マスタリングTCP/IP 応用編

11300j038
加藤尚樹

概要

- 内容量が多かったため、時間内にすべてを説明するのは不可能だと考え、この中でも最も重要であるIPとTCP説明していきたいと思います。

IP(Internet Protocol)

- IPの2つの基本機能
  - アドレス付け
  - フラグメンテーション

基本機能を提供する機能の分類

- TOS (Type Of Service)
  必要な特定の「サービスの品質」を指定する機能
- 生存時間
  データグラムが送信されてから破棄されるまでの時間。
- オプション
  オプションを設定することによってより有益な制御機能を提供している。

基本機能を提供する機能の分類

- ヘッダのチェックサム

  ヘッダ内の情報のチェック

- エラーレポーティング

  IP自体にはエラー報告機能はないが、IPの上位プロトコルであるInternet Control Message Protocol (ICMP)によってエラーなどの状態メッセージが運ばれる

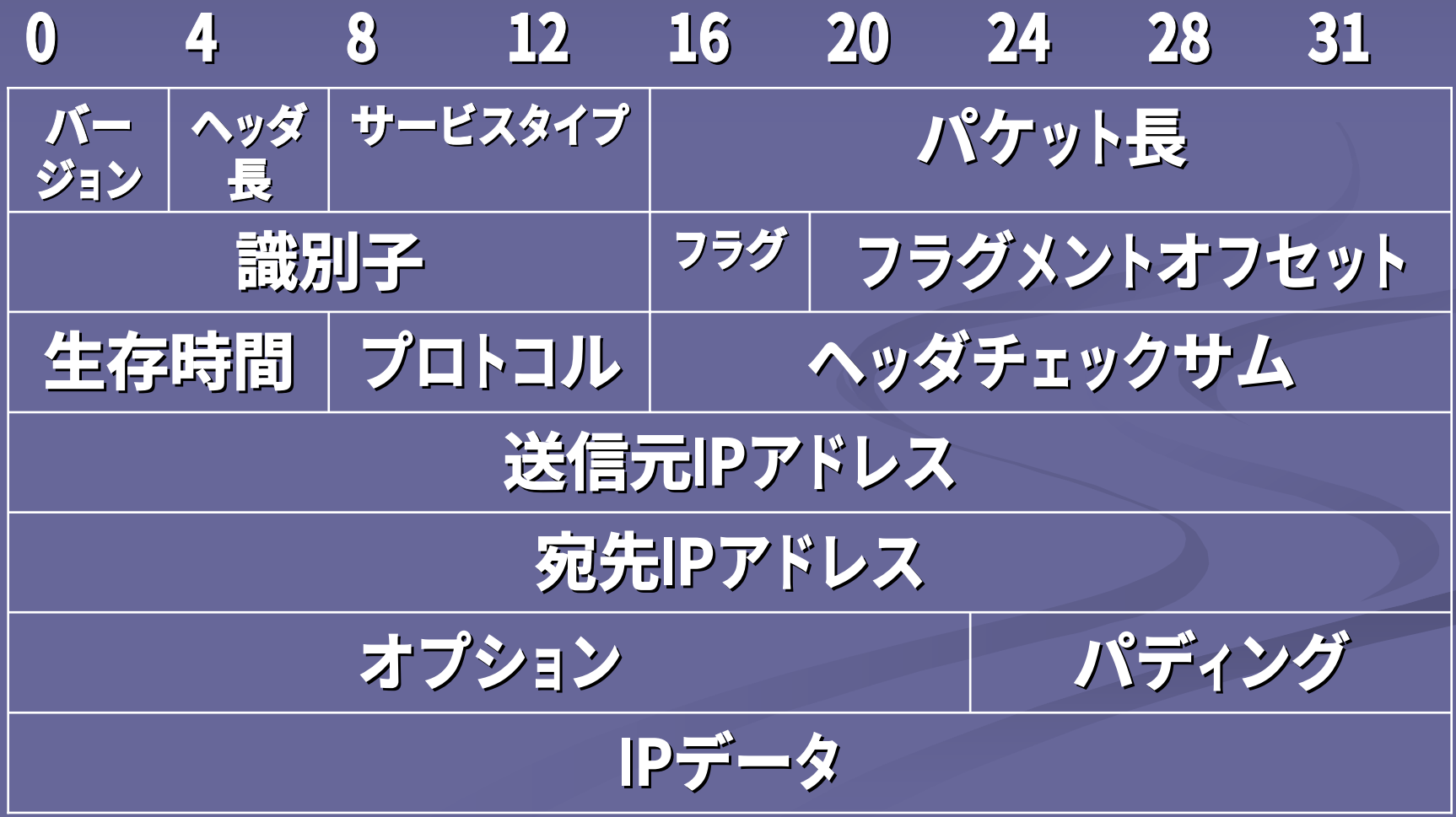
IPデータグラム

- MTU

  IPデータグラムは物理フレームかMACフレームによって運ばれる。このカプセル化によってデータグラムのサイズはフレームのデータエリアの長さを越えることができないため、制限がつく。これをMTU（ネットワーク最大転送単位）という。

IPデータグラム

- フォーマット



フォーマット詳細

- バージョン(4bit)

  IPのバージョンを示す。これによってIPヘッダのフォーマットがわかる。

- ヘッダ長(4bit)

  IPヘッダの長さを示す。現在では32bitとされており、これによってIPヘッダの開始位置、及び終了位置を知ることができる。

- サービスタイプ(8bit)

  必要なサービス品質をホストに指定することができる。次のページに詳細を示す。

サービスタイプのビット

0	1	2	3	4	5	6	7
優	先	度	D	T	R	0	0

- 優先度

  この最初の3bitによって0～7間での値を設定し、優先順位を与える。

- D,T,R

  これらの各ビットはそれぞれ、必要な遅延、スループット、信頼性を指定している。サービスタイプのビットの意味を次に示す。

D	T	R	意味
0			通常の遅延で処理する
1			低遅延で処理する
	0		通常のスループットのパスを与える
	1		高スループットのパスを与える
		0	通常の信頼性のパスを与える
		1	高い信頼性のパスを与える

