卒業論文　　2011年度 (平成23年度)

Mtrace Looking Glass Systemによる IPマルチキャストネットワークの状態監視と分析

慶應義塾大学 環境情報学部

小野山 裕己

指導教員

慶應義塾大学 環境情報学部

村井 純

中村 修

慶應義塾大学 大学院 政策・メディア研究科

朝枝 仁

平成24年2月14日

卒業論文要旨 2011 年度 (平成 23 年度)

Mtrace Looking Glass System による IP マルチキャストネットワークの状態監視と分析

本研究では，IP マルチキャストの経路情報をリアルタイムに探索可能な Mtrace Looking Glass システムを実現する．このシステムは，IETF で提案されているインターネットドラフト Mtrace2 で定義された経路情報を IP マルチキャストルータから取得する．Mtrace Looking Glass Server は複数台が連携して動作することで，異なる AS に配置された場合でも連携して経路を遡って探索することが可能である．また，ネットワーク管理者のポリシーによって，ログイン機能あるいは SimpleNetwork Management Protocol (SNMP) を選択して，IP マルチキャストの経路状態をルータから取得することができる．これらの機能により，今まで取得することが難しかったデータをリアルタイムに取得することが可能となり，障害の発生地点・原因の早期解明につながる．さらに，このシステムを応用したクローラー機能により，任意の時間粒度でデータを取得・蓄積を行える．これにより，フラッピング，IP マルチキャスト経路のライフタイム，Autonomous System (AS) を跨ぐマルチキャスト経路の状態，Any-source multicast (ASM) における送信者の移り変わりや SessionAnnouncement Protocol(SAP) の利用実態を分析し，DoS 攻撃やルーティングプロトコルの障害検知を提示することが出来る．

キーワード
1．IP マルチキャスト, 2．経路探索, 3．Mtrace2,

慶應義塾大学大学 環境情報学部

小野山 裕己

Abstract of Bachelor Thesis

Academic Year 2011

IP Muticast Network Monitoring and Analysis by Mtrace Looking Glass System

In this thesis, the "Mtrace Looking Glass" is designed and implemented to obtain IP multicast routing states from multicast routers activating in the Internet. Data given by the Mtrace Looking Glass can be shown in the same format defined in the IETF standard Mtrace2 function. Mtrace Looking Glass servers can be installed in different AS domains. When Mtrace Looking Glass traces multicast routes beyond AS domains, Mtrace Looking Glass servers located in different AS domains communicates with each other and shows the concatenated output. In order to support different security level, the Mtrace Looking Glass server provides different access methods – a remote login function or SNMP – to multicast routers, depending on operators' policy. The Mtrace Looking Glass also has a crawler function. It periodically monitors IP multicast routing tree's and data sender's lifetime, multicast routing states, and actual usage of SAP messages at certain intervals. It contributes to IP multicast router operations as it discovers routing problems such as route flapping or DoS attacks.

Keywords :
1.IP Multicast , 2. traceroute,3. Mtrace2

Keio University, Faculty of Environment and Information Studies

Yuki Onoyama

目 次

図 目 次

表 目 次

第1章　序論

本章において，本論文の背景，目的を明らかにする．また，本論文の構成を示す．

1.1　背景

インターネットは現在も広域通信基盤としての進化・成長を続けており、トラフィックは年々増加傾向にある．特に昨今では大容量コンテンツである動画のストリーミングがトラフィックにおける大きな割合を占めている．図 1.1 は，2011 年に Cisco 社が発表した IP トラフィック予測 [1] を表す．縦軸はエクサバイトを表している．2013 年にトラフィックに占める動画の割合が 60%を超えると発表されており，今後もその需要が拡大していくことが予想される．中でも，YouTube[2] やニコニコ動画 [3] はその代表的なサービスであり，Hulu[4] なども急激な成長を遂げている．このような蓄積型動画サービスに対し、Ustream[5] やニコニコ生放送 [6]、IPTV[7] といったリアルタイムストリーミングサービスもまた需要が高まっている．

一対多型の通信を実現する技術として、IP マルチキャストがある．IP マルチキャストの具体的な説明は第 2 章において行うが，送信者側の計算機資源やネットワーク帯域資源の利用を効果的に抑えることが可能となる．しかし，IP マルチキャスト通信を実現するためには様々な問題がある．まず，経路上の全ルータがマルチキャストに対応していなければならない．マルチキャストは経路制御プロトコルがユニキャストに比べて複雑であり，管理が難しい．既にひかり TV[8] や BBTV[9] は IPv6 マルチキャストによる配信が行われているが，特にドメイン間の経路制御は難しく、インタードメインの利用に限られている．また，IP マルチキャストは受信者の有無でその経路が常に変化するため、どのような経路が構成されているかを把握することが難しく，トラブルシューティングやマルチキャストネットワークのオペレーションも非常に難しくなっている．

以上のことから，今後予想される大容量コンテンツの配信，特にリアルタイム性の高い映像配信を実現する技術として，計算機資源及びネットワーク資源を効果的に抑えることができる IP マルチキャストは不可欠であると言える．しかし，IP マルチキャスト通信の実現には様々な障壁が存在する．このため，IP マルチキャストの経路をリアルタイムに明らかにすることで，トラブルシューティングや IP マルチキャストネットワークのオペレーションを容易にすることが求められている．

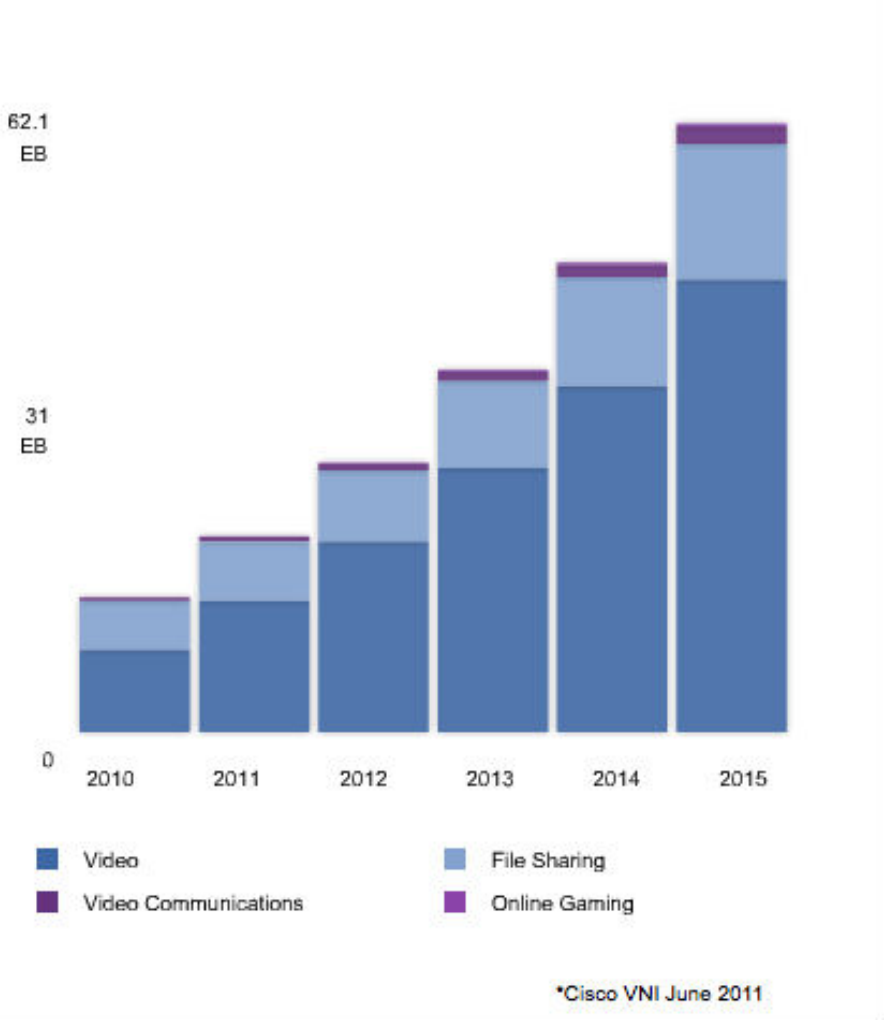

図 1.1: Cisco 社発表のトラフィック予測 (2011 年 6 月)

1.2 本研究の目的

本研究では，ラストホップルータからファーストホップルータまで遡って，IP マルチキャストの経路情報をリアルタイムに探索可能とするシステムを構築する．これにより，障害の発生地点・原因がリアルタイムに解析可能となり，オペレーション・トラブルシューティングのコスト削減につなげることができる．また，このシステムを応用して，一定時間ごとに IP マルチキャストの経路情報を取得・蓄積する．フラッピング，ディストリビューションツリーのライフタイム，AS を跨ぐマルチキャストルーティングの動作，ASM における送信者の移り変わりや Session Announcement Protocol (SAP)[10] の利用実態を明らかにすることで，今まで取得出来なかった情報を明らかにすることができ，今後の IP マルチキャストネットワークのデザインや IP マルチキャストプロトコルの発展に役立て

ることが可能となる．

1.3 本論文の構成

本論文の構成を以下に示す．第２章では，本研究の要素技術である IP マルチキャストについて述べる．第３章では，本研究の要求事項及びそれを実現するシステムの提案を行う．第４章では，３章で提案したシステムの設計について述べる．第５章では，４章の設計に基づいた実装について述べる．第６章では，本システムの実装についての評価と本システムを用いて取得したデータの分析を行う．第７章では，本研究のまとめを示し、今後の課題並びに可能性について言及し，結論とする．

第2章　既存技術

本章では，本研究の要素技術である IP マルチキャストについて整理し，既存の経路探索手法について述べる．

2.1　IP マルチキャスト

IP マルチキャストは単一の送信者が複数の受信者に対して同一内容のパケットを一斉に配信する際，送信者が単一のパケットを配信するだけで複数の受信者にパケットを届けることができる技術である．ユニキャストの場合，受信者の数だけ送信者がパケットを配信しなければならないが，IP マルチキャストは経路上のルータがディストリビューションツリーに従ってパケットの複製を行い，受信者に配信を行う．特にリアルタイム性の高い大規模配信を行う場合，ユニキャストに比べて送信者の計算機資源やネットワーク資源そのものの消費量を効果的に抑えることができる．

図 2.1 は IP マルチキャスト通信の概要を示しており，用語説明を下記に示す．

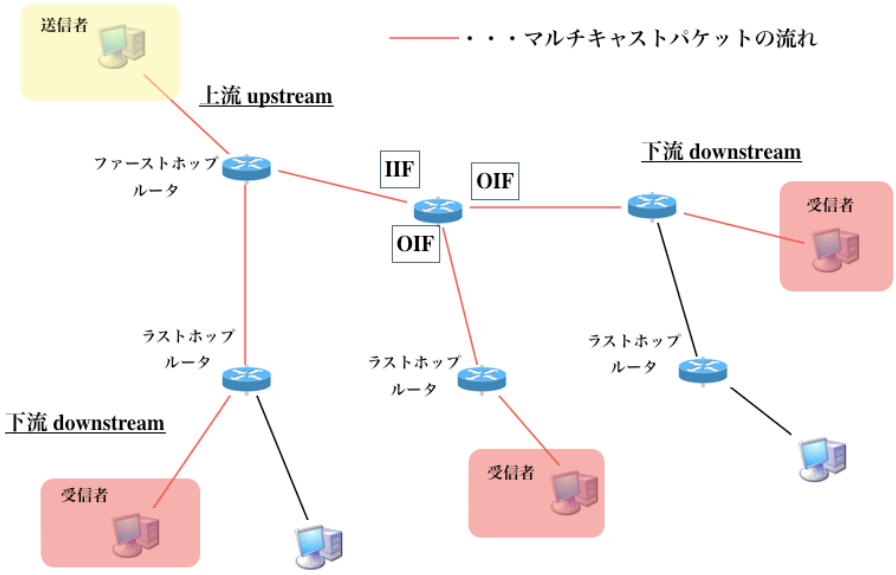

図 2.1: IP マルチキャストの概要図

Downstream（下流）

受信者の方向を Downstream，または下流という．

Upstream（上流）

送信者の方向を Upstream，または上流という．

Incoming Interface

上流からパケットを受け取るインタフェースのことである．Incoming Interface は，一つのディストリビューションツリーにつき必ず一つである．

Outgoing Interface

上流から受け取ったパケットを下流に向かって転送するインタフェースのことである．Incomig Interface と異なり，Outgoing Interface は複数存在する場合もある．

ファーストホップルータ

送信者が接続されているルータのことである．

ラストホップルータ

受信者が接続されているルータのことである．

2.1.1 マルチキャストアドレス

IP マルチキャストは，ユニキャストのアドレスとは別に特別なマルチキャストアドレスを必要とする．受信者は送信先にマルチキャストアドレスを指定し，参加要求を行うことでそのマルチキャストアドレスで分けられるマルチキャストグループに参加する．また，送信者はマルチキャストアドレスを宛先に指定することで，経路上のルータが適切にパケットを複製してマルチキャストグループに参加している受信者にデータを届ける．IPv4 マルチキャストアドレスの範囲を図 2.2，IPv6 マルチキャストアドレスの範囲を図 2.3 に示す．

IPv4 では 224.0.0.0〜239.255.255.255 の範囲，IPv6 では FF00::/8 のアドレスで定義される．ルーティングプロトコルなどの制御情報を送信するためにあらかじめ予約されているアドレスもこの中に存在するが，各用途についての説明は省略する．

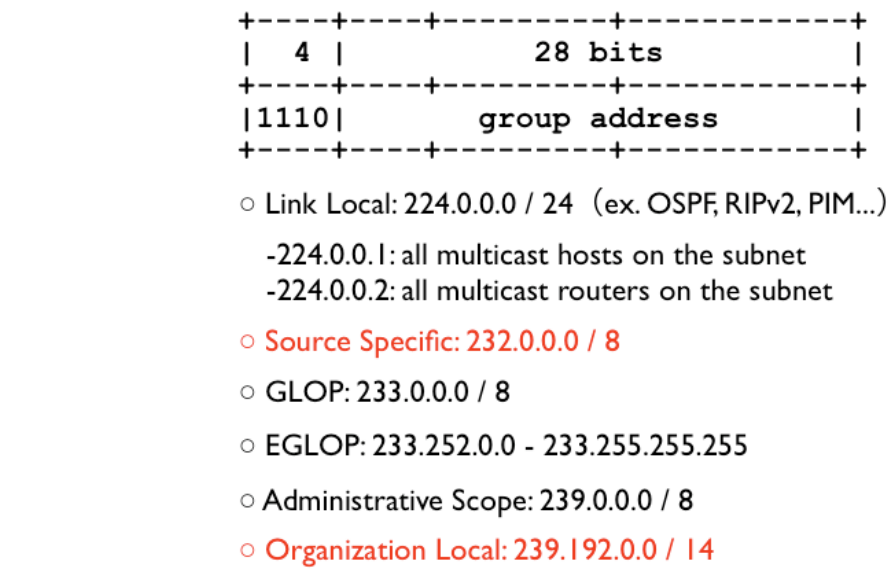

図 2.2: IPv4 マルチキャストフォーマット

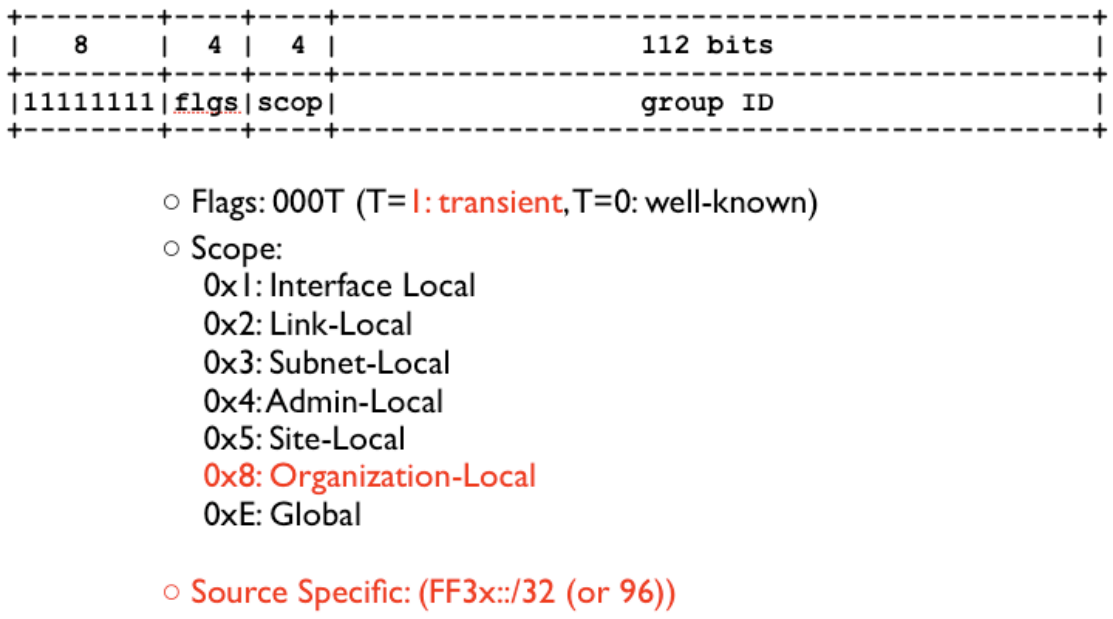

図 2.3: IPv6 マルチキャストフォーマット

2.1.2 ディストリビューションツリー

IP マルチキャスト通信を行うには，経路上の全ルータで IP マルチキャストのルーティングプロトコルを動作させる必要がある．そのルーティングプロトコルが受信者の有無に

