Vol.2019-MBL-93 No.2
Vol.2019-ITS-79 No.2
2019/11/20

vCPEを利用したNTMobileシステムの運用効率化に関する検討

加藤 宏理[1,a)]　松岡 穂[2]　柳瀬 知広[2]　鈴木 秀和[1,b)]　内藤 克浩[3]　渡邊 晃[1]

概要：IPv4/IPv6 の混在や NAT の導入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）がある．NTMobile では専用のサーバ群を利用して通信開始時にエンドツーエンド暗号化トンネルを構築するシグナリング処理や，IPv4/IPv6 間でパケットを中継する必要がある．そのため，サーバ群の設置場所に伴う通信開始時のオーバヘッド増加や，最適な中継サーバの選択方法などの課題があった．これらの課題を解決するために，本稿では NTMobile のサーバ群を仮想化し，ISP のエッジに設置される vCPE（Virtual Customer Premises Equipment）に配置する方式を検討する．これにより，シグナリング時間の短縮や最適な通信経路の実現に加えて，QoS 機能の実現など，NTMobile システムの運用効率化を図る．

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 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．

この課題を複雑化したネットワークにおいて解決し，通信接続性と移動透過性を同時に実現する技術として NTMobile（Network Traversal with Mobility）がある [1,2]．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．NTMobile における代表的な専用サーバとして，仮想 IP アドレスの管理および配布，さらにはエンド端末間のトンネル経路構築指示を行う DC（Direction Coordinator）と，IPv4/IPv6 デュアルスタックネットワークに設置される RS（Relay Server）がある．RS を中継することにより，IPv4 と IPv6 の差を吸収し，バージョンの違いを意識しない双方向通信を実現する．

しかし，従来の NTMobile ではこれらの機能を実現するために，エンド端末は通信開始時に必ず DC と事前に UDP トンネルを構築するためのシグナリング処理を行う必要がある．そのため，DC が高負荷状態であったり，エンド端末との間のネットワークが輻輳していたりすると，通信遅延が増加してしまう．さらに，高スループットを実現できる最適な RS を選択しようとすると，通信経路が冗長になってしまい，通信遅延が増加してしまう可能性も考えられる．従って，最適な RS を選択する手法が検討されているものの，実現には至っていない [3]．

この課題を解決するため，本稿では NTMobile サーバ群を仮想化し，vCPE（Virtual Customer Premises Equipment）への実装を検討する．vCPE は ISP（Internet Service Provider）のネットワークエッジに設置される汎用サーバであり，ホームネットワークや公衆無線 LAN などに設置されている CPE におけるネットワーク機能を仮想化して代行する仕組みである．そのため，vCPE 上で動作する DC や RS とエンド端末との間の通信遅延の削減が可能であり，またエンド端末間の通信経路上に存在するため通信経路の冗長化の解決が可能である．更に vCPE の持つ仮想マシン管理機能を用いて負荷分散を実現することにより，常に最適な NTMobile サーバの選択が可能となる．

[1] 名城大学理工学部
Department of Science and Technology, Meijo University
[2] 名城大学大学院理工学研究科
Graduate School of Science and Technology, Meijo University
[3] 愛知工業大学情報科学部
Department of Information Science, Aichi Institute of Technology
[a)] hirotoshi.kato@ucl.meijo-u.ac.jp
[b)] hsuzuki@meijo-u.ac.jp

© 2019 Information Processing Society of Japan

以下，2 章では NTMobile の概要と課題を示し，3 章では本稿で検討する手法について述べ，4 章で検討手法の評価を行い，5 章でまとめる．

2. 従来の NTMobile

2.1 概要

NTMobile のシステム構成を図 **1** に示す．NTMobile は DC（Direction Coordinator），RS（Relay Server）といったサーバと NTMobile が実装された端末（以下，NTM 端末）によって構成される．また，システムにおいて発信側の NTM 端末を MN（Mobile Node），受信側の NTM 端末を CN（Correspondent Node）と呼称する．

DC は各 NTM 端末に対する仮想アドレスの割り当てとブロックチェーンを用いた NTM 端末の管理を行い，NTM 端末の要求に応じて NTMobile 通信に用いる UDP トンネルの構築指示を行う．RS は通信を行う NTM 端末同士が異なる NAT の配下にある場合や，それぞれの端末が IPv4 と IPv6 のネットワークに属している場合など，直接通信が行えない状況においてパケットを中継する役割を持つ．NTM 端末は NTMobile 用の端末情報として楕円曲線暗号の秘密鍵，公開鍵，公開鍵のハッシュにより求められる ID，署名を持つ．端末上のアプリケーションは DC によって割り当てられた仮想アドレスを使用することで実 IP アドレスの変化を隠蔽し，実アドレスによってカプセル化され UDP トンネルを用いたトンネル通信を行う．これにより端末の移動や実ネットワークの構成に影響されない通信が可能となる．

