

情報科学演習 C 資料

文責: 小島 英春*

2018 年 4 月 9 日

1 2 点間ネットワークと放送

我々のもっとも身近にある「通信ネットワーク」
として「電話網」や「テレビ放送網」などが挙げら
れる (図 1)。この 2 つは以下のような特徴を持って
いる。

- 電話網は、特定の電話機間における 1 対 1 の通
信を提供する。このような接続形態を **2 点間リ
ンク (point-to-point link)** と呼ぶ。
- テレビ放送網はキー局から多数のテレビ受像機
への 1 対多、片方向の通信を提供する。このよ
うな接続形態を **放送 (broadcasting)** と呼ぶ。

計算機ネットワークの場合にも、**伝送媒体**に応じ
てどちらかの接続形態が使われる (図 2)。たとえ
ば、2 点間リンクとして以下のような伝送媒体が使
われる。

- 銅線。RS-232C 規格などの信号ケーブルを自
分で敷設することもできるし、既設の電話線な
どを利用することも可能。
- 光ファイバ。高速な 2 点間リンクを張ることが
できる。
- マイクロ波や赤外線など直進性が極めて高い電
磁波。道路をまたいだ敷地間や、異なる建物間
などケーブルを敷設するのが困難な際に有効。

一方、放送型の伝送媒体には以下のようなものが挙
げられる。

- 銅線。ただし、2 点間に張るのではなく、1 本
のケーブルを引き回し、すべての計算機を並列
に接続する (図 3)。同軸ケーブルを利用する旧
式のイーサネットなどがこの方式。
- 直進性のあまり高くない電磁波。直進性の高い
電磁波でも、通信衛星から地上へ送るなら、放
送型媒体として使用できる。無線 LAN など。

計算機ネットワークの場合ソフトウェアによって
通信をかなり自由にコントロールできるので、2 点
間リンク型の伝送媒体をあたかも放送型であるかの
ように、また、放送型伝送媒体を 2 点間リンクであ
るかのように利用できる。

つまり、物理的伝送媒体が 2 点間リンク型であっ
ても、中継によって全計算機に同じデータを『放送』
することができる。図 2(a) を例に取ると、計算機
A が同じデータを C, D 両方に送出し、D が A か
らのデータを B に中継すれば、計算機 A から B,
C, D への『放送』が可能である。

逆に放送型伝送媒体上でも、特定宛先以外の計算
機ではデータを無視することで 2 点間リンクのよう
に通信することができる。図 2(b) を例に取ると、計
算機 A が送出する通信データに計算機 B 宛である
ことを書き込み、計算機 C, D は自分宛でないデー
タを無視すれば、計算機 A から B だけへの通信が
可能である。

2 回線交換とパケット交換

電話網やテレビ放送網と計算機ネットワークの違
いは他にもある。電話やテレビでは、通信開始前に
通信経路を確保し (電話ではダイヤルした時。テレ
ビではテレビ局が周波数の割当てを受けて放送設備

