Vol.2023-DPS-195 No.35
Vol.2023-MBL-107 No.35
Vol.2023-ITS-93 No.35
2023/5/19

CYPHONICにおける NAPT越え時の通信経路最適化に関する設計及び実装

眞玉 和茂[1,a)]　小村 聖[,b)]　後藤 廉[1,c)]　鈴木 秀和[2,d)]　内藤 克浩[3,e)]

概要：現在のインターネットでは, Network Address Port Translation (NAPT) や Internet Protocol (IP) のバージョンの違いなどが原因となり, デバイスが直接通信できないケースが増加している. 一方, 近年のサービス形態は多様化しており, 一部サービスでは, クライアントサーバー型の通信よりも, Peer-to-Peer (P2P) 型の通信が親和性が高いことも増えつつある. 著者らは, IP 通信における通信接続性や移動透過性を実現する技術として, CYber PHysical Overlay Network over Internet Communication (CYPHONIC) を提案してきた. CYPHONIC では, クラウドサービスがデバイスの通信を中継することにより, NAPT 越えを実現している. しかし, 通信の中継に伴う, クラウドサービスへの負荷集中や経路冗長化が課題として残されている. 本稿では, クラウドサービスを用いて各デバイスを接続した後, 特定の条件においてのみ通信経路を直接通信に切り替える経路最適化手法を提案する. 提案手法では, NAPT におけるポート変換技術にはいくつかの種類が存在し, その組み合わせによってはデバイス間の直接通信が可能であることに着目する. 提案手法の導入に伴い, 既存の CYPHONIC の通信シグナリングを拡張することにより, クラウドを経由して通信を確立した後に, 相互のデバイスが直接通信可能なのかを確認する機能を追加する. 評価実験では, NAPT の種類の組み合わせにより, どのような条件において提案手法が有効に働くのかについて確認を行い, 経路最適化手法の有効性を明らかにする.

キーワード：NAPT, 通信接続性, 移動透過性, オーバーレイネットワーク, IoT

1. はじめに

近年, Internet of Things (IoT) を活用したサービスが注目される. IoT サービスは, IoT デバイスが相互に連携したシステムを示し, スマートホームやスマートファクトリー等の多岐にわたる分野で活用される [1,2]. 端末の連携に利用されるネットワークモデルには, クライアントサーバ型と Peer-to-Peer (P2P) 型が存在する [3,4]. クライアントサーバ型は, 情報の管理やサービスの提供を行うサーバと, 提供されるサービスを利用するクライアントで構成される. クライアントサーバ型は, 常にサーバを経由した通信を行う必要があるため, クライアント間の通信時における経路の冗長化が問題視される. また, 利用クライアントの増加に伴い, サーバへの負荷集中が懸念される. P2P 型は, サーバの機能を併せ持つクライアントで構成されるため, 処理の一極集中を軽減し, 最短経路での通信が可能となる. しかし, 通信プロトコルの影響により, P2P 型での通信が困難な場合がある.

IoT デバイスがインターネット上での通信を行う際, Internet Protocol (IP) を利用する [5]. IP 通信では, 端末の識別子として一般に IPv4 アドレスが利用される. しかし, IPv4 はアドレスの数が 2^{32} 個のみであるため, デバイスの増加に伴い, IP アドレスの枯渇が懸念された [6].

IP アドレスの枯渇問題に対して, Network Address Port Translation (NAPT) や IPv6 が用いられる [7,8]. NAPT は, 内部ネットワークで利用するプライベート IP アドレスを, インターネット上で利用可能なグローバル IP アドレスに変換する機能を持つ. NAPT を導入することにより, 1 つのグローバル IP アドレスを多数の端末で共有する

[1] 愛知工業大学大学院経営情報科学研究科
Graduate School of Business Administration and Computer Science,
Aichi Institute of Technology, Nagoya, Aichi 464-0807, Japan
[2] 名城大学大学院理工学研究科
Graduate School of Science and Technology, Meijo University, Nagoya, Aichi 468-0073, Japan
[3] 愛知工業大学情報科学部
Faculty of Information Science,
Aichi Institute of Technology, Nagoya, Aichi 464-0807, Japan
[a)] matama@pluslab.org
[b)] hjr3ikmr@pluslab.org
[c)] r0719en@pluslab.org
[d)] hsuzuki@meijo-u.ac.jp
[e)] naito@pluslab.org

Vol.2023-DPS-195 No.35
Vol.2023-MBL-107 No.35
Vol.2023-ITS-93 No.35
2023/5/19

ことが可能となる．一方で，NAPT は外部ネットワークから NAPT 配下に存在する端末を隠蔽する．そのため，外部ネットワークに存在する端末は，NAPT 配下に存在する端末への通信開始が困難になる．この問題は，NAPT 越え問題と呼ばれる [9]．

