3ZC-2

TCP/IP 可視化ツールの開発-アプリケーション層の可視化

柳瀬　貴博†　　荒井　正之†

帝京大学理工学部情報科学科†

1 はじめに

インターネットの普及に伴い、情報の専門教育において TCP/IP の教育がきわめて重要になってきた。しかし、従来の学習方法を用いた場合、決まった通信パターンの学習しか行えない、実感がわかないなどの理由で、抽象的性格の強い TCP/IP の概念を学習することが困難であった。

本研究の目的は、アプリケーションプロトコルに特化した可視化ツールを作成することである。このツールを用いて、プロトコルの実際の動作の理解を促すことができると考えている。なお、本ツールでは、アプリケーションプロトコルがやりとりするデータをメッセージと表記する。

2 ツールの概要

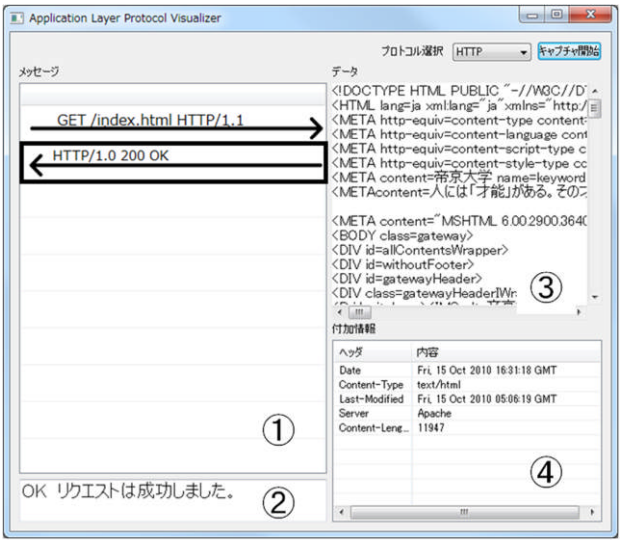

図 1. プログラム実行画面

図 1 は、本ツールの実行画面である。

画面は大きくわけて 4 つに分かれており、それぞれ、①「シーケンス図」②「メッセージの説明」③「やりとりしたデータの中身」④「付加情報」を表示する。

①では、メッセージのやりとりをシーケンス図として表す。また、任意のメッセージを選択し、実際に通信したデータや付加情報を表示する。図 1 では、サーバーからのレスポンス「HTTP/1.0 200 OK」が選択されている。

②では、①で選択されたメッセージの解説を表示する。①で選択されているメッセージは、サーバーへのリクエストが成功したというレスポンスなので、その説明である「OK リクエストは成功しました。」という説明文を②に表示している。

③では、実際に受信したデータの内容を表示する。図 1.①では、①で選択されたレスポンスで送られてきたデータ「index.html」の中身を表示している。表示される内容は、「テキスト」「静止画」の場合、ツール画面にそのまま表示される。「動画」「音声」の場合、ファイル名を表示し、外部メディアプレイヤーへのリンクを表示する。その他のバイナリデータについては、ファイル名の表示のみ行う。表示内容の選別は、メッセージの付加情報を使用する。

④では、メッセージの付加情報を表形式で表示する。図 1.④では、「送信日時」「コンテンツ形式」「最終更新日時」「使用サーバー」が表示されており、図 2 のようにヘッダ名にマウスカーソルを合わせると、ヘッダの解説が表示される。

図 2.付加情報欄に表示されるツールチップの例

3 実装

本ツールでは、Jpcap[1]というクラスライブラリを使用している。これは、Java 上でパケットキャプチャを実現するための、pcapAPI ラッパーである。Windows では、pcapAPI として WinPcap[2]ドライバを使用しており、Linux では lbpcap[3]を使用する。

3.1 データの取得方法と範囲

本ツールでキャプチャを開始すると、通信する全てのパケットを監視し、リモートホストの指定ポートへのアクセスを探索する。アクセスを検出した時、そのパケットの SYN フラグが true だった場合、そのホストの IP・ローカル IP・アクセス先ポートを元にフィルタリングを行いながら、通信パケットの中で必要なパケットを取得していく。取得したパケットの SYN フラグと FIN フラグの数をカウントしており、FIN フラグが SYN フラグと同じ数検出されてから一定時間後まで SYN フラグを検出しなかった場合、キャプチャを終了する。

本ツールでキャプチャされる通信データは、

TCP/IP Visualization Tools for Application Layer Protocols
†Takahiro YANASE, Masayuki ARAI, School of Science & Engineering, Teikyo University

