ネットワークの基礎も身につく

実習しながら学ぶ CCNA

黒木啓之／長屋未来◉共著

CUTT
カットシステム

・本書の内容についてのご意見、ご質問は、お名前、ご連絡先を明記のうえ、小社出版部宛文書（郵送またはE-mail）でお送りください。
・電話によるお問い合わせはお受けできません。
・本書の解説範囲を越える内容のご質問や、本書の内容と無関係なご質問にはお答えできません。
・匿名のフリーメールアドレスからのお問い合わせには返信しかねます。

はじめに

ネットワークはスマートフォンやそれに伴うSNSの普及により，現代の生活になくてはならないものとなっており，将来的にもまだまだ発展していきそうです．そのため，今後ネットワークを扱うことのできる技術者の不足が懸念されます．

ネットワークスキルを認定する資格試験はいくつかありますが，その中でもCCNA（Cisco Certified Network Associate）は，ネットワーク機器のスタンダードであるCisco社製機器を扱うことを前提にしているものの，試験内容はネットワーク全般をカバーしており，Cisco機器を扱う技術者だけでなく，ネットワークの基本から応用までの知識を得たい人にもオススメな資格になっています．しかしながら，難易度は高く，たやすく合格ができる資格ではありません．

またCCNAの内容を約20年教えている過程で，学生たちがどのようなことを良く間違えるのか，ということがわかって来ました．そのため，ここで得られたノウハウをどこかで表現できないかな，と思った所，本書の執筆の打診をいただき，書かせていただきました．

本書はCCNAおよびネットワークを学習する方々にわかりやすく，さらに学習する内容を淡々と記述するのではなく，簡単で理解がしやすいものについては簡潔に，また難しかったり理解しにくいものについては図を使うなどの工夫をして書いています．

一方，現在のCCNA200-301試験登場時にそれまであった「シミュレーション問題」が廃止されましたが，近年それが復活しました．また，何しろCCNAの内容は本を読むだけでなく実際に手を動かした方が理解が深まります．このようなことから，本書は実習を多く掲載しています．

本書は，次のような特徴があります．

- 本の内容はCCNAを取得するための内容になっていますが，ネットワークの基礎から応用までを学習するためにも書かれています．また大学や高専，高校，専門学校等の授業でも使えるように，CCNAの内容を踏襲しつつ，ネットワークの内容を広くカバーし，さらに実習で理解を深めるという形にしてあります．
- CCNAの内容については，全てを網羅するものではなく重要と思われるところをピックアップして記述しています．そのため，CCNA試験を完璧な形で受験するにはもう少し別の形で勉強する必要はありますが，短期間で学習する本では物足りないが，膨大なページ数の本を読むのは苦しい，という方に向いている内容となっています．
- 本書は実習が多く掲載されています．これは前述の通り近年復活し，また他書で少なくなってきたシミュレーション問題へ対応するためです．全てではありませんが，前に戻ったり忘れて

しまって進めないことがないよう，実習はそれぞれで完結するように，前に学習したことでも再度掲載してあります．

- 実習はできるだけ実行するよう推奨しています．これは書面だけの理解よりは手を動かすことで理解が深まり，また実習は以前に学習した内容も踏まえて行われることも多いので復習になるからです．さらに学習後，資格習得後には知識があるだけでなく実際にネットワーク機器を扱うことのできる人材になっていただきたいとの目的もあります．やはり，知識だけでなく実際に操作できる人の方が重宝されます．私が教えた卒業生達は実習をたくさん行った結果，社会に出ても機器がすぐに操作できる実践的な技術者として活躍しています．
- 脚注に英単語の訳や補足説明を入れています．特にカタカナで書かれている重要語はそれをそのまま覚えなければいけない，と思っている人が多いですが，これを英語に直し，さらにそれを日本語にすると，その役割や機能がそのまま表現されていることが多いです．たとえば，「ポート番号」の「ポート」が「港」の意味だと分かると，ポート番号自体の理解が深まります（本文中にはさらにイメージが掲載されています）．また本文中に書くと読む妨げになってしまうが，理解の助けになったり，難しいがちょっと知っていたりすると問題が解きやすくなるなどの内容も脚注に入れています．

