

# Zabbix を利用した NTMobile サーバ統合管理システム

鈴木 洸太<sup>1,†1,a)</sup> 加藤 宏理<sup>1,†2</sup> 鈴木 秀和<sup>1,b)</sup> 内藤 克浩<sup>2</sup>

受付日 2021年11月30日, 再受付日 2022年6月9日,

採録日 2022年7月15日

**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 NTMobile (Network Traversal with Mobility) が提案されている。NTMobile に基づく通信は、エンド端末と複数の NTMobile 独自サーバが連携して通信コネクションを確立するため、障害発生時の原因究明には関係するすべてのサーバを調査する必要がある。そのため、大規模環境下での運用時や開発時に NTMobile の監視負担増加が懸念されている。本稿では、統合監視システムの Zabbix を用いて NTMobile サーバの管理システムを提案する。NTMobile サーバのメモリ上でキャッシュされた情報を可視化するプロセスを導入することにより、通常の Zabbix では管理できないプロセスの内部情報を監視する手法を実現した。仮想環境において動作検証および評価を行った結果、サーバ管理にかかわる作業工程や時間を削減できることを確認した。

**キーワード:** NTMobile, サーバ監視, 障害検知, Zabbix

## Integrated NTMobile Server Management System using Zabbix

KOTA SUZUKI<sup>1,†1,a)</sup> HIROTOSHI KATO<sup>1,†2</sup> HIDEKAZU SUZUKI<sup>1,b)</sup> KATSUHIRO NAITO<sup>2</sup>

Received: November 30, 2021, Revised: June 9, 2022,

Accepted: July 15, 2022

**Abstract:** Network Traversal with Mobility (NTMobile) has been proposed as a technology that provides both IP mobility and End-to-End connectivity in IPv4/IPv6 networks. Since NTMobile-based communication establishes communication connections between end nodes and multiple original NTMobile servers, it is necessary to investigate all the servers involved in order to determine the cause of a failure. Therefore, there is a concern that the monitoring burden of NTMobile will increase when operating in a large-scale environment or during development. In this paper, we propose a management system for NTMobile servers using Zabbix, an integrated monitoring system. By introducing a process that visualizes information cached in the memory of NTMobile servers, we have realized a method of monitoring internal information of processes that cannot be managed by standard Zabbix. As a result of operation verification and evaluation in a virtual environment, we confirmed that the proposed method can reduce the work processes and time involved in server management.

**Keywords:** NTMobile, server monitoring, fault detection, Zabbix

<sup>1</sup> 名城大学大学院理工学研究科  
Graduate School of Science and Technology, Meijo University,  
Nagoya, Aichi 468-8502, Japan

<sup>2</sup> 愛知工業大学情報科学部  
Faculty of Information Science, Aichi Institute of Technology,  
Toyota, Aichi 470-0356, Japan

<sup>†1</sup> 現在, 株式会社パソナテック  
Presently with Pasona Tech, Inc., Chiyoda, Tokyo 100-8228,  
Japan

<sup>†2</sup> 現在, 西日本電信電話株式会社  
Presently with Nippon Telegraph and Telephone West Corporation,  
Osaka, Osaka 534-0024, Japan

<sup>a)</sup> kota.suzuki@ucl.meijo-u.ac.jp

<sup>b)</sup> hsuzuki@meijo-u.ac.jp

## 1. はじめに

スマートフォンやタブレットなどのモバイル端末や5Gなどの無線通信技術の急速な普及により、移動しながら通信を行う機会が増加している [1]。しかし、端末の移動により接続先ネットワークが変化した場合や、Wi-Fi から5Gのように通信インタフェースを切り替えた場合、端末に割り当てられるIPアドレスもともに変化してしまう。そのため、移動前のIPアドレスで確立した通信フローを移動後では識別することができなくなるため、通信が断絶

してしまう課題が存在する。この課題を解決する技術として移動透過性プロトコルが提案されている [2], [3].

一方、現在のモバイルインターネット環境は IPv4 と IPv6 が混在しており、端末の移動先ネットワークによっては、移動透過性プロトコルを実装していても通信が断絶する場合がある。たとえば、IPv4 ネットワークでは、IPv4 グローバルアドレスの枯渇問題に対応するために NAT (Network Address Translation) を導入し、プライベートネットワークを構築することが一般的である。しかし、NAT の外側から内側のネットワークに対して通信を開始できない NAT 越え問題が発生する。さらに、IPv4 と IPv6 の間では互換性が無いため、直接通信を行うこともできない。そのため、端末の接続先ネットワークの種類や通信相手端末のネットワーク環境に影響されることなく、通信接続性を確保しつつ、移動透過性を実現することが要求されている。特に、働き方改革の一環として遠隔地から組織内のネットワークへ安全に接続する需要が高まっており、この問題の解決は重要である [4].

