

AnT オペレーティングシステムにおける TCP/IP 通信機能の実現

井上 喜弘^{†1} 谷口 秀夫^{†1} 山内 利宏^{†1}

マイクロカーネル構造のオペレーティングシステムは、OS 機能の大半をプロセスとして実現し、高い適応性と堅牢性を持つ。しかし、通信機能を実現する OS サーバ間でデータの授受処理が発生し、このデータ授受オーバーヘッドが通信機能を低下させる懸念がある。ここでは、マイクロカーネル構造を有する *AnT* オペレーティングシステム上に実現した TCP/IP 通信機能について報告する。*AnT* は、プロセス間の複写レスでのデータ授受機能を用いた高速なサーバプログラム間通信機構を有する。この通信機構を利用して、TCP/IP 通信機能を実現するプロセス間で送受信データを複写レスで授受可能にした。また、実現した TCP/IP 通信機能について、パケット送信処理時間と RTT について評価した。

Implementation of the TCP/IP communication function in the *AnT* operating system

YOSHIHIRO INOUE,^{†1} HIDEO TANIGUCHI^{†1}
and TOSHIHIRO YAMAUCHI^{†1}

An operating system based on microkernel architecture realizes most of OS functions as processes and has adaptability and toughness. However, the interprocess communication overheads may decrease communication function. In this paper, we discussed the TCP/IP communication function of realizing the *AnT* operating system based on microkernel architecture. *AnT* has a high-speed communication mechanism between processes used the copy-less data transmission. With this communication function, we could transfer data by copy-less between processes of realizing the TCP/IP communication function. In addition, we evaluated processing time of packet transmission and RTT for TCP/IP communication function of *AnT*.

1. はじめに

計算機サービスが多様化し、それを支える基盤ソフトウェアも複雑になっている。また、Web サービスやメールのように他の計算機と通信するサービスが普及し、計算機間で通信を行う機会が増えたことで、通信の高速化が求められている。一方で、計算機サービスの多様化に伴うプログラムの相互作用による不具合や、ネットワークを介した攻撃の脅威は大きな問題となっている。このような背景から、多様化する計算機サービスを支える高い適応性と不具合や攻撃に耐えうる高い堅牢性を持つオペレーティングシステム(以降、OS)の重要性が増している。

OS のカーネル構造には、モノリシックカーネル構造とマイクロカーネル構造^{1)–6)}がある。モノリシックカーネル構造は、OS 機能をカーネルとして実現する。このため、モノリシックカーネル構造の OS では基盤ソフトウェアの複雑化に伴いカーネル構造が複雑になり、新しい機能の追加や削除が困難になる。一方、マイクロカーネル構造は、割り込み処理や例外処理などの最小限の OS 機能をカーネルとして実現し、デバイスドライバやその他の OS 機能をプロセスとして実現する。このため、新しい OS 機能をプロセスとして実現することにより、その追加や削除を容易に行うことができる。また、各 OS 機能はプロセスとして独立しているため、不具合の発生によるシステム全体への影響を最小限に抑えることができる。しかし、マイクロカーネル構造の OS における通信処理では、通信機能を実現するプロセス(OS サーバ)間でデータの授受処理が発生し、このデータ授受オーバーヘッドが通信機能を低下させる懸念がある。

ここでは、マイクロカーネル構造を有する *AnT* オペレーティングシステム⁷⁾(An operating system with adaptability and toughness)(以降、*AnT*) 上に実現した TCP/IP 通信機能について報告する。*AnT* は、プロセス間の複写レスでのデータ授受機能を用いた高速なサーバプログラム間通信機構^{8)–10)}を有する。この通信機構を利用して、TCP/IP 通信機能を実現するプロセス間の送受信データを複写レスで授受可能にした。また、実現した TCP/IP 通信機能について、評価の結果を報告する。

^{†1} 岡山大学大学院自然科学研究科

Graduate School of Natural Science and Technology, Okayama University

2. AnT オペレーティングシステム

AnT は、マイクロカーネル構造を有する OS である。OS 機能やデバイスドライバを OS サーバとして実現することにより、プログラムの追加や削除を容易に行うことができる。また、機能を OS サーバ毎に分散させることにより、プログラムの暴走によるシステム全体の破壊を防止できる。AnT は多重仮想記憶であり、特徴的な領域としてコア間通信データ域¹¹⁾ (ICA : Inter-core Communication Area) がある。ICA の特徴として、以下の 3 つがある。

- (1) ページ (4 KB) を単位とし、n ページ分の領域の確保と解放
- (2) 確保した領域 (n ページ) の実メモリ連続の保証
- (3) 2 仮想空間の間での領域の貼り替え

