

実機ベース汎用大規模実験環境の状態の保存・復元機構

野中雄太^{†1} 知念賢一^{†1,†2} 宮地利幸^{†2}
三輪信介^{†3} 篠田陽一^{†4,†2,†3}

ネットワーク技術を検証するためには実機を用いた実験環境を用いる必要があるが、検証はパラメータの変更や不具合の修正を行いながら繰り返し行うため、検証のたびに手動で実機を用いた実験環境を構築することは現実的ではない。そのため、実機を用いた実験環境のネットワークポロジやソフトウェアの構成等の状態を保存しておき、再検証の際に復元できればネットワーク技術の検証を容易にできる。また、実験環境の状態を復元する際に保存時と違う機器を用いて柔軟に行うことができれば、多人数での利用時や機器の故障時にも対応できる。

本研究では、実機を用いた大規模な実験環境の状態を柔軟に保存・復元するための機構を提案・設計する。本論文ではその設計と実装、そして評価について述べる。実装と評価は情報通信研究機構 北陸リサーチセンターにあるネットワークテストベッド、StarBED において実施した。

Suspend and Resume of States of Experiment States on Actual Node base Test-bed

YUTA NONAKA,^{†1} KEN-ICHI CHINEN,^{†1,†2} TOSHIYUKI MIYACHI,^{†2}
SHINSUKE MIWA^{†3} and YOICHI SHINODA^{†4,†2,†3}

To reduce setup time and human error of experiment to evaluate new technologies, the mechanism that suspends and resumes the situation of them is needed. To approach real behavior of network complex, we employ actual node based experimental facilities and run various network experiments with dynamic construction. However it takes long time to construct them. In addition, it also causes human mistakes when operator does it manually. As the scale of experiments increasing, these issues are increasing. Then, we design suspend and resume mechanism of experiment situation for such facilities. Since large-scale experimental facilities should be used in many people to save cost, the mechanism have to care such case. Our mechanism can also resume situation with alternative resources when resources suspended are occupied.

This paper describes design, implementation and evaluation of our mechanism. The mechanism is developed on the StarBED.

1. はじめに

ネットワーク技術の実装を開発する際には、動作検証が不可欠である。動作検証を行うためには、検証対象となる技術に応じた構成の実験環境が必要になる。

実験環境は検証の段階によって変化し、同じ構成の実験環境を何度も利用することが多い。一般に実験環境は小規模で単純な構成から大規模で複雑な構成へと段階的に変化するが、実装の不具合が発見された場合などに、その前段階の構成の実験環境に戻して再実験が行われる。また、ネットワーク技術の検証では、同じ構成の実験環境での試験を、実装に与えるパラメータ等を変更して何度も繰り返し行うことも多い。

以上のような検証の繰り返しの際して、同じ構成の実験環境を検証のたびに手動で構築するのは手間がかかる上、ミスが発生することも考えられる。この問題は実験環境が大規模になるほど顕著になる。そこで、実験環境を構成する機器やネットワークポロジといった状態を保存・復元する機構が必要である。この機構があれば、検証の過程で何度も利用される実験環

^{†1} 北陸先端科学技術大学院大学 情報科学研究科
School of Information Science, Japan Advanced Institute of Science and Technology

^{†2} 情報通信研究機構 北陸リサーチセンター
Hokuriku Research Center, National Institute of Information and Communications Technology

^{†3} 情報通信研究機構 情報通信セキュリティ研究センター
Information Security Research Center, National Institute of Information and Communications Technology

^{†4} 北陸先端科学技術大学院大学 情報科学センター
Center for Information Science, Japan Advanced Institute of Science and Technology

境をその都度手動で構築し直す必要がなくなり、実験環境の構築にかかる人的・時間的コストを削減できる。

さらに、ある種の技術を検証するための実験環境の構成は類型化できる。例えば、サーバ負荷試験に用いられる実験環境の構成は、対象となるサーバの種類によらず、ある程度類似したものになることが予想される。実験環境の状態を保存・復元する機構があれば、実験環境の状態を他の検証に再利用することもでき、検証を容易にできる。

本論文では、実機を用いた実験環境の保存・復元機構の設計と実装、そして実装の評価について述べる。

2. 実験環境とその保存・復元手法

2.1 対象とする設備と実験環境

ネットワーク技術の検証は、その技術が運用される環境に近い環境で行う必要がある。このような環境では、実機を用いることが望ましい。これは実環境での実装を利用できるためである。実機を用いて様々なネットワークトポロジを構築するためには、多数の機器が必要になるが、多数の機器を個人で独占することは困難であるため、多人数での利用を考慮する必要がある。

そのため、本論文では多人数で共用される実機群を対象とする。多人数で共用する機器群全体を「実験設備」と呼び、実験設備の中で特定の実験者が検証に用いる機器群を「実験環境」と呼ぶ。

2.2 実験環境の状態の柔軟な復元

多人数での共用を前提とした実験設備上で実験環境の状態の保存・復元を行う場合、保存時の実験環境で使用していた機器等の共用の資源が、復元時には他の実験者に使用されている場合が考えられる。このような場合、保存時とは異なった資源を用いて保存時と同様の実験環境を復元することができれば、実験者が実験環境の状態を復元できる可能性が高まり、実験設備を有効に利用することができる。さらに、多数の実機を用いた実験設備では機器が故障する頻度も高くなるが、保存時の機器と異なった機器で復元できれば、故障時にも対応できる。