IPv6 は，アドレスの数が 2^{128} 個あるため，アドレスの枯渇問題に対処可能である．しかし，IPv4 と IPv6 はパケットの構造が異なり，互換性がないため，相互の通信が困難である [10]．また，現在のネットワーク環境では，IPv4 から IPv6 への移行が完了していないため，IPv4 と IPv6 が混在している．

また，近年の端末に着目すると，スマートフォン等の無線端末は複数のインターフェースを備え，インターフェースを切り替えてネットワークを移動することが可能である．しかし，IP は端末の移動が考慮されていないため，ネットワークの移動に伴い，トランスポート層の通信が切断される課題があり，移動透過性の問題と呼ばれる．

以上より，端末間での相互通信を利用したサービスの活用には，これらの課題を包括的に解決する必要がある．筆者らは，上記の課題を解決する技術として，CYber PHysical Overlay Network over Internet Communication (CYPHONIC) と呼ばれる技術の提案及び開発を行なってきた [11, 12]．CYPHONIC は，通信端末に仮想の IP アドレスを割り当てることにより，実ネットワークの影響を隠蔽するオーバーレイネットワークを提供する．既存の CYPHONIC は，クラウドサービスが通信する両端末のネットワーク情報を管理・連携する事で，NAPT 越えを実現する．また，通信する両端末が NAPT 配下に存在する場合は，クラウドサービスが通信を中継する．しかし，NAPT の種類や組み合わせによっては，両端末間での直接通信が可能な場合があるため，経路冗長化の問題がある．

本研究では，クラウドサービスが中継を行う通信において，直接通信が可能な場合の経路最適化処理を提案する．提案手法では，NAPT の種類毎の特性に着目し，直接通信での NAPT 越えが可能な場合に，経路の切り替えを行う．NAPT には，IP アドレスとポート番号を変換する規則が複数パターン存在する．また，変換規則の中には，一定の条件を満たすことにより，外部のネットワークからの通信が可能な場合が存在する．そこで，NAPT の種類と組み合わせにおいて，一定の処理を行うことにより，直接通信による NAPT 越えを実現する．

2. CYPHONIC の概要

CYPHONIC では，通信を行うエンド端末間のネットワーク上に UDP トンネルを構築する．エンド端末は UDP トンネルを介した通信を行うことにより，安全な End to End (E2E) 通信が可能である．図 1 に CYPHONIC の概要図を示す．CYPHONIC は，クラウドサービスである CYPHONIC Cloud とエンド端末である CYPHONIC Node から構成される．CYPHONIC Cloud は，以下の 3 種類のサービスから構成される．

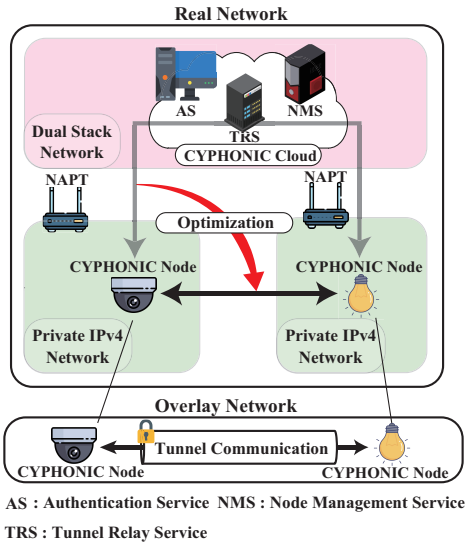

図 1 CYPHONIC の概要

- Authentication Service (AS)
  AS は，CYPHONIC Node のアカウント情報を管理し，CYPHONIC Node の認証処理を行う．また，AS は CYPHONIC Node の識別子である Full Qualified Domain Name (FQDN) と仮想 IP アドレスを割り当て，NMS との暗号化通信で利用する共通鍵を配布する．
- Node Management Service (NMS)
  NMS は，CYPHONIC Node の IP アドレスやポート番号等のネットワーク情報を管理し，CYPHONIC Node 間のトンネル構築処理を制御する．
- Tunnel Relay Service (TRS)
  TRS は，NAPT 間の通信や IPv4/IPv6 間の通信など，直接通信が困難な CYPHONIC Node 間の通信を中継する．

CYPHONIC Node は，CYPHONIC Cloud と連携し，相手の CYPHONIC Node との E2E 通信を行う端末である．CYPHONIC Node は，トンネル通信前の処理とトンネル構築時の処理がある．

トンネル通信前の処理では，認証処理と登録処理を行う．認証処理では，AS と TLS 通信を行うことにより，CYPHONIC Node が正規のユーザであることを証明する．この際，AS から配布された NMS のアドレス情報と暗号鍵を利用して登録処理を行う．登録処理では，NMS と暗号化通信で CYPHONIC Node のネットワーク情報を登録する．

