

SDN 環境下での効果的な マルチパスルーティングの考察

Consideration of Efficient Multipath Routing under Software Defined Networking

井原 栄[†]
Sakae Ihara

今泉 貴史[‡]
Takashi Imaizumi

1. はじめに

近年、ネットワーク技術は急速に変化しており、その管理や運用の面で、ネットワークアーキテクチャの物理的制限がボトルネックとなっていた。そこで新たに注目されたのが Software Defined Networking (SDN) の概念である。SDN を一般化する Open Network Fundation (ONF) によれば、SDN とは経路制御を行うコントロールプレーンとデータの転送を行うデータプレーンを分離して扱い、ネットワークの仮想化を実現することである。管理者は物理的な制限なく自由なネットワークの構築が可能になる。[1]

SDN の環境では、制御層からの要求に柔軟に対応するために、物理層は各端末がメッシュ状に構成されると必要がある。そのようなトポロジで重要になるのがマルチパスルーティングである。ネットワークをメッシュ状に接続する事は、経路に冗長性を持たせることであり、高帯域幅、耐障害性などの効果が期待できる。しかし、一般的なルーティング手法であるシングルパスルーティングでは単一の経路しか利用されないことで資源の無駄になっている。

SDN のマルチパスに対応したルーティングプロトコルとして研究されているのが、IETF において策定されている transparent interconnection of lots of links(trill)[2] や IEEE で策定されている IEEE 802.1aq Shortest Path Bridging(SPB)[3] などである。これらは基本的に全て、等コストマルチパスを利用している。現実のネットワークでは、複数の同じコストの経路が存在する可能性は低いなど、負荷分散の観点から不十分と考えられる。

上記の問題に対応したルーティングとして、Multinomial Logit Based (MLB) ルーティング制御方式が提案されている [4]。これは心理学で利用される非集計ロジットモデルをネットワークのルーティング制御に利用したもので、確率式に基づいてパケットを各経路へ分配する。

しかし MLB ルーティング制御では、各経路にどれくらい分配するかを決定する確率変数の設定方法が定まっていなかった。ネットワーク資源を無駄なく効率的に利用するためには、ネットワークの状況に応じて経路を選択する必要がある。現実にはトラフィックは予測不可能なものであり、最適な経路の選択確率は動的に変化する。

本論文では、SDN の特徴である一元管理を使用して、MLB ルーティングを拡張したより効率的なルーティン

グ手法を提案する。

2. MLB ルーティング

先に述べたように MLB ルーティングでは、行動心理学で扱われている数学モデルである非集計ロジットモデルを利用している。まず非集計ロジットモデルの概要を簡単に述べた後に、その定式化を行う。

2.1. 非集計ロジットモデル

2.1.1. 非集計ロジットモデル概要

人が何かを選択するとき、ある個人を選択を一意に特定することは不可能であり、それらは確率的に表現する必要がある。そのような人の選択確率をモデル化したのが非集計ロジットモデルである。

例として人が道を選択する状況を考える。それぞれの選択肢には、「景色が綺麗」、「道が広いから走りやすい」など、それを選ぶ理由となる属性値が含まれている。基本的にはこの属性値が最大のものが選択されるはずである。しかし実際には、個人の気まぐれによって属性値が低い選択肢が選ばれる可能性がある。非集計ロジットモデルでは属性値という基準を導入したうえで、そのランダム性を確率分布によって表現する。

ルーティングにおいて、この選択の確率的分布はそのまま選択経路の分散に利用できる。帯域幅などのコストに依存した経路を選択するが、その選択は確率によるので一か所にトラフィックが集中することを防ぎ、結果的に負荷分散を実現する。