- 残りの2ビットについて
  - 第6bit はRFC1349によって『金銭的コストの最小化』の意味を持つ
  - 第7bit は予備として空けられており通常は0となっている。

TOSの設定例（奨励値）

プロトコル	TOS値				説明
	D	T	R	M	
TELNET	1	0	0	0	遅延を最小化
FTP制御	1	0	0	0	遅延を最小化
FTPデータ	0	1	0	0	スループットを最大化
SMTPコマンド	1	0	0	0	遅延を最小化
SMTPデータ	0	1	0	0	スループットを最大化
NNTP	0	0	0	1	コストを最小化
SNTP	0	0	1	0	信頼性を最大化

フォーマット詳細 (続き)

- パケット長(16bit)
  - データグラム全体の長さをヘッダとデータを含めてオクテット単位で示す。最大サイズは64KBである。
  - よってどのホストも576オクテットまでのデータグラムは受信できなければならないというのが受け入れられている。
  - また、576オクテットを超えるデータグラムを送信できるのは、受信側が受信できるとわかっているときに限り行われる。

上限値と下限を設ける考え方

必須IPヘッダのほかに適切なサイズを保証するのが目的である。

例：

IPヘッダで許される最大のサイズ60オクテット
適切なデータサイズが仮に512オクテット
60+512=572<576

標準IPヘッダ20オクテットとした場合
20+512+α＜576　（αは上位プロトコルヘッダなど）

フォーマット詳細 (続き)

- フラグ(3bit)
  - フラグフィールドには3つのフラグが用意されているが最初の1bitは必ず0であり使用されていない。
  - 第2ビットはフラグメンテーション不可フラグで、これが1の場合、どんな場合もフラグメンテーションを行ってはいけない。
  - 最終ビットは後続フラグメントがあるがあることを示すビットで、複数のフラグメントを持つデータグラムの場合に、識別子フィールド及び、フラグメントオフセットフィールドとともに使われる。

フォーマット詳細 (続き)

- フラグメントオフセット(13bit)
  - データグラム全体におけるこのフラグメントの相対位置を受信ホストに伝える。
  - オフセットは必ず8オクテット(64bit)で示され、この単位をオフセットブロックという。
  - オフセットブロックは必ず、最終オフセットを除き8の倍数であり、最短長も8である。
- 生存時間(8bit)
  - このデータグラムがインターネット上で存在していることのできる最大時間を示す。
  - 単位は秒で、少なくともルーターを一つ通るごとに1減少される。
  - 0になるとデータグラムは破棄され、ICMPによって、送信元に伝えられる。

フォーマット詳細 (続き)

- プロトコル(8bit)
  - データグラムのデータ領域で運ばれる上位プロトコルを示す。これによってデータグラムを自己識別型としている。以下に主なプロトコルと番号を示す

コード	プロトコル
01	ICMP
04	IP
06	TCP
11	UDP

フォーマット詳細（続き）

- ヘッダチェックサム
  - これはIPデータグラムの完全性をチェックするもので以下のような手順によって実現される。

1. ヘッダの一連を16bit単位に分割する。
2. 1の補数演算を用いてワードを合計する。
3. 桁上げをすべて加算して最後の結果の1の補数を取る。
4. 受信ホストで同様の計算を行い、チェックサムと照らし合わせて、破損がないことを確認する。

フォーマット詳細（続き）

- 送信元アドレス
  送信元のアドレス
- 宛先アドレス
  最終宛先のアドレス
- オプション
  後に説明
- パディング
  長さは1～3オクテットでデーターグラムヘッダが32bitになるように調整するために存在する。

IPデータグラムのオプション

- オプションの構成
  基本的に次の二種類のフォーマットに分けられる
  - オプションタイプのみのもの
  - 可変長オプション

IPデータグラムのオプション（続き）

オプションクラス	オプション番号	オプション長	オプション名と説明
0	0	-	オプションリスト終了
0	1	-	No Operation
0	2	11	セキュリティ
0	3	可変長	ルーズソースルーティング
2	4	可変長	インターネットタイムスタンプ
0	7	可変長	レコードルート
0	8	4	Stream ID
0	9	可変長	ストリクトソースルーティング

ルーズソースルーティングとストリクトソースルーティング

- オプションデータの構成
  - 1オクテットのポインタと複数の4オクテットフィールドで、最終宛先までの経路上にある1つのホストのIPアドレスである。
  - 受信したホストはオプション長フィールドとポインタフィールドを調べる。ポインタのほうが長ければ、データとして使用済みであるためそれ以降のルーティングはヘッダにある宛先フィールドのみによって行われる。

		ポインタ	IP (1)	IP (1)	・・・	IP (N)

ルーズソースルーティングとストリクトソースルーティング

- ルーズソースルーティングとストリクトソースルーティングの違い
  - ルーズソースルーティングはオプションに保存されているアドレスまでに中間ルーターを経由するルートを選択することができる。
  - ストリクトソースルーティングの場合は、保存されているアドレスまで直にルーティングされ、もしそのアドレスのルーターが故障してた場合、たとえほかのルートがあったとしてもそのデータグラムは破棄される。

インターネットタイムスタンプ

- 自分と中間ホストの遅延時間をホストが計算することができるオプション

TCP

- 特徴
  - コネクション型
  - ほかのホストアドレスを指定したり、フラグメント化や再構成、優先順位の指定をしたりすることができない。
  - 下層プロトコルの信頼性についてはほとんど前提をおかず、汎用性を持つ。
  - 応答確認機能を持ち、上位層は信頼性を求める必要がない。
  - 複数のアプリケーションをサポートするように設計されている

