

平成29年度 卒業論文

和文題目

**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**

英文題目

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

情報工学科 渡邊研究室
(学籍番号: 140441100)

稲垣 智

提出日: 平成30年2月9日

名城大学理工学部

概要

同一 LAN 内での通信を前提にすると、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あるため、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。DSMIPv6 (Dual Stack Mobile IP version 6) や HIP (Host Identity Protocol) , NTMobile (Network Traversal with Mobility) はこのような目的を満たす可能性のある技術である。

しかし DSMIPv6 や HIP はそれぞれに固有の技術的課題を抱えているとともに、カーネルの改造を必要とするため、普及が難しいという課題を抱えている。それに対し NTMobile は技術的課題がないうえ、カーネルを改造しない実装が可能である。しかし、既存アプリケーションの実装を若干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利用できないという課題がある。

そこで本論文では様々な OS に標準的に実装されている TUN/TAP インタフェースを用いて、NT-Mobile をアプリケーションとして実装する。この手法により 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、かつカーネルや既存アプリの改造を必要としないシステムを実現できる。

目次

第 1 章	はじめに	1
第 2 章	NTMobile	2
2.1	NTMobile の概要	2
2.2	NTMfw と R-NTMfw	2
第 3 章	提案方式	5
3.1	アドレス情報登録処理	5
3.2	通信開始時の処理	5
第 4 章	実装	8
4.1	TUN アプリケーションのモジュール構成	8
4.2	動作検証	9
第 5 章	評価	10
5.1	定量評価	10
5.1.1	評価構成	10
5.1.2	iPerf を用いたスループットの測定	10
5.1.3	処理時間の測定	11
5.1.4	考察	13
5.2	定性評価	13
第 6 章	まとめ	15
	謝辞	17
	研究業績	21

第1章 はじめに

TCP/IP が設計された当初は、IP アドレスが枯渇することは想定されていなかった。しかしインターネットの普及に伴い、当初の想定をはるかに超えるネットワーク規模となった。そのため、IPv4 アドレスの枯渇を解決するため、NAT (Network Address Translate) を利用したプライベートアドレスによるネットワークが広く利用されるようになった。これによりインターネットは延命したが、グローバルアドレス空間側の端末からプライベートアドレス空間側の端末へ通信の開始が行えないという制約のあるネットワークになった (NAT 越え問題)。IPv6 ネットワークへの移行が徐々に進められているが、IPv4 ネットワークとの互換がないことから、IPv6 ネットワークへの完全移行には時間を要することが予想される。そのため、しばらくの間は IPv4 アドレスと IPv6 アドレスの混在環境が続くと予想されている。また、現在の IP ネットワークでは、通信端末がネットワークを切り替えると、IP アドレスが変化するため、通信を継続することができないという課題がある。このように現状のネットワークには様々な制約があり、ネットワークの形態を問わず通信開始を行うことが可能な通信接続性や、通信中に移動しても通信を継続できる移動透過性技術を実現できると有用である。これにより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 (Local Area Network) のように扱うことが可能になる。

ネットワークをフラット化できる技術の候補として DSMIPv6 [1], HIP [2], NTMobile [3-5] がある。しかし DSMIPv6 は IPv4 環境において必ず HA (Home Agent) を経由した通信となるため、通信経路が冗長になることや、全ての移動端末に IPv4 グローバルアドレスが必要となり、アドレス枯渇問題に逆行するなどの課題を抱えている。HIP は NAT を跨る移動が複雑になり、シグナリングに時間を要するという課題を抱えている。また、DSMIPv6 と HIP に共通する課題として、カーネル空間に実装する必要があるため、これらの技術が普及しない原因となっている。NTMobile は、DSMIPv6 や HIP が抱える技術的課題はない。また通信接続性や移動透過性の機能を NTMobile フレームワーク (NTMfw) [6] と呼ぶアプリケーションライブラリとして実現しているため、カーネルの改造が不要である。しかし、NTMfw は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。

そこで本論文では、様々な OS に標準実装されている TUN/TAP インタフェースを利用し、NTMfw の機能を別のアプリケーションとして実現することで、既存のアプリケーションをそのまま使用できる方法を提案する。また、提案方式を Linux 上に実装し、動作の確認を行ったため報告する。

以後、2 章で NTMobile について述べ、3 章では提案方式の実現方法について述べる。4 章では実装を行い、5 章では評価を行う。最後に 6 章でまとめる。

第2章 NTMobile

本章では、NAT 越え問題の解決や移動透過性を実現する技術であり、本提案のベースとなるシステムである NTMobile について述べる。

2.1 NTMobile の概要

図 1 に NTMobile のネットワーク構成図を示す。NTMobile は NTMobile 機能を実装したエンド端末（以後 NTM 端末）、NTM 端末のアドレス情報の管理、および通信経路の指示を行う DC（Direction Coordinator）、エンドエンドで直接通信ができない場合にパケットの中継を行う RS（Relay Server）から構成される。DC 及び RS はデュアルスタックネットワーク上に設置する必要がある。また、これらの装置群はネットワーク規模に応じて複数台設置による負荷分散が可能である。