図 1 は、代替機器群を用いて実験環境の状態を復元した例である。この例では保存時の状態 (図 1(a)) のクライアント A 群が利用不可能な場合に図 1(b) のようにクライアント C 群を代替として復元している。

また、保存した実験環境の状態を部分的に復元できれば、保存した状態の他の実験への再利用が容易になる。例えば、実験環境のネットワークトポロジとソフトウェア構成を保存したとして、復元時にはネット

ワークトポロジだけを復元することが考えられ、この場合、ネットワークトポロジとソフトウェア構成の両方を復元する場合よりも短時間で復元が可能になると予想される。

2.3 実験環境の構成要素と状態

実験環境にはハードウェアやソフトウェアといった様々な要素が存在しており、これらの要素によって、物理トポロジ、論理トポロジ、ソフトウェアの 3 つの状態が構成される。

まず、ハードウェアとして、計算機、スイッチ・ルータといったネットワーク機器、それらを物理的に接続する各種ケーブルが存在する。これらのハードウェア同士が物理的に接続されることによって、実験設備のネットワークの物理トポロジが構成されている。次に、計算機やネットワーク機器の設定によって構成される論理トポロジがある。これには、スイッチの VLAN によるトポロジ等がある。そして、各計算機のソフトウェアがある。これには計算機上のソフトウェアの構成やプロセスの状態等がある。

これら 3 つの状態を保存・復元することで、実験環境全体の状態を保存・復元できる。また、論理トポロジとソフトウェアを保存・復元する仕組みを分割して実装することによって、ネットワークトポロジだけの復元といった部分的な状態の復元が可能になると考えられる。

2.4 各状態の保存・復元手法

2.4.1 各状態の性質

物理トポロジ、論理トポロジ、ソフトウェアの状態にはそれぞれ特徴があり、実験環境の状態を保存・復元する際にはそれを考慮しなければならない。

まず、物理トポロジを構築するためには手動で配線作業を行う必要がある。この作業をソフトウェアで自動化することは困難である。

次に、論理トポロジは、ソフトウェアで変更を行える。また、多くの共通のプロトコルで運用されているという特徴がある。つまり、ネットワーク機器のベンダや計算機の OS により設定のためのコマンドや設定ファイルの書式が異なっても、それらの設定は TCP/IP をはじめとした共通のプロトコル群に準拠しているものも多く、それらに準拠した設定の意味には互換性がある。

論理トポロジが様々な機器や OS で同じ意味を持つものが多いのに比べ、ソフトウェアの状態は複雑である。まず、Microsoft Windows、Mac OS X、各種 Linux ディストリビューション等の OS が存在し、それぞれ動作に必要とするハードウェアの条件が異な

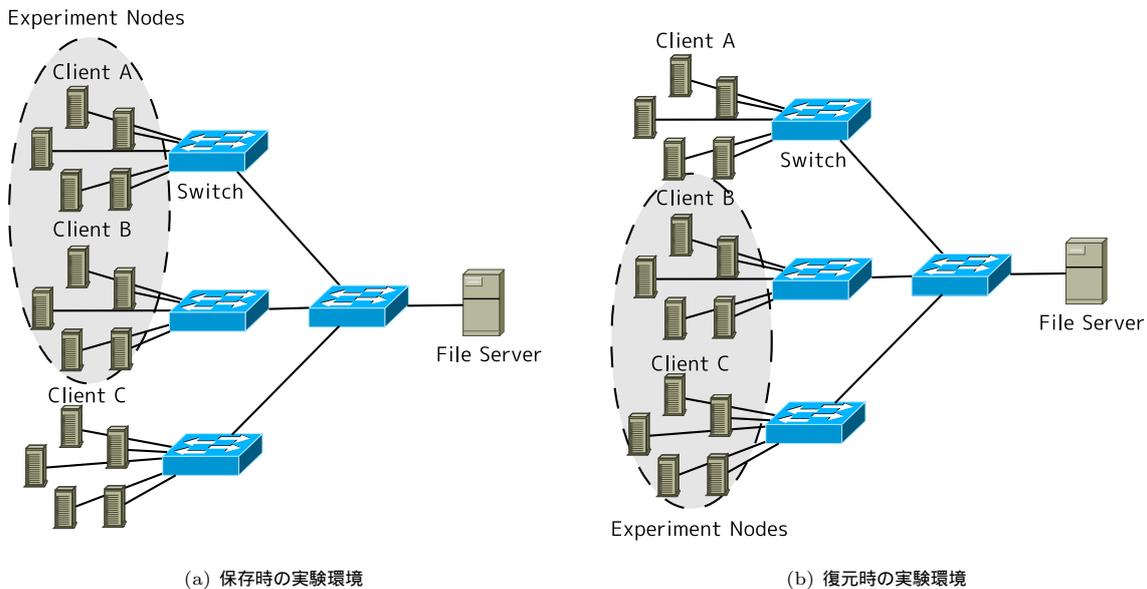


図 1 柔軟な復元の例

る。また、各 OS ではそれぞれ異なったファイルシステムを必要とし、動作可能なアプリケーションも OS によって異なる。

2.4.2 保存・復元手法の検討

以上の特徴を考慮して、各状態の保存・復元手法を検討する。物理トポロジに関しては、ソフトウェアで変更することが困難であるため、保存時には保存時の物理トポロジの構成を取得・記録し、復元時には保存時の物理トポロジと復元時の物理トポロジを比較して利用する機器を選択することになる。そのための仕組みとして、実験設備上の物理トポロジの構成を手動で登録する仕組みや、自動で物理トポロジを把握する仕組みが必要となる。

