2019.5 第 1 版

パケットの解析

1. ネットワーク情報を調べる

実験１ ネットワークコマンド(ipconfig /all)を使って、自分の PC に設定されているネットワーク情報①～⑥を調べ、実験ノートに記録する。

① 自分の PC の MAC アドレス
② IP アドレスが動的に割りつけられたものか？静的に割りつけられたものか？
③ 自分の PC の IP アドレス
④ サブネットマスク
⑤ デフォルトゲートウェイの IP アドレス
⑥ 優先 DNS サーバーの IP アドレス

コマンドプロンプト(通称 Dos 窓)を開き ipconfig /all と入力すると、ネットワーク情報が表示される。以下の実習例に従って、①～⑥のネットワーク情報を実験ノートに記録しなさい。

```
C:¥>ipconfig /all
Windows IP 構成
イーサネット アダプター ローカル エリア接続:
① 物理アドレス. . . . . : 6C-62-6D-6B-8A-68
② DHCP 有効 . . . . . : はい
  自動構成有効. . . . . : はい
③ IPv4 アドレス . . . . . : 172.17.181.1(優先)
④ サブネット マスク . . . . . : 255.255.0.0
⑤ デフォルト ゲートウェイ . . . . . : 172.17.254.254
⑥ DNS サーバー. . . . . : 172.17.254.254
```

この(赤色)情報を記録する
ただし、このテキストに書かれている内容はサンプル

2. パケットモニターを使ってみる

実験２ 自分の PC から簡易 Web サーバー(172.17.181.100)へ、ネットワークコマンドの ping を使って検査パケット(ICMP : Internet Control Message Protocol)を送る。このときにやり取りするパケット(要求パケットと応答パケット)を、パケットモニター(Wireshark)を使ってキャプチャし、以下のパケット内の情報①～③を実験ノートに記録する。

① 送信元 IP アドレス
② 宛先 IP アドレス
③ プロトコルの名称

以下の手順に従って実験を進めなさい

(1)ネットワークコマンドの ping 実行前に、パケットモニターを使ってパケットをキャプチャする
ディスクトップにあるアイコンをクリックして、Wireshark を起動する。ツールバーの[interface]ボタンを押して、キャプチャインタフェース画面を表示させ、使用している LAN カードを選択して、[Start]ボタンを押す。※詳しくは「パケットモニター(Wireshark)の使い方」参照。

(2)コマンドプロンプトを開き、ネットワークコマンドの ping を簡易 Web サーバー(教卓 PC:172.17.181.100)へ送る
検査パケットが簡易 Web サーバー(教卓 PC)へ到達すると、以下のような応答が表示する。未到達の場合は、原因を見つけて解決し、再度ネットワークコマンド ping を実行する。

```
C:¥>ping 172.17.181.100

172.17.181.100 に ping を送信しています 32 バイトのデータ:
172.17.181.100 からの応答: バイト数 =32 時間 =16ms TTL=53
172.17.181.100 からの応答: バイト数 =32 時間 =9ms TTL=53
172.17.181.100 からの応答: バイト数 =32 時間 =15ms TTL=53
172.17.181.100 からの応答: バイト数 =32 時間 =31ms TTL=53

172.17.181.100 の ping 統計:
    パケット数: 送信 = 4、受信 = 4、損失 = 0 (0% の損失)、
ラウンド トリップの概算時間 (ミリ秒):
    最小 = 9ms、最大 = 31ms、平均 = 17ms
```

(3)パケットキャプチャを停止し、パケットの内容を確認する

ツールバーの[Stop]ボタンを押し、キャプチャを停止する。パケット一覧部に表示されているたくさんのパケットから、フィルタを使ってネットワークコマンド ping の ICMP の要求パケット(Echo request)と応答パケット(Echo request)のみを表示させる。※「パケットモニター(Wireshark)の使い方」参照。