最後に，この執筆に共著者として関わっていただいた長屋未来さんには，実習の部分を大変丁寧に執筆していただいた上に，たくさんのアドバイスをいただきました．また本書の出版社である株式会社カットシステムの石塚勝敏社長をはじめ社員の方々には，本書を執筆させていただくチャンスをいただいた他，本書作成の過程で多大なる協力をいただきました．皆さんには感謝を申し上げます．

筆頭執筆者　黒木啓之

この本の読み方

● 図中のネットワーク機器のシンボル（アイコン）は，それぞれ次のようになっています．

ルータ　スイッチ　L3 スイッチ　ブリッジ　ハブ　PC

また，図の一部では型番や役割の区別をする必要がない場合，また説明の都合上で次のシンボルを使っています．

ルータ　スイッチまたはハブ　バス型ネットワーク※1

● ルーターが「ルータ」，ユーザーが「ユーザ」など，最後の「ー」が省略されていますが，これは工学の分野ではマイナス記号やハイフンと間違うなどのことから省略されることが多く，本書はその慣例にならって省略されています．

●「設定」のタイトルで実習ができるようになっているものは，Cisco 社の Packet Tracer（パケットトレーサ）でできるようにしてあります（Packet Tracer は現在ユーザ登録をすればインストール・利用することができます）．それで動作確認もしています．またその際，

- シンボルに書かれている型番は，Packet Tracer で使うものを示しています（別のルータやスイッチでも同じコマンドで設定は可能な場合がありますが，インターフェイスの数の違いやコマンドが一部利用できないなどがあります）．
- 実線（―）はストレートケーブル，破線（- - -）はクロスケーブルで接続することを意味しています．
- Cisco 社のネットワーク機器のコマンドは，他のコマンドと区別がつけば省略が可能となっています．そのため，一部省略形で書かれています（例：`configure terminal` → `conf t`，`int` → `interface`）

※ 1　これは初期のネットワークがバス型ネットワーク（10BASE-2 や 10BASE-5）のため，ネットワークの基本はこれを使って説明されることが多いです．また現在では，バス型ネットワークはハブでの接続で同様のことができます．ただし，現在ではこの形のネットワークはほとんどありません．

目次

第 1 章

ネットワークの基礎

1.1 2 進数，16 進数，論理積

2 進数

2 進数は，0 ～ 9 までの数字を 10 個使う 10 進数に対して，0 と 1 の 2 個の数字を使う数です．ネットワークも PC も内部ではこの 2 進数で数が表現されており，動作もこの 2 進数を用いて行います．

例えば，10 進数の「2」は 2 進数では「10」となります．これは，

- 10 進数　　0，1，2，3，4，5，6，7，8，9，10
- 2 進数　　0，1，10，11，100，101，110，111，1000，1001，1010

のように，10 進数は 10 個進んで桁が変わるのに対し，2 進数は 2 個進んで桁が変わります．

また，ネットワークの動作を理解するために，10 進数を 2 進数に変換することがよくあります．それには大きく 2 つの方法があります．10 進数の 235 を 2 進数に変換する場合を例に説明します．

1 つ目は，「2 で割っていって余りがあるかどうかを計算していく」方法です．図 1.1 のように

2 で割っていき，余りが 1 か 0 かを書いていき，0 となったら（割れなくなったら）その余りを下から並べれば変換終了です．結果として，11101011 となります．

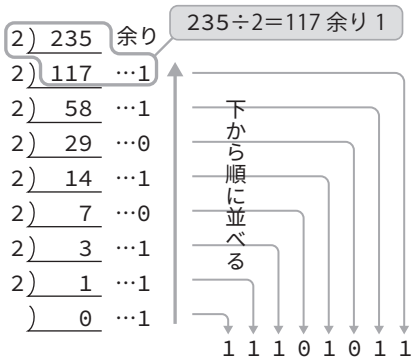


図 1.1 ● 10 進数の 235 を 2 進数に変換する例

2 つ目は，計算を苦手としない人向けの方法です．2 進数には各桁でそれぞれ図 1.2 のように 128，64，32，16，8，4，2，1 の意味を持っていることから，変換したい数をこの数の分引いていき，引けたら 1，引けなかったら 0 とする方法です．