論理トポロジを構成するための設定は、前述した通り機器のベンダや計算機の OS に関わらず同一の意味を持っているものが多い。各ベンダ等によって独自のプロトコルを用いる場合もあるが、一般的には共通のプロトコルで論理トポロジの構築は可能である。そのため、共通のプロトコルに準拠した論理トポロジに関する情報を機器のベンダや OS に依存しない情報に抽象化して保存し、復元時に論理トポロジの情報を利用する機器に合わせて設定することで、保存時と異なった機器を用いて、保存時と等価な実験環境を復元できる。また、論理トポロジは物理トポロジ上に構成されるため、論理トポロジを保存・復元するためには物理トポロジを把握しなければならない。そのための方法として、前述した物理トポロジを把握する仕組みを利

用して取得することが考えられる。

ソフトウェアの状態に関する要素は、導入されている OS やアプリケーションの構成などの静的な要素と、プロセスの動作状況やメモリ・CPU の使用状況などの動的な要素に分けられる。静的な要素の保存・復元手法としては、ハードディスクのパーティションをディスクイメージとして保存する方法や、各ファイルシステムに合わせたバックアップ手法が考えられる。また、導入されているソフトウェアによって必要とされる機器のスペックが異なるため、多人数での利用を考慮する場合には、復元時に用いる計算機上で保存時のソフトウェアが動作するかどうかを判定する必要がある。その方法として、物理トポロジを把握する際に計算機のスペックも同時に把握し、それを利用して対象ソフトウェアが動作するかどうかを判定する仕組みが考えられる。

ソフトウェアの動的な要素の保存・復元手法としては、数多く提案されているプロセスマイグレーションの方法を利用することが考えられる。ただし、それらには特殊な OS を必要とするなどの制約があるため、実験者の求める様々な OS に対応することが困難である。そのため、プロセスマイグレーションに関しては本論文での対象としない。

以降では、物理トポロジを把握する仕組み、論理トポロジ、ソフトウェア構成をそれぞれ保存・復元する仕組みについて詳細な設計を述べる。

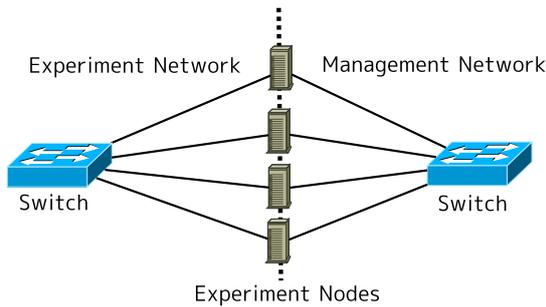


図 2 実験用と管理用ネットワーク

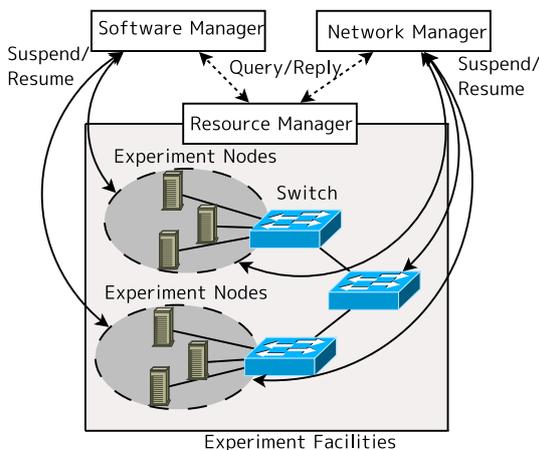


図 3 提案手法概念図

3. 保存・復元機構の設計

3.1 実験設備の設計

本論文で対象とする実験設備は、2.1 節での定義に加え、図 2 のような、実験用ネットワークと管理用ネットワークが分かれているものとする。実験用ネットワークは実験者が検証に用いるためのネットワークであり、管理用ネットワークは保存・復元のために必要な通信を行うためのネットワークである。2 つのネットワークを分離したのは、検証に用いているネットワークに保存・復元のためのトラフィックが混ざることや、検証の状況により到達性がなくなった計算機に対しても、保存・復元のための通信を行う必要があるという理由である。

また、本論文で保存・復元の対象とする実験設備の機器は計算機と VLAN を扱えるネットワークスイッチのみとし、論理トポロジは VLAN を用いて仮想的に構築することとする。これは、検証に使われる実験環境の機器の多くは計算機であり、その状態を保存・復元することで多くの技術の検証に対応できると考え