2.1.2. 非集計ロジットモデルの定式化

ある個人が N 個の選択肢集合 $J \in \{1, 2, \dots, N\}$ から、選択肢 j を選択する状況を考える。選びやすさという基準を以下のように定義する。

$$U_j \stackrel{\text{def}}{=} [j \text{ を選択したことで得られる効用}] \quad (1)$$

全ての選択肢はこの属性値を持っていて、これが高いものほど選択されやすくなる。しかし実際の選択では属性値に依存しないランダム性が含まれるので、これを以下のように分解する。

$$U_j = V_j + \epsilon_j \quad (2)$$

[†]千葉大学大学院融合科学研究科

[‡]千葉大学統合情報センター

V_j は観測可能な値である。 ε_j は確率項であり、効用を確率変数として表現している。このようにして選択を確率的な事象として表現する。 U_j は確率となり、 V_j が高いものほど選ばれやすいが、その確率は ε_j によって変動する。

非集計ロジットモデルでは、確率項についてガンベル (Gumbel) 分布を仮定している。式 2 の効用をガンベル分布に当てはめて変形したのが、式 3 の選択確率式である。

$$P_j = \frac{\exp[-\gamma U_j]}{\sum_{j \in N} \exp[-\gamma U_j]} \quad (3)$$

式 3 は j を選択する確率であり、 γ 値に依存する値となる。

ここで γ 値は選択確率を調整する確率変数である。 γ 値を大きくすることでコストの影響度が大きくなり、 V_j が最大のものを選ぶ可能性が高まる。逆に 0 に近づけるとすべての選択肢が等確率で選ばれるようになる。この γ 値を調整することで、コストに差がある経路をどの程度利用するかを制御することができる。従って γ 値はコストへの依存度と表現することができる。

本研究ではこの γ 値を利用して経路分散を実現する。詳しくは 3 章で述べる。

2.2. 転送確率計算法

MLB ルーティングは、非集計ロジットモデルにネットワークのルーティングを当てはめたものである。ここでは前のセクションで示した非集計ロジットモデルの選択確率 (式 3) に必要な情報を当てはめていく。

MLB 法では全ての選択肢経路の集合を考慮に入れるが、現実にはそれらを計算で求めることは困難である。しかし実際にそれらを計算する必要はなく、この先で示すマルコフ配分や重み行列といった概念を導入することで、リンクコストで定義する行列の計算のみに帰着できる。これが MLB ルーティング制御の優れたところであり、結果的に非常に少ない計算量ですべての経路を考慮したルーティングを実現する。

ここでは先に概念的な定式化を行い、それに基づく実際の計算方法を解説する。

2.2.1. 定式化

ノード集合 V 、リンク集合が E の有効グラフ $G = (V, E)$ を考える。ノード $i \rightarrow j$ のリンクを e_{ij} 、コストを c_{ij} とする。またあるノード v に対し、それに隣接しているノード集合を $Z(v)$ と定義する。

ここであるパケットをノード o からノード d を転送する場合を考える。選択肢となりえる全ての経路の集合を Φ^{od} とし、あらかじめ与えられているものとする。先に述べたように実際にはこの集合を求める必要はない。

これらをも非集計ロジットモデルにあてはめると以下の式ようになる。

$$P_r^{od} = \frac{\exp[-\gamma \cdot C_r^{od}]}{\sum_{r \in \Phi^{od}} \exp[-\gamma \cdot C_r^{od}]} \quad (4)$$

ここで、 C_r^{od} は $o \rightarrow d$ への r 番目の経路コストを意味し、経路を形成する各リンクコストの和で与えられるものとする。

2.2.2. ノードごとのテーブルの計算

式 4 は式 5 に示すマルコフ配分と等価であることが証明されている。[4]

$$p(j|i) = \exp[-\gamma c_{ij}] \frac{W_{jd}}{W_{id}} \quad (5)$$

$$W_{id} \stackrel{\text{def}}{=} \sum_{r \in \Phi^{id}} \exp[-\gamma C_r^{id}] \quad (6)$$

遷移 $o \rightarrow d$ の途中、あるノード i についたとする。そのとき、 i が隣接ノード j に遷移する確率が $p(j|i)$ である。式 5 より、その次の遷移先 j はそれ以前のルートに関係なくなる。

