Vol.2015-CSEC-69 No.34
Vol.2015-IOT-29 No.34
2015/5/22

自律分散型適応的経路制御手法のOpenFlow実装

柏崎 礼生[1,a)]　下條 真司[1,b)]　髙井 昌彰[2,c)]

概要：計算機やデバイスが複数のネットワークに接続するマルチホーミングにより構成されたマルチホームネットワークにおいて各拠点の経路制御ノードが自律的にネットワーク情報を収集し、遅延時間に基づく自律分散型の経路制御手法を著者らは提案した。本手法ではパケット単位での確率論的分散が行われるため、遅延時間から求められる配分割合を高精度で実現できる特徴があるが、フローが分割されるため通信品質を劣化させる問題点が指摘されていた。本稿では提案手法を改善し OpenFlow を用いた実装について議論を行う。

An implementation of an autonomous distributed adaptive network routing by OpenFlow

Hiroki Kashiwazaki[1,a)]　Shinji Shimojo[1,b)]　Yoshiaki Takai[2,c)]

Abstract: Multihoming means connecting a computer or a device to more than one network. Authors proposed an adaptive routing method with an autonomous distributed manner based on latency time. In the method, each router probe network information autonomously and make its own routing table independently. Though the method has an advantage on sufficient accuracy of load balancing ratio calculated from latency information with per packet processing, a issue is pointed that per packet processing degrade communication quality because of separation of a flow. To deal with the problem, an improved method and its implementation by OpenFlow are discussed in this paper.

Keywords: multihoming, adaptive routing, latency

1. はじめに

インターネットを利用したサービスを提供あるいは利用する際に、その利用の価値や重大性の度合いに依存して、耐障害性を高めるための冗長回線を用意することがある。全ての回線を常に利用するアクティブ・アクティブ構成は、複数の回線を稼働系と待機系に分けて、稼働系の主回線で障害が発生した場合にのみ待機系である冗長回線を利用するアクティブ・スタンバイ構成と比べ、回線投資に対する効率が高い。アクティブ・アクティブ構成では複数の回線に対してトラフィック要求を分散させる機序が必要である。トラフィック要求が常に一様である場合、各々の回線の帯域幅を制約条件として、トラフィック要求の分散を静的に定めることができる。回線が共用である場合、制約条件は変動し、この分散を静的に定めることはできない。またトラフィック要求が動的に変化する場合もまた、この分散を静的に定めることができない。ネットワークに対して技術や科学原理を計測、特性評価、モデリングやトラフィック制御に応用してネットワークの性能評価や性能最適化問題に対処する取り組みをトラフィックエンジニアリングという [1–3]。アクティブ・アクティブ構成で複数の回線を有効活用するためには、トラフィック要求や回線の可用帯域の変化といったネットワークの変化に対してトラフィック要求の配分を変化させる適応的なトラフィックエンジニアリングが必要である。

適応的なトラフィックエンジニアリングを実現する方法

1 大阪大学
Osaka University
2 北海道大学
Hokkaido University
a) reo@cmc.osaka-u.ac.jp
b) shimojo@cmc.osaka-u.ac.jp
c) takai@iic.hokudai.ac.jp

ⓒ 2015 Information Processing Society of Japan

はそのアプローチの観点から大きく二つに分類することができる。一つは中央集権型の手法であり、もう一つは自律分散型の手法である。中央集権型の手法では、対象とする系全体の変化を観察するスーパーノードがあり、系を構成する各経路制御ノードが従う経路制御表はこのスーパーノードにより与えられる。一方、自律分散型の手法では各経路制御ノードが局所的な情報から個別の経路制御表を作成し、それに従うことで系全体のトラフィックエンジニアリングを実現する。中央集権型の手法ではスーパーノードが各拠点間のトラフィック要求が辿る経路の割り当てを俯瞰することができるため、ループのない経路割り当てを実現することができる。クラウドコンピューティング環境を利用した中央集権型の手法が提案されているが [4]、ノード数の増大に対して組み合わせ総数は等比級数的に増大する問題点が指摘されている。自律分散型の手法では系全体の評価を行わないため、計算量爆発の問題は発生せず、各ノードが保有しなければならない情報量や計算能力も比較的小さい。人工生命のアプローチを利用した自律分散的な手法が提案されているが [5,6]、経路のループを回避するために全てのパケットに経由情報を記載しているオーバーヘッドが大きい点が指摘されている。

