

# vCPE を用いた NTMobile サーバの 運用効率化に関する基礎的検証

加藤 宏理<sup>1</sup> 松岡 穂<sup>1</sup> 鈴木 洸太<sup>1</sup> 鈴木 秀和<sup>1</sup> 内藤 克浩<sup>2</sup>

概要：IPv4 アドレスの枯渇に伴う NAT の導入や IPv6 への移行により複雑化したネットワークにおいて、移動透過性を実現する技術として NTMobile (Network Traversal with Mobility) が提案されている。しかし NTMobile では通信開始時に行われるシグナリング処理で生じる遅延や、直接通信できない場合に用いられる中継サーバ経由時に生じる経路冗長化、中継サーバの最適な選択が実現されていない等の課題がある。本稿ではこれらの課題を解決するため、ISP のネットワークエッジに設置されネットワーク機能を提供する汎用サーバである vCPE (Virtual Customer Premises Equipment) 上への NTMobile サーバ群の実装を提案し、ネットワークエミュレータの Mininet を用いた基礎的な検証により有用性を確認する。

## Basic Verification of Operational Efficiency of NTMobile Server Using vCPE

HIROTOSHI KATO<sup>1</sup> MINORU MATSUOKA<sup>1</sup> KOTA SUZUKI<sup>1</sup> HIDEKAZU SUZUKI<sup>1</sup>  
KATSUHIRO NAITO<sup>2</sup>

### 1. はじめに

現在のインターネット環境は IPv4 アドレスの枯渇に伴う NAT (Network Address Translation) の導入や IPv6 アドレスへの移行により複雑化している。また、スマートフォンをはじめとするモバイル端末の普及により、4G/LTE や Wi-Fi などを場所や状況に応じて使い分けながら通信を行う機会が増加している。しかし、通信インタフェースを切り替えると接続先ネットワークが変化してしまうため、IP アドレスが変化してしまう。これにより、通信の断絶が発生するという課題が存在する。

複雑化したネットワークにおいてこの課題を解決し、移動透過性を実現する技術として NTMobile (Network Traversal with Mobility) がある [1]。NTMobile ではエンド端末に対して移動によって変化しない一意な仮想 IP アドレスを専用サーバが割り当て、この仮想 IP アドレスを設定し

た IP パケットを UDP トンネルを用いて伝送することにより、アプリケーションに対してネットワークの切り替えによる実 IP アドレスの変化を隠蔽する。しかし NTMobile ではエンド端末同士の通信開始時に必要なシグナリング処理で生じる遅延や、直接通信できない場合に端末間の通信を中継する中継サーバ経由時に生じる経路冗長化、中継サーバの最適な選択が実現されていない等の課題がある。

これらの課題を解決するため、筆者らは ISP (Internet Service Provider) のネットワークエッジに設置されネットワーク機能を仮想的に提供する装置である vCPE (Virtual Customer Premises Equipment) [2] に着目し、vCPE 上へ NTMobile サーバ群を運用する方式を提案している [3], [4]。vCPE がネットワークエッジに設置され、ユーザがインターネットへ通信を行う際の中継地点である点に着目し、エンド端末と NTMobile サーバ群との通信時間削減や中継サーバ経由時の経路冗長化を解決し、vCPE の備える NFV (Network Functions Virtualization) [5] を用いた負荷分散によって、経路・負荷の両面で最適な中継サーバの選択を実現する。しかし、従来手法に加えて vCPE のパケット転送処理が必要となることにより、処理時間が増加する懸念

<sup>1</sup> 名城大学大学院理工学研究科  
Graduate School of Science and Technology, Meijo University

<sup>2</sup> 愛知工業大学情報科学部  
Faculty of Information Science, Aichi Institute of Technology

