研究ノート

RIPパケットを用いた論理ネットワーク状況視覚化

佐井慧彦*・森口一郎**

要旨：本研究では，RIPとtracerouteを利用し論理ネットワーク図を自動で出力するシステムを構築した．本システムは，まずRIPパケットをキャプチャしネットワークアドレスやサブネットマスク，hop数を抽出する．抽出後，ネットワークまでに経由するルータアドレスをtracerouteで取得する．そして，ルータアドレスのネットワーク所属情報をもとに論理ネットワーク図を自動で出力する．出力されたネットワーク図はWebブラウザ上でユーザに公開し，ユーザが図のルータをクリックしたとき，MRTGで作成したトラフィックグラフを表示する機能も実装した．

キーワード：RIP，SMNP，MRTG，traceroute，論理ネットワーク図

Visualization of the logical network situation using RIP packets

Akihiko SAI* and Ichirou MORIGUCHI**

Abstract: In this study, we developed a system that automatically outputs logical network diagrams using RIP packets and traceroute. First, the proposed system captures routing information protocol（RIP）packets and extracts the network addresses, subnet masks, and hop counts from them. Second, the system collects the router addresses that were passed to the network based on the traceroute results. Then, it automatically outputs logical network diagrams considering the network group information of router addresses. These network diagrams are provided on a Web server, which also display traffic graphs constructed by multi router traffic grapher（MRTG）when users click routers on them.

Keywords: RIP, SNMP, MRTG, traceroute, logical network diagram

* 東京情報大学　総合情報学部　総合情報学科（2024年３月卒業予定）
株式会社インターネットイニシアティブ（IIJ）（2024年４月より）
Faculty of Informatics, Tokyo University of Information Sciences
Internet Initiative Japan Inc.
** 東京情報大学　総合情報学部　総合情報学科
Faculty of Informatics, Tokyo University of Information Sciences

2023年10月５日受付
2023年１月17日受理

1．はじめに

近年，デジタル機器の増加や携帯電話など通信端末の普及［1］によって，通信は日常生活に必要な要素として成長し［2］，ネットワークを構築する機会も増え，管理の重要性も高まった［3］．しかし，ネットワーク管理は正常に通信ができているかの監視を日常的に行う必要があるほか［4］，ネットワーク構築時，ネットワークを直感的に把握するためケーブルの配線をまとめた物理ネットワーク図，データの流れをまとめた論理ネットワーク図を作成しなければならない．しかし，通信が生活インフラとなった今，ネットワーク管理者は少人数である場合が多く，負担が大きいため，これは大きな問題といえる．また，管理者でない組織内ユーザは通常，ネットワークがどのような状態，構造であるかの確認ができない．

これらの問題に対し，秋田，永田，谷口らは，SDN（Software Defined Network)［5］の1つであるOpenFlowを各ルータに実装し，ネットワークの接続状態やトラフィックを取得し，これらをサーバに保存しWebサービスとしてユーザに提供する研究を行った［6］．しかし，管理者が前もってOpenFlowに論理ネットワーク情報を与えなければならないほか，OpenFlow技術を使用しなければネットワークの状態を自動取得することができない問題がある．また尾形，貫井らはtracerouteの結果からネットワークトポロジーを出力する研究を行ったが［7］，SNMP［8］を使用しておらず，トラフィック情報などのネットワーク状態を取得していない．SNMPを利用したネットワーク構造取得研究には太田，神宮らのSNMP対応スイッチを利用しローカルネットワークの階層構造を出力する研究や［9］，辻本，藏内，井口らのSNMPでMIB-Ⅱに登録されているOSFP-LSDBなどのルーティング情報を取得しその情報を参考にネットワーク構成を出力する研究がある［10］．しかし，これらSNMPに依存したネットワーク構造取得はシステム実行者がルータの管理者であり，SNMPで情報を要求する許可がないと構造推定に必要な情報を取得することができないほか，管理者が管理範囲をシステムに入力しなければならない．

これらの問題に対し，本研究ではシステム実行者が管理しているしていないに関わらず，ネットワークを解析し自動的に論理ネットワーク図作成を行うためにRIPパケットをキャプチャし，パケット内に記載されている各ネットワークの端末にtracerouteを実行してその結果を参考に論理ネットワーク図を自動で出力するシステムを構築した．出力したネットワーク図は先行研究［6］と同じようにWebサービスとしてユーザに提供し，さらに提供した図のルータがユーザにクリックされたときそのルータがシステム実行者の管理しているルータであるならSNMPで取得したトラフィック情報とCPU使用率も表示することでネットワーク通信状態も提供した．また，複数の実行サーバで作成した論理ネットワーク図を1つの図にするために，2つのネットワーク情報のルータを比較し，同じIPアドレスを所有していた場合と，属しているネットワークを集合の1要素としたとき包含関係にあった場合は，同じルータと判断しまとめる機能も実装した．

本システムを仮想環境で構築したネットワークで実行した結果，スター型のネットワークであればシステム実行者の管理外のネットワークであっても正しい論理ネットワーク図を作成できることが確認できた．また，複数の実行サーバを設置する場合は，部分的であるがループが確認できる論理ネットワーク図を作成することができた．しかし，すべてのネットワークをRIPで公開していない管理外ネットワークの場合，一部のネットワークしかグラフ化できないことも判明した．

2．本システムの重要プロトコル，ツール

先行研究ではネットワーク通信状況の把握と，管理外ネットワークのグラフ化が問題点であったが，本システムはRIPとMRTGを利用して解決している．

2.1　RIP

ルーティングプロトコルの1つであるRIP（Routing Information Protocol）はUDP520番ポートを使用し，version1ではブロードキャスト，version2ではマルチキャスト通信で各ルータが所有しているネットワーク情報を他のルータと共有する［11］．このRIPにはクラスアドレスを想定して設定したバーション1とサブネットマスクを考慮したバージョン2があり，今回は現在主流のバージョン2のみを想定して本システムを構築した．

