TCP/IP 入門への第一歩

by sakae kumehara

2005/8/22

本書は、工学部 2 年生を対象とした共通講義として「インターネットの現状と将来」と題して行った講義のテキストのうち、「インターネットの現状」の部分に加筆修正を加えたものです。講義は、受講生がインターネットの基本技術に関して知識を持たないことを前提にして行いました。本書では、初心者ができるだけイメージを抱きやすいように、詳細部分の説明は省いて、基本的な部分だけを強調しています。なお、これからインターネットについて本格的に勉強しようとする人の手がかりになるように APPENDIX を追加しています。APPNEDIX では、若干技術的な説明をしています。

[　目次　]

１　インターネットはどのように発展してきたか

インターネットの起源はアメリカ国防総省の音頭取りで始まったARPANET(アーパネット、Advanced Research Projects Agency Network)にまで遡ることができます。この当時(1969年)は、アメリカとソビエトはいわゆる米ソ冷戦時代で、いつ第3次世界大戦が勃発してもおかしくない状態でした。このような時代にアメリカの国防総省はARPANETを提唱／構築しました。ARPANET開発の趣旨は、ソビエトから発射された原子爆弾がアメリカ合衆国のどこかに着弾してもダウンしないネットワークの構築です。電話のように送信者と受信者間に物理的な回線をつないでから通信を始める方式では、その要求を満たすことができません。敵からの爆撃にもっとも耐性の強いネットワークとして提唱されたのが、パケット転送という方式です。パケット転送では、データをパケット（Packet）と呼ばれる小さなブロックに分割し、それを別々に送信します。

ARPANETは、カリフォルニア大学ロサンゼルス校(UCLA)、スタンフォード研究所(SRI)、カリフォルニア大学サンタバーバラ校(UCSB)、ユタ大学(Utah)の4拠点のコンピュータを結んで始まりました。

ARPANETには当時のアメリカにも数台しかないような高性能のコンピュータが接続されました。研究者達は、これらの高性能コンピュータの計算能力やデータベースなどを、ネットワークを介して利用する方法を開発しようとしました。研究者は、ネットワークを介して結ばれていますので、共同研究に関する打ち合わせは電子メールを使って行っていました。そのうちに電子メールの面白さに気づいていったようです。ネットワークは次第に電子メールシステムを開発ターゲットとするシステムに変わっていきました。

ARPANETの構築が始まった1969年は、AT&Tのベル研究所(現在のルーセントテクノロジー)で、UNIX(代表的なネットワークOS)の開発がスタートした年です。UNIX はフリーの OS として大学や研究所の間に広まっていきます。そして、ARPANETではインターネットのバックボーン技術としてTCP/IPが開発され、それがUNIXシステムに同梱されて配布されことになります。当時ARPANETに参加できたのは国防総省と研究契約を締結することのできる裕福な大学や研究機関だけでしたが、ARPANETでの研究成果はTCP/IPの形で次々にUNIXシステムに導入されることになると、金持ちでない大学もコンピュータネットワークを構築することができるようになりました。やがて、アメリカにはいくつものコンピュータネットワークが構築されました。そのようなネットワークを相互に接続したのがCSNET(Computer Science Network)です。CSNETはアメリカのNFS(National Science Foundation、科学研究の資金援助を行う米国の政府機関なのでその意味では日本の文科省に当たる)のスポンサーシップによって構

築されました。
　ARPANET がインターネットの基本技術である TCP/IP を開発し、その TCP/IP で構築されたネットワークを相互に接続し、今日のインターネットの原型となるオープンなネットワークを作り上げたのが CSNET だといっていいでしょう。

２　インターネットを支える基本技術　IP

インターネットはTCP/IPという技術で作られています。TCP/IPとはTransmission Control Protocol / Internet Protocolという意味です。Protocol(プロトコル)とは、政治の世界で使われる場合は、外交儀礼とか議定書などという意味です。つまり、外交上の約束事です。言葉や政治制度の違う国同士がうまく付き合っていくためには何らかの約束事が必要です。それがプロトコルです。このプロトコルという言葉が通信の世界でも使われます。通信の世界でも約束事という意味です。通信をする同士も何らかの約束事に基づかないと意思の疎通ができません。IP(インターネットプロトコル、Internet Protocol)とは、インターネットを介して2つのコンピュータ間でデータを交換するときの約束事です。インターネットは数百にも及ぶプロトコルの塊ですが、その代表がTCPとIPです。なかでも、インターネットのもっとも基本的なプロトコルといえるのが、IPです。

IPの技術は郵便の制度によく似ています。手紙を送信する場合、封筒の表には相手方の住所を書き、裏面には自分の住所を書きます。そして、封筒の中には相手に送信すべきデータが入れられます。インターネットを介してデータを相手側に送信する場合は、適当な大きさにデータを切り分けます。データが大きすぎる場合は、ネットワークを利用する他の人に迷惑をかけます。小さすぎる場合は技術的に不都合なことがあります。従って、データをネットワーク上に送信する場合は、データをある適当な大きさの範囲の中に納める必要があります。この小さなデータのかたまりはパケットと呼ばれます。このパケットの先頭部分(ヘッダと呼ばれます)には、そのパケットの宛先はどこか、送信者はだれかなどが書かれます。宛先はどこかというのは、コンピュータネットワークの世界では、宛先に当たるコンピュータはどんなコンピュータで、そのコンピュータはどのネットワークに属しているかということです。コンピュータネットワークでは、コンピュータのことは通常ホストといいます。ホストというと昔のことを知っている人は、大型コンピュータをイメージするかも知れませんが、ここでホストというときは大型コンピュータの意味ではありませんので注意してください。宛先に当たるホストはどれで、そのホストはどのネットワークに属しているかは、IP アドレスという識別子で示します。IP アドレスは、32ビットの2進数で表しますが、通常は人間に分かりやすいように、8ビットずつを10進数表示して、それをドット”.”で結んだ形式で表します。この方式をドット区切り10進表示といいます。たとえば、「20．2．2．200」というような感じになります。この中にどのネットワーク、そのネットワーク内のどのホストという情報が含まれます。このままでは、少し柔軟性に欠けますので、現在の形式は「20．2．2．200/24」などのような形で使われます。このIPアドレスの意味は、20．2．2．200の先頭24ビットがネットワークの識別子で、25ビット目から最後の32ビット目までがそのネットワークに属するホストの識別子ということです。

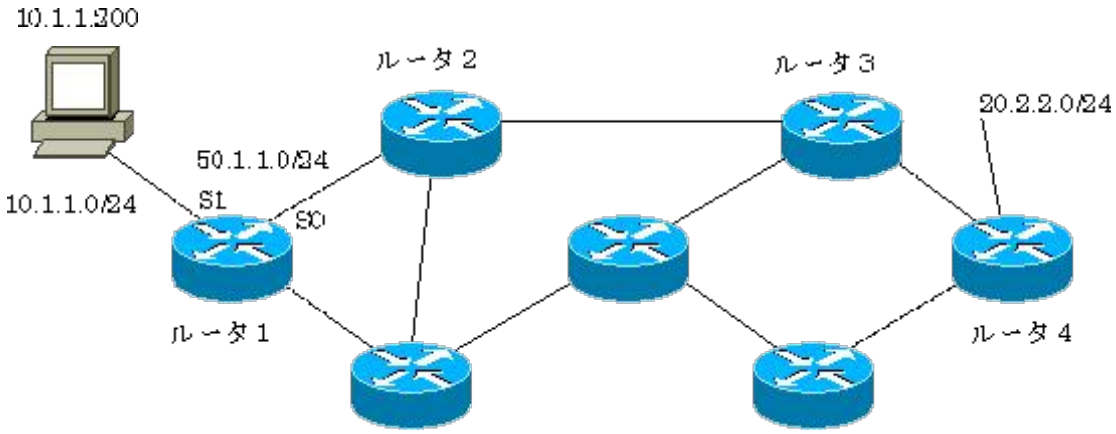
(先頭)

送信元アドレス ＝30.3.3.30	宛先アドレス ＝20.2.2.200	データ

上の図はパケットを簡略化して示したものです。送信者がどんなネットワークに属するどんなアドレスのホストであるのか、受信者がどんなネットワークに属するどんなアドレスのホストなのかがパケットのヘッダに記述されます。このほかにもいろいろな情報がヘッダに記述されます(ここから IP ヘッダの説明へリンク) が、ここでは省略します。

上の図は、送信者側(の IP)が作ったパケットです。このパケットには IP アドレスのどこまでがネットワークで、どこからがホストを表すかの目印(先ほどの例でいうとスラッシュの後の数字)は何も記述されていないことに注意してください。

ここまでの説明は、ネットワークを介してやりとりされるデータの形式についてです。しかし、やりとりされるデータの形式を決めても、そのパケットを運んでくれる装置がないとうまくいきません。郵便制度でいうと郵便局にあたります。インターネットでは、郵便局に当たるのがルータというネットワーク機器です。



ルータ同士のネットワーク

インターネットはルータのネットワークと言っていいでしょう。郵便制度の基礎は郵便番号でそれは郵便番号簿にまとめられていますが、ルータも郵便番号簿にあたるデータベースを持っています。ルータが持っているデータベースをルーティングテーブルといいます。ルーティングテーブルは、どの宛先ネットワークにパケットを送信するには、そのパケットを誰に渡したらいいか、つ

まり、どの宛先ネットワークにパケットを送信する場合には、どのルータにその転送を依頼したらいいかという情報が書かれたデータベースです。

次に示すのはルータ１のルーティングテーブルを簡略化したものです。

宛先アドレス/マスク	メトリック	ネクストホップ	インターフェース
20.2.2.0/24	3	ルータ２	S0
10.1.1.0/24	0	直接接続	S1
50.1.1.0/24	0	直接接続	S0
・・・	・・・	・・・	・・・

ルータが知っているのはネットワークです。ルータの知識は個々のあて先アドレスにパケットを届ける場合にどうするかというレベルのものに過ぎません。あて先アドレスが自分と同じネットワークの場合は自分で直接届けます。しかし、直接接続していないあて先の場合は、自分で届けることはできません。その場合には、他のルータに転送を依頼します。このルータをネクストホップ（NextHop）といいます。つまり、あて先ネットワーク毎のネクストホップを知っているネットワーク機器がルータということになります。ルータはパケットをネクストホップルータへ転送します。そのパケットを受信したルータも、受信パケットの宛先にマッチしたルートのネクストホップへと転送します。パケットがあて先ネットワークに到達するまでこれを繰り返します。あて先ネットワークに到達すると、そのパケットを受信したルータはあて先が自分と直接接続するネットワークであることが分かりますので、自分で配送します。

上の図でいうと、宛先が 20.2.2.0/24 というネットワークの場合は、ネクストホップはルータ２、つまり 20.2.2.0/24 宛にパケットを送信する場合は、ルータ 2 に転送するということになります。後は、ルータ 2 に任せます。ルータ２も、自分のルーティングテーブルを参照して、それをネクストホップルータに渡すだけです。これを続けていくと、あて先ネットワークまで到達できます。

上のルーティングテーブルで、宛先が 20.2.2.0/24 のエントリのインターフェースが S0 になっています。これは、ルータ１が宛先アドレス 20.2.2.0 のパケットを受信した場合、ルーティングテーブルに照らしてルータ 2 にパケットを転送しますが、そのとき、そのパケットを送り出す口がシリアル 0(Serial0)という名前のインターフェースだということです。インターフェースについては後で説明します。

ルーティングテーブルの宛先フィールドの部分を見てください。この部分にはネットワークのアドレスが表示されています。ネットアークのアドレスとは何でしょうか。先ほど IP アドレスはネットワーク部とホスト部の組み合わせ

で出来ているといいました。ネットワークアドレスというのは、ホスト部が全部 0 で出来ているアドレスのことです。全部 0 で出来ているとは、2 進数表示で全部 0 ということです。

先ほど IP が作るパケットのアドレス部にはどこまでがネットワーク部であるかを示す指標がどこにも付いていないといいました。どこまでがネットワーク部かはルータがルーティングテーブルに照らして判断します。では、ルーティングテーブルはどうやって作るのでしょうか。これは、ネットワーク技術者が手作業で作る場合もありますが、大抵の場合はルーティングプロトコルといわれるプログラムがルータ上で動いていて、そのプログラムが、他のルータ上で動いているルーティングプロトコルと会話をすることで情報を収集し、作ります。IP アドレスのうち、どこまで(つまり上位から何ビット目まで)がネットワークアドレス部であるかは、ルーティングプロトコルの種類によっては自動的に判断できる場合と、できない場合があります。古いルーティングプロトコル(通常レガシープロトコルなどと呼ばれます)は、アドレスの先頭部分のビットの並びから自動的に判断しますが、この方法は不都合が多いことが分かっていますので、最近のプロトコルはどこまでがネットワークアドレスとなるかは、ネットワーク技術者のルータに対する設定によって決めています。つまり、ネットワーク技術者が、どこまでをネットワークアドレスとするのが最適であるかネットワーク環境全体を考慮に入れてネットワーク毎に判断し、それをルータに教え込むということになります。

20.2.2.200/24 の/24 の部分をマスクとか、ネットワークマスク(あるいはサブネットワークマスク)といいます。/24 とは、1 が 24 個続いた数字を意味します。これは 2 進数で 1 が 24 個続いた数字という意味です。これを IP アドレスにマスクをするようにかぶせます。数学的にいいますと AND 計算します。IP アドレスは 32 ビットの 2 進数で表現できますので、マスクも 32 ビット必要です。従って、24 ビットマスクとは 11111111111111111111111100000000 という 2 進数になります。これと 20.2.2.200 を 2 進数表示したものの各桁同士の AND 計算(桁上がりはしません)をします

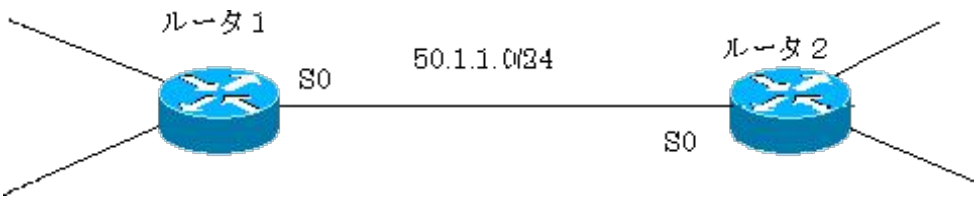
```
      00010100 00000010 00000010 11001000  (=  20.  2.  2.200)
AND)  11111111 11111111 11111111 00000000  (= 255.255.255.  0)
      --------------------------------------------------------
      00010100 00000010 00000010 00000000  (=  20.  2.  2.  0)
```

AND 計算の結果は、20.2.2.0 です。つまり、ルータは、受信したパケットの宛先アドレス 20.2.2.200 とルーティングテーブルの 20.2.2.0 のエントリを比較する場合は、20. 2. 2. 0 のエントリからマスク情報(/24)を取り出して、受信パケットの宛先アドレスと AND 計算をして、算出した結果と、ルーティングテーブルのエントリ(の宛先アドレス、つまり 20.2.2.0)を比較します。その結果

が一致すれば(正確にいうと一致するビットの長さが一番長いエントリがマッチエントリとして選択されます)、そのエントリが示すルートを最適なルート(経路)と考えます。

ただし、最適ルートといってもネクストホップがどれで、メトリックがどれだけかというだけです。ここで、メトリックとは距離のことです。メトリックは英語ではMetricと書きます。先ほどルーティングプロトコルということをちらっと話しましたが、このメトリックとは抽象的な距離で各ルーティングプロトコルによって具体的な意味は違います。一番レガシーなRIPというプロトコルでは、メトリックはルータを何台越えるかという意味です。ルータを越えることをホップといいますが、ルータ１から見ると、20.2.2.0にたどり着くまでにはルータをいくつ越えるかが20.2.2.0へのルートのメトリックということになります。このメトリック値が最小のルートが最適ルートと判断されます。先ほど示した図の例では、メトリック３、４、５などのいくつかのルートがありますが、メトリックが３のルートが最適なルートです。実は、インターネットでは、宛先にたどり着くのに複数のルートがあることが通常です。インターネットは、ソ連の核爆弾にもっとも破壊されにくいネットワークといいましがが、それはこういう意味です。そして、実際にいくつかあるルートの中の最適ルートがルーティングテーブルに格納されます。

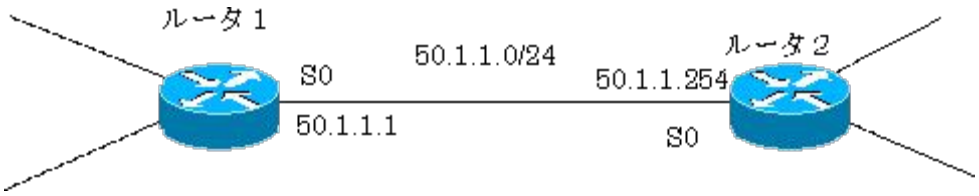
次にネクストホップルータとインターフェースについて説明します。ネクストホップルータについては、最適ルートに当たるルータのパスを想定した場合に、自分から見て、次のホップのことだということはすでにお話しました。この「次」とはどういう意味なのでしょうか。この「次」という意味は、隣接ルータのうち、あて先ネットワークへのパス上に存在しているルータという意味です。では、隣接とはどういう意味でしょうか。はじめにインターフェースについて説明します。インターフェースとは、ルータがネットワークと接続している部分です



上の図では、ルータ１のインターフェースS0とルータ２のインターフェースS0は共に同じネットワーク50.1.1.0/24に属しています。このように２つのルータが同じネットワークに接続している状態を隣接といいます。ルータ１とルータ２は隣接ルータということになります。つまり、ネクストホップルータは隣接ルータですから、同じネットワークを共有しているルータ同士ということになります。実は、パケットは、ネットワークを共有するコンピュータ(ルータもコンピュータです)同士の間でしか、受け渡しをすることができません。パケットは、必ずあるルータのインターフェースから、同じネットワークに所

属する隣接ルータのインターフェースへと転送されます。上の図でいえば、ルータ１の S0 から、ルータ２の S0 へという形で転送されていきます。同じネットワークに所属しないインターフェース同士で直接パケットの送受信が行われることはありません。