たためである。例えば、計算機をルータとして動作させることによって、ルーティングを考慮した複雑な環境での検証にも対応できる。

設計した実験環境の状態の保存・復元機構の概念図を図 3 に示す。まず、実験設備の物理トポロジや機器のスペックは資源管理機構 (図中 Resource Manager) によって把握される。また、資源管理機構では実験設備の使用状況も把握する。ネットワークの保存・復元機構 (Network Manager) は、実験者の指定した実験環境の物理トポロジを資源管理機構から取得し、物理トポロジにしたがって必要な機器の論理トポロジを保存・復元する。ソフトウェア管理機構 (Software Manager) は実験者が指定した計算機のソフトウェア構成をハードディスクのパーティションのディスクイメージやファイルシステムのバックアップとして保存・復元する。

以下に、各機構の詳細について述べる。

3.2 資源管理機構

資源管理機構は実験設備で共用されている資源を把握し、ネットワーク保存・復元機構やソフトウェア管理機構が実験環境の状態を保存・復元する際に必要となる情報を与える機能を持つ。資源管理機構が把握する情報は以下の通りである。

- 物理トポロジ
- 機器スペック
- 機器識別子 (ホスト名)
- 資源利用状況

実験設備の物理トポロジを構成する要素は、計算機のネットワークインタフェース (以降 IF) とスイッチの IF との接続状態、スイッチの IF とスイッチの IF の接続状態、などの機器同士の接続状態である。資源管理機構は、各機器の各 IF がどの機器のどの IF に接続されているかを把握することで、実験設備全体の物理トポロジを把握する。また、保存時と復元時の実験環境のネットワークの同一性を保つためには、各機器間の帯域等のリンク特性も復元可能である必要がある。そのため、資源管理機構は各機器の IF の規格についても把握する。また、実験設備内の各機器を識別するためのホスト名も資源管理機構は保持する。

実験設備内の機器のスペックは、ソフトウェアを保存・復元する際に重要になる。ソフトウェアの動作に関連するものとして、CPU の種類・速度、メモリの種類・容量、チップセットの種類、ハードディスク容量が挙げられる。CPU、メモリ、チップセットは、保存したソフトウェアが動作するかどうかと、動作速度に関連する。ハードディスク容量は保存したソフトウェ

アを導入できるかどうかに関連する。そのため、これらの情報を資源管理機構は把握する。

実験設備内の共用資源の利用状況は、多人数で実験設備を利用するために把握しておく。把握する情報としては、実験設備内の計算機の利用状況、実験設備内で利用されている VLAN 番号が挙げられる。

3.3 ネットワーク保存・復元機構

ネットワーク保存・復元機構は、実験環境のネットワークトポロジの保存・復元を行う。この機構は、実験環境の保存・復元時に資源管理機構と通信を行い、物理トポロジ等の情報を取得する。そして、保存・復元対象機器の設定情報の取得、対象機器の設定を行い、実験環境の論理トポロジを保存・復元する。論理トポロジを保存・復元するために必要な情報は以下の通りである。この情報と、資源管理機構が持つ情報を組み合わせることで物理トポロジと論理トポロジからなるネットワークトポロジを保存・復元できる。

- 計算機の IF の IP アドレス、経路情報
- 計算機の IF が接続されたスイッチの IF の VLAN 番号

図 4 はネットワーク保存・復元機構の動作を示したものである。図 4(a) は保存時の、図 4(b) は復元時の動作を示す。

保存時、実験者はネットワークの状態を保存したい実験環境上の計算機群をネットワーク保存・復元機構に指定する。計算機群を指定されたネットワーク保存・復元機構は、資源管理機構と通信を行い、計算機群とスイッチの名前とそのスイッチのポートとの対応に関する情報などの、物理トポロジの情報、計算機群のスペックの情報を取得する。その後、ネットワーク保存・復元機構は指定された計算機群と接続されているスイッチの IF の VLAN 番号を取得する。また、指定されたホスト名の計算機群の持つ経路情報と実験用 IF に設定された IP アドレスを計算機群から取得する。そして、計算機群のホスト名、実験用 IF の用途 (管理用ネットワークに接続された IF か、実験用ネットワークに接続された IF か)、各インタフェースの規格、各 IF に接続されたスイッチのホスト名、VLAN 番号、機器のスペックを記述したファイルを出力する。このファイルを State File と呼ぶことにする。このとき、State File に記述される情報は、ネットワーク機器のベンダや OS に依存しない書式で保存される。

状態の復元時には State File を読み込み、それに基づいてスイッチ及び計算機群を設定し、ネットワークトポロジを復元する。実験設備のどの機器を用いてネットワークトポロジを復元するかは、保存時の機器

での復元、実験者の指定した機器による復元、資源管理機構が割り当てた任意の機器による復元が考えられる。それらのネットワーク保存・復元機構はどの場合でも資源管理機構に使用権限を確認する。これは、他の実験者が使用中の機器の使用を避けるためである。また、保存時のネットワークトポロジをどこまで厳密に復元するかは実験者が指定する。例えば、リンク特性が違ってネットワークの形さえ同じであれば復元を許す、等の指定が考えられる。

この機構により、以下のようなパターンで実験環境のネットワークトポロジの保存と復元が可能となる。

- (1) 同構成での復元
保存時と同じ計算機、IP アドレス、VLAN 番号を用いて復元する
- (2) 代替計算機を用いた復元
実験環境を構成する計算機を指定して復元する
- (3) 代替 VLAN を用いた復元
保存時に利用していた VLAN 番号が使用中である場合、別の VLAN 番号を用いて復元する
- (4) 代替リンク特性を用いた復元
保存時とはリンク特性が異なるネットワークトポロジでの復元