RIPバージョン２のデータフォーマット（図１）のoperation部は通信の段階を表しており，送信は0x01，受信は0x02が格納される．address family部は認証機能，route tag部は他のルーティングプロトコルとの連携のために使用される．network address部はルータが持っているネットワーク情報のネットワークアドレス，subnet mask部はそのサブネットマスク，next hop部は対象のネットワークへ送信する場合に次に経由させるルータのアドレス，distance部はネットワークの距離が挿入されており，RIPの場合hop単位で表現している．これらnetwork address部からdistance部はネットワークごとに１セットとなっており，送信元のルータが把握している情報分追記される．

図１　RIPv2のフォーマット

2. 2　MRTG

MRTG（Multi Router Traffic Grapher）[12]はルータのトラフィック状態をSNMP通信で取得し，トラフィック状態変動をPNG画像形式で生成するツールである（図２）．本研究ではSNMP通信が許可されているルータからインターフェース情報を取得し，論理ネットワーク図を正確に把握，トラフィック状態を表示するために使用した．

ネットワークを介して機器を管理するプロトコルのSNMP（Simple Network Management Protocol）は，機器の情報をツリー構造で表したMIBツリー形式[13]で要求内容を記し，リクエストする．そのため，MRTGのMIBツリー関連設定を変更することでCPU使用率やメモリ状態もグラフ化できるため，本システムはトラフィック状態のほかにCPU使用率もグラフ化するように設定した．

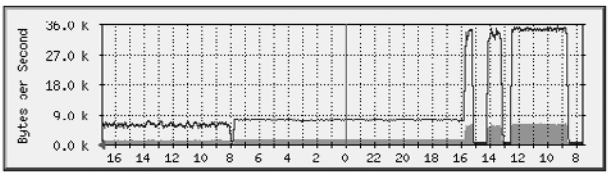

図２　MRTGで作成したトラフィックグラフの例

３．本システムの概要

RIPとtracerouteを使用し図を出力する機能と，複数のグラフを１つにまとめ，MRTGにより作成されたトラフィックグラフと１つにまとめたグラフをWebサービスとしてユーザに提供する機能を実行サーバ上に構築した．

3. 1　本システムの構成要素

本システムはネットワーク情報出力，グラフ情報出力，グラフ提供CGIのように機能を３分割してプログラムを構築した．

ネットワーク情報出力プログラムは実行サーバが属するネットワーク内に流れているRIPパケットをキャプチャしネットワークの存在を把握する．キャプチャ後，把握したネットワークに通信可能な機器を探すホストスキャンを行い，発見できた任意の端末にtracerouteを実行し，その実行結果を参考にネットワーク接続情報を解析する（図３）．解析後，tracerouteで発見できたルータとMRTG設定ファイル群を照合し，情報を修正して接続情報として出力する．グラフ情報出力プログラムでは，実行機器から出力された接続情報と別の実行サーバとの通信で取得した複数の接続情報をマージし，グラフ情報を出力させた．そしてユーザがネットワーク図を要求した際にグラフ提供CGIがグラフ情報をもとにネットワークグラフを出力し，ユーザに提供する（図４）．

今回，接続状態把握時に実行するパケットキャプチャはWireshark，ホストスキャンはNmapで実装し，ネットワーク情報出力プログラムとグラフ情報出力プログラムおよび情報通信プログラムはすべてJavaで構築し，グラフ提供CGIはシェルスクリプトとvis.js［14］で作成した．

