TCP/IP概論

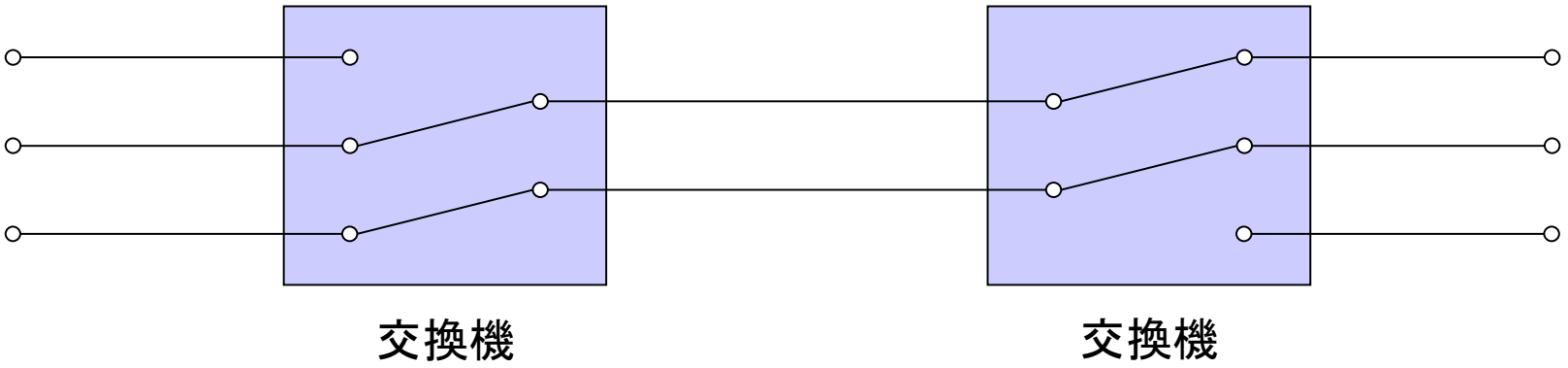
名無しさん@なんでも実況V大学

今日の授業

- インターネットの基本的構造
  - 回線交換とパケット交換
  - レイヤリングモデル

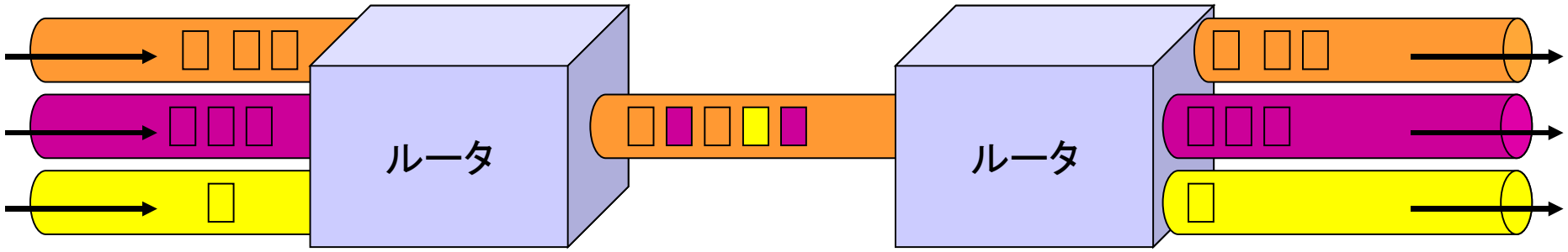
回線交換とパケット交換

- 回線交換の例
  - ユーザーが途中の回線を占有する
  - 途中の回線を越えるユーザーが起きると「回線が使えなくなる」
    - 例・地震のとき等
  - 電話回線など



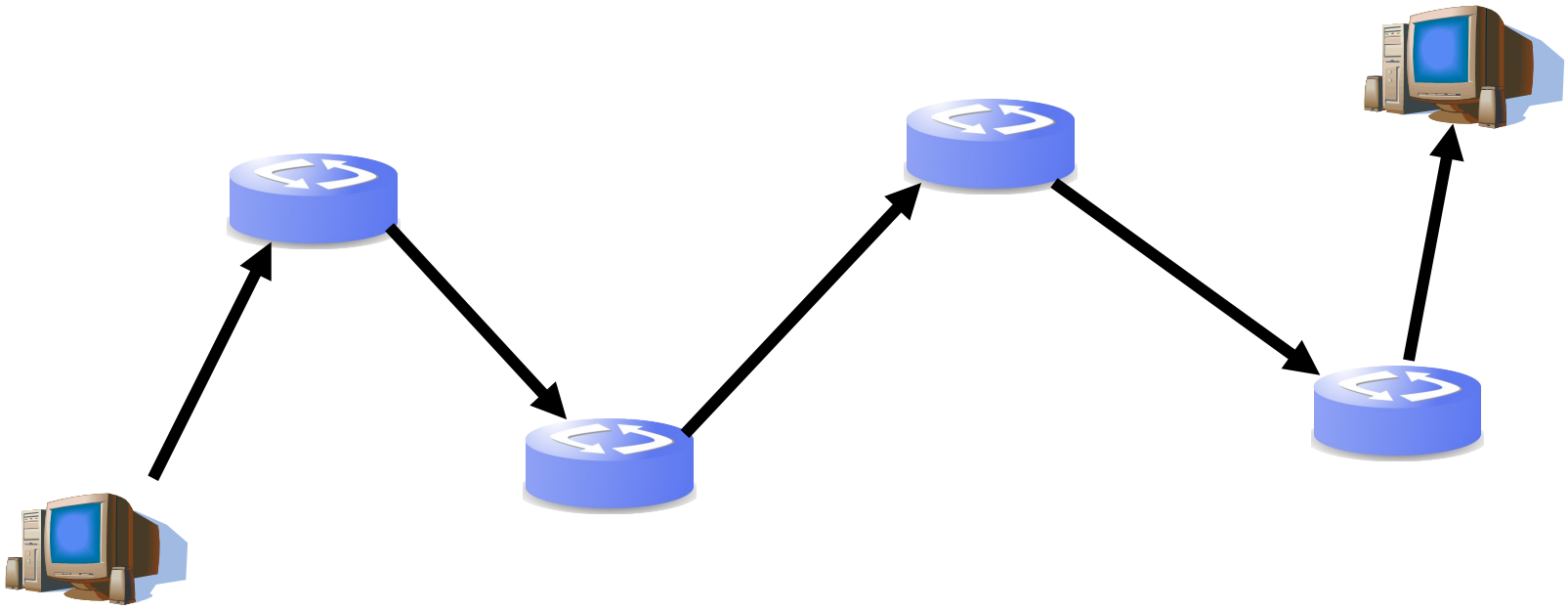
回線交換とパケット交換

- パケット交換の例
  - ユーザーが途中の回線を共有する
  - データは「パケット」と言う単位に分割されて送られる



パケット交換

- パケットは中継機を経由して宛先に届く
  - TCP/IP（インターネット）の場合はルータと呼ばれる
  - ブロードバンドルータとかでおなじみ？



Traceroute

- 実際にどういう風に飛んでいるか見てみましょう！
  - コマンドプロンプトを開く
    - （ファイル名を指定して実行→cmd)
  - 「tracert (宛先アドレス)」と入力
  - 宛先までどんなルータを経由しているかわかる

経路選択

- じゃあ、どうやって道をえらんでいるんだろう？
  - 経路選択問題
- 身近なところで、やはり自分のPCの経路を見てみましょう

経路表

- PCの経路表を見る場合
  - コマンドプロンプトを開く
    - （ファイル名を指定して実行→cmd)
  - 「route print」と入力

- ばらばらと出てきましたが・・・・
  - 結局は外に行くパケットは「default」に
  - 家のルータなどで一つのISPとつないでいる、つまり出口が一つしかないため

