情報処理学会論文誌　Vol. 52　No. 3　1280–1290 (Mar. 2011)

多様な仮想網が構築されるMPLS障害に適した分散協調障害処理技術

渡辺　修[†1,†2]　小柳恵一[†1]

近年，MPLSのコア網上に様々なレイヤ2仮想網サービス（Ethernet，Frame Relay，ATM）を構築し，ネットワークプロバイダが保有する自社網の統合や，仮想閉域網サービスとしてのエンドユーザへの提供が行われている．1つのコア網上に様々なサービスを提供する仮想網を構築することでコスト的な効果が大きい．しかし，1つの物理的障害が複数個所の仮想網の障害へと派生し，障害の影響範囲が大きくなるため，運用においては一連の障害の原因を早期に突き止め対応することが重要となる．本論文では，MPLS上で複数のレイヤ2仮想網サービスが提供されている環境において，通信サービスに関係し，かつ各ルータで分散管理されたエンティティの依存関係から障害の原因となったイベントを抽出する新たな方式を提案する．従来，障害の依存関係を解決するためには，中央の管理システム内にあらかじめ設計され，静的に保持された依存関係ルールを用いていた．この従来の方式では，ネットワークが拡大するにつれ，ネットワークの実態と中央管理システムで管理された静的情報の不一致などの問題が発生していた．このような問題を解決するために，本提案では，グローバルに展開された各ルータの持つローカルな知識を用いて依存関係ルールを動的かつリアルタイムに生成することで，仮想網に影響を及ぼす物理的障害の部位特定が可能となることを示す．

Distributed and Co-operated Fault Processing for Various Virtual Networks over MPLS

Osamu Watanabe[†1,†2] and Keiichi Koyanagi[†1]

Recently, service providers build various layer 2 virtual networks (Ethernet, Frame Relay and ATM) over MPLS core network. This MPLS convergence significantly reduces cost of ownership, although at the same time, this convergence increases operational complexity. Because one physical failure induces various failures at virtual networks. Operators are required to find the root cause event as soon as possible. This paper proposes the methodology of root cause event analysis based on communication service entity relationship of MPLS layer 2 virtual network services. Our methodology does not have to store entire dependency rule in NOC NMS, instead, dynamically generates entity dependency information using router's own local information. Our methodology enables physical fault isolation by means of dynamically generated entity relationship from router's local information.

1. はじめに

仮想化技術の進展にともない，物理的な網上に仮想的な網を構築するケースも増えてきた．これら仮想網の構築では，物理的な回線工事が不要，プロビジョニング時間の短縮化，仮想網をプライベート網として機密性を保つ，物理的な資源の有効利用など，多々の有利な点がある．しかし，他方で1カ所の物理的な障害が複数の仮想網に影響を及ぼし，速やかに物理的障害箇所を特定して復旧するなど，より効率的なオペレーションが求められる．

特にサービスプロバイダにおいては，MPLS（Multi Protocol Label Switching）[1]技術を利用したレイヤ2仮想網により従来のEthernet，Frame Relay，ATMといった個別に構築した網を統合するほか，レイヤ2仮想閉域網（L2VPN）サービスとしてエンドユーザに提供することが行われている[2]．これらの複数のレイヤ2サービスは単一のMPLS網上で提供することが可能であるため，ネットワーク基盤を有効に活用することができる[3]．反面，下位レイヤの障害が複数の仮想網の障害として波及していくことになり，障害発生時には速やかに原因を突き止めて対処することが重要になる．従来，障害の派生関係は中央のネットワーク管理装置側にルールベース，モデルベースで実装され，障害箇所特定に使用されることが多かった．この場合，ネットワーク管理装置で保持しているルール，モデルが日々運用され構成変更が頻繁に発生する実際のネットワークの構成と一致しなくなるという現実の問題がある．

そこで，本論文では，各ルータ内のエンティティの依存関係の情報を，障害発生時にイベントログに付与することで，障害発生時点の依存関係を反映させた全体のイベントログの依存関係を構築・解析し，障害発生原因に最も近いイベントログを特定する方式を提案する．具体的には次の手順による．(1) ルータ内で発生するイベントログを自ルータ内で捕捉し，MIB（Management Information Base）などのエンティティ依存関係の情報を検索し，依

†1 早稲田大学大学院情報生産システム研究科
Graduate School of Information, Production and Systems, Waseda University
†2 シスコシステムズ合同会社
Cisco Systems G.K.

存関係情報を付与したイベントログを管理システムに対して送出する．(2) 管理システムでは，付加されたエンティティ依存情報から，イベントログの依存関係（これをイベントグラフと呼ぶ）を導出する．(3) 前のステップで得られたイベントグラフを解析し，障害原因に最も近いイベントログを特定する．

以上のような今回提案する手法は，既存のルールベース，モデルベースの手法に比較して次の点で有用である．(1) ネットワークの構成情報を中央のネットワーク管理装置側で維持する必要がなく，ネットワーク構成変更に柔軟に対応することができる．(2) 評価において示すように，十分に実用的なオンライン処理能力が実現可能である．

以下，2 章では仮想網における障害分析の困難さに関する問題分析と，障害原因を特定する関連研究について述べ，3 章では本論文で提案する分散した構成情報から障害原因を特定する手法を説明する．4 章で，本提案の実装と評価について述べる．

2. 問題分析と関連研究

2.1 問 題 分 析

図 1 は，単一の網（物理網と呼ぶ）の上に複数の仮想網が重畳されて運用されている例を示す．仮想網のリンク（仮想リンク）やノード（仮想ノード）は，他の仮想網のリンク，ノードとは独立しており，相互の仮想網からは隠ぺいされる．これら仮想リンク・ノードの機能は物理網で提供されるプロトコルとサービスとして提供される．したがって，物理網で発生する障害は，重畳している複数の仮想網で提供しているサービスに伝播していくことになり，障害の影響度は従来の物理網で提供していた単一サービスに比較して大きなものとなる．

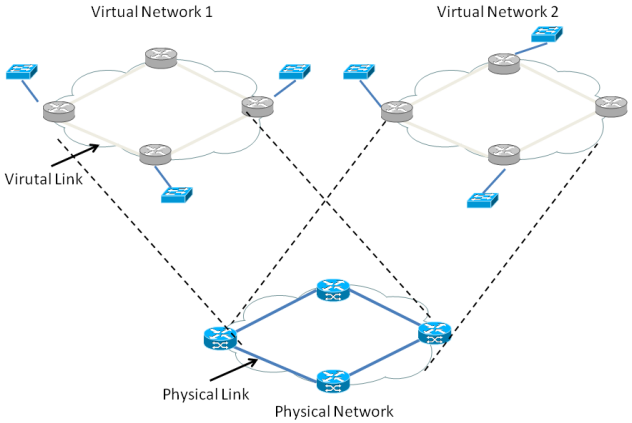

図 1　物理網に重畳した仮想網サービス
Fig. 1　Virtual network services over physical network.

