プラットフォーム独立な高速ネットワーク機器の性能評価ツールの開発

山之上 卓, 戸田哲也

九州工業大学情報科学センター

概要

ギガビットスイッチなどの高速ネットワーク機器の性能評価を行うプラットフォーム独立なツールの開発について述べる．このシステムは，対象となるネットワーク機器に接続された多数のコンピュータ(ノード)を一個所から遠隔操作し，ノード間で一斉にデータを流すことによって，ネットワーク機器の性能を評価するものである．Netperf などの既存のツールとの比較も行った．

Development of a Platform Independent Performance Measurement Tool for High Speed Network Equipment

Takashi Yamanoue, Tetsuya Toda

Information Science Center, Kyushu Institute of Technology

Abstract

Development of a platform independent tool, which evaluates the performance of high speed network equipment such like a gigabit switch, is shown. This tool evaluates the performance of the equipment by exchanging data between a large number of computers (node) simultaneously across the equipment. These nodes are controlled from a remote node. We also compare this tool with other network benchmark tools such like the Netperf.

1. はじめに

分散システムの構成要素であるネットワーク機器は分散システムの物理的および論理的な中心部分に位置しており，分散システムのなかでも重要なものの一つである．ネットワーク機器の性能そのもの性能を測定することができれば，分散システムの構築や設計に役立てることができる．近年，ネットワーク機器の高速化が進み，カタログ値で 1Gbps の性能を持ったスイッチ(SW)なども普及している． しかしながら，これらの機器が本当にそれだけの性能を発揮できるかどうかについて，従来は高価なネットワークテスタなどを使って計測する必要があり，エンドユーザ側で調査することは難しかった．

我々は高速ネットワーク機器に多数のコンピュータ(ノード)を接続し，そのノードを遠隔操作し，ノード間で一斉にデータ転送を行うことによってネットワーク機器の性能を評価するツールの開発している．このツールを使って無線 LAN とギガビット SW の性能測定を行った．またネットワークベンチマークテストとして標準的に使われている netperf[1]との比較も行った．

本ツールは Java で開発されており，様々なプラットフォームで利用可能である． このツールは，プラットフォーム独立な分散システムの評価システム[3]のアプリケーションの一つである．

