

平成30年度 修士論文

和文題目

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

英文題目

**A Proposal for a Communication System that  
Eliminates Restrictions of IP Networks Without  
Any Changes of End Nodes and its Implementation**

情報工学専攻 渡邊研究室

(学籍番号: 173426006)

尾久 史弥

提出日: 平成31年1月29日

名城大学大学院理工学研究科

## 概要

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) 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の仕様であるため、普及が進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) 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修正して開発する場合は、ソケット API (Application Programming Interface) を書き換える必要がある。また、アプリケーションのソースコードが公開されていない場合は、NTMfw を利用することができない。そこで、本研究では TUN (TUNnel) と NTMfw の一部を改造した R-NTMfw (Remodeled NTMfw) を利用した TUN 利用型 NTMobile を実現し、既存のアプリケーションは一切の改造を行わずに NTMobile の機能を利用できる方式を提案する。実装及び動作検証により、既存のアプリケーションに対して一切の変更を行うことなく、NTMobile の機能を利用できることを確認した。また、既存のアプリケーションが TUN 利用型 NTMobile を利用して NTMobile 通信を行った時のスループットを測定した結果、72.96Mbps であることが分かった。このスループットの値は、一般的なビデオチャットアプリケーションにおいて、高品質なビデオ通話やグループ通話が可能なことを示し、一般的な用途において実用できる性能であることを確認した。

# 目次

<b>第 1 章</b>	<b>序論</b>	<b>1</b>
<b>第 2 章</b>	<b>既存技術</b>	<b>4</b>
2.1	カーネル空間を改造する方式	4
2.1.1	概要	4
2.1.2	課題	4
2.2	カーネル空間を改造しない方式	6
2.2.1	NTMobile の概要	6
2.2.2	NTMobile 通信開始時の動作	7
2.2.3	NTMfw の概要	7
2.2.4	NTMfw の動作	9
2.2.5	課題	10
<b>第 3 章</b>	<b>TUN 利用型 NTMobile</b>	<b>11</b>
3.1	TUN の概要	11
3.2	R-NTMfw の概要	11
3.3	提案方式のカプセル化処理に着目した動作概要	11
3.4	TUN 利用型 NTMobile の実現に向けた検討と実装方針	13
3.4.1	TUN に割り当てる IP アドレス	13
3.4.2	BSD ソケット API による NTMobile シグナリングの開始	13
3.4.3	DNS パケットをルーティングするための仮想 IP アドレスの導入	14
3.4.4	名前解決要求の宛先として用いる仮想 IPv4 アドレスの定義	14
3.4.5	任意の DNS サーバーを利用できることを考慮した名前解決要求の宛先の設定	14
3.4.6	IP フラグメンテーションの課題解決	14
3.5	TUN 利用型 NTMobile の動作概要	15
3.5.1	起動時の処理	15
3.5.2	NTMobile 通信開始時の動作	15
3.5.3	代替 DNS サーバーによる一般端末との通信開始時の動作	16
<b>第 4 章</b>	<b>実装</b>	<b>18</b>
4.1	起動時の処理	18
4.2	モジュール構成	18

<b>第 5 章</b>	<b>動作検証と評価</b>	<b>20</b>
5.1	NTMobile 通信の動作検証 . . . . .	20
5.1.1	通信接続性試験 . . . . .	20
5.1.2	移動透過性試験 . . . . .	21
5.2	スループットによる性能評価 . . . . .	22
5.2.1	評価構成 . . . . .	22
5.2.2	スループットの測定 . . . . .	23
5.2.3	処理時間の測定 . . . . .	23
5.2.4	考察 . . . . .	25
5.3	代替 DNS サーバーを利用した時に発生するタイムアウト時間の測定 . . . . .	26
5.4	TUN 利用型 NTMobile の課題 . . . . .	27
<b>第 6 章</b>	<b>結論</b>	<b>29</b>
	謝辞	30
	参考文献	31
	研究業績	33

# 第1章 序論

今日のインターネットは、IPv4 (Internet Protocol version 4) グローバルアドレスの枯渇問題 [1] に伴い、NAT (Network Address Transmission) を導入して IPv4 プライベートアドレスを導入している。NAT が導入されているネットワークでは、グローバルネットワーク側からプライベートアドレス側へ通信を開始できない NAT 越え問題と呼ばれる課題がある [2]。また IPv6 (Internet Protocol version 6) アドレスへの移行が進められているが、IPv4/IPv6 アドレスには互換性がなく、直接通信を行うことができない。このことから NAT 環境や IPv4 と IPv6 が混在した環境においても相互通信を保証する通信接続性が求められている。

一方、スマートフォンをはじめとするモバイル端末が急激に普及し、携帯網へのトラフィックが爆発的に増加している [3]。そこで、携帯電話事業者は携帯網のトラフィックを削減するため、Wi-Fi (Wireless Fidelity) へのトラフィックオフロードを進めている [4]。TCP (Transmission Control Protocol) /IP ネットワークにおいて、ネットワークを切り替えると IP アドレスが変化して、今まで行っていた通信が断絶してしまうため、IP アドレスが変化した場合でも通信を継続できる移動透過性技術が求められている [5]。

通信接続性と移動透過性を実現する技術として、DSMIPv6 (Dual Stack Mobile IPv6) [6] や HIP (Host Identity Protocol) [7], NTMobile (Network Traversal with Mobility) [8] [9] [10] が提案されている。DSMIPv6 と HIP は IETF (The Internet Engineering Task Force) により、既に標準化された技術ではあるが、アーキテクチャの構造上、ユーザ空間に実装することが難しく OS (Operating System) のカーネル空間に実装することが必須であるため、普及がすすまない課題がある。特にスマートフォンではカーネルを改造すると、メーカーのサービス保証を受けることができない。これに対し NTMobile は、通信ライブラリである NTMfw (NTMobile framework library) をアプリケーションに組み込むことで利用できるため、カーネルを改造しないでユーザ空間で利用することができる。[11]。しかし、NTMfw は新規に作成するアプリケーションの場合は問題ないが、既存のアプリケーションを修正して開発する場合は、プログラム内のソケットインターフェースを書き換える必要がある。また、アプリケーションのソースコードが公開されていない場合は、NTMfw を利用することができない。

これらの既存技術の課題から通信接続性と移動透過性を提供する技術はユーザ空間において実装し、アプリケーションを改造せずに利用できることが求められている。ユーザ空間において通信接続性と移動透過性を提供できる技術は NTMobile のみであることから、アプリケーションは BSD ソケット API (Berkeley Software Distribution Sockets Application Programming Interface) を利用して NTMobile の機能を利用できると有用である。アプリケーションが BSD ソケット API を利用して通信を行う時の動作に着目すると、アプリケーションが送信したデータを基に、カーネル空間

の TCP/IP プロトコルスタックで IP パケットが生成される。NTMobile のカプセル化パケットは、IP パケットに対して UDP (User Datagram Protocol) カプセル化を行うパケットの構造であるため、上記のアプリケーションが送信した IP パケットをそのまま利用することができる。このことから、アプリケーションが送信した IP パケットをカーネル空間からユーザ空間に取りこみ、再度カーネル空間において UDP カプセル化を行うことで、アプリケーションを改造しないで NTMobile 通信を実現できる。

アプリケーションが送信した IP パケットをカーネル空間からユーザ空間に取り込む処理は、TUN (TUNnel) を利用することで実現できる。TUN は、IP パケットをカーネル空間からユーザ空間へ取り込むことができる機能であり、仮想の NIC (Network Interface Card) を提供する [12]。これまでに TUN を利用した NTMobile (TUN 利用型 NTMobile) の提案が行われ、通信開始時の動作検証が行われていたが、通信処理時の実装が完了しておらず、性能評価も行われていない [13]。また、NTMobile の実装は TUN 利用型 NTMobile の他にも複数のモデルが提案されており、仕様の把握が困難であったことから、これらの仕様を統一的な枠組みとして再定義された [14]。以降、新しい NTMobile の仕様に基づいた TUN 利用型 NTMobile は検討がされていない。

そこで本研究では、TUN と NTMfw の一部を改造した R-NTMfw (Remodeled NTMfw) を利用することで、新しい NTMobile の仕様に基づいた TUN 利用型 NTMobile を実現し、既存のアプリケーションは一切の改造を行わずに NTMobile の機能を利用できることを提案する。既存のアプリケーションが BSD ソケット API を利用して NTMobile の機能呼び出す動作は、従来の TUN 利用型 NTMobile の動作を流用し、TUN インターフェースに仮想 IP アドレスを割り当てて、仮想 IP アドレス宛のパケットは全て TUN インターフェースにルーティングするようにした。また、NTMobile では NTMfw が提供するソケット API により、名前解決処理と NTMobile シグナリング処理を行う必要があるが、既存のアプリケーションは BSD ソケット API で名前解決処理を行う必要がある。BSD ソケット API により名前解決処理が行われると、カーネル内部で名前解決要求パケットが構築され、当該パケットにより名前解決処理が実行される。この動作に着目して、提案方式でも名前解決要求パケットを TUN インタフェースにルーティングし、名前解決処理と NTMobile シグナリングのトリガとした。TUN 利用型 NTMobile の新たな要求として、ユーザ空間上で IP パケットを操作できるため、IP パケットをデータとみなして、カーネル内部で UDP カプセル化を行う R-NTMfw を新たに提案して利用する。また、アプリケーションの通信やシステム構成によっては、ユーザが任意の DNS (Domain Name System) サーバーを利用して、一般通信を行いたい場合がある。この要求に対して、DNS サーバーの優先順位に基づいた名前解決要求パケットのルーティング方法により実現できることを提案する。

以上の検討を実装及び動作検証を行った結果、既存のアプリケーションを一切変更することなく NTMobile の機能を利用できるようになった。また、アプリケーションは通常の一般通信も、併行して行うことができることを確認した。既存のアプリケーションが TUN 利用型 NTMobile を利用して NTMobile 通信を行った時のスループットを測定した結果、72.96Mbps であることが分かった。このスループットの値は、一般的なビデオチャットアプリケーションにおいて、高品質なビデオ通話やグループ通話が可能なことを示し、一般的な用途において実用できる性能であることを

確認した。

以降、第2章において、通信接続性と移動透過性を実現する既存技術について述べ、3章で提案方式であるTUN利用型NTMobileについて述べる。4章ではTUN利用型NTMobileの実装について述べ、5章では、実装したTUN利用型NTMobileの動作検証と評価を行う。最後に第6章でまとめる。

## 第2章 既存技術

本章では、IPv4/IPv6 混在環境において通信接続性と移動透過性を実現する技術について述べる。

### 2.1 カーネル空間を改造する方式

カーネル空間を改造して実装することが必須である DSMIPv6 と HIP について説明する。

#### 2.1.1 概要

DSMIPv6 は、Mobile IPv6 [15] を IPv4/IPv6 混在環境に対応するように拡張したものである。すなわち、IPv6 の移動透過性技術をベースとして、NAT 越え、IPv4/IPv6 間通信を可能にした方式である。IPv4/IPv6 デュアルスタックネットワークに設置する HA (Home Agent) と移動端末 MN (Mobile Node) がトンネル経路を構築する。MN はホームネットワークで取得する HoA (Home Address) と移動先のネットワークで取得する CoA (Care-of Address) の2種類のアドレスを持つ。MN がどのようなネットワークに移動しても HoA は変化せず、CoA のみが変わる。MN の上位アプリケーションは HoA を用いて通信しており、CoA の変化を隠ぺいすることができる。IPv6 オンリーのネットワークでは経路最適化により直接通信を行うが、それ以外の通信はすべて HA が介在することにより実現する。

