

ネットワークの基礎Ⅲ

0. 目次

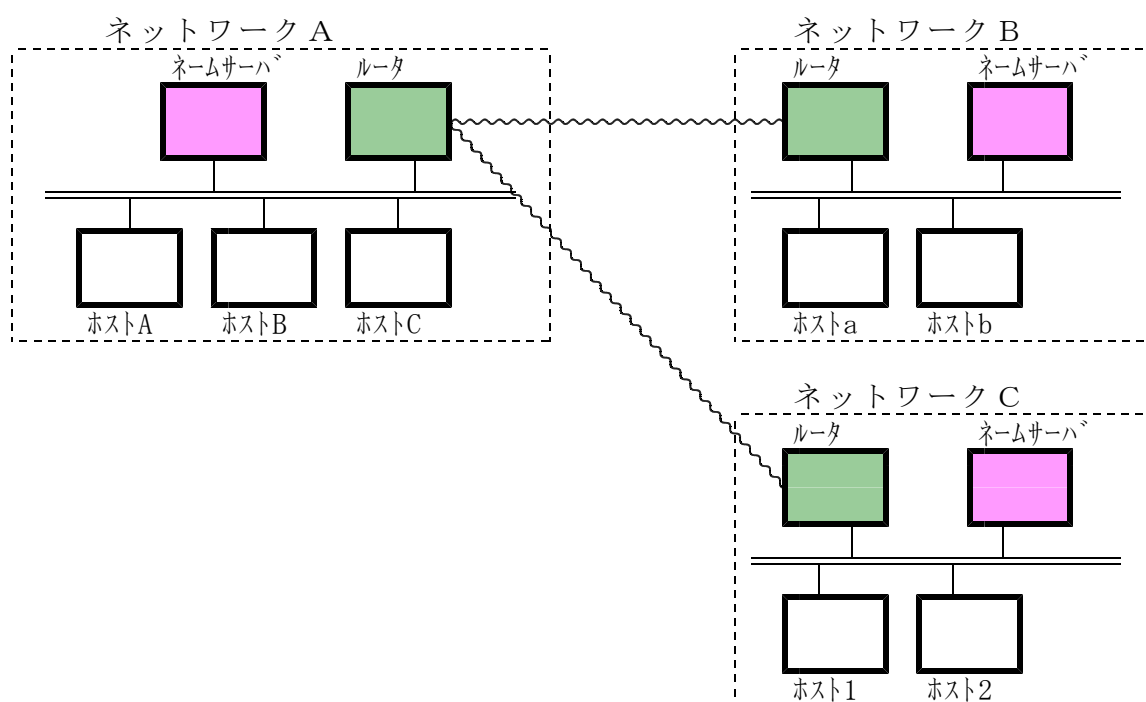
1. パケット

- 1. 1 パケットの構造
- 1. 2 IPアドレス
- 1. 3 リモートホストとのデータ転送時間の測定。
- 1. 4 パケットがリモートホストにたどり着く経路
- 1. 5 パケットの経路制御

2. TCP/IP

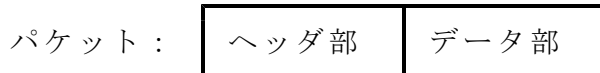
1. パケット

送信元ホストから送り出されるデータは、いくつかの**パケット**（小包）に分割されてネットワーク上に送り出される。このパケットには、送信元IPアドレス、送信先IPアドレス、データが書かれており、ネットワーク上の多くの**ルータ**（パケットに書かれたIPアドレスを見て、次の送り先として適切なコンピュータを見つけるコンピュータ。すなわち、異なるネットワークアドレスを持つネットワーク間の相互通信を行う。そのために必要な情報は適当な時間間隔で収集し経路制御表に保存している）を経由して送信先ホストに届けられる。送信先ホストでは、届いたパケットをまとめてデータを取り出す。まだ届いていないパケット（パケットには、パケットの総数と順番の情報がある）があれば、送信元IPアドレスを基にデータの再送を要求する。



1. 1 パケットの構造

パケットはヘッダ部とデータ部から構成される。ヘッダ部には、送信先 IP アドレスや送信元 IP アドレスなど、パケットを相手のコンピュータに届けるための情報が書かれる。データ部には相手のコンピュータに届けたい情報が書かれる。



1. 2 IP アドレス

IP アドレスは32ビットからなる。2進表記ではわかりにくいため、8ビットずつピリオドで区切って10進数で表す。

2進表記 : 11000000 00000011 00000001 00001111

10進数ドット表記 : 192. 3. 1. 15

32ビットはネットワーク部とホスト部に分かれる。157. 80. 82. 0/24は、ネットワーク部として24ビット使うことを意味する。

ネットワーク部	ホスト部
***** ***** *****	*****

157. 80. 82. 0	ネットワークのアドレス ネットワーク自体を指す
157. 80. 82. 1~157. 80. 82. 254	コンピュータのアドレス 254台のコンピュータを接続 できる。
157. 80. 82. 255	ブロードキャストアドレス

ブロードキャストアドレスは、同一ネットワーク内のすべての IP アドレスに対して同じパケットを送信する場合に使う。ネットワーク番号+ホスト番号(すべて1)と書く。157. 80. 82. 0/24の場合、157. 80. 82. 255となる。

IP アドレス (127. 0. 0. 1) には、localhost というホスト名がつけられ、ホスト内で通信をするときに用いられる。

```

1 nslookupコマンドでlocalhostのIPアドレスを調べる。
2 C:\WINDOWS\system32> nslookup localhost
3 Server: hit000.ipc.ibaraki.ac.jp
4 Address: 157.80.11.22
5
6 Name: localhost
7 Address: 127.0.0.1