IP ネットワークのトラフィックエンジニアリングにおいて、パケット単位での配分がトラフィック要求を複数の回線に最も細かく配分することができる。しかしパケット単位での配分はパケットの到着順序を乱すことがある。バッファ領域を確保することでパケットのリオーダリングができるが、バッファ領域にパケットを蓄積する遅延が通信品質に悪影響を及ぼす。フロー単位でのトラフィック要求の配分においてはこの問題は発生しない。ONF[*1] が提唱する OpenFlow はその仕様に定められているように [*2] フロー単位で制御を行う。今後 OpenFlow 対応スイッチの需要が増大するならば、OpenFlow で実現可能なトラフィックエンジニアリング手法はより安価に高速なフロー制御を実現することができるようになる。本稿では自律分散型の適応的なトラフィックエンジニアリング手法を OpenFlow で実装可能なアプローチに改良し提案する。

2. 経路制御

自律分散型の手法では各ノードが自律的に情報を収集し、各ノードが独立した経路制御表を作成する必要がある。提案手法では各ノードは隣接ノードに対して探査のためのパケット (プローブ) を送信する。プローブのペイロードにはそのプローブが任意のノードで生成された時、あるいは到着した時のそのノードの識別子と時刻情報が追記される。この時刻情報はノード間の伝送遅延時間を計算するために記述される。そのため各ノードの時刻は 100 マイクロ秒オーダーで同期が行われているものとする。

[*1] https://www.opennetworking.org
[*2] https://www.opennetworking.org/ja/sdn-resources-ja/onf-specifications/openflow

2.1 プローブ

各ノードはすべての目的ノードに関する経路制御表 (次ノード候補の集合) を有している．ノードの識別子を N_i と表すとき、ある目的ノード N_d に向けた次ノード候補 N_a には，長さ w のスコア列 $L_{N_dN_a} = \langle S^1_{N_dN_a}, S^2_{N_dN_a}, \cdots, S^w_{N_dN_a} \rangle$ が与えられる．このスコアはノードに到着したプローブに付随する時刻情報から得られる．ノード N_p にプローブが到着した場合，そのプローブの付加情報は例えば $\langle N_1, t_1, N_2, t_2, \cdots, N_{p-1}, t_{p-1}, N_p \rangle$ のようになっている．ここで T_i はその前に書かれた識別子を持つノードで生成された、あるいはそのノードに到着した時刻情報である。プローブは全てのノードに対して定期的に発信される。プローブが後述する配分割合に従って次ノードへと転送され、目的ノードに到着するとペイロードに記載された情報が転載されたフィードバックパケットが生成される。フィードバックパケットはペイロードに記載された経路の逆順を辿って送信元ノードへと送り返される。フィードバックパケットを受け取った各ノードはそのペイロードの情報をもとに目的ノードに対して次ノードを選択した場合、到着までにどれだけの時間を要したかの総遅延時間情報を得ることができる。この付加情報に含まれる始点ノード N_1 とすべての経由ノード $\{N_1, N_2, \cdots, N_{p-1}\}$ をそれぞれ目的ノードとする経路制御表において，次ノード候補 N_{p-1} に対するスコアの先頭 $S^1_{N_mN_{p-1}}, (1 \le m \le p-1)$ に，各経由ノードから現在パケットが到着したノード N_p までの総遅延時間を書き込む (図 1)．すなわち $S^1_{N_mN_{p-1}}$ は (1) 式により求められる．

