社団法人　情報処理学会　研究報告
IPSJ SIG Technical Report

2003－ＱＡＩ－ 9 (11)
2003／10／8

OSPFネットワークおけるトラヒック情報の可視化手法の実装

吉田　薫† 永見 健一†† 江崎 浩†

† 東京大学 大学院 情報理工学系研究科　〒113-8656 東京都 文京区 本郷 7-3-1
†† 株式会社インテック・ネットコア　〒136-0075 東京都江東区新砂 1-3-3
E-mail: †kaoru@hongo.wide.ad.jp, ††nagami@wide.ad.jp, †hiroshi@wide.ad.jp

あらまし　インターネットの急速な発展に伴い、ネットワークトポロジはますます複雑化してきている。その結果、トラフィックの急激な増大によって局所的に輻輳が発生したり、その逆にトラフィックの流れないリンクができるという問題が起きている。本研究では、ネットワークトポロジを可視化し、トラフィック情報を付与することで、ネットワーク資源が有効活用されているかの監視を行うことを目的としている。また、効率的に運用されていない場合にはそれに対する改善手法を提案する。
キーワード　OSPF, トラヒック, トポロジ, 可視化

Implementation of Traffic data visualization on OSPF Network

Kaoru YOSHIDA†, Kenichi NAGAMI††, and Hiroshi ESAKI†

† The University of Tokyo, 7-3-1 Hongo Bunkyo-ku, Tokyo 113-8656, Japan
†† Intec Netcore, Inc. , 1-3-3 Shinsuna Koto-ku, Tokyo 136-0075, Japan
E-mail: †kaoru@hongo.wide.ad.jp, ††nagami@wide.ad.jp, †hiroshi@wide.ad.jp

Abstract　According to the rapid growth of the Internet, the Network Topology of the Internet itself and of the AS (Autonomous System) has become more complex than ever. Also, due to the sudden bursty and high bandwidth traffic, the network is sometimes delivered unexpeted congestions within their networks, or some link(s) in their network does have very low usage. Many ISPes and campus network operators have struggled to run their network more efficiently, so that their network has fewer congestions (high load links/nodes) and fewer low load links/nodes. In this paper, we focus on the monitoring and visualizing the network topology and the resources utilization in thier AS. The proposed system can take care both layer 3 boxes (i.e., routers) and layer 2 boxes (i.e, switches).
Key words　OSPF, Traffic, Topology, Visualization

1. はじめに

ブローバンドなインターネット環境の家庭への普及が急速に進展しており、その結果個人が映像や音声のような広帯域で高速なディジタル通信行うことができるようになった。このような広帯域のディジタル通信を収容しサービスするために、インターネットのバックバーンではギガビット単位でのパケット転送を行うためのネットワーク基盤の整備が進められている。インターネットはネットワークのネットワークであり、特にバックボーンネットワークにおいてはトポロジは単純なスター状ではなく、複雑な網の目状の構成となっているのが一般的である。複雑な構成下において、データを送信元の計算機から、送信先の計算機へと正しく配送する機能のことをルーティングまたは経路制御と呼ぶ。経路選択 [1] の方式としては RIP(Routing Information Protocol) や OSPF(Open Shortest Path First)、IS-IS(Intermediate System-to-Intermediate System) などが実際のインターネットにおいて適用されている。これら、インターネットにおいて現在適応されている経路制御方式は、各ノードが、隣接するノードから転送される制御情報を基に、自律的に IP パケットがの転送経路を決定している。経路制御では、正確には、IP パケットが次に転送されるべきノードの決定のみを行なうためネットワーク全体の経路を把握しているわけでは、必ずしもない。その結果、あるデータ通信に着目し、IP パケットが具体的にどのような経路を経由して目的の計算機に転送されるか/転送されているのかを正確に把握することは、現実には、容易ではない。その結果、ネットワークの運用管理者が想像していない場所においてトラヒックの輻輳が起こってしまい効率的なネットワーク運用が困難になってきてしまっている。また、ネットワーク内にあるリンクに障害が発生すると、他のリンクに負荷がかかることになるが、どの程度の負荷を生じるかということを正確に把握することできず、最悪の場合には連鎖的な輻輳を起こすなど、ネットワークに混乱を来たす場合もある。