ICA を利用したプロセス間でのデータ授受は、授受するデータを格納した ICA をデータ授受元プロセスの仮想空間から剥がし、データ授受先プロセスの仮想空間へ貼り付けることで行える。つまり、2 つの仮想空間の間で ICA の貼り替えを行うことで、通信する 2 つのプロセス間でのデータ授受を複写レスで実現する。

また、複写レスデータ授受機能を利用した高速なサーバプログラム間通信機構がある。この機構は、制御用情報とデータ情報を各々 ICA に格納し、サーバプログラム間でのデータ授受を複写レスで実現している。また、1 つの制御用 ICA を複数のプロセスで持ち回り、依頼情報を積み重ねること (多段依頼) でオーバーヘッドを抑制する。さらに、多段依頼に対する結果返却は、逐次的な返却ではなく、結果の返却を必要とする依頼元のプロセスへ直接的に返却し、処理を高速化している。

3. マイクロカーネル構造における TCP/IP 通信機能実現時の問題点

TCP/IP は、インターネットを含む通信ネットワークで広く利用されているプロトコルであり、基本的な処理の流れは、コネクション確立、データ転送、およびコネクション切断である。

マイクロカーネル構造の OS ではデバイスドライバやファイル管理といった一部の OS 機能を OS サーバとして実現する。このため、モノリシックカーネル構造の OS と比較して、以下の 2 つの処理オーバーヘッドが発生することで TCP/IP 通信性能が低下する。

- (1) OS 機能呼び出し時における OS サーバ間でのデータの複写処理オーバーヘッド
- (2) プロセス切り替えに伴う空間切り替えによるオーバーヘッド

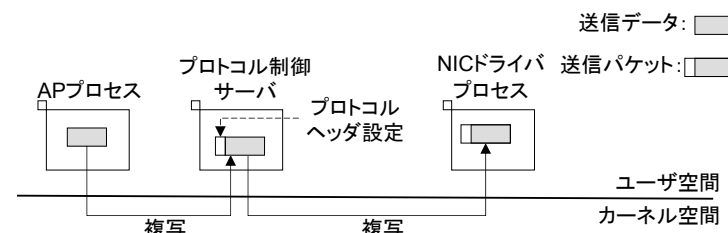


図 1 マイクロカーネル構造 OS におけるパケット送信処理の例

OS サーバ間でのデータの複写処理オーバーヘッドの例として、マイクロカーネル構造を持つ MINIX⁴⁾⁻⁶⁾ では、OS サーバ間で互いのメモリ領域を参照できない。このため、プロセス間でのデータ授受時にシステムタスク (カーネルモードの処理を行う OS サーバ) へデータ複写の依頼を発行する。依頼を受けたシステムタスクは、データ授受元の OS サーバの領域からデータ授受先の OS サーバの領域へデータを複写する。

マイクロカーネル構造の OS におけるパケット送信処理の例を図 1 に示す。図 1 では、TCP/IP 通信を実現する OS 機能として、各ヘッダの設定やパケットを生成するプロトコル制御部と NIC (Network Interface Card) ドライバをそれぞれプロセスとして実現しているものとする。マイクロカーネル構造の OS におけるパケット送信処理では、はじめに AP プロセスからプロトコル制御サーバへ送信データの複写が発生する。プロトコル制御サーバでは、複写された送信データから送信パケットを生成する。生成された送信パケットは、プロトコル制御サーバから NIC ドライバプロセスへと複写される。このように、1 つの送信データを送信する場合でも、最低 2 回のデータ複写が発生する。受信時も同様に、NIC ドライバプロセスとプロトコル制御サーバ間およびプロトコル制御サーバと AP プロセス間でそれぞれデータの複写が発生する。さらに、実際の通信では、サイズの大きなデータの送信を行う場合やネットワークが混雑している場合がある。これらの場合、パケットの喪失や送信データの分割が発生し、再送処理や送信データの分割処理により送信するパケット数が増加し、結果としてプロセス間でのデータ複写回数が増加する。このプロセス間でのデータ複写処理によるオーバーヘッドが通信性能に大きく影響を与える。