フィルタ

Filter: icmp　Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
2507	49.7816110	192.168.0.9	59.106.13.138	ICMP	74	Echo (ping) request id=0x0001, seq=13/3328, ttl=128 (reply in 2508)
2508	49.8005560	59.106.13.138	192.168.0.9	ICMP	74	Echo (ping) reply id=0x0001, seq=13/3328, ttl=53 (request in 2507)
2553	50.7931390	192.168.0.9	59.106.13.138	ICMP	74	Echo (ping) request id=0x0001, seq=14/3584, ttl=128 (reply in 2554)
2554	50.8172630	59.106.13.138	192.168.0.9	ICMP	74	Echo (ping) reply id=0x0001, seq=14/3584, ttl=53 (request in 2553)
2556	51.8087470	192.168.0.9	59.106.13.138	ICMP	74	Echo (ping) request id=0x0001, seq=15/3840, ttl=128 (reply in 2557)
2557	51.8329630	59.106.13.138	192.168.0.9	ICMP	74	Echo (ping) reply id=0x0001, seq=15/3840, ttl=53 (request in 2556)
2569	52.8244090	192.168.0.9	59.106.13.138	ICMP	74	Echo (ping) request id=0x0001, seq=16/4096, ttl=128 (reply in 2570)
2570	52.8398710	59.106.13.138	192.168.0.9	ICMP	74	Echo (ping) reply id=0x0001, seq=16/4096, ttl=53 (request in 2569)

①送信元 IP アドレス　②宛先 IP アドレス　③プロトコル名　応答パケット　要求パケット

3. TCP コネクション(接続)を解析する

簡易 Web サーバー(172.17.181.100)のホームページ閲覧時のパケットをキャプチャし、キャプチャしたパケットの中から、フィルタを使って TCP パケットのみを表示させる。

以下の手順に従って実習を進めなさい

(1) パケットモニターを使ってパケットキャプチャする

(2) ブラウザ(IE など)を使って簡易 Web サーバのホームページを表示する
　ブラウザのアドレスバーに簡易 Web サーバの IP アドレス(172.17.181.100)を入力する

(3) パケットキャプチャを停止する

(4) パケット一覧部に表示されているたくさんのパケットから、フィルタを使って TCP パケットのみを表示させる。このとき、以下のように TCP と自分の PC の IP アドレス(172.17.181. ? /16)を and(論理積)にすると、目的のパケットを見つけやすい。

Filter: tcp and ip.addr == 192.168.0.9/24

プロトコル(tcp)　and (論理積)　自分の PC の IP アドレス (172.17.181. ? /16)

実験 3　パケット一覧部を上部から下部に向かって、Info の列に[SYN]、[SYN,ACK]、[ACK]と表記している3つのパケットを探し、これらのパケットの情報①～⑤を実験ノートに記録する。

① 送信元 IP アドレス
② 宛先 IP アドレス
③ 送信元 MAC アドレス
④ 宛先 MAC アドレス
⑤ TCP ヘッダ内の flags(16 進数)=>例:0x???

＜パケット一覧部＞

325	3.37393700	192.168.0.9	103.243.222.37	TCP	66 60904→80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1
418	3.49118800	103.243.222.37	192.168.0.9	TCP	66 80→60904 [SYN, ACK] Seq=0 Ack=1 Win=26580 Len=0 MSS=1414 SACK_PERM=1 WS=512
421	3.49122800	192.168.0.9	103.243.222.37	TCP	54 60904→80 [ACK] Seq=1 Ack=1 Win=262144 Len=0

①送信元 IP アドレス　②宛先 IP アドレス　Info 列 : [SYN] [SYN, ACK] [ACK]

＜パケット詳細部＞

```
⊟ Ethernet II, Src: AsrockIn_83:e6:95 (bc:5f:f4:83:e6:95), Dst: NecPlatf_aa:4c:f0 (a4:12:42:aa:4c:f0)
  ⊞ Destination: NecPlatf_aa:4c:f0 (a4:12:42:aa:4c:f0)
  ⊞ Source: AsrockIn_83:e6:95 (bc:5f:f4:83:e6:95)
    Type: IP (0x0800)
```

＜パケット詳細部＞

```
⊟ Transmission Control Protocol, Src Port: 60904 (60904), Dst Port: 80 (80), Seq: 0, Len: 0
    Source Port: 60904 (60904)
    Destination Port: 80 (80)
    [Stream index: 27]
    [TCP Segment Len: 0]
    Sequence number: 0    (relative sequence number)
    Acknowledgment number: 0
    Header Length: 32 bytes
  ⊞ .... 0000 0000 0010 = Flags: 0x002 (SYN)
```

実験 4　パケット一覧部において、Info の列に[FIN,ACK]、[ACK]と連続で表記されている2つのパケットを探し、これらのパケットの情報①～⑤をノートに記録する。