TCPの動作

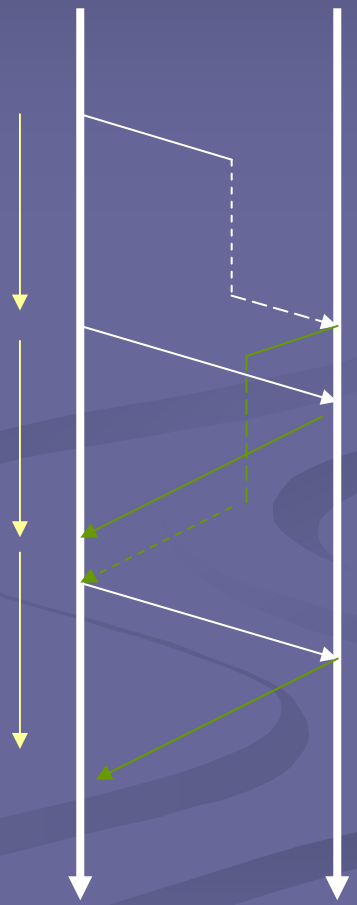
- 基本的なデータ転送
  - TCPはデータの発信と受信を同時に行うことができる・・・・・・・・・TCP層が全二重の性質を持つ
  - データのかたまりをセグメントと呼ぶ。
  - 一般的にセグメントと送信タイミングはTCPモジュールによって最適な値に決められるが、ユーザーが命令を出すことによって自由に帰ることもできる。

TCPの信頼性

- TCPの信頼性・・・・どのセグメントにも応答確認が必要である
- 応答確認信号はリモートホストから来るデータとともにTCPセグメントとして送られる。
- 応答確認信号はデータが送られた際に、タイマーが設定されその時間内に返される必要がある。

TCPの信頼性

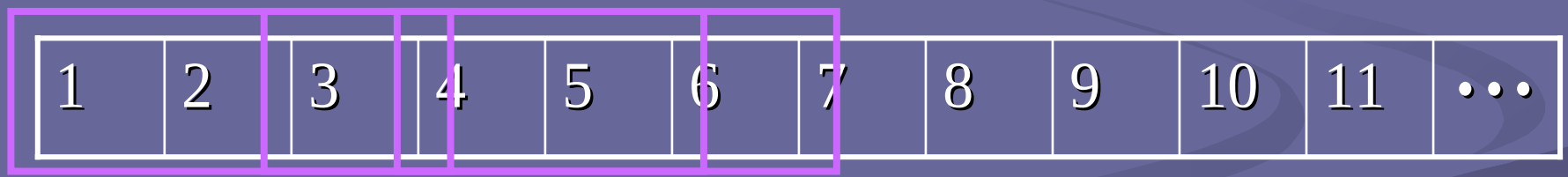
- 右図のようにセグメントの送信または応答信号の送信が遅れた場合、そのセグメントは再送され、遅れたセグメントは破棄される。
- セグメントおよび応答信号は消失してしまう場合があるが、この場合はタイムアウトになるので、再送される。



スライディングウィンドウ

- 今まで説明してきた方法では、セグメント送信後、応答確認が帰ってくるまで次の信号が遅れない。そこでスライディングウィンドウと呼ばれる方式が用いられる。下の例はウィンドウサイズを3とした場合である。

初期ウィンドウ　セグメント1の応答が到着　ウィンドウが移動　セグメント4の応答が到着　ウィンドウが移動



フロー制御

- TCPは送信側が送信できるデータ量を受信側が決定している。
- これを実現するために受信ホストは応徳確認の際ウィンドウを設け、受信できるオクテット数を示す。
- 注意点
  - 送信ホストに置かれたバッファは再送が必要になる可能性があるため応答があるまで削除できない。
  - また、受信データは受信処理がほかの処理によって時間がかかることも考えられるので、すぐには削除できない。

多重化

- TCPはさまざまなプロセスに同時にコネクション型の環境を提供するためにポートと呼ばれるアプリケーションアドレスを提供する。これとIPアドレスを結合することによってソケットと呼ばれるものを生成し、インターネット上のどこからもアプリケーションプロセスを個別に識別できるようにしている。
- 例
  - HTTP:80,FTP:21,TELNET:23，etc…

コネクション

- TCPの信頼性はコネクションの両端が同一シーケンス番号を持つことを前提としているため、コネクションの確立、維持、可能な限り閉じるという動作をしなくてはならないコネクションの確率（オープン）には次の2種類がある
  - パッシブオープン

    積極的にコネクションを求めるのではなく、送られてくるサービス要求を受け入れるときに用いられる。

    例：HTTP,FTPサーバー、TELNETサーバー

コネクション（続き）

- アクティブオープン

  ほかのホストとコネクションを積極的に開始しようとする場合に用いられる。

  例；ブラウザ、FTPクライアント、TELNETクライアント

- 注意

  パッシブオープン同士ではコネクションを結ぶことは不可能である。また、コネクションを確立できるのは双方がリモートプロセスのポートを知っている場合のみである。通常ウェルノウン（Well known）のポートに関しては自動的に割りあてられる。

TCPセグメントヘッダ

各フラグの説明

- URG

  緊急フラグで緊急ポインタが有効で使わなくてはならないことを示す

- ACK

  応答確認フラグで応答確認番号が有効であり、
  セグメントが応答確認を持っていることを示す

- PSH

  少量のデータをセグメントで転送するためのもの

- RST

  リセットフラグでリセットしなければならない場合に送られる。

各フラグの説明

- SYN

  シンクロナイズフラグでシーケンス番号が有効で先頭シーケンス番号として扱わなければならないことを受信側に教える。

- FIN

  終了フラグで送信側にはデータがなく、コネクションを閉じることを伝えるためにある。ただし、相手側はデータが残っている可能性があり、その場合は双方向から単方向リンクになる。

TCPオプション

- TCPヘッダは複数のオプションを指定可能だが現在定義されているのは3つだけである。
  - オプションリスト終了

    オプションフィールドの終わりを示すために設計されたが、実際に適用できるサイズが現在のところ4オクテットなので実際使われることはない
  - No Operation
    - オプション間でのミリデタとして使うように設計されているが実際使えるのが最大セグメントサイズしかないので現れるのは1回のみになり使われない