経路表に手を加えると・・・・

route add (宛先アドレス) mask 255.255.255.255 gateway 127.0.0.1

宛先のアドレスは、次に 127.0.0.1(自分)に送る

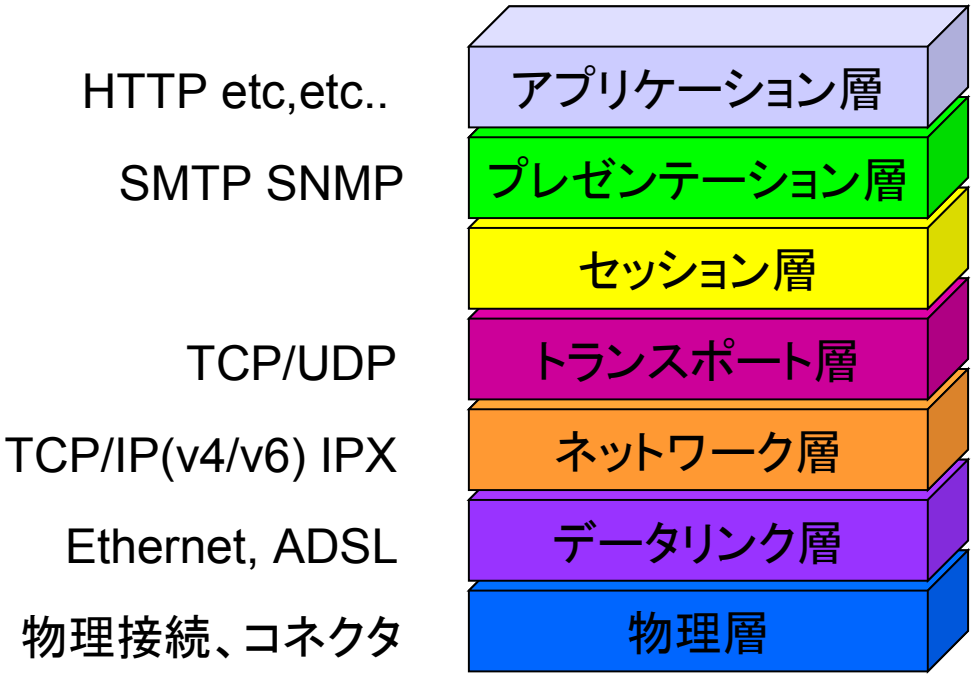
永遠に届かない！！

まとめ

- 回線交換とパケット交換
  - パケット交換方式
  - 経路選択・経路表

レヤリングモデル

- 必ず出てくるOSI参照モデル
  - 正直「もう見飽きた」という人が多いでしょうが・・・



そもそもなんでこんな層に分ける必要があるの？？

レイヤリング無しで
ブラウザのプログラムを書く

/* URL をサーバに送る */
＃場合１：TCP/IP を使ってる場合　・・・・・
＃場合１の１：無線を使ってる場合　・・・・・・
＃場合１の１の１：暗号化無し無線使ってる場合・・・・・
＃場合１の１の２：WEPを使ってる場合・・・
＃場合１の１の２の１：802.11a を使ってる場合
　電波をこういう波形でこういう風に出して、こういう電波がきたらどうこう

＃場合１の１の２の２：802.11bを使ってる場合
　電波をこういう波形でこういう風に出して、こういう電波がきたらどうこう

＃場合１の２：有線を使ってる場合
＃場合１の２の１：Ethernetを使ってる場合
＃場合１の２の１の２：1000Base-T を使ってる場合
　電線にこういう風に電気を流して、こういう電気が来たら・・・
　　：（中略）
＃場合２：IPXを使ってる場合・・・・

ブラウザのプログラムを書く

```
/* URL をサーバに送る */
#場合1:TCP/IP を使ってる場合 ・・・・・
#場合1の1:無線を使ってる場合 ・・・・・・
#場合1の1の1:暗号化無し無線使ってる場合・・・・・
#場合1の1の2:WEPを使ってる場合・・・
#場合1の1の2の1:802.11a を使ってる場合
  電波をこういう波形でこういう風に出して、こういう電波がきたらどうこう

#場合1の1の2の2:802.11bを使ってる場合
  電波をこういう波形でこういう風に出して、こういう電波がきたらどうこう

#場合1の2:有線を使ってる場合
#場合1の2の1:Ethernetを使ってる場合
#場合1の2の1の2:1000Base-T を使ってる場合
  電線にこういう風に電気を流して、こういう電気が来たら・・・
     :(中略)
#場合2:IPXを使ってる場合・・・・
     :
     :
```

```
 ∧_∧
(･ω･;)
U   U)
 し ⌒J
```

・・・・さすがにこんなことまで
把握してプログラム書けないんよ

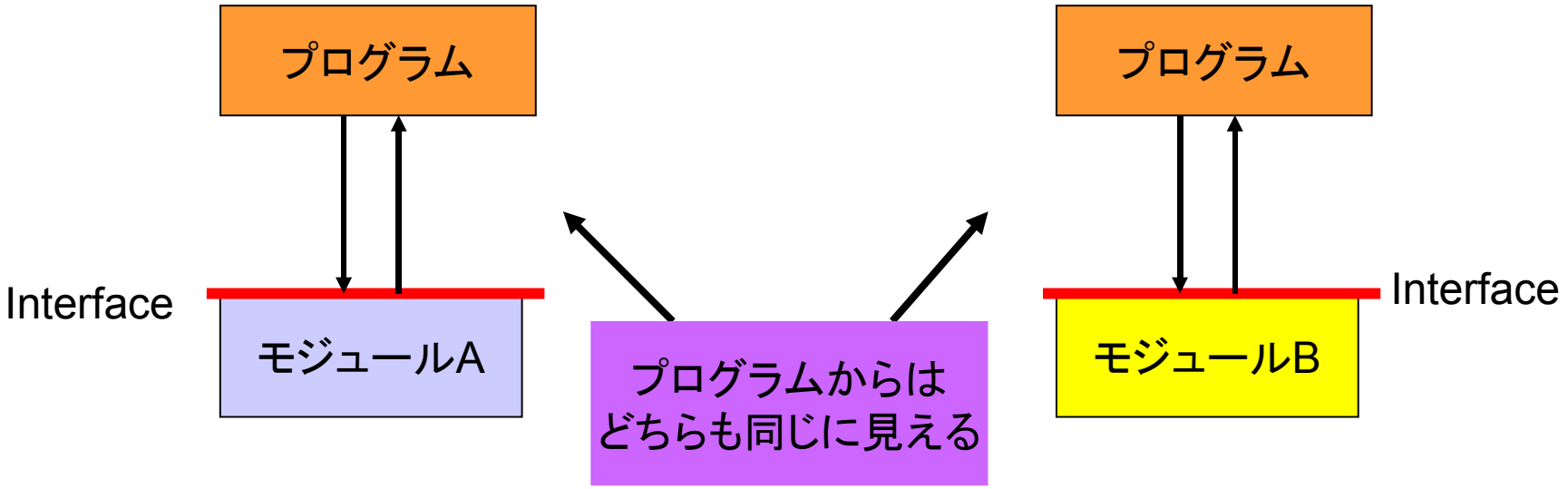
```
      |
 ＼  ＿  ／
＿ (m) ＿  ﾋﾟｺｰﾝ
     |ミ|
 ／  .` ´  ＼
  ∧_∧
 ( ･ω･)
 (U  U
 し ⌒J
```

各部分はそれぞれ専門の奴に
書いてもらえばいんじゃね？

→ **抽象化の概念**

抽象化

- 階層ごとに「インターフェース」を決めて、それぞれを分離する
- 階層の上のプログラムはインターフェースの向こう側に誰がいるか関知しない



身近な(？)例

- デバイスドライバ
  - プログラムを書く人はどんなデバイスが繋がっているか意識せずに書ける
  - DirectX
    - グラボがRadeonだろうがGeForceだろうがお構いなし
    - (・・・・でも、最近は2強だし、特定ボード用にチューンすることも珍しくないけどね・・・・)
- プラグイン