① 送信元 IP アドレス
② 宛先 IP アドレス
③ 送信元 MAC アドレス
④ 宛先 MAC アドレス
⑤ TCP ヘッダ内の flags(16 進数)

4. Web アクセスを解析する

実習3や実習4でキャプチャしたパケットの中から HTTP のみを表示させる。パケット一覧部に表示されているたくさんのパケットから、以下のようにフィルタを使って HTTP パケットのみを表示させる。

Filter: http and ip.addr == 192.168.0.9/16

プロトコル　and（論理積）　自分の PC の IP アドレス（172.17.181. ? /16）

実験 5　パケット一覧部で Info の列に GET / HTTP/1.1 と表記されているパケットを探し、このパケットの情報①～⑥を実験ノートに記録する。

① 送信元 IP アドレス
② 宛先 IP アドレス
③ 送信元ポート番号
④ 宛先ポート番号
⑤ TCP ヘッダ内の flags(16 進数)=>例：0x???
⑥ Hypertext Tranfer Protocol(HTTP)ヘッダ内の User-Agent(使用ブラウザ名)

＜パケット一覧部＞

```
332 6.87579100 192.168.0.9        173.194.117.240    HTTP    1108 GET / HTTP/1.1
```

①送信元 IP アドレス　②宛先 IP アドレス　Info 列：GET / HTTP/1.1

＜パケット詳細部＞

```
⊟ Transmission Control Protocol, Src Port: 63841 (63841), Dst Port: 80 (80), Seq: 1, Ack: 1, Len: 1054
    Source Port: 63841 (63841)
    Destination Port: 80 (80)
    [Stream index: 5]
    [TCP Segment Len: 1054]
    Sequence number: 1    (relative sequence number)
    [Next sequence number: 1055    (relative sequence number)]
    Acknowledgment number: 1    (relative ack number)
    Header Length: 20 bytes
  ⊞ .... 0000 0001 1000 = Flags: 0x018 (PSH, ACK)
```

```
⊟ Hypertext Transfer Protocol
  ⊞ GET / HTTP/1.1\r\n
    Accept: text/html, application/xhtml+xml, image/jxr, */*\r\n
    Accept-Language: ja\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/42.0.2311.135 Safari/537.36 Edge/12.10240\r\n
```

⑥使用ブラウザ名

実験 6 パケット詳細一覧部で Info の列に HTTP/1.1 200 Document follows と表記されているパケットを探し、このパケットの情報①～⑤を実験ノートに記録する。

① 宛先 IP アドレスおよび送信元 IP アドレス
② 宛先 MAC アドレスおよび送信元 MAC アドレス
③ TCP ヘッダ内の flags(16 進数)
④ HTTP ヘッダ内の Server 及び Content-type の内容
⑤ Line-based text data 以下の全内容(数行になることもある)

> **注意**:目的のパケットが取得できないときには、以下の手順に従って、インターネット一時ファイルを削除してから再度 Web アクセスしてください。インターネット一時ファイルを削除する手順は、以下の通り。
> ＊「コントロールパネル」→「ネットワークとインターネット」→「インターネットオプション」→「全般」→「閲覧の履歴」→「削除」→「すべてを削除」→「はい」→「OK」
> 上記手順でも、目的のパケットが取得できないときには、Ctrl キー ＋ f5 キーを実行して強制的に Web アクセスする

5. ARP(Address Resolution Protocol:アドレス解決プロトコル)を解析する

相手の IP アドレスを指定して通信するためには、MAC アドレスも必要となる。この2つのアドレスを結び付けるのが、ARP の役割である。ARP 通信は、IP アドレスをキー(手がかり)として MAC アドレスを問い合わせる ARP 要求と、この問い合わせに対して IP アドレスに対応した MAC アドレスを返答する ARP 応答から成立している。

簡易 Web サーバ(172.17.181.100)の IP アドレスを指定して、ネットワークコマンドの ping を使って検査パケットを送る。この時にやりとりするパケットをキャプチャし、キャプチャした複数のパケットの中から ARP パケットを見つけて指定した情報を実験ノートに記録する。

以下の手順(1)～(4)に従って実習を進めなさい

(1) 自分の PC 内に保存されている ARP キャッシュ(MAC 情報)をすべて削除する
コマンドプロンプトを開き、ARP キャッシュを削除するコマンド(arp -d)を実行する