```

(参考) 現在使われている IP アドレスはバージョン 4 で IPv4 と表す。次世代の IP アドレスは128ビットの長さをもつ。IPv6 と表す。IPv6 では IPv4 で生じているアドレスの枯渇問題が解決される。IPv4 では約 4.3×10^9 個、IPv6 では約 3.4×10^{38} 個の IP アドレス数となる。

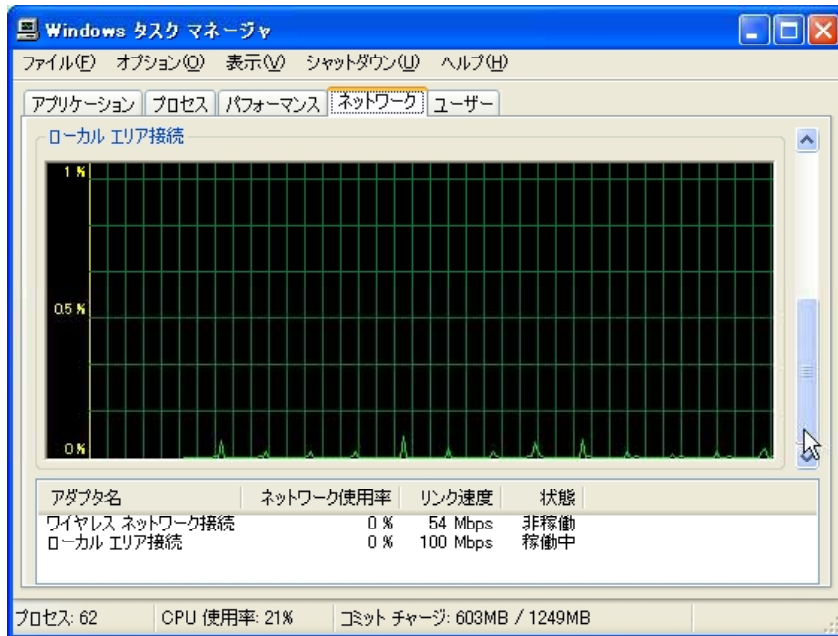
●WindowsXPでネットワーク状況の確認

①タスクバーを右クリックし、**タスクマネージャー**を選択する。

「Windowsタスクマネージャ」ウィンドウが表示される。

ネットワークタブを選択する。

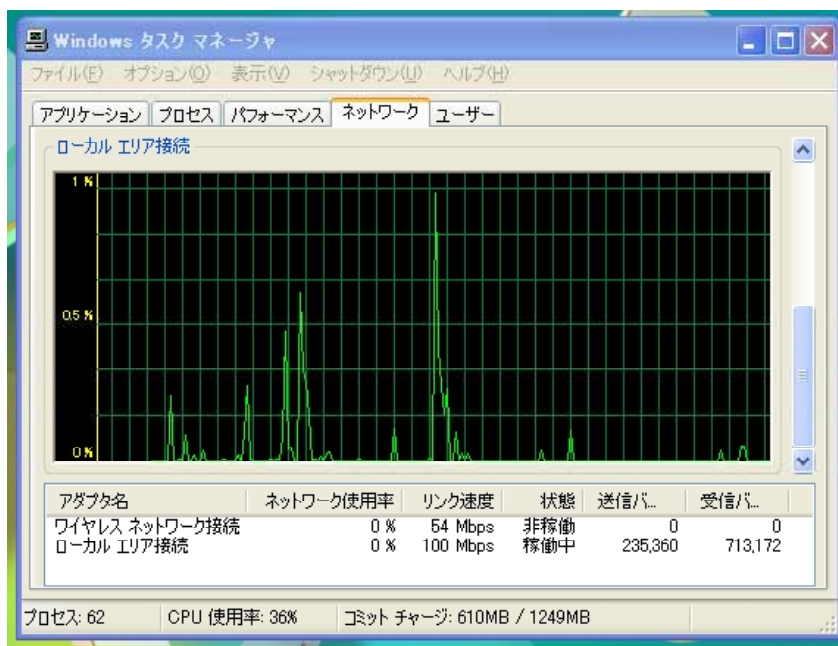
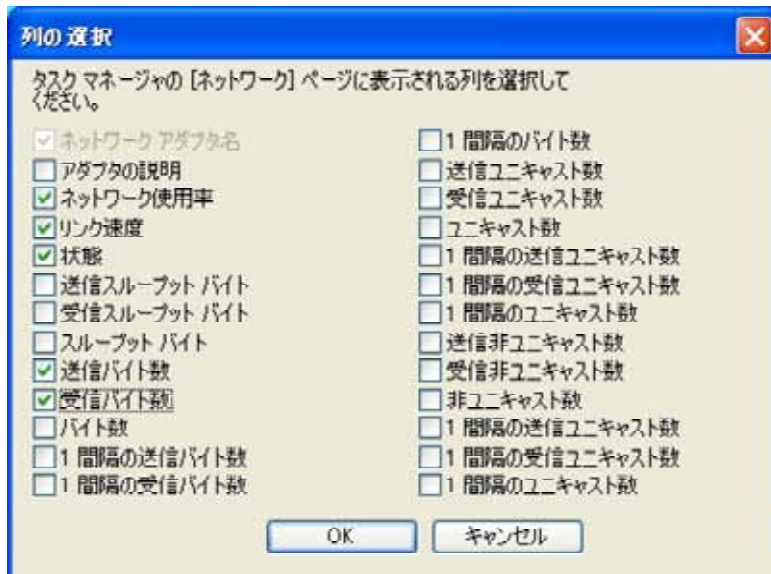
最初、ワイヤレスネットワーク接続が表示されているので、右のスクロールバーを下に移動し、ローカルエリア接続の表示に変える。



