Vol.2022-MBL-104 No.26
Vol.2022-UBI-75 No.26
Vol.2022-CDS-35 No.26
Vol.2022-ASD-24 No.26
2022/9/6

CYPHONICによるエンドツーエンド暗号化通信のためのファイアウォール越え解決手法

堀崎 翔太[1,a)]　眞玉 和茂[2]　内藤 克浩[3]　鈴木 秀和[1,b)]

概要：IPv4/IPv6 混在環境下で通信接続性と移動透過性を同時に実現する技術として，CYPHONIC（CYber PHysical Overlay Network over Internet Communication）が提案されている．CYPHONIC を導入したモバイルノードや IoT デバイス間はエンドツーエンド暗号化通信を実現することができる．しかし，CYPHONIC プロトコルは UDP4501 番ポートを利用するため，既設のファイアウォールの設定を変更しないと暗号化通信を実現することができない．本稿では，次世代 HTTP プロトコルとして標準化された QUIC を利用して CYPHONIC のプロトコル仕様を拡張し，ファイアウォールを通過することが可能なエンドツーエンド暗号化通信の実現手法を提案する．

Solving Firewall Traversal for CYPHONIC-based End-to-End Encrypted Communication

Shota Horisaki[1,a)]　Kazushige Matama[2]　Katsuhiro Naito[3]　Hidekazu Suzuki[1,b)]

1. はじめに

スマートフォンをはじめとするモバイルノード台数の増加やインターネットサービスの普及により，ネットワークトラフィック量は増加の一途を辿っている [1]．現在，インターネットで利用されている TCP/IP では，IP アドレスを識別子として通信を行っている．しかし，増え続けるモバイルノードの数に相反して，現在も最も利用されている IPv4 アドレスの数は 2^{32} 個に制限されてしまう．そのため，各ネットワークに NAT（Network Address Translation）[2] を導入することにより，1 つのグローバル IP アドレスを多数のプライベート IP アドレスを割り当てたノードで共有する構成が前提となっている．

NAT は外部のネットワークから NAT 配下に存在するノードを隠蔽する特性を持つ．これにより，外部ネットワークに存在するノードは，NAT 配下のノードに対して通信を開始することができない．これは NAT 越え問題として広く知られており，NAT をまたがってアプリケーション間の双方向通信を実現するために様々な NAT トラバーサル技術が提案されている [3], [4]．また，アドレス空間が 128 ビットに拡張された IPv6 への移行が世界的に進んでおり，すべてのノードに一意の IP アドレスを割り当てることが可能である [5]．しかし，IPv4 と IPv6 は互換性が無く，異なるバージョン間では通信を行うことができない．そのため，現在のインターネットは IPv4 と IPv6 が混在した環境になっている．

さらに，TCP/IP アーキテクチャではノード間の通信を IP アドレスやポート番号等の組で管理している．しかし，IP はノードが移動することを考慮して設計されていないため，接続先ネットワークが変化した場合，ノードの IP アドレスが変化してしまう．そのため，モバイルネットワーク環境において，ノードが通信中にネットワークを移動してしまうと，移動前のフローを識別することができず，通信

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

© 2022 Information Processing Society of Japan

が断絶してしまう．この課題に対しても，これまでに様々な解決策が提案されている [6], [7], [8].

これらの課題に加え，インターネットでは様々な脅威が存在するため，暗号化通信技術やアクセス制御技術などを導入して，安全な通信を実現することが必須となっている [9]．しかし，これらの種々の課題に対する個別の解決策は多数提案されている一方で，包括的に解決する仕組みについては十分な議論が行われていない．

このような背景から，筆者らは IPv4/IPv6 混在環境において，ネットワーク環境に影響されない通信接続性とネットワークが切り替えられた場合にも通信可能な移動透過性を同時に実現する技術として，CYPHONIC（CYber PHysical Overlay Network over Internet Communication）を提案してきた [10], [11]．CYPHONIC は，ノード間の通信に先立って最適かつ安全な通信経路を確立し，オーバーレイネットワーク上でエンドツーエンド暗号化通信をアプリケーションに提供するネットワークアーキテクチャである．CYPHONIC プロトコルは UDP4501 番ポートを用いてノード間に暗号化 UDP トンネルを構築するために，CYPHONIC シグナリングを行うために既設のファイアウォール（以下，FW：FireWall）の設定を変更しなければならない．しかし，企業や大学などのネットワークに存在する FW は，セキュリティの観点から容易にポート開放することは推奨されていない．

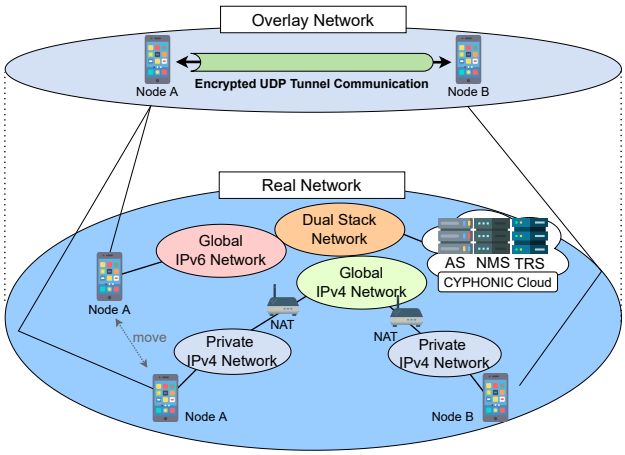
そこで，本稿は次世代 HTTP（Hyper Text Transfer Protocol）プロトコルとして標準化された QUIC [12], [13], [14], [15] を CYPHONIC に適用することにより，セキュアな FW を通過可能なエンドツーエンド暗号化通信を実現する手法を提案する．以下，2 章で既存の CYPHONIC の概要と現状の課題について整理する．3 章では課題解決に向けた手法を示し，4 章で動作検証および定性的な評価を行う．そして，最後に 5 章でまとめる．

