6D-04

OpenFlow と Open vSwitch による IP スプーフィング機能の実装

三宅　信之†　　栗山　俊通†　　田村　孝之†

三菱電機(株)　情報技術総合研究所†

1. はじめに

ネットワーク機器やシステムを検証する際，複数の送信元 IP アドレス，要件によっては数万の送信元 IP アドレスからのアクセスがなければ正しい検証ができないことがある．例を挙げると，ファイアウォールや IDS，IPS と言った IP アドレスが重要になる NW 機器の機能・性能検証，IP アドレスによるアクセスログ解析機能の検証などである．このような検証においては，マシンに複数の IP アドレスを持たせる IP エイリアス機能や，送信元 IP アドレスを擬装し複数のアドレスからアクセスがあるかように見せかける IP スプーフィングが有効である．

これらの機能を使用するには，検証用アプリケーションによる実装が必要である．しかし，通常 IP アドレスは OS に管理されていることから，ユーザによって万単位の IP アドレスを使い，かつ性能を保つためには，煩雑な実装を必要とする．これに対し，OpenFlow を使用することで，比較的容易に高速動作する IP アドレスの擬装が設定でき，TCP/IP，UDP/IP が成立するよう透過的に IP パケットを書き換えることにより，特定の検証用アプリケーションに依存せずに使用することが可能である．

2. OpenFlow と Open vSwitch

OpenFlow とは，ユーザ作成のアプリ(コントローラ)により，スイッチに送信されたパケットの転送先決定や IP アドレスを含む各フィールドの内容書き換えを行うことで，動的にネットワークを制御する技術である．更に，コントローラがフローエントリ(一致条件・書き換え内容・転送処理)をスイッチに保存することで，同一条件のパケットをスイッチ内で高速に処理できる．

Open vSwitch[2]は Open Flow 対応の仮想スイッチである．OpenFlow 対応の H/W スイッチは，エントリモデルで登録可能フローエントリ数が数千～1 万程度だが，仮想スイッチである本スイッチでは 10 万以上のフローエントリの登録が可能である．また本スイッチは処理の一部がカーネル空間に置かれており，高速に動作する．

An Implementation of IP Spoofing using OpenFlow and Open vSwitch

Nobuyuki Miyake† Toshimichi Kuriyama† Takayuki Tamura†

†Information Technology R&D Center, Mitsubishi Electric Corp.

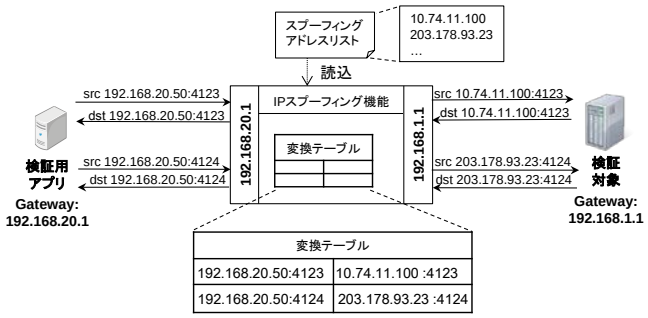

図 1 IP スプーフィング機能の全体像

3. IP スプーフィング機能

本稿で実現する IP スプーフィング機能の全体像を図 1に示す．端的に言えば，本機能は検証用ツール側の IP アドレスとポート番号毎に変換する IP アドレスが異なる NAT ルータである．

本機能はまず，検証用アプリと検証対象とは異なるマシンで動作し，検証用アプリと検証対象の通信を中継する．検証用アプリから送信される TCP/IP・UDP/IP のパケットの送信元 IP アドレスを，送信元 IP アドレスと送信元ポート番号の組によって決まる擬装アドレスに変換し，検証対象に転送する．また，検証対象から擬装アドレス向けに送信される TCP/IP・UDP/IP のパケットの宛先アドレスを検証用アプリの IP アドレスに変換し，検証用アプリに送信する．擬装アドレスはユーザ設定のスプーフィングアドレスリストから決定する．またルータとして，検証用アプリと検証対象のゲートウェイアドレスを持たせることで，それぞれのマシンからパケットを本機能に送信させる．

4. OpenFlow による実現方式

まず，物理的な配置として，OpenFlow スイッチを検証用アプリと検証対象の通信を中継できるようマシンを配置する．

ルータ機能を持つ OpenFlow コントローラは，いくつかのフレームワークでサンプルや作成のための指針が公開されているため，それを参考に実装可能である．このルータ機能をベースに，パケット転送処理において IP スプーフィングできるよう変更する．具体的には，OpenFlow コントローラは検証ツールから，登録済みのフローエントリにマッチしない TCP/IP・UDP/IP パケット(未登録パケット)が送信される毎に，図 2の 2 つのフローエントリを OpenFlow スイッチに登録する．

