

高速通信プロトコルを実現する

1E-7

トランスポートライブラリの実装と評価

三宅 優 加藤 聰彦 鈴木 健二

国際電信電話株式会社 研究所

1. はじめに

近年ネットワークの伝送速度が高速化されるに伴い、高機能かつ高速な通信プロトコルの機能や実装方法の検討が重要となっている。筆者らは、 x -kernelを用いてXTP^[1]の実装を行い、実装したXTPプログラムが伝送遅延の有無にかかわらず高いスループットを達成可能であることを確認している^[2]。しかし、このプログラムは筆者らの開発した通信ボードや特定のOSを持つワークステーションでしか動作しない。そこで、多機種のワークステーション上で高速通信を実現するために、本プログラムをベースとしたトランスポートライブラリを作成した。ライブラリ形式を採用することによりアプリケーションに適したデータ転送アルゴリズムが、容易に実装可能となるとともに、一般的なUNIXが提供している関数のみを使用することにより、他のUNIXベースのOSへの移植が容易となっている。これまでに、本ライブラリを実装し、その性能評価を行い、さらにFTPに組み込んでその有用性を実証したので、本稿では、それらの結果について報告する。

2. トランスポートライブラリの実装概要

2.1 プロトコル構成

図1に、トランスポートライブラリを使用して通信を行う場合のプロトコル構成を示す。トランスポートライブラリは、UNIXをベースとするOS上で動作し、アプリケーションとリンクして使用され、シーケンス番号の管理、フロー制御、誤り制御等の処理を行う。また、本ライブラリは、カーネル内のUDPを利用して通信する。このため、受信したデータのコネクションとの対応付けは、UDPのポートを使用してカーネル内で実行される、IPにより到達可能な範囲であれば相互通信できる、等の利点を持つ。

2.2 通信プロトコル仕様

プロトコル仕様に関しては、基本的にXTP^[1]をベースにしている。但し、明示的なコネクションフェーズを設け、アプリケーションが利用するバッファサイズや、UDPのバッファサイズ、使用するUDPのポート番号を交渉させることとした。また、ルーティングはIPにより実行されるため、XTPのルーティング処理部は削除した。更に、誤り回復方式として、go-back-Nと選択再送の両方式を使用した。

2.3 プロトコルの実装

トランスポートプロトコルは、高スループットを実現させるとともに、UNIX OSにおけるユーザレベルで動作させるために、以下のような実装方法を採用して

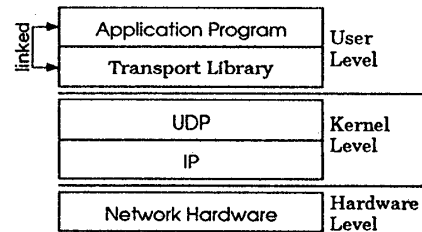


図1: プロトコル構成

いる。

- フロー制御とレート制御のパラメータの変更を協調して行い、輻輳により発生するパケットロスを抑制する。このために、輻輳制御手順に従って決定されるデータ送信側のウィンドウサイズと、観測された往復遅延時間とにより、レート制御の送信レートを定める方法を採用した^[2]。
- アプリケーションが確保したバッファを、そのまま送受信バッファとして利用することにより、データコピーを避ける。このため、送信関数はバッファ全体の受信確認が行われてからリターンすることとし、さらに、通信相手からの受信確認の遅れによりデータを連続して送信できなくなる事態を防ぐために、送信関数の引数として複数のバッファを持たせることを可能とした^[3]。
- ソケットバッファのサイズが64K(53K)バイト以上確保できないOSに対処するため、ライブラリレベルで複数のソケットバッファを確保して仮想的にバッファサイズを拡大することにより、バッファオーバフローを回避する。
- インターバルタイマと、タイマ割り込みを利用して各タイマーの制御を行う。ただし、レート制御のタイマーは、非常に間隔の短い制御を必要とするため、`gettimeofday()`と`select()`の組合せで実現する^[3]。

3. 性能評価

3.1 評価方法

図2に、性能評価に使用したシステムの構成を示す。使用したワークステーションは、SunOS4.1.2が動作するSPARCserver 670MP(SPARC CPU 40MHz x 2)と、Solaris2.3が動作するSPARCstation 20(SuperSPARC CPU 60MHz x 2)である。これらを、140Mbpsの伝送速度を持つATM LANに接続し、ワークステーション間の経路上に、遅延およびエラーを挿入可能でDS-3(45Mbps)のインタフェースを持つ装置を接続した。伝送路のボトルネックは、DS-3の45Mbpsであり、フレーミングおよびATMセルのヘッダのオーバーヘッドにより、

"Implementation and Evaluation of Transport Library for High Speed Communication"

