

[スキルアップに効く技術と知識]

# インフラエンジニアの 教科書

*The textbook of the  
infrastructure engineer*

Yutaka Sano  
佐野 裕



**現場で役立つ本当の「力」をつける!**

数年、実務経験を積んだ人が次のステップに進むために知っておきたい技術や知識をわかりやすく解説します。日々のルーティンワークからは学びにくい原理原則・ノウハウが満載!

**視野を広げ成長を目指す人に贈る渾身の続編。  
前作と併せて読みたいインフラエンジニアのバイブル!**

[スキルアップに効く技術と知識]

# インフラエンジニアの 教科書

*The textbook of the  
infrastructure engineer*

Yutaka Sano  
佐野 裕



## ■権利について

- 本書に記述されている社名・製品名などは、一般に各社の商標または登録商標です。
- 本書では™、©、®は割愛しています。

## ■本書の内容について

- 本書は著者・編集者が実際に操作した結果を慎重に検討し、著述・編集しています。ただし、本書の記述内容に関わる運用結果にまつわるあらゆる損害・障害につきましては、責任を負いませんのであらかじめご了承ください。
- 本書の内容は2016年7月現在の情報を基に記述しています。

### ●本書の内容についてのお問い合わせについて

この度はC&R研究所の書籍をお買いあげいただきましてありがとうございます。本書の内容に関するお問い合わせは、「書名」「該当するページ番号」「返信先」を必ず明記の上、C&R研究所のホームページ(<http://www.c-r.com/>)の右上の「お問い合わせ」をクリックし、専用フォームから送りいただくか、FAXまたは郵送で次の宛先までお送りください。お電話でのお問い合わせや本書の内容とは直接的に関係のない事柄に関するご質問にはお答えできませんので、あらかじめご了承ください。

〒950-3122 新潟県新潟市北区西名目所4083-6 株式会社 C&R研究所 編集部  
FAX 025-258-2801  
「インフラエンジニアの教科書2 スキルアップに効く技術と知識」サポート係



## はじめに

「日々インフラエンジニアとして頑張っているけど、自分にはどうも何かが足りない気がする」と思ったことがあるかもしれません。OJT(On the Job Training)による実地訓練を通してインフラエンジニアとしての経験を積んでいるうちにさまざまなことができるようになったとしても、インフラエンジニアに必要な基本的な知識をしっかり押さえないまま経験だけが増えていくと、「一応いろいろなことができるようになったけれども、詳しく知らないし、他の人に説明できない」といったことが増えてくるようになります。

そこで本書では、日常業務でよく耳にするけれども、OJTや日常業務からでは身に付けにくいインフラエンジニアの必須知識を中心に内容を盛り込みました。

前作『インフラエンジニアの教科書』では、新卒の新入社員がまずは知っておくべき用語を一通り記しましたが、本作『インフラエンジニアの教科書2 スキルアップに効く技術と知識』では、実務経験を数年積んだ人が次のステップに進むために知っておくべき技術や用語をなるべくわかりやすく、まんべんなく記すと共に、他の本では省かれるようなちょっとした疑問点や、実は重要だけでも意外と気付きづらい技術的なポイントなどについても漏れなく記載しました。本書を読んでから他の技術書を読むときと違って見えるようになるはずです。

また、後半では世界規模のインターネットサービス運営についての内容にも触れました。近年多くのIT企業が世界に向けて挑戦が続いていますが、その際意識すべきことや注意すべきことなどをまとめてみました。

業務の役に立てていただければ幸いです。

2016年7月

佐野 裕





# 目次 contents

## ● CHAPTER-01

---

### プロトコル

001	プロトコルの基本 .....	14
●	インターネットで用いられるプロトコル群 .....	15
●	プロトコルのアップデート .....	17
●	独自プロトコル .....	18
002	TCP/IP .....	20
●	TCP .....	20
●	UDP .....	22
●	IP .....	23
●	ICMP .....	24
003	プロトコルの通信パターン .....	27
●	TCP上で1つのコネクションを用いる一般的なプロトコル .....	27
●	TCP上で2つのコネクションを用いるプロトコル .....	27
●	UDPで送りっぱなしのプロトコル .....	28
●	UDPで双方向通信を行うプロトコル .....	29

## ● CHAPTER-02

---

### OS

004	プロセスとスレッド .....	34
●	早わかり プロセスとスレッド .....	34
●	プロセス .....	35
●	ファイル記述子 .....	36
●	プロセスの親子関係と新しいプロセスの生成 .....	37
●	プロセス間通信 .....	39
●	プロセスの状態遷移 .....	39
●	シグナル .....	41
●	プロセスの終了 .....	42
●	スレッド(軽量プロセス) .....	43
005	OSによるプロセス処理の仕組み .....	46
●	タイムスライス .....	46
●	コンテキストスイッチ .....	46
●	Copy On Write (CoW) .....	48
●	スケジューラとディスパッチャー .....	48