メニューバーから**表示**を選択し、**列の選択**を選択する。

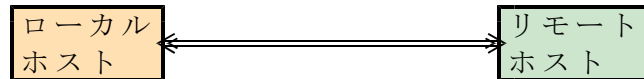


- ② 「列の選択」 ウィンドウが表示される。
 送信バイト数 と 受信バイト数 にチェックを入れる。
 OK ボタンをクリックする。



1. 3 リモートホストとのデータ転送時間の測定。

ping コマンドを使いローカルホストとリモートホスト間のデータ転送時間（往復遅延時間）の測定を行う。この結果から、リモートホストが近いか遠いかわかる。リモートホストは、ホスト名またはIPアドレスで指定する。



time	パケットを送出してから帰ってくるまでの時間（単位はミリ秒）。往復遅延時間という。
ttl (time to live)	パケットを送出したコンピュータが初期値（32または64または255）をパケットに設定し、途中ルータを経由するごとに1引かれる。ttlの値が0になると、そのパケットは破棄される。

```

1 ローカルホストからリモートホスト(hcs.ipc.ibaraki.ac.jp)へpingコマンド
2 を実行。
3 C:\WINDOWS\system32>ping hcs.ipc.ibaraki.ac.jp
4 Pinging hcs.ibaraki.ac.jp [157.80.75.61] with 32 bytes of data:
5
6 Reply from 157.80.75.61: bytes=32 time<1ms TTL=253
7 Reply from 157.80.75.61: bytes=32 time<1ms TTL=253
8 Reply from 157.80.75.61: bytes=32 time<1ms TTL=253
9 Reply from 157.80.75.61: bytes=32 time<1ms TTL=253
10
11 Ping statistics for 157.80.75.61:
12     Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
13 Approximate round trip times in milli-seconds:
14     Minimum = 0ms, Maximum = 0ms, Average = 0ms
15 ローカルホストからリモートホスト(acs.ipc.ibaraki.ac.jp)へpingコマンド
16 を実行。
17 C:\WINDOWS\system32>ping acs.ipc.ibaraki.ac.jp
18 Pinging acs.ibaraki.ac.jp [157.80.77.61] with 32 bytes of data:
19
20 Reply from 157.80.77.61: bytes=32 time=2ms TTL=251
21 Reply from 157.80.77.61: bytes=32 time=2ms TTL=251
22 Reply from 157.80.77.61: bytes=32 time=2ms TTL=251
23 Reply from 157.80.77.61: bytes=32 time=2ms TTL=251
24
25 Ping statistics for 157.80.77.61:
26     Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
27 Approximate round trip times in milli-seconds:
28     Minimum = 2ms, Maximum = 2ms, Average = 2ms
29
  
```

```

30 pingコマンドの使用法は、pingと入力すると表示される。
31 C:¥WINDOWS¥system32>ping
32 Usage: ping [-t] [-a] [-n count] [-l size] [-f] [-i TTL] [-v TOS]
33           [-r count] [-s count] [[-j host-list] | [-k host-list]]
34           [-w timeout] target_name
35 Options:
36     -t           Ping the specified host until stopped.
37                 To see statistics and continue - type Control-Break;
38                 To stop - type Control-C.
39     -a           Resolve addresses to hostnames.
40     -n count     Number of echo requests to send.
41     -l size      Send buffer size.
42     -f           Set Don't Fragment flag in packet.
43     -i TTL       Time To Live.
44     -v TOS       Type Of Service.
45     -r count     Record route for count hops.
46     -s count     Timestamp for count hops.
47     -j host-list Loose source route along host-list.
48     -k host-list Strict source route along host-list.
49     -w timeout   Timeout in milliseconds to wait for each reply.

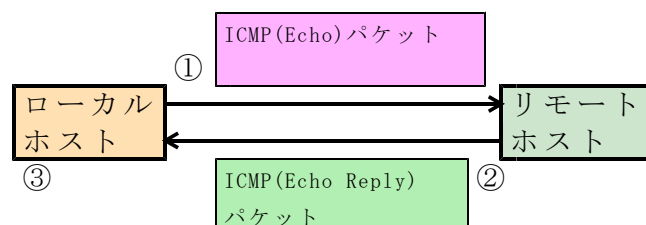
```

- ・途中でパケットが消失することもある。

●pingの仕組み

ICMP(Internet Control Message Protocol)という制御プロトコルを使う。

- ①ローカルホストからリモートホストに向けて、ICMP(Echo)パケットを送信する。
- ②ICMP(Echo)パケットを受信したリモートホストは、ローカルホストに向けて、ICMP(Echo Reply)パケットを送信する（このように動作するように決められている）。
- ③ローカルホストは、ICMP(Echo Reply)パケットが返ってきた場合、リモートホストが動作していると判断し、いつまで待っても返ってこない場合、動作していないか、存在しないと判断する。



1. 4 パケットがリモートホストにたどり着く経路

パケットがネットワークを通過してリモートホストにたどり着く経路を `tracert` コマンドを使って調べる。往復時間（ローカルホストから各行のルータまで）を3回調べ表示する。5秒以上応答がない場合、*が表示される。機器を経由することをhop(ホップ)という。したがって、30 hops maxは最大30ホップまで調べたことを意味する。

