

## リアルタイムパケットの送信制御

野村 哲弘<sup>†</sup> 石川 裕<sup>†</sup>

時間制約のあるタスク間の通信における送出制御について、各メッセージパケット毎に時間制約情報を付加してパケット配送時間を保障する方法を提案する。カーネル内で各パケットの送信時間を算出してそれらの時間制約を満たすように送信順序を調整して送出する。時間制約を満たせなくなるパケットについては発信元アプリケーションに例外を通知して破棄する。これらにより送信キュー内で優先度逆転現象を解決する。また、上記機構を Linux 上で Qdisc(キューイング規則) を用いて実装し、様々な時間制約を持つパケットを混在させた環境でもそれぞれの時間制約を満たす通信が行なわれることを示す。

### Transmission Control on Real-time Packets

AKIHIRO NOMURA<sup>†</sup> and YUTAKA ISHIKAWA<sup>†</sup>

In the transmission control of communication between tasks whose packet's delivery time has deadline, a new method, using the delivery time information instead of the priority of task running, is proposed. The transmission order is determined based on the delivery time, in order to guarantee the deadline constraint of each packet. If the packet no longer satisfies its deadline, an exception is delivered to the application and the packet is dropped. In this way, the priority inversion problem is resolved in the transmission queue. This mechanism is implemented in Linux using the Qdisc (queuing discipline) framework. We show that the communication is done with the deadline constraint varies between the packets.

#### 1. はじめに

今日のネットワーク技術の進歩に伴い、リアルタイム通信への需要は日増しに高まってきている。このような中で、殆どのリアルタイム送信制御アルゴリズムでは“通信の優先度は送信タスクの実行優先度に比例している”という前提の元に成り立っている。この仮定は常に正しいものではなく、優先度の逆転現象を発生させてしまうことがある。

本論ではパケットの時間制約自体をそのパケットの送信優先度として扱う手法を提案する。本提案手法は時間制約情報を扱うことに特化しているため、シンプルな待ち行列モデルで構成されている。Linux 上で本提案手法を実装し、時間制約を守って送信が行われていることを示す。

#### 2. 提案手法

パケットの送出時間制約の保障のためには、そのパケットが通過するネットワークインタフェース上の全てのパケットを制御しなければならない。このため、

リアルタイム通信を行なうネットワークインタフェースの物理層の手前にそのインタフェースを通過する全パケットを監視するモジュールを設置する。本モジュールは2つの待ち行列を持ち、片方は時間制約のあるパケットを、もう片方は時間制約の無いパケットを保持する。時間制約付きパケットの待ち行列では、パケットを時間制約順に整列させると同時に制約の充足可能性をチェックする。

時間制約のあるパケットが本モジュールに到着したとき、パケット長とバンド幅から送出にかかる時間を算出する。各パケットの送出時間の和と時間制約情報から時間制約を満たせなくなるパケットが存在しないかを確認する(図1)。パケットの挿入によって時間制約を満たせなくなるパケットが発生する際は発信元アプリケーションに例外を送出し、そのパケットを破棄する。そうでなければモジュール上の優先度付き待ち行列に追加する。

ネットワークインタフェースからパケットを要求されたときには、時間制約のあるパケットを保持しているときはそのうちの一番制約が厳しいパケットを、無い場合は時間制約の無いパケットを送出する。