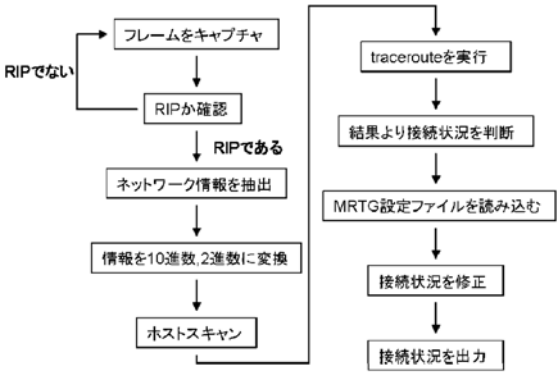

図３　接続状態取得のための処理の流れ

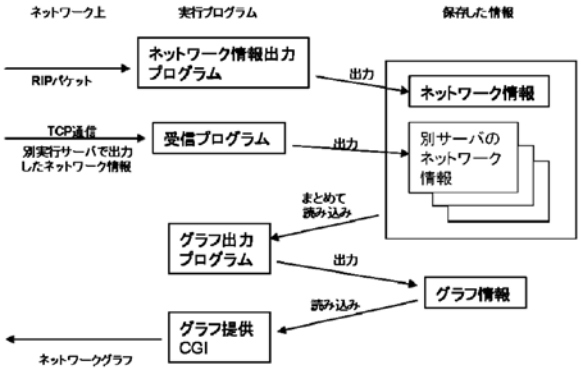

図４　システム全体の処理の流れ（情報送信は省略）

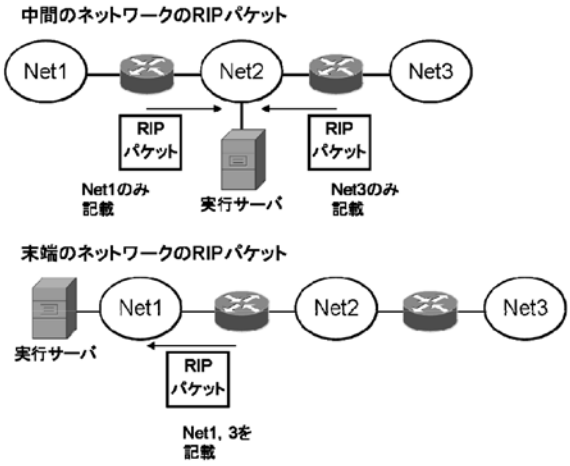

図５　中間ネットワークと末端ネットワークのRIPパケット

このように実行サーバはRIPパケットをキャプチャできれば論理ネットワーク構造取得処理を開始するため実行サーバの配置場所は任意である．しかし実行サーバが属するネットワーク内に複数のRIPパケットが流れている場合，それぞれのパケットに記入されていないネットワークが存在する．本システムはRIPパケットでネットワークグラフ化範囲を判断しているため，まとまったネットワーク情報が流される末端ネットワークで本システムを実行することが推奨される（図５）．

3. 2　tracerouteによる接続状況把握

本システムではRIPパケットからネットワーク情報を抽出し，把握できたネットワークに属する機器に対しtracerouteを実行して，その結果から経由したルータのアドレスを入手する．このようにして取得したアドレスがどのネットワークに属しているかを判断することで，対象のネットワークまでに経由したネットワークを知ることができ，tracerouteの表示順を参考に並べることで経由したネットワークの順番を知ることができる（図6）．例えば10.0.0.0/8から192.168.0.0/16の端末に向けてtracerouteを実行したときにtracerouteの結果が図６のように10.0.0.1，172.16.0.1，192.168.0.1の順番で表示され，RIPパケットから10.0.0.0/8，172.16.0.0/12，192.168.0.0/16のネットワークが把握できているとする．tracerouteの結果のアドレスがどのネットワークに属しているかあてはめると，10.0.0.1は10.0.0.0/8，172.16.0.1は172.16.0.0/12，192.168.0.1は192.168.0.0/16となる．この結果とtracerouteの表示順より10.0.0.0/8から192.168.0.0/16への接続状況は10.0.0.0/8，10.0.0.1，172.16.0.0/12，172.16.0.1，192.168.0.0/16となっていることが確認できる．この処理をすべてのネットワークに実行しネットワー

ク全体構造を出力した.

ルータにtracerouteを実行したとき，２通りの通常とは違う結果が出力される（図７）．図７①のように1hop前のネットワークに存在するアドレスが返答された場合は，実行サーバから近い方のルータのインターフェースが返答されたことになるため，目標のネットワークの1hop前にそのアドレスのネットワークが存在すると判断できる．図７で例えると，192.168.0.1にtracerouteを行った結果，最後に172.16.0.1が返答され，192.168.0.0/16の一つ前に172.16.0.0/16があると判断できる．また，図７②のように1hop前のネットワークがtracerouteの結果に記入されていない場合，実行サーバが属する10.0.0.0と目標のネットワーク192.168.0.0の間にネットワークが存在しないと判断され，正確な接続状況を取得することができない．そのため1hop前のネットワークが記入されていないtracerouteの結果は参考にせず，同ネットワークの別のアドレスにtracerouteを再実行する必要がある．通常n hop先のネットワークに対しtracerouteを実行したとき出力される結果はn+1個のアドレスが表示されるため（図８①），図８②のようにn hop先の端末に実行したに関わらずtracerouteの実行結果がn個のアドレスの場合ルータに実行したと判断させた.