NTMobile では NTM 端末に対して FQDN（Fully Qualified Domain Name）と、実ネットワークに依存しない仮想 IP アドレスを割り当てる。NTMobile では DNS の問い合わせをトリガとして通信経路の構築を行う。DC は DNS サーバとしての機能を有しており、NTM 端末から問い合わせを受けた FQDN を元に 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 を行っているため、DC からの通信経路指示をいつでも受信することができる。通信経路を構築した後、アプリケーションは仮想 IP アドレスに基づいたパケットを生成する。NTMobile では仮想 IP アドレスに基づくパケットを全て実 IP アドレスでカプセル化し、通信相手に送信する。通信中に端末がネットワークを切り替えると、実 IP アドレスは変化するが仮想 IP アドレスは変化しないため、通信を継続できる。

また、NTM 端末間で直接通信が行えない場合は RS を経由した通信を行う。RS は IP バージョン間の違いをカプセル化により吸収したり、通信を行う両 NTM 端末が異なる NAT 配下に存在する場合に通信の中継を行うことで通信接続性を保証する。

2.2 NTMfw と R-NTMfw

NTMobile では上記の機能を、NTMfw と呼ぶアプリケーションライブラリとして提供している。図 2 に NTMfw のモジュール構成を示す。アプリケーションは NTMfw が提供するソケット API を使用して通信を行うことで NTMobile の機能を利用できる。アプリケーションからデータを受信した NTMfw ソケット API は仮想 IP スタックへと処理を渡す。仮想 IP スタックは lwIP（A Lightweight TCP/IP stack）を用いており、受け取ったデータに対し、TCP/IP ヘッダや UDP ヘッダ等の付与を行うモジュールである。仮想 IP スタックには仮想 IP アドレスが紐付けられているため、受信したデー

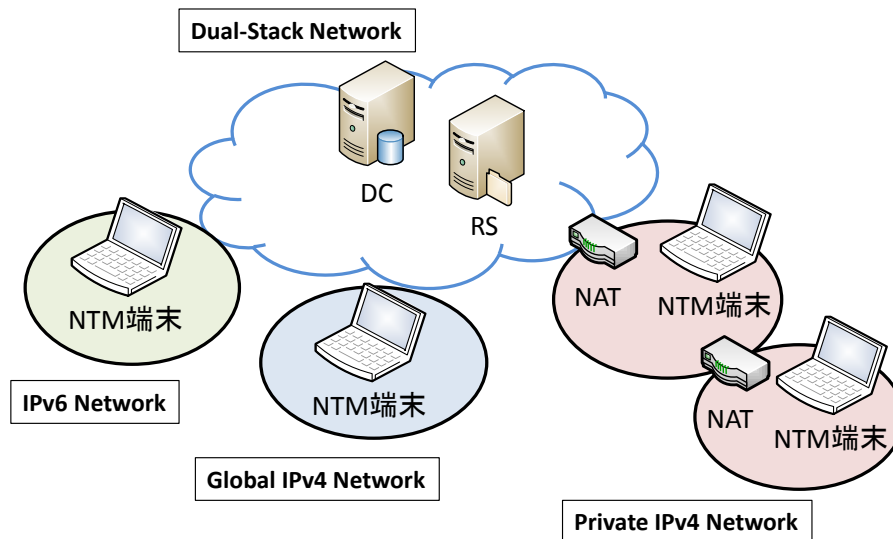


図1 NTMobileのネットワーク構成

タに仮想 IP アドレスによるヘッダ付与が行われる。このようにして生成されたパケットをパケット操作モジュールへ渡し、そこでNTMobile固有のヘッダ（NTMヘッダ）の付与及びMAC付与が行われ、暗号化された後にカーネルのソケットAPIに渡すことで、実IPアドレスによるカプセル化が行われて実ネットワークへ送信される。また、通信相手から受信したパケットはパケット送信時の逆の手順を辿ることでアプリケーションへ渡される。

また、NTMfwから仮想IPスタックを除いたR-NTMfw（Remodeled-NTMfw）[7]と呼ばれるものが存在する。こちらはアプリケーションから渡されるデータが既にパケットの形をしている場合に、仮想IPスタックによる無駄なヘッダ付与を行わないために作成されたものである。本提案ではこちらのR-NTMfwを利用する。

