

さて、このコラムもようやく7回目ということで、1年の半分以上を過ぎました。会誌にふさわしいコラムの話題を考えるのに、毎月悩んでいますので、いいものがあればお寄せください。今回はインターネットのアプリケーションプログラムが動作するのに欠かせないTCPの話です。

TCP (Transmission Control Protocol)¹⁾ は、インターネット上で信頼性ある通信を実現するのに欠かせないインターネット標準プロトコルです。基盤となっているネットワークプロトコルのIPは、よくご存知のように、信頼性のない、コネクションレス型のパケット配送プロトコルです。これを使って、ネットワークプログラム、アプリケーションを構築するには、通信誤り、パケット順序の入れ換え、パケットの重複、性能向上といったさまざまな要素を考慮する必要があります。このような考慮をすべてのアプリケーションプログラマが扱っていたのでは大変です。そこで、当初から信頼性のあるストリーム指向のプロトコルが開発され、実用に供されています。それがインターネット標準であるTCPです。

TCPのインターネット標準を規定するRFCは85ページもあり、それに加えて、さまざまな改良を規定するRFCが出されており、全容を理解するのは大変です。RFC793で示されていない実装の細部に関していくつかは、RFC1122²⁾ で述べられています。で、標準っていうくらいだから、それで全部なのかというと、そうでないのが発展し続けるインターネットの世界のいいところです。TCPの改良が、職人芸のように続けられてきており、非常に多様な通信媒体が使われるインターネットで長年のあいだ使われているの

です。

TCPを非常に簡単に表現してしまえば、再転送付き肯定確認応答プロトコルです。送信側は、データを送るとき何バイト目まで送ったかを、シーケンス番号として覚えておき、受信側は、受信に成功したとき、肯定確認応答として、何バイト目まで受信したかのシーケンス番号を送信側に返します。一定時間内に確認応答が返ってこなければ、同じデータを定められたアルゴリズムで計算された時間をおいた後で再送します。TCPの性能は、この再転送時の時間間隔や、一度に何バイト送るかを定めるウィンドウサイズといったものに大きく影響されます。したがって、この再転送間隔の決定アルゴリズムや、ウィンドウサイズの調

整、確認応答の出し方といった要因に関して、非常に多くの提案がなされ、シミュレーション結果や実装結果が論文になっています。そうした実装を元に議論が重ねられ、いくつかがInternet-Draftになり、そのうちのいくつかがRFCとなり、さらにいくつかがインターネット標準になってきました^{3)~5)}。

歴史的には、56kbpsから1.5Mbpsくらいの広域ネットワークと10MbpsくらいのLANでの利用が多かったわけですが、通信媒体の多様化と高速化により、TCPに課せられた仕事も複雑にならざるを得ません。たとえば、TCPのシーケンス番号は32ビットですが、ギガビットクラスのネットワークになって、これが1周して溢れたらどうなるのかとか、衛星通信や大陸間を結ぶ回線のように、遅延は比較的大きいが帯域幅も大きい回線の場合、ウィンドウサイズを大きくしないと思ったように転送速度が増えません⁶⁾。また、低速のモデムで公衆網を使って接続する場合、TELNETで1文字入力するごとに、IPヘッダとTCPヘッダ合わせて40バイトものオーバーヘッドがあるのはたまりません。これらに関しても、そのときどきに、解決方法が提案されてきました。

たとえば、CSLIP⁷⁾ と呼ばれる仕組みが、低速回線で使うTCPのために研究されました。基本的なアイデアは、20バイトのIPヘッダと20バイトのTCPヘッダのうち約半分は、あるTCPコネクションについて変化しないことを利用するものです。もしシリアル回線の両端のノードが、コネクションに関する状態情報を記憶しておけば、TCPのパケットごとに40バイトのIP/TCPヘッダを送らずにその半分と1バイト程度のコネクシ

ョン番号を送って済ませられます。さらに、IPのパケット長やヘッダチェックサムも、データリンク層の情報や、リンクの両端での再計算によって実は省略でき、16バイトくらいになります。さらに、FTPのデータ転送などで実際に変化するの、パケットID、チェックサム、そしてシーケンス番号だけで、シーケンス番号も多くの場合1ずつしか増えないので、4バイトも使うことはありません。というわけで、CSLIPは、ヘッダを5バイトないし6バイトにすることを目標としています。このような職人芸的な改良がいろいろなところで行われています。