● アフィニティー(Affinity) .....	50
<b>□□6 インフラエンジニアのプロセス管理 .....</b>	<b>53</b>
● シングルプロセスかつシングルスレッドの処理速度向上 .....	53
● CPUコアを無駄なく使う .....	53
● コンテキストスイッチの発生を抑える .....	53
● 特定プロセスの暴走 .....	53
● 割り込み処理 .....	54
<b>□□7 メモリ管理 .....</b>	<b>55</b>
● 早わかり メモリ管理 .....	55
● 物理メモリ空間と仮想メモリ空間 .....	55
● スワップ .....	57
● ページング .....	57
● NUMA(非対称メモリアクセス) .....	58
<b>□□8 OSによるメモリ処理の仕組み .....</b>	<b>62</b>
● ページキャッシュ .....	62
<b>□□9 インフラエンジニアのメモリ管理 .....</b>	<b>63</b>
● スワップの発生 .....	63
● 空き物理メモリが足りなく見える .....	64
● バックアップとページキャッシュ .....	64
<b>□10 ファイル管理 .....</b>	<b>66</b>
● ファイルのメタデータ .....	66
● ファイルシステム .....	67
● 分散ファイルシステム .....	68
<b>□11 Linux/UNIXのファイル管理 .....</b>	<b>69</b>
● ファイルのメタデータ .....	69
● ディレクトリエントリ .....	69
● 使用中ファイルの削除 .....	70
● シンボリックリンクとハードリンク .....	70
● バインドマウント .....	72
● ファイルシステムのマウント .....	73
<b>□12 Windowsのファイル管理 .....</b>	<b>76</b>
● Windowsでのメタデータ .....	76
● ファイル名の文字コード .....	77
● シンボリックリンクとジャンクションとハードリンク .....	78
<b>□13 インフラエンジニアのファイル管理 .....</b>	<b>80</b>
● ファイルを消しても空き容量が増えない .....	80

● ディスク容量に空きがあるのにファイルを生成できない	80
● ログ出力でディスク容量を使い果たした	81
● 想定よりもハードディスクの空き容量が少ない	82

## ● CHAPTER-03

### ネットワーク

□ 1 4 ネットワークとネットワーク機器	84
● 集約と切り替え	84
● ルータ	86
● スイッチ	88
● 経路情報	93
□ 1 5 ネットワーク機器の主要機能	97
● VLAN	97
● ジャンボフレーム	100
● NAT(Network Address Translation)	101
● ACL	103
● SSLオフロード	103
□ 1 6 インフラエンジニアのネットワーク管理	104
● ファームウェアのアップデート	104
● フェイルオーバーとスイッチオーバー	104
● 主に監視するのはdiscard(packet loss)とunbind	105
● IPアドレス計画	106
● トポロジー制約	107
● ベンダー統一 vs. マルチベンダー	107
● ネットワーク機器の構造はサーバと同じ	108
● サーバをネットワーク機器のように使うこともできる	108
● ネットワーク経由でのファイルコピーが遅い	109

## ● CHAPTER-04

### データベース

□ 1 7 データベース	112
● RDBMSの種類	112
● LAMPとLAPP	115
● RDBMSのプロセスアーキテクチャ	115
● データベースの選定	117

<b>□ 1 8</b>	<b>データベースの用語と主要機能</b> .....	118
●	SQL .....	118
●	トランザクションとコミット／ロールバック .....	118
●	トランザクション分離レベル .....	120
●	ロックとデッドロック .....	122
●	インデックスとフルスキャン .....	123
●	ストアードプロシージャ .....	124
●	コネクションプール .....	125
<b>□ 1 9</b>	<b>データベースの高度な機能</b> .....	126
●	データ保全とレスポンス向上のための機能 .....	126
●	スケールアウトのための機能 .....	127
●	高速処理のための機能 .....	129
●	可用性向上のための機能 .....	130
<b>□ 2 0</b>	<b>インフラエンジニアとデータベース</b> .....	134
●	データベースの日常監視 .....	134
●	SQL文の品質向上対策 .....	135
●	ディスク/I/O負荷対策 .....	135
●	MySQLマスターサーバのSPOF(単一障害点)解消 .....	135
●	バックアップとリカバリ .....	138
●	RDBMSのバージョンアップ .....	138
●	RDBMS起因でない原因不明のパフォーマンス低下対応 .....	139
●	RAIDコントローラーのライトバック・ライトスルー設定 .....	140

## ● CHAPTER-05

# WEBのサーバサイド開発言語

<b>□ 2 1</b>	<b>オープンソース系開発言語</b> .....	144
●	WEBサーバと開発言語との連携 .....	144
●	データベースとの接続 .....	147
●	フレームワーク .....	147
●	ライブラリ/リポジトリ .....	148
●	PHPアクセラレータ .....	149
<b>□ 2 2</b>	<b>Java</b> .....	150
●	早わかり Java .....	150
●	Javaプログラムのコンパイルから実行までの流れ .....	151
●	Java SEとJava EE .....	152
●	WEBアプリケーションサーバ .....	153

● JavaでWEBアプリケーションを動かすために必要なもの .....	155
● サンプルとしてのJavaプログラムの実行 .....	157
● JSPとしてのJavaプログラムの実行 .....	157
● ヒープ領域とメモリ管理 .....	158
● ガベージコレクション(GC) .....	159
● その他 .....	160