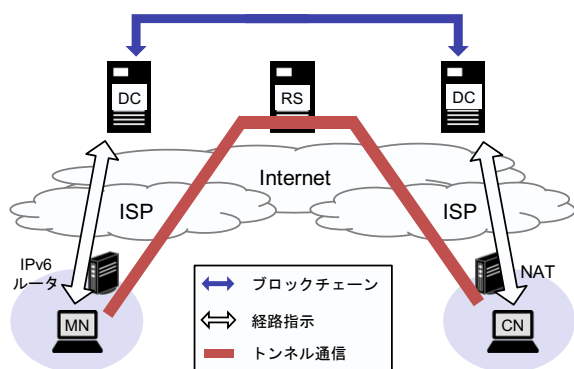


図 1 NTMobile の構成

があった。

そこで本稿では、ネットワークエミュレータの Mininet[6] を用いて NTMobile サーバ群が vCPE 上に実装された環境を模した仮想ネットワークを作成し、NTMobile サーバを vCPE 上へ実装した場合に生じる通信遅延等の検証を行う。

以下、2 章で NTMobile の概要について述べ、3 章で提案方式について述べる。4 章で Mininet を用いた検証と評価について述べ、5 章でまとめる。

## 2. NTMobile

### 2.1 概要

NTMobile は現在の複雑化したネットワークにおいて移動透過性を実現するシステムである。本稿はこの NTMobile のうち、ブロックチェーン型 NTMobile[7] を用いて検証を行った。図 1 にブロックチェーン型 NTMobile の構成を示す。

NTMobile はインターネット上に設置された専用のサーバ群と、NTMobile が実装されたエンド端末によって構成される。代表的なサーバとして、端末への仮想アドレス割当てとブロックチェーンを用いた端末管理を行う DC (Direction Coordinator)、端末同士が異なる NAT の配下に存在する場合や IP アドレスのバージョンが異なる場合などの直接通信を行えない状況においてパケットの中継を行う RS (Relay Server) が存在する。また、NTMobile が実装されたエンド端末を NTM 端末と呼称する。NTM 端末は初回起動時に楕円曲線暗号の秘密鍵と公開鍵を作成し、公開鍵のハッシュから作成される ID と秘密鍵による署名を端末情報として保持する。

### 2.2 位置登録

NTM 端末は起動時や移動後など、仮想アドレスの割り当てを要求するため DC に対して位置登録を行う。図 2 に位置登録シーケンスを示す。NTM 端末は自身の端末情報に加え、宛先となる DC の情報やタイムスタンプなどを Node Transaction として送信する。Node Transaction を受信した DC は、Node Transaction を検証することによ

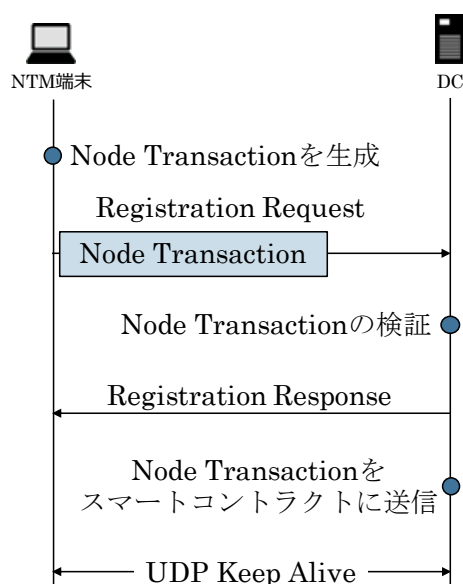


図 2 位置登録のシーケンス

り、NTM 端末の認証を行う。認証結果が正常な場合は仮想アドレスを NTM 端末へ送信し、Node Transaction をスマートコントラクトへ送信することにより、ブロックチェーンを用いた端末情報の管理を行う。また、このとき DC は NTM 端末の実 IP アドレスと NAT の外側の IP アドレスを取得する。その後 DC は NTM 端末との通信路を維持するため、一定間隔で UDP によるキープアライブを行う。

### 2.3 トンネル構築シグナリング

NTM 端末が通信を開始する際、DNS による名前解決が発生するが、NTMobile では DNS クエリをトリガーにトンネルを構築するシグナリング処理を開始する。トンネル構築シグナリングは、経路指示処理とトンネル構築処理が行われる。

