

エンドノードを変更することなく IP ネットワークの制約を除去する通信システムの提案と実装

173426006 尾久 史弥

渡邊研究室

1. はじめに

IP ネットワークは通信インフラとして定着しているが、様々な制約があり、自由に通信ができないという課題がある。そこで、自由に通信を開始できる通信接続性とネットワークを切り替えても通信を継続できる移動透過性が求められている。通信接続性と移動透過性が実現できる技術として、DSMIPv6(Dual Stack Mobile IPv6)、HIP(Host Identity Protocol)、NTMobile(Network Traversal with Mobility)[1] が提案されている。しかし、DSMIPv6とHIPは、カーネル空間を改造することが前提であるため、普及がすすまないという課題がある。NTMobileは、通信ライブラリである NTMfw(NTMobile framework library)[2] を用いて新規にアプリケーションを開発することで、カーネル空間を改造しないで利用できるが、既存のアプリケーションを利用する場合は、ソケット API を書き換える必要がある。そこで本稿では、NTMobile の機能をアプリケーションとして実現し、デバイスにインストールするだけで既存のアプリケーションをそのまま利用できる TUN 利用型 NTMobile を提案する。

2. NTMobile

2.1 NTMobile の概要

NTMobile は、NTMobile の機能を持つ NTM 端末、実 IP アドレスと仮想 IP アドレスの管理、および通信経路を指示する DC(Direction Coordinator)、直接通信が行えない場合にパケットの中継を行う RS(Relay Server) により構成される。

NTMobile では DC が NTM 端末に対して、位置に依存しない仮想 IP アドレスを割り当て、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。アプリケーションによって生成された仮想 IP アドレスに基づくパケットは、NTMobile の機能により実 IP アドレスでカプセル化され、通信相手に送信される。通信中に端末がネットワークを切り替えると実 IP アドレスが変化するが、仮想 IP アドレスは変化しないので通信を継続できる。DC は DNS サーバの機能を包含し、通信相手の NTM 端末の名前解決を行うとともに、NTM 端末に対して最適な通信経路の指示を行う。NTM 端末は DC に対して、定期的に Keep Alive を行っており、DC からの通信経路の指示をいつでも受信することができる。DC が NTM 端末に対して適切な経路を指示することにより、グローバル空間からプライベート空間に向けて通信を開始することができる。端末同士が直接通信ができない場合は、RS を経由した経路を DC が指示する。

2.2 NTMfw ライブラリ

NTMfw は、NTMobile の機能を実現する通信ライブラリである。BSD(Berkeley Software Distribution) ソケット API と互換性のある NTMobile 用の NTM ソケット API を提供しており、ユーザは、NTMfw を一般通信ライブラリのように利用してプログラムを記述することができる。新規に開発するアプリケーションは、NTMfw を利用して、

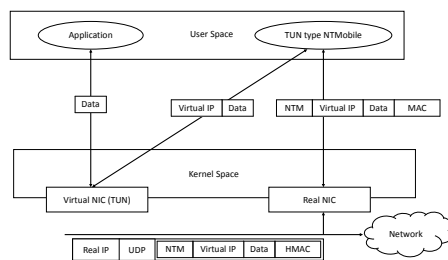


図 1: TUN 利用型 NTMobile のカプセル化の様子

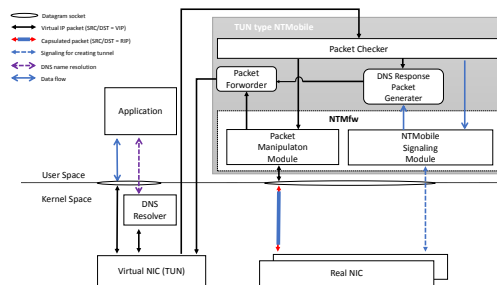


図 2: TUN 利用型 NTMobile のモジュール構成

アプリケーションを実装する。しかし、既存のアプリケーションは、プログラムの通信処理部を BSD ソケット API から NTM ソケット API に書き換える必要がある。

2.3 NTMfw を利用する時の課題

アプリケーションを新規開発するには、仕様の検討や実装など、それなりの時間を要する。また、既存のアプリケーションを修正する場合、一般のユーザがソケット API を書き換えるのは困難である。そもそも、プログラムのソースコードが公開されていない場合は、アプリケーションを改造できない。このように、NTMfw を利用する方法は、開発者への負担が大きく、広く普及することが困難になる可能性がある。

3. 提案方式

端末に提案方式をインストールすることで、既存のアプリケーションをそのまま利用できる TUN 利用型 NTMobile を提案する。

3.1 概要

TUN は多くの OS で提供されている機能で、IP パケットをカーネル空間からユーザ空間へフックすることができ、IP パケットの操作とカプセル化を実現できるインタフェースである。TUN インタフェースに仮想 IP アドレスを割り当てることで、アプリケーションが送信する仮想 IP アドレス宛のパケットは、全て TUN インタフェースにルーティングされる。また、NTMobile では名前解決要求をシグナリング処理のトリガとするため、DNS パケットも TUN イ

インターフェースにルーティングするように設定を行う。図1にTUN利用型NTMobileにおけるカプセル化の様子を示す。アプリケーションが送信したデータはカーネル内部に処理が渡され仮想IPパケットが生成される。TUN利用型NTMobileは、生成された仮想IPパケットをTUNインターフェースから読み込み、NTMobileの機能によりNTMヘッダおよびMAC(Message Authentication Code)ヘッダを付与する。その後、実インターフェースに向けて送信することにより、カーネル内部でUDPによるカプセル化が行われる。

3.2 モジュール構成

TUN利用型NTMobileのモジュール構成を図2に示す。アプリケーションからTUN利用型NTMobileに送信されるパケットは、通信用/DNSパケットの2通りであり、パケットチェッカーにより処理が分岐される。DNSパケットの場合は、NTMfwのシグナリングモジュールによりシグナリング処理が行われ、処理が成功すると通信相手の仮想IPアドレスを取得できるので、その仮想IPアドレスを用いてDNSレスポンスパケット生成機能にてDNS応答パケットを生成する。その後、パケットフォワーダーを経由してアプリケーションに対して名前解決要求の応答を行う。通信用パケットの場合は、NTMfwのパケット操作機能に処理が渡され、NTMobileによるヘッダ付与および暗号化等が行われ、実インターフェースに処理が渡される。

3.3 TUN利用型NTMobileの動作

図3にアプリケーションがTUN利用型NTMobileを利用してNTMobile通信を利用する際の動作シーケンスを示す。TUN利用型NTMobileを利用して通信を開始する場合、アプリケーションはNTM端末のFQDNを指定してDNSクエリパケットを送信する。TUN利用型NTMobileは、DNSクエリパケットを受信してFQDNを解析する。解析した結果、NTMobile端末のFQDNである場合は、NTMobileの機能により通信相手とNTMobileシグナリングを実行して、トンネル経路を構築するとともに、通信相手の仮想IPアドレスを取得する。NTMobileシグナリング終了後にTUN利用型NTMobileは、取得した仮想IPアドレスとアプリケーションが送信したDNSクエリパケットからDNSレスポンスパケットを生成し、アプリケーションに渡す。アプリケーションは、通信相手を仮想IPアドレスとみなすため、以降の通信は全て仮想IPアドレス宛にパケットが送信される。TUN利用型NTMobileは、TUNインターフェースから仮想IPパケットを読み込み、NTMobileの機能によりNTMヘッダおよびMACヘッダを付与して実インターフェースに処理を渡す。実インターフェースでは、UDPでカプセル化して通信相手に送信する。通信相手から応答がある場合は、NTMobileの機能によりデカプセル化して仮想IPパケットを取得する。また、取得した仮想IPパケットをTUNインターフェースに書き込むことにより、アプリケーションにデータを渡す。

3.4 IPフラグメンテーションの課題解決

ネットワークには、1回の転送ごとに送信できるデータの最大長であるMTU(Maximum Transmission Unit)が決められており、パケット長がMTUを超えるとパケットのフラグメンテーションが発生し、通信効率が低下する。NTMobileのようにパケットをカプセル化するプロトコルは、プロトコルが付与するヘッダ長によりMTUを超えるという課題がある。

TUN利用型NTMobileでは、TUNインターフェースに適切なMTUを設定することにより、フラグメンテーションが発生しないようにする。

3.5 一般通信とNTMobile通信の併用

アプリケーションが一般通信またはNTMobile通信を利用するかを選択できる。これは、通信開始時の名前解決処理を、TUN利用型NTMobileと既設のDNSサーバーで分担することで実現できる。NTMobile通信を利用する場合は、TUN利用型NTMobileが名前解決処理とNTMobileのシグナリング処理を行い、NTMobile通信を開始する。一般通信を利用する場合は、既設のDNSサーバーが名前解決処理を行い、一般通信を開始する。

表 1: スループットの計測結果

区分	時間 (Mbps/s)
TUN利用型NTMobile経由	73.0
一般通信	93.8

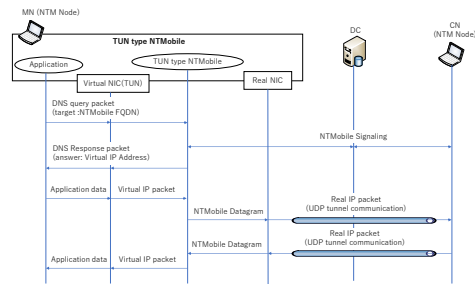


図 3: NTMobile 通信開始時の動作シーケンス

4. 実装と評価

提案方式の全ての機能をLinux上にC言語で実装し、IPv4ネットワークにおいて動作検証を行った。動作検証の結果、デバイスに提案方式をインストールするだけで、一般通信とNTMobile通信の併用が可能になることを確認した。

スループットの測定により提案方式の性能評価を行った。計測結果を表1に示す。提案手法を利用することにより、一般通信と比べてスループットが22%低下することがわかった。提案方式の処理の中にはNTMobileによる暗号化/復号処理が含まれており、スループットに影響を与えていると考えられる。暗号化/復号処理を行っていても、実用的な範囲で動作することが分かった。

5. まとめ

本稿では、NTMobileの機能をユーザ空間のアプリケーションとして実現する方式を提案し、実装と評価を行った。動作検証により、アプリケーションに対して一切の変更をすることなく、NTMobileの機能を利用できることを確認した。また評価により、十分な性能で動作することを確認した。今後は、IPv6に関連する動作検証を進め、全てのネットワーク環境において動作を確認する予定である。

参考文献

- [1] 上野尾一真, 鈴木秀和, 内藤克浩, 渡邊 晃: IPv4/IPv6混在環境で移動透過性を実現するNTMobileの実装と評価, 情報処理学会論文誌, Vol. 54, No. 10, pp. 2288-2299 (2013).
- [2] 納堂博史, 鈴木秀和, 内藤克浩, 渡邊 晃: エンドツーエンド通信をアプリケーションレベルで可能にする通信ライブラリの実現と評価, 情報処理学会論文誌, (TBD).

