ネットワークにおけるパケットアナライザの作成

(情報処理) 中村 友彦

１．序論

近年、コンピュータネットワークは大きく発展し、その管理が非常に重要になってきている。ネットワークを常に安定した状態で稼動させるためには、ネットワークを圧迫する不正なパケットを、常に監視する必要がある。本研究では、ネットワーク上を流れるパケットを全て捉え、そのヘッダ部分をファイルに蓄積し、それらを解析するアナライザプログラムを作成した。

２．開発環境

OS には Linux（カーネル 2.4）を採用し、プログラミング言語には C を用いた。パケット取得実験は、基本的に本学科コンピュータ室の mei サーバー上で行った。この場合、ネットワークのデータリンク層は Ethernet である。プログラム上で Ethernet フレームを直接取得するには、データリンクへのアクセスライブラリが必要である。通常 Linux では、ソケットシステムコールを利用してアクセスするが、今回は効率的にアナライザを開発するために、汎用ライブラリとしてアメリカ ローレンス バークレイ国立研究所で開発された libpcap を利用した。

３．開発プログラム

パケット取得プログラムと、統計解析プログラムという 2 段階処理の形式をとった。この形式をとることで、パケット取得時に、特定のプロトコルやホストにフィルタリングをかけずにデータを蓄積し、そのデータをいろいろな面から解析が行えるというメリットがある。なお、パケット取得に際しては、プライバシーを守るため、データの内容を切り落とし、ヘッダ部分だけを観察した。

（1）パケット取得プログラム

ライブラリ libpcap を使用してパケットを取得する。プログラム実行時に、観測時間、保存ファイルを指定できるようにした。ここでのデータ処理は、必要最小限の情報にとどめ、ファイルサイズを小さくして取り込むことに力を入れた。

＜libpcap の使用方法＞

① pcap_open_live 関数に、以下の引数を与え libpcap を初期化した。初期化が失敗すればエラーメッセージを表示し、プログラムを終了する。

- ネットワークインタフェース　→eth0 を指定
- 取得するデータの最大バイト数
- 読み出しのタイムアウト時間（ミリ秒）
- 無条件受信モード（promiscuous）　→通常のモードでは、自分宛て以外のパケットを破棄するが、このモードでは、ネットワーク上の全てのパケットを取り込める

ようにする。

② pcap_loop 関数に以下の引数を与え、パケットの取得を繰り返す。パケット取得のたびに packet_print 関数を呼び出し、パケットの処理を任せる。

- パケットキャプチャディスクリプタ　→パケット取得操作に必要な情報を保持
- パケット取得数　→無限ループとした（signal 関数で観測時間を制御）
- パケットを処理する関数へのポインタ　→packet_print 関数を指定

＜packet_print 関数＞

取得するパケットは、実験環境では Ethernet フレームなので、上位プロトコルの選定はタイプフィールドを参照する。選定後、上位プロトコルヘッダを参照する関数を指定する。例えば、タイプが 0x0800 なら上位は IP なので、IP ヘッダを参照する ip_print 関数に処理を任せる。

（2）統計解析プログラム

パケット取得プログラムで蓄積されたデータを、ファイルから読み取り、情報を様々な角度で解析するプログラムとした。下の図１は、パケット構造を詳細に出力したものである。図２は通信に必要な情報のみを詳細に出力したものである。他にも、特定プロトコルに関するデータのみを取り出すなど、解析方法はいろいろある。

図１ パケット構造 (一部)

```
Ethernet Frame       (87 バイト)
--Ethernet--------------------(14)
始点 MAC アドレス   0:b0:d0:ab:59:7e
終点 MAC アドレス   8: 0:20:86:3c:1e
タイプフィールド 0x0800 (IP)
--IP ヘッダ-------------------(20)
バージョン          4 (IPv4)
ヘッダ長            5 (= 20 バイト)
サービスタイプ      00000000
パケット長          73 (IP+UDP)
識別子              14413
フラグ              000
```

図２ 通信詳細 (一部)

```
20:17:08 (80 バイト)
   0:b0:d0:ab:59:7e-mei.kpu.ac.jp(udp:1685)
-> 8: 0:20:86:3c:1e-aoi(udp:domain)

20:17:08 (144 バイト)
   8: 0:20:86:3c:1e-aoi(udp:domain)
-> 0:b0:d0:ab:59:7e-mei.kpu.ac.jp(udp:1685)

20:17:08 (88 バイト)
   0:b0:d0:ab:59:7e-mei.kpu.ac.jp(udp:1685)
-> 8: 0:20:86:3c:1e-aoi(udp:domain)
```

４．結論

現在の進行状況は、統計解析プログラムの解析方法を検討、開発している。また、Tcl/Tk によるユーザーインターフェースの導入も検討している。なお、このプログラムは Unix のソケットルーチンを使うためスーパーユーザーの権限でしか走らない。制限されたユーザーが、限られたプログラムだけ root 権限で実行できる sudo プログラムを用いて実験を行った。