

# 生化学反応モデルに着想を得た VNF 制御手法に関する一検討

黒川 稜太<sup>†</sup> 長谷川 剛<sup>††</sup> 村田 正幸<sup>†</sup>

<sup>†</sup> 大阪大学大学院情報科学研究科 〒 560-0871 大阪府吹田市山田丘 1-5

<sup>††</sup> 大阪大学サイバーメディアセンター 〒 560-0043 大阪府豊中市待兼山町 1-32

E-mail: <sup>†</sup>{r-kurokw,murata}@ist.osaka-u.ac.jp, <sup>††</sup>hasegawa@cmc.osaka-u.ac.jp

あらまし ネットワーク機能仮想化技術 (Network Function Virtualization: NFV) では、様々な仮想ネットワーク機能 (Virtual Network Function: VNF) が汎用サーバ上で実現され、それらが適応的に移動、追加、及び削除されることによって、運用コストや設備投資コストを抑え、またシステム障害等の環境変動に対して柔軟に対応することが可能になる。NFV に基づくネットワークシステムを効率的に運用するためには、各 VNF の配置、サーバ資源の割り当て、及びネットワークフローの経路等を適切に決定することが求められる。さらに、NFV のようなネットワークサービスは、需要変動、ネットワークの輻輳、及びシステム障害等に素早く対応するために、自律分散的に動作することが望ましい。我々の研究グループでは、生化学機構に着想を得た、タプル空間モデルに基づくサービス空間構築手法を提案している。本報告では、その手法を NFV フレームワークへ適用し、その実装方針を検討した結果を示す。また、NFV において重要な機能であるサービスチェイニング要求について、IETF で標準化が進められている Network Service Header を用いた実装方針を示す。最後に、アプリケーションの実装方針について述べる。

キーワード 生化学反応式, ネットワーク機能仮想化技術 (NFV), Software Defined Network (SDN), サービスチェイニング (Service Function Chaining: SFC), Network Service Header (NSH)

## 1. はじめに

近年、スマートフォンやタブレット端末の普及、及びモノのインターネット (Internet of Things: IoT) [1] 技術の発展など、ネットワークを取り巻く環境の変化により、ネットワークに接続されるハードウェア機器の種類が増加し、トラフィック量も増加している。新しいネットワークサービスを立ち上げるには、専用のハードウェア機器が必要となり、設置するための空間や稼働するための電源を必要とするため、コスト増大による利益減少やエネルギー消費量の増大等の問題が深刻となっている [2]。また、ハードウェア機器の保守運用やシステム障害等の環境変動への対応にも手間がかかり、柔軟性が低い。

このような問題に対処するための 1 つの技術として、ネットワーク機能仮想化技術 (Network Function Virtualization: NFV) [3] がある。図 1 に、NFV が適用されたネットワーク環境の例を示す。NFV は、仮想ネットワーク機能 (Virtual Network Function: VNF) を汎用サーバ上で実現して提供する技術であり、通常はハードウェア機器で実行されるネットワーク機能がソフトウェアとして実現されるため、運用コストや設備投資コストを抑え、またシステム障害等の環境変動に対して柔軟に対応することが可能になる。VNF は、ファイアーウォール [4] や Deep Packet Inspection (DPI) [5] 等のネットワーク機能や、Evolved Packet Core [6] のようなモバイルコアネットワークの機能など、多種多様なネットワーク機能を対象に考えられている [7]。その結果、様々な VNF が汎用サーバ上に配置され、複数の VNF が 1 台のサーバ資源を共有したり、1 つの VNF が複数のサーバ上で分散実行される [8, 9]。

また、NFV では、ネットワークフローが適用されるべき VNF の順序を示すサービスチェイニング要求を持つ。図 1 は、NFV 環境に到着したネットワークフローが、そのパケットが持つサービスチェイニング要求に従って、複数の VNF が順に適用され、フローが退出する様子を表す。このような NFV に基づく

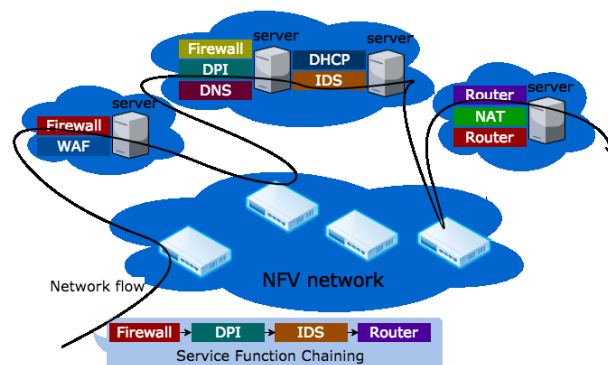


図 1 ネットワーク機能仮想化 (NFV) 環境

ネットワークシステムを効率的に運用するためには、各 VNF の配置、サーバ資源の割り当て、及びトラフィック量やサーバ負荷に応じたネットワークフローの経路などを適切に決定する必要がある。さらに、NFV のようなネットワークサービスは、需要変動、ネットワークの輻輳、及びシステム障害などの環境変動に素早く対応するために自律分散的に動作することが望ましい。