ブラウザのプログラムを書い「た」が・・・

```
/* URL をサーバに送る */
＃場合1：TCP/IP を使ってる場合　・・・・・
＃場合1の1：無線を使ってる場合　・・・・・・
＃場合1の1の1：暗号化無し無線使ってる場合・・・・・
＃場合1の1の2：WEPを使ってる場合・・・
＃場合1の1の2の1：802.11a を使ってる場合
  電波をこういう波形でこういう風に出して、こういう電波がきたらどうこう

＃場合1の1の2の2：802.11bを使ってる場合
  電波をこういう波形でこういう風に出して、こういう電波がきたらどうこう

＃場合1の2：有線を使ってる場合
＃場合1の2の1：Ethernetを使ってる場合
＃場合1の2の1の2：1000Base-T を使ってる場合
  電線にこういう風に電気を流して、こういう電気が来たら・・・
      :（中略）
＃場合2：IPXを使ってる場合・・・・
      :
      :
```

```
  :∧_∧
:((ﾟωﾟ`)):  な、何日も徹夜してやっと
    U　U)　　書き上げたお・・・・・・
    し ⌒J
```

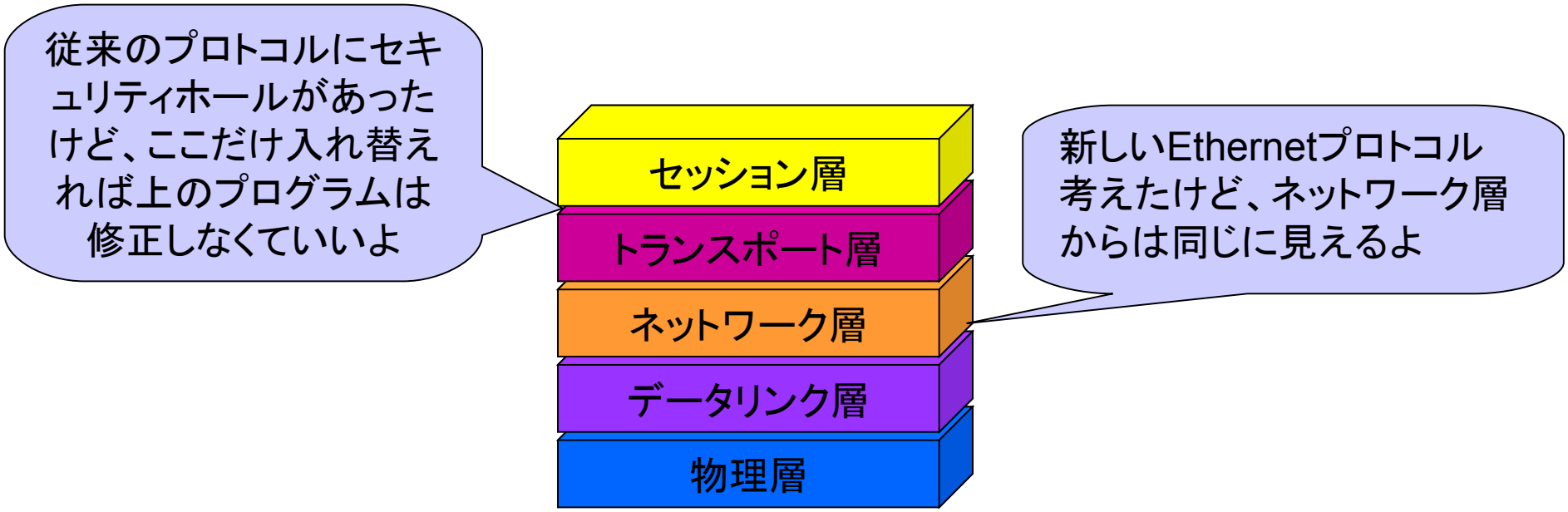
```
  :∧_∧　　！？
:((´ﾟ'ωﾟ)):
    U　U)
    し ⌒J
```

今日C社から新しい無線カードが出たそうなんで、対応してくださいね

```
     ∧_∧
y=-(　ﾟωﾟ) ・・;; ﾀｰﾝ
   |　　U
   し ⌒J
```

抽象化の利点(2)