このような問題を回避、解決し、効率的なネットワークの運用を行う手法のことをトラヒックエンジニアという。トラヒックエンジニアリングを実現する方法としては、コンストレイントルーティングという手法を用いることにより、通常の選択される経路とは異なるものを明示的に指定する方法や、MPLS(Multi Protocol Label Switching) を用いた経路制御で選択される経路以外を LSP(Label-Switched Path) によって設定する方法などが代表的である。これらの他にもトラヒックエンジニアリングを実現するための方法が数多くなされているが、決定的な解決法は未だ見出されておらず、継続的な研究開発と実システムを用いた実証的評価検証が進められている状況である。本研究では、レイヤ 3 における経路決定方法の一つである OSPF というプロトコルを用いているネットワークにおいてネットワーク状態を可視化し、より効率的にネットワーク状態を把握し、ネットワークを管理する事を支援するシステムの研究開発を行なう事を目的としている。

本論文の構成としては、2. 章において、本研究が対象としている OSPF の説明をし、本研究に関連のある研究について述べる。3. 章において、本研究で用いるツールの説明を行う。4. 章では、3. 章で説明したツールをより有用なものとするのに必要なシステム要求について述べる。

2. 関連研究

本章では、まず本研究で着目してるルーティングプロトコルである OSPF の説明を簡単に行う。ついで、ネットワーク上を流れるトラヒックを収集するための既存ツールの代表的な例を挙げる。最後に、ネットワーク可視化に関する既存研究の代表例について述べる。

2.1 OSPF

OSPF [2] は AS(注1) 内部で使用されている IGP(Interior Gateway Protocol) の一つであり、RFC2328 [3] においてバージョン 2 が定義されている。大規模なネットワークに適用可能な仕様となっている。OSPF の特徴としては以下の 4 点が上げられる。

- リンクステートアルゴリズムを採用し、きめ細かなネットワークトポロジの定義と表現
- フラッディング手法を用いて迅速な収束性
- コスト (メトリック) に基づいた動的な経路制御
- エリアの概念を用いた高いスケーラビリティーの実現

2.1.1 リンクステートルーティングプロトコル

OSPF はリンクステートアルゴリズムを使用した経路制御プロトコルであり、各ルータはリンクステートを隣接ルータに広告する。リンクステートにはルータが接続しているリンク (e.g. イーサネットのネットワーク) の属性や、そのリンクのネットワークアドレス、コストなどの情報が含まれている。各ルータは任意のルータがどのリンクにどのように接続されているかという情報をリンクステートより知る事ができる。各ルータはこのリンクステート情報をもとに AS 内におけるネットワークの形状と経路を把握する。リンクステート情報データベースから各ルータは Shortest Path First アルゴリズムを用い、自分自身からネットワーク内に存在する全ての計算機への最短パスから生成される経路情報を生成する。生成された情報は木構造をなし、スパンニングツリーと呼ばれる。そして、その経路情報よりルーティングテーブルを作成する。各ルータがネットワークの構成を把握しているので、変化が起きた際に素早くルーティングテーブルを再構築する事が可能となる。

2.1.2 メトリック

メトリックとは、宛先ネットワークとの距離を表すパラメータである。本研究においてはコストをルータ間の重みとし、メトリックを送信元から宛先までのコストの総和及び複数ホップのコストの和として定義する。同じネットワークの経路情報が複数方向から来た場合、ルータはメトリックの小さい経路の方を近い、すなわち良い経路であると判断し、そちらへのパケットの転送経路をその宛先ネットワークへのパケットの転送経路として選択する。コストはネットワーク管理者が各ルータのイ

(注1) : AS:ISP や学術組織など同一のポリシで運営されている組織

ンターフェイスで設定する。コストはリンクステートがルータのインターフェイスを通過する際に足されていく。経路計算にはダイクストラアルゴリズムが用いられる。

2.2 トラヒック情報の取得

トラヒック情報の取得方法には様々なものがある。本節ではまずネットワークを遠隔から管理する際に広く用いられているSNMP について説明する。次いで、近年普及し始めたサンプリングによるフロー情報収集プロトコルである NetFlow,sFlow について述べる。

2.2.1 SNMP

SNMP とは、Simple Network Management Protocol の略であり、ネットワークに接続された機器類をネットワーク経由で監視するためのプロトコルである。RFC1157 [4] でバージョン 1、RFC1902～1906 [5] でバージョン 2 が RFC3410 [6] でバージョン 3 がそれぞれ定義されている。SNMP は複雑で大規模なネットワークを人間の手で管理せず、自動的に管理情報を収集設定することで、管理者の負担を軽減することを目的として考案されたネットワーク管理プロトコルである。SNMP の提供する基本機能を用いて以下のようなネットワーク管理、監視機能 (タスク) を実現することができる。これらの情報から、表 1 に示すのようなネットワーク管理、監視機能 (タスク) を実現することができる。