そのような動作を表現する 1 つの方法として、自律分散性や自己組織性の高い、生化学機構を用いる手法がある。我々の研究グループでは、生化学反応式を用いたタプル空間モデルに基づくサービス空間構築手法を提案している [10-12]。この手法では、サーバをタプル空間と考え、サービス要求、サービス需要、及びサーバ資源を化学物質として表現し、サーバの挙動をタプル空間内の化学反応式として記述する。さらに、複数のタプル空間を接続してネットワークを構成することで、複数のサーバから構成される大規模ネットワークシステムにおける、サービスの拡散やサービスに対する要求の移動を表現することができる。また、化学反応式はそれぞれのタプル空間で独立に定義されて動作するため、自律分散的な挙動をモデル化するのに適し

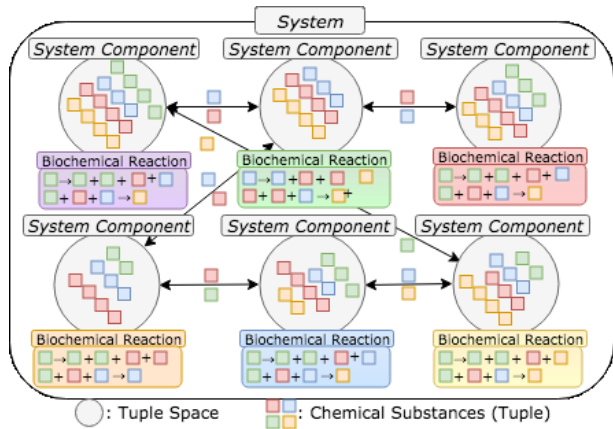


図2 生化学反応式を用いたタプル空間モデル [12]

ている。しかし、過去の研究では、コンピュータシミュレーションによる基本的な動作検証、及び簡易な実験環境における検証のみが行われており、NFV 環境への適応については示されていない。

そこで本報告では、上述の手法を NFV 環境へ適用するために、NFV フレームワーク [13] における提案手法の実装方針について検討する。初めに、文献 [12] で示されている、生化学反応式を用いたタプル空間モデルに基づくサービス空間構築手法を NFV へ適用する方法に関して説明する。次に、NFV フレームワークに提案手法を適用する方法について説明する。さらに、NFV のオープンソースソフトウェア (Open Source Software: OSS) である Open Platform for NFV (OPNFV) [14] を用いた NFV フレームワークの実装環境を示し、提案手法の実装方針を述べる。また、NFV において重要な機能であるサービスチェイニング要求について、Internet Engineering Task Force (IETF) で標準化が進められている技術の 1 つである Network Service Header (NSH) [15] を用いた実装方針について述べる。最後に、提案手法の有効性を検証するために、動画アプリケーションを想定した提案手法の実装方針を述べる。

本報告の構成は以下の通りである。まず 2 章では、本報告で用いる、生化学反応式を用いたタプル空間モデルに基づくサービス空間構築手法を NFV へ適用する方法について述べる。3 章では、NFV フレームワークにおける提案手法の実装方針について述べる。最後に 4 章で本報告のまとめと今後の課題について述べる。

## 2. 生化学反応式を用いたタプル空間モデルに基づくサービス空間構築手法

本章では、文献 [12] で示されている、生化学反応式を用いたタプル空間モデルに基づくサービス空間構築手法を NFV へ適用する方法に関して説明する。

図 2 に、生化学反応式を用いたタプル空間モデルの概要を示す。タプル空間を化学反応が起こる場とする。タプル空間内のタプルは化学物質に相当し、その量は化学物質の濃度に相当する。タプル空間内で起こる化学反応を定義することで、タプル濃度の変化等の挙動が定められる。また、タプル空間の外部からの化学物質の投入や、生成物を外部に送出する挙動を記述することによって、タプル空間外との相互作用を実現することができる。各タプル空間で起こる化学反応は独立に起こるため、自律分散的な動作を表現することができる。

上述のモデルを NFV へ適用する場合には、タプル空間は VNF を提供するサーバに、タプルは VNF、フローのパケット、及びサーバの資源量等にそれぞれ対応付ける。以降では、文献 [12] で提案されている、NFV の様々な挙動を実現するための化学反応式を示す。

### 2.1 サービスチェイニング要求

あるフローに対して、そのサービスチェイニング要求により実行される VNF を順に  $f_1, f_2, f_3, \dots$  とすると、サービスチェイニング要求  $c$  を式 (1) のように表す。

$$c = \{f_1, f_2, f_3, \dots, f_{end}\} \quad (1)$$

サービスチェイニング要求  $c$  を持つフローに対して、VNF  $f_1$  が実行された場合、サービスチェイニング要求  $c$  は式 (2) のように変化する。

$$c \leftarrow c \setminus \{f_1\} = \{f_2, f_3, \dots, f_{end}\} \quad (2)$$

サービスチェイニング要求  $c$  が次に実行を要求している VNF を  $f^1(c)$  と表す。式 (2) の例では、 $f^1(c) = f_2$  である。以下では、化学物質の添え字  $f$  はフローを、 $c$  はサービスチェイニング要求を表す。