図 2 は，図 1 の仮想網サービスの物理網，仮想網のエンティティ相互の関連を，文献 4) をベースにエンティティ依存関係としてモデル化したものである．仮想網のサービスを提供するエンティティは，物理網の階層化されたエンティティが提供するサービスと考えることができる．ここで依存関係とは，次の 2 種類のエンティティ間の障害の伝播の関係である．

- 上位レイヤの通信エンティティは下位レイヤのエンティティに依存する．
- 異なるノードのエンティティは通信している同位レイヤのエンティティと相互に依存関係にある．

Entity **x** ⟶ *Entity* **y** の矢印はエンティティ y の機能はエンティティ x に依存していることを示し，エンティティ x からエンティティ y へ障害発生のイベントが派生する関係を示す．図 2 のサービスエンティティとは，物理網の複数ノードの協調動作によって提供される仮想網サービスを示す．このように仮想網の環境では，物理網の一か所の障害が，物理網の

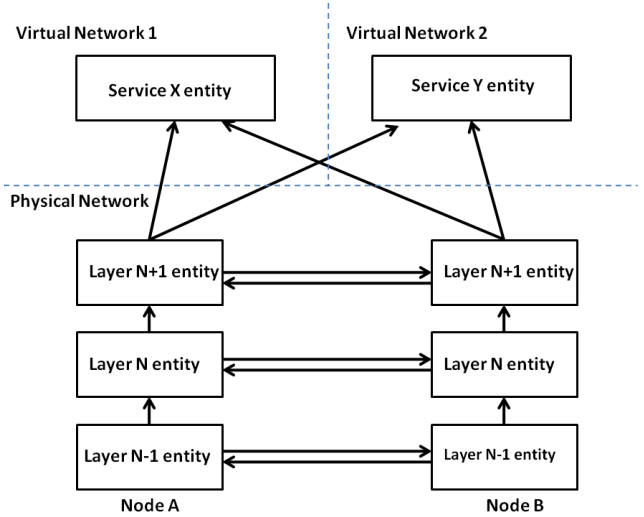

図 2　階層化エンティティの依存関係
Fig. 2　Layered entity and its dependency model.

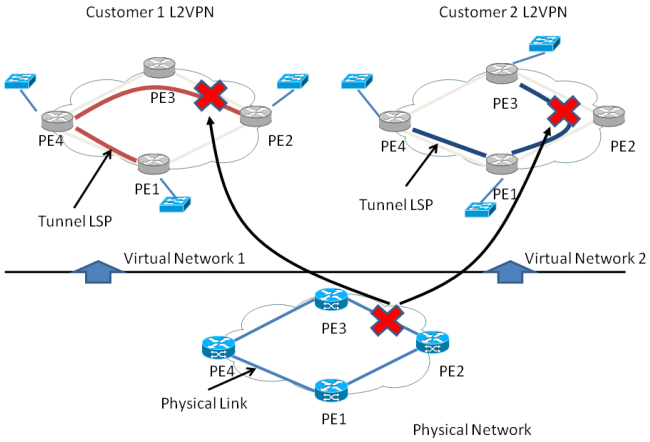

図 3　MPLS による仮想網サービス
Fig. 3　Virtual network services over MPLS.

上位層へ伝播し，さらに仮想網へと伝播して広がっていくことが特徴である．管理システムへは地理的に独立した各ノードから個別に障害イベントとして報告されてくるが，障害箇所を特定するためには，これら多数のイベントから伝播の元となった原因のイベントを抽出する必要がある．従来は障害の派生関係を，中央のネットワーク管理装置側にルールベースまたはモデルベースで実装し，障害箇所特定に使用されることが多かった．この場合，管理装置で持つルールまたはモデルは，ある特定の時期の網構成を示した静的なものであるため，日々運用されている実際のネットワークの構成とは一致しなくなるという問題がある．

2.2　MPLS における仮想網

図 3 は単一の MPLS 網上で提供されている複数のレイヤ 2 サービスを示している．レイヤ 2 のフレームは MPLS のラベル付きパケットでカプセル化されて MPLS 網上を転送される．カプセル化されたパケットは MPLS LSP（Label Switch Path）上を転送されていく．MPLS LSP はレイヤ 2 サービスで必要とされる帯域確保，FRR（Fast ReRoute）[5] の必要性から RSVP TE（Traffic Engineering）[6] のトンネルを 2 地点間で張り，そこにレイヤ 2 のフレームを MPLS のラベルでカプセル化したパケットを通過させることでサービスを提供する．

図 3 では，1 つの物理的な MPLS 網の上に複数の L2VPN サービスを提供している．PE（Provider Edge）ルータ間で RSVP TE による Tunnel LSP を張り，L2VPN の拠点間にサービスを提供している．ここで PE2–PE3 間の物理リンクに障害があった場合，この障害により Customer1 VPN で使用している PE2–PE4 間のトンネル，Customer2 VPN1 で使用している PE1–PE3 のトンネルに障害が派生し，Customer1，Customer2 の両方に障害の影響が及ぶこととなる．大規模ネットワークではトンネルの数は数千以上に及び，大量のイベント情報から原因となった障害と派生した障害を区別することは容易ではない．

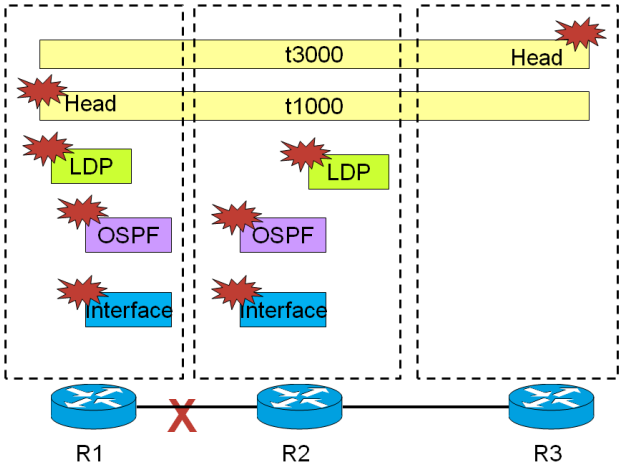

図 4　MPLS TE でのリンク断障害
Fig. 4　Alarms caused by link down at MPLS TE network.

