

シミュレーションによる NETCONF 検証環境の設計と実装

岡部 誠人[†] 廣津 登志夫[‡]

法政大学情報科学部

1. はじめに

現在、ルーターやスイッチなどのネットワーク機器の管理方法として、Command line interface (CLI) や Simple Network Management Protocol (SNMP) を用いた手法がある。CLI はコマンドによる操作によりネットワーク機器を設定するが、異なるベンダーのネットワーク機器では同じ設定内容に対するコマンドが異なるという欠点がある。SNMP はネットワーク機器やコンピュータなどの監視や制御を行うためのプロトコルで、異なるベンダーのネットワーク機器であっても標準的な設定情報や統計情報を取得可能である。しかし、主に対象のネットワーク機器から情報を取得する使用方法が多い。これらの欠点を改善した、NETCONF[1]の標準化が進められている。NETCONF では、異なるベンダーのネットワーク機器であっても、統一的な設定をする事が可能である。また、設定内容や統計情報の取得も可能である。

この NETCONF を用いた大規模なネットワークの構築や運用を行う際には、事前に検証環境で正しく動作するか確認する必要がある。例えば、設定の変更によるネットワークの到達性が保証や、設定変更中にネットワーク機器への NETCONF 接続が遮断されないかなどがこの検証に相当する。検証においては、記述内容が実通信に与える影響を確認可能であることが必要である。しかし、大規模なネットワークでは、実際の機材を用いた検証環境の構築は難しい。そこで、本研究では実ネットワークを模倣したシミュレーション環境を用意して、NETCONF の動作検証を行う環境の設計と実装を行う。

2. NETCONF

NETCONF とは IETF により標準化が進められている次世代のネットワーク機器管理用プロトコルである。管理者側の端末と管理されるネットワーク機器の間で XML により記述されたコマンドや設定内容を交換する。

表 1 NETCONF の基本的な端末の操作方法

操作	内容
get-config	設定内容の取得
get	設定内容の取得 (running のみ)
edit-config	設定内容の変更
copy-config	設定のコピー
delete-config	設定の削除 (running は削除不可)
lock	設定に対する変更を禁止
unlock	lock された設定の変更禁止の解除
close-session	セッションの終了
kill-session	セッションの強制終了

設定は用途別に、端末が起動した際に読み込まれる startup, 現在動作している running, 端末の設定に影響しない candidate の 3 種類がある。これらの設定に対する NETCONF プロトコルの基本的な操作は表 1 に示す通りである。

各設定の構造については、YANG[2]により定義される。YANG とは NETCONF で操作する設定のデータ構造や操作方法などが定義されたデータモデリング言語である。YANG により定義されたデータ構造を YANG モジュールと呼び、機能ごとに複数の YANG モジュールが存在する。新たな YANG モジュール内で、すでにある YANG モジュールをインポートし機能を付け加えることが可能である。NETCONF で交換する設定は YANG で定義された構造の XML となっており、標準的な YANG モジュールに従うことで異なるベンダーにおいても統一された設定が可能となる。

3. シミュレータと連携する NETCONF 検証環境

本研究では、大規模ネットワークの NETCONF による設定の検証のために、実機のテスト環境でなくシミュレータを利用可能とする。この際、シミュレータによる検証が完了した設定が、そのまま実環境に投入可能であることが望ましい。

図 1 にシミュレータと実ネットワークの両方で検証・設定可能な設定構築環境の構成を示す。

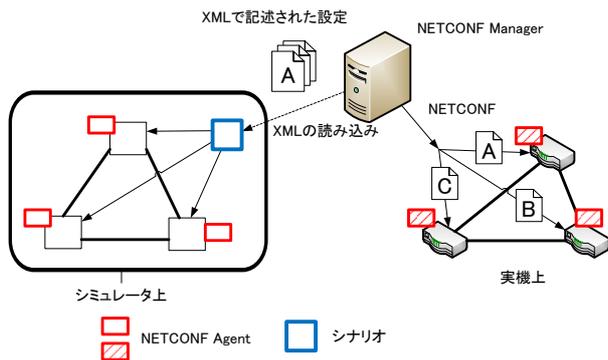


図 1 NETCONF の動作の概略図

NETCONF Manager は管理者側の端末となり，NETCONF Agent は XML で記述された設定からノードの設定の変更を行う．シミュレータ上では，ネットワークの構成とシミュレーションの手順を記述したシナリオの処理により XML で記述された設定を読み込み，各シミュレータのノードに設定をする．実機上では各ノードにセッションを張り設定の適用を行う．