エンドノードを変更することなく IPネットワークの制約を除去する 通信システムの提案と実装

名城大学大学院 理工学研究科 情報工学専攻

渡邊研究室 173426006 尾久史弥



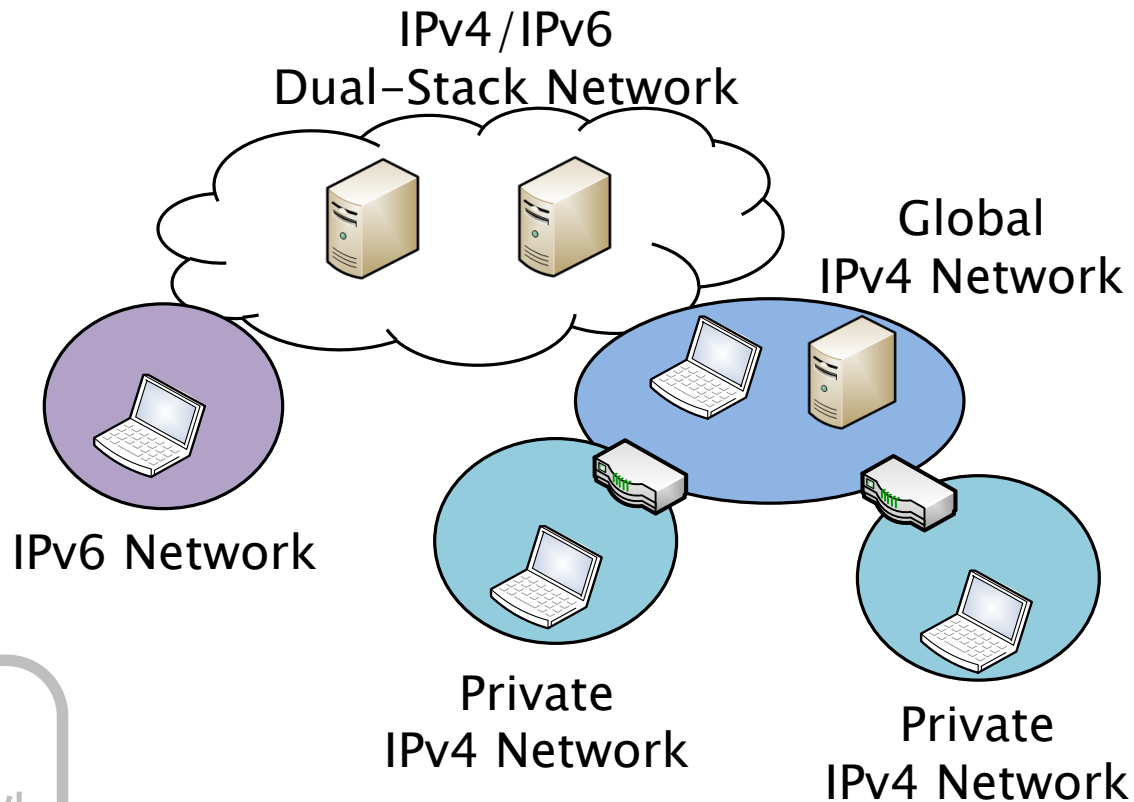
研究背景

■ IPv4グローバルアドレスの枯渇問題

- NATの導入
- IPv6アドレスの導入

複雑なネットワーク構成

- NAT越え問題
- IPv4/IPv6の非互換性
- 移動透過性の課題



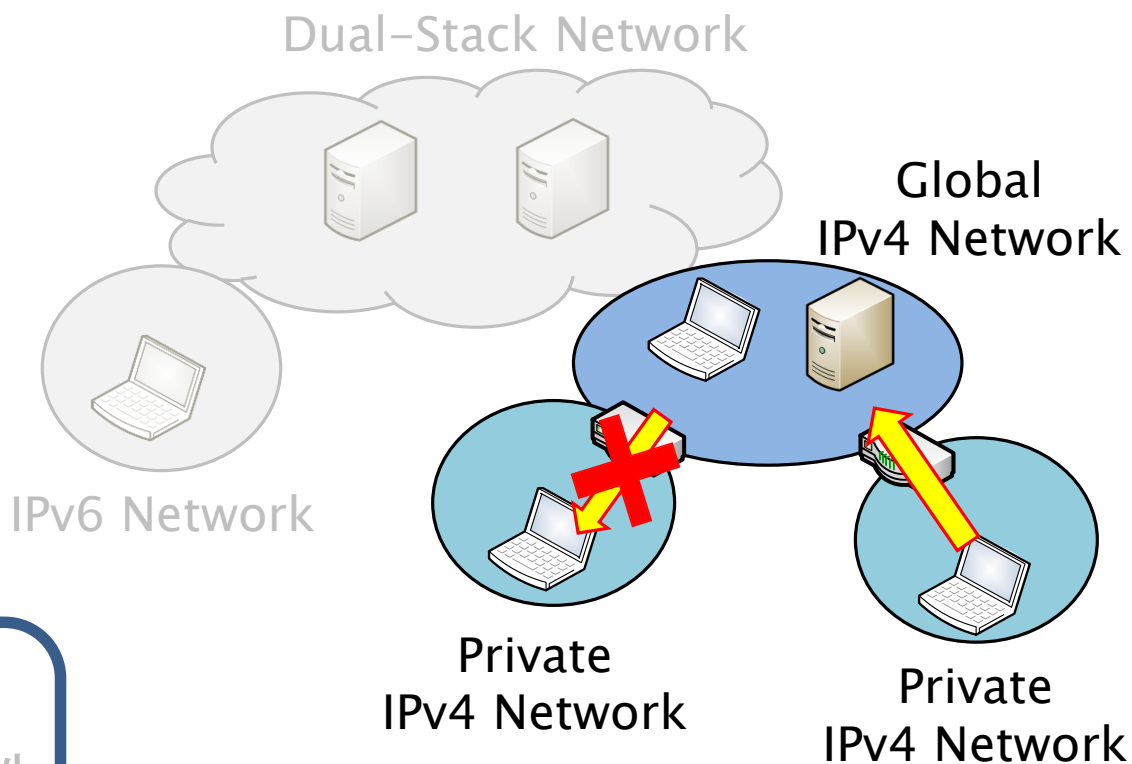
研究背景

■ IPv4グローバルアドレスの枯渇問題

- NATの導入
- IPv6アドレスの導入

複雑なネットワーク構成

- **NAT越え問題**
- IPv4/IPv6の非互換性
- 移動透過性の課題



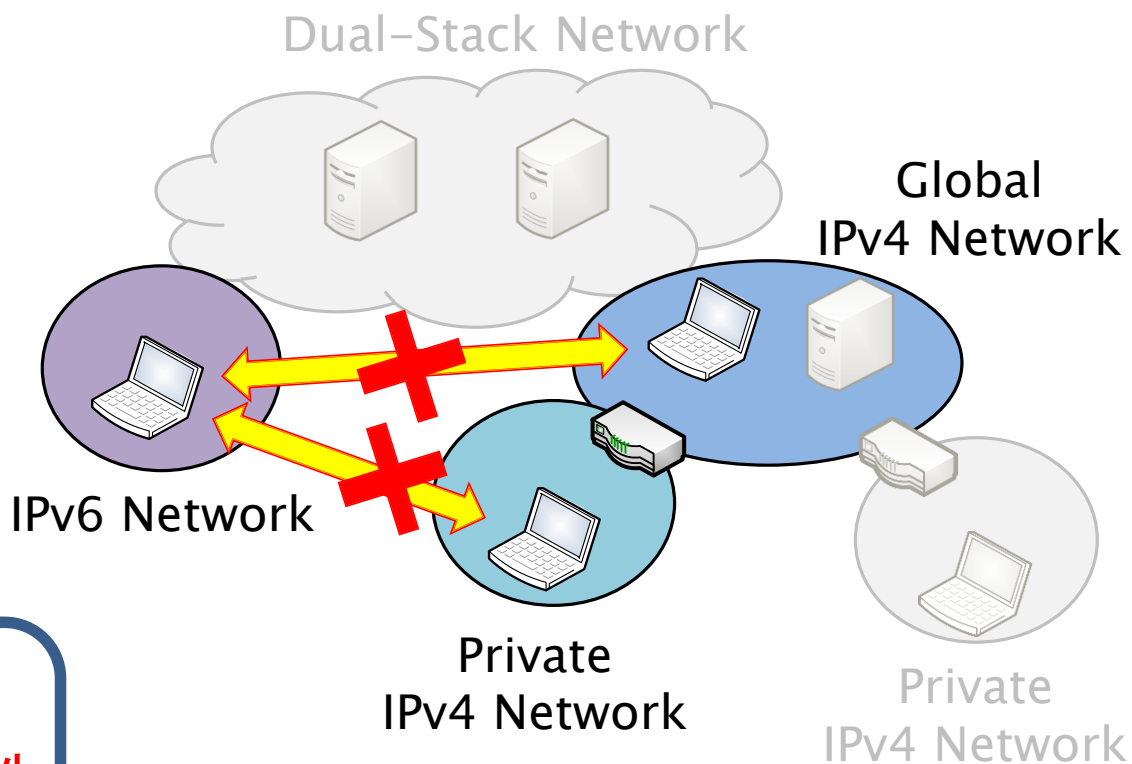
研究背景

■ IPv4グローバルアドレスの枯渇問題

- NATの導入
- IPv6アドレスの導入

複雑なネットワーク構成

- NAT越え問題
- **IPv4/IPv6の非互換性**
- 移動透過性の課題



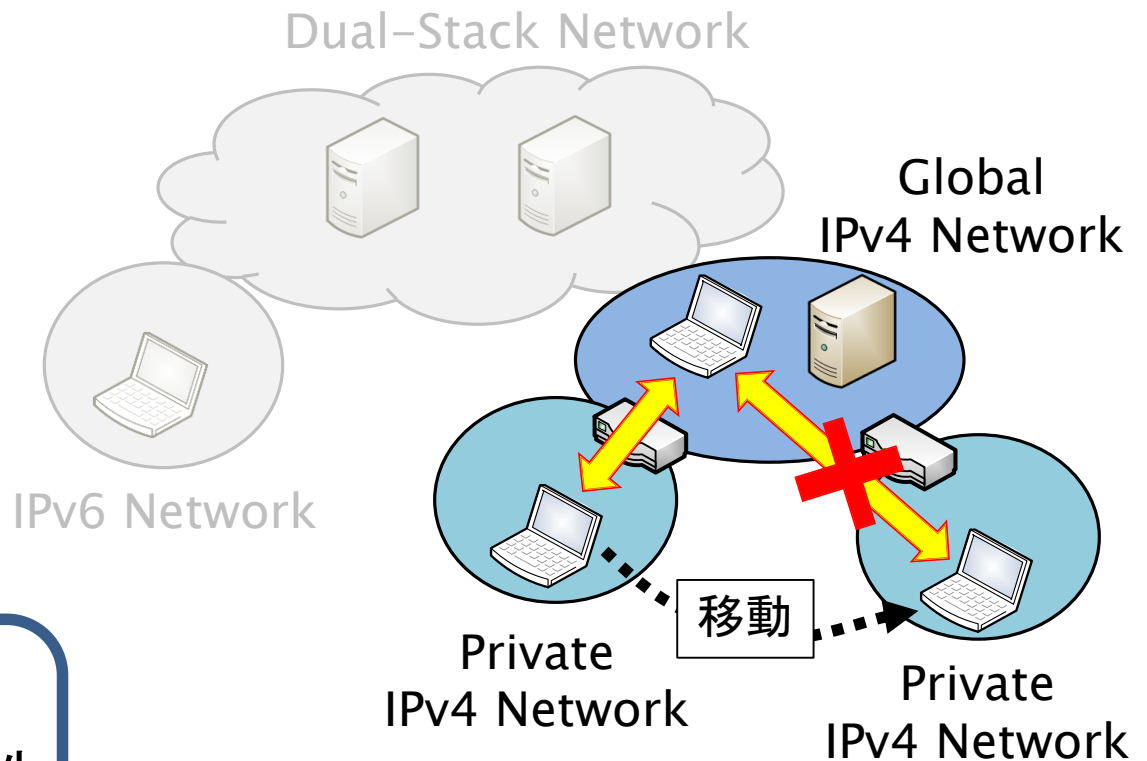
研究背景

■ IPv4グローバルアドレスの枯渇問題

- NATの導入
- IPv6アドレスの導入

複雑なネットワーク構成

- NAT越え問題
- IPv4/IPv6の非互換性
- **移動透過性の課題**



既存技術

- カーネル空間を改造する方式
 - DSMIPv6 (Dual Stack Mobile IPv6)[1]
 - HIP (Host Identity Protocol)[2]

- カーネル空間を改造しない方式
 - NTMobile (Network Traversal with Mobility) [3, 4, 5]

[1] Soliman, H.: Mobile IPv6 Support for Dual Stack Hosts and Routers, RFC 5555 (2009).

[2] Moskowitz, R., Heer, T., Jokela, P. and Henderson, T. R.: Host Identity Protocol Version 2 (HIPv2), RFC 7401 (2015).

[3] 内藤克浩, 上酔尾一真, 西尾拓也, 水谷智大, 鈴木秀和, 渡邊 晃, 森香津夫, 小林英雄
:NTMobile における移動透過性の実現と実装, 情報処理学会論文誌, Vol. 54, No. 10, pp. 367-3 (2013).

[4] 鈴木秀和, 上酔尾一真, 水谷智大, 西尾拓也, 内藤克浩, 渡邊 晃
: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. 1, pp. 367-379 (2013).

[5] 上酔尾一真, 鈴木秀和, 内藤克浩, 渡邊 晃:IPv4/IPv6 混在環境で移動透過性を実現するNTMobile の実装と評価,
情報処理学会論文誌, Vol. 54, No. 10, pp. 2288-2299 (2013).

既存技術

■ カーネル空間を改造する方式

- DSMIPv6 (Dual Stack Mobile IPv6)[1]
- HIP (Host Identity Protocol)[2]

■ カーネル空間を改造しない方式

- NTMobile (Network Traversal with Mobility) [3, 4, 5]

[1] Soliman, H.: Mobile IPv6 Support for Dual Stack Hosts and Routers, RFC 5555 (2009).

[2] Moskowitz, R., Heer, T., Jokela, P. and Henderson, T. R.: Host Identity Protocol Version 2 (HIPv2), RFC 7401 (2015).

[3] 内藤克浩, 上酔尾一真, 西尾拓也, 水谷智大, 鈴木秀和, 渡邊 晃, 森香津夫, 小林英雄
:NTMobile における移動透過性の実現と実装, 情報処理学会論文誌, Vol. 54, No. 10, pp. 367-3 (2013).

[4] 鈴木秀和, 上酔尾一真, 水谷智大, 西尾拓也, 内藤克浩, 渡邊 晃
: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. 1, pp. 367-379 (2013).

[5] 上酔尾一真, 鈴木秀和, 内藤克浩, 渡邊 晃:IPv4/IPv6 混在環境で移動透過性を実現するNTMobile の実装と評価,
情報処理学会論文誌, Vol. 54, No. 10, pp. 2288-2299 (2013).

カーネルを改造する方式の課題

カーネルに実装することが前提の仕様

DSMIPv6はネットワーク層を改造して実装

HIPはネットワーク層とトランスポート層の間に実装

アプリケーションレベルで実現することができない

- ❌ カーネルの改造は、一般ユーザには困難
- ❌ スマートフォンに実装されていない
- ❌ カーネルを改造すると、メーカーのサポートを受けられない
- ❌ カーネルの頻繁なバージョン更新に追従ためのコストが高い

既存技術

■ カーネル空間を改造する方式

- DSMIPv6 (Dual Stack Mobile IPv6)[1]
- HIP (Host Identity Protocol)[2]

■ カーネル空間を改造しない方式

- **NTMobile (Network Traversal with Mobility) [3, 4, 5]**

[1] Soliman, H.: Mobile IPv6 Support for Dual Stack Hosts and Routers, RFC 5555 (2009).

[2] Moskowitz, R., Heer, T., Jokela, P. and Henderson, T. R.: Host Identity Protocol Version 2 (HIPv2), RFC 7401 (2015).

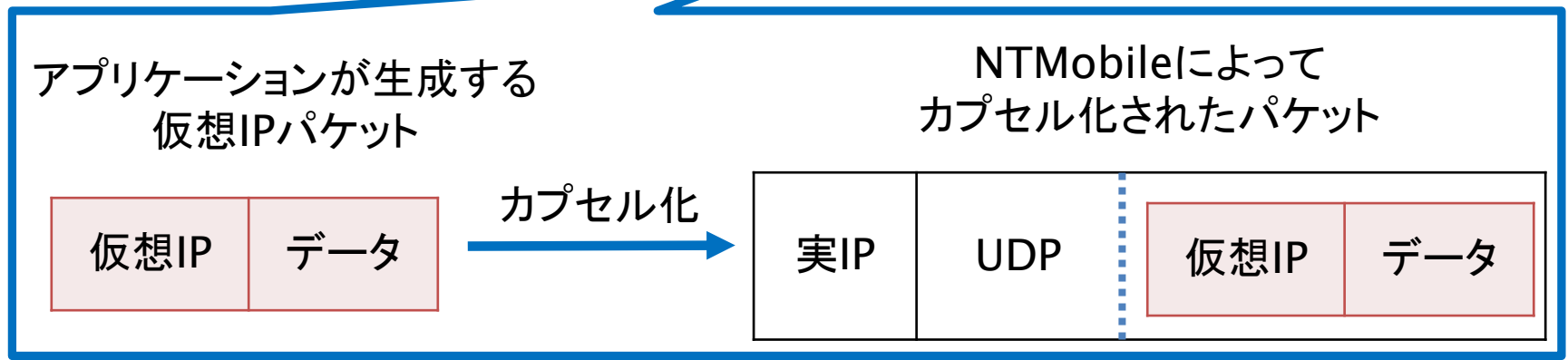
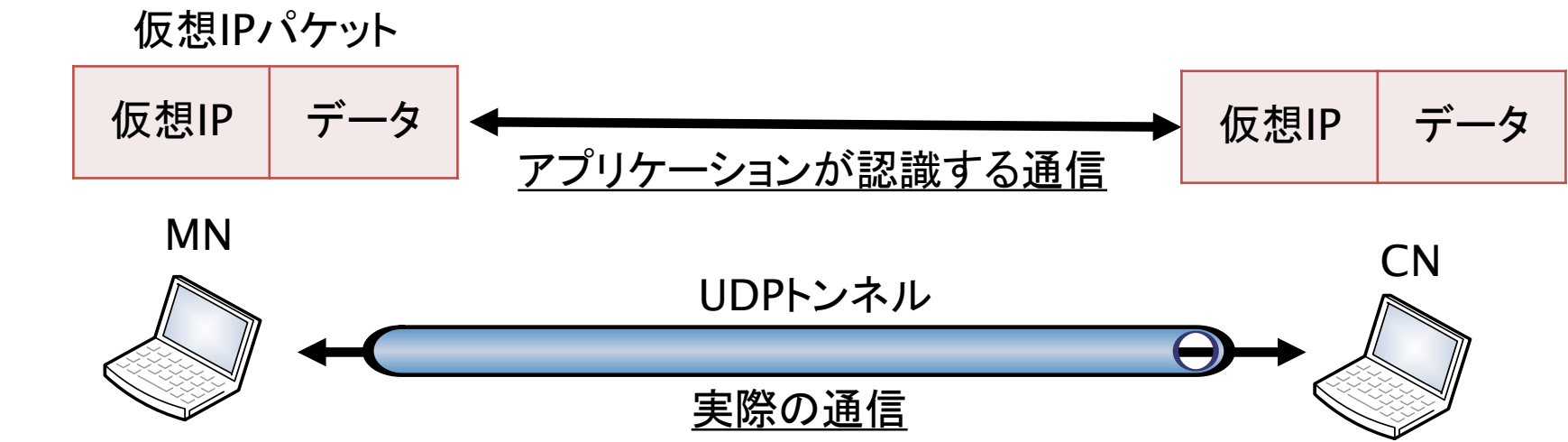
[3] 内藤克浩, 上酔尾一真, 西尾拓也, 水谷智大, 鈴木秀和, 渡邊 晃, 森香津夫, 小林英雄
:NTMobile における移動透過性の実現と実装, 情報処理学会論文誌, Vol. 54, No. 10, pp. 367-3 (2013).