2. CYPHONIC

2.1 概要

CYPHONIC は，ノードに固定の仮想 IP アドレスを割り当て，通信開始時に通信相手ノードとの間に暗号化 UDP（User Datagram Protocol）トンネルを構築することにより，オーバーレイネットワーク上の通信をアプリケーションに対して提供する．ノードが移動して IP アドレスが変化した場合は，暗号化 UDP トンネルを再構築するが，仮想 IP パケットに記載する送信元/宛先 IP アドレスは変化しないため，通信フローは維持され，移動透過性を実現することができる．さらに，IPv4 と IPv6 の互換性問題を解決しており，NAT の有無に関係なくエンドツーエンドの暗号化通信を確立することができる．

図 1 に CYPHONIC の概要を示す．CYPHONIC は，クラウドサービスと CYPHONIC を導入した CYPHONIC ノードにより構成される．クラウドサービスは，以下の 3 種類から構成される．



AS:Authentication Service　NMS:Node Management Service　TRS:Tunnel Relay Service

図 1　CYPHONIC の構成

- AS（Authentication Service）
  CYPHONIC ノードの認証を行う．CYPHONIC ノードの識別子である FQDN（Fully Qualified Domain Name），公開鍵証明書，NMS との暗号化通信に使用する共通鍵の生成と配布を行う．また，CYPHONIC ノードの機器情報を管理しており，中間認証局としての役割を持つ．
- NMS（Node Management Service）
  CYPHONIC ノードの実 IP アドレスや NAT の有無などのネットワーク情報を管理し，ネットワーク情報に応じて CYPHONIC ノード間のトンネル構築処理を制御する．CYPHONIC ノードに対して，仮想 IP アドレスと経路構築時に CYPHONIC ノード間での通信に使用するトンネル鍵，一時鍵の生成と配布を行う．
- TRS（Tunnel Relay Service）
  IP アドレスの違いや CYPHONIC ノードがそれぞれ異なる NAT 配下に存在する場合のように，CYPHONIC ノード同士で直接相互通信ができない場合に通信を中継する．

AS，NMS および TRS は IPv4/IPv6 どちらのネットワークからでも利用できるよう，デュアルスタックネットワークに配置する．

2.2 シグナリング

CYPHONIC のシグナリングには，主に起動時に実施する認証処理と登録処理，通信開始時に実施する経路構築処理とトンネル構築処理の 2 つに分けられる．以下に従来手法における具体的な手順について示す．

2.2.1 認証および登録処理

図 2 に公開鍵証明書を用いた認証処理および登録処理のシーケンス図を示す．認証処理（Login Request/Response）

© 2022 Information Processing Society of Japan

Vol.2022-MBL-104 No.26
Vol.2022-UBI-75 No.26
Vol.2022-CDS-35 No.26
Vol.2022-ASD-24 No.26
2022/9/6

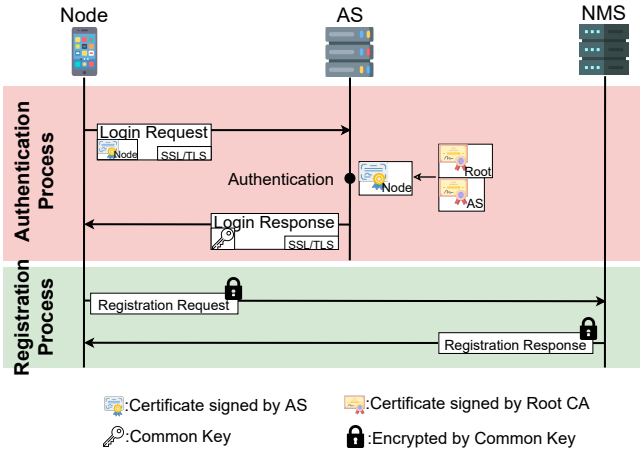


図 2 従来手法の認証および登録処理

表 1 シグナリングパターン

CYPHONIC ノードの位置		通信相手側 IPv6	通信相手側 IPv4-G	通信相手側 IPv4-P
通信開始側	IPv6	パターン 1	パターン 3	パターン 3
	IPv4-G	パターン 3	パターン 1	パターン 2
	IPv4-P	パターン 3	パターン 2	パターン 3

IPv4-G：グローバル IPv4　IPv4-P：プライベート IPv4

では，CYPHONIC ノードが起動した際に，AS に対して CYPHONIC ノードの真正性を示すため，AS から事前に配布された公開鍵証明書を用いて TLS 通信によりログイン認証を行う．認証成功後，NMS に登録処理（Registration Request/Response）を行う．

登録処理では，AS から配布された共通鍵を用いて NMS と暗号化通信を行い，CYPHONIC ノードのネットワーク情報を登録し，仮想 IP アドレスを取得する．

2.2.2 経路構築およびトンネル構築処理

CYPHONIC ノードの通信開始時に，DNS（Domain Name System）名前解決処理や CYPHONIC ノードの接続先ネットワークが変化するハンドオーバ処理をトリガとしてノード間に経路構築処理が行われる．経路構築処理は，双方の CYPHONIC ノードのネットワーク環境に応じて適切な経路が選択され，**表 1** に示す 3 パターンが存在する．