　同じネットワークに所属しているインターフェースのアドレスのネットワーク部は必ず同じになっています。



上のネットワークでは、マスクが 24 ビット長ですので、ネットワークのアドレスは 50.1.1.0 です。このような表現をプリフィックスという言い方をすることがあります。ルータ１の S0 とルータ２の S0 のプリフィックスは共に 50.1.1.0 となっています。

　では、ここまでの話を復習しましょう。同じネットワークに所属するインターフェースを持っているルータ同士が隣接ルータです。同じネットワークにインターフェースを持つルータ同士は、直接パケットの交換を行うことができます。つまり、直接パケットのやりとりが出来るルータ同士が隣接ルータということになります。ルータは複数のネットワークに所属しています。そのため、ルータ網は、同じネットワークに属するインターフェース間のパケット転送を繰り返すことで、遠くのネットワークにパケットを送り届けることができます。遠くのネットワークにパケットを転送する時に、ルータが参照するデータベースがルーティングテーブルです。

※）　通常ルータ同士はポイントツーポイントという形で接続することが多いのですが、それは上の図のルータ１とルータ２のように１対１で接続するタイプのネットワークです。このような形のネットワークではサブネットワークマスクを/30 とするのが普通ですが、ここでは初心者に分かりやすいように/24 で説明します。

３　LAN と TCP/IP の仲介役　ARP

先ほどは IP 同士が通信をすればそれで OK というような話し方をしましたが、実はそんな簡単にはいきません。なぜなら IP 同士はつながっていないからです。実際はデータリンク層の世話になります。データリンク層は正確には TCP/IP ではありません。これは LAN プロトコルといわれる範疇の事柄で、皆さんが大学内で日常的に使っているのはイーサネットといわれるものです。このイーサネットを使って通信をする場合について説明します。

イーサネットで通信をする場合のアドレスを MAC アドレスといいます。先ほど IP アドレスのネットワークアドレス部が同じもの同士で直接データのやりとりをすると説明しました。IP アドレスのネットワークアドレス部が同じ、つまりネットワークアドレスが同じもの同士だけが直接通信をすることができ、そのときに使うのが MAC アドレスです。MAC アドレスは２進４８ビットのアドレスです。

（MAC アドレスの例　16 進表示）　00-e0-63-9a-ea-fd

４８ビットの２進数では分かりにくいので 16 進表示を使うのが通常です。8 ビットずつ 16 進表示をします。先頭の 24 ビットはインターフェースのベンダー(製品の販売をする会社。製品のメーカや販売代理店のこと)を一意に識別する番号です。後半の 24 ビットはそのベンダー内で一意の識別番号です。

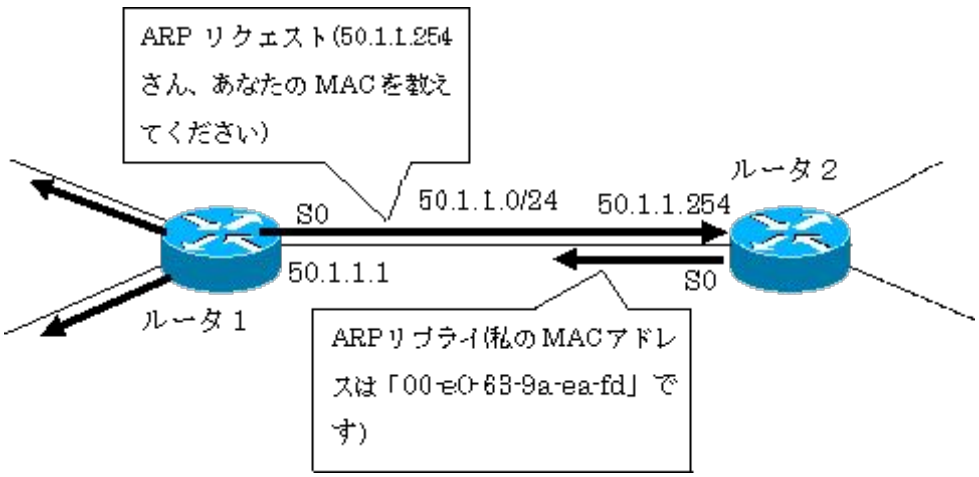
通信プログラムを使うユーザは相手のコンピュータ（通信したいプログラムが動作しているコンピュータ）の MAC アドレスをたぶん知りません。たぶん、知っているのは IP アドレスです（本当は、これも知らないでしょう。知っているのはドメイン名で、これを DNS の助けを借りて、IP アドレスに変換しますが、この点はここでは考えないことにします）。従って、与えられた IP アドレスから MAC アドレスを取り出すための仕組みが必要になります。これを ARP といいます。ARP とは、Address Resolution Protocol という意味です。日本語で言えばアドレス解決プロトコルということになります。

データリンク層では、IP から受け取った(送信依頼された)パケットを隣接のルータあるいはホスト上のデータリンク層に渡します。データリンク層で通信を制御しているのはハードウェア的には NIC(Network Interface Card)といわれるボードです。ソフトウエア的には、NIC 上のチップにインストールされているプログラムが機能します。データリンク層のプログラムが認識できるデータの固まりはフレームというものです。したがって、IP から送信依頼の形で受け取ったパケットをフレームの形にする必要があります。つまり、パケットをフレームで包みます。具体的にはパケット先頭にフレームヘッダを付加し、最後にトレーラというものを追加して、パケット全体をフレームで包み込むこと

になります(パッケージングあるいはカプセリング)。

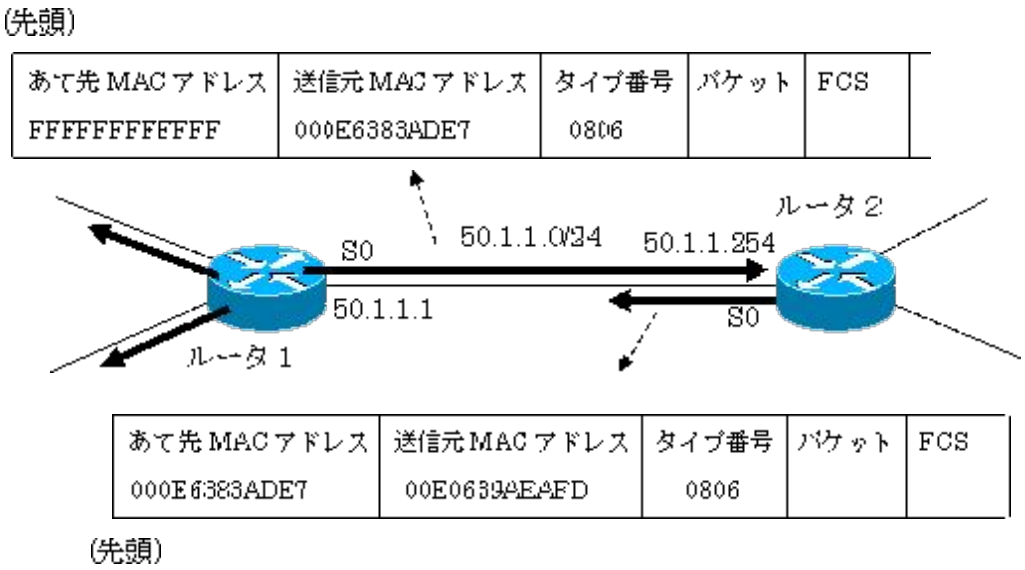
ARP では、ARP リクエストと ARP リプライを使います。ARP リクエストは、特定の IP アドレスを持ったルータあるいはホストの（インターフェースの）MAC アドレスを聞き出すことです。あて先は、どこだか分かりません。相手の IP アドレスは知っていますが、その相手が、どのインターフェースと直接接続しているか分からないからです。もちろん相手が直接接続していることが前提です。直接接続していないなら、もともとフレームの交換ができないのですから、MAC アドレスを聞き出しても無意味です。自分のいずれかのインターフェースとネットワークを共有しているはずですので、手当たり次第に MAC アドレスを聞き出します。この方法をブロードキャストといいます。

次の図で説明することにします。ルータ 1 上の IP が 50.1.1.254 の IP アドレス上の IP にパケットを送信しようとしていると仮定しましょう。ルータ 1 上の ARP は 3 つのインターフェースから、「50.1.1.254 の IP アドレスをお持ちの方、MAC アドレスを教えてください」という ARP リクエストをブロードキャストします。



この場合、ルータ 1 が自分のすべてのインターフェースから送り出す ARP リクエストはフレームでカプセル化されていますが、あて先 MAC アドレスは、すべてのあて先を示す「FFFFFFFFFFFF」となっています。もし、50.1.1.254 のホスト上の ARP が生きていれば、ARP リプライを使って MAC アドレスを知らせてくるはずです。ARP リプライは、ARP リクエストを送信した特定の相手の MAC アドレスに対して送信します。このように特定の相手に送信する方法をユニキャストといいます。

以上の説明を具体的にフレームで図解すると次のようになります。



苦労して入手した MAC アドレスは、しばらくの間キャッシュに保存します。Dos 窓から arp ?a コマンドを実行すると、その時点で記憶している MAC 情報が表示されますので確認してみてください。

MAC アドレスの次にタイプ番号フィールドがありますが、これは何を意味するのでしょうか。このタイプ番号は、IP 層のプロトコルの識別子です。ARP パケットを作ったのはネットワーク層の ARP で、それを受信するのも ARP です。ARP の識別子(タイプ番号)は 0x0806 です。したがって、タイプ番号 0x0806 のフレームを受信した NIC は、フレームヘッダ（と FCS）を取り除いて、上の層の ARP にパケットを渡さなくてはなりません。0x は後に 16 進数が続くことを示しています。フレームの最後に FCS というフィールドがありますが、これはフレームをチェックするためのフィールドです。

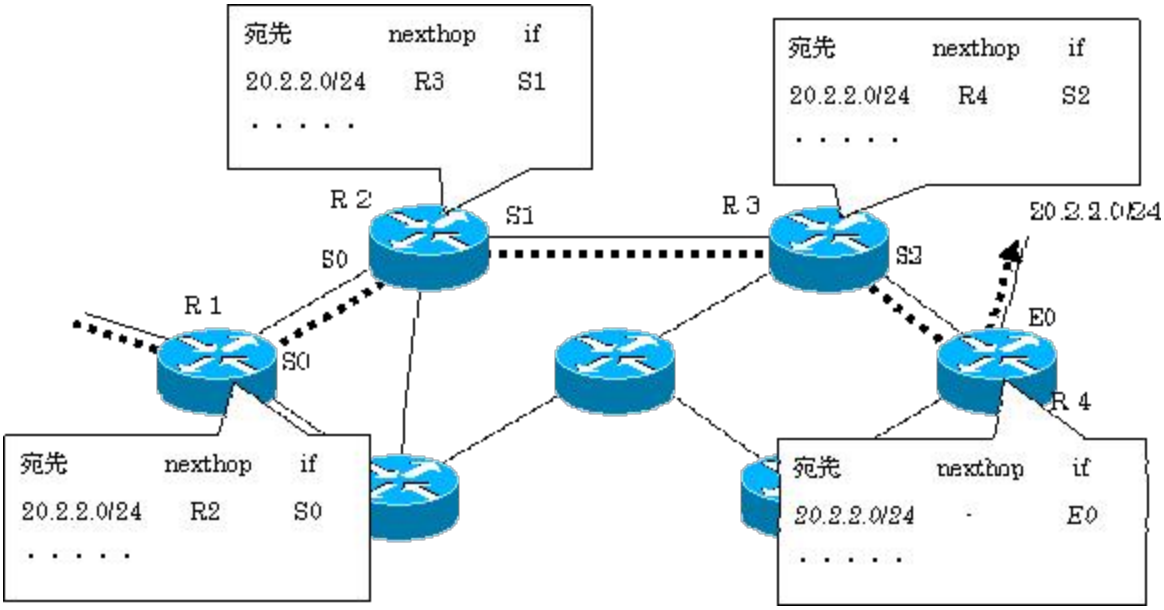
IP アドレスがあるのにどうして MAC アドレスで通信をするのでしょうか。受信パケットの IP アドレスと自分の IP アドレスのマッチングを行うのは、ネットワーク層の IP です。ネットワーク層のプロトコルは OS のカーネルに属しています。つまり、受信パケットの IP アドレスと自分の IP アドレスのマッチングを行うのは、カーネルの仕事ということになります。カーネルは様々なタスクをスケジューリングします。カーネルは大変忙しいのです。もし仮に、LAN 内でも IP アドレスで相手を識別するということになると、OS の仕事がさらに増えることになります。MAC アドレスを使えば、OS は、NIC が自分宛と判断したものだけを処理すれば済みます。

そんなに便利なMACアドレスなら、IPアドレスなど使わないで、MACアドレスだけで通信すればいいのにと思いませんか。しかし、事はそんなに簡単ではありません。MACアドレスにはネットワークアドレスの概念がないのです。MACアドレスだけ書いて、データのかたまりを送っても、そのデータのかたまりをどこに送ったらいいのか。そのデータのかたまりの宛先として指定された（MAC アドレスを使って指定された）ホストがどこにあるのか誰も知りません。つまり、インターネットのどこでどんなMACアドレスのホストが動いているのか管理する仕組みがないのです。このことをMACアドレスは「位置情報を持たない」という言い方をすることもあります。IPアドレスは位置情報を持っています。マスクを使うとIPアドレスからネットワークアドレスを取り出すことが出来るということはこういう意味なのです。

４　IP パケットを運ぶための仕掛け　ルーティング

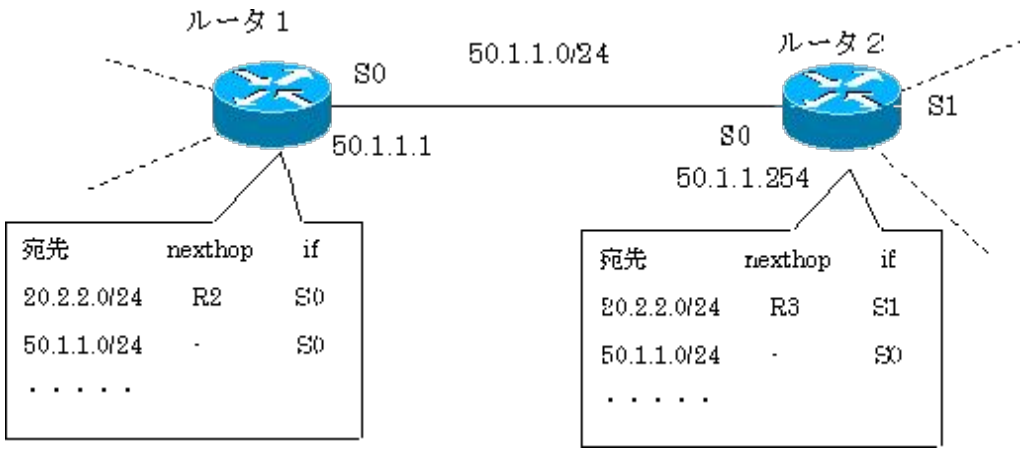
ルータは遠くのネットワークにパケットを届ける手段を持っていますので、フレームを使って LAN 上のルータにパケットを送れば、後はルータ同士がルータ網を使って運んでくれそうですね。その想像が正しいかどうか、少しルータの仕組みを調べてみましょう。

次の図では、各ルータが持っているルーティングテーブルを簡略化したものを示しています。



※)図中のルーティングテーブルの if はインターフェース(interface)の意味です。

IP アドレスは、ホストに対して割り当てられているのではなく、インターフェースに対して割り当てられています。ルータはこのインターフェースを複数もち、それぞれが異なるネットワークに属しています。次の図は上のネットワークから R1 と R2 の部分を取り出したものです。R1 と R2 は 50.1.1.0/24 というネットワークを共有していますので、ルーティングテーブルには、あて先として 50.1.1.0/24 が載っています。このネットワークへは直接接続していますので、ネクストホップ（NextHop）は空欄になっています。



今、ルータ１があて先アドレス 20.2.2.200 のパケットを受信したとします。このパケットを転送する場合について考えてみましょう。ルータ 1 の IP はあて先エントリとあて先アドレス 20.2.2.200 のマッチングを行い、(ルーティングテーブル上の)あて先ネットワークアドレス 20.2.2.0 のエントリが最適ルートであると判断します。パケットをネットワーク 20.2.2.0 に送信するためには、インターフェース S0 経由で NextHop のルータ 2 に転送する必要があります。

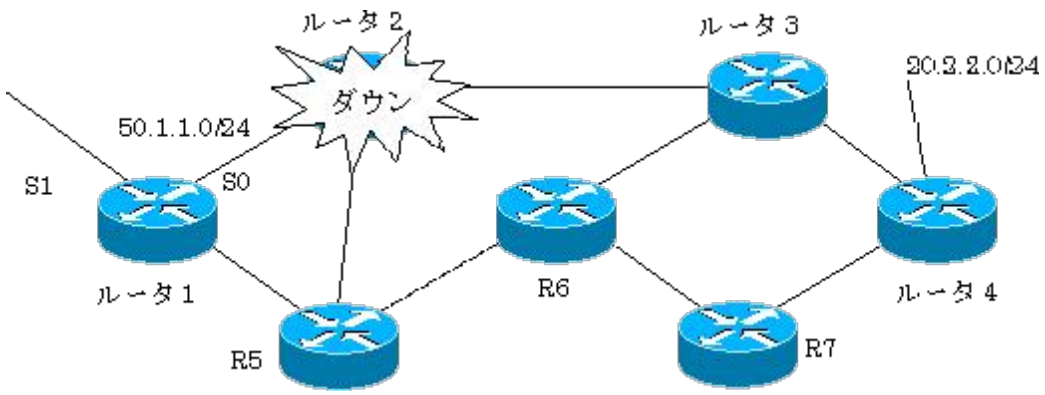
ルータ１のルーティングテーブルでは、簡単のために R2 と書きましたが、実際のルーティングテーブルでは、ルータ 2 の IP アドレスが書かれています。この IP アドレスは、LAN で直接送信できるアドレスでなくてはなりませんので、ネットワークアドレス部がルータ 1 と共通です。パケットを送り出すインターフェース S0 は 50.1.1.1 で、パケットを受け取るルータ 2 のインターフェース S0 のアドレスは 50.1.1.254 です。ルータ 1 の S0 とルータ 2 の S0 は同じネットワークを共有していることになりますので、LAN でパケットの受け渡しができるということになります。