## ● CHAPTER-06

### 共通鍵暗号方式と公開鍵暗号方式

□ 2.3 共通鍵暗号方式と公開鍵暗号方式 .....	162
● 共有鍵暗号方式 .....	162
● 公開鍵暗号方式 .....	162
● さまざまな公開鍵暗号方式のアルゴリズム .....	163
● RSA暗号 .....	164
□ 2.4 SSL通信とSSLサーバ証明書 .....	165
● 早わかり SSL通信 .....	166
● SSL通信の用途 .....	166
● SSL通信の仕組み .....	167
● SSLサーバ証明書 .....	168
● ワイルドカード証明書 .....	169
● EV SSLサーバ証明書 .....	169
● 認証局(CA) .....	169
● 中間CA証明書 .....	170
● クロスルート証明書 .....	172

## ● CHAPTER-07

### 障害対策と障害対応

□ 2.5 障害対応の心構え .....	176
● 障害対応の心構え .....	176
□ 2.6 スピード .....	177
● サーバ状態の収集 .....	177
● システム構成や開発担当者の事前把握 .....	180
□ 2.7 原因分析力 .....	181
□ 2.8 情報収集力 .....	183

## ● CHAPTER-08

### よく知られたセキュリティ攻撃

□ 2.9	有名なセキュリティ攻撃手法 .....	186
●	ポータスキャン .....	186
●	バックドア .....	189
●	ブルートフォースアタック .....	190
●	ソーシャルハッキング .....	191
●	DoS攻撃とDDoS攻撃 .....	191
□ 3.0	設定ミスで生じるセキュリティホール .....	193
●	メールサーバでのオープンリレー .....	193
●	DNSオープンリゾルバ .....	194
●	DNS amp(増幅)攻撃 .....	196
●	DNSキャッシュポイズニング攻撃 .....	197
□ 3.1	開発時に生じるセキュリティホール .....	198
●	バッファオーバーフロー攻撃 .....	198
●	クロスサイトスクリプティング(XSS)攻撃 .....	199
●	SQLインジェクション攻撃 .....	199

## ● CHAPTER-09

### インターネットの運用と発展をつかさどる組織や団体

□ 3.2	インターネットの標準化やポリシーを取りまとめる組織 .....	202
●	ISOC(インターネット協会) .....	202
●	IAB(インターネットアーキテクチャ委員会) .....	202
●	IETFとIESG .....	204
●	IRTFとIRSG .....	204
□ 3.3	インターネット資源を管理する団体 .....	205
●	ICANNとIANA .....	205
●	ドメイン名の割り当てと管理 .....	206
●	IPアドレスと自律システム(AS)番号の割り当てと管理 .....	207
●	JPNIC .....	209
●	JPRS .....	209

## ● CHAPTER-10

# RFCの読み方と作られ方

□ 3.4	RFCとは	212
●	RFCの歴史的経緯	212
●	RFC流標準化	213
●	RFCのカテゴリ	213
●	標準化過程(Standards Track)	214
●	実験的(Experimental)	214
●	歴史的(Historic)	214
●	情報(Informational)	214
●	現状で最良の慣行(BCP: Best Current Practice)	214
□ 3.5	RFCの見方	216
●	RFCの検索	216
●	RFC文書の構成	218
□ 3.6	RFCができるまで	220
●	RFC掲載までのプロセス	220

## ● CHAPTER-11

# 世界規模のインターネットサービス運営

□ 3.7	遠距離を前提とした通信対策	224
●	レイテンシー問題	224
●	TCPウィンドウサイズを大きく設定する	225
●	データを圧縮して送信する	227
●	プロトコルを変える	227
□ 3.8	クライアントとサーバの距離を短くする	228
●	世界中にPOPを置く	228
●	クラウドサービスの海外リージョンをPOPとして使う	228
●	主要ISPやIXとピアリングする	229
●	CDNサービスを利用する	229
□ 3.9	グローバルサービスでのデータ保存	231
●	1カ所にデータを集約する構成	231
●	複数拠点ごとにデータを保存できるようにする構成	232
●	同じタイミングで同じデータが異なる拠点で更新された場合の対応	233
●	ユーザの属する地域を決める	234



□ 4 0	グローバルコミュニケーション	235
●	言語と文化の問題	235
●	社内言語	237
●	社内システム	237
●	ITインフラポリシーの統一	239
●	時差や祭日の問題	240
●	海外出張	241

## ● CHAPTER-12

# インフラエンジニアとして目指す方向

□ 4 1	インフラエンジニアとして目指す方向	244
●	バランスの良い設計/構築/運用経験を積む	244
●	体系化された方法論として学びにくい現状	244
●	技術の世界はきりが無い	244
●	技術力を深める	245
●	スピードを速める	245
●	言い切る力をつける	246
●	クラウド系エンジニア vs. オンプレミス系インフラ	246
□ 4 2	インフラエンジニアのレベルアップ	247
●	オペレーター	247
●	システム導入が中心のエンジニア	247
●	障害原因分析を行えるエンジニア	248
●	特定分野でトップレベルに詳しいエンジニア	248