このような背景から、筆者らは IPv4/IPv6 混在環境において通信接続性と移動透過性を同時に実現する NTMobile (Network Traversal with Mobility) を提案してきた [5]-[7]. NTMobile はエンド端末の IP 層で動作するシステムであり、NTMobile を導入したエンド端末同士は暗号化されたトンネルを動的に構築し通信を行う。そのため、既存のアプリケーションを変更すること無く、ネットワークの制約を除去して安全なエンドツーエンド通信が可能となる。

NTMobile に基づく通信では、エンド端末間で暗号化トンネルを動的に形成するために、DC (Direction Coordinator) や RS (Relay Server) と呼ぶ NTMobile 独自のサーバ群が定義され、トンネル形成に必要なエンド端末の情報や RS 中継の有無、暗号鍵などの情報を取り交わす。さらに、NTMobile 特有の処理時に出力されるログは、DC や RS が連携しているため複数台のサーバへ分散することや、サーバが処理時に生成する情報をメモリ上だけにキャッシュすることもある。したがって、サーバ管理者が NTMobile の処理に関して監視や調査を行うには、複数台のサーバからログを取得して突き合わせて調査したり、メモリ上の情報を参照するためにメモリダンプを行って解析する必要があった。このような状況下において、NTMobile 特有の処理において障害が発生した場合、管理者はトンネル形成にかかわった DC や RS を特定し、各種ログなどの情報を収集して解析する必要があり、エラーの原因究明に時間を要する懸念がある。そのため、NTMobile を安定的に運用するために、NTMobile サーバ群を低負担かつ高効率なサーバ管理手法は非常に重要である。

これまでに、上記の課題を解決すべく、NTMobile サーバ群を統合監視する手法が提案されている [8]. NTMobile

独自の拡張 MIB (Management Information Base) を定義し、各サーバの死活監視や CPU 使用率等の負荷情報を SNMP (Simple Network Management Protocol) を用いて取得する。しかし、この手法では拡張 MIB に基づいた規定的な状態情報しか取得できなかったため、NTMobile の一連の処理が正常に動作していない場合の状況分析や原因の特定に必要な任意の情報を網羅的に収集することができなかった。また、NTMobile 機能を有するエンド端末はホームネットワークや社内 LAN、セルラーネットワークに接続している状況がほとんどであり、エンド端末と DC および RS 間には NAT やファイアウォールが存在するため、環境によって SNMP による情報取得が不可能な場合も考えられる。

本稿では、ネットワーク機器等の統合監視システムである Zabbix を活用して NTMobile サーバ群を低負担かつ高効率に監視するシステムを提案する。NTMobile サーバ群のハードウェアリソース情報の監視以外にも、Zabbix 上で NTMobile プロセスがメモリ上でキャッシュしている内部情報の監視を通じた NTMobile 特有の処理に基づく障害の原因調査が行えるようにする。Linux 環境上に実現したプロトタイプを用いて動作検証を行い、提案システムを導入することにより、管理者が監視作業に費やす負担がどの程度削減できるかを示す。

## 2. NTMobile

### 2.1 概要

NTMobile では、ネットワークに依存せず端末の識別子としての役割を持つ仮想 IP アドレスと、接続先ネットワークから割り当てられる位置情報としての役割を担う実 IP アドレスの両方を利用して通信を行う。エンド端末のアプリケーション間で交換されるメッセージには仮想 IP アドレスに基づく IP パケットを利用しており、実際のネットワークでルーティングさせるために実 IP アドレスに基づく UDP/IP ヘッダでカプセル化を行う。

NTMobile を実装したエンド端末は、起動時に DC より仮想 IP アドレスが割り当てられる。通信開始時において、NTMobile を実装した通信相手端末との間で暗号鍵の交換や UDP トンネルの動的な構築を行い、仮想 IP アドレスに基づく IP パケットを UDP トンネルを通じて通信相手端末へ伝送する。これにより、ネットワークの切り替えや通信経路上にある NAT の影響を受けない通信が可能である。さらに、移動や通信インタフェースの切り替えよりエンド端末の接続先ネットワークが変化した場合でも、仮想 IP アドレスは変化しないため通信が断絶しない。

### 2.2 構成

図 1 に NTMobile の概要を示す。NTMobile は下記に示す装置より構成されている。

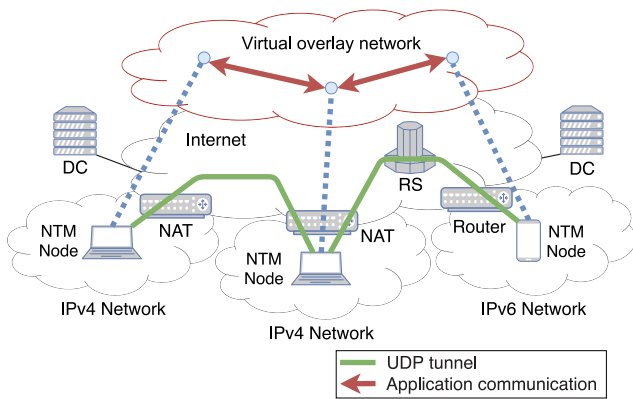


図1 NTM Mobileの概要

Fig. 1 Overview of NTM Mobile.

● NTM 端末

NTM Mobile が実装されたエンド端末である。便宜上、通信開始側 NTM 端末を MN (Mobile Node), その通信相手 NTM 端末を CN (Correspondent Node) と表記する。NTM Mobile を利用するために、任意の DC へログイン処理を行い仮想 IP アドレスを取得する。この仮想 IP アドレスを用いて通信を行う。

● DC (Direction Coordinator)

管理下にある NTM 端末に仮想 IP アドレスの割り当てや、NTM 端末の実 IP アドレス情報などの管理を行う。さらに、NTM 端末の通信開始時に両端末間で最適なトンネル経路を構築するため、関連する端末へトンネル構築に関わる指示を送る。

● RS (Relay Server)

NTM Mobile は経路最適化 [9] により可能な限り NTM 端末間で直接相互通信を行うが、IPv4/IPv6 間の互換性がないネットワーク間で通信する場合や、NAT の種類によっては経路最適化が行えない。このような NTM 端末同士で直接相互通信が行えない場合に通信を中継する。

