

謝辞

この本を読んでもくださる方に
気力をくれる友人に
大切な人に
感謝と本書をささげます

初版前書き

ネットワーク対応製品を開発していても、プログラマは TCP/IP やイーサネットのことをあまりよく理解していない。そんな現状があります。

そうでなくても、ゲーム用の NIC なんてオカルトな製品が出てくる世の中。同じお金で、Intel か 3com の NIC を買っておいたほうが幸せになれる、ということを理解するためには、ネットワーク技術の知識が欠かせません。

本書は、そのための解説書として作成しました。TCP/IP 理解の一助となれば幸いです。

改訂版前書き

コミックマーケット 78 で本書の初版を出しました。90 ページ近いとはいえ、コピー本としては 500 円という無茶な値段*1にもかかわらず、もっていった 20 部が午前中に完売という、ありがたくも自分にとっての誤算となる事態となりました。

その後、見本誌を自分でチェックすると細かい間違いも誤字脱字もいっぱいみつけて…テクノポリスかゲーメストか、という有様に、単なる再版でなく、全面的なリライトをしたのが、本書です。

IPv4 という、当面使い続けることになるとしても、いつか IPv6 に取って代わられてほしいテクノロジーをもとに解説していくのもどうかと思いました。せっかく同人誌なのだから、IPv6 で TCP/IP の教科書というのも…さすがに無理です。

本というのは、著者が理解していない項目を文章にしても、読者には理解してもらえません。本書は、著者の判る範囲で TCP/IP についてまとめています。

第三版前書き

コミックマーケット 79 ではじめてオンデマンド本にした改訂版も、コミックマーケット 80 にて完売させていただきました。そうするとまた直したくなるもので…それなんてオライリー商法、という感じではありますが、現在の知識をもとに、記事の追加修正を行いました。こっそり間違いも直しています。

一番大きな変更は、章立てを見直したことでしょう。本書の二章と三章、七章から九章までは前の版ではひとつの章でした。分割することで、異なることの詰め込み感があったことを少しなりとも解消できたかなと思います。

*1 キンコーズでのコピー代とホチキスの 3 号針の代金を回収させていただいた程度です。

また IPv4 か、という感想を頂く覚悟はしていますが、やはり IPv4 は現在のインターネット技術の基本です。完売した*2とはいえまだまだ現場では現役でしょう。*3IPv6 を理解する基礎体力作りには、是非本書をご活用ください。

インターネットで使用されている技術の基礎を、少しなりとも理解していただく助けになれば幸いです。少なくとも、参考文献に期した書籍をひもとくきっかけになればと思います。

第四版前書き

当サークルで毎回サークルカットに書いているくせに、出そうで出ない本であった本書ですが、5年のブランクを経て、ようやく、更なる改訂版が出せました。今回は、念願の IPv6 対応です。

今回は、章立てを大胆に入れ替えてみました。これまではネットワークアクセス層から上のレイヤに向かっての説明を行っていたのですが、アプリケーション層からトランスポート層への説明、ネットワークアクセス層の説明を経て、いわば上のレイヤと下のレイヤの知識を付けてから、インターネットプロトコル層について学ぶ形にしてみました。

また、本書では UDP が一番最後の章という変則的な後世になっています。これは、パケットそのものである UDP は、いわばパンツをはいた IP である、という理屈のもと、UDP の理解には IP の理解が必要なのではないか、そう考えたからです。

また、すっかり当サークルの顔になっている、インフラエンジニアの毒舌な妹に、章の序文と、いもうとコラムというコーナーを担当して貰いました。本書では、インフラエンジニアの毒舌名妹は、TCP/IP について、いろいろな関連知識の説明を担当しています。

特に IPv6 についてはまだまだ書き足りないところもありますが、IPv4 を使う TCP/IP と、IPv6 を使う TCP/IP、その同じところ、違うところの知見を得ていただければ、筆者としては幸いです。

第六版前書き

世の中は、スマホアプリを始めとして、ネットワーク対応のアプリケーションがあふれています。その状況で、TCP/IP について、基本的な部分を纏めた本というのはあまりありません。流行りの言語やフレームワークと比べると、新規性がないせいか数量的な見劣りがします。

そんな中、本書は、TCP/IP のことをわからないと言えないままネットワークアプリケーションを開発しているエンジニアが、TCP/IP の基本を理解するための本として、加筆訂正を行いました。

特に、IPv6 については、IPv4 ベースの TCP/IP の理解を前提としているネットワークエンジニア向けの本が多いため、TCP/IP の一部として初学者が学ぶためのテキストは多くありません。そこで、今回の版では、第四版から IPv6 に関する記述を増やし、TCP/IP のなかにある IPv6 を意識して貰えるようにしました。また、第四版は、TCP/IP のレイヤーの順番に拘らない解説を試みたのですが、立ち入った説明をするための前提条件の説明のため、代三版以前の構成に戻しています。

あわせて、プロキシ、NAT、NAPT に関する記事を追加しました。これらは現在では

*2 本書改訂版でなくもちろん IPv4 アドレスのことです。

*3 IPv6 環境での疑似ヘッダってどうなるんだろう…

家庭のネットワークにも存在するものです。ネットワークアプリケーションを開発するときに意識しておくべきものであるため、コミックマーケット 92 で出した同人誌から再録し、本書の内容に合わせての加筆訂正を行っています。

皆様が、TCP/IP について知る一助となれば、著者としてこれに勝る喜びはありません。

想定する読者

IT 関連でプログラマなどをしていて、普段はネットワークのことを気にしない人に、TCP/IP と、その物理インフラの代表であるイーサネットについて、について最低限理解してもらおう、というコンセプトで書いています。想定する読者は、TCP/IP について、正面から勉強したことはないけれど、何となくネットワークアプリケーションのプログラムを書いている、そんなエンジニアです。

そのため、インフラエンジニアの視点で見ると若干物足りない部分があるかと思いますが、この点は今後の本で埋めていきたいところです。

本書の内容

本書では、TCP/IP について、IPv4 と IPv6 を一度に学習するというコンセプトで、TCP/IP の解説を行っています。

■**第一章** TCP/IP というプロトコルの概論と、おおまかな全体像と、プロトコルスタックの概念についてについて説明します。TCP/IP は、複数のプロトコルが、役割を分担しながら他のプロトコルにサービスしたり、他のプロトコルからサービスを受けたりして通信をすることについての説明です。

■**第二章** ネットワークの物理媒体とその上での通信である、ネットワークアクセス層の概論と、一番簡単なネットワークである、エンドとエンドにホストがある、一対一通信のネットワークについての説明です。

■**第三章** ネットワークアクセス層で、一つの伝送媒体を複数のホストが共有するネットワークについて、説明を行います。その代表として、イーサネットの説明を行います。

■**第四章** トランスポート層についての概論的な説明です。トランスポート層における確実な通信とは何か、確実な通信が要らない場合、どのように通信を行うか、上位のアプリケーションとの対応付けはどのようにおこなうか、UDP や TCP の理解のために必要な知識についての説明をします。

■**第五章** トランスポート層のプロトコルのひとつである UDP について説明します。UDP は、インターネットプロトコル層の機能に、ポートによる通信多重化とエラー検出を付加した、比較的簡単な簡単なトランスポート層となります。

■**第六章** トランスポート層のプロトコルのひとつ、ストリーム通信を提供する TCP について説明します。TCP は下位のプロトコルが何であるかにもかかわらず、確実な通信を提供するプロトコルです。

■**第七章** アプリケーション層についての説明となります。電子メールの SMTP など、アプリケーション層のプロトコルについて説明をします。また、アプリケーションで IPv6

対応についても説明をします。

■**第八章** アプリケーション層での通信を中継するプロキシについて説明します。プロキシという、ルータと別の概念で中継をするというのはどういうことかについての解説です。