## ● COLUMN

■ ポステルの法則 .....	18
■ コネクションとセッションの違い .....	26
■ pingやtrace routeを遮断する方法 .....	26
■ ウェルノウンポートとエフェメラルポート .....	30
■ ファイアウォールの動的ポートフィルタリング機能 .....	31
■ cron設定でよく見る記述の意味 .....	45
■ CPUコア数とIntel HTのスレッド数との関係 .....	51
■ デーモンやサービスの定義 .....	52
■ スワップとページングの違い .....	60
■ 仮想サーバのCPU割り当て .....	61
■ カレントディレクトリと親ディレクトリへのリンク .....	75
■ ネットワークが難しいところ .....	96
■ MySQLでシャーディングを行う方法 .....	133
■ リストア(復元)とリカバリ(復旧) .....	142
■ クラウド上でのデータベースの利用 .....	142
■ 修理交換後は元々の設定が引き継がれない場合がある .....	142
■ 文字化け問題 .....	149
■ Scala言語 .....	160
■ SSLとTLS .....	173
■ SSL証明書がなくてもSSL通信ができる理由 .....	173
■ 無料で利用できるSSL証明書発行機関創設 .....	174
■ 良い本の選び方 .....	184
■ DNSは同じ意味を指す用語がたくさん .....	197
■ SPF vs. Sender ID .....	215
■ RFC掲載の参考サイト .....	221
■ 企画や開発側での工夫 .....	230
■ LINEの自動通訳機能 .....	242
■ 標準化のし過ぎは技術的弱体を招くことがある .....	249
■ 就職難は関係ない .....	249
● 索引 .....	252



# CHAPTER 01 プロトコル

## 》》》 本章の概要

プロトコルとは通信規約のことです。ネットワーク上で通信を行うためには通信する同士が同じプロトコルを用いて通信することが大前提となります。

インターネット上には世界中の人がPC、スマートフォン、ポータブルゲーム機、TVなど、さまざまな機器を接続します。国もベンダーもハードウェアもOSも異なる機器同士が何の問題もなく同じネットワーク上で通信しあえるのはすべての機器が同じプロトコルを用いているからに他なりません。

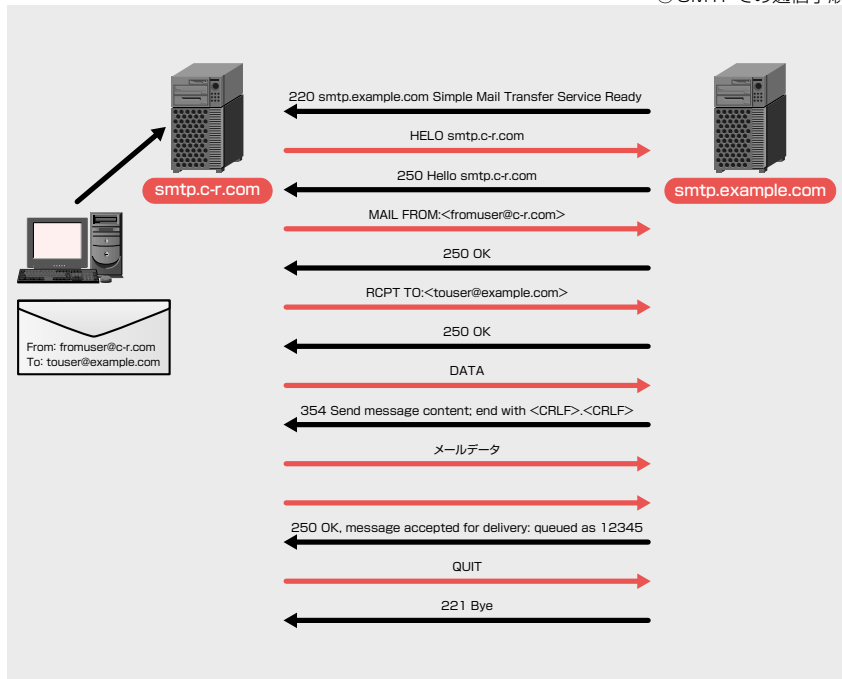
プロトコルの概念をよく理解しておかないとインフラを構築・運営・障害対応する際に支障が出る場面がありますので、インフラエンジニアの基本知識としてきっちり押さえていきましょう。

# プロトコルの基本

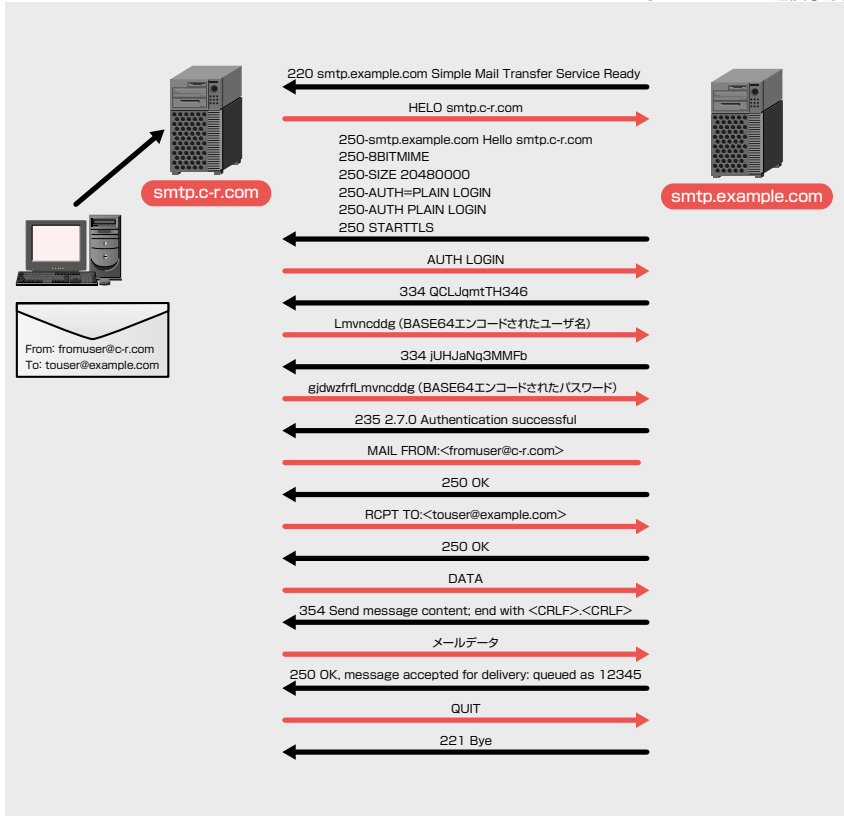
プロトコルとは、コンピュータやネットワーク機器などの電子機器が相互に通信する際に用いられる通信規約のことです。世界中のどの機器やソフトウェア同士でも、同一プロトコルがサポートされていれば相互で通信することができます。

