

階層間マッピングによるネットワーク構成可視化技術 Visualization Technology of Network Topology by Layer Mapping

市原 利浩† 魚住 光成† 荒井 和昌‡

Toshihiro Ichihara Mitsunari Uozumi Kazumasa Arai

1. はじめに

近年、仮想スイッチや VLAN (Virtual Local Area Network), VPN (Virtual Private Network) 等を組み合わせた設計技術により、ネットワークを構成する物理的な機器、回線を仮想化し、論理的なネットワークを複数構築することが可能となっている。これに伴う、ネットワーク管理の複雑化により、従来の物理的なネットワークの管理だけでは、論理的な接続関係や工事の影響範囲の把握が困難となっている。

現状、論理構成を含むネットワーク構成の妥当性は、各機器から収集した接続情報から、相互の接続関係を人手でトレースして把握している。その為、ネットワーク構築時や構成変更時の確認作業の高負荷や、確認ミスによるネットワーク障害の発生などの課題があった。

そこで、我々は、各機器から収集した情報を基に、OSI (Open Systems Interconnection) 参照モデルにおけるレイヤー 1 (L1) の接続状態、レイヤー 2 (L2) のスパンニングツリー (STP : Spanning Tree Protocol) 状態、レイヤー 3 (L3) のルーティング経路を把握し、各レイヤーの状態を重畳して 3D 表示を行うネットワーク構成可視化技術を開発した。本稿では、このネットワーク構成可視化技術について、我々のアプローチとその試行結果を報告する。

2. ネットワーク管理の課題

2.1 手作業による構成の把握

ネットワークを構成する機器は、主にルーター、スイッチ、ブリッジ、リピータ、サーバなどから成る。スイッチには、イーサネット層の packet 転送を行う L2 スwitch (スイッチングハブ) と、ネットワーク層での転送処理を行う L3 スwitch がある。

これらの機器から成るネットワークの構成を確認する場合、下記の 3 つのレイヤーごとに切り分けて、考える必要がある。

- (1) 物理層 (L1) : 物理的な結線状態
- (2) データリンク層 (L2) : イーサネット packet を転送する論理的な接続状態。STP によるループ経路の遮断状態や VLAN による論理ネットワークの構成状態
- (3) ネットワーク層 (L3) : ルーティングテーブルによる IP packet の転送経路

ルーターや L2 スwitch, L3 スwitch は、管理用として、個々に IP アドレスを持ち、自身の機器に物理的に接続している機器情報や、STP によりブロックしているポート番号などのネットワーク I/F を識別する情報を保有する。

また、ルーターは、IP packet の転送先情報であるルーティングテーブルを保持する。

これらの情報は、ネットワークに接続した管理用端末から各機器にログインし、コマンド入力によって収集可能である。

ネットワーク管理者は、収集した各機器の接続情報を解析し、トレースすることで、ネットワークの物理的・論理的な構成を描画している。従来、これらの作業を全てで行っているため、非常に負荷の高い作業となっている。

2.2 ネットワーク構成管理ツール

2.1 節に示すように、ネットワーク構成の描画を全て手作業で行うには、非常に時間を要するため、自動的に接続情報を収集し、描画するツールが存在する。しかし、これらのツールは、L1 の物理的な結線状態を表示しないものが多く、また、L1~L3 の結線状態を表示するものは、各レイヤーの構成を個別に表示するため、各レイヤーの状態を比較して問題の切り分けを行うことが難しい。

3. 我々のアプローチ

3.1 可視化技術の概要

今回我々が開発した可視化技術は、各機器が保有する接続情報の自動収集、収集した情報の解析、解析結果からのネットワーク構成の 3D 表示を行うものである。2.2 節課題を解決するため、ネットワーク構成の表示では、L1~L3 の構成を重畳表示する (図 1 参照)。これにより、レイヤー間の依存関係などの確認作業を行い易くする。

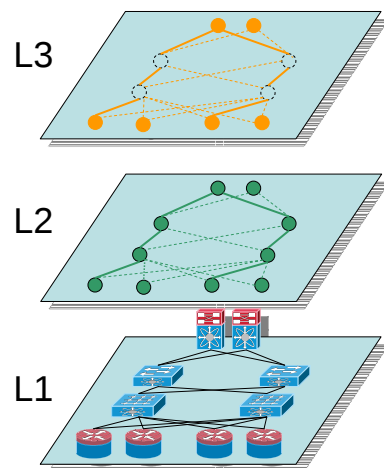


図 1 : L1~L3 の重畳表示の例

図 1 は、L1~L3 の接続状態を表示した例である。L1 部分の構成図は、物理的な結線状態を示している。ケーブル故障などによる断線が発生すると、その箇所の結線は表示されなくなる。L2 部分は、論理的なネットワーク構成を示す。STP によって遮断している経路は点線表示し、全ての

†三菱電機 (株) ,Mitsubishi Electric Corporation

‡三菱電機インフォメーションシステムズ (株) ,
Mitsubishi Electric Information Systems Corporation

機器を表示する。L3 部分は、各ルーター間の IP パケットの経路を表示する。通常、L3 相当のネットワーク構成には、イーサネット層の機器である L2 スイッチは含まれないが、本可視化技術では、L2 の経路情報と L3 のルーティング情報をマッピングすることで、ルーター間の L2 スイッチも表示する。これによって、IP レベルのネットワークにおける物理的な結線状態を把握し易くする。