LAN での送信は、前に説明したように MAC アドレスを使います。しかし、ルータ 1 はルータ 2 の S0 の MAC アドレスを知りませんので、ARP を使って聞き出します。そして、ARP で聞き出した MAC アドレスをつけたフレームで、パケットを包んで(カプセル化)、ルータ 2 に送り出します。

ルータ２はそのフレームを受け取ります。フレームを受け取るのはルータ 2 の NIC です。NIC はあて先 MAC アドレスが自分宛であることを確認し、それをネットワーク層に渡します。タイプ番号は渡す相手がネットワーク層の誰であるかを指示しています。タイプ番号が 0x0800 ならば、フレームヘッダを外して(フレームの場合は、最後部に FCS というフィールドがついていますので、細かいことを言えば FCS も外して)、パケットを IP に渡します。IP はパケットのあて先 IP アドレスを確認します。自分宛ではないようです。また、自分の

属しているネットワーク宛でもなさそうです。そこで、ルーティングテーブルを確認して、ネクストホップを探し、そのネクストホップに転送を依頼します。ネクストホップに渡すときはARPを使って、MACアドレスを聞き出し、フレームで包んでから渡します。これをあて先ネットワークまで繰り返します。あて先ネットワークに直接接続しているルータまで辿り着くと、ルータは自分で直接パケットを送り届けます。もちろんこの場合も、ARPを使ってMACアドレスを聞き出し、フレームでカプセル化する必要があることは今までの説明の通りです。

ルータはルーティングテーブルの内容に従って、パケットの転送をするわけですが、それではそのルーティング情報はどのようにして手に入れるのでしょうか。これにはスタティックな方法とダイナミックな方法があります。スタティックな方法はネットワーク管理者が手動でルータに教えます(つまり、ネットワーク管理者が設定します)。これに対して、ダイナミックな方法は、ルータ同士が会話をすることで、宛先ネットワークへのルートを学習し、最適ルートを選択して、それをルーティングテーブルに格納します。ルータ同士の会話の方法にはいくつかの方言があります。これをルーティングプロトコルといいます。一番古い方言はRIP（Routing Information Protocol）と言います。RIPは、いまでも小さなネットワークではよく使われています。そのほかRIPとよく似たプロトコルにIGRPがあります。これはルータベンダとして有名なCisco Systemsのプロトコルです。その他に大きなネットワークでの利用に適したOSPFやEIGRP（EIGRPもCisco Systemsのプロトコル）などというプロトコルがあります。以上のルーティングプロトコルは、主に同じ管理方針に従って管理運用されているネットワークドメインの間で(通常は同じISPに属しているネットワーク間と思ってください)ルーティングのための情報を交換する場合のプロトコルです。複数のISPに跨るようにして、ルーティング情報を交換するためには、BGPという別種のルーティングプロトコルが必要となります。ただし、BGPは他のプロトコルとはかなり性質の異なるルーティングプロトコルですので、今回はBGPを無視して説明しています。そのため、ここまでの説明の中にはBGPには当てはまらないところもありますので注意してください。



インターネットのそもそもの出発が障害に強いネットワークを作ることだったといいました。そのことについて少し話をしておきます。上のネットワークで今、ルータ R2 がダウンしたとします。このときその他のルータは情報を交換して、個々の宛先ごとにルートを計算します。ルーティングプロトコルに RIP を使用している場合、宛先 20.2.2.0 へは、ルータ R1、R5、R6、R3、R4 と辿るルートと、R1、R5、R6、R7、R4 と辿るルートの 2 通りのルートが最適ルートということになります。ただし、ルーティングプロトコル毎にどのルートを最適とするかの判断基準は異なります。このように、たとえネットワークのどこかに障害が発生してルーティングテーブルと実際のネットワークとの間に矛盾が発生しても、ルーティングプロトコルが機能することで、すぐに実際にネットワークとルーティングテーブルの間の整合性は回復します。ネットワークに障害が発生し、ルーティングテーブルとの間に矛盾が発生し、やがてその矛盾が解消することをネットワークが収束するといいます。この収束時間はルーティングプロトコルの種類によって異なります。収束時間が短いほど、そのルーティングプロトコルは障害に強いということになります。このようにルーティングプロトコルがネットワーク障害の状態から自動的に抜け出す手段を持っていることが TCP/IP が傷害に強いといわれる理由です。

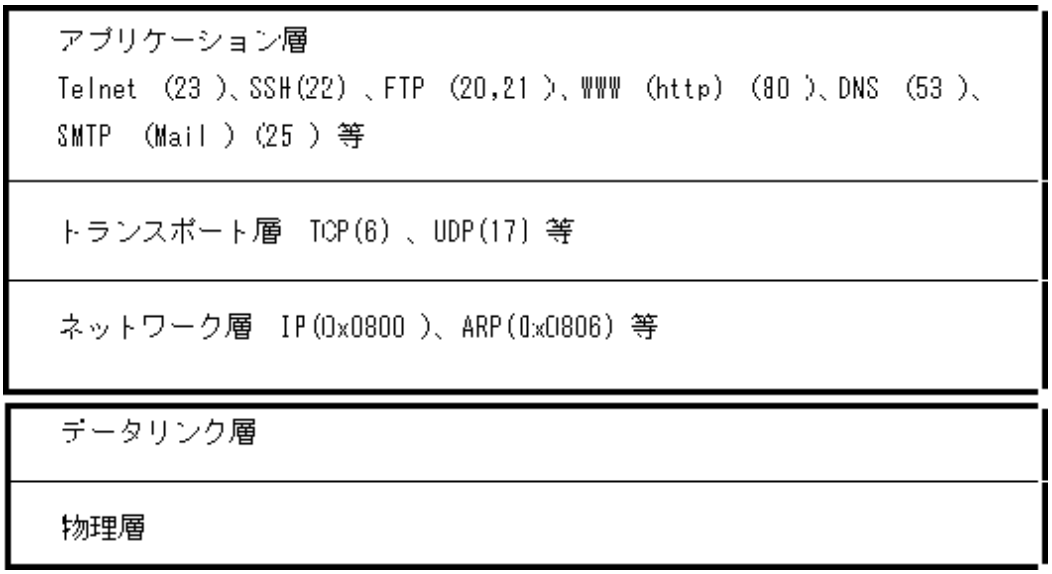
５　階層構造のプロトコル群

　インターネットのプロトコルは IP だけではありません。インターネットは非常の多くのプロトコルから成り立っています。そのプロトコルの集まりが TCP/IP と呼ばれるものです。TCP/IP はプロトコルがスタック構造に積み上げられたものです。スタック構造とは、干草などを積み重ねた構造のことです。下から順番に積んでいき取り出すときは、上から順に取り出します。上とか下という言葉にあまりこだわらないでください。要は、INPUT をした逆順に、新しいものから OUTPUT していくという構造になっているのがスタック構造です。TCP や IP、その他たくさんのプロトコルが階層構造で集まっているものを TCP/IP プロトコルスイートといいます。スイートとは、suite と書きます。つまり、一揃いということです。ホテルのスイートルーム(寝室、浴室、居間などが一式そろっています)とか、スーツ(上着とズボンで一揃い) などと同じですね。

スタック構造は 3 階層からできています。下から、ネットワーク層、トランスポート層、アプリケーション層と呼ばれます。各階層にはさまざまなプロトコルが存在しますが、とても全部を示すことはできませんので、代表的なものだけ示します。

アプリケーション層 Telnet （23 ）、SSH(22) 、FTP （20,21 ）、WWW （http) （80 ）、DNS (53)、SMTP （Mail)(25)等 ※)カッコ内はポート番号
トランスポート層 TCP(6) 、UDP(17) ※)カッコ内はプロトコル番号
ネットワーク層 IP(0x0800)、ARP[0x0806) 等 ※)カッコ内はタイプ番号

各層には多くのプロトコルがありますので、それを識別する何かが必要です。その識別子も図中に示しました。通常はこの下に実際にコンピュータ同士を結んでいる層がおかれます。物理的にコンピュータ同士をつないでいるのは、皆さんが実際に大学で使っているネットワークを例にとると、LAN(Local Area Network)と呼ばれるネットワークです。LAN は、物理層とデータリンク層から構成されます。



通常、ユーザに対してサービスを提供するのは一番上の層のアプリケーションプロトコルです。

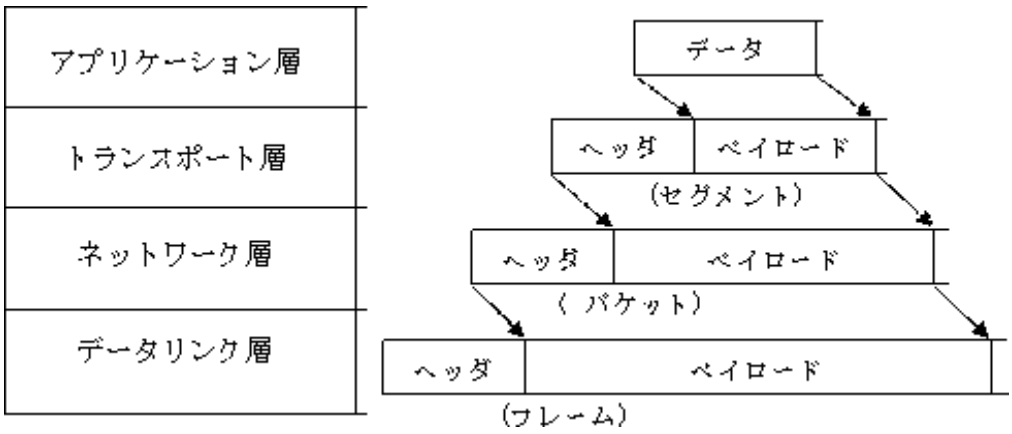
２台のコンピュータ上のプログラム同士が通信をする場合は、このアプリケーション層のプロトコルを利用します。アプリケーション同士が通信をするということは、データを交換するということです。送信すべきデータはいろいろの大きさであることが予想されますが、これを適当な大きさのデータに分割して送信します。

相手側のアプリケーションは、送信側のアプリケーションと通信できるタイプのアプリケーションでなくてはなりません。送信側が SMTP というプロトコルを利用したメールプログラムであるなら、受信側も SMTP というプロトコルを利用したメールプログラムです。送信側と受信側で送受信の手順について同意していなくては通信ができません。同意に基づく通信を行うためには、双方で同種のプロトコルを使います。通信をするということはただデータを相手に送信すればいいというわけではありません。通信を制御する必要があります。通信を制御するためにはいろいろの情報を通信相手との間で、共有する必要があります。この制御情報は、通常は、お互いに交換しあうデータの先頭部分に付加して送信します。

データを相手に送信すればいいのですが、アプリケーション層同士は直接物理的に接続していませんので、下の層(トランスポート層)に送信を依頼します。下の層の代表的なプロトコルは TCP と UDP です。例えば、TCP に送信を依頼するとしましょう。TCP でも、送信側と受信側で予めデータ送受信に関する同意が必要です。送信側が TCP ならば、受信側も TCP の手順に沿って受信しないとお互いに話が通じません。

ところでTCPの層も直接接続していないことは一目瞭然です。今度も下の層(ネットワーク層)に依頼します。ここでは、IPに頼むとしましょう。頼まれたIPは当然相手側のプロトコルスタックのIPに対して通信をしようとします。しかし、またしても、ここでも直接通信できませんのでまたまた下の層(データリンク層)に送信を頼むことになります。

このように一番上のアプリケーションプログラムから始まって次々に下の層のプロトコル(の約束事に従って作成されたプログラム)に依頼することになります。しかし、頼むにはそれなりの流儀があります。通常は、これこれのことをしたいのでデータの送信を頼むということになります。これは上の層のプログラムから下の層のプログラムにデータが渡される場合に、その先頭部分に追加します。これは上の層から下の層への申し送り状ということになります。この申し送り状をヘッダといいます。従って、上の層から下の層へ申し送りがなされるたびにヘッダが付加されていきますので、ヘッダがどんどんながくなります。これをカプセル化といいます。



アプリケーション層のデータは適当な大きさに分割されたデータです。TCPでデータを送信する場合は、ある一連のデータの中の一片ですので、データストリームといいます。これをトランスポートに渡します。トランスポート層をトラック(車)にたとえると、このデータは荷物ということになりますので、ペイロード(支払いの発生する荷物ということです)といわれます。トランスポート層で、ペイロードにヘッダをつけたものをセグメントといいます。このセグメントはネットワーク層に渡されます。これも、ネットワーク層にとってはペイロードです。このペイロードにヘッダがつけられます。これをパケット(データグラム)といいます。さらにこのパケットがペイロードとして、データリンク層に渡され、データリンク層では、フレームを作ります。

フレームはネットワーク経由で隣接デバイスまで届けられます。隣接デバイスのデータリンク層は、フレームからパケットを取り出します。そのあて先IPアドレスが自分宛ならば、さらに上の層に届けられます。あて先IPアドレスが自分宛でなければ、またパケットをフレームでカプセリングして、あて先へ

向けて転送します。

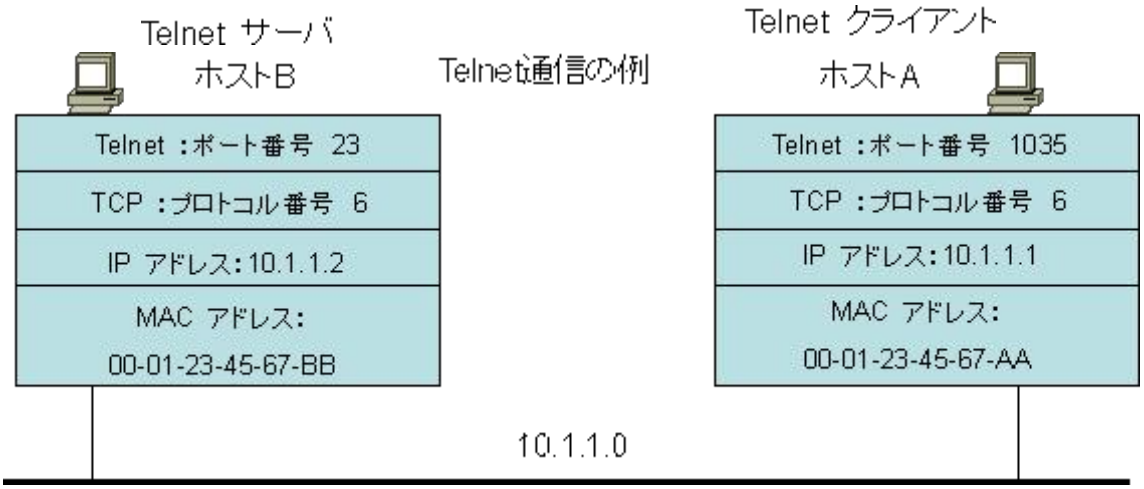
TCP/IP では各層でデータの塊(データパッケージと呼ばれます)に違う名前をつけていますので、ここでまとめておきます。TCP を利用する場合は、ネットワーク層はパケット(あるいはデータグラム)、トランスポート層はセグメント、アプリケーション層はストリームです。UDP を使う場合は、ネットワーク層はデータグラム（あるいはパケット）、トランスポート層はデータグラム(あるいはパケット）、アプリケーション層はメッセージです。また、パケットという用語には、データを分割して伝送する方式という意味もありますので、データパッケージとしての呼び名というよりももっと一般的な用語として使用されている場合もあります。

６　アプリケーションサービスを下支えする　TCP と UDP

トランスポート層のプロトコルとしては、TCP と UDP が代表的です。UDP は TCP の簡易版というべきプロトコルです。TCP のことが分かれば UDP もだいたいの想像がつきますので、はじめに TCP について説明します。

６．１　高信頼性トランスポートサービス　TCP

はじめに、トランスポート層で TCP を使って、Telnet の通信をする場合について具体的に説明してみましょう。

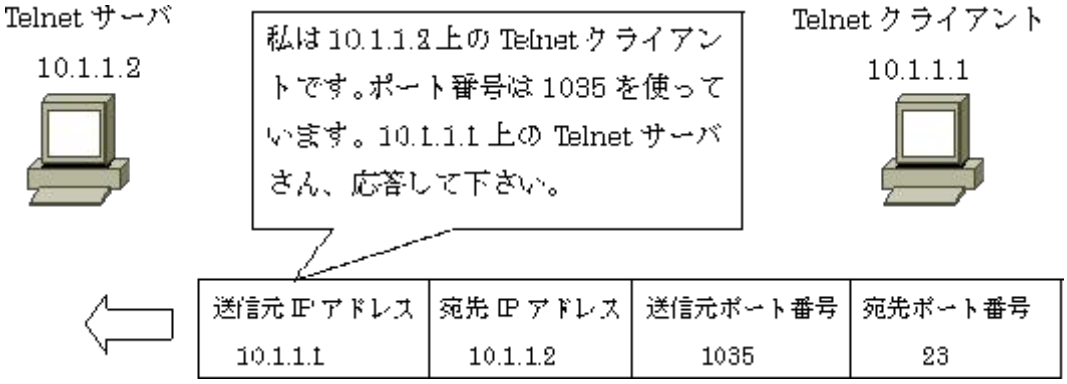


上の図では、ホスト A とホスト B が同じネットワーク上に設置されています。インターネットのプログラムは通常、クライアントサーバシステムという仕組みで出来ています。クライアントサーバシステムというという仕組みでは、通信はクライアントからアクティブに始められます。サーバはクライアントからのアクティブな要求をパッシブにひたすら待ち続け、クライアントからの要求があれば、それに実直に応答しようとします。

Telnet クライアントは 10.1.1.2 という IP アドレスを持ったコンピュータ上にある Telnet サーバに要求を行います。

ホスト A 上の Telnet クライアントはホスト B 上の Telnet サーバに話しかけます。相手は「IP アドレス 10.1.1.2 上の Telnet サーバさん」と指定します。では Telnet サーバさんとはどうやって指定すればいいでしょうか。アプリケーションプロトコルはポート番号という番号を使って識別します。アプリケーションプロトコルサーバは、固有のポート番号を持っています。なぜ固定の番

