

# アプリケーション識別型トラフィック制御システムの開発

式町 龍聖<sup>1</sup> 神屋 郁子<sup>2</sup> 下川 俊彦<sup>1</sup>

**概要:** 近年、コンテンツ配信サービスの増加とリモートワークの普及により、ネットワークトラフィックが増加し、アプリケーションごとのネットワーク要件も多様化している。しかし、既存のルーティング手法ではアプリケーションごとの要件を考慮した経路選択が行われなため、ネットワーク性能や QoE の低下が問題となっている。本研究では、DNS によるアプリケーション識別と SRv6 による経路制御を組み合わせた、アプリケーション識別型トラフィック制御システムを開発した。評価実験では、低遅延・帯域幅・両方を要求する 3 種類のアプリケーションについて ECMP と比較した。その結果、提案手法は全アプリケーションで要件に応じた経路選択が可能であり、ECMP で生じていたクライアント間の QoE 不均衡を解消できることを確認した。

**キーワード:** トラフィックエンジニアリング, DNS, SRv6, QoS

## Development of Application-Aware Traffic Engineering System

RYUSEI SHIKIMACHI<sup>1</sup> YUKO KAMIYA<sup>2</sup> TOSHIHIKO SHIMOKAWA<sup>1</sup>

**Abstract:** In recent years, the growth of content delivery services and the spread of remote work have increased network traffic, diversifying the network requirements of individual applications. However, conventional routing methods do not consider per-application requirements, leading to degraded network performance and Quality of Experience (QoE). In this study, we developed an application-aware traffic control system that combines DNS-based application identification with SRv6-based path control. In our evaluation, we compared the proposed method with ECMP for three types of applications requiring low latency, high bandwidth, and both. The results confirmed that the proposed method can select paths according to application requirements and eliminates the QoE imbalance among clients that occurred with ECMP.

**Keywords:** Traffic Engineering, DNS, SRv6, QoS

### 1. はじめに

近年、動画配信やオンラインゲームなどのコンテンツ配信サービスが増加し、リモートワークの普及も相まってネットワークトラフィックが増加している。総務省の報道資料 [1] によると、2024 年 5 月から 2025 年 5 月の 1 年間で、ダウンロードトラフィックが 15.3%、アップロードトラフィックが 14.3%増加している。トラフィックの増加

は、パケットロスの発生や転送速度の低下を招き、ネットワーク性能の低下につながる。

また、アプリケーションの多様化に伴い、許容遅延や必要帯域幅などのネットワーク要件も多様化している。例えば、Netflix[2] では 4K 動画の場合 15Mbps 以上の帯域幅が必要とされている。また、音声通話は ITU-T 勧告 G.114[3] において、片方向遅延 150ms 以下が会話品質の許容範囲として推奨されている。このようにアプリケーションごとに求められるネットワーク要件は異なる。

しかし、インターネットにおけるルーティングは、宛先アドレスや固定的なメトリックに基づいて経路を決定し、アプリケーション種別や要件を考慮しない。そこで本研

<sup>1</sup> 九州産業大学大学院 情報科学研究科  
Graduate School of Information Science, Kyushu Sangyo University

<sup>2</sup> 福岡女子大学  
Fukuoka Women's University

究では、アプリケーション利用者の体感品質 (QoE) 向上を目的として、DNS を利用したアプリケーション識別と、SRv6 を利用した経路制御を組み合わせた、アプリケーション識別型トラフィック制御システムを開発する。

## 2. トラフィック制御技術

### 2.1 セグメントルーティング

セグメントルーティング (Segment Routing) は、パケットが経由する経路をあらかじめ指定することで、柔軟な経路制御を可能にするルーティング方式である [4]。セグメントルーティングでは、特定の動作や位置をセグメント (Segment) として表現する。セグメントは、ルータやリンクなどのネットワーク要素を識別するための単位であり、セグメント識別子 (SID) によって一意に表される。複数の SID を順序付けたセグメントリストをパケットに付加することで、送信元がパケットの通過経路を指定できる。

### 2.2 SRv6

SRv6 (Segment Routing over IPv6) は、IPv6 ネットワーク上でセグメントルーティングを実現する技術である [5]。SRv6 では、SID を IPv6 アドレス形式で表し、SRH (Segment Routing Header) [6] としてパケットに付加する。SRH には、複数の SID を保持するセグメントリストと、未処理の SID 数を示す SL (Segment Left) などが格納される。

SRv6 では、パケット処理に関与するルータの役割を 3 種類に分類する。Source Node は SRH を付加してパケットを送信するルータである。Transit Node は SRH を保持したまま通常の IPv6 ルーティングで中継するルータである。Endpoint Node は現在の SID を自身の SID として認識し、対応する機能 (Function) を実行するルータである。主な SID 機能として以下がある。