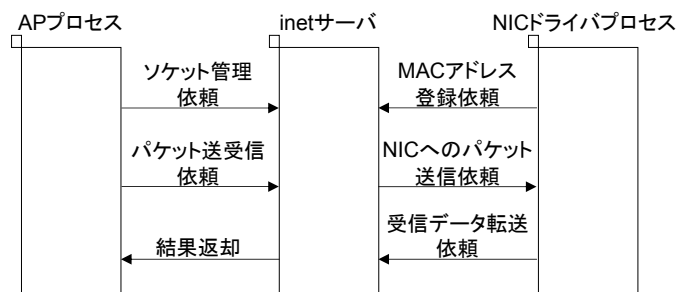


図2 TCP/IP 通信の基本構造

表1 inet サーバの提供機能一覧

処理依頼	機能	関数名	処理内容
ソケット管理 依頼	ソケット生成	socket()	ソケットを生成する
	ソケット bind	bind()	ソケットへアドレスを割り当てる
	ソケット listen	listen()	ソケットを接続待ちにする
	ソケット connect	connect()	相手との接続を確立する
	ソケット accept	accept()	接続を受け入れる
	ソケット close	close()	ソケットを開放する
パケット送受信 依頼	パケット送信	send()	パケットを送信する
	パケット受信	recv()	パケットを受信する
MAC アドレス 登録依頼	MAC アドレス 登録	ifattach()	inet サーバへ MAC アドレスを 登録する
受信データ転送 依頼	受信データ転送	ethinput()	NIC ドライバプロセスから データを受信する

4. 設 計

4.1 設 計 方 針

AntにおけるTCP/IP通信機能を設計するにあたり、以下の2つを設計方針とした。

(1) プロセス間での送信データの複写レス授受

送信処理において、送信データが複数のプロセスに受け渡される。このため、送信データを送信データ格納用ICA(データ用ICA)に格納し、処理依頼時にデータ用ICAを依頼先のプロセスへと貼り替えることにより、複数プロセス間での送信データの複写レス授受を実現する。

(2) サーバプログラム間通信処理の高速化

多くのOS機能をプロセスとして実現するため、プロセス間でのデータ授受が頻発する。このため、ICAを用いた高速なサーバプログラム間通信機構を用いて、APプロセス-NIC間で発生するサーバプログラム間通信処理を高速化する。また、直接返却を利用し、結果返却の回数を削減する。

4.2 基本構造

TCP/IP通信機能の基本構造を図2に示す。プロトコル制御サーバ(inetサーバ)とNICドライバプロセスの2つのプロセスによりTCP/IP通信機能を実現する。

inetサーバは、ソケットの生成やパケットの送受信など、TCP/IP通信に利用する各種機能をAPプロセスとNICドライバプロセスに提供する。inetサーバは、APプロセスやNICドライバプロセスからの処理依頼を取得し、必要であれば処理結果を依頼元プロセス

へ返却する。inetサーバへの処理依頼は、inetサーバが提供する機能を利用する。inetサーバが提供する機能の一覧を表1に示す。ソケット管理依頼とパケット送受信依頼は、inetサーバがAPプロセスに対して提供する機能であり、MACアドレス登録依頼と受信データ転送依頼は、inetサーバがNICドライバプロセスに対して提供する機能である。ソケット管理依頼は、TCP/IP通信におけるコネクション確立とコネクション切断時にAPプロセスが利用する機能である。パケット送受信依頼は、TCP/IP通信におけるパケット送受信時にAPプロセスが利用する機能である。MACアドレス登録依頼は、MACアドレスをinetサーバに登録する際にNICドライバプロセスが利用する機能である。受信データ転送依頼は、NICが受信したデータをinetサーバに転送する際にNICドライバプロセスが利用する機能である。

NICドライバプロセスは、inetサーバからのパケット送信依頼を取得し、NICへパケットを転送する。また、NICがパケットを受信すると、NICから受信データを受け取り、inetサーバに対しパケット受信依頼を発行する。

inetサーバの内部構造を図3に示す。inetサーバは、依頼管理部、ソケット管理部、およびプロトコル管理部の3つの管理部からなる。依頼管理部は、APプロセスやNICドライバプロセスからの処理依頼を取得する。取得した依頼によって、ソケット管理部やプロトコル管理部を呼び出す。また、各管理部からの処理結果を受け取り、必要に応じて依頼元のプロセスへ結果を返却する。ソケット管理部は、ソケットの生成、削除、および状態遷移などのソケットに関する処理を行う。また、コネクション確立要求パケットやコネクション切断要求パケット送信時にプロトコル管理部を呼び出す。プロトコル管理部は、パケットの送

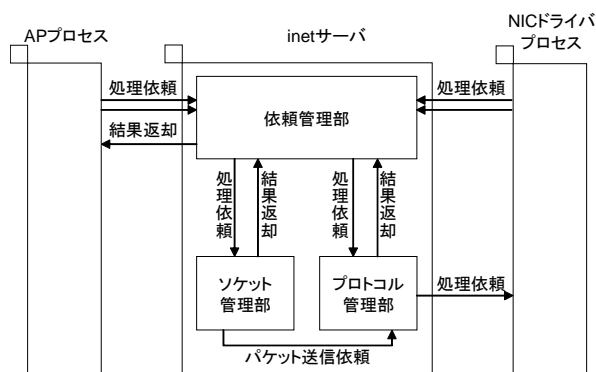


図 3 inet サーバの内部構造

信処理における送信データへの各プロトコルヘッダ設定処理やパケット受信処理における受信パケットへの各プロトコルヘッダ削除処理などのパケット送受信時のプロトコル管理処理を行う。また、パケット送信時には、NIC ドライバプロセスに対して NIC へのデータ送信依頼を発行する。

