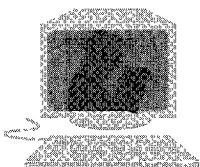


# インターネットの レイヤリング

太田 昌孝

東京工業大学 総合情報処理センター  
mohta@necom830.hpcl.titech.ac.jp



## OSIのレイヤリング

複雑になりがちなプロトコルを少しでも単純にすべく、プロトコルの役割分担をはっきりさせ、それぞれの内部を知らなくても独立にプロトコルを設計できるようにしたい。その分担方法として、抽象度を利用し、生のデータに近い部分(物理層)からはじめて最も抽象度の高い部分(アプリケーション層)という形で分けるのがレイヤリングの考え方である。コンピュータでいえばサブルーチンやライブラリに相当する考えで、そうたいしたことはない。

古典的なレイヤリングモデルとして、OSIの7層参照モデル(図-1)がある。物理層というのは、情報をどう物理現象に対応づけるかといったことを受け持つ。光がなければ0、あれば1という具合である。コンピュータでいえば入出力機器内部でタイミングを制御するファームウェアなどに相当する部分であろうか。データリンク層というのは、個々の物理層の違いを吸収してネットワーク層と物理層の間を埋めるものである。物理層が3つ以上の機器を含むが物理層ではそれらが区別できない場合や、複数の物理層を統合する場合には、局所的な機器の区別や通信も行う。コンピュータでいえばデバイスドライバに相当する部分であろう。そして、ネットワーク層というのは、複数のデータリンク層を統合して、端末同士に大域的な通信をさせる部分である。コンピュータでいえばファイル管理やプロセス間通信に相当しよう。

さて、ここまでに「統合」という言葉が2度出てきたが、もちろんこれは問題である。別々の場所で同じようなことをやるのはきちんとした機能分担ができていないということで、プログラムでもプロトコルでも混乱のもとであり、うまく混乱を治めたとしても、無駄が生じることになる。

そこで、最近のインターネットでは、データリンク層の機能を最小限に抑えようという考えがある。インターネットではデータをパケットで運ぶが、物理現象は直接パケットに対応せずにビット列にしか対応しないことが普通なので、この間の対応はデータリンク層の役割として残ることになる。物理層が3つ以上の機器を含みつつそれらの区別が物理層で行えない場合(たとえば無線)は、さらにデータリンク層の役割は大きくなる。一方、ネットワーク層を光ファイバー上の光の物理層(余計なことをしなければ2つの機器を結ぶだけ)に直接乗せようというIP over WDMの場合は、データリンク層の役割はパケット化だけで十分である(細かいことをいうと、IPv6インターネットのネットワーク層ではエラー検出をしないので、データリンク層の機能としてエラー検出の機能も望まれる)。

さて、ネットワーク層では個々の機器が区別できたが、同じ機器間でもファイル転送、音声通信、画像通信などといった異なる種類の通信が行われるわけで、これらを区別し、必要に応じて帯域や遅延などのQoS保証の手配をするのがトランスポート層である。コンピュータでいえば、プロセス管理に相当しよう。

この上には、セッション層、プレゼンテーション層、アプリケーション層が続くのだが、筆者にはこれらの層がどう役割分担しているのかまったく理解できなかったし理解しようという気も起きなかった。一応役割分担らしきものはあったような気がしたが、どうも恣意的だし、「統合」とか「エラー検出」だとかいった機能があちこちで見受けられ、まったく整理が行われていないのである。

コンピュータで考えれば、これらの層は個々のプロセス内部の構造を規定しているわけである。個々のプロセスの内部がどうなっているかなど、アプリケーションによって違うのが当たり前なのだから、参照モデルなど作ろうというのが間違っている。ひところ「構造化プログラミング」という言葉