●WindowsXPから `tracert` コマンドの実行

コマンドプロンプトを開き、`tracert` コマンドを実行する。

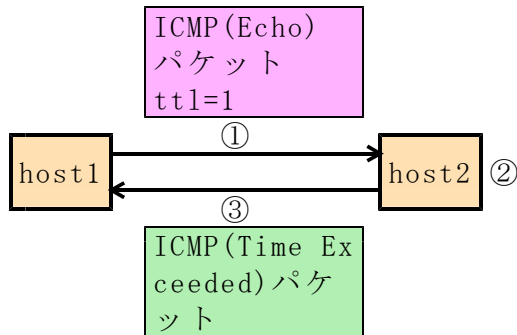
```

1 ローカルホストからリモートホスト(hcs.ipc.ibaraki.ac.jp)へtracertコ
2 マンドを実行。
3 C:\WINDOWS\system32>tracert hcs.ipc.ibaraki.ac.jp
4 Tracing route to hcs.ibaraki.ac.jp [157.80.75.61]
5 over a maximum of 30 hops:
6   1    <1 ms    <1 ms    <1 ms    157.80.82.11
7   2    <1 ms    <1 ms    <1 ms    157.80.2.12
8   3    <1 ms    <1 ms    <1 ms    hcs.ibaraki.ac.jp [157.80.75.61]
9 Trace complete.
10 ローカルホストからリモートホスト(mcs.ipc.ibaraki.ac.jp)へtracertコ
11 マンドを実行。
12 C:\WINDOWS\system32>tracert mcs.ipc.ibaraki.ac.jp
13 Tracing route to mcs.ibaraki.ac.jp [157.80.76.61]
14 over a maximum of 30 hops:
15   1    <1 ms    <1 ms    <1 ms    157.80.82.11
16   2    <1 ms    <1 ms    <1 ms    157.80.2.11
17   3    1 ms     1 ms     1 ms     157.80.1.12
18   4    1 ms     1 ms     1 ms     157.80.3.12
19   5    <1 ms    <1 ms    <1 ms    mcs.ibaraki.ac.jp [157.80.76.61]
20 Trace complete.
21 tracertコマンドの使用法は、tracertと入力すると表示される。
22 C:\WINDOWS\system32>tracert
23 Usage: tracert [-d] [-h maximum_hops] [-j host-list] [-w timeout]
24         target_name
25 Options:
26     -d                Do not resolve addresses to hostnames.
27     -h maximum_hops  Maximum number of hops to search for target.
28     -j host-list      Loose source route along host-list.
29     -w timeout        Wait timeout milliseconds for each reply.

```

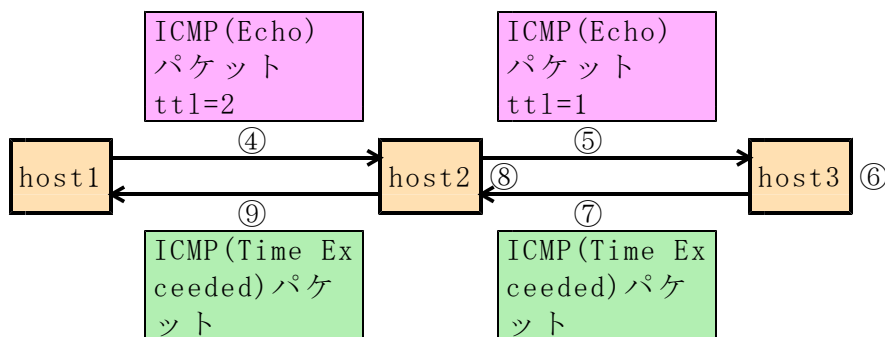

● tracerouteコマンドの仕組み

ICMP(Internet Control Message Protocol)という制御プロトコルを使う。
 ホスト(host1)からホスト(host3)までの経路を調べる。



- ① tracerouteコマンドは、まずttlを1としてICMP(Echo)パケットをネットワークに送出する。
- ② 1ホップ目のルータ(host2)は、受信したICMP(Echo)パケットのttlから1を引く。0になると、ICMP(Echo)パケットを破棄する。
- ③ 同時に、ICMP(Time Exceeded)パケットをネットワークに送出する。
 ICMP(Time Exceeded)パケットには、ルータ(host2)のIPアドレスが含まれる。
 この結果、送信元ホスト(host1)では、1ホップ目ルータ(host2)のIPアドレスが得られる。

