

第2章 LANの仕組み

パソコンLANを構成する要素はノード間をつなぐケーブルと生成されるデータ（電気信号）です。

この章では、そのケーブルの配線方式やデータの送り方、またノード間で正しくデータをやりとりするためのプロトコルと呼ばれる約束事などパソコンLANの仕組みについて見ていきましょう。

1. LANのトポロジー

トポロジーとは、ネットワークを構成する通信路（通信ケーブル）と接続されたノード（コンピュータや各種周辺装置など）による配線形態のことを指します。そのトポロジーには主なものに次の3つが挙げられます。

- (1) スター型
- (2) バス型
- (3) リング／ループ型

では、それぞれの特徴を見ていきます。

(1) スター型

ネットワーク上の全てのノードが一つの制御装置を中心として放射線状に配置された配線形態です。星の形に似ていることから「スター型」と呼ばれています。一番身近な例としては交換機による電話の接続があります。この場合、全ての電話機が交換機に接続され、他の電話機との通話はこの交換機を通してのみ成り立ちます。この交換機と電話機の間を中央制御装置とノードに置き換えたシステム形態がスター型LANなのです。この形態は構造がシンプルなためメーカー側からすると対応するソフトウェア開発が容易であるという長所を持つ反面、接続されるノード数が増加した場合、中心に配置される中央制御装置（ホストコンピュータ）の性能がシステム全体のパフォーマンスに非常に大きく影響する短所も持っています。そこで現在では、中央制御装置の代わりにハブ（HUB）と呼ばれる集線装置を配置し、単に複数の通信路を確保する形態に移行しました。

この形態では、トラブルが発生した場合でもノードごとに問題のある箇所を切り分けることが可能なので、システム全体を停止させずにケーブルの差し替えや接続箇所の変更などで比較的簡単に修復する事ができます。

スター型LANの代表的なものには10BASE-Tがあります。これは、ケーブルにツイストペアケーブルを使用し、その両端はモジュージャックを使用しているため、ハブとコンピュータを接続するには、ジャックを差し込むだけで済むため誰でも大変手軽に接続でき便利です。

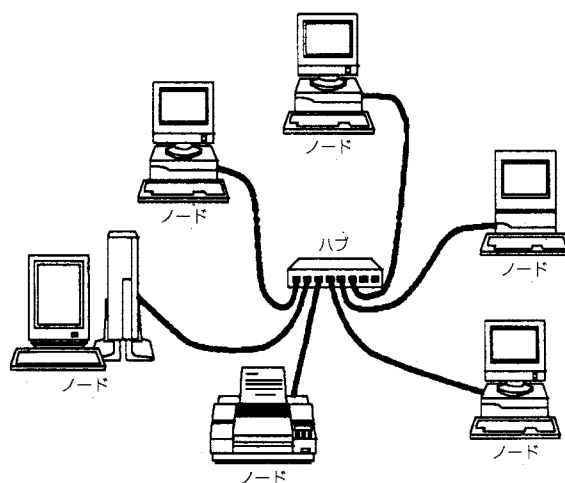


図2-1 スター型LAN

※ハブ (HUB)

ハブは1本の幹線通信路から複数の通信路を引き出すために必要な集線装置です。ハブは入力された電気信号の増幅・再生を行うリピータの機能も合わせ持っています。この働きにより複数のノードに安定したデータ分配が行えるようになっています。

ハブは最大4台までのカスケード接続も可能です。また1台につき約4~12台分のノード端子があるので、かなり数多くのノード接続が可能となります。

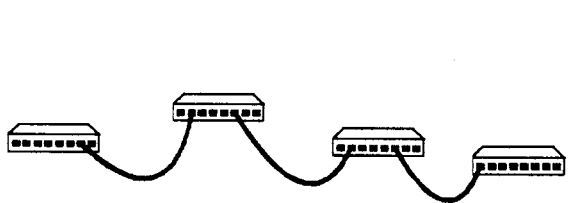


図2-2 カスケード接続

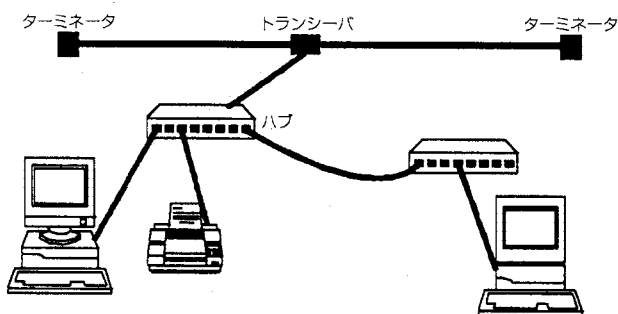


図2-3 ハブを使用した配線

(2) バス型

バス型とは幹線となる1本のトランスケーブル（通信路）を中心として、そこから適当な間隔をあけてブランチケーブルを伸ばし複数のノードを配置することによって構成される直線的な形状の配線形態です。

この形態では、通信路上をデータが両方向に向かって流れていきますがケーブルの終端まで届いた際のデータの反射によるデータ破壊を防ぐために必ずトランスケーブルの両端にはターミネータと呼ばれる終端抵抗を取り付けます。

この形態は、構造がシンプルであるために導入しやすく、建物の天井や床下などの隠れた位置によく配線されています。また、LANの主流として最も広く普及している形態です。代表的な例としてはイーサネットLANがあります。転送速度は10Mbpsとなっており高速な通信が可能です。