図 3 に経路指示シーケンスを示す。送信側の NTM 端末を MN、受信側の NTM 端末を CN、MN および CN が位置登録を行った DC をそれぞれ  $DC_{MN}$ 、 $DC_{CN}$  と呼称する。MN は通信開始時に  $DC_{MN}$  へ Direction Request メッセージを送信する。このメッセージは通信相手端末となる CN の ID が記載されている。Direction Request を受信した  $DC_{MN}$  は CN の ID を元にスマートコントラクトを用いてブロックチェーン上の Node Transaction を検索し、CN の端末情報を取得する。この CN の端末情報に記録された  $DC_{CN}$  に対し、 $DC_{MN}$  は CN の端末情報を要求する。 $DC_{CN}$  は自身が持つ CN の端末情報を  $DC_{MN}$  に対して送信する。CN の端末情報を受信した  $DC_{MN}$  は自身が保持する MN の端末情報とあわせて、UDP トンネルの最適経路を計算する。このとき、図 1 のように両端末が直接通信できない場合は RS 経由の経路となる。 $DC_{MN}$  は MN と CN

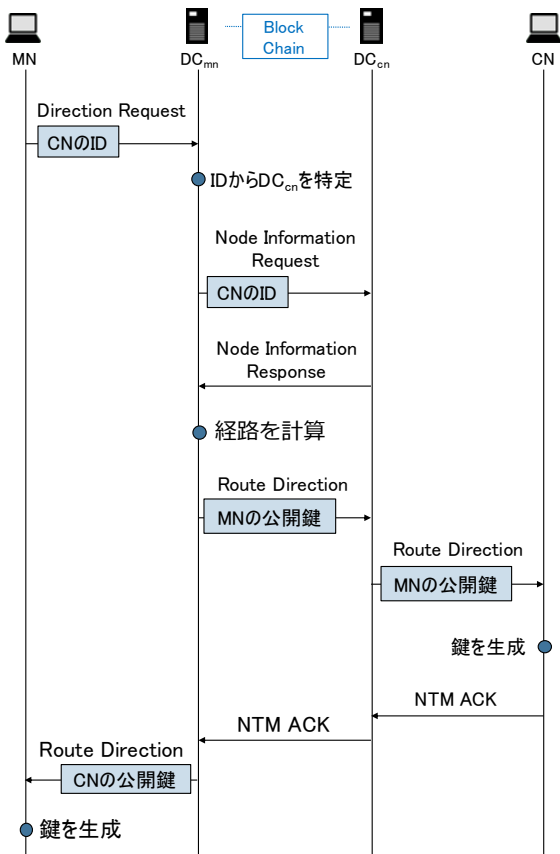


図 3 経路指示シーケンス

に対して Route Direction メッセージを送信し、トンネル構築先の指示と通信相手の公開鍵を配付する。このとき、CN に対しては DC<sub>CN</sub> 経由で送信される。Route Direction を受信した両端末は相手端末の公開鍵と自身の秘密鍵で Diffie-Hellman 鍵共有を行い、エンド鍵を生成する。

次に、図 4 に RS を経由する場合のトンネル構築シーケンスを示す。Route Direction を受信した MN は、指示された RS に対して Tunnel Request メッセージを送信する。Tunnel Request を受信した RS はそのメッセージを CN へ転送する。ここで、RS と CN 間には NAT が存在するが、CN は Route Direction メッセージ受信時にトンネル構築先として指示された RS に対して Hole Punching を実施しているため、RS は NAT を越えて CN へ Tunnel Request を届けることができる。CN から MN に対しては RS を経由して Tunnel Response メッセージが応答される。この一往復のやり取りにより、経路指示の際に生成された共通のエンド鍵を保有していることを確認する。

以上の処理により、MN と CN は RS を中継した暗号化 UDP トンネル経路を構築することができる。その後の MN と CN 間の通信は仮想アドレスに基づく IP パケットを実 IP アドレスでカプセル化し、ペイロード部分をエンド鍵で暗号化して行われる。RS ではカプセル化ヘッダの付け替えのみを行うため、エンドエンド通信の秘匿性は保証される。