DC や RS はデュアルスタックネットワークに接続しており、ネットワークの規模に応じて分散設置することが可能である。

2.3 通信シーケンス

図 2 に IPv4 ネットワークに接続されている NTM 端末と IPv6 ネットワークに接続されている NTM 端末間で UDP トンネルを構築して通信を開始するまでの通信シーケンスを示す。NTM 端末は端末起動時に、現在の実 IP アドレスや公開鍵情報を DC に登録するログイン処理を行う。DC はログインした NTM 端末に仮想 IP アドレスを割り当て、NTM 端末のアドレス情報や NTM 端末の公開鍵情報を内部のデータベースに追加する。NTM 端末は通信開始時に通信相手と通信を開始する要求を登録先 DC へ送信する。その後、DC は受信した要求の内容を基に通信

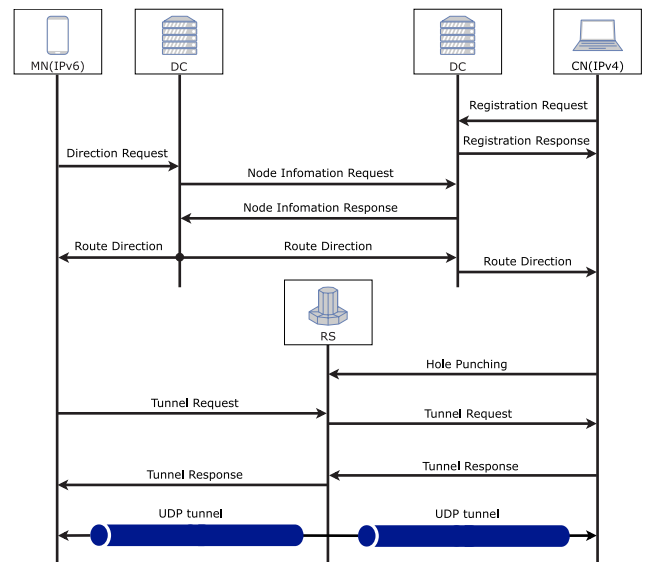


図2 NTM Mobileの通信シーケンス

Fig. 2 NTM Mobile communication sequence.

相手 NTM 端末の管理先 DC を特定し、NTM 端末間のトンネル構築を行うシグナリング処理を実施する。ただし、NTM 端末同士が直接相互通信できないと DC が判断した場合は RS を経由する経路でトンネル構築を行う。

このように、NTM Mobile に基づいた通信は、DC および RS がメッセージの交換を通じて NTM 端末間でトンネルを構築し端末間で通信を行う。

2.4 運用時における課題

NTM Mobile を実環境での運用を想定した場合や開発時において、以下に示す課題が懸念される。

- (1) NTM Mobile 特有の処理にてエラーが発生した場合、サーバリモートアクセスを行い、状況の調査を行う必要がある。特に、NTM Mobile に基づく通信は DC や RS が連携して NTM 端末間の通信経路を構築するため、処理時に出力されるログが複数のサーバへ分散してしまう。そのため、サーバ管理者は複数のサーバからログを取得し、突き合わせて調査する必要がある。さらに、文字列による情報のため、視認性が悪く情報の見落としなどヒューマンエラー発生懸念もある。
- (2) 障害発生時の原因特定に有用な情報が内部情報としてメモリ上のみ存在し、サーバ管理者が参照するためにメモリダンプ処理を行う必要がある。障害発生原因の調査時において、RS が持つリレーテーブルは通信経路の追跡を行う際に有用な情報であるが、メモリ上のみ存在するため、調査に時間を要する。
- (3) RS は通信の中継を行うために両 NTM 端末と UDP トンネルを構築するが、構築時にエフェメラルポートの範囲内 [10] にある UDP ポート番号を UDP トンネルに割り当てる。つまり、使用可能な UDP ポート番号が枯渇状態にある場合、UDP トンネルを新規構築

することができない。しかし、トンネル構築指示を出す DC はすべての RS が持つ割り当て可能な UDP ポート番号の残数を把握しておらず、固定的に選択した RS に対してトンネル構築指示を送るため、RS の状態によっては処理エラーになる可能性がある。また、DC ではログインしている NTM 端末に対して仮想 IP アドレスの割り当てやトンネル構築指示の送信などの処理を行うため、管理下にある NTM 端末台数の増加による通信負荷により処理遅延の発生が予測される。そのため、各 RS で使用可能な UDP ポート番号の残数や DC にログインしている NTM 端末を管理する必要がある。

### 3. 提案システム

#### 3.1 監視システムの要求事項

2.4 節で示した運用時に懸念される課題を踏まえて、NTMobile サーバ群の統合監視システム構築にあたり、システム構築後の運用の手間や NTMobile 自体のサービス品質低下を防止する仕組みとして、本稿では下記の要件を満たす必要があると考えた。

##### 多数の機器の監視が可能

NTMobile は DC や RS がネットワーク規模に応じて分散配置されるため、複数機器を同時にかつ高効率な監視ができることは必須である。

##### 多様な監視データの収集手段があること

監視情報の収集による通信帯域の逼迫を最小限にするため、監視情報取得の頻度や取得方法を柔軟に選択できることが重要である。

##### IPv4/IPv6 混在環境下でも監視データの収集が可能

DC や RS は IPv4/IPv6 が混在したネットワーク環境下において分散的に配置されるため、監視対象となる機器が出力する監視情報を接続先ネットワークによらず収集できることが必須である。