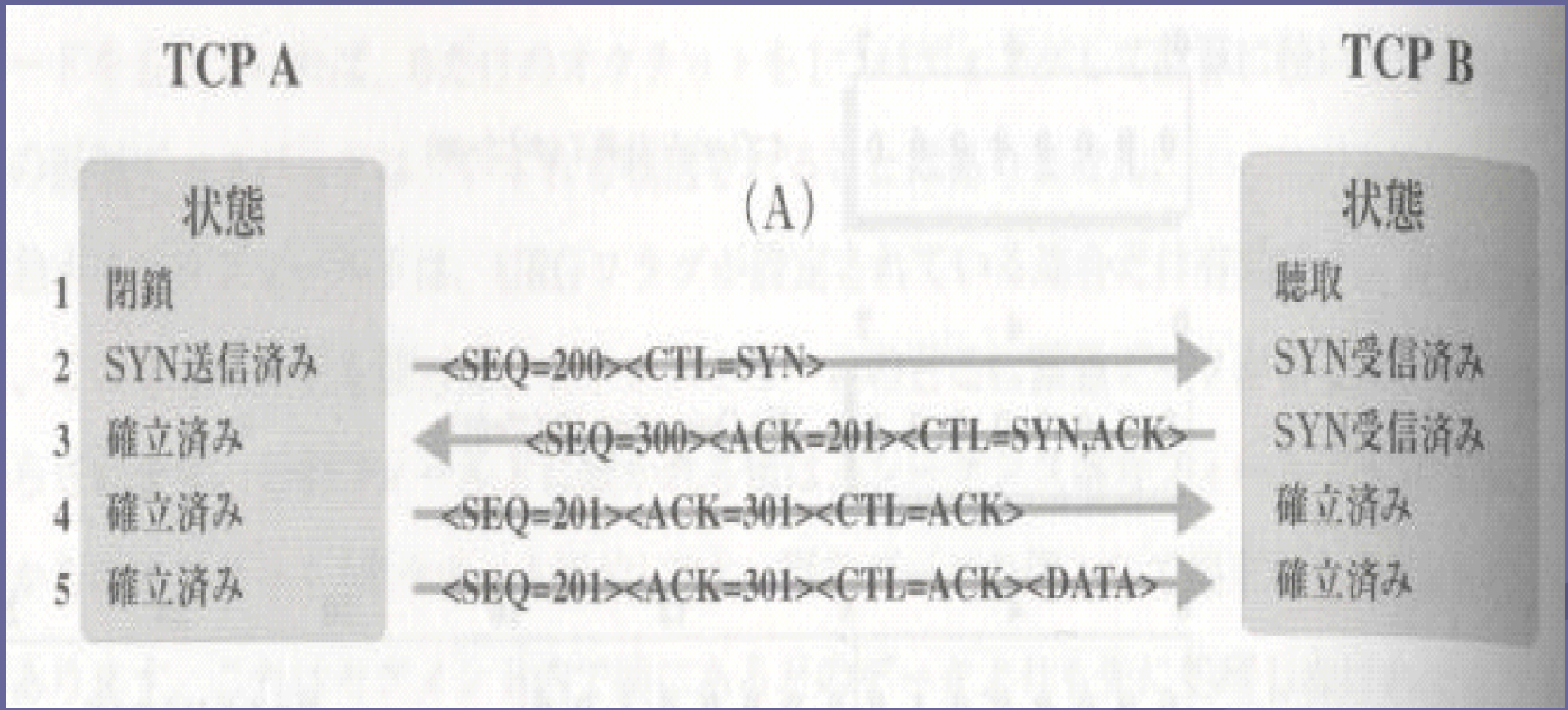
TCPオプション

- 最大セグメントサイズ

  常に32bitの長さで送信側TCPモジュールが受信可能な最大セグメントサイズを示す。このオプションが現れるのは1回のみでSYNビットが設定されていて、コネクションが確立される段階のみ。これが設定されていなければ自由なセグメントサイズを用いることができる。