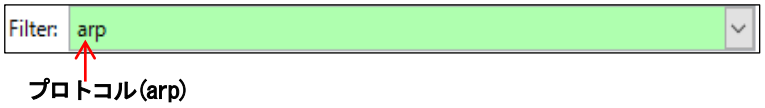
```
C:¥>arp -d
```

※このコマンド(arp -d)が実行できない場合は、手順(2)へ進む

(2) ブラウザ(IE など)を使って簡易 Web サーバのホームページを表示する
ブラウザのアドレスバーに簡易 Web サーバの IP アドレス(172.17.181.100)を入力する

(3) パケットキャプチャを停止する

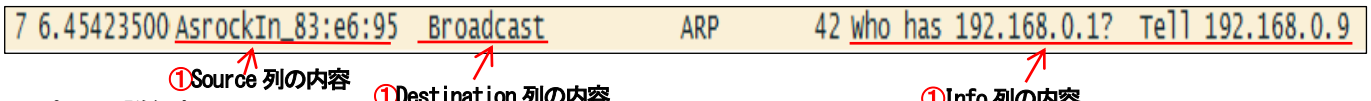
(4) パケット一覧部に表示されているたくさんのパケットから、フィルタを使って ARP パケットのみを表示させる



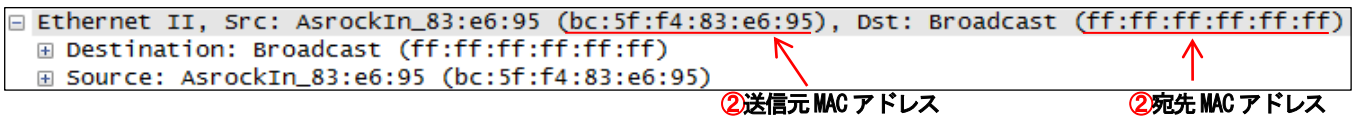
実験 7 パケット一覧部に表示されているパケットの中から Info の列に who has ＊＊＊＊？と表記されているパケットを探し、このパケットの情報①～④を実験ノートに記録する。

① パケット一覧部の Source 列、Destination 列、Info 列の内容
② Ethernet II ヘッダ内の送信元 MAC アドレス(16 進数)、宛先 MAC アドレス(16 進数)
③ Address Resolution Protocol ヘッダ内の Sender MAC address, Sender IP address
④ Address Resolution Protocol ヘッダ内の Target MAC address, Target IP address

＜パケット一覧部＞



＜パケット詳細部＞



```
⊟ Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IP (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    Sender MAC address: AsrockIn_83:e6:95 (bc:5f:f4:83:e6:95)
    Sender IP address: 192.168.0.9 (192.168.0.9)
    Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
    Target IP address: 192.168.0.1 (192.168.0.1)
```

③ Sender MAC address

③ Sender IP address

④ Target MAC address

④ Target IP address

実験 8　パケット一覧部に表示されているパケットの中からInfo 列に＊＊＊＊ is at ＊＊＊＊と表記されているパケットを探し、このパケットの情報①～④を実験ノートに記録する。※パケットの情報③④は実験7と同じ方法で取得する

① パケット一覧部の Source 列、Destination 列、Info 列の内容
② Ethernet Ⅱ ヘッダ内の送信元 MAC アドレス（16 進数）、宛先 MAC アドレス（16 進数）
③ Sender MAC address, Sender IP address
④ Target MAC address, Target IP address

＜パケット一覧部＞

8	6.45467900	NecPlatf_aa:4c:f0	AsrockIn_83:e6:95	ARP	60	192.168.0.1 is at a4:12:42:aa:4c:f0

①Source 列の内容　①Destination 列の内容　①Info 列の内容

2019.5 第 1 版

パケットモニター(Wireshark)の使い方

1. Wireshark とは

ネットワーク上（LAN ケーブルに流れている）のパケットを取得して、その中の情報を画面に表示するソフトウェア（LAN アナライザーまたはパケットモニター）の1つに Wireshark がある。Wireshark は、非常に高機能なオープンソース（ソースコードが公開されている）の LAN アナライザで、誰でも自由にコンピュータにインストールして利用することができる。（公式サイト http://www.wireshark.org）