よってディストリビューションツリーを構築し，マルチキャストパケットの配信を行う．IP マルチキャストのルーティングプロトコルは様々なものが存在するが，PIM-SM[11] が利用されているネットワークがほとんどであるため，本節では PIM-SM の動作に従って説明を行う．

Any Source Multicast (ASM)

Any Source Multicast は，送信元ツリーと共有ツリーの 2 つのディストリビューションツリーを組み合わせ，Rendezvous Point を中心としたマルチキャストパケットの配信を行う．RP とは，送信元ツリーと共有ツリーの境界のルータであり，受信者の参加要求と送信者のパケットの待ち合わせ場所である．PIM-SM における Any Source Multicast の動作について，RP への参加要求を図 2.4，共有ツリー作成の様子を図 2.5，送信者への参加要求を図 2.6，共有ツリーと送信元ツリーが重複した状態を図 2.7，RP への停止要求を図 2.8，送信元ツリーの様子を図 2.9 に示す．

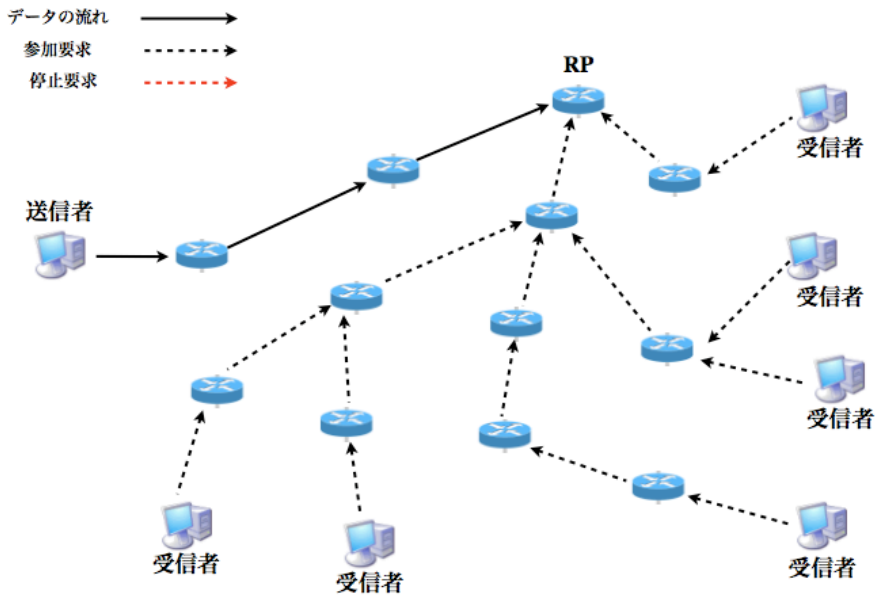

図 2.4: マルチキャストツリー構築の流れ 1 -RP への参加要求-

まず始めに，送信者は宛先にマルチキャストアドレスを指定する．宛先にマルチキャストアドレスが指定されたパケットを受け取ったファーストホップルータは，あらかじめ決定されている RP へと転送を行う．一方で受信者は，接続されているルータに対してそのマルチキャストアドレスへの参加要求を行う．ラストホップルータとなるそのルータは，参加要求を受け取ると同様にあらかじめ決定されている RP に向かって参加要求を行う．この際，RP への参加要求はユニキャストのルーティングテーブルを参照して行われる．

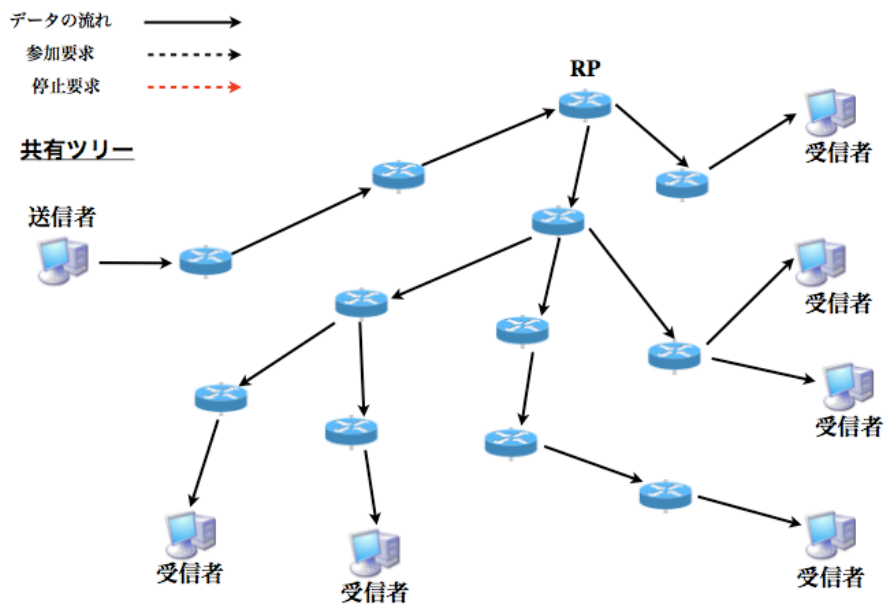

図 2.5: マルチキャストツリー構築の流れ２ -共有ツリー-

RP は，参加要求を受け取ったインタフェースにマルチキャストパケットを送信する．その他のルータも同様に，参加要求を受け取ったインタフェースからマルチキャストパケットを送信する．この繰り返しをラストホップルータまで行うことで，受信者のみに適切にマルチキャストパケットを転送することができる．参加要求を受け取ったインタフェースが複数ある場合，その数だけマルチキャストパケットを複製して転送を行う．このディストリビューションツリーを共有ツリーという．共有ツリーは，マルチキャストアドレスごとに構築され，(*, G) で表される．受信者は，送信者のアドレスを知らなくても G で表記されるマルチキャストアドレスを知っていればマルチキャストパケットを受信することができる．

共有ツリー (*, G) に従ってマルチキャストパケットが配信される場合，必ず RP を中心とするため，最適な経路でルーティングが行われているとは限らない．そこで，ラストホップルータは共有ツリー (*, G) に従って配信されているマルチキャストパケットから送信者のアドレスを参照し，送信者が存在する方向に向かって新たに参加要求を行う．この際，RP への参加要求と同様にユニキャストのルーティングテーブルを参照して行われる．

ラストホップルータが送信者が存在する方向に向かって行った参加要求は，最終的にファーストホップルータが受け取る．ファーストホップルータは，参加要求を受け取ったインタフェースからマルチキャストパケットを送信する．その他のルータも同様に，参加要求を受け取ったインタフェースからマルチキャストパケットを送信する．この繰り返しをラストホップルータまで行うことで，送信者から受信者まで最適な経路でマルチキャストパケットを転送することができる．このディストリビューションツリーを送信元ツリーという．送信元ツリーは，送信者のアドレスとマルチキャストがセットとなって構築され，

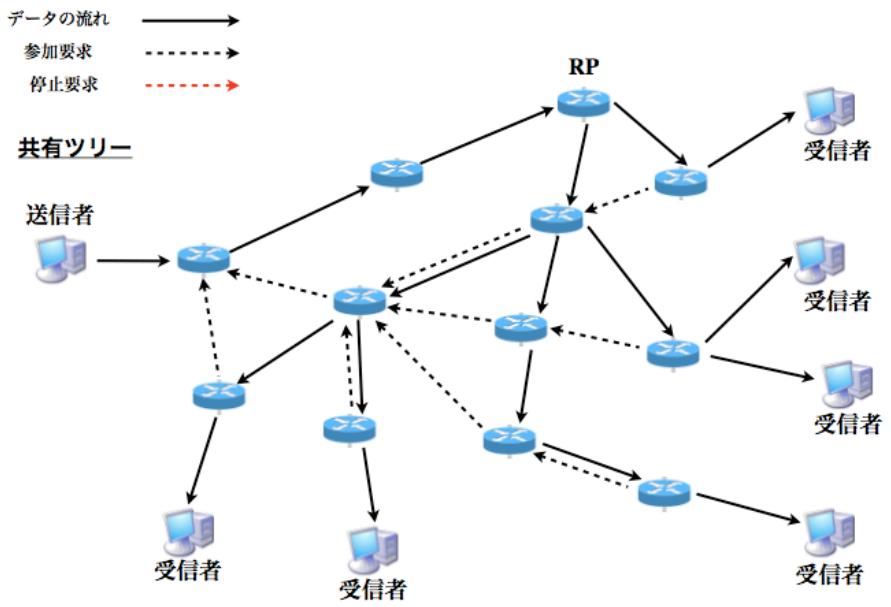

図 2.6: マルチキャストツリー構築の流れ３ -送信者への参加要求-

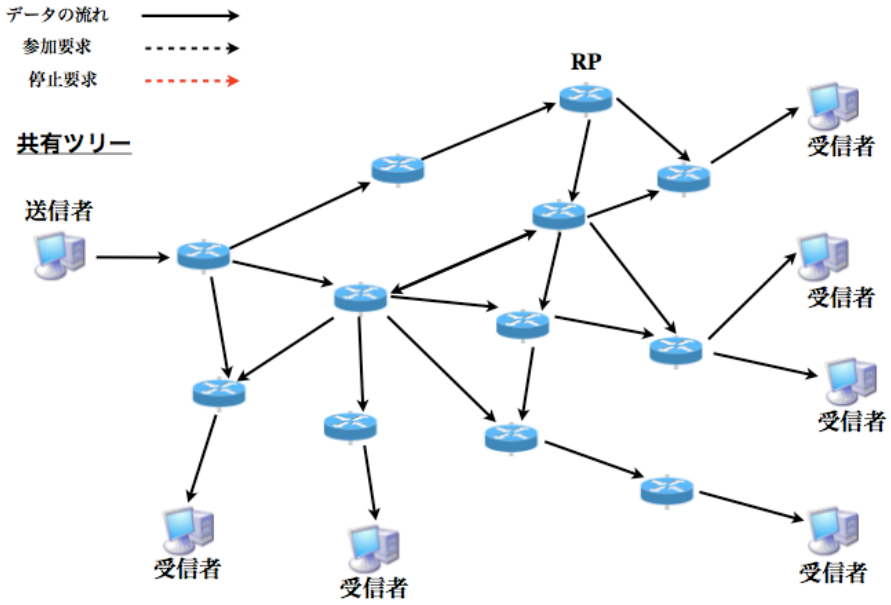

図 2.7: マルチキャストツリー構築の流れ４ -共有ツリーと送信元ツリーの重複-

(S, G) で表される．S は送信者のアドレスであり，G はマルチキャストアドレスである．

しかし，この状態では共有ツリー (*, G) に従って配信されたマルチキャストパケットと送信元ツリー (S, G) に従って配信されたマルチキャストパケットが重複してしまう．そ

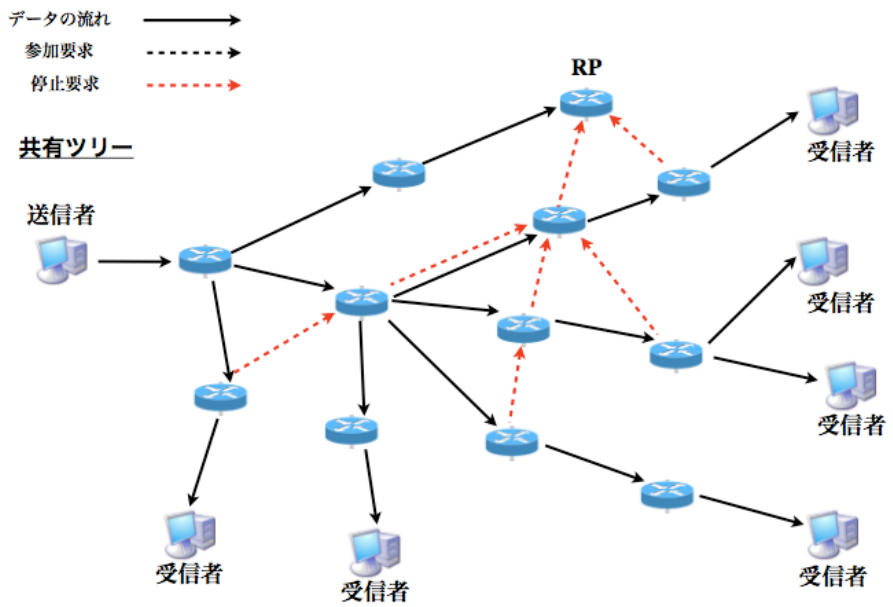

図 2.8: マルチキャストツリー構築の流れ５ -RP への停止要求-

こで，ラストホップルータは送信元ツリー (S, G) に従って配信されたマルチキャストパケットを受け取った段階で，ネクストホップルータへ停止要求を行う．共有ツリー (*, G) の経路上のルータは，全ての Outgoing Interface から停止要求を受け取った場合，さらに次のネクストホップルータへと停止要求を行う．このようにして最終的には RP へと停止要求が行われる．図 2.9 は最終的に送信元ツリー (S, G) のみになった状態を表している．

ここまで ASM におけるディストリビューションツリー構築の流れを説明した．しかしながら，実際のネットワークでは受信者の参加要求のタイミングは異なり，図 2.4 から図 2.9 のように共有ツリー (*, G) ができてから送信元ツリー (S, G) へと切り替わるまで奇麗に行われることは非常に少ないと言え，受信者の出現や消失の度にディストリビューションツリーが切り替わる．経路ができたり消えたりと非常に複雑な動きをしていることは容易に想像できる．そのため，いつどのような経路が構築されているかを把握することも困難である．

Source Specific Multicast (SSM)

Source Specific Multicast は， 1 対多型の通信に特化してマルチキャストパケットの配信を行う．送信元ツリー (S, G) のみを構築し，初めから最適な経路でマルチキャストパケットを配信する．前節で述べた通り，IPv4 では 232.0.0.0/8，IPv6 では FF00::/8 のマルチキャストアドレスで定義される．SSM は，PIM-SM の派生型であり，PIM-SSM とも表記される．PIM-SSM のディストリビューションツリー構築の流れとして，送信者への参加要求を図 2.10，送信元ツリーの様子を図 2.11 に示す．