図3のようなパケット単位で取得している。そのため、本ツールでは、取得したパケットからアプリケーションメッセージを再構築するための機能を備えている。例として、図3の太枠で囲まれたデータは分割されてしまったため、2パケットで1つのメッセージにする。この機能により、アプリケーションメッセージは図4のように復元される。図 4 は(a)メッセージ(b)付加情報(c)本体のデータが復元された例である。

IP Header	TCP Header	Application Message
192.168.0.5	59823	GET /index.html HTTP/1.0
192.168.0.1	80	
192.168.0.1	80	HTTP/1.0 200 OK Server: Apache
192.168.0.5	59823	
192.168.0.1	80	[DATA]
192.168.0.5	59823	

図3.取得されるパケットの一部
*[DATA]は分割されたデータの一部

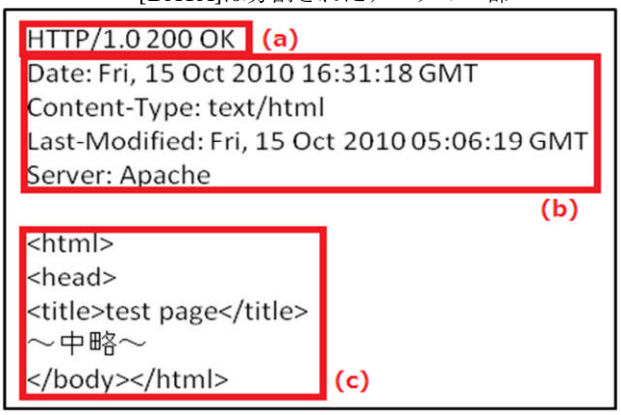

図4.復元されたアプリケーションメッセージの一例

3.2 各種プロトコルへの対応とデータ構造

アプリケーションプロトコルでは、プロトコルごとにリクエストコマンドやレスポンスコードが定義されており、また送受信するメッセージの構造はプロトコルごとに異なる。よって本ツールでは、取得したアプリケーションメッセージを解析するための定義データをプラグイン形式で追加・編集するための機能を備えている。

定義データは、図 5 のような XML ファイルにまとめられている。図 5.①には、受信するメッセージから必要なデータを取り出すための情報が入っている。図 5.①では、<split>に区切り文字の情報が入っており、<order>に参照するデータの場所が入っている。この例で図 4.(a)を処理すると、[HTTP/1.0][200][OK]の 3 つのトークンに分割され、このとき参照するデータが前から 2 番目、すなわち[200]ということになる。

②と③には、リクエストメソッド, レスポンスステータスの解説用データが入っている。④には、付加情報の解説用データが入っている。

```
<?xml version="1.0" encoding="shift_jis"?>
<protocol>
 <full_name>Hyper Text Transfer Protocol</full_name>
 <clipped_name>HTTP</clipped_name>
 <messagematching>                                          ①
  <split>¥s</split><order>2</order></messagematching>
～中略～
 <message>
  <requestArray>
   <request><method>GET</method>
    <explanation>指定されたURIのリソースを取得します。      ②
</explanation></request>
   <response><status>200</status>
   <explanation>OKリクエストは成功しました。               ③
</explanation></response>
～中略～
   <header>
    <name>Date</name>
    <explanation>送受信した日時</explanation>              ④
   </header>
  </addition>
</protocol>
```

図5.定義データの一例

3.3 可視化方法

まず、パケットキャプチャにより図3のようなデータを取得する。このデータを再構築することで、図4のようなアプリケーションメッセージを取得する。このアプリケーションメッセージを、図 5 の①を元に、「メッセージ」「付加情報」「データ本体」に分割する。メッセージは、図 5.②,③のデータを元に図 1.②に解説データを表示し、付加情報は、図 5.④のデータを元にツールチップに表示するデータを変更する。データ本体は、付加情報内の Content-Type をもとに図 1.③のように表示する。たとえば、図4の3行目の Contents-Type は text であるで、ファイルの中身を表示している。

4 おわりに

本研究は、パケットキャプチャを元にアプリケーションプロトコルのメッセージ交換時の通信手順とそのデータの可視化を行うツールを提案した。本ツールは、アプリケーションプロトコルによるメッセージ交換の可視化機能、通信したデータの表示機能を備えており、通信の仕組みを、実データを用いて理解するのに役立つと考えられる。今度の課題として、実授業におけるツールの評価がある。

5 参考文献

[1]Jpcap
http://netresearch.ics.uci.edu/kfujii/Jpcap/doc/
[2]WinPcap http://www.winpcap.org/
[3]tcpdump/libpcap Public repository
http://www.tcpdump.org/