表 1 SNMP の主な機能

タスク	機能
障害管理タスク	ネットワーク上で発生した障害を検出するためのタスクであり、障害が検出された場合に、クライアントプロセスに対して障害の通知を行なう。
アカウント管理タスク	ネットワークの使用状況を監視するためのタスクであり、ネットワークへのアクセス方法や用途、使用頻度や使用量などの情報を収集し、統計情報をクライアントプロセスへ通知する。
通信機器管理タスク	ルータなどの通信機器が正常に動作しているかの監視を行なったり、制御するためのタスクである。
パフォーマンス管理タスク	ネットワークの負荷の状況を監視するためのタスクであり、ネットワークの負荷の情報を収集し、統計情報として、ユーザに通知する。
セキュリティ管理タスク	ネットワークに接続されている計算機やルータなどの通信機器へのアクセスを制御し、不正な侵入者の検知を行ない、検知した場合にはユーザに対して通知する。

一般に SNMP においては複数のエージェントと一台のマネージャが存在している。エージェントとしては例えばルータなどが挙げられ。マネージャは複数のエージェントから渡された情報を一元的に収集し、管理することができる。本研究では上記の機能群の中の主に障害管理タスク及びパフォーマンス管理タスクの部分を用いる。

ネットワーク管理者などは、snmpget,snmpwalk などというアプリケーションプログラムを用いる事でネットワーク上にある機器の管理情報を収集する事が可能となっている。

2.2.2 NetFlow

NetFlow は、現在 IETF IPFIX ワーキンググループ [?] で議論されている規格であり、バージョン 9 まで定義されている。バージョンによって収集されるデータが異なるが、ルータやスイッチ上を流れるパケットから送信元、宛先 IP アドレス、送信元、宛先ポート番号、パケット数などの情報を収集することができる。ルータやスイッチなどで得られたデータを、定期的にデータ集積用の計算機に送信し、蓄積されたデータを解析する事でネットワーク上のフロー傾向を知る事が可能となっている。既に、CISCO Systems 社や Juniper Networks 社によって対応した製品が出されている。。

2.2.3 sFlow

sFlow [8] は、InMon 社が RFC3176 で定義している技術であり、「スイッチとルータを含むデータネットワークでトラフィックをモニターする技術」である。サンプリングをハードウェアレベルで処理しているので、負荷を小さくできると言う特徴を持っている。スイッチやルータのインターフェイスにパケットが到着するとスイッチ／ルーティング機能により、宛先インタフェースが割り当てらるが、この時無作為にサンプリングされる。このサンプリングされたパケットのデータを基に、そのルータ、スイッチを通過したトラフィックの送信元と送信先の統計情報を生成する技術である。Foundry NETWORKS 社などによって製品が出されている。

2.3 ネットワークの可視化

ネットワーク状態を可視化するためのツールは現在すでに様々なものが出ている。一例としては、caida [9] が提供している skitter というツールが上げられる。skitter によって視覚化されたネットワークの図を図 1 二示す。このツールは以下の 4

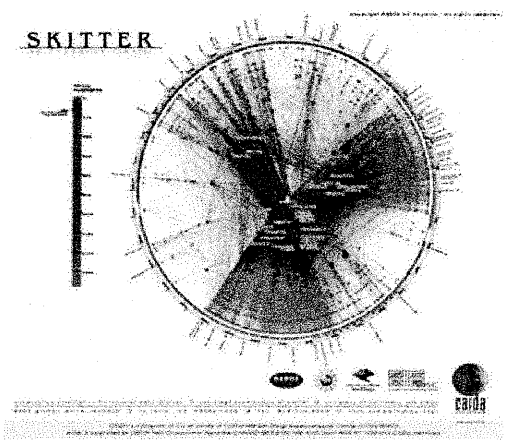

図 1 skitter によるネットワークトポロジ図

つの特徴を持っている

（1） IP による転送経路の計測

ある送信元から宛先へ至る経路上で経由する計算機から

（2） round trip time(RTT) の計測
任意の宛先に対して ICMP(Internet Control Message Protocol) 要求を行い、その宛先に到達するまでにかかる時間の測定を行っている。