HIP は、IP アドレスが持つ通信識別子と位置識別子の役割のうち、端末識別子を分離し、端末識別子として HI (Host Identifier) を導入する。IP 層と TCP/UDP 層との間に新たに HIP 層を定義する。HIP 層においては、IP アドレスと HI のマッピングを管理し、上位層では HI を用いて通信を識別する。IP アドレスは位置識別子の役割のみを担うので、移動によって IP アドレスが変化しても、HI は変化せず移動透過性を実現する。HIP は既存技術を可能な限り流用するという方針を持ち、NAT 越え技術として ICE (Interactive Connectivity Establishment) [16] を改造する形で実現する。

#### 2.1.2 課題

DSMIPv6 は IPv4 ネットワークに対応するため、HA を経由して Mobile IPv4 [17] の技術をそのまま利用できるようにしている。Mobile IPv4 は IP in IP 処理を行い、Mobile IPv6 は IPv6 オプションを利用することから、カーネル内の IP 層を改造するのが前提の仕様である。したがって、DSMIPv6 をアプリケーションレベルで実現することは困難である。また、HIP は IP 層とトランスポート層の間に HIP 層を設ける構造上、カーネルの改造が必須である。

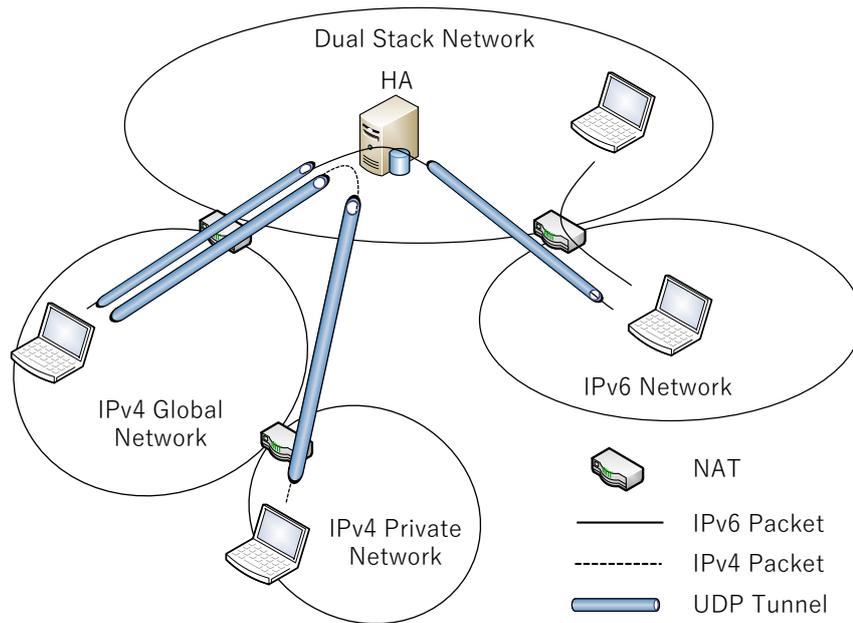


図1 DSMIPv6の構成

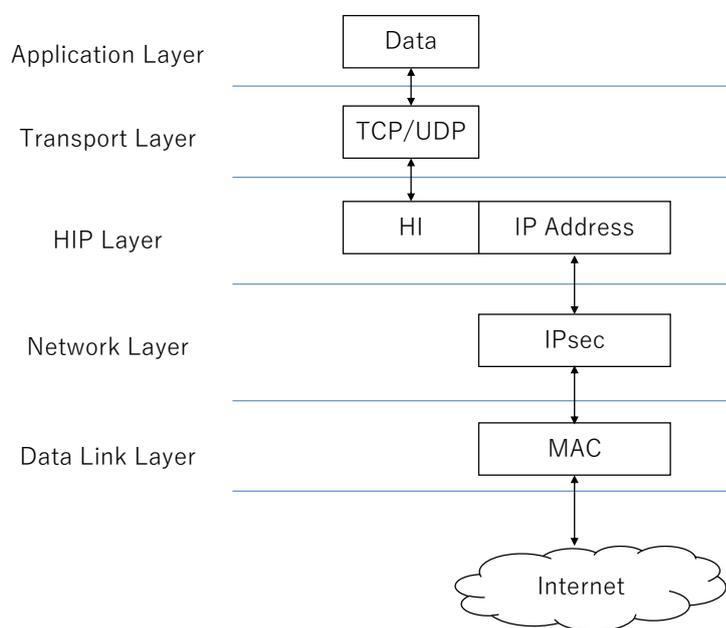


図2 HIPのレイヤーモデルの構成

DSMIPv6とHIPは、IETFにより標準化されているが、カーネル空間を改造することが必須である。例えば、OSのカーネルへ機能実装が可能なLinuxでは、短期間に膨大な量の変更がOSに対して行われている[18]。この変更の中には、カーネル内部のインターフェースの変更が含まれる場合があり、上記の内部インターフェースを利用しているプログラムは、その変更に合わせて修正

を行う必要がある。この OS の頻繁なバージョンの更新に追従しながらシステムを運用することは、プログラムの管理と運用の面からコストが大きく、かつ複数 OS のサポートを考慮すると膨大な量の開発や運用のコストが生じる。

一方、スマートフォンには DSMIPv6 と HIP は組み込まれていないため、スマートフォンを改造してこれらの技術を組み込む必要がある。しかし、スマートフォンではカーネルを改造すると、メーカーのサービス保証を受けることができなくなる。

## 2.2 カーネル空間を改造しない方式

カーネルを改造する方式では、スマートフォンをはじめとするモバイル端末に適用することが難しく、普及が進まない課題がある。そこで、通信ライブラリとして移動透過性と通信接続性の機能を提供することで、カーネルの改造が不要である NTMobile が提案されている。

### 2.2.1 NTMobile の概要

NTMobile は、IPv4/IPv6 ネットワークが混在した環境において、ネットワークのアドレス空間やアドレス体系に依存することなく、端末の通信接続性と移動透過性を同時に実現する通信技術である。NTMobile の機能を持つ端末は、IPv4/IPv6 ネットワークが混在した環境で自由な双方向通信が可能で、かつ通信中に移動しても通信を継続することができる。

図 3 に NTMobile のシステム構成を示す。NTMobile は、NTMobile の機能を持つ NTM 端末、NTM 端末のアカウントの管理および認証を行う AS (Account Server)、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)、NTM 端末がエンドツーエンド通信が行えない場合にパケットの中継を行う RS (Relay Server) によって構成される。

NTMobile では、DC が NTM 端末に対して、NTM 端末であることが識別可能な FQDN (Fully Qualified Domain Name) と所属する IP ネットワークに依存しない仮想 IP アドレスを割り当て、アプリケーション上では仮想 IP アドレスに基づいた通信を行う。NTMobile の機能を利用するアプリケーションは仮想 IP アドレスを通信識別子とするため、通信中に端末がネットワークを切り替えて実 IP アドレスが変化しても、仮想 IP アドレスは変わら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共に、NTM 端末に対して最適な通信経路の指示を行う。また、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、NAT 越えを実現することができる。

両 NTM 端末が直接通信を行うことができない場合や、NTM 端末の通信相手が一般端末 GN (General Node) である場合に RS を経由した通信を行う。直接通信が行えない場合とは、両エンド端末が異なる NAT 配下に存在する場合や、一方が IPv4 ネットワーク、もう一方が IPv6 ネットワークに存在する場合がこれに相当する。RS を経由する場合でも、複数の RS の中から 1 つを選択することで、冗長の少ない通信経路で通信を行うことができる。

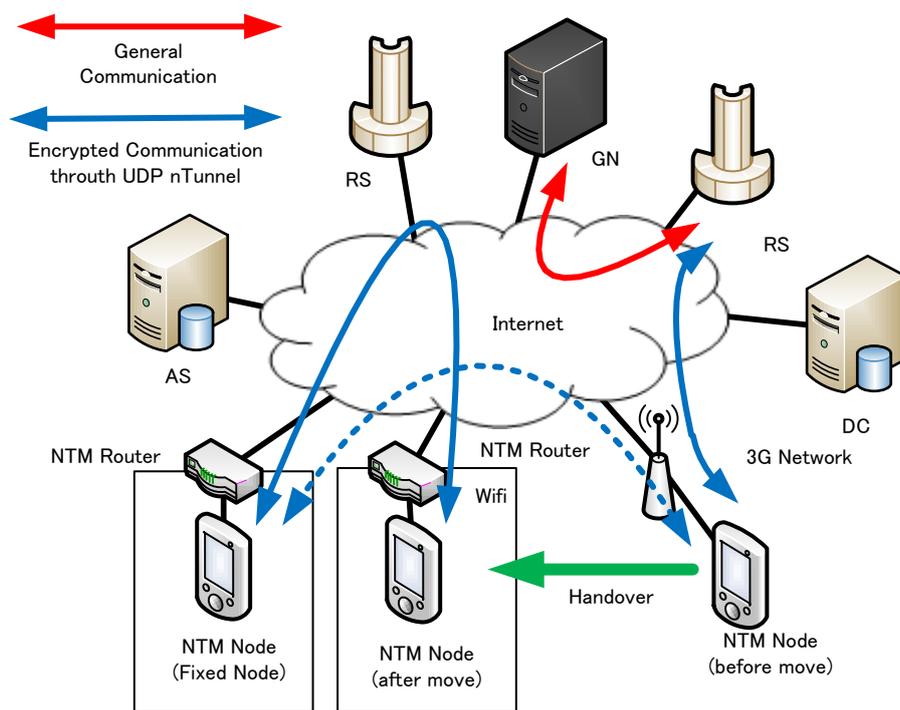


図3 NTMobile のシステム構成

## 2.2.2 NTMobile 通信開始時の動作

図4にNTMobile通信開始時の動作シーケンスを示す。MN及びCN（Correspondent Node）は、NTM端末である。また、説明の省略化のために、NATの存在は省略している。通信開始時にMNは、CNのFQDNを指定して名前解決要求を行う。NTMfwはCNとNTMobileのシグナリング処理を行い、MNとCN間でNTMobileのシグナリング処理を行う。また、NTMfwはシグナリング中のやり取りの中でCNの仮想IPアドレスを知ることができる。シグナリング処理終了後にNTMfwは、名前解決要求の応答として、CNの仮想IPアドレスを返す。MNは仮想IPアドレスを通信相手と認識し、仮想IPアドレス宛にパケットを送信する。このパケットは、NTMobileの機能によりカプセル化されCNへ送信される。MNとCN間で既にトンネル経路が構築されており、CNからMNへNTMobileの機能によるカプセル化されたパケットが送信された場合は、MNはカプセル化されたパケットを受信後に、NTMobileの機能により当該パケットをデカプセル化して、アプリケーションにデータを渡す。

## 2.2.3 NTMfw の概要

NTMfwは、BSDソケットAPIと互換性のあるNTMソケットAPIを提供する通信ライブラリであり、NTMfwを利用してプログラムを記述することで、NTMobileの機能を利用することができる。NTMfwはTCP/IPプロトコルスタックの機能を持ち、TCP/IPプロトコルスタックの機能により生成した仮想IPアドレスに基づくパケットを、カーネル内部で実IPアドレスでカプセル化し