このように、等価なマルコフ配分が存在することを利用して、各ノードごとで確率 $p(j|i)$ に従って分配すればよい。また $p(j|i)$ は各ノードが保持するルーティングテーブルにそのまま適応する事が出来る。宛先と隣接ノードの関係をこのルーティング方式に従って計算する事でルーティングテーブルが作成できる。

表 1: テーブル例

宛先	次ノード	転送確率
ホスト A	j_1	0.5
	\vdots	\vdots
	j_N	0.1
ホスト B	j_1	0
	\vdots	\vdots
	j_N	0.2

ただし j_n は隣接ノード集合 Z に含まれる。また、表 1 のホスト B の j_1 向けのような、確率 0 のノードは保持する必要がないので、実際のテーブルはさらに小さくなる。

W_{id} を求めることができればこのようにルーティングを行うことが可能になる。ここで W_{ij} を要素に持つ行列 W を転送重み行列と名付け、その求め方を次のセクションで示す。

2.2.3. 転送重み行列について

前の節で述べた転送重み行列 W の求めかたについて述べる。式6が W_{ij} の定義であるが、計算にはあらゆる経路コストを足し合わせる必要があり、そのまま計算するのは適当でない。これを行列の計算によって導出する方法が存在し、結果的に一度の逆行列の計算によってすべての W_{ij} を求めることができる。

各要素が式7となる $N \times N$ の行列 A を用意する。

$$a_{ij} = \begin{cases} \exp[-\gamma c_{ij}] & (\text{リンク } e_{ij} \text{ が存在する}) \\ 0 & (\text{それ以外}) \end{cases} \quad (7)$$

この時、行列 W は以下のように表せる。[4]

$$W = A + A^2 + A^3 + \dots \quad (8)$$

$$W_{ij} = \sum_{r \in \Phi^{ij}} \exp[-\gamma C_r^{ij}] \quad (9)$$

また、 A が Haskins-Simon 条件を満たすとき、

$$[I - A]^{-1} = I + A + A^2 + A^3 + \dots \quad (10)$$

となる。

式8と式10より行列 W は以下のように表現できる。

$$W = [I - A]^{-1} - I \quad (11)$$

W は経路の重みを表した値であり、これを利用することで全ての経路を考慮したルーティングが可能になる。

2.2.4. Haskins-Simon 条件

転送重み行列 W を逆行列を用いた計算で求める過程で、Haskins-Simon 条件を満たす必要がある事を説明した。Haskins-Simon 条件は式12の等式と等価である。

$$\rho(A) = \max\{|\lambda|\} < 1 \quad (12)$$

ここで、 $\rho(A)$ は A のスペクトル半径であり、最大固有値 $\max\{|\lambda|\}$ である。 A の構造上 γ 値を減少させると $\rho(A)$ の値は上がっていく。よって γ 値には下限が存在することになる。実際には γ 値をそれ以上の値に設定すればよいだけであり、条件を満たすことは容易である。これについて詳しくは次の章で説明する。

2.3. 既存手法の課題

MLB 法では不等コストコストマルチパスを少ない計算量で実現する。等コストマルチパスに比べてより効果的な負荷分散が期待できる。

しかし MLB 法の問題点として、 γ 値の設定方法が確立されていない点がある。 γ 値を適切な値に設定しないと、負荷分散が十分に行われなかったり非効率な経路を選択する可能性がある。既存手法では適切な値を一意に定めていたが、実際にマルチパスのメリットを考えると経路の分配は、トラフィック量の変化に応じて変化させるべきである。トラフィックが多いときは負荷分散の必要性が高まるが、逆にトラフィックが少ないときに、コストの高い経路を通ることはパフォーマンスを下げることになる。このように γ 値の適切な値は、トラフィックが一定でないものである以上動的に変化するものと考えられる。

また γ 値はトポロジーに対して一つのパラメータなので、もし γ 値の設定でトラフィックを考慮した分散を実現しようとしても局所的な設定が行えないことから、効率的でない経路を選択する可能性が残る。