■**第九章** プロキシという概念でインターネットプロトコル層の中継をおこなう NAT についての説明をします。TCP/IP の知識を踏まえて、中継のためにヘッダ情報を書き換えることの副作用について乃解説を行います。

■**第十章** トランスポート層の中継を行うプロキシである NATP について説明します。IPv4 ベースのネットワークで、少ないグローバルアドレスを共有するための技術であり、現在のインターネットを理解するためには必要な概念です。

免責事項

本書に書いてあることは、筆者知識のレベルでまとめたものです。ですが、内容が正しいとは言いきれません。初版でも改訂版でも相当やらかしています。また、学校のレポート、業務などのコードを書く際に、本書の内容を信じて書いて損害が生じて、筆者にその責任はありません。

くれぐれも、自己責任と十分な検証の上、ご利用ください。

表紙イラスト

ゆうちゃん (コース英知)

目次

第 1 章	TCP/IP の概論	1
1.1	鳥類キャリアによる IP 伝送	1
1.2	伝書鳩でインターネットの通信をするための仕組み	2
1.3	インターネットプロトコルスイートと TCP/IP	6
1.4	エンドツーエンド原則	8
1.5	レイヤごとの役目とデータの名前	9
1.6	カプセル化とトンネリング	11
1.7	TCP/IP と OSI 参照モデル	14
第 2 章	ネットワークコミュニケーション層 (その 1) 概論と二つの端点を直結するネットワーク	19
2.1	ネットワークコミュニケーション層	19
2.2	いちばん簡単なネットワーク	21
2.3	SLIP	21
2.4	PPP	23
2.5	ループバックインタフェイスと自分宛の通信	24
第 3 章	ネットワークコミュニケーション層 (その 2) 複数のインタフェイスを結ぶネットワーク	27
3.1	複数のインタフェイスが接続されたネットワーク	27
3.2	イーサネット	28
3.3	イーサネットにおける共有伝送媒体の制御	32
3.4	MAC アドレスによるインタフェイスの区別	37
3.5	イーサネットフレーム	40
3.6	衝突ドメインの接続	43
3.7	現在のイーサネット	46
3.8	イーサネットとネットワークのループ	52
3.9	イーサネットの規格	55
3.10	ベースバンドとブロードバンド	57
第 4 章	インターネットプロトコル層 (その 1) インターネットプロトコルの概論	61
4.1	インターネット	61
4.2	そもそもインターネットとは何か	68
4.3	インターネットプロトコル層	69
4.4	インターネットプロトコル層	69
4.5	インターネットプロトコルアドレス	70

第 5 章	インターネットプロトコル層 (その 2) アドレスとルーティング	75
5.1	IP アドレス	75
5.2	グローバルなアドレスとプライベートなアドレス	84
5.3	IPv6 のマルチキャストアドレス	88
5.4	特別な意味を持つ IP アドレス	90
5.5	データグラムとヘッダ	91
5.6	経路集約と CIDR	96
5.7	ICMP	100
5.8	IP アドレスとネットワークコミュニケーション層のアドレスのマッピング	102
5.9	データグラムの寿命	105
5.10	フラグメントとゲートウェイ	105
第 6 章	トランスポート層 (その 1) トランスポート層の概論	109
6.1	トランスポート層とはなにか	109
6.2	トランスポート層の機能	112
6.3	疑似ヘッダ	115
第 7 章	トランスポート層 (その 2) UDP	119
7.1	UDP	119
7.2	UDP ヘッダと疑似ヘッダ	120
第 8 章	トランスポート層 (その 3) TCP	123
8.1	確実な通信	123
8.2	フロー制御	130
8.3	データの再送と送出量の制御	133
8.4	TCP ヘッダと疑似ヘッダ	136
第 9 章	アプリケーション層	141
9.1	アプリケーション層とはなにか	141
9.2	ポート番号の選択	142
9.3	アプリケーション層のプロトコル	144
9.4	ソケットとアプリケーションの実装	151
9.5	サーバの実装	154
第 10 章	プロキシ	159
10.1	アプリケーション層のプロキシ	159
10.2	プライベートネットワークからのアクセス	160
10.3	プロキシのかたち	162
10.4	プロキシの性質	165
10.5	プロキシの役目	165
10.6	リバースプロキシ	166
第 11 章	NAT	169
11.1	インターネットプロトコル層のプロキシ	169
11.2	アクセスとアドレス変換	169
11.3	静的 NAT	171
11.4	動的 NAT	172

第 12 章	NAPT	175
12.1	ホストによるグローバルの IP アドレスが少ない	175
12.2	NAPT	175
12.3	NAPT の通信の方向	177
12.4	デュアルスタックと NAPT	178
付録 A	TCP に関する補講	181
A.1	遅延 ACK の無効化	181
A.2	最大セグメントサイズ (MSS Maximum Segment Size)	181
A.3	Nagle のアルゴリズム	182
A.4	スロースタート	183
A.5	輻輳回避アルゴリズム	184
A.6	再転送と送信内容の再構築	186
付録 B	アプリケーション層に関する補講	189
B.1	魔法使いの弟子シンドローム	189
参考文献		193
あとがき		197

目次

1.1	インターネットプロトコルスイート	6
1.2	レイヤの上下関係	7
1.3	カプセル化	11
1.4	設計思想としてのトンネリング	13
1.5	TCP/IP と OSI 参照モデルの対応	15
2.1	ループバックインタフェース	25
3.1	ALTO ALOHA の概念図	31
3.2	衝突の検出	33
3.3	半二重と全二重	36
3.4	MAC アドレスの構造	38
3.5	イーサネットフレーム	40
3.6	衝突ドメイン	43
3.7	リピータとブリッジ	45
3.8	STP ケーブル	48
3.9	10Base-T、100Base-TX のピンアサイン	49
3.10	クロスケーブル	50
3.11	Cat.5e、Cat.6 ケーブル	51
3.12	リピータによるループ	53
3.13	誤学習とブロードキャストストームの発生	54
3.14	3Com EtherLink III PCI	57
3.15	マンチェスター符号と 4B5B+MLT3	59
3.16	記念館三笠で展示されている三六式無線機 (レプリカ)	60
4.1	二つのネットワークを結ぶ	63
4.2	三つ以上のネットワークを結ぶ	64
4.3	もっとたくさんのネットワークを結ぶ	64
4.4	Tier1 プロバイダとプライベートピア	66
4.5	デュアルスタック	71
5.1	IPv4 アドレスの構造	76
5.2	IPv6 アドレスの構造	80
5.3	ユニキャストアドレス	82
5.4	マルチキャストアドレス	83
5.5	エニーキャストアドレス	84
5.6	グローバルユニークアドレス	86
5.7	リンクローカルアドレス	87

5.8	ユニークローカルアドレス	87
5.9	マルチキャストアドレス	88
5.10	IPv4 ヘッダ	92
5.11	IPv6 ヘッダ	93
5.12	IPv6 のネクストヘッダ	94
5.13	IPv6 のホップバイホップヘッダによるジャンボグラム	95
5.14	経路集約	98
5.15	サブネット分割	98
5.16	ICMP データグラム	100
5.17	ICMP データグラムの構造	101
5.18	ARP 要求と ARP 応答	103
5.19	NDP による MAC アドレス問い合わせ	104
5.20	インターネット物理モデル	108
6.1	ポート番号によるアプリケーションの区別	113
6.2	インターネットプロトコル層が IPv4 の疑似ヘッダの構造	116
6.3	インターネットプロトコル層が IPv6 のときの疑似ヘッダの構造	116
7.1	UDP データグラムの構造	121
7.2	UDP 疑似ヘッダを含めた UDP データグラム	121
8.1	セグメント一つの送信と応答	124
8.2	シーケンス番号と確認応答	125
8.3	3way Handshake	126
8.4	コネクション切断	128
8.5	TCP ハーフクローズ	128
8.6	遅延 ACK とピギーバック	129
8.7	バルク送信とウィンドウサイズ広告	131
8.8	ウィンドウサイズ広告 (1)	132
8.9	ウィンドウサイズ広告 (2)	132
8.10	ウィンドウサイズ広告 (3)	132
8.11	ウィンドウサイズ広告 (4)	133
8.12	重複 ACK	134
8.13	タイムアウト	136
8.14	TCP ヘッダの構造	137
8.15	TCP オプションフィールド	137
8.16	疑似ヘッダを含めた TCP セグメント	140
9.1	RRQ/WRQ パケット	149
9.2	DATA パケット	149
9.3	ACK パケット	149
9.4	ERROR パケット	150
10.1	プロキシの概念	160
10.2	プロキシとルータの違い	161
10.3	プロキシ	162
10.4	ゲートウェイ	163