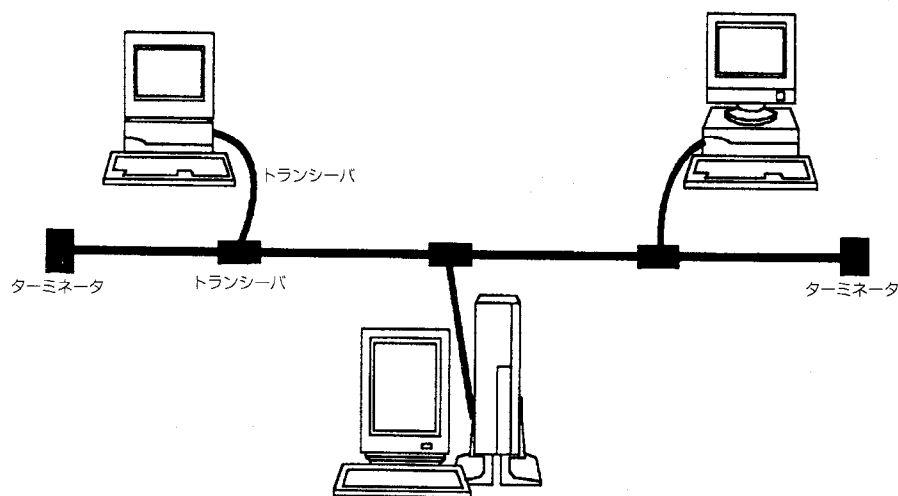


図2-4 バス型LAN

※トランシーバ

イーサネット規格の一つである10BASE-5の配線用機器です。幹線ケーブルである10BASE-5のThickケーブル（イエローケーブル）とパソコンを接続する際に幹線ケーブル側にケーブルに突き刺すようにして取り付けます。

(3) リング／ループ型

リング型は、ケーブルの両端を結び、「輪（リング）」のような形状に構成し、そこに各ノードを配置する配線形態です。

リング型の幹線内ではデータは一方方向に流しますの
で、FDDI（Fiber Distributed Data Interface）のよ
うな高速伝送媒体を使用することが可能です。したが
って、高速で長距離の大規模LANに適しています。と
ころが導入費用が高価なため、機能自体は高い評価
を受けていますが、実際はまだバス型ほど広い普及は
していません。しかし、近い将来はバス型と並んで広
く国内でも普及することは間違いのないでしょう。

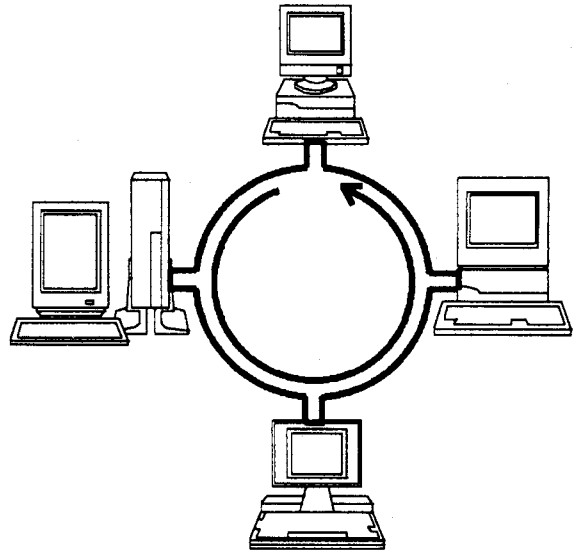


図2-5 リング／ループ型LAN

2. LANのデータ転送方式

パソコンLANのアクセス方式には次のものがあります。

- (1) CSMA/CD方式 バス型
- (2) トークンパッシング方式 バス型・リング型

それぞれの通信方式の特徴を見ていきましょう。

(1) CSMA/CD方式

CSMA/CD（Carrier Sense Multiple Access with Collision Detection：搬送波感知多重アクセス／衝突検出）方式は、名前が示すように、ケーブル上のキャリアの存在を随時検査して（データ転送が行われているかどうか（信号の有無、電圧の変化）を調べること）、キャリアが存在していなければデータを転送する仕組みです。しかし、たまたま複数のノードから同時にデータが転送されることが考えられ、その場合データの衝突が起こりデータを正しく転送することができなくなります。そのために、衝突検出を行う機能も持ち合わせています

それでは細かな手順について見ていきましょう。

(a) 送信手順

- ① 現在のケーブルの利用状況（キャリアが存在するかどうか）を調べる。
- ② キャリアが存在する（他のノードがデータを転送中である）場合は、そのままある一定時間待って再度キャリアを調べる。もし、キャリアが存在しなければ、準備しておいたパケット（データに送信元・送信先のアドレスなどを付加してひとまとめたもの）を送出する。
- ③ 他のノードから同時に送信されたパケットと自分の送信したパケットとの衝突がないかどうか通信路を監視する。もし、衝突を検出した場合はネットワーク上の全てのノードに衝突が生じたことを通知し、自分が送信したパケットを無効として一定時間待機した後再度②の手順に戻る。
- ④ ③で衝突が無ければ送信終了となる。