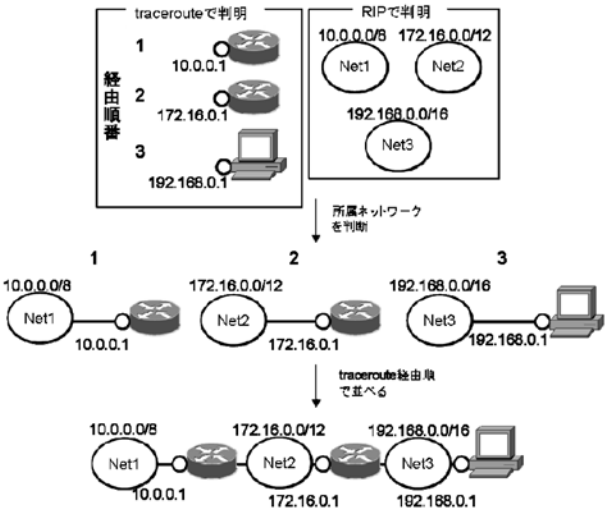

図６　tracerouteを使用した接続状態の取得方法

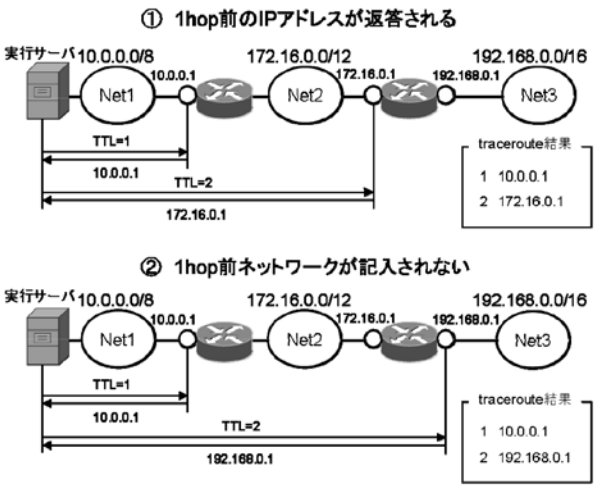

図７　ルータにtracerouteを実行した場合の返答

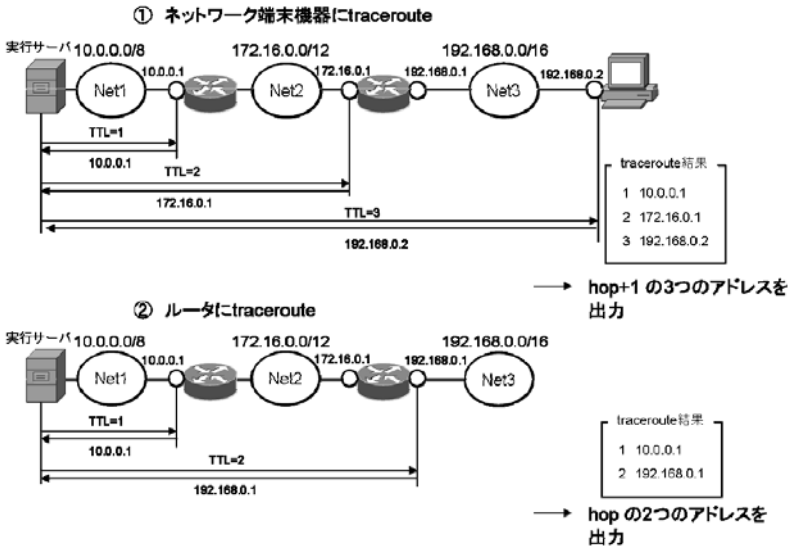

図８　2hop先のネットワークのルータと末端機器にtracerouteを実行した場合の返答

3.3　インターフェースの記録

tracerouteの実行結果は最後のアドレス以外は必ずルータのアドレスになるため，ルータのアドレスを記録しグラフに表示する機能も追加した．しかし，この処理のみでは末端のネットワーク側のルータのアドレスを把握できないため，ホストスキャンで取得したすべてのホストアドレスにtracerouteを行い，ルータに実行した場合その時に使用したホストアドレスをルータの末端側のアドレスとした．

本システムは出力されたアドレスの数でルータにtracerouteを実行したか判断しているが，ネットワーク部が途中まで同じネットワークが存在した場合は，ルータが所有していないアドレスがルータのインターフェースとして記録されてしまう場合がある．例えば図９の場合は，172.22.0.0/16に属するルータのインターフェースアドレスを172.22.1.0/24の端末のアドレスと誤認してしまう可能性がある．また，マスク値が小さいネットワークはホストアドレスが非常に多くアドレスの調査に時間がかかる場合もあるため，ネットワークのマスク値が一定以下（実験では24bit未満）の場合，ルータの末端側のアドレス調査は実行させないシステムにした．

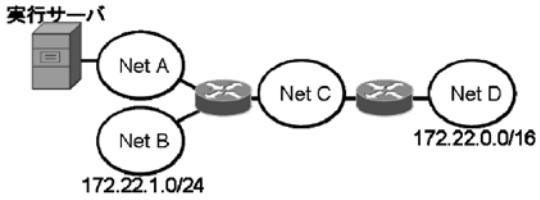

図９　インターフェース取得に失敗するネットワーク例

3.4　MRTGとの連携

本システムはMRTGを利用してシステム実行者が管理するルータのトラフック状態をグラフ化し提供するが，MRTGは管理対象のルータとSNMP通信ができる場合そのルータ用のMRTG設定ファイルを作成し，このファイルをもとにMRTGグラフを作成する．そのためMRTG設定ファイルを読み込み，提供したネットワークグラフでユーザがルータをクリックした時，MRTGグラフを表示させた．

MRTG設定ファイルには，ルータのインターフェースごとに設定が記入されており（図10），この設定のMRTG_INT_IPという項目にインターフェースアドレスが記入されている．このアドレスとtracerouteで発見したルータアドレスを照合することで，２つの情報の対応付けができる．

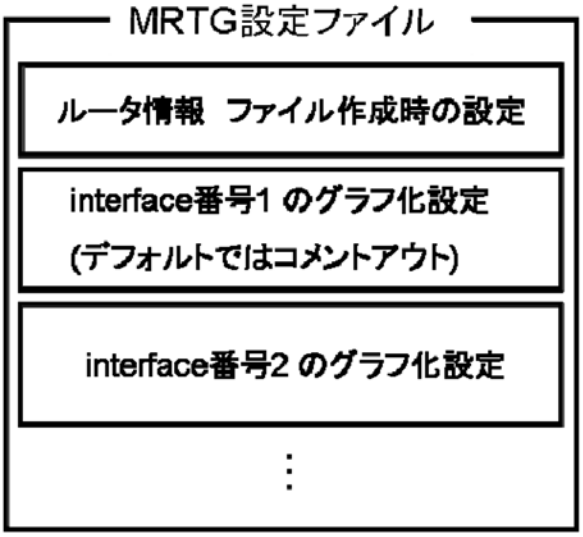

図10　MRTG設定ファイルのフォーマット

3.5　グラフマージ機能

ループが存在するネットワークではtracerouteが通過しない経路が存在し，正常にグラフを出力できないため，複数の実行サーバを別のネットワークにも設置することで片方のサーバでは把握できない経路を取得できる（図11）．この複数のサーバで出力した接続情報を１つにマージし，グラフ情報として出力する機能も本システムで実装した．

