5V-06

NTMobileにおけるプロセス単位の通信制御機構の提案

吉田 貴裕 †1　松岡 穂 †2　渡邉 悠雅 †2　鈴木 秀和 †1
†1 名城大学理工学部　†2 名城大学大学院理工学研究科

1 はじめに

IPv4/IPv6 混在環境において移動透過性と通信接続性を同時に実現する NTMobile（Network Traversal with Mobility）[1] の安全性を高めるために，通信相手の FQDN に基づいて通信可否を制御する機能が提案されている [2]．しかし，従来の仕様では通信が許可されている通信相手の場合，NAT やファイアウォールを越えてエンドツーエンドの暗号化通信ができてしまうため，マルウェアなどの不正なプロセスであっても自由に通信が行えてしまうという課題がある．

本稿では，NTMobile に基づく通信の安全性を向上させるために，プロセス単位の通信制御機構を提案する．

2 NTMobile

図 1 に NTM 端末に実装される TUN 利用型 NTMobile [3] のモジュール構成を示す．NTMobile デーモンは，アプリケーションプロセスが送信した DNS 問い合わせパケットとデータパケットを TUN の仕組みでフックし，パケットチェッカーによりパケットの種別を判断する．DNS 問い合わせパケットであれば NTMobile によるトンネル構築シグナリング処理を開始し，データパケットであると判断された場合は構築された UDP トンネル経路を利用して通信相手へ送信される．

文献 [2] では，DNS 問合せパケットに記載されている FQDN から NTMobile 通信の可否を判断し，許可されていればトンネル構築処理が行われる．一度，通信相手との間に UDP トンネルが構築されると，マルウェアや企業が利用を禁止しているプロセスであっても，NTMobile に基づく End-to-End の暗号化通信が行えてしまう課題がある．

3 提案手法

3.1 概要

前述の課題を解決するために，フックしたデータパケットを UDP トンネル経路に流し込む前，すなわちパケットチェッカーにフィルタ処理を追加する．このフィルタ処理は送信しようとしているデータパケットを生成したアプリケーションプロセスを検知し，ユーザが定義したフィルタルールに基づいて通信可否を制御する．フィルタルールは，ポート番号/プロセス名および通信可否情報により構成され，ブラックリスト方式及びホワイトリスト方式のどちらかで定義する．

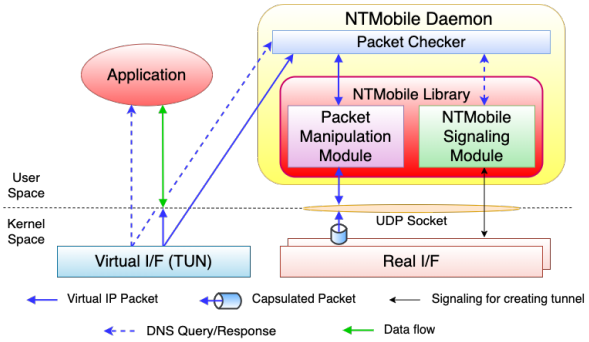

図 1　TUN 利用型 NTMobile のモジュール構成

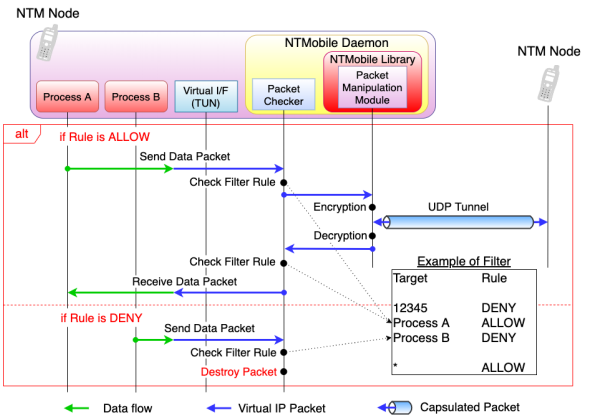

図 2　提案手法を実装した NTMobile 通信シーケンス

3.2 動作

図 2 に提案手法を実装した NTMobile 通信におけるシーケンスを示す．通信を行う両エンド NTM 端末にフィルタルールを定義しているものとする．通信開始側 NTM 端末はフックしたデータパケットの送信元ポート番号とプロトコル番号から，生成されているソケットを特定する．さらに，当該ソケットをオープンしたプロセスを特定し，定義されたフィルタルールをチェックする．当該プロセスの通信が許可されていれば，従来通