たとえば、メールを送信するときに用いられるSMTP(Simple Mail Transfer Protocol)やESMTP(Extended SMTP)というプロトコルでは図のような手順で通信を行うことができます。このように、相互通信はプロトコルが定義されているからこそ実現します。

●SMTPでの通信手順



## ● ESMTPでの通信手順



なお、SMTPはRFCと呼ばれる文書の7504番、ESMTPは6531番で定義されています。

- SMTPのRFC

URL <http://www.rfc-editor.org/info/rfc7504>

- ESMTPのRFC

URL <http://www.rfc-editor.org/info/rfc6531>

## ● インターネットで用いられるプロトコル群

通信を行うときには様々なプロトコルを組み合わせで用います。プロトコルのまとまりをプロトコル群と呼びます。

インターネットに関するプロトコル群としては、TCP/IP参照モデルと呼ばれる4階層のモデルと、OSI参照モデルと呼ばれる7階層のモデルがあります。

## ◆ TCP/IP参照モデル

TCP/IP参照モデルとは、IETFにより管理されているインターネットで使われる4階層のプロトコル群のことです。インターネット・プロトコル・スイートもしくはTCP/IP4階層モデルなどとも呼ばれます。

● TCP/IP参照モデル

階層	階層名	代表的なプロトコル	説明
第4層	アプリケーション層	HTTP、SMTP、DNS、FTP、SSH、TELNETなど	ユーザに直接サービスを提供するユーザプロトコル(TELNET、FTP、SMTPなど)と、共通的なシステム機能を提供するサポートプロトコル(SNMP、BOOTP、RARPなど)の2つのカテゴリで使われる
第3層	トランスポート層	TCP、UDP	アプリケーションのためにエンドツーエンドの通信サービスを提供する
第2層	インターネット層	IP、ICMP、IGMP	ICMPは制御プロトコルであり、IPの上の層に位置するが、インターネット層に属する
第1層	リンク層	ARP	ネットワークインタフェースのためのプロトコル

## ◆ OSI参照モデル

OSI参照モデルとは、国際標準化機構(ISO)によって1977年から1984年にかけて策定された7階層で構成されたプロトコルのモデルです。

ネットワーク機器で使われるL2・L3・L4・L7スイッチがOSI参照モデルの階層に対応しています。LxスイッチのLはLayer(層)を意味します。たとえば、第2層データリンク層で通信を行うスイッチがL2スイッチ、第3層ネット

階層	階層名	説明	
第7層	アプリケーション層	具体的な通信サービス(たとえばファイル・メールの転送、遠隔データベースアクセスなど)を提供。HTTPやFTPなどの通信サービス	
第6層	プレゼンテーション層	データの表現方法(たとえばEBCDICコードのテキストファイルをASCIIコードのファイルへ変換する)	
第5層	セッション層	通信プログラム間の通信の開始から終了までの手順(接続が途切れた場合、接続の回復を試みる)	
第4層	トランスポート層	ネットワークの端から端までの通信管理(エラー訂正、再送制御など)	
第3層	ネットワーク層	ネットワークにおける通信経路の選択(ルーティング)。データ中継	
第2層	データリンク層	直接的(隣接的)に接続されている通信機器間の信号の受け渡し	
第1層	物理層	物理的な接続。コネクタのピンの数、コネクタ形状の規定など。銅線・光ファイバ間の電気信号の変換など	

ワーク層で通信を行うスイッチがL3スイッチとなります。

ところで、TCP/IP参照モデルとOSI参照モデルは似ている部分が多々ありますが、TCP/IP参照モデルを定義しているIETFから「OSI参照モデルに準拠することは意図しない」というアナウンスがされています。

● プロトコルのアップデート

プロトコルに不備がある場合や時代の変化でプロトコルの機能拡張が必要な場合などに、プロトコルがアップデートされることがあります。

プロトコルがアップデートされると、新しいプロトコルに対応したバージョンのソフトウェアやファームウェアがリリースされる場合もありますが、古いソフトウェアやハードウェアの場合はすでにサポートが終了していて最新版がリリースされないこともよくあります。

このように、世界中に新旧のプロトコルが混在する状況はプロトコル設計者がコントロールできる領域でないので、プロトコルをアップデートする場合は新旧両方が混在することを前提として設計されます。

