広島工業大学紀要研究編
第44巻（2010）pp.223-227
論　　文

ネットワークトラフィックの可視化による通信データの監視および制御に関する研究

長坂　康史*・福田　宏見**

（平成21年10月31日受理）

A Study on Monitoring and Controlling of Communication Data by Network Traffic Visualization

Yasushi NAGASAKA and Hiromi FUKUDA

(Received Oct. 31, 2009)

Abstract

The Internet has become one of the most important tools. A huge amount of data is communicated in the Internet all over the world. The management of data traffic is very important under such a situation, because the computer transmits lots of unnecessary data though it is infected by a computer virus and the network is not well managed. A monitor of data traffic, therefore, should be required for recognizing such a situation.

This paper describes the monitoring and control system of communication data. The system consists of an agent and a manager. The agent monitors the communication data in his own computer and reports the result to the manager. On the other hand, the manager receives and visualizes it to find abnormal data traffics with several kinds of graphs. The system also has the functionality to restrict the specified data transmission. The proposed system makes it possible that an event and a bottleneck of network system can be found and solved easily.

Key Words: network traffic, traffic control, traffic visualization

1．はじめに

昨今のインターネットの普及により，企業や学校などに設置されるほとんどのコンピュータにインターネットを利用するための環境が整備されており，コンピュータを利用するユーザは，誰でも簡単にインターネットを利用することができる。

しかし，ネットワークシステムの管理が不十分であると，コンピュータウィルスへの感染や致命的な障害が発生する可能性が高くなり，インターネットを快適に利用することができなくなる。コンピュータウィルスに感染すると，不要なデータを大量に送信し，そのデータの宛先コンピュータに負荷を掛けるだけでなく，１つのコンピュータがネットワークを占有し，ネットワーク全体のパフォーマンスを低下させる可能性がある。また，１つのコンピュータがコンピュータウィルスに感染すると，ネットワークを介して他のコンピュータにも感染が広まり，システム全体に大きな障害をもたらすこともある。そして，その感染の拡大はとても早いため，即座に対応する必要がある。

また，データ収集システムなどにおいてもシステムの管理は重要である。データ収集システムでは，それぞれのコンピュータにおいて様々なデータがやりとりされる。この

* 広島工業大学情報学部情報工学科
** 中国管区警察局

各コンピュータの通信データ量を監視することによって，システムのボトルネックや障害をいち早く発見することができ，システムの運用を支援することができるだろう[１]。

現在，コンピュータの通信データを監視する代表的な方法として，ネットワーク上を流れるパケットをキャプチャし，そのパケットに含まれる各種プロトコルヘッダの情報や通信データ量などを調べ，どのような通信が行われているか分析するというものがある。しかし，ネットワーク上を流れるパケットから得られる情報をシステムの管理に役立てるためには大きな労力を必要とする。

そこで本研究では，パケットキャプチャによって得られる情報を基にネットワークトラフィックの可視化を行い，さらに，ネットワークに接続されている各コンピュータの通信データを監視・制御するシステムの開発を行い，ネットワークシステムの管理を効率化することを目的とする。

２．開発システム

2.1　概要

本システムは，ネットワークに接続された複数のコンピュータの情報を１つの管理用コンピュータが収集し，管理用コンピュータにおいて，各監視対象コンピュータの情報を可視化して表示する。また，管理用コンピュータでは各監視対象コンピュータの通信データを制御する。図１に本システムの構成例の概要を示す。

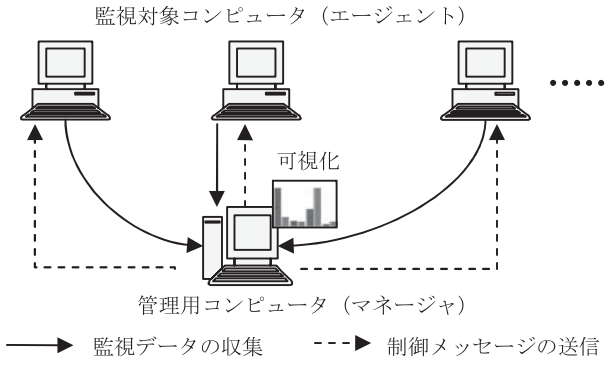

図１　システムの概要

図１に示すように，本システムは，エージェントとマネージャで構成される。これらの詳細な説明を以下に示す。

⑴　エージェント

エージェントは，監視対象コンピュータそれぞれに配置されるソフトウェアである。監視対象コンピュータにおいて送受信されたパケットをキャプチャし，それによって得られた情報をマネージャに送信する。このキャプチャデータの送信は，指定された一定間隔ごとに行い，その間に監視対象コンピュータにおいて送受信されたパケットの情報をマネージャに渡す。また，マネージャから送られてくる制御メッセージを受信すると，ファイアウォール機能によって通信データの制御を行なう。さらに，ファイアウォールによって破棄されたパケットの情報をマネージャに送信する。この情報もキャプチャデータと同様に，指定された一定間隔ごとに送信し，その間に破棄されたパケットの情報をマネージャに渡す。このエージェントは C++ 言語で開発した。