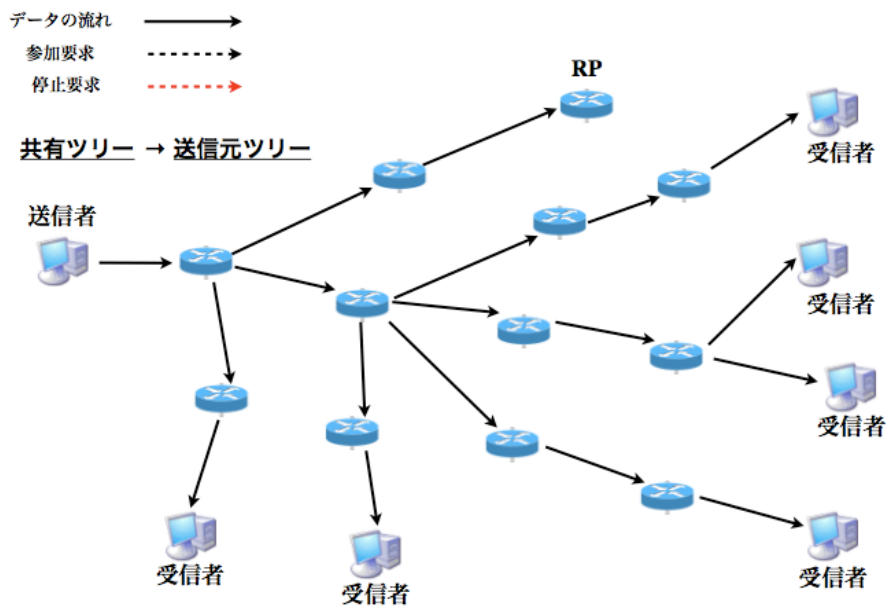

図 2.9: マルチキャストツリー構築の流れ 6 -送信元ツリー-

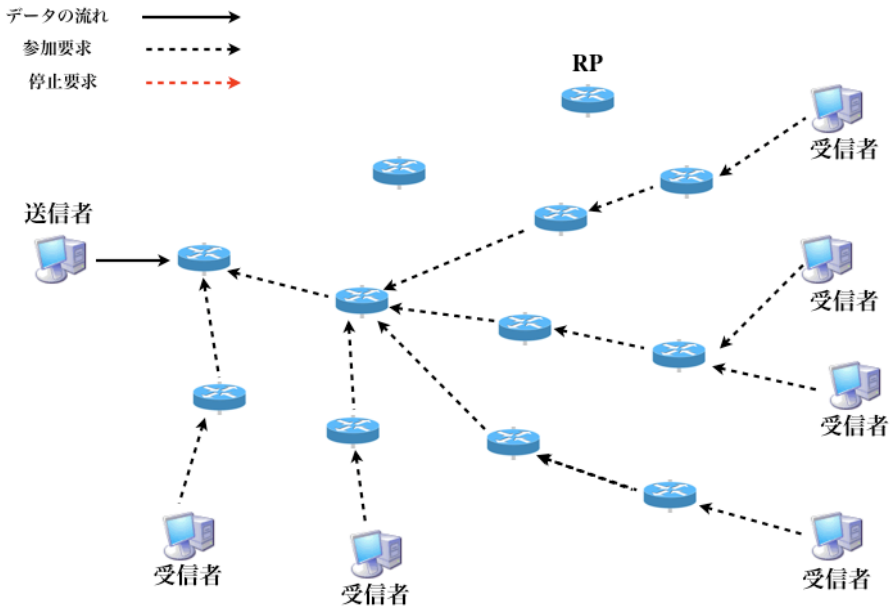

図 2.10: マルチキャストツリー構築の流れ SSM -送信者への参加要求-

ASM と同様まず始めに，送信者は宛先にマルチキャストアドレスを指定する．一方で受信者は，接続されているルータに対して送信者のアドレスとマルチキャストアドレスを指定して参加要求を行う．ラストホップルータとなるそのルータは，参加要求を受け取

ると送信者の方向に向かって参加要求を行う．この際，送信者の方向への参加要求はユニキャストのルーティングテーブルを参照して行われる．

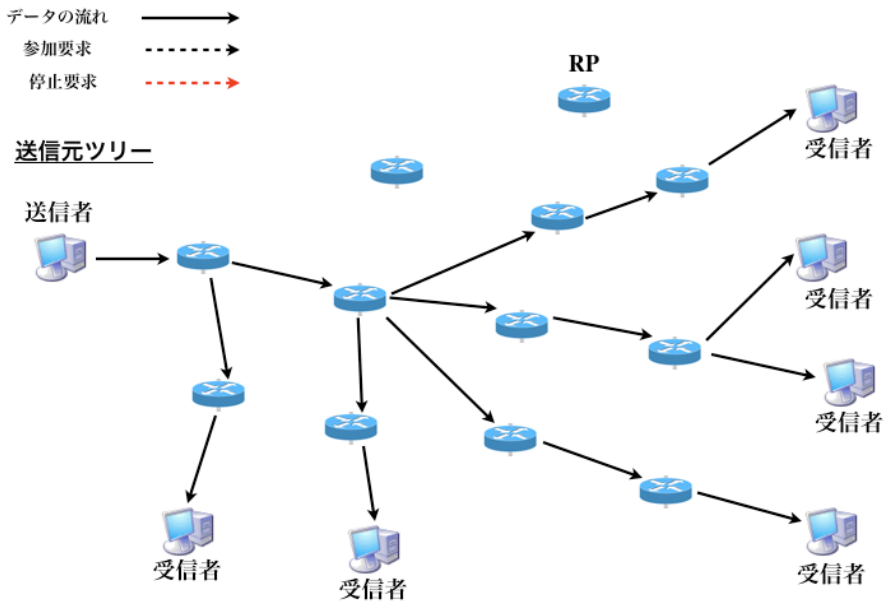

図 2.11: マルチキャストツリー構築の流れ SSM -送信元ツリー-

ラストホップルータが送信者が存在する方向に向かって行った参加要求は，最終的にファーストホップルータが受け取る．ファーストホップルータは，参加要求を受け取ったインタフェースからマルチキャストパケットを送信する．その他のルータも同様に，参加要求を受け取ったインタフェースからマルチキャストパケットを送信する．この繰り返しをラストホップルータまで行うことで，送信者から受信者まで最適な経路でマルチキャストパケットを転送することができる．

SSM は，ASM と比べてディストリビューションツリーの構築が複雑ではない．さらに，送信者のアドレスとマルチキャストアドレスがセットで指定されるため，マルチキャストアドレスの重複が起こっても送信元ツリー (S, G) の重複が起こることもない．しかし，受信者がマルチキャストアドレスに加えて，あらかじめ送信者のアドレスを知ることができなければ参加要求を行うことができない．

2.1.3 Session Announcement Protocol(SAP)

Session Announcement Protocol は，受信者が送信者のアドレスを知ることができる方法の一つである．送信者は，あらかじめ決められた SAP 用のマルチキャストアドレスに対して通知を行う．そうすることで，受信者が SAP 用のマルチキャストアドレスから送

信者のアドレスとマルチキャストアドレスを取得することができる．IANA[12] によって，224.2.0.0-224.2.255.255 の範囲が予約されている．

2.2 Mtrace2

Mtrace2[13] は，ラストホップルータからファーストホップルータまでホップバイホップで経路を遡って IP マルチキャストの経路情報を探索する手法である．その動作概要を図 2.12 に示す．

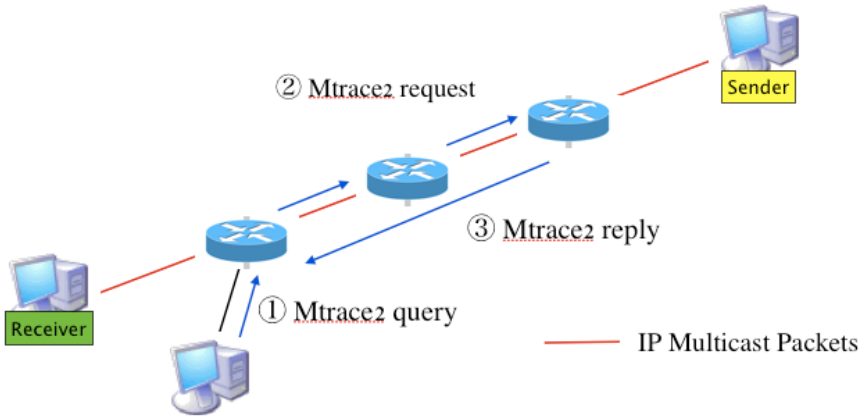

図 2.12: Mtrace2 動作概要図

Mtrace2 クエリを受け取ったラストホップルータは，マルチキャスト経路情報を取得した上で上流の隣接ルータに Mtrace2 リクエストを送る．Mtrace2 リクエストを受け取ったルータは，同様にマルチキャスト経路情報を取得した上で上流の隣接ルータに Mtrace2 リクエストを送る．ファーストホップルータまでたどり着くと，ラストホップルータへ Mtrace2 レスポンスを返す．経路上のルータが Mtrace2 に対応していない場合など，ある地点から遡れなかった場合も同様に最後のルータがラストホップルータに Mtrace2 レスポンスを返す．Mtrace2 クエリのフォーマットを図 2.13，IPv4 の Mtrace2 Standard Response Block のフォーマットを図 2.14，IPv6 の Mtrace2 Standard Response Block のフォーマットを図 2.15 に示す．

しかし，Mtrace2 はセキュリティ上の配慮からラストホップルータに接続されているノード以外からのクエリを受け付けない仕様となっており，外部からクエリを送って経路探索を行うことはできない．

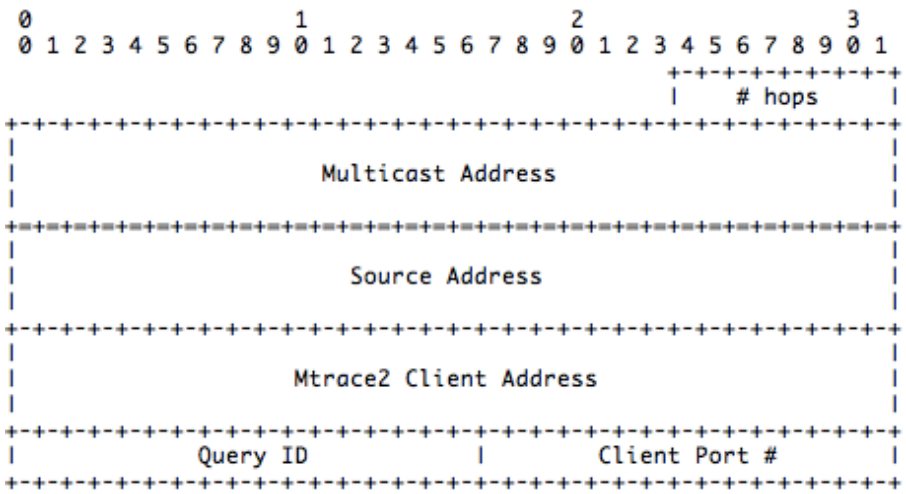

図 2.13: Mtrace2 クエリヘッダーフォーマット

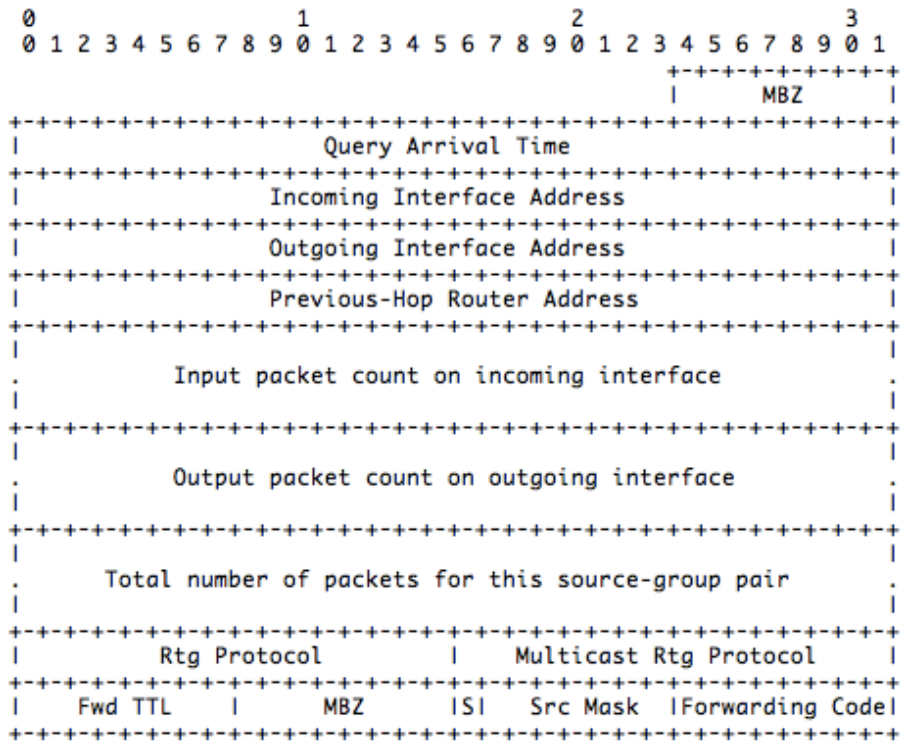

図 2.14: IPv4 Mtrace2 Standard Response Block

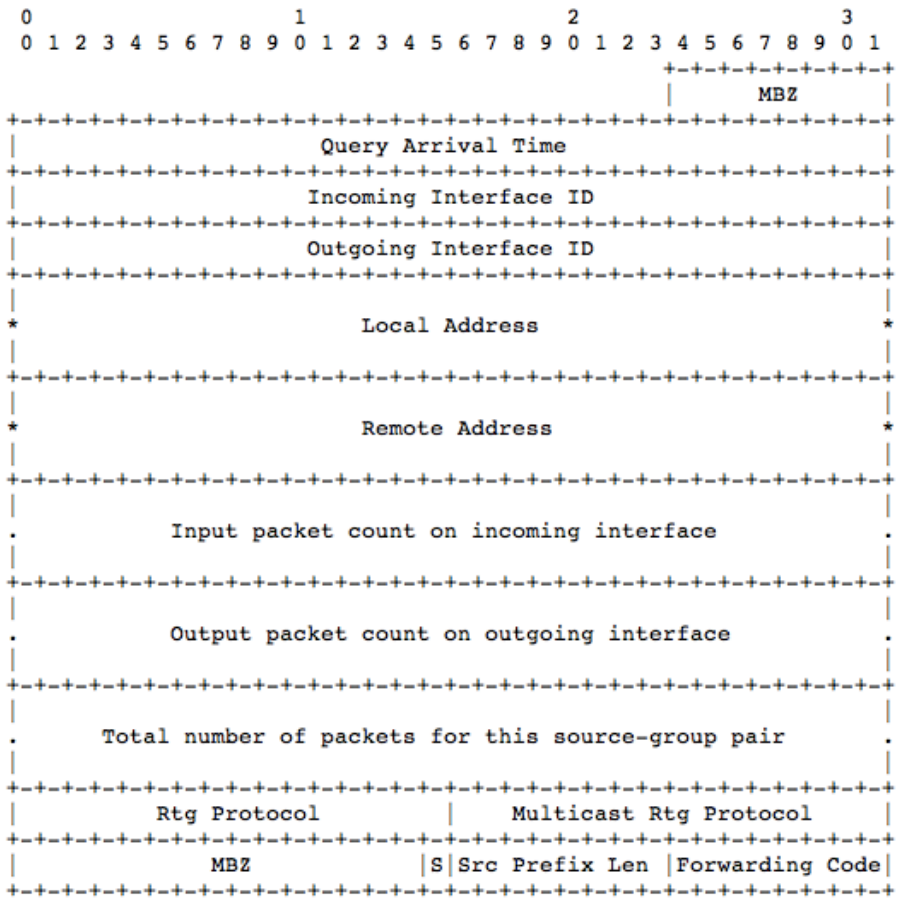

図 2.15: IPv6 Mtrace2 Standard Response Block

2.3 まとめ

本章では，本研究の要素技術である IP マルチキャストについての整理を行った．ASM におけるディストリビューションツリー構築は非常に複雑であり，いつどのような経路が構築されているかを把握することが難しい．一方で，SSM はディストリビューションツリーの構築は ASM に比べ複雑ではないが，受信者が送信者のアドレスとマルチキャストアドレスをセットで指定して参加要求を行わなければならない．そのため，受信者の負担が大きく，データの受信自体に障壁が生まれている．しかし，ASM と SSM にはこのような違いがあるにも関わらず，利用比率に関して明らかになっていない．その理由としては，先に述べたように経路が頻繁に切り替わることや，受信者の数を把握できないことなどが挙げられる．さらには，ASM と SSM の利用比率を明らかにしていく上で，送信者のアドレスを告知する SAP の利用者はどれだけいて，どの程度の時間存在していたのかを解明する必要もある．SAP を利用している送信者のアドレスに対して，SSM として参加要求を行った受信者がいた場合は経路上のルータに送信元ツリーが構築されるからである．

また，既存の経路探索手法である Mtrace2 に関しても述べた．Mtrace2 は，ラストホップルータからファーストホップルータまでホップバイホップで経路を遡って探索する手法であるが，セキュリティ面を考慮してラストホップルータに接続されているノードからしかクエリを受け付けない仕様になっている他，Mtrace2 に対応していないルータやセキュリティ上の理由から Mtrace2 リクエストを始めとしたメッセージをブロックしているルータが経路上に存在した場合は，そのルータから先の経路を探索することができない．そのため，現在 IP マルチキャストルーティングが行われているネットワークにおいて，その経路情報を取得することは困難であると言える．しかし，Mtrace2 において取得するデータやその経路探索手法は大いに参考になる，