### 2.2 VNF への資源の割り当てと実行

サービスチェイニング要求  $c$  を持つフローがサーバへ到着すると、サーバにおいて対応する VNF が存在する場合には、パケットに対して VNF が適用される。このとき、サービスチェイニング要求が 1 つの VNF から構成されている場合には、VNF が適用された後にそのパケットは消失する。一方、サービスチェイニング要求が複数の VNF から構成されている場合には、パケットが持つサービスチェイニング要求から適用した VNF を削除する。また、需要の少ない VNF は衰退し、需要の多い VNF は成長させることが求められる。このような挙動を反応式 (3), (4) のように表す。

$$VNF_{f^1(c)} | PKT_c \xrightarrow{r_{us}} \begin{cases} VNF_{f^1(c)} | VNF_{f^1(c)} | PKT_c \setminus \{f^1(c)\} \\ |toserve(VNF_{f^1(c)}, PKT_c) (c \setminus \{f^1(c)\}) \neq \emptyset \\ VNF_{f^1(c)} | VNF_{f^1(c)} \\ |toserve(VNF_{f^1(c)}, PKT_c) (c \setminus \{f^1(c)\}) = \emptyset \end{cases} \quad (3)$$

$$VNF_f \xrightarrow{r_{ds}} 0 \quad (4)$$

$VNF_{f^1(c)}$  は、サービスチェイニング要求  $c$  が次に実行を要求している VNF を表し、その濃度が大きいほどその VNF に対する需要が多いことを表す。 $PKT_c$  は、サービスチェイニング要求  $c$  を持つフローを構成するパケットを表す。 $toserve(VNF_{f^1(c)}, PKT_c)$  は、サービスチェイニング要求  $c$  を持つフローに対して、次に適用すべき VNF が実行されたことを表す。

また、VNF が配置されているサーバの資源量に応じた VNF の実行速度の制約を記述するために、以下の反応式 (5) - (7) を導入する。

$$VNF_{f^1(c)} | PKT_c | RSRC_t \xrightarrow{r_{us1}} MEDIANE_{f,t} \quad (5)$$

$$MEDIANE_{f,t} \xrightarrow{r_{us2}} VNF_{f^1(c)} | PKT_c | RSRC_t \quad (6)$$

$$MEDIANE_{f,t} \xrightarrow{r_{us3}} \begin{cases} VNF_{f^1(c)} | VNF_{f^1(c)} | PKT_c \setminus \{f^1(c)\} | RSRC_t \\ |toserve(VNF_{f^1(c)}, PKT_c) (c \setminus \{f^1(c)\}) \neq \emptyset \\ VNF_{f^1(c)} | VNF_{f^1(c)} | RSRC_t \\ |toserve(VNF_{f^1(c)}, PKT_c) (c \setminus \{f^1(c)\}) = \emptyset \end{cases} \quad (7)$$

$RSRC_t$ 、及び  $MEDIANE_{f,t}$  は、サーバ  $t$  における利用可能な資源量、及びサーバ  $t$  が提供している VNF  $f$  に割り当てられた資源量をそれぞれ表す。反応式 (5) 及び (6) は、サービスチェイニング要求  $c$  のフローを構成するパケットが適用すべきサーバ  $t$  に到着する際に、各 VNF の需要の大きさに応じて資源が割り当てられる様子を表す。反応式 (7) は、VNF  $f$  に割り当てられた資源量に基づいて、VNF が実行されることを表す。

### 2.3 VNF の拡散

サーバにおける需要の多い VNF が、他のサーバへ VNF が拡散することを表現するために、以下の反応式 (8) を導入する。

$$VNF_f \xrightarrow{r_{ms}} VNF_{\sim f} \quad (8)$$

反応式 (8) は、サーバにおける需要の多い VNF がその濃度

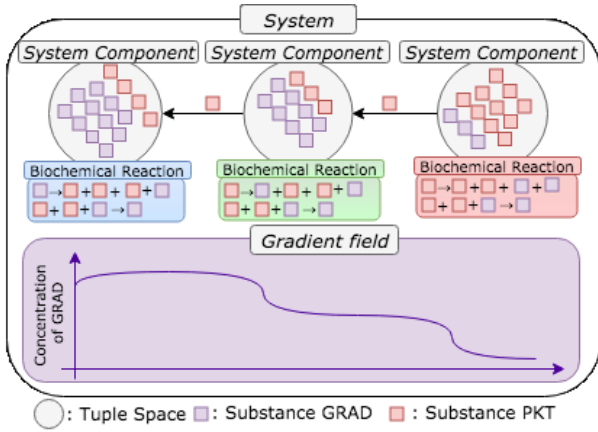


図 3 勾配場によるフローのパケットの移動 [12]

に比例した速度で周囲の接続されたサーバに拡散し、分散実行されることを表している。

#### 2.4 フローの移動

あるサーバに存在する所望の VNF を要求するフローに対して、資源量の枯渇等でそのサーバ上で VNF を実行できない場合には、フローのパケットは対応する VNF が提供されているサーバへ移動することが求められる。また、その移動先は、要求する VNF が提供されているサーバに近づく方向であることが望ましい。このような挙動を実現するために、各サーバにおける VNF の需要とサーバの資源量に基づく勾配場を形成し、フローのパケットが勾配場に従って移動することを目的とし、以下の反応式 (9) - (12) を用いる。