号でなければならないのでしょうか。インターネットの通常のプロトコルはクライアントからアクティブに、パッシブなサーバに対して話しかけるという手順(クライアントサーバシステム)を採用しています。クライアントからアクティブに話しかけますので、サーバのポート番号は予め決まっていなくてはなりません。しかも、クライアントがそれを予め知っている必要があります。したがって、サーバのポート番号は固有で、しかも公知の番号でなくてはなりません(Well-Known ポート番号と呼ばれます。UNIX システムでは/etc/services に記載されています)。
Telnet サーバのポート番号は、23 番ですから、ホスト A 上の Telnet クライアントは IP アドレス 10.1.1.2 のコンピュータ上のポート番号 23 のプロトコルさんという呼びかけをして相手を識別します。この時、ホスト A 上の Telnet クライアントは当然自分を名乗らなくてはなりません。もちろん、10.1.1.1 上のポート番号は何番という形の名乗り方になります。ただし、クライアント側のポート番号は固定的なものではありません。その通信に限って使い捨てする番号です。なぜでしょうか。クライアントはアクティブに話しかけますから、話しかける際に、自分は今回の接続では何番のポート番号を使用しますとその場で宣言すれば済みます。従って、クライアントのポート番号は予め決まった固定ポート番号である必要はありません。このように一時使用のポート番号は、エフェメラル(ephemeral、短命の)ポート番号と呼ばれます。この番号は、Telnet クライアントが、今回の接続に関してはこの番号のポートで返事を待ちます、という意味が含まれます。通常は 1024 番以上のポート番号が使われます。



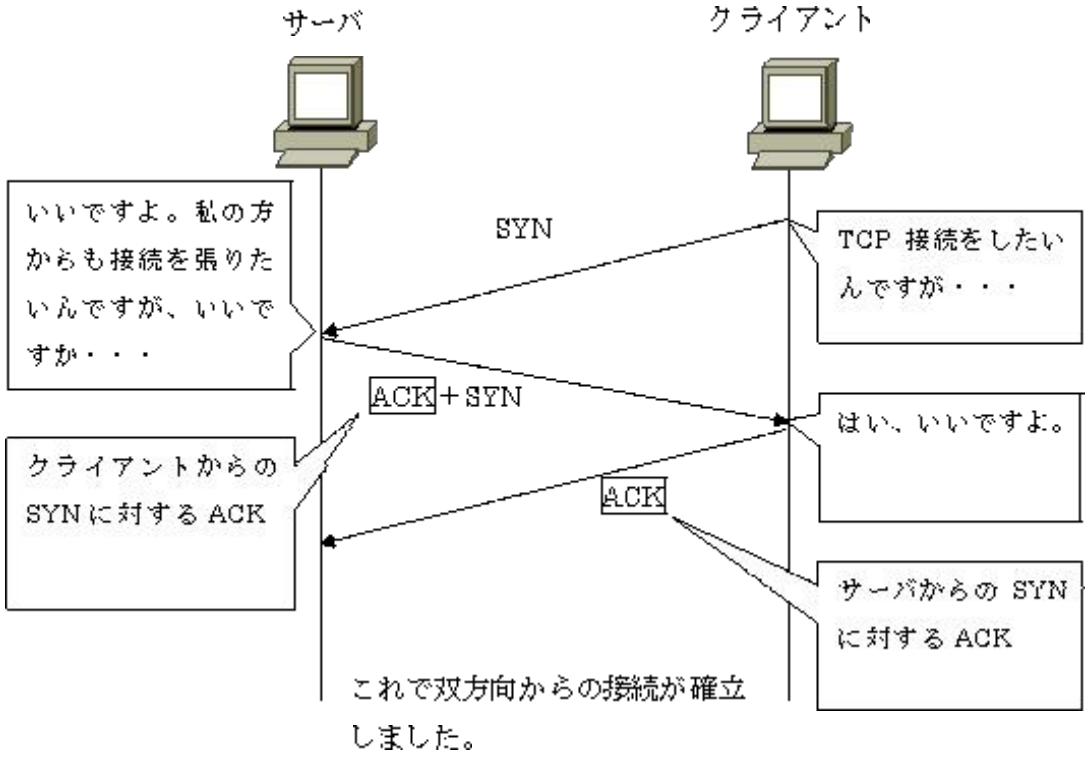
上の図では、ホスト A 上の Telnet クライアントは、「私は IP アドレス 10.1.1.1 のコンピュータ上のポート番号 1035 番のクライアントプログラムです。IP アドレス 10.1.1.2 のコンピュータ上のポート番号 23 番のサーバプログラムさん、私と通信をしてください。」という感じで話しかけています。

ここでは、この呼びかけに単体で反応するものがあるかのように書かれていますが、これもイメージが浮かびやすいように簡略化しています。実際は、

「10.1.1.2」さん、と呼ばれて自分への呼びかけであると認識するのは、10.1.1.2 のホスト上の IP です。そして、「ポート番号 23」さん、と呼ばれて、自分への呼びかけであると認識するのは、10.1.1.2 上の Telnet サーバです。

Telnet はトランスポート層のプロトコルとして TCP を使っていますが、TCP を使った接続の形をコネクション型の接続といいます。コネクション型の接続では、通信を行う前に接続を確立し、それを維持し、必要なデータの交換が済めば、接続を終了します。接続を確立し、維持し、終了するためには、接続する TCP 同士の間で制御情報を共有する必要があります。接続確立、接続維持、データ交換、接続終了のための制御情報はセグメントのヘッダ部分に記述します。

TCP では、接続の確立のために、3 ウエイハンドシェークという方式を使います。



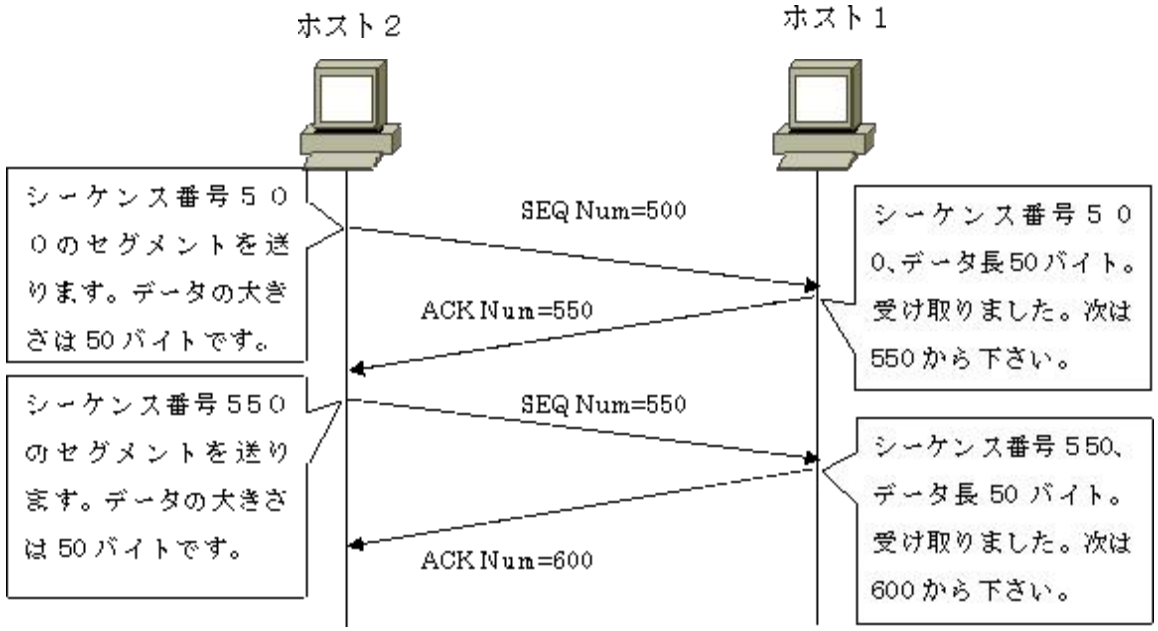
クライアント側が送信したパケットでは、セグメントヘッダの SYN というビットフィールドがセットされています。サーバはそれに対して、SYN と ACK というビットフィールドのセットされたパケットを送り返してきます。最後にクライアントが ACK ビットフィールドのセットされたパケットを送り返します。この 3 回の挨拶でコネクションが確立します。これを 3 ウエイハンドシェークといいます。これで接続が確立し、その接続上でパケットの交換が行われます。

SYN とは同期(SYNchronization)、ACK とは確認応答(ACKknowledgement)という意味です。SYN も ACK もセグメントヘッダの制御(Control)ビットフィールドの中のサブフィールドです。

Telnet クライアントから送信されたパケットが相手側のコンピュータまで到達した場合はどのように処理されるのでしょうか。Telnet クライアントが送信したパケットのヘッダには IP アドレスが書かれていますので、サーバ側のネットワーク層プロトコルである IP は自分宛のパケットであることを確認できます。このパケットは上の層に渡します。トランスポート層の誰に渡すかは、プロトコル番号というフィールドに記述されます。TCP の場合は、プロトコル番号が「６」になっています。UDP の場合は、プロトコル番号は「17」です。パケットのままでは、上の層の TCP は何が何だか分かりませんので、IP はパケットのヘッダを取り除いてセグメントを取り出し、相手に渡します。セグメントヘッダにはポート番号などの制御情報が記述されています。上のアプリケーション層に渡すときの指標はポート番号です。ポート番号が 23 になっていれば、Telnet サーバに渡すことになります。TCP はセグメントヘッダを切り落として、データのかたまりを Telnet サーバに渡します。これで、データが Telnet サーバから Telnet クライアントまで送り届けられました。

クライアントとサーバ間のデータ送信はたぶん 1 回で終わらないことが多いでしょう。しかも、困ったことにインターネットでは、後から送信したデータが先に届いてしまう可能性がないわけではありません。1 回ずつのデータが同じ経路を辿って宛先まで送られるかどうかは保証されていないからです。そこで、データの順番が狂ってもあとで順序よく並べ直すことが出来るようにヘッダにはシーケンス番号(Sequence Number)という目印が付けられます。データを順番通りに届けるのは TCP の役割ですので、シーケンス番号はセグメントヘッダに記述されます。

TCP では、シーケンス番号は送信されるストリームのオクテット数を表します。ここまでは、アプリケーションプロトコルを Telnet として説明してきましたが、Telnet はネットワーク仮想端末(NVT)プロトコルといわれるもので、ローカルホストからリモートのネットワークホストへ、ローカルホストと同じようにアクセスできるようにします。そのため、通常は１バイトずつデータをやり取りしますので、ここでの説明は Telnet の説明としてあまり、適切ではありません。単にシーケンス番号の説明と割り切ってください。説明のイメージとしては FTP です。シーケンス番号を使って、データを送信する様子を簡略化して図解すると次のようになります。

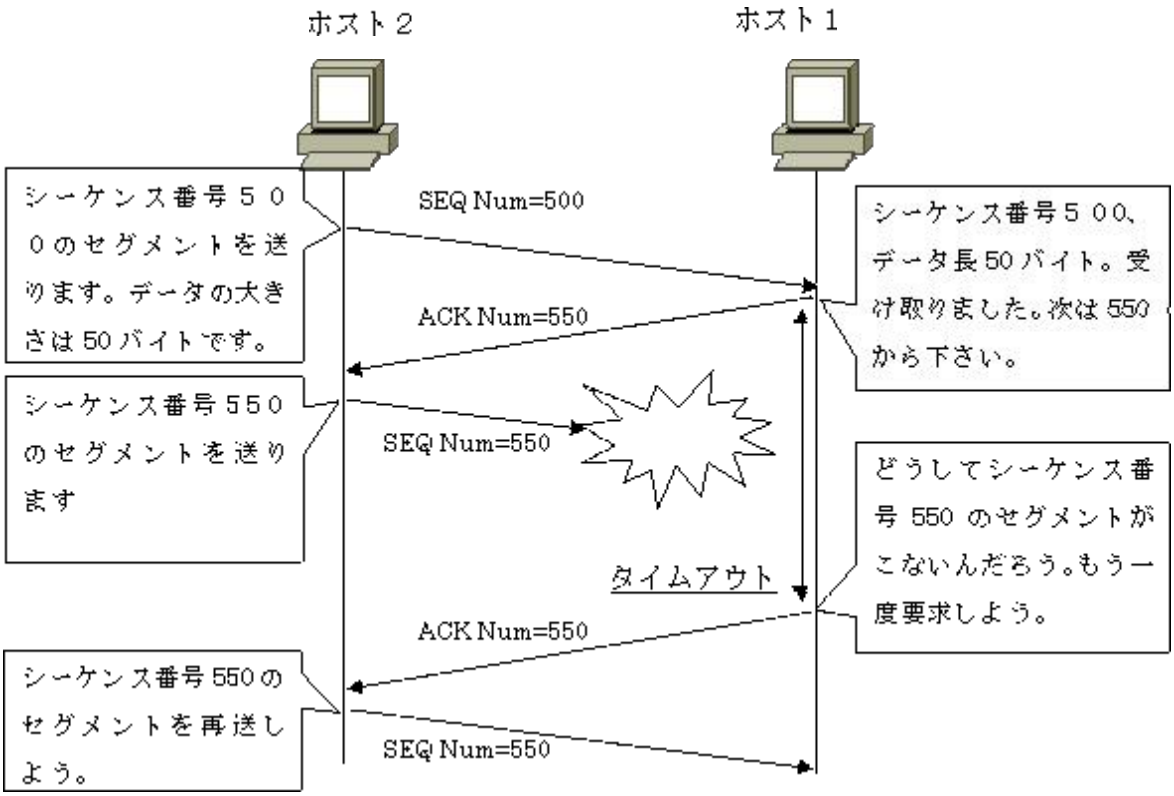


シーケンス番号は、そのセグメントに含まれるデータの先頭のオクテットが、送信すべきデータストリーム全体の中で占める位置を示しています。先頭オクテットが 500 オクテット(バイト)目から 50 オクテットを送信すると、クライアント側は 549 オクテットを受信したことになります。ACK 番号は 550 です。この ACK 番号には、549 オクテットまではすでに受信したという確認応答の意味と、次にデータは 550 オクテット目から送ってくれという意味を併せ持っています。ACK 番号は先ほど 3 ウエイハンドシェークの箇所で説明した ACK ビットフィールドとは違いますので、セグメントヘッダで確認しておいてください。

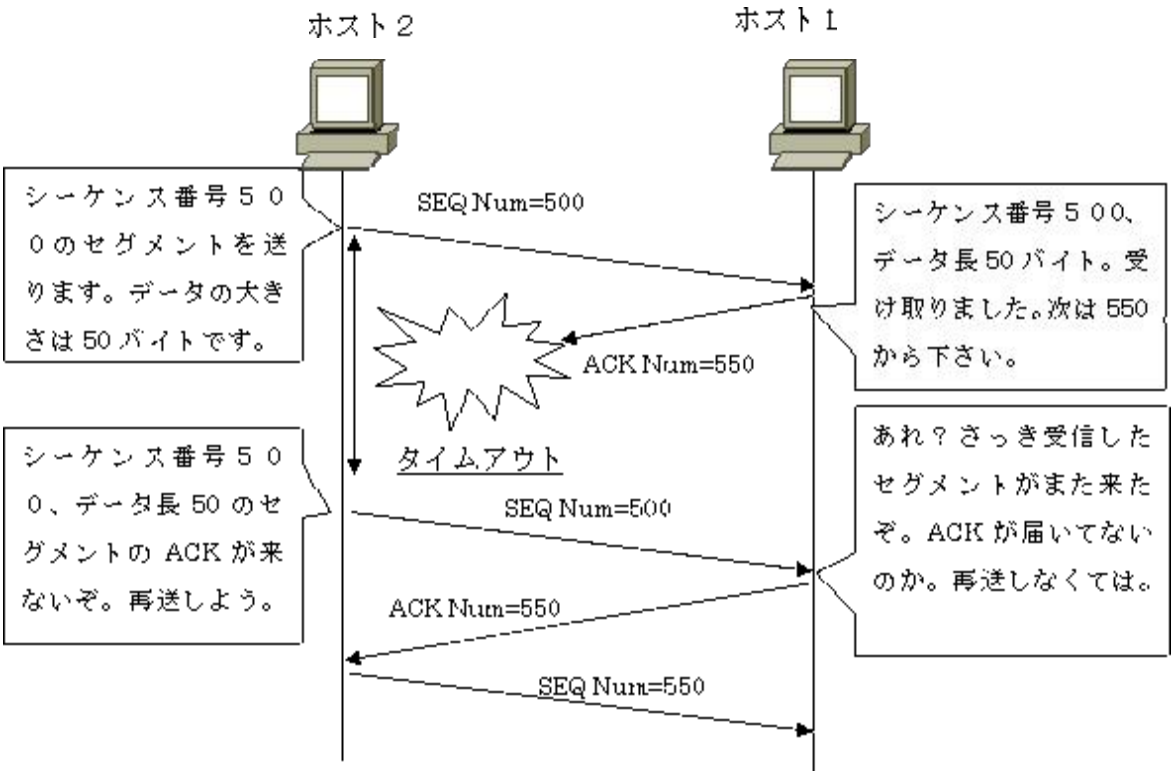
上の説明だと、クライアントからの要求/応答と、サーバからのリプライが交互に行われるように思われるかも知れません。しかし、実際はクライアントの応答を受ける前に、サーバがどんどんデータを送信したり、クライアントがサーバからの何回かのデータ送信に対して、1 回の応答で済ましたりといった効率的なやり取りも可能です。

送信データが途中で壊れてしまった場合、TCP はエラー状態を立て直すことができます。次に 2 つの例を示しています。最初の例は、ホスト 1 が ACK で応答/要求を行った後、一定時間経過しても要求したパケットが送られて来ないので、再度 ACK を送信しています。ホスト 1 は「シーケンス番号 550 のセグメント」を要求し、同時にタイマーをスタートさせます。その後、「シーケンス番号 550 のセグメント」を受け取ることなくタイマーがタイムアウトしてしまったので、再度「シーケンス番号 550 のセグメント」を要求し、これに対して、ホスト２側で再送を行っています。この例の図では、ホスト 2 からの「シーケンス番号 550 のセグメント」がネットワークの途中で壊れていますが、ホスト