2.3　MPLS TE

本提案の実装例として**図 4** のような MPLS TE（Traffic Engineering）サービスを提供する簡単なネットワークを考える．TE サービスを提供するためには，OSPF（Open Shortest Path Fast）のようなルーティングプロトコルおよび LDP（Label Distribution Protocol）の機能が動作していることが必要になる．図 4 は 3 つのルータからなり，R1 から R2 を経由して R3 まで，TE のためのトンネル（t1000，t3000）を 2 本張った簡単なネットワークである．TE トンネルは 1 方向なので双方向に計 2 本必要である．このときルータ R1-R2 間の物理リンクの断があったとする．この障害から以下のイベントログが各ルータ，各レイヤのエンティティから派生的に発生する（**図 5** を参照）．

- 物理リンクのインタフェースリンク断とリンクレイヤプロトコル断（3 カ所）図 5 の 003，004，008

```
001 R1: *Dec 28 15:14:14.023: %OSPF-5-ADJCHG: Process 100, Nbr 1.1.1.104 on Serial2/1 from FU...
002 R1: *Dec 28 15:14:14.024: %LDP-5-NBRCHG: LDP Neighbor 1.1.1.104:0 (3) is DOWN (Interface ...
003 R1: *Dec 28 15:14:16.014: %LINK-5-CHANGED: Interface Serial2/1, changed state to administ...
004 R1: *Dec 28 15:14:17.014: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/1, chan...
005 R2: *Dec 28 15:14:23.430: %LDP-5-NBRCHG: LDP Neighbor 1.1.1.100:0 (1) is DOWN (TCP connec...
006 R1: *Dec 28 15:14:32.269: %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel1000, cha...
007 R3: *Dec 28 15:14:37.249: %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel3000, cha...
008 R2: *Dec 28 15:14:43.313: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, chan...
009 R2: *Dec 28 15:14:43.314: %OSPF-5-ADJCHG: Process 100, Nbr 1.1.1.100 on Serial2/0 from FU...
```

図 5　リングダウンによって発生したイベントログ
Fig. 5　Event logs caused by link down at MPLS TE network.

- OSPF のネイバーロス（2 カ所）図 5 の 001，009
- LDP のネイバーロス（2 カ所）図 5 の 002，005
- MPLS TE トンネルの head のルータよりトンネルダウン（2 カ所）図 5 の 006，007

これらのイベントは 1 つの原因から連鎖的に派生し，送出元ルータも異なる．また，送信タイミングもルータの負荷などの状況により，原因となったログが先に届く保障もない．実際のネットワークでは MPLS TE トンネルの数が多く，原因となったリンク断にともなうルータからのリンク断メッセージは大量のログに埋没してしまい，リンク断およびその個所を原因として特定することは容易ではない．

2.4 関 連 研 究

障害箇所を特定する問題に対してはいくつかのアプローチがある．1 つはルールを蓄積したエキスパートシステムによる障害解析である（文献 7)）．単純なルールの蓄積では対処できない場合も多く，ネットワークの構成とルールの同期も困難な場合もあり広く使われるには至っていない．通信システム中のエンティティとその関連を形式的に記述して，警報を発出したエンティティから関連する警報を特定し障害箇所を特定する手法も提案されている[8]．しかし，すでに述べたようにサービスの追加・変更が頻繁に行われる網では，網の実態と管理システム内のエンティティおよびその関連情報の整合性をとることが困難である．また，障害伝播モデル（FPM: Fault Propagation Model）を因果関係（Causal Graph）または依存関係（Dependency Graph）[9]，ベイジアンネットワーク[10] でモデル化し，原因を特定するシステムがある．FPM ベースのシステムでは，障害箇所特定の精度向上のためには，障害伝播の正確な知識を事前に保持しておくことが必要となる．しかし，前記と同様の理由で網の実際と障害伝播の知識の整合をとることが困難である．本論文では，エンティティの障害派生の依存関係を分析することで原因となったイベントを特定する手法を提案する．依存関係をルータ内の知識で補い，中央のネットワーク管理装置には事前の定義を不要とすることで，事前知識を管理システムに蓄える必要がなく，管理システムの運用負荷を減らし，つねに実網の状態を反映したエンティティの依存関係が維持できる．

3. 提 案 方 式

本章では，本論文で提案する方式について概要を述べ，次に MPLS TE により提供される仮想網を例に各手順について詳細を説明する．本論文で提案する手法は，仮想網一般に適用可能である．本論文で提案する方式の特徴は，エンティティのルータ内に閉じた部分的な依存関係の情報を障害発生時にルータから管理システムに報告し，管理システムで全体的な依存関係に組み立て，障害原因箇所を特定することにある．仮想網の例としては MPLS で提供される L2VPN サービスを対象とした．これはエンティティの依存関係に関する情報が MPLS に関連する MIB で定義済みであり一般的に実装可能であるからである．

3.1 概　　要

本論文で提案する方式では，1 つの原因から次々に派生するイベントログの依存関係を構築し，障害発生原因に近いログを運用者に提示することで運用負荷の低減を目的とする．本提案で使用する用語を定義する．

イベント　ネットワークで起きた障害である．リンクダウンやプロトコル障害などである．

イベントログ　ルータが検出したイベントを管理システムに通知するログを示す．イベントの情報が簡潔に記述されている．通知手段は syslog[11] が使用される．

エンティティ　ネットワークの各層におけるプロトコルの処理単位を示す．リンク層であればインタフェースごとに別エンティティであり，上位層であればプロトコル単位である．サービス層であればネットワークにまたがって 1 つのサービスが 1 つのエンティティと考える．

依存関係　2 種類の依存関係が考えられる．同一装置内において，上位層のエンティティが下位層のエンティティの機能を利用する関係と，別装置において同じ層のエンティティどうしが同一プロトコルで互いに通信する関係である．あるエンティティで発生したイベントは，依存関係にあるエンティティへ派生してイベントを発生させる．

