

第 4 章

タイマシステムとその他の諸問題

第3章までは、TCPのプロトコルとしての特徴とデータ転送の実際を見てきました。本章では、未解決のいくつかの問題を取り上げます。

前半ではTCPのタイマシステムとタイムアウト時間に関する問題点を取り上げ、後半では残されたいくつかの問題点について述べます。特に後半は、新しく RFC となった TCP に関するいくつかの話題を取り上げます。

本章の中でもっとも重要な項目は、データの再転送を決定するための情報となるタイマシステムとタイムアウト時間に対する考察です。

4.1 TCPのタイマシステム

TCPは、データ転送時やアイドル時も含めて、いくつかのタイマシステムを持っています。例えば、第3章で述べたゼロウィンドウ検査では、ウィンドウが拡大したかどうかを定期的に問い合わせることを示しました。もっとも重要なタイマはデータ転送時に起動されるもので、このタイマがタイムアウトすることでデータの再転送が決定されます。

TCPが持つタイマシステムの中で特に重要なものは次の4つと考えられます。

- ① **タイムアウトタイマ**：データ転送時に設定されて、データ転送に対するタイムアウトを検出する
- ② **持続タイマ**：継続的に起動されているタイマで、このタイマを利用してゼロウィンドウ検査などが行われる
- ③ **終了時タイマ**：TIME-WAITからCLOSED状態に移移するための待ち時間のタイマで、第2章で示したように2MSLの時間(多くの実装では2分間)待ってからCLOSED状態に移移する
- ④ **キープアライブタイマ**：TCPの無応答時間に対するタイマ

これらの中で④キープアライブタイマに関しては、それが実装されているかどうかについては実装系に依存しています。不幸にして、インターネットアプリケーションの多くは、自分の接続相手先からの通信がな

くなくても接続が解放されるまでじっと待ち続けます。トランスポート層のプロトコルであるTCPも例外ではありません。

そのため、場合によっては、長時間にわたる無応答状態でもTCPのコネクションが解放されない場合があります。このような状態ははなはだ都合が悪いものとなります。例えば、コネクションの片方のエンドポイントが何らかの理由でクラッシュし、ハーフオープン状態となったとします。このようなハーフオープン状態のコネクションが多数発生すると、場合によってはそれ以上の新しいコネクションを受け付けることができなくなるかもしれません。

そのために、キープアライブタイマは多くの実装系において採用されています。ただし、キープアライブタイマを使うかどうかはTCPの規定にはありません。実装されているタイマを使うかどうかは、アプリケーションプログラムの設計者に任されています。

タイムアウトタイマについては、次節以降に説明します。ここでは、他の3つのタイマについて簡単に概観しておきます。

4.1.1 持続タイマ

既に述べたように、持続タイマは受信側からゼロウィンドウサイズを告知された場合に起動されます。ゼロウィンドウサイズが告知されると、送信側はウィンドウサイズをゼロとしそれ以上のTCPセグメントの送出を止めます。そして、受信側のバッファに空きができてウィンドウサイズが拡大するのを待ちます。通常は、受信側からゼロ以外のウィンドウサイズが送られてきたところでウィンドウサイズが拡大し、持続タイマも当然リセットされます。

一方、何らかの理由で受信側からの新たなウィンドウサイズの告知が送られてこなければ、持続タイマは切れます。そして、送信側はゼロウィンドウ検査を行います。ここで、受信側からの応答(たとえそれが、ウィンドウサイズがまだゼロであるということを告げるものであっても)が返ると、持続タイマはリセットされます。

タイマのタイムアウト時間は、通常のTCPのタイマバックオフ戦略(次節参照)に従います。

4.1.2 終了時タイマ

終了時タイマとは、TCPがコネクションを確立する場合に、TIME-WAIT状態からCLOSED状態への遷移を制御するためのタイマシステムです。

通常は、TIME WAIT状態に遷移した直後に起動され、2MSLだけ待ってCLOSED状態に遷移し、コネクションが完全に解放されます。

4.1.3 キープアライブタイマ

キープアライブタイマは、TCPの標準仕様では何も規定されていません。しかしながら、現実的な問題として無応答のコネクションを強制的に切断する必要が生じた場合に、キープアライブタイマが実装されていないとTCPが能動的にコネクションを解放する手段がなくなってしまいます。そのための弊害は既に表示したとおりです。