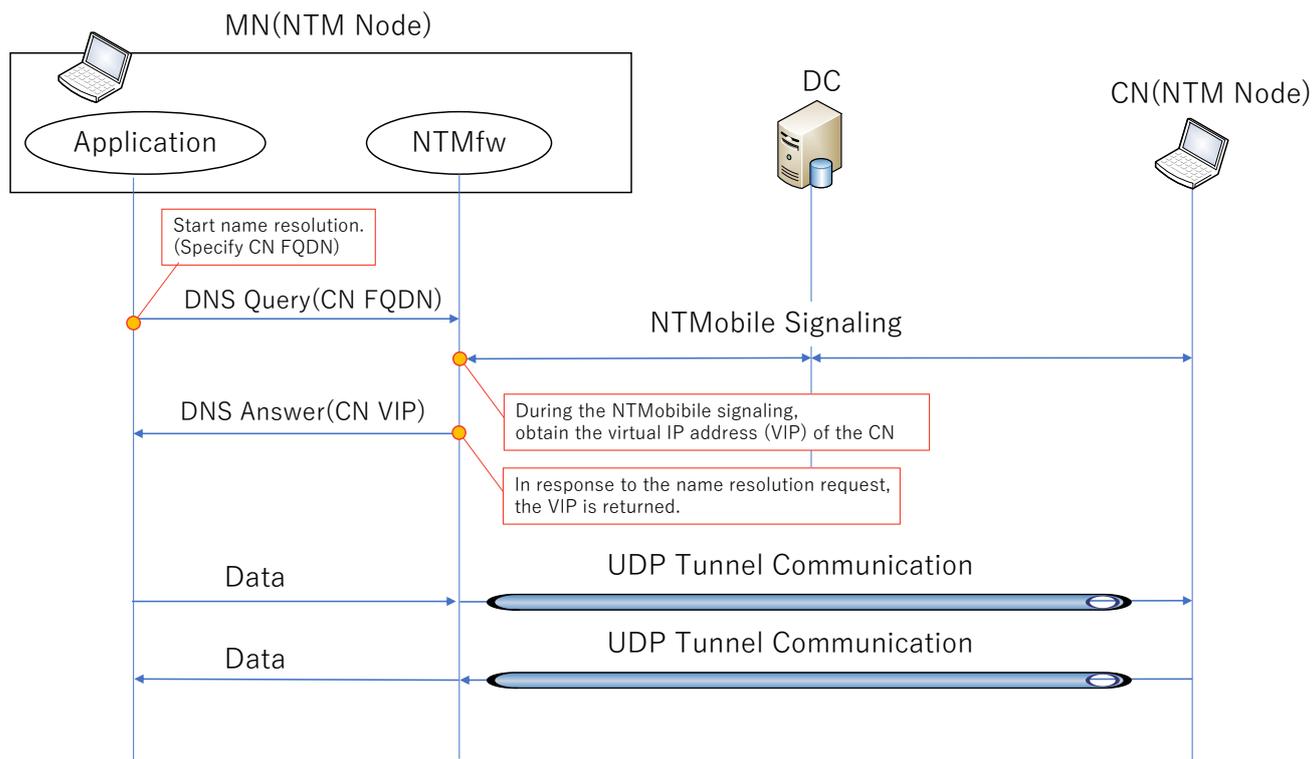


図4 NTMobileの通信開始時の動作シーケンス

て通信相手に送信する。

NTMfwのモジュール構成を図5に示す。NTMfwは次のモジュールから構成される。

- **NTM Socket API**  
BSDソケットAPIに代わってアプリケーションに提供するソケットAPIである。NTMソケットAPIの名称は、例えばntmfw\_sendtoのように、BSDソケットAPIの名称の前にntmfw\_を付与し、機能的にはBSDソケットと互換性を持つ。
- **Negotiation Module**  
NTMobileの初期化処理とシグナリング処理を行うモジュールである。NTMobileの制御メッセージの処理や、アドレス情報の監視を行う。NTMソケットAPIの名前解決を担う関数が呼び出された場合や、他端末から通信要求があった場合、このモジュールでトンネル構築処理が行われ、トンネルテーブルが更新される。
- **Packet Manipulation Module**  
通信パケット、およびシグナリングパケットの生成、解析処理を行う。通信パケットに対しては、NTMobileヘッダの付与、暗号化/復号処理、MAC (Message Authentication Code) 認証処理を実行する。BSDソケットAPIを用いて実アドレスによるパケットの送受信を行う。
- **Virtual TCP/IP Protocol Stack**  
上位アプリケーションが送受信するデータのTCP/IP処理を実行する。TCP/IPプロトコルスタックとしてlwIP (A Lightweight TCP/IP stack) が利用されている。

- Tunnel Table

通信相手ごとに FQDN, 仮想 IPv4/IPv6 アドレス, 実 IPv4/IPv6 アドレス等をメンバとするエントリを持つ。

### 2.2.4 NTMfw の動作

NTMfw のカプセル化処理に着目した動作シーケンスを図 6 に示す。アプリケーションが送信したデータは TCP/IP プロトコルスタックに渡され、TCP/UDP ヘッダ及び仮想 IP アドレスに基づく IP ヘッダが付与される。その後、パケット操作機能により NTMobile 通信であることを示す NTM ヘッダが付与され、暗号化、MAC 付与等の処理を経て、OS に処理が渡される。OS では、上記パケットを UDP でカプセル化する。これら一連の処理により、パケットはトンネル経路を経由して相手端末に届けられる。パケットの受信処理は、上記と逆の手順により実現される。

NTMfw は DNS 要求をトリガとして、NTMobile のシグナリング処理も実行する。アプリケーションは、NTMobile のシグナリング動作を意識する必要はない。また、NTMfw は通信中に自らの IP アドレスを監視している。IP アドレスの変化を検出すると、これをネットワークが切り替わったものとして認識し、DC との間で再度シグナリングを実行する。この動作により実 IP アドレスが変化した場合においても、アプリケーションは実 IP アドレスの変化に気づくことなく通信が継続される。

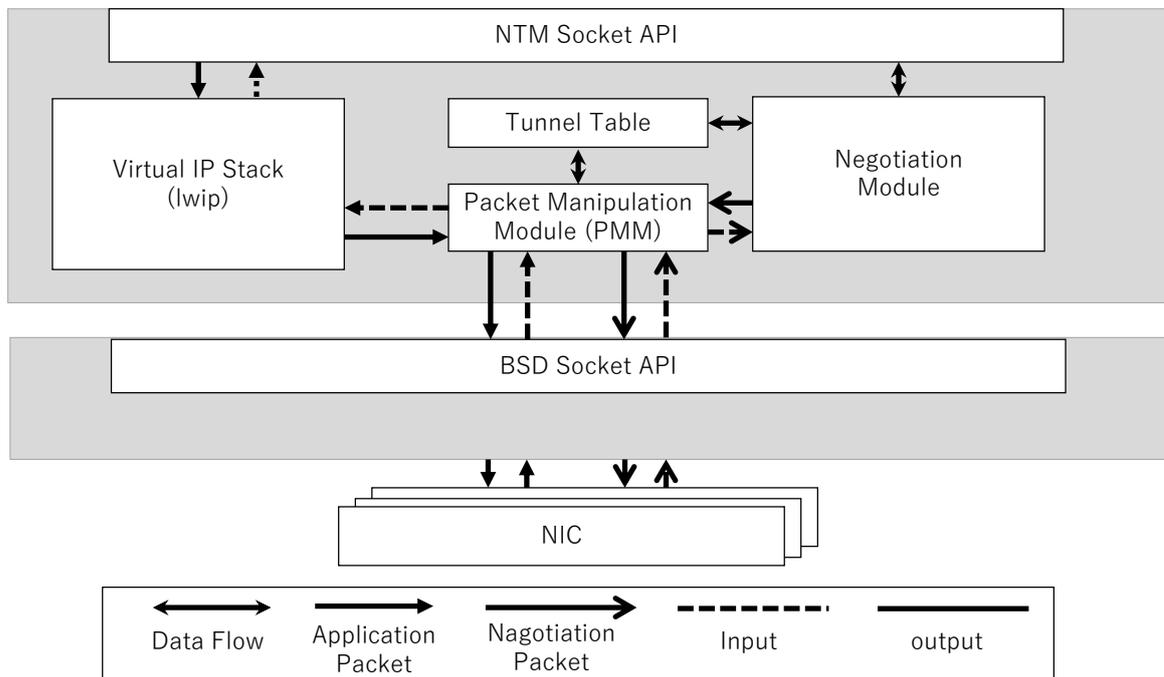


図 5 NTMfw のモジュール構成図

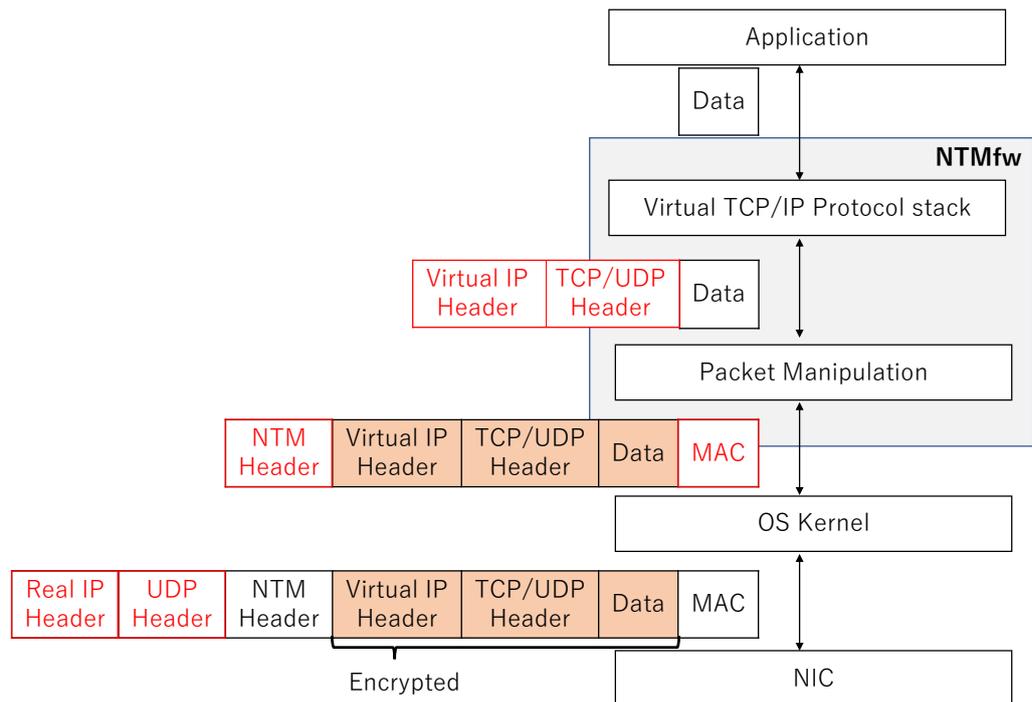


図 6 NTMfw のカプセル化通信の実現方法

## 2.2.5 課題

ユーザが NTMobile の機能を使いたい場合は、NTM ソケット API を使って新規にアプリケーションを開発するか、既存のアプリケーションの BSD ソケット API を NTM ソケット API に書き換える必要がある。アプリケーションを新規に開発するには、仕様の検討や実装など、それなりの時間を要する。既存のアプリケーションを修正する場合、一般通信が必須とされる部分があると、プログラムの処理を詳しく調査する必要があり、単純なソケット API の書き換えでは、仕様を満たすことができない可能性がある。また、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。

## 第3章 TUN 利用型 NTMobile

本章では、アプリケーションが送信したパケットをカーネル空間からユーザ空間に取り込む処理を TUN で行い、TUN から取り込んだユーザ空間の IP パケットをデータとみなして、カーネル空間でカプセル化を行うように NTMfw を改造した R-NTMfw を用いることで、アプリケーションが一般の通信を利用して NTMobile の機能を利用できる TUN 利用型 NTMobile を提案する。ユーザは、端末に TUN 利用型 NTMobile をインストールする作業のみで利用可能であり、アプリケーションが指定する通信相手によって、NTMobile/一般通信を利用するか選択できる。

### 3.1 TUN の概要

TUN とは、オープンソースの仮想インタフェースであり、IP パケットをユーザ空間へ取り込むことができるため、VPN (Virtual Private Network) アプリケーションのカプセル化処理を実装する場合に広く利用される。ユーザ空間にパケットをフックしたい場合は、当該パケットを適切に TUN インタフェースにルーティングされるように設定する必要がある。