*文責は小島だが、本資料は演習 C を担当された先生方によ
って管理されている資料を基に作成した。

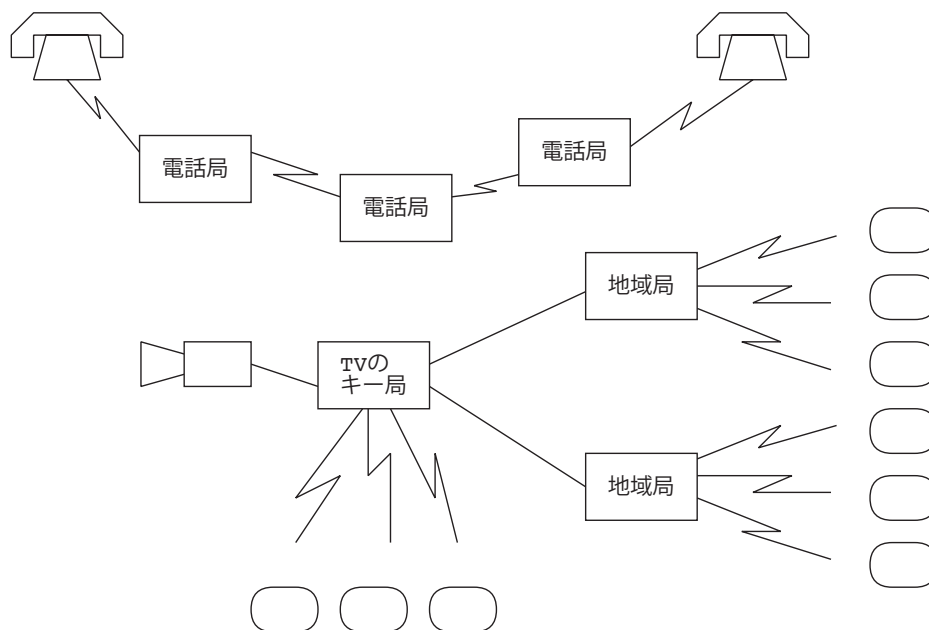


図 1: 電話網 (2 点間リンク) とテレビ (放送)

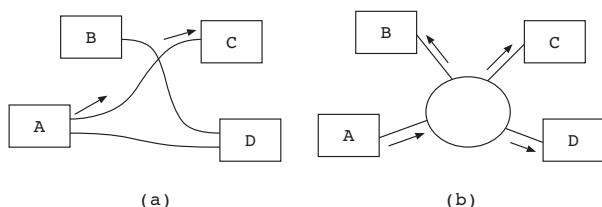


図 2: 2 点間リンク (a) と放送 (バス) 型リンク (b)

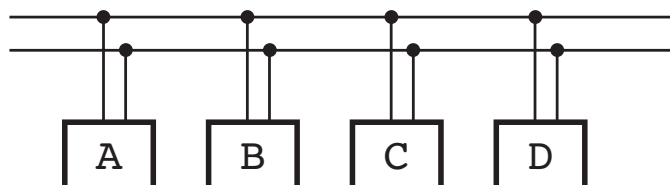


図 3: バス型接続

を準備した時)、通信中その経路はずっと確保したままである。例えば、電話の場合、話中に双方が黙っている時間があっても、回線は確保されたままで、他者がそれを利用することはできない。また、テレビの場合だと夜間の放送時間帯外に他者がその周波数を使うことは普通許されない。そのかわり、電話はいつでも再び話しはじめることができるし、テレビ局は朝になれば混信の心配無しに放送を再開することができる。このような接続形態を回線交換と呼ぶ。

計算機ネットワークでは普通回線交換はあまり採用されない。特定の 2 台の計算機間の通信には、普段はほとんどやりとりするデータがないくせに、「ファイル転送」などの仕事が発生した瞬間に、どっと大量のデータがやりとりされるという性質があるからである。

そこで、計算機ネットワークでは通信したいデータがある範囲の大きさ (数十～数千バイト程度) のパケットと呼ばれるかたまりにまとめ、そのパケットをやりとりすることで通信を行なう。もちろん、1 つのパケットに入り切らない量のデータを送りたい時は複数のパケットに分けて送るわけである。このような接続形態をパケット交換と呼ぶ。電話が回線