- 階層ごとに入れ替えができれば楽

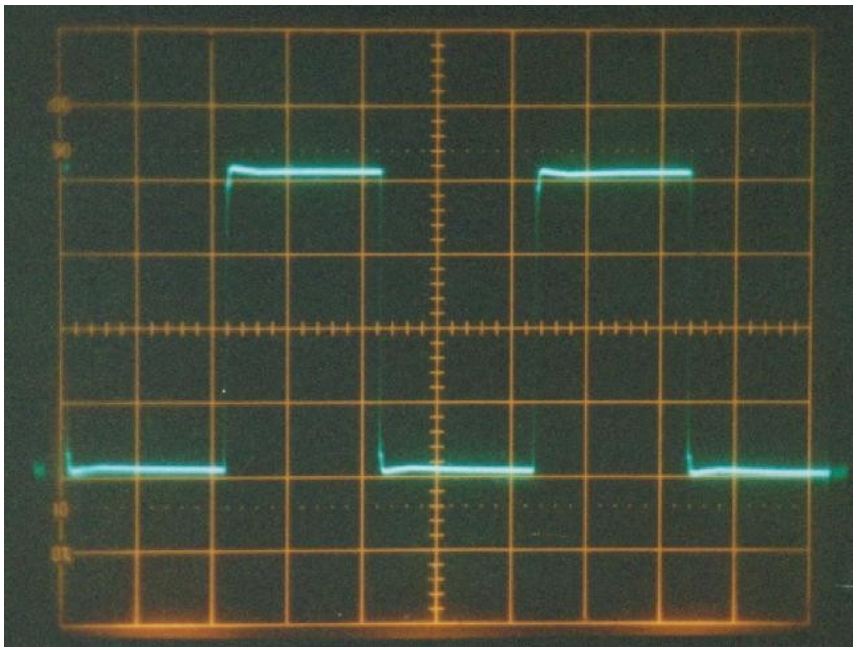


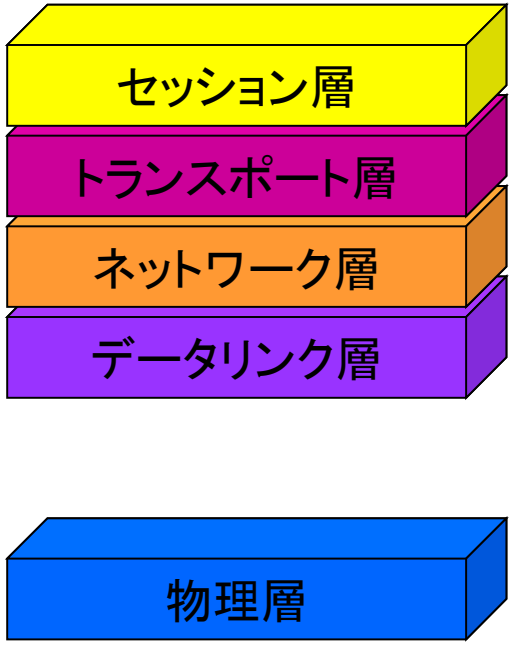
実際にどう動いている？

- 実際の通信を詳しく見てみましょう

実際にどう動いている？

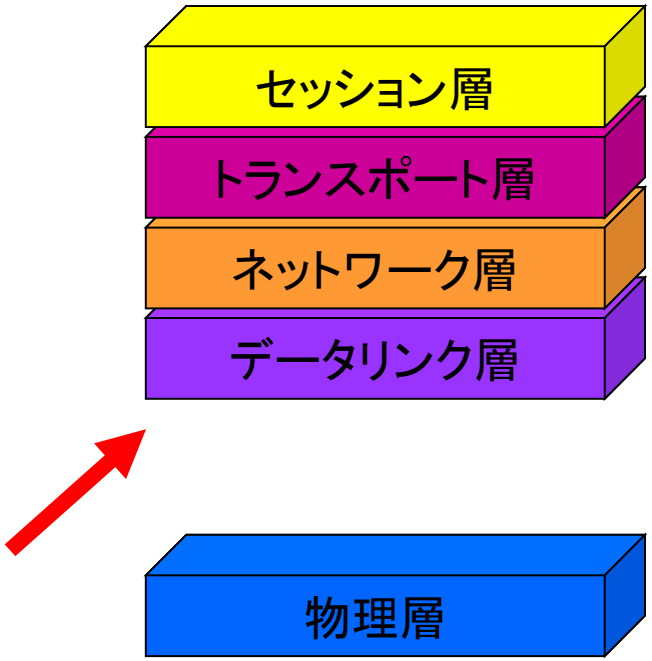
- 実際の通信を詳しく見てみましょう





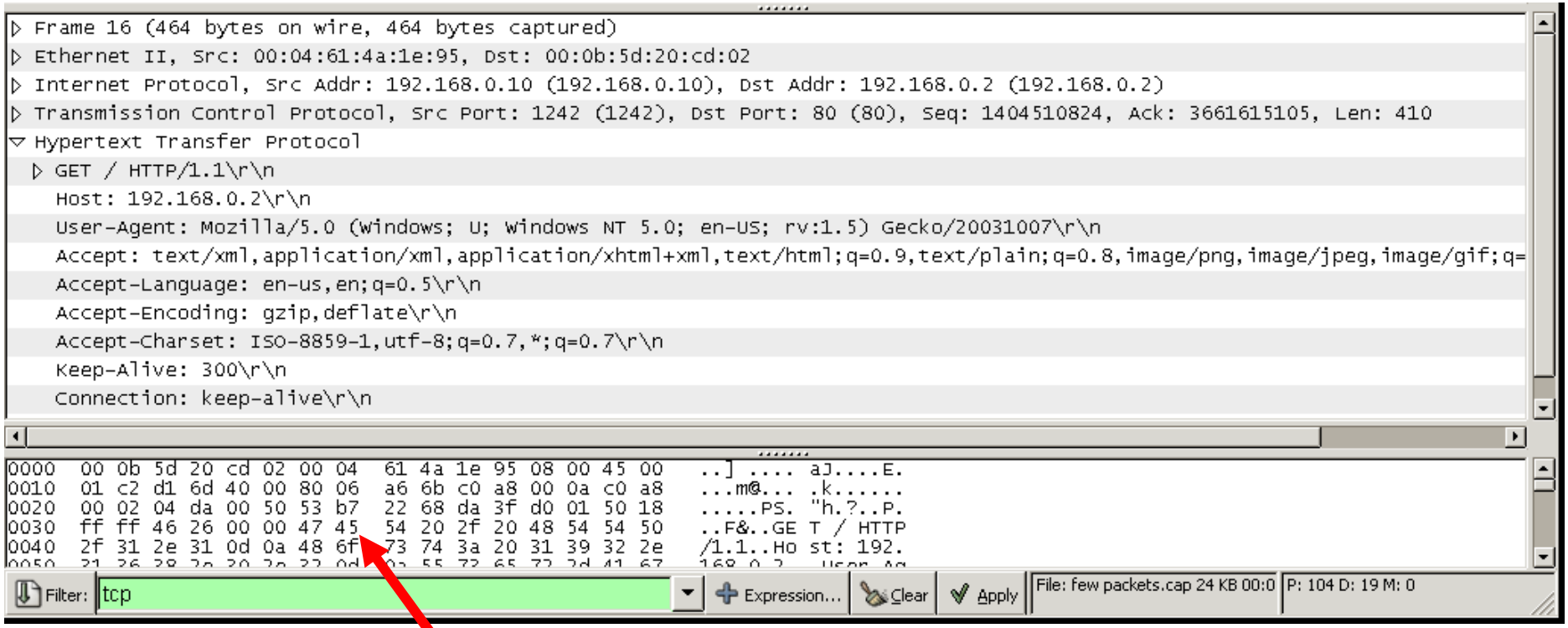
実際にどう動いている？

- さっきのはレイヤが低すぎたので、もうちょっと上の層で見てみましょう
- 実際どんなデータが流れている？



実際にどう動いている？

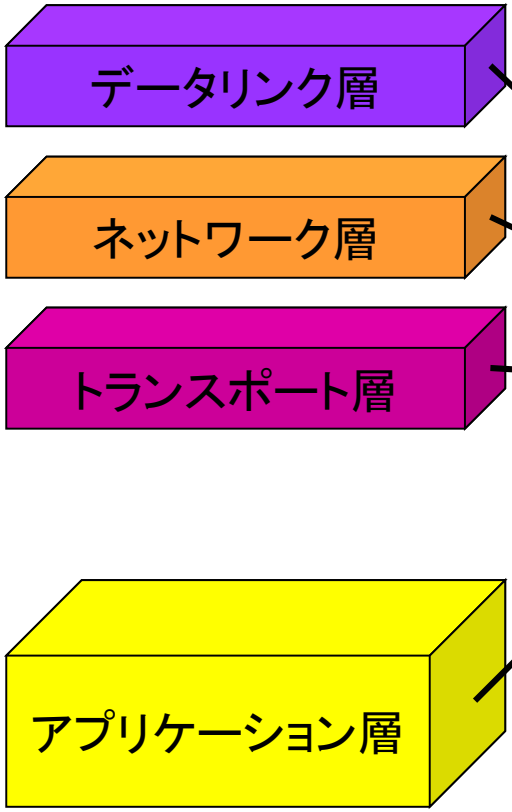
- Ethereal で見てみましょう
  - Ehtereal オフィシャルサイトのスクリーンショットより



実際に流れているデータはコレ（16進数）

実際にどう動いている？

下の層からくっついて一つのパケット（正確にはフレーム）を作り上げている




```
▷ Frame 16 (464 bytes on wire, 464 bytes captured)
▷ Ethernet II, Src: 00:04:61:4a:1e:95, Dst: 00:0b:
▷ Internet Protocol, Src Addr: 192.168.0.10 (192.1
▷ Transmission Control Protocol, Src Port: 1242 (1
▽ Hypertext Transfer Protocol
  ▷ GET / HTTP/1.1\r\n
    Host: 192.168.0.2\r\n
    User-Agent: Mozilla/5.0 (Windows; U; Windows N
    Accept: text/xml,application/xml,application/x
    Accept-Language: en-us,en;q=0.5\r\n
    Accept-Encoding: gzip,deflate\r\n
    Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
    Keep-Alive: 300\r\n
    Connection: keep-alive\r\n
```

