4N－4　細粒度の並列処理を用いた通信プロトコルハードウェアの実装方式に関する検討

佐藤 友実† 加藤 聰彦†† 鈴木 健二††
†電気通信大学 †† 国際電信電話(株)研究所

1. はじめに

近年、超高速ネットワークの導入に伴い、高性能な通信システムの実現が重要となっている。622Mbps の OC-12 回線上の ATM や 800Mbps の HIPPI など、高いスループットが要求される場合には、高位レイヤのプロトコル処理も含めてハードウェアにより実装することが有効であると考えられる[1, 2]。ハードウェアによる実装では、プロトコル処理を細かく分割し、それらを並列に実行させることにより、全体の性能を向上させることが可能である。そこで、プロトコル処理から、粒度の細かい並列処理部分の抽出を検討しつつ、設計を進める必要がある。本稿では、TCP (Transmission Control Protocol) の受信処理を例にとり、1つのレイヤ内のプロトコル処理を、ハードウェアを用いて並列性を考慮しつつ実装する方法について、検討した結果を示す。

2. 実装方針

ハードウェアによる細粒度の並列処理を用いて、通信プロトコルを実装するために以下の方針を用いる。

(1) レイヤ単位、メッセージ単位、1つのレイヤ内の機能単位などの粒度の並列化手法を組み合わせて用いて、プロトコル処理において独立に実行できる部分をできるだけ並列化する[1]。

(2) 各レイヤのプロトコル処理は順に実行される必要がある場合が多いため、1つのメッセージに対する複数のレイヤ処理は順番に実行する。さらに、複数のメッセージに対して、各レイヤの処理をパイプライン的に多重処理を行うことにより全体の性能を向上させる[1]。

(3) 1つのレイヤ内の処理については、独立なパラメータチェックなど、並列に実行可能な部分(並列実行モジュール)をできるだけ抽出し並列度を上げる。その場合、応答パケットの作成など、前もって独立に処理可能な部分については、その処理結果を使用するか否かを判断する前に、先取りして実行する。

(4) レイヤ内の並列実行モジュール間で同期を取る必要がある場合は、専用のシグナルをモジュール間に導入する。また、モジュールが共通に使用する制御情報についてはレジスタに設定し、レジスタへのアクセスのための同期制御は行わないようにする。

3. TCP 受信処理の実装方式の検討

3.1 構成

TCP を処理するハードウェアとして、以下のような構成を想定する。

(1) TCP ハードウェアには、外部から TCP セグメント全体のデータがワード単位に与えられる。与えられた各ワードに対して、チェックサムの検査を行い、それと並行して、状態遷移や出力セグメントの作成などを行う。

(2) TCP ハードウェアの中には、図1に示すように、以下のレジスタを用意する。

- 処理中の TCP パケットのヘッダのパラメータを格納するレジスタ
- IP 受信処理時に識別した IP アドレスおよびパケット長を格納するレジスタ (その値は、TCP 受信処理開始時に IP 処理ハードウェアから与えられていることを想定している)
- コネクション情報を保持するレジスタ (本検討では1本の TCP コネクションを想定している)
- コネクション情報の作業用レジスタ

これらのレジスタはハードウェア内から同期制御なしに参照可能とする。

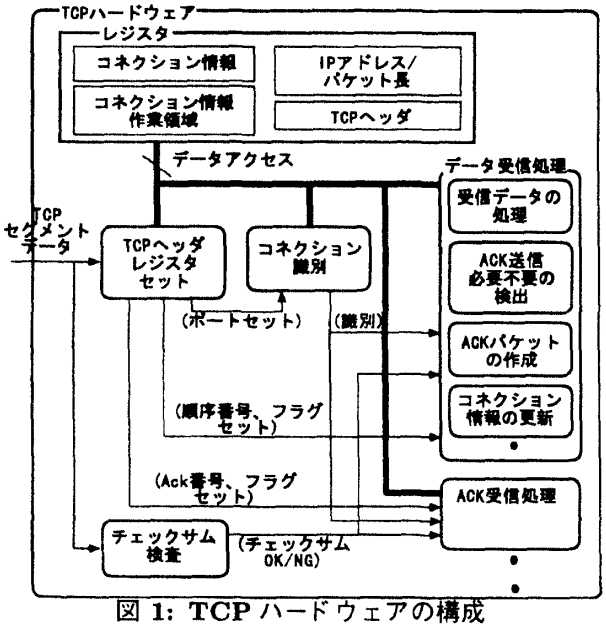

図 1: TCP ハードウェアの構成

(3) 並列実行モジュールとして、図1に示すように、TCP のヘッダ情報を対応するレジスタに設定するモジュール、コネクションを識別するモジュール、データ受信処

"A Study on Hardware Implementation of Communication Protocols by Use of Parallel Processing with Small Granularity"

Tomomi SATO†, Toshihiko KATO†† and Kenji SUZUKI††
†The University of Electro-Communications
††KDD R&D Laboratories

理モジュール、ACK 受信処理モジュール、チェックサム検査モジュールなどが用意される。これらの間には図中のシグナルが定義される。さらに、データ受信処理モジュールは、受信データの処理などのより小さい並列実行モジュールに区分される。