##### NTMobile 独自項目の監視が可能

システムリソース使用量やシステム死活監視など標準的な監視項目以外にも、NTMobile で使用されるパラメータやメモリ上にキャッシュされる内部情報も監視できることは必須である。

##### 再実装なく新規監視項目に対応可能

NTMobile は現在開発段階であり、今後も新機能が実装されることが予想される。これに伴い、監視内容を拡張する場合でも監視プラットフォーム自体に変更を加えない実現方法が望ましい。

#### 3.2 提案システム概要

3.1 節で示した要求事項を基に、NTMobile サーバ群の統合監視システムを構築するために、Zabbix を利用する。Zabbix [11] とは、ネットワーク上に分散する監視対象機

器を統合監視するオープンソースで提供されている管理ソフトウェアである。Zabbix を利用する理由として、標準的なサービスやサービスに依存しないログ等の監視に加え、拡張次第で任意の値を取得および監視できる汎用性の高さ、監視項目等の設定等を複数の監視対象機器に共有できるため Zabbix 自体の管理負担が増大しないこと、GUI を用いた監視が可能であり、監視情報の閲覧や監視項目の設定をマウス操作で完結することが挙げられる。実際に、Zabbix はネットワーク機器の監視以外にもセンサー情報の分析基盤システムなど幅広い場面で利用されている [12], [13]。

Zabbix を利用し NTMobile サーバ群を統合的にかつ詳細に監視できるよう、以下に示す NTMobile および Zabbix の機能拡張を提案する。

#### 3.3 構成

図 3 に提案システムの概要を示す。監視対象となる DC および RS から監視情報を取得するために、デーモンプロセスである Zabbix Agent を各監視対象サーバへ導入する。また、Zabbix Agent が取得した監視情報を回収し総括する Zabbix サーバを NTMobile システム内に導入する。なお、IPv4 および IPv6 上に存在する監視対象機器から監視情報を取得できるよう、Zabbix サーバはデュアルスタックネットワーク上に設置する。これにより、各監視対象サーバから回収した情報を Zabbix 上で集中的に監視することが可能である。

#### 3.4 監視項目

Zabbix では、サーバ死活監視やログファイル回収など一般的に利用される監視項目は標準で搭載されている。これらに加え、NTMobile の動作状況をより詳細に監視するために、NTMobile に関する監視項目を追加する。

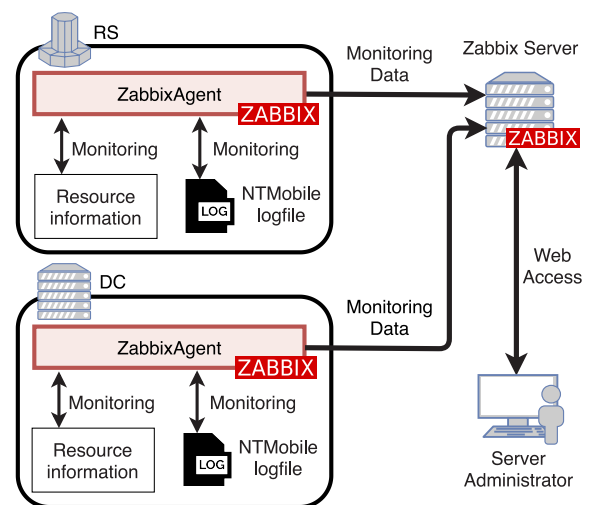


図 3 提案システムの概要

Fig. 3 Overview of proposal system.

### 3.4.1 障害検知

NTMobile 特有の処理に関する障害の検知には、各サーバが出力するログの内容監視およびプロセス死活監視により行う。図 4 に NTMobile が出力するログを用いた障害検知手法の概要を示す。ログファイルの内容検知には Zabbix トリガー [14] 機能を利用する。Zabbix トリガーは、事前に定義した条件式に基づき回収した監視情報を評価し、条件を満たす場合に指定した動作アクションを実行する機能である。提案システムでは、NTMobile 特有の処理に関する障害を管理者が迅速に検知できるように、NTMobile が出力するログ内にキーワードを埋め込む。埋め込んだキーワードが含まれるログを Zabbix サーバが回収した際に、その内容に基づいた障害情報を Zabbix 上に表示するよう Zabbix トリガーの条件式を定義する。

加えて、監視対象上で稼働するプロセスを監視し、NTMobile に関連するプロセスが動作しているか確認することで、NTMobile のシステムダウンをプロセス死活監視からも検知する。

以上により、DC や RS 上で発生した NTMobile 特有の処理に関する障害や関連するプロセスの稼働状況を Zabbix 上で検知することが可能になる。

### 3.4.2 内部情報可視化

RS が持つ内部情報としてリレーテーブルが定義されている [5]。リレーテーブルとは、RS が通信の中継を行うために NTM 端末と RS 間で構築される UDP トンネルと NTM 端末の仮想 IP アドレスを対応付けた情報である。この情報は通信障害発生時の原因究明において通信経路を追跡する際に有用な情報であるが、RS のメモリ上のみ存在する。提案システムでは、Zabbix が提供する機能であるスクリーン [15] を利用し、メモリダンプ操作を必要としないリレーテーブルの可視化を実装する。

スクリーンとは、Zabbix が取得した監視情報を 1 つの画面内に一覧表示し、管理者が監視対象のシステム状況について直感的に概要を把握することを目的とした機能である。スクリーンで表示ができる内容として、各種監視情報のほかに HTTP を用いて取得した HTML ページの表示がある。HTML ページの表示が可能であることに注目し、

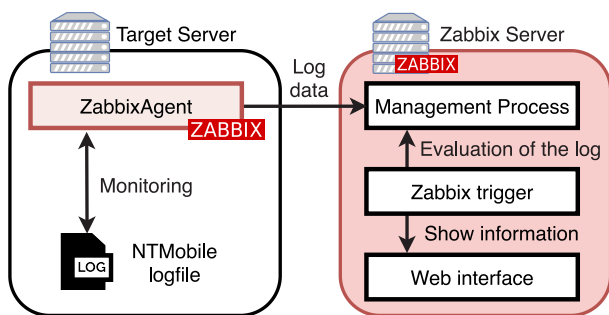


図 4 障害検知手法の概要  
Fig. 4 Overview of fault detection method.

