一般社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report
SeMI2023-39(2023-07)

CYPHONICクラウドにおける監視システムの設計と基礎評価

鈴木　誉写[†]　眞玉　和茂[†]　後藤　　廉[†]　ウジイエ ギレルメ セイジ[†]

内藤　克浩[††]　鈴木　秀和[†††]

† 愛知工業大学大学院経営情報科学研究科　〒470–0356 愛知県豊田市八草町八千草 1247
†† 愛知工業大学情報科学部　〒470–0356 愛知県豊田市八草町八千草 1247
††† 名城大学情報工学部　〒468–8502 名古屋市天白区塩釜口一丁目 501 番地
E-mail: †{homare448,matama,r0719en,useidy}@pluslab.org, ††naito@pluslab.org, †††hsuzuki@meijo-u.ac.jp

あらまし　著者らは，インターネットにおける安全な端末間通信，通信接続性および移動透過性を実現可能な技術として，CYber PHysical Overlay Network over Internet Communication（CYPHONIC）を提案している．CYPHONIC の管理者は，CYPHONIC のクラウドシステム（以下 CYPHONIC クラウド）の状態を把握する際に多大な労力を要していた．そこで本研究では，CYPHONIC クラウドの状態を可視化する監視システムを実現した．検証から，提案システムが CYPHONIC クラウドの状態を容易に観測可能とすること，また監視システムとして妥当な性能を発揮することを示した．提案システムは，CYPHONIC における今後の効率的な開発と運用に対して有用である．
キーワード　オーバーレイネットワーク，CYPHONIC，マイクロサービスアーキテクチャ，可観測性，監視システム

Initial Design and Evaluation of the Monitoring System for the CYPHONIC Cloud

Yoshiya SUZUKI[†], Kazushige MATAMA[†], Ren GOTO[†], Guilherme Seidy UJIIE[†],

Katsuhiro NAITO[††], and Hidekazu SUZUKI[†††]

† Graduate School of Business Administration and Computer Science, Aichi Institute of Technology
1247 Yachigusa, Yakusa Cho, Toyota City, Aichi Prefecture 470–0392 Japan
†† Faculty of Information Science, Aichi Institute of Technology
1247 Yachigusa, Yakusa Cho, Toyota City, Aichi Prefecture 470–0392 Japan
††† Faculty of Information Engineering, Meijo University
1–501 Shiogamaguchi, Tempaku-ku, Nagoya 468–8502, Japan
E-mail: †{homare448,matama,r0719en,useidy}@pluslab.org, ††naito@pluslab.org, †††hsuzuki@meijo-u.ac.jp

Abstract　The authors have proposed CYber PHysical Overlay Network over Internet Communication (CYPHONIC) as a technology to achieve secure inter-terminal communication, communication connectivity, and transparent mobility on the Internet. CYPHONIC developers have faced significant challenges in obtaining an accurate understanding of the state of the CYPHONIC Cloud, necessitating substantial effort. To address this issue, we develop a monitoring system that visualizes the status of the CYPHONIC Cloud. Through the verification, we demonstrate that the proposed system facilitates easy observation of the CYPHONIC Cloud's condition and performs effectively as a monitoring system. This system proves valuable for the future efficient development and operation of CYPHONIC.
Key words　Overlay network, CYPHONIC, Microservice architecture, Observability, Monitoring system

1. はじめに

従来の中央集権型システムは，単一障害点やサーバのボトルネックといった問題が懸念されている [1]．そこで，サーバを介さず，ネットワーク上に分散配置された端末が直接データを共有する方式が注目されている [2]．この方式は，耐障害性や負荷分散といった観点から優れており，ビデオ会議システムやファイル共有システム，仮想通貨等のサービスで利用されてい

る [3], [4]．しかし，端末間でデータを共有する際には，セキュリティとネットワークの観点から課題が残されている．

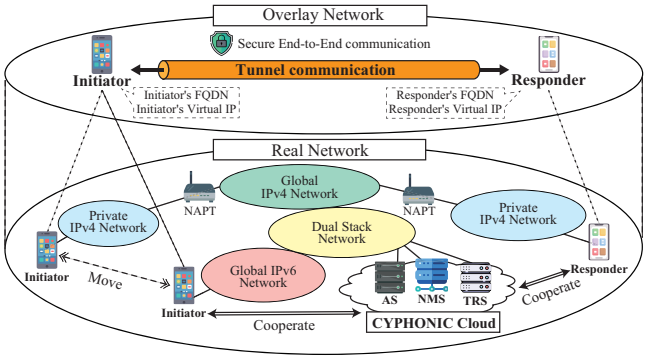
セキュリティの課題とは，安全な端末間通信の実現である．様々なネットワークに分散した端末が通信する場合，セキュアゾーンの境界を明確化することが困難である．故に，境界型セキュリティに限定しない，新たなセキュア通信機構が必要である [5]．そのため，認証済み端末間による，End-to-End（E2E）で暗号化された相互通信を実現する技術が必要である [6]．