- パターン 1
  1 つ目のパターンは，**図 3** のように，通信経路上に NAT が存在せず，両ノードが互いに直接通信することができる場合である．
  経路構築処理では，NMS が登録処理で得た CYPHONIC ノードのネットワーク情報をもとに，最適な通信経路を決定し，トンネル構築時の CYPHONIC ノード間での通信に使用するトンネル鍵，一時鍵の生成を行う．その後，両 CYPHONIC ノードに仮想 IP アドレスを含む相手 CYPHONIC ノードのアドレス情報，トンネル鍵，一時鍵を通知し，トンネル構築処理に移行する．

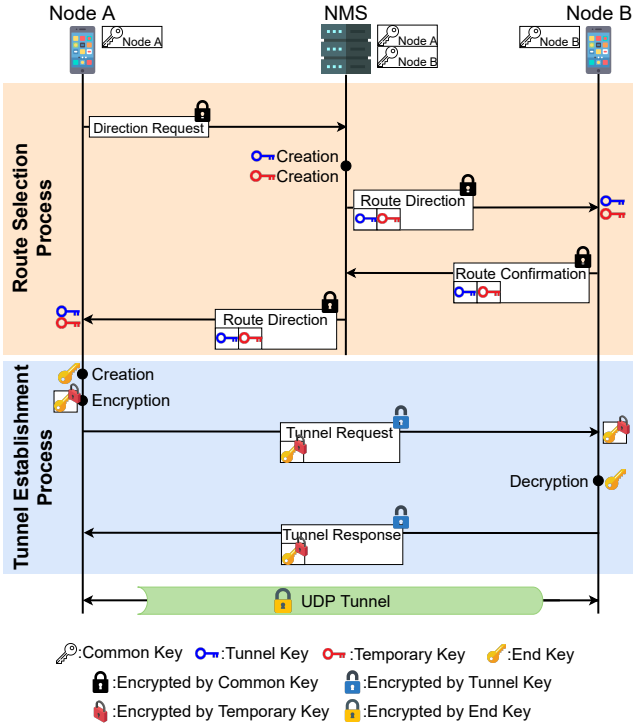


図 3 従来手法の経路構築およびトンネル構築処理（パターン 1）

トンネル構築処理では，CYPHONIC ノードがトンネル暗号化通信で用いるエンド鍵を生成し，一時鍵で暗号化したものを相手に通知し，UDP トンネルを構築する．NMS と CYPHONIC ノード間の通信は共通鍵，トンネル構築時の通信はトンネル鍵，トンネル通信はエンド鍵を用いてそれぞれ暗号化通信を実現する．

- パターン 2
  2 つ目のパターンは，**図 4** のように，通信経路上に NAT は存在するが，どちらか片一方のみ設置されている場合である．このパターンでは，ホールパンチングの役割も兼ねて，NAT 配下の CYPHONIC ノード側からからトンネル構築要求を行うことにより，CYPHONIC ノード間での暗号化エンドツーエンド通信が可能である．
- パターン 3
  3 つ目のパターンは，**図 5** のように，両 CYPHONIC ノードが異なる Symmetric NAT 配下に存在したり，IPv4 オンリーのネットワークと IPv6 オンリーのネットワークのように直接通信できない場合である．この場合は，TRS を経由した UDP トンネル構築処理を行うため，TRS のアドレス情報を両 CYPHONIC ノードに通知する．また，新たに Relay Request により NMS が TRS にトンネル鍵を通知する処理や，TRS と相互通信を行うためのホールパンチングに関する処理が加わる．CYPHONIC シグナリングおよびトンネル通信は，クラウドサービスが，CYPHONIC プロトコルに含まれる識別子を用いて各通信の管理を行う．

© 2022 Information Processing Society of Japan

Vol.2022-MBL-104 No.26
Vol.2022-UBI-75 No.26
Vol.2022-CDS-35 No.26
Vol.2022-ASD-24 No.26
2022/9/6