10.5	トンネル	164
10.6	リバースプロキシ	167
11.1	静的 NAT	171
11.2	動的 NAT	173
12.1	NAPT	176
12.2	ポートフォワードイング	178

1

TCP/IP の概論

TCP/IP はいまいちわかりにくいですよ。

インターネットを経由しての通信は、TCP/IP で行われているのは知っているでしょう。でも、アプリケーションから見たとき、TCP とか IP とかがどんな役割をしているから、離れたサーバとクライアントが通信できるのか。また、LAN とインターネットとはどんな風に繋がっているのか。そんな部分がよくわからない。

まずは、TCP/IP というもののイメージを掴んでみることにしましょう。

この章では、通信を行う両社を、慣例に従って Alice と Bob という名前で表記します。

1.1 鳥類キャリアによる IP 伝送

伝書鳩でインターネット通信ができる。そのための規格があることをご存知だろうか。

インターネットにおける規格を提案する RFC^{*1}の 1149 番で、鳥類キャリアによる IP 伝送 (IPoAC IP over Avian Carrier) という規格が提案されている。^{*2}

IPoAC について簡単に説明すれば、紙に通信したいデータを書いて、伝書鳩^{*3}にくくりつけて飛ばす、そうやってインターネットの通信を行う手順の提案である。このように、IPoAC は、鳩に持たせるための紙に書いたデータを作成する規約、鳩の挙動などについて言及し、伝書鳩をインターネットの通信に使用できるようにした規格の提案として提出されたものだ。

種を明かせば、IPoAC は 1990 年 4 月 1 日に発行された提案であり、ジョーク RFC と呼ばれるもののひとつである。RFC では、毎年 4 月 1 日非は、このようなジョーク RFC を発行する習慣がある。

IPoAC はジョーク RFC ではあったが、実証実験が行われている。その実証実験では、鳩が宛先に辿り着かなかったことによって生ずるデータの損失が多く、鳩なので伝送遅延も大きい。つまり、通信のやり直しや通信そのものにかかる時間が大きいという、ある意味当然の結論が出た。だが、IPoAC は、これらの問題を許容すれば、伝書鳩でインター

^{*1} Request for Comment

^{*2} <https://tools.ietf.org/html/rfc1149>

^{*3} 規格では avian carrier であり、伝書『鳩』と明記はされていない

ネットの通信を行うことが可能であることも、その実証実験でしめした。

いもうとコラム 惑星間インターネット

RFC1149 はいわゆるジョーク RFC でしたが、インターネット通信において、送ったデータのロスや応答時間 (レイテンシ) が大きくなる環境があります。それは、宇宙です。さよならジュピターでも、カイパーベルト領域ま行った探査船のコンピュータであるナヴァホと、月に設置されたコンピュータのティム・ラビットがレイテンシを越えて会話するシーンがありました。

このような環境のインターネット通信は、惑星間インターネット (Interplanetary Internet) として研究開発が行われています。

1.2 伝書鳩でインターネットの通信をするための仕組み

鳩は遅い。そして、送り出した鳩が、目的とする通信相手にたどり着くわけでもない。また、順番通りに届くわけでもない。だが、鳩を伝送媒体に用いて、でインターネットで行われている通信が成立する。

そのような条件で通信ができるようにするには、どうすればよいのだろうか。それは、伝送遅延を許容し、鳩は確率的にしか相手のところにたどり着かず、送り出した順番と到着した順番が一致しないことを前提に通信の仕組みを作る。つまり、鳩に完璧さを求めず、それをサポートする仕組みをつくれがよい。

伝書鳩の役割とは、通信を行う双方の間で、情報を物理的な空間を越えて運ぶことである。鳩なので、途中で餌になる虫を見つけたり、発情期で交尾相手を見つけたり、鷹に追いかけて逃げたりするかもしれない。これらの可能性は、IPoAC でも検討されている。鳩は、通信の焚いての所に到着する時間がわからない。それどころか、相手のところに到着しないこともある。送り出した順番に鳩が到着することもないだろう。

まず、鳩なのだから、必ずしも通信の相手側に到着するわけではない。これを前提条件として、伝書鳩で通信をするための仕組みを考えて聞くことにしよう。

1.2.1 鳩を使って確実な通信をする方法

現在のインターネットは、確実な通信を行っている。だが、IPoAC を使ってインターネットの通信を行うのなら、データを運ぶ手段が鳩であるという前提で、確実な通信をおこなう手段をかなが得なければならない。それは、相手にデータが届いたことがわかるまで、同じデータを送り直し続ける、ということである。この過程をくりかえしていけば、確率的にいつか通信が成功するだろう。

最初に、Alice と Bob で、鳩が行き来できる状況なのかを確認する。その確認が取れないと、いくら鳩を飛ばしても通信が成立しない可能性があるということだ。

次に、鳩が届けたデータを受け取った側は、Alice に向けて飛ぶ鳩を使って、データを受け取ったという返事をするにすることにする。このとき、データを運ぶ手段は鳩しかない。電話など、鳩以外の手段で通信を届けることはできない。

三つ目、データを送信した側は、一定時間待って、データを受け取ったことを知らせる鳩が通信お相手から来なければ、送り出した鳩が到着しなかったとみなして、もう一度鳩を送り出す。これは、相手がデータを受け取ったことが確認できるまで3は、そのデータは届いていないと見なして再送する、ということである。

最後に、送信するデータには、送信した順番を再現するための通し番号を書いておく。受信したというメッセージには、その番号も書き添え、何番目のデータを受け取ったかがわかるようにする。

この手順は、TCP/IP で、確実な通信を行うため仕組みを、ごく簡単にしたものである。

1.2.2 鳩の遣り取りができるかの確認

最初に、Alice から、鳩が相手にたどり着ける状況にあるのかを確認する。その方法は、通信を開始したい、という内容のデータを持った鳩を、Bob にとばす。そのはとを受け取った Bob は、鳩が届いた、こちらも通信できる、というデータを持たせた鳩を、Alice にとばす。Alice にその鳩が到着したら、Alice は、鳩を飛ばしても大丈夫だと知ることができる。次に Alice は、鳩が到着した、というデータを持たせた鳩を、Bob に送る。これによって、Bob は、Alice と鳩の遣り取りができる状況なのを知ることができる。

このように、Alice と Bob が鳩を送りあって状況を確認することは、TCP の 3way-Handshake という動作似相当する。

Alice と Bob の協調

次に、IPoAC で、Alice と Bob はどのくらい協調して通信を行うのだろうか。前提として、この両者は、鳩以外の情報伝達手段を持っていないことを再度確認しておきたい。変な言い方ではあるが、我々はインターネットで通信を行う際に、インターネットしか情報伝達手段を持っていない。

Alice は、データを送る必要がある場合は、Bob への事前連絡なしに鳩を放つ。IPoAC では鳩より早い情報伝達手段がない前提である。そのため、事前連絡のしようもない。そのため、Bob は、いつ鳩がきてもいいように受け入れる準備をしている。だが、鳩がこない限りは何もしない。大切なことなのでもう一度書くと、鳩より速い通信手段はないのだ。なので、Alice に事前の連絡を取ることはできない。