次章では，本章で述べた既存技術を踏まえて本研究の目的を達成するための手法について述べる．

第3章　アプローチ

本章では，目的とするIPマルチキャストネットッワークのモニタリング及び統計・分析に対する要求事項を述べ，その手法についての提案を行う．

3.1　本研究の目的・要求事項

本研究では，IPマルチキャストの経路をリアルタイムに探索可能とし，障害の発生地点・原因をリアルタイムに解析可能とする．これにより，オペレーション・トラブルシューティングのコスト削減につなげることを目的とする．また，一定時間ごとにIPマルチキャストの経路情報を取得・蓄積し，分析を行うことで，フラッピング，ディストリビューションツリーのライフタイム，ASを跨ぐマルチキャストルーティングの動作，ASMとSSMの利用比率，ASMにおける送信者の移り変わりやSAPの利用実態を明らかにし，今後のIPマルチキャストネットワークのデザイン及びプロトコルの発展に役立てることを可能とする．

本研究の目的を達するため，以下の項目が必要となる．

ラストホップルータに接続されているノード以外からの探索

IPマルチキャストネットワークの経路探索手法であるMtrace2は，セキュリティ上の理由でラストホップルータに接続されているノードからしかクエリを受け付けない．そのため，本システムでは経路上のルータからIPマルチキャストの経路情報を取得することができる特別な権限を持たせ，本システムを介すことで遠隔からマルチキャストネットワークのモニタリング及び分析を可能とする必要がある．

ホップバイホップでの経路探索

ディストリビューションツリーは，ユニキャストの経路と異なり送信者と受信者が存在して初めて構築される．送信者の存在及び受信者の有無で経路が常に変化するため，どのような経路が構成されているかを把握することが難しい．また，マルチキャストパケットを受け取ったルータは，上流方向の近接ルータのアドレスは情報として保持しているが，下流方向についての情報はパケットを送り出す自分自身のインタフェースのみしか保持しておらず，下流方向の隣接ルータもしくは受信者のアドレスは保持していない．IPマ

ルチキャストの経路を確実に探索するためには，ラストホップルータからネクストホップルータのアドレスである上流の隣接ルータのアドレスを取得しファーストホップルータまで順々に遡っていくことが必要となる．

複数の MtraceLG を介した経路探索

特別な権限を持たせた Mtrace Looking Glass Server（以下「MtraeLG」）を構築した場合，AS 内の全ルータを 1 台の MtraceLG で一括管理することはセキュリティ面から現実的ではない．そこで一つの AS 内に複数の MtraceLG を設置することを想定し，MtraceLG 間で連携ができるようにする必要がある．この場合，同 AS 内に限らず，AS を跨ぐマルチキャストルーティングの経路探索も MtraceLG 間の連携のみで行うことができる．MtraceLG が管轄するルータの範囲として LG ドメインを設け，管轄外であった場合も他の MtraceLG へクエリを送信することができるように設計を行う．クエリを受けた MtraceLG は，クエリを投げた MtraceLG へ探索結果を返す．これにより，ファーストホップルータまで遡った時の MtraceLG が最初の MtraceLG へと適切にデータを返すことが可能となる．

一定時間ごとのデータ収集

IP マルチキャストネットワークの分析を行うためには，その経路情報を一定間隔ごとに同条件で取得する必要がある．MtraceLG にクローラとしての機能を持たせて一定時間ごとにデータを自動収集し，あらかじめ用意したデータを蓄積するサーバに送信を行う．一定時間ごとのデータが蓄積されることで，後からそのデータを用いて時間ごとの変化を用いて分析を行うことが可能となり，IP マルチキャストネットワークの分析を行うことが出来る．

3.2 システム概要

本節では，前節で述べた要求事項に基づいて，MtraceLG を用いて経路情報を取得するシステムモデルを提案する．また，このシステムを用いた一定時間ごとの統計システムの提案も併せて行う．

3.2.1 システムフロー

本研究の目的を満たすため，遠隔からマルチキャスト経路情報を取得できるよう MtraeLG を構築し，ルータからマルチキャスト経路情報を取得する．そこで取得した上流の隣接ルータのアドレスを元にラストホップルータからファーストホップルータまでホップバイホップで遡って経路を探索する．一つの AS 内に複数の MtraceLG が設置されることを想定し，LG ドメインとして管轄するルータの範囲を設ける．MtraceLG がクエリを受けて

から経路探索を行う様子を図 3.1 に示し，箇条書きで順序を説明する．

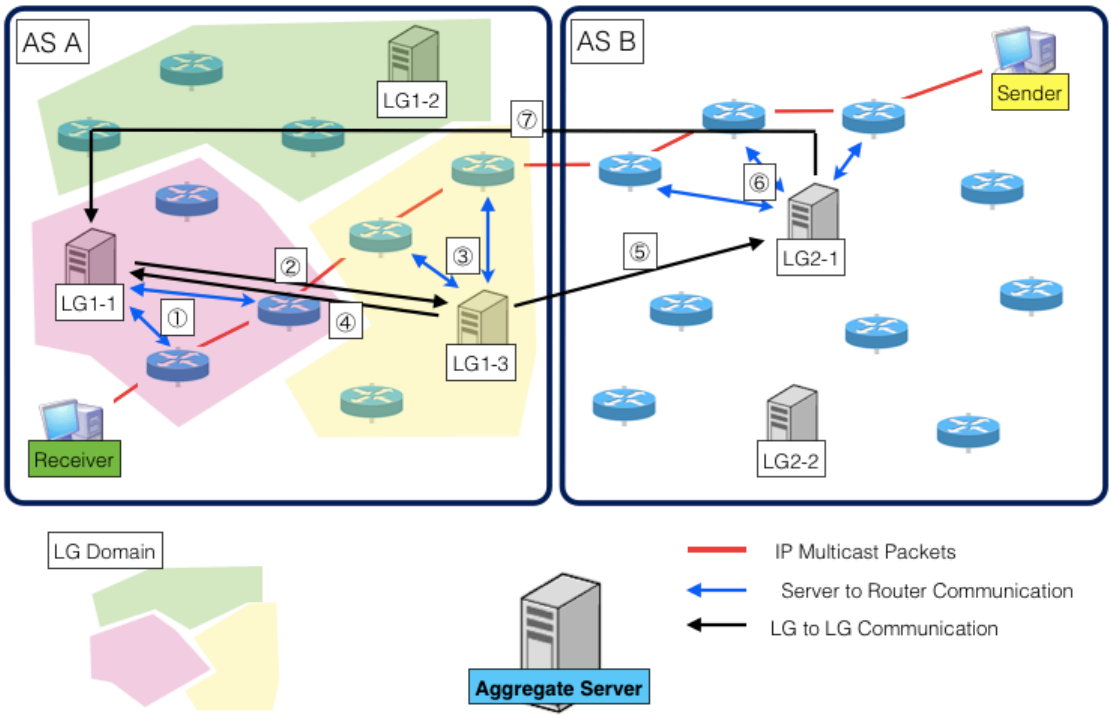

図 3.1: システム概要図

1. クエリを受けた MtraceLG は，ルータから上流の隣接ルータのアドレスなどの情報を取得する．その上流の隣接ルータのアドレスを参照し，ネクストホップルータからも同様に情報を取得する．
2. 上流の隣接ルータのアドレスが LG ドメイン外の場合，他の MtraceLG にクエリを投げる．
3. クエリを受けた他の MtraceLG は，同様に経路を遡る．
4. 上流の隣接ルータのアドレスが AS 外の場合も同様に他の MtraceLG にクエリを投げる．この際，一番最初の LG に取得した情報を返す．
5. クエリを受けた他の MtraceLG は，同様に経路を遡る．
6. ファーストホップルータまで遡ると，最初の MtraceLG へと取得した情報を返す．

上述の動作を行うことで，ラストホップルータからファーストホップルータまでホップバイホップで経路を遡って情報を得ることが可能となる．

MtraceCrawler

前節の要求事項で挙げた一定時間ごとのデータ収集を行うため，上記システムを応用した MtraceCrwaler を構築する．

MtraceCrawler は，ファーストホップルータからラストホップルータまでホップバイホップで経路を遡るのではなく，LG ドメイン内の一つ一つのルータから個別にマルチキャスト経路情報を取得し，その都度あらかじめ用意した Aggregate Server へと取得したデータを送信する．後から取得したデータの分析を行う際に，上流の隣接ルータのアドレスを参照してデータを繋ぎ合わせることで，ディストリビューションツリーがどのように構築されているかを時間単位で把握することが可能とする．これにより，マルチキャストツリー構築の複雑性を明らかにすることができる．この際，マルチキャストアドレスを取得することで，その範囲から ASM と SSM の利用比率，ソースアドレスと組み合わせて分析することで送信者ごとの SAP 利用実態も解明可能となる．また，どこで障害が起きたのかを調査することもできる，MtraceCrawler の動作を図 3.2 に示し，箇条書きで順序を説明する．

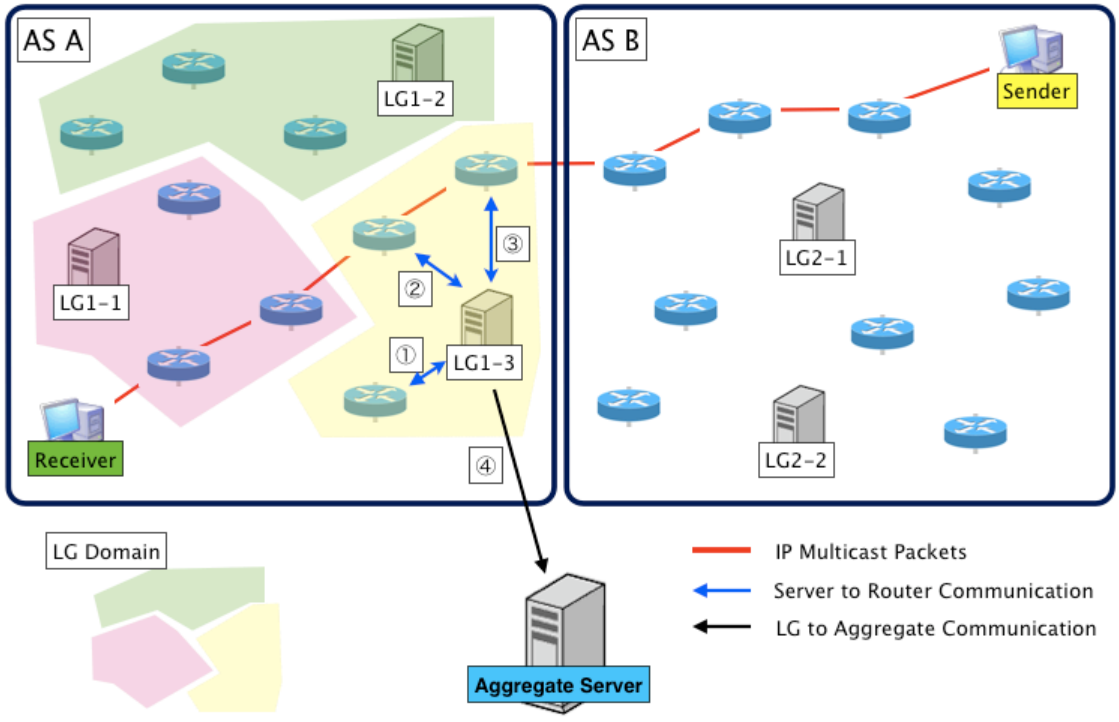

図 3.2: MtraceCrawler 動作概要図

1. MtraceLG は，一定時間置きに LG ドメイン内のルータから情報を取得する．
2. 複数のルータを管理している場合は，1 と同様の動作を繰り返す．

3. 2 と同様の動作を行う．

4. LG ドメイン内全ての探索を終えると，ルータごとにあらかじめ用意した集積サーバへと取得したデータを送信する．

上記動作を一定時間ごとに繰り返し，後から上流の隣接ルータのアドレスを参照して繋ぎ合わせることで，本研究の目的である分析を行うことが可能となる．

3.3 IP マルチキャスト経路情報の取得手法

本節では，ルータから IP マルチキャストの経路情報を取得する手段について述べる．現状，下記の項目が考えられる．

SNMP MIB

IPv6 マルチキャストに対応した MIB である IP Multicast MIB[14] をサポートしていないルータが多く，MrouteMIB[15] や PIMMIB[16] などから IPv4 マルチキャストに関するデータしか取得することができない場合がほとんどである．しかし，プライベート MIB を除く OID が RFC で標準化された仕様であることから，Cisco や Juniper といったルータのメーカーによってコマンドが異なるケースごとに実装を行わなくていいメリットがある．また，毎回ルータを操作して情報を取得する必要は無いので，プログラムエラーやヒューマンエラーによる障害の可能性も下記の２つと比べると低い．

ルータへのリモートログイン

実ネットワークで動作しているルータにリモートログインを行うことは，プログラムの不具合による意図しない設定変更の可能性などリスクが高く，パスワード流出の危険性もある．さらに，機種によってコマンドが異なるため，機種ごとの実装が必要となりシステムが複雑となる．しかし，SNMP の手法と異なり IPv6 マルチキャストでもコマンドを実行するだけで経路情報を取得することができる．

仮想的な PIM ネイバーとなるルータの設置

１台のルータを用意し，MtraceLG の管轄内全てのルータと PIM ネイバーとなる手法である．この際，ルーティングがされないように設定を行う．そのため，上記２つの手法と比べてルータの負荷が最も少ない手法と言える．さらに，新たに任意のルータを設置するため，リモートログインの手法を用いた場合もルータの機種にとらわれずにシステムを構築することができる．そのため，システムも複雑にならずに汎用性も高い．しかし，常時ルーティングに大きな支障を及ぼす恐れがあり，リスクが非常に高く，設置するコスト

もかかる.

上記３つの手法のメリット・デメリットから，本システムではIPv6のデータも安定して取得することができるリモートログイン手法をメインにしつつも，ネットワーク管理者が複数の選択肢から取得手法を選択できるようSNMP MIBの手法も取り入れることとする.

3.4 まとめ

本章では，本研究の目的とするIPマルチキャストネットッワークのモニタリング及び統計・分析に対する要求事項を述べ，その手法についての提案を行った．次章では，本章で述べた要求事項を満たすシステムの設計について述べる.

第4章　設計

本章では，本研究で提案するシステムの設計について述べる．本システムの動作についてモジュールごとにまとめ，説明する．

4.1　設計要件

本システムに必要となる機能は以下の通りである．

- LG ドメインのルータ管理機能
- ルータにリモートログインし，コマンドを実行して情報を取得する機能
- ルータの MIB から情報を取得する機能
- ホップバイホップで経路を遡る機能
- 経路探索のクエリを受けつける機能
- 一定時間ごとに自動的にデータを収集する機能

4.2　モジュール構成

本システムを構成するモジュールは以下の通りである．モジュール図を図 4.1 に示す．

HTTP Server

経路探索のクエリを受け付けるモジュールである．ルータ名を選択し，マルチキャストアドレス，選択したルータの下流方向の IP アドレスを渡すことで，後述の Mtrace Walker へとデータを渡して IP マルチキャストの経路探索を開始する．(S, G) の場合は送信者の IP アドレスも渡す．Mtrace Walker から返信を貰うと，その結果を返す．

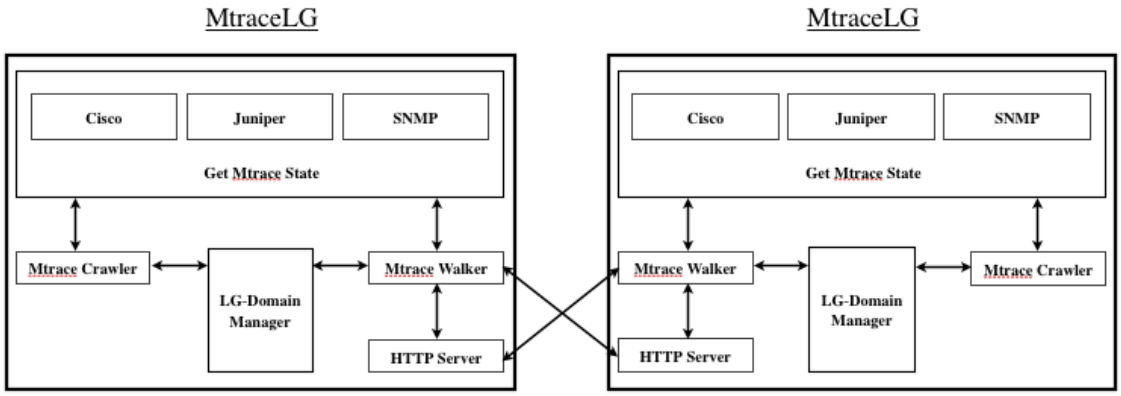

図 4.1: モジュール図

LG-Domain Manager

LG ドメイン内のルータ及び他の MtraceLG に関する情報が格納されているデータベースの管理を行うモジュールである．LG-Domain Manager が保持する情報を表 4.1 に示す．

