

第 5 章

TCPの実装モデル

これまでは、TCPのプロトコルを中心に解説しました。その結果、TCPは多くの重要な機構を含んでおり、プロトコルそのものが複雑であることがわかりました。

TCPの実装では、それらの複雑に絡み合った機構を整理して、一つのプロトコルとして機能するように組み上げる必要があります。そこで第5章では、TCPの実装を考えるうえでのポイントとなるモデルを提示します。

5.1

入出力モデル

TCPの実装を考える前に、TCPの役割とそのためにより必要となるOSとの折衝について再度考察する必要があります。このためには、TCPの入力と出力、およびそのためのデータ構造を把握することが重要となります。そこで、本章の最初では、TCPとUDPがIPを使ってネットワークに対してどのような入出力を行うのかについて見ておきます。

5.1.1 ネットワークへのデータ出力

TCPは、上位のアプリケーションプロトコルとプロトコルポートを介して結び付けられます。例えば、上位のプロトコルからネットワークへの出力が要求されると、図5.1に示したように、ポートごとに用意されたバッファに置かれます。

TCPのプロセス(以下、これを「TCPプロセス」と呼ぶ)は、これらのバッファからIPへ出力するためのデータを作成し、IPのプロセス(以下、「IPプロセス」と呼ぶ)へ送ります。データを受け取ったIPは、そのデータをデータグラムにカプセル化して、ネットワークのデバイスへと送り出します。

このとき、TCPプロセスはプロトコルポートを使い、複数のアプリケーションにポートごとのバッファをつなぐことで、アプリケーションプログラムに対するマルチプレクシングを行います。

一方、IPプロセスはIPのキューに置かれたデータを順次取り込んで、IPデータグラムを作成し、デバイスごとの出力キュー(図では「出力パケットのキュー」)に置かれます。

このとき、複数のネットワークデバイスがあるとルーティングによって決定されたネットワークデバイス

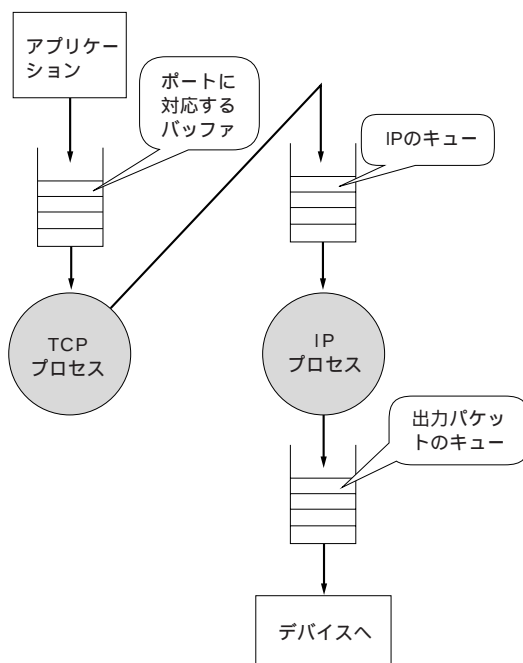


図5.1 ネットワークへの出力

のキューに出力されます(たとえそれがマルチホームホストであり、IPデータグラムのフォワーディングが行われなくても、IPプロセスはどのネットワークデバイスに出力するかを決めるためにルーティングテーブルを参照する)。

5.1.2 ネットワークからのデータ入力

一方、ネットワークから到着したデータは、各ネットワークデバイスの入力バッファに置かれます。IPプロセスは、これらの入力バッファから取り出したパケットからIPデータグラムを抽出します。そして、上位層のプロトコルによって必要なキュー上に置きます。例えば、図5.2は上位層としてTCPを考えた場合のモデルです。

すでに示したように、TCPではコネクションをエンドポイントの組として管理しているので、TCPプロセスへの入力キューは、セグメントに対するポートごとに存在すると考えられます。

TCPプロセスは、ポートごとのキューから取り出したデータをコネクションに関連付けられた入力バッファに置きます。各コネクションのエンドポイントに位置するアプリケーションは、該当するバッファからデータを取り出します。

5.1.3 タイマプロセス

TCPがネットワークにデータを出力する場合、必ずタイマを起動します。起動されたタイマは、タイムアウト時間内にACKが返って来るとクリアされます。そのため、入力されたデータにACKが含まれていた