Bob から送られる、到着したという情報についても同様である。Bob は、Alice への事前連絡なしに、到着したという返事を持たせた鳩を放つ。Alice は、到着した、という連絡を持った鳩がいつ来てもいいように準備している。だが、鳩が来ずにに時間切れになったら、こんどは Alice が Bob への事前連絡なしに、先ほど送り出した鳩と同じデータを持った鳩を送り出す。

これは、TCP/IP における、TCP がデータを確実に届ける手順二層等する。

到着した鳩の順番が違っていった場合

到着した鳩が持っていたメッセージの順番が違っていった場合は、どのようにすれば良いだろうか。たとえば、1番データを持った鳩が来たが、次に来たのは3番のデータを持った鳩であった場合である。2番のデータが来てないことはどう連絡すればいいだろうか。

Bob は、1番のデータが届いた、という応答を、鳩を使って Alice に送る。だが、2番のデータに関しては何もしない。そもそも、2番のデータが存在するかわからないのだ。来ないという連絡のしようもない

Alice は、2番のデータが「受信できた」というメッセージが来なかったことで、2番以降のデータを送り直す。2番のデータだけ送り直すのではない。一見無駄が多いようだが、こうすることで、どの鳩の到着の連絡があったかの管理をせずに済むというメリットがある。

1.2.3 鳩の到着の連絡を省略する通信

ここまで説明した方法は、データを受け取った、というメッセージのやりとりが成立するまで、り鳩が再び放たれ続ける。正直なところ、これは面倒だし、鳩の到着と、それを知らせる鳩の送付のコストは、Alice、Bobのどちらにとっても高い。そのため、受信した、という確認を省略する通信というものがある。

たとえば、一匹の鳩の脚にくくれる程度の問い合わせと、おなじく一匹の鳩の脚にくくれる答えで終了する通信があるとすると、このとき、到着した、という連絡にかかる手間は、行われる通信のデータ量に対して、あまりにも大きい。そのため、現実性がなくてもかまわない場合は、到着したという連絡を省略することで、通信のコストダウンを計ることができる。

このような、受信の確認を省略する通信は、TCP/IPにおけるUDPに沿うとする。

1.2.4 鳩が飛ぶ先の表し方

IPoACで通信を行うとき、鳩はどこに向けて飛ばすのか。伝書鳩は、ある宛先にむけて飛ぶように訓練される。つまり、宛先ごとに、そこに向けて飛ぶ鳩を用意して、決定した宛先によって鳩を選択する。

鳩を飛ばす人間は、宛先をどのように管理するのだろうか。それには、住所表示を使うことにしよう。そうすることで、鳩を管理する人間が、どこに飛ぶ鳩を使うのか管理しやすくなる。

1.2.5 二つの住所表示

京都市は、住所の表し方が二つある。何区何町何丁目何番地、という表し方は、通常の住所表示として使われる。だが、古い表現として、街路が碁盤の目になっている京都では、場所のブロックがどの通りに面しているかを、東西方向、南北方向の通りの交差点から、東西南北どちらに進めばいいかで表現するやりかたがある。

たとえば、京都市役所は、現在使用されている住所表記では京都市中京区押小路河原町西入榎木町450-2であるが、古い表記では京都市中京区寺町通り御池上ル本能寺前となる。だが、京都市役所を宛先とする郵便を出すときは、このどちらで記載しても届く。^{*4}

京都の住所表記にtuいて説明したのは、ひとつの場所を表す住居放棄に、複数の表現方法がある場所としてだ。古い表記は比較のおおざっぱ、新しい表記は細かい。だが、実際の地図上の場所は同じである。新しい表記は、より細かく場所を特定することができる。つまり、本能寺前に市役所以外の建物があったとしても、番地の番号まで使えば、市役所とは別の住所としてで表すことができるだろう。

だが、そんな違いがあったとして、メッセージを作る人間は、京都市役所という場所を宛先として指定する。住所の表記の違いは、京都市中京区寺町通り御池上ル本能寺前に向けて飛ぶ鳩を使うのか、京都市中京区押小路河原町西入榎木町450-2にむけて飛ぶ鳩を使うのか、という使い分けを行うことに相当する。

この二種類の表記があることが、後に説明するIPv4とIPv6に対応する。

^{*4} 京都では、古い住所表記に使われる通りの名前を、「まるたけえびすに、おしおいけ、あねさんろっかく」というような歌にして覚えていた。この歌はいくつかあるので、興味があれば調べてほしい。

1.2.6 鳩と人間の役割分担

次に、IPoACを使った通信での、鳩と人間の役割分担について考えてみよう。そのために、鳩と人間がIPoACにおいてどのように振る舞うかを再確認したい。

1.2.7 IPoACとインターネットプロトコルモデル

では、実際のインターネットにおいて、鳩はどこにいるのだろうか。正確に言えば、鳩に相当するのはどの部分なのだろうか。それを考えるために、IPoACを、実際のインターネットに相当するモデルに当てはめてみよう。ここまで説明した内容で、メッセージを作ってからとはを送り出すまでの手順を分けて書いてみると、表1.2.7のように現れる。

役割
メッセージをデータとしてを作る
データの送受信の管理
鳩を飛ばし受け入れる
鳩がデータを運ぶ

表 1.1 人と鳩の役割分担

これをインターネットの用語に置き換えてみよう。説明はこの先で行うので、今はインターネットでの用語が何になるかだけ見てもらうのでかまわない。

役割	インターネットでの名称
メッセージをデータとしてを作る	アプリケーション層
データの送受信の管理	トランスポート層
鳩を飛ばし受け入れる	インターネットプロトコル層
鳩がデータを運ぶ	ネットワークコミュニケーション層

表 1.2 インターネットの用語との対応関係

表1.2.7のように、メッセージを作るものをアプリケーション層、データの送受信を管理するトランスポート層、鳩を送り出し、受け入れるのがインターネットプロトコル層、鳩そのものが、ネットワークアクセス層とよばれる、それぞれの層（レイヤー）に相当する。

この四つの機能が菱餅のように重なり合っているイメージでとらえられることから、いわゆるTCP/IPのことを、インターネットプロトコルモデルと呼ぶ。

1.2.8 プロトコル

プロトコル (Protocol) という言葉は、何を意味するのだろうか。辞書では、外交手順、儀礼、議定書、という意味の単語である。ネットワークにおけるプロトコルとは、ネット

ワーク機器が相互に通信を行うための規約、規格のことである。TCP/IP という名前は、Transmission Control Protocol/Internet Protocols というように、二つのプロトコルという言葉が含まれている。

IP は、複数の異なるネットワークの間で通信を行うための規約である。また、TCP は、IP のサービスをデータの伝送のために利用して、確実にデータを届けるための規約である。

TCP/IP は、TCP と IP だけでなく、UDP や各種のアプリケーション層のプロトコルというように、複数のプロトコルが集まった規格である。このような、各種のプロトコルが集まって成り立っている規約を、プロトコルスイートと呼ぶ。インターネットプロトコルスイートは、インターネットの通信のための規約の集合体である。

いもうとコラム IPoAC は何を定義しているのか

実際のところ、IPoAC は何を定義しているのでしょうか。それをインターネットプロトコルスイートの言葉を使うと、ネットワークコミュニケーション層である鳩に、どのようにインターネットプロトコル層でのデータを載せ、取り扱うのか、という部分になります。

これは名前からもわかることで、IP データグラムを鳩に乗せて運ぶから、IP over Avian Carriers となるのです。

1.3 インターネットプロトコルスイートと TCP/IP

下のレイヤーのサービスを利用する



上のレイヤーにサービスを提供する

図 1.1 インターネットプロトコルスイート

インターネットの通信方法である TCP/IP は、このように、役割分担した機能を組み合わせることによってできている。そのため、このモデルをインターネットプロトコルスイート、と呼んでいる。また、その機能の一つ一つに層（レイヤ）という言葉がつくのは、図 1.1 のように、上下に層となって重なっているように表されるためだ。