3. TCMLB ルーティング

本研究では、新たに各リンクごとにトラフィック量を導入してそれぞれの分配確率を制御する方法として、Two Cost MLB ルーティング手法を提案する。

3.1. 提案手法の概要

提案手法のねらいは、普段は最少コストの経路を利用するが、トラフィックが増大しているときにはコストの高い経路も利用することで全体の効率を上げることである。

MLB ルーティングにおける γ 値は、まさしく上記の経路選択を制御する値である。トラフィックの増加に伴い γ 値を 0 に近づけることで、コスト差がある経路を利用することができる。しかし γ 値はトポロジーにおいて一つのパラメータであり、全体の確率に影響する。局所的なトラフィックの増加に対応するためには、 γ 値のみの操作では不十分である。

そこで本研究では各リンクごとに二つのコストを利用することを提案する。

- 静的コスト
- 動的コスト

あらかじめ各リンクごとに不変な静的コストで MLB ルーティングを行い、その結果を踏まえ、各リンクのトラフィック量に応じて γ 値を制御する方法を考える。これにより、トラフィック量に応じて選択できる経路が増加するので、より効果的な負荷分散を行える。

ここで例として図における $i \rightarrow j$ の遷移に対して、 γ 値による選択確率の変化を図2に示す。図1のように

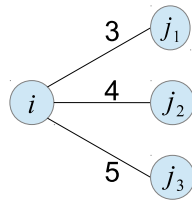


図 1: コスト差のある 3 つのノード

コストに差を持たせた三つのノードを用意し、それらの選択確率が γ 値を変更することでどのように遷移するかを図 2 に表した。

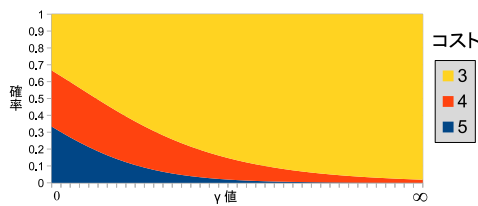


図 2: γ 値による選択確率の変化例

単純に各ノードのトラフィック量を利用して γ 値を変動させることは、ノードごとに複数回の MLB ルーティングを計算する必要があり、スケーラビリティに問題が発生する。

そこで本研究では、 γ 値による確率変化をトラフィックに帰着させるために、新たに、各リンクのトラフィック量から確率を決定する関数を用意する。各ノードにおける隣接ノードの選択確率は図 2 のように変化するので、あらかじめ二つの γ 値を用意し、静的コストによる MLB ルーティングの計算を行い確率の最大値と最小値を計算しておく。選択確率はトラフィックによりそこで求めた確率の間を遷移するので、簡単のために今それを直線の式と近似し、上で得られた確率を一次関数に変更する。図 3 に例を示す。

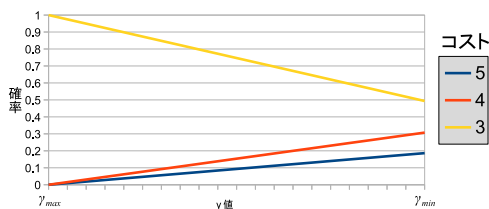


図 3: γ 値による選択確率の変化例

図の左側はトラフィック量が少なき、右側がトラフィック量が多いときの確率である。二つの γ 値とトラフィックが最大最少のときの確率を決定するもので、トラフィックが 0 のときに γ 値最大のときの確率となるようにする。また、トラフィック量はそのリンクの許

容量を超えたときに γ 値が最少における確率になるようにする。

この最小値は、トラフィックの増加に対して、どの程度までコストが高い経路を利用するかを決定する要素になる。これを低く設定すると、非効率な経路を多く選択しすぎる可能性があるため、あらかじめ適切な値を設定する必要がある。

SDN の制御部では、ノードごとに表 2 のようなテーブルを保持しておく。

表 2: 制御部で保持するテーブル例

宛先	次ノード	転送確立を導く関数
ホスト A	j_1	$p(j_1 i)$
	\vdots	\vdots
	j_N	$p(j_N i)$
ホスト B	j_1	$p(j_1 i)$
	\vdots	\vdots
	j_N	$p(j_N i)$