この操作をあと2回繰り返す。



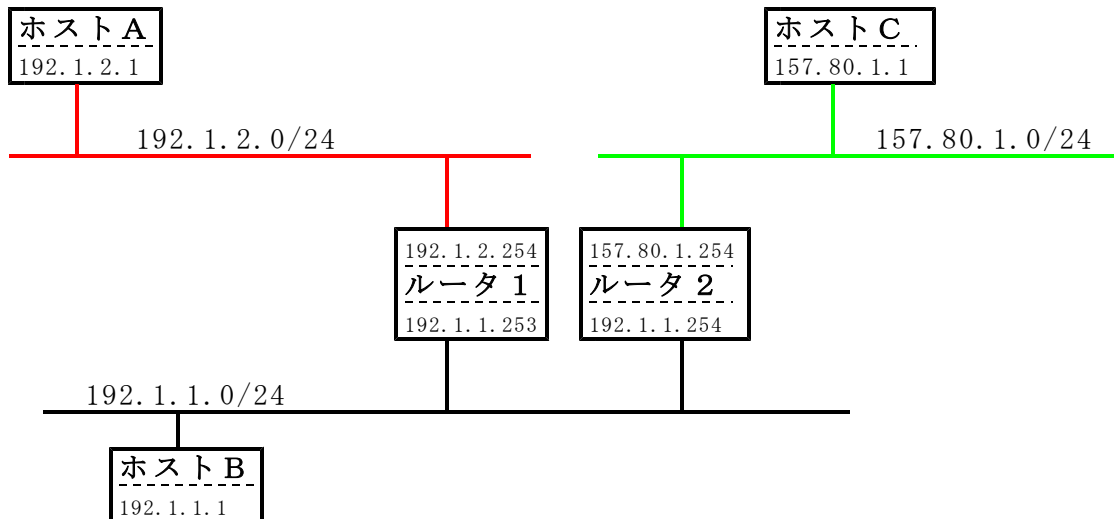
- ④ tracerouteコマンドは、ttlを2としてICMP(Echo)パケットをネットワークに送出する。
- ⑤ 1ホップ目のルータ(host2)は、受信したICMP(Echo)パケットのttlから1を引き1とし、ネットワークに送出する。
- ⑥ 2ホップ目のルータ(host3)は、受信したICMP(Echo)パケットのttlから1を引く。0となるので、ICMPパケットを破棄する。
- ⑦ 同時に、ICMP(Time Exceeded)パケットをネットワークに送出する。
 ICMP(Time Exceeded)パケットには、ルータ(host3)のIPアドレスが含まれる。
- ⑧ 1ホップ目のルータ(host2)は、受信したICMP(Time Exceeded)パケットをネットワークに送出する。
- ⑨ この結果、送信元ホスト(host1)では、2ホップ目ルータ(host3)のIPアドレスが得られる。