3.4 ソフトウェア管理機構

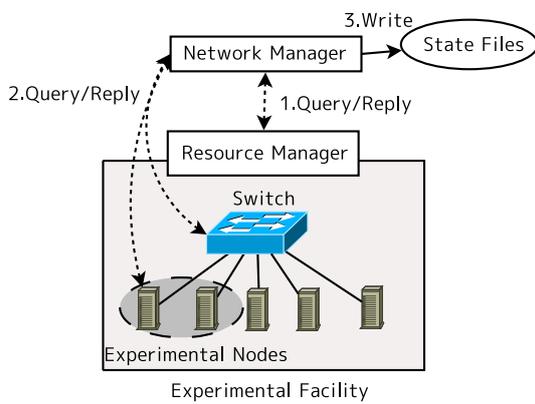
ソフトウェア管理機構は、実験者が指定した計算機のソフトウェア構成を保存・復元する。

ソフトウェアの状態の保存と復元の方法として以下が挙げられる。

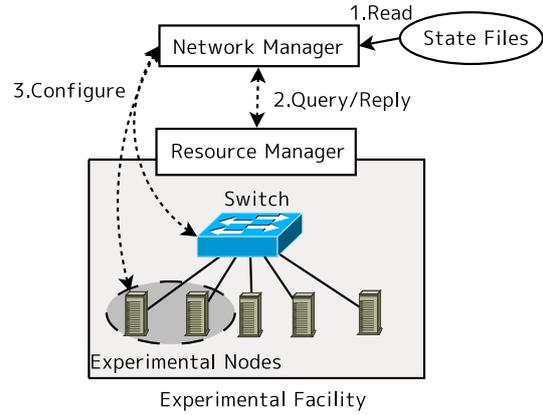
- (1) 固定した構成の OS とアプリケーションを用いる方法
- (2) 計算機のハードディスクをディスクイメージとして保存・復元する方法
- (3) ファイルシステムとしてバックアップする方法

固定した構成のソフトウェアとして、実験に多く使われる OS とソフトウェアをあらかじめハードディスクの特定のパーティションに用意しておけば、部分的な状態の復元に柔軟に対応できる。これは、ソフトウェアを復元する際にハードディスクへの書き込みを行わないため、高速に実験環境の構築が行えるためである。また、検証の最初の段階においては、実験者が自分でソフトウェアを導入する手間を省くことができるため、実験の準備時間も削減できる。

ディスクイメージとしてソフトウェア構成を保存する方法は、保存対象の OS に依存せずに実験者が必要とするソフトウェア構成を再現できる。そのため、未知の OS にも対応することが可能である。しかし、計算機によっては作成したディスクイメージを導入して

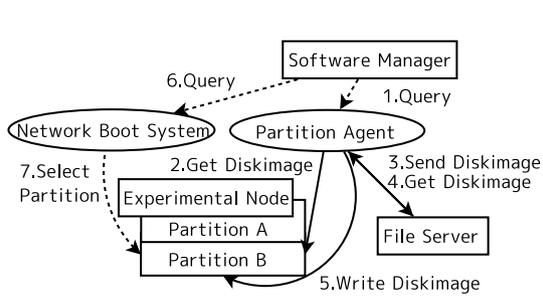


(a) 保存時の動作

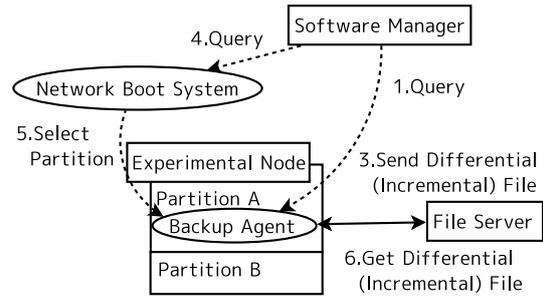


(b) 復元時の動作

図 4 ネットワーク保存・復元機構の動作



(a) ディスクイメージ使用時



(b) ファイルシステム使用時

図 5 ソフトウェア管理機構の動作

も動作しない場合が考えられる。また、常にハードディスク上の保存対象のパーティションのデータ全てを書き出すため、保存・復元にかかる時間も大きくなる。

ファイルシステムとしてバックアップする方法は、保存・復元の対象となるファイルシステムによって保存・復元の方法の詳細が違いため、処理が複雑になるという問題がある。しかし、差分バックアップや増分バックアップ等のバックアップの手法により、ディスクイメージとして扱う方法に比べて高速にソフトウェア構成を保存・復元することが可能である。また、前述した固定した構成のソフトウェアを利用し、その変更点をバックアップすることもできる。

以上の 3 つの方法を組み合わせるソフトウェア管理機構を設計した。前述した 3 つの方法の特徴から、計算機のハードディスクを複数のパーティションに分割し、各パーティションを固定ソフトウェア用と実験者が任意に OS を導入できるパーティションに用いる

ことにした。これにより、固定した OS とソフトウェアを利用する実験者にも、独自のソフトウェアを用いた実験を行いたい実験者にも対応できる。保存・復元方法としては、固定ソフトウェアはファイルシステムとして変更点をバックアップし、任意の OS 用のパーティションはディスクイメージとして保存・復元する。