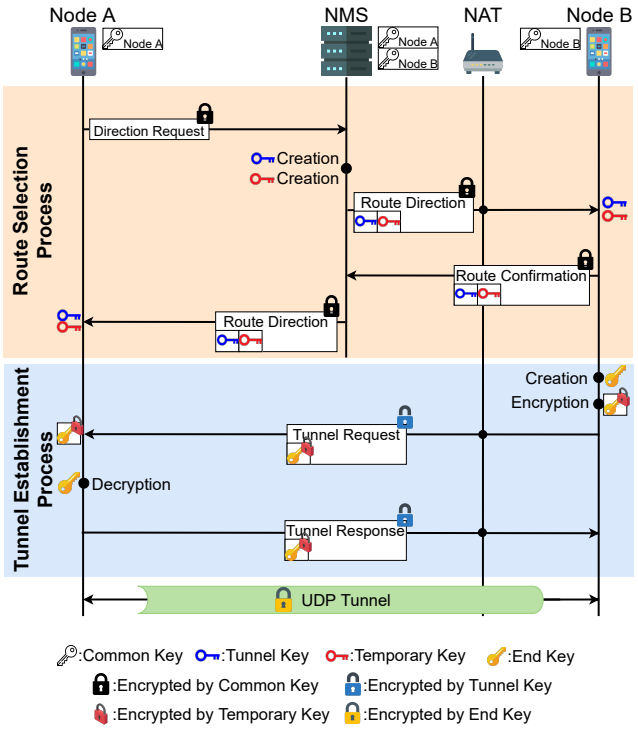


図 4 従来手法の経路構築およびトンネル構築処理（パターン 2）

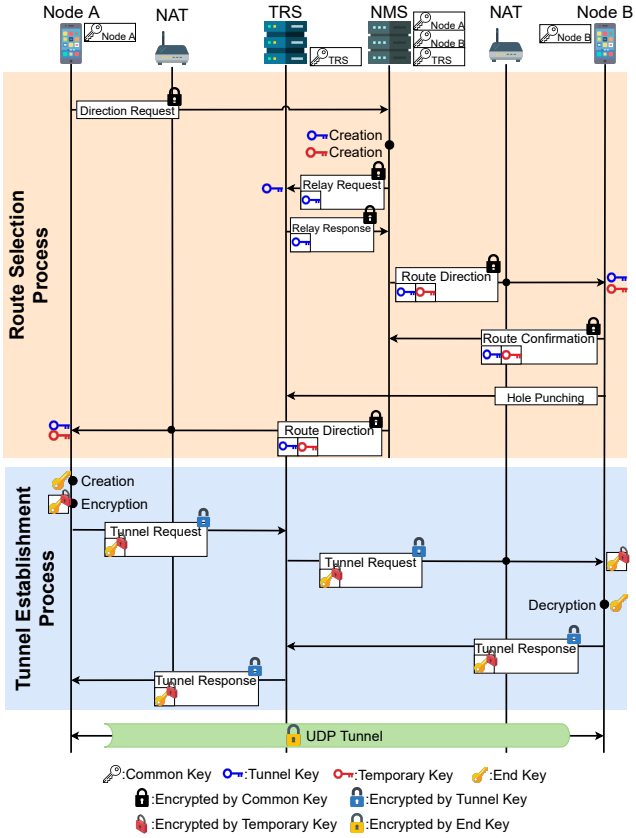


図 5 従来手法の経路構築およびトンネル構築処理（パターン 3）

2.3 現状の課題

CYPHONIC プロトコルは，UDP のポート番号 4501 番を用いてトンネル通信を行う．このポート番号は Well Known ポート番号を用いている HTTP や HTTPS とは異なり，企業などの強固なセキュリティ対策が適用されている一般的な FW では解放されていない．そのため，FW の設定を変更しなければ，CYPHONIC による暗号化通信を実現することができない．しかし，セキュリティ確保の観点から，企業では容易に FW のポートを開放することは推奨されておらず，既設の FW に対して逐一設定変更を行うことは困難である．したがって，CYPHONIC ノードが移動しても，移動先で接続したネットワークに設置されている FW の設定を変更することは不可能であるため，FW が原因で CYPHONIC によるセキュアなエンドツーエンド通信および移動透過性の実現が阻害されてしまう課題がある．

3. 提案手法

3.1 課題解決のアプローチ

前述した FW が原因で通信が阻害される事象は，CYPHONIC に限らず，多くの一般的なアプリケーションに共通する課題である．移動することが前提のスマートフォンにおけるアプリでは，HTTP/HTTPS 通信を利用することにより，この課題に対処している．これは，企業のような強固なファイアウォールであったとしても，Web の通信は通常許可しているため，接続先ネットワークに依存することなく，FW を越えてアプリとサービス間の通信を実現することができる．

そこで，CYPHONIC にもこのアプローチを採用することにより，FW の通過問題を解決することを検討した．HTTP/HTTPS 通信はトランスポート層プロトコルとして TCP を採用している．一方，CYPHONIC はできる限りノード間はエンドツーエンドで暗号化通信できること，スループット特性を維持できることなどから，多くの NAT 越え技術でも利用されている UDP を採用している．そのため，CYPHONIC に HTTP/HTTPS を適用すると，高スループットで NAT 越え可能なエンドツーエンド暗号化通信が不可能になってしまい，本来の目的を達成できなくなってしまう．

ここで，筆者らは 2021 年 5 月に標準化されたトランスポート層プロトコルである QUIC に着目した．QUIC は UDP をベースにしている一方，TCP の輻輳制御や再送制御などの信頼性を確保する機能や，TLS 相当の暗号化機能も有しており安全性も担保されている．さらに，次世代 HTTP/3 で利用されているプロトコルであることから，HTTP/HTTPS と同様に FW の影響を受けないと考えられる．

3.2 概要

そこで本稿では，CYPHONIC に QUIC を適用することにより，FW 越えが可能なエンドツーエンド暗号化通信を実現する CYPHONIC-over-QUIC を提案する．特に，

© 2022 Information Processing Society of Japan

Vol.2022-MBL-104 No.26
Vol.2022-UBI-75 No.26
Vol.2022-CDS-35 No.26
Vol.2022-ASD-24 No.26
2022/9/6