RS 上でリレーテーブルの情報を HTML にレンダリングする機能を NTMobile に追加実装を行い、Zabbix のスクリーンを通じてリレーテーブルを可視化できるようにする。

図 5 にリレーテーブルを Zabbix 上に可視化する手法の概要を示す。リレーテーブルを Zabbix 上で可視化するため、RS に可視化プロセスを追加実装する。そして、RS プロセス内で用いられているリレーテーブルに関する変数を取得できるよう RS プロセスの機能拡張を行い、UNIX ドメインソケットを用いたプロセス間通信により可視化プロセスにリレーテーブルに関する情報を送信する。可視化プロセスでは受け取った情報は Web フレームワークを用いて HTML ページにレンダリングする。

これにより、内部情報を取得するプロセスと Zabbix で用意されているスクリーン機能を組み合わせる方法を用いることで、NTMobile プロセスが内部情報としてメモリ上にキャッシュしているリレーテーブルの情報を直接監視することが可能となる。

### 3.4.3 負荷検知

RS はトンネル作成時に UDP ポート番号をエフェメラルポートの範囲内から割り当てる。しかし、トンネル構築指示を出す DC ではそれぞれの RS におけるエフェメラルポート範囲内の UDP ポート番号残数を把握しておらず、使用可能なポート番号が枯渇している RS にトンネル構築指示を出した場合トンネル構築処理に失敗してしまう。また、DC ではログインしている NTM 端末に対して仮想 IP アドレスの割り当てやトンネル構築指示の送信などの処理を行うため、管理下にある NTM 端末台数の増加による通信負荷により処理遅延の発生が予測される。そこで提案システムでは、RS の割り当て可能な UDP ポート番号の残数や DC が管理している NTM 端末の台数を Zabbix 上で監視できるよう NTMobile および Zabbix を拡張する。

図 6 にエフェメラルポート範囲内にある、RS が割り当て可能な UDP ポート番号の残数を監視する手法の概要を示す。RS で利用可能な UDP ポート番号残数は RS のサー

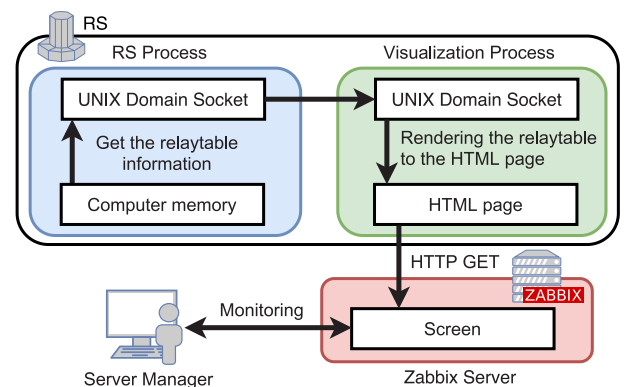


図 5 内部情報可視化手法の概要  
Fig. 5 Overview of internal information visualization method.

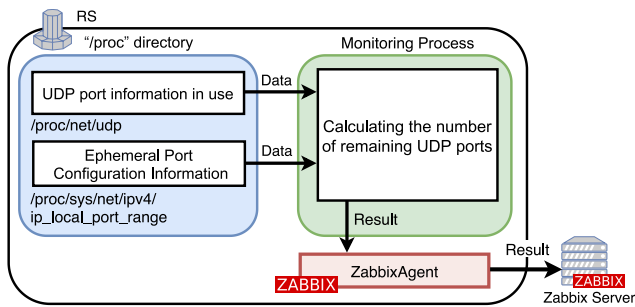


図 6 UDP ポート残数監視手法の概要

Fig. 6 Overview of UDP port remaining monitoring method.

バ内で持つリソース情報から取得する。一般的な Linux システムの場合、使用中の UDP ポート情報およびエフェメラルポート範囲の設定情報が「proc」ディレクトリ内に格納されている [16]。これらの情報から、割り当て可能な UDP ポート番号の残数を取得する監視プロセスを RS のプロセスと並列に処理されるように追加する。また、図 7 に DC の管理下にある NTM 端末台数を監視する手法の概要を示す。DC は、管理下にある NTM 端末について管理するデータベースを DC 内部で持つ。このデータベースにクエリを送信し NTM 端末台数を取得する機能を DC プロセス内に追加する。

上記より取得した各情報を Zabbix 上で監視するために、UserParameters [17] を用いる。UserParameters は、Zabbix で事前に定義されていない監視を行うために独自に定義した監視項目と監視情報を対応付けるキーの役割を持つ値である。これらの情報を監視する監視項目を UserParameters を用いて定義し、実装した追加機能が出力した情報を Zabbix Agent を通じて Zabbix サーバが回収し監視できるようにする。

以上により、Zabbix 上で各 RS で割り当て可能な UDP ポート番号の残数および各 DC で管理している NTM 端末台数を統合的に監視することが可能になる。