- End: 次の SID を宛先アドレスへ書き換えて転送する
- End.X: 特定の隣接リンクに転送する
- End.DT4/DT6: SRH を外して IPv4/IPv6 ルーティングを行う

SRv6 では、Source Node でセグメントリストを設定することで、任意の経路を指定したパケット転送が実現できる。本研究では、アプリケーションのネットワーク要件に基づいてセグメントリストを動的に生成し、経路制御を行う。

## 3. アプリケーション識別

### 3.1 DNS を用いた識別方式

本研究では、DNS を用いたアプリケーション識別方式を採用する。クライアントがサーバに接続する際、まず DNS サーバに問い合わせでドメイン名から IP アドレスを取得

する。この DNS クエリに含まれるドメイン名は、通信先のアプリケーションを識別する手がかりとなる。

従来のアプリケーション識別手法には課題がある。ポート番号に基づく識別は、多くのアプリケーションが HTTPS (TCP/443) や QUIC (UDP/443) を使用するようになり、識別が困難になっている。Deep Packet Inspection (DPI) は、TLS や QUIC による暗号化の普及により適用範囲が限定されている。DNS を用いた識別方式は、これらの課題を回避できる。

### 3.2 e2LD による識別

具体的には、ドメイン名から e2LD (effective 2nd-level domain) を抽出し、アプリケーションの識別に用いる。e2LD とは、パブリックサフィックス (例: .com, .co.jp) とその直下の 1 ラベルを組み合わせたドメイン名を指す。パブリックサフィックスの一覧は Public Suffix List (PSL) [7] として公開されており、本研究ではこのリストを参照して e2LD を抽出する。

e2LD を用いることで、1つのアプリケーションが多数のサブドメインを使用する場合でも、少数のエントリで識別できる。例えば、Zoom は docs.zoom.us, us04web.zoom.us, log-gateway.zoom.us など多数のサブドメインを使用するが、e2LD を用いることで zoom.us の 1 エントリで識別できる。

## 4. システム設計

### 4.1 対象とするアプリケーション

本システムでは、ネットワーク要件の観点から以下の 3 種類のアプリケーションを対象とする。

- 遅延の小ささを要求するアプリケーション (例: VoIP, Web 会議)
- 帯域幅の広さを要求するアプリケーション (例: 動画ストリーミング)
- 遅延の小ささと帯域幅の両方を要求するアプリケーション (例: クラウドゲーミング)

本システムでは、アプリケーションのネットワーク要件として許容遅延および必要帯域幅を定義し、これらの要件とネットワーク状態に基づいて経路を選択する。

### 4.2 システム構成

本システムは、アプリケーション識別サブシステム、モニタリングサブシステム、経路選択サブシステムの 3 つから構成される (図 1)。

アプリケーション識別サブシステムは、DNS サーバから DNS ログを受信し、ドメイン名から e2LD を抽出してアプリケーションを識別する。識別結果は経路選択サブシステム

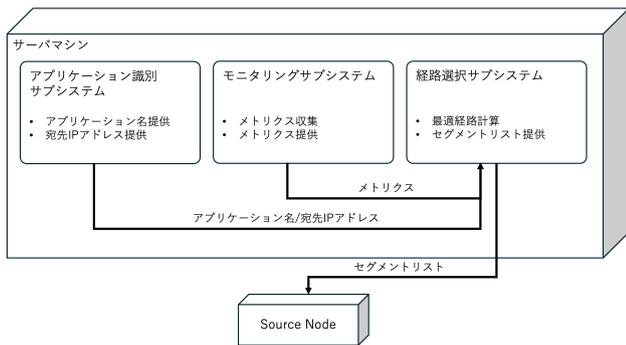


図 1 システム構成

ムに提供される。

モニタリングサブシステムは、ネットワークのトポロジ情報とリンクメトリクス（遅延、スループット）を収集する。収集した情報は経路選択サブシステムに提供される。

経路選択サブシステムは、アプリケーションのネットワーク要件とリンクメトリクスに基づいてリンクコストを決定し、最適経路を算出する。算出した経路はセグメントリストとして Source Node に設定される。

#### 4.3 リンクコスト決定

リンクコストは、アプリケーションのネットワーク要件に応じて以下のように決定する。

遅延の小ささを要求するアプリケーションでは、リンクの遅延をそのままリンクコストとして用いる。これにより、遅延が小さいリンクを優先的に選択できる。

帯域幅の広さを要求するアプリケーションでは、帯域幅使用率をリンクコストとして用いる。リンク容量を  $C$ 、現在のスループットを  $T$ 、アプリケーションが要求する帯域幅を  $B_{req}$  とすると、リンクコスト  $w_{bw}$  は式 (1) で表される。

$$w_{bw} = \frac{T + B_{req}}{C} \quad (1)$$

遅延の小ささと帯域幅の両方を要求するアプリケーションでは、まずリンクの遅延が許容遅延以内であるかを評価し、許容遅延を超えるリンクは除外する。次に、残ったリンクに対して帯域幅使用率をリンクコストとして用いる。