情報を１つに合わせるためには，同一のルータを特定し，１つのノードにまとめる必要がある．このため，グラフのルータ同士を比較し，同じインターフェースを所持している場合または所属しているネットワークを集合の要素として包含関係の場合は同じルータと判断した（図12）．例えば図12の場合，情報Aルータが Net1，Net2，Net3に，情報Bルータが Net1，Net2に属している．この属しているネットワークを要素として情報を集合化した場合，情報Aの要素はNet1，Net2，Net3，情報Bの要素はNet1，Net2となる．この２つの集合を比較し包含関係の場合は２つの情報のルータは同じと判断する．例の場合はこの関係が成立するので情報A，Bをマージし，１つのルータとなる．

このように複数のサーバを設置することでループを把握した論理ネットワーク図を提供することができる．しかし，先行研究［７］のようにtraceroute

は最短経路しか通過できないためサーバの設置場所によって把握できない経路が存在する．例えば図13のネットワークの場合Net1の他にNet7に追加で実行サーバを設置した場合それぞれの実行サーバで作成したグラフをマージすることで正しい論理ネットワーク図を提供することができる．しかしNet3またはNet4に追加で実行サーバを設置した場合，最短経路が図14のようになりループが把握できない．このようにサーバの設置場所によって結果が変わってしまうため複数の経路を通過するような末端ネットワークに実行サーバを設置することを推奨する．

なおシステムがSNMP通信によりインターフェース情報を取得することが可能である場合，その情報を参考に論理ネットワーク図を作成する．そのため実行者が管理している範囲のネットワークはループありなし関わらず１つのサーバでも正しくグラフ化できる．

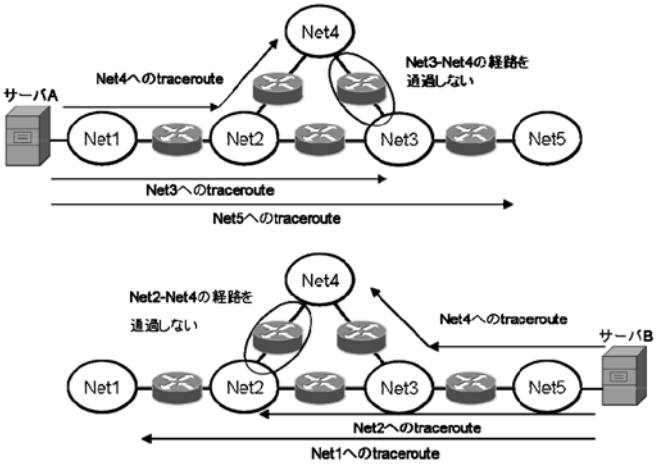

図11　tracerouteが通過しない経路があるネットワーク例

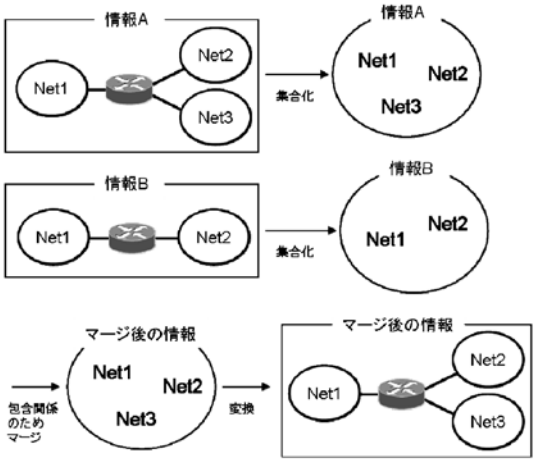

図12　包含関係によるマージ

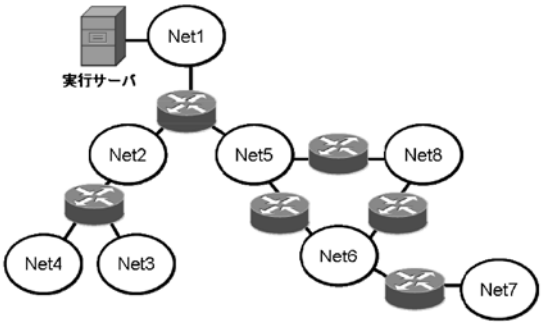

図13　論理ネットワークマージが失敗するネットワーク例

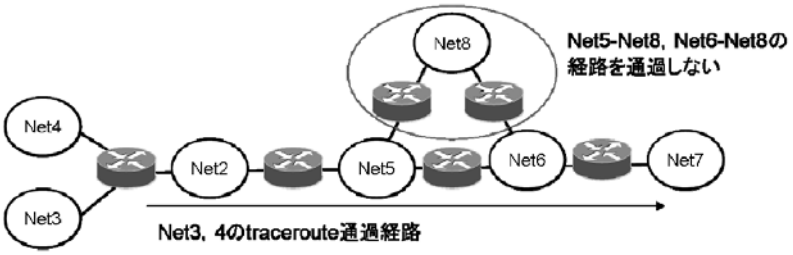

図14　図13内のNet3，4のtracerouteの経路（Net1は省略）

3.6　ネットワーク図提供CGI

前節までで出力したグラフ情報を使用し，JavaScriptとCGIを組み合わせてネットワーク図をWebブラウザ上でユーザに提供した．

WebブラウザはHTMLを解釈してユーザにコンテンツを提供するため，HTMLソースコード内でネットワークグラフを作成しなければならない．そのため，数的グラフを表示するためのJavaScript用の外部ライブラリであるvis.jsのネットワークグラフ専用版を利用しネットワークグラフを作成した．

visネットワークグラフはノード同士の接続状態が変化しないグラフとなっているが，実際のネットワークは動的に接続状態が変化する可能性がある．このため，前節までで出力されているネットワーク接続情報（グラフ情報）を参考にvisネットワークグラフを動的に出力するCGIをシェルスクリプトで構築した．