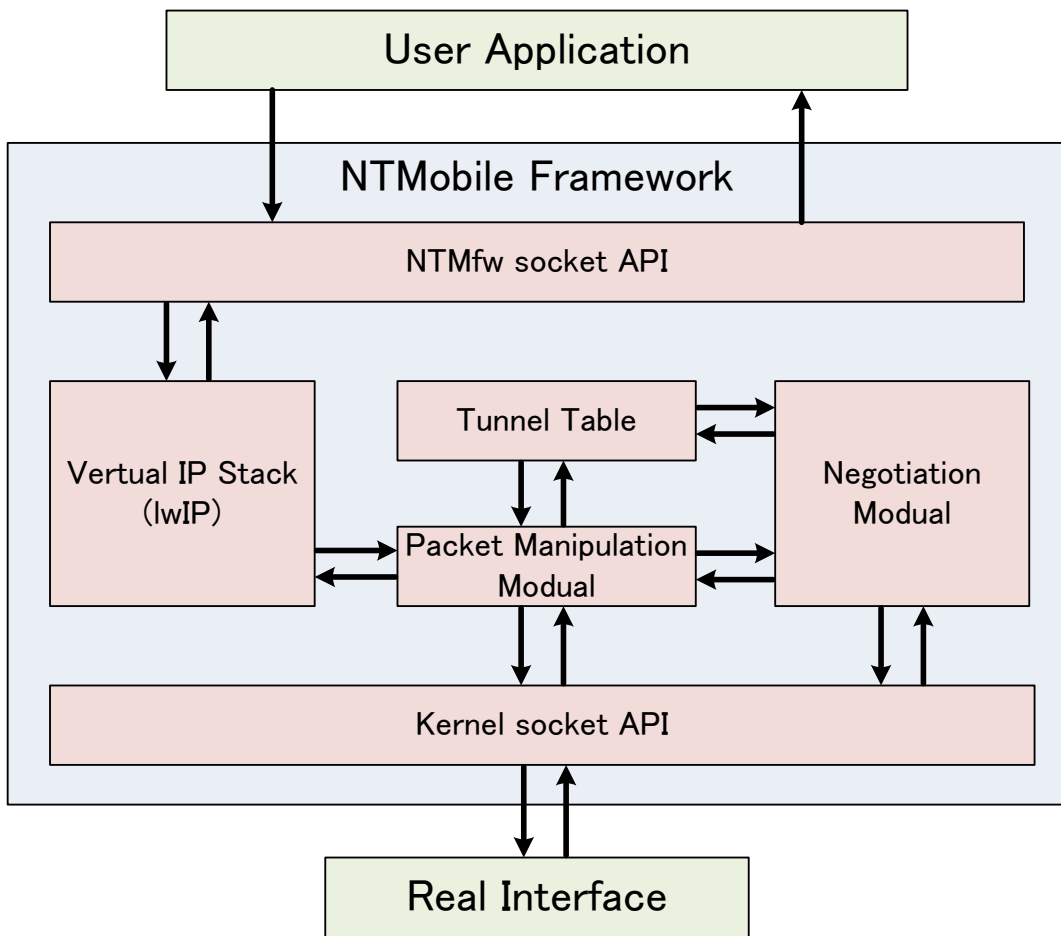


図2 NTMfw のモジュール構成

第3章 提案方式

本章では提案方式である TUN アプリケーションの動作について述べる。提案方式では NTMobile 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 のカプセル化に利用できる。また、TUN インタフェースは IP パケットをフックし、TAP デバイスはイーサネットフレームをフックする。今回は IP パケットへの操作を行うため、TUN インタフェースを利用する。提案方式では TUN インタフェースに仮想 IP アドレスを割り当てる。これによりフックされるパケットは仮想 IP アドレスによるヘッダ付与が行われる。

図 3 に提案方式におけるパケットのカプセル化の様子を示す。初めに一般のアプリケーションがデータを送信する。こちらは TUN インタフェースを経由して仮想 IP アドレスによるヘッダ付与が行われた後、TUN アプリケーションへフックされる。TUN アプリケーションは R-NTMfw の機能を用いて受信したパケットに NTM ヘッダ付与、及び MAC 付与を行い、暗号化した後に実インタフェースへ渡す。実インタフェースで実 IP アドレスによるカプセル化を行い、ネットワークへ送信する。

以後の説明では、通信を行う一般のアプリケーションをユーザアプリと表記する。

3.1 アドレス情報登録処理

TUN アプリは起動時、及びハンドオーバー時に R-NTMfw の機能を利用し、DC に対してアドレス情報の登録処理を行う。そちらの応答として仮想 IP アドレスが割り当てられる。仮想 IP アドレスを受け取った 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 IP 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。