1	1	1	0	1	0	1	1
128	64	32	16	8	4	2	1

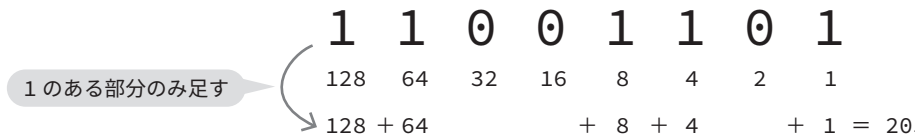
図 1.2 ● 2 進数の各桁の意味

例えば 235 は，

235 から 128 は引ける	8 桁目を 1 として，235 から 128 を引く	（= 107）
107 から 64 は引ける	7 桁目を 1 として，107 から 64 を引く	（= 43）
43 から 32 は引ける	6 桁目を 1 として，43 から 32 を引く	（= 11）
11 から 16 は**引けない**	5 桁目を 0 とする	（= 11 のまま）
11 から 8 は引ける	4 桁目を 1 として，11 から 8 を引く	（= 3）
3 から 4 は**引けない**	3 桁目を 0 とする	（= 3 のまま）
3 から 2 は引ける	2 桁目を 1 として，3 から 2 を引く	（= 1）
1 から 1 は引ける	1 桁目を 1 として，1 から 1 を引く	（= 0 で終了）

となります．

逆に，2進数を10進数に変換することがあります．これは，前述のようにそれぞれの桁が128，64……の意味を持っていることから，2進数で1があるところの数を足せばよいのです．つまり2進数11001101は，



となります．ネットワーク関係で覚えておきたい2進数の値を表1.1に示します．

表1.1 ●ネットワーク関係で重要な2進数の値

64	01000000	128	10000000	248	11111000
63	00111111	192	11000000	252	11111100
127	01111111	224	11100000	254	11111110
191	10111111	240	11110000	255	11111111

16進数

16進数は，

0，1，2，3，4，5，6，7，8，9，A，B，C，D，E，F，10

のように，16個進んで桁が変わる数です．ただし10進数で10以上は

10 → A，11 → B，12 → C，13 → D，14 → E，15 → F

と，英大文字を利用して1桁で表現します．

2進数の例での235を16進数に変換すると，図1.3に示す手順で「EB」となります．

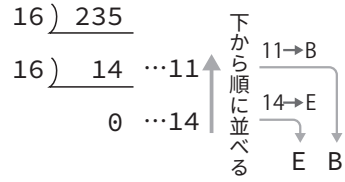


図1.3 ● 10進数の235を16進数に変換する例

なお，16 進数の特徴として，「2 進数を下から 4 桁ずつに区切り，それぞれを 16 進数に変換する」ことで，2 進数から 16 進数へ変換をすることができます．2 進数「11101011」を 4 桁ずつ「1110」と「1011」に分けると，それぞれ 10 進数で「14」と「11」，16 進数では「E」と「B」になるので，先の結果と同じになります．

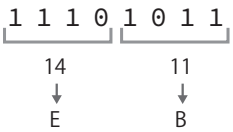


図 1.4 ● 2 進数を下から 4 桁ずつに区切り，それぞれを 16 進数に変換

論理積

ネットワークでは，2 進数のかけ算（積）が必要なときがあります．これを**論理積**と言って，普通のかけ算と区別します．

2 進数は 0 と 1 しかないので，かけ算は右の 4 通りしかありません．

「論理積」などと難しいことを言いましたが，実は普通のかけ算と同じです．0 に何をかけても 0 ですから上の 3 つは 0 となりますが，1 × 1 は 1 となっているだけです．

