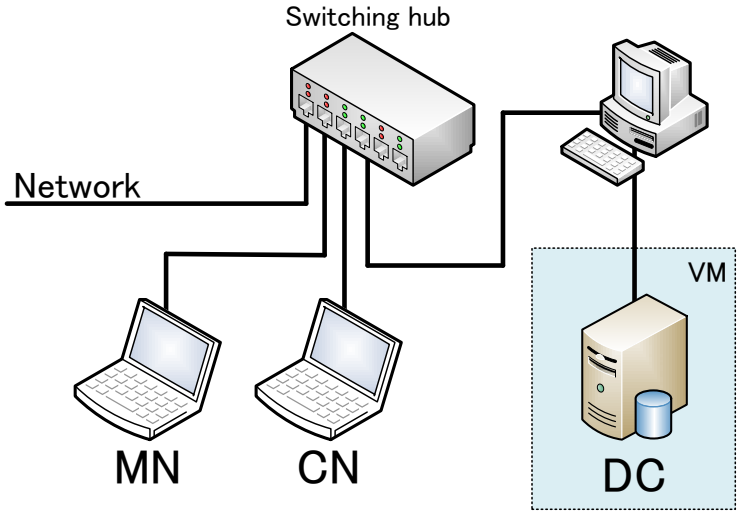


提案方式の動作シーケンス



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
  - 一般アプリケーションが送信するパケットを NTMobile通信に変換できることを確認
- 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



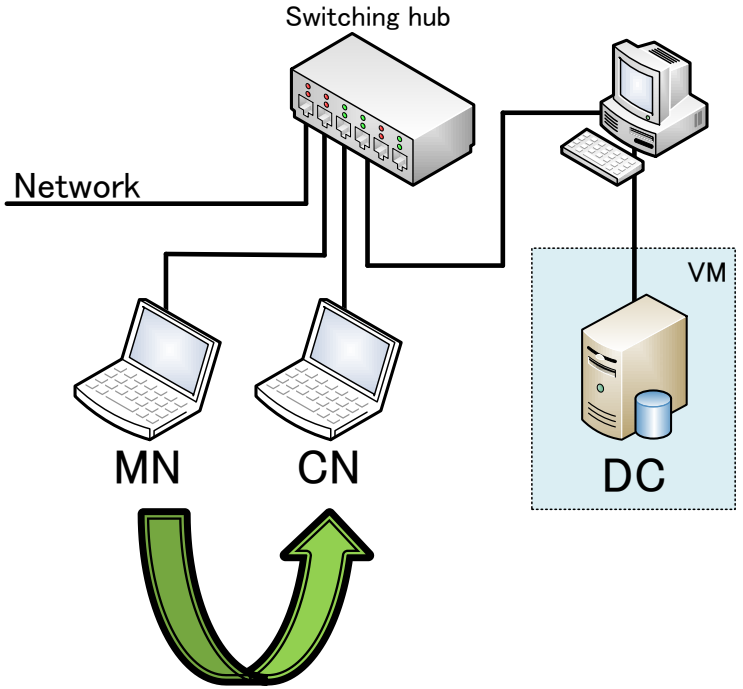
MN, CNの仕様	
OS	Ubuntu 14.04 32bit
CPU	Intel Core i5-2520M 2.50GHz
Memory	1944360 kB

評価 (1 / 3)

