

GINE を用いた大規模ネットワークエミュレーションと管理機能の追加

浅野 洋介†

南山大学大学院 数理情報研究科†

1 はじめに

広域ネットワークで使用するアプリケーションを性能評価するためにさまざまなネットワークエミュレータが存在している。オープンソースで end-to-end の通信を模倣できるエミュレータは Nistnet である。これはリンクエミュレーションに対応しているが、ネットワークアドレスのプレフィックスを指定することができない。また NCTUns[4] は GUI も実装されているが、独自カーネルが必要である。さらに FreeBSD の IMUNES は、ホストやルータを模倣することができるが、独自拡張カーネルの再構築が必要である。そして南山大学の後藤邦夫教授が考案した Goto's IP Network Emulator(以下, GINE) [3] は、多数のルータやリンクで構成される広域ネットワークを模倣することができるネットワークエミュレータである。各リンク毎に与えられた確率分布に従って、遅延やパケットロスなどの通信障害を容易にエミュレートすることができ、バンド幅を独自に与えることができる。

本研究では、新機能として GUI を用いたネットワークトポロジの読出/保存機能や実行時の操作などの管理機能を GINE に実装する。そして動的ルーティングプロトコルを用いた広域ネットワークトポロジを構築する方法を提案する。

2 ソフトウェアアーキテクチャ

本節では GINE のシステム構成を説明する。GINE でのエミュレーションはカーネルレベルではなく、ユーザプロセスレベルで実現されている。そしてシステムコールを多用するため C++ とマルチスレッドを使用するために GNU common C++[1] の Thread クラスを用いて記述されている。

2.1 リンクエミュレーション

リンクエミュレーションとはホスト間をつなぐリンクを表現し、パケット遅延や損失を発生させることができる機能である。これは NistNet で使用できる機能であるが、NistNet では IPv6 に対応していない。GINE では IPv4/v6 アドレス、プロトコル、ポートのフィルタリングが可能で、プロトコル、行き復りのリンクごとに違うパラメータの障害 (フレーム遅延/損失、回線容量の制限など) を発生させることができるリンクを表現する。またこのリンクは一般的なデータリンク層のフレームバッファとして使用する独自 Queue プログラムによって実現している。

2.2 ホスト/ルータエミュレーション

FreeBSD の IMUNES にはホストやルータを模倣する機能がある。GINE でもこの機能を使用する方法と

してネットワークスタックの仮想化を使用している。これはルーティングやフォワーディング機能を含めたネットワーク部分を仮想化し、ネットワークホストを構築する方法である。この機能は Linux 2.6.26 以降から Network Namespace(以下, NETNS)[2] として標準搭載されており、カーネル修正は不要であるため容易に使用することができる。この機能を GINE に導入することで、実ホスト、実ルータ等の機能をより容易に実装できるようになった。図 1 に GINE によって構築できるネットワークトポロジ例を示す。

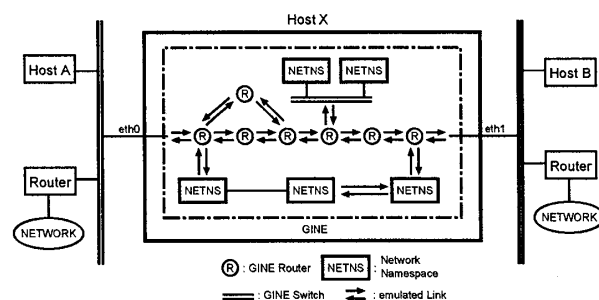


図 1: ネットワークトポロジ

NETNS にルータ/ホストの機能を持たせることでより現実的なネットワーク空間を構築できるようになる。また仮想ルータ (GINE Router), 仮想スイッチングハブ (GINE Switch) などをプログラムによって生成し、NETNS と組み合わせることでさまざまなネットワークトポロジを構成することができる。

2.3 改良点の概要

後藤研究室において GINE の改良や拡張が進められてきた。しかし機能が増えてきたことで管理する機能が必要になってきた。本研究では次の 3 点について改良と機能を追加する。

1. NETNS の自動生成と端末を用いた管理
2. オブジェクトの保存/読出機能
3. GUI 機能

3 システムの実現

2.3 節を実現する方法について説明する。

3.1 仮想ネットワークスタックの自動生成と端末管理

仮想ネットワークスタック (NETNS) を用いるとき、NETNS ごとにネットワークインタフェース設定等のコマンド実行が必要となる。しかし NETNS 数を増やすと、同数の端末を起動させると無駄なメモリ消費量が増えエンドユーザの操作も困難になるため、NETNS を GINE 内で管理しバックグラウンドで処理する機能が必要である。

これを改善するために、fork 関数を用いてプロセスを分岐させ、擬似端末で親プロセスと子プロセス (NETNS) をつなぎ、親プロセスから子プロセスにコ

Evaluation of GINE's new Functions for Large-scale Network Emulation

† Yosuke ASANO

Graduate School of Mathematical Sciences and Information Engineering, Nanzan University

マンド文字列を渡して実行することで実現した。しかし一部の NETNS に対しては、IP アドレスの付与等のマニュアル操作も必要であるため、必要時に端末を開閉する機能も実現した。端末には xterm を使用し、作成済の擬似端末のスレーブ側を接続する。