$$VNF_f | RSRC_t \xrightarrow{r_{rg}} VNF_f | RSRC_t | GRAD_f \quad (9)$$

$$GRAD_f \xrightarrow{r_{dg}} 0 \quad (10)$$

$$GRAD_f \xrightarrow{r_{mg}} GRAD_f^+ (GRAD_f^-) \quad (11)$$

$$PKT_c \xrightarrow{r_{mf}} PKT_c^+ (GRAD_f^+ | c) \quad (12)$$

$GRAD$  は、勾配場を形成するための物質である。式 (9) は、 $GRAD$  が  $VNF$  と  $RSRC$  の濃度に応じた速度で生成されることを表している。式 (10) は、 $GRAD$  が一定の速度で消失することを表している。すなわち、 $VNF$  の濃度が小さく、式 (9) による  $GRAD$  の生成が行われないようなダブル空間では、 $GRAD$  の濃度が徐々に小さくなる。式 (11) は、 $GRAD$  がその濃度の小さい周囲の接続されたダブル空間へ拡散することを表している。従って、 $GRAD$  による勾配場は、 $VNF$  と  $RSRC$  の濃度が大きく、 $GRAD$  が大きな速度で生成されるダブル空間を頂上とし、その周囲に向かって裾野が広がるように形成される。式 (12) は、 $PKT$  が  $GRAD$  の濃度の高い周囲の接続されたダブル空間へ移動することを表している。これらの反応式を導入することによって、フローのパケットが、対応する VNF がより多く、かつ資源量がより多く残存しているサーバの方向へ移動する、という挙動が実現される。図 3 に、式 (9) - (12) によって実現される勾配場の形成、及びフローのパケットが移動する様子を示す。

#### 2.5 複数の VNF の共存

1 台のサーバ上に複数の VNF が共存する場合、各 VNF の需要に応じてサーバ資源を割り当てることで、需要に応じた資源共有を行うことが求められる。これを実現するために、上述した化学反応式を VNF 毎に定義する。

#### 2.6 化学物質濃度の更新

反応式における化学物質  $VNF$ ,  $PKT$ ,  $RSRC$ ,  $MEDIATE$ , 及び  $GRAD$  の濃度について、NFV に基づくネットワークシステムでは以下のように設定する。

- $VNF$ : 初期濃度値を設定し、反応式に応じて更新する。
- $PKT$ : サーバに到着するフローのパケット数、及び VNF が適用され、サーバから退出するパケット数に応じて決定される。
- $RSRC$ : サーバにおける CPU 資源量を利用率で表現し、反応式に応じて更新する。
- $MEDIATE$ : 各 VNF に与える CPU 資源量を利用率で表現し、反応式に応じて更新する。
- $GRAD$ : フローのパケットを複数のサーバへ分散させるための指標であり、反応式に応じて更新する。

これらの化学物質濃度の更新は、文献 [16] に示されている、化学反応式モデルの数値計算手法に基づき、一定の時間間隔毎に行う。その間隔で区切られた時間をタイムスロットと定義する。具体的には、各タイムスロットの開始時点で、1 つ前のタイムスロットの間にサーバに到着したパケット数、及びサーバから退出したパケット数に応じて、 $PKT$  の濃度を設定する。次に、タイムスロット内で各化学反応式が実行される回数を反応物の濃度と反応速度係数から算出し、反応式を実行して各化学物質の濃度を更新する。最後に、各 VNF へ割り当てる資源を表す物質である化学物質  $MEDIATE$  の濃度に応じて、各 VNF へ CPU 資源を配分し、フローのパケットを複数のサーバへ分散させる指標となる物質である化学物質  $GRAD$  の濃度に応じて、フローのパケットを複数のサーバへ分散させる。

### 3. NFV フレームワークに基づいた提案手法の実装方針

本章では、NFV フレームワークに基づいた、2 章で説明した手法の実装方針について述べる。

#### 3.1 NFV フレームワークにおける提案手法の位置付け

図 4 に、NFV の標準化を進めている European Telecommunications Standards Institute (ETSI) の配下に設立された Industry Specification Group (ISG) が文献 [13] で提示している NFV フレームワークに、提案手法の実現に必要な機能を配置した図を示す。提案手法を実現する機能を Biochemical Reaction Manager (BRM) と定義する。また、表 1, 2 にそれぞれ、図 4 における主要な機能である NFV Architectural Layers と NFV Management and Orchestration (NFV MANO) の役割と、機能間を接続するインタフェースの説明を示す。表 1 では、NFV Infrastructure, Virtualized Infrastructure Manager, VNF Manager, 及び NFV Orchestrator をそれぞれ、NFVI, VIM, VNFM, 及び NFVO と省略して表記している。

NFV フレームワーク [13] には、仮想化されたネットワーク機能である VNF, VNF が導入されるリソースに相当する NFVI, 及び NFVI と VNF の管理・調整を行う NFV MANO の 3 つの主要な機能が存在する。NFV フレームワークにおいて提案手法を実現するために、以下のように既存の機能における役割を拡張し、新たな機能・役割を追加する。

- サービスチェイニング要求を実現するために、VNF に NSH の更新機能を追加する。NSH の詳細は 3.3 節で説明する。
- フローの経路制御を実現するために、SDN Switch 及び SDN Controller をそれぞれ、NFV Architectural Layers 及び NFV MANO に配置する。また、サービスチェイニング要求を実現するために、SDN Controller に NSH の挿入、更新、削除等の設定を実行する機能、及び SDN Switch に NSH の挿入、更新、削除等を実行する機能を追加する。NSH の詳細は 3.3 節で説明する。
- 負荷に応じた VNF への資源の割り当てや、VNF の適応的な負荷分散、VNF の拡散及び集約等を実現するために、提案手法を実現する機能である BRM を NFV MANO に配置する。BRM は、ネットワーク環境に応じてダブル空間を定義し、反応式の実行結果に基づいて、様々な挙動を実現する。

