様　式　Ｃ－１９、Ｆ－１９－１、Ｚ－１９　（共通）

科学研究費助成事業　　研究成果報告書

平成　３０　年　　６　月　１９　日現在

機関番号：　１２６０１
研究種目：　研究活動スタート支援
研究期間：　2016～2017
課題番号：　１６Ｈ０６６７８

研究課題名（和文）ソフトウェアにより仮想化されたネットワークインフラの監視手法の確立

研究課題名（英文）Observation methods for Softwarerized Network Infrastructure

研究代表者

岡田　和也（OKADA, KAZUYA）

東京大学・情報基盤センター・助教

研究者番号：１０７３２３４９

交付決定額（研究期間全体）：（直接経費）　　2,200,000円

研究成果の概要（和文）：本研究では、Network Function Virtualization (NFV) における監視手法の確立を目指し、容易に構築可能なNFV環境としてコンテナ技術を利用したNFV基盤を研究開発した。コンテナ技術を利用することで、容易に且つ迅速にNFV環境を構築可能な仕組みを実現した。経由するNFの数とパケットサイズを変更し転送性能を評価した。また、約100名が参加した研究会にて提案機構を用いたサービスを提供し、実トラフィックを問題なく転送できることを確認・評価した。

研究成果の概要（英文）：Network Function Virtualization (NFV) needs observation mechanism for stable service operation. For achieving the purpose, we proposed and implemented a lightweight NFV infrastructure called clnfv. clnfv can use for multiple evaluations of NFV research. We evaluated its network performance by changing packet length and the number of NF. We also deployed an NFV service on a research workshop which has over 100 attendees and provides stable service through the infrastructure.

研究分野：　ネットワークアーキテクチャ, MEC, SDN

キーワード：　SDN　NFV

1．研究開始当初の背景

固定・移動体通信事業者では，通信量の増加に耐えうる強固なネットワークが必要とされると同時に，顧客毎に要望の異なるセキュリティ機能，認証，コンテンツキャッシュといった様々な付加機能の提供が求められている．こうした背景から通信事業者では，通信量の増加に伴う設備投資費と運用費を削減しつつ，柔軟に付加機能を顧客に提供するための技術としてネットワーク機能仮想化（NFV: Network Function Virtualization）技術が注目されている．NFV では，ソフトウェアにより仮想化されたルータ，NAT，ファイアウォール等のネットワーク機能（VNF: Virtual Network Function）を用いてサービスを構成する．VNF は，仮想マシン（VM）を利用した仮想アプライアンスとして提供される．そのため，複数の VNF を一つのサーバ上に多重化し，ネットワークサービスに必要な機器数を削減し設備投資費を大幅に削減できる（図 1）．

一方，VNF 間の転送は従来の IP アドレスによる経路とは異なり，利用者単位で転送先の VNF が決定される．従って，IP ネットワークで用いられている死活監視，経路確認手法を適応できず，VNF の障害を迅速に検知できない問題がある．しかし，通信事業者において安定したサービスを提供するには，NFV 環境で発生する障害を検知するための監視機構が必要不可欠である．

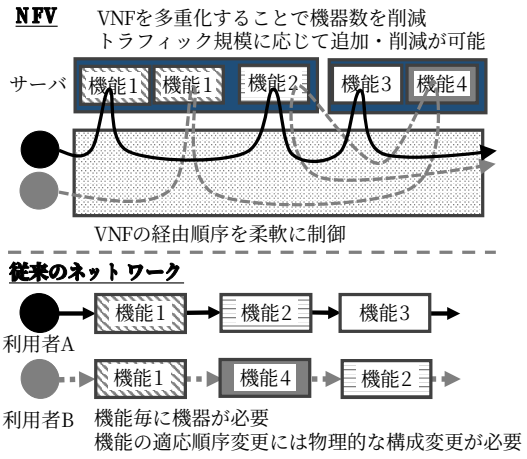


図 1. 従来のネットワークと NFV の比較

2．研究の目的

NFV ではソフトウェアによる機能の仮想化と多重化により論理構造と物理構造の乖離が大きく，通信障害の原因究明が困難になる．特に VNF 間のパケット転送には既存のヘッダ情報に加えて，パケットに埋め込まれた顧客毎の識別子も用いられる．そのため，従来の IP ネットワークで用いられていた ICMP による監視手法だけでは NFV ネットワーク上の障害を検知できない．通信機能のソフトウェア化は設備投資削減と顧客の要求の多様化から避けられない．一方で安定した通信サービスの提供には，正常にサービスが機能していること監視する機構が必要不可欠である．NFV における監視の問題点をまとめると次の 2 点になる（図 2）．