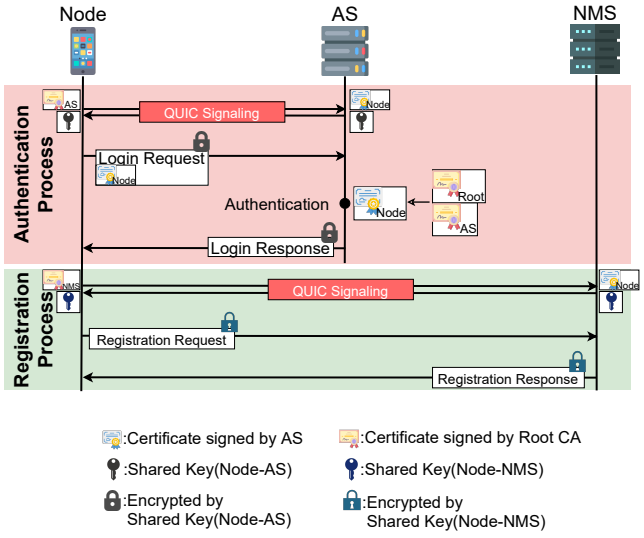


図 6 提案手法による認証および登録処理

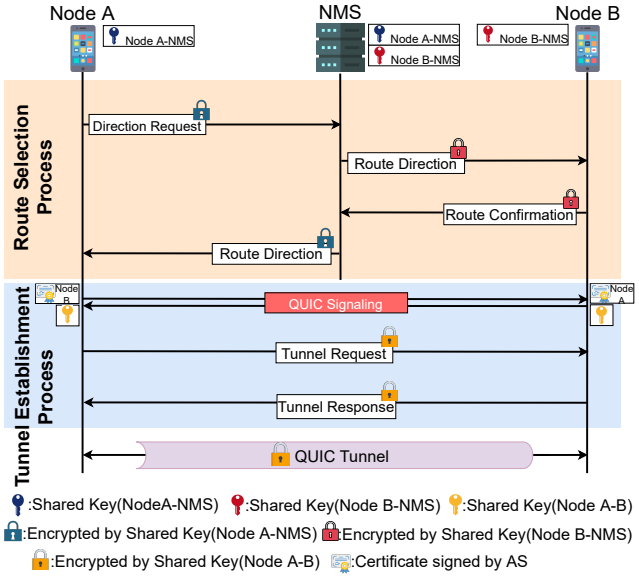


図 7 提案手法による経路構築及びトンネル構築処理（パターン 1）

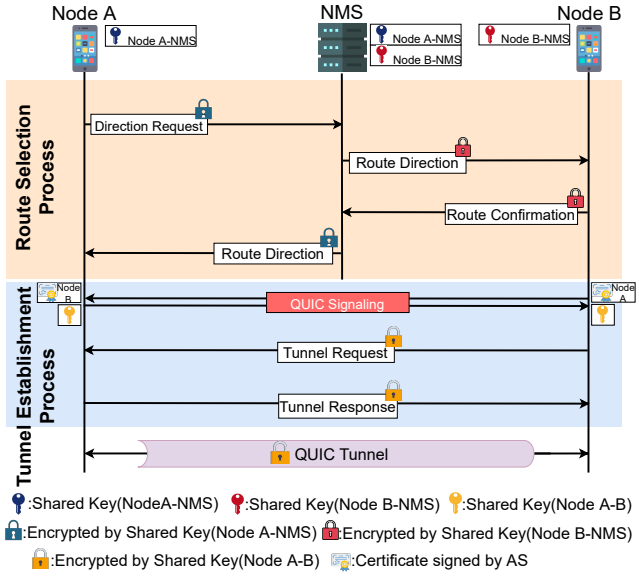


図 8 提案手法による経路構築及びトンネル構築処理（パターン 2）

CYPHONIC ノードが強固な FW があるネットワーク配下に存在する時に CYPHONIC が提供する通信接続性の確保について焦点を当てる．

従来の CYPHONIC（以下，従来手法）で UDP を使ってトンネル通信を実現していた機能を，CYPHONIC-over-QUIC（以下，提案手法）では QUIC を使ってトンネル通信を実現するように拡張する．また，従来手法では暗号鍵の交換や暗号化処理機能を CYPHONIC クラウドサービスが提供していたが，提案手法では QUIC が標準で有している TLS1.3 の一部機能に代替する．

3.3 シグナリングの変更

3.3.1 認証および登録処理

図 6 に提案手法による認証処理および登録処理のシーケンス図を示す．認証処理の前に，CYPHONIC ノードが起動したタイミングで，QUIC シグナリングを実行する．QUIC シグナリングでは，CYPHONIC ノードと AS がそれぞれ，事前に AS によって配布された公開鍵証明書，Root CA から発行された公開鍵証明書を用いて相互認証を行い，CYPHONIC ノードと AS 間の通信を暗号化する鍵を共有する．認証処理では，公開鍵証明書の情報を元にデータベースにアクセスし，登録されている情報と照合する．認証成功後，AS は CYPHONIC ノードの FQDN，NMS のアドレス情報を通知するが，CYPHONIC の枠組みで行っていた共通鍵の生成および配布を廃止する．代わりに，登録処理の前にも NMS に対して同様の QUIC シグナリングを行い，CYPHONIC ノードと NMS 間の通信を暗号化する共通鍵を QUIC における TLS の枠組みで共有する．

3.3.2 経路構築およびトンネル構築処理

- パターン 1・パターン 2

  図 7，**図 8** に提案手法によるパターン 1，パターン 2 のシーケンス図を示す．NMS は経路決定後，両 CYPHONIC ノードに，仮想 IP アドレスを含めた相手 CYPHONIC ノードのアドレス情報，FQDN を通知するが，従来手法で行っていたトンネル鍵と一時鍵の生成と配布を廃止する．トンネル構築処理およびトンネル通信で用いるエンド鍵は，CYPHONIC ノード間での QUIC シグナリングで共有する．パターン 1 では，通信開始側 CYPHONIC ノード A から通信相手側 CYPHONIC ノード B に対して，QUIC シグナリングによる接続要求とトンネル構築要求を行う．パターン 2 では，NAT 配下に存在する CYPHONIC ノード B から QUIC シグナリングとトンネル構築要求を行う．これにより，経路構築処理，トンネル構築処理およびトンネル通信も UDP から QUIC による通信に置き換えることが可能となる．

© 2022 Information Processing Society of Japan

Vol.2022-MBL-104 No.26
Vol.2022-UBI-75 No.26
Vol.2022-CDS-35 No.26
Vol.2022-ASD-24 No.26
2022/9/6