表 1.2 ● 2 進数のかけ算

X	Y	結果
0	0	0
0	1	0
1	0	0
1	1	1

1.2 プロトコル

プロトコルとは，通信での約束事のことです．通信規約などとも言います．データを確実に通信する場合には，相互で同じプロトコルに従わなければなりません．分かりやすく言えば，多くの国の人がいる場で「ここでは英語を話す」というプロトコルに従えば，会話（通信）が正しく伝わります．

1.3 LAN と WAN

LAN（Local Area Network）とは，主に会社や学校，家庭など小さな領域のネットワークのことを言います．LAN は，現在では LAN ケーブルや無線 LAN で繋がっている部分を言います．また，スイッチなどのネットワーク機器のみで構成されている部分を言います．

WAN（Wide Area Network）とは，全世界的な領域のネットワークのことを言います．WAN は，現在ではインターネットと呼ばれる部分を言います．また，ルータなどのネットワーク機器で構成されている部分を言います．

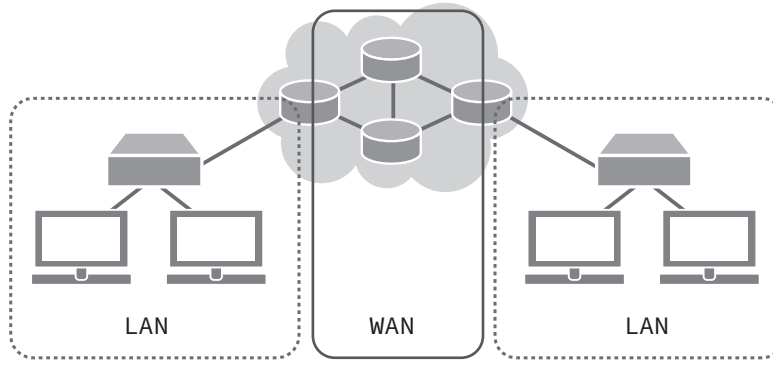


図 1.5 ● LAN と WAN

1.4 OSI 参照モデル

OSI 参照モデルとは

OSI 参照モデルとは，ネットワーク開発を迅速かつ円滑に進めるために作られたネットワークの動作を「層」という形で表したものです．

OSI 参照モデルは次の 7 層で構成されています※ 1．

※ 1　OSI は Open System Interconnection の略．また OSI 参照モデルを第 7 層から最初の文字を取って「アプセトネテブ」として覚えると，覚えやすいです．

第 7 層　アプリケーション層
第 6 層　プレゼンテーション層
第 5 層　セッション層
第 4 層　トランスポート層
第 3 層　ネットワーク層
第 2 層　データリンク層
第 1 層　物理層

図 1.6 のように 2 つの通信機器で通信する際，各層では同じ規格（プロトコル）で運用します．例えば，「ネットワーク層相互では IP アドレスを使う」のようにしておけばよい，ということです．もし新しい規格を採用しようとした場合には，上下の層に送るデータはそのままにすれば，その層だけを新しいものに変え，他の層はそのままにすることができます．そのため，すべてのプロトコルを見直す必要がなく，迅速かつ円滑に開発をすることができます．

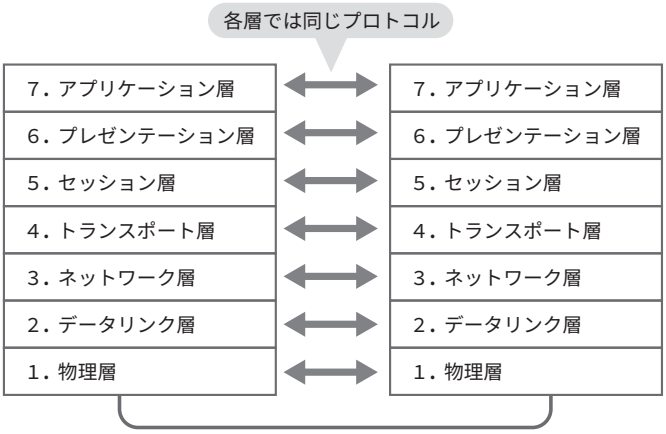


図 1.6 ● OSI 参照モデルに基づく通信

各層について簡単に説明します．

● **アプリケーション層**は，メールやテキストエディタなどのソフトウェアに当たる部分です．ホームページを見る場合にはウェブブラウザから「このページを見たい」などのデータを送り出し，ホームページのサーバからはそのホームページのデータを受け入れる層となります．