3.2 処理手順

TCP ハードウェアは、以下のような処理手順により実行される。また、その時間的な順序関係は、図2に示す通りである。

(1) TCP ハードウェアに入力されるセグメントのデータは、TCP ヘッダレジスタセットモジュールに与えられる。このモジュールは、TCP ヘッダのパラメータを、対応するレジスタに格納し、以下に示すように、他のモジュールにパラメータをセットしたことをシグナルにより通知する。

(2) (1) と同時にチェックサム検査モジュールにも、セグメントのデータが渡される。このモジュールでは、セグメントのヘッダとユーザデータに対して、チェックサムを計算する。

(3) TCP ヘッダレジスタセットモジュールは、先頭の4バイトを与えられると、発信ポートと宛先ポートのレジスタに格納し、コネクション識別モジュールにポートをセットした旨のシグナルを通知する。

(4) コネクション識別モジュールは、IP アドレスレジスタと TCP ヘッダレジスタ中のポートの情報から、対応するコネクション情報を検索する。検索終了後、データ受信処理、ACK 受信処理モジュールにコネクション識別のシグナルを送る。

(5) TCP ヘッダレジスタセットモジュールは、順序番号とフラグのパラメータ値を設定した後、データ受信処理モジュールにシグナルを送る。

(6) データ受信処理モジュールでは、ヘッダレジスタのセットと、コネクション識別のシグナルを受けると処理を開始する。ここでは、

- コネクション情報の順序番号とヘッダの順序番号の比較による、上位への通知／内部的に保持／廃棄などの受信したデータの処理、
- delayed flag や順序番号の比較などにより、ACK パケットを送出するか否かを決定する処理、
- ACK パケットの作成処理

などが並行して実行される。これらの処理は、コネクション情報の状態に応じて実装される。これらの結果は、受信した TCP セグメントチェックサムの検査の結果を待つために、コネクション情報作業領域に保持される。また、上記のモジュールはそれぞれの処理を終了すると、コネクション情報を更新するモジュールにシグナルを通知する。

(7) (5) と同様に、TCP ヘッダレジスタセットモジュールは、Ack 番号とフラグのパラメータ値を設定した後、ACK 受信処理モジュールにシグナルを送る。ACK 受信処理モジュールでは、レジスタセットとコネクション識別のシグナルにより、(6) と同様に処理を開始する。

(8) チェックサム検査モジュールは、処理を終了すると、データ受信処理および ACK 受信処理のモジュールにシグナルを通知する。

(9) データ受信処理モジュールでは、チェックサムの検査結果のシグナルにより、コネクション情報を更新するモジュールで、パケットの出力などを含む最終的な処理を行う。

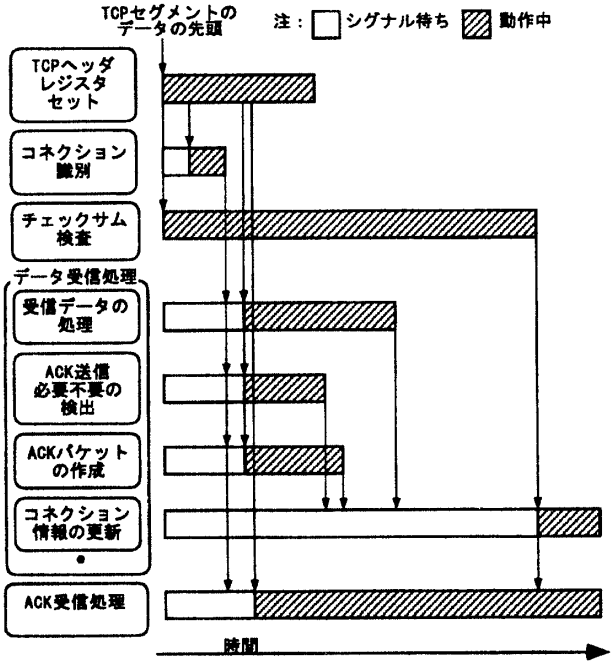

図 2: 処理の時間順序関係

これらの処理においては、TCP ヘッダなどのレジスタに、各時点では、1つのモジュールしかアクセスしないようにした。このため、レジスタに対する同期制御は不必要となっている。このような手順を採用することにより、TCP のプロトコル処理を機能ごとに分け、それぞれを並列に処理することが可能となる。

4. おわりに

本稿では、高位レイヤの通信プロトコルのハードウェア上への実装に対して、TCP のプロトコル受信処理を細粒度の並列処理を用いて実装する方式について検討した結果を示した。本方式により、TCP の処理に高い並列度を導入することができ、高性能な実装が可能となると考えられる。現在、VHDL (VHSIC Hardware Description Language) を用いて、本方式に基づいた TCP ハードウェアの実装を行っている。

参考文献

[1] 加藤, "通信プロトコルのハードウェアによる実装に関する一検討," 情処第 53 回全国大会, March 1996.

[2] 松田, 松田, "高速プロトコル処理のためのハードウェア (SHiPP) の検討," 信学技報 SSE96-10, May 1996.