オーバーレイネットワークの
マルチ OS 対応を想定した端末機能の研究

B23705 氏家せいじ 指導教員 内藤克浩

キーワード: Overlay Network Protocol, Zero-Trust, Operating Systems

1 はじめに

近年のセキュリティアプローチは, ネットワークの安全性を保証するアプローチから通信端末自身を安全に接続するアプローチへと移行している [1]. 本研究室では, ゼロトラストモデルを採用した通信技術として, CYber PHysical Overlay Network over Internet Communication (CYPHONIC) を提案している [2]. CYPHONIC は, 仮想 IP アドレスを用いてオーバーレイネットワークを構築する. これにより, 物理ネットワークの影響を受けない継続的な通信を提供し, 安全性, 通信接続性, 移動透過性を実現する. また, CYPHONIC を用いた通信を行う端末には専用の端末プログラムが導入されている. 従来の CYPHONIC は Linux をベースに実装および基礎評価が行われてきた. 一方で, 現代のインターネットに接続する端末は、Linux, Windows, macOS など, 多種多様な Operating System (OS) 上で動作しており, CYPHONIC が広範な利用を目指すには, これら異なる環境での互換性を確保する必要がある. CYPHONIC は特定の OS に依存することなく, あらゆるプラットフォームで動作可能であるべきであり, OS の制約を享受してはならない.

本研究では, CYPHONIC のエンドノードに存在する端末プログラムに着目し, マルチ OS への対応を想定した設計および実装を行う. 端末プログラムには, オーバーレイネットワーク上を通信を実現するための諸機能が備わっている. これらの諸機能を OS に依存する機能と依存せず動作する機能で分離する. そして, 依存する機能を一般的に使用されるデスクトップ OS である Linux, Windows, macOS 上で動作可能となるよう実装を施すことでマルチ OS への対応を図る. 加えて, OS に起因して分岐する機能を実行前に決定することで分岐による性能劣化が発生しない枠組みを導入する. また, 端末機能をユーザが管理可能なシステムを作成する. 管理システムから端末機能の起動や停止, 通信管理を行うことで CLI による煩雑なプロセスを簡易化する.

2 CYPHONIC

図 1 に CYPHONIC の概要を示す. CYPHONIC は, CYPHONIC クラウドと CYPHONIC ノードから構成される. CYPHONIC クラウドは, CYPHONIC を用いた通信をサポートするクラウドサービスである. CYPHONIC ノードは, CYPHONIC クラウドと連携しオーバーレイネットワーク上の通信を実現する端末である. CYPHONIC ノードはログイン後に, CYPHONIC クラウドから FQDN と仮想 IP アドレスが提供される. その後, 提供された仮想

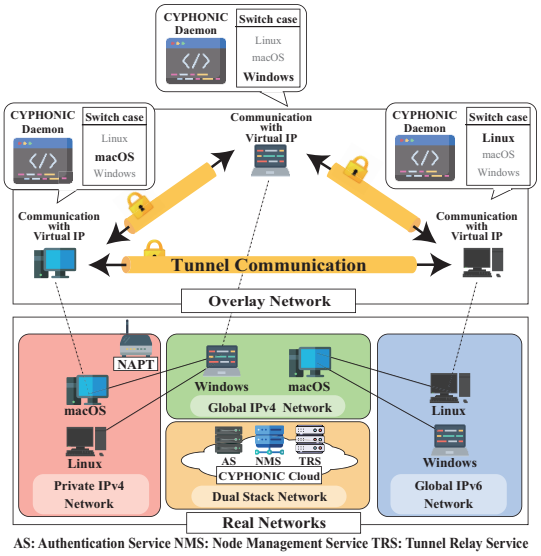

図 1: Overview of CYPHONIC

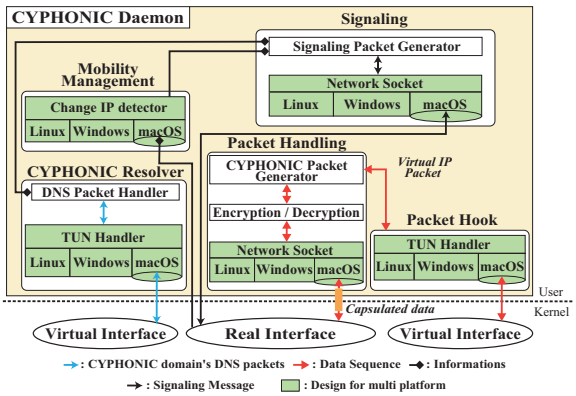

図 2: Sytem model of CYPHONIC Node

IP アドレスを利用し, 通信相手と共通鍵を用いて暗号化された通信を行う.

CYPHONIC ノードには, 端末機能として CYPHONIC Daemon が搭載されており, オーバーレイネットワーク上の通信を実現するための諸機能を備えている. CYPHONIC Daemon には, アプリケーションの仮想 IP アドレスを用いた通信を実現させるために, ソケット通信を実現する機能や仮想インターフェイスを操作する機能, 実 IP アドレスが変更検知を行う機能のような OS に依存する機能と, 通信データの暗号化やクラウドとのシグナリングを行う機能のような OS に依存しない機能が混在している.

