3A-2

AnT における NIC ドライバプロセスの実現

岡本 幸大† 乃村 能成‡ 田端 利宏‡ 谷口 秀夫‡

†岡山大学工学部 ‡岡山大学大学院自然科学研究科

1. はじめに

オペレーティングシステム(OS)の信頼性向上のために，デバイスドライバを OS 核内ではなく，プロセスとして構成する手法が提案されている[1][2]．プロセスは OS からの独立性が高いため，デバイスドライバの不具合が OS 全体に及ぼす危険性を低下させることができる．言い換えれば，この手法は，OS 核の管理する資源を最小化し，多くの処理を OS 核外部に構成するということである．このため，デバイスドライバプロセスを実装する場合，マイクロカーネル上での実装が適している，

しかしながら，上記手法は，デバイスドライバと OS 核との連携オーバヘッドが大きくなるため，入出力性能の低下が懸念される．例えば，NIC ドライバでは，OS 核とバッファ管理やプロトコル処理との連携が必要であり，その連携によるオーバヘッドが増大する．

そこで，信頼性と性能のトレードオフから，実装の形態としていくつかの形態が考えられる．本稿では，マイクロカーネル構造をもつ ***AnT*** オペレーティングシステム[3]上の NIC ドライバを取りあげ，２つの形態を実装し，性能面について比較した結果を述べる．

2. 実装

2.1 形態

NIC の制御に関連する処理は，大きく３つに分けることができる．

(1) NIC 制御のためのドライバ部
(2) プロトコル処理部
(3) バッファ管理部

ドライバ部をプロセス化した場合，実装の形態として以下の案がある．

(案1) 全てを個別プロセスとして実装
(案2) 連携の多い組み合わせを 1 つのプロセスとして実装
(案3) ドライバ部のみをプロセスとして実装

マイクロカーネルの利点を確保するために，性能が許す限り，実現形態としては(案1)>(案2)>(案3)の順が望ましい．

そこで，(案1)の場合と，(案2)の一例としてバッファ管理部をライブラリとして実装した場合について考察する．理由として，通信の際，バッファの確保や解放，その他の処理は度々発生するので，バッファ管理はドライバ部やプロトコル処理部との連携が多いためである．

具体的には，***AnT*** オペレーティングシステム上に(案1)と(案2)の形態を実装する．バッファ管理機構としては広く実装されている mbuf 機能をプロセス，及びライブラリとして実装する．これらの実装にもとづいて，両者のプログラム構成の違いと，それに伴い OS に要求される機能，及び２つの実装での性能の違いについて評価する．

2.2 mbuf 管理プロセスの実装

ドライバ部，プロトコル処理部，バッファ管理部をそれぞれ個別のプロセスとして実装した場合の通信処理の流れを図１に示し，以下に説明する．

(1) 応用プログラム(AP)プロセスから mbuf 管理プロセスに mbuf 取得要求が出されると，管理プロセスは，必要なメモリを OS 核から確保し，その領域を mbuf として要求元 AP プロセスへ渡す．
(2) AP プロセスが送信データを格納しイーサネットプロトコル処理プロセスへ mbuf を渡すと，受け取ったプロセスは管理プロセスへ mbuf の受

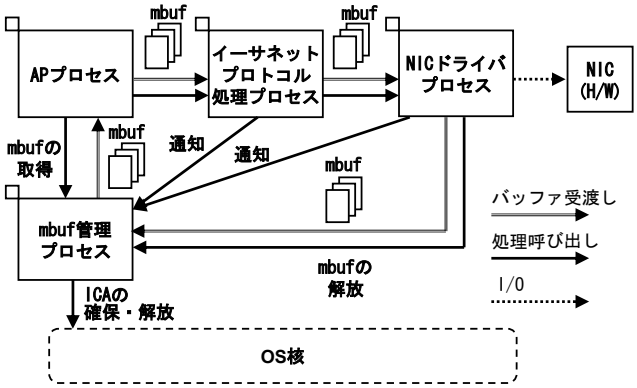

図１ mbuf 管理プロセスにおける処理の流れ

Realization of NIC Driver Process for ***AnT***
Kouta Okamoto†, Yoshinari Nomura‡,
Toshihiro Tabata‡ and Hideo Taniguchi‡
† Faculty of Engineering, Okayama University
‡ Graduate School of Natural Science and Technology, Okayama University

け取りを通知する．これは，どのプロセスが mbuf を使用しているか管理するためである．NIC ドライバへ mbuf を渡す際も同様の処理を行う．
(3) NIC ドライバプロセスは，データの送信を終了すると，管理プロセスへバッファの解放要求を出し，管理プロセスに mbuf を渡す．