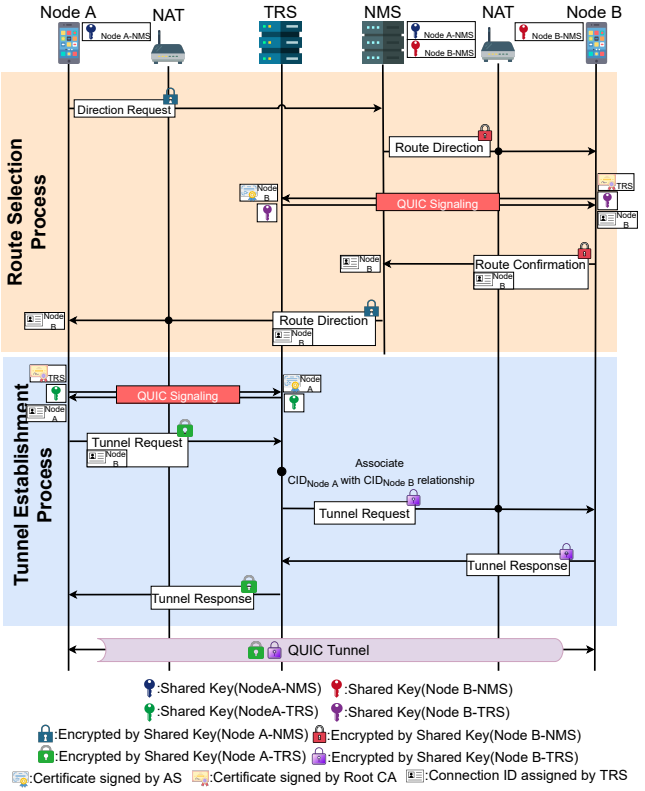


図 9 提案手法による経路構築及びトンネル構築処理（パターン 3）

- パターン 3

  図 9 に提案手法によるパターン 3 のシーケンス図を示す．提案手法では，TRS に対するトンネル鍵と一時鍵の通知やホールパンチングに関する処理を廃止する．代わりに，通信相手側 CYPHONIC ノード B は NMS から送られてきたアドレス情報をもとに，TRS に対して QUIC シグナリングによる接続要求と鍵の共有を行う．この際，TRS は CYPHONIC ノード B に対して，一意な Connection ID“CID_B” を割り当てる．CID とは，QUIC に定義されている通信識別子であり，IP アドレスやポート番号が変化しても通信フローを同定するために導入されている．CYPHONIC ノード B は，取得した CID_B を NMS 経由で CYPHONIC ノード A に通知する．その後，CYPHONIC ノード A も TRS との QUIC シグナリングによって CID_A を取得し，トンネル構築要求に CID_B を含めて TRS に通知する．TRS は，2 つ CID と CYPHONIC ノードのアドレス情報を関連付ける．これにより，以降の通信は QUIC ヘッダ内に含まれる CID によって宛先 CYPHONIC ノードを判断することができる．

4. 実装・評価

4.1 実装

本稿では，従来の CYPHONIC ノード，AS および NMS に対して，提案手法の実装を行った．CYPHONIC は，クラウド環境との親和性が高く，安定した並列処理が可能な Go 言語と関連する Go ライブラリを利用している．そのため，本実装は Go 言語で実装された QUIC オープンソースライブラリである `quic-go` [16] を用いた．

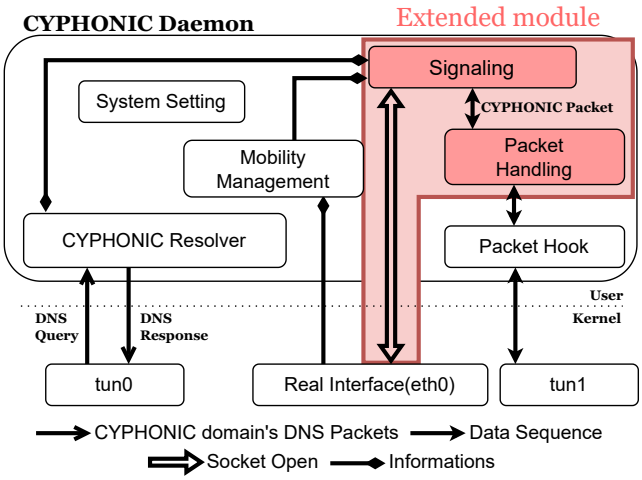


図 10 CYPHONIC のモジュール構成

図 10，**図 11** に本実装におけるモジュール構成図とパケット処理フローを示す．赤枠で囲まれたシグナリング，パケットハンドリングに関するモジュールに対して変更を加えた．QUIC は，UDP ベースで実装されたプロトコルであるため，トランスポート層の中で UDP と QUIC の 2 種類のプロトコルが存在する．従来手法では，パケットの完全性を保証するため，CYPHONIC 独自の HMAC（Hash Based Message Authentication Code）を付与していた．一方，提案手法では QUIC がパケットの完全性を保証するため，従来手法で必要だった CYPHONIC 独自の HMAC を廃止することができる．

4.2 動作検証

提案手法による動作検証を行うにあたり，コンテナ型の仮想環境においてアプリケーションを実行するためのプラットフォームである Docker を用いた．Docker 上にパターン 1 及びパターン 2 を模した仮想環境を構築し（NMS の IPv6 アドレスは `2001:db8::3`），2 台の CYPHONIC ノード間（IPv6 アドレスはそれぞれ `2001:db8::5`，`2001:db8::6`）でコマンド `ping` を用いた疎通確認を実施した．

図 12 に CYPHONIC ノードによる Wireshark のパケットキャプチャ結果の一部を示す．図 12 上部のパケットの一覧のうち，赤枠で示した「`STREAM(0)`」と示されているパケットが ICMP パケットを QUIC でカプセル化したものであり，2 往復分に該当する．そのうち最初の ICMP Echo Request を含むパケット（図 12 の (1)-1）のパケットをダンプした内容を図 12 下部に示す．実 IPv6 ヘッダ，UDP ヘッダおよび QUIC から構成されており，青枠で囲まれた部分（`Protected Payload`）が QUIC で暗号化されたスト