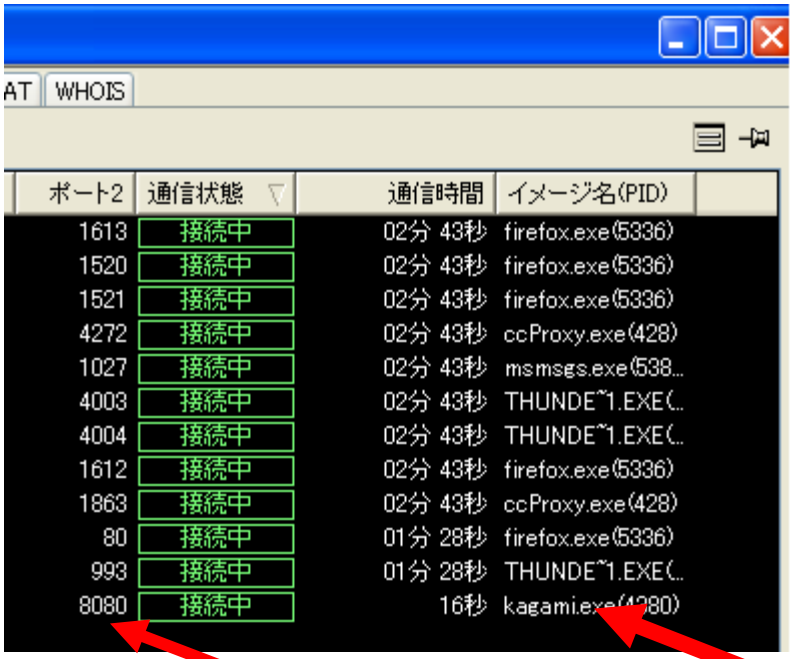
データリンク層って？

- Ethernetの層
- いわゆるLANの規格で、MACアドレスというアドレス体系を使って通信
  - MACアドレスはインターフェース（ポート）ごとに固定

ネットワーク層は？

- 主にインターネットプロトコル（IP）の層
  - みなさんおなじみの・・・・
  - 他のネットワーク層もあるけど・・・・
- IPアドレスを使って通信する

トランスポート層は何を見てるの？



TCP Monitor Plus
http://hp.vector.co.jp/authors/VA032928/

ポート番号とプログラムの対応関係を
見ることができる

OS はこういうデータベースを持っている
（厳密には一つにまとまってるとは限らない）

プログラム（プロセス）のリスト

ポート番号のリスト

Fportの場合

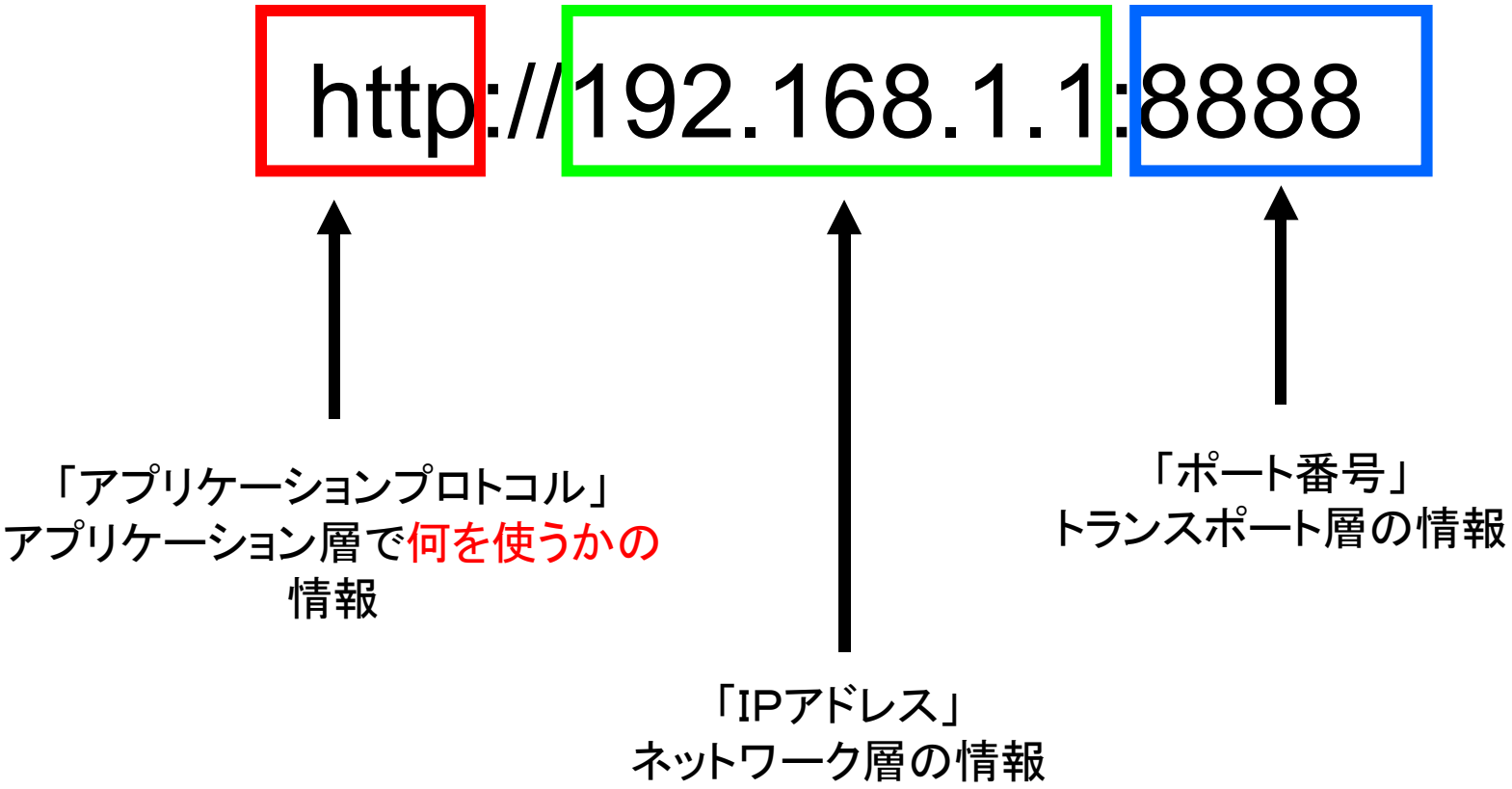
- TCPMonitor Plus は接続しているTCPポートのみ表示
- Fportは Listen している TCP/UDPポートも見れる
  - http://www.foundstone.com/us/resources/termsofuse.a

アプリケーション層は何よ？

- 様々なアプリケーションで使われるプロトコル
  - HTTP
  - FTP
  - SMTP(mail)
  - RTP
- 詳しい内容は省略
  - 上の層がデータを輸送することが目的とすれば、これより上の層はもっと具体的な要求とか
  - ページ要求、転送要求、発呼、etc….

実況Vではおなじみの「URL」とは

- いろいろな層の情報の集合体



そのほかいろいろなURL

mms://192.168.1.1:3214/

ftp://192.168.1.1/

省略されている場合は
デフォルトのポートを使う
「Well known port」

http://192.168.1.1/test.cgi?param=hoge

アプリケーション層「で」使われる情報
アプリケーション層の種類を示すものより
上のレイヤの情報

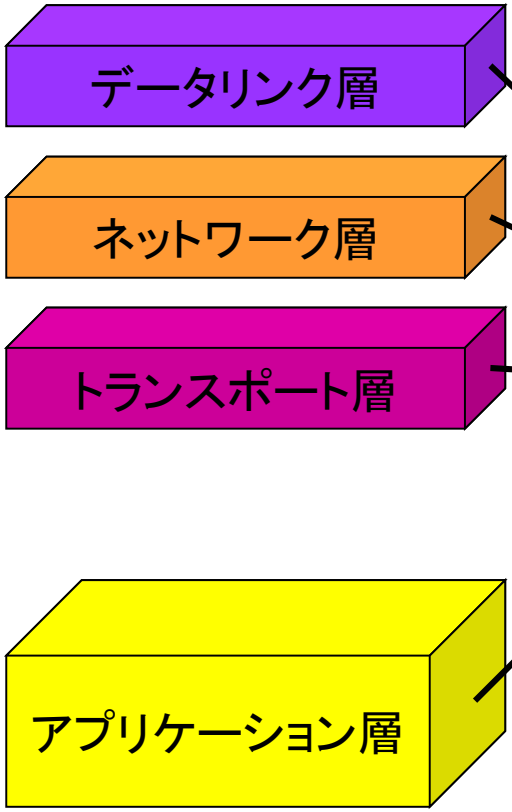
実はもっと上もあるんです

http://192.168.1.1/test.cgi?param=hoge

- 意味は「test.cgi というプログラムに param=hoge という情報を渡す」
  - つまり、さらに上の存在に渡すということで、webサーバアプリケーションはそれが何を意味しているか知らない
  - ここにも階層構造が