- **プレゼンテーション層**は，データの表現方法（プレゼンテーション）を決める部分です．例えば，データを圧縮したり，暗号化したり，文字コードなどのデータの表現方法を司っています．

- **セッション層**は，通信の始まりから終わりまで（= セッション）を司る部分です．具体的には，通信経路の確立・切断を管理します．また，セッションのスケジューリングやログインなどの認証などもここで行われます．

- **トランスポート層**は，（1）データを保証するかしないか（TCP/UDP），（2）どのプロトコル（アプリケーション）からデータが来たのか（データを送るのか），（3）送ったデータが正しく送られているのか（3 ウェイハンドシェイクやデータの到着確認）などを行う部分です．このトランスポート層におけるデータのことを**セグメント**と言います．（1）～（3）の機能は後述します．

- **ネットワーク層**は，WAN でのデータの行き先を決めるアドレス（IP アドレス）を使ってデータを送受信する部分です．データを送り出す場合，上位（トランスポート層）から来たデータに宛先および送信元 IP アドレスを付けます．このネットワーク層におけるデータのことを**パケット**と言います．

- **データリンク層**は，LAN でのデータの行き先を決めるアドレス（MAC アドレス）を使ってデータを送受信する部分です．データを送り出す場合，上位（ネットワーク層）から来たデータに宛先および送信元 MAC アドレスを付けます．このデータリンク層におけるデータのことを**フレーム**と言います．

- **物理層**は，ケーブル※2 や電気信号などの「実態があるもの（物理的なもの）」の部分です．光ケーブルか銅線のケーブルかなど，どのようなケーブルを使うかや，どのような方式の信号で通信するかなどを決めている部分です．

※2　Cisco では，ケーブルのことを「メディア」と言います．

1.5 トランスポート層の機能

トランスポート層は，前述した通り多くの機能（役割）を担っています．ここでは，このトランスポート層の機能を詳しく説明します．

TCP と UDP

データの中には，ファイルや画像など一部でも欠落してしまうと困るものと，ライブ配信のように，音声や動画などが一部欠落してもリアルタイム性を重視したいものがあります．この欠落したら困るデータは **TCP**（Transmission Control Protocol）で，欠落しても良いデータは **UDP**（User Datagram Protocol）で送ります．

TCP は相手と接続の確認を取ったり，データが正しく宛先に送られているかなどのチェックを行いますが，UDP はそのようなチェックを行いません．TCP のようなチェックを行うものを**コネクション型**，UDP のようなチェックを行わないものを**コネクションレス型**と言います．

ポート番号

データを受信した場合，そのデータをメールアプリに渡すのか，あるいはブラウザに渡すのかを決めておく必要があります．この，どのアプリケーション（プロトコル）に渡すのかを番号として表現し，その番号を**ポート番号**と言います．

ポート番号は，図 1.7 のように指定されたポート（港）に着岸してデータの積み下ろしを行うことをイメージすると分かりやすいでしょう．

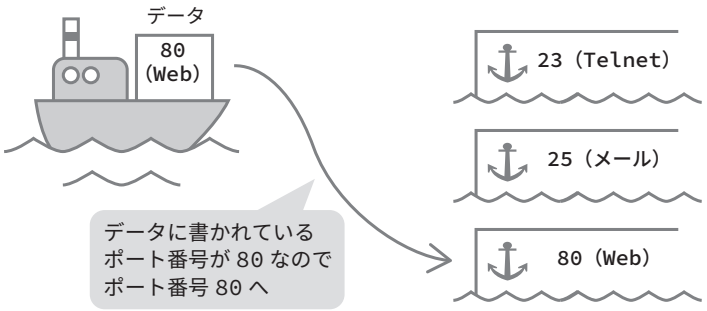


図 1.7 ●ポート番号のイメージ

主要なプロトコルとポート番号を表 1.3 に示します．

表 1.3 ●主要なプロトコルとポート番号

プロトコル	ポート番号	TCP/UDP
ftp	20，21	TCP
ssh	22	TCP
telnet	23	TCP
SMTP	25	TCP
TFTP	69	UDP
HTTP	80	TCP
POP3	110	TCP
NTP	123	UDP
SNMP	161	UDP
HTTPS	443	TCP