3.2 通信開始時の処理

図 4 に通信開始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

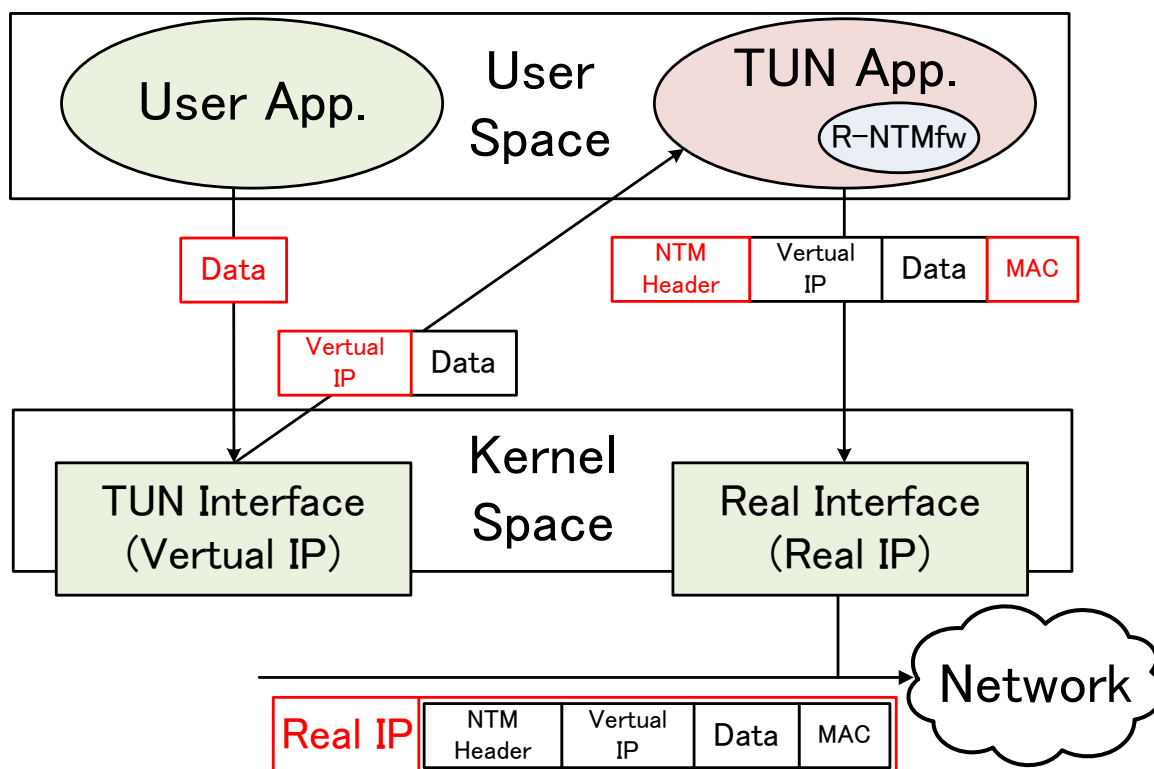


図3 提案方式におけるパケットカプセル化の様子

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、R-NTMfw の機能により NTMobile シグナリング処理を実行し、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宛先が仮想 IP 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。また、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。

また、ユーザアプリが指定する FQDN が一般端末のものであった場合は、受信したパケットをそのまま RAW ソケットで DNS サーバへ送信する。DNS サーバから得られた応答を DNS 応答に記載し、ユーザアプリへ返信することで、一般端末との通信も行うことができる。