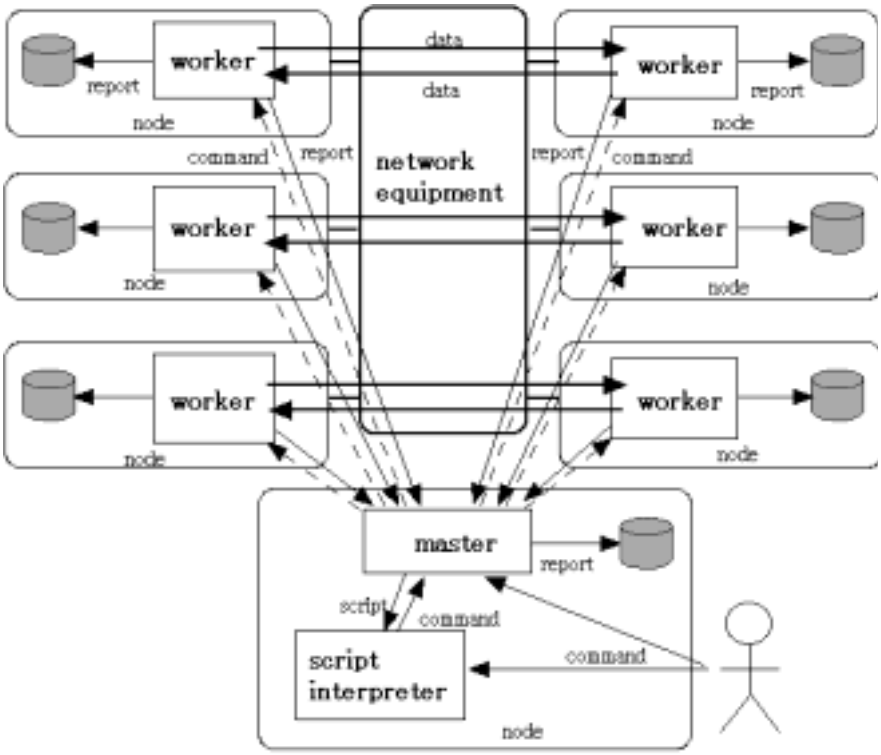

図 1　高速ネットワーク機器の性能評価ツールの概要

本稿では２章で本性能評価ツールの概要，３章で測定実験，４章で関連研究との比較を行った後，５章でまとめと今後の課題などについて述べる．

2. 性能評価ツールの概要

図 1に本性能評価ツールの概要を示す. 評価の対象となるネットワーク装置(network equipment)にコンピュータ（node）を接続し，その node の上で worker プログラムを動作させる．worker プログラムは master プログラムから指示(command)を受けることによって，ネットワーク装置をまたいだ worker プログラム間のデータ交換を行い，このときの転送時間などの測定結果を各 node で記録するとともに，master プログラムにも測定結果の一部を送る．command 列を script interpreter によって自動的に実行させることにより，あらかじめ定めた手順で各 worker プログラムに様ざまな動作をさせることができる．多数の node 間でデータ交換を一斉に開始することにより，対象のネットワーク装置に大量のデータを流した場合の性能の計測を行うことができる．ギガビット SW などの高速 SW にビッグパイプを持った中速 SW を接続し, その中速 SW に多数の node を接続し，高速 SW をまたぐようにノード間でデータ交換させることによって，高速 SW が実際にどのくらいの性能を持つか，調べることができる．

worker プログラムも master プログラムも，分散システムの性能評価システム[3]の，アプリケーションの一つである．これらは一つの「グループ」に参加した「ノードシステム」上で動作し，グループ内で遠隔操作や情報交換が行われる．

以下，2.1 で worker プログラム，2.2 で master プログラム，2.3 で script interpreter について説明する．

2.1 worker プログラム

worker プログラムは，master プログラム から command を受け取り，それを実行することによって，他の worker プログラムへの TCP 接続や，接続した worker 間のデータ交換などを行う．TCP 接続時に要した時間や，データ交換に要した時間などは, csv 形式で記録される．現時点ではまだ udp は扱えない．

図 2に worker プログラムの GUI を示す.

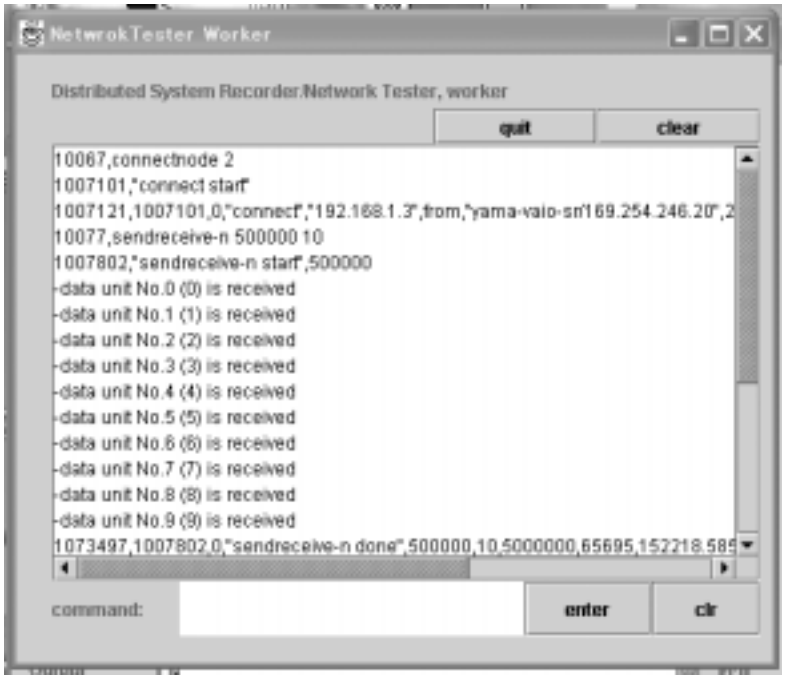

図 2　worker プログラムの GUI

「quit」ボタンをクリックすると worker プログラムを終了する．「clear」ボタンをクリックすると，中心の大きな矩形の「出力領域」の内容が消去される．ここには入力されたコマンドとその実行結果が表示される．下の「command」ラベル右の小さな矩形領域が command 入力領域である．ここに master プログラムから送られた command が入力される．command は各 worker プログラムで直接入力することもできる．その右の「enter」ボタンをクリックすると command が実行される．このボタンのクリックも，master プログラムから遠隔で行われる．「clr」ボタンをクリックすると command 入力領域がクリアされる．worker プログラムで実行可能な command には以下のようなものがある．

