インターネット計測とデータ解析 第 11 回

長 健二朗

2013 年 6 月 19 日

番外編

第 11 回 パケット解析 (6/19) 6 限 (18:10-19:40) $\lambda 13$

- UNIX コマンド
- パケットキャプチャリング
- プロトコル解析
- 演習: プロトコル解析

便利な UNIX コマンド

- テキストファイルを扱う場合に便利な UNIX コマンド
  - sort, head, tail, cat, cut
  - diff, tee, grep, uniq, wc
  - join, find, sed, awk, screen

sort コマンド

sort コマンド: テキストファイルの行をソートして並び替える

```
$ sort [options] [FILE ...]
```

- options (課題で使いそうなオプション)
  - -n : フィールドを数値として評価
  - -r : 結果を逆順に並べる
  - -k POS1[,POS2] : ソートするフィールド番号 (1 オリジン) を指定する
  - -t SEP : フィールドセパレータを指定する
  - -m : 既にソートされたファイルをマージする
  - -T DIR : 一時ファイルのディレクトリを指定する

例: file を第 3 フィールドを数値とみて逆順にソート、一時ファイルは”/usr/tmp”に作る

```
$ sort -nr -k3,3 -T/usr/tmp file
```

head コマンド

head コマンド: ファイルの先頭部分を出力

- デフォルトは 10 行表示

```
head [-n lines | -c bytes] [file ...]
```

例:

```
$ sort -n -k3,3 | head -n 10
```

tail コマンド

tail コマンド: ファイルの末尾部分を出力

- デフォルトは 10 行表示

```
tail [-F | -f | -r] [-q] [-b number | -c number | -n number] [file ...]
```

- 良く使うオプション
  - -f : ファイルを監視して末尾に追加された部分を出力

例:

```
monitor a log file:
$ tail -f /var/log/httpd-access.log
```

cat コマンド

cat コマンド: ファイルの内容を (連結して) 出力

```
cat [-benstuv] [file ...]
```

例:

```
$ cat file1 file2 > file3
```

cut コマンド

cut コマンド: 指定したファイルの各行から、それぞれの一部分を切り出して出力

```
cut -b list [-n] [file ...],
cut -c list [file ...],
cut -f list [-s] [-d delim] [file ...]
```

- 良く使うオプション
  - -b BYTE-LIST：バイト位置を指定
  - -c CHAR-LIST：文字位置を指定
  - -f FIELD-LIST：フィールド位置を指定
  - -d DELIM：フィールドの区切り文字を指定

例:

```
extract users' login names and shells from the system passwd file:
$ cut -d : -f 1,7 /etc/passwd
show the names and login times of the currently logged in users:
$ who | cut -c 1-16,26-38
```

diff コマンド

diff コマンド: 複数ファイルを行ごとに比較し、異なる行を表示

```
diff [OPTION]... FILES
```

- 良く使うオプション
  - -u : unified diff format で出力

例:

```
$ diff -u file1 file2
```

tee コマンド

tee コマンド: 標準入力から読んだ内容を、標準出力とファイルの両方に出力

```
tee [-ai] [file ...]
```

例:

```
$ ls | tee output.txt
```

grep コマンド

grep コマンド: PATTERN に指定した文字列を含む行を出力する

```
grep [options] PATTERN [FILE...]
grep [options] [-e PATTERN | -f FILE] [FILE...]
```

例:

```
search lines including 'abc':
$ grep 'abc' file
count the number of lines starting with 'abc':
$ grep -c '^abc' file
```

uniq コマンド

uniq コマンド: ソートされたファイルから重複行を削除

```
uniq [-c | -d | -u] [-i] [-f num] [-s chars] [input_file [output_file]]
```

- 良く使うオプション
  - -d : 重複行のみ表示

例:

```
$ cat file1 file2 | sort | uniq > file3
```

```
$ sort file | uniq -d
```

wc コマンド

wc コマンド: ファイルの行数、単語数、バイト数を出力

```
wc [-Lclmw] [file ...]
```

join コマンド

join コマンド: 共通のフィールドでソートされたファイルの各行を結合する

```
join [-a file_number | -v file_number] [-e string] [-o list] [-t char]
    [-1 field] [-2 field] file1 file2
```

例:

```
$ cat file1
1001    orange
1002    apple
1003    grape
$ cat file2
1001    400
1002    250
1004    500
$ join file1 file2
1001 orange 400
1002 apple 250
$ join -a1 -a2 -e NULL -o '0,1.2,2.2' file1 file2
1001 orange 400
1002 apple 250
1003 grape NULL
1004 NULL 500
```

find コマンド

find コマンド: ファイルの検索

```
find [-H | -L | -P] [-EXdsx] [-f pathname] pathname ... expression
find [-H | -L | -P] [-EXdsx] -f pathname [pathname ...] expression
```