1) VNF の死活・経路監視が困難：VNF の中には，コンテンツキャッシュ，DPI のように装置が IP アドレスを持たない透過型の VNF も存在する．従来の監視に用いられていた ping, traceroute コマンドでは，装置が応答できないため死活監視，経路確認ができない．

2) VNF の正常性確認が困難：VNF では機能に応じてパケットの書き換えや，内容に基づいた応答，破棄が行われる．そのため，たとえパケットが宛先端末に到達したとしても，パケットが正常に処理されたか否かを判別できない．

3．研究の方法

本研究では，検証可能な NFV 環境を実現するために，Linux で用いられているコンテナ技術を用いて構築可能な Container-based Lightweight NFV (clnfv) を提案した．clnfv は，コンテナ基盤として広く利用されいている Docker を NF のホスティング環境として利用し，NF 間のチェイニングを Open vSwitch と OpenFlow により実現する．clnfv を設計・実装し，NFV としての転送性能をトラフィックジェネレータを用いてベンチマークを行った．その結果，コンテナ技術を活用した NFV の性能と問題点を明らかにした．また，約 100 名が参加した学術系イベントにおいて clnfv を用いたネットワークサービスを参加者に提供し，86 時間の連続稼動達成と，最大 120 Mbps のトラフィックを収容した．clnfv の構成を図 2 に示す．

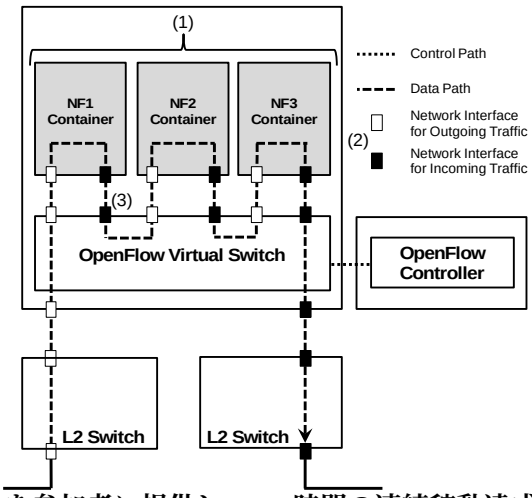


図 2. clnfv の構成

clnfv は一般的なソフトウェアの組み合わせのみで構成された導入が容易で軽量な NFV 基盤である．clnfv の目的はコンテナ仮

想化と OpenFlow のみを用いて NFV 基盤に必要な NF 仮想化とサービスチェイニングの 2 要件を満たすことである．以下では clnfv の構成要素について説明する。

・仮想化

clnfv はコンテナ仮想化を用いて NF を実現する．コンテナ仮想化は仮想マシンを用いた完全仮想化や準仮想化と比較してパフォーマンス，メモリやストレージといった資源の使用量といった観点で有利である．また，コンテナとホストとの通信は完全仮想化されたゲスト OS とホストの通信と比較して，ハードウェアのエミュレーションや，仮想化自体のオーバーヘッドが無いため，軽量である． また，LXC や Docker などのコンテナ仮想化基盤はコンテナをイメージという単位で管理し，外部の Docker や LXC の環境に容易に移植することができる．そのため，第三者が構築した NF を再利用することも容易であり，NF 構築にかかるコストを小さくできる．

・サービスチェイニング

clnfv は複数の NF を利用者が選択的に適用するサービスチェイニングを行う． サービスチェイニングは先行研究や標準化団体の提唱する規格などにおいて様々な手法が研究されている．しかしながら，未だデファクトスタンダードと呼べるものは存在せず，運用可能な形で実装されたプロトコルも少ない．clnfv では導入の容易さを主眼に置き，OpenFlow を用いたサービスチェイニングを選択した． OpenFlow は現時点で一般的に使用されているプロトコルの中で，サービスチェイニングのような特殊なパケット転送に耐えるだけの柔軟性を備えている． また，OpenFlow のコントローラ，スイッチにはいずれも精査された実装が複数存在する．このことは結果的に管理者が構築に際してプロトコルのデバッグなどする必要性を最小限に抑え，構築コストを削減できると考える．