ネットワークの課題とは，通信阻害および端末移動への対処である．端末は一般に Internet Protocol（IP）を使用して通信する [7]．IP のバージョンには IPv4 と IPv6 が存在するものの，双方は互換性を持たないため相互通信が困難である [8]．また，異なる Network Address Port Translation（NAPT）の配下に存在する端末同士が通信する際には，NAPT 越え問題への対処が必要である [9]．そのため，IP バージョンの差異や NAPT 越え問題が存在する環境においても通信を可能にする，通信接続性が求められる [10], [11]．加えて現代のネットワーク環境では，スマートフォンの利用をはじめとする移動体通信の需要が増加している [12]．故に，端末移動時においてシームレスな通信継続を実現する，移動透過性も求められる [13]．

そこで著者らは，通信接続性や移動透過性，安全な端末間通信を包括的に実現する技術として CYber PHysical Overlay Network over Internet Communication（CYPHONIC）を提案および開発している [14], [15]．CYPHONIC は，IPv4/IPv6 間の通信や NAPT 越え，端末移動に伴うネットワーク切り替え時の通信継続を実現可能である．さらに CYPHONIC は，端末認証と暗号化された E2E 通信を提供する．以上より CYPHONIC は，セキュリティの課題とネットワークの課題を解決可能である．

CYPHONIC では，CYPHONIC クラウドと呼ばれるクラウドシステムが，端末認証や端末間通信のサポートを行う．著者らは，CYPHONIC クラウドの開発において，マイクロサービスアーキテクチャを採用した．マイクロサービスアーキテクチャを採用したシステムは，水平スケーリングを伴う柔軟な規模拡張が可能となる [16]．一方，水平スケーリングによってインスタンス数が動的に増減する環境は，多大な監視負担が懸念される．故に，マイクロサービスアーキテクチャを採用したシステムでは，可観測性の確保が求められる [17]．本研究における可観測性とは，観測者に対してシステムが自身の内部状態に関する洞察を与える性質および機能を指す．以上のことから，CYPHONIC クラウドには可観測性を確保する機構が求められている．一方，従来の CYPHONIC クラウドが有しているシステムの状態を示す機能は，簡素なロギングのみである．そのため，運用や開発に伴う状態観測に際して，ログ情報を突き合わせる作業に多大な労力を要していた．このような背景から，CYPHONIC クラウドには，内部状態を観測可能とするための監視システムが必要である．

そこで本稿では，CYPHONIC クラウドにおいて，可観測性を確保する監視システムの設計および実装を示す．まず，CYPHONIC の全容を詳説し，監視対象である CYPHONIC クラウドの課題を明らかにする．続いて，提案する監視システムの設計を述べ，その概念実装を行う．その後，提案システムの有効性と性能に関する検証の結果を述べる．有効性に関する検証の結果から，提案システムが CYPHONIC クラウドの内部状態を観測可能にすることを示す．また，性能に関する検証の結果から，監視エージェントおよび拡張を施した CYPHONIC クラウドが，それぞれの役割に対して妥当な性能を有することを示す．これにより，提案システムが実用的な監視の水準と通信の品質を両立して提供し，CYPHONIC のさらなる開発と運用に有用であることを明らかにする．



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

図 1　CYPHONIC の概要

2. CYPHONIC

2.1 概　　要

図 1 に CYPHONIC の概要図を示す．CYPHONIC は，通信に仮想 IP アドレスを使用することにより，オーバーレイネットワークを構築する．オーバーレイネットワーク上の通信は，NAPT や IP バージョンの差異といった実 IP アドレスに起因する問題の影響を受けない．さらに，仮想 IP アドレスは，端末移動に伴いネットワークが切り替った場合でも不変である．これにより CYPHONIC は，アプリケーション開発者が通信接続性および移動透過性の課題を考慮する必要を排除する．加えて，CYPHONIC を利用した場合，認証済み端末が暗号化されたデータを直接通信するため，安全な E2E 通信が実現される．