また，本システムはルータのMRTGグラフを表示させるため，MRTG設定ファイルを参考に，ノードクリック時に対応しているMRTGグラフを表示するようCGIを構築した［15］．

4．本システムの評価実験

本システムが正しく論理ネットワーク図を作成できるか実験を行い確認するため，VMware Workstation Player version 16.2.4でCentOS7をOSとして仮想マシンを複数生成し，仮想ネットワークを構築した．1つのホストマシン内で仮想ネットワークを構築しているため，仮想マシン間の通信速度は10Mbpsに制限した．また，VMware仮想環境にはルータ用のOSをインストールできないため，仮想マシンでも動作するソフトウェアルータであるQuaggaを利用してルータを構築した．

本システムはWebサービスでネットワーク図を提供するため，今回はWebサーバとしてApacheバージョン2.4.46をシステム実行サーバにインストールし，Webサービスを実装した．

また，本システムは提供するネットワークグラフにMRTGグラフも表示させるため，MRTGバージョン2.17.4も実行サーバにインストールした．トラフィックグラフはデフォルトの設定で作成したが，CPU使用率もグラフ化するため設定ファイルの最後の行に図15の内容を追記した．図10内の［ファイル名］は出力後のMRTGグラフの名前となっており，今回は［監視するルータアドレス］_cpuとした．

```
Target[[ファイル名]]: .1.3.6.1.4.1.2021.10.1.5.1&.1.3.6.1.
4.1.2021.10.1.5.2:[コミュニティ名]@[ルータアドレス]
MaxBytes[[ファイル名]]: 100
Options[[ファイル名]]: gauge,absolute,nopercent,noinfo
Title[[ファイル名]]: CPU load average
YLegend[[ファイル名]]: CPU Load
LegendI[[ファイル名]]: one min average
LegendO[[ファイル名]]: five min average
ShortLegend[[ファイル名]]: (%)
Legend1[[ファイル名]]: one min average
Legend2[[ファイル名]]: five min average
PageTop[[ファイル名]]: <H1>CPU Load Average</H1>
```

図15　CPU使用率グラフ化のためのMRTG設定例

5．結果

本システムが正しくネットワークグラフを取得できるか確認するために，様々なパターンのネットワーク上で本システムを実行した．

5.1　仮想環境での結果

実行者の管理ネットワーク上で正常に動作するか確認するため複数の仮想ネットワークを作成し，本システムを実行した．なお表示した論理ネットワークグラフなどはfirefoxブラウザで表示している．図16は現在のネットワークで最も利用されているスター型であり，実行サーバから見た場合はツリー型となる．図16のネットワークNet1で本システムを実行し論理ネットワーク図を自動的に出力，Webブラウザで表示した結果が図17であり，把握できたルータのインターフェースは輪郭に接して表示させた（図18）．しかし，3.2節で述べたように今回は24bit以下のネットワークに対してはインターフェース調査を行わないようにしているため，表示できてない部分も存在する．

ルータのポート（丸部分）に色がついている場合は，MRTGに登録している（システム実行者がSNMP通信によりインターフェース情報を取得可能）ルータであり，クリックすることでトラフィックグラフやCPUグラフを表示できるほか（図19），そのルータの輪郭に接してインターフェースアドレスをすべて表示した．