（3） 経路の変化を追跡
定常的に測定しているデータから RTT から経路変化が配送中か、応答中かの識別をすることができる。

（4） ネットワークの接続状態の可視化
ある送信元から多くの宛先への経路を有向グラフ化する事で、接続状態を可視化する事ができる。

3. OSPF ドメインにおけるトラヒックの可視化システム

本研究では、OSPF ネットワークをレイヤ 3 レベルで可視化することのできるを拡張したツール [10] に実装した。本章ではこのツールの機能説明を行う。このツールの特徴としては以下の 4 点が上げられる。

- OSPF ネットワークのリンク状態の可視化
- 各リンクを流れるトラヒック量の表示
- OSPF ネットワーク上でのフローシミュレーション
- レイヤ 2 スイッチ概念

3.1 OSPF ネットワークのリンク状態の可視化

2.1.1 節で述べたように、OSPF ネットワークにおいては、OSPF により経路情報の交換を行っている計算機は全て同一経路情報を共有している。こうした OSPF の特性を利用して本ツールでは、AS 内の一つの計算機に対して snmpwalk を実行することにより、その計算機が保持している OSPF の LSA(Link-State Advertisement) 情報を収集し、ファイルを生成する。

そして、得られた LSA ファイルを基に、図 2 のように OSPF を利用している各ルータ間のリンクの情報、そのネットワークの構成を視覚化することができる。情報を収集し、表示するツールは Perl/Tk で実装されている。

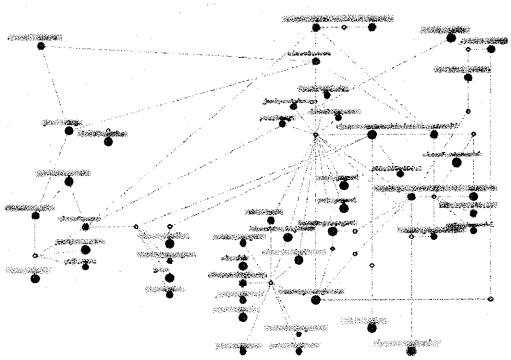

図 2 WIDE インターネットにおけるルータ間のリンク状態

これにより、OSPF の転送経路としてどの計算機と計算機が接続しているか、どの計算機群が同一ネットワークセグメント内に存在しているかという情報を視覚的に把握する事が可能となる。

3.2 各リンクを流れるトラヒック量の表示

- 定期的に snmpwalk を行い、各リンクの帯域幅情報、トラフィック情報の取得
- 取得した情報から帯域幅に応じた線の太さ
- この時、各リンクには入力と出力があるので、有向グラフにする
- 帯域占有率に対して段階的な閾値を導入することにより、各リンクの利用率 (負荷率) の視覚化

OSPF を利用してる全てのルータに対して snmpwalk を実行することにより、各計算機のネットワークインターフェイスのインターフェイス名、帯域幅、入出力のデータ量を収集する。これにより、snmpwalk に対してルータが応答しない場合にはそのルータ自身が落ちていることがわかり、snmpwalk の結果よりインターフェイスがダウンしていれば、そのインターフェイスからでているリンクのみが落ちていることがわかる。また、データを正常に収集できた場合には各リンクの帯域幅及び負荷率という情報を得ることができる。実際に、WIDE インターネットバックボーンネットワークにおいて、測定した結果を図 3 に示す。この測定ではネットワーク上の計算機全てではなく一部の計算機に対してのみ、snmpwalk を実行した。トラフィックの測定を行ったリンクが黄緑色に、測定を行っていない場所が灰色になっていることが分かる。また、測定を行った場所が全て黄緑色であり、帯域にまだ十分な余裕がある事がわかる。

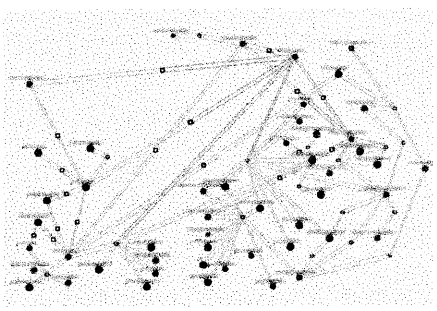

図 3 WIDE インターネットの各リンクの帯域幅及びトラフィック量

3.3 OSPF ネットワーク上でのフローシミュレーション

本ツールの大きな特徴として一つの計算機からそれを送信元とするネットワーク内の全ての計算機への経路を表示することが可能である。これは、ある計算機からネットワーク内の全ての計算機に対してダイクストラアルゴリズムによる最短パスを計算することにより得られる。この機能を利用し、送信元と宛先が選択された際にその経路の探索を行う。次いで計算され選択された経路上にある各リンクに対して指定されたフローを注入する。これを複数回繰り返すことにより、ネットワーク上に実際に近い形でのトラフィックフローを再現することが可能となっている。そして、送信元から送信先に流れているトラフィックの情報を、その経路上にあるリンクに付加していくことで各