(b) 受信手順

- ① ケーブル上のキャリアを常時監視する。
- ② 自分宛のアドレスを持つパケットが流れてきた場合にそれを取り込む（関係ないパケットはそのまま無視する）。

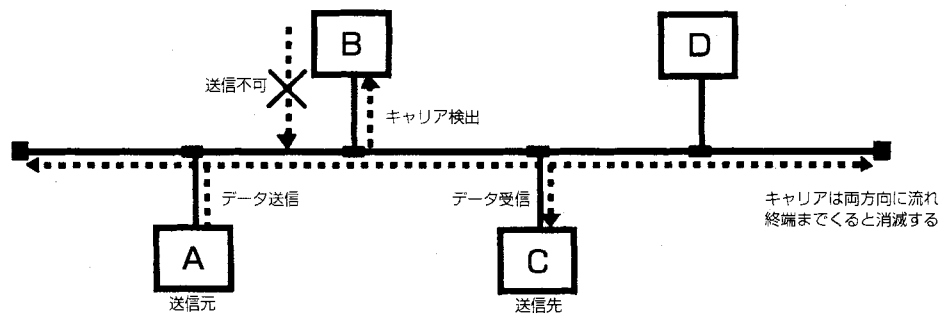


図2-6 CSMA/CD方式

この方式では、通信路を監視するだけという具合に、制御システムがシンプルのためネットワーク構築が比較的容易で様々なLANで広く活用されています。

通信速度は、バス型のイーサネットLANに代表されるような10Mbpsの高速な通信が可能です。

ネットワークパフォーマンスの面で見るとトラフィック（ネットワーク上のデータの通信量）が少ない小規模なLANであれば非常に優れた通信効率を発揮できますが、その反面、ノード台数が増加すると急激にトラフィックが増加し、極端に通信効率が低下してしまう欠点もあります。

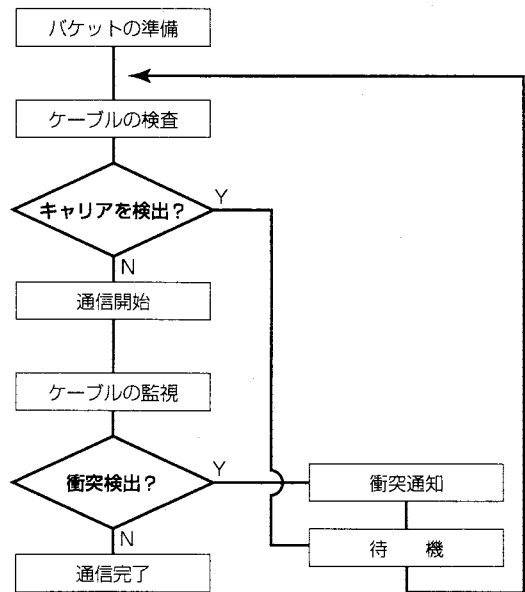


図2-7 CSMA/CD方式での送信フロー

※パケット (Packet)

LAN上では複数のノードが接続されており、1台があまりに長いデータを送信すると他のノードが送信できなくなってしまいます。そこで、LAN上でデータを送信する場合はデータの大きさを約1~4kバイト程度に分割して送ります。この分割されたデータのことをパケット（「小包」という意味）と呼びます。

パケットの中には伝送先・伝送元のアドレスや分割の順序などの必要なデータが含まれています。受信側では受け取ったパケットを組み立て直して復元します。

(2) トークンパッシング方式

この方式は論理的（トークンバス方式）、あるいは物理的（トークンリング方式）に各ノードを閉鎖的なリング状に配置し、フリートークンと呼ばれるある特定のビットパターンのパケットを高速で巡回（パッシング）させて送信権を取得させる方式です。

リング内ではトークンは1つだけしか存在しませんから、CSMA/CD方式のように複数のノードが同時にパケットを送信して衝突してしまうことはありません。つまり、パケットの衝突によるデータ破壊が発生しないように工夫された方式なのです。

では、詳しい手順について見ていきましょう。

(a) 送信手順

- ① データを送信したいノードはケーブル上に流れてきたフリートークンを取得する。これによって、そのノードは送信権を取得したことになる。なお、データの送信を行わない場合は、フリートークンを無視してそのまま流す。
- ② 取得したフリートークンに、送信したいデータや送信先・送信元アドレスを付加しビジートークンに切り替えてケーブル内に戻し送信する。
- ③ 送信先のノードから受信情報を付加したトークンが流れてくるので、それを取得する。
- ④ 再びフリートークンを通信路上に戻し、他のノードに送信権を渡す。