時間ごとにトラフィック量を収集し関数に当てはめ出した確率により各ノードのテーブルを更新する。

このようにして γ 値により変化する経路の選択確率をトラフィックに帰着させる。この関数に各リンクごとのトラフィック量を表す値を代入することで、トラフィックを考慮したルーティングが可能になると考える。

3.2. 範囲設定

ここでの範囲設定とは、静的コストによる MLB ルーティング計算の際に利用する γ 値の最大値と最小値を求めることである。ここで求める γ 値による確率の範囲の間で経路選択が行われるため、適切な値を設定する必要がある。

3.2.1. γ 値の範囲

先にも述べたように MLB ルーティングにおいて γ 値はコストへの依存度と表現できる。この制御によって各経路の選択確率が変動する。 γ 値の変化によって、パケットの転送がどのように変わるか図 4 に示す。

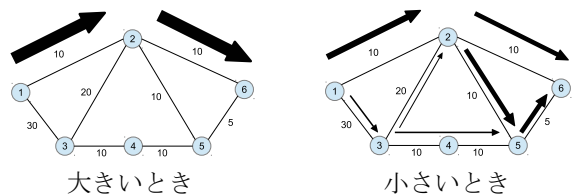


図 4: γ 値に対するパケットの遷移変化

トラフィックコストが増減したときに、図 4 の間を遷移するようにするのが提案手法となる。このセクション

では、どの程度の確率を遷移するかを決定するために γ 値の最大値と最小値を設定する。この範囲設定を行わないと、特にトラフィックが増加したときに、必要以上に非効率な経路を選択し続ける可能性がでてくる。以下に適切な γ 値の設定方法について考察する。

3.2.2. 最大値の設定

最大値については、最短経路を想定しているので γ 値を極限にとった確率となる。実際には γ 値が極限の場合の計算をする必要はなく、適当な γ 値で計算した結果から最も確率が高いものを 1、それ以外を 0 にすることで最短経路になる。また最大の確率を持つノードが複数あった場合に、それらを等確率にすることで等コストマルチパスにもなる。

3.2.3. 最小値の設定

一方、最小値についてはあらかじめ適切な値を与える必要がある。

γ 値を 0 にして計算することで、全ての隣接ノードが等間隔で選ばれるようになる。MLB ルーティングでは全ての経路を考慮しているため、非常に効率の悪い経路を選択する可能性がある。

本研究の目的はトラフィックの増加に対して、コストが高い経路を利用できるようにすることである。コスト差が少ない場合はこの方法が有効であるが、コスト差が大きい場合は効果が悪くなる。 γ 値の最小値はこのコストの許容量を制御する値になるので、ネットワーク全体を効率よく利用するためにこの値を適切に設定する必要がある。

本研究では最小値の設定法として、MLB ルーティングにおける計算上の限界値を利用する方法と、設計者がその値で分散が高すぎると判断した場合の調整法をしめす。

3.2.4. 計算上の下限

前章で述べたように、MLB ルーティングはリンクコストに基づいた行列の計算を行う。この過程で、 γ 値には下限が生じる。この下限値を最少の γ 値として利用することができる。

転送重み行列 W を求める際に Haskins-Simon 条件を満たす必要があった。リンクコストを表す行列に対し、式 13 を満たせばよいことになる。

$$\max_j \left[\sum_{j=0}^N \exp[-\gamma \cdot c_{ij}] \right] < 1 \quad (13)$$

式 13 の \max_j は γ 値を下げることで増加していくので、 γ 値をある一定値以下にすることで条件が満たされなくなる値が存在する。これが γ 値の下限となり、そのトポロジーにおいて最もコストに依存していないルートを通る可能性が高い状態である。

3.2.5. 下げすぎないための調整

上記の条件でトポロジーによる下限が求められたが、その γ 値では低すぎる場合を考える。その場合は最小値を手動で設定する必要があるが、以下の計算式を利用することで各確率を制御することができる。