１のタイマーの観点からは、ホスト１からの「ACK 番号 550 のセグメント」が壊れても同じです。ホスト１が「ACK 番号 550 のセグメント」を発信したときに、タイマーをスタートさせ、タイマーがタイムアウトするまでにホスト２からの「シーケンス番号 550 のセグメント」を受信できなかったという点では同じだからです。

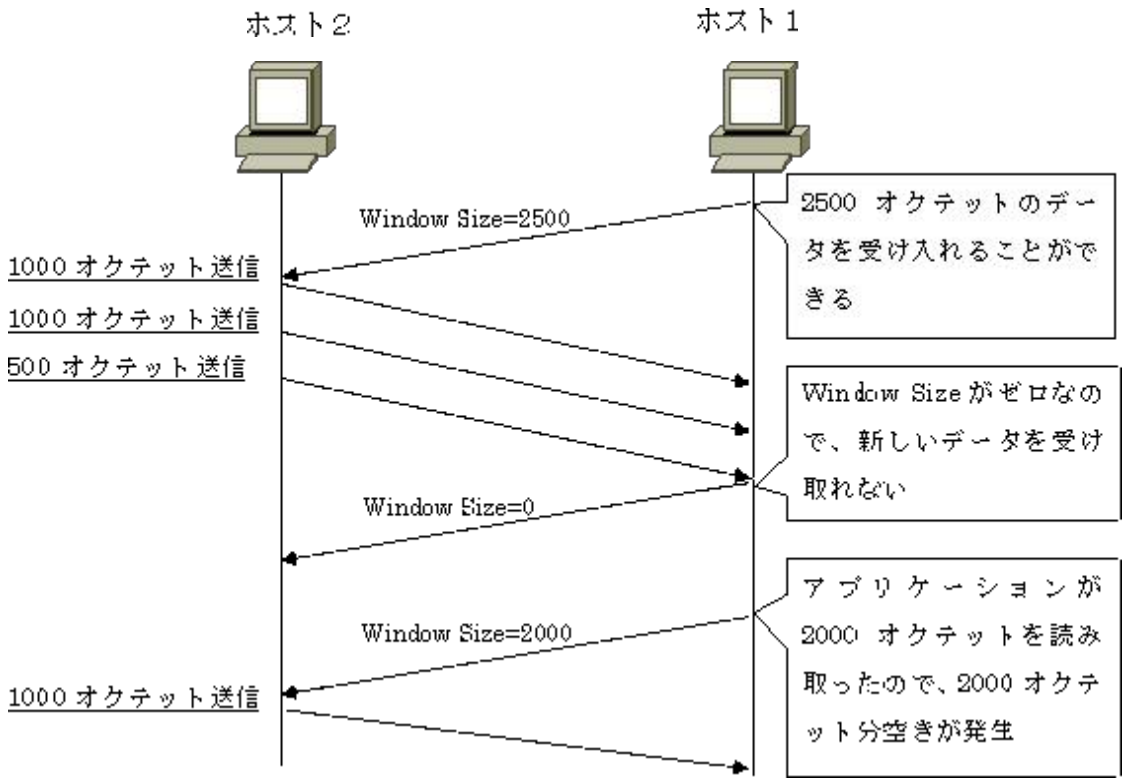


２つ目の例では、ホスト２が「シーケンス番号 500、データ長 50 バイトのセグメント」を送信した後、応答確認のセグメントを受信しないうちにタイムアウトしたので、同じセグメントを再送しています。この場合も、先ほどの例と同じで、ホスト２は「シーケンス番号 500、データ長 50 バイトのセグメント」を送信すると同時にタイマーをスタートさせ、タイマーがタイムアウトするまでに、そのセグメントの確認応答を受け取ることができなかったので、ネットワークに障害が発生したと思って(というより TCP/IP の設計者がそう判断したということですが)、同じセグメントを送信しています。２つ目の例は、ホスト２のタイマーという観点から説明しているだけで、１つ目の例とまったく同じだといってもいいでしょう。また、ホスト２のタイマーという観点から見ると、ホスト２が発信した「シーケンス番号 500 のセグメント」が途中で壊れても、ホスト１の「ACK 番号 550 のセグメント」が途中で壊れても結果はまったく同じです。

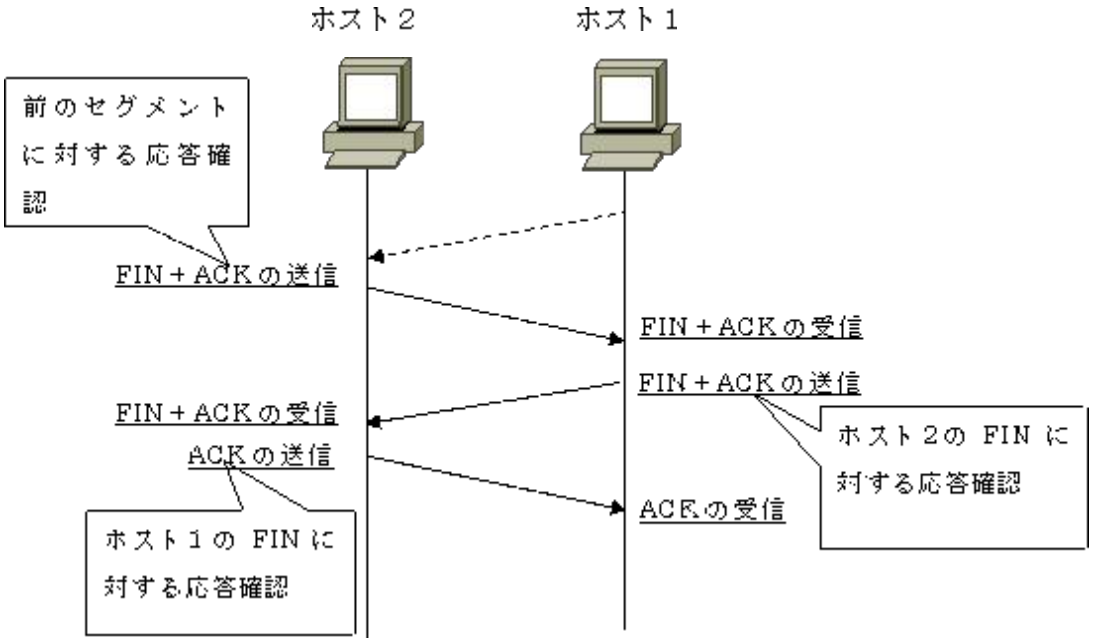


以上の説明は、イメージが浮かびやすいように簡略化しています。しかし、シーケンス番号の仕組みがあまり簡単だとセキュリティ等に問題が生じますので、実際はもっと複雑になっています。通常は ISN(Initial Sequence Number 、初期シーケンス番号)を決めて、ISN に送信したデータのオクテット数を加算しています。ISN の決め方は OS によって異なります。

送信するデータの大きさはどうでしょうか。データの大きさはネットワーク毎に決まっています。その限度は、ネットワークに直接接続しているデータリンク層に聞けばわかります。しかし、相手がいまどれくらいの大きさのデータを受け取ることが出来るかはまた別の問題です。データを受け取るときはいったんある場所に置きます。この一時的に保存しておく場所は大きさが予め決まっています。相手側の TCP から受け取るデータの量、上のアプリケーション層のデータ処理のすすみ具合等でこの一時保管の場所の空き具合が変わってきます。そこで、データを受信し、それに対する返事を出すときに、今自分がどれくらいの大きさのデータまでなら受信可能かを知らせてやります。この自分が今どれだけ受信可能という量のことをウィンドウサイズといいます。データを送信する側がこのウィンドウサイズを無視すると、受信側ではデータの取りこぼしが発生する可能性があります。



このようなことを何回かして、多い場合は何百回、何千回と行って、用が済んだらコネクションを閉じます。コネクションを閉じるときは、コネクションを閉じますよと合図をします。コネクションを閉じる場合はFIN ビットを使います。コネクションは、クライアントからと、サーバから二重に確立していますので、閉じるときもクライアント側とサーバ側からそれぞれ FIN ビットフィールドをセットしたパケットを送信する必要があります。



これで、コネクション型の接続が一回終了したことになります。

６．２　　ベストエフォット型のサービス　UDP

UDP はコネクションレス型です。コネクション型の TCP では、コネクションを確立し、維持/管理し、それを終了するという動作が必要になりますが、UDP ではそんなことは必要ありません。具体的には、3 ウエイハンドシェークの手順に従って接続を確立することもありません。ACK を使って応答確認することもありません。シーケンス番号を使ってセグメントを認証することもありません。Window サイズを使ってフローを制御することもありません。FIN を使って、接続を終了することもありません。つまり、データを送ってもそれを受け取ったという返事はありません。データを送ってもそれが相手に届いているかを確認する手だては用意されていません。相手がどれくらいのデータを受け取る能力があるかも分かりません。こんな信頼性に欠ける通信方式は利用価値があるのでしょうか。実はこれがあるのです。UDP は大変便利な通信の方式です。

UDP は TCP と比較すると簡単な方式ですので、ヘッダが小さくなります。ヘッダが小さいのでパケット全体も小さくなります。ネットワークの帯域幅(リンクの伝送能力の尺度で、一般に bps(bits per second)で表されます)が小さなところでは、小さなパケットの方が好都合です。それから、UDP はパケットが小さいだけでなく、接続確立/維持の仕組みがありませんので、そのためのパケットも必要ありません。また、UDP は仕組みが簡単ですので、UDP を使ったアプリケーションはコンパクトになります。
たとえば、ネットワークを管理したりする仕組みには UDP が最適です。ネットワークを管理する仕掛けは、SNMP(Simple Network Management Protocol)というプロトコルです。SNMP では、ルータやスイッチなどのネットワーク機器やパソコンなどでエージェントとうプログラムを稼働させます。エージェントは自分が稼働しているルータ等のデバイス上で様々な情報（例えば、一定時間あたりのパケットの送受信数、CPU やメモリの稼働状況等々、非常に広範囲の情報）を収集し、データベース化します(このデータベースを MIB(Management Information Base)といいます)。エージェントと通信して、データをかき集めてくるプログラムをマネージャといいます。通常、ネットワーク上の特定のホスト上でマネージャプログラムを動かし、そのホストにデータを集積し、GUI 表示します。SNMP が使うこの方式のことをマネージャエージェント方式といいます。インターネットの方式はほとんどがクライアントサーバシステムという方式に則っていますが、SNMP ではエージェントマネージャシステムを使っています。
SNMP では、マネージャが定期的にエージェントからデータを収集しますので UDP がぴったりです。SNMP は UDP を使うことで接続管理のための手間を省き(送信するパケットの量が減ります)、さらに小さなパケットを使うことで、帯域消費量を少なくすることができます。定期的に通信をするということは、1 回くらいデータの送受信に失敗してもそれほど影響がないということです。したがって、UDP のような信頼性の低い通信方式でも問題はありません。そもそ

も、SNMP を使うのは、ネットワークが細くて(帯域が小さいことを細いなどということもあります)心配だという場合が多いのです。ある高速道路が狭くて渋滞や事故が多発しているので管理のために、警察が沢山のパトカーを出動させたとしたらどういう結果になるでしょうか。たぶん、ますます結果が悪くなります。この際、どうしても必要ならば白バイを走らせるということになるでしょうか。これと同じで、SNMP パケットが流れて却ってネットワークがうまく機能しなくなってしまうというのでは本末転倒ということになります。
それから、UDP を利用するアプリケーションはプログラムが小さくなります。どうしてもプログラムを小さくしないといけない場合もあります。例えば、ハードディスクを使えない場合などです。ハードディスクは可動部分があるので連続運転をすると故障の確率が高くなります。そこで、何年もの間稼働し続ける必要のあるデバイスではハードディスクを使いません。例えば、ルータなどはいったん動かしたら何年間もそのまま動かし続けます。ルータには通常ハードディスクはついていません。このようなものをディスクレスマシンといいます。ルータは OS や設定ファイルなどを ROM や RAM、フラッシュメモリ、NVRAM(不揮発性の RAM)などのチップに保存して、利用しています。その他にも、ルータにとって、どうしても必要なプログラムもあるでしょう。たとえば、TFTP というプロトコルです。ルータは、設定情報をサーバに保存したりサーバからダウンロードしたり、あるいは新しい OS をダウンロードしたりする際に、ファイル転送プロトコルを必要とします。ファイル転送プロトコルとしては、FTP(File Transfer Protocol)がありますが、これは TCP を使いますので、大きな容量のプログラムになります。チップにインストールするにはもっと単純なプロトコルの方がいいんです。プロトコルを簡略化(Trivial)するためには、UDP を使う必要があります。そこで、UDP を使って、FTP を簡略化したものが TFTP(Trivial File Transfer Protocol) です。

７　インターネット発展の牽引役　アプリケーションサービス

6.1では、インターネットサービスの例としてTelnetを使って説明しましたが、Telnet以外のも様々な種類のインターネットサービスがあります。皆さんに一番なじみがあるのがメールとWebでしょう。メールはインターネットサービスのプロトコル名としてはSMTP（Simple Mail Transfer Protocol）、WebはWWW（World Wide Web、世界的に張り巡らされた蜘蛛の巣の意味）とか、HTTP（Hyper Text Transfer Protocol）とか呼ばれます。
皆さんはWebサーバに接続する際、Webブラウザを使います。たぶん、Webブラウザのアドレスバーでは***.co.jpのような形で宛先サーバを指定しているでしょう。また、メールを使う場合も、アカウント名@***.co.jpのようにして相手先を指定します。この「***.co.jp」のようなものをドメイン名といいます。Webは、実際にはこのドメイン名の後に、そのドメイン名で指定されたコンピュータ上に存在する何々のページという形式で特定のWebページを指定します。メールの場合は、ドメイン名でメールサーバを指定します。ここでは詳しいことは省略します。いままでの話で、どうも相手のコンピュータの指定は、IPアドレスで行わないといけないのかなという点は理解していただけたものと思います。しかし、実際の利用では、ユーザはドメイン名を使うわけですから、ドメイン名をIPアドレスに変換する仕掛けが必要なことが分かっていただけると思います。この仕掛けのことをDNS（Domain Name Service、あるいはDomain Name System）といいます。DNSが発明されて相手を名前で指定することが出来るようになりました。
メールは離れたネットワークのユーザ同士を結びつける役割を果たします。初期のインターネット開発のエネルギーは如何にしてメールシステムを便利にするかということだったといっていいでしょう。Webはもともと情報閲覧用の単純なシステムでしたが、インターネットの商用利用のきっかけとなりました。インターネットの爆発的な人気はWebの発展によるものといっても言い過ぎではないでしょう。Webシステムは現在もものすごい勢いで進化しています。

８．標準化の仕組み

ここまで、読み進んでくると多くの方は、インターネットを非常にうまく管理された巨大な構築物のように想像するのではないでしょうか。しかし、インターネットを強権的に支配している団体はどこを見渡しても見つかりません。インターネットは、誰か少数の人間が支配しているような世界ではありません。インターネットは各種のホストやネットワークが自主的にプロトコル規格に従っているだけです。では、そのプロトコル規格とはどんなものでしょうか。インターネットの中心に IETF(インターネット推進専門委員会)というオープンな組織があります。IETF はネットワーク設計者や、製品ベンダーや研究者の集まったオープンな組織です。IETF では、刊行物の形で規格書を発行し、インターネットに参加するものはその規格書に従うという形で標準化が進められます。IETF の発行する規格書は RFC(Request For Comment)と呼ばれます。

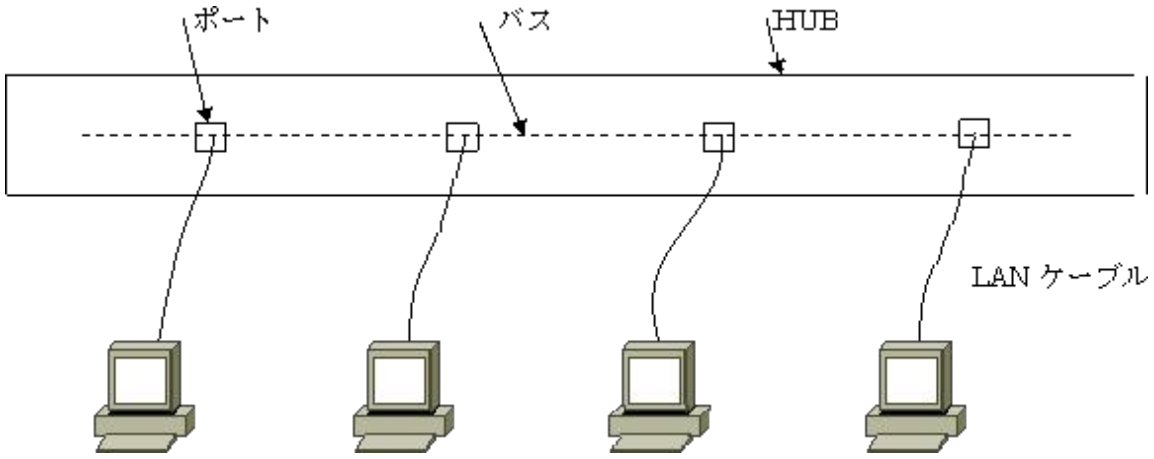
APPENDIX

■　LAN

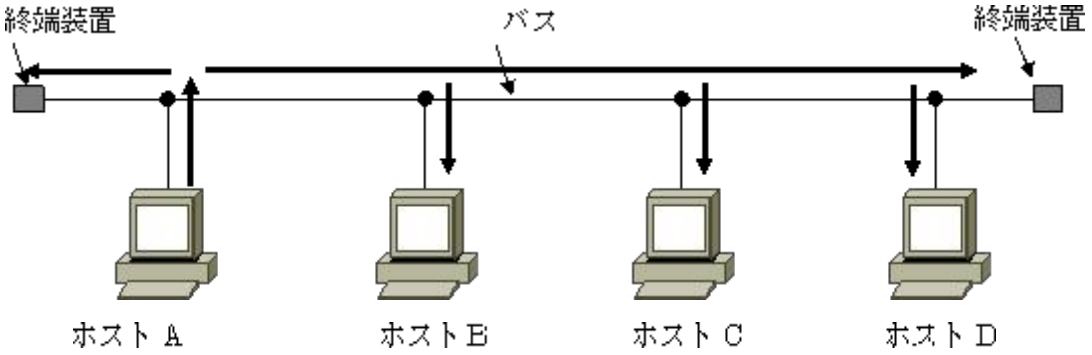
LAN(Local Area Network、ローカルエリアネットワーク)とは、数キロメートルの範囲内で使うように設計されたネットワークです。従来は公道を横切らないネットワークを想定していますが、無線 LAN が登場し、公道をまたがった LAN も可能になりました。LAN は一般に、イーサネットやトークンリング、FDDI などのネットワーク形態をとります。