[4] 鈴木秀和, 上酔尾一真, 水谷智大, 西尾拓也, 内藤克浩, 渡邊 晃
: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. 1, pp. 367-379 (2013).

[5] 上酔尾一真, 鈴木秀和, 内藤克浩, 渡邊 晃:IPv4/IPv6 混在環境で移動透過性を実現するNTMobile の実装と評価,
情報処理学会論文誌, Vol. 54, No. 10, pp. 2288-2299 (2013).

NTMobileのUDPTトンネル通信

- アプリケーションは位置に依存しない仮想IPアドレスに基づいて行う
 - 端末が移動しても仮想IPアドレスは変化しない

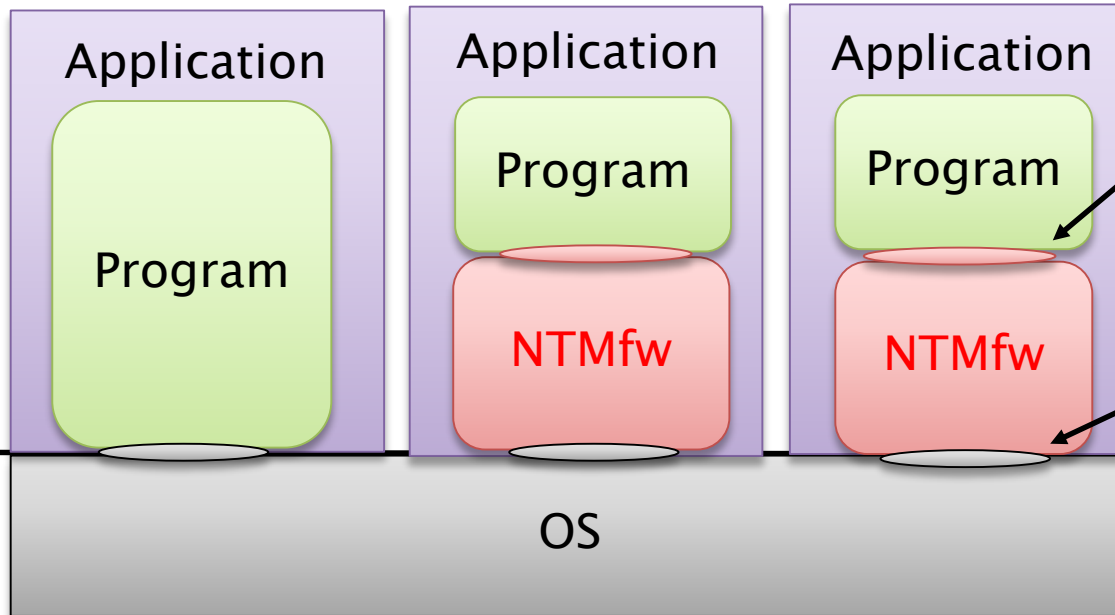


NTMobileの利用方法

- **NTMfw(NTMobile framework library)**を利用してアプリケーションを実装
 - 一般のソケットAPIと互換性のあるNTMソケットAPIを提供
 - NTMobileを利用したいアプリケーションに対して実装

一般通信のみの
アプリケーション

NTMobileを利用したい
アプリケーション



NTMソケットAPI

`ntmfw_getaddrinfo()`
`ntmfw_sendto()`

BSDソケットAPI

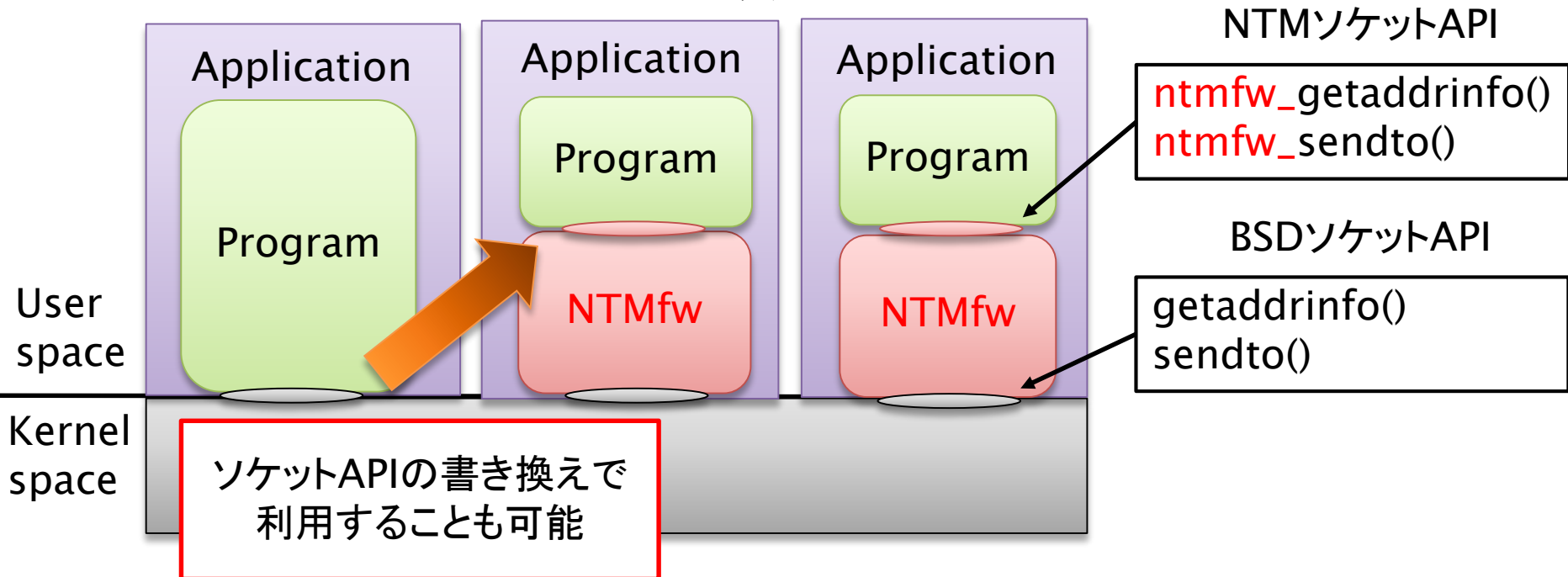
`getaddrinfo()`
`sendto()`

NTMobileの利用方法

- **NTMfw(NTMobile framework library)**を利用してアプリケーションを実装
 - 一般のソケットAPIと互換性のあるNTMソケットAPIを提供
 - NTMobileを利用したいアプリケーションに対して実装

一般通信のみの
アプリケーション

NTMobileを利用したい
アプリケーション



NTMfwを利用する場合の課題

アプリケーションを新規に開発する場合

- ✖ 仕様の検討や実装などに時間が必要

アプリケーションを修正して開発する場合

- ✖ 一般通信/NTMobile通信が混在する場合、ソケットAPIの一括書き換えでは、仕様を満たすことができない
- ✖ 一般的なユーザがプログラムを書き換えることが困難

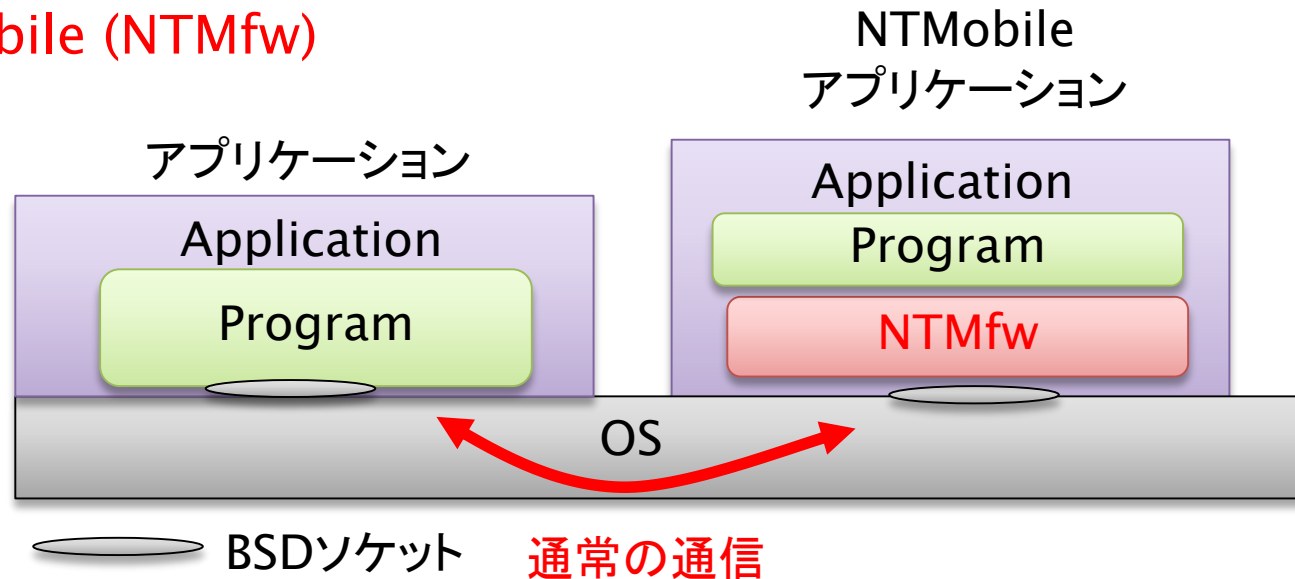
開発者への負担が大きい

ソースコードが公開されていない場合は利用できない

研究目的

前提条件

- ✓ ユーザ空間への実装
- ✓ アプリケーションを改造しない
- ✓ NTMobile (NTMfw)