$o \rightarrow d$ の最短経路パスをコストを C_{sp}^{od} 、その選択確率を P_{sp}^{od} とする。ここで C_{sp}^{od} に対して k_c 倍のコストである経路の選択確率が、 P_{sp}^{od} に対して k_p 倍以下になる値を計算する。

$$k_p P_{sp}^{od} = k_p \frac{\exp[-\gamma \cdot C_{sp}^{od}]}{\sum_{r \in \Phi^{od}} \exp[-\gamma \cdot C_r^{od}]} < \frac{\exp[-\gamma \cdot k_c \cdot C_{sp}^{od}]}{\sum_{r \in \Phi^{od}} \exp[-\gamma \cdot C_r^{od}]} \quad (14)$$

式 14 より式 15 が求まる。

$$\gamma > \frac{-\log k_p}{(k_c - 1)C_{sp}^{od}} \quad (15)$$

このように設計者が望む k_c と k_p を与えれば、その条件を満たす γ の下限を求めることができる。このようにして設計者側で確率を考慮した下限を計算することが可能である。

先ほど求めた下限値では低すぎる場合にこの方法で最低値を調整することでより効果的な分散を実現できる。

3.3. トラフィックコストによる関数の決定法

各ノードごとの選択確率の変動の範囲を求めた。確率はトラフィック量に応じてその間を遷移する。

ここでは各リンクから収集したトラフィック量を利用して実際のルーティングに用いる確率を決定するために、利用するトラフィック量と、確率を求める関数を作成する。

3.3.1. 利用するトラフィック量について

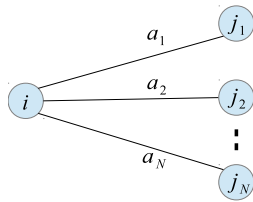
各ノードの選択は選択肢となる全てのノード間のトラフィックが影響する。よってこれらを一つのパラメータ T で表し関数へ適用できるようにする。

3.3.2. トラフィック量の関数への適応

i からのパケット送信において、次のノードへの選択確率について考える。

図 5 における a_1, a_2, \dots, a_N は各リンクにおけるトラフィック量を表す。

i からその隣接ノードへの遷移確率は、選択肢となるノード全てのトラフィック量を考慮すべきである。よってそれぞれ j の選択確率は上記のすべての a に依存する。

図 5: i の隣接ノードに動的コスト

また関数に適用させるために、各リンクごとに最も選択経路が分散している状態におけるトラフィック量を設定しておく。ここで A_{1max} を a_1 における限界値として取り決める。これを超えた場合は許容量を超えたとしてもっとも分散が多い状態にする。

本研究ではこの A_{max} を帯域幅とする。

$$T = \frac{1}{N} \left(\frac{a_1}{\max(a_1)} + \frac{a_2}{\max(a_2)} + \dots + \frac{a_N}{\max(a_N)} \right) \quad (16)$$

計算に利用する各トラフィック量は、限界値を超えないものとして収集したときに調整する T はそれぞれのリンクのトラフィック量に比例し、各リンクが許容量を超えた時点で 1 となる。

3.3.3. 関数の設計

以上によって求めた T を利用して、確率を導出する関数を作成する。

ノードの選択確率は先ほど求めた範囲の間で遷移する。よってトラフィックの変動に合わせてこの範囲の値を決定することでネットワークの状況に応じた経路選択が可能になると考える。

本研究ではこの選択確率とトラフィック量を簡単のために一次関数として実行する。

先ほど求めた範囲における γ 値で計算した結果を以下のように表記する。

- $P_{max}(j)$: γ 値が大値の時の確率
- $P_{min}(j)$: γ 値が最少の時の確率

今静的コストによる MLB ルーティングによって以上を求めた状態で、あるノードからノード j へ遷移を考える。その確率はトラフィック量が少ないときに $P_{max}(j)$ に近づき、トラフィックが限界に達したときに $P_{min}(j)$ となるようにする。またさきほど求めたトラフィック量を表す T を用いる。