■　イーササーネットと HUB

HUB はイーサネットの 10Base-T という仕様で使っている集線措置です。装置内にバス構造のネットワークがあります。バスとは、お風呂(bath)のことではありません。バスとは、もともとの意味は、bus、もっと言うと omnibus、つまり乗り合いバスのことです。みんなで乗り合いバスのように協同で使うネットワークをバス形式といいます。バス形式のネットワークを簡単に示します。点線で描いたネットワークがバスです。装置の中にバス構造のネットワークを実装したのが HUB というネットワーク装置です。



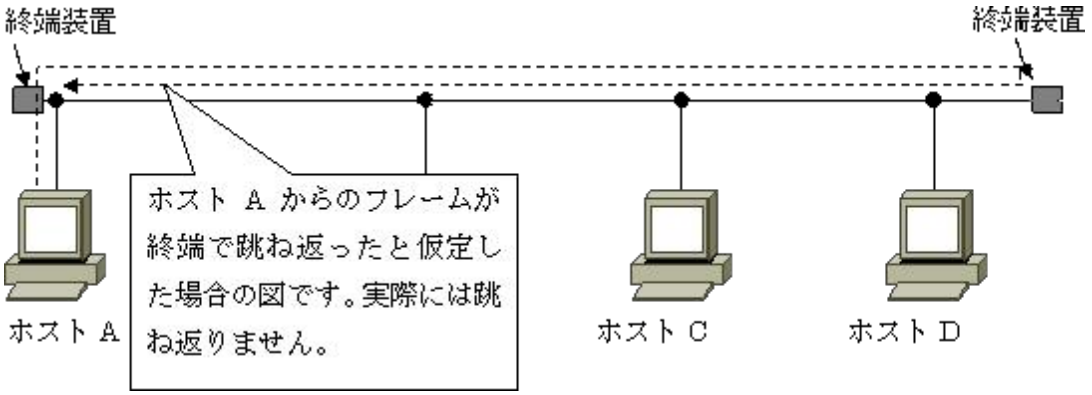
各ホストは、LAN ケーブル(10Base-T ケーブル、あるいはツイストペアケーブルなどと呼ばれます)で、集線装置である HUB のポートに接続されています。



バス形式のネットワークでは、そのネットワークに属するホストがそのネットワークを共有します(Shared Network という言い方もあります)ので、たとえばホストAがホストBに向けてフレームを送信しても、そのフレームはホストCにもホストDにも届きます。したがって、ホストAとホストBが通信をしている間はホストCとホストDはネットワークを使えないことになります。このバス形式のネットワークを使ってデータリンク層のレベルで通信を制御しようとする方式の代表がイーサネットです。イーサネットはXerox社のパロアルト(PARC、Palo Alto Research Center)のコンピュータサイエンス研究所に在籍していたメトカフ博士によって発明されました。博士はこの発明の特許権をXeroxから買取り3Com社をスタートさせています。

イーサネットでは、CSMA/CD 方式というデータの転送方式を採用しています。CSはCarrier Sense、MAはMultiple Access、CDはCollision Detectという意味です。MAとはみんながネットワークを使っていいということです。ホストAもホストBもホストCもホストDもネットワークにアクセスすることができます。CSとはネットワークにフレームを発送するときは、他のホストがネットワークを使っていないかよく検査しなさいということです。誰も使っていなければネットワークにフレームを流し込むことができます。しかし、ホストAと同時にホストCもCarrierをSenseしていて、「よし誰も使っていないぞ」と思い込んで、殆ど同時にネットワークにフレームを流し込むこともあります。そのような場合は、フレーム同士が衝突して壊れてしまいます。ですから、フレームをネットワークに流し込んだら、それでいいと安心しないで、その後もCollision(衝突)がないかよく調べないといけません。そして、衝突を検知したらジャム信号を発信し、しばらくの間待ってから再度フレームを送信します。再送したらまた衝突ということがないように、各自乱数発生器を使って待ち時間を決めることになっています。これがCSMA/CD方式です。

ではどうやって衝突を検知するのでしょうか。フレームが衝突すると、規定外の小さな破片になります。このような破片(衝突破片)をラント(runt)といいます。フレームの衝突破片を受信したら衝突が発生したと思っていいでしょう。では、いつラントを受信したら衝突と認識すべきでしょうか。



上の図はホスト A から発信されたフレームが終端で跳ね返って来た場合の想像図です。ただし、実際には跳ね返ってきませんので勘違いしないでください。跳ね返ると衝突が発生しますので、終端装置で跳ね返りを防いでいます。終端装置はつまり、跳ね返りを防止する装置なのです。したがって、終端装置で跳ね返ってくるなどということはありませんし、あっては困ることです。しかし、ここでは跳ね返ってくると想像してください。ネットワークの一番端のホストから発せられたフレームがもし戻ってくるとしたらかかる時間をラウンドトリップ時間(往復時間です)といいます。ラウンドトリップ時間が経過するよりも早くラントを受信したらそれは、他のホストから発せられたフレームが途中で衝突をし、その結果ラントが発生したものと考えるべきでしょう(ホスト A が発信したフレームとの衝突と考えるべきでしょう)。

そこでイーサネットでは、ネットワークの大きさをある一定の大きさに限定し、その最大のネットワークを仮定したときのラウンドトリップ時間を計り、その時間以内にラントを受け取れば衝突とみなすということにしています。この時間をスロットタイムといいます。このことは、イーサネットの直径をスロットタイムの半分以上に拡張すると衝突の検知ができないということを意味します。標準イーサネット(10Mbps)と、ファーストイーサネット(100Mbps)では、この時間は 512 ビット時間とされています。512 ビット時間は、512 ビット長のフレームが最大サイズのイーサネットシステムの両端のあるステーションを往復するのにかかる時間に、若干の時間を加味した時間です。標準イーサネットは 10Mbps ですから、51.2 マイクロ秒、ファーストイーサネットは 100Mbps ですから、5.12 マイクロ秒ということになります。

フレーム送信後、スロットタイムが経過するまでの間が衝突を心配する時間帯です。スロットタイムの半分が経過した時点で、フレームはイーサネットネットワークの端まで到達します。ネットワークの他のステーションがキャリアセンスできずに、フレームを送信した場合、衝突破片がスロットタイム内に必ず到達します。スロットタイムの半分が経過した後は、他のステーションは、フレームを送信しようとしてキャリアセンスをすれば、必ずキャリアの検出ができますので、フレームの送信を行わないはずです。したがって、衝突破片を受信することなしに、スロットタイムが経過すれば、最初のステーションもも

はや衝突について心配する必要がなくなります。
　では衝突破片についてはどう考えればいいでしょうか。衝突破片とは規定よりも短いフレームということですので、最初に送信したフレームがあまりに短すぎると、衝突破片かどうかの区別がつきません。そこで規定では、最小のフレームの長さは512ビット(64バイト) ということにしています。フレームヘッダ(とトレーラ)は合計で18バイトですから、データフィールドの長さ(TCP/IPを使う場合はパケットの長さ)は46バイトということになります。

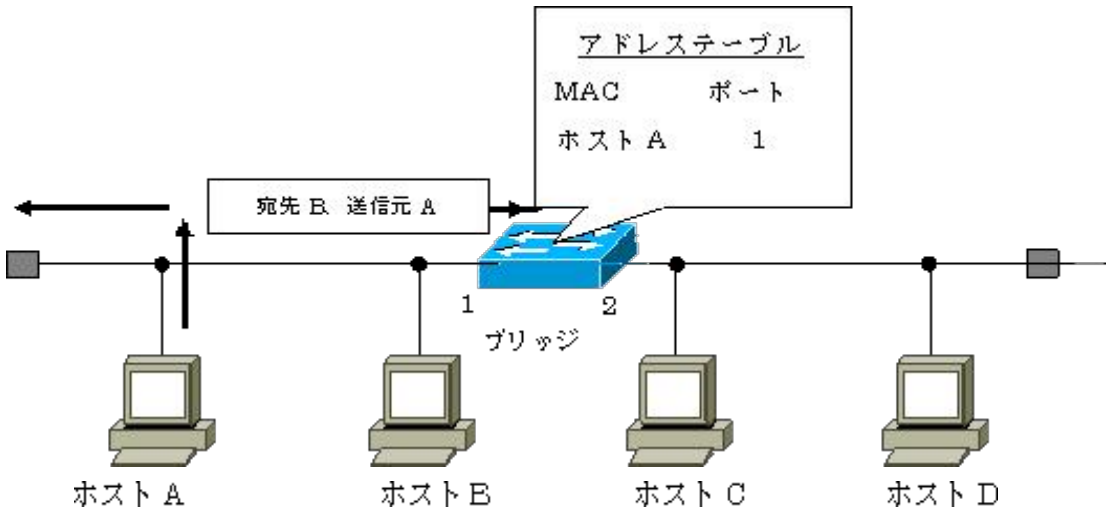
　最小のフレームを512ビットと規定して、スロットタイムを512ビット時間と規定すると、正常な衝突(イーサネットにとって衝突は通常はエラーではありません)は、最初の512ビット(つまり通常のフレームでしたらフレームの先頭部分です)を送信している途中に発生(検知)します。そして、512ビット時間後はもう衝突は発生しないものと考えてもいいというのがイーサネットの約束事ですので、512ビット時間以降に衝突が発生すると困ったことになります。

　イーサネットネットワークの直径をスロットタイム時間の半分よりも大きくしてしまうと、衝突が発生しても衝突破片はスロットタイム経過後に検出されます。このような衝突を遅れ衝突といいます。遅れ衝突は再送されません。なぜでしょうか。512ビット時間経過前の衝突はイーサネットにとっては正常な衝突ですので、フレームを再送するだけの話です。この仕組みを維持するために、スロット時間が経過するまでは、送信したフレームのコピーをバッファに保存し、再送時はそのコピーを送信します。スロットタイムが経過すれば、バッファに入れたデータはもう使用しないはずですので、破棄してしまいます。遅れ衝突に対して再送することになると、このメカニズムを維持することができません。つまり、イーサネットネットワークの直径をスロットタイム時間の半分以上に設計してしまうと、イーサネットは正常に機能しないということになります。

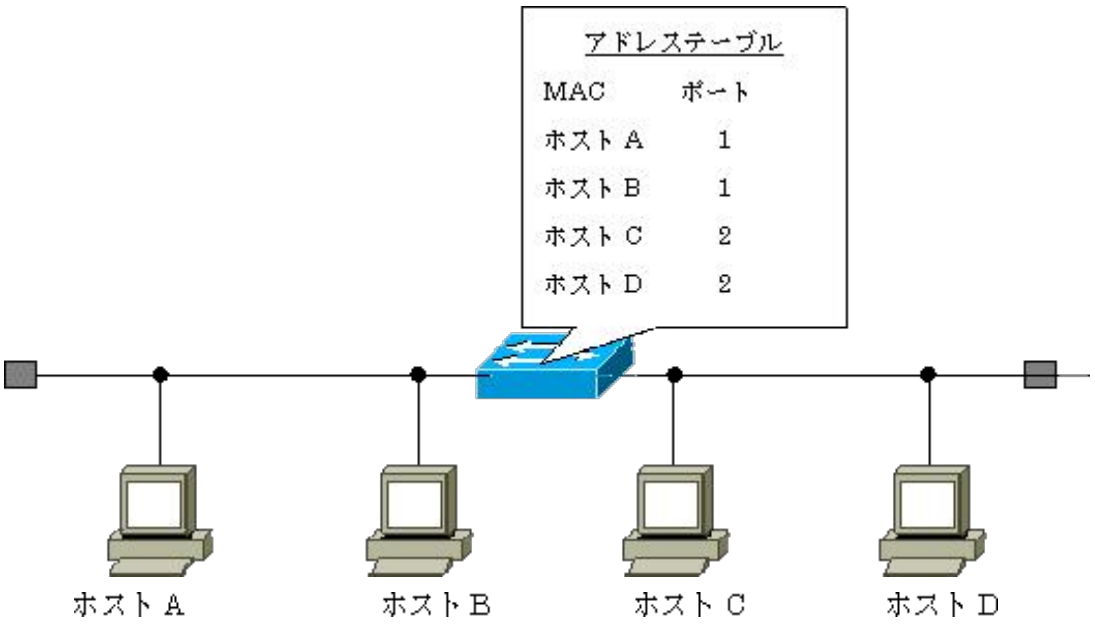
■　スイッチングHUB

　HUBはネットワークを単純に共有するための装置で、ブロードキャストだけでなく、マルチキャスト、ユニキャストのフレームもすべてネットワーク全体にいきわたります。
　ネットワーク機器としてHUBを使うことにはいくつかの問題がありますが、その代表が衝突とセキュリティです。その問題点を解消するために開発されたのが、スイッチです。一般的にはスイッチングHUBと呼ばれています。
　スイッチングHUBはもともとブリッジと呼ばれていました。その後、従来のブリッジとの性能の違いを強調するためのベンダーの営業戦略の一環として、スイッチ（あるいはスイッチング HUB)という言葉が使われるようになりました。原理的にはスイッチングHUBもブリッジと同じです。

HUB はポートの先にどんなステーションが接続されているか知る能力がありません。そこで、フレームを受信した以外のすべてのポートから、フレームを送り出します。ブリッジには学習機能がついています。ブリッジは学習機能を使ってポートの先に接続されたステーションを学習し、あて先ステーションに接続されたポートだけからフレームを送り出します。



上の図で説明します。ホスト A は宛先 MAC アドレスがホスト B のフレームをネットワークに送り出しています。ブリッジはポート 1 でこのフレームを受信し、ポート 1 の先にホスト A(の MAC アドレスのインターフェース)が存在していることを認識し、アドレステーブルにホスト A(の MAC アドレス)とポート 1 のペアのエントリを追加します。このような作業が何度か行われると、最終的にブリッジのアドレステーブルは次のようになります。

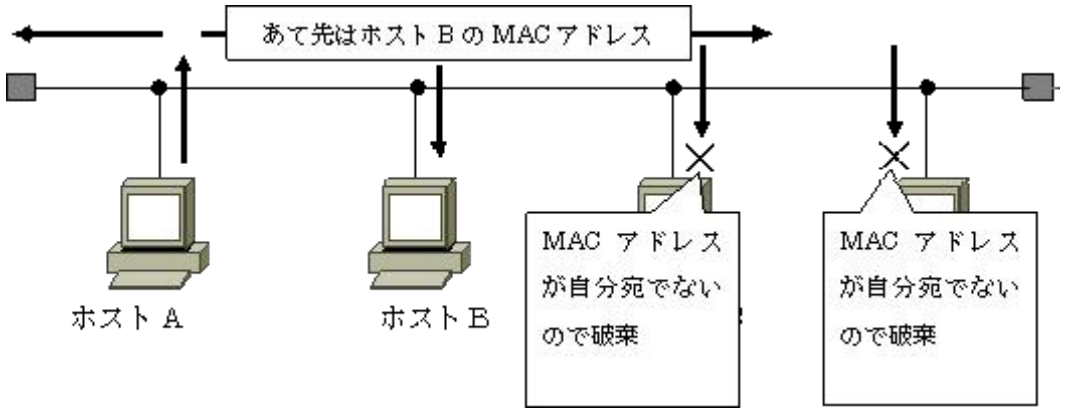


このアドレステーブルを適用すると、ホストAとホストBの間のフレームはブリッジによって破棄されポート2から送出されることはありません。したがって、ホストAとホストBの間でフレームの交換がなされている間、ホストCとホストDの間でもフレームの交換を行うことができます。ブリッジは、あるポートからフレームを受信した場合、そのフレームを受信ポート以外から送り出すかどうかアドレステーブルを参照して判断し、宛先が受信ポートの先にあれば破棄し(フィルタリングといいます)、受信ポート以外のポートの先に宛先があれば転送(あるいはフラッディング)します。どこのポートから転送したらいいでしょうか。受信ポート以外の全てのポートから送り出すか(フラッディングといいます)、あるいは宛先MACのある特定のポートから転送するかということになります。これは、デバイスの機能あるいは、ネットワーク管理者の設定によって決まってきます。

従来のブリッジを高機能にしたものがスイッチングHUBです。機能的には同じものですが、スイッチメーカは自社のデバイスの高機能性をアピールするためにブリッジという名前よりももっとインパクトのある名前が必要だったのでしょう。メーカは新しい高機能なブリッジにスイッチングHUBという名前をつけました。従来のブリッジはソフトウエアで動作しますが、スイッチングHUBはブリッジと同じ機能をASIC(Application-Specific Integrated Circuit)という半導体技術で実現しています。スイッチングHUBはハードウェアによって動作していますので、ブリッジに比べてパフォーマンスが格段に向上しています。スイッチASIC技術は絶え間ない進化を続け、現在はスイッチチップセットが市場に投入され、ポート密度が高く、高パフォーマンスなスイッチが実現されています。

以上の説明は宛先 MAC アドレスが特定のインターフェースを指している場合、つまりユニキャストアドレスの場合です。ブリッジ(スイッチング HUB)は、マルチキャストフレームとブロードキャストフレームはフラッディングします。

Hub を使うとブロードキャスト、マルチキャスト、ユニキャストのいずれのフレームもネットワーク全体にいきわたりますが、通常の場合は NIC が、MAC アドレスが自分宛でないフレームを破棄してしまいます。