例:

```
print files with ".rej" suffix:
$ find . -name "*.rej" -print

print ".o" files older than 1 year
$ find . -name "*.o" -mtime +365 -print

remove empty files:
$ find . -empty -exec rm {} \;
```

sed コマンド (streaming editor)

sed コマンド:

```
sed [-Ealn] command [file ...]
sed [-Ealn] [-e command] [-f command_file] [-I extension]
    [-i extension] [file ...]
```

- 良く使うオプション
  - -e command : コマンドの追加
  - -f command_file : 指定したファイルに記述されているコマンドを追加

例:

```
replace "old" by "new":
$ echo "old songs in old books" | sed 's/old/new/g'

print line 3-5:
$ sed -n '3,5p' file
```

awk コマンド

awk コマンド:

- 演算機能を持つプログラム言語
- 1 行プログラムが書き易い

```
awk [ -F fs ] [ -v var=value ] [ 'prog' | -f progfile ] [ file ...  ]
```

例:

```
swap column1 and colimn2 and add sum to column3:
$ echo "12 56" | awk '{print $2,$1,$1+$2}'

extract the capacity in percent from the df command:
$ df | awk 'match($0, /[0-9]+%/) {print substr($0, RSTART, RLENGTH - 1)}'
```

screen コマンド

screen コマンド: 仮想端末 (標準コマンドでないので、インストールする必要あり)

- ひとつのターミナルで複数の仮想ターミナルを使用できる
- 端末を切り離す機能がある
  - シェルで処理を実行中に端末を切り離しても処理はバックグラウンドで継続し、後で再接続すればもとのシェルに戻れる
    - screen : screen の起動
    - ”ctrl-a d” : detach 端末の切り離し
    - screen -r : 切り離されている端末の接続

パケットキャプチャリング

- ネットワークに流れるパケットを収集
  - 通常パケットの先頭 N バイトを記録
- 詳細なプロトコルレベルの解析が可能
  - トラブルシューティングには欠かせないツール
- 制約
  - データサイズが大きい
  - プライバシーへの配慮が必要
    - 利用アドレス
    - 通信の内容

タッピング

- タッピング: ネットワークトラフィックへのアクセス
- ホストレベル
  - 自分宛以外のパケットも受信
  - ネットワークカードのプロミスキュアス機能の利用
- ネットワークレベル
  - 昔のイーサネットはセグメント上の全てのパケットが見えた
  - いまどきのスイッチは学習機能を持ち不要なパケットを流さない
  - タッピングの準備
    - スプリッタ: 光レベル、電気レベル (耐故障性への配慮)
    - スイッチ・ルータでのポートミラーリング
  - 無線の場合は利用しているチャネルの全トラフィックが見える