1

プロトコル

2

3

4

5

6

7

8

9

10

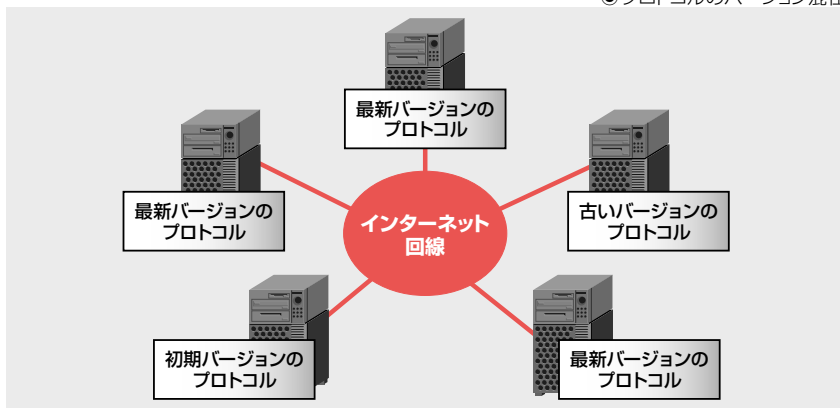
11

12

●OSI参照モデル

	スイッチング先決定時に用いる情報	対応するネットワーク機器
	HTTPのURL	L7スイッチ／ロードバランサー
	IPアドレス + TCP/UDPポート番号	L4スイッチ／ロードバランサー
	IPアドレス	L3スイッチ／ルータ
	MACアドレス	L2スイッチ／スイッチングハブ、ブリッジ
		L1スイッチ／リピータハブ





## ● 独自プロトコル

自分で開発する独自アプリケーションでは独自プロトコルを用いることができます。たとえば、オンラインゲームを開発する場合はクライアント側とゲームサーバ側で通信が必要となりますが、このとき、ゲームサーバを、HTTP上で動く独自プロトコルで構成したり、もしくはパフォーマンスを重視したいのでTCP上で動くHTTPよりオーバーヘッドの少ないスリムなプロトコルで構成したりといったことができます。独自プロトコルですので自分が好きなようにプロトコルを設計できます。

ただ、さまざまな状況に対応できるプロトコルの設計は容易ではなく、実際にプロトコルを設計してみると既存のプロトコルが本当によくできていることを実感することになるでしょう。



## COLUMN

### ポステルの法則

「送信するものに関しては厳密に、受信するものに関しては寛容に」という、ポステルの法則と呼ばれる通信における設計原則があります。

これはRFCで定義されている規約に明確に違反しているものはエラーとすべきですが、必ずしも違反といえないものについては受け入れましょうという考えです。

特にメール関連ではポステルの法則の恩恵をよく受けることとなります。メール関連ソフトウェアにはさまざまな実装が存在します。その中に

は実装に癖があるメールサーバやメールクライアントが多々ありますが、それらに対して、RFC違反だからとエラーを返す代わりに過ちはできるだけ許容しましょう、という実装がポステルの法則に則った理想的な実装です。

ポステルの法則は、アメリカのコンピュータ科学者のジョン・ポステル博士(Jonathan Bruce Postel, 1943年8月6日 - 1998年10月16日)に由来しています。ポステル博士は、1974年にカリフォルニア大学ロサンゼルス校(UCLA)からコンピュータサイエンスの博士号を授与され、UCLA在学中から初期のARPANetに関わっていました。

1977年から南カリフォルニア大学の情報科学研究所でインターネットの発展と標準化に多大な貢献をし、「インターネットの神」と呼ばれ、TCP/IP、SMTP、DNS、FTP、Telnetの開発に携り、RFCの編集者のほかに、IANAの発起人として知られています。

ポステル博士は55歳で死去しましたが、その際にRFCに追悼文が公表されています。

- RFC 2468 - I REMEMBER IANA(ヴィントン・サーフによる追悼文)

**URL** <https://tools.ietf.org/html/rfc2468>

- RFC 2441 - Working with Jon(ダニー・コーエンによる追悼文)

**URL** <https://tools.ietf.org/html/rfc2441>

● ジョン・ポステル博士

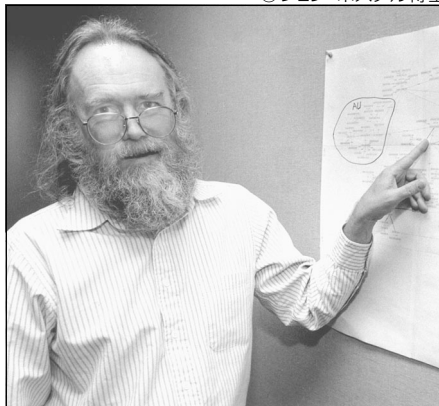


Photo by Irene Fertik, USC News Service. Copyright 1994, USC

# TCP/IP

TCP/IPとはインターネット上で標準的に用いられるプロトコル群の総称を指します。ここでは特に重要なTCP、UDP、IPそしてICMPのプロトコルについて記します。