clnfv は L2 リンクの間に物理的に挟み込む L2 透過での設置を想定している．これは，設置に際しての導入コストを極力小さくするためである．先行研究などにおいて示されるサービスチェイニングは利用者端末や他のネットワーク機器に対して特別な設定を要するものが多い．一方で L2 透過での設置は物理的な結線のみで対応できる．これに加えて clnfv では VLAN ID を用いてフローの識別を行うため，VLAN タグをパケットに付与するための設定が必要になるが，VLAN は一般的なスイッチなどには標準で付属している機能であり，導入が容易である．

clnfv のサービスチェイニングは OpenFlow を用いてパケットに付与された VLAN ID と利用者ごとに定義された NF のチェインを一意に対応させ，フローの識別を行う． フローの識別を行った後はフローに一意に対応した ID を IPv4 の TOS (DSCP)フィールド (6bit) に埋め込み，以降は DSCP フィールドに埋め込まれた ID をもとにチェイニングを行う． 元々の VLAN タグをフロー識別に用いない理由は，Linux において Raw Socket などを用いて NF を実現している場合にコンテナ内部で OS が受信したパケットの VLAN タグを Raw Socket に渡る前に外してしまい，チェイニングが正常に出来ないためである．本稿では IPv4 の TOS (DSCP)フィールドを用いているが，OpenFlow の仕様で書き換えを許されていればどのフィールドを用いても構わない． ただし，チェイニングの仕組みに影響を与えないために，用いるフィールドが NF によって変更されないことが保証されていなければならない．DSCP フィールドは一般的な NF においては変更される可能性が低く，関連研究などにおいてもフローの識別に用いられている．そのため，チェイニングに用いる情報を埋め込むフィールドとしてふさわしいと考える．

clnfv は L2 透過での設置を想定しているため，OpenFlow 仮想スイッチと NF は上流側から入力されたパケットは下流側に，下流側から入力されたパケットは上流側に出力する必要がある． そのため，仮想スイッチと NF にそれぞれ上流側と下流側のネットワークインターフェイスを用意するという設計を採用した．そのため，NF を実装する場合，必ず上記の規則に従い，パケットを処理しなければならない． インラインで動作する多く NF（IPS/IDS，ファイアーウォール，DPI など）は，特別な設定をせずともこのような挙動をするため問題はない．

仮想スイッチはチェイニングの際，パケットが入力されたポートと DSCP フィールドにもとづいて経路の決定を行う． 例えば，図[fig:clnfv-arch] (3)に示した NF1 の上流側インターフェースからパケットが入力された場合，OpenFlow 仮想スイッチはそのパケットの DSCP フィールドを参照し，対応するチェインの特定を行う． そのチェインにおいて次の NF が NF2 であった場合，（入力は上流側のインターフェースからであったため）NF2 の下流側からパケットを入力する．

clnfv のプロトタイプにはコンテナ仮想化基盤に Docker v17.05.0-ce，OpenFlow スイッチに Open vSwitch v2.5.2，OpenFlow コントローラには Ryu v4.16 を用い，OpenFlow は v1.3 を使用した.いずれのソフトウェアも Linux や FreeBSD といった主要な OS 上で動作し，導入が容易である．実装に際しては Linux ディストリビューションの Ubuntu16.04- LTS を用いた.

表 1. 評価環境

構成要素	詳細
L3 switch	Dell S3048-ON
Linux server	Ubuntu 16.04 LTS Intel(R) Xeon(R) E5-1410@ 2.80GHz (8 cores) 24GB RAM
Docker	version 17.05.0-ce
Open vSwitch	version 2.5.2
OpenFlow	version 1.3