ただ、このように NIC が、MAC アドレスが自分宛でないフレームを破棄しないで、それを受信し IP 層まで上げて欲しいという場合があります。たとえば、ネットワーク管理をするためにはホスト C 上にインストールしたプログラムで、ホスト A とホスト B の間のフレームのやり取りを監視できると好都合です。そのためには、MAC アドレスが自分宛でないフレームも受信して IP 層まで引き上げることができなくてはなりません。

無差別(無制限、プロミスク、promiscuous)モード(mode)を使うと、NIC は自分以外をあて先 MAC アドレスとするフレームを受信します。LAN アナライザなどをインストールすると、デフォルトで無差別モードに設定され、自分宛でないフレームも受信し始めます。この技術は、ネットワークトラフィックを調べてネットワーク障害を検出しようとする際には非常に有力な手段を提供しますが、クラッカーにも非常に強力なクラッキング手段を提供することになります。LAN アナライザを使うと、パスワードやデータの中身が見えてしまうからです。盗聴(スニッフィング)を防止するためには、データを暗号化することが大切ですが、スイッチング HUB を活用することもできます。スイッチング HUB の 1 つのポートを 1 つのホストで独占すれば、スニッフィングに対して効果的です。

■　イーサネットはどこまで伸ばせるのか

　イーサネットでネットワークの大きさを制限する要素は２つあります。１つは伝送距離が長くなるとデータの電気的な波形が乱れるということです。もう１つは、ネットワークの大きさが大きくなりすぎるとイーサネットのプロトコルである CSMA/CD 方式が機能しなくなってしまうということです。

物理的な制限を回避する方法がリピータです。しかし、リピータを使用しても、CSMA/CD 方式の制限を越えることはできません。ブリッジ(スイッチ)を使うと、物理的な制限を回避することも、CSMA/CD 方式による制限を回避することもできます。さらに、イーサネットでも CSMA/CD 方式を使わない方式(全二重モード)になると、もともと CSMA/CD 方式に基づく制限とは無関係ということになります。

●物理的な理由にもとづく距離制限
　フレームは電気信号として送信されますから、伝送距離が長くなると減衰します(ケーブルの信号伝送特性)。イーサネットでは、様々なメディア方式(10BaseT など)が規定され、各メディアで使用するケーブルのタイプ毎にケーブルの最大長が決められています。ここでポイントになるのが、どこまで波形が乱れずに(読み取り可能な状態で)伝送されるかということですので、途中で波形を増幅し整形する装置を挟めば、伝送距離を延長することができます。この装置がリピータです。イーサネットでは HUB(ハブ)がリピータの役割を果たします。ただし、HUB を使うと、HUB の転送処理によって遅延が発生することになりますので注意してください。

●CSMA/CD 方式にもとづく距離制限
　スロットタイムとの関係で、LAN リンクの大きさは制限を受けることになります。

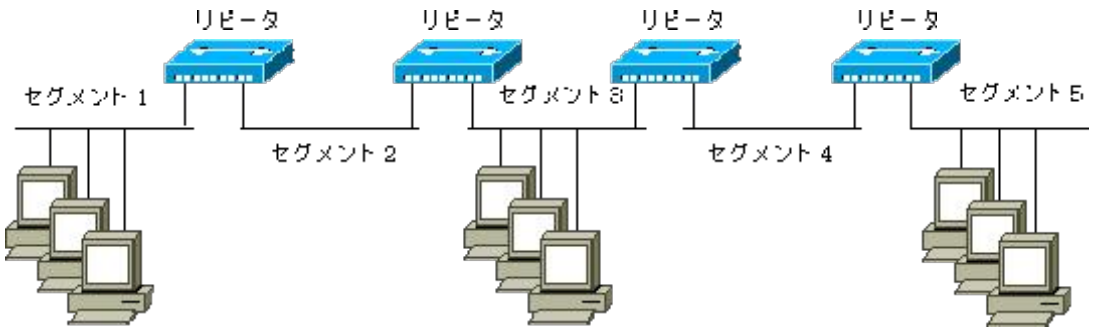
●半二重モードと全二重モード
　イーサネットには、半二重モードと全二重モードという２通りのモードがあります。半二重モードについて CSMA/CD 方式が機能しますが、全二重モードの場合は、CSMA/CD 方式そのものが当てはまりません。イーサネットというと、CSMA/CD 方式とほとんど同義語と思われていますが、全二重モードという CSMA/CD 方式と何のかかわりもない方式が開発されました。これを果たして、イーサネットと呼んでいいものやらいささか疑問なのですが、一応全二重のイーサネット方式と呼ばれています。したがって、全二重の場合にはスロットタイムにもとづく距離制限は当てはまらないということになります。

全二重のセグメントでは、ケーブル長を制限するのは信号伝送特定だけですので、ケーブルの種類によっては半二重の場合よりもかなり距離を延長することができます。ツイストペアケーブルの距離特性はケーブルの伝送特性による制限ですので、全二重にしても最大距離が長くなるわけではありません。これに対して、光ファイバは半二重動作時におけるタイミングの調整のために距離制限を受けますので、全二重で使った場合は、距離が大幅に長くなります。光ファイバを全二重で使えば、伝送特性による制限だけとなります。光ファイバにはシングルモードとマルチモードがありますが、シングルモードの方が信号をより遠くに伝送することができます。

●リピータによるネットワーク延長

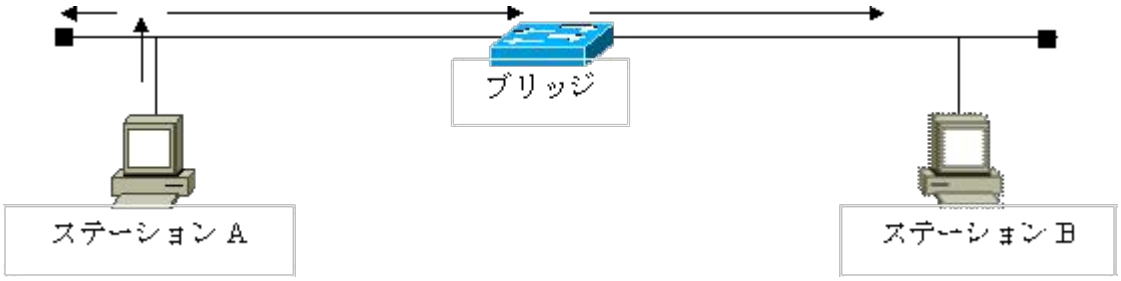
リピータを使うと、物理的な理由にもとづくネットワークの大きさの制限を回避することができます。ただし、リピータを使えば、スロットタイムの制限が許す限りいくらでもネットワークを大きくしていいかというとそうではありません。銅線や光ファイバ上を伝送される場合の伝送遅延、リピータ(ハブ)での転送処理にもとづく転送遅延などを考慮して使っていいリピータの数やセグメントの数は制限されています。１０Mbps のイーサネット(標準イーサネット)では、この制限は「5/3/1 規則」といわれます。

「5/3/1 規則」では、リピータを使って接続できるセグメントの数を５つ、そのうちデバイス(エンドステーション)を接続できるのは３つのセグメントとしています。したがって、２つのセグメントはリピータ同士を接続するだけで、ホストを接続することはできません。そして、この５つのセグメントで１つのコリジョンドメインを形成します。



●ブリッジ(スイッチ)によるネットワーク延長

ブリッジはコリジョンドメインを分割し、衝突を転送しません。また、スロットタイムは再確立します。ステーション A から送信されたフレームが壊れずにブリッジに到達した場合、ブリッジはフレームをバッファに保存し、コピーを他のポートからステーション B に向けて送信します。このフレームがステーション B に到達する直前に、ステーション B がフレームを送信したとすると、ブリッジはスロットタイム時間内に(フレームの送信中に)ジャム信号を受け取ることができますので、バッファに保存してあるフレームを送信することができます。つまり、各セグメント単位でスロットタイムを持っていることになります。ということはセグメントごとに物理的な制限と、CSMA/CD 方式のスロットタイムに基づく制限を守っていれば、理論的にはいくらでもイーサネット LAN を拡大することができます。



ただし、実際にはいろいろの条件が絡んでくるので、イーサネット LAN の大きさには制限があります。ブリッジを何段も接続すると、ブリッジの転送処理に起因する遅延が蓄積することになります。また、ブリッジはマルチキャストフレームとブロードキャストフレームをフラッディングし、その分帯域を消費します。LAN が大きくなりすぎると、ネットワークは、マルチキャストフレームやブロードキャストフレームのフラッディングに起因するコリジョンの頻発で、機能麻痺に陥る可能性があります。コリジョンは、マルチキャストフレームやブロードキャストフレームだけでなく、セグメントを越えたユニキャストフレームによっても引き起こされます。

●ファーストイーサネット

100Mbps のファーストイーサネットは 10Mbps と比べると 10 倍の速度で動作しますので、すべてのタイミングが 1/10 になります。スロットタイムは 5.12 マイクロ秒になりますので、レイトコリジョンを避けるためにはネットワークの大きさを小さくしなくてはなりません。ファーストイーサのネットワークはリピータあるいはブリッジ(スイッチ)で拡張しない限り 200m 位の大きさになります。

●ギガビットイーサ

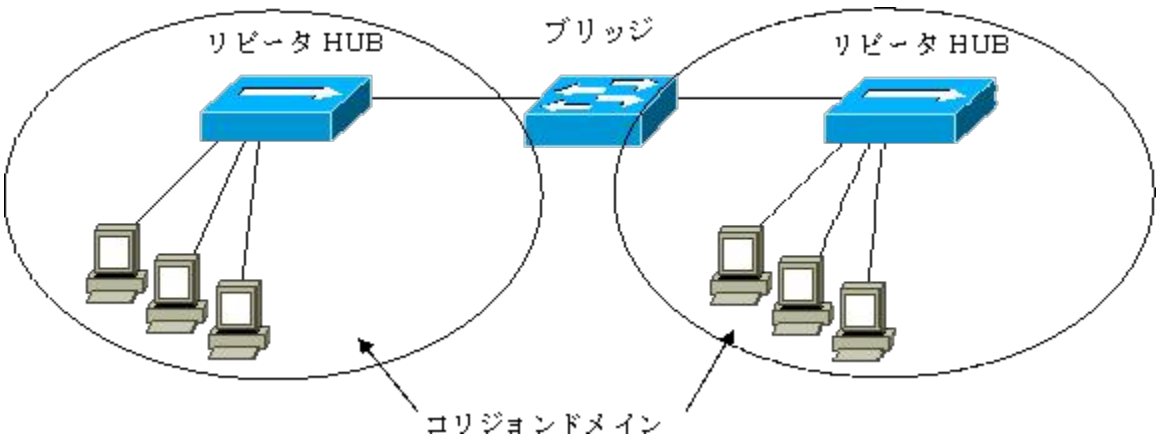
ギガビットイーサのスロットタイムを考える場合に、そもそもギガビットイ

ーサで半二重モードがありうるのかという疑問が浮かびます。半二重モードで1000Mbps の速度が出せるのかです。1000Mbps で動作するには光ファイバでなくてはいけないのではないか。光ファイバを使うのなら、全二重だろうと考えるかも知れませんが、ツイストペアケーブル(銅線)を使って CSMA/CD 方式モードで動作する 1000BaseT という方式があります。

ギガビットイーサで半二重モードを採用したとすると、スロットタイムは100Mbps の 5.12 マイクロ秒の 1/10 で 0.512 マイクロ秒ということになるのでしょうか。だとすると、ギガビットイーサの場合の LAN の直径(ダイアミータ、diameter)は 20m 位になります。これでは使い物になりません。この計算の元になっているのは、最小フレームの単位が 64 バイト長ということです。しかし、フレームが短い場合には、その後ろに詰め物をしてやればフレームが長くなります。その結果、フレームの送信をし終わるまでの時間が長くなります。これを「キャリア拡張」といいます。ギガビットイーサのキャリア拡張では、最小のフレームサイズを 4,096 ビットにまで長くしています。その結果、スロットタイムを 4.096 マイクロ秒にまで伸ばすことができました。

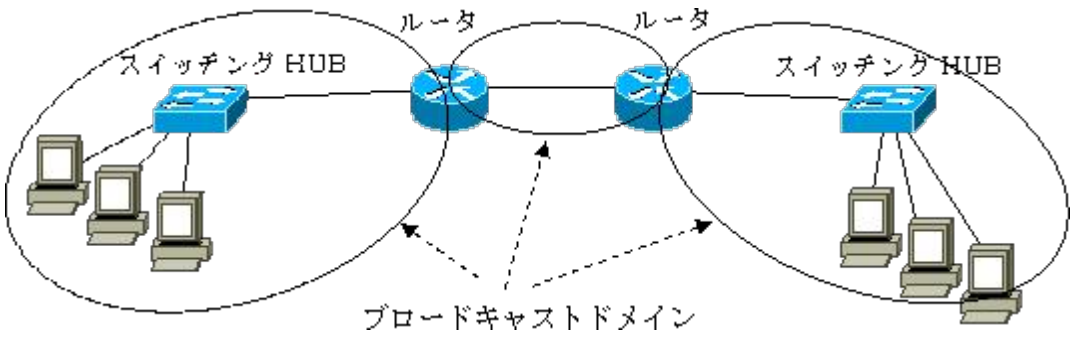
■　コリジョンドメイン

コリジョンドメインとは、CSMA/CD 方式のネットワークにおいて衝突(Collision、コリジョン)が発生する範囲(ドメイン)です。



■　ブロードキャストドメイン

HUB やスイッチング HUB では、ブロードキャストフレームをブロックすることができません。ブロードキャストフレームが届く範囲をブロードキャストドメインといいます。ブロードキャストフレームをブロックするのはルータですので、ルータはブロードキャストドメインを分割するデバイスということになります

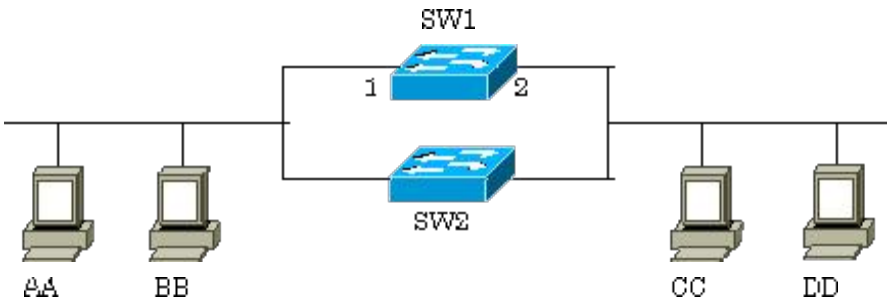


■　半二重通信と全二重通信

全二重通信は、送信と受信を同時に行うことができます。これに対して、半二重通信では 1 本の媒体を共有して送信と受信を行いますので、送信と受信を同時に行うと衝突が発生してしまいます。

■　スパニングツリープロトコル

スイッチ(ブリッジ)によって構成されたネットワーク（スイッチドネットワーク）を障害に強いネットワークにするためには冗長構成にする必要があります。しかし、スイッチで構成されたネットワークに冗長構成を導入するとループが発生します。



上のようなもっとも単純な冗長構成でも問題が発生します。ホストAのMACアドレスをAA、ホストBのMACアドレスをBB、ホストCのMACアドレスをCC、ホストDのMACアドレスをDDと仮定します。ホストAからブロードキャストフレームが発信されたとします。このブロードキャストフレームをSW1が受信すると、SW1のアドレステーブルには「MACアドレスAA、ポート1」のエントリが追加されます。そして、そのフレームはポート2からフラッディングされます。そのすぐ後にSW2もポート1からブロードキャストフレームを受信します。そして、「MAC アドレス AA、ポート 1」のエントリを追加します。そして、SW2もこのフレームをポート2からフラッディングします。そして、そのすぐ後に、先ほどSW1がポート2からフラッディングしたフレームをポート2で受信し、アドレステーブルを「MACアドレスAA、ポート2」と書き換え、さらにポート1からフラッディングします。SW1もSW2がポート2からフラッディングしたフレームをポート2で受信し、アドレステーブルを「MACアドレスAA、ポート2」と書き換え、さらにポート1からフラッディングします。これが繰り返されると、アドレステーブルが収束できず、またその間にブロードキャストフレームが嵐のように増加していきます（ブロードキャストストーム）。マルチキャスト、ユニキャストでも不都合が起こりますが、ここでは省略します。

解決策は、物理的な冗長構成に対して、ループのないツリー構造(スパニングツリー)を抽象的に構成し、そのツリー状のネットワーク上でフレームの交換を行うことです。このアルゴリズムをスパニングツリーアルゴリズムといいます。スパニングツリーアルゴリズムを利用して、ループのない抽象的なネットワーク上でフレームの交換を行うためのプロトコルをスパニングツリープロトコルといいます。

■　スパニングツリーアルゴリズムの概略

スパニングツリーアルゴリズムの概略について説明します。スパニングツリープロトコルはIEEE802.1Dによって標準化されています。障害に強いネットワークを構築しようとすると、ネットワークに冗長性を持たせて、どこかが故障してもネットワーク全体としては機能するという状態にしなくてはなりませ

ん。しかし、ネットワークに物理的な冗長性を持ち込むと、ブロードキャストストームが起こります。これを解決するのがスパニングツリーアルゴリズムです。スパニングツリーアルゴリズムは、物理的にループのあるネットワーク上にツリー構造を構築し、そのツリーに沿ってフレームを送信するアルゴリズムです。スパニングツリーとは、「枝の広がった木」という意味です。ネットワークに参加しているすべてのLANに対して枝を広げている木を想定してください。この木はネットワーク上のすべてのLANをつなげているけれども、あくまで木ですから、ループはありません。ネットワーク上のすべてのLANをツリー構造で接続し、そのツリーに沿ってフレームを送信すれば、ブロードキャストストームは発生しません。