＜主な利用目的＞

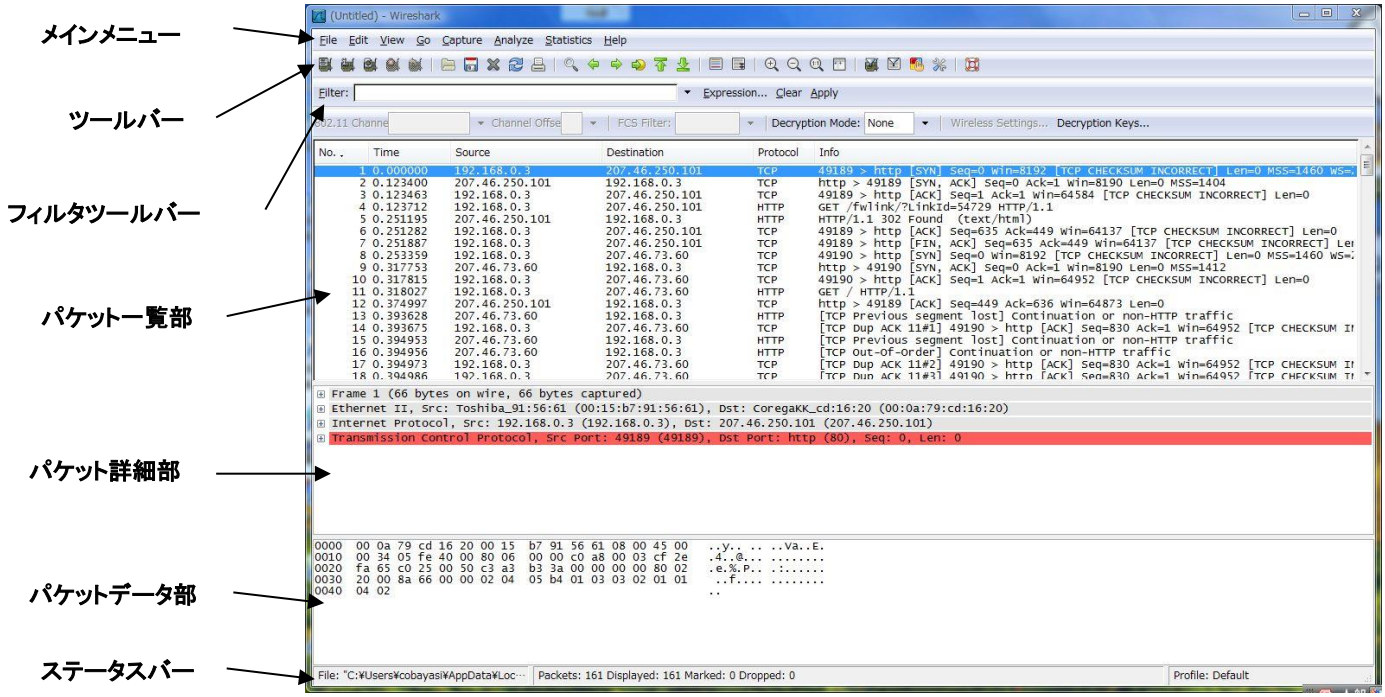
- ネットワーク管理者がネットワークのトラブルを解決するため
- セキュリティ技術者がセキュリティの問題を解決するため
- 開発技術者がプロトコルの実装についてデバッグ（誤り確認）するため
- ネットワークプロトコルの内部について学習するため

＜主な機能＞

- ネットワークインタフェースから送出されるパケットをキャプチャ（取得する）する
- パケットの内容を詳細表示する
- パケットキャプチャしたデータを開いたり保存したりできる
- 他の LAN アナライザが取得したキャプチャデータを変換して読み込み、書き出しができる
- さまざまな条件でパケットを絞り込んで検索できる
- 特定のパケットのみ色づけできる
- さまざまな統計情報を作成できる

2. ユーザインタフェース

Wireshark は、パケットをキャプチャするための様々なメニューやツールバーで構成されている。また、右クリックで呼び出したサブメニューで、オプションなどを指定することができる。ここでは、キャプチャしたパケットを解析するために必要な Wireshark の画面構成などについて述べる。



＜画面構成と各部の名称＞

[メインメニュー]