(b) 受信手順

- ① ケーブルを監視し、自分宛のアドレスを持つトークンが流れてきたら取り込む。
- ② 送信元のノードに受信情報を付加したトークンを返す。

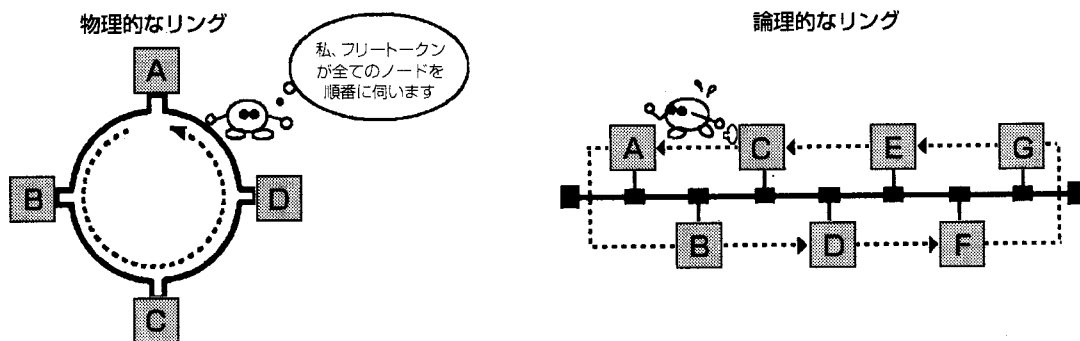


図2-8 トークンパッシング方式

このトークンパッシング方式の場合、パケットの衝突が発生しないためノード数を増加しても全体のパフォーマンスが極端に低下することはありません。しかし、制御システムはトークンの伝送制御（トークンを正しい順序でノード間に渡す、トークンの紛失や複数発生など）やリング動作の管理、特に論理的リングを形成するトークンバス方式の場合は更に論理的リングを維持するための幾つかの制御が必要になるため複雑になります。したがって、導入する場合のコストを考えると比較的割高になります。

※CSMA/CD方式とトークンリング方式の比較

どちらの方式を利用するのが果たして良いのでしょうか？

フリートークンが回ってくるまでデータ送信が出来ないトークンパッシング方式の方が一見非効率的に感じるかもしれませんが、実際にはそうではないようです。もう一度両者の仕組みについて考えてみます。

CSMA/CD方式の場合、データを送信しようとした時に通信路上に既にキャリアが存在している場合は、ある時間（ノードごとに乱数を利用して待ち時間を決定しています）だけそのまま待機します。その後、再度送信を試みますがその際にもまたキャリアを検出すると再び待機することになります。問題になるのは、この通信路が空になるまで繰り返し待機しなければならないという点です。待機している間に通信路が空き、その間に他のノードがデータを送信してしまった場合、待機時間が終わって再度送信を試みますが、通信路は空ではありませんからまた待機することになります。データを送信するノードは、複数存在するわけですから、極端なケースですが、最悪の場合タイミングが悪ければいつまでも待機してデータを送信することが出来ないというケースも考えられます。（つまり「早い者勝ち!!」という方式なのです）

この現象はノード数が増加し、その各ノードが頻繁にデータを送信しようとする程、発生する可能性が高くなると考えられ、結局ノード数が増える程ネットワークのパフォーマンスが低下することになります。

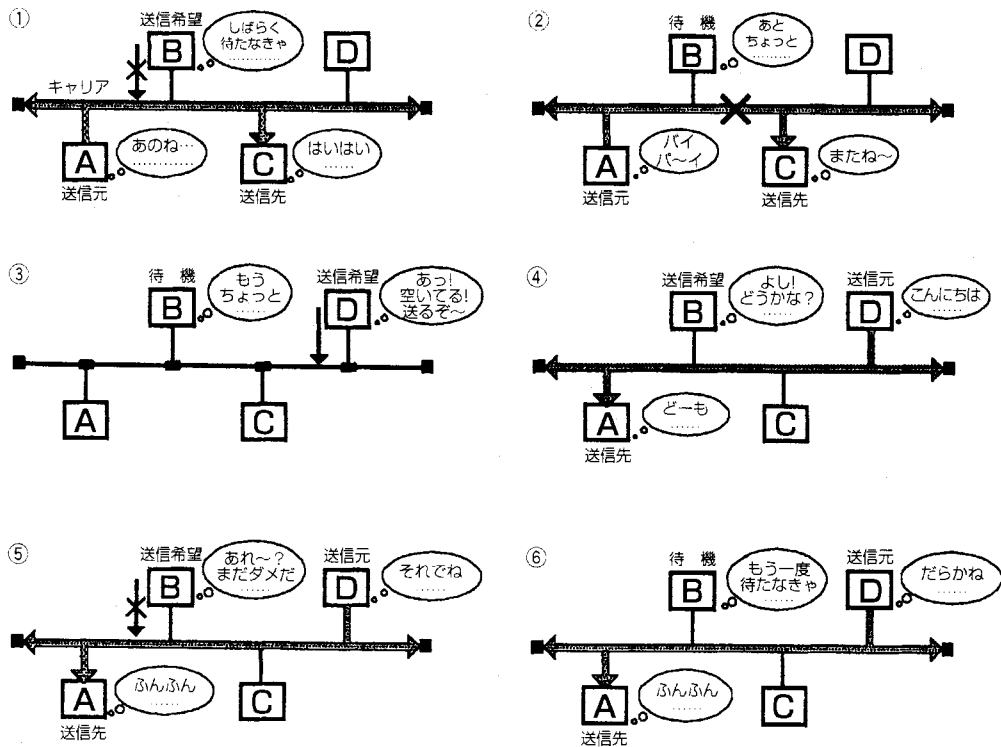


図2-9 CSMA/CD方式

トークンリング方式の場合はフリートークンが回ってくるまでは確かに待機しなければなりません、このフリートークンは、たとえ全てのノードが送信しようと待機していても必ず順番に全てのノードに回ってくることになります。どういうことかという、あるノードがフリートークンを取得し、データを付加してビジートークンとして送信します。送信先のノードはアドレスを見て、自分宛のトークンと判断したら受け取ります。そして受信情報を付加して送信元ノードにトークンを返します。送信元ノードが再びフリートークンを通信路に戻す時、他のノード全てがフリートークンをひたすら待っていたとすると、このフリートークンは今データを送信したノードのすぐ下流のノードに必ず渡ることになります。この繰り返しのことを考えると理解できると思いますが、フリートークンを取得するために通信路を監視していればそれはいずれ必ず回ってくるようになります。したがって、リングに接続するノード数を増加させても極端にパフォーマンスが低下する心配はありません。つまり、性能的には安定していると言えるでしょう。