$$S^1_{N_mN_{p-1}} = \sum_{i=m}^{p-1} d_i \tag{1}$$

経路制御表に新しいスコアが書き込まれる前に，古いスコア列の要素を 1 つずつ後退させ，列長を越えたスコアを破棄する．すなわち，スコア列には常に最新の w 個のスコアが保存される．IPv4 を用いた広域イーサ網での IP パケット経路制御を想定した場合，各ノードを一意に定める情報に 16 ビット，情報情報に 16 ビットと想定し 1 ホップごと合計 4 バイトの情報がペイロードに追加されるとしてもプローブのペイロードを 1400 バイトとして，350 ホップの情報を積載することができる。またプローブを 1 秒間に 1,000 パケット送信した場合、回線が 1Gbps のときプローブにより 0.14%が専有される。

2.2 配分割合の決定

各ノードはフローをどの隣接ノードに転送するかを定める。複数の転送先がある場合、目的ノードと次ノードに対

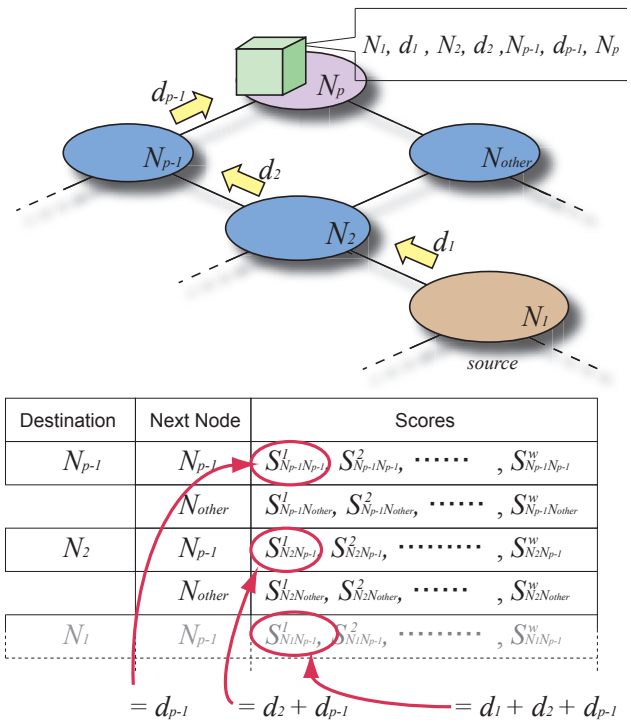

図 1　スコア付き経路制御表
Fig. 1　Routing table with scores.

応する経路制御表のスコア列を参照し，その配分割合を決定する。配分割合は次の順序で決定される。

(1) 目的ノード識別子
(2) 送信元ノード識別子
(3) ある目的ノードのあるポート
(4) ある送信元ノードのあるポート

次ノードとしてすべての隣接ノードが候補に挙げられるが，既に経由したノードおよび目的ノードへの到達が不可能なノードは除外する．候補に残った隣接ノードを N_{a_i}, $(1 \leq i \leq r$, r は候補総数）とする．

候補が複数存在する場合, 目的ノード N_d, 候補ノード N_{a_i} のスコア列に対する代表値 Q_{a_i} を, $Q_{a_i} = \sum_{m=1}^{w} C_m S^m_{N_d N_{a_i}}$ とするスコア列全体に対する加重平均で与える．ここで C_m は負の傾きを持つ線形の加重関数である．

Q_{a_i} は遅延時間であり，すべての候補の中で最小の Q_{a_i} が選ばれることが望ましいが，決定論的に選択すると各ノードが収集できる遅延時間情報が偏向してしまう．そこで次ノードは，すべての候補の代表値 Q_{a_i} を $\lambda(>0)$ 乗した値の逆数で加重された確率分散で決定されるものとする (図??)．すなわち候補ノード N_{a_i} が次ノードとして選ばれる確率 P_{a_i} は (2) 式で表される．

$$P_{a_i} = \left(\frac{1}{Q_{a_i}}\right)^{\lambda} \left(\sum_{j=1}^{r} \left(\frac{1}{Q_{a_j}}\right)^{\lambda}\right)^{-1} \qquad (2)$$