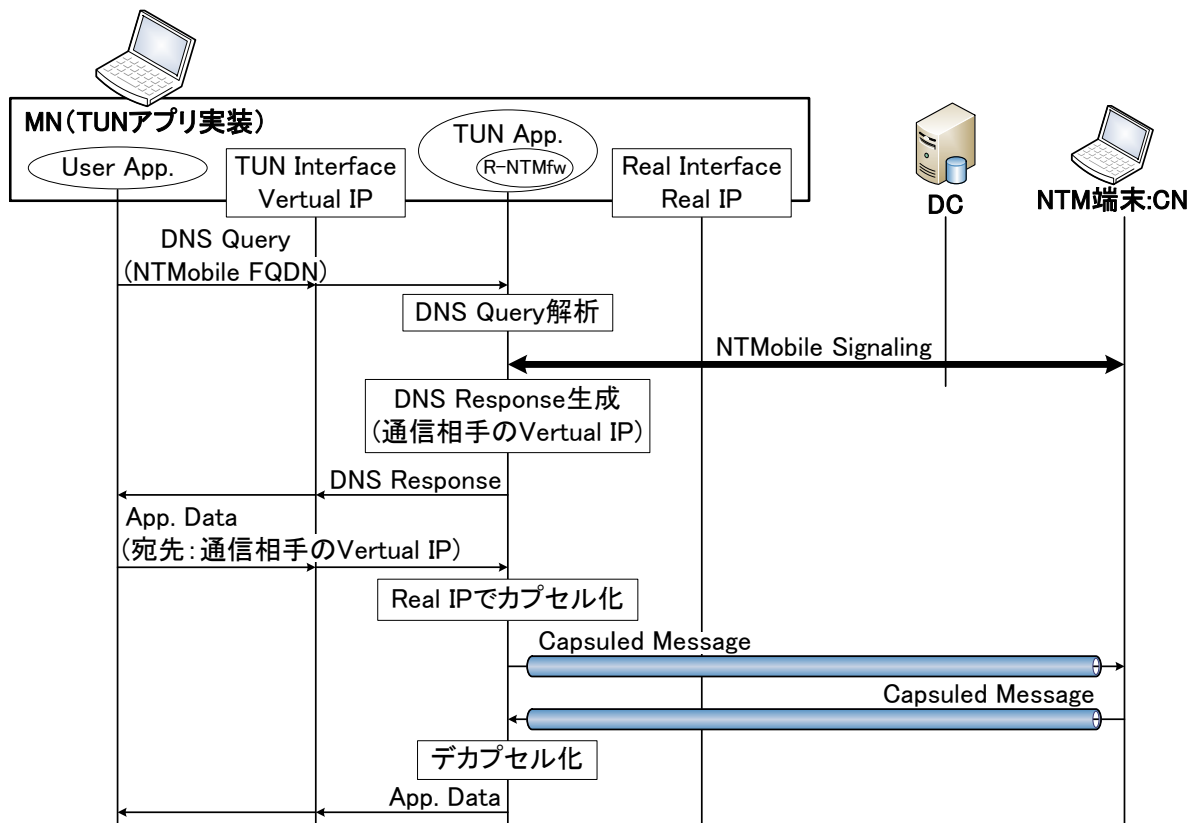


図 4 提案方式における通信開始時の動作シーケンス

第4章 実装

本章では、提案方式の実装及び動作検証について述べる。実装は Linux 環境にて行った。

4.1 TUN アプリケーションのモジュール構成

図 5 に TUN アプリケーションのモジュール構成を示す。以下にそれぞれのモジュールについて説明する。なお、R-NTMfw については 2.4 節で述べているため、ここでは省略する。

- **Initiation Module**

TUN アプリケーションの起動時に、R-NTMfw の機能を用いてアドレス情報登録処理を行うモジュールである。

- **TUN IF (Inter Face) Setup Module**

TUN インタフェースを起動し、仮想 IP アドレスの割り当てを行うモジュールである。

- **Packet Manipulation Module**

パケットの中継及び解析を行うモジュールである。このモジュールがユーザアプリからのパケットを受信する。受信したパケットが DNS クエリであった場合は通信相手 FQDN の解析を行う。ここで通信相手が NTM 端末であった場合は R-NTMfw の機能を用いて NTMobile シグナリング処理を行い、トンネル経路構築処理を行う。トンネル経路構築処理が終了した後、通信相手の仮想 IP アドレスを DNS レスポンスに載せユーザアプリへ返信する。また、通信相手が一般端末であった場合は、RAW ソケットにより DNS クエリをそのまま DNS サーバ宛に送信する。その後 DNS サーバから得られた応答をそのままユーザアプリへ返信する。

また、受信したパケットが DNS 以外のパケットであった場合は、R-NTMfw が提供するソケット API を呼び出し、処理を R-NTMfw へ移す。

これらのモジュールのうち、TUN IF Setup Module は今回新たに実装を行ったモジュールである。その他のモジュールは、エンド端末に隣接設置することで一般通信を NTMobile 通信に変換する NTMobile Adaptor [7] のモジュールを大いに利用した。変更点として、Initiation Module の処理により割り当てられた仮想 IP アドレスを TUN IF Setup Module へ渡す必要があるため、両モジュール間の中継処理部を実装した。また、Packet Manipulation Module において、TUN インタフェースからのパケット受信部、及び TUN インタフェースへのパケット送信部の実装を行った。