では、ここまで何となく使ってきた TCP/IP という用語は何であろうか。TCP はトランスポート層のプロトコルの一つ、IP はインターネットプロトコル層のプロトコルの名称である。一見すると、表??の、トランスポート層とインターネットプロトコル層のプロトコルの名前をつなげただけに見える。だが、TCP/IP と言うときは、インターネットプロトコルスイートそのものを現す名前となる。

1.3.1 レイヤの上下関係とサービス

インターネットプロトコルスイートの各層は、自分より下の層が自分にサービスすることを前提に、自分より上の層に対してサービスを行う。

もういちど IPoAC で説明すれば、インターネットプロトコル層は、ネットワークコミュニケーション層である、鳩がデータを運ぶのを利用してデータを送り出し、受け入れる。そして、受け入れたデータをトランスポート層にわたし、トランスポート層から送り出すべきデータを受け取る。インターネットプロトコル層は、鳩がデータを運んでくれば、途中でどんな飛び方をするか、そして、鳩が到着するかどうかについては、全く関知しない。そして、トランスポート層には求められるサービスを提供するだけで、トランスポート層が何をしているかは全く関知しない。

このことを、あるレイヤーが下から数えて n 番目にあるとき、 n 層のサービスは、 $n-1$ 層からサービスを受け、 $n+1$ 層にサービスする、というように言い表す。

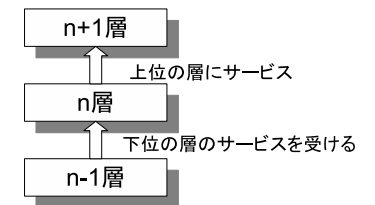


図 1.2 レイヤの上下関係

1.3.2 レイヤ間の依存関係とプロトコルスタック

層の間の依存関係はどうであろうか。実は、それぞれの層は、お互いに全く依存しあわない。ネットワークコミュニケーション層として鳩を使うかわりに、糸電話を使っても、瓶詰めの手紙を海に流しても、はじめてのお使いをする姪っ子に持たせても、通信は成立する。このことを表す言葉として、Two can and tin. というフレーズがある。意識すれば「糸電話でもいいよ」となる。

これは、あるレイヤは、直接に接していないレイヤのことは全く気にする必要がないということでもある。トランスポート層は、情報を運ぶ何かが鳩なのか糸電話なのか、それとも姪っ子なのかを考える必要がない。

また、本書ではインターネットプロトコル層について、バージョン 4 の IPv4 とバージョン 6 の IPv6 の両方について、説明を行う。インターネットプロトコル層が、IPv4 と IPv6 のどちらでも、トランスポート層、ネットワークアクセス層は、プロトコルの変更なく通信を行うことができる。^{*5}

トランスポート層の置き換えの例もある。インターネットの通信で WAN 高速化を行

*5 実装の観点で見れば、インタフェイスの違いなどから、全く変更が必要ないわけではない。

う機器がある。これは、対向する機器の間でトランスポート層で、TCP や UDP などの従来のプロトコルと違うものに置き換えて、通信時間の短縮を計る。このような通信が成立するのは、インターネット層が、トランスポート層が何をやっているかを関知しないためである。

ここまで説明したように、TCP/IP は異なった役割をもつプロトコルが、お互いにサービスを提供したりされたりして成り立っている。図にすると、各層のプロトコルを上下に重ねたように表される。このように、役割を分けたプロトコルが上下に重なる形で全体像が作られるプロトコルスイートを、プロトコルスタックと呼ぶ。

1.4 エンドツーエンド原則

TCP/IP には、二つの意味を持つエンドツーエンド原則というルールがある。ひとつは、通信の処理は、通信の当事者である両端でのみ行い、途中経路は関与しないという設計思想である。そして、もうひとつは、通信の当事者は対等であるべきという思想的なものとなる。

1.4.1 処理は両端(エンド)でのみ行なう

アプリケーション層やトランスポート層による制御は、通信の端点(エンド)側だけで行い、途中の経路は通信の導管に徹する、という設計思想である。TCP/IP の用語では、途中の経路は、ネットワークコミュニケーション層と、インターネットプロトコル層までが関与する。そして、トランスポート層やアプリケーション層での通中に介入しない。そうすることで、通信のエンドとなる機器にのみ、トランスポート層やアプリケーション層を実装すればよいことになる。

TCP/IP は、途中経路の実装を簡単にし、経路の敷設を行いやすくする設計思想であった。トランスポート層以上の動作には、かなりの CPU のソースを必要とした。そのため、重い処理をするノードを減らしたかったという、歴史的な事情もある。

設計士粗糖としての根拠として、根拠として、コンピュータリソースが潤沢となった現在では、厳密に守られることはなくなった。プロキシなど、トランスポート層以上の動作をする者が、通信の途中に介入することもあるためである。

1.4.2 通信の両端は常に対等である

もう一つは、インターネットに接続されたすべてのエンドは、対等な立場で通信が可能であるべきという原則である。対等な立場というのは、インターネットに接続されたすべてのホストは、自分を含むすべてのホストに対して送信を行うことができ、逆に、すべてのホストからの通信を受信できるべき、ということである。

もっとも、現在のインターネットはこのエンドツーエンド原則の理念は失われている。犯罪対策のブロッキングは、経路で他のホストへの通信を制限してしまう措置である。だが、エンドツーエンド原則が提唱された当時は、インターネットに接続していた組織がすべて顔見知りであった、いわゆる性善説が成り立っていた時代であることを記載しておかなければ不公平になるであろう。

いもうとコラム 実際のエンドツーエンド

エンドツーエンドは、今ではあまり現実的でない考えであるという見方もあります。たとえば、プロキシやファイアウォールは、インターネットプロトコル層よりも上のレイヤーで通信を処理し、中継したり、通信を遮断します。

それでも、両端から見てインターネット層の通信で結ばれているように見れば、通信は成立するということでもあります。それが、インターネットプロトコルの通信を、アプリケーション層のデータとして運ぶ「トンネル」の考え方につながっています。

1.5 レイヤごとの役目とデータの名前

では、各層の役目を、もう少しだけ、インターネットの用語を使って説明し直そう。また、隠れいやーで取り扱うデータには、それぞれ名前がある。

1.5.1 ネットワークコミュニケーション層

同じネットワークの中で、ネットワークに接続されたインタフェースを区別し、通信するための層である。同じネットワークとは、二つ以上の機器が、共有する伝送媒体によって直接に接続されたものである。ケーブルや信号などの電気的な規格と、それを利用してどのような情報を送るか、それらをまとめが概念がネットワークコミュニケーション層である。

概念と書いたのは、ネットワークアクセス層そのものは TCP/IP では定義されていないためだ。本書では後の章で、説明のために PPP やイーサネットを取り上げているが、これらはインターネットプロトコルスイートの中でなく、別の規格をネットワークコミュニケーション層として利用しているものである。

ネットワークコミュニケーション層のデータで、可変長のものを、フレームと呼ぶ。本書で解説するネットワークコミュニケーション層のデータは可変長なので、フレームとなる。また、本書では詳細には立ち入らないが、固定長のデータを用いるとき、そのデータのことを、セルと呼ぶ。たとえば、ATM(Asynchronous Transfer Mode) という規格では、53byte 固定長のセルを通信の単位としている。また、光方式の WAN で使われる GPON(Gigabit Passive Optical Network) は、固定長のセルの中に、可変長のイーサネットフレームをただ入れて伝送する方式である。

1.5.2 インターネットプロトコル層

インターネットプロトコル層は、ネットワークコミュニケーション層というネットワークが複数あった場合に、そのネットワークとネットワークの間での通信を担当する。ただし、インターネットプロトコル層は、ネットワーク間の通信手段を提供するのが役目であり、確実に通信が成立しているかは保証しない。