CYPHONIC の実現にあたり，端末は仮想 IP パケットを実ネットワークを通して送受信する必要がある．そのため，実際には，仮想 IP パケットを端末上のデーモンプロセスが実 IP パケットにカプセル化し，User Datagram Protocol（UDP）トンネルを構築して通信する．UDP 通信における安全性や通信接続性は，端末と CYPHONIC クラウドの連携によって確保される．

2.2 構 成 要 素

CYPHONIC を利用して通信する端末を，CYPHONIC ノードと呼ぶ．CYPHONIC ノードは，一連の通信プロセスを開始する Initiator と，コネクション確立に応答する Responder に類別される．また，CYPHONIC クラウドは，端末を認証する Authentication Service（AS），端末のネットワーク情報を管理する Node Management Service（NMS），通信を中継する Tunnel Relay Service（TRS）から構成される．以下に，CYPHONIC の構成要素の詳細を示す．

- CYPHONIC ノード

CYPHONIC ノードは，起動時に AS に対して端末認証を行い，その後 NMS に対してネットワーク情報の登録処理を実行する．Initiator は，Fully Qualified Domain Name（FQDN）によって Responder を指定し，NMS に対して通信開始を要求する．その後，NMS から受信した経路指示に従って，Responder または TRS との UDP トンネルを確立する．また，CYPHONIC ノードの実 IP アドレスが変更された際には，NMS に対してネットワーク情報の再登録処理を実行する．加えて，通信相手との間に新たな UDP トンネルを確立する．アプリケーションデータの通信を行う際には，双方の CYPHONIC ノード間で直接交換した暗号鍵を使用する．これにより，TRS を含めた第三者に対して通信内容を秘匿する．

- Authentication Service (AS)

AS は，CYPHONIC ノード起動時に認証を行い，オーバーレイネットワークの安全性を担保する．AS は，個々の CYPHONIC ノードに紐付くデバイス ID とパスワードによるベーシック認証または証明書認証を行う．また，認証処理に伴い，FQDN や仮想 IP アドレスを選定し，CYPHONIC ノードに付与する．加えて，CYPHONIC ノードと NMS 間の通信で使用する暗号鍵を生成し，共有する．

- Node Management Service (NMS)

NMS は，CYPHONIC ノードの認証処理後もしくは実 IP アドレスが変更された際に，ネットワーク情報の登録処理を受け付ける．これにより，CYPHONIC ノードの実 IP アドレスやそのバージョン，仮想 IP アドレス，NAPT の有無等を管理する．さらに，Initiator からの通信開始要求を受け取った際には，FQDN から仮想 IP アドレスを解決する処理や，通信経路の指示を行う．経路指示に際して，可能な場合には，両 CYPHONIC ノードに対して直接通信を指示する．一方で，直接通信が困難であると判断した場合，TRS に通信中継を依頼する．NMS は通信中継の依頼を発出すると同時に，両 CYPHONIC ノードに対して，TRS を介した通信を行うよう指示する．加えて，CYPHONIC ノードと TRS 間の通信で使用する暗号鍵を生成し，共有する．

- Tunnel Relay Service (TRS)

NMS からの通信中継依頼を受け取った TRS は，双方の CYPHONIC ノードとの間に UDP トンネルを確立する．その後，CYPHONIC ノード間の通信を中継する．TRS の通信中継が必要となる場合の具体例として，両 CYPHONIC ノードが異なる NAPT の配下に接続されている場合が挙げられる．この場合，TRS は CYPHONIC ノードからの UDP Hole Punching による NAPT 越えを利用し，通信中継を実現する．また，TRS は IPv4/IPv6 デュアルスタックを利用し，IP バージョンが異なる端末同士での通信もサポートする．

以上のように，CYPHONIC は，認証済みノードによる安全な E2E 通信を実現する．また，NMS によるネットワーク環境に合わせた経路指示と TRS による通信中継によって，通信接続性を確保する．さらに，NMS によるネットワーク情報の管理および CYPHONIC ノードによる動的な UDP トンネルの確立によって，移動透過性を確保する．