一方、導入コストの面で比較すると先にも述べた通り、CSMA/CD方式は制御構造が比較的簡単なため対応するハードウェアやソフトウェアの開発も比較的容易です。その結果、安価で優れた製品が数多く市場に出回っているためより低コストで簡単にLANが導入できます。逆にトークンパッシング方式では、障害対策や管理などに関する制御などの複雑な機能をネットワークを制御するNIC（ネットワークインターフェースカード）中のハードウェアに持たせます。したがってその導入にかかる費用は高くなるようです。

結局、比較するとどちらにも一長一短があるということになり、選択する際は「どちらの方式が良いか」ではなく「目的の業務にあう方式はどちらか」という点を重視して選択することが大事なようです。

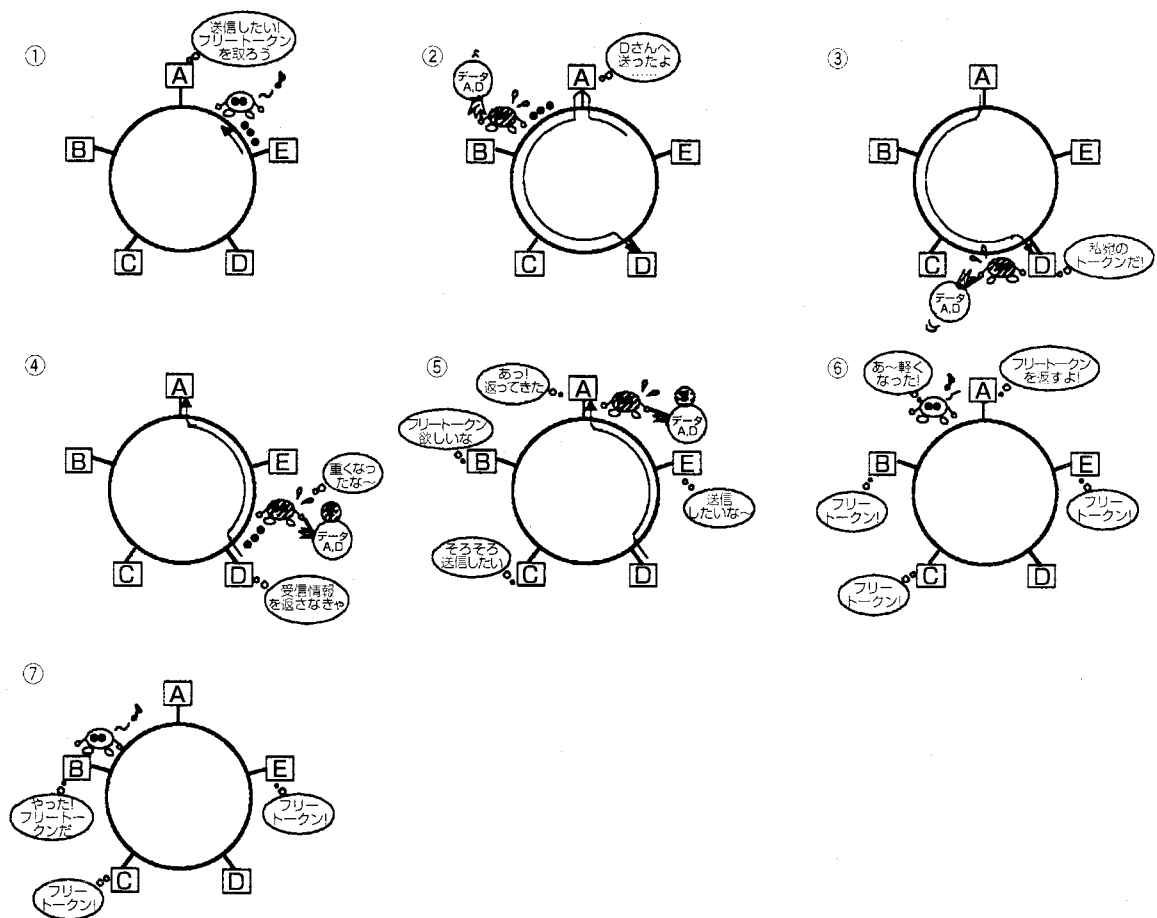


図2-10 トークンパッシング方式

3. 通信プロトコル

ネットワークと言えば必ず「プロトコル」という言葉を耳にするでしょう。あるいは、「プロトコル」と言う言葉そのものではなくネットワーク上で利用される様々なプロトコルの具体的な名称自体を耳にしているのかもしれない。では、そのプロトコルとは一体何なのでしょう？ また、どのような種類のプロトコルが存在するのでしょうか。

(1) プロトコルとは

プロトコル (Protocol) とは、もともと「外交上の儀礼」という意味を持ちますが、ネットワーク用語としては「通信規約」と訳されています。広い意味で捉えると、データの通信路であるケーブルやそこを流れるデータそのものである電気信号にいたるまでの通信に関する規約一般のことをプロトコルと呼んでいます。ただし、ここまで広い意味で捉えずに、通信に関する規約の中でも主に通信手順に関わる部分を示すことが多いようです。