IP アドレスは、インターネットプロトコル層でネットワークとそこに接続されたホストを特定するための識別手段であり、インターネットにおける文字通りの住所である。

インターネットプロトコル層では、これまで使われてきた IP バージョン 4 と、より多くの IP アドレスが使える、新しい規格の IP バージョン 6 という、二つのバージョンのインターネットプロトコル層の規格が、2018 年現在は併用されている。IPv4 と IPv6 は

同時に使用することが可能であり、同時に使用されていることを、IPv4をインターネットプロトコル層とするスタックと、IPv6をインターネットプロトコル層とするスタックの二つガルという意味で、デュアルスタックという。

インターネットプロトコル層のデータを、IPデータグラムと呼ぶ。また、IPパケット、という呼ぶこともある。これは、送り出され卵だけのデータを、電信（テレグラム）や小包（パケット）に見立てたものである。

1.5.3 トランスポート層

トランスポート層には大きく二つの役割がある。一つは、ポート番号とよばれる、通信を行うアプリケーションを特定するための番号を提供して、一つのIPアドレスを用いて複数のアプリケーションが同時に通信できるようにする、多重化である。多重化は、TCPとUDPに共通する機能である。

もう一つは、TCPの昨日で、エンドツーエンドでの確実な通信を担保することである。確実な通信が成立している条件は、通信相手がデータを受信したことを確認した状態であるとする。また、データの到着順、つまり通信内容の送信順を確認して、データのつじゅしん側でその順番を再現する、つまり、通信の順番も担保している。

トランスポート層のデータは、プロトコルごとに異なる名前と呼ばれる。TCPのデータをセグメント、UDPのデータを、UDPデータグラムと呼ぶ。後ほど説明するが、UDPのデータは、IPデータグラムとほぼ同じ性質を持つため、区別してUDPデータグラムと呼ばれる。

コネクションとコネクションレス

TCPのように、確実な通信を行うプロトコルは、その動作から通信のエンド同士が直接接続されているかのような状態をエミュレートしていると考えられる。そのため、コネクション型、もしくはコネクション指向の通信と呼ぶ。古い資料では、電話交換網でエンドとエンドの間が電話線で結ばれている状態に見立てて、仮装交換回路（バーチャルサーキット）と呼んでいるものがある。

一方のUDPは、確実な通信を行うために必要な、「受信した」追う乙の送付やデータの到着順の管理などは行わない。つまり、コネクション指向の動作はおこなわない。そのためコネクションレス型と呼ぶ。

1.5.4 アプリケーション層

アプリケーション層は、通信のエンドとエンドで通信を行う主体である。つまり、TCP/IPというのは、このアプリケーション間の通信を行うために存在すると言っている。アプリケーション間の通信を、プロセス間通信とよぶ場合がある。

アプリケーション層とは、トランスポート層以下が提供する通信を使って、他のアプリケーション層と通信する。他のホストの別のアプリケーションと通信するときの規約は、アプリケーション層のプロトコルと呼ばれる。また、アプリケーション層のプロトコルごとに、トランスポート層でTCPを使うか、UDPを使うかが決められる。

たとえば、SMTPやHTTPといった、確実な通信を前提としたプロトコルを使用するときは、トランスポート層にTCPを使用する。また、DNSのような、問い合わせと応答に対して、受信したという通信のコストが大きいプロトコルや、SNMPのようにひたすらデータを待つプロトコルの場合は、UDPが使用されることが多い。

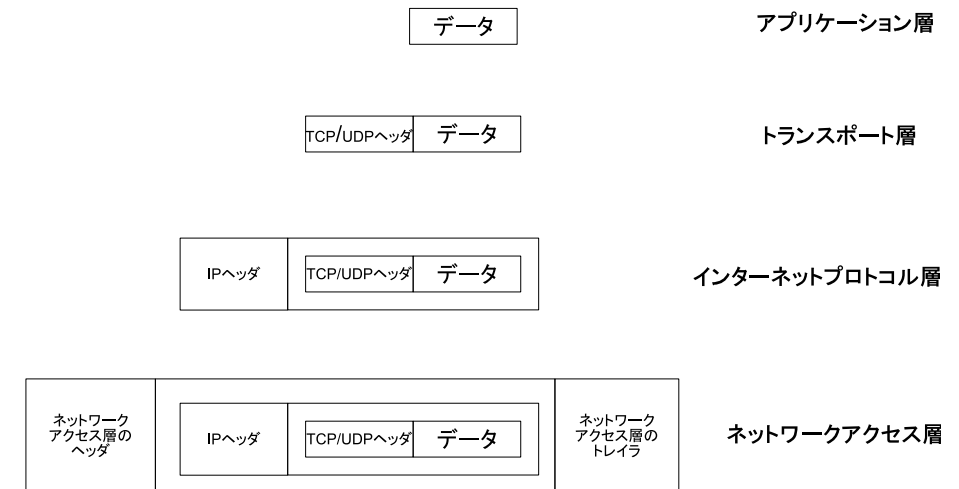


図 1.3 カプセル化

最も現在は、コンピュータリソース全体で、TCPの通信を行うために占めるコストが下がったこともあって、これまでUDPを使用してきたプロトコルをTCPで置き換える場合がある。

アプリケーションのデータは、下位のプロトコルにTCPを用いるときには、ストリームと呼ばれることが多い。UDPの場合は、特別な名称はない。だが、問い合わせと応答の組み合わせとなるアプリケーションでは、クエリとリザルトというように呼ばれることもある。

1.6 カプセル化とトンネリング

実際にインターネットプロトコルスイートで通信を行う場合は、カプセル化とトンネリング、という二つの概念を意識することとなる。それについて説明を行おう。

1.6.1 カプセル化

TCP/IPのように、複数のレイヤからなるプロトコルには、カプセル化という概念がある。この概念を、アプリケーション層からどのようにデータがネットワークコミュニケーション層に送り出されるかを見てみよう。

アプリケーション層から発行されたデータは、トランスポート層で扱えるデータにしてやらないと、トランスポート層から送り出すことができない。トランスポート層から送り出せるデータにするには、トランスポート層のデータとして必要なデータを追加する。具

体的には、アプリケーション層のデータの先頭に、トランスポート層のヘッダを付加する。また、インターネットプロトコル層で扱えるデータのサイズに切り分けることも、トランスポート層で行う。

通信に使うための情報部分を、データ先頭にあることか羅、ヘッダ情報、あるいは単にヘッダという。ネットワークコミュニケーション層では末尾に着けることもあり、そのような付加情報をトレイラとよぶ。また、上位レイヤーから受け取ったデータを、ペイロードと呼ぶ。

あるレイヤーが、上位のレイヤーのデータを加工して、自分のレイヤーのデータにしてから、次のレイヤーに渡すことを、カプセル化と呼ぶ。粉薬をカプセルに入れて、固形の薬という形に変えるイメージでカプセル化と呼ぶ。このカプセル化は、アプリケーション層のデータをトランスポート層に送り出すときだけではない。トランスポート層から送り出されたデータは、インターネットプロトコル層で扱えるようにする必要がある。このときも、インターネットプロトコル層に必要なデータを付加する、カプセル化が行われる。更に、インターネットプロトコル層からネットワークアクセス層にデータが送り出される際も、同様にカプセル化が行われる。

最終的に、アプリケーション層のデータは、トランスポート層、インターネットプロトコル層、ネットワークアクセス層という三重のカプセルに包まれて、ビット列としてネットワークに送り出される。

1.6.2 カプセルをはがす

では、受信側に届いたデータはどうなるのであろうか。データを受信したネットワークコミュニケーション層は、ネットワークコミュニケーション層で通信するためのデータを外し、インターネットプロトコル層にデータを渡す。インターネットプロトコル層、トランスポート層でも同様に、自分が通信に使うデータを外して、一つ上のレイヤに残りデータを渡す。