次ノードの選択確率を定めるパラメータ λ は，候補の中から相対的に少しでも優位なノードが選択される確率を強める働きがある．明らかに $\lambda \to \infty$ では，最良の候補が確率 1 で選ばれることになる．

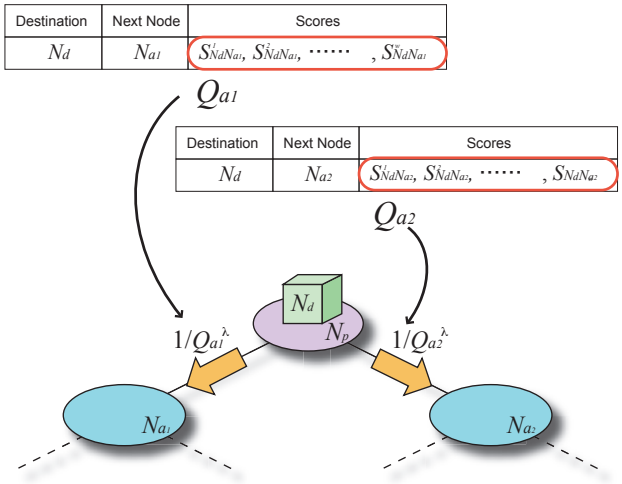

図 2　配分割合の算出
Fig. 2　Calculation of distribution ratio

3. OpenFlow での実装

前節で得られた配分割合に従ってトラフィック要求を配分する手法を xFlow と OpenFlow を用いて実装する。配分割合はトラフィック要求量の割合である。任意のノードは、そのノードから発生するトラフィック要求量とそのノードに流入するトラフィック要求量を知ることができる。全てのパケットの送信元ノード情報と目的ノード情報をノード識別子や IP アドレスだけでなくポートまで把握することの負荷は大きい。そこで処理負荷を軽減するためにフローサンプリング技術、特に xFlow と呼ばれる sFlow [7], NetFlow [8], およびその IETF 標準技術である IPFIX [?,9], PSAMP [11] を用いる。任意のノードの配下で発生するトラフィック要求とそのノードを通過するトラフィック要求をサンプリングすることにより、各フローの大きさを把握することができる。各ノードは全ての目的ノードに対する配分割合に対して、その配分割合に沿う形でこのフローを、フローのトラフィック要求量の小さな順序で割り当てていく。フローの粒度は配分割合の決定と同様に以下の粒度の順序で分類されているものとする。

(1) 目的ノード識別子
(2) 送信元ノード識別子
(3) ある目的ノードのあるポート
(4) ある送信元ノードのあるポート

配分割合に従うように最初は目的ノードで分類されたフローを割り当てる。配分割合は目的ノードごとに与えられているため、目的ノードで分類されたフローを割り当てることができるのは、目的ノードに対する配分割合が 100%の時のみである。そうでない場合は次の送信元ノード識別子

を元にしたフローの分割をして割り当てる必要がある。最も小さなフローから順番に複数の次ノードに対して割り当てていくが、どの次ノードに割り当てても与えられた配分割合を超過してしまうフローが現れた時は、次の粒度である目的ノードのポートで分割する。これを繰り返すことにより与えられた配分割合を実現するフローの配分を実現することができる (図 3)。

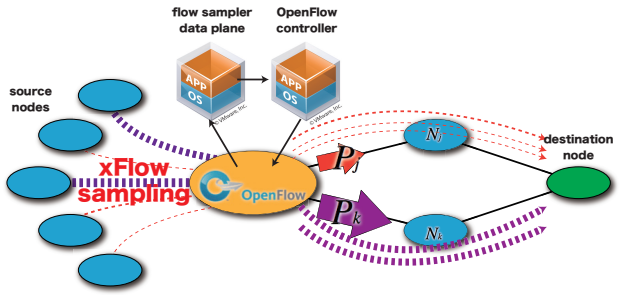

図 3　xFlow と OpenFlow を用いたトラフィック要求の配分
Fig. 3　Distribution of traffic demands with xFlow and OpenFlow