図 5 に、ソフトウェア管理機構 (Software Manager) の構成と動作を図示した。図 5(a) と図 5(b) はそれぞれ、ソフトウェア構成の保存・復元にディスクイメージとファイルシステムを用いる場合の動作となる。

どちらの場合も、ネットワークを利用して計算機の起動パーティションを指定する。この機能を持つのが、図中の Network Boot System である。これには PXE 等を利用する。またディスクイメージとしてソフトウェア構成を保存・復元する機構が図中の Partition Agent であり、固定ソフトウェアのファイルシステムのバックアップとレストアを行うのが図中の Backup Agent

である。

保存時、計算機のファイルシステムをバックアップする場合、Software Manager は Backup Agent を用いてファイルシステムの変更点をバックアップする。Backup Agent はあらかじめ固定ソフトウェア群に組み込んでおく。計算機のハードディスクのディスクイメージを保存する場合、Software Manager は Partition Agent を用い計算機のハードディスクの対象パーティションのバイナリデータを取得し、ファイルサーバに転送する。

復元時、保存した計算機のディスクイメージを復元する場合、Partition Agent はディスクイメージを保存してあるファイルサーバからディスクイメージを取得し、復元対象の計算機のハードディスクに書き込む。そして、ネットワークを利用して起動パーティションを指定して起動させる。その後、計算機のファイルシステムとして扱う場合には Backup Manager によるレストアが行われる。

4. 実装

ネットワーク保存・復元機構のスイッチの状態を保存・復元する部分を情報通信研究機構 北陸リサーチセンターの実験施設である StarBED^{1),2)} に実装した。StarBED に導入したのは、StarBED が本論文で対象としている管理用ネットワークと実験用ネットワークが存在している実験設備であり、さらに多数の機器を用いた大規模な実験環境を構築できるためである。資源管理機構として、StarBED の資源管理機構である erm を用いた。erm は、StarBED 上の機器について、以下の情報を保持する。

- 各機器のホスト名
- 計算機の IF の情報
管理用 IF か実験用 IF か、規格、接続スイッチホスト名、接続スイッチ IF
- スイッチ間の接続状態
- スイッチのコマンド体系

Network Manager は実験者に指定された計算機の接続されているスイッチの状態を State File に保存し、それを用いて復元を行う。State File の書式は図 6 とした。

実験者に保存対象の計算機のホスト名を指定されると、Network Manager は erm と協調して以下の動作を行う。

- (1) erm と通信を行い、計算機の IF の情報、スイッチのコマンド体系を取得する
- (2) 計算機の IF の情報から、実験用 IF が接続され

ているスイッチにログインする

- (3) 計算機の IF が接続されているスイッチ IF の VLAN 番号を取得する
- (4) erm から得た情報と、スイッチから得た VLAN 情報を基に、State File を出力する

保存時に、Network Manager はスイッチにログインして設定を取得している。スイッチの各ベンダのコマンド体系に合わせた手順でコマンドを入力し、取得した設定をベンダに依存しない書式に変更する。現在対応しているベンダとコマンド体系は Foundry Networks の Ironware と Cisco Systems の Cisco IOS である。

復元時の Network Manager の動作は以下の通りである。

- (1) erm と通信を行い、計算機の IF の情報、スイッチのコマンド体系を取得する
- (2) 計算機の IF の情報から、実験用 IF が接続されているスイッチにログインする
- (3) 計算機の IF が接続されているスイッチ IF の VLAN 番号を設定する

Network Manager は、復元時に、実験者が復元対象の計算機を指定しなければ保存時の機器で復元を行い、復元対象の計算機が指定された場合はその計算機で復元する。また、保存時の VLAN 番号が他の実験者に使用されていた場合には、実験者からの指示により別の VLAN 番号を利用しての復元も可能である。

5. 実行と評価

5.1 概要

本実装を用いることで以下のような復元が可能であることを検証した。

- (1) 保存時の計算機・VLAN での復元
- (2) 代替計算機での復元
- (3) 代替 VLAN での復元
- (4) 他ベンダのスイッチでの復元

検証には図 7 のような実験環境を用い、複数のベンダの機器に対応していることを確認するため、スイッチには Foundry Networks の BigIron MG8 と Cisco Systems の Catalyst 6509 を用いてそれぞれの検証を行った。

以下に上記の機能の検証の結果と、本機構を用いた VLAN の保存にかかる時間を測定した結果について述べる。

5.2 保存時の計算機・VLAN での復元

初期の状態では、図 7 の 001 から 003 までの計算機の実験用 IF が VLAN850 に所属しているとする。

計算機	管理用 IF 識別子	実験用 IF	接続スイッチ名	接続スイッチポート	VLAN 番号
-----	------------	--------	---------	-----------	---------