最終的に、アプリケーション層は、送信元のアプリケーション層が発信したデータのみ受け取る。

1.6.3 トンネリング

トンネリングという言葉には、二つの似て異なる意味がある。それは、エンドツーエンドにおける、設計思想としてのトンネリングと、プロセス間通信の手段に一般名称としてのトンネリングである。

設計思想としてのトンネリング

設計思想としてのトンネリングは、カプセル化を通信のエンドから見た視点で説明したものである。

インターネットプロトコルスイートのあるレイヤとレイヤの間の通信は、いわば同じ階層にあるレイヤの間でトンネルを通して、途中で何があれを無視して、直接向かい合っているイメージである。そのため、同じ階層にあるあるレイヤからレイヤへの通信を、トンネリングという。これは、エンドツーエンド原則における、通信の両端のトランスポート層、アプリケーション層は、そのしたの経路が何であれ、直接通信をしているものとしてやりとりする、ということである。

下位のレイヤーが何であるかに関わらず
他のホストの同じ層と通信

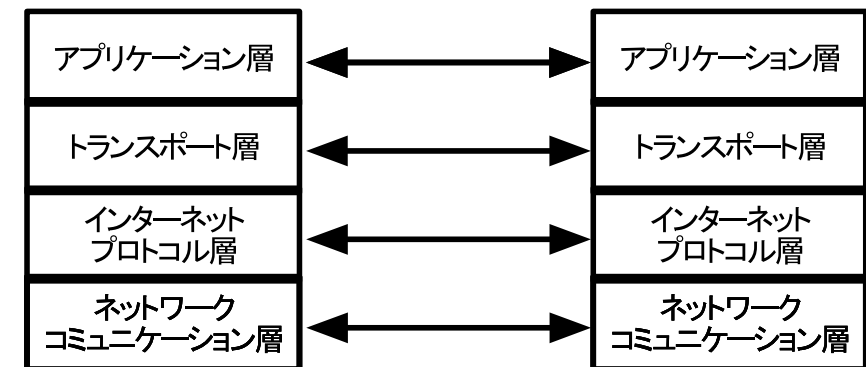


図 1.4 設計思想としてのトンネリング

プロセス間通信の手段としてのトンネリング

トンネリングにはもう一つの側面がある。同じ階層にあるレイヤからレイヤの通信は、かならずしもひとつ下のレイヤの通信を使う必要はない。例えば、ネットワークコミュニケーション層のフレームのビット列を、アプリケーションのストリームとして送ったと考えてみよう。そうすれば、本来は同じネットワークでしかできないネットワークコミュニケーション層の通信が、違うネットワークに置かれた機器同士で可能となる。当然、エンドにはそれぞれ、ネットワークアクセス層のフレームをキャプチャし、それをアプリケーションのデータとして送り出すアプリケーションを動かしておかなければならない。

また、インターネットプロトコル層のペイロードとして、IP データグラムを載せて通信することで、インターネットプロトコルでは直接通信できないネットワーク間の通信が可能となる。これが、VPN(Virtual Private Network) の概念である。

いもうとコラム 闘士ゴーディアン

1979年に放送された、闘士ゴーディアンというタツノコプロのロボットアニメを知っていますか、あるいは、覚えていますか。この作品の主役ロボットは、いわば着用型のパワードスーツです。

主人公のダイゴ大滝は、プロテッサという小型ロボットを着用します。ユニークなのはここからで、プロテッサは、デリンガーという一回り大きいロボットを着用します。更にデリンガーは、ガービンというもう一回り大きいロボットを着用します。

長々とゴードリアンの話をしたのは、TCP/IPにおけるカプセル化のイメージがまさにこの形だからです。アプリケーション層のデータがダイゴ大滝であるとすれば、プロテッサー、デリンガー、ガービンの着用は、各層のカプセル化に相当する、というわけです。

余談ながら、ゴードリアンは玩具先行のデザインなので、DX超合金では、関節の処理に破綻がありません。ダイゴ大滝、プロテッサー、デリンガー、ガービンと着用した状態で、ガービンの間接が稼動するという今見てもすごいおもちゃです。

1.7 TCP/IP と OSI 参照モデル

ネットワークの役目を層として表すものとして、OSI 参照モデル (Open Systems Interconnection Reference Model)^{*6}、と呼ばれるモデルがある。OSI 参照モデルは、TCP/IP とは全く別に、IEEE によって制定された規格である。

古いネットワークの教科書では、TCP/IP でなく、OSI 参照モデルが取り上げられていることがある。かつては、研究所発祥の TCP/IP は、そのうち役目を終えて、OSI 参照モデルという正しい存在に置き換えられると洋装されていた。だが、実際には、TCP/IP は研究所のネットワークを飛び出し、世界中でネットワークを繋ぐために使われている。

OSI 参照モデルは七つのレイヤを持つ。そのレイヤの名前と、対応するレイヤ番号は、表 1.7.3 となる。

レイヤ番号	レイヤ名
L7	アプリケーション層
L6	プレゼンテーション層
L5	セッション層
L4	トランスポート層
L3	ネットワーク層
L2	データリンク層
L1	物理層

表 1.3 OSI 参照モデル

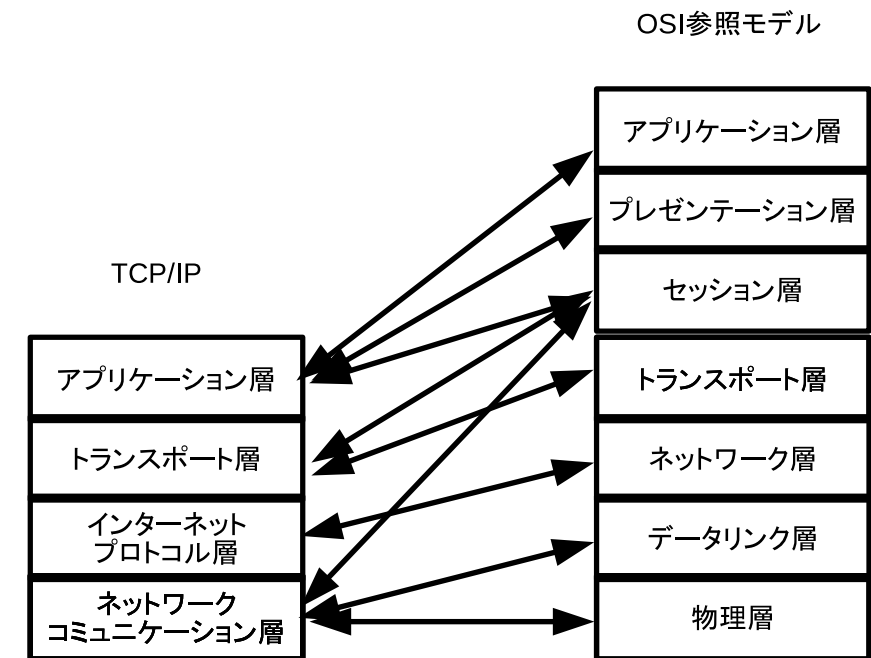
1.7.1 OSI 参照モデルと TCP/IP のマッピング

TCP/IP と OSI 参照モデルをにマッピングして説明することが多い。それは、機能がおおまかにマッピングできるからである。

たとえば、TCP/IP のネットワークコミュニケーション層は、OSI 参照モデルでは、物理層とデータリンク層をあわせたものに相当するとされる。同様に、OSI 参照モデルのネットワーク層はインターネット層、OSI 参照モデルのトランスポート層は TCP/IP でもトランスポート層に相当する。また、OSI 参照モデルのそこから上の層は、TCP/IP のアプリケーション層に相当する。名前は同じだが、OSI 参照モデルとインターネットプロトコルスイートのアプリケーション層とが、直接対応しているわけではない。