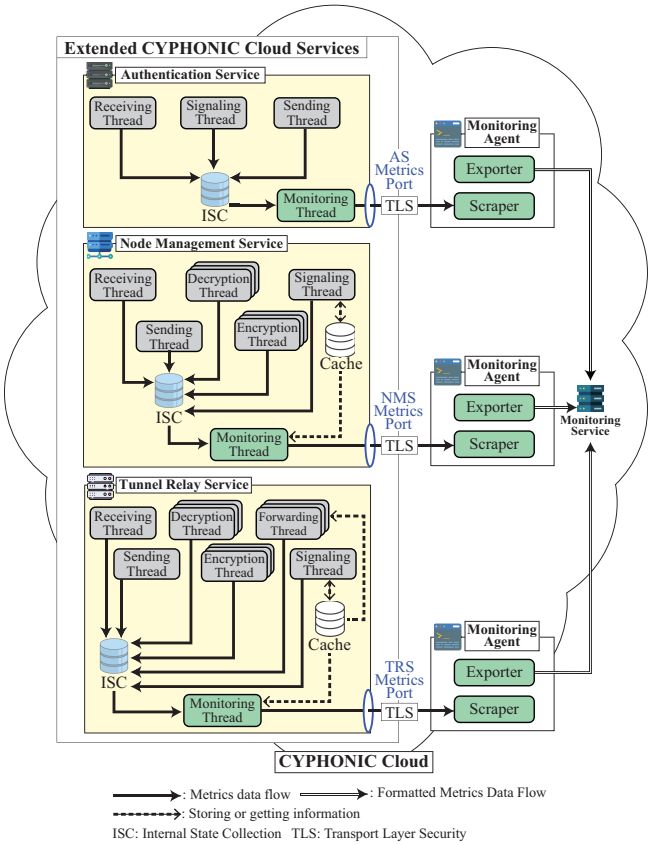


図 2　提案システムモデル

上述のように，CYPHONIC クラウドでは，細分化された役割を担う複数のサービスから構成されている．故に，CYPHONIC クラウドは，サービスごとの柔軟な水平スケールが可能であり，規模拡張による負荷分散を前提として設計されている．

2.3　運用時の課題

CYPHONIC の通信に障害や遅延が発生した場合，その根本的な原因となっている CYPHONIC クラウドのサービスインスタンスを迅速に特定することが重要である．従来の CYPHONIC クラウドでは，ログ情報を目視することがシステムの状態を把握する唯一の手段であった．しかし，水平スケーリングによってサービスインスタンス数が増加する環境においては，手動でログ情報を照合し，システムの状態を観測することは困難を極める．故に，各サービスインスタンスの内部状態を直感的に把握するために，監視システムの実現が求められている．

3.　提案する監視システム

3.1　概　　要

図 2 に提案するシステムモデルを示す．CYPHONIC はあらゆるシステムで活用可能な技術であり，異常発生が様々なケースに影響するため，迅速な異常検知が求められる．また，CYPHONIC クラウドは水平スケーリングを前提としているため，監視サーバの柔軟な負荷対応が求められる．以上より，監視システムの形態として，監視対象の異常検知が容易であり，かつ監視サーバの設定変更のみで負荷制御が可能である，Pull 型を採用した．監視システムを実現するにあたり，監視対象が自身の内部状態

を示す指標を出力する必要がある．このような指標は，一般にメトリクスと呼ばれる [18]．一方で，従来の CYPHONIC クラウドはメトリクスを出力する機能を有しておらず，監視システムの導入が困難であった．そのため，CYPHONIC クラウドの各サービスを，メトリクス出力機能の追加を以て拡張する．また，長期的かつ直感的なシステム監視には，メトリクスの蓄積と可視化が不可欠である．そこで，メトリクスの蓄積と可視化を担う CYPHONIC クラウドのマイクロサービスコンポーネントとして，Monitoring Service（MS）を追加する．加えて，監視対象と MS の間でデータ形式やプロトコルが異なる場合に双方の連携を可能にする，監視エージェントを提案する．

3.2 CYPHONIC クラウドサービスの拡張

CYPHONIC クラウドサービスの拡張にあたり，まず Internal State Collection（ISC）を追加する．ISC は，メトリクスの記録に使用される，各スレッドから操作可能な共有変数である．

次に，既存のスレッドに対してメトリクス記録機能を追加する．想定される操作の例として，パケット受信時の処理を担当する Receiving Thread にて，ISC 内の受信パケット数を記憶するカウンタをインクリメントする処理等が挙げられる．