また、将来の NTMobile の新機能実装により、監視項目を新設する必要がある場合でも、UserParameters を利用することで、Zabbix 上で監視を行うことも可能である。

## 4. 実装・評価

### 4.1 実装

提案システム動作確認および提案システムを用いた監視作業における所要時間を確認するために、NTMobile の監視に用いる追加機能を DC および RS に実装し、提案システムを構築した。なお、今回の実装にあたり、以下の点に配慮した。

- 監視プロセスは NTMobile と並列して動作させるため、システムリソースの消費を最小限にするために軽量の並列処理が実装できる Go 言語を用いて実装した。

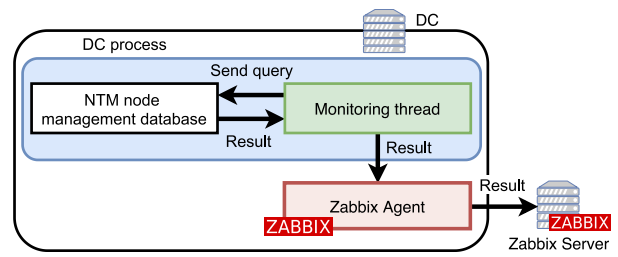


図 7 DC 管理下にある NTM 端末台数監視手法の概要

Fig. 7 Overview of the method of monitoring the number of NTM nodes managed by the DC.

表 1 動作検証環境の諸元

Table 1 Specifications of operation verification environment.

	Zabbix サーバ	DC・RS
CPU	仮想 1 コア	仮想 1 コア
メモリ	2GB	2GB
ストレージ	10GB SSD	25GB SSD
OS	Ubuntu 16.04.6 LTS	Ubuntu 18.04.6 LTS
Zabbix	Zabbix 4.2	Zabbix Agent 4.2.3

- 通信帯域に与える影響を抑えるために、障害に関わる重要度の高い項目であるエラーログ、エフェメラルポート使用率や NTM 端末台数は取得間隔、それ以外の重要度の低い項目は取得するインターバルを長く設定した。
- 各サーバから収集した情報から重要な情報の見落としを防止するために、Zabbix のカラー表示機能に対応するように NTMobile のログに含まれるメッセージの重要度を見直した。

### 4.2 動作検証

Zabbix サーバは VirtualBox を用いて仮想マシン<sup>\*1</sup>で構築した。また、監視対象となる DC および RS は、それぞれ VirtualBox を用いて起動した仮想マシン上で実行し、それぞれに Zabbix Agent を導入した。なお、動作検証に使用した環境については表 1 に示すとおりである。

図 8 に Zabbix 上における NTMobile サーバ群の監視画面を示す。今回実装した監視項目について、Zabbix により情報の取得および統合監視が行えることを確認した。ログの重要度に応じてカラー表示を適切に設定することにより、大量な監視情報の中から必要とする情報を順次に見つけることが可能になり、重要な監視情報の見落としによるヒューマンエラーの防止にもつながる。

<sup>\*1</sup> Zabbix が提供する仮想マシンイメージである Zabbix Appliance を使用している。



図 8 提案システムの監視画面  
Fig. 8 Monitoring screen of proposed system.

表 2 異常の原因究明における作業工程の内容および平均作業時間の比較

Table 2 Comparison of work processes and average work time in investigating the cause of communication errors.

作業概要	従来手法における作業工程	平均作業時間 [s]	提案手法における作業工程	平均作業時間 [s]
監視準備	ターミナル起動	12.48	ブラウザ起動	14.67
	NTMobile サーバに SSH でログイン		Zabbix サーバにログイン	
リレーテーブル情報の取得	NTMobile プロセスの PID 調査	99.10	可視化プロセスと Zabbix による情報収集	—
	GNU Debugger の起動			
リレーテーブルの調査	NTMobile プロセスのメモリダンプ	61.30	「スクリーン」ページ表示	16.80
	ダンプファイル表示		リレーテーブル調査	
ログファイルの収集	ログファイル取得	53.70	Zabbix が収集	—
障害状況の調査	ログ調査	34.30	「障害」ページ表示	18.20
	障害状況の把握		障害状況の把握	
プロセス稼働状況の調査	プロセス稼働一覧の表示	34.30	「障害」ページ表示	18.20
	プロセス稼働状況の調査		プロセス稼働状況の調査	

### 4.3 監視作業時間

#### 4.3.1 実験概要

本稿では提案手法の導入により、管理者の負担がどの程度軽減できるかを明らかにするために、NTMobile プロトコルの処理過程で異常が発生して端末間の通信が開始できない状況が発生したことを想定し、その原因を究明するまでに要する時間を計測した。管理者が手動で各種情報を収集・分析する手法（以後、従来手法）と提案手法の両方を使用して、それぞれの監視作業全体に要した時間を計測した。従来手法では、複数の NTMobile サーバに存在するログファイルを集集し、RS においてはメモリ上にキャッシュされているリレーテーブルの情報をダンプして解析する作業が必要になる。一方、提案手法では Zabbix サーバにアクセスし、統合管理されている情報を解析する作業を行った。

なお、これらの監視作業は、Linux サーバの初歩的な運用スキルがあり、かつ NTMobile に基づく通信の仕組みやプロトコル仕様を正しく理解していないと実施できないため、筆者が所属する研究室において NTMobile の開発に従事している 3 名に被験者となってもらった。また、今回は NTMobile 環境の最小構成である DC および RS を 1 台ずつ稼働している状況で実験を行った。