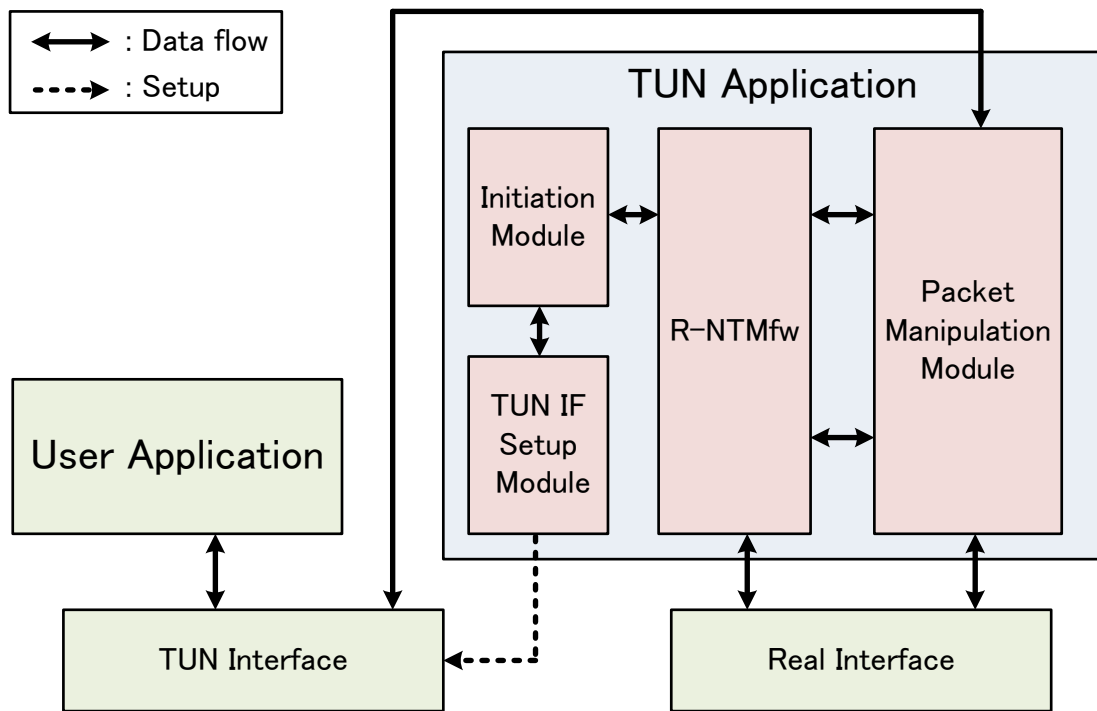


図5 TUN アプリケーションのモジュール構成

4.2 動作検証

TUN アプリを Linux 上で一部実装し動作検証を行った。動作検証を行うにあたって、DNS クエリのルーティングテーブルの設定は静的に行った。検証方法は提案方式を実装した PC 端末を 2 台準備し、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。

第5章 評価

本章では、4章のモジュールを実装した TUN アプリケーションの評価を行う。

5.1 定量評価

本節では定量評価について述べる。定量評価では iPerf を用いたスループットの測定と、処理時間の測定の2通りの方法により評価を行った。

5.1.1 評価構成

図 6 に評価に用いるネットワーク構成を示す。MN 及び CN としてラップトップ PC を 2 台用意し、それぞれに提案方式の TUN アプリを起動させた。表 1 に MN 及び CN に用いたラップトップ PC の仕様を示す。DC はデスクトップ PC 内に VMware Workstation Player を用いて、仮想マシンとして準備した。これらの MN、CN 及びデスクトップ PC をスイッチングハブで繋ぎ、IPv4 ネットワークに接続した。また、MN-CN 間でのトンネル経路構築はあらかじめ完了させた。

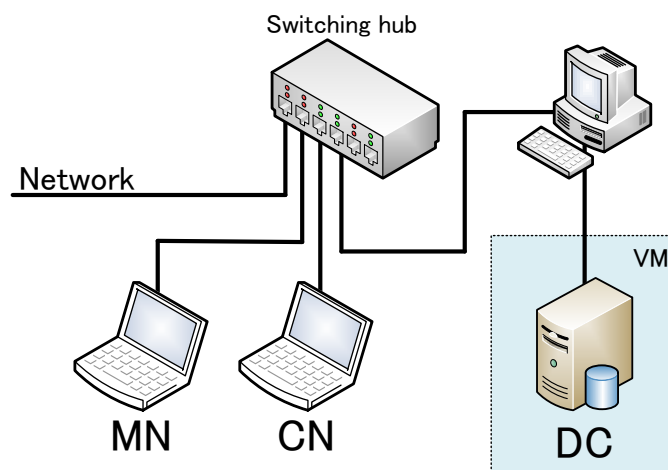


図 6 評価構成

5.1.2 iPerf を用いたスループットの測定

図 6 の構成のもとで、MN から CN へ向けて iPerf によるスループットの測定を行った。通常の通信と TUN アプリを経由する通信においてそれぞれ測定を行い、結果を比較した。測定時の条件は以

表 1 MN 及び CN の仕様

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

表 2 iPerf によるスループットの測定結果

TUN アプリ経由	一般の通信
4.73 Mbits/s	95.27Mbits/s

下の通りである.

- UDP 通信による測定を行った.
- NTMobile のカプセル化によるヘッダオーバーヘッドを考慮し, iPerf によるパケットサイズを 1400 に設定した.
- 10 秒間の測定を 10 回ずつ行い, 平均値を評価結果とした.

表 2 に測定結果を示す. 測定結果より, TUN アプリを経由した通信は, 一般通信の 4.96% のスループットまで低下することがわかった.

5.1.3 処理時間の測定

図 6 の構成のもとで, MN から CN の仮想 IP アドレスへ向けて「ABCDE」の文字列をデータとする UDP パケットを 10 回送信し, TUN アプリの処理に要した平均時間の測定を行った. 図 7 に送信側における評価項目を, 図 8 に受信側における評価項目を示す. また, 表 3 に送信側の評価結果を, 表 4 に受信側の評価結果を示す.

図 7 における TUN インタフェース経由時間はユーザアプリがパケットを送信し, TUN アプリがパケットを受信するまでに要する時間である. TUN アプリ処理時間は TUN アプリがパケットを受信してから, 処理を R-NTMfw に渡すまでに要する時間である. R-NTMfw 処理時間は R-NTMfw がデータを受信してから, パケットを実ネットワークへ送信するまでに要する時間である.

図 8 における R-NTMfw 処理時間は R-NTMfw が実インタフェースからパケットを受信してから, TUN アプリの処理モジュールへ渡すまでに要する時間である. TUN アプリ処理時間は TUN アプリがパケットを受信してから, ユーザアプリにパケットを送信するまでに要する時間である. TUN インタフェース経由時間は TUN アプリがパケットを送信し, ユーザアプリがパケットを受信するまでに要する時間である.

また, これらの測定には C 言語にて標準的に実装されている `clock_gettime` 関数 [8] を使用した. 図 7 及び図 8 にて定義した各処理時間の先頭と末尾において当該関数を用いて時間を取得し, 末尾の時間と先頭の時間の差分を処理時間とした.