### 3.2 R-NTMfw の概要

R-NTMfw は、NTMfw が持つ仮想 TCP/IP プロトコルスタックの機能を除去した通信ライブラリである。アプリケーションが仮想 IP パケットをデータとして利用可能である場合は、R-NTMfw を利用して NTMobile の機能を持つアプリケーションを開発できる [19]。

R-NTMfw のモジュール構成を図 7 を示す。また、R-NTMfw におけるカプセル化処理に着目した動作シーケンスを図 8 に示す。アプリケーションが送信した仮想 IP パケットはパケット操作機能に渡され、NTMobile 通信であることを示す NTM ヘッダ、暗号化処理、MAC 付与等の処理を経て、OS に処理が渡される。OS では、上記パケットを UDP でカプセル化する。これら一連の処理により、パケットはトンネル経路を経由して相手端末に届けられる。パケットの受信処理は、上記と逆の手順により実現される。

### 3.3 提案方式のカプセル化処理に着目した動作概要

図 9 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 インタフェースから読み込み、NTMobile の機能により NTM

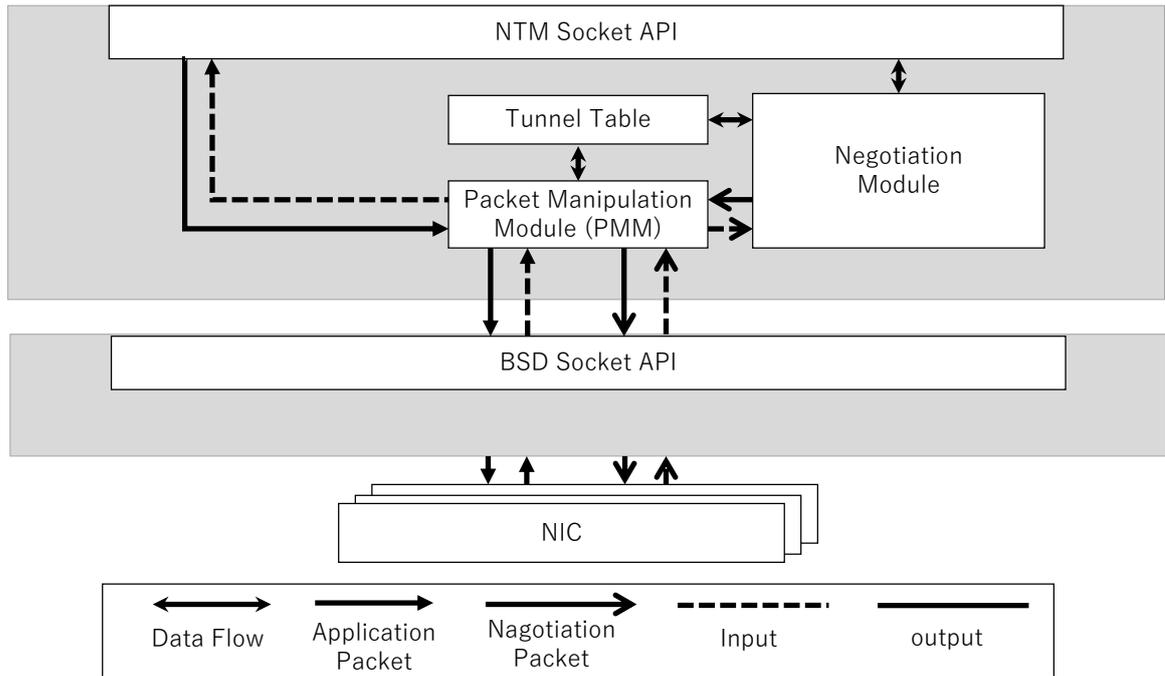


図 7 R-NTMfw のモジュール構成図

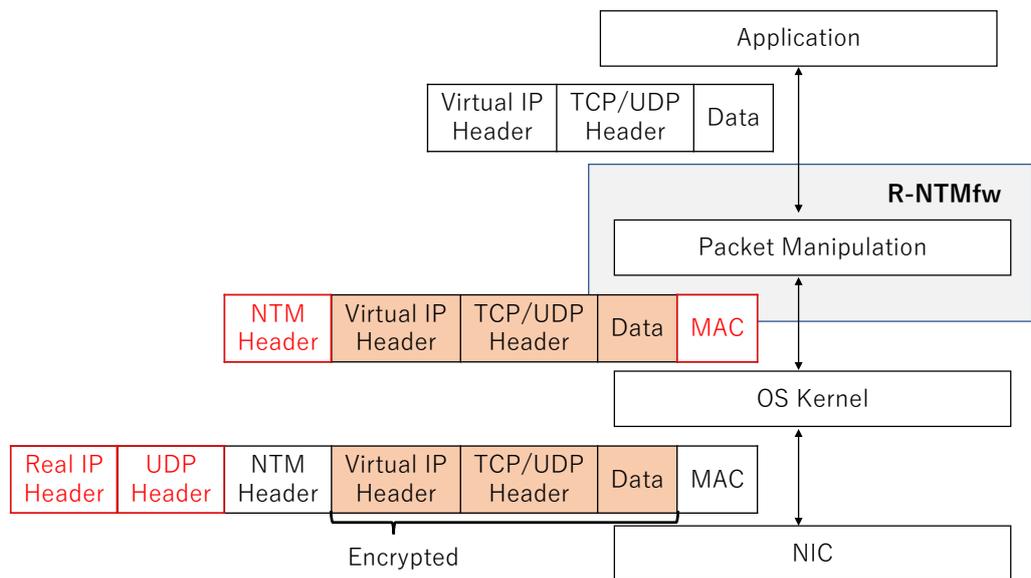


図 8 R-NTMfw のカプセル化通信の実現方法

ヘッダおよび MAC が付与される。その後、実インタ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。以上の動作により、アプリケーションのプログラムに対して、一切の変更をすることなく NTMobile の機能を利用することができる。

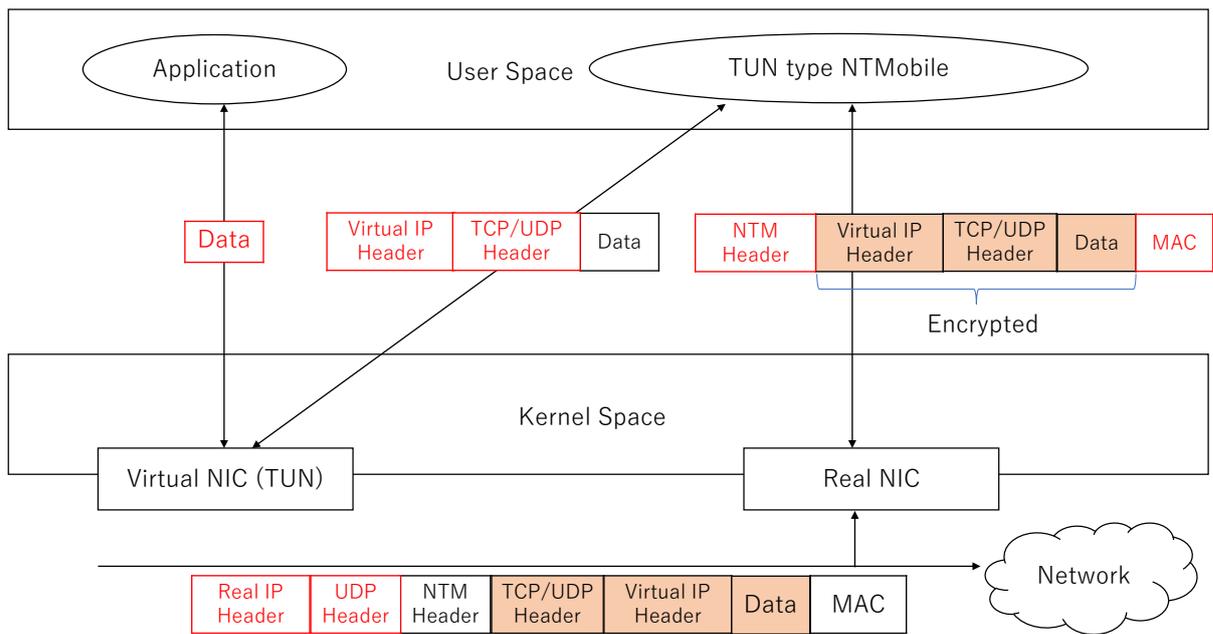


図9 TUN 利用型 NTMobile におけるカプセル化の様子

### 3.4 TUN 利用型 NTMobile の実現に向けた検討と実装方針

本節では、TUN 利用型 NTMobile の実現に向けて行った検討と実装の方針について述べる。

#### 3.4.1 TUN に割り当てる IP アドレス

TUN を利用してユーザ空間にパケットを取り込むには、当該パケットを適切に TUN にルーティングする必要がある。TUN 利用型 NTMobile では TUN インターフェースに仮想 IP アドレスを割り当てて利用する。これにより、NTMobile 通信を利用するアプリケーションが送信する仮想 IP アドレス宛の IP パケットは、全て TUN インターフェースにルーティングされる。

#### 3.4.2 BSD ソケット API による NTMobile シグナリングの開始

NTMobile では名前解決要求を NTMobile シグナリング処理のトリガとして用いている。BSD ソケット API により開発されたアプリケーションが名前解決要求を行う時に、C 言語においては `getaddrinfo` 関数が利用される [20]。当該関数を利用すると、OS 内部で名前解決要求パケットが構築され DNS サーバーと DNS プロトコルによる名前解決処理を行う。TUN 利用型 NTMobile では、この動作に着目し、名前解決要求パケットを通信相手との NTMobile シグナリングのトリガとする。また、NTMobile シグナリング処理の終了後に通信相手の仮想 IP アドレスで名前解決応答を行うようにする。

### 3.4.3 DNS パケットをルーティングするための仮想 IP アドレスの導入

名前解決要求の宛先がデフォルトゲートウェイである場合に、デフォルトゲートウェイ宛のパケットを全て TUN インターフェースにルーティングする設定を行うと、TUN 利用型 NTMobile が送信するデフォルトゲートウェイ宛のパケットが TUN インタフェースにループバックしてしまう。DNS プロトコルで利用される 53 番ポートを TUN インタフェースにルーティングする設定も考えられるが、TUN 利用型 NTMobile が行う名前解決要求により送信される DNS パケットも同様に、TUN インタフェースにループバックしてしまう。

そこで、名前解決要求の宛先の一つに仮想 IP アドレスを持った仮想的な DNS サーバーを追加するように OS に対して設定を行う。これにより、名前解決要求により生成される DNS パケットの宛先が仮想 IP アドレスになるため、TUN インタフェースにルーティングされる。また、NTMobile における仮想 IP アドレスの名前解決を行う役割がある DC とは、実 IP アドレスを利用して通信を行うため、パケットのルーティングにおける課題は起こらない。

### 3.4.4 名前解決要求の宛先として用いる仮想 IPv4 アドレスの定義

NTMobile では、仮想 IPv4 アドレスとして、IANA (Internet Assigned Number Authority) で規定された「198.18.0.0/15」[21] のアドレス帯域を利用している。この帯域を、DC 同士が相互に管理し、全ての NTM 端末の仮想 IP アドレスが重複しないように割り当てている。このため、DNS サーバーに割り当てる仮想 IPv4 アドレスは固定化し、DC が NTM 端末に割り当てる仮想 IP アドレスの範囲から除去する必要がある。また、固定化した DNS サーバーの仮想 IPv4 アドレスを全ての TUN 利用型 NTMobile で利用し、仮想 IPv4 アドレスの消費を抑える必要がある。

NTMobile では、以下の固定化されたアドレスが予約されている。NTMobile を利用するアプリケーションは、「198.18.0.0」を仮想 IP ネットワークのネットワークアドレスとして認識する。文献 [22] では、「198.18.0.1」を予約アドレスとすることを提案している。このことから、DNS サーバーに割り当てる仮想 IPv4 アドレスは、「198.18.0.2」として固定化する。