表 4.1: LG-Domain Manager が保持する情報

ルータ情報	ルータの名前，ルータのアドレス， データを取得する手法（Cisco，Juniper，SNMP） データを取得するために必要な情報（パスワードなど）
他の MtraceLG 情報	他の MtraceLG のアドレス 他の MtraceLG が管理するルータのアドレス

Mtrace Walker からあるアドレスが自身の LG ドメイン内かどうかの検索の依頼を受けると，データベースを検索して LG ドメイン内かどうかの結果を返す．LG ドメイン内であった場合，そのアドレスからデータを取得するための手法も併せて返す．LG ドメイン外であった場合は，そのアドレスを管理する MtraceLG のアドレスを返す．どの MtraceLG が管理するか不明な場合は，他の MtraceLG へ問い合わせを行う．この際そのルータのアドレスを管理する LG が判明するので、データベースに保管する．また，Mtrace Crawler から自身が管轄するルータの探索依頼を受けると，LG ドメイン内の全てのルータのアドレスを返す．

Mtrace Walker

ラストホップルータからファーストホップルータまでホップバイホップで経路を遡っていくモジュールである．HTTP Server から受け取ったルータのアドレスを利用して，ルータからデータを取得するための情報の検索を LG-Domain Manager に依頼する．LG-Domain Manager から受け取ったデータ取得手法が Cisco へのリモートログインであった場合は Get Mtrace State (Cisco) へ，Juniper へのリモートログインであった場合は Get Mtrace State (Juniper) へ，SNMP であった場合は Get Mtrace State (SNMP) へと HTTP Server から受け取ったクエリとともにデータを取得するために必要な情報を渡す．

Get Mtrace State から返信を受け取ると，上流の隣接ルータのアドレスを LG-Domain Manager へ渡し，探索を依頼する．LG ドメイン内であった場合は，同様に適切な Get Mtrace State へと HTTP Server から受け取ったクエリとともにデータを取得するために必要な情報を渡す．LG ドメイン外であった場合は，他の MtraceLG へとクエリを投げる．

ファーストホップルータまで遡り終わった場合や他の MtraceLG に投げたクエリから返信があった場合は，その結果を HTTP Server へと返す．

Get Mtrace State (Cisco)

Cisco ルータにリモートログインを行いデータを取得するモジュールである．コマンドを実行して得たデータを整形して返す．Mtrace Walker からの依頼では，ルータのアドレス，ユーザーネーム，パスワード，プロンプト，マルチキャストアドレス、送信者のアドレス，下流方向のアドレスを受け取り，送信者のアドレスが存在すれば送信元ツリー (S, G) の探索，存在しなければ共有ツリー（*, G）の探索を行う．Mtrace Crawler からの依頼では，ルータのアドレス，ユーザーネーム，パスワード，プロンプトのみを受け取り，存在する全てのディストリビューションツリーに関するデータの探索を行う．

Get Mtrace State (Juniper)

Juniper ルータにリモートログインを行いデータを取得するモジュールである．動作は Get Mtrace State (Cisco) と同様であるため，省略する．

Get Mtrace State (SNMP)

ルータの MIB からデータを取得するモジュールである．snmpwalk で得たデータを整形して返す．データを取得するためには，v1，v2c の場合はルータのアドレス，コミュニティ，v3 の場合はルータのアドレス，ユーザーネーム，認証用パスワードが必要である．Mtrace Walker からの依頼では，マルチキャストアドレス，送信者のアドレス，ダウンストリームのアドレスも加えて受け取り，送信者のアドレスが存在すれば送信元ツリー (S, G) の探索，存在しなければ共有ツリー（*, G）の探索を行う．Mtrace Crawler からの依頼では，存在する全てのディストリビューションツリー関するデータの探索を行う．

Mtrace Crawler

一定時間ごとに自動でデータを収集するモジュールである．まず，LG-Domain Manager へ探索依頼を行い，LG ドメイン内の全てのルータのアドレス，データを取得する方法，データを取得するために必要な情報を受け取りリスト化する．そのリストを順々に参照し，データを取得する方法が Cisco へのリモートログインであった場合は Get Mtrace State (Cisco) へ，Juniper へのリモートログインであった場合は Get Mtrace State (Juniper) へ，SNMP であった場合は Get Mtrace State (SNMP) へと探索を依頼する．受け取ったデータはルータごとに時間別で保存する．

4.3 動作概要

本節では，本システムの動作概要をコミュニケーションフローを MtraceLG とルータ，MtraceLG 間，MtraceLG と Controller，MtraceCrawler にわけて説明を行う．

4.3.1 コミュニケーションフロー

前節で述べた各モジュールのコミュニケーションフローを箇条書きで述べる．また，全体コミュニケーションフローにおけるモジュールの関係及び動作概要を図 4.2 に示す．

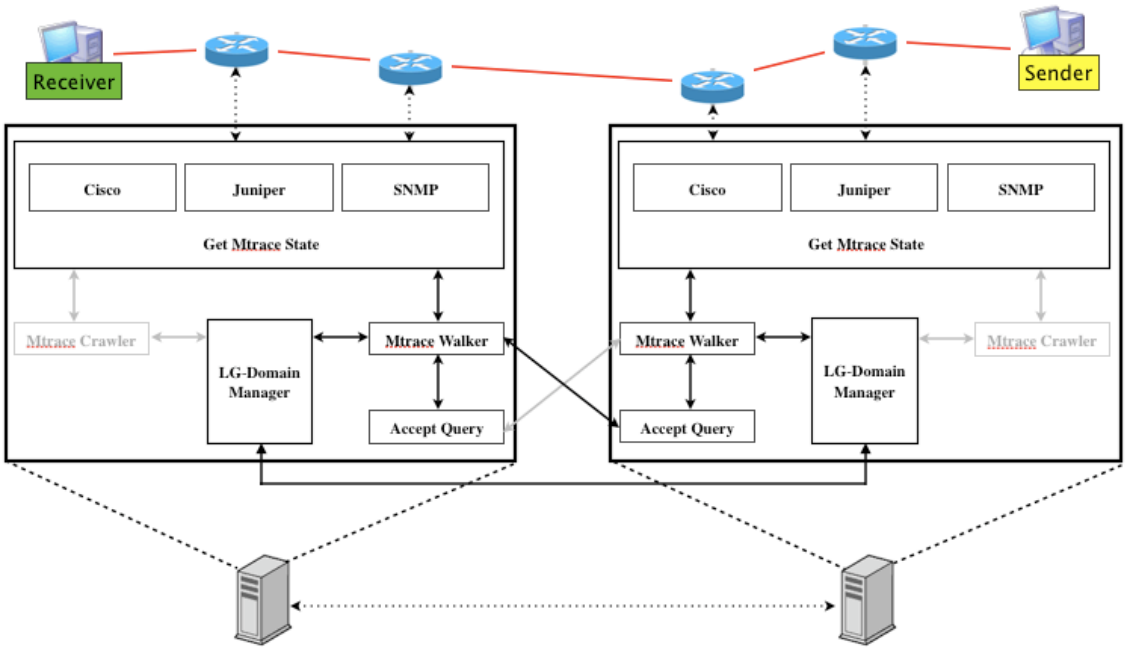

図 4.2: モジュール動作概要図

1. HTTP Server がクエリを受ける．
2. クエリを受けた HTTP Server は，ルータ名，マルチキャストアドレス，送信者のアドレス，そのルータの下流方向のアドレスを Mtrace Walker に渡す．
3. HTTP Server からクエリのデータを渡された Mtrace Walker は，ルータ名を LG-Domain Manager に渡し，ルータからデータを取得する手法及びそのために必要な情報の検索を依頼する．
4. LG-Domain Manager は，ルータ名からデータを取得する手法及びそのために必要な情報を Mtrace Walker へと返す．
5. LG-Domain Manager から返信を受けた Mtrace Walker は，ルータからデータを取得する手法に応じて Get Mtrace State の Cisco，Juniper，SNMP のいずれかへマルチキャストアドレス，送信者のアドレス，下流方向のアドレスを渡し，探索の依頼を行う．
6. 探索の依頼を受けた Get Mtrace State は，ルータから情報を取得し，Mtrace Walker へと返す．
7. Mtrace Walker は，上流方向の隣接ルータのアドレスがある場合は LG-Domain Manager に渡し，LG ドメイン内のルータかどうか判断する．上流方向の隣接ルータのアドレスが無い場合は，他の MtraceLG へ問い合わせを行った上で取得した情報を HTTP Server に結果を返す．
8. 上流方向の隣接ルータが LG ドメイン内である限り，5，7 を繰り返す．
9. LG-Domain Manager からのルータを管理する他の LG に関する返信，もしくは他の MtraceLG から問い合わせに対して返信があった場合，Mtrace Walker はその MtraceLG へクエリを投げる．LG ドメイン外であり，かつ他の MtraceLG で無かった場合，HTTP Server に結果を返す．
10. クエリを受け取った他の Mtrace LG は，1，9 を繰り返す．
11. 情報を取得した他の MtraceLG は，HTTP でクエリを投げた MtraceLG へと返す．

4.3.2 Mtrace Crawler

Mtrace Crawler とその他モジュールがどのように連携して動作するかを箇条書きで述べる．Mtrace Crawler と各モジュールの関係及び動作概要を図 4.3 に示す．

1. Mtrace Crawler は，crontab の設定により一定時間ごとに動作し，ホップバイホップでの探索は行わない．

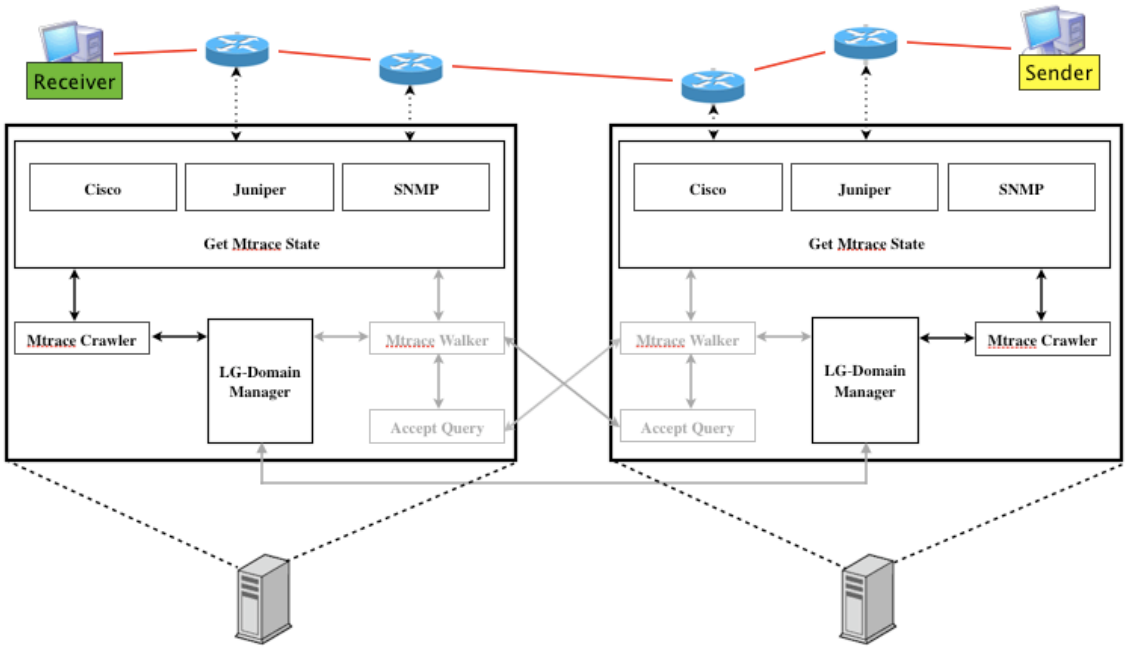

図 4.3: Mtrace Crawler で用いるモジュール

2. Mtrace Crawler が起動すると，まず始めに LG-Domain Manager に探索依頼を行い，LG ドメイン内の全てのルータのアドレスに関連した情報をリスト化する．
3. リストを順々に参照し，ルータからデータを取得する手法に応じて Get Mtrace State の Cisco，Juniper，SNMP へ探索依頼を行う．
4. 探索依頼を受けた Get Mtrace State は，ディストリビューションツリーをリスト化する．リストを順々に参照し，ルータから情報を取得し，Mtrace Crawler に返す．
5. 結果を受け取った Mtrace Crawler は，時間別にルータごとに探索結果を保存する．今回は，あらかじめ用意した Aggregate Server へ送信する．
6. crontab で設定するため，動作を一定時間ごとに半永久的に繰り返す．

4.4 まとめ

本章では，３章で述べた提案に基づき具体的な設計について述べた．次章では，本章の設計に基づいた実装を行う．

第5章　実装

本章では，４章で述べた設計を基に MtraceLG の実装について述べる．

5.1　実装概要

本節では，４章で述べた各モジュールごとに実装について述べる．

5.1.1　実装環境

実装環境を表 5.1 に示す．

表 5.1: ソフトウェア環境

OS	Fedora 14 2.6.35.6-45
プログラム言語	python 2.7
コンパイラ	GCC 4.5.1
データベース	SQLite 3.6.23.1

5.1.2　LG-Domain Manager

LG-Domain Manager は，データベースの集合体である．各モジュールは，必要に応じて LG-Domain Manager から情報を取得して利用する．あらかじめ LG-Domain Manager に情報を格納しておくことで，MtraceLG は機能することができる．データベースのテーブルごとに格納する情報を表 5.2 に示し，説明を行う．

ルータ ID

ルータ固有の ID である．int 型の数字が 1 から順に割り当てられる．

ルータ名

ルータの名前である．MtraceLG にルータを登録する際に，登録者が入力する．結果を表示する際に，ルータ名がともに表示される．

表 5.2: LG-Domain Manager テーブル一覧

router	ルータ ID，ルータ名，ルータの IP アドレス データの取得手法，アクセス手法
cisco_access	ルータ ID，ルータの IP アドレス，ユーザー名 パスワード，プロンプト
juniper_access	ルータ ID，ルータの IP アドレス，ユーザー名 パスワード，プロンプト
snmp_access	ルータ ID，ルータの IP アドレス，SNMP バージョン コミュニティ名（ユーザー名），v3 パスワード
peer	MtraceLG の ID，MtraceLG 名，MtraceLG の IP アドレス

ルータの IP アドレス

ルータの IP アドレスである．IPv4，IPv6 であるかは問わない．ネットワーク管理者がルータ登録時にアドレスを 1 つしか入力しなかった場合でも，データの取得手法に応じてルータのインタフェース全てのアドレスを取得して保存することができる．

データの取得手法

ルータからどのようにしてデータを取得するかを判断する．1 〜 3 の int 型が格納されている．今回の実装では，Cisco ルータへのリモートログインは 1，Juniper ルータへのリモートログインは 2，SNMP を用いる場合は 3 とした．

アクセス手法

自身の管轄内なのか，それとも他の MtraceLG なのかを 1 もしくは 2 の int 型で判断する．

ユーザー名

ルータにリモートログインする際に必要となるユーザー名である．ルータ登録時に必ず記載する．

パスワード

ルータにリモートログインする際に必要となるパスワードである．ルータ登録時に必ず記載する．

プロンプト

上述のユーザー名，パスワードを用いてリモートログインを行った際のルータのプロンプトである．プログラム内で利用するため，必ず必要となる．

SNMP バージョン

SNMP を用いてルータから情報を取得する際に必要となる．v1，v2c もしくは v3 が入る．ルータ登録時に必ず記載する．

コミュニティ名（ユーザー名）

SNMP を用いてルータから情報を取得する際に必要となる．SNMP バージョンが v1, v2 の場合はコミュニティ名が，v3 の場合はユーザー名が入る．

v3 パスワード

SNMP バージョンが v3 の場合に，ルータから情報を取得する際に必要となる．ルータ登録時に SNMP バージョンを v3 にした場合のみ，記載が必要となる．

MtraceLG の ID

MtraceLG 固有の ID である．int 型の数字が 1 から順に割り当てられる．

MtraceLG 名

MtraceLG の名前である．他の MtraceLG を登録する際に，登録者が記載する．

MtraceLG の IP アドレス

MtraceLG の IP アドレスである．登録者が，他の MtraceLG を設定する際に入力する．

5.1.3 HTTP Server

HTTP Server では，ルータ，マルチキャストアドレス，下流方向のアドレスをクエリとして受け取り，Mtrace Walker へと探索を依頼するモジュールである．(S, G) の場合は送信者のアドレスも渡す．クエリを受け取るウェブユーザインタフェースを図 5.1 に示す．