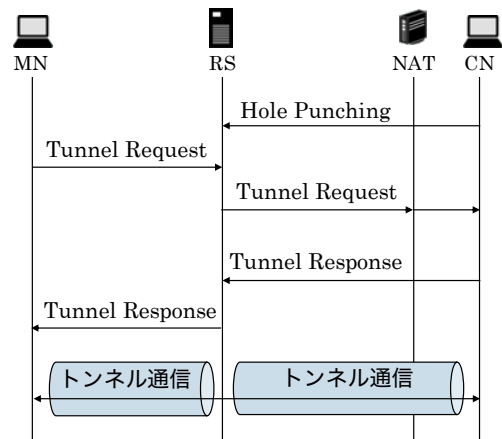


図 4 トンネル構築シーケンス

## 2.4 課題

トンネル構築シグナリングでは、MN および CN と各 DC<sub>MN</sub>/DC<sub>CN</sub> との間、さらに RS との間で複数の制御メッセージの交換が発生する。DC や RS は全ての NTM 端末からメッセージを受け取れるよう、クラウド上やインターネット上に構築する必要があるため、通信開始までに発生するトンネル構築処理は伝送遅延や、DC および RS の負荷状況により処理時間が増加する可能性がある。

また、経路指示の際に DC<sub>MN</sub> は必要に応じて中継すべき RS を指示するが、選択した RS によってエンドツーエンド遅延やスループット性能が変化してしまう可能性がある。仮に高スループットが得られると推定された RS を中継すると、伝送遅延がかえって増加してしまい、アプリケーション性能に影響を及ぼす可能性もある。文献 [8] において最適な RS 選択手法が検討されているものの、実現には至っていない。

## 3. 提案方式

### 3.1 概要

近年クラウドシステムの利用に伴う伝送遅延の課題を解決するため、MEC (Mobile Edge Computing) の活用が注目されている。MEC では処理の低遅延化を図るため、エッジコンピュータを用いて処理を行う。本稿ではこのアプローチと同様に、NTMobile サーバ群の vCPE 上での稼働を提案する。

vCPE は仮想的にネットワーク機能を提供する汎用サーバである。専用のハードウェアによって構成され、ネットワーク機能を提供する CPE (Customer Premises Equipment) を仮想化し、ユーザからのパケットに対しネットワーク機能を適用してインターネットへと転送する役割を持つ。これらを実現するため vCPE にはネットワーク仮想化技術が用いられており、SDN を用いた柔軟なパケットの転送機能と、NFV を用いた仮想マシン管理機能を備える。CPE はユーザ側に設置される設備であるが、vCPE は ISP のネットワークエッジに設置される。そのため、ISP は新

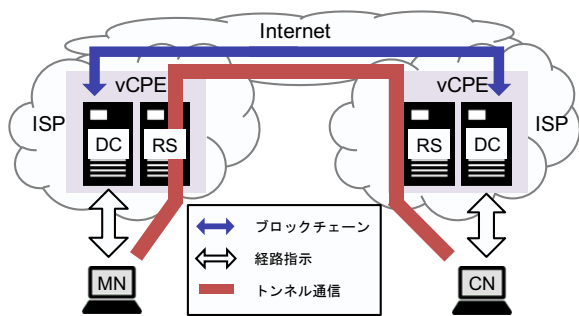


図 5 提案方式の概要

規サービスの導入が容易になり、またメンテナンス性の向上を図ることができるなどの利点が挙げられる。

vCPE がネットワークエッジに設置された汎用サーバである点、ユーザからのパケットがインターネットへ転送される際に必ず経由する設備である点、NFV による仮想マシン管理機能を備えている点に着目し、従来の NTMobile に存在する NTMobile 通信時の伝達遅延や NTMobile サーバの負荷による処理時間増加、最適な RS の選択が実現できていないといった課題の解決を図る。