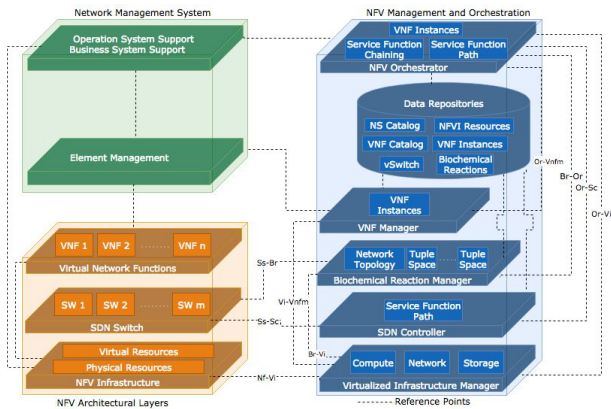


図 4 NFV フレームワーク [13] の拡張

機能名	概要・役割
NFVI	VNF が導入されるリソースであり、ハードウェア資源とソフトウェア資源から構成される
VIM	NFVI として用いられる計算資源、ネットワーク資源、及びストレージの構成情報を管理し、仮想マシンへの各資源の割り当てや、資源の再利用等を実行する
VNFM	VNF のライフサイクル管理やヘルスマonitoring、Auto Healing/Auto Scaling 等を実行する
NFVO	VNF のテンプレート化やサービスチェイニング要求の管理、VNF のオーケストレーション等を実行する
VNF	パケットに対するネットワーク機能の適用や、NSH の更新等を実行する
SDN Switch	フローエントリに従ったパケットの転送や、NSH の挿入、更新、削除等を実行する
SDN Controller	SDN Switch に対するフローエントリの設定や、NSH の挿入、更新、削除等の設定を実行する
BRM	ネットワークトポロジの管理や、トポロジ情報に基づいたタプル空間の定義、反応式の実行等を実現する
Data Repositories	NFVI, VNF, フローエントリ、反応式等の情報を保持するデータベースである

表 1 各機能の役割

本研究では、NFV を実現するための OSS として、OPNFV [14] を用いる。OPNFV は、Linux Foundation によって設立され、既存の OSS を統合することによって、NFV フレームワーク全体を実装することを目的としており、OpenStack [17], OpenDaylight [18], Kernel-Based Virtual Machine (KVM) [19] などが用いられている。また、OPNFV においては、SDN を実現するための通信プロトコルとして、OpenFlow を用いることが想定されているため、本報告では、OpenFlow を用いた提案手法の実装方針を検討する。

### 3.2 提案手法の実現

図 5 に、OPNFV を用いたシステム構成と、図 4 における各機能の配置を示す。図 5 では、簡単のために、1 台の物理マシン上に図 4 の環境を実装することを想定しており、複数の仮想マシンと仮想スイッチが存在する。仮想マシンは、NFV 環境をインストールする役割を持つ Jump Server、NFV MANO の役割を持つ Controller、及び NFV Architectural Layers の役割を持つ Compute の 3 種類が存在する。また、OPNFV のインストール時に、ノードのプロビジョニングのために用いられる Admin、テナント用トラフィックのために用いられる Tenant、外部ネットワークに接続するために用いられる Public、ストレージの I/O 処理のために用いられる Storage の 4 種類のネットワークが存在する。

表 3 に、各機能の図 5 の示すシステム上での実現方法の概要を示す。NFV MANO 及び NFV Architectural Layers の機能

インタフェース名	用途
Or-Vnfm	NFVO が、VNFM から必要な VNF の情報を取得する等で用いられる
Vi-Vnfm	仮想マシンの状態の情報交換等で用いられる
Or-Vi	仮想マシンの状態の情報交換等で用いられる
Or-Sc	SDN Controller が、NFVO からフローの経路情報を取得する等で用いられる
Ss-Br	BRM が、SDN Switch からフローレートの情報を取得する等で用いられる
Br-Vi	VIM が、BRM から <i>MEDIATE</i> の濃度を取得する等で用いられる
Nf-Vi	VIM が、NFVI へ資源を割り当てる等で用いられる
Br-Or	NFVO が、BRM から <i>GRAD</i> , <i>VNF</i> の濃度を取得する等で用いられる
Ss-Sc	SDN Controller が、SDN Switch へフローエントリを設定する等で用いられる

表 2 各インタフェースの用途

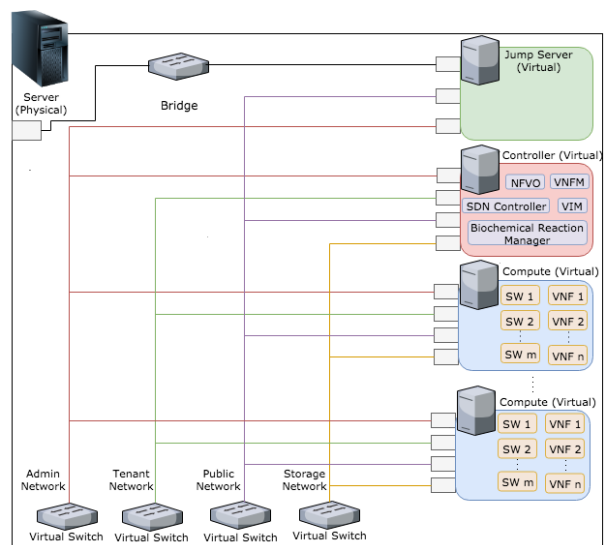


図 5 OPNFV を用いたシステム構成図

はそれぞれ Controller 及び Compute 上に配置し、OPNFV に統合されている既存の OSS や各ベンダーが提供する OSS を用いて実装する。