CSLIPは主としてモデムを使って接続する末端の低速回線でのTCPの性能向上をねらったものですが、インターネット全体を見た場合には、輻輳制御が重要です。インターネット上を流れるトラフィックが増え、関係する送信者と受信者の数が増え、ネットワークの振舞いが複雑になるにつれ、ますます輻輳制御は難しくなります。ネットワークが混雑していなければ、遅延時間と帯域に合わせて、TCPは一定間隔でパケットを出していけば、理論値に近い回線の利用効率が得られます。事情を複雑にしているのは、混雑によってパケットが落ち、そのため再転送までの適切な待ち時間の予測が困難になることです。この輻輳を回避するための技術と、いったん輻輳状態になって、その後回復するための技術がTCPには組み込まれています。最も基本的なものがSlow-Startと呼ばれているもので⁴⁾、回線の混み具合に適応して、適切な送信速度に達する工夫の1つです。TCP Vegas¹⁰⁾や、ECN (Explicit Congestion Notification)¹²⁾をはじめとする比較的新しい研究も数多くあり、評価も定

まりつつあります。また、TCPを用いる通信の両端だけでなく、中間ノードがどのように輻輳制御に介入できるかの研究もあり、混みそうになったら適当にパケットを落すRED (Random Early Detection) と呼ばれる中間ノードのパケット待ち行列 (queue) が溢れる前にランダムにパケットを落して終端ノードに輻輳を気づかせるような機構⁹⁾など、さまざまな解析や工夫が提案されています¹¹⁾。

また、無線によるモバイルコンピューティングが普及するにつれ、遅延変動が大きな回線や、不安定な回線を利用するためのTCPも必要で、いろいろな研究がなされています⁸⁾。

WWWで用いられるHTTPのような比較的短く転送量も小さなセッションに対応して、シーケンス番号の初期値の交換など、TCPの初期状態を確立するための3ウェイハンドシェイクを省略したTransaction TCP (T/TCP) のような研究もあります¹⁴⁾。適切なウインドウなどをあらかじめ予測できるような他のTCPコネクションとパラメータを制御情報の記憶個所で共有するとどうなるかという研究もあります¹³⁾。

このように、単に再転送付き肯定確認応答プロトコルといっても、さまざまな改良と新しい提案がなされているわけです。幸いFreeBSDやNetBSD、Linuxといったソースコードが利用可能なオペレーティングシステムが利用しやすい時代ですので、実装するもよし、またネットワークシミュレータで性能評価するもよしです。

—VanとSallyを知らなきゃもぐり—

参考文献

- 1) Postel, J.: RFC793 Transmission Control Protocol (Sep. 1981).
 - 2) Braden, R.: RFC1122 Requirements for Internet Hosts—Communication Layers (Oct. 1989).
 - 3) Nagle, J.: RFC896 Congestion Control in IP/TCP Internetworks (Jan. 1984).
 - 4) Jacobson, V.: Congestion Avoidance and Control, ACM SIGCOMM '88 (Aug. 1988).
 - 5) Karn, P. and Partridge, C.: Improving Round-Trip Time Estimates in Reliable Transport Protocol, ACM SIGCOMM '87 (Aug. 1987).
 - 6) Fox, R.: RFC1106 TCP Big Window and NAK Options (June 1989).
 - 7) Jacobson, V.: RFC1144 Compressing TCP/IP Headers for Low-speed Serial Links (Feb. 1990).
 - 8) Brown, K. and Singh, S.: M-TCP: TCP for Mobile Cellular Networks, ACM Computer Communication Review, Vol.27, No.5 (Oct. 1997).
 - 9) Lin, D. and Morris, R.: Dynamics of Random Early Detection, ACM SIGCOMM '97 (Sep. 1997).
 - 10) Brakmo, L. S., O'Malley, S. and Peterson, L.: TCP Vegas: New Techniques for Congestion Detection and Avoidance, ACM SIGCOMM '94 (Aug. 1994).
 - 11) Paxson, V.: End-to-End Internet Packet Dynamics, ACM SIGCOMM '97 (Sep. 1997).
 - 12) Floyd, S.: TCP and Explicit Congestion Notification, ACM Computer Communication Review, Vol.24, No.5 (Oct. 1994).
 - 13) Touch, J.: RFC2140 TCP Control Block Interdependence (Apr. 1997).
 - 14) Braden, R.: RFC1644 T/TCP—TCP Extensions for Transactions Functional Specification (July 1994).
- (平成10年8月24日受付)