#### 4.4 最適経路算出

経路算出では、始点から終点までの経路のうち、リンクコストの総和が最小となる最短経路を Dijkstra アルゴリズムで求める。アプリケーションごとにネットワーク要件が異なるため、同じリンクメトリクスであっても、算出される経路がアプリケーションごとに異なる。

### 5. システム実装

本システムは Rust 言語で実装した。各サブシステムは Docker コンテナとして独立して動作し、Docker Compose

で管理される。

アプリケーション識別サブシステムでは、dnstap[8] と dnscollector[9] で DNS ログを受信し、publicsuffix クレートで e2LD を抽出する。アプリケーションと e2LD の対応表は YAML ファイルで管理され、ホットリロードに対応している。

モニタリングサブシステムでは、各ルータ上の Telegraf[10] からメトリクスを収集する。収集するメトリクスは、LLDP 隣接情報、インターフェースの送信バイト数、隣接ルータへの ping 応答時間である。

経路選択サブシステムでは、Dijkstra アルゴリズムで最短経路を算出し、Source Node 上の経路制御プロセスに送信する。経路制御プロセスは ip route コマンドで SRv6 経路をカーネルのルーティングテーブルに登録する。

## 6. 評価

### 6.1 評価概要

本システムがアプリケーションのネットワーク要件に基づいた経路選択ができるか、またその経路選択が QoE に与える影響を評価する。比較対象として、フローのハッシュ値に基づいて経路を選択する ECMP[11] を用いる。

QoE 指標として、低遅延を要求するアプリケーションには E-model[12] による R 値を用いる。帯域幅を要求するアプリケーションには、実効スループットと必要帯域幅から算出するスループット余裕度を用いる。

### 6.2 評価環境

評価実験は、Containerlab[13] を用いて構築した仮想ネットワーク上で実施した。ルータには FRRouting[14] を使用し、SRv6 機能を有効化した。トポロジは、15 台のクライアント、6 台のルータ、3 台のサーバで構成される（図 2）。

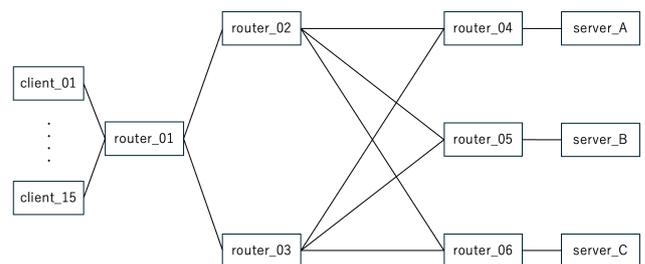


図 2 評価用ネットワークトポロジ

評価に用いたアプリケーションの要件を表 1 に示す。

表 1 アプリケーションの要件

アプリケーション	許容遅延 (ms)	必要帯域幅 (Mbps)
A (低遅延要求)	50	-
B (帯域幅要求)	-	50
C (両方要求)	100	30

## 6.3 実験結果

### 6.3.1 アプリケーション A (低遅延要求)

提案手法では、全クライアントに対して低遅延経路が選択された。RTT 測定結果を表 2 に示す。

表 2 アプリケーション A の RTT 測定結果 (平均値)

クライアント	ECMP (ms)	提案手法 (ms)	R 値 (ECMP)	R 値 (提案)
client_01	100.9	5.4	90.5	91.6
client_02	100.9	5.4	90.5	91.6
client_03	0.2	5.5	91.7	91.6
client_04	0.2	5.4	91.7	91.6
client_05	0.2	5.4	91.7	91.6

ECMP では、client\_01-02 が高遅延経路に振り分けられ RTT が 100.9ms となり、許容遅延 50ms を超過した。提案手法では全クライアントで RTT が 5.4-5.5ms となり、許容遅延を満たした。

### 6.3.2 アプリケーション B (帯域幅要求)

提案手法では、全クライアントに対して高帯域幅経路が選択された。両手法とも全クライアントでスループット平均値が約 100Mbps となり、必要帯域幅 50Mbps を上回った。本実験条件ではリンク容量に対してトラフィック量が少なかったため、両手法に差は見られなかった。

### 6.3.3 アプリケーション C (両方要求)

提案手法では、ネットワーク状態に応じて動的に経路が選択された。RTT 測定結果を表 3 に示す。

表 3 アプリケーション C の RTT 測定結果 (平均値)

クライアント	ECMP (ms)	提案手法 (ms)	R 値 (ECMP)	R 値 (提案)
client_11-14	101.1	65.3	90.5	90.9
client_15	200.4	65.4	89.3	90.9