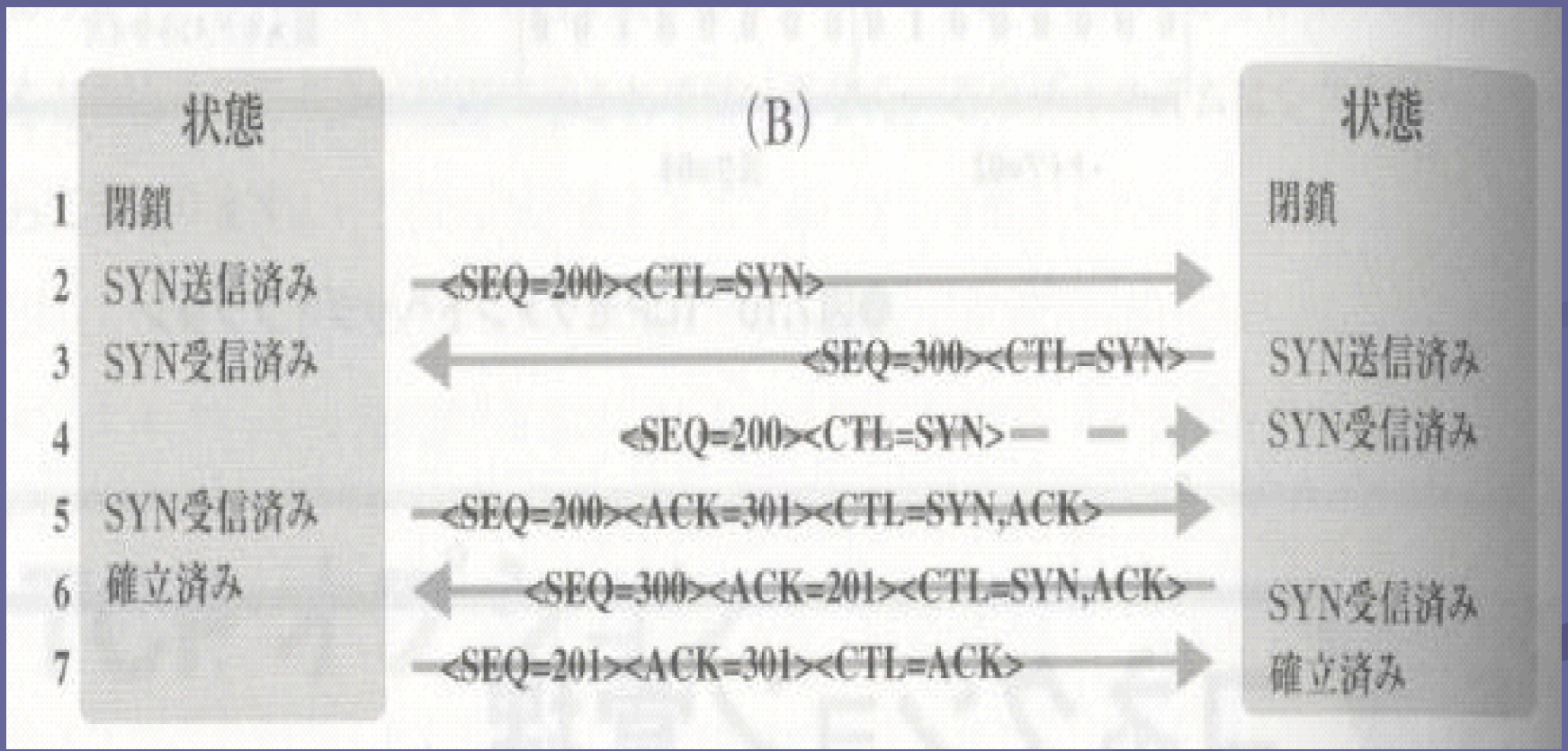
コネクション管理

- スリーウェイハンドシェーク



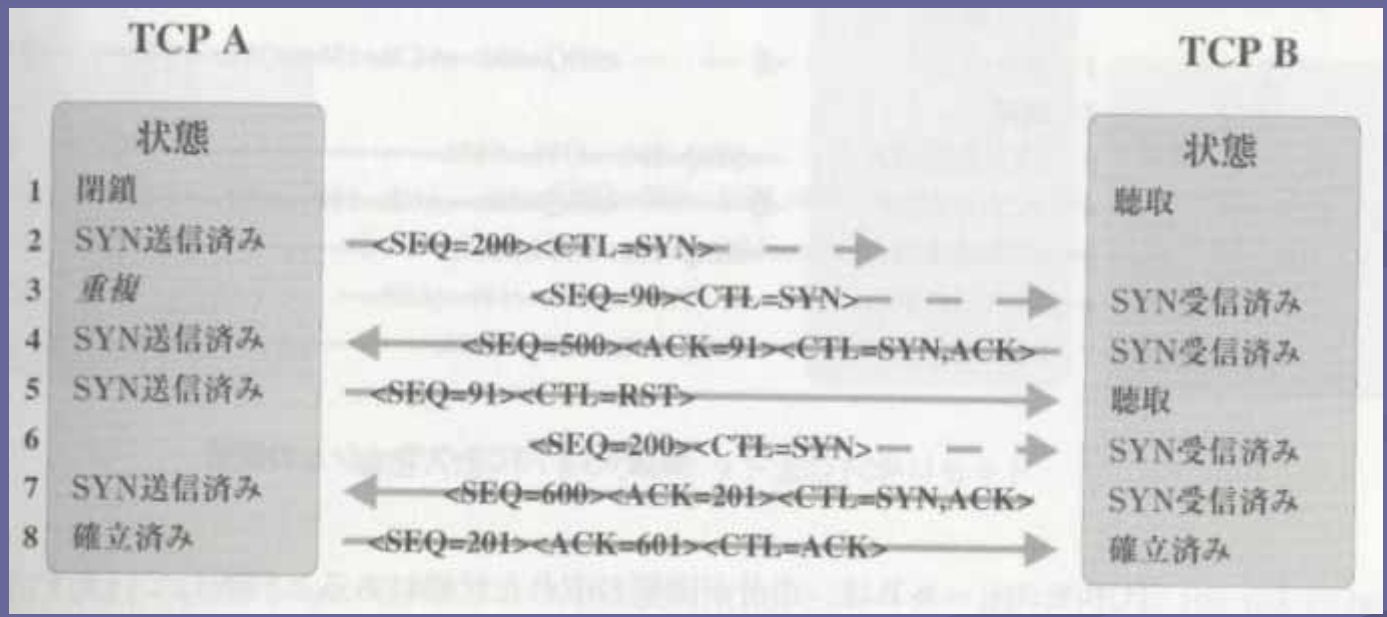
コネクション管理

- 同時にコネクションを開始しようとした場合



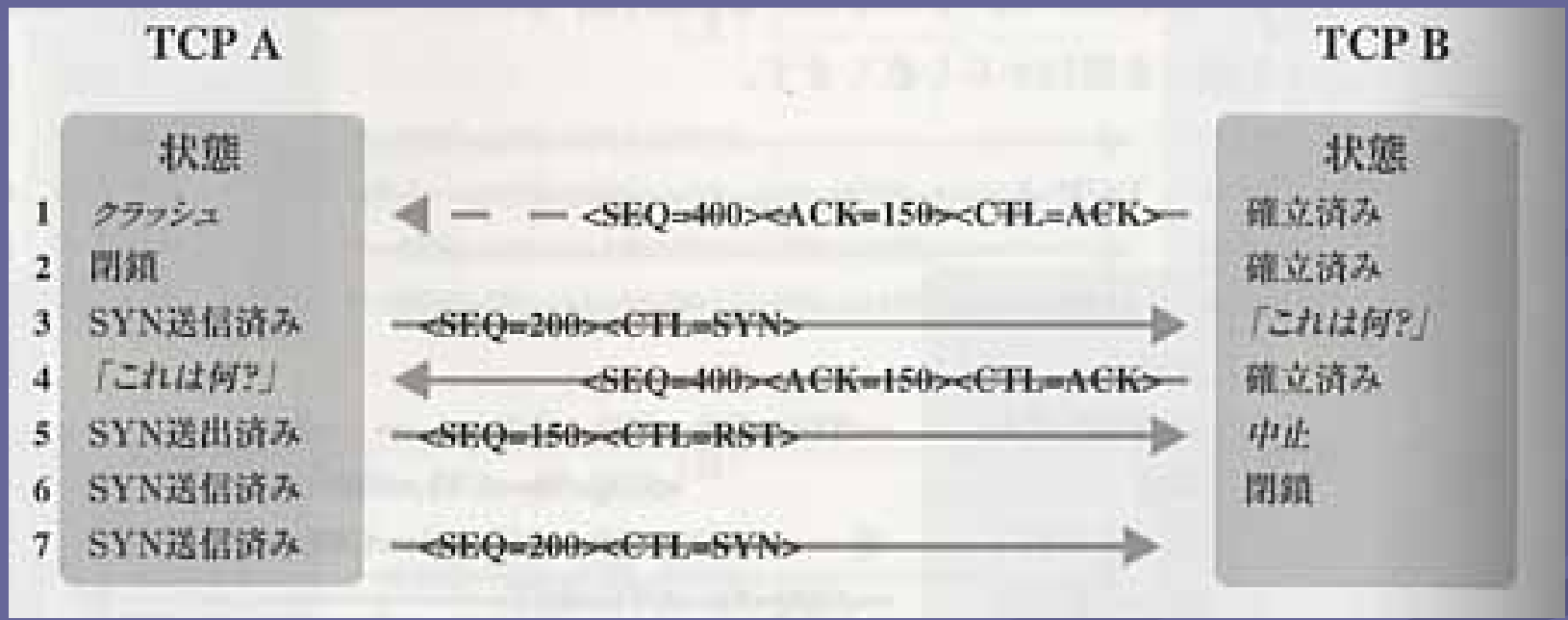
コネクション管理

- SYNフラグを設定した重複セグメントが確立段階の途中で受信された場合



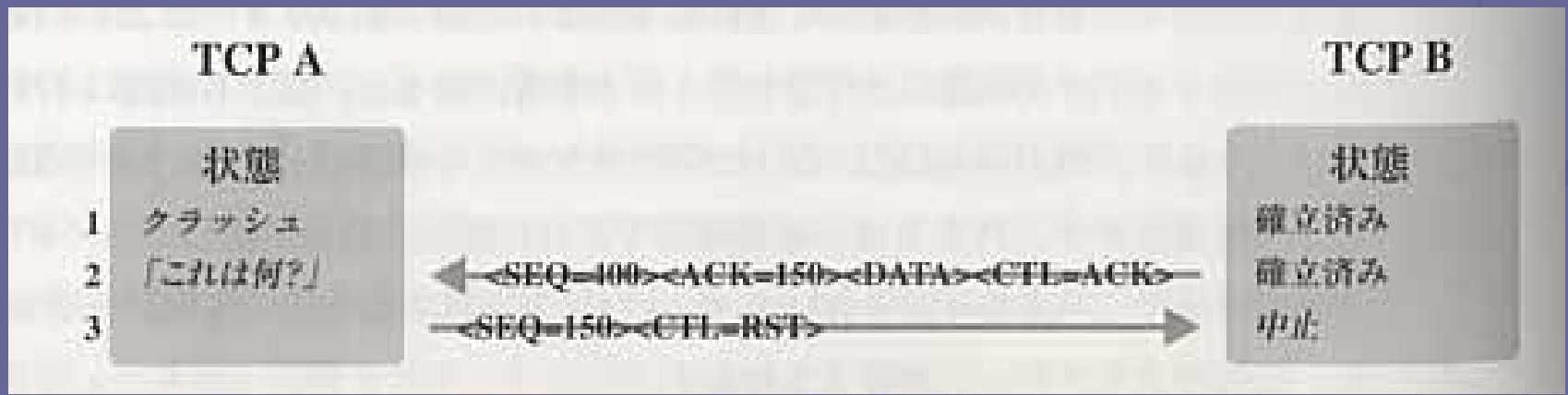