4.3 プロセス間での複写レス送信データ授受

モノリシックカーネル構造の OS におけるパケット送信処理では、AP プロセスが持つ送信データをカーネル空間に複写する処理が発生する。一方、*AnT* におけるパケット送信処理では、プロセス間での送信データの授受処理が発生する。このとき、AP プロセスでデータ用 ICA を取得し、AP プロセスの持つ送信データをデータ用 ICA に複写することにより、データ用 ICA の貼り替えを利用したプロセス間での送信データの複写レス授受が可能になる。また、AP プロセスがデータ用 ICA に送信データを直接格納することにより、パケット送信処理を完全複写レスで行うことが可能となる。

パケット送信時の送信データの流れを図 4 に示す。パケット送信処理では、AP プロセスが `send()` を呼び出すことで、inet サーバへのパケット送信依頼を発行する。このとき、データ用 ICA を取得し、取得したデータ用 ICA に送信データを格納する。その後、ICA の貼り替えを行い、依頼情報を格納した制御用 ICA とともにデータ用 ICA を inet サーバへ送信する。inet サーバでは、取得したデータ用 ICA 内の送信データに対して各プロトコ

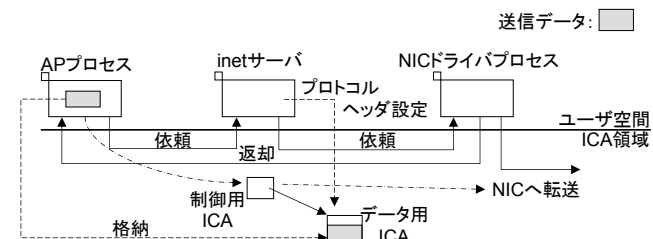


図 4 パケット送信時の送信データの流れ

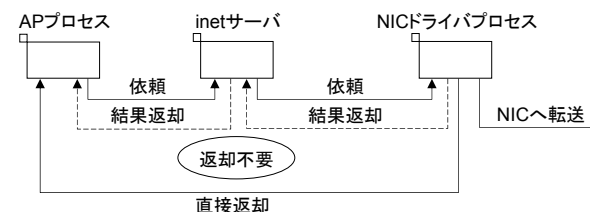


図 5 パケット送信時の直接返却の様子

ルヘッダを設定する。プロトコルヘッダ設定後、inet サーバは、NIC へのパケット転送依頼を NIC ドライバプロセスに発行する。このとき、ICA の貼り替えを行い、依頼情報を格納した制御用 ICA とともにデータ用 ICA を NIC ドライバプロセスへ送信する。このように、パケット送信処理では、送信データをデータ用 ICA へ格納することで、プロセス間の送信データの授受を複写レスで行うことが可能になる。

4.4 直接返却を利用したプロセス間データ授受回数の削減

パケット送信時、AP プロセスは inet サーバへパケット送信依頼を発行し、inet サーバは NIC ドライバプロセスへ NIC へのパケット送信依頼を発行する。このように、パケット送信時には多段依頼が発生する。通常、多段依頼の結果返却では、中継した OS サーバを経由した逐次的な返却が行われる。しかし、パケット送信処理の場合、中継した inet サーバはパケットの生成と各プロトコルヘッダの設定のみを行うため、結果返却の際に inet サーバを経由する必要はない。このため、結果返却時に inet サーバへの結果返却を省略するこ

表 2 評価環境

	送信計算機 (測定計算機)	受信計算機
CPU	Intel [®] Celeron [®] D (2.8GHz)	Intel [®] Pentium [®] 4 (2.8GHz)
メモリ	256MB	512MB
OS	AnT または FreeBSD 4.3-RELEASE	FreeBSD 4.3-RELEASE
NIC	Realtek 8139 PCI Ethernet Card	Realtek 8139 PCI Ethernet Card

とで、返却処理を削減できる。

パケット送信時の直接返却の様子を図 5 に示す。パケット送信時の結果返却は、直接返却を用いて NIC ドライバプロセスから AP プロセスへ直接結果を返却する。具体的には、inet サーバから NIC ドライバプロセスへの処理依頼時に、結果返却で inet サーバを経由しないように制御用 ICA に情報 (直接返却の旨の情報) を設定し、NIC ドライバプロセスから AP プロセスへ直接結果を返却する。これにより、inet サーバを経由して結果を返却する場合に比べ、1 回分の返却処理を削減することが可能になる。