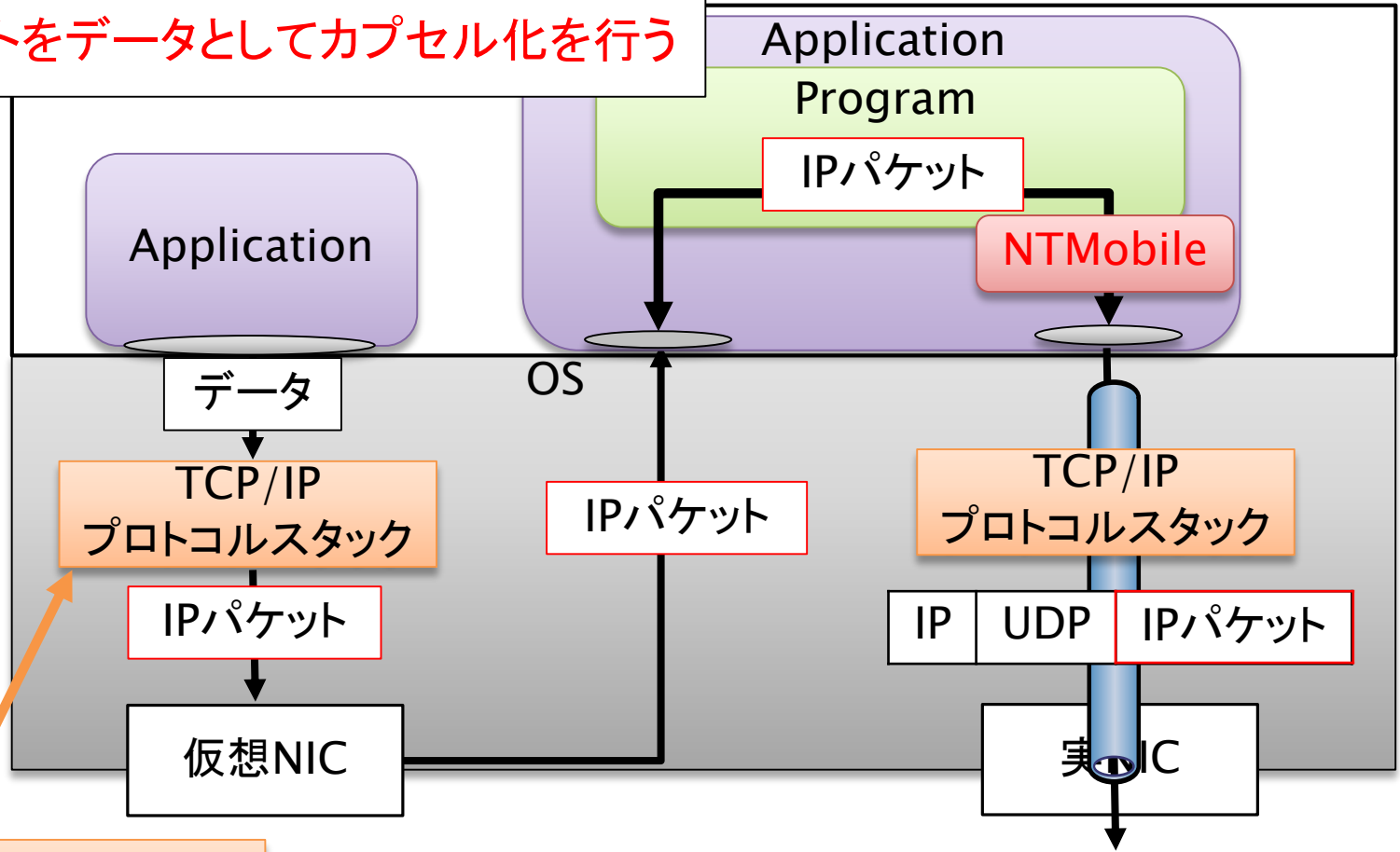
研究目的

NTMfwを利用したアプリケーションを実現し、
アプリケーションを変更することなくNTMobileの機能を
提供することを提案

アプリケーションを改造しない方式の検討

検討事項

- ✓ IPパケットをユーザ空間に取り込む
- ✓ IPパケットをデータとしてカプセル化を行う



アプリケーションが送信したデータからIPパケットを生成

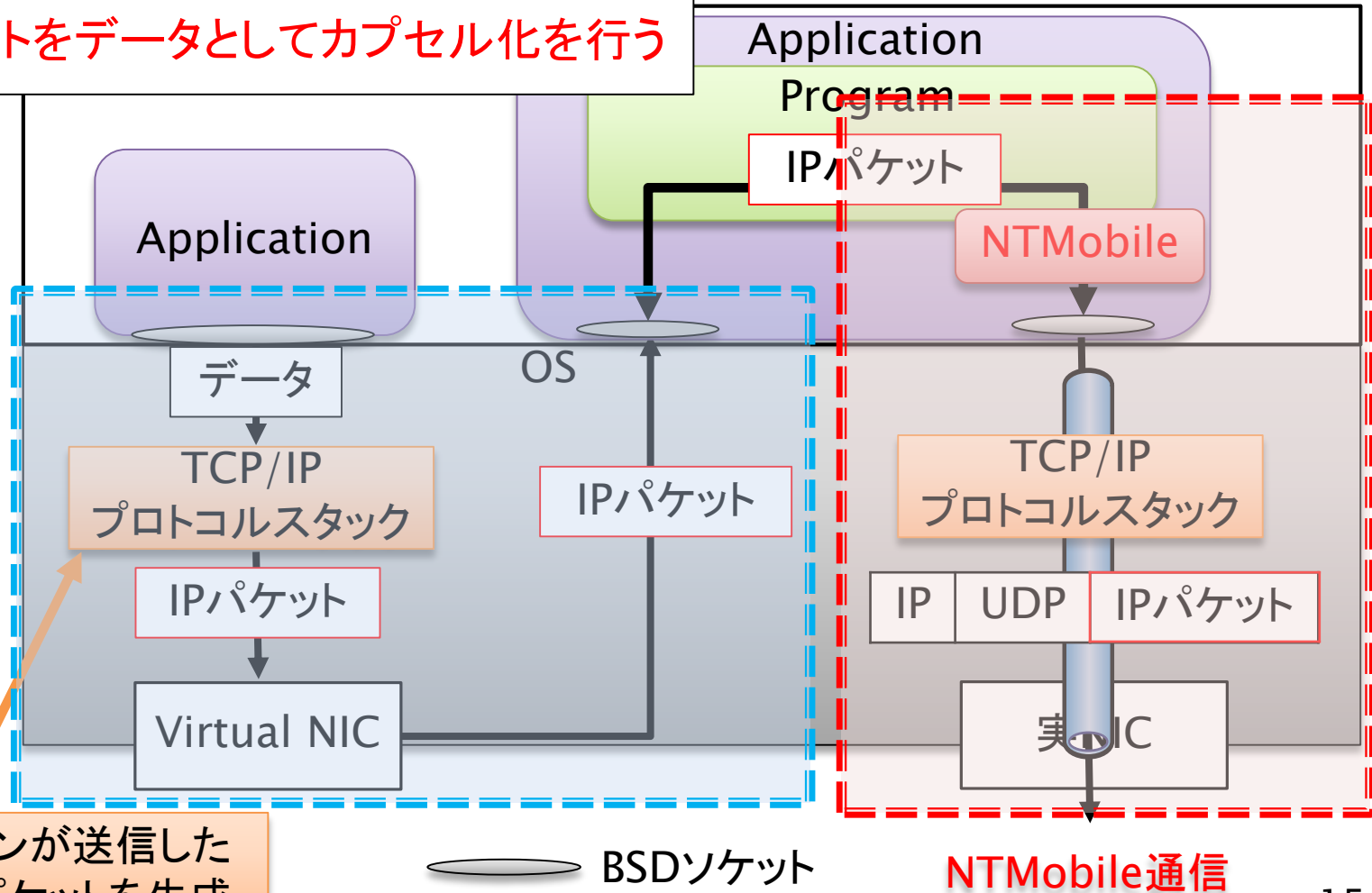
○ BSDソケット

NTMobile通信

アプリケーションを改造しない方式の検討

検討事項

- ✓ IPパケットをユーザ空間に取り込む
- ✓ IPパケットをデータとしてカプセル化を行う

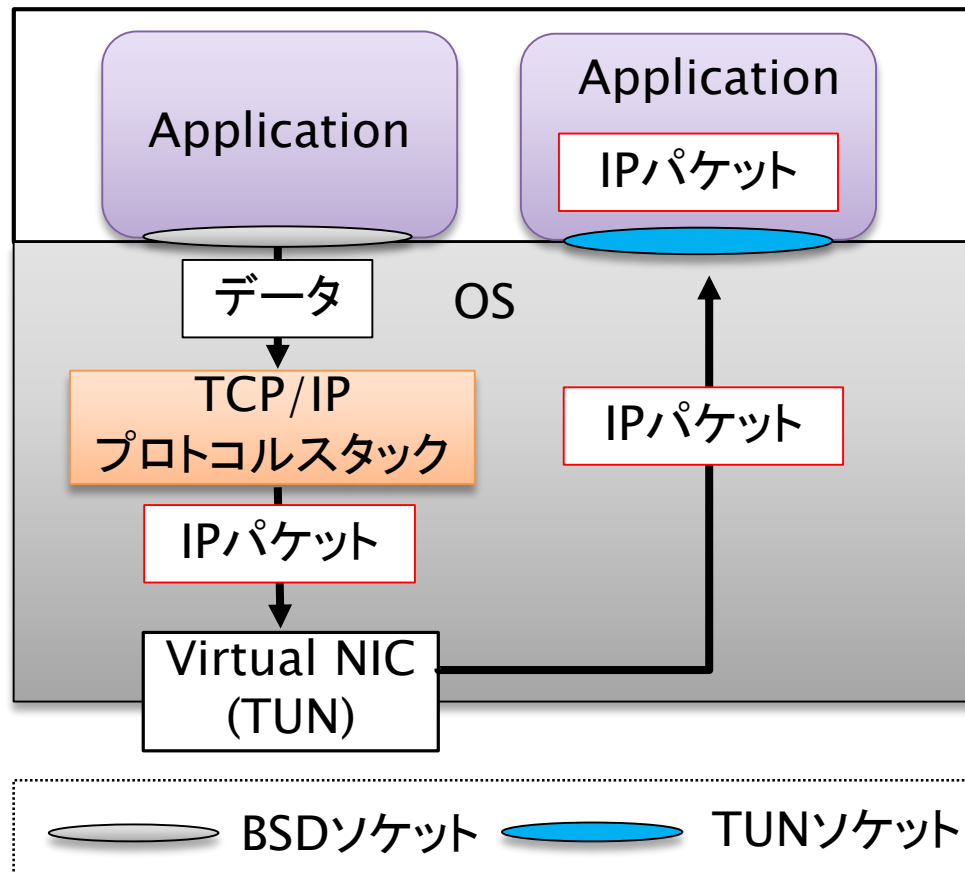


アプリケーションが送信したデータからIPパケットを生成

NTMobile通信

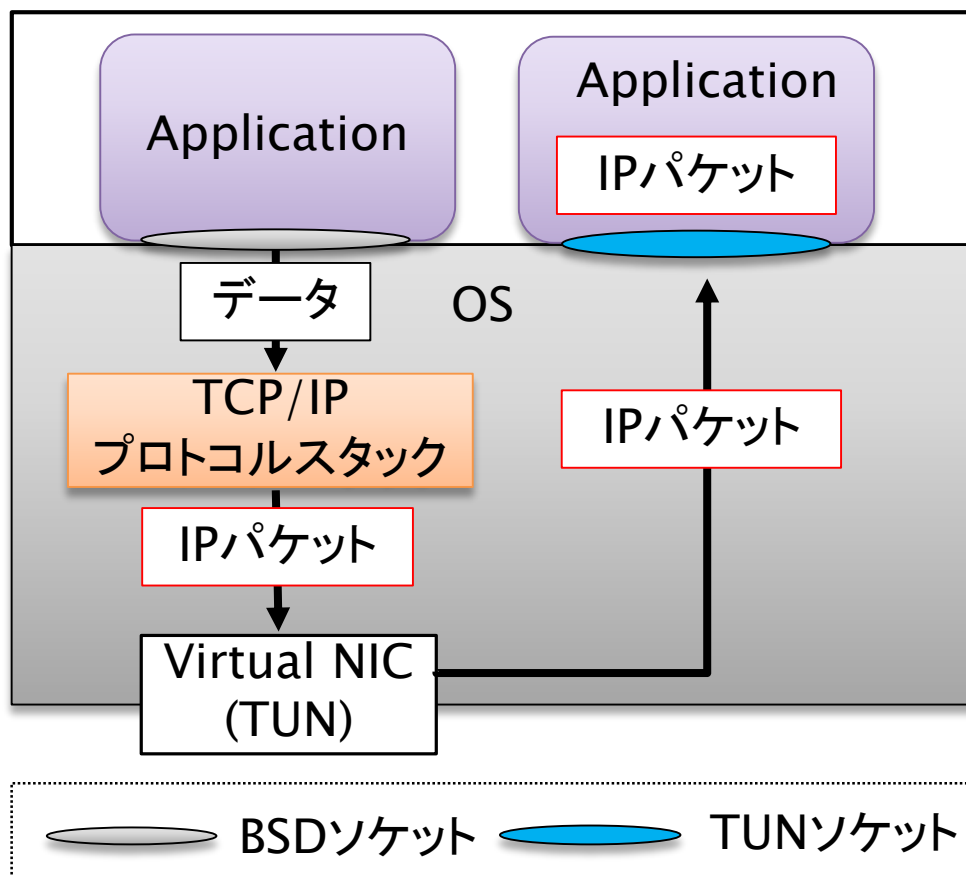
TUN(TUNnel)

- ユーザ空間にIPパケットを取り込めるサービス
 - TUNインタフェースを定義することで利用可能
- 実際のインタフェース同様にIPアドレスなどを設定して用いる



TUN(TUNnel)

- ユーザ空間にIPパケットを取り込めるサービス
 - TUNインタフェースを定義することで利用可能
- 実際のインタフェース同様にIPアドレスなどを設定して用いる



アプリケーションによるTUNの利用 (1/4)

- TUNに仮想IPアドレスを割り当てる

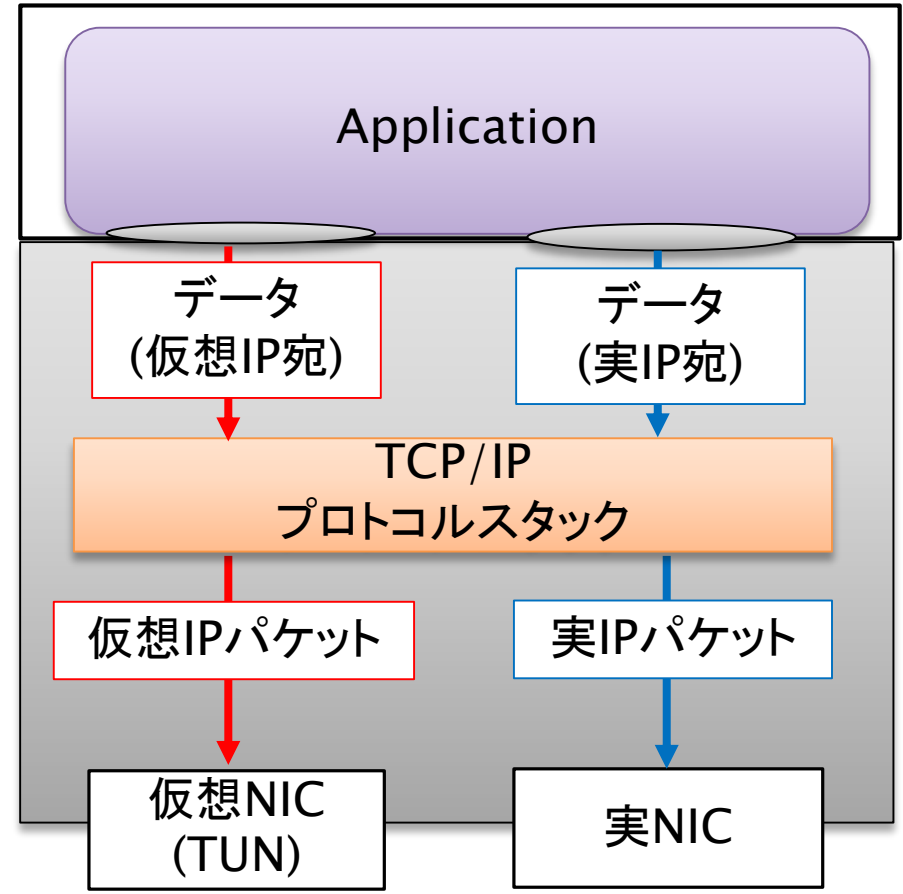
仮想IPパケット → TUN
実IPパケット → 実NIC

- NTMobile通信を行いたい場合

- 仮想IPアドレスで通信
- TUNを経由のNTMobile通信

- 一般通信を行いたい場合

- 実IPアドレスで通信
- 従来通り一般通信を行える



仮想IPアドレス

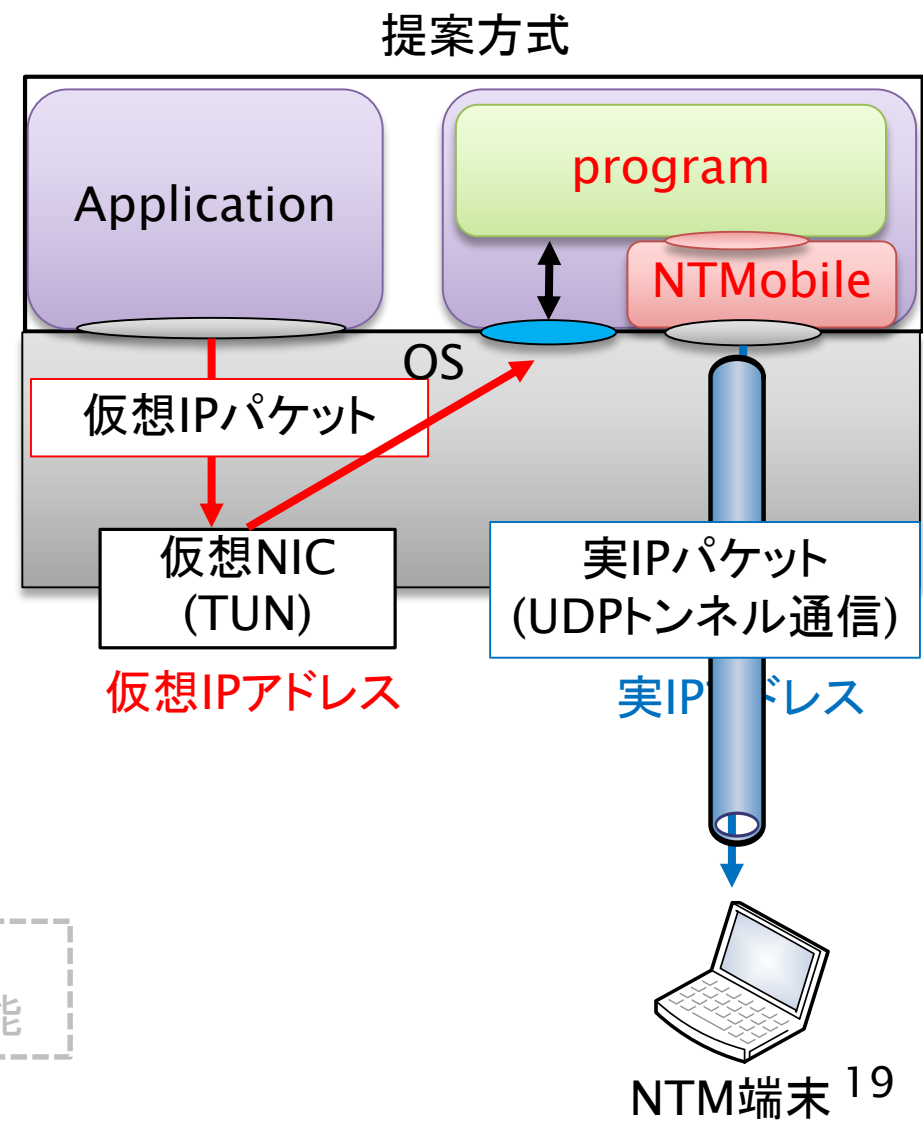
実IPアドレス

通信相手の宛先によって
NTMobile/一般通信を利用するか選択可能

アプリケーションによるTUNの利用 (1/4)

- TUNに仮想IPアドレスを割り当てる
 - 仮想IPパケット → TUN
 - 実IPパケット → 実NIC
- NTMobile通信を行いたい場合
 - 仮想IPアドレスで通信
 - TUNを経由のNTMobile通信
- 一般通信を行いたい場合
 - 実IPアドレスで通信
 - 従来通り一般通信を行える