最後に，Monitoring Thread を追加する．Monitoring Thread は，Transport Layer Security （TLS） サーバとして振る舞うスレッドであり，Metrics Port を開く．Metrics Port は，従来の CYPHONIC の通信とメトリクス収集用の通信を区別するポート番号である．メトリクスを要求する通信を Monitoring Thread が受信した場合，ISC や内部キャッシュからメトリクスを取得し，返信する．この際，TLS クライアント認証を活用することで，攻撃者によってサーバの内部状態が観測される事態を防止する．Monitoring Thread は，特定の監視ツールへの依存を回避するために，高い汎用性を持つデータ形式である JavaScript Object Notation（JSON）を用いてメトリクスを出力する．

3.3 Monitoring Service（MS）の追加

MS は，一定時間ごとに監視対象ソフトウェアからメトリクスを取得し，内部データベースに蓄積する．さらに MS は，蓄積したメトリクスを可視化することで，観測者に対して直感的な監視を提供する．

3.4 監視エージェントの追加

監視エージェントは，各サービスと MS の間を仲介し，双方の連携を実現するソフトウェアである．監視エージェントの役割として，CYPHONIC クラウドサービスが出力したメトリクスデータの整形や，監視対象と MS が使用するプロトコルが異なる場合における齟齬の解消が挙げられる．加えて，監視エージェントは，監視対象サービスの死活状況を MS に伝達する．図 2 に示されているように，監視エージェントは，メトリクスを取得する Scraper Module とメトリクスを MS に渡す Exporter Module から構成される．

4. 実　　装

4.1 CYPHONIC クラウドサービスの拡張

CYPHONIC クラウドサービスにメトリクス出力機構を追加するにあたり，既存実装に倣い Go 言語を使用した．ISC は，

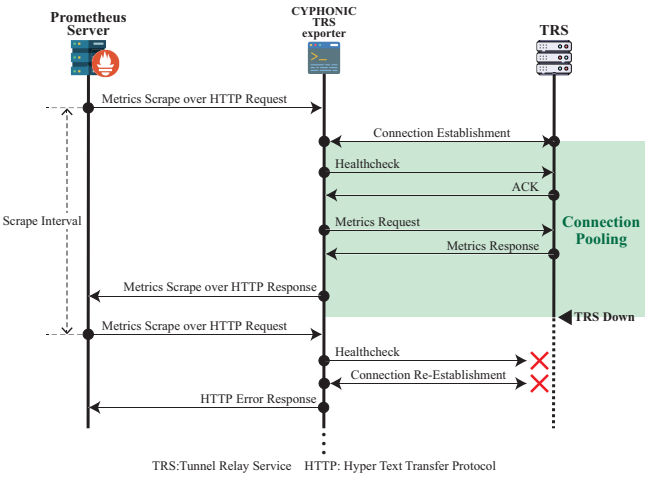


図 3　CYPHONIC TRS exporter の通信シーケンス

複数スレッドから同時に操作された場合でも，保持しているデータの整合性を保つ必要がある．そのため，ISC の作成には，atomic パッケージと排他制御を活用した．

Monitoring Thread の実装では，Go 言語が提供する軽量スレッドである，Goroutine を活用した．CYPHONIC クラウドの正しい内部状態を観測するためには，Monitoring Thread が突発的なメトリクスの変動を見逃さないことが重要である．そこで，瞬時的なメトリクスを扱うデータ構造の実装にあたり，定期的な観測と移動平均の算出を行う機構を活用した．

Monitoring Thread がメッセージを受信した場合，その種類に基づいて以下のメッセージを返信する．

- ACK メッセージ

Healthcheck メッセージを受信した場合，ACK メッセージを返信する．クライアントは，ACK メッセージの受信可否から，サービスや TLS コネクションの状態を確認する．

- Metrics Response メッセージ

Metrics Request メッセージを受信した場合，JSON 形式のメトリクスが含まれた Metrics Response メッセージを返信する．

4.2 Monitoring Service（MS）の導入

MS の実装には，サードパーティ製の監視ツールを活用した．採用する監視ツールは，CYPHONIC の設計思想に対する適合の観点から，マイクロサービス設計思想を採用した監視ツールが望ましい．また，CYPHONIC クラウドサービスは水平スケールを前提として設計されているため，監視ツールには監視対象を動的に補足する機能が求められる．以上の要件を満たすことから，MS として導入する監視ツールに Prometheus を採用した [19]．一方，Prometheus が提供する User Interface（UI）は簡素であるため，データ監視ツールとして Grafana を併用した．