図 6 State File の書式

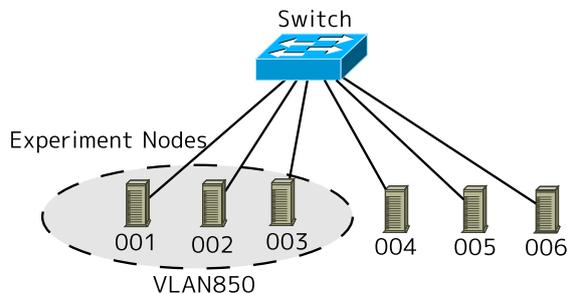


図 7 実験環境初期状態

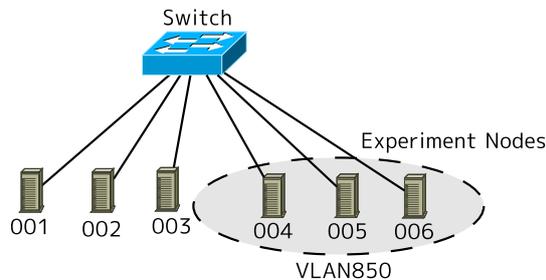


図 8 代替計算機での復元

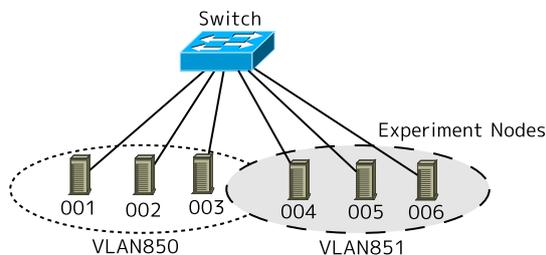


図 9 代替 VLAN での復元

001 から 003 が接続されたスイッチの状態を Network Manager で保存した後、スイッチの VLAN850 を消去した。その後、保存した状態を復元すると、001 から 003 の実験用 IF が VLAN850 に再び所属した。このように、保存時と同じ計算機、VLAN 番号を用いた復元が実現された。

5.3 代替計算機での復元

先の保存された 001 から 003 までの状態を、004 から 006 の計算機を指定して復元させた。この場合もあらかじめスイッチから VLAN850 を消去して復元し

た。その結果、図 8 のように 004 から 006 の実験用 IF が VLAN850 に所属した。これによって、実験環境を構成する計算機を指定して復元する機能を確認できた。

5.4 代替 VLAN での復元

先の保存された状態を 004 から 006 の計算機を指定して復元させるときに、VLAN850 には 001 から 003 の実験用 IF 所属しているままとし、使用中の VLAN を避けて復元するように指定して状態を復元した。その結果、図 9 のように 004 から 006 の実験用 IF は VLAN851 に所属した。これによって、使用中の VLAN 番号を自動的に避けて状態を復元する機能が動作していることが確認できた。

5.5 複数のベンダの機器への対応

また、一方のスイッチを利用した実験環境で出力した State File を、もう一方のスイッチを利用した実験環境で復元する実験を行った。対象のスイッチは、BigIron MG8 と Catalyst 6509 の両方である。評価の詳細は前述した代替計算機での復元、代替 VLAN での復元と同様であり、結果も前述と同様となった。以上より、コマンド体系に依存せず実験環境の VLAN を復元できることが確認できた。

5.6 所要時間

StarBED 上で多数の VLAN の保存と復元を行い、所要時間を測定した。対象としたスイッチは BigIron MG8、Catalyst 6509 である。VLAN の数は BigIron MG5 については 1、10、50、100、200、Catalyst6509 については 1、10、50 と変化させ、それぞれ Linux の time コマンドによって測定を行った。2 つのスイッチで対象とした VLAN の数が異なるのは、StarBED の構成の関係上、各スイッチに同じ条件で接続できる計算機が異なるためである。

表 1 は各スイッチについて保存・復元する VLAN 数を変化させたときの保存・復元の所要時間を示している。どちらのスイッチでも、保存・復元にかかる時間は、VLAN の数に対して線形に増加した。BigIron に接続された 200 台の計算機の VLAN 情報の保存を約 3 分で、復元を約 17 分で行える。Catalyst についても、結果から同程度の時間で保存・復元が行えると予想できる。

保存に比べ復元に時間がかかっているが、これは復

表 1 VLAN の保存・復元時間 (単位:秒)

VLAN(個)	BigIron		Catalyst	
	保存	復元	保存	復元
1	3	7	4	7
10	11	54	12	42
50	47	265	57	220
100	92	529		
200	189	1053		

元の方がスイッチに対して多くのオペレーションを必要とするためである。どちらの場合も、手動で復元する場合に比べ、短い時間で済、コマンド実行の回数が大幅に削減できるため実験者の作業時間も削減できる。

6. 議 論

ネットワーク技術の検証を行うために今日までに様々な手法や実験設備が提案・構築されてきた。その中で、VMware³⁾を用いた仮想マシンを利用するものに VM Nebula⁴⁾があり、多数の実機を用いるものに Netbed⁶⁾、StarBED がある。

VMware は Windows、Linux 上で x86/x64 アーキテクチャをエミュレートした仮想マシンを動作させることのできるソフトウェアである。これを用いることで少数の計算機で多数の計算機を必要とする検証を行うことが可能である。さらに、VMware には多数の仮想マシンを一元的に管理する機能があり、仮想マシンの状態を保存・復元する機能も持つ。しかし、1 台の計算機に導入できる仮想マシンの数には限界があるため、ある程度以上大規模な検証は複数の計算機を利用することになる。そのとき、計算機同士を接続するネットワーク機器の状態を保存する機能は VMware にはない。現在は、VM Nebula を拡張し、多人数での利用等を考慮した VM Nebula2⁵⁾の開発が進められている。