提案手法を実現するプログラムは、Controller 上で 1 つのプロセスとして動作させる。プログラムは、ネットワーク環境に応じてタプル空間を定義し、反応式を独立に実行する。Compute 上の各 SDN Switch に到着したフローに対して、*GRAD* の濃度に応じてフローの経路を決定し、SDN Controller を通じて、SDN Switch に経路設定を行う。また、SDN Switch に到着するフローレートの情報は BRM に伝達され、反応式における *PKT* の濃度に反映される。VNF への資源割り当ては、*MEDIATE* の濃度に応じて、VIM によって実行される。さらに、*VNF* の濃度に応じて、NFVO を通じて、VNF の起動、及び停止を実行する。

### 3.3 サービスチェイニング要求の実現

IETF が提案している Network Service Header (NSH) [15] の概要と、NSH を用いたサービスチェイニング要求の実現方法について説明する。

#### 3.3.1 Network Service Header

NSH は、サービスチェイニング要求において、所望の順序で VNF を適用するために必要なフローの経路情報、及び経路上の位置情報をパケットのヘッダに格納する。図 6 に、NSH のフォーマットを示す。NSH は、基本情報やペイロードのプロトコルに関する情報を格納する Base Header、フローの経路情報と経路

機能名	実現方法
NFVI	KVMを用いて, Compute上で実現する
SDN Switch	Open vSwitchを用いて, Compute上で実現する
VNF	各ベンダーが提供するOSSを用いて, Compute上の仮想マシンで実現する
VIM	OpenStackを用いて, Controller上で実現する
VNFM	OpenStack Tackerを用いて, Controller上で実現する
NFVO	OpenStack Tackerを用いて, Controller上で実現する
BRM	提案手法のプログラムを用いて, Controller上で実現する
SDN Controller	OpenDaylightを用いて, Controller上で実現する

表3 各機能の実現方法

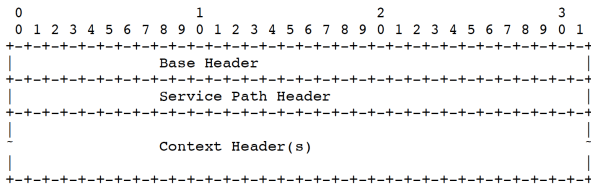


図6 Network Service Header (NSH) [15]

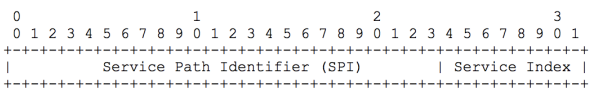


図7 NSH Service Path Header [15]

機能名	役割	実現方法
SFP	SFCの具体的なSFの経路の保持(例: DC1のDPI1 → DC3のNAT2)	NFVO上で実現する
SC	各フローのSFPの決定	NFVO上で実現する
	パケットへのNSHの付与	SDN Controller上で実現する

表4 NSHの実現に必要な機能とその実現方法

上の位置情報を格納する Service Path Header, 及びメタデータを格納する Context Header(s) で構成される。図7に, サービスチェイニング要求の実現に必要な情報を格納する NSH Service Path Header の構成を示す。NSH Service Path Header は, フローの経路を一意に識別する Service Path Identifier (SPI) と経路上の位置情報を示す Service Index (SI) で構成される。

### 3.3.2 NSHの実現

IPパケットにおけるNSHの実現方法の1つとして, OptionフィールドにNSHを格納することが考えられる。しかし, 現状のOpenFlowの仕様では, Optionフィールドに基づいた制御を行うことはできない。従って, 本研究においては, IPヘッダのType Of Service (TOS) フィールドにNSHを格納する。TOSフィールドは8 [bit] であるため, SIのみを格納する。SIには, 初期値としてフローが通過するVNFの数を設定する。フローにVNFを適用する度にSIの値を1減らし, SIが0である場合には, サービスチェイニング要求における全てのVNFが適用されたと判断する。SPIは, フローに経路を割り当てるために必要となるフロー識別子であるが, 本研究においては, 5-tuple (送信元IPアドレス, 送信先IPアドレス, 送信元ポート番号, 送信先ポート番号, プロトコル) を用いる。

表4に, NSHの実現に必要な機能とその実現方法の概要を示す。表4では, データセンタ, Service Function Path, 及びService Classifierをそれぞれ, DC, SFP, 及びSCと省略して表記している。

表4の機能を用いたフローの処理方法について説明する。SDN Switchに, 到着したフローを処理するフローエントリが設定されていない場合は, 以下のように動作する。