#### 4.3.2 実験結果

表 2 に両手法における作業概要とそれらの平均所要時間を示す。監視準備に関する作業は提案手法のほうが若干遅いが、それ以外の各作業時間は従来手法より提案手法のほうが短時間で完了していることが確認できる。リレーテーブルの調査については、従来手法ではメモリダンプ処理とダンプファイルの解析を行う必要があり、作業スキルが求められる。被験者はこれらの作業方法を熟知している

が、平均 160 秒程度を要していた。これに対して提案手法では、Web ページに表示された情報を閲覧するだけで解析できるようになったため、約 17 秒まで短縮化することに成功した。

#### 4.3.3 考察

メモリダンプは当該サーバに搭載されているメモリ全体的内容をファイルに出力する。従って、サーバのメモリ容量に準じて出力に要する時間が増減する。今回、RS のメモリ容量は 2 GB であるため、実運用時はさらに大容量のメモリを搭載していることを想定すると、さらに作業時間が増加する。一方、提案手法は RS プロセスからリレーテーブルの情報を直接取得するように拡張したため、メモリダンプは必要なく、サーバの搭載メモリ容量に関係なく短時間で情報の収集・可視化が可能である。

また、今回の評価実験では最小構成の環境で実施したため、被験者はどの NTMobile サーバにログインすればよいのかを特定する必要がなかった。大規模環境で運用する場合、従来手法では障害に関与している NTM 端末の情報を管理している DC を特定する必要がある。また、NTM 端末が通信の中継に利用する RS は DC が動的に決定するため、特定した DC に保存されているログを解析しなければならない。これらの作業時間が表 2 の所要時間に加わることになる。一方、提案手法はすべての DC のログ情報を一元管理できているため、これらの作業時間は大幅に削減できる。そのため、規模が大きくなるほど、従来手法と提案手法の差が大きくなることが分かる。

#### 4.4 大規模運用時における必要スペック

NTMobile サーバ群を Zabbix で統合管理する場合、3.4 節に示した障害検知および負荷検知に関する情報収集は一般的なサーバの監視と同等と挙動であり、NTMobile の使用に伴って負荷が増加することはない。メモリ上にキャッシュされている内部情報の可視化については、独自の監視プロセスを稼働させる必要はあるが、これは Zabbix Agent が稼働する NTMobile サーバ側で動作する。そのため、Zabbix サーバは Zabbix Agent から送信された HTTP GET メッセージを受信して表示するだけであり、この処理も NTMobile 利用時に発生する固有の処理ではなく、一般的な Zabbix サーバと同じ挙動になる。

以上を踏まえると、NTMobile を大規模で運用する場合も、同じ規模の一般的なサーバを監視する場合と同等の負荷が Zabbix サーバに発生することが推測される。文献 [18] に表 3 のように監視対象の規模に応じた Zabbix サーバのハードウェア要件が示されており、NTMobile サーバ群の規模に応じて Zabbix サーバのハードウェア仕様を決定できると考えられる。

表 3 Zabbix サーバの必要スペック

Table 3 Required specifications of Zabbix server.

規模	監視対象台数	CPU	メモリ
小規模	100	1 コア	1GB
中規模	500	2 コア	2GB
大規模	>1,000	4 コア	8GB
超大規模	>10,000	8 コア	16GB

## 5. 残された課題と今後の展望

NTMobile は現在開発中の技術であり、大規模環境下における実運用および評価を行う段階には至っていない。今後は、大規模環境を想定して大量の監視情報が発生した際の NTMobile サーバ群および統合監視システムに発生するハードウェアリソースの負荷や、管理者の負担軽減効果を評価する必要がある。また、監視作業時間に関する定量評価では被験者数を増員したり、サーバ管理技能や NTMobile 運用の習熟度に応じて、提案手法の改善効果をさらに詳細に分析する必要がある。

本稿では各サーバに点在している NTMobile の挙動にかかわる情報を Zabbix を用いて一箇所で統合管理できる仕組みについて示した。次のステップとして、提案手法により集積された様々な情報から動的に異常を検出する仕組みを追加することを予定している。これにより、NTMobile のプロトコル仕様を熟知していない管理者であっても、どの処理過程において、何が原因で通信が失敗しているのかを即座に判断できるようになると考えられる。

## 6. おわりに

本稿では、ネットワーク機器の統合監視システムである Zabbix を活用して、NTMobile サーバ群を高効率かつ低負担に統合管理するシステムについて提案した。また、Zabbix が標準的にサポートしていない NTMobile プロセスがメモリ上にキャッシュしている内部情報を可視化する仕組みを示し、動作検証により実現できることを示した。評価実験を行った結果、提案システムを導入することにより NTMobile サーバ群の監視に関わる作業工程や所要時間を短縮できることを確認した。今回提案した独自のサーバプロセスで管理する情報を Zabbix で統合管理する手法は、NTMobile だけではなく、既存サービスや研究開発中の様々なプロジェクトにも応用可能であり、システム運用時の監視以外にも開発補助ツールとして活用することも可能である。

### 参考文献

- [1] 総務省：令和 2 年版情報通信白書，pp.6-24，日経印刷株式会社 (2020)。
- [2] Le, D., Fu, X. and Hogrefe, D.: A review of mobility sup-



port paradigms for the internet, *IEEE Communications Surveys Tutorials*, Vol.8, No.1, pp.38-51 (2006).