この操作をあと2回繰り返す。

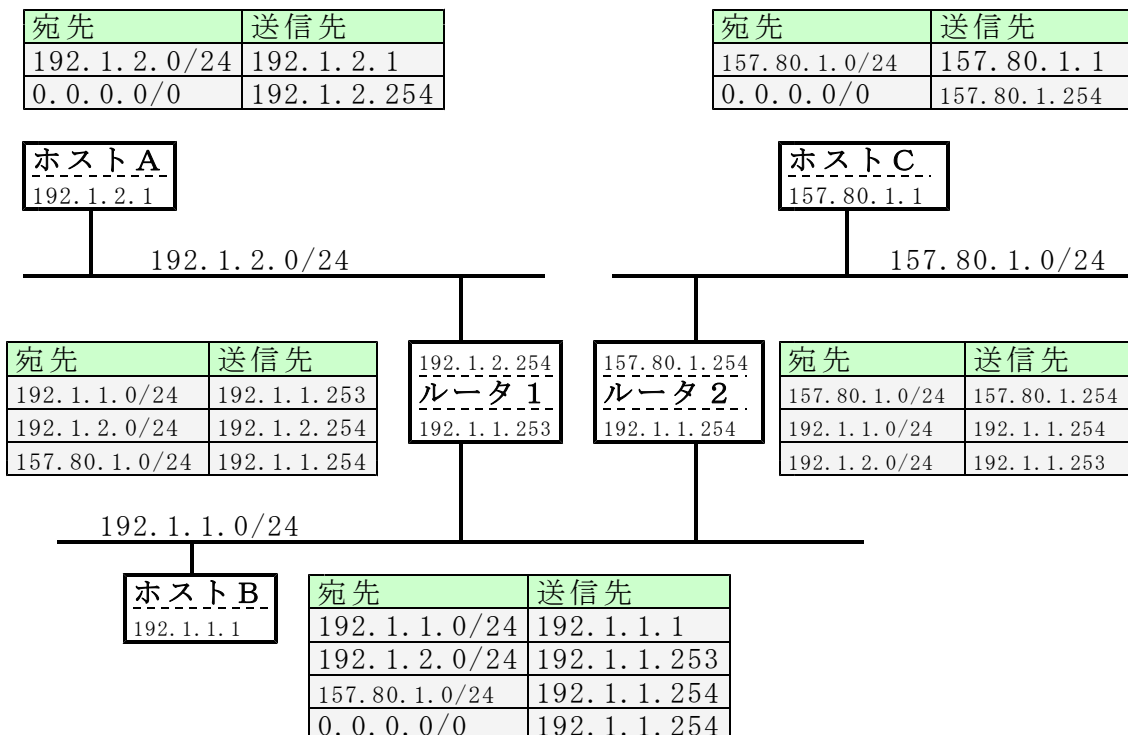
1. 5 パケットの経路制御

つぎのネットワークでパケットの経路制御を考える。各ホストにはIPアドレスがひとつ、ルータは接続するネットワーク数のIPアドレスを持つ。

●ネットワーク

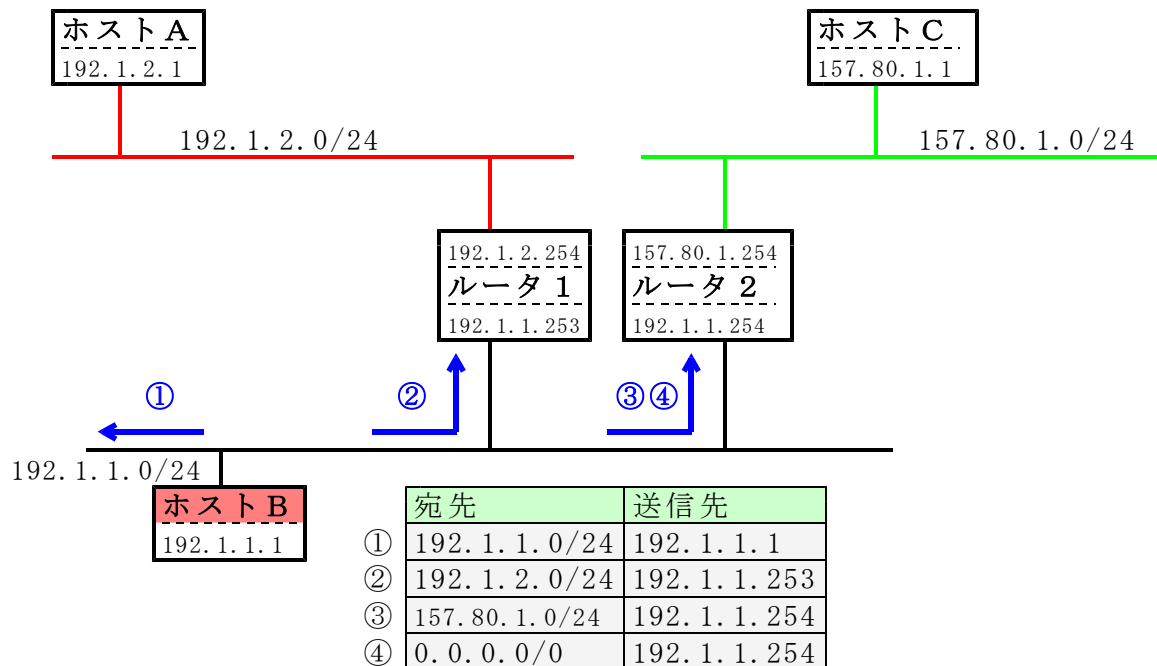


●ネットワーク + 経路制御表



経路制御表は手作業で作成する方法とルーティングプロトコルという機能を用いて自動的に作成する方法がある。

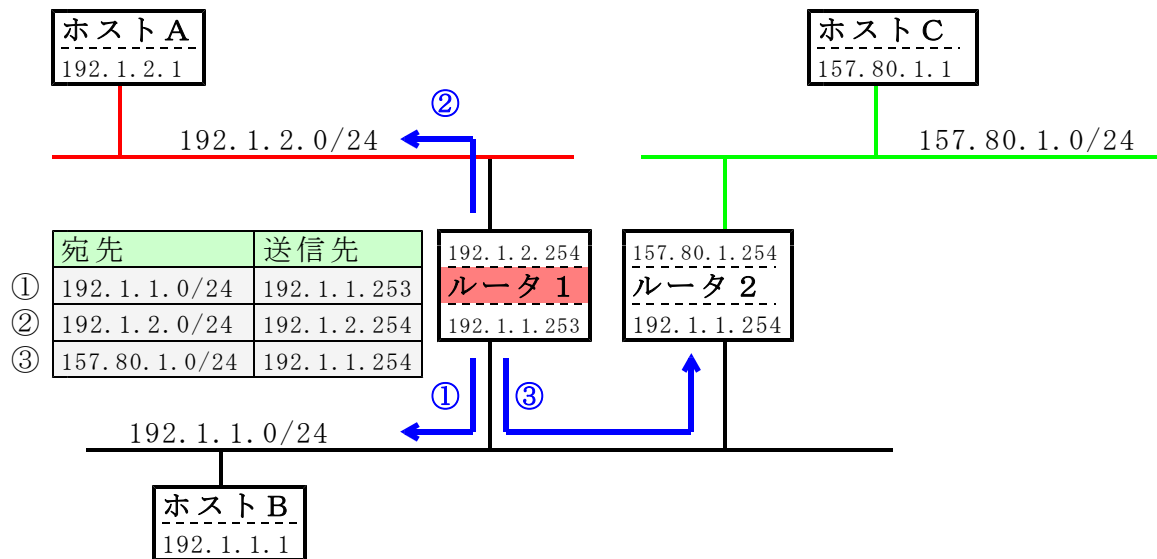
- ホスト B は経路制御表を用いて、パケットの送信先ホストを決める。



ホスト B の経路制御表で、

- ① パケットの宛先が自分が接続しているネットワーク (192.1.1.0/24) の場合、送信先を自分自身の IP アドレスと指定している。この結果、パケットを同じネットワークのホストに直接送信する。
- ② パケットの宛先が他のネットワーク (192.1.2.0/24) の場合、ルータ 1 (192.1.1.253) へパケットを送信する。
- ③ パケットの宛先が他のネットワーク (157.80.1.0/24) の場合、ルータ 2 (192.1.1.254) へパケットを送信する。
- ④ パケットの宛先が経路制御表のどの宛先とも一致しない場合、0.0.0.0/0 に対応するルータ 2 (192.1.1.254) に送信する。

- ルータ 1 は経路制御表を用いて、パケットの送信先ホストを決める。



ルータ 1 の経路制御表で、

- ① 受信したパケットの宛先がネットワーク (192.1.1.0/24) の場合、ネットワーク (192.1.1.0/24) のホストに送信する。
- ② 受信したパケットの宛先がネットワーク (192.1.2.0/24) の場合、ネットワーク (192.1.2.0/24) のホストに送信する。
- ③ 受信したパケットの宛先がネットワーク (157.80.1.0/24) の場合、ルータ 2 (192.1.1.254) へパケットを送信する。