- **connect <ホスト名>**
  ホスト名または IP アドレスで示されたノードの worker プログラムに TCP の接続を行う．各 worker プログラムは，TCP のサーバを兼ねている．接続に要した時間が測定される．
- **connectnode <ノード番号>**
  ノード番号で示されたノードの worker プログラムに TCP の接続を行う．接続に要した時間が測定される．
- **disconnect**
  どこかの worker プログラムと接続していた場合，その接続を切断する．
- **sendreceive-n <data-size> <times>**
  接続先の worker プログラムに<data-size>バイトのデータを送信し，echo back されるデータを受信する．これを<times>回繰り返す．<data-size>のデータが往復するたびに，それに要した時間が計測されると共に，それぞれの片道平均の伝送速度と<times>回の往復における平均速度が測定される．全二重の伝送路でデータの往復が行われた場合，片道平均の伝送速度がネットワークの規格速度を超える場合がある．
- **set <command>**
  command 入力領域に<command>を入力する．
- **enter**
  「enter」ボタンをクリックする．master プログラムが set コマンドを使って，異なるノードの worker プログラムの command 入力領域に，異なるコマンドを入力し，すべての worker プログラムで一斉に enter コマンドを実行することによって，各 worker プログラムで異なる動作を同時にさせることが可能である．

図 2の出力領域に表示されている「1067, connectnode 2」は，記録が開始されてから

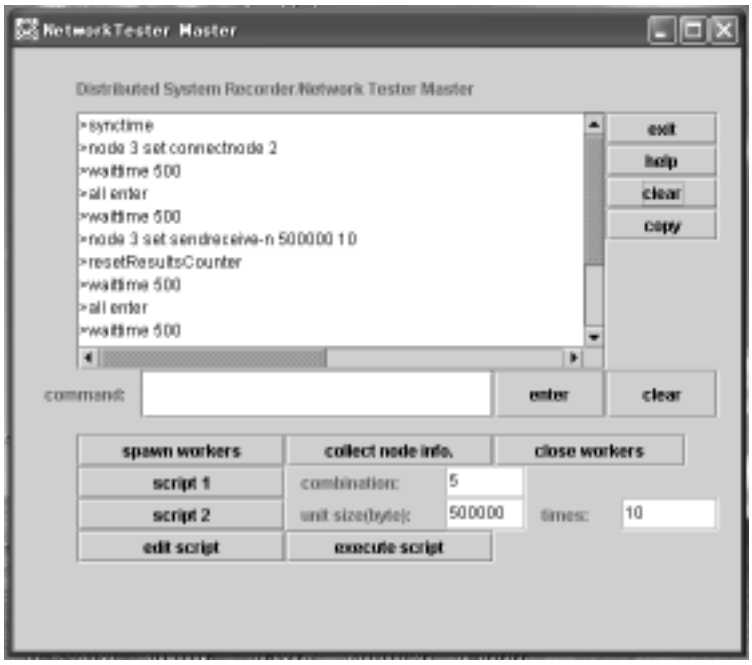

図 3　master プログラムの GUI

1006700ms 後の時点で，２番のノードへの接続が指示されたことを表している．「1007121, 1007101, ”connect”, “192.168.1.3”, from, ”yama-vaio-sr…”, …」は，1007101ms に接続が開始され，1007121ms で接続が完了し，接続先が 192.168.1.3，接続元が yama-vaio-sr であることなどを表している．「10077，sendreceive-n，500000，10」は，1007700ms で，接続先ノードとの間で 500KB のデータを 10 回送受信する指示があったことを表している．「1007802，”sendreceive-n start”, 500000」は，1007802ms の時点で，送受信が開始されたことを表している．「1073497，1007802，“sendreceive-n done”, …」は, 1073497ms の時点で sendreceive-n コマンドの実行が終了したことなどを表している．実行開始が 1007802ms なので，このコマンドによるデータ転送に 65695ms を要したことがわかる．この時間や片道平均伝送速度などもこの行に記述される．また，この情報は，master プログラムにも送信される． この行の後に，このコマンドで実行されたデータ転送の各回における転送時間や片道平均伝送速度などが表示される．