⑵　マネージャ

マネージャは，管理用コンピュータに配置されるソフトウェアである。エージェントから送信されたキャプチャデータおよびファイアウォールによって破棄されたパケットの情報を可視化して表示する。また，監視・制御のための GUI（Graphical User Interface）を提供する。さらに，監視対象コンピュータの通信データを制御するために，制御メッセージをエージェントに送信する。このマネージャは Java 言語で開発した。

2.2　通信データの監視

本研究では，パケットキャプチャを用いて，コンピュータの通信データの監視を行なう。パケットキャプチャとは，NIC（Network Interface Card）に届いたパケットを採取することであり，NIC を通して送受信したパケットの内容を全て見ることができる。パケットにはプロトコルごとのヘッダが含まれており，このヘッダの種類やヘッダ内の IP アドレスやポート番号によって，通信相手やパケットの種類を分析することができる。また，パケットの長さや個数を見ることでトラフィック量を監視することができる。

本研究では，各コンピュータにおいてパケットキャプチャを行い，キャプチャしたパケット内のヘッダの種類によってパケットを以下の３種類に分類する。

⑻　TCP パケット

TCP または UDP ヘッダを含むパケット

⑹　IP パケット

IP ヘッダを含み，TCP および UDP ヘッダを含まないパケット

⑽　Other パケット

TCP，UDP および IP ヘッダを含まないパケット

上記のように，キャプチャされたパケットから Ethernet，IP，TCP および UDP ヘッダを順に読み込み，ヘッダ内の情報からパケットの種類を判断する。TCP または UDP ヘッダを正常に読み込めた場合には TCP パケットに分類し，それ以外のパケットで IP ヘッダを正常に読み込めた場合には IP パケットに，IP ヘッダを正常に読み込めない場合には Other パケットに分類する。

また，キャプチャしたパケットから表１に示す情報を抜き出し保持する。パケットの種類によってはヘッダ内にこれらの情報が含まれていない場合があり，各パケットの種類ごとに保持する情報が異なる。表１に各種パケットごとに，キャプチャしたパケットから情報を抜き出し保持する場合には○，保持しない場合には×として示している。また，本研究ではキャプチャされたパケットがUDPヘッダを含む場合も上記のTCPパケットに分類する。しかし，UDPヘッダにはウィンドウサイズが存在しないので，キャプチャされたパケットがTCPヘッダを含む場合にはウィンドウサイズを保持するが，UDPヘッダを含む場合には保持しない。これを表１に△として示している。

表１　パケットキャプチャによって保持する情報

情報	TCP パケット	IP パケット	Other パケット
パケットの長さ［Byte］	○	○	○
パケットの個数	○	○	○
プロトコルの種類	○	○	×
送信元IPアドレス	○	○	×
宛先IPアドレス	○	○	×
送信元ポート番号	○	×	×
宛先ポート番号	○	×	×
ウィンドウサイズ	△	×	×

そして，表１に示した情報を特定のフォーマットの文字列に変換し，その文字列をマネージャに送信する。また，一定間隔でキャプチャデータを送信することによって，各コンピュータのトラフィック量（１秒間あたりに届いたパケットの量（bps：bit per second）や１秒間あたりに届いたパケットの個数）を監視することができる。

2.3　ネットワークトラフィックの可視化

パケットキャプチャによって得られる情報は文字や数字のみである。そこで，この情報を管理用コンピュータにおいて可視化し，より効率的に分析が行えるようにし，管理の効率化を図る。

本研究では，文字や数字で表される情報をグラフに変換することで可視化を行う。グラフの作成は，JFreeChartライブラリを用いた。JFreeChartは，Javaベースで開発されたグラフの作成に特化したライブラリである。

また，グラフの作成を効率良く行なうため，各コンピュータから収集したキャプチャデータ（表１に示した情報）および，それに加えて，キャプチャデータの送信元IPアドレスを用いて，送信・受信パケットの長さ（Byte），個数をそれぞれ保持する。さらに，様々なグラフに対応できるように，各コンピュータにおいて送受信された全パケットの長さ・個数の合計などを，表１に示した情報や送信・受信パケットの長さ・個数から算出する。

このように，パケットキャプチャによって得られる情報やそれを基に算出した各種の情報をグラフによって表示することでネットワークトラフィックの可視化を行い，効率的なシステム管理を支援する。