(1) 検証用ツールに対するフローエントリ
マッチ条件:
・Ingress Port 検証用アプリ側のポート
・IP src 未登録パケットの送信元 IP アドレス
・IP proto 未登録パケットのプロトコル
・TCP/UDP src port 未登録パケットのポート番号
アクション:
・SET_NW_SRC 擬装 IP アドレス
・FORWARD 検証対象側のポート
(2) 検証対象に対するフローエントリ
マッチ条件:
・Ingress Port 検証対象側のポート
・IP dst 擬装 IP アドレス
・IP proto 未登録パケットのプロトコル
・TCP/UDP dst port 未登録パケットのポート番号
アクション:
・SET_NW_DST 未登録パケットの送信元 IP アドレス
・FORWARD 検証用アプリ側のポート

図 2 登録するフローエントリ

ここで，Ingres Port は物理ポート，IP src・IP dst は送信元・宛先 IP アドレス，TCP/UDP src port・TCP/UDP dst port は送信元・宛先 TCP/UDP ポート番号が一致することを意味する．また，SET_NW_SRC・SET_NW_DST はそれぞれ送信元・宛先 IP アドレスの書き換え，FORWARD は指定の物理ポートに送信することを意味する．

本方式では検証対象に対して万単位の同時接続を行う場合，OpenFlow スイッチにその 2 倍のフローエントリを登録しなければならない．仮想スイッチである Open vSwitch はパラメータを調整すれば少なくとも 10 万個のフローエントリをカーネル上にキャッシュとして同時に持つことができ，万単位の同時接続でも高速な動作が期待できる．コントローラと Open vSwitch は同一マシン上で動作可能なため，図 3の構成で本機能が実現できる．

5. 評価

本稿で述べた手法を実装し機能評価と性能評価を行った．まず機能評価であるが，負荷ツールである JMeter と組み合わせ，100 個の IP アドレスを擬装して http 及び https のプロトコルで評価用 WEB サーバにアクセスしたところ，問題なく動作することが確認できた．

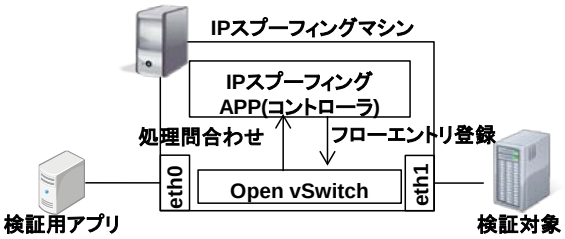

図 3 Open vSwitch を使った構成

表 1 評価環境

CPU	Intel® Core™ i7 (3.07GHz)
メモリ	12GB
NIC	Intel® e1000e (1GbE) × 2
OS	CentOS 6.4 64bit
仮想スイッチ	Open vSwitch 1.11
OpenFlow Framework	trema 0.4.5

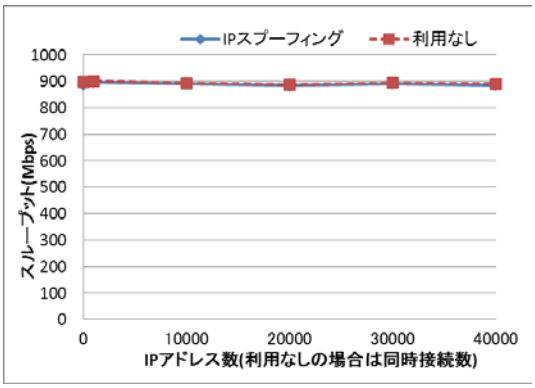

図 4 スループット

また，表 1の環境に IP スプーフィング機能を載せ，自作の評価ツールを使ってスループットの測定を行った．このツールによって指定数の TCP セッションを張り片方から 3 分間データを送信した．通信規格にはギガビット・イーサネットを利用した．評価の結果を図 4に示す．図より，本機能を使用することによるスループットの低下は見られなかった．また，これは TCP 以下のレイヤの各種ヘッダを除いた値であり，ヘッダを考慮に入れると，920Mbps 前後になる．TCP による通信をした上で帯域を 9 割以上使っていることから，より大きな帯域の通信規格を使うことで更にスループットが上がることが期待できる．

6. おわりに

本稿では OpenFlow と Open vSwitch を利用した IP スプーフィング機能の提案を行った．通常，マシンの持つ IP アドレスは OS に管理されており，ユーザによる編集には様々な制限がある．本稿の手法では OpenFlow の動的なフィールド書き換えの仕組みを利用することにより，多数の IP アドレスを擬装し，それぞれの IP アドレスで TCP・UDP による通信を成立させることが可能である．

今後は本機能を様々な評価に活用していく予定である．

参考文献

[1] 高宮安仁ら,OpenFlow 実践入門, 2013
[2] http://openvswitch.org/