5. 評価

5.1 評価の観点

AnT の TCP/IP 通信における以下の 2 つの高速化処理について、効果を明らかにする。

(1) 送信データの複写レス授受による複写回数の削減

モニリシックカーネル構造の OS では、パケット送信処理時に AP プロセスが持つ送信データをカーネル空間へ複写する処理が発生する。これに対し、**AnT** では、データ用 ICA に送信データを格納することで、パケット送信処理における送信データの複写処理を削減できる。この効果を明らかにする。

(2) プロセス間通信処理の高速化

マイクロカーネル構造の OS では、TCP/IP 通信機能を実現するプロセス間でのデータの授受オーバーヘッドが発生する。これに対し、**AnT** では、ICA を利用したサーバプログラム間通信機構を利用してプロセス間のデータ授受処理を高速化している。さらに、直接返却を利用してプロセス間でのデータ授受処理回数を削減している。この効果を明らかにする。

AnT の TCP/IP 通信における高速化処理の効果を明らかにするため、パケット送信処理時間とラウンドトリップタイム (RTT) を測定した。また、比較対象として、FreeBSD 4.3-RELEASE におけるパケット送信処理時間と RTT を測定した。評価環境を表 2 に示す。

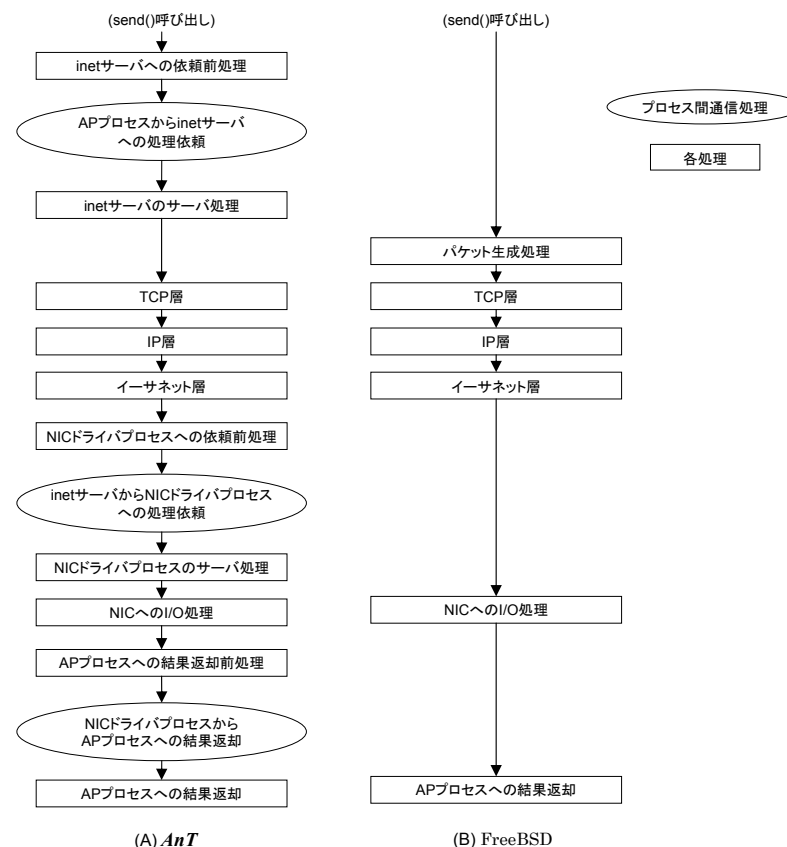


図 6 パケット送信処理内容の比較

5.2 パケット送信処理

AnT と FreeBSD のパケット送信処理内容の比較を図 6 に示す。図 6 より、**AnT** と FreeBSD で共通する処理として、TCP 層、IP 層、イーサネット層、NIC への I/O 処理、および AP プロセスへの結果返却がある。**AnT** 特有の処理として、処理依頼と結果返却の前処理 (inet サーバへの依頼前処理、NIC ドライバプロセスへの依頼前処理、および AP プ