コネクション管理

- クラッシュした場合



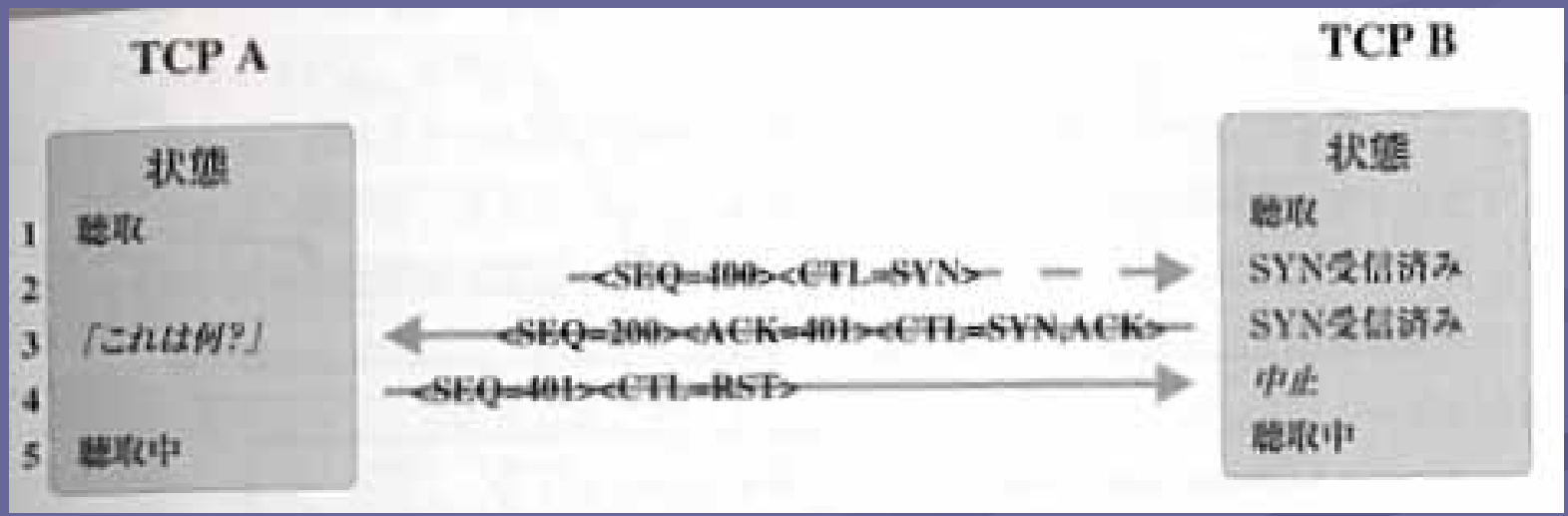
コネクション管理

- アクティブ側がリセットの原因になる場合



コネクション管理

- 両方がパッシブコネクションで開かれており、SYNセグメントを聴取している場合



コネクション維持

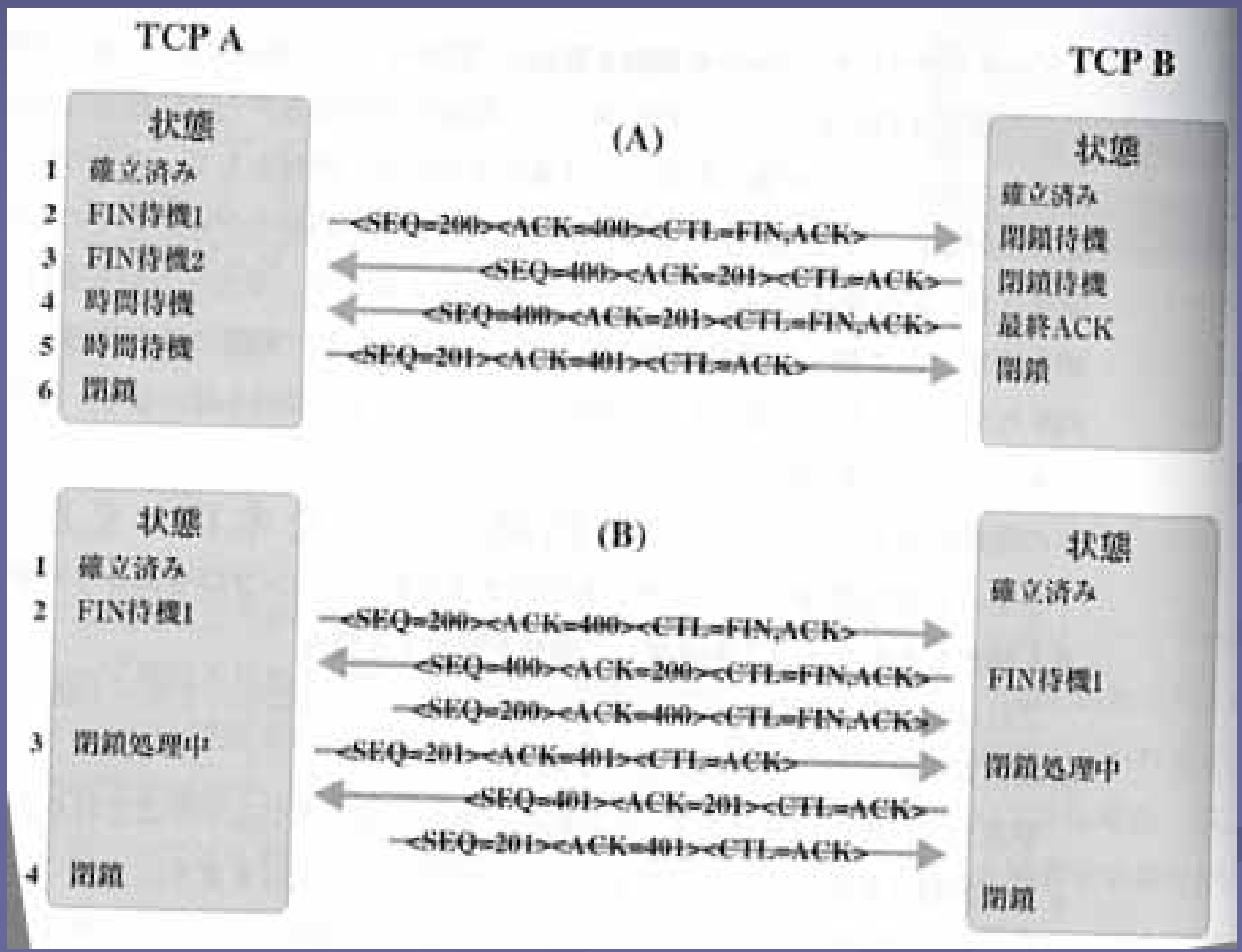
- コネクションは転送されたデータの各オクテットの応答確認によって維持される
- ネットワークの故障などによってセグメントの消失、破損したセグメントを受信した場合、タイムアウト期間の経過後に再送される。
- 再送によって重複セグメントが生じた場合はシーケンス番号と応答確認番号を確認して破棄しなくてはならない。
- 送信ホストは次に使うシーケンス番号を注意深く決めなくてはならない。受信側が次に受け取るシーケンス番号を覚えているからである。同様に送信側は送信したが応答がないシーケンス番号の最も古いものを覚えている。
- コネクションが遊休の状態の場合これらの変数はすべて同じになる