提案方式の概要を図 5 に示す。

### 3.2 NTMobile の処理

従来の NTMobile では NTMobile サーバ群はクラウドやネットワークに設置された汎用サーバ上で稼働するが、提案手法ではネットワークエッジに設置された vCPE 上で稼働する。そのため提案手法では NTM 端末と NTMobile サーバ間の往復遅延時間の削減が可能となる。提案手法においても従来手法と同様に、NTMobile 通信を行うため位置登録やシグナリングなどの処理が必要になる。特にシグナリング処理では NTM 端末と NTMobile サーバ間で最大 4 往復分の通信が生じるため、処理時間の短縮が見込める。加えて NTM 端末と NTMobile サーバ間の通信がネットワークエッジ内で完結するため、インターネットの輻輳による遅延時間増加などの影響を抑えることが可能となる。

vCPE では仮想されたネットワーク機能を複数の仮想マシンによって実現し、ユーザに対して提供する。そのため vCPE ではこれらの仮想ネットワーク機能を稼働させるため NFV を用いて仮想マシンの管理を行い、負荷に応じたオートスケーリングやロードバランシングなどを実行する。vCPE 上で NTMobile サーバ群を稼働させることでこれらの機能を活用し、NFV による仮想マシンの管理が可能となり、負荷分散の実現と負荷による処理時間増加を抑えることが可能となる。

### 3.3 最適な RS の選択

vCPE はユーザから送信されたパケットに対してネットワーク機能を適用し、インターネットへ転送する役割を持つ。そのため vCPE はユーザがインターネットへアクセス

する際の経路上に位置し、ユーザからのパケットが必ず到達する設備となる。RS が vCPE 上で稼働することにより、MN と CN が RS 経由で通信を行う場合においても、直接通信を行う場合と同じ最適経路で通信を行うことが可能となる。これにより RS がインターネット上に設置される従来手法では RS を経由する場合に必ず生じていた経路冗長化が解消される。

経路冗長化の解消と前述した NFV による負荷分散が実現することで、vCPE 上で稼働する RS が NTM 端末に対して経路・負荷の両面で最適な RS となる。そのため DC は経路指示における RS の選択時に、自身と同じ vCPE 上の RS を経由する経路を指示することで、容易に最適な RS を選択することができる。

### 3.4 提案手法における懸念点

一方で vCPE 上で NTMobile サーバを稼働させることで処理時間が増加する可能性も存在する。vCPE において SDN を OpenFlow を用いて実現していた場合、内部におけるパケットの転送は OpenFlow Switch によって行われる。OpenFlow Switch は自身の持つ転送ルールにマッチしないパケットが転送された場合、OpenFlow Controller へ転送ルールの追加を要求する。そのため位置登録を行う際や MN と CN が通信を行う際など、新規に通信を開始する際には転送ルールの追加が行われ、遅延時間が増加する可能性がある。

またエッジサーバはクラウド上のサーバに比べリソースが低いことが一般的である。そのため vCPE に用いられる汎用サーバの性能によっては NTMobile サーバの処理時間が従来手法に比べ増加する可能性がある。

## 4. 検証・評価

### 4.1 検証環境

提案手法における vCPE への実装による NTMobile 通信への影響を評価するため、SDN の検証に利用されるネットワークエミュレータの Mininet を用いて仮想ネットワークを構築した。図 6 に構築した仮想ネットワークの概要を示す。この仮想ネットワークはブロックチェーン型 NTMobile の検証の際に用いられた環境 [7] を元に、NTMobile サーバ群が vCPE 上で稼働しているネットワークを想定して構築した。なお、各 RTT は文献 [9] を参考として、MN から MN 側の vCPE までを 10ms に、CN から CN 側の vCPE までを 20ms に、vCPE 間を 100ms に設定した。

表 1 に検証に用いたホスト PC の仕様とソフトウェアを示す。各 NTM 端末 (MN および CN) や NTMobile サーバ (DC および RS) は Mininet によって作成された仮想ホスト上に構築される。

この環境において位置登録とシグナリングを行い、要し



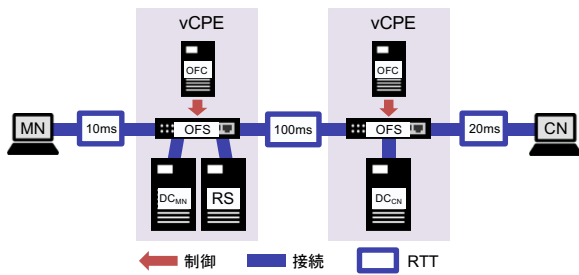


図 6 検証環境

表 1 検証環境構築用ホスト PC の仕様およびソフトウェア

CPU	Core i7-2600
メモリ	12GB
OS	Ubuntu 18.04 LTS
OpenFlow Controller	OpenDaylight ver 0.8.4
OpenFlow Switch	Open vSwitch
ネットワークエミュレータ	Mininet