3.2 オブジェクト保存/読出機能の実現

さまざまなアプリケーションには任意のデータをファイルへ保存、ファイルから読み出しする機能が必要であることが多い。特に動的に変化するモデルや複数の型を持つオブジェクトなどを生成するためには、オブジェクトの状態等を保存し復元する (オブジェクトの永続化) 機能が必要である。

本研究では GNU common C++ の Persistent 機能 [1] を用いる。この機能を実現するためには保存したいクラス全てで、Persistence を生成するためにクラスの型を定義する BaseObject クラスを継承する必要がある。加えてクラス内で宣言された変数等を保存/読み出すために write, read メソッドを定義する必要がある。そのため BaseObject クラスを継承する GineBaseObject クラスを作成し、これを Persistence を実現したいクラスで継承させることにした。

この機能を用いることで保存/読み出しすべき情報は参照したオブジェクトも重複することなく処理できることがわかった。GINE で生成したオブジェクトはほとんどが双方向リスト形式で表現されているため、先頭オブジェクトのみを指定すれば末尾オブジェクトまで保存することができる。

3.3 GUI 機能の実現

現行 GINE の一般的な使用法は、ネットワークポロジごとに C++ で用意したライブラリクラスを使ってメインプログラムを記述し、それをコンパイルしてコマンドベースで実行している。しかしこの方法では、どのようなネットワークポロジで動作しているのかプログラムを確認しないとわからなかったり、プログラム自体を書くことができないユーザにはエミュレータの利用自体が困難である。

本研究では GUI 作成ツール Qt を用いて GINE のライブラリを操作する GUI ツールを作成し、GINE を容易に操作できるようにする。Qt を用いて本研究で実現したい GUI 機能は次の 4 つである。

1. GINE クラスで生成するオブジェクトをボタンとして配置する。
2. オブジェクトボタンを描画フィールドにドラッグ & ドロップで配置する。
3. 各オブジェクトボタンをダブルクリックすると別ウィンドウが開き、詳細な情報を設定する。次のような例を実現する。
 - FrameQueue ボタン: 遅延, ロスなどの値を入力するなど
 - NETNS ボタン: 3.1 節で作成した xterm を開くなど
4. スタートボタンをクリックしたとき、スレッドがスタートして通信を開始する。またポーズボタンやストップボタンを用意し、時間経過に沿ったシナリオを実現する。

5. ファイルメニューから 3.2 節のオブジェクト保存/読出機能呼び出す。

作成した GUI アプリケーションを図 2 に示す。

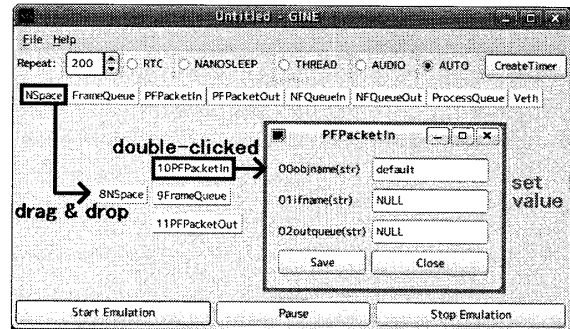


図 2: GUI アプリケーション

このアプリケーションによってプログラムを書くことができないユーザも使用できるようになった。

4 大規模ネットワーク構築方法の提案

NETNS を使用すると、動的ルーティングデーモンを起動させることができる。本研究では大規模ネットワークを構築することを想定し、RIP, BGP といったプロトコルをサポートし、オープンソースで公開されている Quagga を使用する方法を紹介する。

1 台のホスト上で大規模なネットワークを構築するためには、GINE で起動した複数の NETNS 上でそれぞれ Quagga を動作させ、経路交換させる必要がある。しかし Quagga のデーモンは NETNS の生成時には仮想化されない。そこで設定ファイルとプロセスファイルをそれぞれ NETNS ごとに作成し、NETNS 上でそれら指定してデーモンを起動することで、同時にデーモンを使用できる。

5 おわりに

本研究では、ネットワークエミュレータに新機能を搭載することができた。しかし GUI アプリケーションに関しては最小限の機能までの搭載に留まった。

今後の課題として、大規模なネットワークを構築できるように 3.3 節で作成した GUI アプリケーションをさらに改良する必要がある。

参考文献

- [1] Free Software Foundation: *Gnu common C++*, <http://www.gnu.org/software/commoncpp/>.
- [2] Network Namespace: *Linux Containers*, <http://lxc.sourceforge.net/network.php>.
- [3] Sugiyama, Y. and Goto, K.: Design and Implementation of a Network Emulator using Virtual Network Stack, *Proc. of the Seventh International Symposium on Operations Research and Its Applications, Lecture Notes in Operations Research, Vol.8*, pp.351–358 (2008).
- [4] Wang, S., and Chou, C.: The Design and Implementation of the NCTUns Network Simulation Engine, *Elsevier Simulation Modelling Practice and Theory*, Amsterdam, Nederlanden, pp.57–81 (2007).