コネクション終了

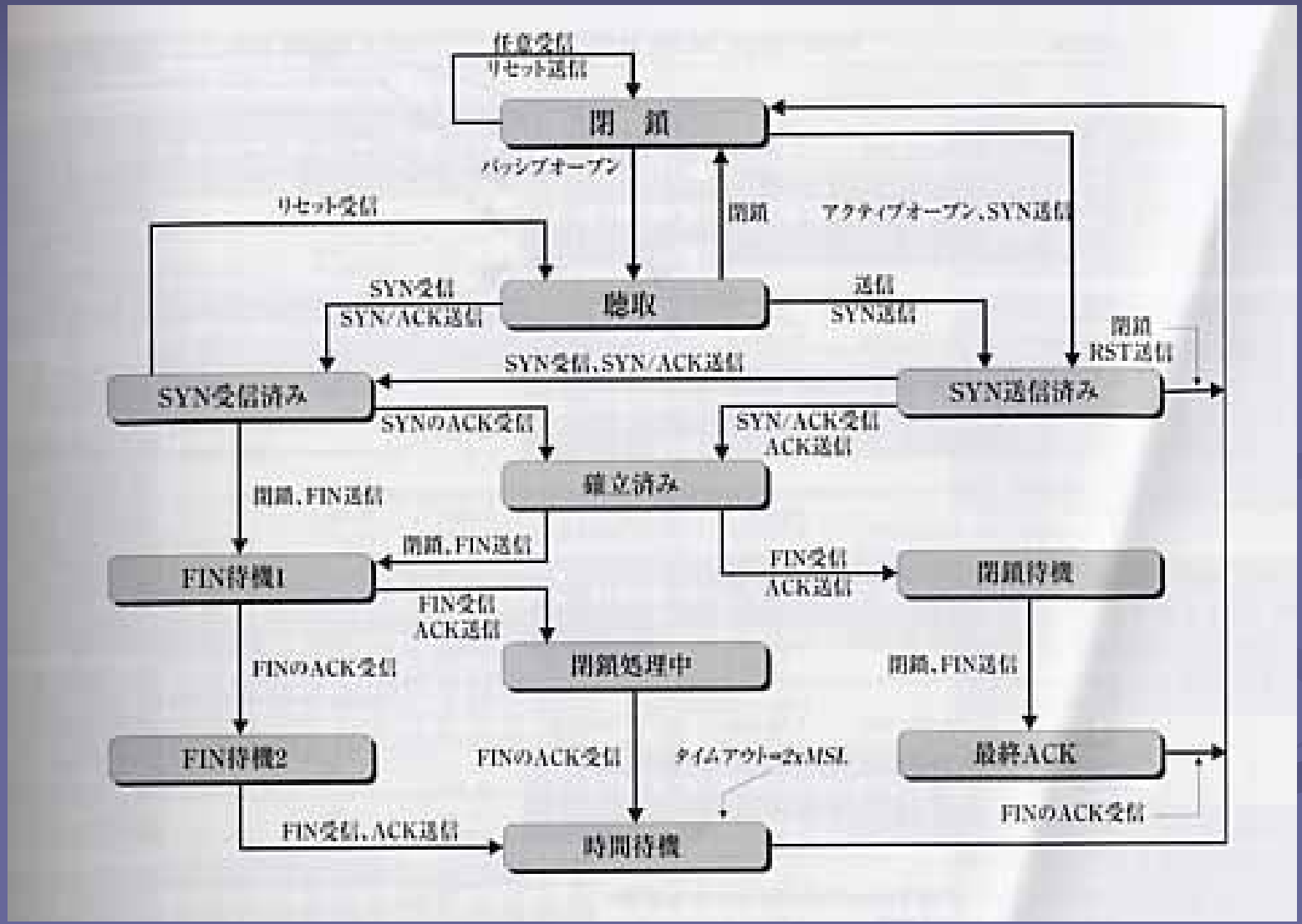
コネクションが終了されるのは大きく分けて次の3つである

- ユーザアプリケーションが閉鎖を開始し、TCPモジュールにコネクションの閉鎖を行わせる場合
- リモートのTCPモジュールが自分のアプリケーションプロセスからの要求により、FINフラグを設定したセグメントを送信して閉鎖を開始する場合
- コネクションの両ホストが同時に閉鎖を開始しようとする場合

コネクション終了


TCP A
TCP B
状態
1 確立済み
2 FIN待機1
3 FIN待機2
4 時間待機
5 時間待機
6 閉鎖
(A)
<SEQ=200><ACK=400><CTL=FIN,ACK>
<SEQ=400><ACK=201><CTL=ACK>
<SEQ=400><ACK=201><CTL=FIN,ACK>
<SEQ=201><ACK=401><CTL=ACK>
状態
確立済み
閉鎖待機
閉鎖待機
最終ACK
閉鎖
状態
1 確立済み
2 FIN待機1
3 閉鎖処理中
4 閉鎖
(B)
<SEQ=200><ACK=400><CTL=FIN,ACK>
<SEQ=400><ACK=200><CTL=FIN,ACK>
<SEQ=200><ACK=400><CTL=FIN,ACK>
<SEQ=201><ACK=401><CTL=ACK>
<SEQ=401><ACK=201><CTL=ACK>
<SEQ=201><ACK=401><CTL=ACK>
状態
確立済み
FIN待機1
閉鎖処理中
閉鎖

TCPの遷移状態


任意受信
リセット送信
閉鎖
パッシブオープン
リセット受信
閉鎖
アクティブオープン、SYN送信
SYN受信
SYN/ACK送信
聴取
送信
SYN送信
閉鎖
RST送信
SYN受信、SYN/ACK送信
SYN受信済み
SYN送信済み
SYNのACK受信
SYN/ACK受信
ACK送信
確立済み
閉鎖、FIN送信
閉鎖、FIN送信
FIN待機1
FIN受信
ACK送信
FIN受信
ACK送信
閉鎖待機
FINのACK受信
閉鎖処理中
閉鎖、FIN送信
FIN待機2
FINのACK受信
タイムアウト=2xMSL
最終ACK
FINのACK受信
FIN受信、ACK送信
時間待機