イベントグラフ　あるイベントが発生すると，そのイベントが発生したエンティティと依存

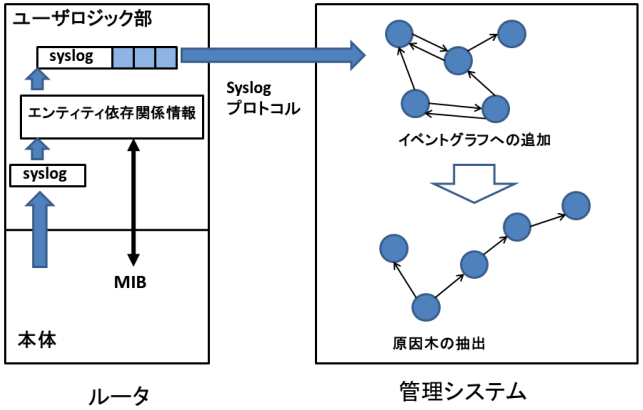

図 6　システム構成
Fig. 6　System overview.

関係にあるエンティティからイベントが派生的に発生する．この依存関係にあるエンティティから発生した一連のイベントログの依存関係を示す有向グラフをいう．

原因木　イベントグラフのある節点を根とした有向木をいい，イベントグラフの部分グラフである．イベントグラフ内の特定のイベントから派生する一連のイベントを示す有向木である．

本提案の方式は以下の手順により処理を進める．

ステップ 1　ルータのユーザロジック追加機能を使い，ルータで発生したイベントログを捕捉し，エンティティ間の依存関係を示す情報をローカルに検索・付加して syslog を管理システムに送信する．

ステップ 2　管理システムでは付加情報を使用して受信したイベントログの依存関係（イベントグラフ）を構成する．

ステップ 3　ステップ 2 で導出した有向グラフは，障害発生の原因となったイベントログと，そこから派生したイベントログを含んでいる．そこで，この有向グラフの部分グラフである有向木のうち，イベントグラフの節点を最も多く含む有向木（原因木）とその根となるイベントを導出することで，一連のイベントログの原因ログを運用者に提示する．1 つの原因木で，イベントグラフのすべての節点を持つことができない場合，原因となるイベントが複数存在しており，複数の原因木が存在する．

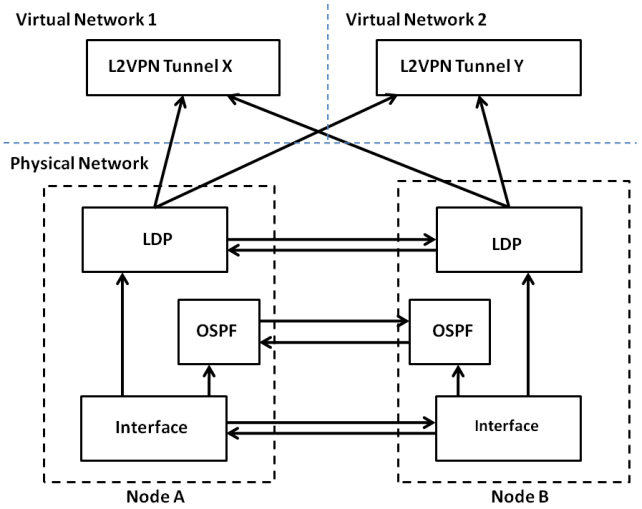

図 7　L2 VPN サービスでのエンティティの依存関係
Fig. 7　L2 VPN service entity dependency relationship.

従来の中央における静的な情報管理方法と比較すると，ステップ 1 のログを捕捉しエンティティ間の関連情報を検索して付加するステップが本提案手法の従来手法に対しての追加コストとなる．実測によれば，このステップによるログの送付遅延は 150 ミリ秒程度の増加であった．

3.2　エンティティのモデル化と依存関係

図 7 は，図 2 の一般化したモデルを MPLS TE のトンネルによって提供される仮想網サービスに適用したものである．仮想網のサービスを MPLS TE トンネルのエンティティと考え，MPLS TE サービスは，物理網の LDP プロトコルによって提供される．LDP のサービスは物理リンク層（Interface）からサービスの提供を受け，対向となるノードの等位の LDP と相互作用する，という関係にある．

3.3　イベントログへの情報付加

本論文で対象としている MPLS TE サービスに関連したエンティティの依存関係は，ルータ内でローカルに保持している以下の情報により取得が可能である．

(1)　リンクレイヤの隣接情報

CDP（Cisco Discovery Protocol）[12] もしくは IEEE802.1AB LLDP（Link Layer

Discovery Protocol)[13] または各ベンダの独自実装が存在しリンクレベルの隣接情報を取得することができる．リンクレイヤの隣接情報によりリンク層のエンティティのイベントの派生関連付けが可能となる．

(2) LDP（Label Distribution Protocol）の隣接情報
LDP MIB[14] により LDP エンティティの隣接情報を取得できる．当ルータの LDP エンティティで障害が発生した場合，隣接している LDP エンティティでも同様の障害が発生する可能性が高い．そこで LDP 隣接情報を参照してイベントに付加することで LDP 層のエンティティ間の派生関連付けが可能となる．

(3) OSPF（Open Shortest Path Fast）の隣接情報
OSPF MIB[15] により OSPF エンティティの隣接情報を取得できる．LDP と同様に派生の関連付けを行うことができる．

(4) TE トンネルの通過情報
MPLS TE MIB[16] により自ルータを通過している TE トンネルの情報を取得できる．自ルータでの LDP 障害により，自ルータを通過する TE トンネルに障害が派生する．LDP のアラーム報告時に，自ルータを通過するトンネルの情報を付加することで，TE トンネルダウンと下位の LDP アラームとを関連付けることができる．

ルータで検出されたイベントは SNMP トラップまたは syslog によって管理システムに送られる．本提案の方式では，イベントログに依存関連の情報を付加して，管理システム側での依存関係分析に使用するものである．したがってメッセージのフォーマットが自由な syslog を使用する．たとえばルータ R1 のインタフェースダウンにともなう syslog

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/1, changed state to down

をルータで検知した場合，自ルータ R1 の Serial2/1 のインタフェースの対向は，CDP または LLDP によりルータ R2 Serial2/0 のインタフェースであることから，R1 は次の情報を syslog メッセージに付加して管理システムに送付する．