© 2022 Information Processing Society of Japan

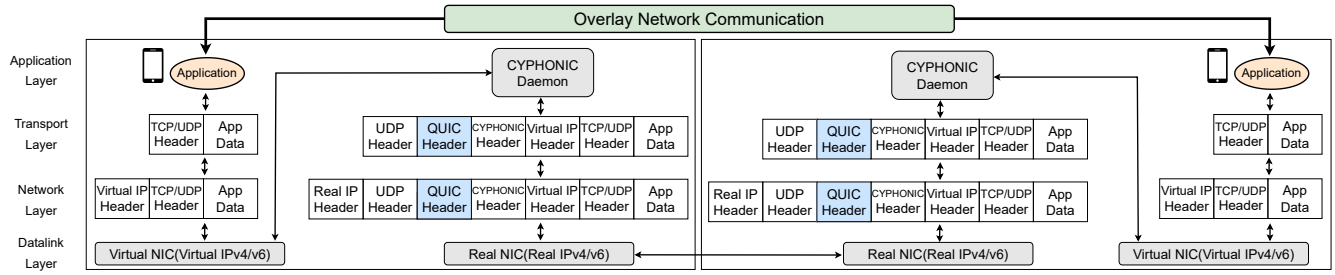


図 11 提案手法によるパケット処理フロー

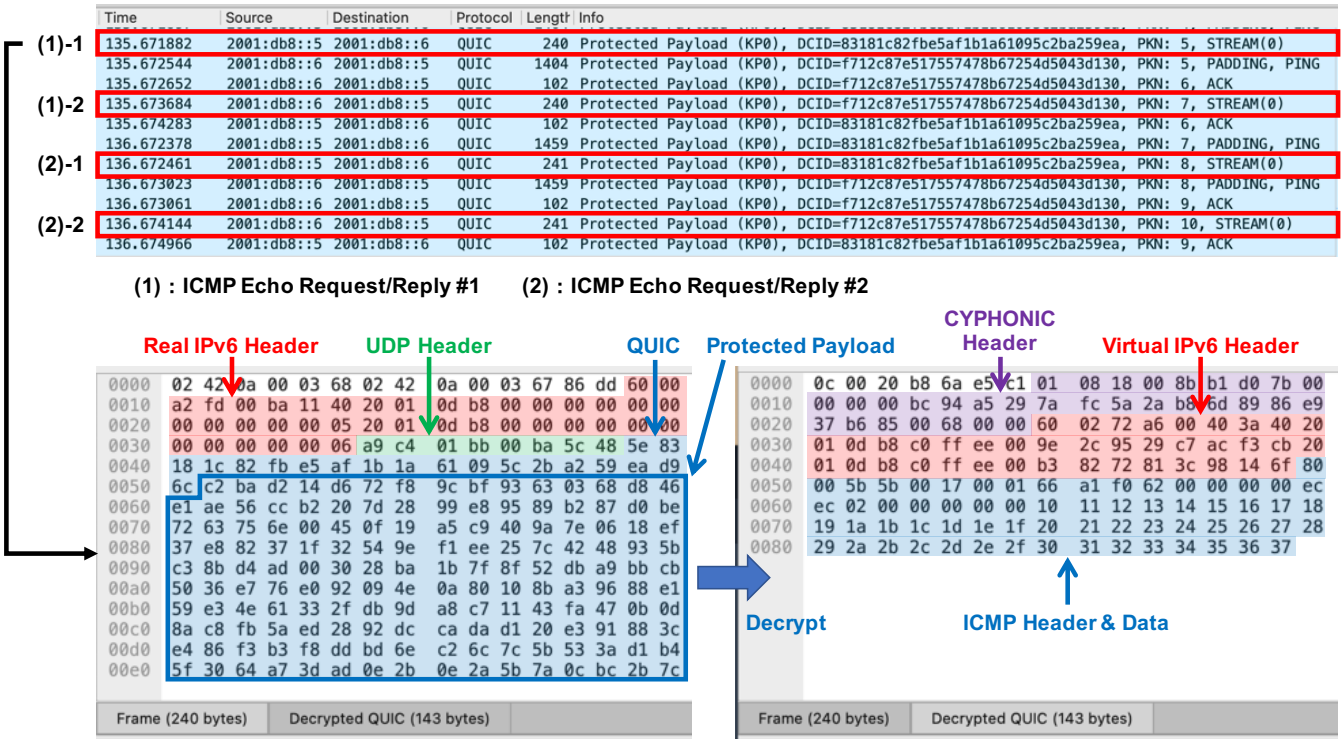


図 12 Wireshark によるパケットキャプチャ結果

リームデータである．このデータを復号して解析したところ，CYPHONIC ヘッダ，仮想 IPv6 ヘッダおよび ICMP Echo Request メッセージが含まれていることを確認できた．以上より，提案手法により CYPHONIC ノード上のアプリケーションが送受信するデータを QUIC トンネルを通じて暗号化してやり取りできることを確認した．

4.3 評価

本稿で提案した手法について，従来手法と定性的に比較評価を行った．**表 2** に提案手法と従来手法の比較表を示す．適用範囲について，従来では CYPHONIC によるシグナリングおよびトンネル通信は UDP4501 番ポートを用いているため，FW によって CYPHONIC のパケットが遮断される問題があった．提案手法では，CYPHONIC に QUIC を用いることによって，移動先ネットワークに強固な FW が設置されていても，その影響を受けない．これにより，CYPHONIC に基づくエンドツーエンド暗号化通信を適用できる範囲が大幅に広がる．

表 2 従来手法と提案手法の比較

	従来手法	提案手法
移動先ネットワークでの通信成否	△	○
ノードへの公開鍵証明書の配付	オプション	必須
ノード間の認証	×	○
クラウドサービスの安全性	△	○
スループット性能	○	○