ここでノード j における選択確率を以下のように設定する。

$$p_j(T) = P_{max}(j) + (P_{min}(j) - P_{max}(j))T \quad (17)$$

図 6 に式 17 を示す。

これに先ほど設定した T を入れることでトラフィックを考慮した経路分散が可能になる。

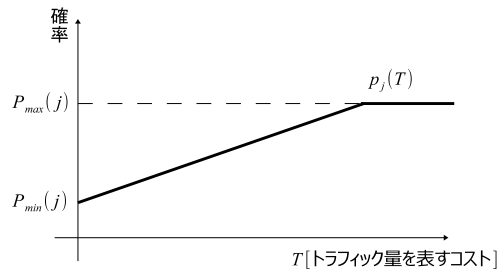


図 6: 選択確率をトラフィックによって求める関数

4. 考察

本論文では既存研究における MLB ルーティングにおいて計算式の拡張を行い、各リンクごとのトラフィック量を利用して γ 値設定をおこなうマルチパスルーティングを提案した。この章では、現在利用されている既存技術と、既存研究となる MLB ルーティングとの比較を考察し、そのあとに本提案の利点と影響について考察する。

4.1. 既存技術との比較

データセンタでのイーサネットファブリックにおいては SPB や trill の実装が主流である。詳しい実装は各ベンダーごとに異なるが、それらとの比較を考察する。

4.1.1. SPB trill

選択経路の数について、SPB や trill では全て等コストマルチパスを採用している。よって利用する経路は各ノード間にある等コストの経路のみである。SPB においては経路の最大数は (ノード数 \times 経路数) が 4096 以下になる数までしか保持できない。これは選択経路を識別するために ID を利用するためである。

その点 MLB、また TCMLB では γ 値が最大の状態では等コストマルチパスなので利用できる経路数は等しく、 γ 値を下げた状態での経路数は増加する。確率により非当コストの経路も利用するといえるので利用できる経路の数はこちらの方が多い。

しかし実際の経路差はトポロジに依存する。既存手法より提案手法の方が効果が上がるのはコストに差がないルートが多く存在する時である。

計算量について制御層への負荷となる。

SPB、trill はともにダイクストラのアルゴリズムを利用する。本手法では MLB ルーティングを行うのはトポロジーに変化が見られた場合であり、時間毎に行うのはテーブルの更新のみである。 $O(N)$ といえるのでコントローラにかかる不可は少ないと考えられる。

4.1.2. MLB との比較

既存手法では γ 値は固定であり、各リンクにコストは一つであったりした [4],[5]。

これに対し TCMLB ルーティング制御では、動的コストを用いることで静的コストを考慮にいれつつ、ダイナミックな経路を分散が可能になっている。これにより非効率な経路を無駄に選択することなく、経路の適切なタイミングで負荷分散を実現する。また、トラフィックを各リンクごとに考慮することで、局所的なトラフィックに対して分散を行い、それに依存しない経路での不必要な負荷分散を回避している。

以上のことから、ネットワーク資源を動的要素を考慮にいれつつ無駄なく利用できていると言えるので、既存の MLB ルーティング手法よりもさらなるパフォーマンスの向上が期待できる。

また計算量についても、時間ごとの MLB の計算はなくなり、TCMLB による関数の計算のみになるので、既存手法より低減を実現している。

4.2. TCMLB ルーティング制御について

4.2.1. 利点

本手法で行ったのは γ 値による経路変化をトラフィック量によって遷移するようにしたことである。トラフィックが少ない場合は γ 値を上昇させ、コスト優先の経路を選択する。逆にトラフィックが増加した場合には、 γ 値が下がった状態して、コストに依存しない経路を選出するようになる。このようにしてネットワークの状況に応じた負荷分散を実現している。

SDN の導入が期待されるネットワークでは、サーバーの高密度化によるトラフィックの局所性が問題となっている。そのような環境において負荷分散は重要であり、この手法の効果が大きいと考えられる。

また実装に関して、TCMLB では SDN を前提としている。実際の動作は制御層における計算と、時間毎の各スイッチのテーブル更新のみである。制御層にかかる不可を少なくしたことで導入が容易となり、また専用のヘッダも必要ないので、透過性に優れているといえる。