本実装では，各処理をプロセス化し独立させているので，信頼性を確保しやすい．しかしながら，mbuf 管理プロセスに対する処理の呼び出しが多く，性能の低下が懸念される．

また，mbuf をプロセス間で受け渡しする場合，プロセス間でのゼロコピー通信機能が有効である．ゼロコピー通信機能がない場合，mbuf の受け渡しの際にメモリコピーが発生してしまう．本実装では，mbuf 用の領域に ***AnT*** で実装されているコア間通信データ域[3](ICA)を使用し，ゼロコピー通信を行うこととした．

2.3 mbuf 管理ライブラリの実装

ドライバ部，プロトコル処理部をプロセス，バッファ管理部をライブラリとして実装した場合の通信処理の流れを図２に示し，以下に説明する．mbuf 管理プロセスの実装と同様に，mbuf として用いる領域に ICA を使用する．

基本的な処理の流れは mbuf 管理プロセスの形態と同様である．しかし，mbuf 管理プロセスへの mbuf の取得，解放，及び通知の処理は発生しない．このため，これらの処理にかかるオーバヘッドを削減でき，処理速度の向上が期待できる．一方，各プロセスでの mbuf 管理ライブラリの処理においてメモリの分割損が生じるため，利用効率が低下する．

3. 評価

各実装について，Pentium４プロセッサ 2.8GHz，RealTek 8139 PCI Ethernet card，及び 512MB RAM を搭載した計算機を用いて，***AnT*** オペレーティングシステム上で評価した．具体

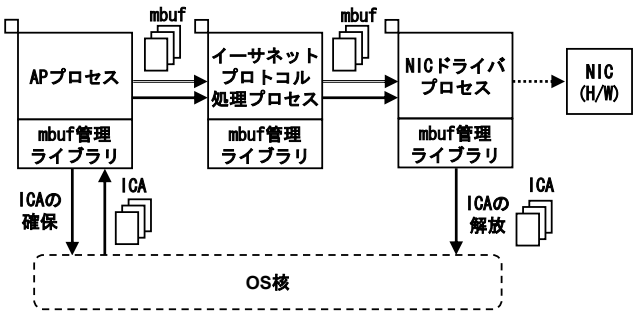

図２ mbuf 管理ライブラリにおける処理の流れ

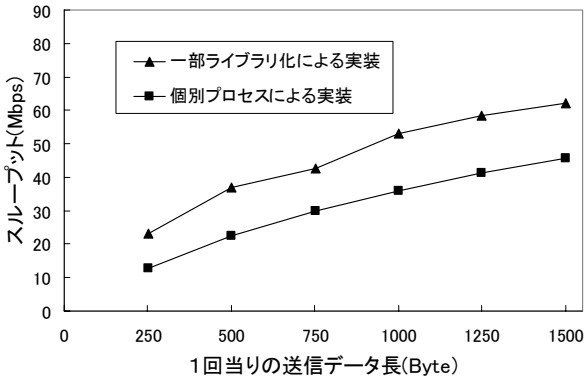

図３ スループット

的には，送信処理を行い，１回当りの送信データ長に対するスループットを測定した．測定結果を図３に示す．図３より，バッファ管理部をプロセスとして実装した場合よりも，ライブラリとして実装した場合では，通信速度が約 1.3 倍速いことがわかる．

4. おわりに

バッファ管理機構をプロセス，及びライブラリとして実装した際の信頼性と性能とのトレードオフについて考察し，その性能の違いを示した．バッファ管理部をライブラリとして実現した場合，プロセスとして実現するよりも通信速度が約 1.3 倍向上する．

残された課題は信頼性に着目した評価である．

謝辞 本研究の一部は，科学研究費補助金 基盤研究(B)「適応性と頑健性を有する基盤ソフトウェアのカーネル開発」（課題番号：18300010）による．

参考文献

[1] 野村裕佑，乃村能成，横山和俊，谷口秀夫，丸山勝巳，“走行モード変更機構を利用したデバイスドライバの実現，” 情報処理学会研究報告，2006-OS-101， Vol.2006， No.15， pp.25-31， Feb.2006.

[2] Andrew S. Tanenbaum, Jorrit N. Herder, and Herbert Bos，“Can We Make Operating Systems Reliable and Secure?”IEEE Computer Magazine, Vol.39, No.5, pp.44-51, May.2006.

[3] 谷口秀夫，乃村能成，田端利宏，安達俊光，野村裕佑，梅本昌典，仁科匡人，“適応性と堅牢性をあわせ持つ ***AnT*** オペレーティングシステム，”情報処理学会研究報告，2006-OS-103, Vol.2006, No.86, pp.71-78, Jul.2006.