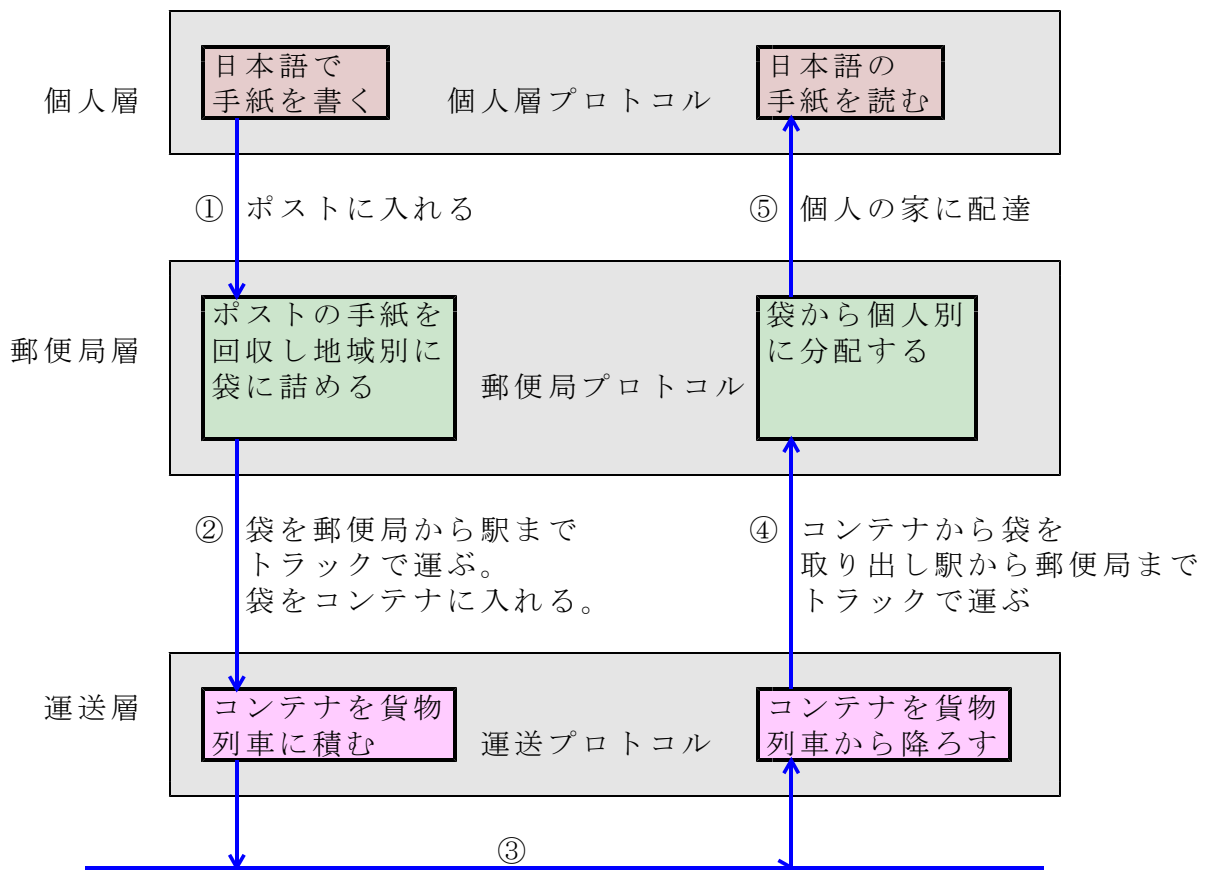
2. TCP / IP

コンピュータなどのハードウェアやソフトウェアを組み合わせるネットワークを構成する。このとき、コンピュータ間でうまく通信を行うには、相互の取り決め（データの表現法、データを送る順序、エラーの回復、相手コンピュータにデータを届ける方法など）をきちんと定義しておかないと混乱する。この取り決めを**プロトコル**という。

ところが、プロトコルを勝手に決めたのでは、より複雑になってしまう危険がある。そこで、プロトコルの定義をネットワーク全体が見通しよくなるように、階層的に分けて決めている。すなわち、目的や機能に応じて明確にグループに分ける。このようにしておくこと、全体の構造と機能を守る限り、個々のプロトコルの開発・保守に専念できる。階層的に決められたプロトコルの集まりで構成されるネットワーク全体を**ネットワークアーキテクチャ**という。

●プロトコルの階層化を手紙を例に示す。

3つの階層（個人層、郵便局層、運送層）とそれぞれのプロトコルで考える。



個人層プロトコル	日本語を使うとか、書類の書式などの約束事を決める。
郵便局プロトコル	手紙の宛先と取り扱い郵便局との対応、速達便・普通便などの約束事を決める。
運送層プロトコル	郵便局と最寄りの駅との対応などの約束事を決める。

●①②③④⑤におけるデータ

①

発信者住所	本文
受信者住所	

②

送り元郵便局 送り先郵便局	発信者住所	本文
	受信者住所	

③

出発駅 到着駅	送り元郵便局	発信者住所	本文
	送り先郵便局	受信者住所	

④

送り元郵便局 送り先郵便局	発信者住所	本文
	受信者住所	

⑤

発信者住所	本文
受信者住所	

最初は、各会社、各国でネットワークアーキテクチャの体系化が進められたが、他社製品同士の接続は困難でハードウェアやソフトウェアを利用できないという問題が残った。すなわち、標準化の重要性が認識された。

I S O（国際標準化機構）やC C I T T（国際電信電話諮問委員会。現在 I T U-T）では、異機種接続のネットワークのプロトコル標準化と国際通信の相互接続が検討され、O S I 参照モデルが示された。

O S I 参照モデル

第7層	アプリケーション層
第6層	プレゼンテーション層
第5層	セッション層
第4層	トランスポート層
第3層	ネットワーク層
第2層	データリンク層
第1層	物理層