4.3 監視エージェントの実装

著者らは，Prometheus 向けの CYPHONIC クラウドサービス用監視エージェントとして，CYPHONIC exporter を実装した．本実装では，CYPHONIC クラウドサービスに倣い Go 言語を使用した．また，Prometheus 向け監視エージェントの開発を支援するライブラリである，Prometheus client libraries を活用した．CYPHONIC exporter の役割は，CYPHONIC クラウドサービス

表 1　CYPHONIC TRS exporter で観測可能なメトリクス

メトリクス名	データ構造	内容
trs_recv_total	Counter	各種パケットの受信回数
trs_send_total	Counter	各種パケットの送信回数
trs_errors_total	Counter	処理ごとのエラー発生回数
trs_latency_seconds	Histogram	処理ごとの遅延時間
trs_queue_elements	Gauge	内部キューの要素数
trs_cache_elements	Gauge	内部キャッシュの要素数
trs_goroutines	Gauge	Goroutine の個数
trs_memstat_bytes	Gauge	メモリ状況
trs_heap_objects	Gauge	割り当て済み変数の個数
trs_num_gc	Counter	GC の実行回数
trs_num_cpu	Counter	CPU コア数
trs_uptime_seconds	Gauge	起動からの経過時間

と MS の間のインターフェイスとなり，双方を連携させることである．そのため，CYPHONIC exporter は，CYPHONIC クラウドサービスに対しては TLS クライアントとして，MS に対しては HyperText Transfer Protocol（HTTP）サーバとして振舞う．

図 3 に，TRS 用の監視エージェントである CYPHONIC TRS exporter の通信シーケンスを示す．Prometheus は，Scrape Interval と呼ばれる一定の時間間隔ごとに，メトリクスを要求する HTTP リクエストを送信する．CYPHONIC TRS exporter が HTTP リクエストを受信すると，指定された TRS との通信が初めてならば，TLS コネクション確立処理を実施する．続いて，Healthcheck メッセージを使用して TRS の死活状況を確認する．通信に失敗した場合，監視対象が再起動されている可能性を考慮し，コネクションの再確立を試みる．コネクション再確立処理も失敗した場合，CYPHONIC TRS exporter は TRS のダウンを断定し，エラー応答を返信する．Prometheus は，HTTP ステータスコードに基づいて監視対象の死活状況を判断するため，エラー応答によって TRS のダウンを認識する．一方，TRS の生存を確認した場合，CYPHONIC TRS exporter はメトリクスを取得する．その後，JSON 形式のメトリクスを Prometheus client libraries が提供するデータ形式に整形する．最後に，HTTP レスポンスを介して MS にメトリクスを伝達する．表 1 は，CYPHONIC TRS exporter が観測可能なメトリクスの一覧である．

CYPHONIC exporter は，公開するエンドポイントとポート番号を，起動時のオプションによって変更可能である．これにより，導入する環境に応じて柔軟な設定変更が可能である．さらに，CYPHONIC exporter の監視対象を指すネットワーク情報は，HTTP リクエストの GET パラメータで柔軟に指定可能である．これにより監視対象の切り替えが可能であるため，CYPHONIC exporter は 1 つのインスタンスで複数の CYPHONIC クラウドサービスインスタンスを監視可能である．

5. 検証および評価

5.1 検 証 項 目

本検証では，本提案の目的である可観測性の確保を，提案システムが実現しているか否かを評価する．一般に，可観測性という語に対して，観測されるべき具体的な指標は定められていない [18]．そのため，可観測性の評価にあたり，本検証において評価対象とする観測情報を明確に定義する必要がある．そこで，今後の CYPHONIC の運用や開発を鑑みて，死活状況，障害情報，パフォーマンス情報の観測が可能であるか否かに基づいて可観測性を評価することとした．加えて，CYPHONIC exporter の性能検証および拡張を施した CYPHONIC クラウドを使用した通信品質の検証を行い，提案システムの実用性を評価する．本検証は，図 4 および表 2 に示されている環境で実施した．

図 4　検 証 環 境

表 2　検証用コンピュータの諸元

ホストマシン	
OS	Ubuntu 22.04
CPU	11th Gen Core i9-11900K @ 3.50GHz 16cores
メモリ	128GB
仮想化ソフトウェア	Kernel-based Virtual Machine (KVM)
仮想マシン	
OS	Ubuntu 22.04
メモリ	2GB（AS，NMS，TRS），1GB（その他）