通信相手の宛先によって
NTMobile/一般通信を利用するか選択可能



アプリケーションによるTUNの利用 (1/4)

- TUNに仮想IPアドレスを割り当てる

仮想IPパケット → TUN
実IPパケット → 実NIC

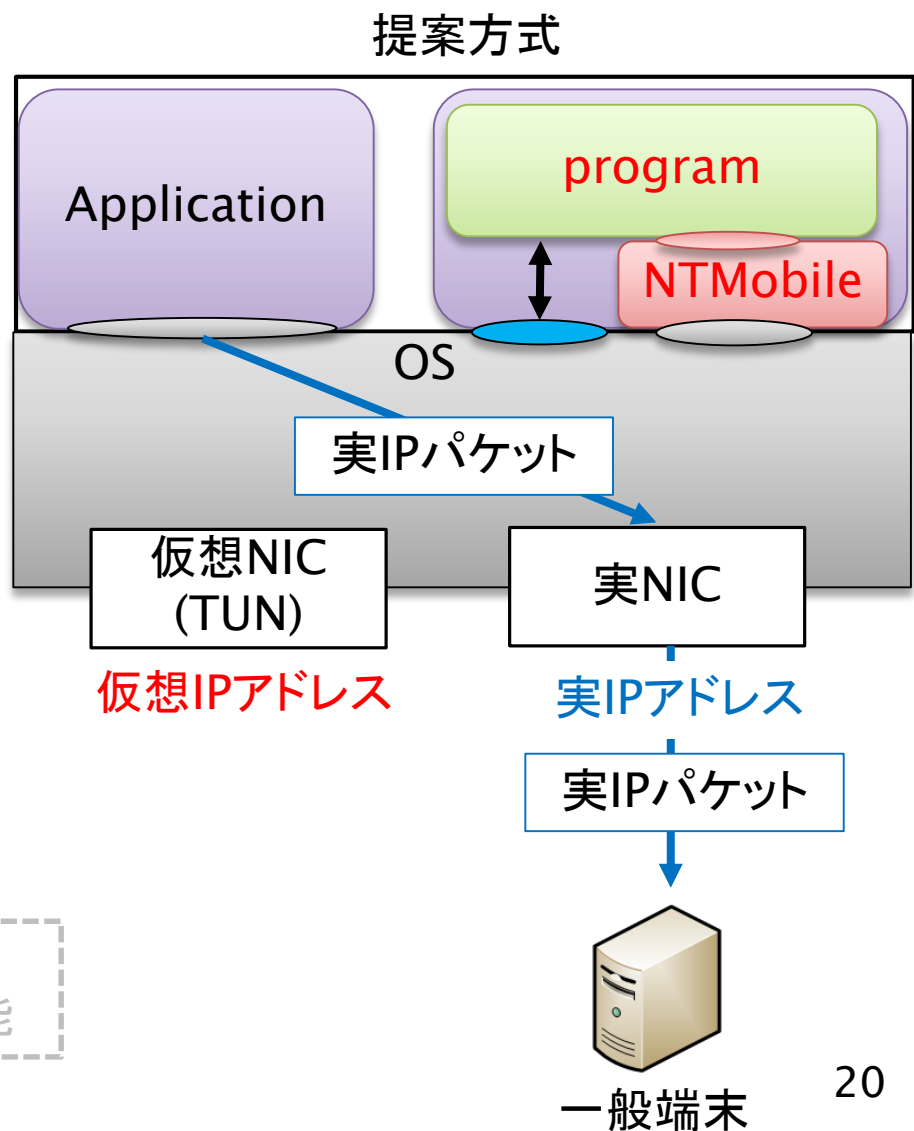
- NTMobile通信を行いたい場合

- 仮想IPアドレスで通信
- TUNを経由のNTMobile通信

- 一般通信を行いたい場合

- 実IPアドレスで通信
- 従来通りの一般通信

通信相手の宛先によって
NTMobile/一般通信を利用するか選択可能



アプリケーションによるTUNの利用 (1/4)