Mtrace Walker からは JSON 形式で返信を受け取る．探索結果が出力された状態を図 5.2 に示す．

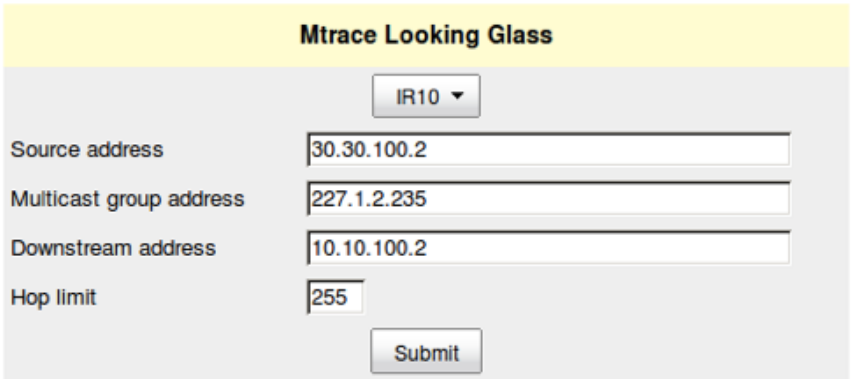

図 5.1: HTTP Server Web ユーザインタフェース

Results

Incoming Interface	Outgoing Interface	Previous Hop Router	Incoming Packets	Outgoing Packets	(S,G) Packets	Routing Protocol	Multicast Routing Protocol	Forwarding TTL	Source Mask	Forwarding Code
10.10.0.2	10.10.100.1	10.10.0.1	0	33812	0	0	0	1	0	NO_ERROR
192.168.10.5	10.10.0.1	192.168.10.6	0	34471	0	0	0	1	16	NO_ERROR
30.30.0.1	192.168.10.6	30.30.0.2	33916	0	0	0	0	1	24	NO_ERROR
30.30.100.1	30.30.0.2	30.30.100.2	0	33922	0	0	0	1	24	RPF_IF

図 5.2: HTTP Server 探索結果

5.1.4 Mtrace Walker

Mtrace Walker は，HTTP Server からクエリを受け取り，ホップバイホップでの経路探索を行うモジュールである．まず，引数としてクエリの値を受け取る．LG-Domain Manager のデータベースにセッションを張り，受け取ったクエリのルータ ID を利用して router テーブルからデータの取得手法を得る．また，データの取得手法を参照して cisco_access テーブル，juniper_access テーブル，snmp_access テーブルを適切に判断して，ルータからデータを取得する際に必要となる情報を得る．その後，Get Mtrace State にクエリとデータを取得する際に必要となる情報を渡す．その様子を図 5.3 に示す．

Get Mtrace State からの返信を辞書型で受け取る．ホップリミットを-1 した後，上流方向の隣接ルータのアドレスが無い，もしくはホップリミットが 0 の場合以外はホップバイホップでの探索を開始する．再び LG-Domain Manager の router テーブルから上流方向の隣接ルータのアドレスを基にルータ ID を取得し，そのルータ ID を利用してルータから情報を取得する手法を検索する．同じく router テーブルのアクセス手法から，上流方向の隣接ルータが自身の LG ドメイン内かどうかを判断し，管轄内であれば同様に Get Mtrace State へと依頼し，管轄外であれば他の LG ドメインへ HTTP を利用してクエリを POST する．その様子を図 5.4 に示す．

```
def traverse_routers(router_id, query_id, source,
                         mcast_group, downstream_addr, querier, hop_limit):
    global Session
    router_record = Session.query(RouterTable).get(router_id)

<<中略>>

    if router_record.rt_type == RT_CISCO:
        router = routerCisco.Router(router_record.access_param(Session))
    elif router_record.rt_type == RT_JUNIPER:
        router = routerJuniper.Router(router_record.access_param(Session))
    elif router_record.rt_type == RT_SNMP:
      router = routerSNMP.Router(router_record.access_param(Session))
```

図 5.3: Mtrace Walker -クエリの受信〜Get Mtrace State への依頼-

5.1.5 Get Mtrace State (Cisco)

Cisco ルータへのリモートログインで情報を取得するべき場合に，Mtrace Walker から依頼を受けるモジュールである．telnet でリモートログインを行い，コマンドを実行することで情報を取得する．実行結果を正規表現で整形し，辞書型で Mtrace Walker へと返す．

Cisco ルータで実行するコマンドを下記に示す．

- show ip mroute
- show ip rpf
- show ip mroute count
- show ip interface brief
- show ip pim rp mapping
- show ip interface brief — include
- show ipv6 mroute
- show ipv6 interface
- show ipv6 mroute count

```
response_dict = router.mtrace(family, source, mcast_group, downstream_addr)

<<中略>>

    if addr_record and addr_record.type == ADDR_TYPE_ROUTER:
        router_record = Session.query(RouterTable).get(addr_record.id)

        if router_record.rt_type == RT_CISCO:
            router = routerCisco.Router(router_record.access_param(Session))
        elif router_record.rt_type == RT_JUNIPER:
            router = routerJuniper.Router(router_record.access_param(Session))
        elif router_record.rt_type == RT_SNMP:
            router = routerSNMP.Router(router_record.access_param(Session))

<<中略>>

    if addr_record and addr_record.type == ADDR_TYPE_PEER_RT:

<<中略>>

        conn = httplib.httplib.HTTPConnection(peer_address,
                                              peer.port, timeout=10)
        conn.request('POST', url_prefix + '/request/', request_body, http_header)
        resp = conn.getresponse()
        answer = json.loads(resp.read())
```

図 5.4: Mtrace Walker -ホップバイホップでの探索-

- show ipv6 pim group-map
- show ipv6 rpf
- show ipv6 neighbor
- show ipv6 pim neighbor
- show ipv6 pim neighbor — include
- show ipv6 pim neighbor detail
- show ipv6 interface brief

5.1.6 Get Mtrace State (Juniper)

Juniper ルータへのリモートログインで情報を取得するべき場合に，Mtrace Walker から依頼を受けるモジュールである．動作は Get Mtrace State (Cisco) と同様で，辞書型で Mtrace Walker へと返す．

Juniper ルータで実行するコマンドを下記に示す．

- show multicast route
- show multicast rpf
- show interface
- show multicast route detail
- show pim rps
- show route detail

5.1.7 Get Mtrace State (SNMP)

リモートログインでは無く，MIB から情報を取得するべき場合に Mtrace Walker から依頼を受ける．OID を参照することで情報を取得する．実行結果を正規表現で整形し，辞書型で Mtrace Walker へと返す．

参照する MIB の OID を下記に示す．

- ipMRouteUpstreamNeighbor 1.3.6.1.2.1.83.1.1.2.1.4
- ipMRouteRtMask 1.3.6.1.2.1.83.1.1.2.1.14
- ipMRouteUpTime 1.3.6.1.2.1.83.1.1.2.1.6
- ipMRoutePkts 1.3.6.1.2.1.83.1.1.2.1.8
- ipMRouteDifferentInIfPackets 1.3.6.1.2.1.83.1.1.2.1.9
- ipMRouteNextHopPkts 1.3.6.1.2.1.83.1.1.3.1.11
- ipMRouteInterfaceInMcastOctets 1.3.6.1.2.1.83.1.1.4.1.5
- ipMRouteInterfaceOutMcastOctets 1.3.6.1.2.1.83.1.1.4.1.6
- pimRPSetGroupAddress 1.3.6.1.3.61.1.1.6.1.1
- pimRPSetAddress 1.3.6.1.3.61.1.1.6.1.3
- pimComponentBSRAddress 1.3.6.1.3.61.1.1.12.1.2

- ifName 1.3.6.1.2.1.31.1.1.1.1
- ifInMulticastPkts 1.3.6.1.2.1.31.1.1.1.2
- ifOutMulticastPkts 1.3.6.1.2.1.31.1.1.1.4
- ipAdEntIfIndex 1.3.6.1.2.1.4.20.1.2
- ipAdEntNetMask 1.3.6.1.2.1.4.20.1.3

5.1.8 Mtrace Crawler

Mtrace Crawler は，Mtrace Walker の一部機能を用いて一定時間ごとに自動で情報を取得するモジュールである．Mtrace Walker では，上流方向の隣接ルータのアドレスが無い場合，もしくはホップリミットが 0 の場合以外はホップバイホップでの経路探索を行っていたが，Mtrace Crawler ではホップバイホップでの経路探索は行わない．あらかじめ LG ドメイン内のルータに関する情報をリスト化し，ルータ 1 台ごとに Get Mtrace State へ情報の探索依頼を行う．

まず始めに LG-Domain Manager のデータベースから自身が管轄するルータをリスト化して取得する．リスト化したルータごとにディストリビューションツリーを取得し，while 文を利用して 1 つずつ Get Mtrace State へと依頼を行う．ディストリビューションツリーごとに Get Mtrace State へ依頼したときに，ホップバイホップでの経路探索を行わないプログラムを図 5.5 に示す．

Get Mtrae State から取得した情報は，ディストリビューションツリーごとにテキストファイル化して集積サーバーである Aggregate Server へと送信を行う．具体的には，Get Mtrace State から辞書型で返信を受け取り，テキストファイルに書き起す．そのファイルを SCP を利用して Aggregate Server へと送信する．その様子を図 5.8 に示す．OS の crontab を設定することで，Mtrace Crawler を一定時間ごとに起動して自動収集を行うことが可能となる．

5.2 まとめ

本章では，4 章で提案した設計を基に行った実装について述べた．次章では，本章で行った実装を実環境で動かした際の評価並びに取得した情報の分析を行う．

```
def traverse_routers_crawler(router_id, query_id, source, mcast_group):
    global Session
    Session = init_db()
    router_record = Session.query(RouterTable).get(router_id)

    mcast_ip = ipaddr.IPAddress(mcast_group)

    if mcast_ip.version == 4:
        family = AF_INET
    else:
        family = AF_INET6

    json_list = []
    router = None

    if router_record.rt_type == RT_CISCO:
        router = routerCisco.Router(router_record.access_param(Session))
    elif router_record.rt_type == RT_JUNIPER:
        router = routerJuniper.Router(router_record.access_param(Session))
    elif router_record.rt_type == RT_SNMP:
       router = routerSNMP.Router(router_record.access_param(Session))

    return(router.mtrace(family, source, mcast_group))
```

図 5.5: Mtrace Crawler -ホップバイホップでの探索は行わない-

```
f = open(file, 'w')

<<中略>>

for (k, v) in json_list.items():
        txt = k + ": " + str(v)
        f.write(txt)

f.close()
scp.sendline('scp -r %s %s@%s:%s' % (file, username, server, dir))
```

図 5.6: Mtrace Crawler -Aggregate Server への送信-

第6章　評価

本章では，５章において実装したMtraceLGの評価について述べる，また，MtraceLGを用いて一定時間ごとに蓄積したデータの分析も行う．

6.1　評価概要

本評価は，本研究の目的とするIPマルチキャストネットワークのリアルタイムでのモニタリングの有効性の検証，及び一定時間ごとに蓄積したデータの分析を目的とする．

定性評価及び定量評価は，以下の項目について行う．

- 定性評価： 設計に基づき実装を行ったMtraceLGの実現した機能
- 定量評価： MtraceLGを用いて一定時間ごとに蓄積したデータの分析

6.2　評価環境

本評価は，以下の実環境において行う．

表 6.1: 評価環境

	MtraceLG	ルータ	データ取得手法
WIDE	mcast.fujisawa.wide.ad.jp	cisco2.notemachi.wide.ad.jp cisco2.fujisawa.wide.ad.jp juniper1.otemachi.wide.ad.jp	リモートログイン
AI3	203.178.143.210	sfc-gate.ai3.net sfc-gate2.ai3.net	リモートログイン
APAN	mtrace2.jp.apan.net	tpr5.jp.apan.net tpr6.jp.apan.net tyo-t1600.jp.apan.net	SNMP

6.3　評価項目

本節では，IPマルチキャストネットワークのリアルタイムでのモニタリング及び一定時間ごとに蓄積したデータの分析についてそれぞれ述べる．

6.3.1 リアルタイムでのモニタリング

実環境において，リアルタイムでのモニタリングを行えるようにした．リモートログイン方式でデータを取得する場合，IPv6 マルチキャストネットワークの経路探索も行うことができる．その様子を図 6.1 に示す．

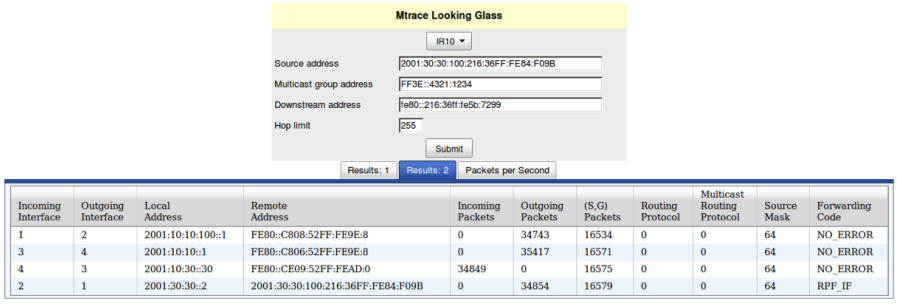

図 6.1: MtraceLG を利用したリアルタイムの経路探索

MtraceLG の利用者は，ルータを選択し，送信者のアドレス，マルチキャストアドレス，ルータから向かって下流方向のアドレスを入力するだけで利用することができる．例えば参加していた（S, G）のデータを受け取ることができなくなった場合にその原因を即座に調べることもできる．その他にも，これから参加したい（*, G）がどのような経路を通っているのか，しっかりとデータを受け取ることができるのかも調べることができる．MtraceLG を利用したモニタリングにより，今までに無かった詳細な IP マルチキャストの経路情報をリアルタイムに取得することが可能となった．

6.3.2 蓄積したデータの分析

分析を行う項目として以下を挙げる．なお，2011 年 12 月 10 日午前 0 時から 2012 年 1 月 31 日午後 24 時の期間で MtraceCraler を動作させて取得した情報を用いて分析を行った．

マップの作成

いつどのような経路が構成されたかのイメージを図に落とし込めるように，Mtrace-Crawler で取得した情報をさらに抽出・整形した．このデータを利用して作成したイメージマップの例を図 6.2 に示す．

線の色によって (S, G) の違いを判別できるようにし，線の太さがパケットカウントを表している．ルータごとに取得した情報から，(S, G) ごとに IIF、OIF，上流方向の隣接ルータのアドレス，パケットカウントを抽出することで図 6.2 のようなマップを作成することができる．本研究ではシステムの挙動上 MtraceCrawler を 10 分単位で動作させたが，作成したイメージマップを見ることで 10 分単位の変化を観察することができる．その様子を図 6.3 及び図 6.4 に示す．

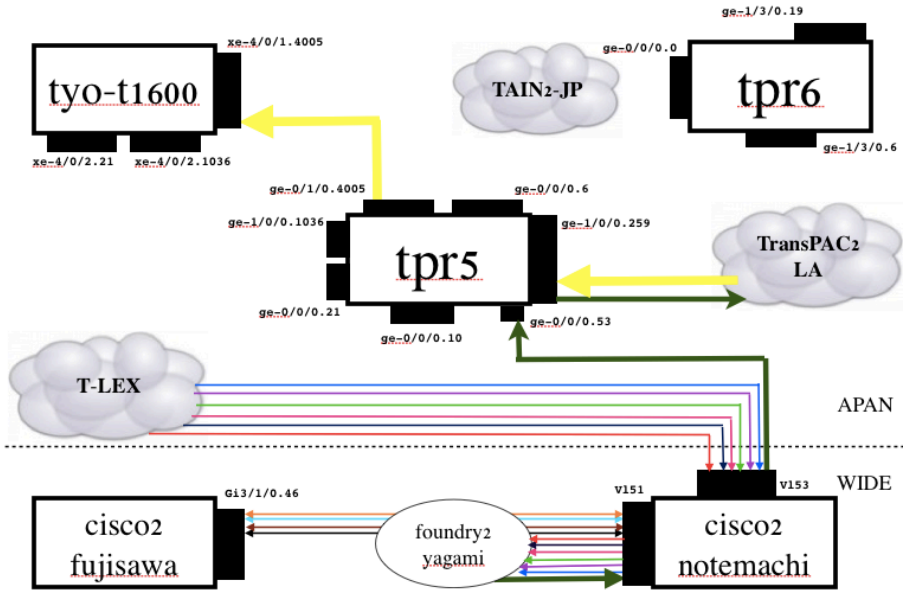

図 6.2: イメージマップの例（2012 年 1 月 30 日 14 時）

図 6.3 及び図 6.4 を閲覧するだけで，様々な情報を得ることができる．そのため，このマップを作成するために抽出し，利用したデータを基に分析を行った．

フラッピングの計測

例えばフラッピングの計測がその一つである．10 分ごとのマップの移り変わりを閲覧することで，経路の切り替わりを見て取ることができる．本研究では 10 分単位での観察しか行うことができなかったため，フラッピングの例を示すことをできなかったが，MtraceCrawler による取得間隔を狭めていくことによってフラッピング計測の可能性を示すことはできた．フラッピングは頻繁に経路が切り替わることであり，不安定な経路において起こる現象である．そのため，フラッピングの計測を行えるようになることで今後のネットワークデザインに大いにこの情報を生かすことが可能となる．なお，マップを作成する際に利用した (S, G) ごとの IIF，OIF，上流方向の近接ルータを観察することで明らかにすることができる．