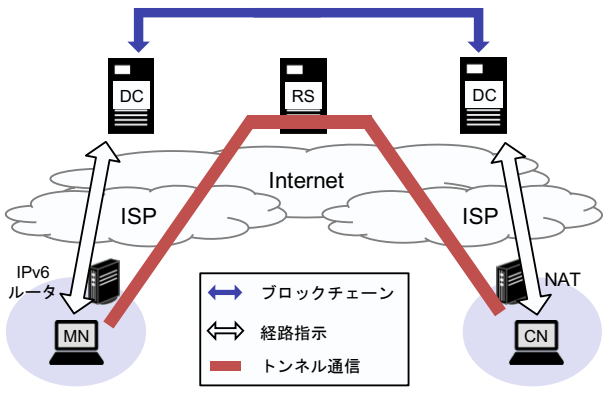


図 **1** NTMobile の概要

2.2 DC による仮想アドレスの割り当てと端末管理

NAT の配下に存在する NTM 端末が DC に対し仮想アドレスの割り当てを要求する際のシーケンスを図 **2** に示す．

NTM 端末は自身の ID と署名，自身を管理する DC と失効フラグやタイムスタンプをフィールドとして持つ Node Transaction を作成し，DC に対して送信する．これが Registration Request である．DC は受け取った Node Transaction を検証することで端末の認証を行う．正しい場合は仮想アドレスを Registration Response に格納して 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でブロックチェーンによる管理を行う．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．

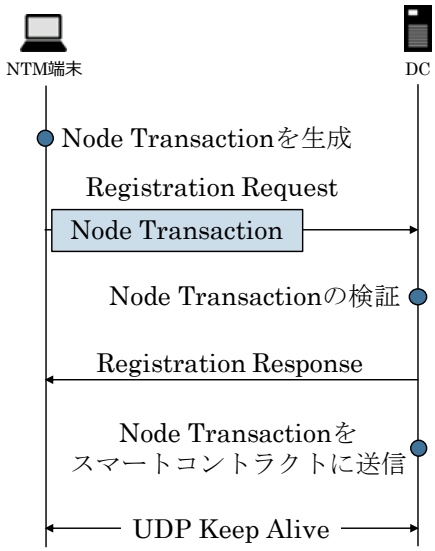


図 **2** 端末登録処理

2.3 NTMobile 通信開始時の処理

IPv6 ネットワークに属する MN が NAT の配下に存在する CN へ NTM 通信を開始する際のシーケンスを図 **3** に，MN と CN が RS を経由したトンネルを構築するシーケンスを図 **4** に示す．

MN と CN に仮想アドレスを割り当てた DC をそれぞれ DC_{MN}，DC_{CN} と呼称する．MN は DC_{MN} に対し Direction Request として CN の ID を送信する．DC_{MN} はスマートコントラクトからブロックチェーンに記録された Node Transaction を検索し，CN の ID から DC_{CN} を特定する．DC_{MN} は DC_{CN} へ CN の端末情報を要求し，取得する．その後自身の保持する MN の情報を加味した UDP トンネルの経路を計算する．計算の結果得られた経路を $DC_{M}N$ は MN と CN に対して送信し，受信した両端末は指示に従いトンネル構築を行う．このとき MN と CN はそれぞれの DC から相手端末の公開鍵が送信される．

トンネル構築において，CN は NAT の配下に存在するため Hole Punching を行い，RS を経由したトンネル構築のため通信を行う．受信した相手端末の公開鍵と自端末の秘密鍵を用いて Diffie-Hellman 鍵共有を行い，生成された鍵によって両端末が構築するトンネルは暗号化される．