- TUNに仮想IPアドレスを割り当てる

仮想IPパケット → TUN
実IPパケット → 実NIC

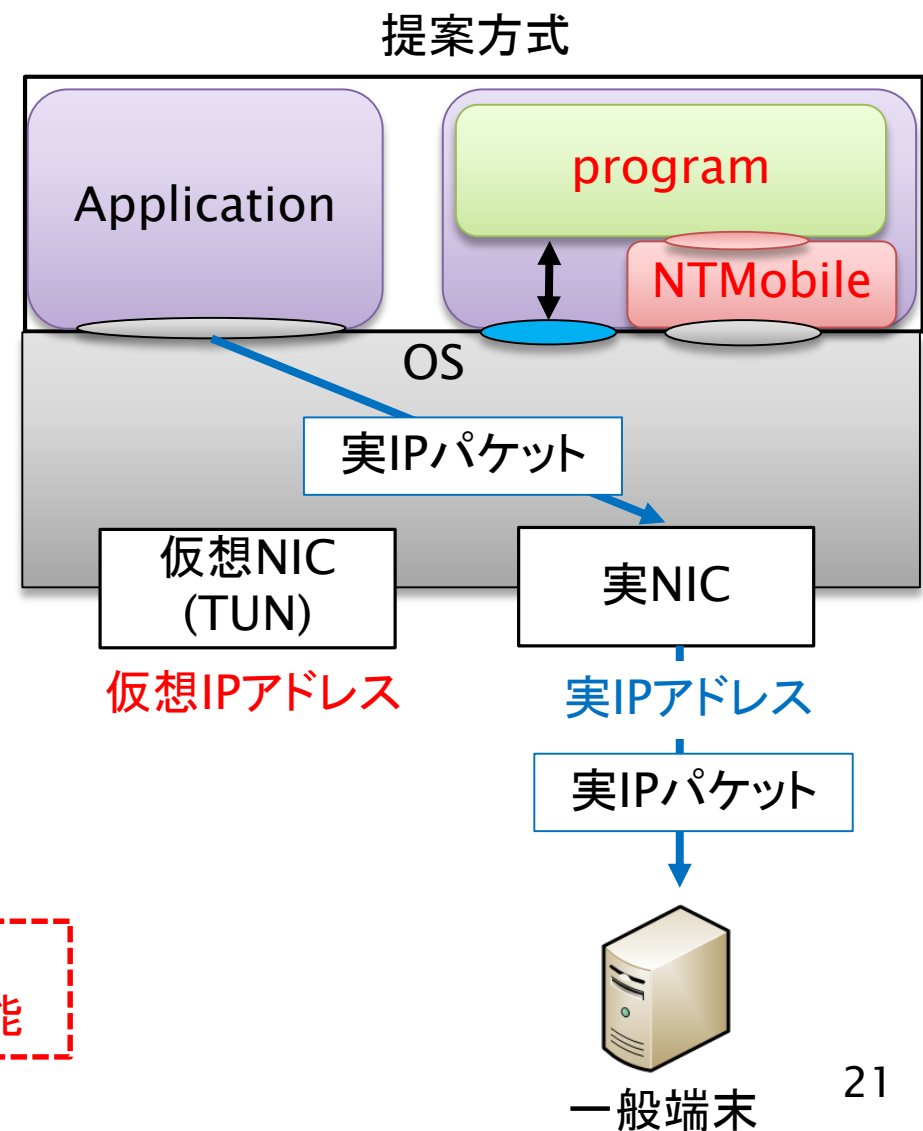
- NTMobile通信を行いたい場合

- 仮想IPアドレスで通信
- TUNを経由のNTMobile通信

- 一般通信を行いたい場合

- 実IPアドレスで通信
- 従来通りの一般通信

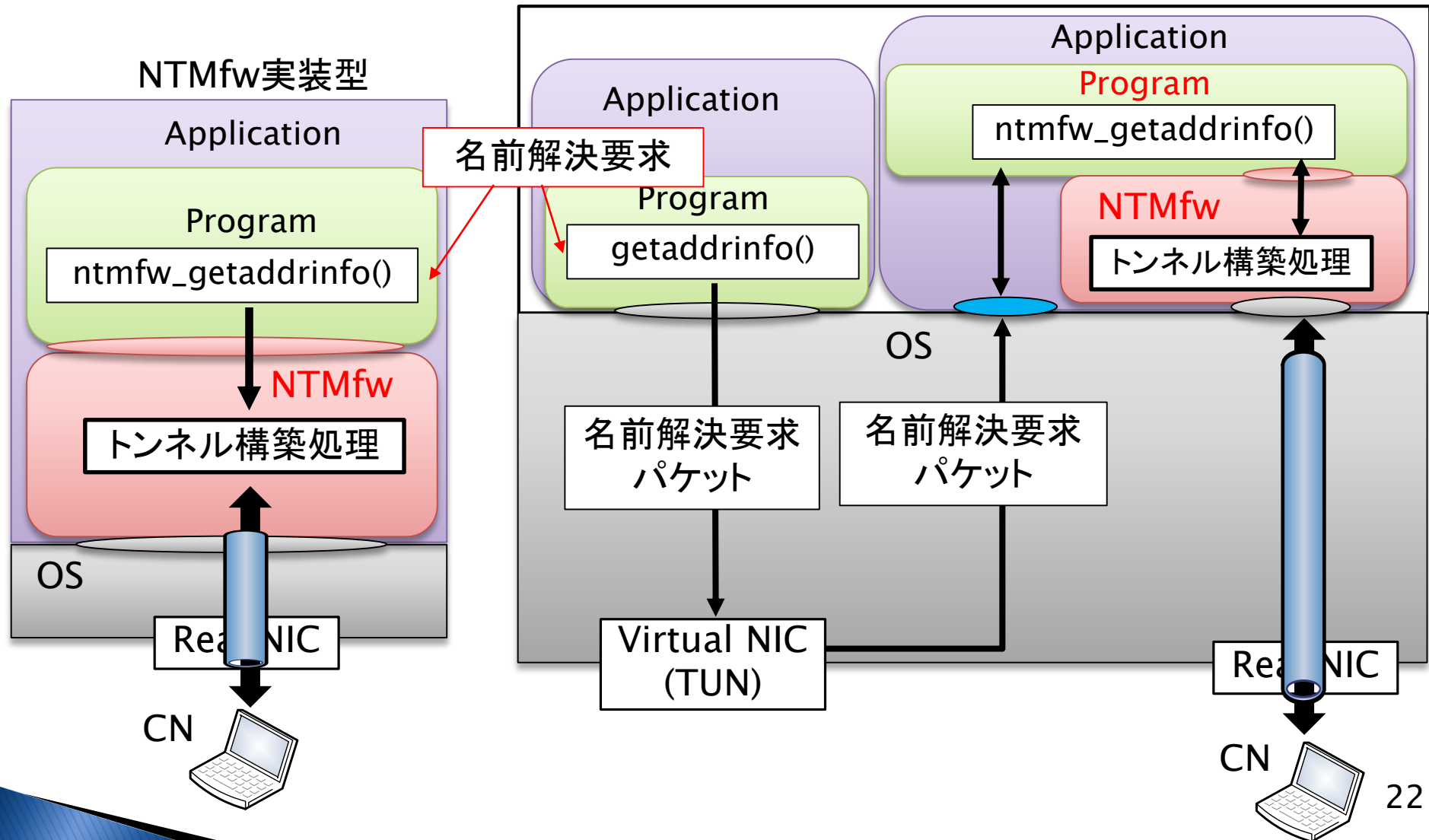
通信相手の宛先によって
NTMobile/一般通信を利用するか**選択可能**



BSDソケットAPIによるトンネル構築処理

- 名前解決要求パケットをトンネル構築のトリガに変更
 - DNSパケットをTUNにルーティングするように設定

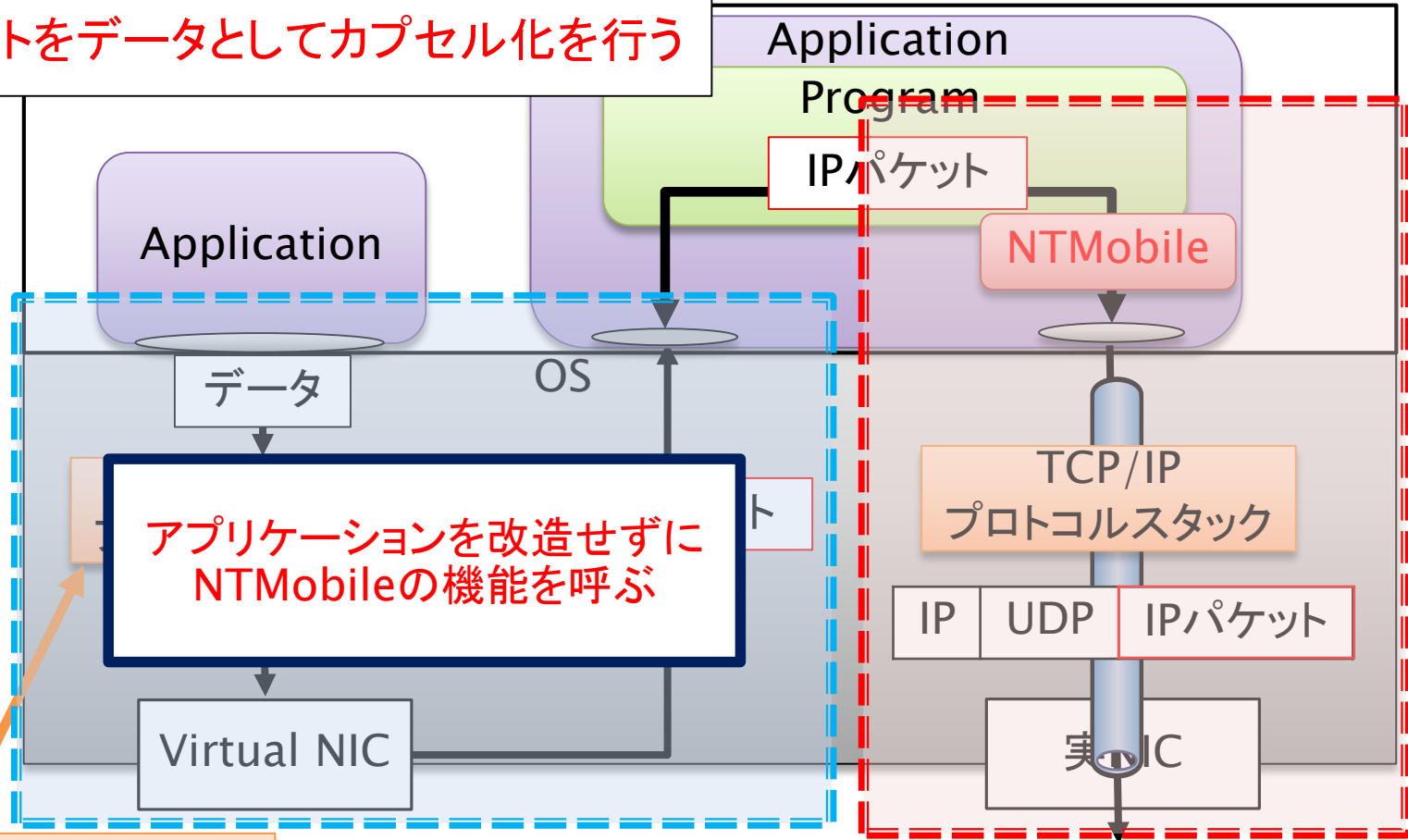
提案方式



アプリケーションを改造しない方式の検討

検討事項

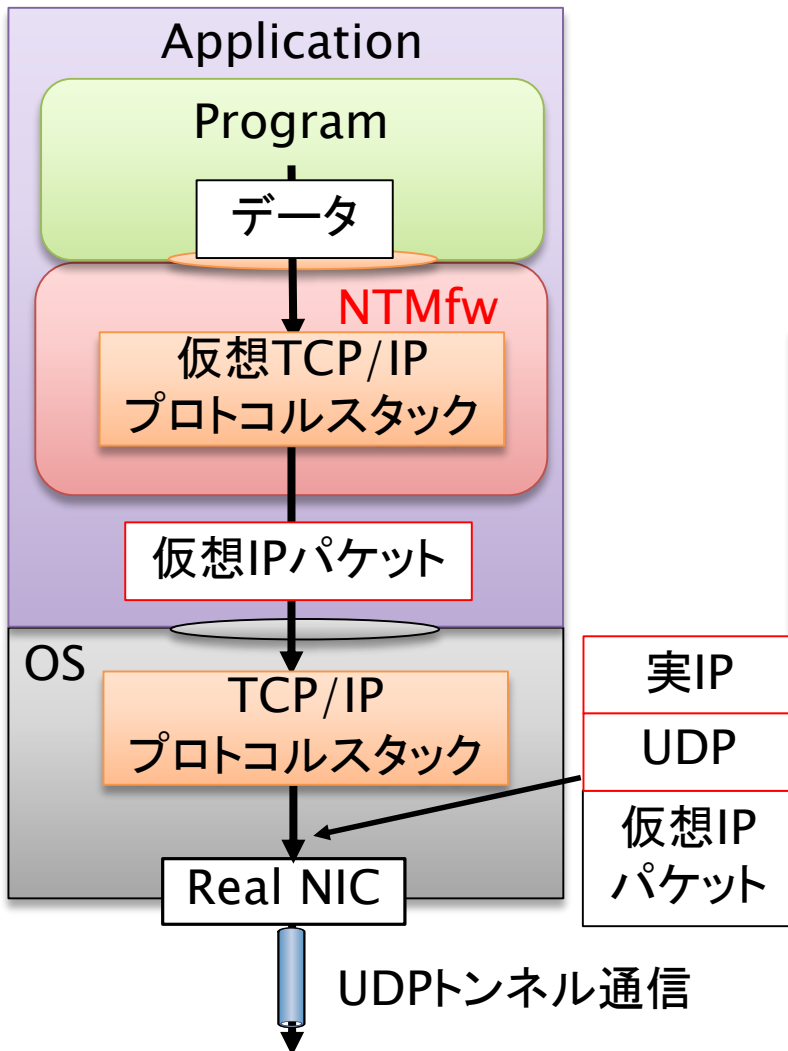
- ✓ IPパケットをユーザ空間に取り込む
- ✓ IPパケットをデータとしてカプセル化を行う