配分されたフローは全て OpenFlow で制御することができる情報により構成されているため、全ての目的ノードに対するフローを OpenFlow コントローラに対して与えることにより、与えられた配分割合に沿うトラフィック要求の配分を実現することができる。

4.　検証実験

本提案手法の初歩的な評価のため、プローブにより配分割合が決定するまでに要する時間をシミュレーションを用いて検証した。評価実験は，国内の主要都市に経路制御ノードが 1 つ以上置かれて敷設された広域ネットワークを状況として想定し，64 個のノードから構成されるランダムネットワークモデルとネットワークの拠点特性を反映させたスケールフリーネットワーク [?] の 2 種類を用いた。各隣接ノード間の遅延時間を示すコストは全て 4(ms) としている。ランダムネットワークは，各ノードから一定距離内にあるノードにリンクを張ることで作成する．また，スケールフリーネットワークは BA モデル [12] を用いて作成する．BA モデルにおける各ノード間のリンク生成確率は (3) 式により与えられる．本論文では初期ノード数 $m_0 = 2$ とし，リンク数のべき指数が $\gamma \simeq 2.8$ である 64 ノードのスケールフリーネットワークを作成した．

$$\Pi(k_i) = \frac{k_i}{\sum_j k_j} \tag{3}$$

ランダムネットワークにおいてノード 38〜45 間を行き交うパケットは，3 つの区間 (9,51), (4,2), (50,22) のいずれかを通る．図 5 は，提案手法 ($\lambda = 4$) において到着したパケットがこの各々の区間のいずれを経由したかの割合を示すものである．測定条件は各ノードがプローブを 1000ms で 1000 個発生させている。各プローブの宛先は一様分布に従う。Dijkstra 法を用いた最短遅延経路は区間 (9,51) を経由するものであり、提案手法の初期状態ではスコア列が空であるため次ノードの配分割合は全て等しく行われる。フィードバックパケットから得られた遅延時間情報によりスコア列が空でなくなると配分確率が変動する。1700ms の時点から区間 (9,51) を含む経路が最も高い確率で選択されるようになり、Dijkstra 法の結果を反映した結果となることが分かる。

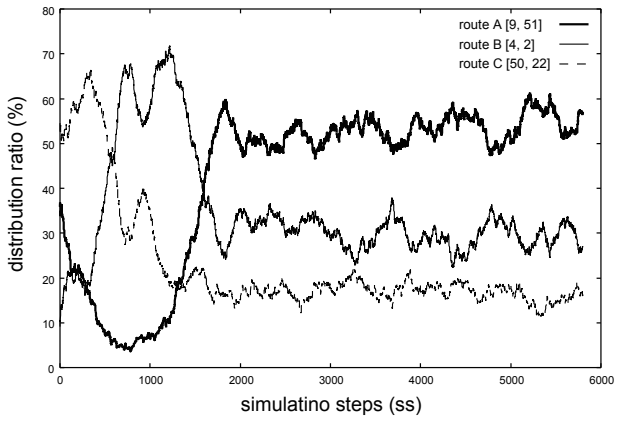

図 5　ランダムネットワークにおける初期状態からの配分割合の推移
Fig. 5　Changes of distribution ratio from initial state in a random network