2.2 master プログラム

master プログラムは，worker プログラム に command を送信することによって worker プログラムを遠隔操作する．図 3に master プログラムの GUI を示す．GUI 右上の「spawn workers」ボタンをクリックするとグループに参加しているノードシステムで worker プログラムが起動される．「collect node info」ボタンをクリックすると，worker プログラムが動作しているノードの番号やホスト名などの情報が master プログラムに収集される．「script2」ボタンをクリックすると，収集された情報を元に，起動された worker 間で一斉にデータ交換を行うコマンド列(script)が自動生成され，これが script interpreter に送られる．「edit script」ボタンをクリックすると，script interpreter の GUI が表示される，この上で script を修正することができる．「execute script」ボタンを押すと，script の実行が開始される．master プログラムで実行可能な command には以下のようなものがある．

- **synctime**
  起動されているすべての Worker プログラムの時間を master プログラムの時間に合わせる．
- **all <worker command>**
  起動されているすべての worker プログラムで<worker command>を実行する．
- **node <node-id> <worker command>**
  ノード番号が<node-id>の worker プログラ

ムで<worker command を実行する.

- **waittime <msec>**
  <msec>時間停止する.
- **waitallresults <number>**
  <number>で指示した数のノードから結果が返ってくるまで停止する.

2.3 script interpreter

script interpreter はここに記述された手順(script)に従って master プログラムのコマンドを実行するものである. Master プログラムで生成された script を取り込むこともできる. この interpreter は分散システムの評価システムのプログラミング環境をそのまま使っている.

3. 測定実験

3.1 無線 LAN の性能測定実験

２台の PC を IEEE802.11b の無線 LAN を使って接続し, この間のデータ転送速度を本ツールを使って計測した. 図 4に実験環境の概略を示す.

PC2 から PC1 に TCP の接続を行い, 1MByte のデータを PC2 と PC1 の間で 10 回往復させ, それらに要した時間や, 片道平均の転送速度を計測した. 接続に要した時間は 10ms, 10 回のデータ転送に要した時間は 40.5 秒, 平均片道転送速度は 3.95Mbps, 最大転送速度は 3.99Mbps, 最小転送速度は 3.8Mbps, 標準偏差は 0.05Mbps であった. この無線 LAN の規格値である 11Mbps の半分も速度が出ていない. このツールはデータの往きと帰りの転送を同時に行なおうとするため, 衝突が多数発生した可能性がある. また無線 LAN の性能は使用環境に大きく影響されることも原因として考えられる.

3.2 ギガビット SW 性能測定実験

ギガビット SW の２つのポートに 1G - 100Mbps スイッチを２台接続し, それぞれの 1G-100Mbps スイッチの 100Mbps ポートにパソコンを接続して本性能評価ツールを動作させることにより, この 1G スイッチの２ポート間のデータ転送性能測定を行った. 各スイッチは全二重通信を設定した. スイッチやパソコンは文献[4]のものを使った.

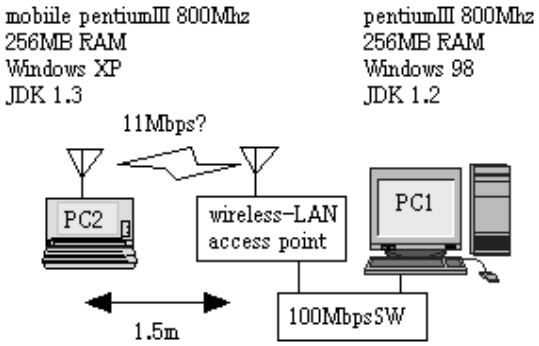

図 4 無線 LAN 性能測定実験

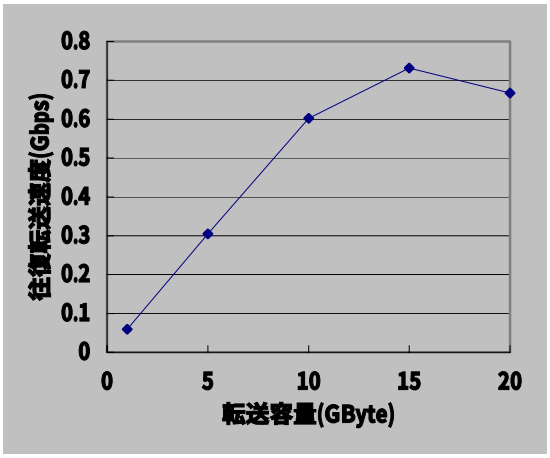

図 5 ギガ SW の性能計測

この実験は, 1Gbps スイッチをまたいで 1, 5, 10, 15, 20 対のノード間で一斉に 1Mbyte のデータを 1000 回往復させて性能計測を行った. １対の場合は 1GByte, 10 対の場合は 10Gbyte のデータがノード対の間で往復することになる. この容量を, それぞれの平均転送時間で割って, ギガビット SW のポート間の転送速度とした. この計測をそれぞれ２回ずつ行って平均を取った. 図 5に実験結果をグラフで示す. ここで横軸はギガビット SW のポート間を往復したデータ容量, 縦軸は, このときの転送速度である.