リンクに流れているトラフィックの量を知ることができる。また、コスト変更を行った場合にも、ネットワーク内を流れる全体のトラフィック量に変化はないと考え、経路を再計算し、その後選択された経路上の各リンクに対してトラフィックを付加していくことにより、変更後のネットワークをシミュレーションすることが可能となる。

各リンクの負荷率に対する閾値を設定可能とすることにより、同じ帯域幅を持っているデータの転送能力の優れている計算機と転送能力の劣る計算機があった場合に、それらに対する注意を払う度合に差をつけることができるようになっている。

シミュレーションの結果を図4に示す。

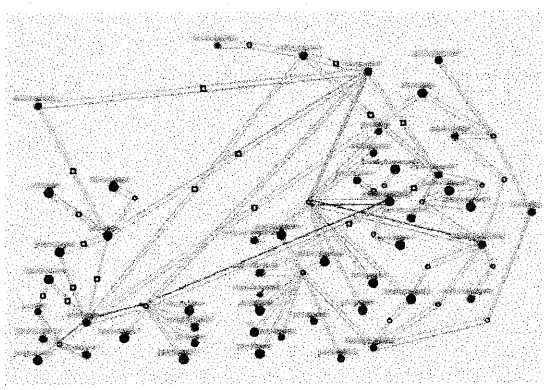

図4　コスト変更前のフロー

これに対して、ネットワーク内においてトラフィックは変化させずに、一箇所のコストを変えた場合の結果を図5に示す。

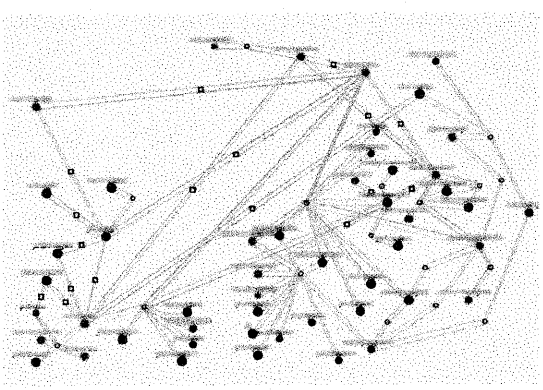

図5　コスト変更後のフロー

このように、コスト変更によって生じるフローの変化を知ることができるようになっている。また、それによる影響を視覚的に把握することができる。輻輳が複数リンクにまたがり連続的に発生している場合には、どのフローが影響しているかということも容易に推測可能となる。

現状ではネットワークのトラフィック情報の収集はsnmpwalkで行っており、各インターフェイスを流れるトラフィックの量を知ることはできるが、ネットワーク全体でどのようなフローが存在しているかということを知ることはできない。つまり、コストを変更した場合にネットワーク全体に与える影響をシミュレーションすることができない。実際のネットワークを再現しようとする際には、各ルータを流れるフローがどの送信元から送信先に向けられているであるかということを知る必要が出てくる。ここで、2.2.2節や2.2.3節で述べたNetFlowやsFlowなどを出力可能なルータやスイッチを複数箇所に設置する事により、フローがどの送信元から送信先へ向けられているものであるかを概算として把握することが可能となると考えられる。この情報を基にネットワークの状態をシミュレータ上に再現し、もし問題のある部分が存在していれば、シミュレータ上でコストの変更などを行い影響を確認することができるようになると考えられる。また、これは実ネットワーク上で作業を行うよりも安全に行うことができる。

3.4　レイヤ2スイッチ概念

レイヤ2スイッチの概念を導入することにより、OSPFからのみでは判断する事ができなかった物理インターフェースとIPアドレスとの対応を取れるようになる。これは例えば、VLANなど一つの物理インターフェイスに複数のIPアドレスが割り当てられている場合や、一つの物理インターフェイスに複数のIPアドレスが割り当てられている場合などに実際のネットワークトポロジを知る上で必要であると考えられる。そして、それをトポロジ上に再現する事により、ネットワーク機器がどのように接続されているかという情報をより詳細に把握可能となる。