スケールフリーネットワークにおいて，障害状態からの回復を評価する。任意の隣接する 2 ノードを結ぶ経路のコスト値を 300 ms に増加させ，2000 ms 後に 4ms に戻す．スケールフリーネットワークでも初期状態から 2000 ms までには配分割合が安定する。2000 ms でコスト値の増大を起こした。図 6 はリンク (9,51)，(22,50)，および (4,2) を経由する配分確率の変化を示した．リンク (9,51) がリンク値を増大させた箇所である。リンク値を増大させたあと、40ms でにリンク (9,51) を通る経路の選択確率は 1.0 % 以下となった。その結果リンク (4,2) を通るトラフィック要求が一時的に増大し、(22,50) を通るトラフィック要求との間で配分割合の振動が発生する。これは安定しないまま 2000 ms が経過する。2000 ms が経過すると (9,51) のリンク値が 4 ms に戻る。戻した直後は (9,51) を通るような配分確率がそもそも低いためプローブに検出されにくい。しかし次第にプローブが (9,51) を通過した結果が反映され、リンク値が戻った後 500 ms で配分割合の回復が始まる。最終的には 1200 ms 後に配分割合は安定するが、これは初期状態の配分割合とは異なる配分となっている。

5.　まとめ

遅延時間を用いた自律分散型適応的経路制御手法の機序

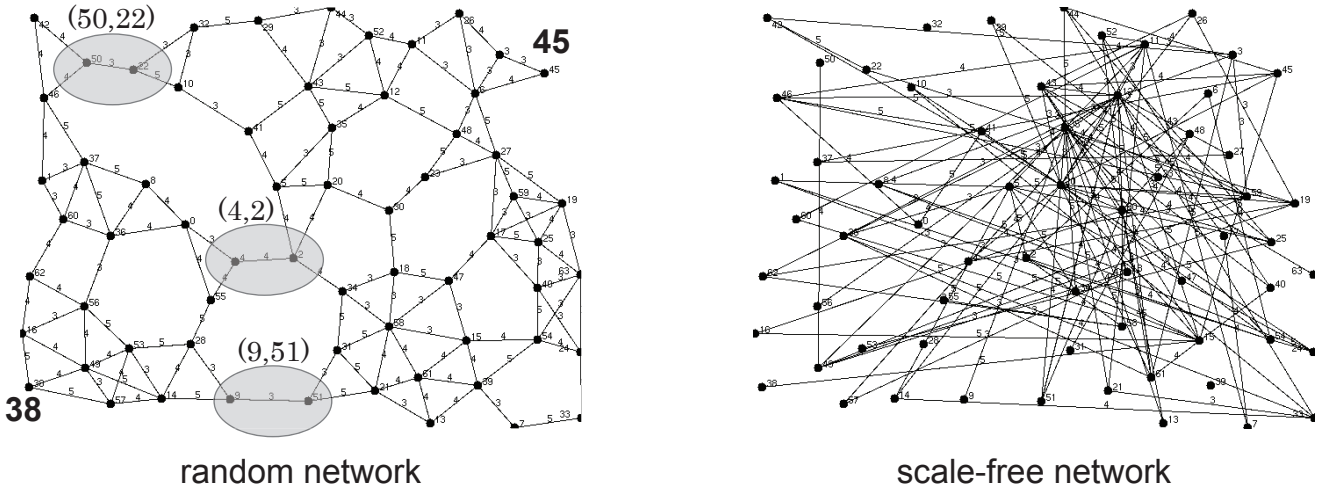

図 4　評価実験で用いるネットワーク
Fig. 4　Topologies of network model for an evaluation

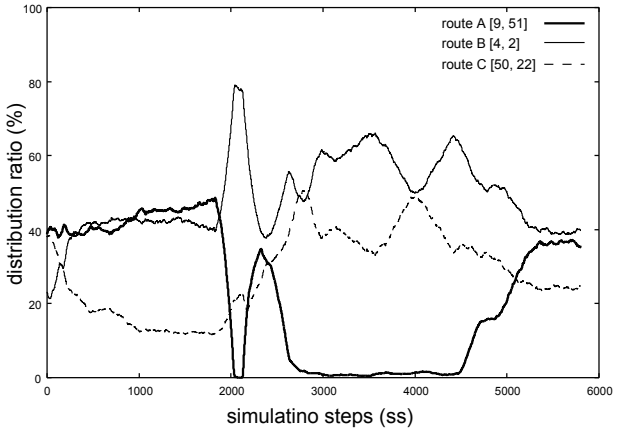

図 6　スケールフリーネットワークにおける障害時の経路配分確率の推移
Fig. 6　Changes of distribution ratio with a traffic disturbance in a scale free Network.