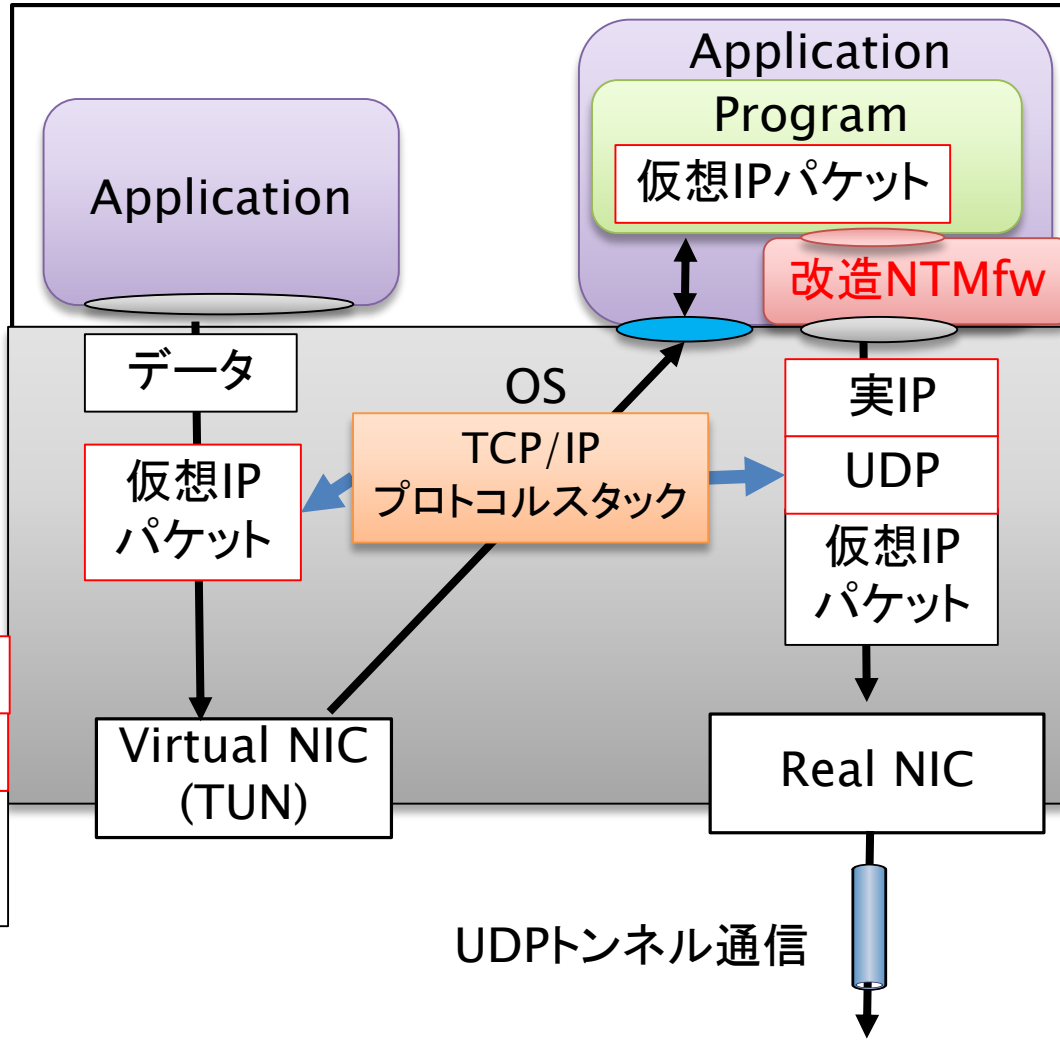
NTMobile通信

カプセル化の実現方法

NTMfw実装型

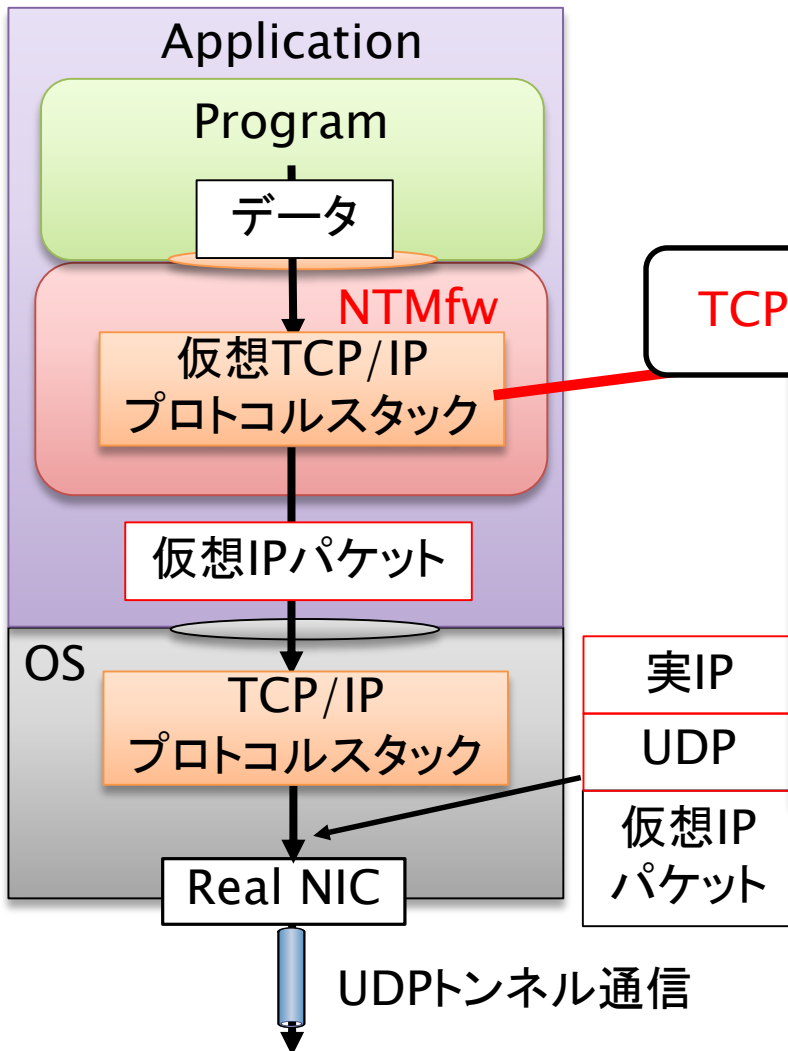


提案方式

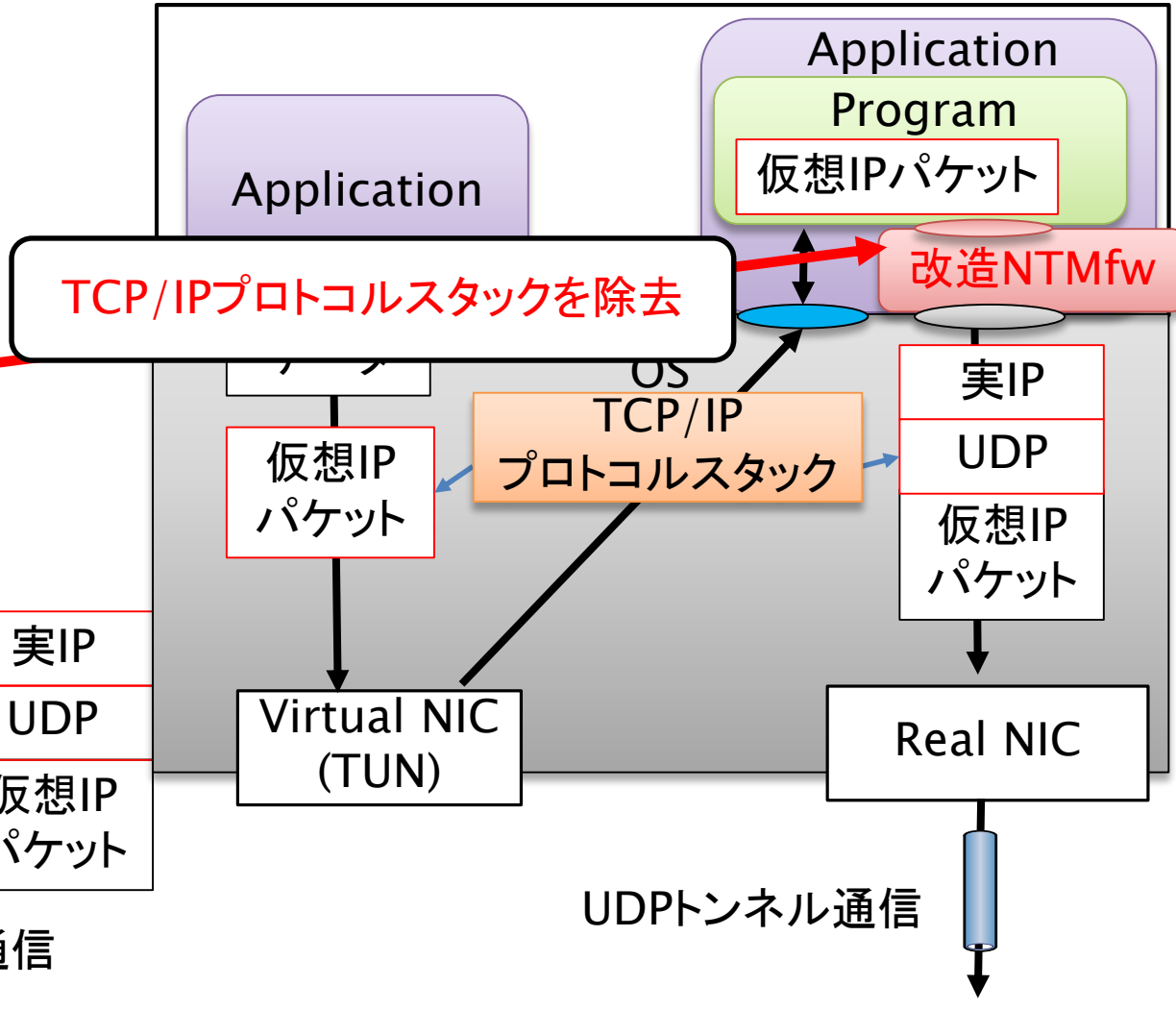


カプセル化の実現方法

NTMfw実装型



提案方式

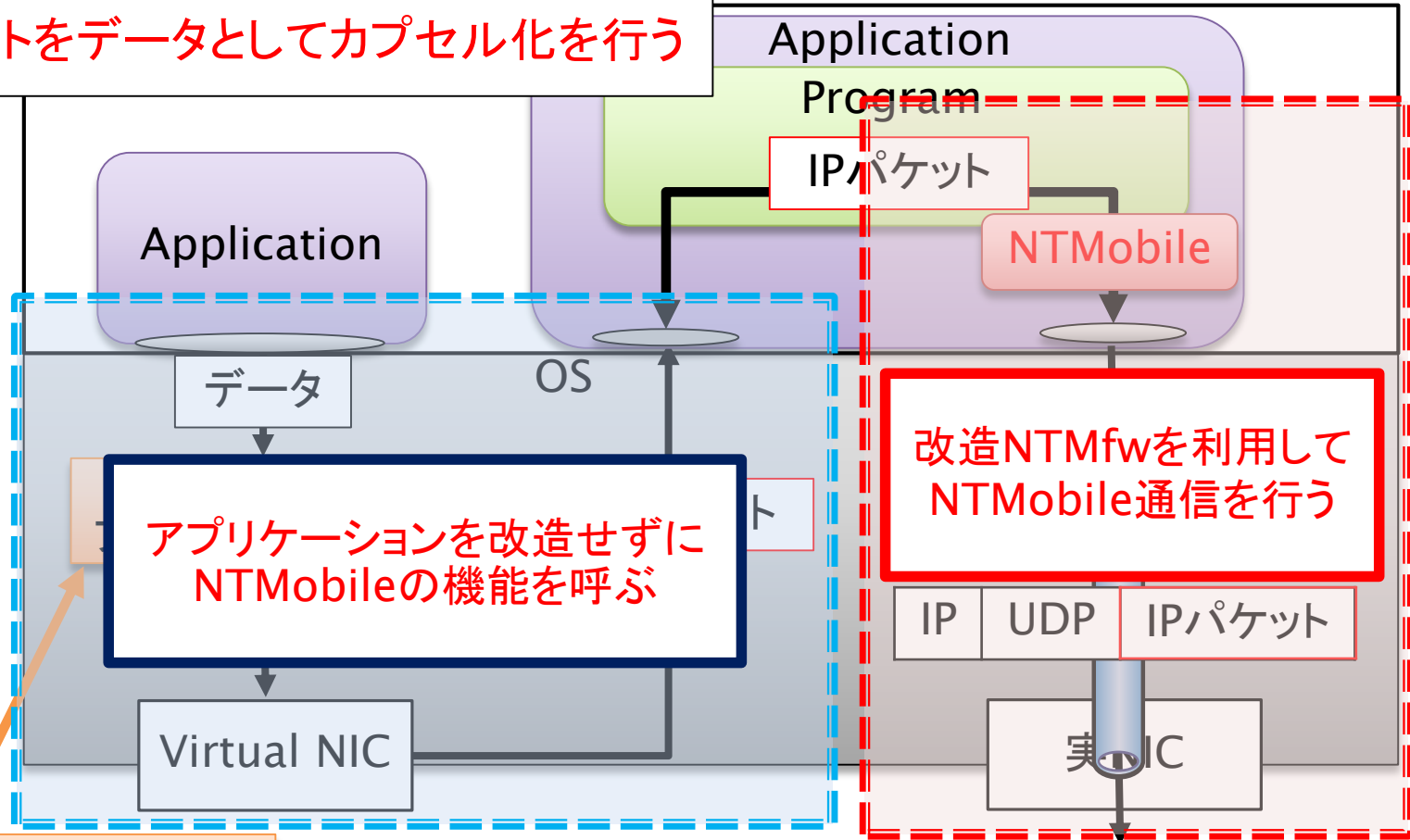


BSDソケット TUNソケット NTMソケット

アプリケーションを改造しない方式の検討

検討事項

- ✓ IPパケットをユーザ空間に取り込む
- ✓ IPパケットをデータとしてカプセル化を行う

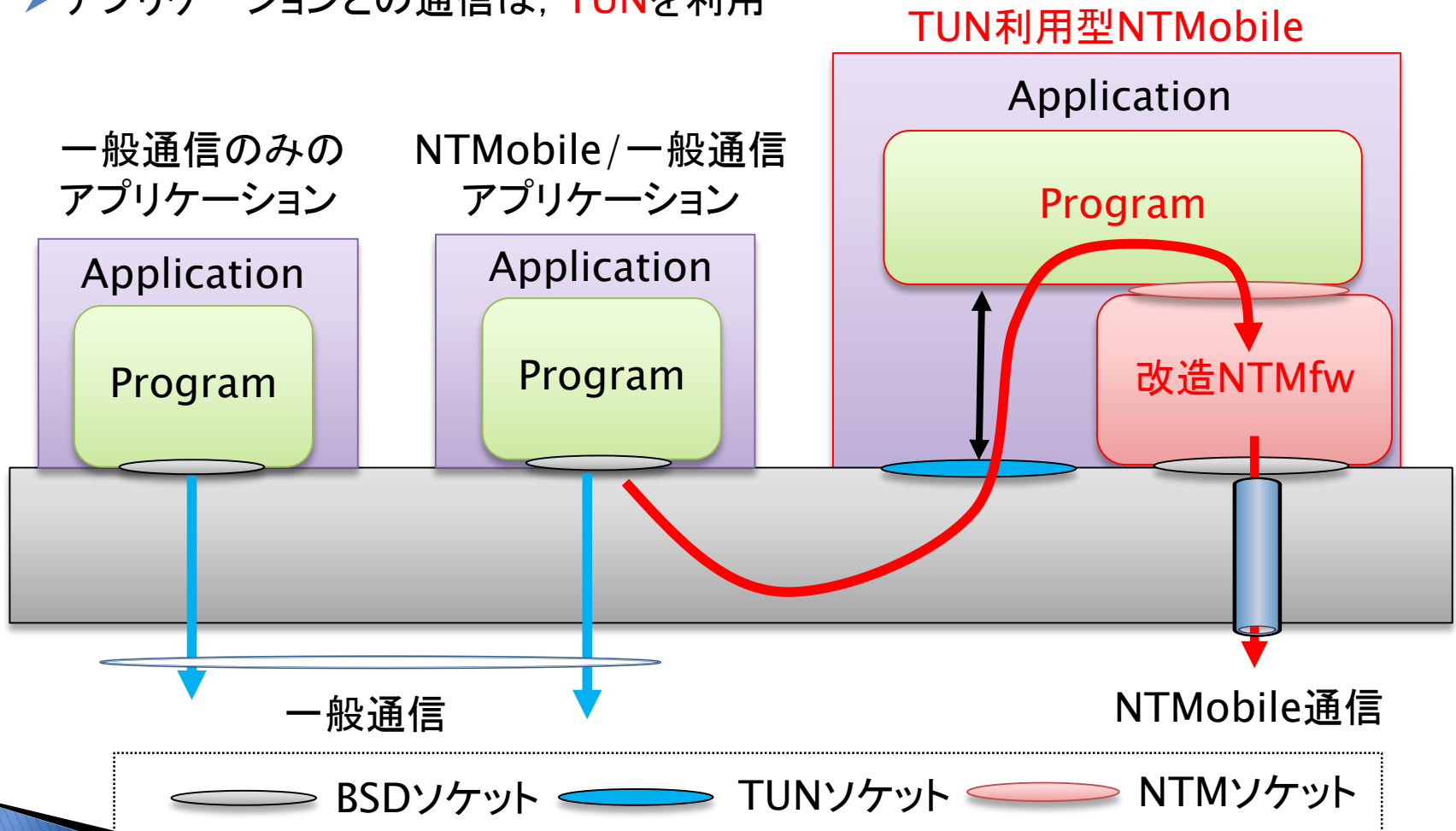


アプリケーションが送信したデータからIPパケットを生成

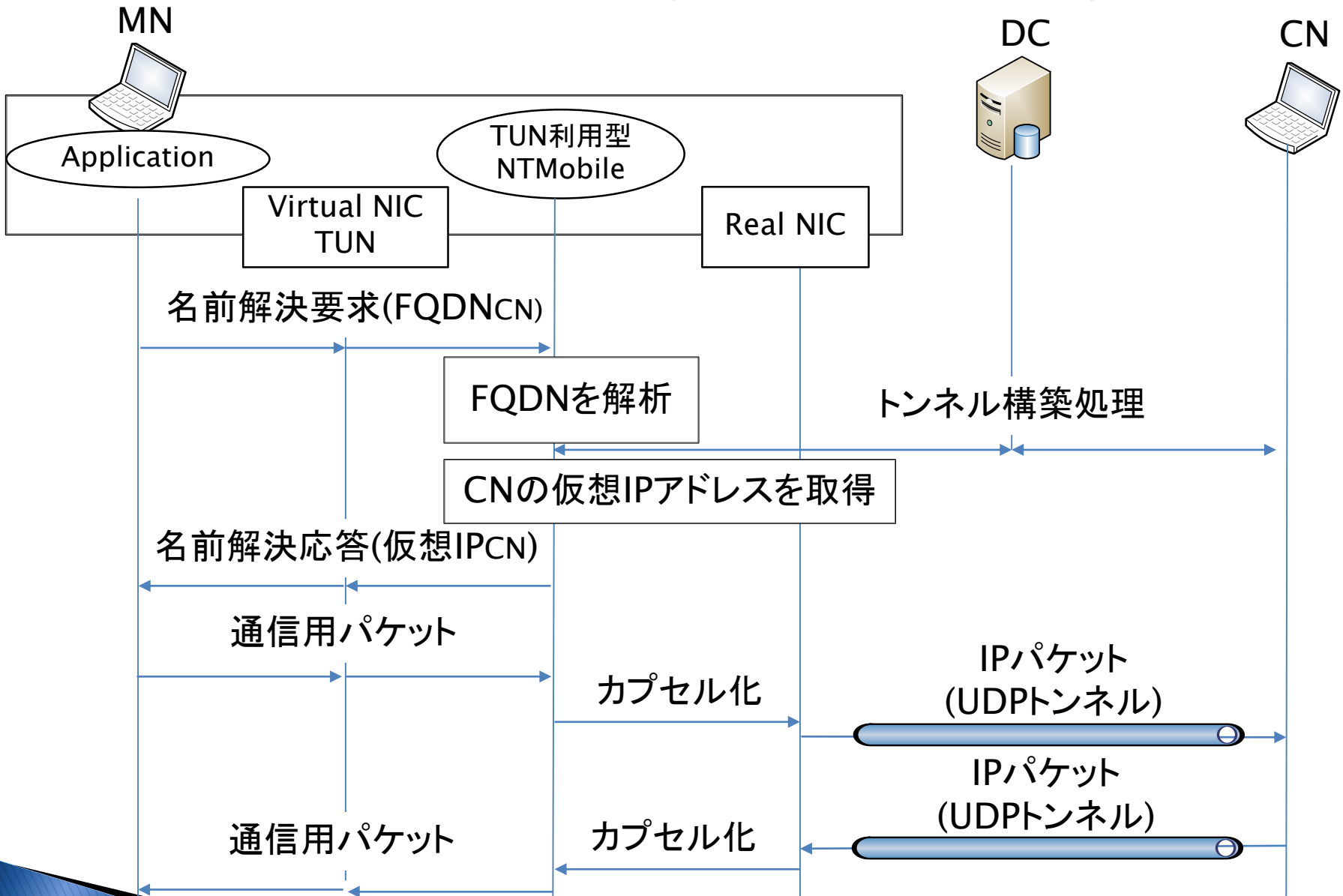
NTMobile通信

提案方式

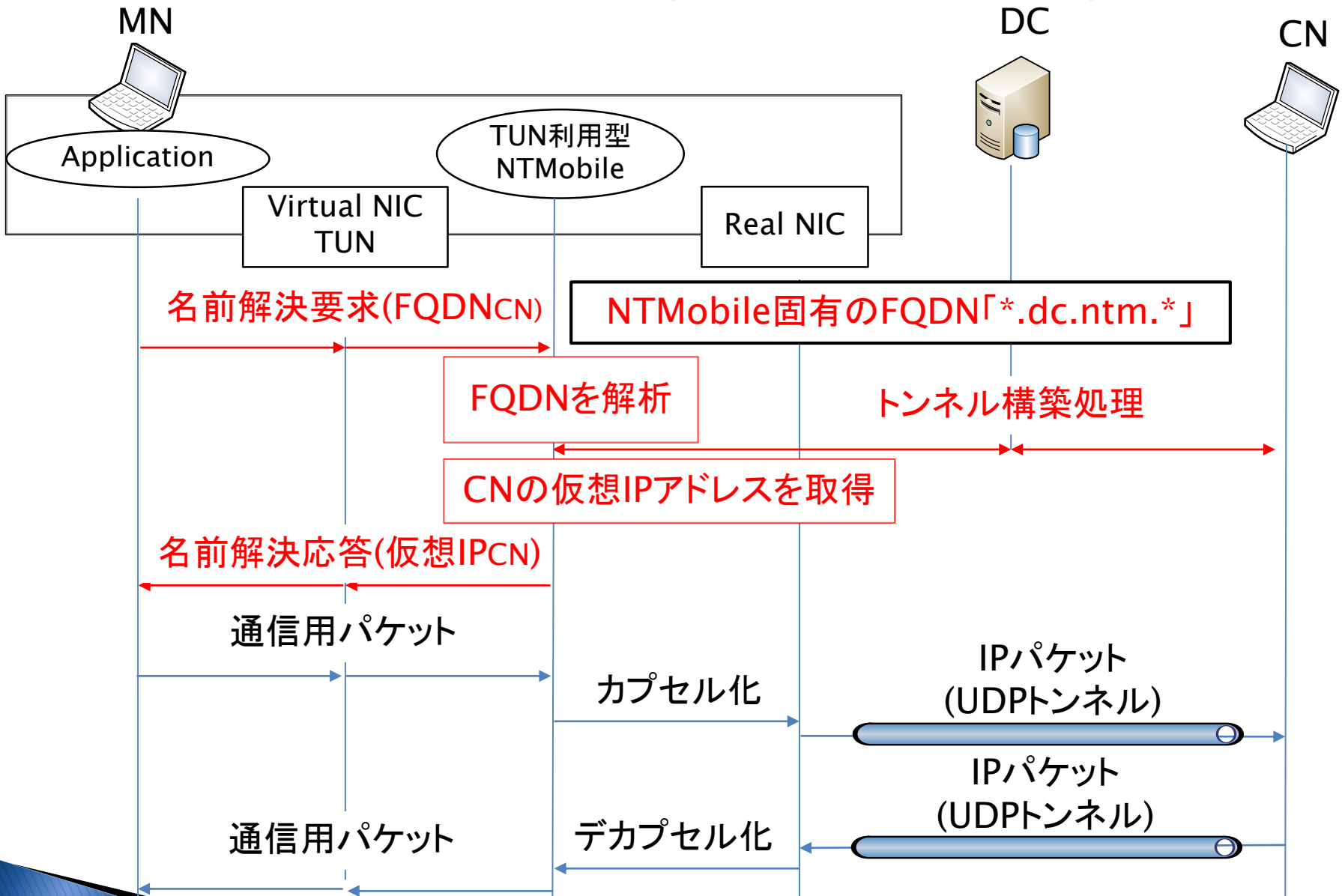
- 改造したNTMfwとTUNにより実現
 - NTMfwからTCP/IPプロトコルスタックを除去
 - アプリケーションとの通信は, TUNを利用



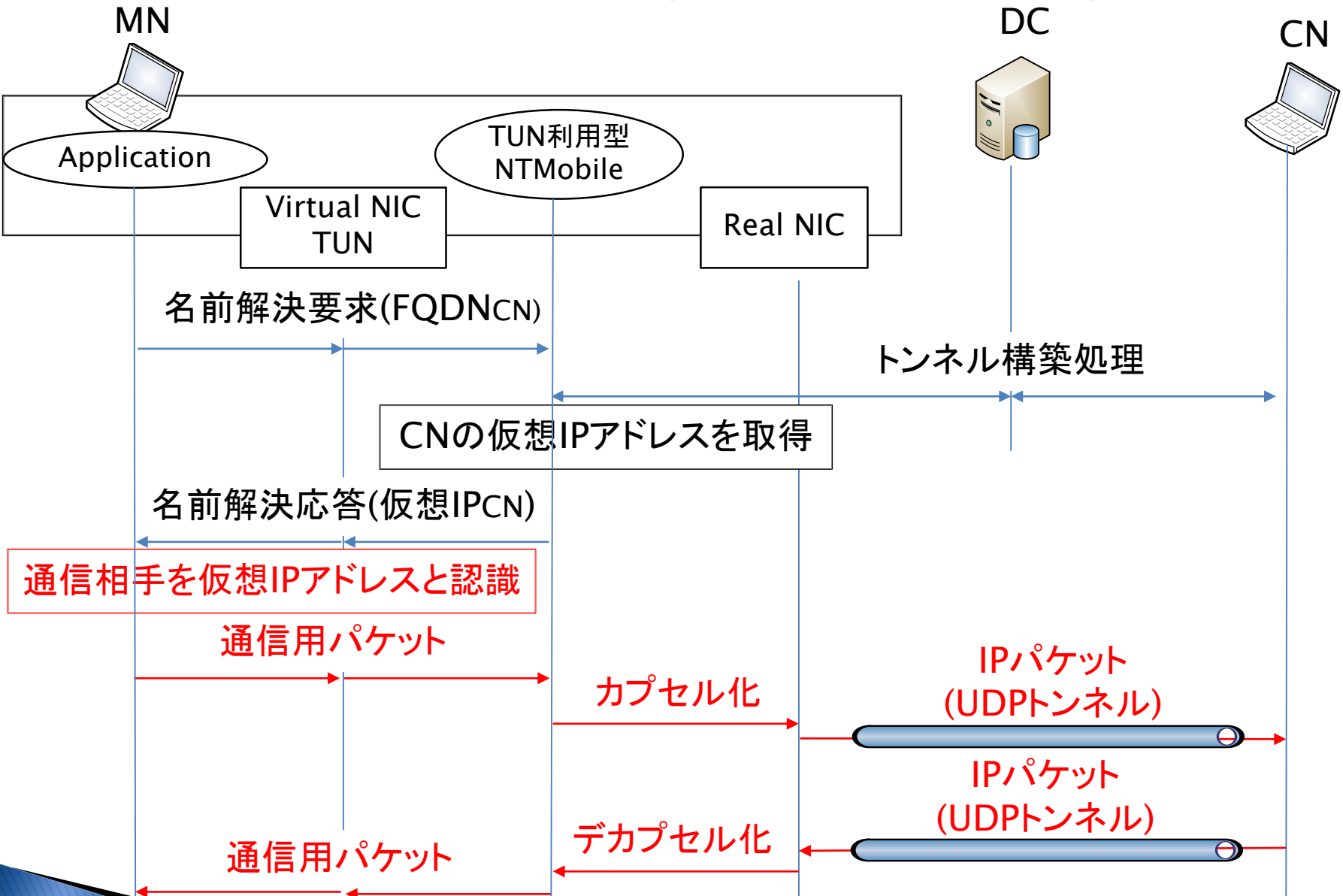
動作シーケンス (通信開始時)