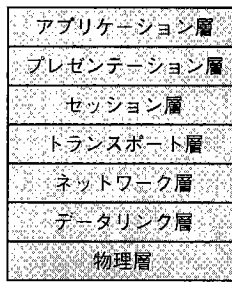


図-1 OSIのレイヤリング構造

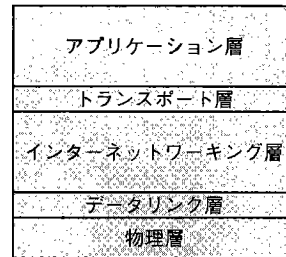
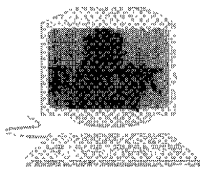


図-2 インターネットのレイヤリング構造

がはやったが、構造化は重要だがどう構造化するかを一般的に決めてもしょうがなかったのと同様、レイヤリングをすることは重要だがどうレイヤリングするかを一般的に決めてもしょうがない。

ところが、当時のネットワークでは、そうもいつてられない事情があった。

ネットワーク（といっても、電話網である）上ではさまざまなサービスが提供されるわけだが、そのためにはネットワーク事業者がネットワーク内部に数々のサーバを設置し、加入者の持つ端末機器をそれに接続させなければならない。異なる事業者の提供する同じサービスに同じ端末が接続できないと、加入者は端末を複数用意しなければいけない。そこで、ISOのような国際標準化機関でサービスごとにプロトコルを標準化しておくことが必要である。また、電話網事業者の考えるサービスは硬直しているわりに複雑なので、似たようなサービスごとにプロトコルやそれを処理するプログラムを全部作り直すよりは、参照モデルを決めておいて、それに従ったほうがいいという寸法である。もちろん、国際標準化機関のやることなので、妥協だらけでうまく分担などできるわけもない。



## インターネットのレイヤリング

インターネットの根本原理はエンドツーエンド原理と呼ばれているが、すべての処理はできるかぎり末端でやろうというものである。ネットワーク内部は単純にとどめ、QoS保証を行わない現状のインターネットでは、ネットワーク中にはインターネットワーキング層（OSIのネットワーク層に相当）以上の処理は行わない。（複雑な）データリンク層でパケットを転送するのがブリッジで、インターネットワーキング層でパケットを転送するのがルータであるが、それ以外の装置も置かない。

サーバを用意するのはネットワーク内部ではなく、サーバ

はネットワーク事業者に限らず、個人を含めただれもが設置することができる。サーバの内容も、電話網事業者が国際標準化機関で改まって議論する硬直したものだけでなく、各自がてんでばらばらに考える種々雑多なものである。一方、いまだき端末の方はコンピュータとしての十分な能力を持っているので、プログラムを入れ替えればどんなプロトコルにも対応できる。

そこで、セッション層とかプレゼンテーション層とかは考えるだけ無駄で、全部アプリケーション層としてまとめてしまえばよい。結局、インターネットのレイヤリングモデルは5層ということになる。

しかし、これをネットワークからみると、トランスポート層とアプリケーション層の境界は端末内部の問題であって、プロトコル上は本質的なことではない。トランスポート層のうちでも、UDPのようにポート番号によって通信を区別する部分は本質的だし、QoS保証が導入されればネットワーク内部でも意味が出てくるので一応2つの層が存在するのはいい。

しかし、一般的にはトランスポート層に属するとされているTCPの再送や速度の制御については、アプリケーション層と区別する意義は特でない。たまたまUNIX系の実装でOS内部でやっている部分をトランスポート層、ユーザプロセスでやっている部分をアプリケーション層と分けたという感がある。リアルタイムOSなどの実装においては、TCPはユーザプロセスから呼ばれるライブラリとなって構わないし、UDPでTCPと同様の再送や速度の制御をやる場合には、ユーザプロセスでやることになる。

しかし、トランスポート層とアプリケーション層の境界をあいまいにしておくと、似たようなことを両者の専門家がばらばらに行うという事態が発生する。なるべく多くの処理をできるだけ末端でやろうというインターネットのエンドツーエンド原理に従えば、トランスポート層はポート番号による区分だけに抑え、TCPの処理などもアプリケーション層に分類するのがむしろ適切であろう（図-2）。

（平成12年9月18日受付）