ルータごとのディストリビューションツリー

マップを見ることで，その時間のルータを流れているディストリビューションツリーを視覚的に観察することができる．10 分単位でその数を観察することで，極端に減少したとき及び増加したときを見て取ることができる．tpr5.jp.apan.net における 2011 年 12 月 27 日の例を図 6.5 に示す．

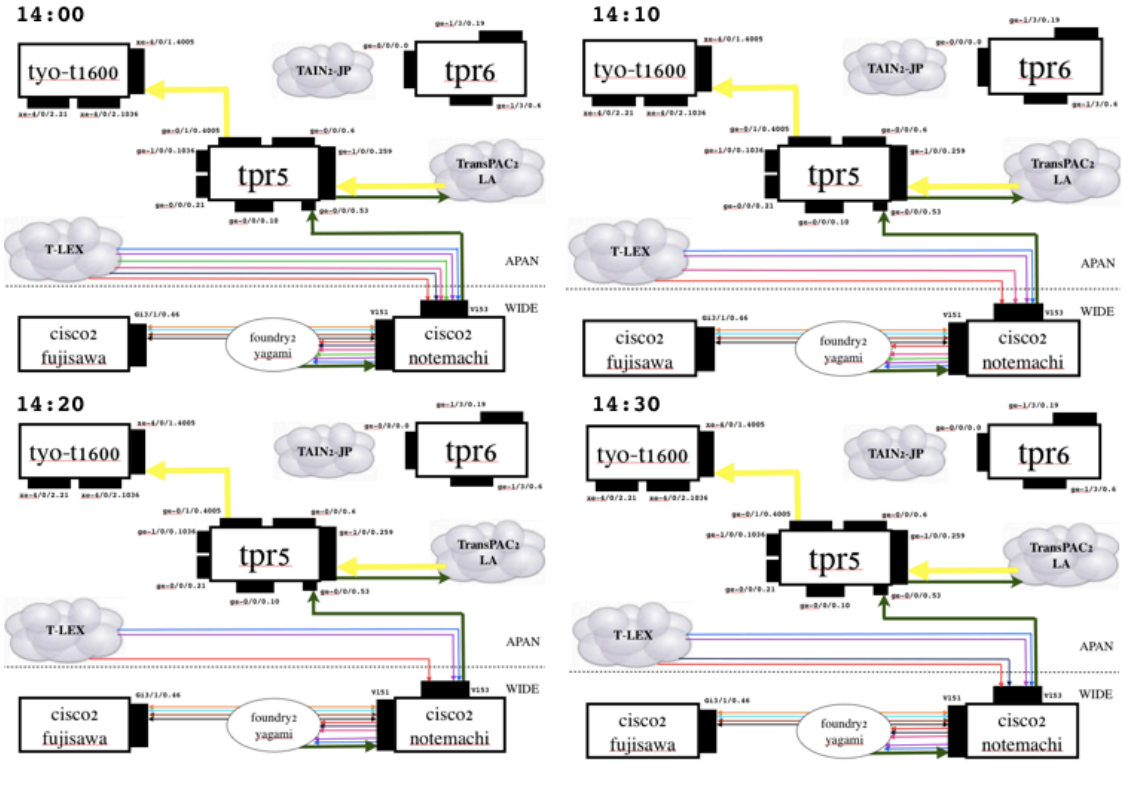

図 6.3: イメージマップの移り変わり（2012 年 1 月 30 日 14 時〜14 時 30 分）

図 6.5 では，10 時 30 分から 10 時 50 分にかけて極端にディストリビューションツリーの数が減少していることがわかる．これは後から判明したことだが，当該時刻にルータのバージョンアップを行っていた．このようにルータのリブートが起きた場合は，ユニキャストにおいても支障が出るためすぐに ping や traceroute ですぐに判断することができる．しかし，図 6.5 の様な急激な減少が起きた際に ping や traceroute で通常通りの反応を得た場合は IP マルチキャストのルーティングプロトコルに障害が起きた可能性が挙げられる．また、計測期間中は確認することができなかったが，図 6.5 とは反対に急激な増加があった場合，IP マルチキャストを利用した DoS 攻撃である可能性を疑うことができる．急激な増減が起きた際に，その前後の (S, G) を見ることで判別が可能であると考えることができる．例えば，あるマルチキャストアドレスを利用した同じサブネットに属する送信者が急激に増加した場合，DoS 攻撃の可能性は高いと言える．

SAP に告知を行う送信者のライフタイム

先ほどのマップでは，複雑になるため SAP のアドレスを省いて作成した．今回は SAP のみのデータを抽出した例を図 6.6 に示す．

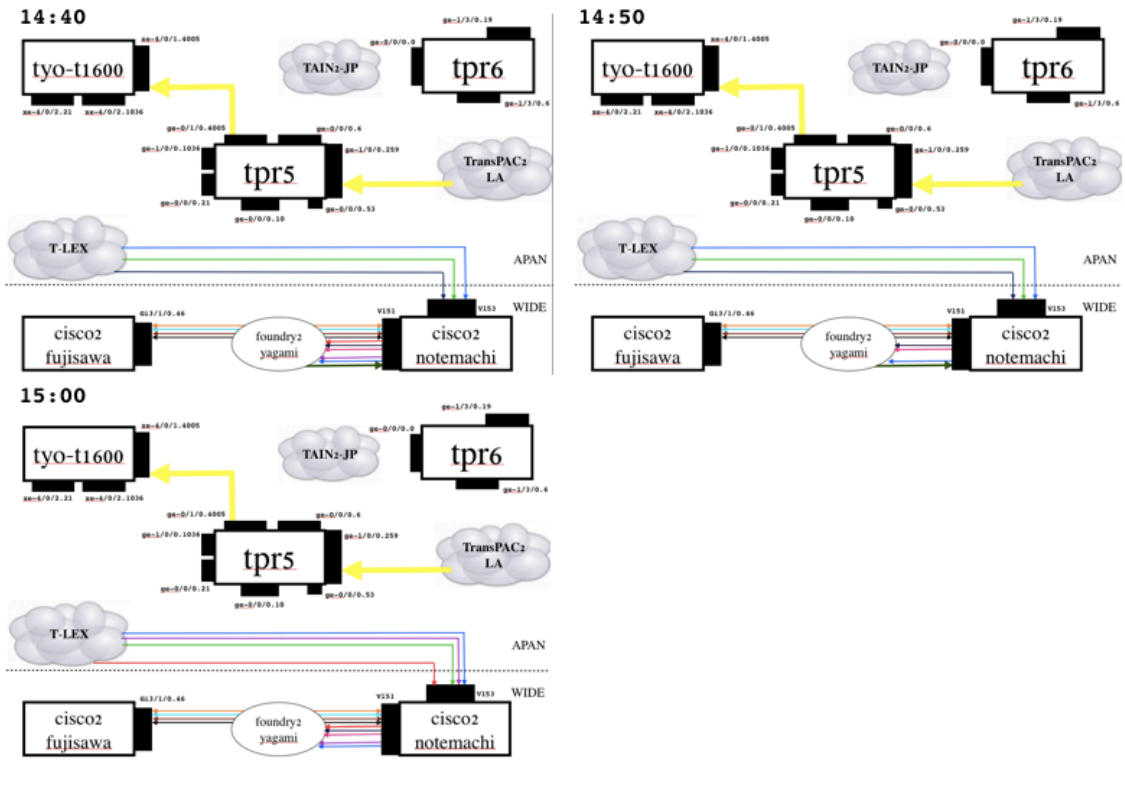

図 6.4: イメージマップの移り変わり（2012 年 1 月 30 日 14 時 40 分〜15 時）

図 6.6 を 10 分ごとに観察することで，SAP を利用して告知を行っている送信者が存在する方向やライフタイムを計測することが可能である．計測期間中に確認することができた送信者をドメイン別にまとめ，出現回数，合計ライフタイム，合計ライフタイムを出現回数で割ったライフタイムの平均時間をまとめた表を図 6.7 に示す．また，合計ライフタイムとライフタイムの平均をグラフにした様子を図 6.8 に示す．

図 6.8 では，緑色の棒グラフが合計時間，青色の折れ線グラフが平均時間を示している．緑色の合計時間が多いにも関わらず青色の点が低い位置にある場合，その送信者が何回も現れては消える，という特性があることを読み取ることができる．図では，右側に推移するほど，つまり合計時間が少ないほど平均ライフタイムが増える場合が多い．この原因の詳細を示していくことも今後求められる．

ASM における送信者のライフタイム

SAP における送信者の詳細を見ることができたが，もちろん同様のデータを利用して SAP 以外の送信者の詳細に関しても明らかにすることができる．ASM における複数送信者のライフタイムをグラフにした例として 2011 年 12 月 11 日の tpr5.jp.apan.net におけるマルチキャストアドレス 224.112.56.29 エントリを図 6.9 に示す．

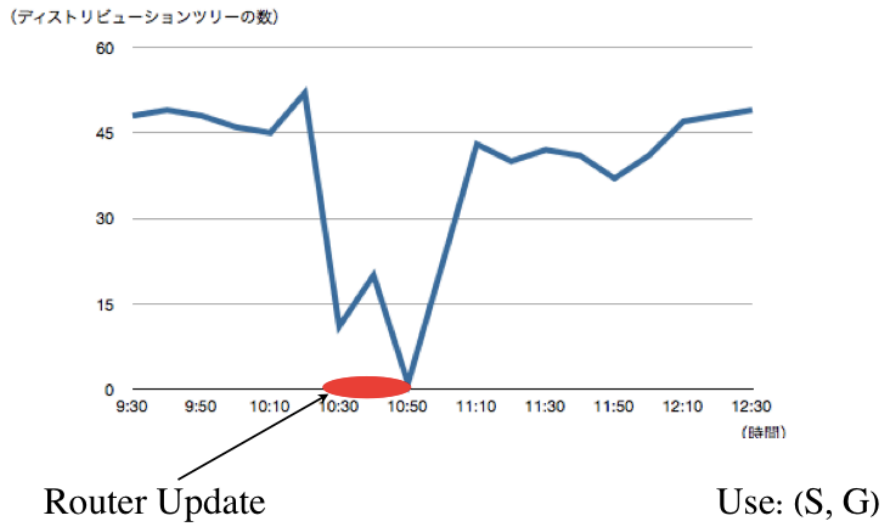

図 6.5: tpr5.jp.apan.net における例（2011 年 12 月 27 日）

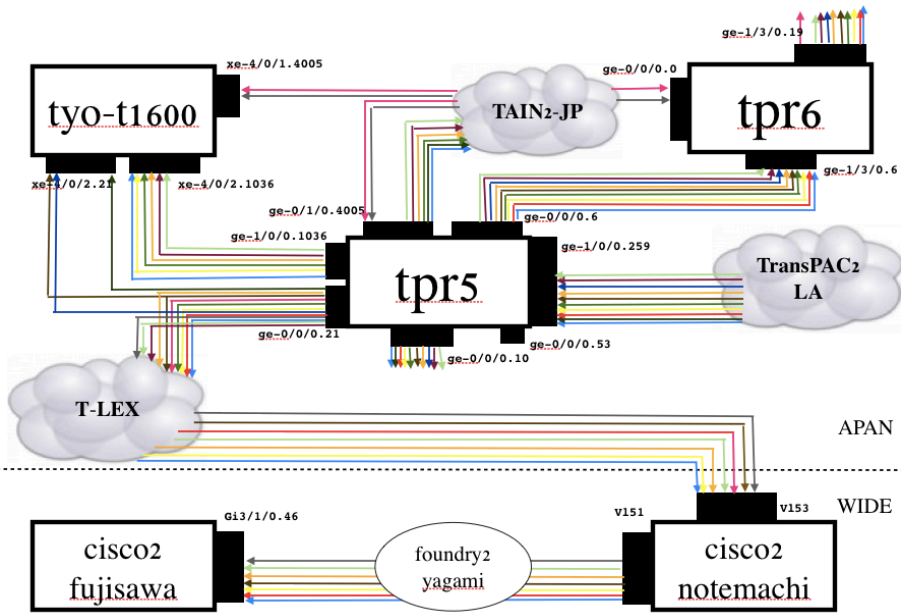

図 6.6: SAP のみのイメージマップ（2012 年 1 月 30 日 14 時）

図 6.9 のグラフから送信者の詳細を見ることができ，どの送信者が同時間帯に同じマルチキャストアドレス宛にデータを送信していたかがわかる．「ルータごとのディストリビューションツリー」で述べたように DoS かどうかを判断する際の情報として利用することができる．図 6.9 における送信者は全て washington.edu のネットワークに属する．

ドメイン	出現回数（回）	合計時間（分）
heanet.ie	884	144270
cam.ac.uk	596	126550
ku.sk	659	94080
utwente.nl	107	53240
edu.ve	347	47910
Pennsylvania State University	395	46900
HZeeland.nl	326	45910
Athens Technical University	461	44900
memphis.edu	438	44570
stuba.sk	520	42760
soton.ac.uk	402	40270
upv.es	253	39600
Ohio State University	78	39060
Tampa University	102	35320
Milton University	207	34050
RWTH-Aachen.DE	63	30060
lut.ac.uk	150	26210
ethz.ch	34	21930
uca.es	115	19930
nrc.ca	42	19190
Renater	100	18550

ドメイン	出現回数（回）	合計時間（分）
upenn.edu	37	17950
New London Connecticut College	37	17040
Carson City University	35	16380
uib.no	24	12280
rediris.es	25	11750
rit.edu	18	9500
uit.no	18	9460
wisc.edu	22	9320
Marine Life Aquarium	11	8880
uni-erlangen.de	23	7,680
umich.edu	19	6800
ayy.fi	10	6750
cornell.edu	10	5050
columbia.edu	4	1470
MIT.EDU	5	1460
ucsc.edu	3	1130
washington.edu	3	850
northwestern.edu	7	690
Iowa City University	3	550
Madrid Universidad	17	430
Honolulu University	3	320