VM Nebula はネットワークセキュリティに関する実験を行うことを目的とし、VMware を利用して作成した仮想マシンを用いて構築した実験環境上で検証を行う実験環境である。VM Nebula では、VMware の持つ仮想マシンの状態を保存する機能を利用し、ワームやウイルスによって変更されたソフトウェアやネットワーク機器の状態を保存した任意の状態に戻すことができる。VM Nebula は特定の目的の実験への対応と目的としているため、本論文で対象とした汎用的な実験環境とはスコープが異なっている。また、複数の実験者で共用することは考慮されていない。

Netbed は ns-2 と実機を利用した実験設備である。

Netbed では実験者がソフトウェアを用いてネットワークの物理的・論理的トポロジを変更することが可能である。Netbed では実験者の要請を受けた実験を、実験機器が空き次第行う。このとき、割り当てられた利用時間を過ぎた実験環境を保存し、次の利用時に実験環境の設定を復元する。ただし、Netbed は独自のインタフェースで実験環境の構築と実行を行うため、Netbed の規格に沿って構築した実験環境のみを対象としている。それに対して、本論文での提案した機構では、構築の手法に関係なく実験環境の状態を保存・復元可能である。

StarBED は一ヶ所に集められた計算機群とスイッチで構成された実験設備であり、実験者は VLAN を利用してネットワークトポロジを構築して実験環境を構築する。StarBED には実験支援ソフトウェア群として SpringOS が用意されている。これを利用することで実験者が実験に必要なネットワークトポロジや実験手順を記述することで自動的に実験を行える。ただし、現状では手動で構築した実験環境の状態を保存・復元する機能はない。StarBED の特徴は、ネットワークスイッチと計算機のみで構成された実験設備であること、その実験支援ソフトウェアがオープンなことである。そのため、同様の設備を個人や組織で構築し、利用することが可能である。本論文での提案する機構もそのような環境を保存・復元することを目的としている。そのため、提案機構と SpringOS を組み合わせることによって、実験の自動遂行から実験環境の保存・復元までの機能を備えた実験設備を構築できるだろう。

7. おわりに

本論文では、実機を用いた大規模実験環境の状態を保存・復元する機構を検討・設計した。実験設備の要素を物理トポロジ、論理トポロジ、ソフトウェア構成の 3 つに分割し、それぞれを管理または保存・復元する機構を設計し、それらを組み合わせることで実験環境の状態を復元する。本論文で設計した機構を実装し、これを用いることで、繰り返し行われることの多い検証において実験環境を検証の都度、手動で構築し直す必要がなくなる。また、保存した状態を別の技術の検証に再利用することも可能になる。

設計した機構のうち、ネットワークを保存・復元する機構の一部のスイッチの VLAN を保存・復元する部分を StarBED で実装して、その挙動を確認した。その結果、保存した状態と異なる計算機、VLAN を用いてスイッチの状態を柔軟に保存・復元できることを確認できた。また、Foundry Networks と Cisco Systems

製のコマンド体系の異なるスイッチに対して VLAN の保存・復元が行え、違うベンダの機器で保存した情報を別のベンダの機器で復元することも可能になっている。そして、手動で実験環境を構築する場合に比べて、オペレーションの数や作業時間といった人的・時間的コストを削減することができた。

今後の課題として、ネットワーク保存・復元機構の機能のうち、計算機の情報保存・復元部分と、ソフトウェア管理機構の実装を行う。また、StarBED の資源管理機構では存在しなかった、計算機のスペックに関する記述を含めた資源管理機構を実装する。そして、設計した機構全体の定性的・定量的評価を行う。

また、本機構を、既存の実験の自動遂行機構と組み合わせ、保存した実験環境の状態を利用して検証の自動遂行が可能になる。現在のところ、StarBED の実験支援ソフトウェア群である SpringOS との連動を考えている。

参 考 文 献

- 1) StarBED Project. <http://www.starbed.org/>.
 - 2) Toshiyuki Miyachi, Ken-ichi Chinen, and Yoichi Shinoda. StarBED and SpringOS: Large-scale General Purpose Network Testbed and Supporting Software. Valuetools 2006, Pisa, Italy, ISBN 1-59593-504-5, Oct, 2006.
 - 3) VMware. <http://www.vmware.com/>.
 - 4) 三輪 信介, 滝澤 修, 大野 浩之. 仮想 PC インターネットセキュリティ実験環境『VM Nebula』の設計と構築. 暗号と情報セキュリティシンポジウム (SCIS). 電子情報通信学会, 2003 年 2 月.
 - 5) 三輪 信介, 大野 浩之. インターネットセキュリティ実験環境『VM Nebula2』の設計と構築. 暗号と情報セキュリティシンポジウム (SCIS2006). 電子情報通信学会, 2006 年 1 月.
 - 6) Brian White, Jay Lepreau, Leigh Stoller, Robert Ricci, Shashi Guruprasad, Mac Newbold, Mike Hibler, Chad Bard, and Abhijeet Joglekar. An integrated experimental environment for distributed systems and networks. pp.255-270, Boston, MA, December 2002. USENIXASSOC.
-