プロトコルには数多くの種類があります。代表的なものにUNIXで標準で採用されている「TCP/IP」やNetWareで使われている「IPX/SPX」、マッキントッシュ用の「AppleTalk」などがあります。送信側と受信側でこのプロトコルがうまくかみ合わなければ通信は成り立ちません。そこで現在、OSに依存しない異機種間通信プロトコルの標準化がISO（国際標準化機構）を中心に進められています。

では、プロトコルの概念について具体的に説明します。

コンピュータは本来、人間の仕事を代行することがその役割です（複雑な処理などでも人間とは比較にならないほど正確に、高速に処理しますが……）。コンピュータに人間の機能の一部を代行させているわけですから、コンピュータの構造や様々な取り決めは、結局人間そのもの、あるいは人間社会の仕組みがモデルとなります。そこで、人間同士が日常生活の中で行うコミュニケーションについて考えてみます。

2人の人間が手紙でコミュニケーションをとる場合、差出人は自分の意志を言葉におきかえて紙に文字として書きます（要するに手紙を書くわけですね！）。これを郵便物としてポストに投函します。すると宛名書きの郵便番号や住所を手がかりに受取人の所まで配達され、それを相手を読むことによって差出人の意志が伝わります。これを図で示すと次のようになります。

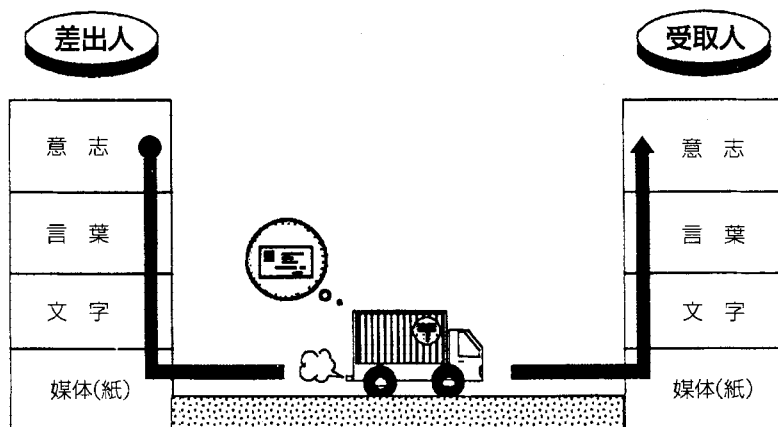


図2-11

差出人と受取人の両者共に4つのプロセス（層）があります（意志、言葉、文字、紙）。この各々のプロセス間にはそれぞれの共通の取り決め（例えば言葉であれば互いに理解できる言語を用いる。文字であれば互いに理解できる文字を使うなど）があるからこそ最終的な意志の疎通が成立します。この取り決めこそがプロトコルなのです。

普段何気なく行っているコミュニケーションも細かく考えてみると、各プロセスで実に様々なプロトコルがあるのです。この共通のプロトコルがあるからこそコミュニケーションが成り立ちます。このように人と人とのコミュニケーションをモデルとして、コンピュータ間のコミュニケーション、つまり通信についてもその流れを幾つかのプロセスに分け、各プロセスそれぞれについてのプロトコルを定義しています。

では、通信の世界では実際にどのようなプロトコルになっているのでしょうか。

(2) OSI参照モデル

先にも述べたように、プロトコルには実に数多くの種類があります。しかし、LANが普及してきた現在、異なるメーカー間のコンピュータもLANで結ぶことが必要となるため、あまりにも多くの種類のプロトコルが氾濫するのはユーザ側にも開発者側にも大変非効率的なことになります。この状況を解決するために、国際的な機関であるISO（International Organization for Standard：国際標準化機構）とCCITT（Consultative Committee for International Telephone and Telegraph：国際電信電話諮問委員会）の連携によってOSI（Open System Interconnection：開放型システム間相互接続）という統一的なネットワークアーキテクチャの国際規格を制定しました。この統一規格を採用することにより、異機種間でも問題なく通信を実現させることを目的としています。また、将来新しいネットワークシステムが開発されたとしても、OSIに準拠していれば現状のシステム間と問題なく通信ができるはずです。

このネットワークアーキテクチャでは、やはり階層（レイヤ）構造が用いられます。通信で必要となる機能を幾つかの機能別に分割し、各層ごとに独立したプロトコルを採用しています。この概念がOSI参照モデルです。では、OSI参照モデルを見てみましょう。

OSI参照モデルは7つの層から成り立ちます。これによりネットワークアーキテクチャをモデル化しています。このモデルはプロトコルを理解していく際の基礎となる重要な概念ですので各層の概略についてこれから説明していきます。

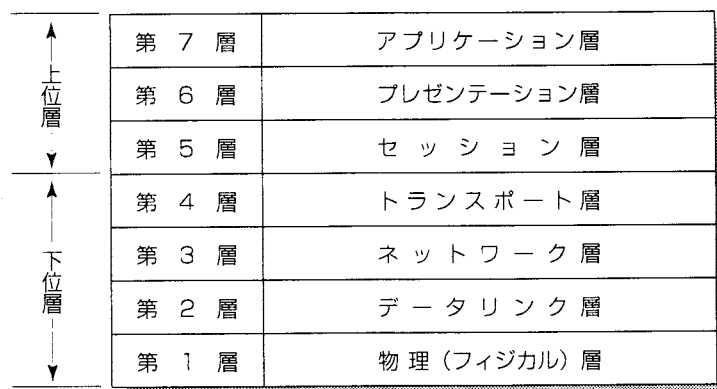


図2-12 OSI参照モデル

7層のうち第5～7層を上位層、第1～4層を下位層と呼んでいます。それぞれの働きを簡単に説明すると以下のようになります。

《上位層》

主に下位層によって提供されるデータ転送機能を利用して、パソコンやワークステーション上のいろいろなアプリケーション業務に対応した通信サービスを実現する機能が中心のプロトコルです。