### 3.4.5 任意の DNS サーバーを利用できることを考慮した名前解決要求の宛先の設定

OS に設定を行う時に、仮想 IP アドレスを持った仮想的な DNS サーバーを優先サーバーとして動作するようにし、ユーザが利用したい DNS サーバーは代替サーバーとして動作するようにする。この設定により、ユーザが任意の DNS サーバーを用いた名前解決処理も可能となる。

### 3.4.6 IP フラグメンテーションの課題解決

ネットワークの規格ごとに 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、一般的にエンドユーザが接続する Ethernet 規格における MTU は 1500Byte である [23]。また、パケット長が MTU を超えると IP プロトコルにより、パケットのフラグメンテーションが発生し、通信効率が格段に低下する。また、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ーによりパケット長が MTU を超えてしまう課題がある [24]。

そこで、TUN 利用型 NTMobile では TUN インターフェースの MTU に、Ethernet 規格の MTU から NTMobile が付与するヘッダーのサイズを引いた値を割り当てる。これにより、TUN インターフェースから受信した仮想 IP パケットの長さは、全て MTU 未満となり、NTMobile のカプセル化が行われたとしても、フラグメンテーションが発生することがなくなる。

表 1 に NTMobile が付与するヘッダーのサイズを示す。NTMobile が仮想 IP パケットに対して、カプセル化したときに NTMobile が付与するパケット長の最大長は、ネットワーク層が IPv6 の場合であり、その長さは 100Byte である。つまり、TUN インターフェースには MTU として 1400Byte を割り当てる。

表 1 NTMobile が付与するヘッダーサイズ

レイヤ	区分	バッファ長 (Byte)
ネットワーク層	IPv4	20
	IPv6	40
トランスポート層	UDP	8
NTMobile プロトコル層	NTM	36
	HMAC	16
	ヘッダー長の合計 (IPv4 は除く)	100

## 3.5 TUN 利用型 NTMobile の動作概要

### 3.5.1 起動時の処理

TUN 利用型 NTMobile を起動時に NTMfw の機能を利用して AS に対してログイン処理を行い、アカウント認証を行う。また、DC に対してアドレス情報の登録処理を行い、その応答として DC から仮想 IP アドレスの割り当てを受ける。NTMobile サーバー群に対しての一連の処理後に TUN 利用型 NTMobile は、TUN インターフェースを作成し、TUN インターフェースの初期化処理を行う。その際に TUN インターフェースの IP アドレスには、DC から割りてられた仮想 IP アドレスを割り当て、MTU には、3.4.6 節で述べた値を割り当てる。また、TUN インターフェースに DNS パケットがルーティングされるように、3.4.4 節で述べた DNS サーバーを DNS リゾルバに追加登録する。

### 3.5.2 NTMobile 通信開始時の動作

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 端末固有の 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。NTM 端末固有の FQDN は、「\*.ntm.\*」で表現され、この文字列が FQDN の中に存在する場合は、通信相手を NTM 端末であると判断する。解析した結果、NTM 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、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フェースに処理を渡す。実インタフェースでは、UDP プロトコル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でアプリケーションにデータを渡す。

### 3.5.3 代替 DNS サーバーによる一般端末との通信開始時の動作

図 10 に TUN 利用型 NTMobile が起動状態時に代替 DNS サーバーを利用したときの一般通信の動作シーケンスを示す。アプリケーションは、通信開始時に一般端末の FQDN を指定して DNS クエリを送信する。最初の DNS クエリは優先 DNS サーバー宛に送信されるので、TUN 利用型 NTMobile は、このパケットを TUN インターフェースから読み込む。その後、DNS クエリパケットに記載されている FQDN を解析する。解析した結果、一般端末の FQDN である場合は、TUN 利用型 NTMobile はこのパケットを破棄する。アプリケーションは、優先 DNS サーバーから応答が返ってこないため、タイムアウト処理をはじめ、代替 DNS サーバーに名前解決要求を行う。名前解決に成功して実 IP アドレスを取得した DNS サーバーは、その実 IP アドレスを DNS レスポンスパケットに記載して、アプリケーションに送信する。アプリケーションは、通信相手を実 IP アドレスと認識するので、実 IP アドレス宛にパケットが送信される。

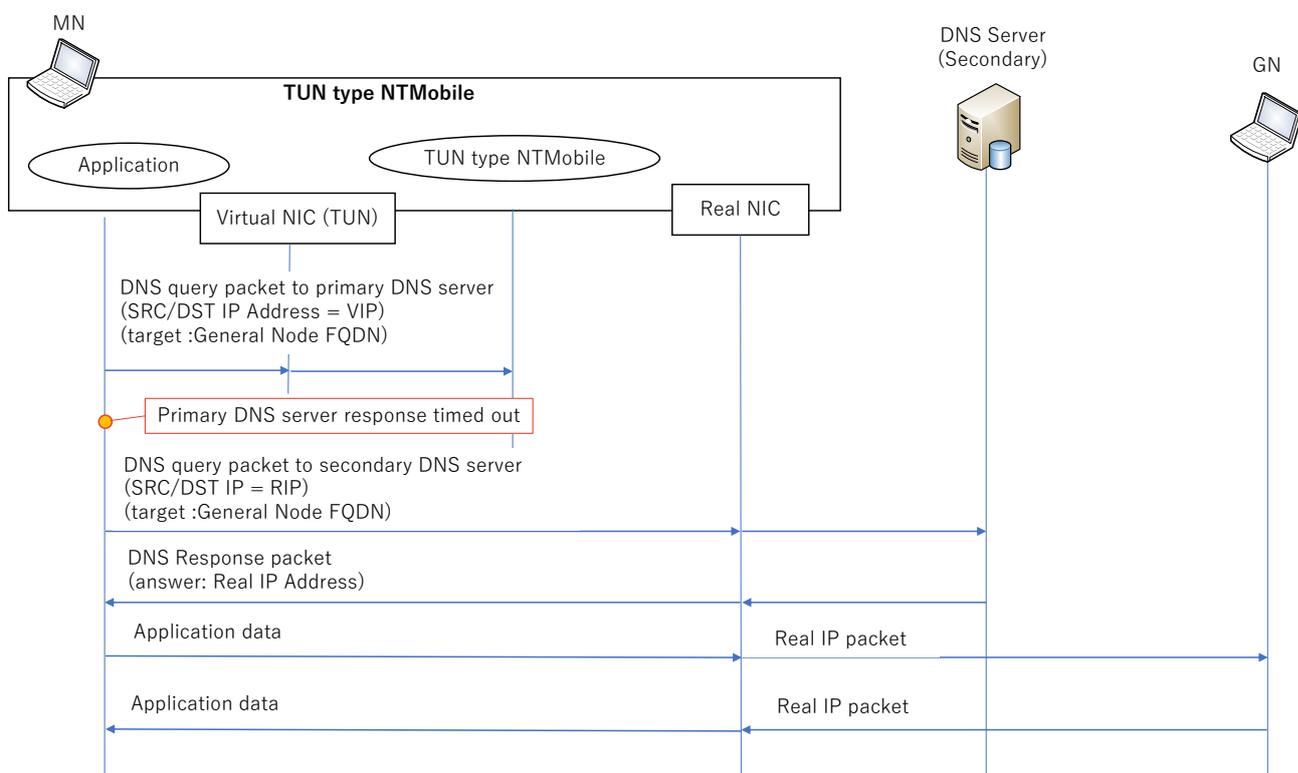


図 10 代替 DNS サーバーを利用したときの一般通信開始時の動作シーケンス

## 第4章 実装

本章では、TUN 利用型 NTMobile の実装について述べる。実装は、Linux 環境にて C 言語で行った。R-NTMfw については、説明の簡単化のため Packet Manipulation Module と Negotiation Module について述べる。

### 4.1 起動時の処理

R-NTMfw の機能により、AS に対してログイン処理を行う。ログイン処理が正常に終了すると、DC から仮想 IP アドレスが割り当てられるので、TUN インターフェスを定義し、仮想 IP アドレスと所定の MTU により初期化を行う。

DNS リゾルバの設定を変更し、TUN インターフェスに DNS パケットがルーティングされるように設定する。DNS リゾルバが問い合わせる DNS サーバーを新たに追加する際の動作について述べる。始めに、DNS リゾルバの設定ファイルを書き換えるため、設定ファイルのコピーファイルを作成する。その後、DNS リゾルバの設定ファイルに、仮想 IP アドレスを持つ DNS サーバーを追加する記述を行う。その際に、追加した DNS サーバーがプライマリ DNS サーバーとして動作するようにする。設定ファイルを編集後に、DNS リゾルバを再起動して設定を反映させる。TUN 利用型 NTMobile が終了しても上記の設定が残るため、元々の設定に戻す必要がある。手順は、以下の通りである。上記でコピーファイルしたファイルを DNS リゾルバの設定ファイルに置き換える。その後、DNS リゾルバを再起動して設定を反映させる。

### 4.2 モジュール構成

TUN 利用型 NTMobile のモジュール構成を図 11 に示す。TUN 利用型 NTMobile は、次のモジュールから構成される。

- Packet Checker

IP パケットを解析して次の処理を決める機能である。アプリケーションからパケットチェッカーに渡されるパケットは、通信用/DNS パケットの 2 通りである。通信用パケットの場合は、通信用パケットをパケット操作モジュールに処理を渡してトンネル通信を行う。DNS クエリパケットの場合は、FQDN を解析する。NTM 端末固有の FQDN である場合は、NTMobile シグナリングモジュールにより NTMobile シグナリング処理を行う。それと同時に、DNS レスポンスパケットジェネレーターに DNS クエリパケットを渡す。一般端末の FQDN である場合は、DNS クエリパケットを破棄する。

- DNS Response Packet Generator

DNS レスポンスパケットを生成する機能である。NTMobile シグナリング処理終了後に、仮想 IP アドレスが DNS レスポンスパケットジェネレーターに渡される。その仮想 IP アドレスを用いて、パケットチェッカーから渡された DNS クエリパケットを基に、DNS レスポンスパケットを生成する。その後、DNS レスポンスパケットをパケットフォワーダーに渡す。

- Packet Forwarder

アプリケーションに IP パケットを転送する機能である。このモジュールに渡された IP パケットを TUN インターフェースに書き込むことで、アプリケーションにデータを渡す。

- Packet Manipulation Module

通信パケット、およびシグナリングパケットの生成、解析を行う。通信パケットに対しては、NTM ヘッダの付与、暗号化/復号処理、MAC 認証処理を実行する。BSD Socket API を用いて実 IP アドレスによるパケットの送受信を行う。

- Negotiation Module

NTMobile の初期化処理とシグナリング処理（通信開始時および移動時のトンネル生成）を行う。また、端末の実 IP アドレスの変化を検出すると、ネットワークが切り替わったものと判断し、移動処理に係わるシグナリングを開始し、新たなトンネル経路を構築する。