表 2 位置登録処理に要した時間

	従来手法 [ms]	提案手法 [ms]
DC の処理	11.3	21.7
OpenFlow の処理	—	34.8
伝送遅延	21.0	10.0
全体	32.3	66.5

た時間を計測した。また、その後シグナリングで構築したトンネルを用いて通信を行い、要した時間を計測した。

## 4.2 位置登録

MN から  $DC_{MN}$  に対して位置登録処理を 100 回行い、処理時間の平均値を算出した。位置登録処理時間の計測には、MN が実行する NTM 端末のプログラムにおいて位置登録処理の packets を送信した時間と、位置登録処理終了後に  $DC_{MN}$  から送信される packets を受信した時間の差分を求める方法を用いた。なお、一連の位置登録処理によって OVS (OpenvSwitch) に追加された MN と  $DC_{MN}$  間の転送ルールは、1 度計測するたびに削除した。

表 2 に位置登録処理に要した処理時間の結果を示す。この値は DC による位置登録の処理時間のほか、OpenFlow による packets 転送処理時間や設定した伝送遅延時間などが含まれている。そのため、 $DC_{MN}$  を構築する仮想ホスト上で NTM 端末のプログラムを実行し、同様に位置登録処理を 100 回行った。ループバックアドレスを用いて同一ホスト上の DC を指定し、OpenFlow による packets 転送処理と伝送遅延を含まない位置登録処理時間を取得した。MN からの位置登録処理時間から伝送遅延と packets 転送処理を含まない位置登録処理時間を引くことで、OpenFlow による packets 転送処理に要した時間を算出した。

提案手法では OpenFlow による転送ルール追加処理に 34.8ms ほど要しており、また位置登録処理に要した時間も

10.4ms ほど増加している。ISP のネットワークエッジへの配置を想定しているため、伝送遅延は減少しているが、これらの処理時間の増加に伴い全体で 34.2ms ほど増加している。しかし、位置登録は端末起動時および移動後などに一度だけ行われる処理であるため、通信ごとに必要となるトンネル構築シグナリング処理に比べ影響は少ないと考えられる。また Web パフォーマンスの指標である RAIL[10]において、ユーザからのアクションに対し、100ms 以内にレスポンスを返すことが可能である場合、ユーザはすぐに結果を得られたと感じている。提案手法の位置登録処理においても、処理時間は 100ms 以内に収まっていることから、位置登録処理時間の増加はユーザにほとんど影響を与えず、実用上問題ないと考えられる。

## 4.3 シグナリング

MN と CN をそれぞれ  $DC_{MN}$  と  $DC_{CN}$  に対して位置登録を行った後に、MN から CN に対し経路指示と RS 経由のトンネル構築からなるシグナリングを 50 回行い、要した時間の平均値を算出した。このとき、MN と  $DC_{MN}$  間、CN と  $DC_{CN}$  間、 $DC_{MN}$  と  $DC_{CN}$  間、 $DC_{MN}$  と RS 間の packets 転送ルールは OVS に登録されているものとした。これは各 NTM 端末とそれぞれの DC の間では位置登録処理が行われ、DC 間ではブロックチェーンの伝搬が生じ、 $DC_{MN}$  は起動時に RS に対して接続を行うことから、シグナリング処理を行う以前より通信が行われており、packets 転送ルールの追加が生じないためである。また一連のシグナリング処理によって各 OVS には MN と RS 間、CN と RS 間の転送ルールが追加されるが、このルールは一度計測するたびに削除している。