- 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

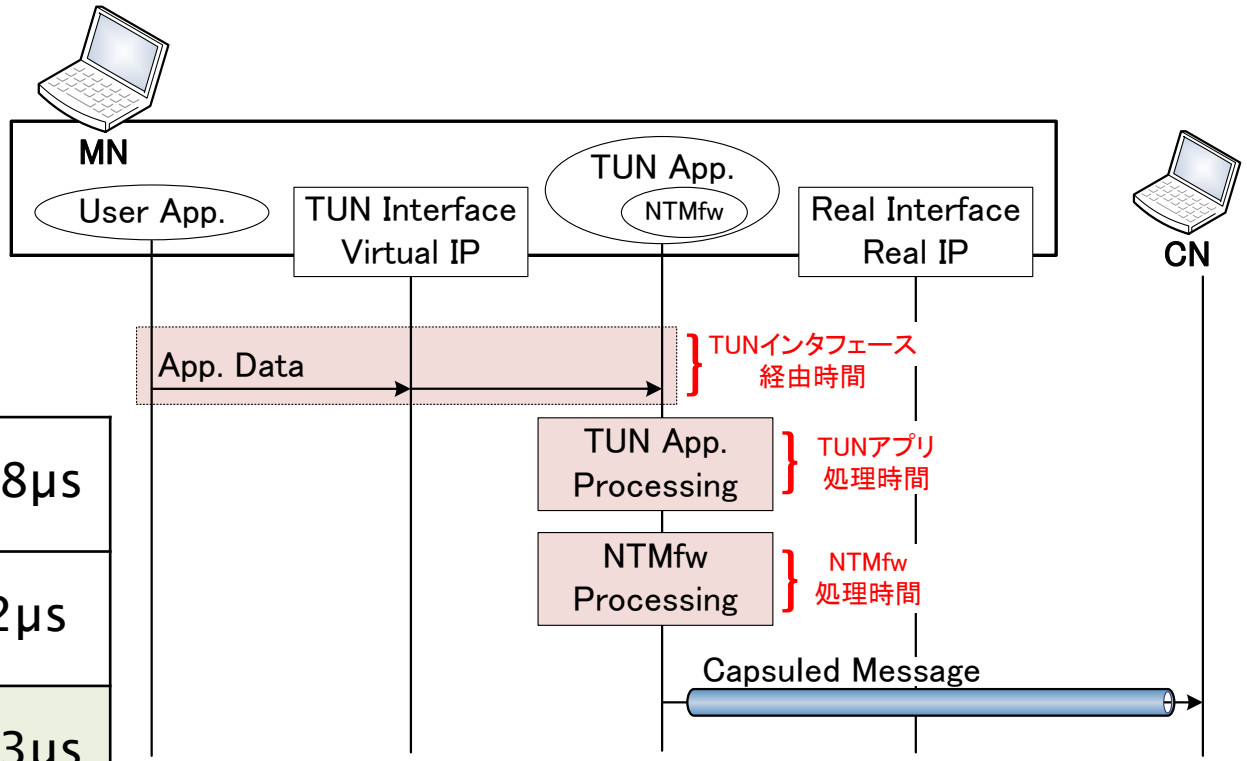
スループットの低下



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

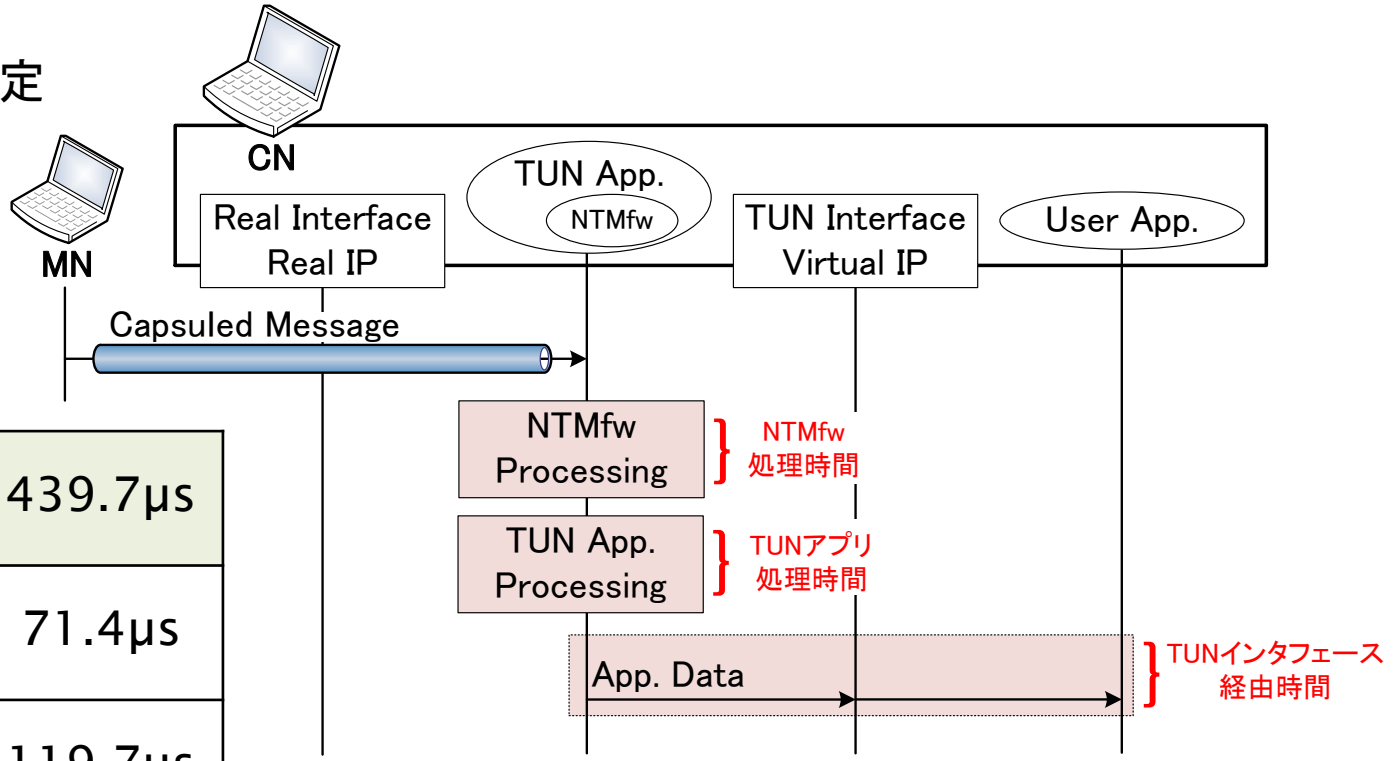


TUNインタフェース 経由時間	152.8μs
TUNアプリ 処理時間	69.2μs
NTMfw 処理時間	320.3μs