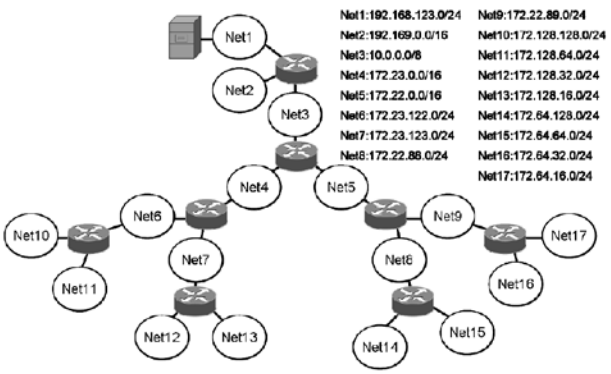

図16　スター型（ツリー型）のネットワーク
（Net4，Net10～17には端末が存在）

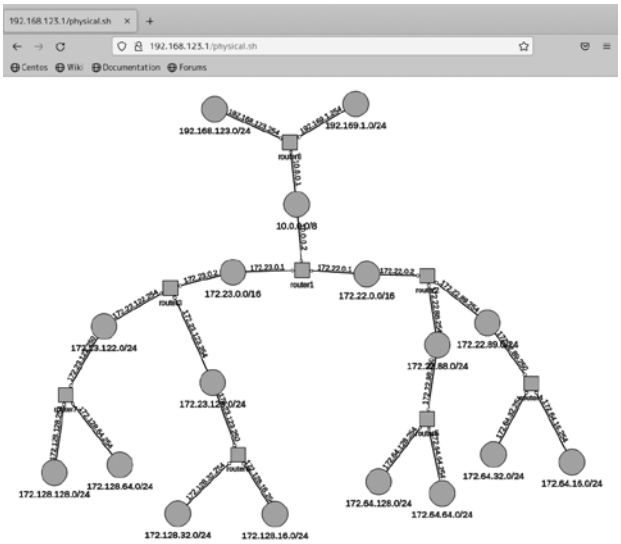

図17　図16内で本システムを実行し作成した
ネットワーク図（firefoxブラウザで表示）

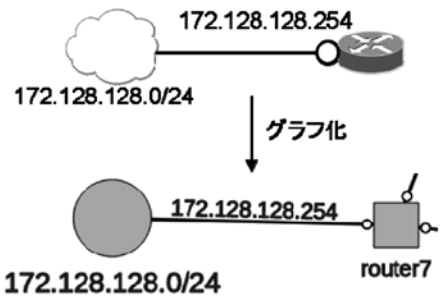

図18　インターフェース表示の例

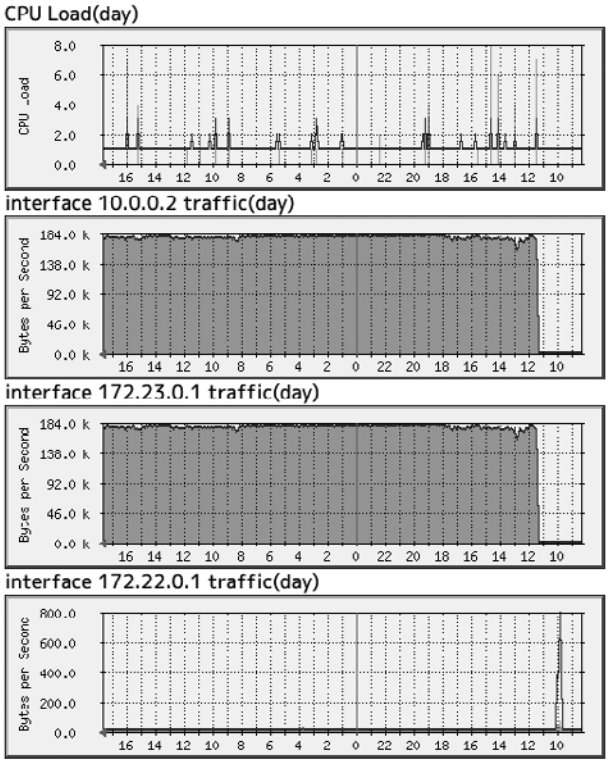

図19　図16 router1をMRTGに登録した場合の
クリック時に表示されるMRTGグラフの例（firefoxで表示）

図20はループが存在している仮想ネットワークでありサーバA，Bからシステムを実行し，出力した論理ネットワーク図が図21，22であるが，１つの実行サーバの情報のみを参考にグラフの作成を試みたとき，tracerouteが通過しない経路は把握することができない．把握できない経路に存在するルータがMRTGに登録されている場合は経路を把握できるが登録を行っていないときはできないため，サーバを複数用意し，それらで作成したグラフを自動で１つのグラフにまとめることで通過しない経路も提供することができる（図23）．しかし，２つの実行サーバでも確認できない経路が存在した場合，さらにサーバを増やす必要があり，サーバを増やすほど本システムの正確性も増すが，現実的には設備のためのコストが増加してしまうことが考えられるため，根本的な解決策とは言えない．

5. 2　東京情報大学内ネットワークでの実行結果

本システムを管理外のネットワークで稼働させた場合でも正常に動作するか確認するため，東京情報大学内の172.22.1.0/24と192.168.124.0/24より本システムを実行し，出力した２つのネットワークグラフ

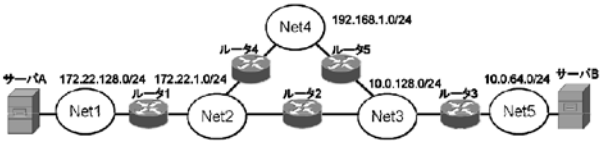

図20　ループが存在するネットワーク

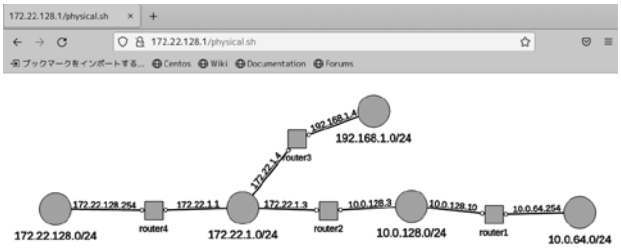

図21　図20内サーバAより実行した場合の
ネットワーク図

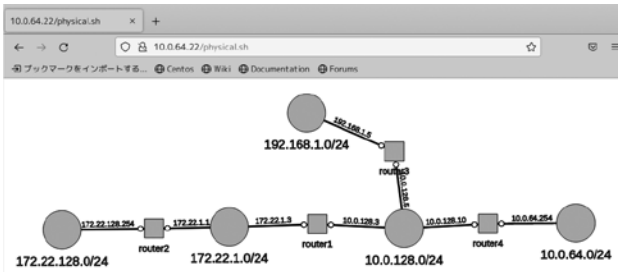

図22　図21内サーバBより実行した場合の
ネットワーク図

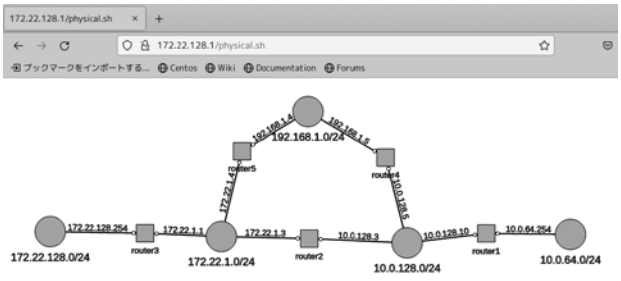

図23　図21，22のグラフをマージしたグラフ

をマージした結果からネットワーク構造を確認した（図24）．なお学内ネットワークは単純なスター型のネットワークである．本システムを実行したサーバは両方ともSNMP通信による情報取得が許可されていないためMRTGグラフはすべてのルータで表示できない．学内ネットワークでの実行結果より出力されたグラフを確認した限り，学内ネットワークはスター型であることが把握できた．しかし，実際の学内ネットワークにはグラフで表示されていないネットワークが全体の大部分を占める．この原因を調査した結果，学内ネットワークは手動設定による静的ルーティングを採用しており，RIPによる動的ルーティングに登録しているネットワークは一部のみであることが判明した．