## TCP

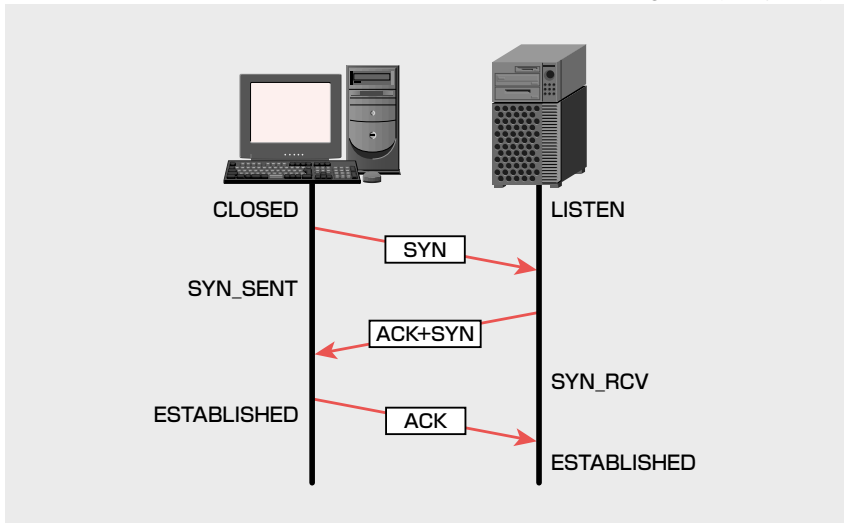
TCP(Transmission Control Protocol)はTCP/IP参照モデルの第3層であるトランスポート層に属するコネクション型プロトコルです。WEB、メール、FTPなどといったクライアントとサーバの間で双方向通信を行うサービスで用いられます。

TCPには、データがきちんと相手に届いたかどうか確認する仕組みがあります。具体的には次のような機能があります。

- パケットを送信する際、都度シーケンスナンバーを更新して送り、受け取った相手方は送信元に受信確認パケットを返す。
- データが相手に届かないか、もしくは送信エラーが発生した場合、それを検知してデータの再送を行う。
- 送信したパケットの順序と受信側で受け取るときの順序が異なった場合に順序を入れ替える。
- データが重複して送られた場合、除去を行う。

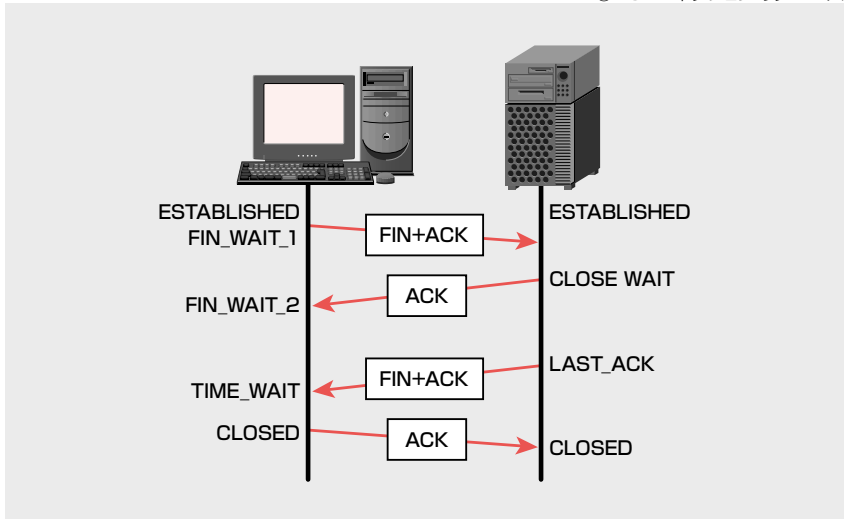
クライアントからサーバに対してTCP接続要求が発生すると、3ウェイハンドシェイクを行ってクライアントとサーバの間でSYNとACKを投げ合うことでコネクションが確立されます。SYNはSynchronize、ACKはAcknowledgeの略で、それぞれデータ送受信の起点と受信確認の意味を持ちます。

## ●3ウェイハンドシェイク



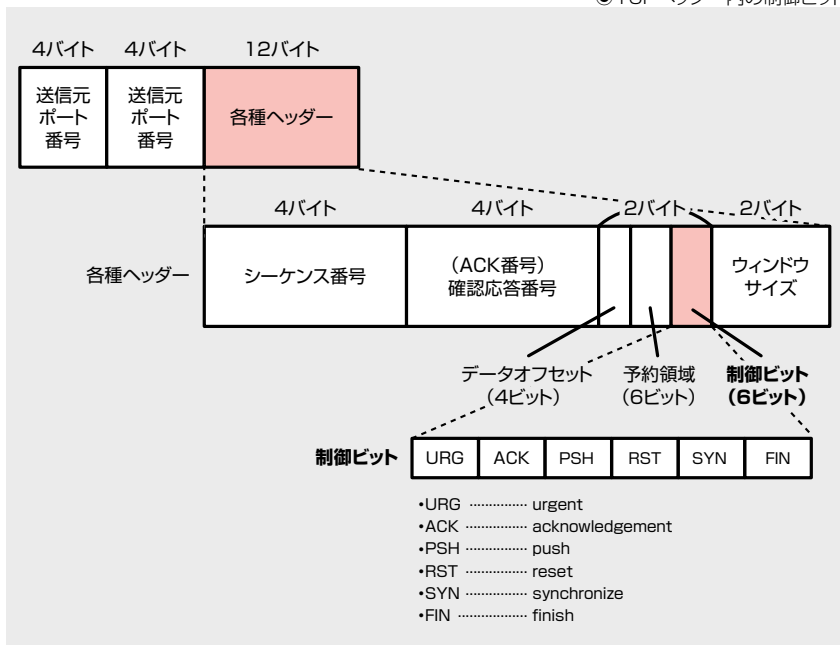
通信を終えてコネクションを閉じる場合は、切断要求としてFIN+ACKパケットを相手に送ります。すると相手から応答としてACKパケットが返ってきます。その後しばらくして相手からも切断要求としてFIN+ACKパケットが送られてくるので、こちらが応答のACKパケットを送り返すことでコネクションが終了します。