表 3 パケット送信処理時間

処理内容	区分	送信データサイズ:256バイト			送信データサイズ:512バイト			送信データサイズ:768バイト			送信データサイズ:1024バイト		
		処理時間(μ秒)		差分 (μ秒)	処理時間(μ秒)		差分 (μ秒)	処理時間(μ秒)		差分 (μ秒)	処理時間(μ秒)		差分 (μ秒)
		AnT	FreeBSD		AnT	FreeBSD		AnT	FreeBSD		AnT	FreeBSD	
inetサーバへの依頼前処理	☆	0.19	-	0.19	0.32	-	0.32	0.35	-	0.35	0.32	-	0.32
APプロセスからinetサーバへの処理依頼	☆	5.13	-	5.13	5.14	-	5.14	5.13	-	5.13	5.15	-	5.15
inetサーバからのサーバ処理	☆	0.08	-	0.08	0.08	-	0.08	0.08	-	0.08	0.08	-	0.08
パケット生成処理	○	-	0.94	-0.94	-	0.99	-0.99	-	1.03	-1.03	-	1.10	-1.10
TCP層		2.67	1.48	1.20	2.89	1.72	1.17	3.15	1.91	1.24	3.49	2.18	1.31
IP層		0.53	0.21	0.32	0.57	0.21	0.36	0.52	0.22	0.30	0.52	0.21	0.31
イーサネット層		0.14	0.27	-0.13	0.14	0.26	-0.12	0.14	0.27	-0.12	0.14	0.27	-0.12
NICドライバプロセスへの依頼前処理	☆	0.13	-	0.13	0.13	-	0.13	0.13	-	0.13	0.13	-	0.13
inetサーバからNICドライバプロセスへの処理依頼	☆	6.91	-	6.91	6.88	-	6.88	6.87	-	6.87	6.89	-	6.89
NICドライバプロセスのサーバ処理	☆	0.04	-	0.04	0.04	-	0.04	0.04	-	0.04	0.04	-	0.04
NICへのI/O処理		4.25	1.71	2.54	4.46	1.99	2.47	4.66	2.11	2.55	4.85	2.27	2.57
APプロセスへの結果返却前処理	☆	0.20	-	0.20	0.20	-	0.20	0.22	-	0.22	0.20	-	0.20
NICドライバプロセスからAPプロセスへの結果返却	☆	3.11	-	3.11	3.10	-	3.10	3.09	-	3.09	3.12	-	3.12
APへの結果返却		0.20	1.21	-1.01	0.21	1.13	-0.92	0.21	1.19	-0.99	0.21	1.21	-0.99
パケット送信処理時間		23.58	5.82	17.76	24.16	6.30	17.86	24.57	6.73	17.84	25.13	7.23	17.90

☆:AnT特有の処理 ○:FreeBSD特有の処理

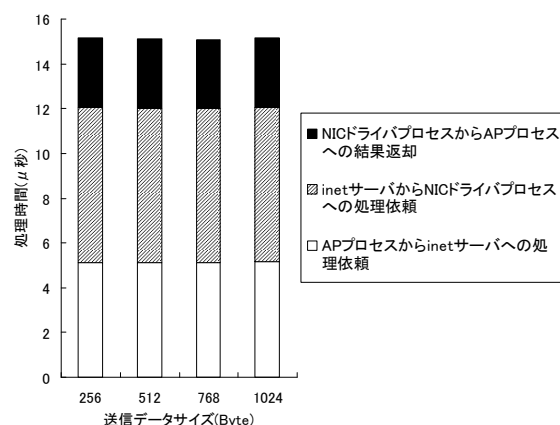


図 7 プロセス間通信処理時間

ロセスへの結果返却前処理),サーバ処理 (inetサーバのサーバ処理, NICドライバプロセスのサーバ処理),およびプロセス間通信 (APプロセスからinetサーバへの処理依頼, inetサーバからNICドライバプロセスへの処理依頼,およびNICドライバプロセスからAPプロセスへの結果返却)がある。FreeBSD特有の処理として,パケット生成処理がある。

パケット送信処理における各処理内容の処理時間を表3に示す。表3から以下のことがわかる。

(1) AnTはFreeBSDと比較してパケット送信処理時間が長い。

例えば,送信データサイズが256バイトのとき,AnTとFreeBSDのパケット送信処理時間は,それぞれ23.58μ秒と5.82μ秒である。つまり,AnTはFreeBSDより17.76μ秒パケット送信時間が長い。このとき,3回のプロセス間通信の処理時間は,それぞれ5.13μ秒,6.91μ秒,3.11μ秒であり,この合計は15.15μ秒である。これは,AnTとFreeBSDのパケット送信処理時間の差(パケット送信処理時間差)の約85%を占める。つまり,プロセス間通信の処理時間がパケット送信処理時間差の要因である。パケット送信処理時間差の要因であるAnTのプロセス間通信処理について分析するため,プロセス間通信の処理時間を図7に示す。図7より,以下のことがわかる。