これによりRIPで公開しているネットワーク以外の自動グラフ化は不可能であり，その場合は事前の静的ルーティング情報取得と手動操作が必要であることが確認できた．

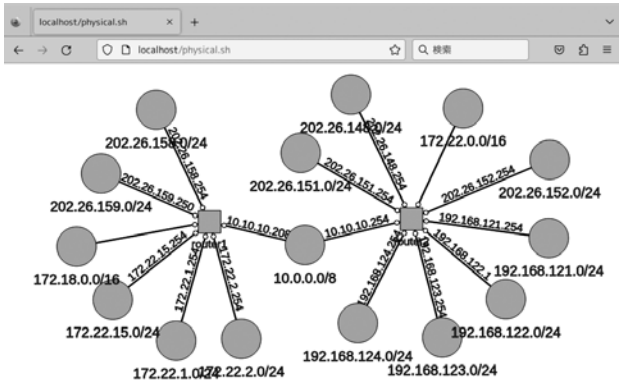

図24　学内ネットワークのグラフ
（172.22.1.0/24と192.168.124.0/24より実行）

6．まとめ

本研究では，RIPパケットでネットワークの存在を把握し，Nmapとtracerouteでネットワークまでの経路を取得して自動的に論理ネットワーク情報を出力する機能と，複数の論理ネットワーク情報をマージし，Webブラウザ上で，ユーザにルータをクリックされた時にMRTGグラフを表示させる機能を持つ論理ネットワークグラフをユーザに提供するシステムを構築した．

仮想ネットワークと学内ネットワークに対し本システムを実行した結果，単純なツリー型のネットワークでは正しくグラフ化を行えることがわかった．またMRTGを採用しているルータは，ルータが持つすべてのインターフェースアドレスとMRTGグラフをネットワークグラフで表示させることができた．さらに，tracerouteでグラフ化を行っ

ているため，ループが存在しているネットワークの場合でも別ネットワークに実行サーバを設置し複数の実行サーバで取得したネットワーク情報をマージすることでグラフ化を行えることも確認できた．また，RIPパケットに全てのネットワークが記入されていない場合は，パケットに記入されているネットワークしか確認できず，自動では正確なグラフを出力できないことも確認できた．

本研究では，RIPを使用しネットワーク図を作成しているが，別のルーティングプロトコルであるOSPF［16］もネットワークアドレスやマスクが存在しRIPと類似している．そのため，本システムを小修正するだけでOSPFからもネットワークグラフを取得可能と考えられる．

本システムは実行サーバを複数設置することでループが存在するネットワーク対策をしているが，多数設置すると設備コストが増加してしまう．さらに，静的ルーティングと動的ルーティングを両方使用しているネットワーク，もしくは静的ルーティングのみのネットワークの自動グラフ化を正確に行えないため，これらを解決する方法を考える必要がある．さらに，L2スイッチのみで分割され，ルータに登録していないVLANネットワークはRIPパケットに含まれないため，本システムはVLANネットワークを把握できず，正確にグラフ化できないことが今後の課題である．

謝　辞

本研究の論文執筆にあたり，東京情報大学内ネットワーク情報を提供して下さり，かつ御助言いただいた井関文一教授に感謝の意を表します．

参考文献

［1］総務省：情報通信分野の現状と課題，https://www.soumu.go.jp/johotsusintokei/whitepaper/ja/r04/html/nd238110.html（参照2023-08-23）．
［2］総務省：インターネットの利用状況，https://www.soumu.go.jp/johotsusintokei/whitepaper/ja/r03/html/nd242120.html（参照2023-08-25）．
［3］経済産業省：我が国におけるIT人材の動向，https://www.meti.go.jp/shingikai/mono_info_service/digital_jinzai/pdf/001_s01_00.pdf（参照2023-08-25）．
［4］Red Hat：ネットワーク管理とは，https://www.redhat.com/ja/topics/management/what-is-network-management（参照2023-08-23）．
［5］NTTコミュニケーションズ：SDNとは，https://www.ntt.com/bizon/glossary/e-s/sdn.html（参照2023-08-25）．
［6］秋田海斗，長田智和，谷口裕治：OpenFlowを用いたネットワーク監視とトポロジーの可視化，第80回全国大会講演論文集，pp.197-198（2018）．
［7］尾形克彦，貫井春美：ネットワークマップの自動作成のためのトポロジー解析，第47回全国大会講演論文集，pp.303-304（1993）．
［8］Case, J. and Fedor, M. and Schoffstall, M. and Davin, J.：A Simple Network Management Protocol：RFC1157（1990）．
［9］太田貴彦，神宮武志：ネットワーク構成把握技術の開発, azbilグループ技術研究報告書, pp.48-52（2021）．
［10］辻本幸徳，藏内将博，井口信和：SNMPを用いたIPネットワーク構成情報の推定，第11回情報科学技術フォーラム講演論文集（FIT），pp.143-146（2012）．
［11］Malkin, G. : RIP Version 2-Carrying Additional Info : RFC1723（1994）．
［12］MRTG : https://oss.oetiker.ch/mrtg/（参照2023-08-23）．
［13］McCloghrie, K. and Rose, M. : Management Information Base-Ⅱ : RFC1213（1991）．
［14］vis-network –npm : https://www.npmjs.com/package/vis-network（参照 2023-08-25）．
［15］Vis Network Examples : https://visjs.github.io./vis-network/examples/（参照2023-08-25）．
［16］Moy, J. : Open Shortest Path First Routing V2 : RFC1583（1994）．