4.　考　　察

3.4節では、既存のOSPFネットワーク可視化ツールの拡張を行い、新たにレイヤ2を考慮したトポロジ描画という概念を導入した。これにより、今までのものに比べよりきめ細かくネットワーク状態を可視化する事が可能となった。しかし、VLANなどのように、SNMPを用いて外部から情報を参照しても、十分な情報を取得することのできないものが数多く存在する。それらのネットワーク情報を取得するためには、現状ではネットワーク管理者が個々の計算機やスイッチ、ルータに入り、その設定を読み取らなければならない。そのため、大規模なネットワークを運用しているISPなどでは、こうした作業を行う事は事実上不可能であり、行おうとすればネットワーク管理者に多くの負担を強いる事になる。現状でそうした設定情報を自動で取得するのは困難であり、情報を一箇所に蓄積する事ができず、視覚的に捉える事ができない。

本章ではこうした事実を踏まえ、ネットワーク管理者に負担をかける事なく、より詳細な可視化されたネットワーク状態を提供するのに必要なシステム要件について検討する。

ネットワークを管理する際、障害が発生した際には迅速に対応する必要がある。つまり、利用されるツールは障害を迅速に把握することが可能で、その障害地点を容易に特定する事が可能である必要がある。

つまり、以下の2つ要件を満たす必要がある。

（1）　より実ネットワーク即したネットワーク図を描画する

事で、直観的な管理を可能とする。

（2） より詳細な設定情報を収集し、反映可能とする。例えば、VLAN などが上げられる。

（3） より詳細なネットワーク情報を可視化することにより、障害発生時などに原因、場所の特定を容易にし、迅速な対応を可能とする。

3. 章で述べたツールは Perl で実装されている。Perl にモジュールを導入する事で、ネットワーク越しに他の計算機に入り、作業する事が可能である。このモジュールを用いる事でネットワーク上にあるルータやスイッチにアクセスし、それぞれの計算機上の設定情報を収集する事ができるようになると考えられる。まず、モジュールを用いる事で情報を十分に収集することができるか検証することが第一の課題である。

ついで、管理者がネットワークを効率的に管理するには、収集できた情報をどのように可視化されることが望まれているかということを検証し、その実装を行う事が第二の課題である。

本研究では、OSPF ネットワークおけるネットワーク状態の可視化を行なった。これにより、ネットワーク上の問題の認識/把握をより容易かつ正確に行なうことが可能となり、それに対する対策を講じるやすくなるようまシステム管理ツールを、ネットワーク運用者に提供することができたと考えられる。

5. おわりに

本研究は、インターネットのルーティングプロトコルとして広く運用されている OSPF に着目し、そのネットワークの状態を可視化する事を目的とした。可視化することにより、ネットワーク上の問題の認識を容易にし、対策を講じやすくする事ができるようになったと考えられる。

今後は 4. 章で述べた考察及び今後の課題について再度十分な検証を行うことが上げられる。ついで、それを元にシステム設計、実装を行い、実ネットワークにおいてその拡張性、規模性の評価を行う予定である。

文　献

[1] 友近剛史, 池尻雄一, 小早川知昭, “インターネットルーティング入門 第一版”, 翔泳社, 2001.

[2] John T. Moy, “OSPF Anatomy of an Internet Routing Protocol”, Addison Wesley, 1998.

[3] J.Moy, “RFC 2328 OSPF Version 2”, RFC2328, April 1998.

[4] J.Case, M.Fedor, M.Schoffstall, J.Davin, “A Simple Network Management Protocol (SNMP)”, RFC1157, May 1990.

[5] J.Case, K.McCloghrie, M.Rose, S.Waldbusser, “ Structure of Management Information for Version 2 of the Simple Network Management Protocol (SNMPv2)”, RFC1902, January 1996.

[6] J.Case, R.Mundy, D.Partain, B.Stewart, “Introduction and Applicability Statements for Internet Standard Management Framework”, RFC3410, December 2002.

[7] IETF IP Flow Information Export WG (ipfix), http://www.ietf.org/html.charters/ipfix-charter.html.

[8] P. Phaal,N. McKee, “RFC 3176 InMon Corporation's sFlow: A Method for Monitoring Traffic in Switched and Routed Networks”, RFC3176, September 2001.

[9] cooperative association for internet data analysis, http://www.caida.org.

[10] 吉田 薫, 江崎 浩, “大規模 OSPF ネットワークにおけるトラフィックエンジニアリングに関する研究”, 2002 年度東京大学工学部電子情報工学科卒業論文, 2003.