Wireshark の様々な機能を呼び出すために使用され、以下の項目がある。

メインメニューの項目と主機能

項 目 名	主 な 機 能
File	ファイルの開閉・保存・結合・印刷など
Edit	パケット検索・強調表示
View	ツールバーや画面表示、パケットの色分け
Go	特定パケットの表示
Capture	キャプチャの開始・終了、キャプチャフィルタ(キャプチャ時の絞り込み)の編集
Analyze	表示フィルタ(絞り込み表示)の作成・編集、パケット解析
Statistics	パケット数の一覧やプロトコル分布などの統計情報の表示
Help	ヘルプ機能

[ツールバー]

よく使う機能をボタン化してある

[フィルタツールバー]

パケットを絞り込む(フィルタ機能)ときに使用する

[パケット一覧部]

キャプチャしたパケットを一覧にして表示する。1つのパケットを1行で表示し、簡単な説明(info)がつけられる。ここでクリックして選択した1つのパケットの詳細な内容が、[パケット詳細部]に表示される。

パケット一覧部・表示内容

項 目 名	内 容
No.	キャプチャした順番を示す番号
Time	キャプチャした時刻
Source	送信元アドレス
Destination	宛先アドレス
Protocol	プロトコルの名称
info	パケットの概要

[パケット詳細部]

[パケット一覧部]で選択したパケットの詳細内容(解析内容)を表示する。また、ヘッダやレイヤに付けられている⊞印をクリックすると、詳細な解析内容が表示される。

[パケットデータ部]

[パケット一覧部]で選択したパケットの内容を、16 進数や ASCII コードなどで表示する。より細かく解析する時に利用する。

[ステータスバー]

現在のプログラムの状態やパケットキャプチャに関する情報が表示される。通常、左側には現在の Wireshark の状態及びキャプチャファイルや選択したフィールドの内容が、右側には現在選択しているパケットの番号が表示されている。

3. キャプチャ操作とパケット表示

ここでは、web サーバー(PC1)からサンプルホームページをダウンロードするときにやり取りされるパケットを例に取って、キャプチャの操作手順とパケットの表示と確認方法を説明する。

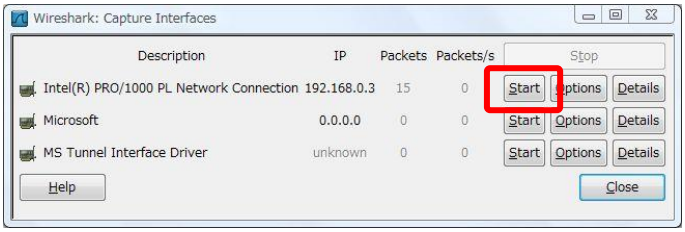
＜キャプチャ操作手順＞

① デスクトップの下に示すアイコンをダブルクリックして Wireshark を起動する。

Wireshark のアイコン

② ツールバーの[interfaces]ボタン(一番左にあるボタン)を押して、キャプチャインタフェース画面を表示させる。

③ 使用している通信アダプタ(LAN カード)を選択して[Start]ボタンを押す。この時点でキャプチャが開始されるので、何らかのパケットが表示される

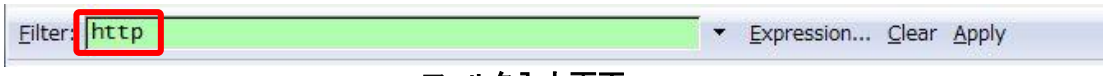


キャプチャ画面

④ ブラウザを起動させ、web サーバー(PC1)のアドレスを指定して、ホームページを表示させる。ホームページのダウンロードが終了したら、ツールバーの[Stop]ボタン(左から4番目のボタン)を押す。

＜パケットの表示＞

キャプチャ終了直後は、目的以外のパケットが表示されているので、表示フィルタを使って絞り込みを行う。今回は、http プロトコルのみを表示するために、以下の通りにフィルタツールバーのボックスに http を入力する。