ECMP では、client\_15 が高遅延経路に振り分けられ RTT 200.4ms となった。提案手法では全クライアントで RTT が 65.3-65.4ms となり、許容遅延 100ms を満たした。

## 6.4 考察

提案手法は、全てのアプリケーションにおいて、それぞれのネットワーク要件に応じた経路選択ができることが確認できた。特に低遅延を要求するアプリケーション A では、ECMP で許容遅延を超過していたクライアントの RTT を約 95ms 低減し、R 値を 90.5 から 91.6 へ改善した。アプリケーション C では、最大 135ms の RTT 低減を達成した。これらの結果から、提案手法は ECMP で生じていたクライアント間の QoE 不均衡を解消し、全クライアントで一貫したネットワーク性能を実現できることが確認できた。

## 7. 関連研究

杉浦らは、SRv6 を用いたアプリケーションの通信特性を考慮した通信経路制御システム Acar[15] を提案した。Acar は帯域幅のみを考慮し、アプリケーションの要求帯域幅を満たす経路を選択する。本研究では帯域幅だけでなく遅延も考慮し、両方を要求するアプリケーションにも対応する点が異なる。

Campo らは、DNS トラフィックから e2LD を抽出してアプリケーションを識別する手法 [16] を提案し、98% の精度でアプリケーションを識別できることを示した。本研究ではこの知見を踏まえ、e2LD によるアプリケーション識別を SRv6 による経路制御に活用した。

## 8. おわりに

本研究では、DNS によるアプリケーション識別と SRv6 による経路制御を組み合わせた、アプリケーション識別型トラフィック制御システムを開発した。評価実験の結果、提案手法は全アプリケーションでネットワーク要件に応じた経路選択が可能であることが確認できた。低遅延を要求するアプリケーションでは、ECMP で許容遅延を超過していたクライアントの RTT を約 95ms 低減した。これらの結果から、提案手法は ECMP で生じていたクライアント間の QoE 不均衡を解消できることが確認できた。

今後の課題として、e2LD では識別できないサブドメインへの対応が挙げられる。例えば、Google が提供するサービスでは calendar.google.com や mail.google.com のように、e2LD より 1 階層下のサブドメインが個別のアプリケーションを表している。これらを識別するためには、e2LD+1 階層下のドメイン名も考慮する必要がある。

## 参考文献

- [1] 総務省: 我が国のインターネットにおけるトラフィックの集計・試算 (2024 年 5 月時点の集計結果の公表), 総務省報道資料 (2025).
- [2] Netflix, Inc.: 推奨されるインターネット接続速度, Netflix ヘルプセンター (2024).
- [3] ITU-T: One-way transmission time, Technical Report Recommendation G.114, International Telecommunication Union (2003).
- [4] Filsfils, C., Previdi, S. et al.: Segment Routing Architecture, RFC 8402 (2018).
- [5] Camarillo, P. et al.: Segment Routing over IPv6 (SRv6) Network Programming, RFC 8986 (2021).
- [6] Filsfils, C., Li, Z. et al.: IPv6 Segment Routing Header (SRH), RFC 8754 (2020).
- [7] : Public Suffix List, <https://publicsuffix.org/>. Accessed: 2025.
- [8] Edmonds, R.: dnstap: flexible, structured event replication format for DNS software, dnstap Official Website (2024). (accessed 2025).
- [9] Machard, D.: DNS-collector v1.14.0, GitHub Releases (2025). (accessed 2025).

- [10] InfluxData Inc.: Telegraf: The plugin-driven server agent for collecting and reporting metrics, Telegraf Official Website (2024). (accessed 2025).
- [11] Hopps, C.: RFC 2992: Analysis of an Equal-Cost Multi-Path Algorithm, <https://datatracker.ietf.org/doc/html/rfc2992> (2000). (accessed 2025).
- [12] : The E-model, a computational model for use in transmission planning, Technical Report ITU-T Recommendation G.107, International Telecommunication Union, Geneva, Switzerland (2015).
- [13] : Containerlab, <https://containerlab.dev/>. Accessed: 2025.
- [14] : FRRouting, <https://frrouting.org/>. Accessed: 2025.
- [15] 杉浦智基: Acar: SRv6 を用いたアプリケーションの通信特性を考慮した通信経路制御システム, 修士論文, 奈良先端科学技術大学院大学 (2022).
- [16] Campo, C., Garcia-Rubio, C., Jimenez-Berenguel, A., Moure-Garrido, M., Almenares, F. and Díaz-Sanchez, D.: Inferring mobile applications usage from DNS traffic, *Ad Hoc Networks*, Vol. 163, p. 103601 (online), <https://www.sciencedirect.com/science/article/pii/S1570870524002129> (2024).