を改善し、xFlow と OpenFlow で実装することができる手法を提案した。プローブにより得られた遅延時間情報をもとに、64 ノードのランダムネットワークにおいて 2 秒未満で初期状態から配分割合が安定することをシミュレーションを用いて示した。現時点では簡易的なシミュレータによる評価実験に留まっているが、より詳細な設定を記述することが可能な NS-3 などのネットワークシミュレータを用いた評価実験を行い、より多様なトポロジやトラフィックパターンを設定し、統計的に有意と言える傾向を見出すことが目的である。また xFlow と OpenFlow による実装可能性を考慮した手法であるため、実機を用いて実装を行い、実ネットワークにおける評価実験を行うことを検討している。

マルチホームネットワークは耐障害性を向上するために用いる利点があるが、その耐障害性が実際の障害時や自然災害時において実際に機能するかを検証する必要もある。現時点ではトラフィック要求の配分割合のみを与えているが、可用帯域推定をもとに配分したトラフィック要求が可用帯域を超過しないような機構を本手法に追加する予定である。またパケット損失が発生した場合における対処について、既存の手法ではペナルティスコアを与えることにより回避しているが、このスコアの値には明確な根拠がない。損失が発生した場合の回避方法について妥当性を有する手法を考案することが今後の課題である。

謝辞　本研究は平成 27 年度北海道大学情報基盤センター共同研究「耐障害性を検証し自律進化する広域分散プラットフォームの検証」，平成 27 年度国立情報学研究所共同研究「学術拠点を利用したデータ基盤サービスとそのアプリケーションに関する検証」による支援，および JSPS 科研費課題番号「26870325」の助成を受けました．

参考文献

[1] M. O'Dell, J. McManus: Requirements for Traffic Engineering Over MPLS, RFC2702 (1999)
[2] Daniel O. Awduche: MPLS and Traffic Engineering in IP Networks, Journal of Communications Magazine, IEEE, Vol. 37, No. 12, pp. 42–47 (1999)
[3] A. Elwalid, I. Widjaja, X. Xiao: Overview and Principles of Internet Traffic Engineering, RFC3272 (2002)
[4] 柏崎礼生, 髙井昌彰: オーバレイルーティング網における広帯域映像配信のための適応的トラフィックエンジニアリング, 情報処理学会論文誌, Vol. 54, No. 1, pp. 404–413 (2013)
[5] 柏崎礼生, 髙井昌彰: 遅延時間情報に基づく適応的ネットワークルーティング, 情報処理学会論文誌, Vol. 47, No. 12, pp. 3308–3318 (2006)
[6] 柏崎礼生, 小林悟史, 河合修吾, 大石憲且, 髙井昌彰: 片方向遅延を用いたネットワークトラフィックの適応的負荷分散手法, 情報処理学会論文誌, Vol. 49, No. 3, pp. 1194–1203 (2008)
[7] P. Phaal, S. Panchen, N. McKee: InMon Corporation's

sFlow: A Method for Monitoring Traffic in Switched and Routed Networks, RFC3176 (2001)
[8] B. Claise, Ed: Cisco Systems NetFlow Services Export Version 9, RFC3954 (2004)
[9] B. Claise, Ed: Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information, RFC5101 (2008)
[10] J. Quittek, S. Bryant, B. Claise, P. Aitken, J. Meyer: Information Model for IP Flow Information Export, RFC5012 (2008)
[11] B. Claise, Ed, A. Johnson, J. Quittek: Packet Sampling (PSAMP) Protocol Specifications, RFC5476 (2009)
[12] Albert-László Barabási, Réka Albert: Emergence of Scaling in Random Networks, Science, Vol. 286, No. 5439, pp. 509–512 (1999)

ⓒ 2015 Information Processing Society of Japan