経路指示の計測は、MN が  $DC_{MN}$  に対し Direction Request を送信した時間と、MN が  $DC_{MN}$  から Route Direction を受信した時間の差分を全体の処理時間とした。また、各 NTM Mobile サーバや CN が packets を受け取った時間と送信した時間を出力し、シーケンスと照らし合わせて差分を求めることにより、各処理時間を算出した。全体からこれらの処理時間と設定した伝送遅延を差し引くことにより、プログラムが packets を送信してから実際に仮想ホストが packets を送信するまでに生じるオーバーヘッドや OVS における転送ルールの追加処理などを求めた。表 3 にシグナリングに要した各処理時間の計測結果を示す。

表 3 より、提案手法における伝送遅延が全体に占める割合は 97.5% 程度である。NTM Mobile サーバや NTM 端末による処理時間の合計は 10.2ms であり、従来手法と比べ 1.2ms ほど増加し全体に占める割合は 2.8% 程度であった。

トンネル構築処理における Tunnel Request と Hole Punching の通信の際に転送ルールの追加が行われるが、トンネル構築処理におけるその他の処理時間は、転送ルールの追加が生じた 4.2 で得られた値に比べ小さな値となって

表 3 トンネル構築シグナリングに要した時間

	従来手法 [ms]	提案手法 [ms]
DC <sub>MN</sub> の処理	2.2	4.0
RS の処理	0.4	0.4
DC <sub>CN</sub> の処理	0.6	0.5
CN の処理	0.7	0.8
伝送遅延	521.3	360.0
その他	—	3.4
合計	525.2	369.1

いる。OVS の転送ルールには処理したパケット数などが記録されているため転送ルールのテーブルを参照したところ、シグナリング処理の前後で新たに転送ルールが追加されたほか、ARP パケット処理用の転送ルールにおけるパケット処理数が増加していた。これは ARP パケットの転送を OpenFlow Controller が処理することを防ぐために OpenDaylight 起動時に OVS へ入力される転送ルールである。この結果からそれぞれのパケットのうち最初の数パケットが上記転送ルールによって処理されたのち、残りのパケットが追加された転送ルールによって転送されていたためにこのような結果になったと考えられる。

NTMobile サーバの処理時間は低下しており、また SDN を用いることで処理時間が追加されているものの全体に占める割合は非常に低い。また、提案手法の計測はシミュレータで行われたものであり従来手法の計測時のような輻輳による遅延を考慮できていないため、実ネットワークを用いた場合、今回得られた結果よりも悪化する可能性がある。しかし、仮に提案手法の伝送遅延が輻輳等により従来手法と同等の値まで悪化した場合においても、従来手法と同等程度の時間でシグナリングを行うことが可能である。

#### 4.4 トンネル通信

シグナリング処理で構築された RS 経由のトンネルを用いて MN と CN 間で通信を行った。まず MN からエンド鍵によって暗号化された「PING」という文字列を送信する。CN は文字列を受信して復号し、「PING」であれば MN に対し「ECHO」という文字列を同様にエンド鍵を用いて暗号化し送信する。MN が文字列を受信し、復号を行う。このトンネル通信を 50 回行い、要した時間の平均値を取得した。このとき、MN から RS 間、CN から RS 間のパケット転送ルールは OVS に登録されている。これは RS 経由のトンネルを構築する際に Tunnel Request などの通信が行われており、パケット転送ルールの追加が生じないためである。

この結果を表 4 に示す。トンネル通信の計測はプログラムにおいて MN が「PING」というメッセージを送信した時間と「ECHO」というメッセージを受信した時間を出力し、求められた差分を全体の処理時間とした。RS にお

表 4 トンネル通信に要した時間

	従来手法 [ms]	提案手法 [ms]
RS の処理	0.2	0.3
CN の処理	0.2	0.1
伝送遅延	195.1	130.0
その他	—	0.5
合計	195.5	131.0

るパケットを受信した時間から中継相手へ送信した時間までを出力し、差分を RS の処理時間とした。CN における「PING」メッセージを受信した時間から「ECHO」メッセージを送信した時間までを出力し、CN の処理時間とした。その他の処理時間は 4.3 と同様に算出しているが、OVS における転送ルールの追加は行われていないため、仮想ホストがパケットを送信する際などに生じるオーバーヘッドなどの値となる。