《下位層》

パソコンやワークステーションなどのパケットデータを信頼性を高く効率よく通信先に転送するための通信機能を提供するプロトコルです。

続いて各層それぞれについて簡単にまとめた表を以下に示します。

表 2-1 OSI参照モデルと各層の機能

	OSI参照モデル	各階層(レイヤ)の機能
7層	アプリケーション層	実際に通信を利用する各種アプリケーションレベルの層。ファイル転送や電子メール、データベースなどの具体的なアプリケーションサービスに関してのコンピュータ間でのデータ形式や手順などについて取り決めが行われる。
6層	プレゼンテーション層	転送するデータが通信するコンピュータ双方で理解できるようにデータ形式を定義したり、機種に依存しているデータの種類やその形式について調整(データの符号化)をする。
5層	セッション層	コンピュータ間で通信を効率良く行うために同期を取ったり、通信方式を明確にしたりして通信の制御を行う。
4層	トランスポート層	コンピュータ間での通信における品質(信頼性)の高いデータ転送を実現させる。転送誤りの検出や回復機能などを持たせ、データを確実に送信する役割がある。
3層	ネットワーク層	通信する上で転送するデータの通信経路を選択し(ルーティング)、通信相手を識別するためのアドレッシング、フロー制御などを行って相手にデータを届けるための中継制御を行う。
2層	データリンク層	物理的な伝送路上での正確な転送を提供する。伝送誤りの検出や再送などの処理を行い、データを確実に伝送路に流す。
1層	物理(フィジカル)層	転送するデータをどのような電気信号に変換するか、あるいは通信媒体の特性やそれとのデータのやりとりなど通信に必要な機器に関する制御を行う。

コンピュータ間でやり取りするデータは、この7つの層(レイヤ)を順に通り返けることにより、データに対して各層それぞれでの処理が行われて送信先のアプリケーションまでデータが正確に送り届けられるようになっていきます。

しかし、現実にはこのOSI参照モデルに厳密に沿って設計されたネットワークアーキテクチャはあまりないようです。業界標準(デファクト・スタンダード)として広く普及している「TCP/IP」のようなオープン指向のプロトコルが多くシステムで既に採用されていることなどの影響もあるようです。ただし、この概念の目指す目的は非常に優れたものであり、将来のオープンシステム環境に対応するために、各メーカーで積極的に今後取り入れられていくでしょう。

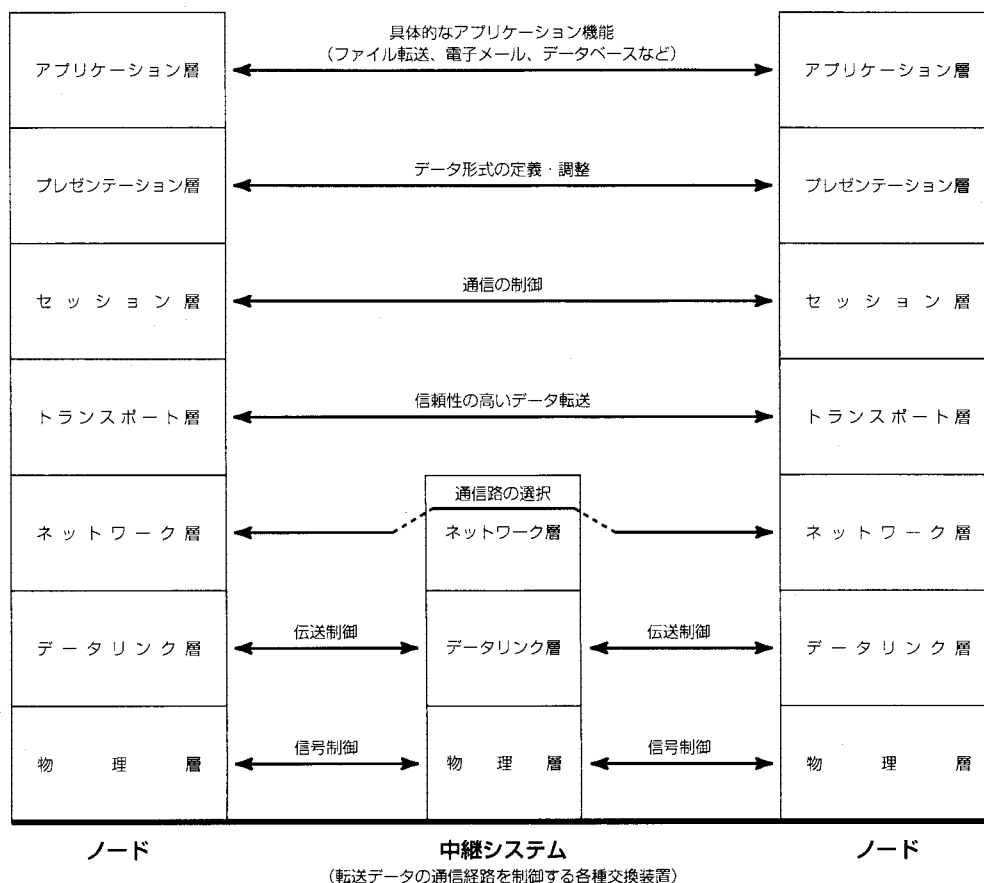


図2-13 各層の機能

(3) 実用プロトコル

OSIの基本概念に基づいてオープンシステム環境が整えられる動きもさることながら、それとは別に、広く普及したことによってそれが事実上の標準となり規格化されたプロトコルもあります。つまり、「業界標準」と呼ばれるものです。業界標準プロトコルとして代表的な「TCP/IP」や、パソコンLANで広く用いられているネットワークOSであるNetWareで使われている「IPX/SPX」プロトコルについて概要を説明しましょう。