3 ウェイハンドシェイク

前述のように，トランスポート層では TCP と UDP という 2 つの通信方法がありますが，特に TCP では，通信をはじめる前に相手と通信できる状態かを確認する**コネクションの確立**を行います．この動作は，図 1.8 に示すようにトランスポート層のデータであるセグメントの中に書かれている，SYN と ACK という**フラグ**を利用して次のような順番で行われます※3．

（1）PC1 は PC2 へ，SYN=1，ACK=0 とした通信をしたい（SYN=1）というセグメントを送ります．
（2）受信した PC2 は PC1 へ，SYN=1，ACK=1 とした通信を了解した（ACK=1），さらにこちらからも通信したい（SYN=1）というセグメントを送ります．
（3）さらに受信した PC1 は PC2 へ，SYN=0，ACK=1 とした通信を了解した（ACK=1）というセグメントを送ります．

もう少し詳しく説明すると，SYN=1 は「通信したいが良いか？」という意味のフラグ，ACK=1 は「通信をするのを了解した！」という意味のフラグです．特に（2）の SYN=1 は，（1）で

※3　フラグ（フラッグ）は 0 か 1 かの値をとり，「1」はフラグ（旗）が上がっていること，「0」はフラッグが下がっていることを意味します．これ以降，これらのフラグやシーケンス番号，確認応答番号を含めたデータのことをセグメント，これらを含めないデータ自身をデータと表現します．

SYN=1 となっているのでそのまま残して PC1 に送り返したのではなく，「PC2 から PC1 へ通信したい」という意味があります．

このことを 3 回のやりとりで確認をすることから**3 ウェイハンドシェイク**と言います．また，TCP はこのように確認をしてから通信を行うために**コネクション型**と言い，逆に UDP はこのように確認をせずに一方的に通信をするため**コネクションレス型**と言います．

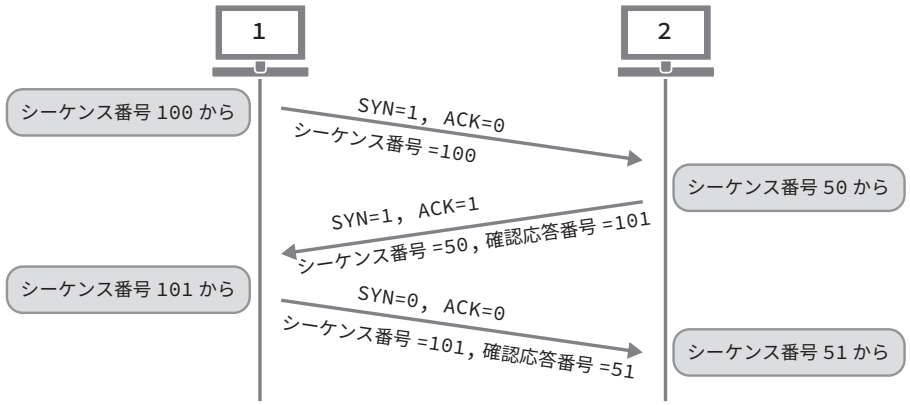


図 1.8 ● 3 ウェイハンドシェイク

さらに，図 1.8 に示すようにこの 3 ウェイハンドシェイクと同時に，セグメントの中にあるシーケンス番号と確認応答番号を利用して，3 ウェイハンドシェイクが終了した後，実際にデータを通信する際のデータの順番を表したり，どこまで通信したかを確認します．