Yutaka MIYAKE, Toshihiko KATO and Kenji SUZUKI
KDD R & D Laboratories

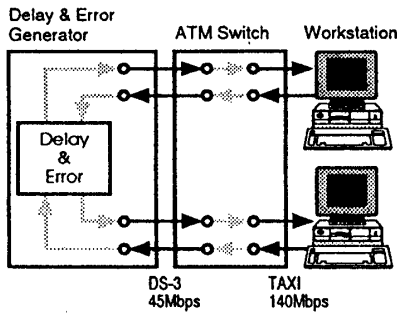


図 2: 評価システムの構成

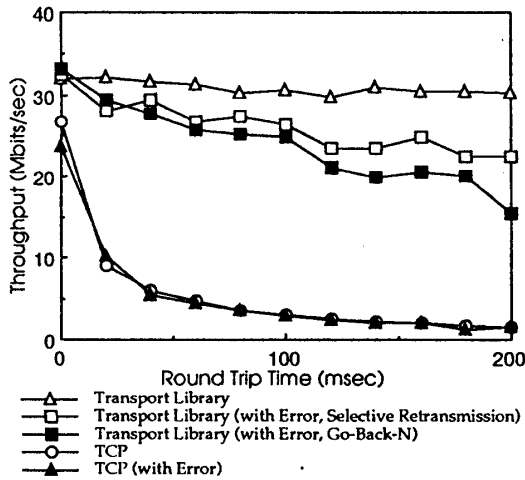


図 3: スループットと往復遅延時間の関係

ATM のペイロードで運ばれるデータの理論的な最大伝送速度は約 36Mbps である。

評価は、伝送路の往復遅延時間を 0ms から 200ms の間で変化させ、ワークステーション間でデータを連続して転送した場合のスループットを測定した。この測定においては、50K バイトのサイズの UDP ソケットバッファを 4 つ使用し、送信パケットサイズは 8192 バイトとした。さらに、エラーと誤り回復方式の関係を調べるために、 1.0×10^{-8} のランダムビットエラーを挿入した場合のスループットも計測した。

比較のために、OS 内に実装されている TCP のスループットを、スループット計測用ソフトである `ttcp` を用いて計測した。このときのソケットバッファサイズは 50K バイト、パケット長は 16384 バイトである。

3.2 評価結果

図 3 に測定結果を示す。トランスポートライブラリのスループットは、遅延に関係なく、約 32Mbps の速度を維持している。特に遅延の影響を受けることなく、つねに OS に実装されている TCP のスループットを上回っている。また、この結果は、実装形態を `x-kernel` 上からライブラリ形式へと変更したが、性能を劣化させることなく実装を行うことができたことを示している。

エラーを挿入した場合は、往復遅延時間の増加に従ってスループットが低下している。これは、遅延時間により、再送要求を通信相手に通知するのに時間がかかり、誤り回復機構を動作させるのが遅れるためである。

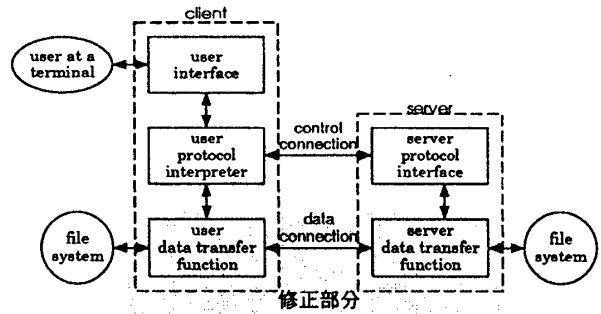


図 4: FTP プログラムの構成と修正部分

遅延	OS 付属の FTP	トランスポートライブラリを使用した FTP
0 ms	16.0 Mbps	19.6 Mbps
100 ms	1.1 Mbps	19.0 Mbps
200 ms	0.6 Mbps	18.0 Mbps

表 1: FTP へ組み込んだ場合のスループット

4. FTP への組み込み

作成したトランスポートライブラリが一般のアプリケーションで利用可能であることを実証するために、ファイル転送プログラムとして広く利用されている FTP への組み込みを行った。図 4 に FTP プログラムの構成と、組み込むために修正を行った部分を示す。FTP の通信では、ユーザからの要求にしたがって制御を行うためのコントロール・コネクションと、ファイルやディレクトリ情報を転送するためのデータ・コネクションが分離されている。そこで、ファイル転送のスループットが影響するデータ・コネクション部分の通信プロトコルを、TCP からトランスポートライブラリを利用する形式に変更した。なおこの変更には、フリーソフトウェアの FTP プログラム (クライアント側は `ncftp`、サーバ側は `wu-ftpd`) を使用した。

表 1 に、OS 付属の FTP とトランスポートライブラリを利用した FTP のスループット比較を示す。ファイル I/O の制限等により、メモリ間転送の場合よりスループットが低下しているが、遅延の無い場合でも、TCP に比べ約 20% のスループット向上を達成するとともに、遅延の増加によるスループット低下も小さいことを確認した。

5. まとめ

本稿では、高速通信を実現する通信プロトコルをライブラリ形式で実装したトランスポートライブラリの実装と評価について述べた。今後は、様々なネットワーク環境で評価を行うとともに、本ライブラリを使用したアプリケーションの開発を行っていく。最後に日頃御指導頂く KDD 研究所 浦野所長に感謝します。

参考文献

- [1] "XTP Protocol Definition Revision 3.6," Protocol Engines Inc., PEI 92-10, 11 January 1992.
- [2] 三宅、加藤、鈴木、"x-kernel 上へ実装した XTP プログラムの性能評価," 第 50 回情報学大会, 1T-7, March 1995.
- [3] Y. Miyake, T. Toshihiko, K. Suzuki, "Implementation Method of High Speed Protocol as Transport Library", 1995 International Conference on Network Protocols, November 1995. (To appear.)