© 2019 Information Processing Society of Japan

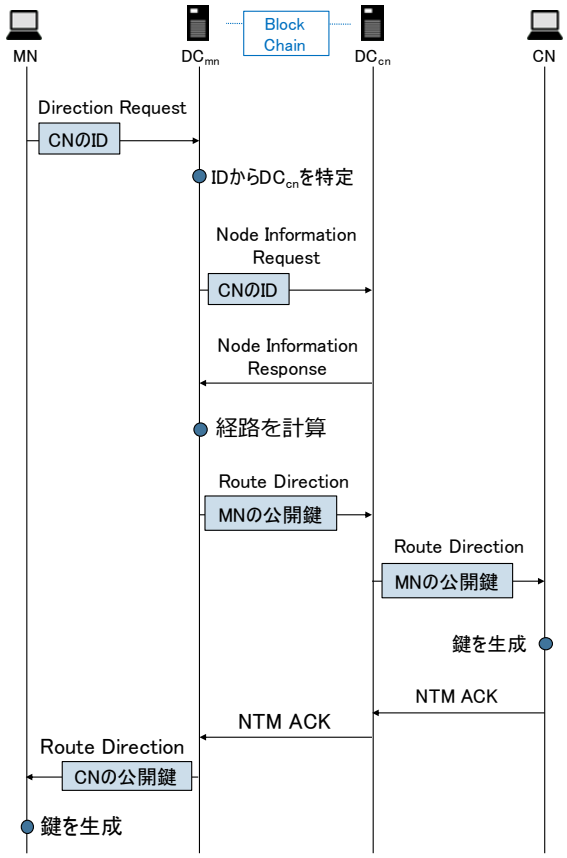


図 3　通信開始時の処理

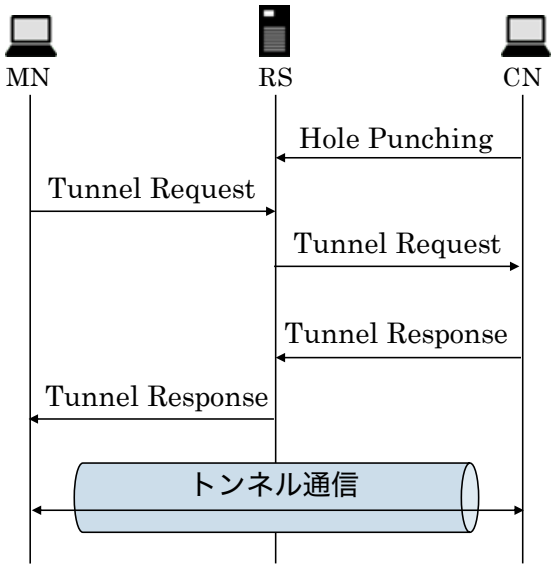


図 4　トンネル構築処理

2.4　課題

2.2 節や 2.3 節において，NTM 端末は DC と複数回の通信を行う．文献 [2] では MN と CN はそれぞれ日本とアメリカに設置された MacBook Pro で稼働し，DC_{MN} と RS は日本に設置された AWS サーバにて稼働し，DC_{CN} はアメリカに設置された AWS サーバにて稼働する環境にて性能評価を行っているが，2.3 の処理において処理時間全体のおよそ 85%が伝送時間が占めており，エンド端末と NTMobile サーバ群間の RTT（Round Trip Time）が大きい環境ではトンネル構築処理に伴うオーバヘッドが増大してしまうという課題がある．

また MN と CN が直接通信できない場合 DC は RS を経由した経路を構築するが，このとき DC は RS の負荷や MN，CN からの通信時間といった要素を考慮し，ネットワーク上に複数存在する RS から最適なサーバを選択する必要がある．更に RS によってパケットが中継されることで，直接通信を行う場合に比べ経路が冗長化し，MN と CN の通信時間に RS までの通信遅延が加算されてしまうという課題がある．

3.　検討手法

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．本稿ではこのアプローチと同様に，NTMobile サーバ群を vCPE へ実装することで NTMobile に存在する課題解決を検討する．vCPE とは，従来 CPE（Customer Premises Equipment）という形でユーザ側に設置された専用のハードウェアにより実現されていたネットワーク機能を，仮想マシンとして実行するプロバイダのネットワークエッジに設置された汎用サーバである．vCPE の実現には SDN（Software Defined Network）[4] と NFV（Network Functions Virtualization）[5] というネットワーク仮想化技術が利用され，仮想マシン間の柔軟なネットワークの構築と仮想化されたネットワーク機能の管理を実現する．

検討手法の概要を**図 5** に示す．NTMobile 通信を行う際に NTM 端末と NTMobile サーバ群との間で行われる通信は検討手法においても従来手法と同様であるが，NTMobile サーバ群が vCPE 上に実装されることでオーバヘッドが削減される．どの程度の削減が可能であるかという試算は 4 章にて行う．また，vCPE はユーザからのパケットが必ず到達する地点であり，到達したパケットを処理してネットワークに転送する役割を持つ．そのため RS を経由する通信においても両端末が直接通信する場合と同じ経路での通信が実現できる．更に vCPE が有するリソース管理やライフサイクル管理といった管理機能を利用し，負荷に応じた NTMobile サーバのオートスケーリングが実現できる．この機能を用いることで NTMobile サーバのインスタンスを複数配置し負荷を分散させ，遅延だけでなく負荷の面でも最適なサーバと通信を行うことが可能となる．

3.2　vCPE への実装により実現可能な機能の検討

NTMobile サーバ群を vCPE へ実装することで NTMobile に存在する課題を解決することを検討したが，本節では更に vCPE への実装により実現可能な機能について検討を行う．