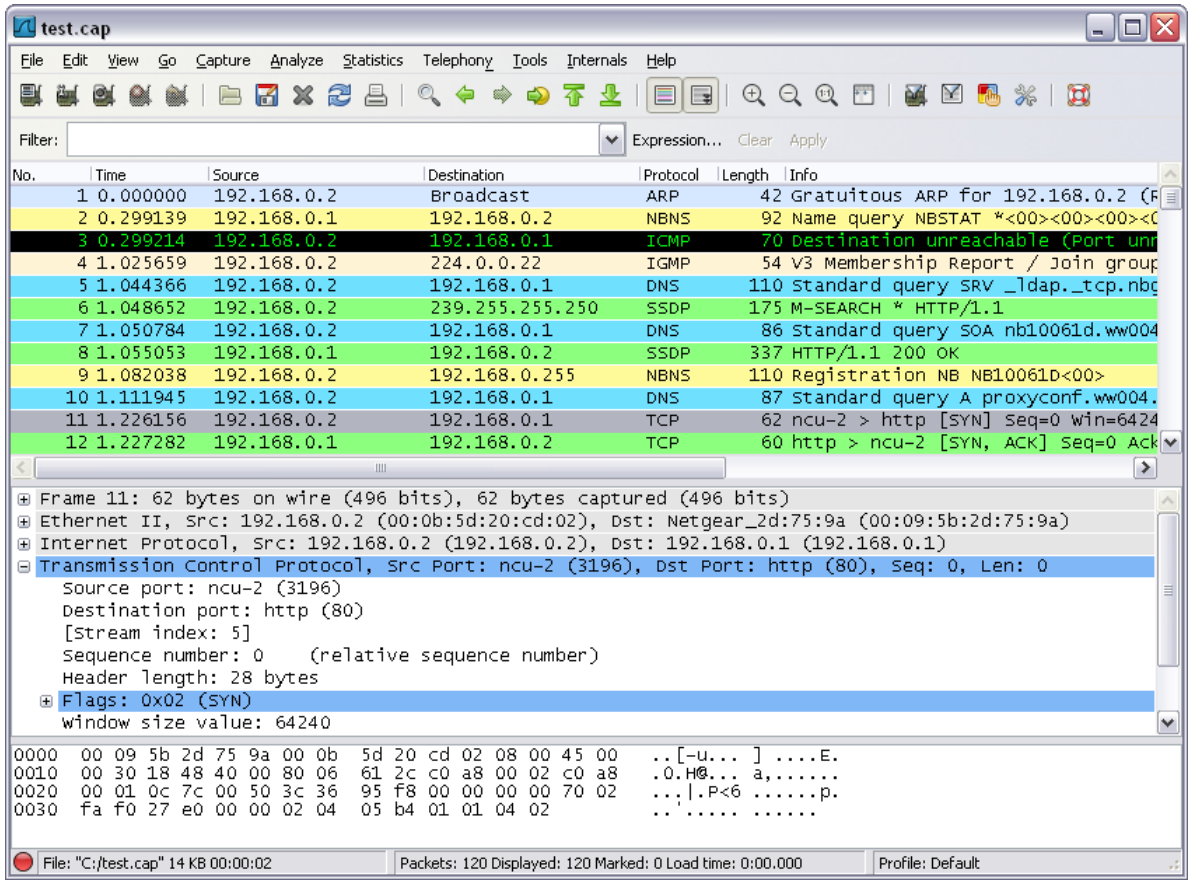
パケットキャプチャリングツール

- さまざまなツール
  - オープンソースツール
    - tcpdump: コマンドラインツール
    - wireshark: GUI ツール
  - 商用製品
    - LAN アナライザ、IDS など

wireshark

- 目的
  - ネットワークのトラブルシューティング
  - セキュリティ問題の検証
  - プロトコル実装のデバッグ
  - ネットワークプロトコル動作の学習
- 特徴
  - オープンソース、マルチプラットフォーム: UNIX,Windows,Mac
  - ライブキャプチャと保存、保存キャプチャファイルの読み込み
  - 既存のファイルフォーマットサポート: pcap, pcap-ng, etc.
  - プロトコル詳細の表示機能 (各種プロトコル解析機能)
  - 多彩なパケット検索、フィルタ機能、色分け表示
  - 統計情報表示機能

wireshark main window



- menu: 各機能のメニュー
- main toolbar: 良く使う機能のアイコン
- filter toolbar: フィルタ操作
- packet list pane: 各パケットのサマリ付きパケットリスト
- packet details pane: 選択したパケットの詳細
- packet bytes pane: 選択したパケットの 16 進数表示
- statusbar: キャプチャデータのサマリ、ステータス

wireshark の使い方

- wireshark user's guide
  - `http://www.wireshark.org/docs/wsug_html_chunked/`
- 実践パケット解析 第 2 版 — Wireshark を使ったトラブルシューティング. オライリージャパン. 2012.

解析演習用データ

- wireshark wiki sample captures
  - `http://wiki.wireshark.org/SampleCaptures`
- capture files used in "Practical Packet Analysis, 2nd Edition"
  - `http://nostarch.com/packet2.htm`
- packet traces from WIDE backbone
  - `http://mawi.wide.ad.jp/mawi/`

演習: パケット解析

実際に wireshark を使ってパケット解析をする

参考文献

[1] Ruby official site. `http://www.ruby-lang.org/`

[2] gnuplot official site. `http://gnuplot.info/`

[3] Mark Crovella and Balachander Krishnamurthy. *Internet measurement: infrastructure, traffic, and applications*. Wiley, 2006.

[4] Pang-Ning Tan, Michael Steinbach and Vipin Kumar. *Introduction to Data Mining*. Addison Wesley, 2006.

[5] Raj Jain. *The art of computer systems performance analysis*. Wiley, 1991.

[6] Toby Segaran. (當山仁健 鴨澤眞夫 訳). 集合知プログラミング. オライリージャパン. 2008.

[7] Chris Sanders. (高橋基信 宮本久仁男 監訳 岡真由美 訳). 実践パケット解析 第 *2* 版 — *Wireshark* を使ったトラブルシューティング. オライリージャパン. 2012.

[8] あきみち、空閑洋平. インターネットのカタチ. オーム社. 2011.

[9] 井上洋, 野澤昌弘. 例題で学ぶ統計的方法. 創成社, 2010.

[10] 平岡和幸, 掘玄. プログラミングのための確率統計. オーム社, 2009.