提案手法のトンネル通信においては全体の 99.2%が通信によって占められており、NTMobile による処理は 0.3%程度であった。また、トンネル通信を行う際には転送ルールの追加が行われなかったため、その他の処理時間が全体に占める割合も 0.4%程度であった。よって、4.3 と同様に輻輳の影響を受けたとしても従来手法と同等程度の時間でトンネル通信を行うことが可能である。

#### 4.5 評価

以上より、提案手法においても従来手法と同様に、問題なく位置登録処理やシグナリング処理、トンネル通信を行うことが可能である。さらに vCPE へ実装することにより、NFV を用いた負荷分散や RS を経由する際に生じる経路冗長化の解決が実現され、DC は NTM 端末に対して最適な RS を容易に選択できることから、提案手法によって NTMobile サーバの運用を効率化することが可能であると考えられる。

### 5. まとめ

本稿では、NTMobile サーバ群を vCPE 上で稼働させて運用効率化を図った提案手法について、Mininet を用いて作成した仮想ネットワークを用いて性能評価を含む基礎的検証を行った。NTM 端末の移動登録処理、トンネル構築シグナリング、トンネル通信についてそれぞれ処理時間を計測した結果、エンド端末間の通信に関わるトンネル構築処理時間とトンネル通信処理時間を削減することができ、提案手法は従来手法より運用効率化を図れることを確認した。

今後は従来手法と異なり、vCPE というユーザに意識されない設備上で稼働する DC に対して位置登録を行うために処理内容の再検討を行い、また NFV を用いた負荷分散の実現を行う予定である。

謝辞 本研究の一部は名城大学学術研究奨励助成制度  
(基礎的研究促進事業費) の助成を受けたものである。

## 参考文献

- [1] 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, oct 2013.
- [2] A. Kimura, S. Kawano, H. Tsuchiya, S. Homma, and A. Okada. Evaluation of virtual customer premises equipment prototype system with open source software. In *2018 IEEE 7th International Conference on Cloud Networking (CloudNet)*, pp. 1–3, 2018.
- [3] 加藤宏理, 松岡穂, 柳瀬知広, 鈴木秀和, 内藤克浩, 渡邊晃. vcpe を利用した ntmobile システムの運用効率化に関する検討. Technical Report 2, 名城大学理工学部, 名城大学大学院理工学研究科, 名城大学大学院理工学研究科, 名城大学理工学部, 愛知工業大学情報科学部, 名城大学理工学部, nov 2019.
- [4] 加藤宏理, 松岡穂, 柳瀬知広, 鈴木秀和, 内藤克浩, 渡邊晃. vcpe を利用した ntmobile システムの運用効率化に関する提案. 第 82 回全国大会講演論文集, 第 2020 巻, pp. 95–96, feb 2020.
- [5] ETSI. Network functions virtualisation (nfv). <https://www.etsi.org/technologies/nfv>.
- [6] Mininet. Mininet. <http://mininet.org>.
- [7] 木村信裕, 柳瀬知広, 田中久順, 鈴木秀和, 内藤克浩, 渡邊晃. ブロックチェーン上のスマートコントラクトを利用した ntmobile システムの基礎的検証. 第 81 回全国大会講演論文集, 第 2019 巻, pp. 65–66, feb 2019.
- [8] Yuka Miyake, Hidekazu Suzuki, Katsuhiro Naito, and Akira Watanabe. Proposal and implementation of a new method of selecting the optimal relay server using ntmobile. In *Proceedings of The Ninth International Conference on Mobile Computing and Ubiquitous Networking (ICMU 2016)*, Vol. 2016, pp. 1–2, sep 2016.
- [9] 菊地俊介, 佐々木健, 大木裕介, 松本亮介. ノード間通信の優位性評価を目的とした fog コンピューティングテストベッドの構築に関する一検討. マルチメディア, 分散協調とモバイルシンポジウム 2019 論文集, 第 2019 巻, pp. 97–104, jun 2019.
- [10] Google Developers. Rail モデルでパフォーマンスを計測する. <https://developers.google.com/web/fundamentals/performance/rail>.