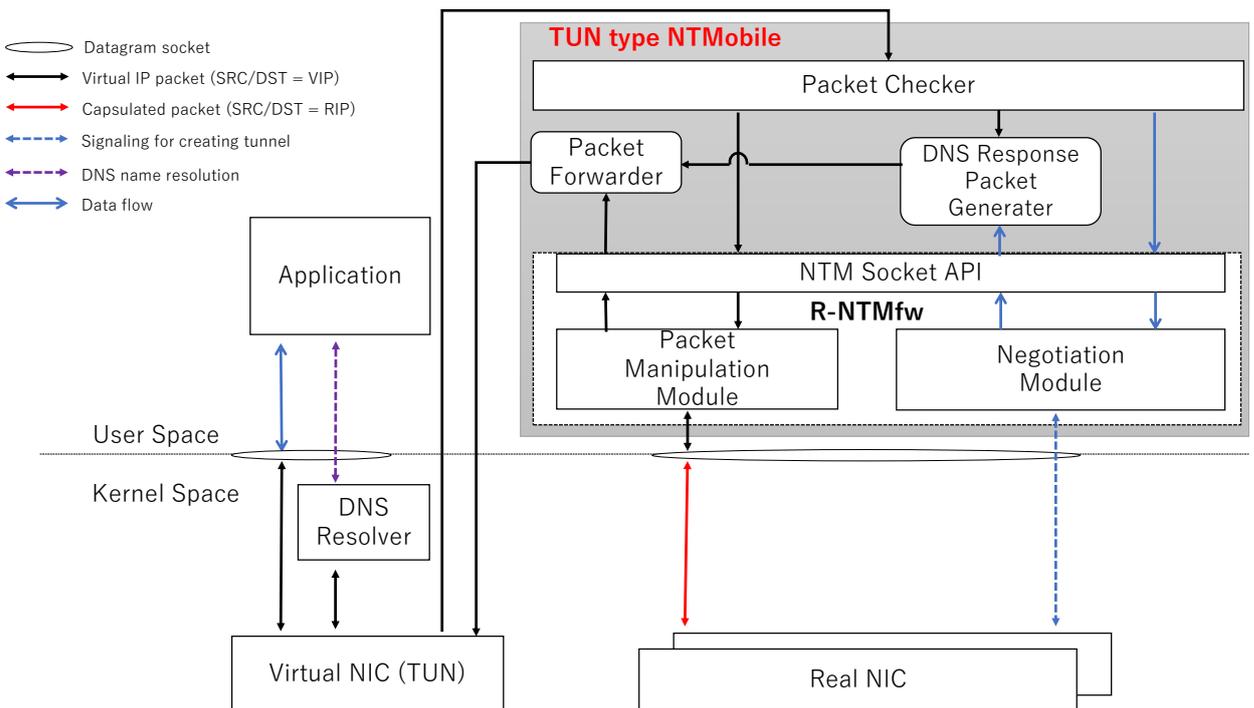


図 11 TUN 利用型 NTMobile のモジュール構成図

## 第5章 動作検証と評価

本章では、4章で示した設計に基づいて構成された TUN 利用型 NTMobile の動作検証および評価を行った。

### 5.1 NTMobile 通信の動作検証

実装した TUN 利用型 NTMobile が通信接続性と移動透過性の要件を満たすかを検証した。

#### 5.1.1 通信接続性試験

MN から CN に対して PING を行うことで、グローバルネットワーク側からプライベートネットワーク側へ通信を開始する通信接続性試験を行った。

試験のネットワーク構成を図 12 に示す。また、試験に用いた MN と CN の諸元を表 2 に、AS, DC, RS の諸元を表 3 に示す。MN と CN は、2 台のラップトップを用意し、TUN 利用型 NTMobile を設置した。また、TUN 利用型 NTMobile は、名前解決要求の宛先の DNS サーバーとして、仮想 IP アドレスが割り当てられた仮想的な DNS サーバーを認識している。AS, DC, RS は、一つのホストマシン上に VMware Workstation Player を用いて仮想マシンとして構築した。有線 LAN ブロードバンドルーターである LAN-BR/4 [25] を NAT として二台用意した。以降、2 台の NAT をそれぞれ NAT1 と NAT2 と呼ぶ。AS, DC, RS, 及び NAT1, NAT2 をそれぞれ IPv4 ネットワークに接続した。その後、NAT1 に MN, NAT2 に CN をそれぞれ接続した。最初、MN が NAT1 に CN が NAT2 にそれぞれ接続されているが、MN と CN が接続されるネットワークの組み合わせに応じて、それぞれの接続先を変更して試験を行った。

試験の結果より、IPv4 ネットワークにおいて、端末が接続されているネットワークに依存せずにデータを送受信することができた。

表 2 MN と CN の諸元

	MN/CN
OS	Ubuntu 14.04 (32 bit)
CPU	Intel Core i5 (2.50 GHz)
Memory	2 GB

表 3 AS, DC, RS の諸元

	Host Machine	Virtual Machine (AS, DC, RS)
OS	Windows 10 (64 bit)	Ubuntu14.04 (32 bit)
CPU	Intel Core i7 (3.50 GHz)	allocate 2core2threads
Memory	16 GB	allocate 2 GB

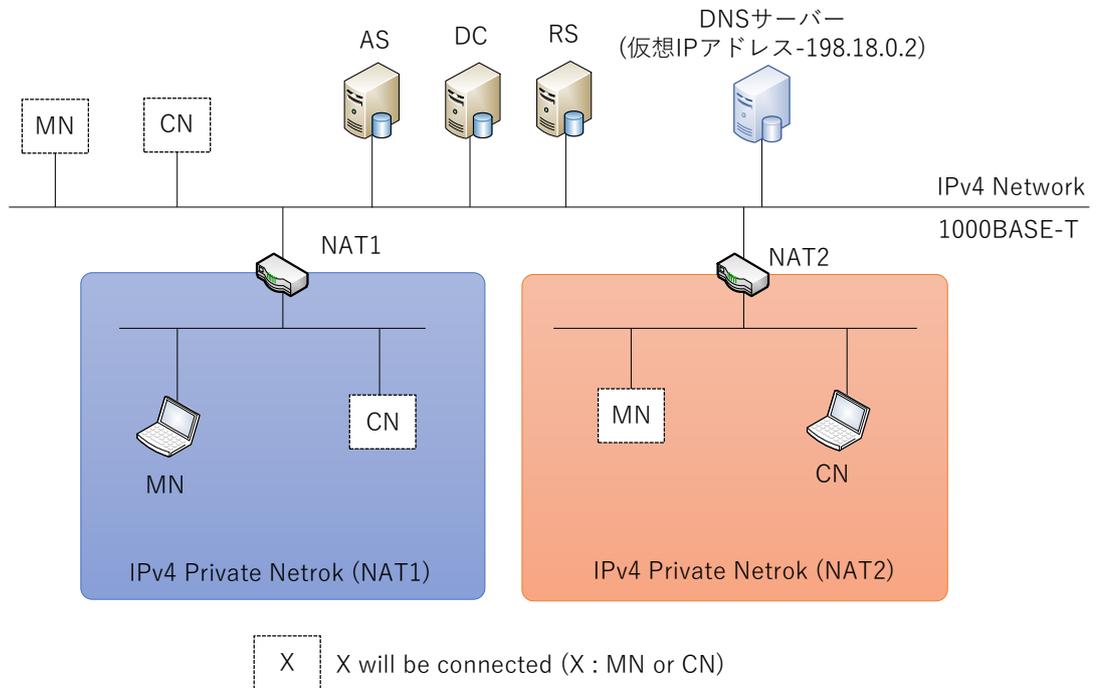


図 12 通信接続性試験のネットワーク構成

表 4 通信接続性試験の結果

		CN		
		IPv4 Private Network 1	IPv4 Private Network 2	IPv4 Network
MN	接続先			
	IPv4 Private Network 1	◎	○	◎
	IPv4 Private Network 2	○	◎	◎
	IPv4 Network	◎	◎	◎

◎: end-to-end, ○: via RS

### 5.1.2 移動透過性試験

MN と CN 間で通信中に、MN のネットワークを切り替えても通信を継続することができる移動透過性を確認した。試験したネットワーク構成と使用した機器は、5.1.1 項の NAT 越え試験と同一である。

MN は CN の FQDN を指定して PING を開始した。その後 MN を現在接続されているネットワークとは異なるネットワークに接続した。

試験の結果より、IPv4 ネットワークにおいて端末をどのようなネットワークに移動しても通信

を継続することができた。

## 5.2 スループットによる性能評価

TUN 利用型 NTMobile のスループットによる性能評価について述べる。性能評価では TUN 利用型 NTMobile を利用した時のスループットと、TUN 利用型 NTMobile と NTMfw のプログラム内の処理時間により評価を行った。

### 5.2.1 評価構成

図 13 に性能評価に用いるネットワーク構成を示す。MN および CN は、端末として Raspberry PI3 Model B を用いて、それぞれに TUN 利用型 NTMobile をインストールした。MN と CN の諸元を表 5 に示す。スイッチングハブは、NETGEAR 製の GS105E-200JPS [26] を使用した。AS, DC, RS は、5.1.1 項の NAT 越え試験と同一である。これらの MN, CN, デスクトップ PC をスイッチングハブで繋ぎ、IPv4 ネットワークを構築した。

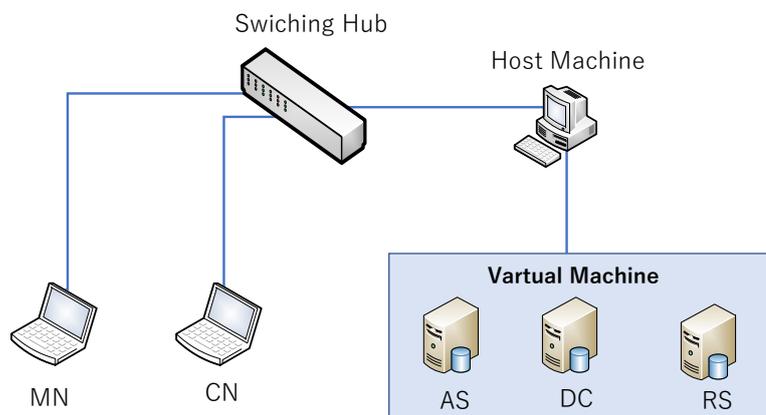


図 13 性能評価のネットワーク構成

表 5 MN と CN の諸元

	MN / CN
OS	Linux raspberrypi 3.18 (32 bit)
CPU	ARMv8 Cortex-A53 (2.50 GHz)
Memory	1 GB

### 5.2.2 スループットの測定

アプリケーションが TUN 利用型 NTMobile を利用したときに、実用上問題ない性能で通信を行うことができるかを確認するために、TUN 利用型 NTMobile を利用した NTMobile 通信と、TUN 利用型 NTMobile を利用しない時の一般通信のスループットを比較した。

図 13 のネットワーク構成のもとで、MN から CN へ向けて、ネットワークの性能測定に用いられる iPerf [27] を利用し、TCP プロトコルによるスループットの測定を行った。試験区分は、TUN 利用型 NTMobile を利用した NTMobile 通信、MTU が 1400 または 1500 の時の一般通信の 3 つとした。TUN 利用型 NTMobile の TUN インタフェースには、1400 の MTU が割り当てられている。また、一般通信の場合は、実インタフェースに対して Ethernet 規格である 1500 の MTU と、TUN 利用型 NTMobile が持つ MTU による NTMobile 通信を比較するための 1400 の MTU を割り当て、それぞれ iPerf による通信を行った。

スループットの測定結果を表 6 に示す。一般通信において MTU を 1500 から 1400 に変更することで低下するスループットは 0.48% であり、同等の性能であることが分かった。TUN 利用型 NTMobile を利用した NTMobile 通信は、MTU が 1400 の一般通信よりもスループットが 22.18% 低下することが分かった。

表 6 iPerf によるスループットの測定結果

区分	MTU	時間 (Mbits/s)
NTMobile 通信	1400	72.96
一般通信	1400	93.75
	1500	94.20

### 5.2.3 処理時間の測定

提案方式を利用することにより低下したスループットの原因を調べるために、5.2.2 項と同じネットワーク構成において、iPerf が送信するパケット長が 1400 バイトになるように設定し、TUN 利用型 NTMobile がパケットの処理に要する処理時間を計測した。MN の評価項目を図 14 に示す。また、CN の評価項目を図 15 に示す。ここで、カーネル内部の処理はユーザ空間の処理に比べて非常に高速であるため、計測時間の範囲から外している。

MN におけるパケットチェッカー処理時間は、仮想 IP パケットを TUN インタフェースから読み込み、NTMfw に処理を渡すのに要する時間である。NTMfw 処理時間は、仮想 IP パケットに対して暗号化、MAC 付与などに要する時間である。