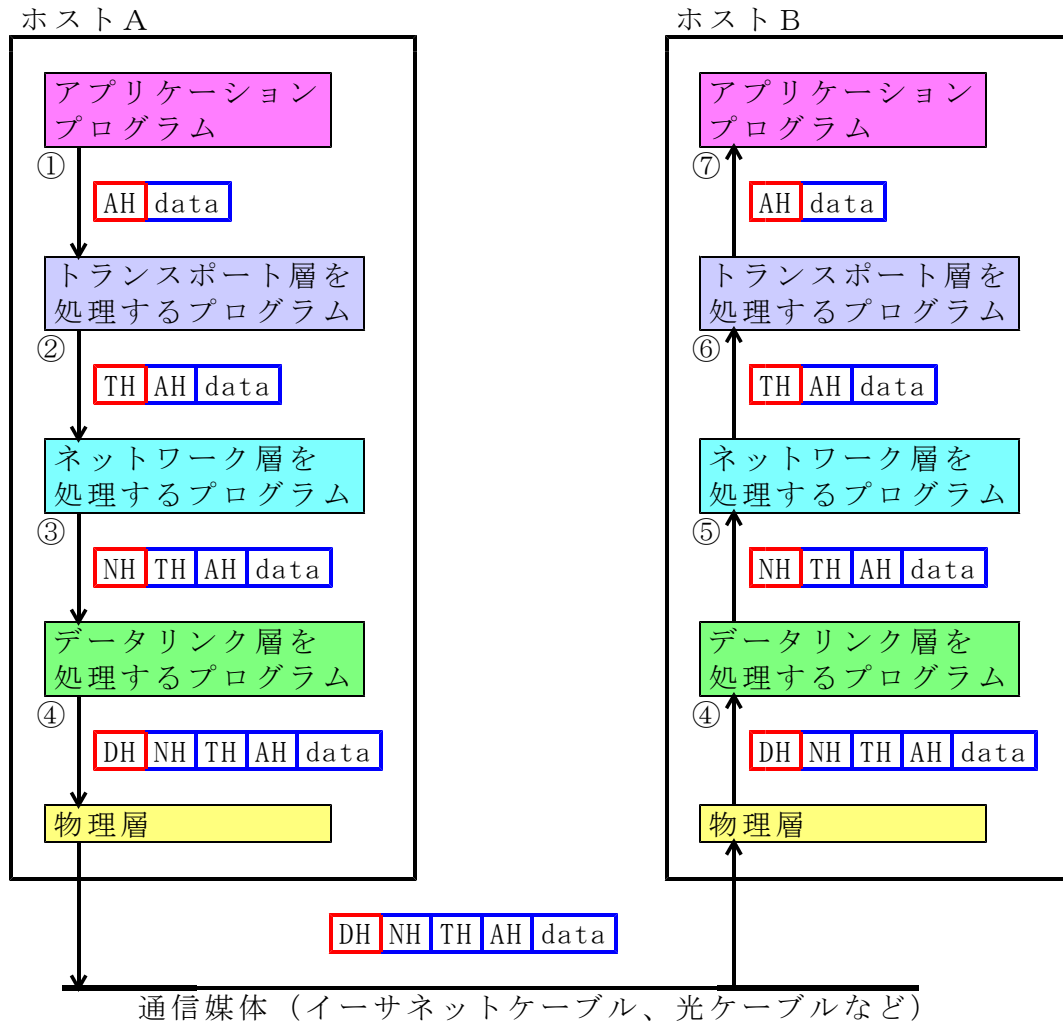
第1層から第4層までを下位層、第5層から第7層までを上位層という。上位層をまとめると考えやすい。応用層と呼ぶ。

層	機能
応用層	アプリケーションプログラム内でのプロトコルを規定する。
トランスポート層	異なる（同一でもよい）ネットワークの2台のコンピュータの プロセス間 で、データを正しく送受信するためのプロトコルを規定する。
ネットワーク層	異なる（同一でもよい）ネットワークの 2台のコンピュータ間 で、データを正しく送受信するためのプロトコルを規定する。
データリンク層	同一ネットワーク内の2台のコンピュータ間 で、データを正しく送受信するためのプロトコルを規定する。
物理層	伝送路（銅線、光ファイバー）の物理的な性質に関するプロトコルを規定する。

このなかでA R P A N E T用に開発されたT C P / I Pというプロトコル群が事実上の標準プロトコルとなった。

層	対応するプロトコル
応用層	F T P, T E L N E T, S M T P, P O P, H T T P
トランスポート層	T C P, U D P
ネットワーク層	I P, I C M P
データリンク層	イーサネット, F D D I, A T M
物理層	

コンピュータ内に各層のプロトコルを実現するプログラムがあり、各層間でデータがやりとりされる。やりとりされるデータは、プロトコルに関する部分（ヘッダという）と交換するデータの部分に分けられる。ホストAのアプリケーションからホストBのアプリケーションにデータが送られる様子は次のようになる。



- ① AHはアプリケーション層のヘッダ。
- ② THはトランスポート層のヘッダ。 **AH data** は全体でデータとみなされる。
- ③ NHはネットワーク層のヘッダ。 **TH AH data** は全体でデータとみなされる。
- ④ DHはデータリンク層のヘッダ。 **NH TH AH data** は全体でデータとみなされる。
- ⑤ NHはネットワーク層のヘッダ。 **TH AH data** は全体でデータとみなされる。
- ⑥ THはトランスポート層のヘッダ。 **AH data** は全体でデータとみなされる。
- ⑦ AHはアプリケーション層のヘッダ。

送信の場合、各層を処理するプログラムは、上位層を処理するプログラムからデータを受け取り、処理するプロトコルのヘッダを付けて下位層を処理するプログラムに渡す。受信の場合、下位層を処理するプログラムからデータを受け取り、処理するプロトコルのヘッダを取り除き、上位層を処理するプログラムに渡す。