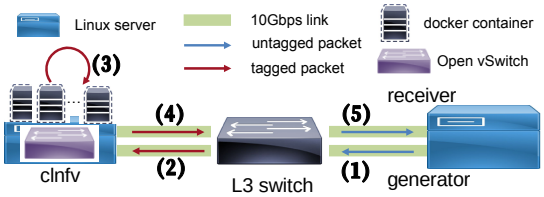


図 3. 評価トポロジ

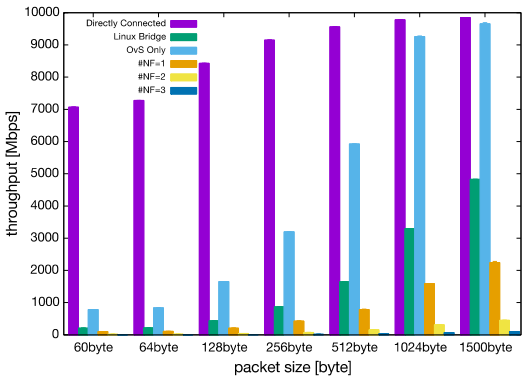


図 4. NF の数と転送性能の関係

・ベンチマーク

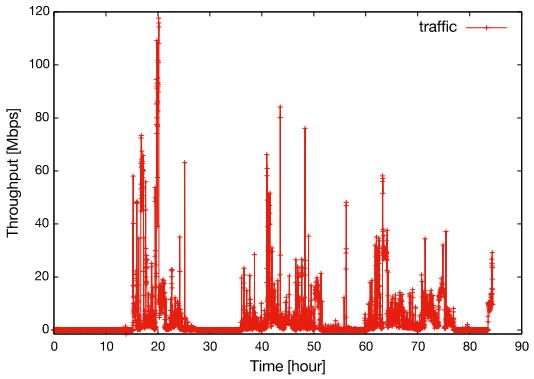
clnfv の基本性能を示すために，図 3 の環境でベンチマーク測定をした．また，評価環境の構成要素の仕様は表 1 に示すとおりである．clnfv は Linux マシン上に構築し，トラフィックジェネレータとレシーバには netmap に付属している pkt-gen を用いた．ベンチマークにおけるデータの流れは以下のとおりである．(1)ジェネレータプロセスがパケットを生成して L3 スイッチへ転送する．(2)L3 スイッチが VLAN タグを付加して clnfv へ転送する．(3)VLAN タグ付きのパケットが clnfv 内でチェイニングされる．(4)clnfv が VLAN タグ付きのパケットを L3 スイッチに転送する．(5)L3 スイッチが VLAN タグを外して，パケットを受信プロセスに転送する．

clnfv の転送性能を示すために，チェイニングで通過する NF の数と，パケットサイズを変化させてそれぞれのスループットを計測した．各 NF は，受信したパケットを Open vSwitch に送り返すだけのものを利用し，パケットの書き換えやその他の処理を行わない．Open vSwitch と NF の間は 2 本の veth で接続されている．コンテナ内における 2 本の veth は，Linux ブリッジで接続されている．NF の数は 0 個 (OvS Only) から 3 個 (NF=3) で変化させる．また，比較として，clnfv を通過しないでデータを転送する場合 (直接接続) と，clnfv の代わりに Linux ブリッジ (Linux Bridge) を用いた場合を計測した．図 3 は，横軸にパケットサイズ [byte]，縦軸にスループット [Mbps] をとり，それぞれの条件下で 60 回計測した結果のエラーバー付き棒グラフである．グラフ中で各パターンごとの結果を色分けして示している．グラフからは，パケットサイズを 60 バイトに設定した場合，clnfv を通過しない場合のスループットが 7Gbps であったのに対し，NF の数が 1 個の場合は 100Mbps，2 個の場合には 20Mbps，3 個の場合には 4Mbps であった．また，パケットサイズを 1500 バイトに設定した場合，clnfv を通過しない場合のスループットは，9.8 Gbps で，NF の数が 1 個の場合は 2.2 Gbps，2 個の場合は 452 Mbps，3 個の場合は 101 Mbps であった．パケットが clnfv を通過するとスループットが低下するのは，いずれの場合も，Docker コンテナ内における 2 本の veth の接続にそれぞれ Linux ブリッジが使用されているからだと考えられる．実際に，図 3 の Linux Bridge の結果を見ると，パケットサイズが 1500 バイトの場合のスループットは 4.8 Gbps を示しており，スループットのボトルネックになったと考えられる．パケットサイズがいずれの場合でも，NF の数が 1 個，2 個，3 個と増えるにつれて，スループットは約 5 分の 1 になった．これは，Linux ブリッジはソフトウェアとして実装されているため，同時に動作する数が増えると CPU 資源が圧迫され，さらに性能が低下したと考えられる．また，Linux ブリッジと Open vSwitch の転送性能を比較すると，パケットサイズがいずれの場合でも Open vSwitch のほうが 2 倍以上の性能を示した．

・実環境における動作検証

実装した clnfv を実際のネットワーク環境に展開し，運用評価を行った．この評価では，3 泊 4 日の学術系イベントネットワークにて clnfv を，イベント参加者の無線 LAN ネットワークに組み込んで検証した．この評価では，ファイアーウォール (iptables) と侵入検知システム (Snort) の 2 つの NF を通過するようなチェインを参加者に提供した．当該ネットワークの日中の接続数は 10 台程度で，最大 17 台であった．図 4 は，運用期間中の帯域利用量を 1 分単位でグラフ化したものである．スループットの最大値は約 120Mbps であり，日中のスループットは平均して 20Mbps 程度であった．図 5 は，運用期間におけるデータ