$[Instance\ R1.Serial2/1][Upper\ R1.LDP\ R1.OSPF][Peer\ R2.Serial2/0]$

$[Instance]$ は自エンティティの識別名，$[Upper]$ は自エンティティに依存している自ルータ内の上位レイヤエンティティ，$[Peer]$ は同位レイヤで対向となっている別ルータのエンティティを示す．エンティティの記法は（ルータ名）.（インスタンス名）としている．物理インタフェースのように複数のインスタンスが存在する場合はルータ内でインスタンスを識別する．たとえば Serial2/0 はルータの 2 番目のモジュールの最初（0 番目）のシリアル回線を示す．また，ルータをまたがるサービスの場合エンティティ名にはルータ名を付与しない．

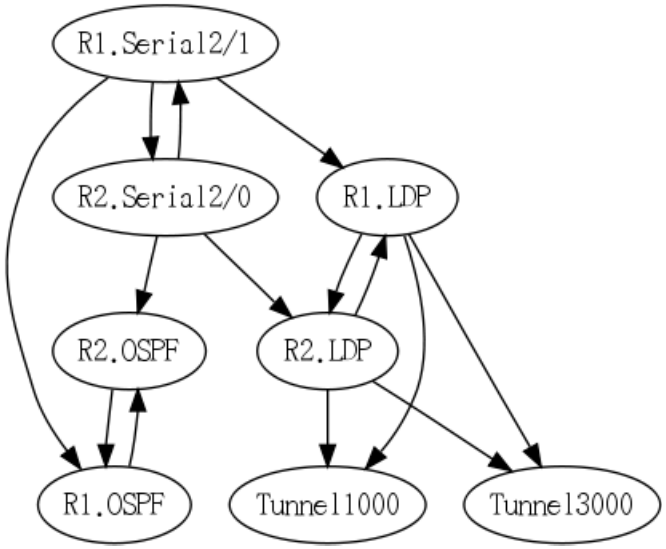

図 8　イベントの依存関係グラフ
Fig. 8　Event dependency graph.

3.4　イベントログ依存関係グラフ

各ルータから管理システムに送信されたイベントログを前項の $[Upper][Peer]$ でタグ付けされた情報を元に依存関係をグラフ化する．依存関係は新規イベントログの $[Insntace]$ で示す節点から，イベントグラフ中の $[Upper]$ および $[Peer]$ で指している既存の節点に対して有向辺を引き，イベントグラフ中のそれぞれの節点の $[Upper]$ および $[Peer]$ で示しているインスタンスが新規イベントログのインスタンスを指していれば，既存の節点から新規の節点に対して有向辺を引く．図 4 の構成で発生したリンク断によって，図 5 で示す一連のイベントログが発生し，ルータ内で付加した依存関係を元に**図 8** のようなイベントの依存関係グラフができる．

矢印は，イベントを発出したエンティティと，そのイベントから派生しうるイベントとの関連を示している．ピアとなるエンティティは相互に関連するため両方向である．レイヤとなっているエンティティは下位から上位に対しての 1 方向となる．この有向グラフを $G = (V, E)$ とする．新イベントログが到着した場合には，すべての節点に対して 1 回走査すれば，グラフ G を更新することができ，イベントグラフの更新のための計算量は $O(|V|)$

である．$|V|$ はグラフ G の節点数である．新規に $|V|$ 個のイベントからイベントグラフを構成するための計算量は，すべての節点の走査を $|V|$ 回繰り返すから $O(|V|^2)$ である．

3.5　原因イベントの導出

最後に，図 8 より，原因となるイベントを特定する．イベントグラフの節点を根として，その節点から訪問可能な節点からなる有向木（原因木）を探索する．最も多くの節点を訪問する有向木を探せば，その根となる節点がイベントの依存関係の原因となったイベントであると判断できる．いくつかのケースがある

1 つの障害で複数のイベントが発生する場合　たとえば 2 地点を結ぶリンクのダウンの場合，リンクの両端のルータのインタフェースダウンのイベントログが発生し，どちらも一連のイベントログの根本原因である．図 8 のイベントグラフからは**図 10** のように R2.Serial2/0 を根とする原因木，図には示していないが同様に R1.Serial2/1 を根とする原因木が導出できる．どちらの節点も原因に一番近いイベントを指しており，運用的にも片方のリンクダウンを示せば十分と考える．

複数の原因が存在する場合　複数箇所で独立にリンクダウン障害が発生した場合，1 つの節点を根とする原因木ではイベントグラフ全体をカバーできない．この場合，イベントグラフを最大にカバーする原因木を探し，次にカバーされていない節点からなる残余のイベントグラフをカバーする原因木を探し，イベントグラフ全体の節点をカバーするまで繰り返す．

ルータダウンの場合　ルータダウンの場合，ダウンしたルータからはイベントログが発出されず，隣接するルータとダウンしたルータの間のリンクダウンが原因イベントとなる．複数の原因が存在する場合と同様に，リンクダウンを原因とする複数の原因木を導出する．

図 9 にイベントグラフから原因木を導出する手順を示す．ある節点を根として，イベントグラフ $G = (V, E)$ を幅優先探索して有向木を探索する．有向木を導出するための計算量は $O(|V| + |E|)$ である．依存関係は，イベントグラフ上の距離が近い節点の方が強いと考えられるから，幅優先探索により根となる節点に距離的に近い節点を優先して探索する．G のすべての $v \in V$ を根として有向木を導出し，一番多く訪問した有向木の根を原因イベントとし，このときの有向木を原因木とする．原因木を導出するために必要な計算量は，幅優先探索を節点数だけ繰り返すから $O(|V|^2 + |V||E|)$ である．$|E|$ は節点数 $|V|$ にほぼ比例する（あるイベントと依存関係を持つイベントの数は大きく変化はしない）と考えることができるので，原因木導出の計算量は $O(|V|^2)$ である．

```
until(|V| == 0){
  foreach(v ∈ V){
    v を根として G に対して幅優先探索を行い有向木 G' = (V', E') を導出
    G に対しての節点訪問率 |V'|/|V| を計算
  }
  最大の節点訪問率を持つ G' を原因木として出力
  G から G' = (V', E') の節点 v ∈ V' と v に入出力する有向辺 e ∈ E を除く
}
```

図 9　原因木の導出手順
Fig. 9　Procedure to derive rooted tree.