このようにして常に時間制約を満たしながらパケッ

<sup>†</sup> 東京大学  
The University of Tokyo

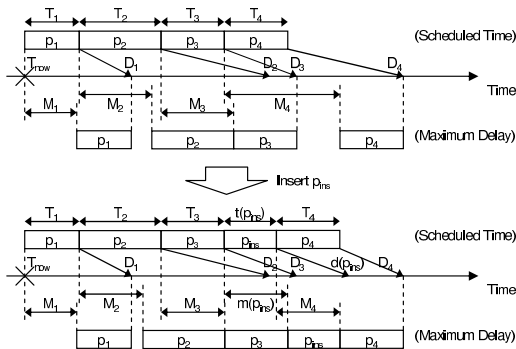


図1 パケット到着時の処理イメージ

Fig.1 Brief image of handling incoming packets

トを送出することが出来る。

### 3. 実装

上記機構を Linux 上で実装する。

#### 3.1 API

リアルタイムパケットをそうでないパケットと区別し、カーネルに時間制約情報を与えるための API が必要となる。しかしながら、新たなシステムコールをシステムに追加すると、本機構を用いていない環境ではプログラムが全く動作しないため、可搬性に欠ける。そこで、新たなシステムコールを使わずに既存のシステムコールで特定の IP オプションヘッダを付けるという形でこの情報を与える。

#### 3.2 パケット制御機構

Linux カーネルには Qdisc(キューイング規則)<sup>1)</sup> と呼ばれるパケット制御機構がある。Qdisc はネットワークインタフェースの入口で通過する全パケットについて優先度付けなどを行うものであり、カーネルモジュールの追加によって新たな Qdisc を導入出来る。Qdisc は Linux カーネルの中で最もデバイスドライバに近い位置にあり、ネットワークデバイスと 1 対 1 の対応が取れ、デバイスの全送信パケットを制御できる。このことから、本提案機構の実装に Qdisc を利用する。なお、Qdisc の後に通ることとなるネットワークインタフェースのドライバ上のバッファを考慮して、送信量を元にバッファの残り容量を推定して送信にかかる時間の計算に反映させることにより、より正確な送信時間の推定を行えるようにする。

### 4. 評価

様々な時間制約を持つパケットを混在させた環境でもそれぞれの時間制約を満たした通信が行なわれることを示す。Intel Pentium 4 3E GHz プロセッサを持つ 2 台のコンピュータ間を Intel E1000(1Gbps) を用

表 1 実験結果 (単位:  $\mu\text{s}$ )

Table 1 Result of Experiment(unit:  $\mu\text{s}$ )

	Task	遅延時間 [ $\mu\text{s}$ ]	
		最大	平均
本機構使用	a	155	66
	b	131	66
本機構未使用	a	13829	12232
	b	13922	13085

Tasks:

- a) 実時間パケット (500 $\mu\text{s}$ ) + 非実時間パケット (バースト)
- b) 実時間パケット (1000 $\mu\text{s}$ ) + 非実時間パケット (バースト)

いて接続し、その間でさまざまな時間制約の組み合わせを持つ通信を行い、パケットの時間制約と送信時間の関係を調べた。

時間制約を持つパケット送信タスクは 20ms 毎に 1 パケットの定期的送信を行う。時間制約の無いパケット送信タスクはバースト通信を行う。表 1 に時間制約が 500 $\mu\text{s}$  および 100 $\mu\text{s}$  の実時間パケット送信タスクと時間制約の無いパケット送信タスクを同時に実行したときの時間制約を持ったパケットの遅延時間を示す。非リアルタイムパケットのバースト転送中においても、本機構によりリアルタイムパケットが指定された時間制約の範囲で送信されることが示された。

### 5. まとめ

リアルタイムパケットの時間制約をそのまま通信の優先度として扱う通信機構とその実装を示した。時間制約を直接扱うことによってその保障をより単純な形で簡便かつ確実に行うことが出来る。

また、Linux カーネル上における本機構の実装を示した。Qdisc 機構によって比較的容易にパケットの送出順制御を行うことが出来る。

この方式において解決すべき問題点として、常に時間制約付きのパケットの送信を優先するため、時間制約の無い通常のパケットに対する帯域保障が為されていないことが挙げられる。また、今後の課題として本実装をリアルタイムスケジューラへと組みこむことによって実際にリアルタイム環境上で本機構によって望まれたリアルタイム通信が行えることを示すことが挙げられる。

謝辞 本研究の一部は、科学技術振興機構 (JST) の戦略的創造研究推進事業 (CREST) の支援を受けた。

### 参考文献

- 1) Hubert, B.: Linux Advanced Routing & Traffic Control HOWTO, <http://lartc.org/>.