- [3] C. Perkins: IP Mobility Support for IPv4, Revised, RFC 5944, IETF (2010).
- [4] 情報流通行政局：令和2年通信利用動向調査の結果，総務省（オンライン），入手先〈[https://www.soumu.go.jp/johotsusintokei/statistics/data/210618\\_1.pdf](https://www.soumu.go.jp/johotsusintokei/statistics/data/210618_1.pdf)〉（参照 2022-05-29）.
- [5] 鈴木秀和，上酔尾一真，水谷智大，西尾拓也，内藤克浩，渡邊 晃：NTMobileにおける通信接続性の確立手法と実装，情報処理学会論文誌，Vol.54, No.1, pp.367-379 (2013).
- [6] 内藤克浩，上酔尾一真，西尾拓也，水谷智大，鈴木秀和，渡邊 晃，森香津夫，小林英雄：NTMobileにおける移動透過性の実現と実装，情報処理学会論文誌，Vol.54, No.1, pp.380-393 (2013).
- [7] 上酔尾一真，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).
- [8] 田内千裕，上酔尾一真，鈴木秀和，内藤克浩，渡邊 晃：NTMobileにおけるサーバ群の統合的管理システムの提案，第76回全国大会講演論文集，Vol.2014, No.1, pp.263-264 (2014).
- [9] 納堂博史，鈴木秀和，内藤克浩，渡邊 晃：NTMobileにおける自律的経路最適化の提案，情報処理学会論文誌，Vol.54, No.1, pp.394-403 (2013).
- [10] Gleason M.: The Ephemeral Port Range, NcFTP Software. (online), available from 〈[https://www.ncftp.com/ncftpd/doc/misc/ephemeral\\_ports.html](https://www.ncftp.com/ncftpd/doc/misc/ephemeral_ports.html)〉 (accessed 2019-11-07).
- [11] Zabbix SIA.: Zabbix:: The Enterprise-Class Open Source Network Monitoring Solution, Zabbix SIA. (online), available from 〈<https://www.zabbix.com>〉 (accessed 2019-05-18).
- [12] 竹内純人：OSSを活用したグラフィカルなセンサデータ監視システムの構築，電気通信大学紀要，Vol.31, No.1, pp.104-113 (2019).
- [13] Gustafsson, J., Fredriksson, S., Nilsson-Mäki, M., Olsson, D., Sarkinen, J., Niska, H., Seyvet, N., Minde, T. B. and Summers, J.: A Demonstration of Monitoring and Measuring Data Centers for Energy Efficiency Using Open-source Tools, *Proceedings of the Ninth International Conference on Future Energy Systems*, Association for Computing Machinery, pp.506-512 (2018).
- [14] Zabbix SIA.: 3 Triggers [Zabbix Documentation 4.2], Zabbix SIA. (online), available from 〈<https://www.zabbix.com/documentation/4.2/manual/config/triggers>〉 (accessed 2019-12-02).
- [15] Zabbix SIA.: 3 Screens [Zabbix Documentation 4.2], Zabbix SIA. (online), available from 〈<https://www.zabbix.com/documentation/4.2/manual/config/visualisation/screens>〉 (accessed 2019-12-02).
- [16] Bowden, T., Bauer, B., Nerin, J., Feng, S. and Seibold, S.: THE /proc FILESYSTEM, The Linux Kernel Organization (online), available from 〈<https://www.kernel.org/doc/Documentation/filesystems/proc.txt>〉 (accessed 2020-10-29).
- [17] Zabbix SIA.: 4 User parameters [Zabbix Documentation 4.2], Zabbix SIA. (online), available from 〈<https://www.zabbix.com/documentation/4.2/manual/config/items/userparameters>〉 (accessed 2020-11-09).
- [18] Zabbix SIA.: Zabbix Manual, Zabbix SIA. (online), available from 〈<https://www.zabbix.com/documentation/4.2/en/manual>〉 (accessed 2022-06-08).



鈴木 洸太（学生会員）

2020年3月名城大学理工学部情報工学科卒業。2022年3月同大学大学院理工学研究科情報工学専攻修士課程修了。2022年4月株式会社パソナテックに入社。在学時代は主としてモバイルネットワークに関する研究に従事。修士（工学）。



加藤 宏理（学生会員）

2020年3月名城大学理工学部情報工学科卒業。2022年3月同大学大学院理工学研究科情報工学専攻修士課程修了。2022年4月西日本電信電話株式会社に入社。在学時代は主としてモバイルネットワークに関する研究に従事。修士（工学）。



鈴木 秀和（正会員）

2004年3月名城大学理工学部情報科学科卒業。2009年3月同大学大学院理工学研究科電気電子・情報・材料工学専攻博士後期課程修了。2008年4月日本学術振興会特別研究員。2010年4月名城大学理工学部助教。2015年4月同大学理工学部准教授。2020年4月より名古屋大学未来社会創造機構モビリティ社会研究所特任准教授を兼任。2022年4月より名城大学情報工学部准教授。モバイルネットワークおよびユビキタスコンピューティング等の研究に従事。博士（工学）。IEEE, ACM, 電子情報通信学会各会員。本会シニア会員。



内藤 克浩（正会員）

1999年3月慶應義塾大学理工学部電気工学科卒業。2004年3月名古屋大学大学院工学研究科情報工学専攻博士課程後期課程修了。2004年4月三重大学工学部電気電子工学科助手。2007年4月同大学助教。2011年9月カリフォルニア大学ロサンゼルス校客員研究員。2014年4月愛知工業大学情報科学部准教授。2016年情報処理学会・長尾真記念特別賞受賞。無線ネットワーク，モバイルコンピューティングの研究に従事。博士（工学）。IEEE, 電子情報通信学会各シニア会員。本会シニア会員。