2.4　通信データの制御

本研究では，以下のような流れで通信データの制御を行なう。

管理用コンピュータにて，各監視対象コンピュータの情報が可視化されたグラフによって通信データの制御が必要かどうか判断する。制御が必要と判断された場合には，監視対象コンピュータへ向けてTCP通信を用いて特定の文字列を送信する。この文字列には，iptablesによって送信・受信を拒否するパケットの内容を指定する項目（どのコンピュータに向けたパケットを拒否するか，どのポート番号のパケットを拒否するか等）が含まれており，この項目を基に文字列を受け取ったコンピュータはiptablesコマンドを実行し，ファイアウォール機能を用いて通信データを制御する。

iptablesはLinuxにファイアウォール機能を実装するソフトウェアで，パケットフィルタリングを行なうことができる。また，送信元・宛先IPアドレスやポート番号を指定することで個々のパケットごとに送信・受信の許可・拒否を設定することができる。なお，パケットのフィルタリングはiptablesにルールを追加することで行なわれる。

本研究では，特定のパケットを破棄するルールをiptablesに追加することで通信データの制限を行なう。また，制限の解除はそのルールをiptablesから削除することで行なう。そして，iptablesによって破棄されたパケットの情報はファイルに出力され，このファイルから表１と同様の情報を読み取り，キャプチャデータと同様のフォーマットの文字列に変換してマネージャに送信する。

このように，ネットワークを介して特定の文字列を送信し，その文字列を基に受信側でiptablesコマンドを実行することで，特定のコンピュータの通信データを別のコンピュータから制御する。

2.5　マネージャGUI

本システムのマネージャは，各監視対象コンピュータの通信データを可視化したグラフや，2.4節に記述した制御メッセージを送信するためのGUIを提供する。図２にマネージャGUI画面を示す。

図２上部のコンボボックスとボタンは，2.4節で記述し

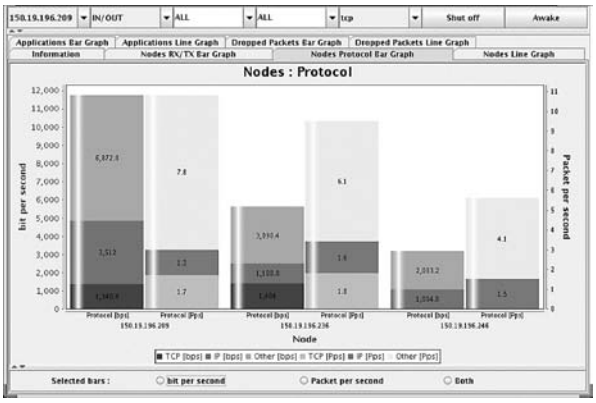

図２　マネージャ GUI 画面

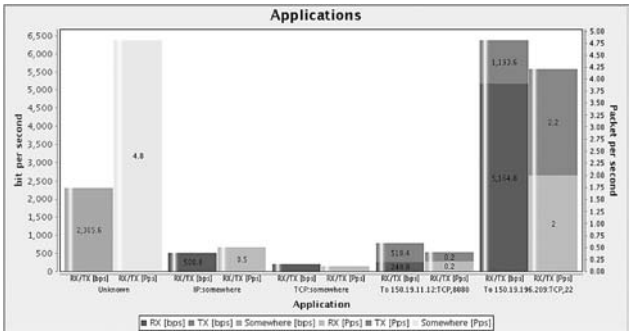

図３　１台のコンピュータにおけるアプリケーションごとの通信データ量を表すグラフ

た制御メッセージを送信するために用いる。

図２の中央部には，様々な角度から分析できるように複数種類のグラフが用意され，それらをタブメニューによって表示している。図２では，各コンピュータの通信データ量を表した棒グラフを表示している。この棒グラフの横軸には各コンピュータの IP アドレスを表示している。縦軸は目盛りが２種類あり，左側の目盛りが１秒間あたりに届いたパケットの量（bps：bit per second）に対応し，右側の目盛りが１秒間あたりに届いたパケットの個数（本研究では，これを Pps：Packet per second と表現する）に対応する。図２のグラフは，横軸の各項目に対して２つの Bar が表示されていて，それぞれ左側が bps のデータを，右側が Pps のデータを表している。このように，bps と Pps の２つの情報を表示することで，より詳細な分析が行えるようにしている。また，このグラフは 2.2 節のパケットの分類方法に基づいて，TCP パケット（TCP），IP パケット（IP），それ以外のパケット（Other）をそれぞれ色分けして表示している。

このグラフによって，どのコンピュータがどのようなデータをどれくらい送受信しているかを確認できる。また，それぞれのコンピュータの通信データ量の比較が容易に行なえる。

また，図２の中央部のタブメニューには合計７つのグラフが用意されており，図２に表示されているグラフの他に，１台のコンピュータにおけるアプリケーションごとの通信データ量を表すグラフなどがある。このグラフを図３に示す。なお，図３のグラフの縦軸は図２と同様である。