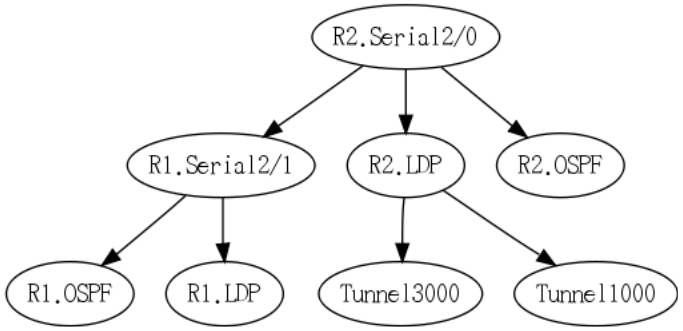

図 10　イベントグラフに対する原因木
Fig. 10　Maximum rooted tree for event graph.

4. 実装と評価

本章では，まず提案方式の実装について述べ，次に処理能力の評価を行い，結果について検討を行う．

4.1　諸元および実装

ステップ 1 のルータ内処理では，ルータには Cisco7301 を使用し EEM（Embedded Event Manager）[17] というユーザロジックの組み込み機能を使って syslog にエンティティの依存関係の情報を追加した．言語は，TCL スクリプト言語を使用し 500 行程度の規模である．

ステップ 2，3 の管理システムとしては Intel Core i5（2.6 GHz）の Ubuntsu 8 の Linux を使用している．管理システムのイベントログ構築部分と原因木導出部分は，Perl を使用し 500 行程度である．処理能力の評価では，大規模ネットワークをシュミレーションする必要があるため次のような手順で MPLS 上に TE トンネルを持つネットワークを構成した．

（1） ルータ数 100 とし，それぞれのルータがリンクを平均 5（平均次数 5）持つランダムグラフを生成（ベースとなる MPLS 網）．
（2） 100 のルータのうち 2 つのルータをランダムに選択してトンネルを張ることを繰り返し，10,000 のトンネルを張る（MPLS 網上の仮想網で使用するトンネル）．
（3） 任意のリンクを選択して物理リンク断とする．
（4） 物理リンク断の結果発生するはずのリンクレイヤのダウン，OSPF/LDP のネイバーロスに相当するログを生成．
（5） ダウンしたリンクを経由しているトンネルを列挙し，トンネルの先頭ルータからトンネルダウンのログを生成．

リンクダウンは 1 カ所から 10 カ所まで変化させ，手順ごとに網は生成し直している．リンクダウンの箇所もランダムに選択しているため生成するログの数にもばらつきが出ている．生成したログに対して本アルゴリズムを走らせてイベントグラフの生成，イベントグラフからの原因木の抽出の評価を行った．リンクダウン数と生成されたイベントログ数の関係を**図 11** に示す．各リンクダウン数につき 3 回試行し記録した．リンクダウンが 1 カ所だけの場合でも結果的に 80 個から 90 個のイベントログが生成される．1 カ所のリンクダウンではリンクレイヤ，LDP，OSPF に関するログが 8 個であり，これらのログ以外はすべて仮想網に影響を与えるトンネルダウンのログであり，リンクダウンを起こしたルータからは遠く離れたトンネルの入り口にあたるルータから送出される，1 カ所のリンクダウンによって，多くの仮想網の障害に波及することになる．

4.2　イベントグラフ生成の評価

図 12 は，イベントログのエントリ数とイベントグラフの生成に要した時間の関係を示している．syslog サーバにいったん蓄えたログを取り出してイベントグラフを生成する場合に相当する（バッチ処理）．横軸はイベント数，縦軸はイベントグラフを生成するために要した時間（秒）である．総イベント数を N とすると，1 つのイベントをイベントグラフに追加するためにすべての節点を走査する必要があり，N のイベント数を処理するためには $O(N^2)$ の計算量となる．

別の方法として，イベントの到着ごとに，逐次的にイベントグラフの構成を変更していく

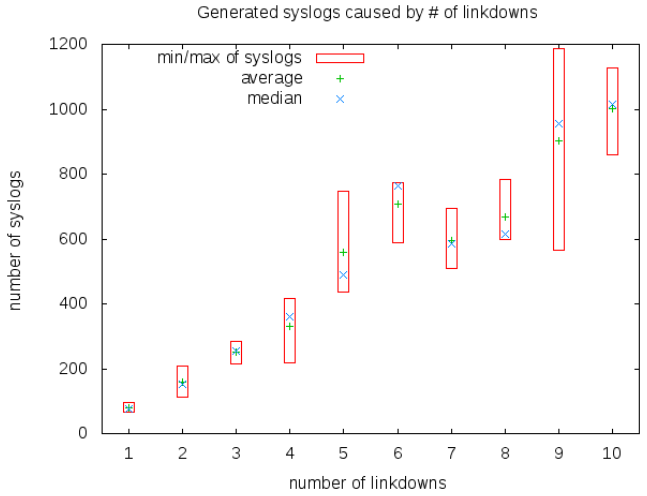

図 11　リンクダウン数と生成されたイベントログ数
Fig. 11　Number of link down events and generated event log entries.

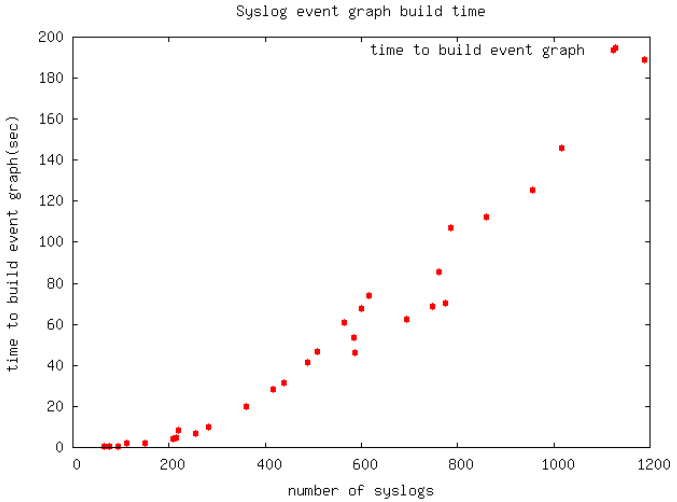

図 12　イベントログ数とイベントグラフ生成時間（バッチ処理）
Fig. 12　Syslog event graph build time (batch style).