## ●TCPコネクションのクローズ



SYN、ACK、FINなどはIPパケットのTCPヘッダー内にある制御ビットに含まれます。

● TCPヘッダー内の制御ビット



## UDP

UDP(User Datagram Protocol)はTCP同様、TCP/IP参照モデルの第3層であるトランスポート層に属する、コネクションレス型プロトコルです。DNS(名前解決)やNTP(時刻合わせ)といった速い応答性が重要なサービスや、音声通話や動画など一部の情報が欠けても問題のないアプリケーションなどで用いられます。

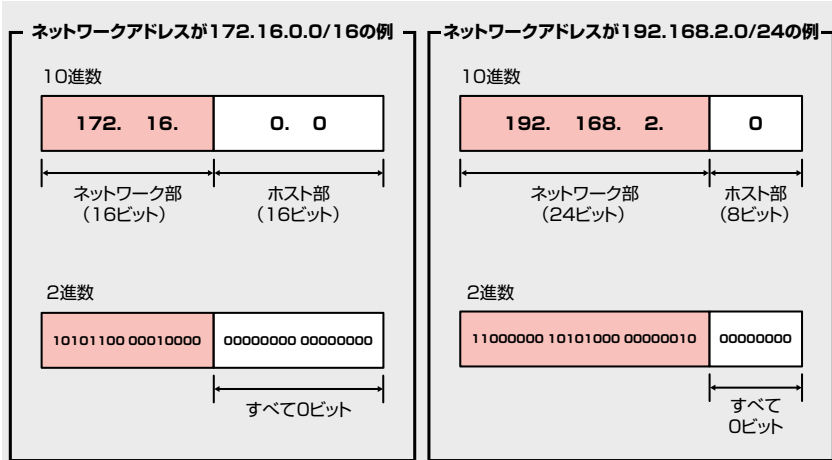
UDPは低品質で高速なプロトコルといわれます。UDPでは送信元は受信先に対してコネクションを張らずに、いきなり一方的にデータを送りつけます。一方的に送りつけるわけなので、送信先がその情報を受け取る保証はなく、またパケットが正常に送信先に届いても、送信先から情報を受け取ったという返答があるわけではありません。

## IP

IP(Internet Protocol)とは、TCP/IP参照モデルの第2層であるインターネット層に属するプロトコルです。IPアドレスのIPはIP(Internet Protocol)のことを指しています。

IPv4ではIPアドレスをネットワーク部とホスト部に分けます。ネットワーク部は各ネットワークに対して固有に割り振られます。一方、ホスト部は各組織で自由に割り振ることができます。

●IPアドレスのネットワーク部とホスト部



ネットワーク部とホスト部の位置を区別するためにネットマスクも併記します。ネットマスクは「255.255.255.0」のような書式で表すか、もしくは「/24」のように「255.255.255.0」を2進数で表した「11111111 11111111 111 11111 00000000」の1の個数を記す書式で表します。

## ICMP

ICMP(Internet Control Message Protocol)はTCP/IPが動作する上で補助的な役割を果たすためのプロトコルで、IPの制御メッセージやエラー通知を転送するのに用いられます。

ICMP上で用いられるコマンドとして有名なものとして、pingやtrace routeで用いられるEcho Replyや、ルータ間のMTU(1回のデータ転送で送信可能なIPパケットの最大値)の調整などがあります。

ICMPはTCPやUDPと同じく、TCP/IP参照モデルの第3層であるトランスポート層に属するプロトコルですが、ルータを越えて他のネットワークと通信する必要からIP上で用いられるため、IPよりは上でTCPやUDPよりは下という位置付けとなっています。

タイプ	コード	内容	
0(Echo Reply)	0	Echo Reply	
	0	Net Unreachable	
	1	Host Unreachable	
	2	Protocol Unreachable	
	3	Port Unreachable	
	4	Fragment Needed and DF was Set	
	5	Source Route Failed	
	6	Destination Network Unknown	
	7	Destination Host Unknown	
	8	Source Host Isolated	
	9	Communication with Destination Network is Administratively Prohibited	
	10	Communication with Destination Host is Administratively Prohibited	
	11	Destination Network Unreachable for ToS	
	12	Destination Host Unreachable for ToS	
	13	Communication Administratively Prohibited	
	14	Host Precedence Violation	
	15	Precedence cutoff in effect	
3 (Destination Unreachable)	0	Redirect Datagram for the Network	
	1	Redirect Datagram for the Host	
	2	Redirect Datagram for the ToS and Network	
	3	Redirect Datagram for the ToS and Host	
5(Redirect)	0	Echo Request	
8(Echo Request)	0	Time to Live exceeded in Transit	
11(Time Exceeded)	1	Fragment Reassembly Time Exceeded	

※出典： <http://www.infraexpert.com/study/tcpip4.html>