(A) 各プロセス間通信の処理時間は,送信データサイズに関わらず一定である。これは,ICAを利用し,プロセス間でのデータ授受を複写レスで行うためである。

(B) 各プロセス間通信は,処理時間が異なる。これは,処理依頼が結果返却であるかの違い,同期処理であるか非同期処理であるかの違い,およびデータ用ICAの有無の違いにより,処理時間が変動するためである¹⁰⁾。APプロセスからinetサーバへの処理依頼はデータ用ICAを含む同期型の通信である。inetサーバからNICドライバプロセスへの処理依頼は,データ用ICAを含む非同期型の通信である。NICドライバプロセスからAPプロセス

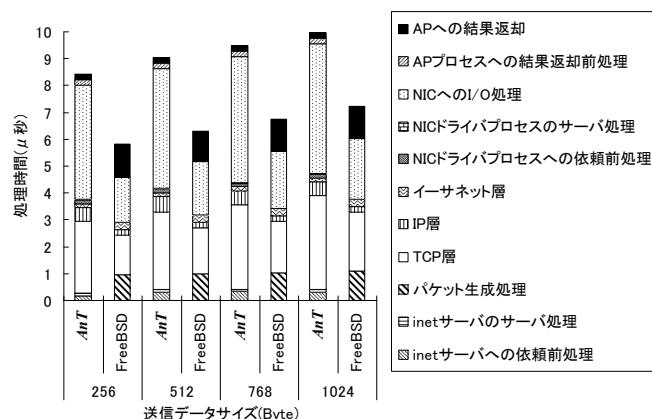


図 8 バケット送信処理時間 (プロセス間通信を除く)

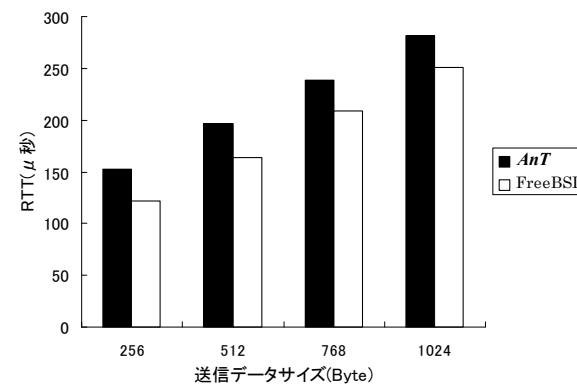


図 9 RTT

への結果返却は、データ用 ICA を含まない同期型の通信である。これらの違いにより、パケット送信処理における各プロセス間通信の処理時間は異なる。

また、プロセス間通信を除いたパケット送信処理時間を図 8 に示す。図 8 より、以下のことがわかる。

(C) *AnT* の TCP 層の処理時間と NIC への I/O 処理時間は、FreeBSD と比べて長い。送信データサイズが 256 バイトのとき、*AnT* と FreeBSD の TCP 層の処理時間は、それぞれ 2.67 μ 秒と 1.48 μ 秒である。つまり、*AnT* では FreeBSD より 1.20 μ 秒処理時間が長い。また、*AnT* と FreeBSD の NIC への I/O 処理時間は、それぞれ 4.25 μ 秒と 1.71 μ 秒である。つまり、*AnT* では FreeBSD より 2.54 μ 秒処理時間が長い。*AnT* の TCP 層の処理では、inet サーバにおけるデータ用 ICA への初アクセスが発生する。データ用 ICA への初アクセスは、そのプロセスにデータ用 ICA が貼り付けられた後、初めてアクセスした際に発生するオーバヘッドであり、TLB ミス等のキャッシュミスによるものである。また、*AnT* の NIC への I/O 処理では、NIC ドライバプロセスにおけるデータ用 ICA への初アクセスが発生し、さらにシステムコールを発行することで NIC への I/O 命令を行うため、処理オーバヘッドが大きくなる。このため、*AnT* の TCP 層の処理と NIC への I/O 処理は、FreeBSD と比較して処理時間が長くなる。しかし、この差は送信データサイズに関わらず一定であるため、パケット送信処理時間差は変化しない。

(2) パケット送信処理時間差は、送信データサイズに関わらず一定である。

送信データサイズが 1024 バイトのとき、*AnT* と FreeBSD のパケット送信処理時間は、それぞれ 25.13 μ 秒と 7.23 μ 秒である。つまり、パケット送信処理時間差は 17.90 μ 秒であり、送信データサイズが 256 バイトのときと比較してパケット送信処理時間差はほとんど変化しない。各 OS は以下の特徴を有している。

(A) *AnT* のパケット送信処理ではパケット生成処理が発生しない。

(B) 図 8 より、FreeBSD のパケット生成処理時間は、データサイズに比例して長くなる。具体的には、送信データサイズの増加に伴い、パケット生成処理時間は 0.94 μ 秒から 1.10 μ 秒まで増加する。これは、送信データの複写処理がパケット生成処理に含まれるためである。このため、複写レスデータ授受の効果により、パケット送信処理時間差は、送信データサイズに比例して小さくなると推察できる。しかし、送信データサイズが 256 バイトから 1024 バイトの間では、パケット送信処理時間差はほとんど変化しない。これは、送信データサイズの増加量が小さいため、パケット生成処理時間増加の影響が小さいと考えられる。これについては、送信データサイズを 1024 バイトより大きくした場合について評価する必要がある。