^{*6} レイヤーが七つあることから、OSI7 階層モデルと記載されることもある

部分的に上位・下位レイヤーの機能を含むところもあり、TCP/IPとOSI参照モデルはそのままマッピングすることはできない



『htbp』

図 1.5 TCP/IP と OSI 参照モデルの対応

このように、OSI 参照モデルとインターネットプロトコルスイートは、一対一でマッピングすることができない。それが、大まかにマッピングできるという表現になった理由である。また、TCP/IP のある層の機能が、OSI 参照モデルでは複数の層にまたがっていたり、OSI 参照モデルのある層の機能が、TCP/IP では対応するとされている層には実装されていなかったりするという事情もある。では、そのような例を、いくつか挙げてみよう。

前述の通り、インターネットプロトコルスイートのネットワークコミュニケーション層は OSI 参照モデルでは物理層とデータリンク層になる。また、OSI 参照モデルのトランスポート層に対応するインターネットプロトコルスイートのトランスポート層の機能の一部は、OSI 参照モデルのセッション層にも含まれている。

また、インターネットプロトコルスイートでは、回線の全二重、半二重の判別が必要であれば、ネットワークコミュニケーション層以下で実装される。だが、OSI 参照モデルで

は回線の全二重、半二重の判別とそれぞれに対応した通信は、セッション層で実現することになっている。

OSI 参照モデルのレイヤ 3 であるネットワーク層では、確実な通信を行うためのコネクション型通信と、そうでない場合のコネクションレス通信の両方が規定されている。だが、TCP/IP のインターネットプロトコル層は、コネクションレス通信のプロトコルである。コネクション型のプロトコルは、その上位のトランスポート層が担当する。

OSI 参照モデルでは、トランスポート層がエラー訂正の機能を持つとされている。だが、TCP/IP には、エラー訂正の機能はない。データの破損を検出する機能は存在する。だが、その場合は単にデータを破棄する。そして、確実に届かなければならないデータであれば、相手が再送することを期待する。インターネットプロトコルスイートの通信で、エラー訂正が必要であれば、アプリケーション層が送出するデータにエラー訂正のための冗長な情報を含めておくか、ネットワークアクセス層で同様に実現するかとなる。ネットワークアクセス層でエラー訂正機能を実装する必要があるのは、たとえば、衛星通信のように、エラー訂正の情報でデータが大きくなる以上に、エラーによる再送のコストが高くなる場合に行われる。

1.7.2 OSI 参照モデルとネットワーク機器の機能

現在では、OSI 参照モデルは、その概念のみが残っており、OSI 参照モデルをリファレンスにした実装は存在しない。現在では、OSI 参照モデルは、ネットワーク機器の機能を表すための用語として、レイヤ番号のみが使われている。この、レイヤ番号を用いた表見は、ネットワークやインフラのエンジニアが用いることが多い。

先ほど説明したように、TCP/IP と OSI 参照モデルのレイヤーは一対一対応しているわけではない。だが、ネットワーク機器の機能を表現するには、ネットワークコミュニケーション層機器ではなく L(レイヤ) 2 機器というような言い方をする。同様に、インターネットプロトコル層機器でなく L3 機器、というように、インターネットプロトコルスイートの各層と「おおむね対応している」OSI 参照モデルの層のレイヤ番号を用いる。

ネットワーク機器のスイッチング HUB は、インターネットプロトコルスイートでは、ネットワークコミュニケーション層に対応する機器である。そのため、ネットワークコミュニケーション層におおむね対応する OSI のレイヤ番号を用いて、L2 スイッチ、あるいはレイヤ 2 スイッチと呼ぶ。SW-HUB が用いるネットワークコミュニケーション層での機能は、OSI 参照モデルでは下から 2 層目、データリンク層に相当する機能の機器であるためだ。

唯一の例外は L1 で、レイヤー 1 でなく、物理層と呼ばれることが多い。たとえば、ケーブルの断線は物理層ので発生した障害である。だが、光ファイバの中の信号を、光学的に多重化する PON や、光とイーサネットのメディアコンバータなど、インターネットプロトコル層に直接サービスしない機器は、L1 機器とよばれることがある。

また、TCP/IP でインターネットプロトコル層の機器であるルータは、OSI 参照モデルではおおむねネットワーク層に相当するとされる。そのため、ネットワーク層のレイヤ番号 3 を用いて、L3 機器と表現する。また、L3 スイッチという機器は、見た目こそスイッチング HUB であるが、インターネットプロトコル層の機能を持っていることを意味する。

1.7.3 高レイヤ機器

ネットワーク機器には、L4 機器、L5 機器、L7 機器と呼ばれる機器もある。これらの機器は、L2 や L3 の機器と対比して、高レイヤ機器と呼ばれることがある。

L4 機器

L4 の機器はトランスポート層に対応する。L5 の機器は、セッション層、L7 の機器は、アプリケーション層である。これらの高レイヤ機器は何を行うための機器なのだろうか。

L4 の機器は、インターネットプロトコル層の IP アドレスと、レイヤ 4 におおむね対応するトランスポート層のポート番号を識別する。それを利用することで、通信がどのサーバのどのアプリケーションにタイして行われているかを判別する。IP アドレスとポート番号の情報を利用して、通信を許可するかどうかを判別するファイアウォールが、代表的な L4 機器である。

L5 機器

L5 は、TCP/IP と対比するとき、トランスポート層とアプリケーション層の間という位置づけになる。そのため、トランスポート層とアプリケーション層の間でストリームを暗号化する SSL/TLS を、L5 に対応づけることがある。^{*7}

L5 に割り当てた SSL/TLS を、SSL アクセラレータや SSL-VPN として提供する機器が、L5 機器と言われることがある。

L7 機器

L7 機器は、レイヤ 7 におおむね対応するアプリケーション層のプロトコルとその状態を判別して、通信を制御する。たとえば Web サーバが複数ある環境で、どれかのサーバに接続するという方法で負荷を分散しつつ、特定のサーバに接続し続ける必要がある、HTTP のセッション維持を行う、ロードバランサを実現することが可能となる。

また、アプリケーションプロトコルのデータを監査してウィルスやマルウェアのチェックを行うスクリーニング用のプロキシを L7 機器と呼ぶこともある。さらに、Web サーバへのアクセスをスクリーニングするソフトウェアと s ちえ実装され WAF(Web Application Firewall) も、ネットワークから見れば、L7 相当となる。

いもうとコラム OSI 参照モデルの L8、L9、L10

パイソンのスケッチ、スペイン宗教裁判^{*8}ではないのですが、七階層ある OSI 参照モデルには、レイヤ 8 からレイヤ 10 までの上位 3 層が存在します。スペイン宗教裁判では、罪が三つしかないはずなのに、異端審問官が四つ数え上げる、というのが笑うところなのですが、OSI 参照モデルは、七層のはずが実際には十層のレイヤーで構成されています。その構成は色々な説があります。たとえば、L8 が経済層、L9 が政治層、L10 が宗教層としたり、L8 がユーザ層、L9 が経済層、L10 が宗教層としたりしています。

^{*7} SSL/TLS は、アプリケーションのストリームを暗号化するという動作から、どうしても OSI 参照モデルに対応づけたいなら、L6 に置くべきと考えられる。

^{*8} Nobody expects the Spanish Inquisition!

レイヤ番号	レイヤ名
L10	宗教層
L9	政治層
L8	経済層

表 1.4 OSI 参照モデルの L8 以上

種を明かせば、この L8 から上のレイヤーは、ジョークの類です。ネットワークの問題は、往々にして、ネットワークの外で、技術的ではない理由によって発生します。事件はデータセンタで起きても障害は会議室で起こる、ということえす。それは経済的事情で機器やそのサポートが買えない、社内政治の事情でベンダや代理店が変更される、上司がクラウドを使わなければ安全、という宗教に捕らわれている、など、現場ではどうにもならない理由で発生する障害を皮肉っています。つまり、L8 以上の問題とは、技術的に解決できるはずのことが解決できない状況です。