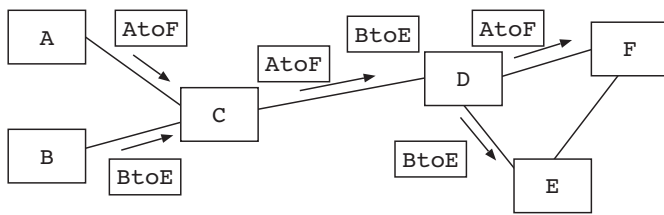


図 4: パケット交換の原理

交換の例だとすれば、パケット交換は葉書や小包に例えることができる。

例えば、図 4 のようなネットワークで、計算機 A から計算機 F へのパケットは、まず計算機 C に転送され、次に計算機 D に転送され、最後に目的地の計算機 F に着くことになる。各中継計算機でパケットはいったん受け取られて格納されることから、パケット交換ネットワークは格納/送出方式 (store and forward) と呼ばれる。この方式は、例えば D→F の転送が失敗した場合でも、D に格納されたパケットをもう一度送るだけですむという利点も持つ。また、例えば、D→F のリンクが壊れてしまっても、A から F へのパケットは D→E→F と迂回転送することで届けることが出来る。このようにパケットの転送経路を適切に制御する管理作業を経路制御と呼ぶ。

さて、計算機 A から計算機 F への通信と同時期に計算機 B から計算機 E への通信も発生したとすると、両者のパケットは C→D の所では同じリンクに相乗りすることになる。B から E への通信がちょっと中断している間は A から F へのパケットを目一杯流しても構わない。つまり、パケット交換では各リンクを回線交換より柔軟に利用することが出来る。そのかわり、A から F へのパケットを次々と流している最中に B から E への通信が再開されたら、この C→D のリンクを両方で公平に使うような管理作業が必要となる。

3 アドレスとポート

パケット交換にせよ回線交換にせよ、通信の際には「送り先」を指定する必要がある。電話の場合は、電話番号がこれに当たるし、葉書や小包なら

住所がこれに当たる。計算機ネットワークではこの情報をアドレスと呼ぶ。各パケットに送り先と送り元の両方のアドレスを書き込む都合上、あまり長い形式では効率が悪いが、ネットワーク中のすべての計算機に互いに相異なるアドレスを割り振れるだけのビット数が必要である。インターネット (the Internet) では現在、32bit のアドレスを用いる IPv4 が主に使われている¹。ただ、0 と 1 を 32 個書き並べるのでは、人間が読み書きし難いので、1~8 桁目、9~16 桁目、17~24 桁目、25~32 桁目をそれぞれ、0~255 の 10 進数で書き、4 つの 10 進数を“.”で繋いだ記法が用いられる。例えば、“10000101 00000001 11110100 01111111”は“133.1.244.127”と表記する²。

普通、1 台の計算機の中には複数のプロセス (A 節参照) が存在しているので、計算機を指定するだけでは通信には不十分である。どのプロセスと通信するかを指定するための情報を一般にポートと呼ぶ。送り先を『これこれのアドレスを持つ計算機の何番ポート』と指定することで、そのポートを使用しているプロセス (普通は高々 1 つ) にデータを送ることが出来る。

パケット送出の際には必ず送り元計算機のアドレスと送出プロセスが使用しているポート番号をパケットに書き込むので、送ったパケットの『返事』を送り返してもらうのは比較的容易である。

4 OSI 参照モデルと TCP/IP

OSI 参照モデルとは、ISO (International Organization of Standardization) によって標準化された通信に関する国際標準 OSI (Open Systems Interconnection) のうち、最もよく知られているものである。OSI 参照モデルは、ネットワークシステムにおいて実現されるべき機能について、階層構造で説明したものである (図 5 参照)。各層は、特定のサービスを上位層に提供し、特定のサービスを階層より受ける。相手のコンピュータ上の同じ層との間に決められた約束ごと (通信する時の方法、アドレ

¹32bit では上記の必要性を満たし難くなってきており、128bit のアドレスを用いる IPv6 の利用も始まっている。

²IPv6 の場合、128bit のアドレスを 16 ビット単位で 16 進数で表記した数値を“:”で区切って表記する。

OSI reference model		TCP/IP model		
7	Application Layer	SMTP, DNS, SNMP, FTP, TELNET, HTTP,... NETBIOS		
6	Presentation Layer			
5	Session Layer			
4	Transport Layer	TCP	UDP	...
3	Network Layer	IP, ICMP		ARP etc.
2	Datalink Layer	Ethernet, Token Ring, X.25, FDDI, ...		
1	Physical Layer			