トンネル構築時の処理では，経路選択処理とトンネル構築処理を行う．図 2 は，トンネルが TRS を経由する場合のシグナリングである．経路選択処理では，NMS が通信を行う両 CYPHONIC Node のネットワーク情報を元に通信経路を指示する．また，NMS が経路指示する際に暗号鍵であ

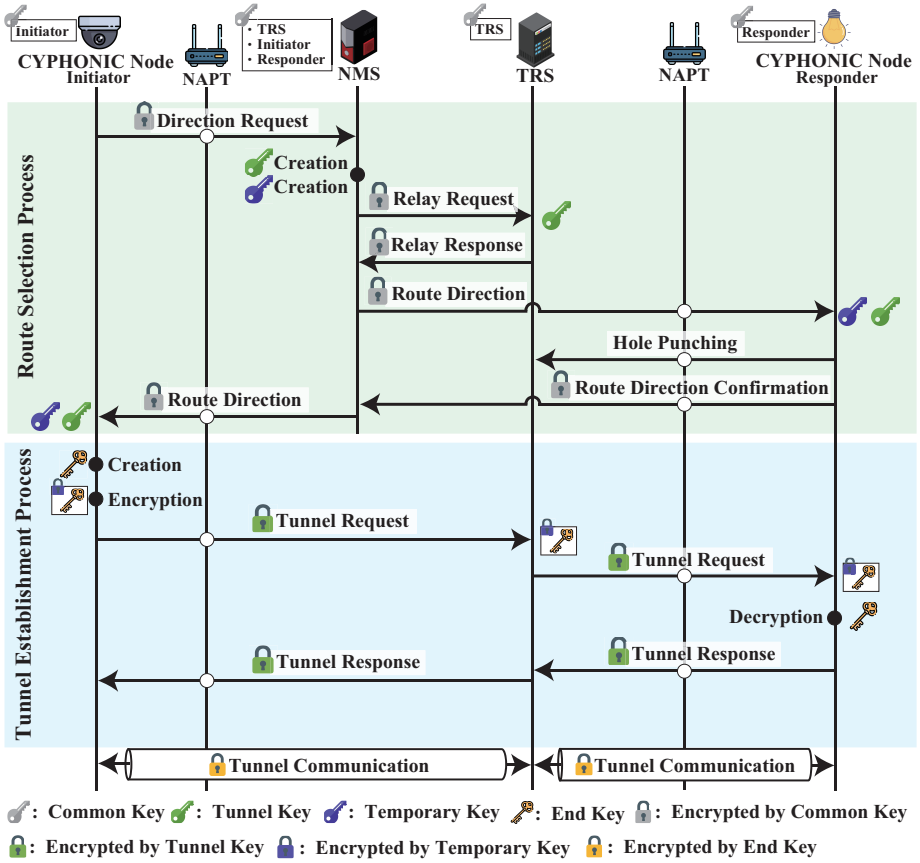

図 2 既存の通信確立処理

る Tunnel Key と Temporary Key を生成し, Tunnel Key を TRS と CYPHONIC Node に, Temporary Key を CYPHONIC Node に配布する. Tunnel Key はトンネル構築処理で利用する通信を暗号化する共通鍵であり, Temporary Key はトンネル通信時に利用する End Key を TRS から秘匿するための共通鍵である. TRS を経由する必要がある場合, TRS から相手先の CYPHONIC Node に対して通信が行えるように, CYPHONIC Node から TRS に対して Hole Punching を行う. トンネル構築処理では, CYPHONIC Node で生成した End Key を Temporary Key で暗号化する. その後, CYPHONIC Node は, Tunnel Key を利用した暗号化通信を用いて, TRS を経由して相手の CYPONIC Node に End Key を配布する.

3. 経路最適化

3.1 概要

本章では, CYPHONIC における NAPT を経由した通信の経路最適化手法について述べる. CYPHONIC では, 通信を行う CYPHONIC Node が双方とも NAPT 配下に存在する場合, TRS が CYPHONIC Node 間の通信を中継することにより NAPT 越えを可能にする. NAPT には, 複数パターンのポート変換規則が存在し, 通信を行う両端末が NAPT 配下に存在する状況においても直接通信が可能な場合がある. しかし, 既存の CYPHONIC では, 両端末が NAPT 配下に存在する場合, 全ての通信において TRS を経由する. したがって, TRS を経由することにより, クラウドを経由するトラフィックの増大を助長するとともに, クラウドを経由する経路の冗長化が問題となる. そのため, 両 CYPHONIC Node 間の通信経路の切り替えを行うことにより, TRS を経由しない通信への経路最適化を行う必要がある. 提案手法では, 経路最適化処理を行うことで, 直接通信が可能な場合に, TRS を経由することなく CYPHONIC Node 間の E2E 通信が可能である. 経路最適化処理では, TRS を経由したトンネルを構築後, CYPHONIC Node 間の E2E 通信が可能であれば, 通信経路の切り替える. 経路最適化では, NAPT の変換情報を相互の CYPHONIC Node が取得することに加え, NAPT の変換テーブルにマッピングを行うことにより, NAPT 配下に存在する CYPHONIC Node 間での E2E 通信を可能にする.