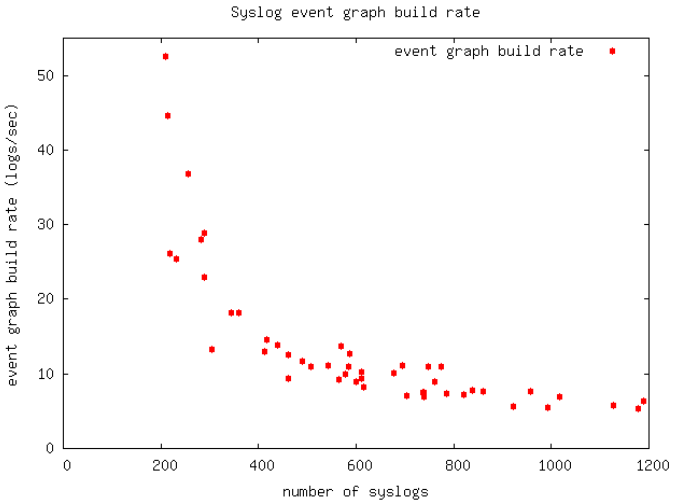

図 13　イベントログ数とイベントグラフ生成速度（オンライン処理）
Fig. 13　Syslog event graph build rate (online style).

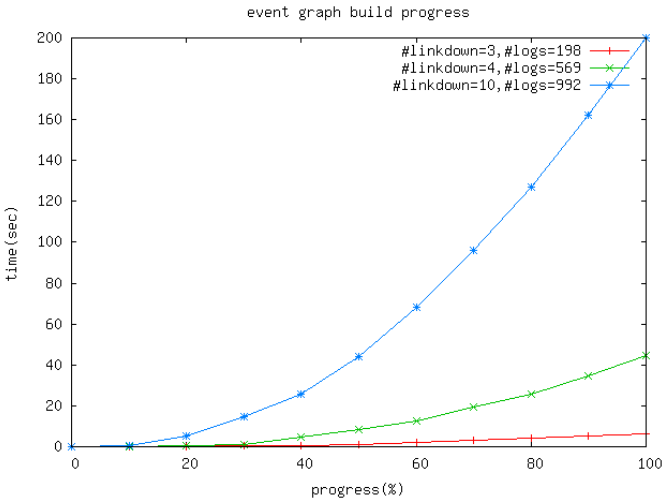

図 14　イベントグラフ生成の処理時間の推移
Fig. 14　Syslog event graph build progress.

ことが考えられる．**図 13** は，イベントグラフ生成時の 1 秒あたりのイベント処理速度とイベント数の関係を示している（見やすさのためイベント数が少ない場合を省き，追加試験を行いサンプル数を増やしている）．バッチ処理の場合は蓄積したイベントログからイベントグラフを生成する場合を想定しているが，この図ではイベントログが到着するたびにイベントグラフを更新するケースを想定している（オンライン処理）．結果的に生成されるイベントグラフは同一である．既存のイベントグラフに新規のイベントの依存関係を追加するための計算量は，$O(N)$ であり，イベントログ数とイベントグラフのイベント処理速度はほぼ反比例する．大規模障害を想定した既存イベントログ数が 1,000 程度の場合では，イベントグラフ更新は毎秒 5 イベント程度であった．

図 14 は，3 種類のログ数の場合においてイベントグラフ生成の処理時間の推移を示した．横軸が全体のログ数に対する進捗を示し，縦軸が経過時間を示す．既述のように新規のイベントログをイベントグラフに追加するためには，既存のイベントグラフの節点をすべて走査する必要があり，既存のイベントグラフの節点数に比例して処理時間が必要であることを示している．

4.3　原因木導出の評価

バッチ処理またはオンライン処理で生成したイベントグラフから原因木を導出する評価を行った．バッチ処理の場合は蓄えたイベントログからイベントグラフを生成した後，オンライン処理の場合は定期的に原因木導出のアルゴリズムを走らせて，その時点の原因イベントを導出することになる．**図 15** は，生成したイベントグラフから原因木を抽出するに要した時間とイベントログのエントリ数との関係を示している．すべての場合において，リンクダウンにともなって直接発生する物理リンクのリンクダウンログを原因イベントとして取り出すことができた．計算量は先に示したとおり $O(N^2)$ である．

実際の運用においては大規模障害の場合であっても，障害部位特定に割くことのできる時間の目安は 10 分以内程度である[18]．100 ノード，10,000 の仮想リンク（TE トンネル）で発生した 1,200 個のイベントログが発生する大規模障害を想定したケースにおいても，イベントグラフ生成に 3 分，原因木を導出して障害箇所を特定するまでに 30 秒程度である．これは実用上目安となる時間内に処理を終えており，実用的に問題のない性能である．

最後にルータから管理システムまでの経路に問題があった場合など，一部のログが欠損した場合について評価を行った．障害原因の誤検出については 2 種類存在する．障害原因

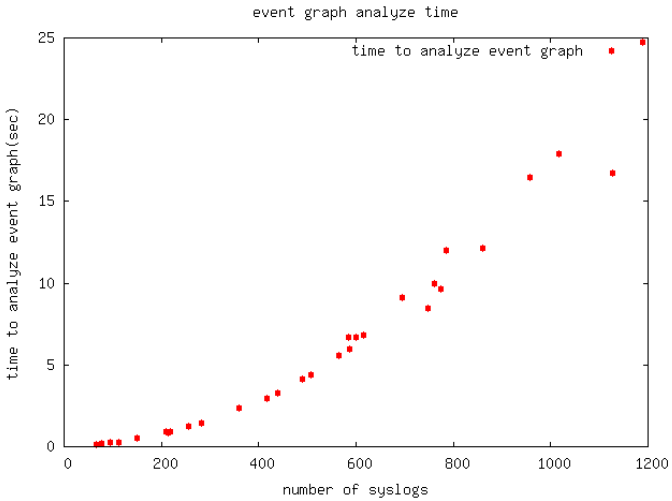

図 15　イベントログ数とイベントグラフ解析時間
Fig. 15　Syslog event graph analyze time.

表 1　ログの欠損があった場合の誤検出数
Table 1　False positve and false negative events.