(a) TCP/IP プロトコル

TCP/IP (Transmission Control Protocol/Internet Protocol) はもともと米国国防総省の軍事ネットワークであるARPAネットワーク用に開発され、その後、国防総省の下部機関であるDARPA (高等研究プロジェクト局) によって手が加えられ、1983年以降実用化されました。そしてUNIXのバークレー版に、このTCP/IPによるネットワーク機能が標準的に実装されたこと、また、その仕様が公開され機種への依存度も少なく、ファイル転送などの利用度の高い機能が標準で用意されているなどによって、世界的な規模で普及し今では業界標準のプロトコルとしての地位を確立しています。

TCP/IPは、関連するいろいろなプロトコルの集まった総称です。ここで関連するプロトコルも含めてOSI参照モデルと対応させながら紹介しましょう。

OSI参照モデル		TCP/IPプロトコル群				
7	アプリケーション層					NFS
6	プレゼンテーション層		SNMP	FTP	Telnet	RPC
5	セッション層					
4	トランスポート層		TCP		UDP	
3	ネットワーク層		IP		ARP	RARP
2	データリンク層		Ethernet、トークンリング、AppleTalkなど			
1	物理（フィジカル）層					

図2-14

《上位層プロトコル》

○ FTP (File Transfer Protocol)

コンピュータ間でファイル転送を行うためのプロトコル。

○ Telnet (Telecommunications Network)

遠隔地のコンピュータを直接利用するためにコンピュータ上に仮想端末を実現するプロトコル。

○ RPC (Remote Procedure Call)

相手コンピュータ中のプログラムを起動し、その処理結果を取得するプロトコル。

○ NFS (Network File System)

相手のコンピュータのファイルシステムをローカルであるかのようにアクセスするプロトコル。

○ SNMP (Simple Network Management Protocol)

ネットワーク管理用の情報をやり取りするプロトコル。

《下位層プロトコル》

○ TCP (Transmission Control Protocol)

トランスポート層に対応したプロトコルで、データを確実に効率良く通信先とやり取りするためのプロトコル。

○ UDP (User Datagram Protocol)

TCPと同じトランスポート層に対応するプロトコルですが、コネクションレスなので相手の受信確認を取得しません。その分TCPよりも手軽に素早いデータのやり取りが行えるプロトコル。

○ IP (Internet Protocol)

ネットワーク層に対応するプロトコルで、TCP/IPで使用するアドレス (IPアドレス) を付加したり、上位層TCPプロトコルへ目的のデータを受け渡す働きをするプロトコル。

○ ARP (Address Resolution Protocol)

通信するには、IPアドレスと通信機器に付けられた固有の物理アドレス (グローバルアドレス) が必要であるが、相手のIPアドレスからその物理アドレスを知るためのプロトコル。

○ RARP (Reverced Address Resolution Protocol)

自分の物理アドレスから相手のコンピュータの働きによって自分のIPアドレスを知るプロトコル。

(b) IPX/SPXプロトコル

NetWareのプロトコルは幾つかの関連プロトコル群も含めて、一般に「IPX/SPX」プロトコルと呼ばれます。TCP/IPプロトコルと対比させると、SPXがTCP（OSIの第4層のトランスポート層）に、IPXがIP（OSIの第3層 ネットワーク層）にそれぞれ対応しています。

このプロトコルは、NetWareの高いシェアに支えられているため、IPXプロトコル対応のネットワーク関連機器は市場に豊富に出回っています。

IPX/SPXプロトコル群をOSI参照モデルと対応させると次のようになります。

OSI参照モデル		IPX/SPXプロトコル群	
7	アプリケーション層		
6	プレゼンテーション層		
5	セッション層		NCP
4	トランスポート層	SPX	SAP
3	ネットワーク層	IPX	
2	データリンク層	Ethernet、トークンリング、AppleTalk Arcnetなど	
1	物理（フィジカル）層		

図2-15

《上位層プロトコル》

○ NCP (NetWare Core Protocol)

サーバとクライアントの間でネットワークOSのサービスによるデータを交換（送信／受信）するためのNetWareの核（コア）となるプロトコル。ただし、これはNovell社独自のもの。

○ SAP (Service Advertising Protocol)

サーバがそのサービスに関する情報をクライアントに知らせるためのプロトコル。ただし、これはNovell社独自のもの。

《下位層プロトコル》

○ SPX (Sequenced Packet eXchange)

配線、パケットの順序だて、パケットの二重化（複数のノードへのデータ転送時）などを保証するプロトコル。

○ IPX (Internetwork Packet eXchange)

パケットを送るためのアドレッシング、ルーティング、パケット交換などを行うプロトコル。