スパニングツリーアルゴリズムでは、スパニングツリーを構築し、維持管理するためにBPDU(Bridge Protocol Data Unit)というフレームを使います。スパニングツリーを構築するために使うBPDUを設定(Configuration、コンフィギュレーション)BPDU、ネットワークトポロジー変更に伴うスパニングツリー再構築ための使うBPDUをTCN(Topology Change Notification、トポロジー変更通知)BPDUといいます。ここでは、設定BPDUを使ってスパニングツリーを構築する場合についてだけ説明します。

設定BPDUのデータ部に含まれる主な内容はルートブリッジID、送信元ブリッジID、ルートパスコスト、ポート識別子(送信元ブリッジのポート識別子)などです。

ルートブリッジIDは、その時点でルートであると想定されている(つまり、その設定BPDUを送り出すブリッジがその時点でルートであると想定している)ブリッジのIDです。送信元ブリッジIDは、その設定BPDUを送信したブリッジのIDです。コスト(ルートパスコスト)は、その設定BPDUを送信しているブリッジからルートへの最短パス(その時点で送信ブリッジが把握している最短パス)のコストです。スパニングツリーアルゴリズムでは、リンクに対するコストが定義されますので、送信ブリッジからルートへの最短パスを通った場合の、リンクコストの合計がルートパスコストになります。

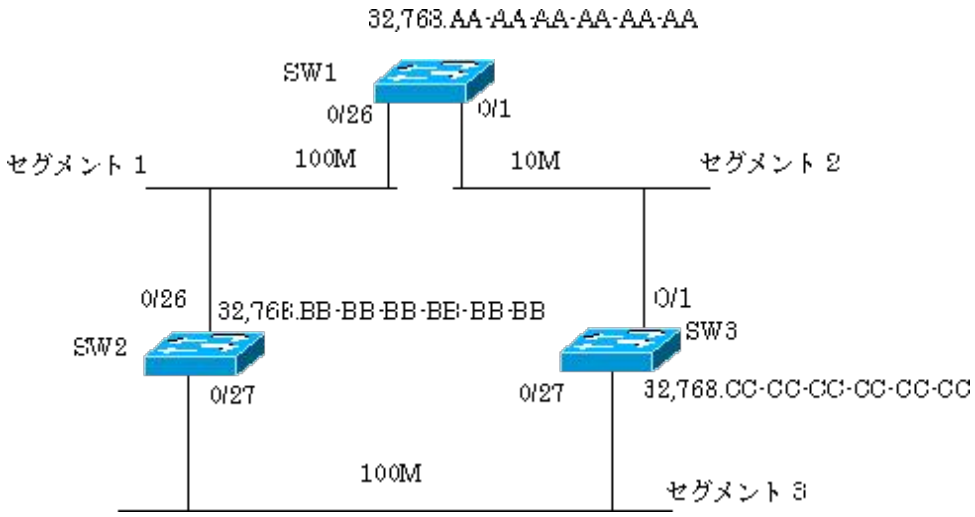
リンクのコスト値は次の計算式で求めます。

コスト値＝1000÷リンクの帯域幅(Mbps)

したがって、標準イーサネット(10Mbps)のリンクコスト値は100、ファーストイーサネット(100Mbps)のリンクコスト値は19ということになります。

ブリッジIDはBIDと表記することにします。BIDは、MACアドレスの前に(ブリッジ)プライオリティと呼ばれる値をつけたものです。この場合のMACアドレスはブリッジを代表するIDとして使いますので、インターフェースのIDとは意味合いが違います。BIDで使うMACアドレスはブリッジのインターフェー

スに割り当てられた MAC アドレスのうちの一番小さな MAC アドレスを使うのが慣例になっています。



上のネットワークを使って説明します。各スイッチの BID は同じプライオリティ 32,768 の次に MAC アドレスが続いています。

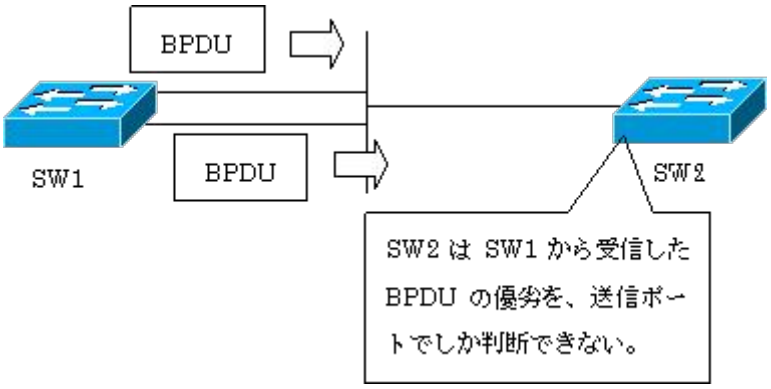
スパニングツリーの設定時に行うべきことは概ね、ルートブリッジの選出、各ブリッジからルートブリッジへの最短パスの計算、各 LAN(セグメント)上で 1 つの代表ブリッジの選出、ブリッジ毎にルートポート(そのブリッジのポートの中でルートブリッジにもっとも近いポート)の選出、スパニングツリーに含まれるポートの選出の５つです。

自分がルートであると認識しているブリッジは BPDU を発信します。自分は、ルートでないと認識しているブリッジは自分から BPDU を発信することはありません。非ルートブリッジは受信した BPDU を転送するだけです。ブリッジは受信した BPDU のうち、ポート毎に最適なものを保存します。最適かどうかの判断は、次のシーケンスで決します。

ルート BID
ルートパスコスト
送信元 BID
送信ポート識別子

BID、ルートパスコスト、送信ポート識別子は、より小さなものが優先されます。送信ポート識別子で決着するのは、ルート BID、ルートパスコスト、送信元 BID がそれぞれ同じ場合です。最終的に、送信ポート識別子を使って判断しなくてはならない場合は、いくつか考えられますが、そのうちの 1 例を次に示します。ただし、これ以降の説明では、この例のような場合は無視し、ルート

BID、ルートパスコスト、送信元 BID までで必ず優劣が決するものと仮定します。



ブリッジは受信した BPDU のうちポートごとに最適な BPDU を保存します。ルートパスコストは、受信した BPDU のルートパスコストにポートのリンクコストを加算したものとします。パスコストは、ブリッジからルートまで戻る際のコストを算出したいので、当該ブリッジから LAN へのコストを加算する必要があります。ブリッジはポート毎に最適 BPDU を保存し、転送する際には、その中からさらに最適 BPDU を選んで、そのポート以外のポートから転送します。その際に送信元 BID は自分の BID に書き換えます。ただし、その時点で、自分をルートと信じているブリッジの場合は、自分が発信すべきものがポートに保存されたものよりもさらに優れているはずですので、それを発信します。

①ルートブリッジの選出

初めにすべてのブリッジが「われこそはルートブリッジである」ということで、自分から BPDU を発信します。この時点では、ルート BID と送信元 BID がともに自分の BID になっています。そして、自分の BPDU よりも優れた BPDU を受信した時点で、ルートブリッジになることをあきらめて BPDU の転送に専念します。

SW1 が発信した BPDU は、SW2 のポート 0/26 で受信され、次のように保存されます。

受信ポート 0/26
ルート BID：32768.AA-AA-AA-AA-AA-AA
ルートパスコスト：19
送信元 BID：32768.AA-AA-AA-AA-AA-AA

SW2 は SW1 からの BPDU を受信する前ならば、自分から BPDU を発信しますが、受信した後は、自分よりも SW1 がルートブリッジに適任であると判断しますので、自分から BPDU を発信することをやめます。SW2 はポート 0/27 で、SW3 経由の BPDU を受信する前は、SW2 の受信 BPDU は SW1 からのものだけなので、こ

れが最適 BPDU となります。SW2 はこの BPDU を受信ポートである 0/26 以外のポートから転送します。

SW3 は 0/1 で受信した BPDU と 0/27 で受信した BPDU を比較して(さらに自分がルートならば発信するはずの BPDU とも比較しますが、説明を省略します)、0/1 で受信した BPDU を最適 BPDU と判断して、0/1 以外のポートから転送します。

最終的に SW1 がルートブリッジに選出されます。SW2 と SW3 の保存する最適 BPDU は次の通りです。

SW2

受信ポート	BPDU
0/26	ルート BID：32768.AA-AA-AA-AA-AA-AA パスコスト：19 送信元 BID：32768.AA-AA-AA-AA-AA-AA
0/27	ルート BID：32768.AA-AA-AA-AA-AA-AA パスコスト：119 送信元 BID：32768.CC-CC-CC-CC-CC-CC

SW3

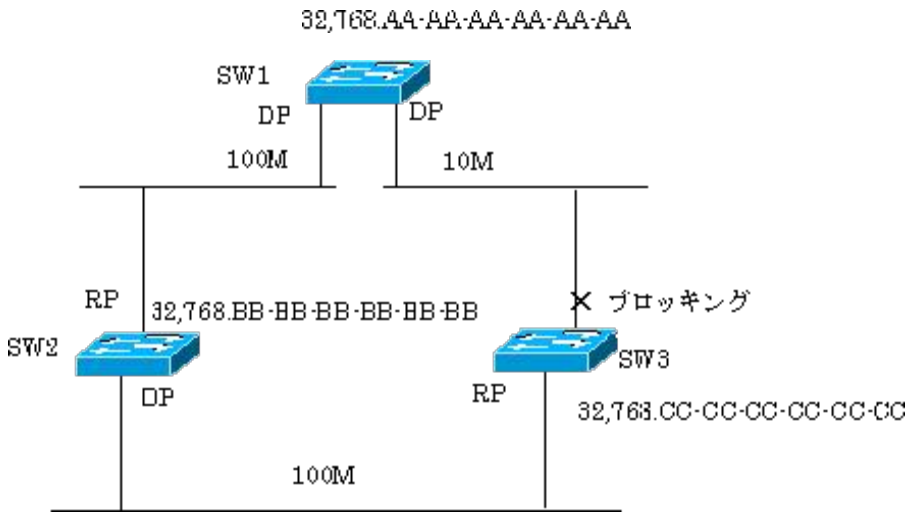
受信ポート	BPDU
0/1	ルート BID：32768.AA-AA-AA-AA-AA-AA パスコスト：100 送信元 BID：32768.AA-AA-AA-AA-AA-AA
0/27	ルート BID：32768.AA-AA-AA-AA-AA-AA パスコスト：38 送信元 BID：32768.BB-BB-BB-BB-BB-BB

② ルートブリッジ以外のブリッジにつき、１つのルートポートを選出
ルートブリッジ以外のデバイスにつき、1 つのルートポートを選出します。ルートポートはそのデバイスのポートの中で、ルートブリッジに一番近いポートです。SW2 では、0/26 が、SW3 では 0/27 がルートポートに選出されます。

③ セグメント(LAN)毎に１つの代表ポートを選出
ルートポート選出の手順と同時進行的に、代表(Designated、あるいは指名)ポート選出の手順が進められます。代表ポートが属するブリッジを代表ブリッジといいます。代表ブリッジに選出されるのはルートブリッジにより近いブリッジです。
ルートブリッジのすべてのポートは代表ポートになります。したがって、セグメント１、セグメント２ではルートブリッジのポートが代表ポートです。セグメント３では、SW2 のアドバタイズするルートパスコストは 19 です。次に、

SW3 のアドバタイズするルートパスコストですが、SW2 経由の BPDU(SW2 が 0/26 で受信したもの)の転送を受ける前は、0/1 で受信したものを最適として判断して、これをセグメント 3 に転送します。しかし、SW2 経由の BPDU を 0/27 で受信した場合は、こちらをより優れたものと判断しますので、これを 0/27 以外のポートから転送します。したがって、この時点で、SW3 はセグメント 3 には BPDU を転送しなくなります。いずれにしてもセグメント 3 では、SW2 が代表ブリッジということになり、SW2 の 0/27 がセグメント 3 の代表ポートとなります。

代表ポートにも、ルートポートにもならないポートは「Non-Designated Port」と呼ばれ、ブロッキング(Blocking)状態になり、ループフリーなパスが完成します。



ブロッキング状態のポートは使えなくなるわけではありませんので注意してください。ブロック状態のポートでは、フレームを受信することも、フレームを送信することも可能ですが、フレームの転送はできません。つまり、ブロック状態のポートで受信したフレームは他のポートから転送されることはありませんし、他のブロック状態でないポートで受信したフレームがブロック状態のポートから転送されることはありません。

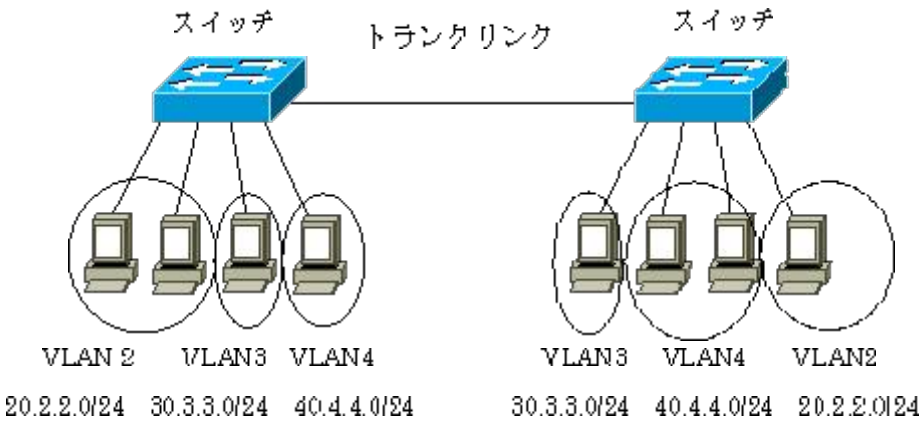
代表ポートはルートブリッジから外に向かうトラフィックの送信に使用されます(したがって、データの転送、コンフィギュレーション BPDU の送信)。これに対して、ルートポートはルートブリッジに向かう方向のトラフィックの送信に使用されます(したがって、TCN BPDU の送信)。

■　VLAN

LAN については、「数キロメートルの範囲内で使うように設計されたネットワークで、従来は公道を横切らないネットワークを想定しています」というように説明しました。これでは、何を言っているのやら分からないと思った人もいるかも知れません。もっと、具体的に言えば、パソコン、スイッチ、HUB、ケーブルなどの集合体で、データリンク層での通信をサポートするものということになるでしょうか。

VLAN は単純な物理的な形状とは関係なく、抽象的に LAN と同じ動作を実現する技術です。今までの考え方では、1 つのスイッチング HUB あるいは HUB につなげられた機器が 1 つの LAN を構成します。しかし、VLAN を使うと物理的な形状とは関係なくなりますので、1 つのスイッチにつながった機器が異なる VLAN に属することもありますし、別のスイッチにつながった機器同士が同じ VLAN に属することもあります。このことを単純に図解すると次のようになります。VLAN に対応したスイッチを接続する場合は、VLAN ごとに別の物理リンクを使う仕様(IEEE802.1D)と、異なる VLAN で 1 つの共通の物理リンク（トランクリンクといいます）を使用する仕様（IEEE802.1Q、ISL）があります。

IEEE802.1D はスパニングツリーについて規定している仕様ですが、VLAN についても規定しています。VLAN ごとに別の物理リンクが必要ということになると、スイッチが離れている場合は使い勝手が良くありません(ケーブルとポートが余分に必要です)。そこで、後になって、異なる VLAN で共通の物理リンクを使用するためのプロトコルが開発されました。IEEE802.1Q は標準のプロトコルで、ISL はシスコ社独自のプロトコルです



トランクリンクを使用するプロトコルでは、異なるVLANのフレームが１つの共通の物理リンク上を流れますので、そのVLNANに属するフレームかを識別するための識別子が必要になります。IEEE802.1Qの場合は、フレームのアドレスフィールドの次にVLAN用のフィールドを挿入しています。これに対して、Cisco社のISLは、ISLフレームでイーサネットフレームをカプセル化する方式を採用しています。

同じVLANに属するホストは、ネットワークで直接接続されたホスト同士ということになりますので、同じネットワークアドレスを持ち、ホスト部だけが異なります。異なるVLANに属するホストは、たとえ同じスイッチングHUBにつながっているとはいえ異なるネットワークアドレスを持ちますので、直接通信することはできません。異なるVLANに属するホスト同士が通信を行うためにはルータを介する必要があります。ただし、レイア3スイッチを使えば、ルータを介在させずにVLAN間の通信が可能になります。レイア3スイッチはネットワーク層(レイア 3)の機能を持ったスイッチです。レイア 3 スイッチには、大きく分けるとルータにスイッチングの機能を持たせたもの(スイッチングルータ)と、スイッチにルーティングの機能を持たせたもの（ルーティングスイッチ）があります。

■　フレームヘッダ

タイプフィールド型(いわゆるDIXフレーム)
次に示すフレームヘッダは DIX(Dec-Intel-Xerox)フレームといわれる仕様です。長い間業界標準(デファクトスタンダード)として利用されてきましたが、現在では正式なIEEE802.3標準として認められています。

タイプ番号は、ネットワーク層のプロトコルを指定するための識別子です。
FCSはフレームチェックシーケンスです。

従来のIEEE802.3標準は、タイプフィールドのところが長さフィールドになっていましたが、これは非常に使いにくいということで業界では殆どDIXフレームを採用してきました。そこで、IEEE802.3では最近(1997年)仕様を変更し、用途に応じて長さとタイプのいずれでも定義できるようになりました。

■　インターフェース

　インターフェースとは、2つのもの間に立って仲介役を演ずるもの、またはその規約です。身近な例では、家庭用電源のコンセントとプラグなどが家庭用電源のインターフェースです。コンピュータ関係では、「ハードウェアインターフェース」、「ソフトウェアインターフェース」、「ユーザインターフェース」の３つのインターフェースが考えられます。
ハードウェアインターフェースは、複数の装置を接続して通信をする際の規約(電気信号の形式、コネクタの形状)、あるいは接続を仲介する装置を指します。コンピュータとネットワーク間を仲介して、シリアル形式のデータ転送を行うためのインターフェースがシリアルインターフェースで、シリアル接続のケーブルやコネクタ(接続口)等を総称します。
　ソフトウェアインターフェースは、プログラム間でデータをやり取りする際の手順や形式を定めたもの、ユーザインターフェースはコンピュータがユーザに対して情報を表示する方法や、ユーザがコンピュータに対して情報を入力するための形式を定めたものです。