3.2 NAPT の種類

NAPT は, 内部及び外部のアドレスとポート番号を変換する技術である. NAPT には, 外部の宛先を問わずに単一の変換規則を持つ Cone 型の NAPT と, 外部の宛先毎に別の変換規則を持つ Symmetric 型の NAPT が存在する. 各 NAPT の詳細を以下に示す.

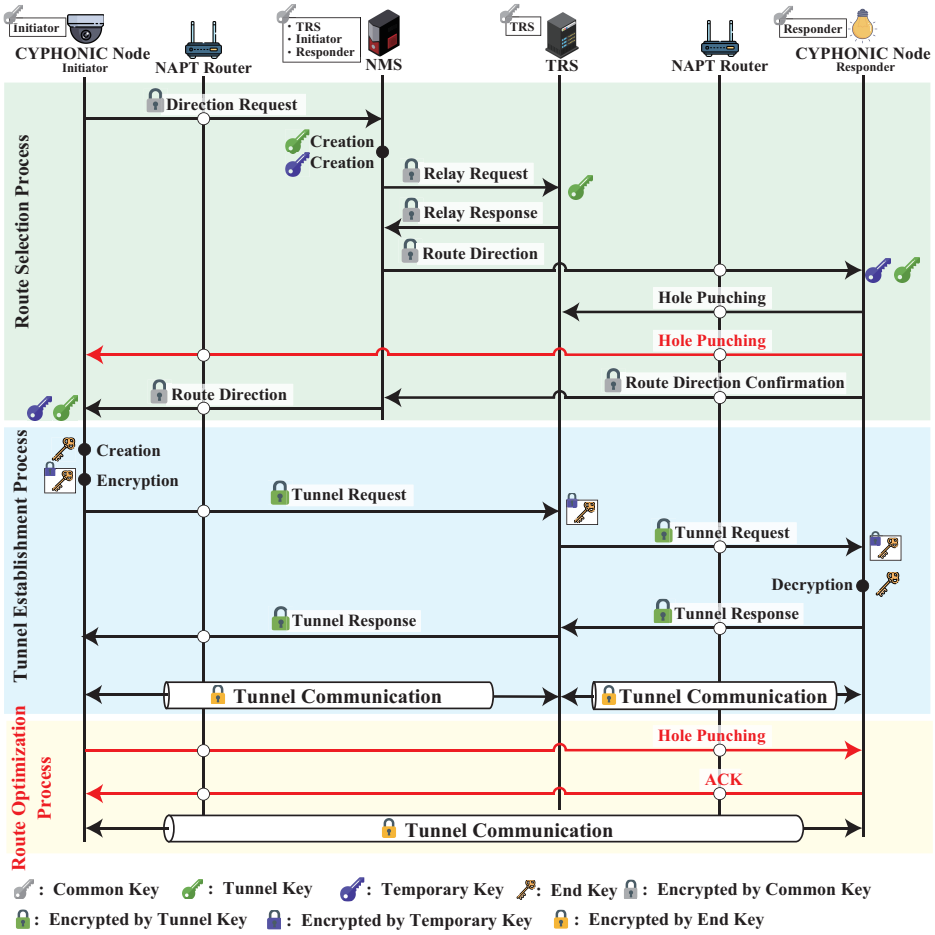

図 3 経路最適化処理

- Full Cone
  Full Cone は, Cone 型の NAPT である. Full Cone は, 一度でも NAPT の変換が行われ, 管理テーブルに対応エントリが存在する場合, 外部のアドレス及びポート番号宛に受信したパケットを対応する内部アドレス及びポート番号に転送する機能を持つ. この時, 一度も送信していない宛先からのパケットを受信した場合においても, パケットを転送する.
- Address-Restricted Cone
  Address-Restricted Cone は, Cone 型の NAPT である. Address-Restricted Cone は, 対応エントリが存在する場合においても, 内側から送信した事のない宛先からのパケットは内側へ転送しない. 内側から送信した事のある宛先からのパケットに限り, 外側の IP アドレス及びポート番号に対応する内部アドレス及びポート番号宛に転送する.
- Port-Restricted Cone
  Port-Restricted Cone は, Cone 型の NAPT である. Port-Restricted Cone は, 対応エントリが存在する場合においても, 内側から送信した事のない宛先からのパケットは内側へ転送しない. 内側から送信した事のある宛先アドレスとポート番号からのパケットに限り, 外側の IP アドレス及びポート番号に対応する内部アドレス及びポート番号宛に転送する.
- Symmetric
  Symmetric は, 最も制約が厳しい NAPT である. Symmetric は, 外側の宛先毎にアドレス及びポート番号の対応付けを行う. そのため, 内側から送信した宛先アドレス及びポート番号からのパケットに限り, 対応する内部アドレス及びポート番号宛に転送する.