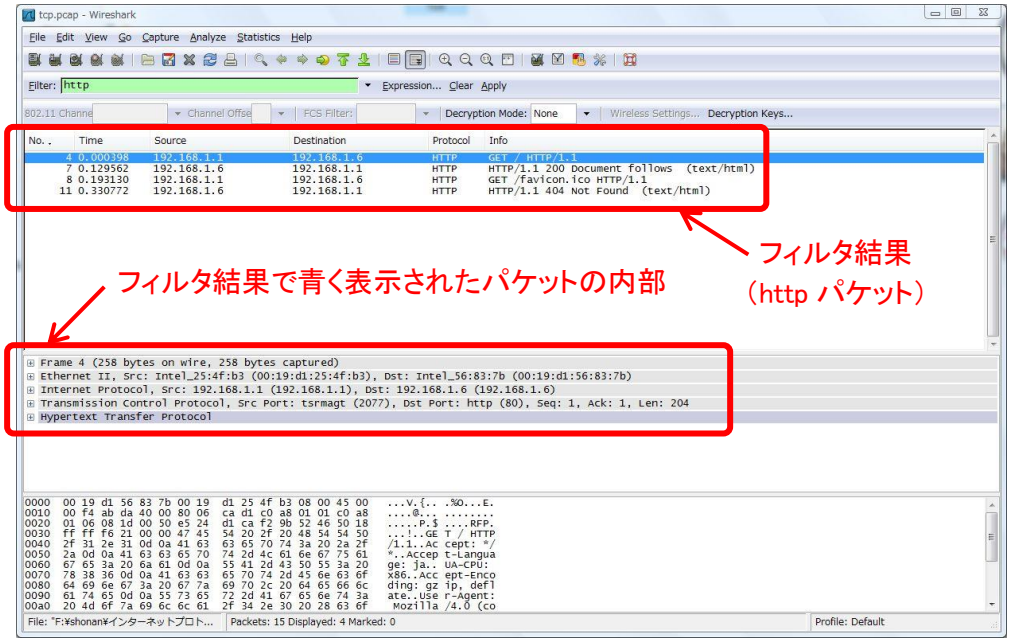


フィルタ入力画面

http を入力し終わるとボックスが緑色に変わるので、Enter キーを押すとパケット一覧部に、以下のように http プロトコルのみが表示される。

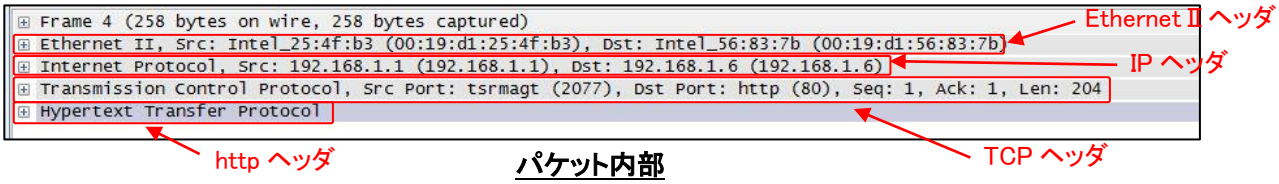
＜パケット(各ヘッダ)の確認＞

ここでは、表示されたパケットヘッダの構成を確認する方法について、1つのパケットを例に挙げて説明する。



フィルタ結果表示

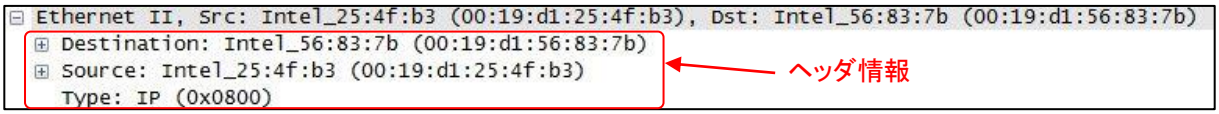
表示されているパケットで[info]の列で GET /http/1.1 と表示されているパケットを選び、パケット詳細部の表示内容を確認する。



パケット内部

上図(パケット内部)の[Ethernet II][Internet Protocol][Transmission Control Protocol][Hypertext Transfer Protocol]は、パケットの**ヘッダ**と呼ぶ部分で複数の要素から構成されており、重要な役割を果たしている。また、最上部の[Frame]は、フレーム情報(プロトコルの構成やサイズなど)を表している。

＊Ethernet II ヘッダ(※行頭のプラス印＋をクリックするとヘッダ情報が表示する)