A Proposal of Per-Process Communication Control Mechanism in NTMobile

Takahiro Yoshida[†1], Minoru Matsuoka[†2], Yuga Watanabe[†2] and Hidekazu Suzuki[†1]
[†1] Faculty of Science and Technology, Meijo University
[†2] Graduate School of Science and Technology, Meijo University

表 1　測定装置の仕様

	NTM Node A	NTM Node B	DC	RS
OS	Ubuntu 16.04 32bit	Ubuntu 16.04 32bit	CentOS 6.9 32bit	CentOS 6.9 32bit
Kernel	4.15.0	4.15.0	2.6.32	2.6.32
CPU	Intel Core i7-3770	Intel Core i7-2600	Intel Xeon E5-2667 v3	Intel Xeon E5-2667 v3
Memory	8GB	10GB	512MB	1536MB

りの NTMobile 通信を行い，拒否されている場合は当該データパケットを破棄する．

なお，通信相手側 NTM 端末も同様に，受信したデータパケットの宛先ポート番号とプロトコル番号からデータを受け取るプロセスを特定し，フィルタルールに基づいて当該パケットの受理または破棄を行う．

4　実装

フィルタルールを NTMobile の設定ファイルとして実装した．NTM 端末起動時にフィルタルールを読み込み，ハッシュテーブルとして管理した．また，フィルタルールにプロセス名を記述する際，プロセス名またはフルパスのどちらかを用いる．フルパスを記述することにより，異なるディレクトリに同じ名前のプロセスが存在しても，個別に制御することができる．

3.2 節で述べたように，プロセス特定後にフィルタルールをチェックするが，この処理を毎回行うとオーバーヘッドが増加してしまう．そこで，一度通信を行った両端末のソケット，プロセス，通信可否の情報を一定時間キャッシュするようにした．また，キャッシュの管理もフィルタルールと同様にハッシュテーブルを用いた．

5　評価

プロトタイプを実装し動作検証を行った結果，プロセスを指定した通信許可/拒否が正常に行えることを確認した．これに伴い，提案手法が通信に与える影響を明らかにするため，提案手法を実装した NTM 端末にてスループット測定を行った．測定装置の仕様および測定環境をそれぞれ表 1，図 3 に示す．DC，RS は ABLENET が提供するクラウドネットワーク上に VPS（Virtual Private Server）として構築した．なお，スループット測定は iperf3 を用いて 100 回行った．また，両エンド NTM 端末をブラックリスト方式に設定し，フィルタルールの数を 100 とした．さらに，従来手法についても同様の環境で測定を行った．

表 2 に提案手法における送受信パケット単位で発生するオーバーヘッド時間の測定結果を示す．キャッシュを用いることにより，オーバーヘッドが約 0.1% となることが分かった．表 3 にスループットの比較結果を示す．

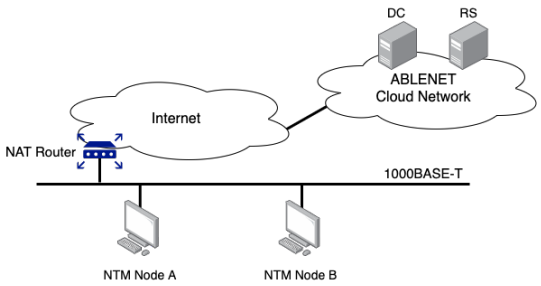

図 3　測定環境

表 2　提案手法のオーバーヘッド時間測定結果

キャッシュ	最小 [s]	平均 [s]	最大 [s]
無し	2.44×10^{-3}	3.22×10^{-3}	3.55×10^{-3}
有り	1.61×10^{-6}	3.25×10^{-6}	5.43×10^{-6}

表 3　スループット測定結果（Mbps）

	最小	平均	最大
従来手法	72.4	80.0	85.4
提案手法	69.8	78.4	84.8

従来手法と比較すると，提案手法のスループットは約 2% の低下に抑えられていた．これはキャッシュ機能を導入した効果によるものと考えられる．したがって，スループット性能は若干低下するものの，実用上の問題が発生することなく，プロセス単位で通信制御を行えるため，セキュリティの向上が期待できる．

6　まとめ

本稿ではユーザが定義するフィルタルールにより，プロセス単位で NTMobile 通信の制御を可能とする手法を実装した．性能評価の結果，スループット特性を維持したまま，通信の制御が行えることを確認した．

参考文献

[1] 上酔尾．他：情報処理学会論文誌，Vol. 54，No. 10，pp. 2288–2299 (2013).
[2] 金松．他：電気学会論文誌，Vol. 137，No. 12，pp. 1571–1579 (2017).
[3] 鈴木．他：DICOMO2014 論文集，Vol. 2014，pp. 1319–1325 (2014).