5.2 可観測性の検証および評価

まず，仮想環境上に，提案システムを含む CYPHONIC クラウドおよび CYPHONIC ノードをデプロイした．MS として使用したソフトウェアのバージョンは，Prometheus が 1.19.5，Grafana が 9.3.2 である．また，それ以外のサービスの実行に使用した Go 言語のバージョンは 1.19.5 である．次に，TRS の内部状態が観測可能であるかどうかを調査するために，CYPHONIC ノード間の通信が TRS を経由するよう，NMS の経路指示を一時的に変更した．最後に，Prometheus と Grafana のダッシュボードを目視することで，可観測性の評価を行った．

第一に，死活状況の監視が可能であるか確認するために，監視対象プロセスの終了と再起動を繰り返した．その結果，表示されている死活状況が切り替わることを確認した．次に，障害情報が観測可能であるか確認するために，不正な CYPHONIC パケットを意図的に送信した．その結果，パケットの種類ごとにエラー発生回数が可視化された．観測者は，この情報を活用し，効率的にエラーの原因調査を実施可能であると考えられる．最後に，パフォーマンス情報が観測可能であるか確認するために，ping を用いて CYPHONIC ノード間のトラヒックを発生させた．その結果，CYPHONIC クラウドサービスのインスタンスごとにメモリ使用量や内部キャッシュの要素数が可視化された．すなわち，観測者はパフォーマンスに影響する事項を監視可能である．以上のことから，提案システムは可観測性を実現していると結論付けた．

5.3 監視エージェントの検証および考察

メトリクス取得処理は，Prometheus が次なる HTTP リクエストを送出する前に完了しなければならない．すなわち，CY-

表 3　CYPHONIC exporter の応答時間

監視対象	応答時間
1 個	1.01 sec
50 個	1.06 sec
100 個	1.11 sec

表 4　CYPHONIC を使用した通信品質

	従来の CYPHONIC	拡張 CYPHONIC
ICMP の RTT	3.58 ms	3.61 ms
UDP のスループット	13.17 Mbits/sec	13.22 Mbits/sec
TCP のスループット	11.04 Mbits/sec	10.85 Mbits/sec

PHONIC exporter の応答時間は Scrape Interval より短い必要がある．そこで，CYPHONIC exporter の応答性能から使用可能な Scrape Interval を確認し，監視システムとしての妥当性を評価する．性能検証ツールには，複数の HTTP リクエストを同時に送信可能であるという点から，siege を採用した．本検証では，N 個の監視対象インスタンスが存在する環境において，それぞれのメトリクスを要求する N 個の HTTP リクエストを，1 つの exporter に対して同時に送信した．検証結果には 100 回の計測の平均値を採用した．本検証の評価基準は 15 秒とした．この秒数は，Prometheus が公開している標準の監視設定における Scrape Interval である．検証で得られた CYPHONIC exporter の平均応答時間を表 3 に示す．検証結果より，CYPHONIC exporter の平均応答時間は 15 秒より大幅に短いことから，実用的な監視システムを実現可能な応答性能であると結論付けた．

5.4　通信品質の検証および考察

本検証では，本提案による CYPHONIC クラウドの機能拡張による，通信品質への影響を確認する．検証に際して，100 台 50 組の CYPHONIC ノード間で，1 分間のトラヒックを同時に発生させた．Round-Trip Time（RTT）の測定には ping を使用し，UDP および TCP 通信の品質測定には iperf3 を使用した．表 4 に，CYPHONIC ノード間の通信品質を示す．検証結果より，従来の CYPHONIC と比較した通信品質劣化は小さく，本提案による CYPHONIC クラウドの拡張に起因する性能低下は許容可能であると結論付けた．

6. ま と め

本稿では，CYPHONIC クラウドの内部状態を観測する監視システムの実現手法を提案した．検証の結果，提案システムが CYPHONIC クラウドの可観測性を確保すると結論付けられた．加えて性能評価の結果，提案システムが監視水準と通信品質の確保を両立可能であると結論付けられた．本成果は，CYPHONIC のさらなる開発と安定した運用の支援に有用である．

一方，追加検証から，CYPHONIC exporter の一連の処理において，監視対象サービスからメトリクスを取得する処理がボトルネックとなっていることが確認された．この事項について，JSON によるテキストベースでの通信が原因であると考察される [20]．故に今後の展望として，データ形式を JSON から Protocl Buffers に変更する等の高速化手法が考えられる [21]．

文　献

[1] S.Li, X.Bai and S.Wei, “Blockchain-based Crowdsourcing Task Management and Solution Verification Method,” 10.1109/CSE53436.2021.00025, October, 2021.