図 5: OSI 参照モデルと TCP/IP 通信の階層構造

スに関する決まり等が含まれる)をプロトコルと呼び、1つのコンピュータ上で、上下の層の間で決められた約束ごと(データの受け渡し方法等)をインタフェースと呼ぶ。OSI 参照モデルにおいては図5のように階層構造が明確に定義されている。以下に各層の役割を簡単に説明する。

- **第7層: アプリケーション層 (Application Layer)**

様々なサービスをユーザに提供する。ここで言うユーザは、端末を操作する人間であったり、端末上で動作するプログラムであったりする。

- **第6層: プレゼンテーション層 (Presentation Layer)**

データの変換や加工を行い、データをユーザの理解できる形にしたり、通信に適した形にしたりする。通信においては、下位層であるセッション層の提供するセッションを用いて通信する。

- **第5層: セッション層 (Session Layer)**

データを送信するコネクションを確立したり、不要になったらコネクションを開放したりする。また、コネクションを流れるデータの量をコントロールし、効率のよい通信を実現する。

- **第4層: トランスポート層 (Transport Layer)**

セッション層から渡されるデータを、確実に相手に送り届けるまでの責任を持つ層である。転送先が相手のマシンのどのプロセスであるかまでを特定してデータの転送を行う。

- **第3層: ネットワーク層 (Network Layer)**

様々なネットワークを経由して、宛先のコン

ピュータまでデータを転送する最に、最も効率のいい経路を選択したり、ある経路が使えなくなった場合に、別の経路を探してそちらへデータを転送したり、といったスイッチングやルーティングを行います。

- **第2層: データリンク層 (Datalink Layer)**

ネットワーク層から指定されたノードとの間に、物理的な通信のための通信路を確立し、データが流れている間の転送エラー等を検知し、必要であればデータを再送したりする。

- **第1層: 物理層 (Physical Layer)**

データを表わすビット情報を回線に送り出すための電気的な変換や、機械的な作業を行う。

インターネットで用いられている TCP/IP 通信は、OSI 参照モデルにそのままマッチングするものではないが、おおまかには図5に示すような対応になる。以下に各層について簡単に説明する。

- **第7～5層**

図5に表記したような、TCP/IP ネットワーク上で動作する様々なプロトコルを用いたアプリケーションが存在する。TCP/IP ネットワークにおいては7～5層の明確な区別はなく、まとめて「アプリケーション層」と言う場合もある。

- **第4層: トランスポート層**

TCP/IP ネットワークでは、TCP (Transmission Control Protocol) と UDP (User Datagram Protocol) がここにあてはまる。TCP はプロセス間にコネクション型の信頼性の高い通信を提供し、相手からの応答、受信確認等を利用し、ネットワーク内でデータが失われた場合にデータを再送したり、ネットワークの混雑状況に応じて転送速度を調節したりする。UDP はコネクションレス型の通信を提供し、相手からの応答やデータ受信確認を行わないため、通信品質は、下位層のネットワークの品質に影響される。従って、通常は上位のアプリケーションが品質の管理を行う。

- **第3層: ネットワーク層**

インターネットにおいては IP (Internet Protocol) がここに該当する。第4層のプロトコ

ルをクライアントとし、複雑に構成されたネットワーク上におけるデータの伝送サービスを提供する。相手先、及び自分自身の特定には IP アドレス (3 章参照) を利用する。この他、IP アドレスと下位の Enthernet アドレス (ハードウェアアドレス) との対応を調べて通知する ARP (Address Resolution Protocol)/RARP (Reverse Address Resolution Protocol) 等もこの層に該当する。ARP は IP アドレスから MAC アドレスを得る (次項参照) プロトコルで、RARP はその逆を行うプロトコルである。