3.3 NAPT の組み合わせ

既存の CYPHONIC において TRS を経由する NAPT 環境のパターンとして, 両端末が異なる 1 台の NAPT 配下に存在する場合と, 多段 NAPT と呼ばれる複数台の NAPT 配下に存在する場合, 同一の NAPT 配下に存在する場合がある. そのため, 各環境に対応した NAPT 越えの実現が必要である.

まず, 両端末が異なる 1 台の NAPT 配下に存在する場合, 端末は互いに相手端末側の NAPT の変換情報を取得する必要がある. また, 少なくとも一方の NAPT が Address-Restricted Cone や Port-Restricted Cone の場合, NAPT へのマッピングを行う必要がある. さらに, 少なくとも一方の NAPT が Symmetric の場合, 宛先毎に変換情報が異

なるため, 外側に存在する相手端末が変換情報を直接取得する必要がある.

次に, 少なくとも一方の端末が多段 NAPT 配下に存在する場合, 構成する NAPT の中で最も強い制約が多段 NAPT 全体の制約として扱われる. NAPT は, Full Cone, Address-Restricted Cone, Port-Restricted Cone, Symmetric の順に制約が強くなる. したがって, 多段 NAPT において外側の NAPT が Full Cone であった場合でも, 内側の NAPT が Symmetric である場合, 多段 NAPT 全体の制約は Symmetric として扱われる.

最後に, 両端末が同一 NAPT に存在する場合, 端末は NAPT を経由することなく, NAPT 配下のネットワークで通信を行うことが望ましい. そのため, 同一 NAPT であるかを確認し, 必要に応じて内部の IP アドレスを用いた通信を行う必要がある.

3.4 経路最適化処理

NAPT 越えは, NAPT へのマッピングと, NAPT の外側に存在する端末が NAPT 配下に存在する端末の外側の IP アドレス及びポート番号を取得する事で可能となる. 経路最適化処理における NAPT のマッピングは, NAPT 配下に存在する端末から外側の端末に対して Hole Punching を送信する事で実現する. また, NAPT の外側の IP アドレス及びポート番号の取得は, NMS が保持する端末の外側ネットワーク情報を利用する事で実現する. Cone 型である, Full Cone, Address-Restricted Cone, Port-Restricted Cone の NAPT は, 外部の宛先を問わずに単一の変換規則を持つため, NMS が保持する NAPT の外部側のネットワーク情報を利用することにより, NAPT 越えが可能である. しかし, Symmetric NAPT は, 外側の宛先毎にアドレス及びポート番号の対応付けを行うため, NMS への登録時と別の変換情報になる. そのため, NMS が保持する NAPT の外部側のネットワーク情報を用いて NAPT 越えを行うことは不可能である. したがって, 相手端末が Symmetric NAPT 配下の場合, 相手端末からの Hole Punching パケットの送信元アドレス及びポート番号に対して通信を行うことで対処する. 以上の処理を行うことで, 両端末が多段 NAPT 配下に存在する場合においても直接通信が可能になる. 両端末が同一 NAPT に存在する場合は, 内部のネットワーク間で通信を行う必要がある. そこで, NMS から取得した両端末の外側 IP アドレスが一致する場合は, 内部の IP アドレスでの通信へ切り替える処理を加える.

3.5 シグナリング

図 3 に TRS 経由時のトンネル構築における経路最適化のシグナリングを示す. Responder は, NMS からの Route Direction を受信後, Initiator の外側の IP アドレス及びポート番号に対して Hole Punching を送信する. Initiator

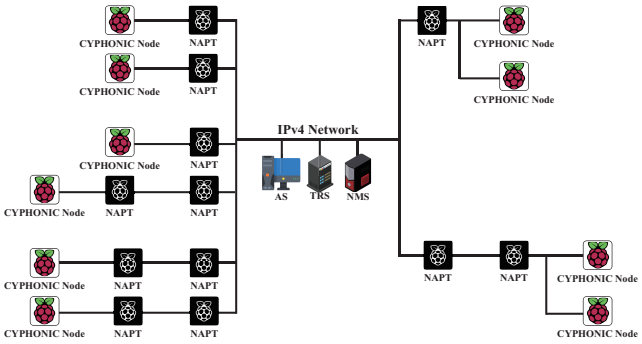

図 4 検証環境

表 1 検証機器

Virtual Machine (AS,NMS,TRS)	
OS	Ubuntu 21.10
CPU	3.50GHz 2cores Intel(R) Core i9-11900K
Memory	2GB RAM
Raspberry Pi 4 Model B (NAPT Router)	
OS	Raspbian GNU/Linux 10.0 (Buster)
CPU	Quad Core 1.5GHz Broadcom BCM2711 64bit
Memory	4GB RAM
Raspberry Pi 3 Model B (CYPHONIC Node)	
OS	Raspbian GNU/Linux 10.0 (Buster)
CPU	Quad Core 1.2GHz Broadcom BCM2837 64bit
Memory	1GB RAM