動作シーケンス (通信開始時)

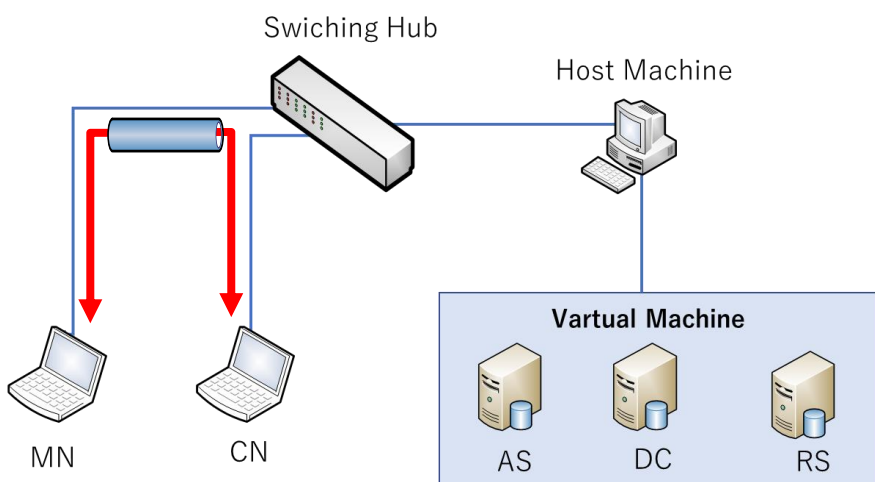


動作シーケンス (通信開始時)



実装と動作検証

- 提案方式の機能をLinux上に実装
 - 端末にインストールすることで利用可能
 - アプリケーションを改造することなくNTMobileの機能を利用可能
- 検証環境 (IPv4ネットワーク)
 - Raspberry PI3(MN, CN)に提案方式を実装
 - 仮想マシン上にAS, DC, RSを構築



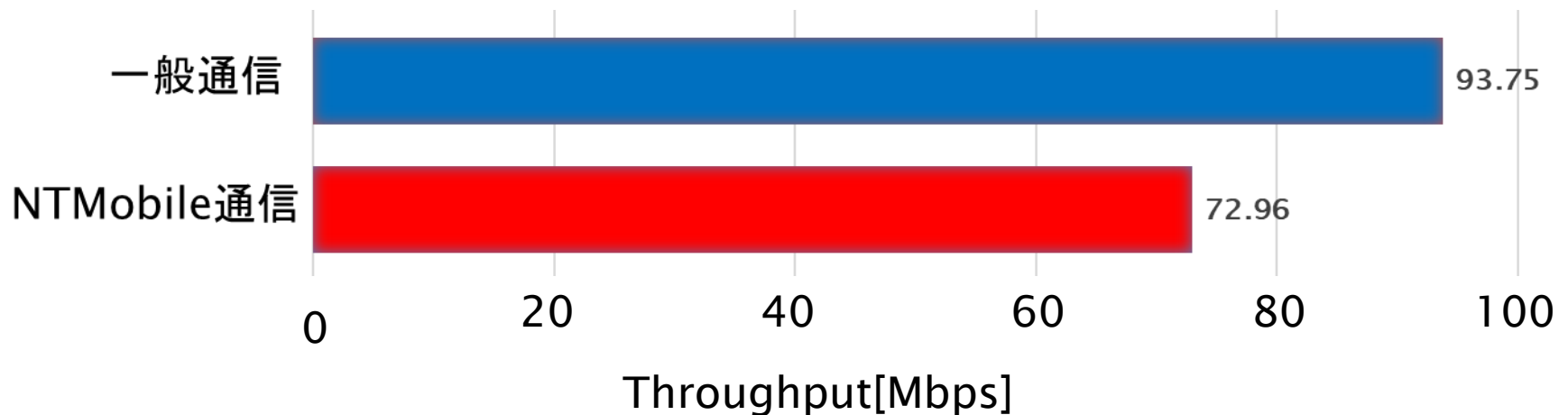
1000BASE-Tの有線LANネットワーク

MN, CNの諸元	
OS	Linux raspberrypi 3.18 (32bit)
CPU	ARMv8 Cortex-A53 (2.5GHz)
Memory	1GB

AS, DC, RS (Virtual Machine)の諸元	
OS	Ubuntu14.04 (32bit)
CPU	2core2thread
Memory	2GB

スループットによる性能評価

- 実用上問題ないかを評価
- 一般通信(提案手法なし)とNTMobile通信(提案手法あり)における比較
 - *Iperfを用いたTCP通信による測定
 - それぞれの通信を10回行い平均値を結果とする

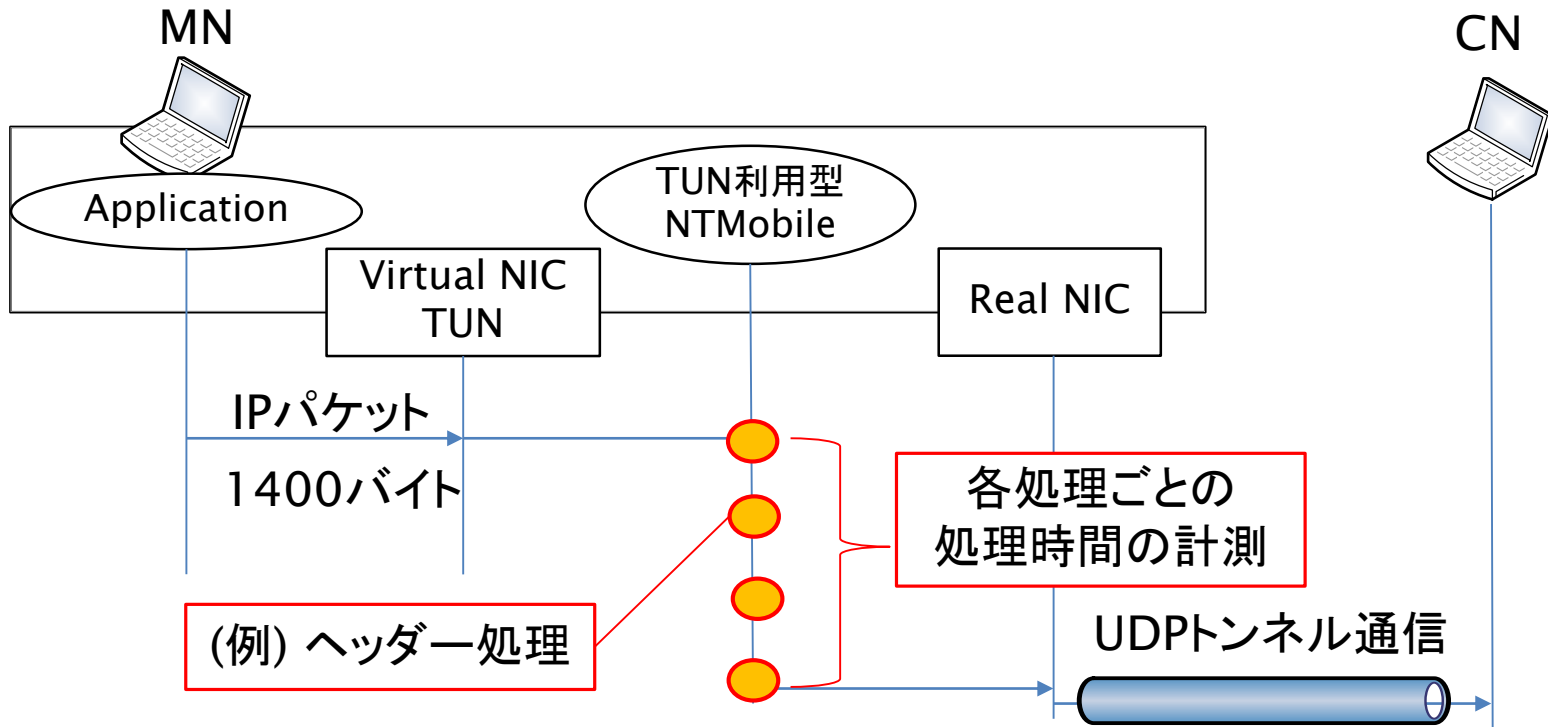


TUN利用型NTMobileを利用すると一般通信と比べて**22%低下**

*Iperf : スループットを測定するツール

処理時間の測定(MN側)

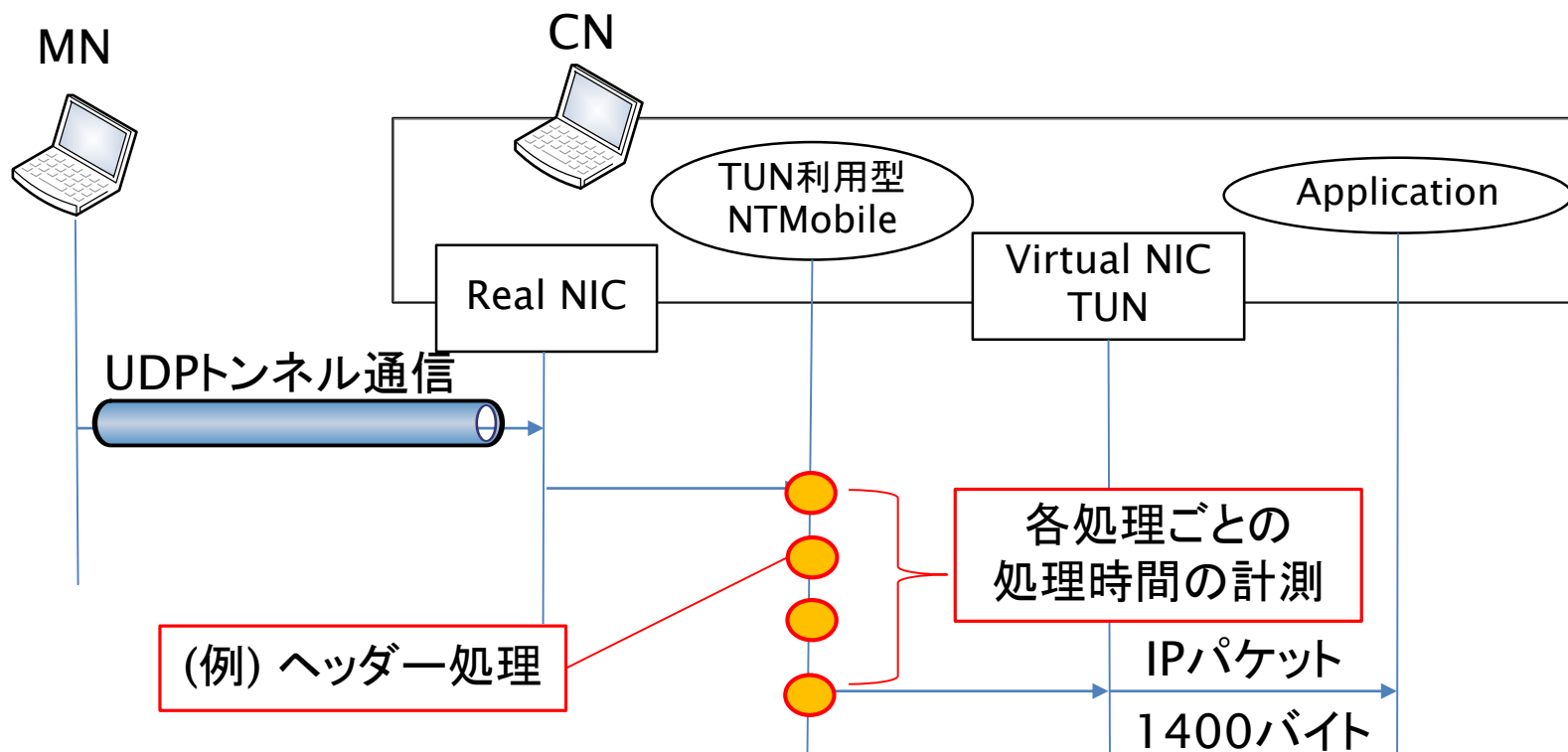
- スループットの低下原因を調査
 - NTMobile通信時の各処理に要する時間を計測
 - アプリケーションは*1400バイトのパケットを送信
 - clock_gettime関数により処理時間を取得



*1400バイト：提案方式において、アプリケーションが送信できるパケットの最大長

処理時間の性能評価 (CN側)

- スループットの低下原因を調査
 - NTMobile通信時の各処理に要する時間を計測
 - アプリケーションは*1400バイトのパケットを送信
 - clock_gettime関数を利用



*1400バイト：提案方式において、アプリケーションが送信できるパケットの最大長

処理時間の性能評価 (結果)

- NTMobileは、セキュリティ機能を有している。
 - 暗号化/復号処理による、パケット盗聴の防止
 - MAC(Message Authentication Code)による、パケット改竄検知

MN側の処理時間測定結果(μ秒)

	区分	処理時間	合計
TUN利用型NTMobile 処理時間	暗号化	116.9	210.0
	MAC生成	30.5	
	その他	62.5	

暗号化, MAC生成
70%占める

CN側の処理時間測定結果(μ秒)

	区分	処理時間	合計
TUN利用型NTMobile 処理時間	復号	125.4	190.7
	MAC検証	40.0	
	その他	25.3	

復号処理, MAC検証
87%占める

性能評価の考察

- 提案方式を使用した場合のスループットは, 72.96Mbps
 - 一般通信に比べて22%のスループット低下
- スループット低下の原因は, セキュリティ処理
 - パケットの機密性, 完全性などを保障するため**必須処理**

*Skypeに必要なスループット

用途	推奨スループット
通話	100 kbps
ビデオ通話(HD)	1.5 Mbps
グループビデオ通話(7人)	8 Mbps

セキュリティ処理を行いながら, 一般的な用途において実用可能

まとめ

- アプリケーションとしてNTMobileの機能を提供
 - TUNインターフェスを利用することで実現
 - 端末のプログラムを一切変更する必要がない
 - 通信相手によって、一般通信/NTMobile通信の選択

- 提案方式の実装と評価
 - 一般通信に比べて22%スループットが低下
 - スループット低下の原因は、暗号/復号処理, MAC生成/検証
 - 提案方式は、充分実用的な性能