15GByte の転送時に転送速度が最大となり, 731.5Mbps であった. この実験時に使用したギガビット SW のモニターでは, 最大 785Mbps の値が表示されていた.

20GByte の転送時, １回目は２０対のうち, ３対が途中で固まり, ２回目は５対が固まった. ここでは最後までデータ転送を行った対の転

送速度を, 20Gbyte の転送を行ったときの転送時間とした.

同じネットワーク機器上で, netperf で 10 対のノード間で一斉に一方向のみのデータ転送を行った場合の通信速度を計測したところ 986Mbps となった. 20 対のノードを使って, 同時に 10 対の双方向転送を行った場合, 通信速度は 780Mbps から 800Mbps であった. netperf を一斉に動かすために ssh やシェルスクリプトなどを利用した.

MRTG でギガビット SW の転送速度を表示しようとしたが, ギガビット SW のモニター表示より一桁少ない値しか表示されなかった. SMTP によるパケット取得がうまくいっていない可能性がある.

本ツールで計測した通信速度の値が netperf やスイッチのモニター表示より若干低めなのは, Java VM の性能の影響などが考えられる.

4. 関連研究

ネットワークの測定ツールとして, netperf[1], DBS(Distributed Benchmark System)[2], P2P モデルを用いたツール[5]などがある.

netperf は基本的には１対１で一方向の通信の性能測定を行うツールである. またプラットフォーム独立ではない.

DBS は複数のホスト間で同時に複数のデータ転送を行うことができると同時に, 各々のデータ転送のスループットの時間変化も測定できる. また, TCP コントロールブロックの値を記録できるなどきめ細かな測定が可能である. しかしながら現時点で DBS は, UNIX 系の OS 上でしか動作しない.

P2P モデルを用いたツール[5]は, ネットワークの構成を推定したりネットワークの論理的なつながりの変化を検地したりすることも可能である. しかしながらこのツールもプラットフォーム独立ではない.

本ツールは現時点では UDP のデータ転送や, 片方向のみのデータ転送時の計測も行えない. DBS ほどきめこまかな測定も行うことはできない. しかしながらスクリプトに従って複数のホストを同時に制御することなどについては, DBS とほぼ同様な機能を持っている. また,本ツールはプラットフォーム独立である. 本ツールの出力は CSV 形式で行われるため, excel など, 既存の表計算プログラムを使って比較的簡単にデータ整理や解析を行うこともできる.

5. おわりに

プラットフォーム独立な高速ネットワーク機器の性能評価ツールの開発について報告した. 今回は, ネットワーク上で行うことができる通信の, ごく一部の組み合わせの測定しか行っていない. 今後はもっと多くの組み合わせによって測定実験をおこない, より精度の高いネットワーク機器の評価を行っていく必要がある. また, 一方向のデータ転送を可能にしたり, UDP の計測を可能にしたりできるよう, 本ツールを拡張する必要もある.

参考文献

[1] Hewlett-Packard Company,”Netperf: A Network Performance Benchmark Revision 2.1”,1995.

[2] Yukio Maruyama, Suguru Yamaguchi, "DBS: a powerful tool for TCP performance evaluations”, Performance and Control of Network Systems, Proceedings of SPIE, Volume 3231, November 1997.

[3] 山之上卓, 望月雅光, 中山仁, 大西淑雅,甲斐郷子,"分散システムとインターネットアプリケーションの性能評価システム", 情報処理学会分散システム/インターネット運用技術研究会報告情報処理学会研究報告, 20-3, pp.55-60, 2000.

[4] 中山仁, 大西淑雅, 望月雅光, 山之上卓, 甲斐郷子, ”Linux thin client を端末とする集合教育用計算機環境の構築”, 情報処理学会分散システム/インターネット運用技術研究会報告, 2000-DSM-18, pp. 31-36, 2000.

[5] 豊野剛, 仲山昌宏,杉浦一徳, ”P2P モデルを用いたエンドノード間トラフィック測定”, 情報処理学会分散システム/インターネット運用技術シンポジウム 2002 論文集,情報処理学会シンポジウムシリーズ, Vol.2002, No.5, pp.39-44, 2002.