実際にどう動いている？

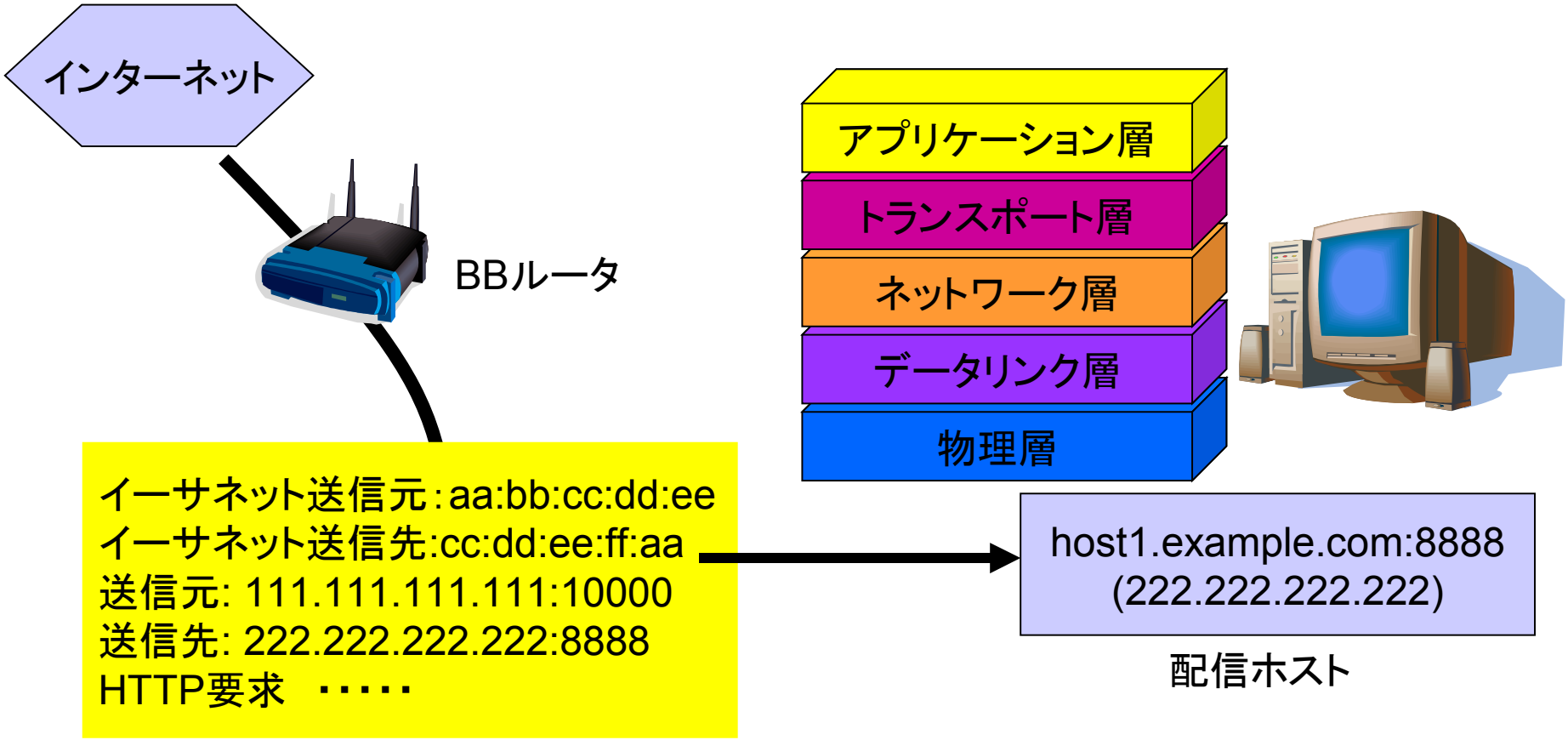
下の層からくっついて一つのパケット（正確にはフレーム）を作り上げている




```
▷ Frame 16 (464 bytes on wire, 464 bytes captured)
▷ Ethernet II, Src: 00:04:61:4a:1e:95, Dst: 00:0b:
▷ Internet Protocol, Src Addr: 192.168.0.10 (192.1
▷ Transmission Control Protocol, Src Port: 1242 (1
▽ Hypertext Transfer Protocol
  ▷ GET / HTTP/1.1\r\n
    Host: 192.168.0.2\r\n
    User-Agent: Mozilla/5.0 (Windows; U; Windows N
    Accept: text/xml,application/xml,application/x
    Accept-Language: en-us,en;q=0.5\r\n
    Accept-Encoding: gzip,deflate\r\n
    Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
    Keep-Alive: 300\r\n
    Connection: keep-alive\r\n
```