シーケンス番号はデータの順番を表すものです．データが大きくて 1 回で送れない場合は，細切れにして複数回に分けて送ります．この細切れにしたデータに番号を打っておき，受信側でデータを 1 つにするときにこのシーケンス番号の順番で戻します．**確認応答番号**は，どのシーケンス番号まで送られたか（実際には，次はどのシーケンス番号から始まるか）を示す番号です．データを送ったときに番号が適切に増えた場合は正しく送られたことが確認でき，番号が適切に増えていない場合は正しく送られていないことが確認できます．

実際には，

（1）シーケンス番号を 100 からとした場合，PC1 は PC2 へ「シーケンス番号 = 100」としたセグメントを送ります．

（2）受信した PC2 は PC1 へ，シーケンス番号に 1 を加えた「確認応答番号 = 101」と同時に，こちらからもシーケンス番号を 50 からとした場合「シーケンス番号 = 50」というセグメントを送ります．

（3）さらに受信した PC1 は PC2 へ，同様にシーケンス番号に 1 を加えた「確認応答番号 = 51」と，さらに受信した確認応答番号をシーケンス番号とした「シーケンス番号 = 101」というセグメントを送ります．

3 ウェイハンドシェイクでは，最初のシーケンス番号を送信元と宛先でやりとりすると同時に，このシーケンス番号に 1 を加えたものを確認応答番号として，通信開始の確認をしています．

データの到着確認

TCP では，さらにセグメントが正しく送信されたかどうかの仕組みを持っています．またここでは，一度に送信できるデータの大きさに制限（Maximum Segment Size; MSS）があることから，MSS が 300 バイトであると仮定し，500 バイトのデータを 300 バイトと 200 バイトで分割し，さらにはシーケンス番号によってデータの順番を確保していることを想定しています（通常 MSS は 1460 バイトです）．

3 ウェイハンドシェイクの後，実際にデータの通信が行われますが，図 1.9 のように，PC1 から PC2 に例えば 300 バイトのデータを送った場合には，PC2 からは確認応答番号として（最初のシーケンス番号が 101 だとすると）401 が送られてきます※ 4．同時に ACK=1 が受信確認のためのフラグとして付いてきます．

さらに PC1 から PC2 に 200 バイトのデータを送った場合には，PC2 からは確認応答番号としてシーケンス番号が 401 だったので 601 が送られてきます．やはり同時に ACK=1 が受信確認のためのフラグとして付いてきます．

※ 4　201 + 300 = 501 が次に送られてくるデータの最初のシーケンス番号となります．「101 から 300 バイト」なので 400 までで（401 ではありません），次に送られてくるデータのシーケンス番号は 401 が期待されます．