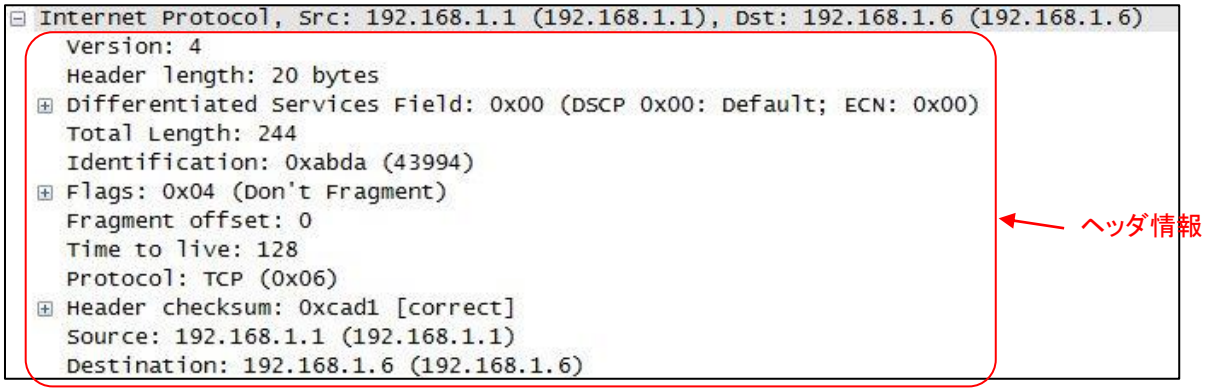


Ethernet II ヘッダ情報

Ethernet II ヘッダの主要素と内容

Destination	宛先 LAN カードの MAC アドレス
Source	送信元 LAN カードの MAC アドレス
Type	上位プロトコルタイプ(ヘッダ形式)

＊IP ヘッダ(※行頭のプラス印＋をクリックするとヘッダ情報が表示する)



IP ヘッダ情報

IP ヘッダの主要素と内容

Vertion	バージョン(通常は 4)
Header length	ヘッダサイズ(通常は 20 バイト)
Differentiated Service Field	パケット優先度(プロトコル優先度)
Total Length	IP パケットサイズ(単位:バイト)
Identification	パケット識別情報
Flags	パケット分割識別子
Fragment offset	パケット分割位置
Time to live	パケットの寿命
Protocol	上位プロトコルタイプ
Header checksum	パケット破損確認情報
Source	送信元 IP アドレス
Destination	宛先 IP アドレス

＊TCP ヘッダ（※行頭のプラス印＋をクリックするとヘッダ情報が表示する）

```
⊟ Transmission Control Protocol, Src Port: tsrmagt (2077), Dst Port: http (80), Seq: 1, Ack: 1, Len: 204
    Source port: tsrmagt (2077)
    Destination port: http (80)
    Sequence number: 1    (relative sequence number)
    [Next sequence number: 205    (relative sequence number)]
    Acknowledgement number: 1    (relative ack number)
    Header length: 20 bytes
  ⊞ Flags: 0x18 (PSH, ACK)
    Window size: 65535
  ⊞ Checksum: 0xf621 [correct]
```

ヘッダ情報

TCP ヘッダ情報

TCP ヘッダの主要素と内容

Source port	送信元ポート番号
Destination port	宛先ポート番号
Sequence number	順序番号（パケットの順番）
Acknowledgement number	応答確認番号（受け取ったパケットの順番）
Header length	ヘッダサイズ（通常 20 バイト）
Flages	TCP 通信制御のための識別子
Windows size	受信バッファサイズ（単位：バイト）
Checksum	パケット破損確認情報

＊http ヘッダ（※行頭のプラス印＋をクリックするとヘッダ情報が表示する）

```
⊟ Hypertext Transfer Protocol
  ⊟ GET / HTTP/1.1\r\n
      Request Method: GET
      Request URI: /
      Request Version: HTTP/1.1
    Accept: */*\r\n
    Accept-Language: ja\r\n
    UA-CPU: x86\r\n
    Accept-Encoding: gzip, deflate\r\n
    User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1)\r\n
    Host: 192.168.1.6\r\n
    Connection: Keep-Alive\r\n
    \r\n
```

ヘッダ情報

httpヘッダ情報

TCP ヘッダの主要素と内容

GET	要求メソッド（Web ページ取得）
Accept	Web ブラウザの書式
Accept-Language	Web ブラウザの利用言語
UA-CPU	使用 CPU（コード形式）
Accept-Encoding	Web ブラウザのデータ形式
User-Agent	Web ブラウザの種類やバージョン
Host	Web サーバーの情報（IP アドレス）
Connection	Web サーバーの接続情報