(1) SDN Controllerは, フローの経路を決定するため

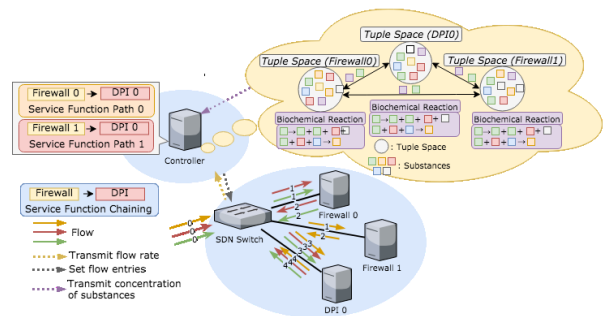


図8 提案手法に基づいたサービスチェイニング要求の実現

に, NFVOに対してSFPを要求する。

(2) NFVOは, GRADの濃度に基づいてSFPを選択し, SDN Controllerへ伝達する。

(3) SDN Controllerは, NFVOから受け取ったSFPに基づいて, SDN Switchにフローエントリの設定, 及びNSHを挿入, 更新, 及び削除する設定を行う。

(4) SDN Switchは, パケットをSDN Switch, あるいはVNFに転送する。

(5) パケットがSDN Switchに到着した場合は(4), VNFに到着した場合は(6)に移移する。

(6) VNFを適用した後, SIの値を1減らし, SDN Switchへパケットを送出する。

(7) SIが0でない場合は(4), 0である場合は(8)に移移する。

(8) サービスチェイニング要求を終了し, SDN SwitchにおいてNSHを削除する。

また, SDN Switchに, 到着したフローを処理するフローエントリが設定されている場合は, (4)の処理から実行する。

### 3.3.3 確率的なフロー経路の決定

一般的に, NFVでは, フロー単位で経路が設定される。一方で, 2章で説明した提案手法は, パケット単位で確率的に挙動が決まり, 経路が決定する。そこで本研究では, 以下のように, 提案手法におけるパケット単位での確率的なフロー経路の決定を, フロー単位で確率的に経路を決定することで近似的に実現する。

図8に, サービスチェイニング要求を持ったフローがSDN Switchに到着する様子を示す。図中の実践矢印及び数字は, フローの経路と通過順を表している。ここでは, フローはファイアウォール処理の後にDPI処理を行うサービスチェイニング要求を持つものとする。ファイアウォールを実現するVNFとしてFirewall 0とFirewall 1の2つが存在し, DPIを実現するVNFとしてDPI 0が存在するため, Controller上のNFVOは, 以下のような2つのSFPを管理している。

- SFP 0: {Firewall 0 → DPI 0}
- SFP 1: {Firewall 1 → DPI 0}

SDN Switchにフローが到着すると, 提案手法のプログラムの実行結果に基づいて, フローに割り当てるSFPを確率的に決定する。具体的には, ファイアウォール処理のためにFirewall 0あるいはFirewall 1のいずれかを用いるかを, それぞれに対応するGRADの濃度を用いて確率的に決定する。例えば, Firewall 0, Firewall 1に対応するGRADの濃度がそれぞれ, 2,000, 1,000である場合には, 到着したフローに対して, 0.67の確率でSFP 0, 0.33の確率でSFP 1をそれぞれ割り当てる。

### 3.4 アプリケーションシナリオ

提案手法の有効性を検証するために, 動画アプリケーションにおけるVNF制御を行うための実装方針を説明する。図9に, 動画配信を提供するStreaming Serverから, ファイアウォール処理の後に, 動画のトランスコーディング処理を行うサービスチェイニング要求を持つフローが到着し, それぞれの処理を受

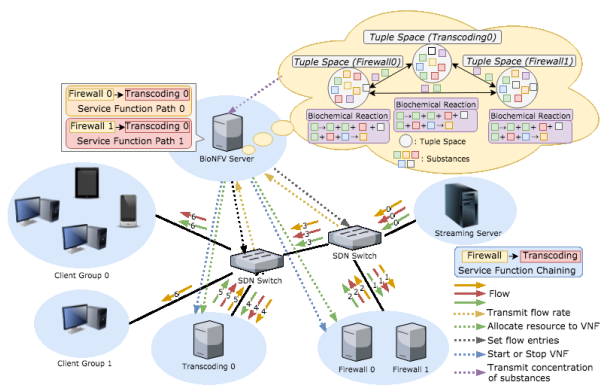


図9 動画アプリケーションにおける提案手法の実現

けた後に Client に動画の送信が行われる様子を示す。図においては、図5における Controller を実現するサーバを BioNFV Server としている。図中の実践矢印及び数字は、フローの経路と通過順を表している。本シナリオを通して、提案手法における以下の機能が実現されることを確認する。

- 負荷に応じた VNF への資源の割り当て
- VNF の適応的な負荷分散
- VNF の拡散、及び集約

図9では、ファイアーウォール処理を実現する VNF として Firewall 0 と Firewall 1 の2つが存在し、動画のトランスコーディング処理を行う VNF として Transcoding 0 が存在するため、以下の2つの SFP が管理されている。

- SFP 0: {Firewall 0 → Transcoding 0}
- SFP 1: {Firewall 1 → Transcoding 0}

BioNFV Server では、ネットワーク環境に応じてタプル空間が定義され、反応式が実行される。タプル空間においては、到着するフローレートに応じて、PKT の濃度が更新される。反応式の実行の結果、MEDIATE の濃度に応じて、Firewall 0, Firewall 1, 及び Transcoding 0 に資源割り当てを実行することによって、負荷に応じた VNF への資源の割り当てが実現される。VNF の濃度に応じて、Firewall 0, Firewall 1, 及び Transcoding 0 の起動、及び停止を実行することで、VNF の拡散、及び集約が実現される。