4.2.2. 効率化について

γ の最小値の設定 トラフィックが許容量を超えていると判断された場合にあらかじめ設定した最小値における MLB ルーティングの計算結果が反映される。

この最小値における確率が、コストに依存しないルートを最も多く選択する状態であり、最小値を低く設定することで、よりコストが高い経路を選択する可能性がある。適切に設定しないと、効率が悪くなる。

トラフィックが増加している状態において、どの程度までコスト差がある経路を選択する事が効率につながるかを実装で検証する必要がある。

関数の設計 TCMLB では、関数を用意して γ 値と確率の関係をトラフィックによって導けるようにした。静的コストによるルーティングを行い、 γ 値を変化させることでその最大の確率と最少の確率を選出した。選択確率はその範囲で変化すると考えられるので、TCMLB では簡単のために直線の式と近似して考えることで、トラフィック依存の確率選択を実現した。

これに関して関数やトラフィックの取り方で、効果が変わると考えられる。トラフィック量と経路選択は線形である必要はないので、静的コストと動的コストの関係を考慮し適切な関数を用意することでパフォーマンスの向上が期待できる。ただし、関数を複雑にすることは計算量の増加につながる。それらはトレードオフなので、コントローラへの負担を考慮しつつ適切な関数を用意することで効率が上がる可能性がある。実際に動作させて適切な値を検証していきたい。

4.2.3. 制限

ループについて 本提案手法はホップバイホップルーティングとなるので、各ノードはパケットがどのような経路を通って送信されたかを意識しない。そのため場合によってはループが発生する可能性があり、ネットワークの利用効率の下がる可能性がある。

確率的な転送になるので、最短経路の方が確率が高いことからブラックホールになることはない。また基本的にループを経由することは、コストの増加につながるため、ほとんどの場合でループの発生確率は無視できるほど小さいと考えられる。

問題となるのは、コストに極端な差があり、ループのホップ数が少ないトポロジーである。この場合最大で二分の一の確率でループバックが発生する。このようにループの発生はトポロジーの形態に依存する。その場合においてなんらかの対処が必要になる。

幸い今回は SDN であるのでトポロジーの監視や制御は容易である。パケットのバックを禁止することや、ループの発生する可能性が高い箇所をあらかじめ抽出し、そのノードにおける経路選択を制限することでループを削除できる。これらの簡単な対処ほうについても考案する必要がある。

5. おわりに

5.1. まとめ

本論文では、メッシュネットワークにおける資源を有効に利用するためのマルチパスルーティング手法として TCMLB ルーティング手法を提案した。これは SDN を前提とした条件で MLB ルーティングを拡張したものである。MLB ルーティングのパラメータである、 γ 値をトラフィック量に依存して制御するために、新たに関数を用意する。この関数に各ノードごとのトラフィック量を代入することで近似的に γ 値の変動によって得られる確率を導き出す。このようにしてトラフィックを考慮した動的なルーティングを実現する。

TCMLB 法によってトラフィック量に応じた経路選択が可能になり、効率的にネットワーク資源を利用できると考えられる。また実装により帯域の増加やスループットの減少など、パフォーマンスの向上が期待できる。

5.2. 今後の課題

実際にこれらを実装することで効果を検証していきたい。また最小値の設定や、関数と動的コストの関係など、さらなる改良の可能性があるので、トポロジーやパラメータを変化させて検証していき、これらの適切な設定を求め、またその自動的な設定方法を考えていきたい。

参考文献

- [1] “open networking foundation”, 2014. [http](http://).
- [2] IETF. Transparent interconnection of lots of links (trill), 2014.
- [3] IEEE. <http://tools.ietf.org/html/rfc6329>, 2012.
- [4] Y. Shimonishi Y. Honma, M. Aida and Iwata A. A new multi-path routing methodology based on logit type assignment. Technical report, Electronics, Information and Communication Engineers, 2009.
- [5] H. Shimonishi Y. Honma M Aida Y. Shinohara, Y. Chiba. An efficient multipath routing algorithm for datacenter networks and its implementation on opneflow-based network. Technical report, Electronics, Information and Communication Engineers, 2009.