

卒業論文 2023 年度（令和 5 年度）

# Linux SRv6 ルーティングインフラにおける SR-Aware ネットワーク機能としての Netfilter の統合

慶應義塾大学 環境情報学部  
澤田 開杜

Linux SRv6 ルーティングインフラにおける SR-Aware ネットワーク機能としての Netfilter の統合
---

本論文では、End.AN.NF という新しい Segment Routing over IPv6 (SRv6) End behavior を提案する。End.AN.NF は Linux netfilter を Linux の SRv6 ルーティングインフラストラクチャへ統合することができる。セグメントルーティング (SR) を利用したサービスファンクションチェイニング環境において、End.AN.NF は、Linux netfilter を SR に対応したネットワークファンクションとして扱えるようにする。End.AN.NF を利用する際、netfilter を利用して作成されたアプリケーションの実装を変える必要はなく、End.AN.NF は SRv6 の基本処理である End behavior を実行しながら、SRv6 でカプセル化された内部のパケットへ netfilter を適用できる。さらに、End.AN.NF は segment id 内の引数部分を利用して、パケットにマークを付けることができる。したがって、netfilter を利用して作成されたアプリケーションは End.AN.NF がパケットバッファに付与したマークを照合することで、適用するルールを変更できる。我々は End.AN.NF を Linux kernel に実装し、その性能評価を行った。計測の結果、End.AN.NF は End.DT4 と H.Encaps を使って SRv6 でカプセル化された内部パケットに netfilter を適用する方法に比べ、27% 高いスループット、及び 3.0 マイクロ秒低いレイテンシを実現した。

キーワード:

1. Service Function Chaining 2. Segment Routing 3. SRv6

慶應義塾大学 環境情報学部  
澤田 開杜

Integrating Netfilter into SRv6 Routing Infrastructure of Linux as an SR-Aware Network Function
---

This paper proposes a new SRv6 End behavior, called End.AN.NF, integrating Linux netfilter as a network function for service function chaining by Segment Routing (SR). End.AN.NF allows netfilter-based applications to be executed as SR-Aware applications without modification, as it applies netfilter to inner packets encapsulated in SRv6 while performing the basic SRv6 End behavior. Furthermore, End.AN.NF utilizes the argument of the segment identifiers to mark packets. Consequently, this enables netfilter-based applications to match the marks on packet buffers and change rules to be applied. We implemented End.AN.NF on the Linux kernel and evaluated its performance. The evaluation shows that End.AN.NF achieves 27% higher throughput and 3.0 microseconds lower latency than applying netfilter to SRv6-encapsulated inner packets by End.DT4 and H.Encaps.

Keywords :

1. Service Function Chaining 2. Segment Routing 3. SRv6

Keio University Bachelor of Arts in Environment and Information Studies  
Kaito Sawada

# 目 次

第 1 章	序論	1
1.1	背景 . . . . .	1
1.2	本論文の目的と構成 . . . . .	2
第 2 章	背景と先行研究	3
第 3 章	設計と実装	4
第 4 章	評価	5
第 5 章	結論	6

# 图 目 次

# 表 目 次

# 第1章 序論

## 1.1 背景

Service Function Chaining (SFC) は, Software Defined Network (SDN) および Network Function Virtualization (NFV) の文脈で研究されているトピックである [1, 2, 3, 4]. SFC では, ネットワーク機能 (NF) を通過する順序や NF のタイプに関する情報を事前に定義し, それらのルールをネットワーク機器に配布する必要がある. SFC ネットワークを構築するネットワーク機器は, 事前に決定されたルールに従って受信したパケットを NF に導く. パケットを NF へ導くためのルールは, SDN コントローラやルーティングプロトコルによってネットワーク機器に配布される. ネットワーク機器は IP ルーティング上の最短経路に関係なく, 配布された SFC ルールに従ってパケットを転送する次のホップを選択する必要がある. また, パケットのヘッダにこれらのルールに合致させるための特別な情報を埋め込む手法が取られることもある. SFC は, クラウドサービスプロバイダ (CSP), アプリケーションサービスプロバイダ (ASP) 及びインターネットサービスプロバイダ (ISP) にとって, 現在の静的な環境に代わる柔軟かつ経済的な選択肢を提供する [5].