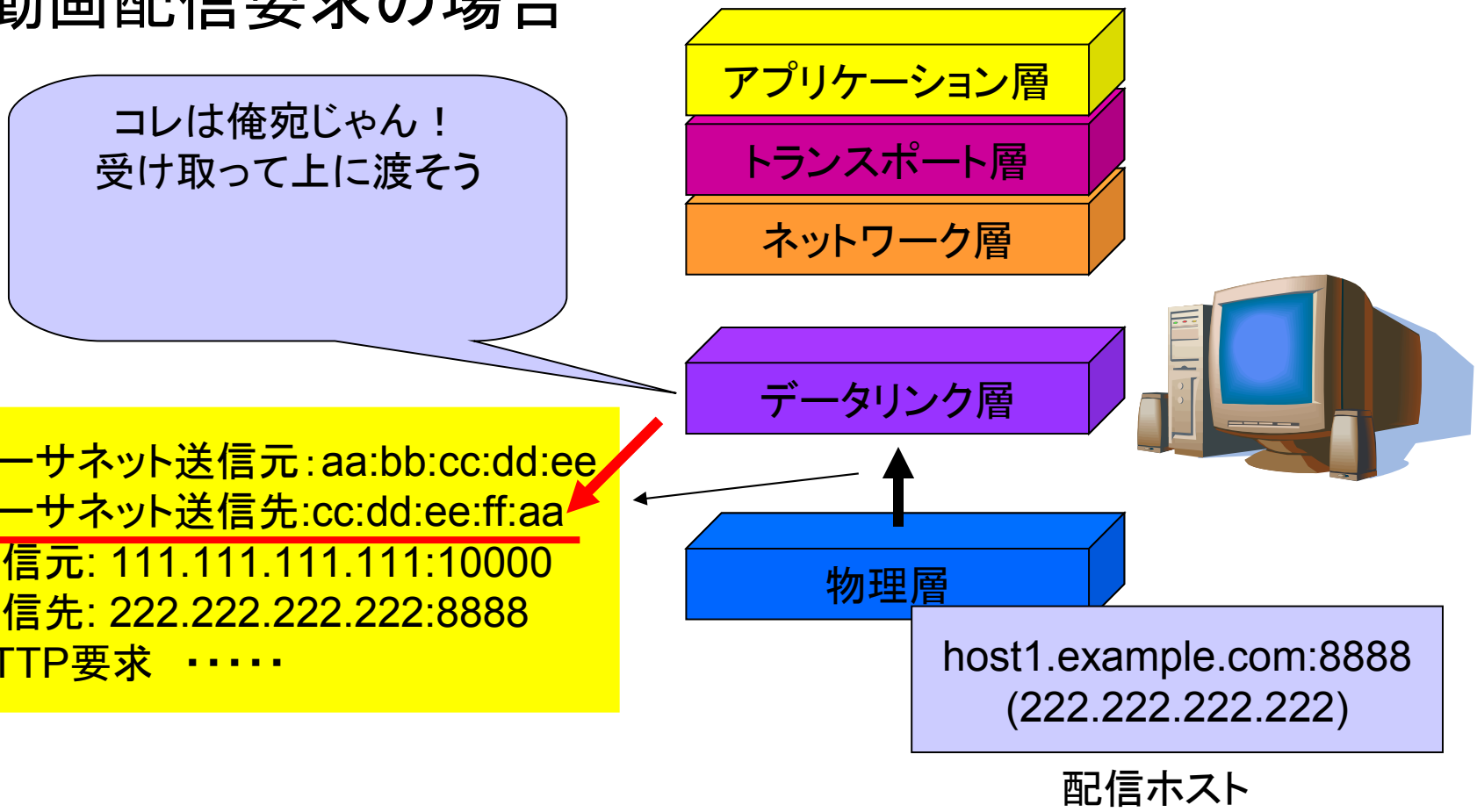
実際にどう動いている？

- とあるパケット：
  動画配信要求の場合



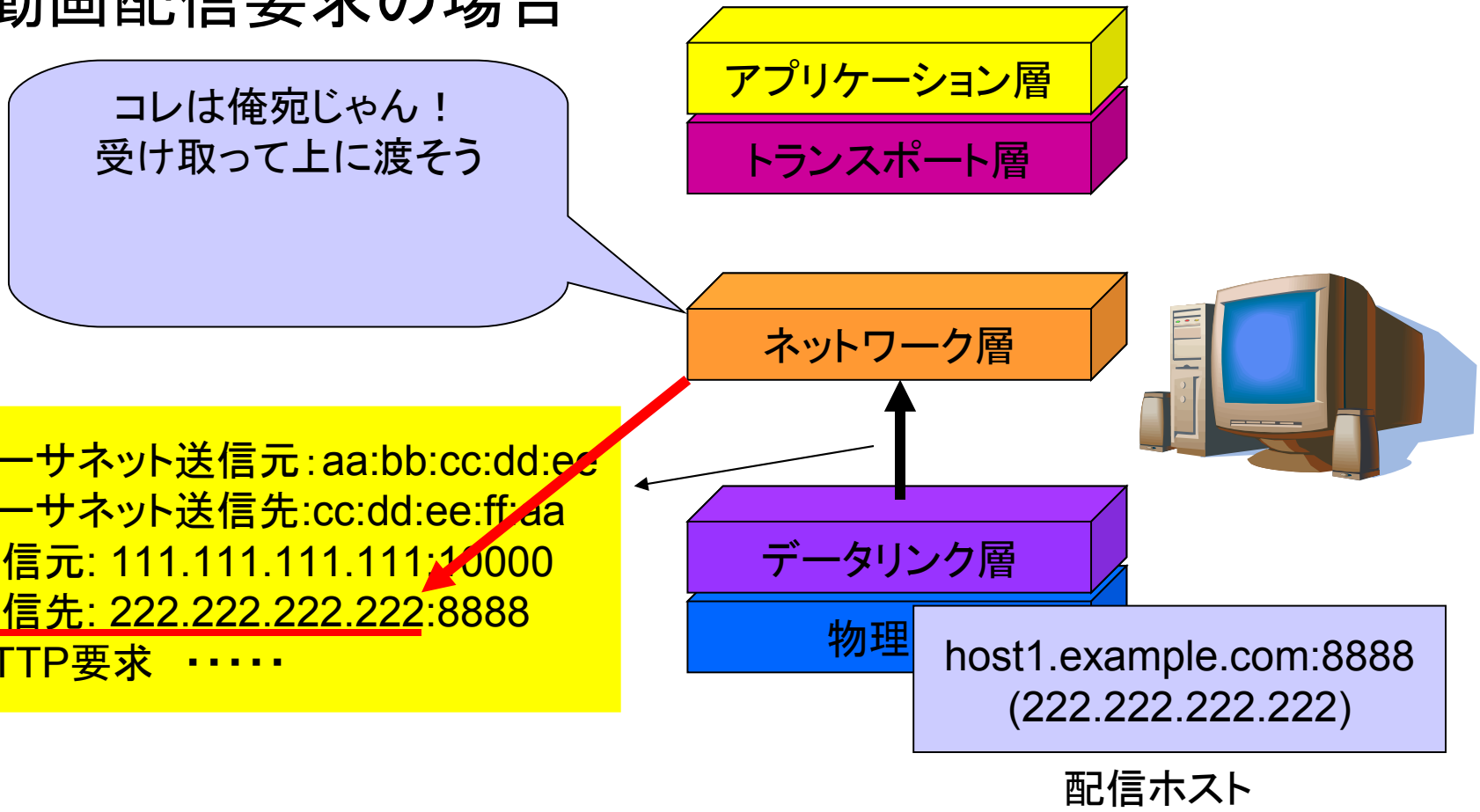
実際にどう動いている？

- とあるパケット：
  動画配信要求の場合



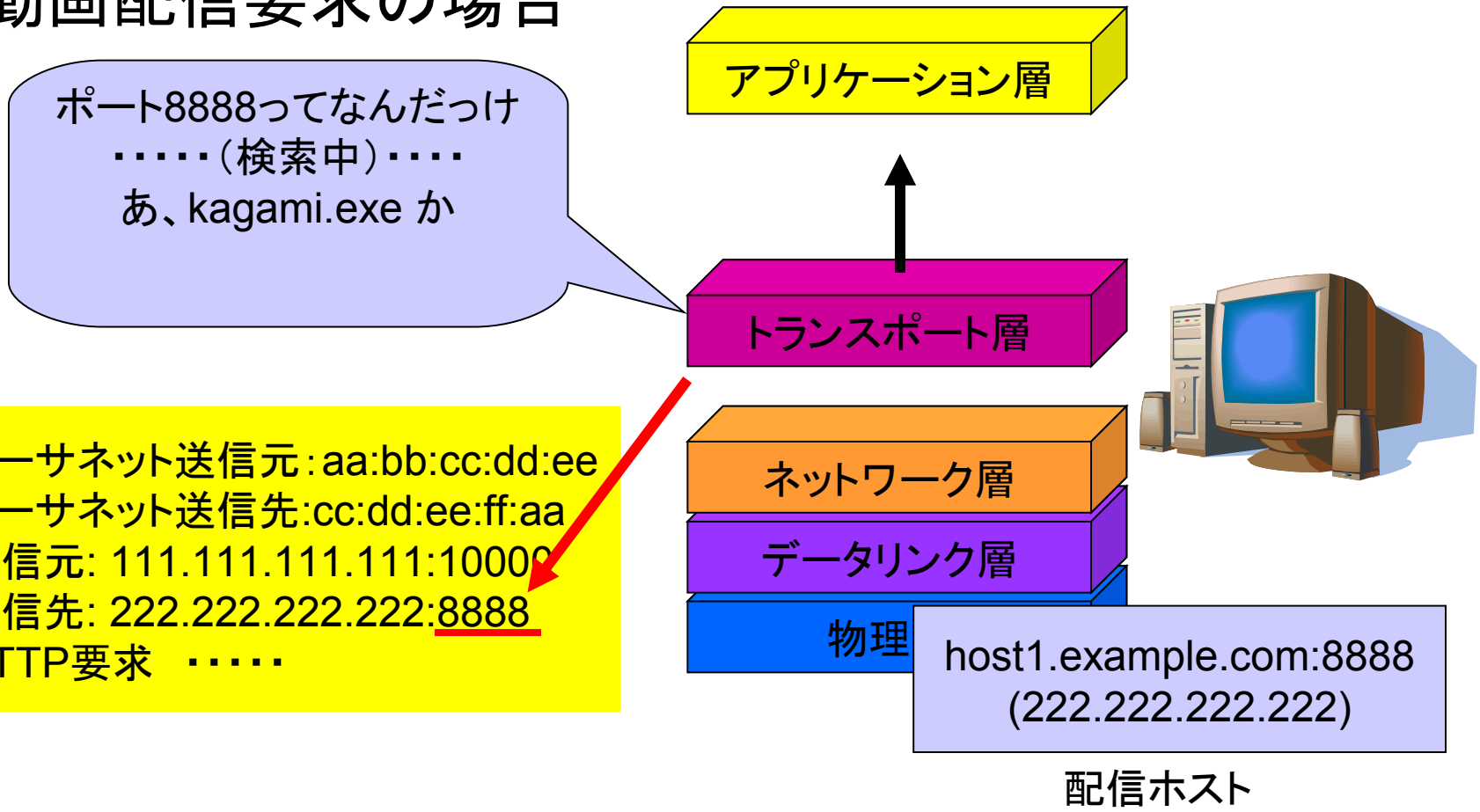
実際にどう動いている？

- とあるパケット：
  動画配信要求の場合



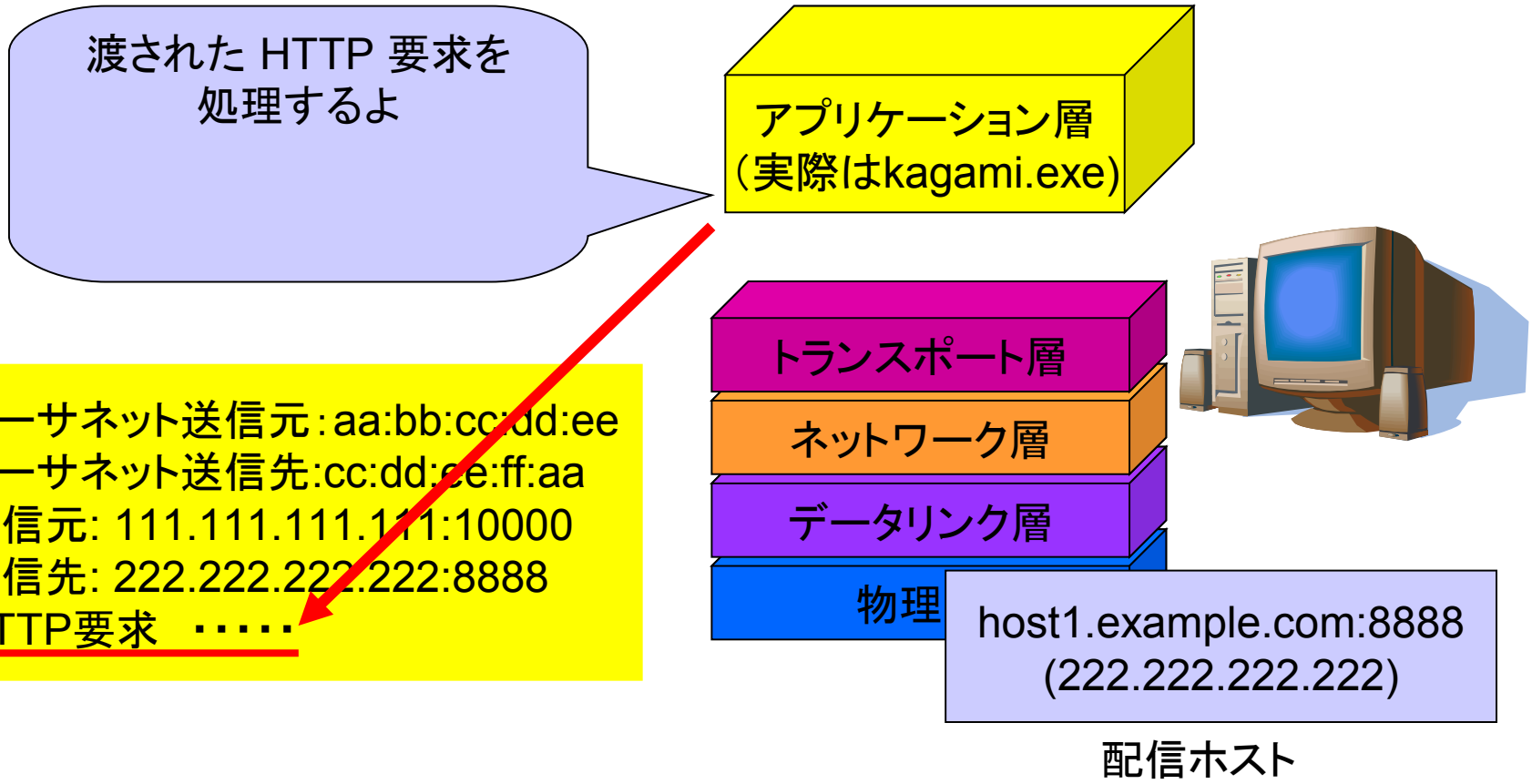
実際にどう動いている？

- とあるパケット：
  動画配信要求の場合



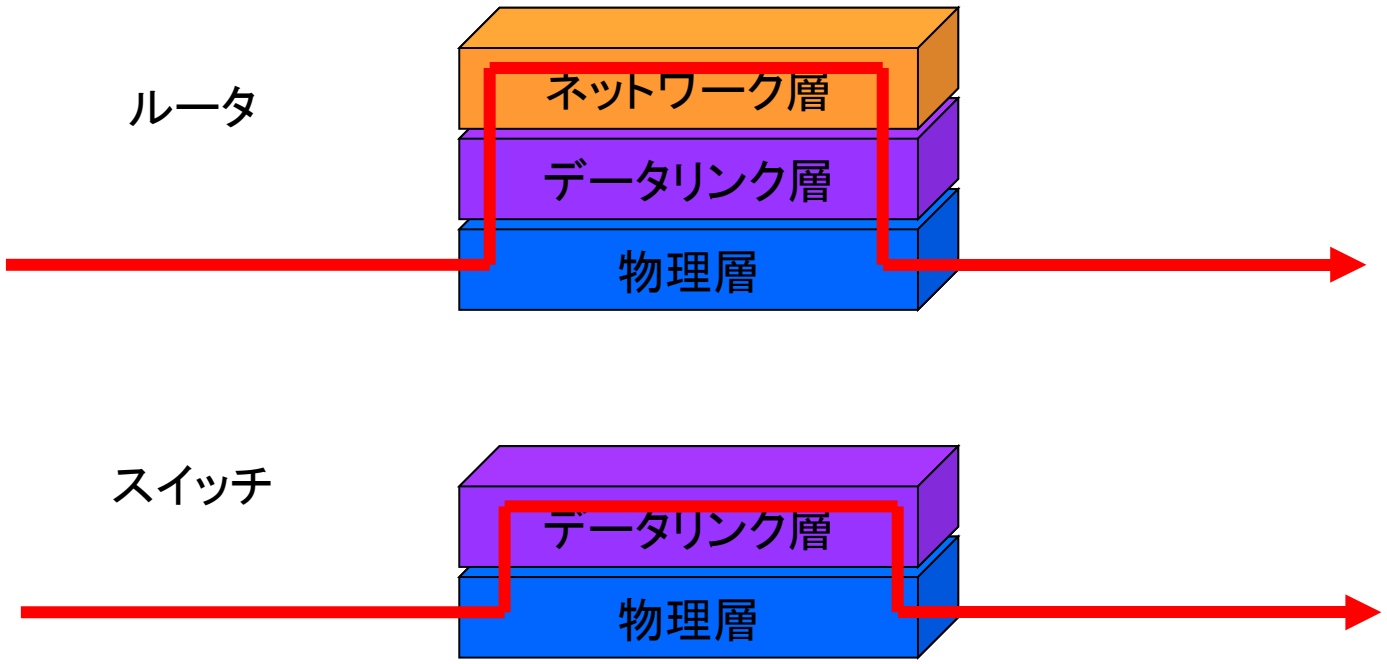
実際にどう動いている？

- とあるパケット:
  動画配信要求の場合



ルータとスイッチ（ハブ）の違い

- ルータはネットワーク層、スイッチはデータリンク層



まとめ

- インターネットの基本的構造
  - 回線交換とパケット交換
  - レイヤリングモデル

VMWare関係

FreeBSDの日本語環境

- 日本語の表示環境
  - ロカール
  - フォント
- 日本語の変換環境
  - 変換サーバ
    - Canna, Wnn, Anthy, egg
  - 変換クライアント
    - Kinput2, xwnmo, scim, uim

フォント

- フォントは基本的に2種類
  - ビットマップフォント
    - ビットマップデータ、固定サイズ
  - アウトラインフォント（ベクトルフォント）
    - ベクトルデータ、可変サイズ
- アウトラインフォントを扱うには専用のライブラリが必要
  - でも今では freetype が有力？

フォント

- 日本語フォントとベクトルフォントライブラリのインストール
  - 東風は・・・・・諸般の事情で使えなくなりました
  - IPA（モナー）フォントが有力？
  - デュアルブートしている場合は裏側の MS フォントを使用すると言う荒技もある