4. 実装

実装するシミュレーション環境として ns-3 と呼ばれるネットワークシミュレータを用いる．ns-3 上でシミュレーション用のネットワーク機器を用意する場合には，物理的なネットワーク機器に相当するノードを生成し，そのノードに対して通信デバイスの機能や通信プロトコルの機能をシミュレートする ns-3 モジュールを追加することで所望のネットワーク機器を動作させる．ns-3 のモジュールの実装では，ノードにモジュールを追加する際に同時にデバイスやプロトコルのパラメータも設定できるようになっている．これに対して，NETCONF の挙動をシミュレートするためには，デバイスやプロトコルを実装したノードを用意して，後から NETCONF でパラメータを設定できるようにする必要がある．

ns-3 では，ns-3 モジュール内にパラメータが定義されているが，モジュールの使用時には直接 ns-3 モジュールのインスタンスを生成するのではなく，ヘルパークラスによりモジュールインスタンスの生成とパラメータの設定を行うようになっている．そこで，YANG の定義毎に該当する ns-3 モジュールのパラメータを変更する関数を用意し，その対応を管理することで NETCONF によるパラメータ設定の処理を行う．この実装について，PointToPoint モジュールの通信帯域と IP アドレスの例を用いて説明する．

PointToPoint モジュールの通信帯域は，PointToPointNetDevice クラス内に定義されている．このクラスはネットワークレイヤのデバイスイ

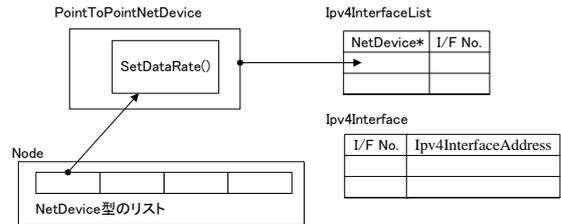


図 2 NetDevice 型のパラメータ変更

ンタフェースを抽象化した NetDevice クラスを継承している．各ノードは NetDevice 型のポインタのリストを保持しており，ノードに通信線を追加する際には PointToPointNetDevice のインスタンスへのポインタを格納する(図 2)．帯域を設定するには，PointToPointNetDevice クラスに定義されている SetDataRate(DataRate)というメンバ関数を呼び出す．NETCONF で帯域を設定する指示を受け取ったら，ノードの保持するリストから対象となるネットワークインタフェースに相当する NetDevice のインスタンスへのポインタを特定し，PointToPointNetDevice にダウンキャストをし，このメンバ関数を呼び出すことでシミュレータ上の通信帯域の設定を実現することができる．

ネットワークインタフェースに対して IPv4 の機能を適用する際には，該当する NetDevice に対して IPv4 インタフェース番号が割り当てられる．このインタフェース番号と IP アドレス情報を保持する Ipv4InterfaceAddress クラスのインスタンスの対応が管理されているので，IP アドレスを設定するインタフェースから対応する NetDevice が特定され，それに割り当てられているインタフェース番号から IP アドレスを管理するインスタンスを辿ることができる．NETCONF でアドレス設定の指示が来たらこのインスタンスに対してアドレス設定のメンバ関数を呼び出す．

5. まとめ

NETCONF による動的なネットワーク構成をシミュレート可能な検証環境を実現した．具体的には，NETCONF の機能を ns-3 の Agent として実装することにより，ns-3 のノードを NETCONF で設定可能となり，その設定は NETCONF に対応した実機環境にもそのまま適用する事ができる．

参考文献

- [1] RFC 6241 Network Configuration Protocol (NETCONF), <https://tools.ietf.org/html/rfc6241>
- [2] YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF), <https://tools.ietf.org/html/rfc6020>