

通信処理に適したタイマ管理方式の提案

本村 公太

NTT通信網総合研究所

1. はじめに

LANなど下位の伝送速度の向上に伴い、それより上位の通信処理速度の遅さが高速データ転送実現のボトルネックとなっている¹⁾。通信処理に要する時間としては、プロトコルヘッダの処理時間の他に、バッファ管理やタイマ管理にともなう処理時間が多くの割合(80%以上)を占める²⁾。このため、タイマ管理の効率化は通信処理の高速化のための重要な課題となる。

本稿では、通信処理でのタイマの使われ方が管理すべきタイマの総数は多いが種類(動作時間間隔)は限定されている点に着目し、その特徴を利用した効率的なタイマ管理方式を提案する。

2. 通信処理におけるタイマ

通信処理において必要となるタイマとしては、例えば、OSIトランスポートプロトコルクラス4の実装では、少なくとも再送タイマ、レファレンス凍結タイマ、無活動時間監視タイマ、ウインドウタイマの4種類が必要である。これらのタイマの内、再送タイマはプロトコル上TPDU毎に必要であるから、厳密に実装すればウインドウサイズ分の管理が必要となる。また、その他のタイマはコネクション毎に管理する必要がある。

スループットの向上のためにウインドウサイズが比較的大きくとられることや、一つの通信処理プログラムがサポートするコネクション数が大きいことを考えると、同時に管理すべきタイマの総数が100以上になる場合も十分考えられる。

以上のことから、タイマの効率的な管理が重要なこと、また、通信処理におけるタイマの特徴として、管理すべきタイマの総数は非常に多いがその種類は限定(上の例では4)されていることが分かる。

3. タイマ管理の基本機能

通信処理プログラムにおけるタイマ管理の位置付けを図1に示す。タイマ管理は、OSが提供するインターバルタイマ等のOSインタフェースを利用して、通信処理プログラム内のタイマ利用者にタイマ管理サービスを提供する。

タイマ管理で必要となる基本的な機能は以下の3つに分類できる。

(1) タイマ開始

タイマ利用者が、タイマ管理に対して指定時間後にタイムアウトを通知するよう依頼する。

(2) タイマ停止

タイマ利用者が、タイマ管理に対して先にタイマ開始を要求したタイマの停止を依頼する。

(3) タイムアウト検査

タイマ管理がクロック受信(インターバルタイマによる割込み)毎に、開始済みのタイマがタイムアウト時刻に達したかどうかを調べる。

タイムアウトに達したタイマがあった場合、タイマ管理からタイマ利用者に対してタイムアウト通知が行なわれる。

4. タイマ管理の評価基準

タイマ管理を効率的に実現するということは、前節で述べた機能を高速に処理できるように、タイマ管理とタイマ利用者とのサービスインタフェースおよびタイマ管理内のタイマ管理のためのデータ構造を決定することであると言える。

このための評価基準として、個々の機能を実現する際のオーバーヘッドと管理すべきタイマの総数との関係を調べる。これは、タイマの総数が小さいときの各方式でのオーバーヘッドの多少の違いよりも、タイマの総数の増加の影響の方がタイマ管理全体の効率への影響としては大きくなると考えられるからである。

開始されているタイマの総数を n とし、オーバーヘッドを O で表す。このとき、

$O(n)$ ならば、オーバーヘッドは n に比例する

$O(1)$ ならば、オーバーヘッドは n に依らないことを表すものとする。

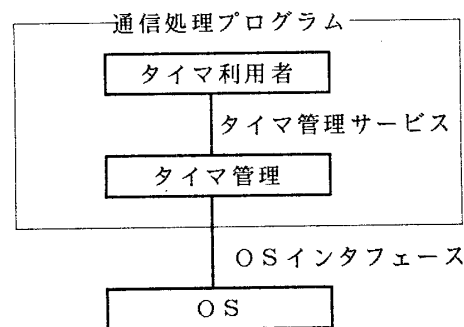


図1 タイマ管理の位置付け

5. 従来の方式

(1) 方式1: 位置固定方式

タイマの開始・停止処理を高速化するためにタイマ生成の手順により予めタイマ管理エレメントの位置を決めておき、それを利用してタイマ管理を行なう³⁾。このときタイマ位置が確定しているのでタイマ開始・停止は $O(1)$ となるが、タイムアウト検査のために各エレメントを一つずつアクセスするのでタイムアウト検査は $O(n)$ となる。