ログ欠損率	0%	10%	20%	30%	40%	50%
False Negative 件数	0	0	1	2	3	2
False Positive 件数	0	26	50	89	101	163

であるにもかかわらず検知しない場合（false negative）と，障害原因ではないにもかかわらず検知する場合（false positive）がある．リンク障害を 10 カ所で発生させたシュミレーション結果のログ（欠損がない場合で 992 個のイベントログが発生する）を使用して 0%から 50%までランダムにログを間引いてログの欠損率を変化させ，障害原因箇所が正しく特定されるか試験を行った．**表 1** に結果を示す．ログの欠損率が高い場合でも検出すべきログを誤って検出しない（false negative）件数は少ないものの，検出すべきでないログを誤って検出する（false positive）件数は非常に多くなる．これは，エンティティの依存関係をイベントに付与される関連付け情報のみで，イベントグラフを構成しようとする本方式に内在する問題である．あるべきログが欠損すると依存関係がそこで分断されてしまい，イベントグラフから原因木を導出する際に，本来の依存関係を抽出できず小さな原因木が無数にでき，これが false positive に対応することになる．このような事態を避けるためにはいくつかの方策が考えられる．(1) ユーザにサービスを提供する網と管理用の網を分けるか，管理メッセージには高いプライオリティを割り当てバッファオーバフロー時などでもメッセージが廃棄されない管理網を設計する．(2) 原因木の大きさを評価対象とする．分断したイベントグラフから抽出した原因木は，それが false positive である場合には，節点の数が非常に少ない木となる（多くの場合，根が 1 つだけの木である）．より「確からしい」指標とともに運用者に提示することで，ログ欠損の影響を小さくできる．原因木の大きさをどのように確からしさの指標に変換するかは今後の課題である．

5. むすび

ネットワークの仮想化技術の進展にともない，網の統合が進んでいる．設備コストの低減に大いに寄与する反面，運用の困難さは増加する．本論文では，物理網上に重畳して構築される仮想網において，ルータの知識を利用してイベントの派生関係を解決し，原因イベントを特定する提案を行った．従来研究では管理システム側での方式に着目するものがほとんどであったが，ルータ内での MIB に蓄積された情報，ルータへの管理機能の組み込みを利用して，ルータ側と管理システム側での協調分散動作を行う仕組みを提案した．本論文では，提案の実装例として MPLS 上の仮想網のリンクレイヤである TE トンネルの障害管理に主眼を置いたが，実際には ATM，FR，Ethernet など L2VPN 上のサービス種別まで拡張する必要があり，今後の課題である．また，さらに大規模な仮想網の場合に問題となりうるイベントグラフ生成，原因木導出のマルチタスク化など実装における性能面の向上も今後の課題である．

参考文献

1) Rosen, E., Viswanathan, A. and Callon, R.: Multiprotocol Label Switching Architecture, IETF RFC3031 (2001).
2) Andersson, L. and Rosen, E.: A Framework for Layer 2 Virtual Private networks (L2VPNs), IETF RFC4664 (2006).
3) Malis, A.G.: Converged Services over MPLS, *IEEE Communication Magazine* (2006).
4) Gopal, R.: Layered Model for Supporting Fault Isolation and Recovery, *IEEE Network Operation and Management Symposium* (2000).
5) Pan, P., Swallow, G. and Atlas, A.: Fast Reroute Extensions to RSVP-TE for LSP Tunnels, IETF RFC4090 (2005).

6) Awduche, D., et al.: RSVP-TE: Extensions to RSVP for LSP Tunnels, IETF RFC3209 (2001).
7) Steinder, M. and Sethi, A.S.: A survey of fault localization techniques in computer networks, *Science of Computer Programming* (2004).
8) Yemini, S.A., Kliger, S., Mozes, E., Yemini, Y. and Ohsie, D.: High speed and robust event correlation, *IEEE Communications Magazine* (1996).
9) Katzela, I. and Schwartz, M.: Schemes for fault identification in communication networks, *IEEE Trans. Networking*, Vol.3, No.6 (1995).
10) Steinder, M. and Sethi, A.S.: Probabilistic Fault Localization in Communication Systems Using Belief Networks, *IEEE/ACM Trans. Netowrking* (2004).
11) Lonvick, C.: The BSD Syslog Protocol, IETF RFC3164 (2001).
12) CiscoSystems: Cisco Discovery Protocol. http://www.cisco.com/en/US/tech/tk648/tk362/tk100/tsd_technology_support_sub-protocol_home.html
13) IEEE: Station and Media Access Control Connectivity Discovery, IEEE Std. 802.1AB (2005).
14) Cucchiara, J., Sjostrand, H. and Luciani, J.: Definitions of Managed Objects for the Multiprotocol Label Switching, Label Distribution Protocol, IETF RFC3815 (2004).
15) Joyal, D., Galecki, P., Giacalone, S., Coltun, R. and Baker, F.: OSPF Version 2 Management Information Base, IETF RFC4750 (2006).
16) Srinivasan, C., Bloomberg, L.P., Viswanathan, A. and Nadeau, T.: Multiprotocol Label Switching Traffic Engineering Management Information Base, IETF RFC3812 (2004).
17) CiscoSystems: Cisco IOS Embedded Event Manager (EEM). http://www.cisco.com/en/US/products/ps6815/products_ios_protocol_group_home.html
18) 上手祐治ほか：大規模ネットワークに適した自動障害部位特定アルゴリズムの実装方式と性能評価，信学技報 ICM2009-14 (2009).

(平成 22 年 5 月 24 日受付)
(平成 22 年 12 月 1 日採録)

渡辺　修（正会員）

1985 年九州大学工学部電子工学科卒業．現在，早稲田大学大学院情報生産システム研究科博士課程在学中，シスコシステムズ合同会社勤務．ビデオ配信システム，ネットワーク管理システムに関する研究とコンサルティングに従事．電子情報通信学会，IEEE，ACM 各会員．

小柳　恵一（正会員）

1975 年慶應義塾大学工学部電気学科卒業．1977 年同大学大学院修士課程修了．1998 年大阪大学大学院博士課程修了．1977 年日本電電公社（現 NTT）入社．工学博士．現在，早稲田大学大学院情報生産システム研究科教授．ネットワークミドルウェア，知能化ネットワークの研究開発に従事．電子情報通信学会，ACM 各会員，IEEE Senior Member.