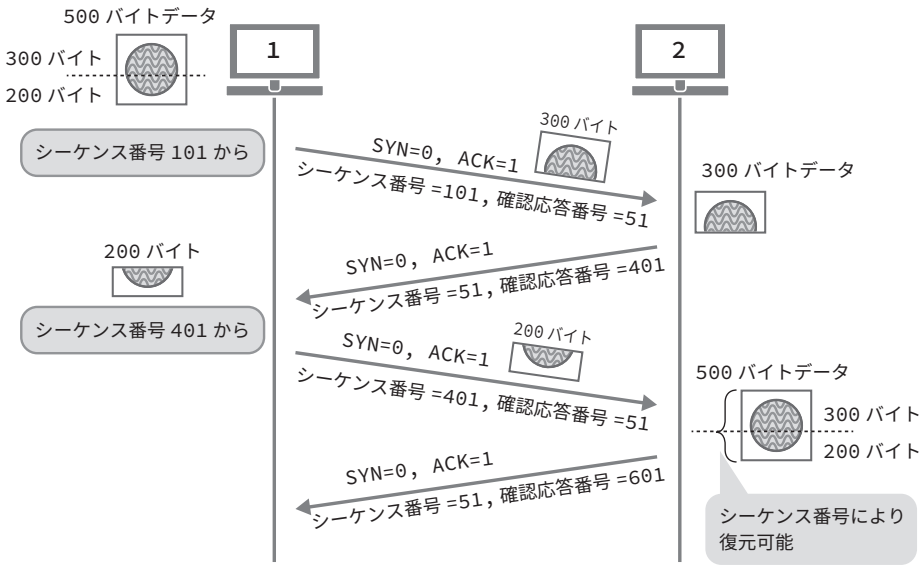


図 1.9 ●データの到着確認

再送の仕組み

データが宛先に送られていない場合，または，データは送られたが確認応答が到着しない場合，データも確認応答も送られたが一定時間内に確認応答が到着しない（送受信が遅延している）場合には，確認応答が来ていないデータを再度送ります．その際，特に遅れてデータが到着したり確認応答が来たりした場合にはデータが重複して相手に送られてしまいますが，これはシーケンス番号で管理されているため重複部分は破棄されます．

すなわち，TCP におけるデータの欠落などは，「確認応答が来なかったデータを確認応答が来るまで再送信する」ことで対応します．

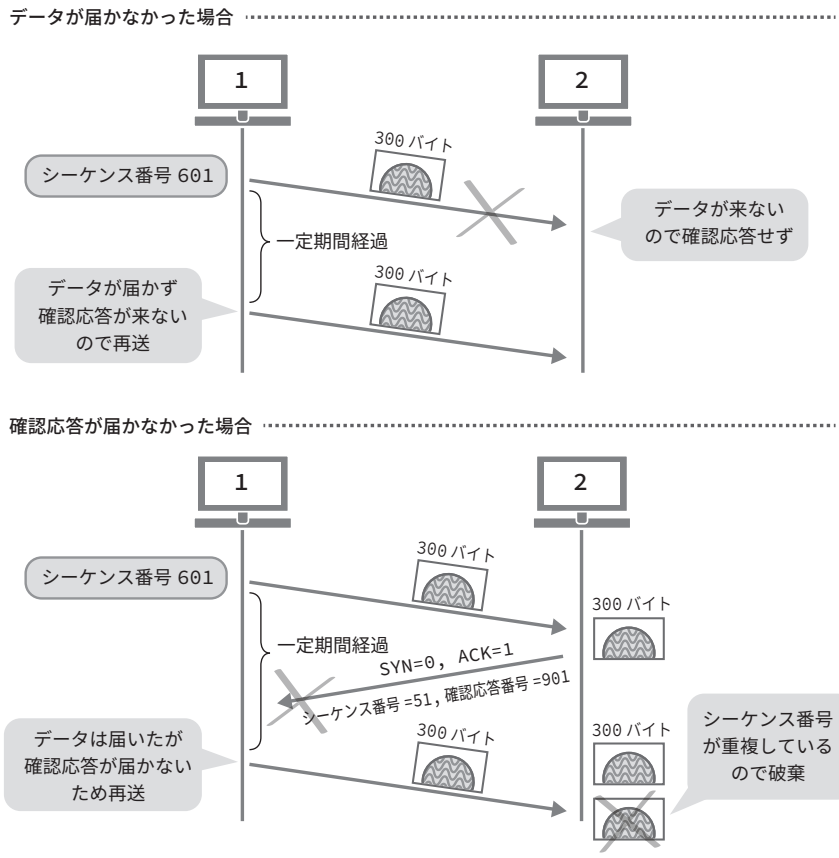


図 1.10 ●再送の仕組み

■ ウィンドウ制御とフロー制御

データの到着確認は，データを確実に相手へ届ける仕組みとしては良いのですが，確認応答が来るまで待っているとなかなか通信が進みません．また，セグメントは基本的には 1460 バイトまで一度に送ることができますが，PC などは一時的にデータを保持しながら受信することができるデータ量が 1460 バイト以上あることがあります．このように，一時的に 1460 バイト以上保持できる場合，確認応答を待たずにドンドン送ることができます．この PC が一時的に保持できるサイズのことを**ウィンドウサイズ**といいます．実際には，3 ウェイハンドシェイクの際にこのウィンドウサイズをお互いに通知し合っています．また，この送信側が送ることのできるウィンドウサイズ分