表 3 より, R-NTMfw の処理に要する時間が 320.3 μ s となっており, 最も時間を要していることがわかった. また TUN インタフェース経由時間と TUN アプリ処理時間は合わせて 222.0 μ s となり, 全体の 40% 程度であることがわかった.

表 4 より, R-NTMfw の処理に要する時間が $439.7 \mu\text{s}$ となっており, 送信側と同様に最も時間を要していることがわかった. また TUN インタフェース経由時間と TUN アプリ処理時間は合わせて $191.1 \mu\text{s}$ となり, 全体の 30%程度であることがわかった.

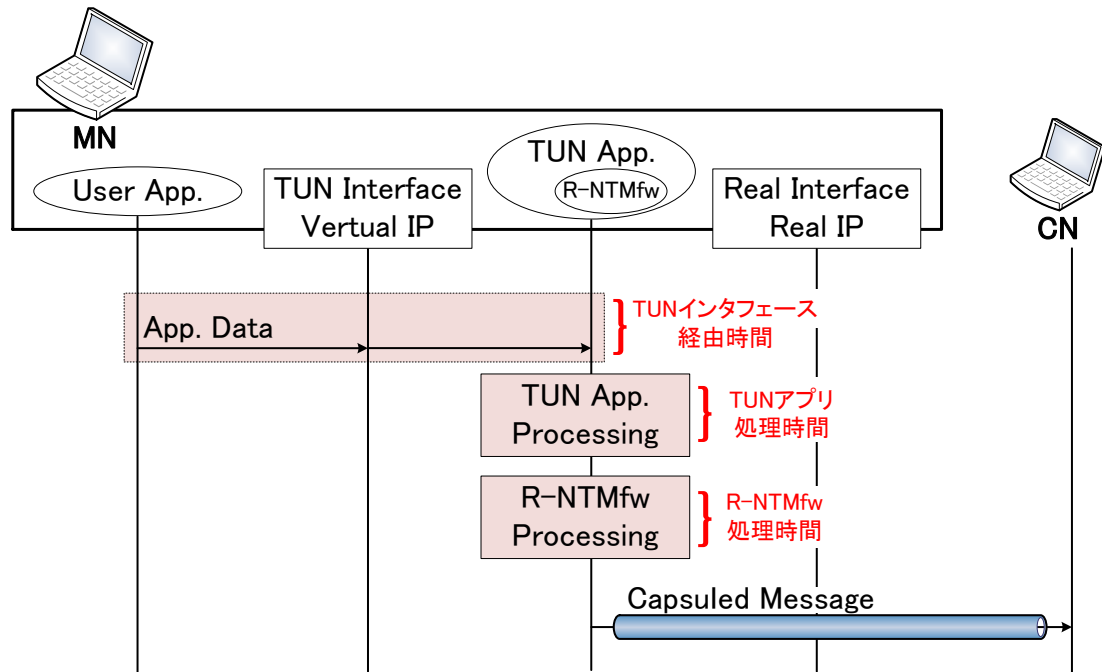


図 7 送信側の評価項目

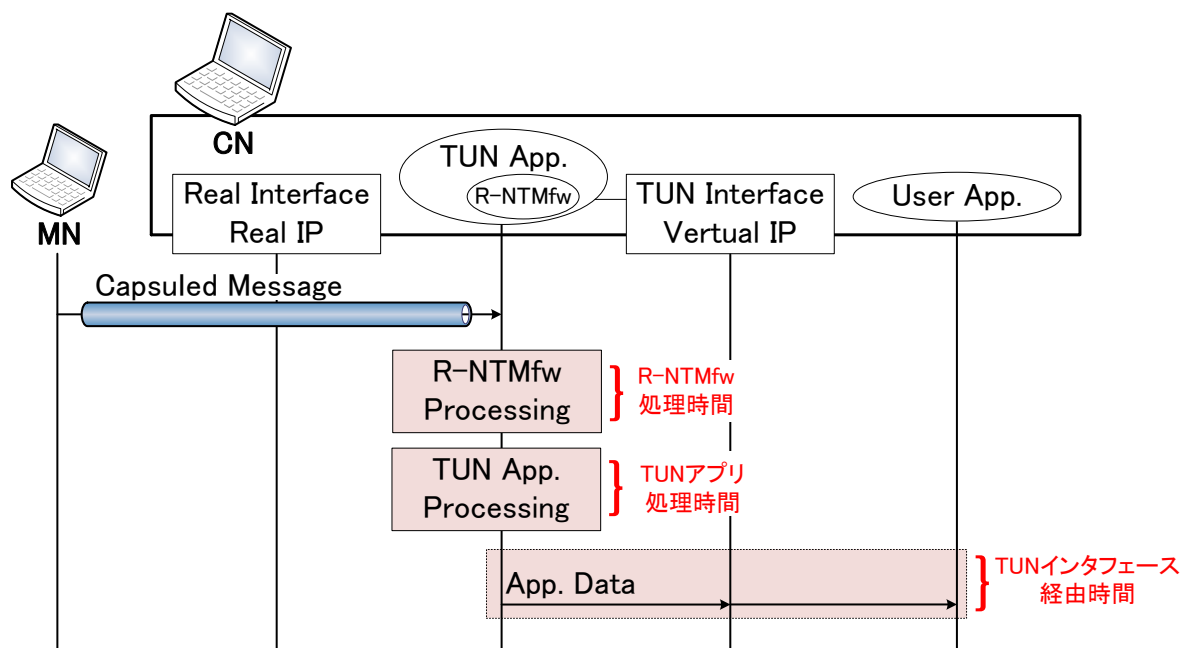


図 8 受信側の評価項目

表 3 送信側の評価結果

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

表 4 受信側の評価結果

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

5.1.4 考察

5.1.2 節の結果より、著しいスループットの低下が見られた。しかし 4.73 Mbits/s のスループットであれば、一般的な画像データや動画データ等を潤滑にやり取りすることが可能であるため、用途によっては利用可能であると考えられる。また、5.1.3 節の結果より、送信側及び受信側とも、R-NTMfw の処理に最も時間を要することがわかった。こちらの原因は、送信側ではメッセージの暗号化処理、受信側ではメッセージの復号処理が含まれているため、これらの処理に時間を要しているからであると考えられる。暗号化技術は現在のネットワークでは必須の課題となっているため、これらの処理に要する時間は必要なものである。そのため、今後は既存の暗号化プロトコルとの性能比較を行い、提案方式の有用性を確認する必要があると考えられる。

5.2 定性評価

表 5 に提案方式と既存技術との比較を示す。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) 既存アプリをそのまま使用できるか
- (3) グローバルアドレスを消費しないか
- (4) 冗長な通信経路を取らないか
- (5) 移動後に素早い通信再開が行えるか

項目 (1) において、DSMIPv6 や HIP ではカーネルの改造を必要とするが、NTMfw や提案方式ではアプリケーションレベルで実現するため、カーネルの改造を必要としない。項目 (2) において、NTMfw では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使用できなかったが、提案方式では別のアプリケーション内で実現するため、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る。また、項目 (3-5) において、提案方式では NTMmobile を利用することで“○”となっている。NTMmobile は実ネットワークに依存しない仮想 IP アドレスを使用しており、また常に最適な通信経路を構築することができる。また移動に係るシーケンスもシンプルなものであるため、これらの項目を“○”としている。

表 5 提案方式と既存技術との比較

	項目 (1)	項目 (2)	項目 (3)	項目 (4)	項目 (5)
DSMIPv6	×	○	×	△	○
HIP	×	○	○	○	△
NTMfw	○	×	○	○	○
提案方式	○	○	○	○	○

第6章 まとめ