に Hole Punching を送信することにより, Responder 側の NAPT は Initiator 宛の変換情報を生成する. また, Responder からの Hole Punching を受信した Initiator は, 送信元アドレス及びポート番号を保持する事により, Responder 側の NAPT が Symmetric である場合の直接通信を可能にする. Initiator は, TRS 経由でのトンネル通信を確立した後, 直接通信への経路最適化処理を行う. Initiator は, Responder からの Tunnel Response を受信後, Responder に Hole Punching を送信する. Responder は, Initiator からの Hole Punching を受信後, 送信元アドレス及びポート番号を保持する事で, Initiator 側の NAPT が Symmetric である場合の直接通信を可能にする. また, Responder は, TRS 経由の通信から Initiator との直接通信への経路に変更する. その後, Responder は, Initiator に ACK を送信する. Initiator は, Responder からの ACK を受信後, TRS 経由の通信から Responder との直接通信へ切り替える. 以上の処理により経路最適化の処理は完了する.

4. 検証および評価

本章では, 提案する経路最適化処理の動作検証を行う. 従来の CYPHONIC において, 両 CYPHONIC Node が NAPT 配下に存在する場合, NAPT の種類と組み合わせに限らず TRS が中継を行なっていた. しかし, NAPT の種類や組み合わせによっては直接通信が可能なパターンが存在し, 経路冗長化の問題があった. そのため, 実際のネット

Vol.2023-DPS-195 No.35
Vol.2023-MBL-107 No.35
Vol.2023-ITS-93 No.35
2023/5/19

表 2 両端末：NAPT 1 台

Initiator \ Responder	FC[*1]	ARC[*2]	PRC[*3]	S [*4]
FC[*1]	○	○	○	○
ARC[*2]	○	○	○	△
PRC[*3]	○	○	○	×
S [*4]	○	○	×	×

表 3 Initiator：NAPT 1 台, Responder：NAPT 2 台

Responder \ Initiator		FC[*1]	ARC[*2]	PRC[*3]	S [*4]
NAPT (Outer)	NAPT (Inner)				
FC[*1]	FC[*1]	○	○	○	○
	ARC[*2]	○	○	○	○
	PRC[*3]	○	○	○	×
	S [*4]	○	△	×	×
ARC[*2]	FC[*1]	○	○	○	○
	ARC[*2]	○	○	○	○
	PRC[*3]	○	○	○	×
	S [*4]	○	△	×	×
PRC[*3]	FC[*1]	○	○	○	×
	ARC[*2]	○	○	○	×
	PRC[*3]	○	○	○	×
	S [*4]	○	△	×	×
S [*4]	FC[*1]	○	△	×	×
	ARC[*2]	○	△	×	×
	PRC[*3]	○	△	×	×
	S [*4]	○	△	×	×

ワーク環境において, 各 NAPT のパターンにおける経路最適化処理の動作検証を行うことで, 経路最適化が行われることを確認する.

4.1 検証実験

本稿では, 経路最適化処理を導入し, 実際に NAPT が存在するネットワーク環境での各 NAPT の種類と組み合わせにおける動作確認を行う. 図 4 に検証環境を示す. グローバルネットワークと見做したネットワークに CYPHONIC Cloud である AS, NMS, TRS と NAPT 機器を配置し, NAPT の配下のネットワークに各 CYPHONIC Node を配置する. 表 1 に実装環境を示す. CYPHONIC Cloud の各サービスは仮想マシンで構築し, CYPHONIC Node は Raspberry Pi 3 を用いる. また, NAPT Router は Raspberry Pi 4 で構築する. 計測パターンは, 各 CYPHONIC Node が存在する NAPT を最大 2 個まで増やした際の各組み合わせである. また, CYPHONIC Node が同一 NAPT に存在する場合においての動作検証も行う.