SFC を実現可能な技術には, いくつかの候補が存在する. 例えば OpenFlow [6], Network Service Header (NSH) [7], MPLS [8] などである. これらの技術はどれも, 最短経路に関係なく, ルールに基づいて受信したパケットを意図した NF に導く, という要件を満たすことができる. OpenFlow では, 経路情報を管理する中央のコントローラが, 実際にパケットを転送する OpenFlow スイッチに対して明示的にパケット転送ルールを設定する. OpenFlow スイッチは, コントローラによって適切に管理されたルールに従い, パケットを意図した NF に転送する. OpenFlow のもつこのアーキテクチャは, 従来のルーティングプロトコルに基づかない柔軟な経路制御を可能にする. NSH は Service Path Identifier (SPI) と Service Index (SI) によって NF を識別する. NSH ノードは, パケットに付与された NSH 内の SPI, SI に基づいてパケットを転送する. NSH は, サービスプレーンと呼ばれる専用のオーバーレイネットワークを作成し, そのオーバーレイネットワーク内でサービスを転送する. このオーバーレイネットワークを構築する, というアーキテクチャにより, NSH では基礎となるネットワークトポロジを変更することなくサービス転送を可能にする. 一方, MPLS では, 直接 NSH を使用する代わりに, MPLS ラベルスタックを利用する. このラベルスタックには, パケットが通過すべきノードの順序がホップバイホップで含まれている. ラベルスタック内で表現されるノードはルータだけでなく, NF も含まれるため, そのラベルスタックに基づいてパケットを転送する事で SFC を実現できる. このアプローチもまた, 基礎となるネットワークトポロジを変更せずに SFC を実現するために必要な, 最短経路によらないパケット転送を達成する.

## 1.2 本論文の目的と構成

本論文における以降の構成は次の通りである．2 章では，サービスファンクションチェーンニングに関する前提知識，及びそれを実現する技術について解説し，本論文の概要について述べる．3 章では，本論文の提案する新たな SRv6 End behavior である End.AN.NF についての詳細な動作，及び実装について述べる．4 章では，実装した End.AN.NF について，レイテンシ及びスループットの性能を特定のを変化させながら性能の計測する．5 章では，本研究における結論と今後の展望について述べ，End.AN.NF に必要なネットワーク制御プレーンについて検討する．



## 第2章 背景と先行研究

## 第3章 設計と実装

## 第4章 評価

## 第5章 結論

## 参考文献

- [1] Bo Han, Vijay Gopalakrishnan, Lusheng Ji, and Seungjoon Lee. Network function virtualization: Challenges and opportunities for innovations. *IEEE Communications Magazine*, 53(2):90–97, 2015.
- [2] Karamjeet Kaur, Veenu Mangat, and Krishan Kumar. A comprehensive survey of service function chain provisioning approaches in sdn and nfv architecture. *Computer Science Review*, 38:100298, 2020.
- [3] Irena Trajkovska, Michail-Alexandros Kourtis, Christos Sakkas, Denis Baudinot, João Silva, Piyush Harsh, George Xylouris, Thomas Michael Bohnert, and Harilaos Koumaras. Sdn-based service function chaining mechanism and service prototype implementation in nfv scenario. *Computer Standards & Interfaces*, 54:247–265, 2017. SI: Standardization SDN&NFV.
- [4] Gianluca Davoli, Walter Cerroni, Chiara Contoli, Francesco Foresta, and Franco Callegati. Implementation of service function chaining control plane through openflow. In *2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, pages 1–4, 2017.
- [5] Deval Bhamare, Raj Jain, Mohammed Samaka, and Aiman Erbad. A survey on service function chaining. *Journal of Network and Computer Applications*, 75:138–155, 2016.
- [6] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. Openflow: Enabling innovation in campus networks. *SIGCOMM Comput. Commun. Rev.*, 38(2):69–74, mar 2008.
- [7] Paul Quinn, Uri Elzur, and Carlos Pignataro. Network Service Header (NSH). RFC 8300, January 2018.
- [8] Adrian Farrel, Stewart Bryant, and John Drake. An MPLS-Based Forwarding Plane for Service Function Chaining. RFC 8595, June 2019.