(2) 方式2: 時間順ソート方式

一般にOSで用いられているタイマ管理方式⁴⁾を適用して、タイマ開始の要求を受け取る毎にそのタイマ管理エレメントをタイムアウト時刻順に並んでいるキューの対応位置に挿入する。このとき、平均的にキュー内の半分のエレメントと比較されるとするとタイマ開始は $O(n/2)$ となる。タイマ開始時の応答として、タイマ位置を返しておき、タイマ停止のときにそれを用いることによってタイマ停止は $O(1)$ となる。また、タイムアウト検査では先頭のエレメントを調べるだけで済むので $O(1)$ となる。

6. 通信処理に適したタイマ管理方式

第2節で述べたように通信処理で使用されるタイマの種類は限定されている。そして、タイマの種類が決れば、タイムアウト時刻はタイマ開始要求された順序に並んでいるはずである。この点に着目し、次に述べるタイマ管理方式を提案する(図2参照)。

(1) タイマ生成

タイマ利用者はタイマ管理に対して、タイムアウトまでの時間間隔を指定した論理タイマの生成をタイマ種別毎に要求する。タイマ管理はタイマ利用者を確認として論理タイマIDを返す。

このとき、タイマ管理は論理タイマID毎にテーブル(キュー)を一つ作成する。

(2) タイマ開始

タイマ利用者はタイマ管理に対して、論理タイマIDを指定してタイマ開始を要求し、タイマ管理は開始したタイマを識別するタイマIDを返す。

このときタイマ管理は、論理タイマIDで示されるキューの最後尾にタイマ管理エレメントを繋ぎ、そのポインタをタイマIDとして返す。

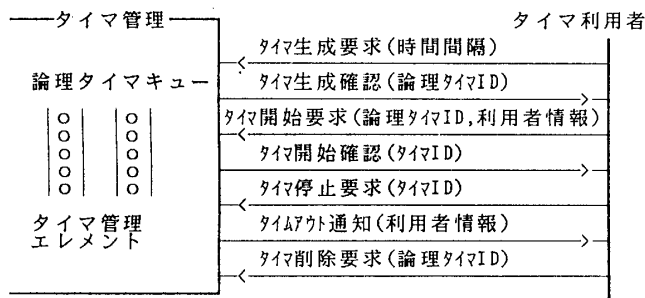


図2 タイマ管理サービスインタフェースとタイマ管理内部のデータ構造

(3) タイマ停止

タイマ利用者はタイマ管理に対して、タイマIDを指定してタイマ停止を要求する。

このときタイマ管理は、タイマIDで示されたタイマ管理エレメントをキューから取り除く。

(4) タイムアウト検査

タイマ管理がクロックを受信毎に、各論理タイマIDに対応したキューの先頭を調べ、タイムアウトに達したものがないかを検査していく。もし先頭エレメントがタイムアウトに達していれば次も調べ、タイムアウトのものがないまで調べていく。

タイムアウトに達したエレメントがあった場合には、タイマ利用者へ通知する。

(6) タイマ削除

タイマ利用者は論理タイマIDを指定して、論理タイマの削除を要求する。

このときタイマ管理は、論理タイマIDに対応したキューを削除する。

本方式を採用した場合、タイマ生成は初期化時にこなせばよいための通信処理中のオーバーヘッドの増加はない。タイマ開始に際しては、論理タイマIDで示されるキューの末尾にエレメントを繋ぐだけであるから、オーバーヘッドは $O(1)$ である。また、タイマ停止は、エレメントへのポインタを使用するので、 $O(1)$ である。タイムアウト検査に関しては、開始済みのタイマ総数 n ではなく、タイマ種類の数 m にほぼ比例するから $O(m)$ となる。

各タイマ管理方式の比較を表1に示す。

通信処理においては、 $m \ll n$ であるから、提案方式のオーバーヘッドは従来の方式に比べてかなり小さく、通信処理を高速化できることが分かる。

表1 タイマ管理方式の比較

	タイマ開始	タイマ停止	タイムアウト検査
方式1	$O(1)$	$O(1)$	$O(n)$
方式2	$O(2/n)$	$O(1)$	$O(1)$
提案方式	$O(1)$	$O(1)$	$O(m)$

n : 開始されているタイマの総数, m : タイマの種類の数

7. おわりに

通信処理におけるタイマの使われ方に着目した効率的なタイマ管理方式を提案し、従来の方式との比較により有効性を示した。

参考文献

- 1) "OSI throughput performance: Breakthrough or bottleneck", Data Communication May 1987
- 2) 永田他:「LANシステムにおけるISOプロトコルインプリメントと処理時間について」, IN86-97
- 3) E.Mumprecht etc.: "TIMERS IN OSI PROTOCOLS - SPECIFICATION VERSUS IMPLEMENTATION", Zurich Seminer 1988
- 4) S.j.Leffler etc.: "The Design and Implementation of the 4.3BSD UNIX Operating System", Addison-Wesley Publishing Company Inc. 1989