[*1] Full Cone
[*2] Address-Restricted Cone
[*3] Port-Restricted Cone
[*4] Symmetric

表 4 Initiator：NAPT 2 台, Responder：NAPT 1 台

Initiator \ Responder		FC[*1]	ARC[*2]	PRC[*3]	S [*4]
NAPT (Outer)	NAPT (Inner)				
FC[*1]	FC[*1]	○	○	○	○
	ARC[*2]	○	○	○	△
	PRC[*3]	○	○	○	×
	S [*4]	○	○	×	×
ARC[*2]	FC[*1]	○	○	○	△
	ARC[*2]	○	○	○	△
	PRC[*3]	○	○	○	×
	S [*4]	○	○	×	×
PRC[*3]	FC[*1]	○	○	○	×
	ARC[*2]	○	○	○	×
	PRC[*3]	○	○	○	×
	S [*4]	○	○	×	×
S [*4]	FC[*1]	○	○	×	×
	ARC[*2]	○	○	×	×
	PRC[*3]	○	○	×	×
	S [*4]	○	○	×	×

4.2 評価

4.2.1 両端末が 1 台の NAPT 配下に存在する場合

表 2 に NAPT を 1 台ずつ経由した場合の経路最適化処理時における検証結果を示す. 表において,「○」は経路最適化が可能な場合,「△」は経路最適化処理を 2 回行うことで経路最適化が可能な場合,「×」は経路最適化が不可能な場合を示す. 両端末が 1 台の NAPT 配下に存在する場合, Cone 型同士の通信は経路最適化が可能であることが確認できた. Responder 側の NAPT が Symmetric の場合を考える. Initiator 側の NAPT が Full Cone の場合, Responder からの Hole Punching が受信可能である. そのため, Responder 側の NAPT が Symmetric である場合においても経路最適化が可能である. Initiator が Address-Restricted Cone の場合, 短期間に再度, 経路最適化処理を実行することで経路最適化が可能である. 1 度目の経路最適化処理では, Responder からの Hole Punching が Initiator 側の NAPT の制約で受信できないが, その後の Initiator 側から Responder への Hole Punching で Initiator 側の NAPT に Responder の IP アドレスのマッピングが行われる. Initiator 側の NAPT にマッピングが行われることで, 2 度目の経路最適化処理の際に, Responder からの Hole Punching を受信可能になるため, 経路最適化が可能となる. 結果として, 一部を除く NAPT のパターンにおいて経路最適化が可能であることを確認した.

4.2.2 端末が多段 NAPT 配下に存在する場合

表 3 に送信側端末が NAPT1 台で相手先端末が NAPT2 台の配下に存在する場合, 表 4 に送信側端末が NAPT2 台で相手先端末が NAPT1 台の配下に存在する場合, 表 5, 6 に NAPT を 2 台ずつ経由した場合の経路最適化処理時における検証結果を示す. 各動作結果から, 経路最適化可能な NAPT の組み合わせには一定の規則があること

表 5 両端末：NAPT 2 台 (Responder の外側 NAPT：Full Cone, Address-Restricted Cone)

Initiator \ Responder		NAPT (Outer)	FC*1				ARC*2			
		NAPT (Inner)	FC*1	ARC*2	PRC*3	S *4	FC*1	ARC*2	PRC*3	S *4
NAPT (Outer)	NAPT (Inner)									
FC*1	FC*1		○	○	○	○	○	○	○	○
	ARC*2		○	○	○	△	○	○	○	△
	PRC*3		○	○	○	×	○	○	○	×
	S *4		○	○	×	×	○	○	×	×
ARC*2	FC*1		○	○	○	△	○	○	○	△
	ARC*2		○	○	○	△	○	○	○	△
	PRC*3		○	○	○	×	○	○	○	×
	S *4		○	○	×	×	○	○	×	×
PRC*3	FC*1		○	○	○	×	○	○	○	×
	ARC*2		○	○	○	×	○	○	○	×
	PRC*3		○	○	○	×	○	○	○	×
	S *4		○	○	×	×	○	○	×	×
S *4	FC*1		○	○	×	×	○	○	×	×
	ARC*2		○	○	×	×	○	○	×	×
	PRC*3		○	○	×	×	○	○	×	×
	S *4		○	○	×	×	○	○	×	×

表 6 両端末：NAPT 2 台 (Responder の外側 NAPT：Port-Restricted Cone, Symmetric)

Initiator \ Responder		NAPT (Outer)	PRC*3				S *4			
		NAPT (Inner)	FC*1	ARC*2	PRC*3	S *4	FC*1	ARC*2	PRC*3	S *4
NAPT (Outer)	NAPT (Inner)									
FC*1	FC*1		○	○	○	○	○	○	○	○
	ARC*2		○	○	○	△	△	△	△	△
	PRC*3		○	○	○	×	×	×	×	×
	S *4		×	×	×	×	×	×	×	×
ARC*2	FC*1		○	○	○	△	△	△	△	△
	ARC*2		○	○	○	△	△	△	△	△
	PRC*3		○	○	○	×	×	×	×	×
	S *4		×	×	×	×	×	×	×	×
PRC*3	FC*1		○	○	○	×	×	×	×	×
	ARC*2		○	○	○	×	×	×	×	×
	PRC*3		○	○	○	×	×	×	×	×
	S *4		×	×	×	×	×	×	×	×
S *4	FC*1		×	×	×	×	×	×	×	×
	ARC*2		×	×	×	×	×	×	×	×
	PRC*3		×	×	×	×	×	×	×	×
	S *4		×	×	×	×	×	×	×	×