図 6.7: 計測期間中のドメイン別 SAP 送信者のライフタイム

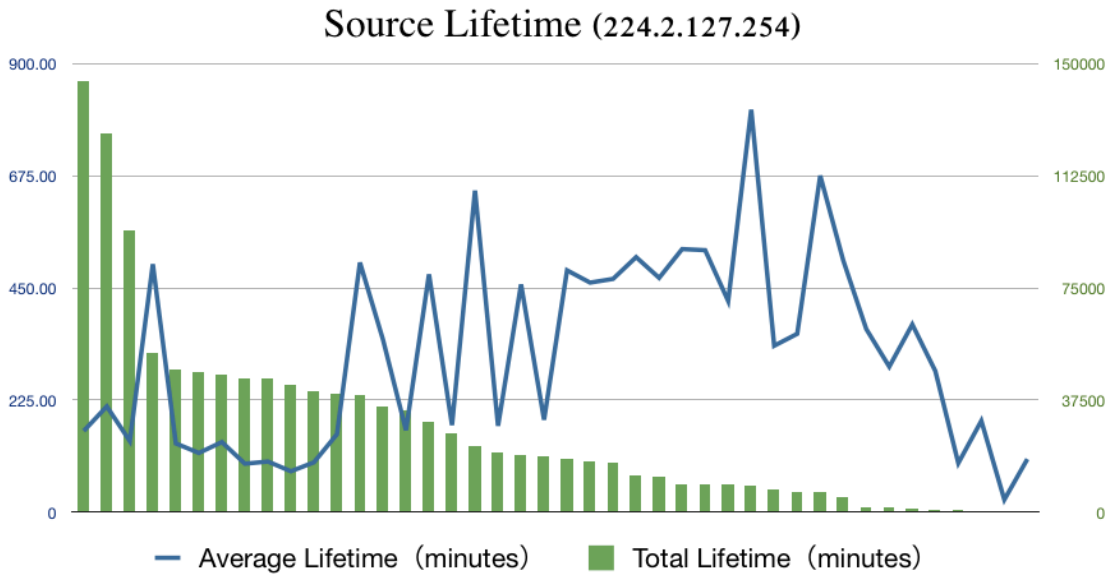

図 6.8: 送信者別のライフタイムの合計時間と平均時間

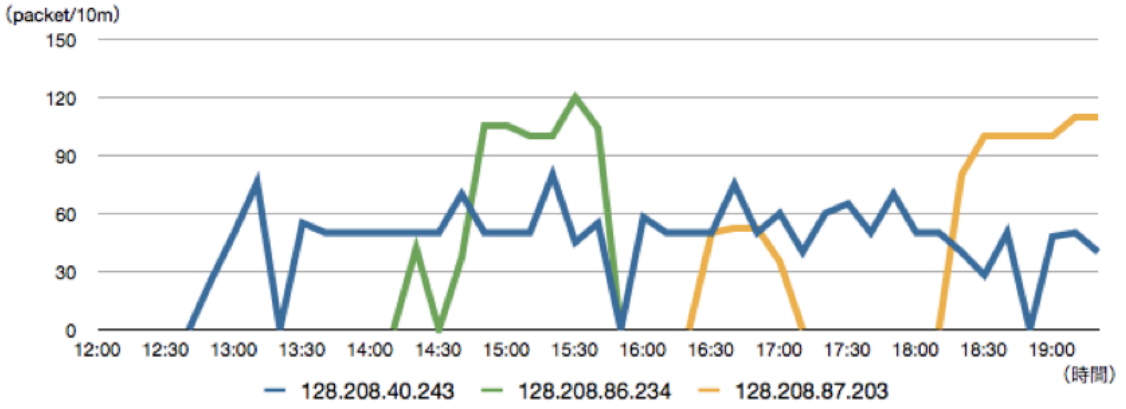

図 6.9: 2011 年 12 月 11 日 224.112.56.29 エントリ（tpr5.jp.apan.net）

縦軸がパケットカウントを表しているが，このパケットカウントが急激に上昇して非常に高い値を示していたり，常に一定であったりした場合に DoS 攻撃を疑うことができる．図 6.9 の場合は，パケットカウントが少ないため DoS 攻撃の可能性は低いと考えられるが SA メッセージの脆弱性をついた攻撃である可能性を否定することができない．そこで，2012 年 1 月 5 日の tpr5.jp.apan.net における 224.112.56.29 エントリを図 6.10 に示す．

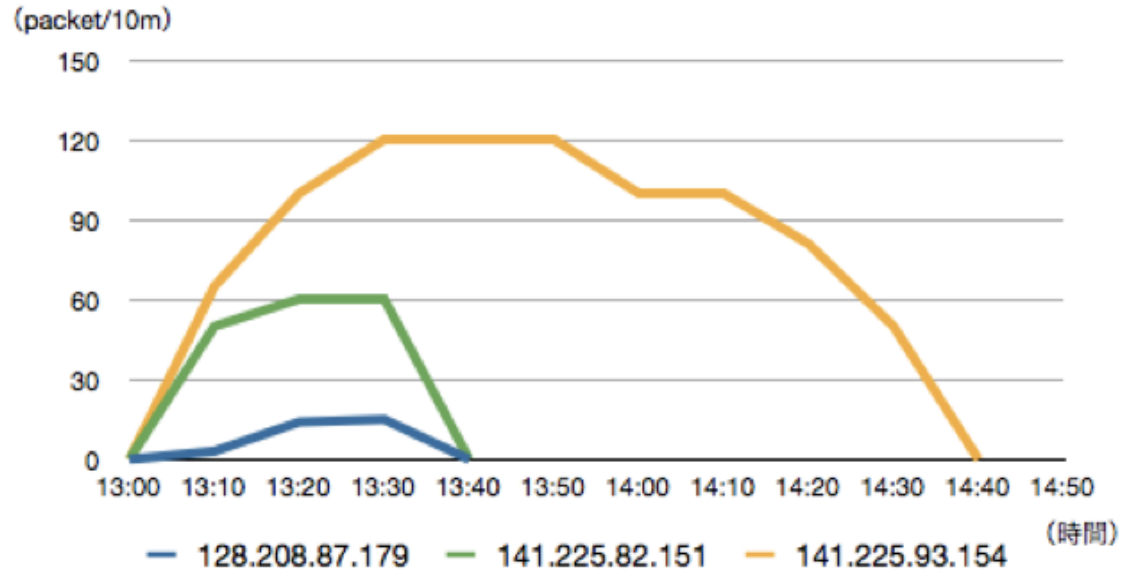

図 6.10: 2012 年 1 月 5 日 224.112.56.29 エントリ（tpr5.jp.apan.net）

図 6.10 では，washington.edu もしくは memphis.edu のネットワークに属する送信者が存在している．そのため，パケットカウントは少ないながらもこの場合はオンラインミーティングなど何らかの情報を実際にやり取りしていたのではないかと推測することができる．なお，これらの送信者はマップより tpr5.jp.apan.net からみて Internet2 方面に位置しており，この方面を上流として下流に受信者がいたことがわかる．

AS を跨ぐマルチキャストルーティングの経路探索

マップでは，AS を跨ぐマルチキャストルーティングの経路及びその移り変わりを見ることができた．マップを作成するために抽出，利用したデータの詳細を制御が難しい AS を跨ぐ経路に特化して観察することが可能である．ここでは，tpr5.jp.apan.net における 2011 年 12 月 12 日 18 時 30 分過ぎの (203.178.138.4, 226.94.1.1) エントリの詳細を表 6.1 に示す．

表 6.2: (203.178.138.4, 226.94.1.1) エントリ -tpr5.jp.apan.net-

Time	Previous Hop Router	Outgoing Interface	Packets
18:30	203.178.133.142	192.203.116.146/30 (ge-1/0/0.259)	966580
18:40	203.178.133.142	192.203.116.146/30 (ge-1/0/0.259)	967802
18:50	203.178.133.142	192.203.116.146/30 (ge-1/0/0.259)	969119
19:00	203.178.133.142	192.203.116.146/30 (ge-1/0/0.259)	971398

表 6.1 から，tpr5.jp.apan.net は (203.178.138.4, 226.94.1.1) のパケットを 203.178.133.142 から受信し，192.203.116.146/30 (ge-1/0/0.259) へと送信していることがわかる．
ここで，cisco2.notemachi.wide.ad.jp における 2011 年 12 月 12 日 18 時 30 分過ぎの (203.178.138.4, 226.94.1.1) エントリの詳細を表 6.2 に示す．

表 6.3: (203.178.138.4, 226.94.1.1) エントリ -cisco2.notemachi.wide.ad.jp

Time	Previous Hop Router	Outgoing Interface	Packets
18:30	203.178.141.141	203.178.133.142	14590359
18:40	203.178.141.141	203.178.133.142	14591603
18:50	203.178.141.141	203.178.133.142	14592887
19:00	203.178.141.141	203.178.133.142	14594041

表 6.2 から，cisco2.notemachi.wide.ad.jp は (203.178.138.4, 226.94.1.1) のパケットを 203.178.141.141 から受信し，203.178.133.142 へと送信していることがわかる．表 6.9 及び表 6.10 から tpr5.jp.apan.net が示す上流方向の隣接ルータのアドレスと，cisco2.notemachi.wide.ad.jp の OIF が一致していることも読み取れる．以上のことから，(203.178.138.4, 226.94.1.1)

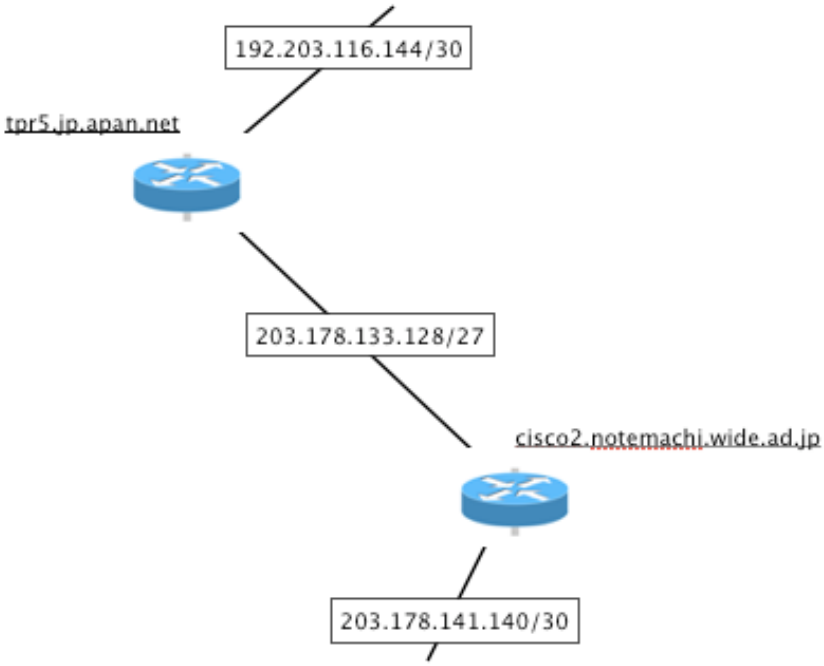

図 6.11: WIDE から APAN へ流れる (203.178.138.4, 226.94.1.1)

エントリは WIDE から APAN を通って受信者へと送信されていることがわかる．その様子を図 6.11 に示す．

このように，下流方向のルータにおける 1 つ手前のルータアドレスを取得して，他のルータに存在する同じ (S, G) エントリの OIF と照合することで経路を繋ぎ合わせて分析することができる．

第7章　結論

本章では，本研究の成果をまとめ，今後の課題及び展望を述べる．

7.1　まとめ

本研究では，IP マルチキャストの経路情報をリアルタイムに取得できる MtraceLG の実装を行った．これにより，リアルタイムでの IP マルチキャストのネットワークをモニタリングでき，リアルタイムに障害の発生地点を明らかにすることが可能となった．

また，MtraceCrawler によって一定時間ごとにマルチキャスト経路情報を自動収集し，分析を行った．実ネットワークにおいてどのようなディストリビューションツリーが構築されているかを時間単位で把握することで，今まで明らかにすることができなかった IP マルチキャストの利用実態を解明することが可能となった．これにより，フラッピング，マルチキャストルーティングプロトコルの障害，DoS 攻撃検知のツールとして非常に有用な可能性を示すことができた．

7.2　今後の課題と展望

本研究における今後の課題及び展望として，以下の項目が挙げられる．

- モニタリング環境の拡大
  今まで把握することができなかった IP マルチキャストの利用実態を解明できるようになったことにより，障害の検知・対応をはじめとした運用及びネットワークデザインに生かすことが可能となった．実環境において構築されているディストリビューションツリーは決して多くなかったが，今後 MtraceLG を普及させていくことでより素早い対応を行うことが可能となるため，モニタリング環境の拡大は必要である．また、モニタリング環境が拡大することにより，MtraceCrawler での統計・分析を広範囲にわたって行うことが可能となる．特に，AS を跨ぐマルチキャストルーティングの統計データを取得することで，更なる有効な分析を行うことができるようになる．

- MtraceLG 間コミュニケーションのセキュリティ機能向上
  前項で挙げたモニタリング環境の拡大を目指す上で，セキュリティ機能の向上は欠かすことができない．今回の実装では，異なる AS に配置された MtraceLG 同士

もあらかじめ接続することによってASを跨いだ経路探索を可能にした．同AS内であれば問題は無いが，異なるASに配置されたMtraceLG同士が直接接続してルータの情報をやり取りすることは安全とは言えない．ユーザー名やパスワードといった直接ルータにアクセスできる情報のやり取りは行わないが，その対策は必要不可欠である．MtraceCrawlerが自動収集した情報をAggregate Serverに送信しているが，ASを跨ぐ際はこれと同じようにあらかじめ用意したサーバを介して通信を行うようにする，などの方法が挙げられる．

- MtraceCrawlerの自動収集時間最適化

  前章の評価における設定時間は10分であった．その理由としては，プログラムの動作時間，ルータ・ネットワークの負荷を考慮してのことである．しかしながら，ディストリビューションツリーの構築の様子を分析したり，ディストリビューションツリーごとのパケットカウントを利用して分析を行おうとした場合，10分という設定時間では満足な分析を行うことができない．例えば，ASMの場合RPTからSPTへと切り替わる様子を分析したくても10分という時間を要することはほとんどなく，パケットカウントも秒単位で上下する値である，そのため，プログラムの動作時間やルータ・ネットワークの負荷を考慮しつつも秒単位での観察ができるような仕組みを構築することが必要である．その方法としては，マルチキャストアドレスと送信者のアドレス，IIF・OIF，そのパケットカウントのみは30秒置きなど短い間隔でルータごとに取得し，その他の情報は10分置きに取得するなど，取得する情報ごとに間隔を設定できるようにすることが考えられる．

- リアルタイムでの自動マッピング

  本研究ではMtraceCrawlerによって取得した情報をさらに整形，抽出することでイメージマップを作成することができた．これと同様に，リアルタイムに自動でマッピングすることが可能となればネットワーク管理者に取って運営を行う上で非常に有用な監視ツールとなる．このリアルタイムで作成したマッピングの蓄積によって，前章で述べたものと同様の分析を行うことが可能である．つまり，監視と分析を兼ね備えたマルチキャストツールとしてさらに価値のあるものとすることができる．

謝辞

本研究を進めるにあたり絶えずご指導を頂きました，慶應義塾大学大学院政策・メディア研究科准教授朝枝仁博士及び mistral メンバーである慶應義塾大学大学院政策・メディア研究科後期博士課程久松剛氏，三島和宏氏，松園和久氏に深く感謝致します．また，徳田・村井・楠本・中村・高汐・バンミーター・植原・三次・中澤・武田合同研究プロジェクトにおいて多くのご指導と助言を頂きました慶應義塾大学環境情報学部長村井純博士，同大学中村修博士，本研究を進めるにあたり多大なご協力を頂いた APAN の田原裕一郎氏，慶應義塾大学大学院政策・メディア研究科後期博士課程アフマド・バスキ氏，堀場勝広氏，AI3 のアン・ウェイ・チャン氏に感謝致します．

そして，本研究を進めるにあたり様々な励ましと助言を頂きました mistralOB の峯木厳氏，内田陽豪氏，現メンバーの佐藤淳哉氏，数井翔氏，合同研究プロジェクトの皆様に感謝致します．最後に，今まであらゆる面で多大な助力をくれた家族と友人に心から深謝と敬愛を表し，謝辞と致します．

以上を持って，謝辞と致します．

参考文献

[1] Inc Cisco Systems, 2011. http://www.cisco.com/.

[2] YouTube, LLC. YouTube WWW page, 2011. http://www.youtube.com/.

[3] niwango, inc. ニコニコ動画 WWW page, 2011. http://www.nicovideo.jp/.

[4] Hulu. Hulu WWW page, 2011. http://www.hulu.com/.

[5] Ustream, inc. Ustream WWW page. 2011. http://www.ustream.tv/.

[6] niwango, inc. ニコニコ生放送 WWW page. 2011. http://live.nicovideo.jp/.

[7] IPTV FORUM JAPAN. IPTV FORUM JAPAN WWW page. 2011. http://www.iptvforum.jp/.

[8] NTT Plala Inc. ひかり TV WWW page. 2011. http://www.hikaritv.net/.

[9] BB Cable Corporation. ブロードバンドケーブル TV BBTV. *http://www.bbtv.com/*, 2011.

[10] M. Handley, C. Perkins and E. Whelan. *Session Announcement Protocol*, October 2000. RFC 2974.

[11] B. Fenner, M. Handley, H. Holbrook, and I. Kouvelas. *Protocol Independent Multicast - Sparse Mode (PIM-SM):Protocol Specification (Revised)*, August 2006. RFC 4601.

[12] IANA. IANA. 2011. http://www.iana.org/.

[13] H. Asaeda, T. Jinmei, W. Fenner, and S. Casner. Mtrace Version 2: Traceroute Facility for IP Multicast. *http://tools.ietf.org/html/draft-ietf-mboned-mtrace-v2-08*, January 2011. Internet-Draft Standards Track.

[14] D. McWalter, D. Thaler and A. Kessler. *IP Multicast MIB*, December 2007. RFC 5132.

[15] K. McCloghrie, D. Farinacci and D. Thaler. *IPv4 Multicast Routing MIB*, October 2000. RFC 2932.

[16] K. McCloghrie, D. Farinacci, D. Thaler and B. Fenner. *Protocol Independent Multicast MIB for IPv4*, October 2000. RFC 2934.