3 マルチ OS を想定した CYPHONIC ノード

マルチ OS を想定した CYPHONIC 端末機能の設計および管理システムの設計を行う. CYPHONIC Daemon が提供する機能の中で仮想インターフェイスやネットワークソケットの作成は, OS 固有の Appli-

cation Programming Interface (API) を利用することで実現されることから, OS 毎に実装が必要となる. 図 2 にマルチ OS を想定した CYPHONIC Daemon のシステムモデル図を示す. 各モジュールを処理目的ごとに機能を組み合わせて構成し, OS に依存する機能としない機能を分離し実装を行う. これにより, 共通の OS に依存しない機能が異なる OS 間で再利用され, OS に依存する機能はそれぞれのモジュール内で OS に対応する機能が処理される. OS に依存する機能として, Network Socket と TUN Handler および Change IP detector が存在する.

Network Socket は CYPHONIC Daemon はアプリケーションが作成した仮想 IP パケットを傍受し, クラウドサービスおよび通信相手に対して, UDP 通信を行う. その際に, 相手とのネットワーク通信を実現する Network Socket を利用する. TUN Handler は, CYPHONIC を用いた通信を行う際の仮想 IP アドレスを利用した通信を実現する. TUN Handler では, OS に対応する仮想インターフェイスを作成し, パケットの読込および書込を行う. Change IP detector は物理インターフェイスの変更を検知し, クラウドサービスへの再登録および, 通信相手とのトンネル再構築を行うことで, 移動透過性を実現する. Change IP detector では, ネットリンクソケットを作成することで物理インターフェイスとの Internal Process Communication (IPC) を実現し, 物理インターフェイスの変更を検知する.

また, 従来の CYPHONIC は CLI を用いて起動や終了処理が行われており, ユーザに対して煩雑となっていた. そのため本研究では, CYPHONIC Daemon を UI 上から操作可能な管理システムを導入し, CYPHONIC Daemon の管理を容易化する. 管理システムは, CYPHONIC Daemon が提供する API を用いて状態管理が可能となる. 主に, ログイン処理の開始命令や, CYPHONIC Daemon の状態取得, 終了命令を行う.

提案するシステムを用いて, CYPHONIC ノードは起動処理, 経路構築処理およびデータ通信処理を以てオーバーレイネットワーク上の通信を実現する. 起動処理では, UI を介してログインを行い CYPHONIC クラウドによる認証を行う. 経路構築処理では, 相手の FQDN 宛のデータ送信を契機に CYPHONIC クラウドから相手の端末情報を取得し, UI 上で通信中の相手情報を管理する. データ通信処理処理では, CYPHONIC クラウドから提供された仮想 IP アドレスを用いてオーバーレイネットワーク上で通信が行われる. 尚 CYPHONIC Daemon は, 関数実行時に分岐処理が発生せず, Daemon の起動段階で OS に対応する機能のみが選定されて動作するため, 分岐処理による性能劣化が発生することなく通信が可能となる.

4 検証

動作検証と性能評価において，Linux，Windows および macOS 上で提案する CYPHONIC Daemon を用いて性能の比較を行った. 本検証の調査対象は，各 OS に対応した CYPHONIC Daemon である. 故に，使用するコンピュータの性能に起因して通信品質に差が生じることは望ましくない. そこで本検証環境では，可能な限り通信端末間の諸元の違いを排除した. 検証結果より，すべての OS において，宛先の FQDN を元に仮想 IP アドレスを取得し，仮想 IP アドレスを用いた疎通を観測したことからオーバーレイネットワーク上の通信が可能であることを確認した. また, 性能評価より, SNS や動画視聴に必要とされる指標以上の性能を確認したことから実運用可能であると結論づけた.

表 1: Result of ICMP evaluation

		Linux	Windows	macOS
RTT	min	1.54 ms	1.00 ms	1.00 ms
	max	5.47 ms	6.70 ms	6.41 ms
	avg	2.99 ms	1.30 ms	1.72 ms

表 2: Result of UDP evaluation

Linux (VM)		macOS (VM)	
Throughput	Jitter	Throughput	Jitter
350 Mbps	0.03 ms	450 Mbps	0.04 ms
Windows		**macOS**	
Throughput	Jitter	Throughput	Jitter
228.00 Mbps	0.16 ms	230 Mbps	0.18 ms

5 まとめ

本研究では, CYPHONIC の端末機能を様々な OS を想定して実装することで, より汎用的な利用を可能にした. 実装では, OS に依存する機能と OS に依存しない機能を分離し, OS に依存する機能にそれぞれ対応する実装を施した. また，検証評価より Linux, Windows，macOS において CYPHONIC が利用可能であることを確認した.

研究業績

- S. G. Ujiie, et al.: “Proposal of CYPHONIC end-device functions on Windows OS,” *GCCE 2023*, October 2023.
- S. G. Ujiie, et al.: “New design for CYPHONIC client program supporting multiple operating systems,” *GCCE 2024*, October 2024.

参考文献

[1] M. Alawneh and I. M. Abbadi, “Integrating trusted computing mechanisms with trust models to achieve zero trust principles,” in *IOTSMS 2022*, pp. 1–6, 2022.

[2] T. Yoshikawa, H. Komura, C. Nishiwaki, R. Goto, K. Matama, and K. Naito, “Evaluation of new CYPHONIC: overlay network protocol based on go language,” in *ICCE 2022*, pp. 1–6, January 2022.