従来手法では，CYPHONIC ノードと NMS 間の認証方法としてパスワード認証と公開鍵証明書認証の 2 種類があった．そのため，パスワード認証モードの場合，CYPHONIC ノードはパスワードを提示すればよく，公開鍵証明書を保有する必要は無かった．一方，提案手法は CYPHONIC ノード間で QUIC による暗号化通信を行うため，公開鍵証明書を必ず保有する必要がある．そのため，CYPHONIC 導入時における初回のサインアップにおいて公開鍵証明書の発行処理を導入する必要はあるが，公開鍵証明書を用いた CYPHONIC ノード間の相互認証が可能になり，通信

© 2022 Information Processing Society of Japan

表 3 パケットサイズの比較

	従来手法 [byte]	提案手法 [byte]
Real IPv6 Header	40	40
UDP Header	8	8
QUIC Header	—	20
CYPHONIC Header	32	32
Virtual IPv6 Header	40	40
IP Payload	1,364	1,360
HMAC	16	—

表 4 CYPHONIC ノード間における ping による RTT 計測結果

	従来手法 [ms]	提案手法 [ms]
最小	1.918	2.365
平均	4.018	4.034
最大	8.148	6.741
標準偏差	1.323	0.911

の安全性を向上させることができる．また，クラウドサービスが暗号鍵の管理をする必要がなくなるため，クラウドサービスに対する攻撃への対策にもなり得る．

表 3 に 1 パケットで送れるデータサイズに関して従来手法と提案手法で比較したものを示す．提案手法は新たに QUIC ヘッダを追加するが，HMAC の削除によって，一度に送ることができる IP ペイロードサイズは 4 バイトの減少に抑えられている．また，**表 4** に 4.2 節の Docker 環境において CYPHONIC ノード間で `ping` を 100 回実行した際の RTT を計測した結果を示す．アプリケーションデータを伝送するパケットの暗号化には AES（Advanced Encryption Standard），128 ビットの暗号鍵が用いられる点も提案手法は従来方式と同様である．以上から，提案手法は従来手法と同等のスループット特性が得られるものと考えられる．なお，文献 [10] によると，iperf3 を用いて 15Mbps の帯域幅で CYPHONIC ノード間でメッセージ交換したときの暗号化通信スループットは 11.16 Mbps を維持していることが示されており，提案手法においてもアプケーション性能に影響を及ぼさないことが期待できる．

5. まとめ

本稿では，CYPHONIC に QUIC を適用することにより，従来手法で課題だった FW を越えたエンドツーエンド暗号化通信を実現する手法を提案した．CYPHONIC シグナリングおよびトンネル通信を QUIC で暗号化通信を行うよう仕様を変更した．プロトタイプ実装して動作検証した結果，正常に動作していることを確認した．

今後は実環境で FW を越えて動作するか検証し，提案手法のシグナリングオーバヘッドおよびスループット特性の評価を行う．また，CYPHONIC が提供する移動透過性を QUIC の Connection Migration 機能に代替するための仕様を検討する．

参考文献

[1] Cisco: Cisco Annual Internet Report (2018–2023) White Paper, https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html (2020).
[2] Egevang, K. B. and Srisuresh, P.: Traditional IP Network Address Translator (Traditional NAT), RFC 3022 (2001).
[3] Yang, L. and Lei, K.: Combining ICE and SIP protocol for NAT traversal in new classification standard, *2016 5th International Conference on Computer Science and Network Technology (ICCSNT)*, pp. 576–580 (2016).
[4] Huang, F., Yu, L., Shen, T. and Hu, S.: The P2P Solution Research and Design Based on NAT Traversing Technology, *2019 IEEE 3rd Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, pp. 1347–1351 (2019).
[5] Hinden, B. and Deering, D. S. E.: Internet Protocol, Version 6 (IPv6) Specification, RFC 2460 (1998).
[6] Le, D., Fu, X. and Hogrefe, D.: A review of mobility support paradigms for the internet, *IEEE Communications Surveys & Tutorials*, Vol. 8, No. 1, pp. 38–51 (2006).
[7] Perkins, C. E.: IP Mobility Support for IPv4, Revised, RFC 5944 (2010).
[8] Tong, H., Wang, T., Zhu, Y., Liu, X., Wang, S. and Yin, C.: Mobility-Aware Seamless Handover With MPTCP in Software-Defined HetNets, *IEEE Transactions on Network and Service Management*, Vol. 18, pp. 498–510 (2021).
[9] Yao, Q., Wang, Q., Zhang, X. and Fei, J.: Dynamic Access Control and Authorization System Based on Zero-Trust Architecture, Association for Computing Machinery, (online), available from ⟨https://doi.org/10.1145/3437802.3437824⟩ (2020).
[10] 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).
[11] Matama, K., Goto, R., Nishiwaki, C. and Naito, K.: Extension mechanism of overlay network protocol to support digital authenticates Protocol, *Proceedings of the 26th World Multi-Conference on Systemics, Cybernetics and Informatics* (2022).
[12] Thomson, M.: Version-Independent Properties of QUIC, RFC 8999 (2021).
[13] Iyengar, J. and Thomson, M.: QUIC: A UDP-Based Multiplexed and Secure Transport, RFC 9000 (2021).
[14] Thomson, M. and Turner, S.: Using TLS to Secure QUIC, RFC 9001 (2021).
[15] Iyengar, J. and Swett, I.: QUIC Loss Detection and Congestion Control, RFC 9002 (2021).
[16] Clemente, L.: lucas-clemente/quic-go: A QUIC implementation in pure go, https://github.com/lucas-clemente/quic-go.

© 2022 Information Processing Society of Japan