フローが到着すると、BioNFV Server は、GRAD の濃度に応じて、フローに対して適用する SFP を確率的に決定し、それに基づいて SDN Switch にフローエントリを設定する。これにより、VNF の適応的な負荷分散が実現される。

到着するフローが減少すると、反応式の実行の結果、VNF の濃度が小さくなる。その場合には、対応する VNF を停止し、停止した VNF を用いる SFP をフローに割り当てないようにする。このようにして、必要に応じた VNF の集約が実現される。

一方、到着するフローが増加すると、反応式の実行の結果、VNF の濃度も大きくなる。その場合には、対応する VNF を起動し、起動した VNF を用いる SFP がフローに割り当てられるようにする。このようにして、必要に応じた VNF の拡散が実現される。

#### 4. まとめと今後の課題

本報告においては、生化学反応モデルに基づくサービス空間構築手法を、NFV 環境に適用するために、NFV フレームワークに基づいた提案手法の実装方針を検討した。具体的には、まず、化学反応式を構成する物質や反応式を、NFV 環境の構成要素や挙動に対応させた。次に、NFV フレームワークに基づいた提案手法、及び NSH を用いたサービスチェイニング要求の実装方針に関して説明した。また、アプリケーションの実装方針について述べた。

今後の課題として、本報告で述べた生化学反応モデルに着想を得た VNF 制御手法の実装、及び実験評価が挙げられる。また、文献 [12] で示されている提案手法において、サーバ間の伝搬遅延時間やリンク帯域を考慮した評価を行うことも重要であると考えられる。

#### 文 献

- [1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Communications Surveys Tutorials*, vol. 17, no. 4, pp. 2347–2376, Fourthquarter. 2015.
- [2] "近未来における ICT サービスの諸課題" available at [http://www.soumu.go.jp/main\\_content/000368347.pdf](http://www.soumu.go.jp/main_content/000368347.pdf).
- [3] ETSI, "Network Function Virtualisation - White Paper 1." available at [https://portal.etsi.org/nfv/nfv\\_white\\_paper\\_1.pdf](https://portal.etsi.org/nfv/nfv_white_paper_1.pdf).
- [4] Ingham, Kenneth and Forrest, Stephanie, "A History and Survey of Network Firewalls," *University of New Mexico, Tech. Rep*, Jan. 2002.
- [5] M. Finsterbusch, C. Richter, E. Rocha, J. A. Muller, and K. Hanssgen, "A Survey of Payload-Based Traffic Classification Approaches," *IEEE Communications Surveys Tutorials*, vol. 16, no. 2, pp. 1135–1156, Secondquarter. 2014.
- [6] A. Gonzalez, P. Gronlund, K. Mahmood, B. Helvik, P. Heegaard, and G. Nencioni, "Service Availability in the NFV Virtualized Evolved Packet Core," in *2015 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, Dec. 2015.
- [7] ETSI, "Network Function Virtualisation - White Paper 2." available at [https://portal.etsi.org/nfv/nfv\\_white\\_paper2.pdf](https://portal.etsi.org/nfv/nfv_white_paper2.pdf).
- [8] X. Li and C. Qian, "A Survey of Network Function Placement," in *2016 13th IEEE Annual Consumer Communications Networking Conference (CCNC)*, pp. 948–953, Jan. 2016.
- [9] J. G. Herrera and J. F. Botero, "Resource Allocation in NFV: A Comprehensive Survey," *IEEE Transactions on Network and Service Management*, vol. 13, no. 3, pp. 518–532, Sep. 2016.
- [10] M. Viroli, M. Casadei, S. Montagna, and F. Zambonelli, "Spatial Coordination of Pervasive Services Through Chemical-Inspired Tuple Spaces," *ACM Trans. Auton. Adapt. Syst.*, vol. 6, no. 2, pp. 1–24, June. 2011.
- [11] G. Hasegawa and M. Murata, "Biochemically-inspired Method for Constructing Service Space in Virtualized Network System," in *Proceedings of ICIN 2016*, Mar. 2016.
- [12] 坂田 航樹, 長谷川 剛, 村田 正幸, "生化学反応モデルに着想を得た仮想ネットワーク機能の配置手法の提案と評価," *電子情報通信学会技術研究報告*, vol. 115, no. 507, pp. 25–30, 2016年3月.
- [13] ETSI GS NFV 002, "Network Functions Virtualisation (NFV); Architectural Framework." available at [http://www.etsi.org/deliver/etsi\\_gs/NFV/001\\_099/002/01.02.01\\_60/gs\\_NFV002v010201p.pdf](http://www.etsi.org/deliver/etsi_gs/NFV/001_099/002/01.02.01_60/gs_NFV002v010201p.pdf).
- [14] "Home - OPNFV." available at <https://www.opnfv.org/>.
- [15] "Network Service Header (NSH)." available at <https://www.rfc-editor.org/rfc/pdf/rfc/rfc8300.txt.pdf>.
- [16] H. Li, Y. Cao, L. R. Petzold, and D. T. Gillespie, "Algorithms and Software for Stochastic Simulation of Biochemical Reacting Systems," Feb. 2007.
- [17] "Open source software for creating private and public clouds." available at <https://www.openstack.org/>.
- [18] "Home - OpenDaylight." available at <https://www.opendaylight.org/>.
- [19] "KVM." available at [https://www.linux-kvm.org/page/Main\\_Page](https://www.linux-kvm.org/page/Main_Page).