本論文では、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 アプリケーションの提案と実装を行った。TUN アプリ内で R-NTMfw の処理を呼び出すことにより、一般のアプリケーションに変更を加えることなく 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。また、Linux 上で TUN アプリケーションの実装を行い、動作検証を行った。そして、TUN アプリケーションを経由する場合と経由しない場合でスループットの比較を行い、スループットが大きく低下することがわかった。また TUN アプリケーション内において、各処理ごとに要する時間を計測し、R-NTMfw の処理に要する時間が最も大きいことを確認した。こちらの原因は R-NTMfw には暗号化/復号処理が含まれているため、それらの処理に時間を要しているからであると考えられる。

今後は DNS クエリのルーティング設定部分の検討を行う予定である。また、既存の暗号化プロトコルとの性能比較を行い、提案方式の有用性を確認する。

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工学部情報工学科 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,様々なご指導を賜りました,名城大学工学部情報工学科 鈴木秀和准教授に深謝致します.

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 内藤克浩准教授に深謝致します.

最後に,本研究を進めるにあたり,様々なご意見を賜りました,渡邊研究室及び鈴木研究室の皆様
に感謝致します.

参考文献

- [1] Soliman, H. : Mobile IPv6 Support for Dual Stack Hosts and Routers, RFC 5555, IETF (2009).
- [2] Jokela, P. : Host Identity Protocol, RFC 5201, IETF (2008).
- [3] 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).
- [4] 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.1, pp.380-397 (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).
- [6] 納堂博史, 八里栄輔, 鈴木秀和, 内藤克浩, 渡邊 晃 : 実用化に向けた NTMobile フレームワークの実装と評価, 電子情報通信学会技術研究報告, Vol.116, No.509, pp.281-288 (2017).
- [7] 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 (2017).
- [8] Man page of CLOCK_GETRES :
https://linuxjm.osdn.jp/html/LDP_man-pages/man2/clock_gettime.2.html

研究業績

研究会・大会等（査読なし）

- (1) 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・電子・情報関係学会東海支部連合大会論文集, Vol. 2017, 講演番号 C3-1, Sep. 2017.
- (2) 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, 第 80 回情報処理学会全国大会講演論文集, Vol.2017, 講演番号 6T-05, Mar. 2018.