CN における NTMfw 処理時間は、パケットの復号処理、MAC 検証を行い、取り出した仮想 IP パケットを TUN 利用型 NTMobile に渡すのに要する時間である。パケットフォワーダー処理時間は、NTMfw から仮想 IP パケットを受け取り、TUN インタフェースに書き込むのに要する時間である。

これらのプログラムの処理時間の測定には C 言語で標準に用意されている `clock_gettime` 関数 [28]

を使用し、モニタリング時刻により計測を行う。モニタリング時刻は、単調増加の時間で表現され、プログラム内部の処理時間を計測するのに適した時刻である。図 14 と図 15 にて定義した各処理の開始時と終了時において当該関数を用いて時間を取得し、それらの時間の差分を処理時間とした。

MN 側の評価結果を表 7 に示す。CN 側の評価結果を表 8 に示す。表 7 の結果より、MN 側では暗号化と MAC 生成の処理に、全体の処理時間の 70% を要することがわかった。表 8 の結果より、CN 側では復号処理と MAC 検証の処理に、全体の処理時間の 87% を要することがわかった。

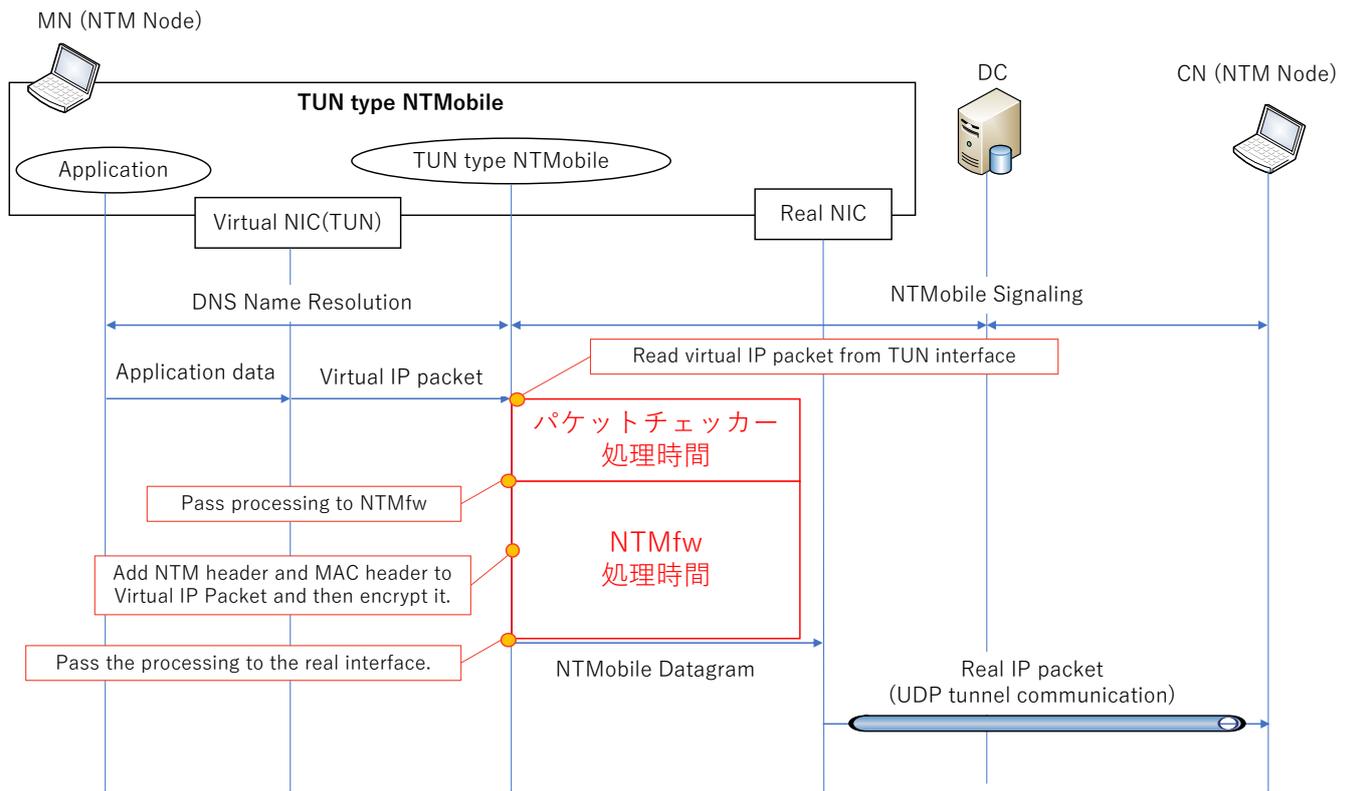


図 14 MN 側の性能評価の項目

表 7 MN の処理時間の結果

区分		処理時間 [ $\mu$ sec]	
パケットチェッカー		15.7	210.0
R-NTMfw	暗号化	116.9	
	MAC 生成	30.5	
	その他	46.9	

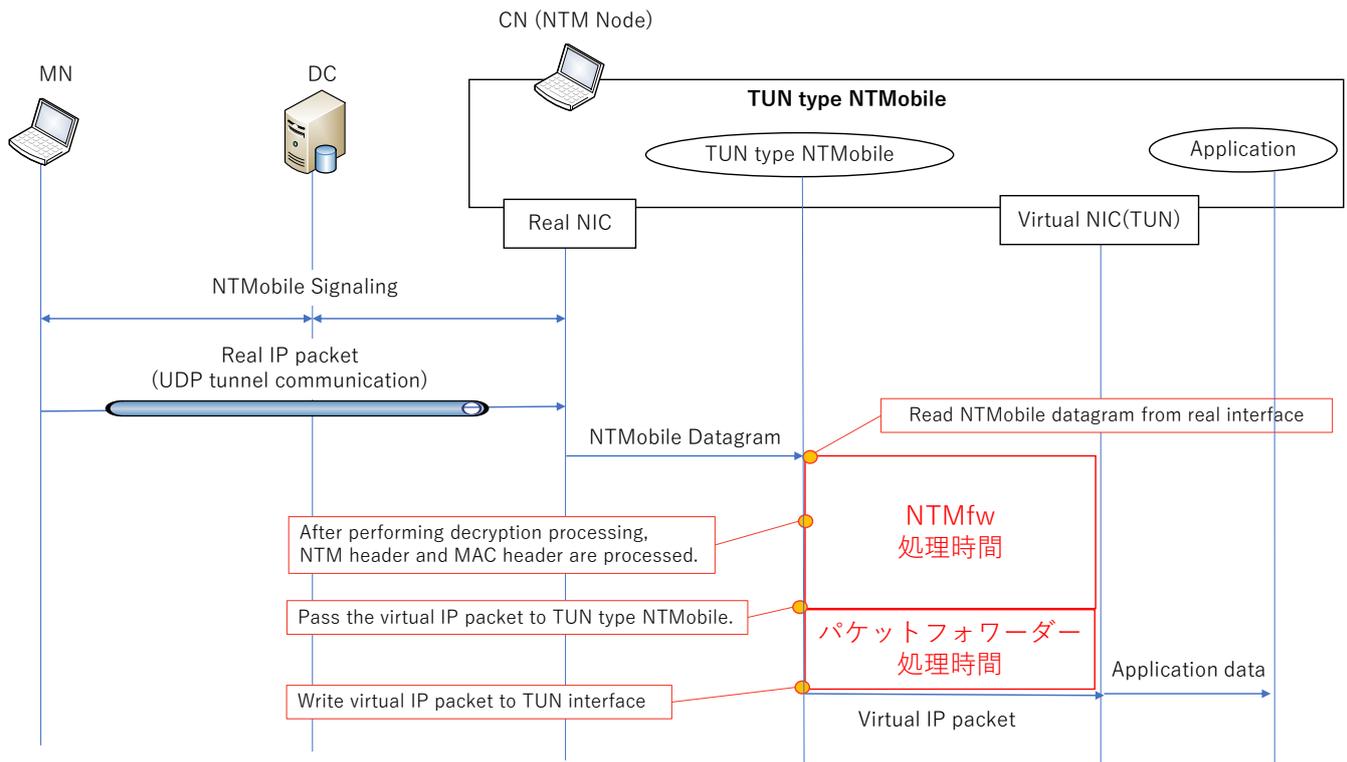


図 15 CN 側の性能評価の項目

表 8 CN の処理時間の結果

区分		処理時間 [ $\mu$ sec]	
パケットフォワーダー		0.6	190.7
R-NTMfw	復号処理	125.4	
	MAC 検証	40.0	
	その他	24.7	

## 5.2.4 考察

5.2.2 項の評価結果により、一般通信と比べて TUN 利用型 NTMobile を経由した通信は 22.18% のスループットの低下がみられた。文献 [11] では、NTMfw の処理時間は、TCP/IP プロトコルスタックの機能および暗号化/復号処理が多く時間を占めることが示されている。TUN 利用型 NTMobile は、NTMfw と比較すると TCP/IP プロトコルスタックの機能をカーネル内部で行うため、処理は非常に高速であると考えられる。しかし、5.2.3 節の評価結果により、引き続き暗号化/復号処理でオーバーヘッドが生じている。また、TUN 利用型 NTMobile はカーネル空間で生成された仮想 IP アドレス宛の packets を TUN インターフェースを経由してユーザ空間に取り込み、再度ソケット通信を通じて、カーネル空間に処理を渡す処理のオーバーヘッドが生じているため、NTMfw と性能を比較するためには、カーネル内部の処理時間を考量した評価を行う必要がある。

スループットの低下の原因は、暗号化/復号処理や MAC 処理であることが分かったが、これらのセキュリティ処理は、通信の完全性や機密性を保証するため必須の処理である。一般的なビデオ

チャットアプリケーションとして Skype [29] に必要とされるスループットを表 9 に示す [30]。TUN 利用型 NTMobile を利用した NTMobile 通信のスループットは 72.96Mbps であり，一般的なアプリケーションに必要とされるスループットを十分に満たす性能で動作できることを確認した。

表 9 Skype における推奨スループット

通話の種類	推奨スループット
通話	100kbps
ビデオ通話	500kbps
グループビデオ通話 (7 人)	8Mbps

### 5.3 代替 DNS サーバーを利用した時に発生するタイムアウト時間の測定

TUN 利用型 NTMobile では，ユーザが任意に利用できる DNS サーバーは代替 DNS サーバーとして設定されているが，アプリケーションが名前解決要求処理を行った時に，優先 DNS サーバーから代替 DNS サーバーに処理が移るまでのタイムアウト時間が発生する。このタイムアウト時間の計測を行った。

試験のネットワーク構成を図 16 に示す。有線 LAN ブロードバンドルータの配下のネットワークは，1000BASE-T の有線 LAN ネットワーク環境である。MN は，ホストマシン上に仮想マシンとして構築し，TUN 利用型 NTMobile を設置および起動した。MN の諸元を表 10 に示す。AS, DC, RS は，ABLENET [31] が提供するクラウドネットワーク上に VPS (Virtual Private Server) として構築した。AS, DC, RS の諸元を表 11 に示す。名前解決要求の宛先として Google の Web サーバー (www.google.co.jp) を利用した。MN には，優先 DNS サーバーとして仮想 IP アドレス (198.18.0.2) が割り当てられた仮想的な DNS サーバーを認識している。また，代替 DNS サーバーは有線 LAN ブロードバンドルータである PR-400MI [32] の DNS 機能を利用する

このネットワーク構成で，タイムアウト時間の計測を行った。最初に MN はパケットキャプチャツールである Wireshark を起動した。その後，Google の Web サーバーの FQDN (www.google.co.jp) を指定して PING を開始した。DNS パケットが優先 DNS サーバーに送信された時刻と代替 DNS サーバーに送信された時刻を記録し，それらの差分をタイムアウト時間とした。上記の計測を 10 回行った平均値を計測結果とする。結果，優先 DNS サーバーから代替サーバーに処理が移るのに要するタイムアウト時間は 5.2 秒であった。