量の累積をグラフ化したものである.運用期間において，合計 160 ギガバイトのデータを

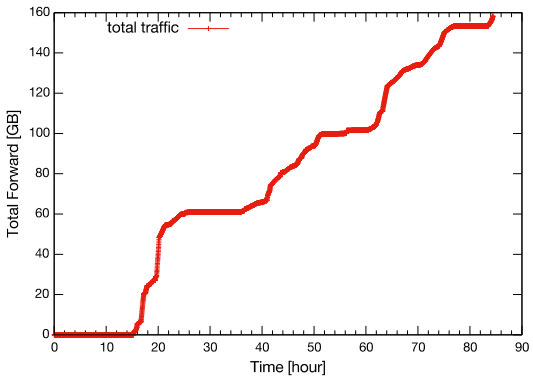


転送した.clnfv は約 86 時間稼働し続けた.このことから，15 人程度のトラフィックであれば 3 日間連続で処理できることを示した.

図 4．clnfv のスループット（時系列）

図 5．clnfv で転送された累積トラフィック量

評価で示された 10Gigabit Ethernet (10G) 環境でのベンチマーク結果は Linux ブリッジが clnfv のパケット転送における大きなボトルネックとなっていることを示している．先行研究においても Linux や FreeBSD といった既存の汎用 OS のネットワークスタックが 10G 環境においてはボトルネックとなりうることが指摘されている． 特に複数の NF を経由する NFV 基盤では，Linux ブリッジによる転送性能の劣化がより顕著に現れてしまう．この問題の解決策としては，netmap や intel 社の DPDK と呼ばれるカーネルバイパス技術



を用いた高速化手法が提案されている． これは，従来カーネルを介して行われていたパケット処理をバイパスし，ユーザランドのプロセスにおいて直接パケット処理を可能にすることで処理の高速化を図るものである．しかしながら，これらの高速化手法は既存のカーネルに対しての変更が必須であり，且つ導入コストが高いのが難点である． clnfv において NF 間のパケット転送に用いている Open vSwitch は，DPDK を用いた高速化の拡張機能を有しており比較的容易にこれらの高速化手法の恩恵を受けることができる．また，Lagopus のような DPDK を用いた OpenFlow スイッチも実装されているため，Open vSwitch を置き換えることで一定のパフォーマンスの向上が見込める． NF 側では，Linux ブリッジを用いる iptables や Snort，Suricata（IDS/IPS)など個別の NF 毎に対応が必要となる． 既に Snort では，DPDK を用いた実装があり，パフォーマンスの向上が見込める． ただし，NF 毎に netmap/DPDK に対応した実装が必要となる．

４．研究成果

本研究では，簡素で軽量な NFV 基盤を目指し，Linux のコンテナ技術を用いた NFV アーキテクチャ clnfv を提案し，Docker を用いた設計と実装を示した．clnfv の特徴は，ネットワークファンクション（NF）の管理と NF 間の連接を一般的なソフトウェアの組み合わせで実現しており，導入と運用が容易な点である．ベンチマークによる転送性能評価の結果，10G ネットワークにおいてパケットサイズが 1500 バイトのときのスループットが，NF の数が 1 個で 2.2Gbps，2 個で 452Mbps，3 個で 101Mbps となり，経由する NF 数が増えるごとに転送性能が大きく劣化することが明らかになった．また，約 100 名が参加した学術系イベントにおいて clnfv を用いたネットワークサービスを参加者に提供し，86 時間の連続稼動達成と，最大 120Mbps のトラフィックを収容し，合計で約 160 ギガバイトのデータを転送した．

今後は，今回明らかになった NF の転送性能の劣化を改善するために，コンテナネットワークにおけるパケット処理高速化技術の活用やその研究を行っていく．

５．主な発表論文等

（研究代表者，研究分担者及び連携研究者には下線）

〔学会発表〕（計 1 件）

早川侑太朗，渡邊 大記，岡田 和也，“clnfv: コンテナを用いた軽量な NFV 基盤”，信学技報，vol．117，no．303，NS2017-111，pp．1-6，2017 年 11 月．

６．研究組織

(1)研究代表者

岡田　和也　（OKADA，Kazuya）
東京大学　情報基盤センター・助教
研究者番号：10732349