- 第 2 層以下

原則的には TCP/IP 通信は第 2 層以下がどのような種類のネットワークであっても動作する。演習室もそうだが、最も一般的なのは Ethernet である。以下 Ethernet を例にして説明を行う。ここの説明は第 1 回目の演習に際して参考になるだろう。

TCP/IP 通信を Ethernet のようなリンクを共有するシステム上で運用する場合、端末間の通信は Enthernet カードに割り当てられた MAC (Media Access Control) アドレス (物理アドレスともいう) を用いて通信する必要がある。MAC アドレスはネットワークカードに物理的に割り当てられたもので、例えば exp204 に付けられているイーサネットカードの MAC アドレスは 00:10:5a:67:92:12 である。

しかし、端末同士が通信を行う際には、ユーザ (人、あるいは端末上で動くプロセス、プログラム) は MAC アドレスを知っている必要はなく、IP アドレスを知っていればよい。これは、第 3 層で動く ARP/RARP が、IP アドレスと MAC アドレスの対応を調べて通知してくれるからである。

A マルチタスクとプロセス

一般的な OS にはマルチタスク、すなわち多数のプログラムを並行して実行する機能がある。大抵の場合、同時に実行するプログラムの数は計算機に搭

載されている CPU の数より多くなるが、計算機がある瞬間に実行しているプログラムの数はたかだか CPU の数に限られる³。だから、正確には『多数のプログラムをあたかも並行して実行しているように見せかける機能』と言うべきであろう。

計算機がどのようにしてプログラムを実行しているかをごく簡単に復習する。機械語命令の列 (= プログラム) が計算機のメモリ上に置かれている。CPU はプログラムカウンタが指している番地のメモリから命令を取り出して実行し、プログラムカウンタを進めることを繰り返す。

『多数のプログラムをあたかも並行して実行しているように見せかける機能』は以下のようにして実現されている。まず、複数のプログラム (ここではプログラム A, B, C としよう) をすべてメモリ上に置いておく。そして、CPU に時計を装備する。一定時間後に CPU に信号を送るように時計をセットし、まずプログラム A の実行に取り掛かる。時計から信号が来たら、プログラム A の実行を一時中断し、後でプログラム A の実行を中断した所から再開できるようにプログラムカウンタやレジスタの値などをメモリ上の適切な場所に退避する。そして、またもや一定時間後に CPU に信号を送るように時計をセットし、プログラム B の実行に取り掛かる。時計から信号が来たら、同様にプログラム B の実行を中断して、プログラム C を実行する。時計から信号が来たら、同様にプログラム A の実行を一時中断した所から再開する。このように多数のプログラムを小刻みに切り替えながら実行するのだが、CPU は高速なのでユーザからは、あたかも多数のプログラムがすべて同時に動いているように見えるのである。

より厳密には、複数のプログラムの一つは OS であり、時計から信号が来ると CPU は必ず OS の実行に切り替わる。OS は実行中の各プログラムの優先順位や CPU 使用時間等を調べ、次に実行すべきプログラムを選択し、その実行に切り替える。

³正確には、1 つの CPU で複数のプログラムを同時に走らせることのできる製品も数多く存在する。1 つの CPU パッケージに複数の CPU コアが内蔵されたマルチコア仕様の物や、CPU コアに工夫をして 1 つの CPU コアが複数の CPU コアとして動作するようにした物などである。いずれの場合も、同時に実行したいプログラム数に対して十分な CPU 数を用意することは非現実的である。

UNIX 系 OS ではこれら実行中のプログラムのことをプロセスと呼ぶ⁴。計算機を使う時、同時に複数の端末エミュレータプログラム、例えば kterm、を実行させていると思うが、そのような場合、プログラムは 1 つでもプロセスは、画面上の kterm のウィンドウの数だけあることになる。kterm 同士でも大抵の場合各ウィンドウ内部の表示が互いに相異なっていることを考えれば当然であろう。また、ps コマンドを用いたより詳細なプロセスの情報を得ることも出来る。

⁴タスク、アクティビティなどと呼ぶ文化もあるようだ。