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

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

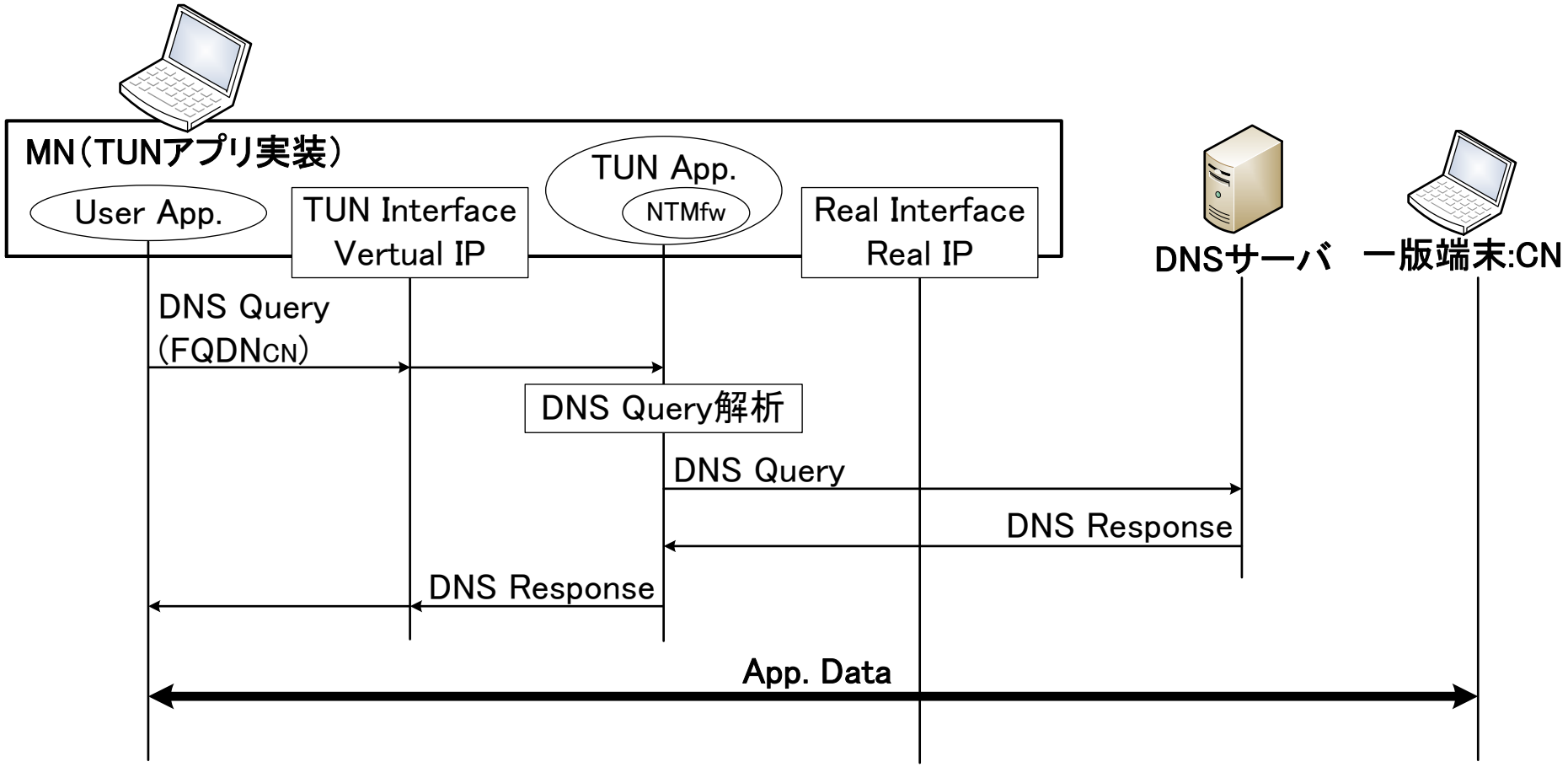
まとめ

- 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



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