© 2019 Information Processing Society of Japan

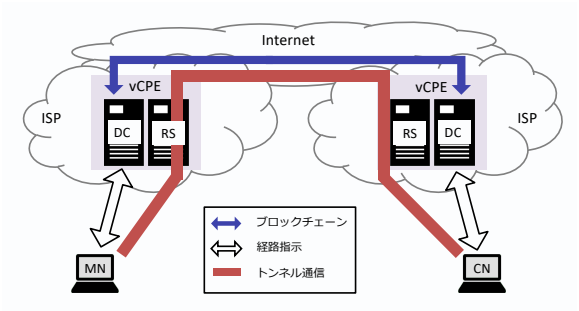


図 5　検討手法の概要

3.2.1　NTMobile サーバによる QoS 制御

SDN においては，SDN コントローラの API を用いることで外部からパケット制御ルールの設定が行える．これを利用し，特定のユーザや特定の通信のパケットを通信速度の速いポートへ割り振る設定を行うことで，帯域の優先をはじめとする QoS 機能が実現可能となる．

3.2.2　非 NTM 端末と NTM 端末間の NTMobile 通信

これは NTMobile を利用し宅外の端末を用いて宅内のスマート家電を操作する先行研究 [6] を応用し，vCPE 配下の NTMobile 未実装端末と外部の NTM 端末に適用するものである．先行研究においては NTMobile の実装が困難なスマート家電と NTM 端末が NTMobile 通信を行うため，中継を行う RCA（Remote Control Agent）が宅内に設置された．この RCA と同様に，NTMobile に関する暗号化やカプセル化などの処理を代行するエージェントを vCPE 上に実装し，NTMobile 通信を中継することで未実装端末と NTM 端末による NTMobile 通信行う．

更にこれを発展させることで，NTMobile 未実装端末同士がお互いの vCPE に実装されたエージェントにより NTMobile 通信を行うことも可能であると考えた．

4.　評価

vCPE に NTMobile サーバを実装することでどの程度の改善が可能であるか試算するため，従来手法の検証によって得られた RTT と，検討手法において試算された RTT を比較し，**表 1** に示す．検討課題の環境は 2.4 節において示したものと同様であるが，NTMobile サーバは vCPE に実装されているものとし，同程度のリソースを割り当てられたものとする．vCPE 上へ実装された NTMobile サーバまでの RTT は．一般的にエッジサーバまでの RTT が最良で数 ms であることから，15ms として試算した．次に表 1 の値を元に検討手法における 2.2 節と 2.3 節の処理時間を試算し，従来手法の検証の際に得られた値と比較したもの**表 2** をに示す．以上より従来手法の処理時間全体に対し検討手法では 16.3%程度の通信時間削減が可能であると考えられる．

表 1　従来手法と検討手法の RTT

	従来手法における RTT（ms）	検討手法における RTT の試算（ms）
MN-DC_{MN}	21	15
MN-RS	22	15
DC_{MN}-DC_{CN}	101	101
CN-DC_{CN}	41	15
CN-RS	162	116

表 2　従来手法と検討手法の処理時間比較

	従来手法における処理時間（ms）	検討手法における処理時間（ms）
2.2 節の処理	32.3	26.3
2.3 節の処理	525.2	440.2

5.　まとめ

本稿では vCPE 上への NTMobile サーバ群の実装を検討した．検討手法によって NTMobile の課題となっていた通信時間の増大と RS 経由時の経路冗長化が解決可能であることを示し，更に QoS 機能や NTMobile 未実装端末との通信など vCPE への実装によって実現可能となる機能の検討を行った．

今後は SDN と NFV を実現する OpenFlow [4] や OpenStack [7] などを用いて vCPE を作成し，NTMobile の実装と検証を行う予定である．

参考文献

[1] 上醉尾一真，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] 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，Vol. 2019, No. 1, pp. 65–66 (2019).
[3] Miyake, Y., Suzuki, H., Naito, K. and Watanabe, A.: Proposal and Implementation of a New Method of Selecting the Optimal Relay Server Using NTMobile, *Proceedings of The Ninth International Conference on Mobile Computing and Ubiquitous Networking (ICMU 2016)*, Vol. 2016, No. 1, pp. 1–2 (2016).
[4] ONF: Open Networking Foundation. `https://www.opennetworking.org`.
[5] ETSI: Network Functions Virtualisation (NFV). `https://www.etsi.org/technologies/nfv`.
[6] 田中久順，鈴木秀和，内藤克浩，渡邊　晃：ECHONET Lite 機器のセキュア遠隔制御システムの評価，技術報告 5，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, 愛知工業大学情報科学部, 名城大学大学院理工学研究科 (2018).
[7] openstack: openstack. `https://www.openstack.org`.

© 2019 Information Processing Society of Japan