図３のグラフは，各通信データの相手コンピュータの IP アドレス，各通信データのプロトコル，ポート番号を横軸に表示させている。また，アプリケーションごとに送信データ（TX），受信データ（RX），送信か受信か判断できないデータ（Somewhere）をそれぞれ色分けして表示している。

上図のように各パケットの IP アドレスやポート番号などによってアプリケーションごとの通信データ量を可視化することで，各アプリケーションの通信データ量を監視し，それぞれを容易に比較することができる。

３．フレームワーク

本研究では，上述した監視，制御，可視化を行なうためのフレームワークをオブジェクト指向を用いて開発した。このフレームワークの概要を図４に示す。このフレームワークはクラスライブラリとして提供され，このクラスを用いることで上述の機能を実現することができる。また，キャプチャデータや制御メッセージの送受信は TCP 通信を用いて行なう。そのため，標準 C ライブラリの Socket 通信に関わる関数のラッパークラスを開発した[２]。そして，これらのフレームワークを用いて本システムの開発を行った。

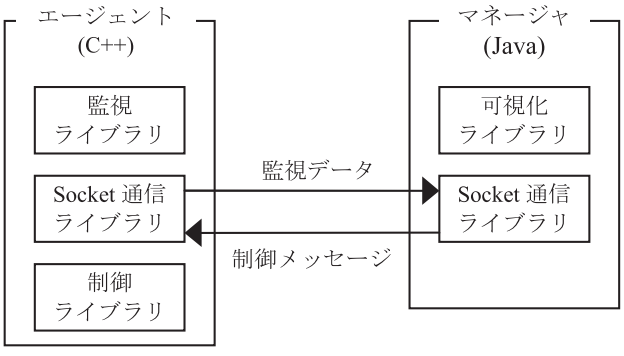

図４　フレームワーク

４．性能評価

本システムの評価を行うために，本システムを使用することでコンピュータにかかる負荷を測定した。CPU 使用率とメモリ使用量を測定し，負荷を調べた[３]。

なお，マネージャが動作するコンピュータの負荷を測定し，エージェントは２台とし，マネージャは２台のコンピュータの通信データを監視する。なお，エージェントか

らマネージャへ監視データを送信する時間間隔は 10 秒と設定している。表２にマネージャを配置したコンピュータの性能を示す。また，表３に測定結果を示す。表３の CPU 使用率については，１秒，５秒，15 秒間隔の CPU 使用率をそれぞれ 10 回求め，平均を算出した。

表２　マネージャを配置したコンピュータの性能

CPU	Intel Pentium 4　3.40GHz
メモリ	1,015,776kB
OS	Scientific Linux CERN release 3.0.8

表３　測定結果

	CPU 使用率（%）			フリーメモリ（kB）
	１秒平均	５秒平均	15 秒平均	
本システム実行前	0.2480	0.1098	0.0999	795,164
本システム実行後	0.8500	1.0483	0.7062	748,084

表３より，本システム実行前の CPU 使用率は最大約 0.2%であり，ほとんど CPU を使用していない状態であった。本システム実行前のフリーメモリは 795,164kB，実行後は 748,084kB であり，本システムの使用メモリは 47,080kB であることが分かる。本システム実行後の CPU 使用率は最大約１％であり，１秒平均，５秒平均，15 秒平均の各項目とも，本システム実行前と比べ増加しているが，１％程度である。また，使用メモリもトータルメモリの数%程度である。これらのことより，本システムを実行することによってコンピュータに高い負荷がかかるということはないと考えられる。

５．まとめ

本研究では，ネットワークシステムの効率的な管理を行なうために，ネットワークに接続されている各コンピュータの通信データを監視・制御するシステムを開発した。また，本システムを実行することによる，コンピュータにかかる負荷を測定した。その結果，本システムを実行することで，コンピュータの動作を阻害することはないと考える。

さらに，本システムでは，監視に用いる情報を単に数値データで表示するだけでなく，グラフなどに可視化することで，コンピュータがどのような通信を行なっているかということが一目で確認できるようになり，システム管理者の負担を軽減し円滑なシステム運用を支援することが可能である。今後は，より分かりやすく可視化を行なえるようにシステム・フレームワークの開発を進め，さらなるシステム管理の効率化を図る。

参考文献

［１］　福田宏見，長坂康史：「ATLAS 実験におけるデータ収集システムの状態可視化システムの開発」，FIT2008　第７回情報科学技術フォーラム，pp.169-170，2008

［２］　福田宏見：「モバイルエージェントを利用した効率的情報検索システムの開発」，広島工業大学，2006 年度卒業研究論文

［３］　藤井峰夫：「加速器科学仮想組織におけるグリッド環境自動監視・診断システム」，広島工業大学，2007 年度修士論文