代替サーバーを利用した一般端末との名前解決処理に毎回 5.2 秒のタイムアウト時間を要するため，ユーザが一般通信を利用する時のボトルネックになる可能性がある。このタイムアウト時間の除去については，今後の課題とする。

表 10 MN の諸元

	Host Machine	Virtual Machine (MN, CN)
OS	Windows 10 (64 bit)	Ubuntu16.04 (32 bit)
CPU	Intel Core i7 (3.50 GHz)	allocate 2core2threads
Memory	16 GB	allocate 1 GB

表 11 AS, DC, RS の諸元

	AS, DC	RS
OS	CentOS 6.9 (32 bit)	CentOS 6.9 (32 bit)
CPU	1 Core	2 Core
Memory	512 MB	1536 MB

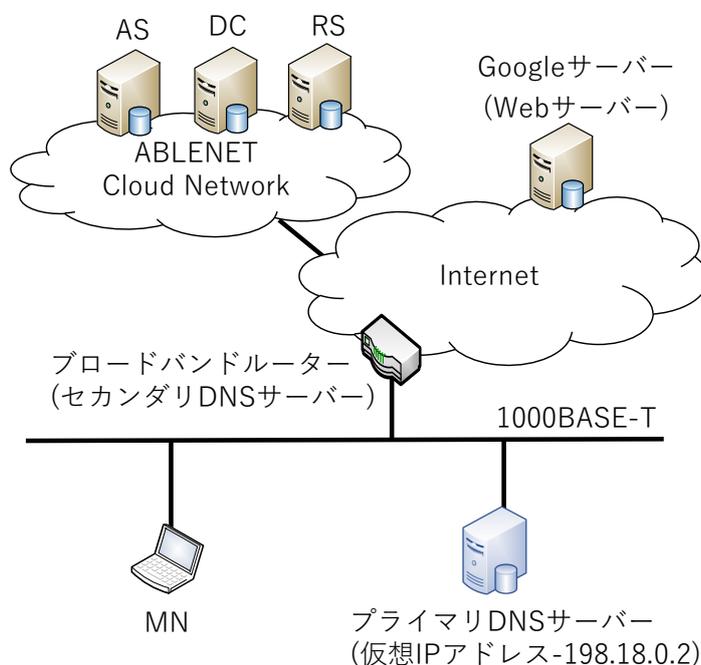


図 16 ネットワーク構成

## 5.4 TUN 利用型 NTMobile の課題

TUN 利用型 NTMobile は、端末にインストールすることのみで NTMobile の機能を利用できるが、端末に TUN ドライバが実装されていない場合は利用することができない。そもそも、組み込み機器のように新規にアプリケーションのインストールができない端末も存在する。

NTMfw は NTMobile の機能を利用したいアプリケーションに対して実装するため、アプリケーション毎に AS によるアカウント認証を行い、NTMobile を利用できる。しかし、提案方式では TUN 利用型 NTMobile が各アプリケーションの代表として AS によるアカウント認証を行い、ア

アプリケーションは名前解決要求を行うことでNTMobileを利用できるため、悪意のあるアプリケーションでも利用できる課題がある。

以上のことから、TUN 利用型 NTMobile をインストールすることが出来ない端末に対しては、端末に接続することで、NTMobile の機能を提供できる仕組みが有用であり検討が必要である。悪意のあるアプリケーションに対しては、NTMobile の仮想 IP ネットワーク上でアプリケーションの通信を制御する仕組みを検討することが必須である。

## 第6章 結論

本論文では、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 TUN 利用型 NTMobile を提案し、実装と評価を行った。動作検証により、アプリケーションのプログラム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、アプリケーションが TUN 利用型 NTMobile を利用した時のスループットの測定を起こった結果、72.96Mbps であることが分かった。このスループットの値は、一般的なビデオチャットアプリケーションをの十分な性能で利用可能であることを示し、一般的な用途でも実用できることを確認した。また、本研究によりユーザは提案方式のアプリケーションをインストールするだけで NTMobile の機能を利用できるため、NTMobile の利用が従来に比べて容易となり、NTMobile の普及への敷居が低くなった。

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。

## 謝辞

本研究を進めるにあたり，多大なるご指導とご教授を賜りました，指導教官である名城大学大学院理工学研究科情報工学専攻 渡邊晃教授に心から感謝いたします。

本研究を進めるにあたり，様々なご指導を頂きました，名城大学大学院理工学研究科情報工学専攻 鈴木秀和准教授に深く感謝いたします。

本研究を進めるにあたり，ご意見並びにご助言を賜りました，愛知工業大学情報科学部情報科学科 内藤克浩准教授に感謝いたします。

本論文を作成するにあたり，快く副査を引き受けていただきました名城大学理工学研究科情報工学専攻 柳田康幸教授に心より感謝いたします。

最後に，本研究を進めるにあたり，数々の有益なご助言を賜りました，渡邊研究室および鈴木研究室の諸氏に感謝いたします。

## 参考文献

- [1] IPv4 アドレスの在庫枯渇に関して - JPNIC. <http://www.nic.ad.jp/ja/ip/ipv4pool/>.
- [2] Egevang, K. B. and Francis, P.: The IP Network Address Translator (NAT), RFC 1631 (1994).
- [3] Index, C. V. N.: Global Mobile Data Traffic Forecast Update, 2016-2021, *Cisco Systems* (2017).
- [4] 岸田重行：Wi-Fi オフロードにまい進する通信事業者，日経コミュニケーション，pp. 84–85 (2011).
- [5] 寺岡文男：インターネットにおけるノード移動透過性プロトコル，電子情報通信学会論文誌 D，Vol. J87-D1, No. 3, pp. 308–328 (2004).
- [6] Soliman, H.: Mobile IPv6 Support for Dual Stack Hosts and Routers, RFC 5555 (2009).
- [7] Moskowitz, R., Heer, T., Jokela, P. and Henderson, T. R.: Host Identity Protocol Version 2 (HIPv2), RFC 7401 (2015).
- [8] 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).
- [9] 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).
- [10] 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).
- [11] 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).
- [12] Universal TUN/TAP driver. <http://vtun.sourceforge.net/tun/>.
- [13] 鈴木秀和，内藤克浩，渡邊 晃：ユーザ空間における移動透過通信技術の設計と実装，マルチメディア，分散，協調とモバイル (DICOMO2014) シンポジウム論文集，Vol. 2014, pp. 1309–1325 (2014).
- [14] 納堂博史，杉原史人，鈴木秀和，内藤克浩，渡邊 晃：NTMobile の実用化に向けた統合的枠組みの検討，第 77 回 MBL・第 63 回 ITS 合同研究会発表，pp. 1–8 (2015).
- [15] Johnson, D. B., Arkko, J. and Perkins, C. E.: Mobility Support in IPv6, RFC 6275 (2011).
- [16] Rosenberg, J.: Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols, RFC 5245 (2010).
- [17] Perkins, C. E.: IP Mobility Support for IPv4, Revised, RFC 5944 (2010).
- [18] torvalds/linux: Linux kernel source tree. <https://github.com/torvalds/linux>.
- [19] 尾久史弥，納堂博史，鈴木秀和，内藤克浩，渡邊 晃：NTMobile の機能を持つアダプタの実現方式の検討，マルチメディア，分散，協調とモバイル (DICOMO2017) シンポジウム (2017).
- [20] The FreeBSD Project. <https://www.freebsd.org/>.
- [21] Bradner, S.: Benchmarking Methodology for Network Interconnect Devices, RFC 2544 (1999).
- [22] 加古将規，鈴木秀和，内藤克浩，渡邊 晃：NTMobile を無限の規模に拡大できる仮想 IPv4 アドレス管理方式の提案，情報処理学会論文誌，Vol. 58, No. 3, pp. 726–735 (2017).

- [23] Hornig, C.: A Standard for the Transmission of IP Datagrams over Ethernet Networks, RFC 894 (1984).
- [24] Savola, P.: MTU and Fragmentation Issues with In-the-Network Tunneling, RFC 4459 (2006).
- [25] LAN-BR/4 - ロジテック株式会社. <http://www.logitec.co.jp/products/lan/lan-br4/index.php>.
- [26] NETGER. <https://www.jp.netgear.com/>.
- [27] iPerf - The ultimate speed test tool for TCP, UDP and SCTP. <https://iperf.fr/>.
- [28] Man page of CLOCKGETERS. [https://linuxjm.osdn.jp/html/LDP\\_man-pages/man2/clock\\_getres.2.html](https://linuxjm.osdn.jp/html/LDP_man-pages/man2/clock_getres.2.html).
- [29] Skype. <https://www.skype.com/ja/>.
- [30] Skype サポート. <https://support.skype.com/ja/faq/FA1417/skype-debi-yao-tonarudai-yu-fu-wojiao-etekudasai>.
- [31] ABLENET VPS. <https://www.ablenet.jp/>.
- [32] 情報機器ナビゲーション. <https://www.ntt-west.co.jp/kiki/download/flets/pr400mi/>.

# 研究業績

## 国際会議（査読あり）

- (1) Fumiya Ogyu, Hidekazu Suzuki, Katsuhiko Naito, Akira Watanabe : Development of End-to-End Communication Adaptor and Implementation, The Tenth International Conference on Mobile Computing and Ubiquitous Networking, Poster and Demo session, Oct. 2017.

## 研究会・大会等（査読なし）

- (1) 尾久史弥, 納堂博史, 鈴木秀和, 内藤克浩, 渡邊晃 : NTMobile アダプタの実現方式の検討, 平成 28 年度電気関係学会東海支部連合大会論文集, 講演番号 B1-3, Sep. 2016.
- (2) 尾久史弥, 納堂博史, 鈴木秀和, 内藤克浩, 渡邊晃 : NTMobile アダプタの実現方式の検討, 情報処理学会第 79 回全国大会講演論文集, 講演番号 6U-2, Mar. 2017.
- (3) 尾久史弥, 鈴木秀和, 内藤克浩, 渡邊晃 : NTMobile アダプタの提案と実装, 情報学ワークショップ 2017(WINF2017) 論文集, 講演番号 PC-28, Mar. 2017.
- (4) 尾久史弥, 納堂博史, 鈴木秀和, 内藤克浩, 渡邊晃 : NTMobile 機能を持つアダプタの実現方式の検討, マルチメディア, 分散, 協調とモバイル (DICOMO2017) シンポジウム論文集, pp.402-408. 講演番号 2F-2, Jun. 2017.
- (5) 稲垣智, 尾久史弥, 鈴木秀和, 内藤克浩, 渡邊晃 : NTMobile における RS を利用しない経路構築手法の提案, 平成 29 年度電気関係学会東海支部連合大会論文集, 講演番号 C3-1, Sep. 2017.
- (6) 小島光野, 尾久史弥, 鈴木秀和, 内藤克浩, 渡邊晃 : NTMobile アダプタの実現と評価, 平成 29 年度電気関係学会東海支部連合大会論文集, 講演番号 C3-3, Sep. 2017.
- (7) 尾久史弥, 鈴木秀和, 内藤克浩, 渡邊晃 : IP ネットワークをフラット化する通信アダプターの提案と実装, 情報処理学会第 80 回全国大会講演論文集, 講演番号 6T-3, Mar. 2018.
- (8) 稲垣智, 尾久史弥, 鈴木秀和, 内藤克浩, 渡邊晃 : 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, 情報処理学会第 80 回全国大会講演論文集, 講演番号 6T-5, Mar. 2018.
- (9) 渡邊悠雅, 尾久史弥, 納堂博史, 鈴木秀和, 内藤克浩, 渡邊晃 : カプセル化パケットのフラグメント処理の検討, 平成 30 年度電気関係学会東海支部連合大会論文集, 講演番号 L1-7, Sep. 2018.