[2] M. S. Islam and R. Hoque, “SIP over Peer-to-Peer — Implications and existing approaches,” 10.1109/ISCI.2011.5958924, March, 2011.

[3] Lin, Chun-Hung Richard and Zhang, Huan and Liu, Jain-Shing and Chen, Shi-Huang, “Implementation of Secure Web Conferencing,” 10.1109/ICCCI49374.2020.9145972, June, 2020.

[4] H. Shen, Z. Li and K. Chen, “Social-P2P: An Online Social Network Based P2P File Sharing System,” 10.1109/TPDS.2014.2359020, October, 2015.

[5] C. Zhang et al., “Tag-Based Trust Evaluation In Zero Trust Architecture,” 10.1109/IAECST57965.2022.10062213, December 2022.

[6] N. F. Syed, S. W. Shah, A. Shaghaghi, A. Anwar, Z. Baig and R. Doss, “Zero Trust Architecture (ZTA): A Comprehensive Survey,” 10.1109/ACCESS.2022.3174679, May, 2022.

[7] Seth, Sameer and Venkatesulu, M Ajaykumar, “TCP/IP Architecture, Design and Implementation in Linux,” Wiley New York, NY, USA, 2008.

[8] N. Chuangchunsong, S. Kamolphiwong, T. Kamolphiwong, R. Elz and P. Pongpaibool, “Performance evaluation of IPv4/IPv6 transition mechanisms: IPv4-in-IPv6 tunneling techniques,” 10.1109/ICOIN.2014.6799698, Feburary, 2014.

[9] G. Kim, J. Kim and S. Lee, “An SDN based fully distributed NAT traversal scheme for IoT global connectivity,” 10.1109/ICTC.2015.7354671, October, 2015.

[10] J. L. Shah and J. Parvez, “Performance evaluation of applications in manual 6in4 tunneling and native IPv6/IPv4 environments,” 10.1109/ICCICCT.2014.6993065, December, 2014.

[11] Sunghyun Yoon, Soon Seok Lee and Sang-Ha Kim, “Seamless and secure P2P communication scheme for mobile Internet devices behind NAT,” 10.1109/ICCE.2012.6161927, January 2012.

[12] “令和 4 年版 情報通信白書,” 総務省, [Online]. Available: https://www.soumu.go.jp/johotsusintokei/whitepaper/ja/r04/html/nd235100.html

[13] C. E. Perkins, “IP Mobility Support for IPv4, Revised,” RFC 5944, November 2010. [Online]. Available: https://www.rfc-editor.org/info/rfc5944

[14] S. Isomura, T. Yoshikawa, H. Komura, S. Kubota, C. Nishiwaki and K. Naito, “Prototyping evaluation of CYPHONIC: Overlay network technology for Cyber-Physical communication,” 10.1109/CCNC49032.2021.9369500, January, 2021.

[15] T. Yoshikawa, H. Komura, C. Nishiwaki, R. Goto, K. Matama and K. Naito, “Evaluation of new CYPHONIC: Overlay network protocol based on Go language,” 10.1109/ICCE53296.2022.9730323, January, 2022.

[16] M. Xu et al., “CoScal: Multifaceted Scaling of Microservices With Reinforcement Learning,” 10.1109/TNSM.2022.3210211, September, 2022.

[17] N. Marie-Magdelaine, T. Ahmed and G. Astruc-Amato, “Demonstration of an Observability Framework for Cloud Native Microservices,” 2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), Arlington, VA, USA, 2019, pp. 722-724.

[18] K. Hoffman, “Beyond the Twelve-factor App: Exploring the DNA of Highly Scalable, Re-silient Cloud Applications,” O’Reilly Media, 2016.

[19] Bastos, Joel and Araújo, Pedro, “Hands-On Infrastructure Monitoring with Prometheus: Implement and scale queries, dashboards, and alerting across machines and containers,” Packt Publishing Ltd, 2019.

[20] T. Daradkeh, A. Agarwal, N. Goely and M. Zaman, “Real Time Metering of Cloud Resource Reading Accurate Data Source Using Optimal Message Serialization and Format,” 10.1109/CLOUD.2018.00067, July, 2018.

[21] I. I. Lysogor, L. S. Voskov and S. G. Efremov, “Survey of data exchange formats for heterogeneous LPWAN-satellite IoT networks,” 10.1109/MWENT.2018.8337257, March, 2018.