各レイヤーの結線状態は、ネットワーク機器の接続情報を含むテキストファイルを字句解析し、ホスト名やポート番号などのネットワーク I/F の ID をトレースすることで、結線状態を作成する。このテキストファイルは、ネットワーク機器が同一の場合でも、OS の種類やバージョンによって、機器が出力するテキスト形式が異なる場合がある。そのため、各種 OS 等に対応した解析用 API と、使用している機器の種類を定義する外部定義ファイルを用意することで、機種依存の部分を吸収する。

3.2 隣接行列による経路管理

各ネットワーク機器をノードとした場合、ネットワークの結線状態を論理的に表す方法として、隣接リスト、結合行列、隣接行列がある。

隣接リストは、各ノードに着目して、個々のノードに直接接続されているノードを列挙する方法であるが、全体の接続関係やループ経路の有無を評価しようとすると、煩雑なアルゴリズムになりやすいという欠点がある。

結合行列は、ノードのリストを行に、ノード間の接続を列とした行列を用いて接続関係を表す方法である。この方法は、ノードの数と辺の数に応じて行列の大きさが変わり、必ずしも正方行列とはならず、行列としての可能な演算に制約が発生する。

隣接行列は、ノードを行と列とした行列として、接続関係を表す方法である。この方法は、常に正方行列であり、ループ評価が行いやすい方法である。欠点として、データの冗長性が高く、計算の際のメモリを消費する。しかし、このメモリ消費は、昨今のコンピュータでは制約にはならず、スパース行列や圧縮アルゴリズムにより、コンパクトに保持可能であるため、本可視化技術では、隣接行列によって経路を管理している。

3.3 各レイヤーの経路導出方法

図 2 に示すように、本可視化技術では、レイヤーごとに隣接行列を保持する。L1 の隣接行列は、解析した各機器の物理結線情報から各機器をトレースしていくことで作成す

る。L2 の隣接行列は、STP によって転送をブロックしているポートと、接続先ホスト名により特定した遮断経路を、L1 の隣接行列の経路から取り除くことによって生成する。

L3 部分の隣接行列は、ルーティング情報と L2 の隣接行列を基に、再帰的探索によって導出する。

このように、L2、L3 の隣接行列は、一つ下位層の隣接行列をベースにすることで、経路導出の効率化を図っている。

4. 終わりに

今回、5～20 程度のルーターと L2 スイッチを利用して、数パターンのネットワークを構築し、機器情報の取得と解析、及び各レイヤー構成の表示を行った。これにより、従来ツールでは、表示されなかった、L2 の遮断されている経路の表示や、ルーター間の L2 スイッチを含む経路の表示が確認できた。更に、L2 と L3 の構成が、L1 の物理構成にマッピングして表示されるため、各レイヤーの依存関係が見やすくなっている。これらにより、構築・変更時のネットワーク構成の妥当性確認や障害発生時の問題点の切り分けが容易になったと考える。

今後の課題としては、実用性を高めるため、適用可能な機種種の拡充や、拡大・縮小表示などによる大規模ネットワークへの対応、選択した機器の詳細情報表示などのユーザビリティ向上が挙げられる。

また、大規模ネットワーク時の隣接行列による経路探索などの演算コストの評価を行い、必要に応じて、より効率的な経路導出方法を検討する。

今後は、実ネットワークへの適用による試使用評価を行う予定である。

参考文献

- [1] CISCO Systems, アーキテクチャ概要：データセンター ネットワークにおける Cisco Catalyst 6500 シリーズおよび Cisco Nexus 7000 シリーズのスイッチング テクノロジーの使用, CISCO Systems, 2008 年
- [2] アラン・ドーラン他, よくわかるネットワークのアルゴリズム, 日本評論社, 2003 年 2 月
- [3] ヘンリー・スウィーザル 他, The Java 3D API 仕様, アスキー, 1999 年 2 月

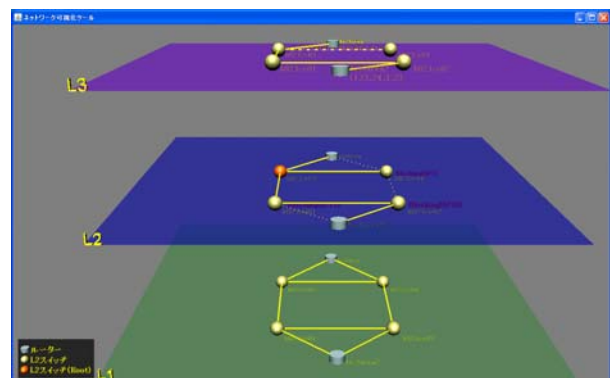
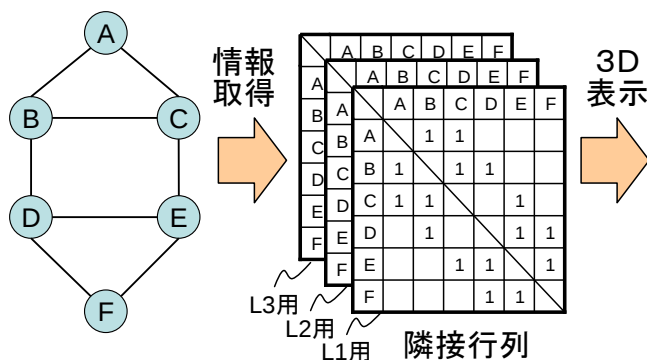


図 2：隣接行列による経路管理