表 7 両端末：同一 NAPT 1 台

FC*1	ARC*2	PRC*3	S *4
○	○	○	○

表 8 両端末：同一 NAPT 2 台

NAPT (Outer) \ NAPT (Inner)	FC*1	ARC*2	PRC*3	S *4
FC*1	○	○	○	○
ARC*2	○	○	○	○
PRC*3	○	○	○	○
S *4	○	○	○	○

を確認した．経路最適化が可能な組み合わせの規則として，制約が強い NAPT の種類が大きく影響を及ぼすことが確認できる．NAPT は，Full Cone, Address-Restricted Cone, Port-Restricted Cone, Symmetric の順に制約が強くなる．例として，Initiator の外側の NAPT が Full Cone で内側 NAPT が Port-Restricted Cone, Responder の外側の NAPT が Symmetric で内側 NAPT が Address-Restricted Cone の場合を考える．Initiator は，Port-Restricted Cone が，Responder は，Symmetric が制約として強く最適化の可否に影響を及ぼす．結果として，過半数を超える場合において経路最適化が可能であることを確認した．

4.2.3 両端末が同一 NAPT 配下に存在する場合

表 7, 8 に同一 NAPT 配下に存在する場合の経路最適化処理時における検証結果を示す．両端末が同一の NAPT 配下に存在する場合，全ての場合において経路最適化が可能であることを確認した．

5. まとめ

本稿では, 移動透過性と通信接続性を実現するプロトコルである CYPHONIC を拡張することにより, クラウドサービスの中継を必要としない通信環境における直接通信への経路最適化手法を提案した. 提案手法では, 特定の NAPT の種類と組み合わせにおいて, デバイス間の通信が直接可能な場合があることに着目し, 最適化可能である場合の通信経路最適化を行った. また, 各 NAPT の種類とその組み合わせにおける経路最適化の可否を検証し, 過半数を超える組み合わせにおいて直接通信への切り替えが行われることを確認した.

謝辞 本研究の一部は JSPS 科研費 (21K11877) の助成を受けたものである. 記して謝意を表する.

参考文献

[1] Rokonuzzaman, M., Akash, M. I., Khatun Mishu, M., Tan, W.-S., Hannan, M. A. and Amin, N.: IoT-based Distribution and Control System for Smart Home Applications, pp. 95–98 (2022).
[2] Okeme, P. A., Skakun, A. D. and Muzalevskii, A. R.: Transformation of Factory to Smart Factory, *2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus)*, pp. 1499–1503 (2021).
[3] Zhu, Y., Wu, W. and Li, D.: Efficient Client Assignment for Client-Server Systems, *IEEE Transactions on Network and Service Management*, pp. 835–847 (2016).
[4] Meftah, L., Rouvoy, R. and Chrisment, I.: Testing Nearby Peer-to-Peer Mobile Apps at Large, *2019 IEEE/ACM 6th International Conference on Mobile Software Engineering and Systems (MOBILESoft)*, pp. 1–11 (2019).
[5] Postel, J.: Internet Protocol, RFC 791 (1981).
[6] Beeharry, J. and Nowbutsing, B.: Forecasting IPv4 exhaustion and IPv6 migration, *2016 IEEE International Conference on Emerging Technologies and Innovative Business Practices for the Transformation of Societies (EmergiTech)*, pp. 336–340 (2016).
[7] Holdrege, M. and Srisuresh, P.: IP Network Address Translator (NAT) Terminology and Considerations, RFC 2663 (1999).
[8] Deering, D. S. E. and Hinden, B.: Internet Protocol, Version 6 (IPv6) Specification, RFC 8200 (2017).
[9] Wang, H.-C., Chen, C. and Lu, S.-H.: An SDN-based NAT Traversal Mechanism for End-to-end IoT Networking, *2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, pp. 1–4 (2019).
[10] Qin, Z.: Seamless converging system for IPv4/IPv6 transition, *2017 9th International Conference on Advanced Infocomm Technology (ICAIT)*, pp. 110–113 (2017).
[11] Yoshikawa, T., Komura, H., Nishiwaki, C., Goto, R., Matama, K. and Naito, K.: Evaluation of new CYPHONIC: Overlay network protocol based on Go language, *2022 IEEE International Conference on Consumer Electronics (ICCE)*, pp. 1–6 (2022).
[12] Horisaki, S., Matama, K., Naito, K. and Suzuki, H.: A Proposal of QUIC-based CYPHONIC for Encrypted End-to-End Communications, *2022 Tenth International Symposium on Computing and Networking (CANDAR)*, pp. 27–35 (2022).