5.3 ラウンドトリップタイム

パケット送信処理時間は、NIC のデータ転送時間を含んでいない。そこで、NIC のデータ転送時間も含めた通信処理時間を明らかにするため、RTT を測定した。具体的には、送

信側計算機から受信側計算機へデータを送信し、受信側計算機からの1バイトのデータを受信するまでの処理時間を測定した。RTTの測定結果を図9に示す。

送信データサイズが256バイトのとき、*AnT*とFreeBSDのRTTは、それぞれ152.82 μ 秒と122.27 μ 秒であり、この差は30.55 μ 秒である。*AnT*のRTTはFreeBSDの約1.2倍であり、*AnT*とFreeBSDのRTTの差は、パケット送信処理時間差と比較して小さい。これは、NICのデータ転送時間が*AnT*のプロセス間通信処理時間と比較して長いためである。しかし、NICのデータ転送時間は、NICの転送速度が上がるにつれて減少する。このため、転送速度の早いNICを利用する場合、NICのデータ転送時間は短くなり、相対的にプロセス間通信処理時間の影響が大きくなると考えられる。このため、マイクロカーネル構造のOSにおけるTCP/IP通信では、プロセス間通信処理時間をいかに短くするかが重要な課題であるといえる。

6. おわりに

マイクロカーネル構造を有する*AnT*オペレーティングシステム上に実現したTCP/IP通信機能について述べた。機能の設計にあたり、プロセス間での送信データの複写レス授受とサーバプログラム間通信処理の高速化の2つを設計方針とした。*AnT*におけるTCP/IP通信の基本構造として、ネットワークプロトコル制御サーバとNICドライバプロセスの2つのプロセスがある。プロセス間での送信データの複写レス授受では、APプロセスの持つ送信データをデータ用ICAに格納し、ICAの貼り替えによりプロセス間で送信データを複写レスで授受できるようにした。サーバプログラム間通信処理の高速化では、ICAを用いた高速なサーバプログラム間通信機構を用いて、プロセス間通信の際に発生するデータ複写処理を削減し、処理を高速化した。また、直接返却を利用することで、1回分の結果返却処理を削減した。

*AnT*に実現したTCP/IP通信機能を評価し、FreeBSDと比較した。評価では、パケット送信処理時間とRTTを測定した。パケット送信処理時間の測定では、*AnT*はFreeBSDよりもパケット送信時間が長く、プロセス間通信処理時間の影響が大きいことがわかった。NICのデータ転送時間を含めたRTTの測定では、*AnT*とFreeBSDのRTTの差は、パケット送信処理時間の差に比べて小さく、NICのデータ転送時間が*AnT*のプロセス間通信処理時間と比較して十分長いためであることがわかった。しかし、転送速度の早いNICを利用する場合、NICのデータ転送時間は短くなり、相対的にプロセス間通信処理時間の影響が大きくなると考えられる。

残された課題として、TCP/IP通信機能を実現するOSサーバ間でのプロセス間通信処理の高速化がある。

参考文献

- 1) J. Liedtke, "Toward Real Microkernels," *Communications of The ACM*, Vol.39, Issue 9, pp.70-77, 1996.
- 2) A.S. Tanenbaum, J.N. Herder, and H. Bos, "Can we make operating systems reliable and secure?," *IEEE Computer Magazine*, Vol.39, No.5, pp.44-51, 2006.
- 3) D.L.Black, D.B.Golub, D.P.Julin, R.F.Rashid, R.P.Draves, R.W.Dean, A.Forin, J.Barrera, H.Tokuda, G.Malan, and D.Bohman, "Microkernel Operating System Architecture and Mach," *Journal of Information Processing*, Vol.14, no.4(19920315), pp.442-453, 1992.
- 4) A.S. Tanenbaum, A.S. Woodhull, "Operating Systems Design and Implementation Third Edition," Prentice Hall, ISBN 978-0131429383, 2006.
- 5) A.S. Tanenbaum, J.N. Herder, H. Bos, B. Gras, and P. Homburg, "Modular System Programming in MINIX 3," *The USENIX Magazine*, Vol.31, No.2, pp.19-28, 2006.
- 6) A.S. Tanenbaum, J.N. Herder, H. Bos, B. Gras, and P. Homburg, "Roadmap to a Failure-Resilient Operating System," *The USENIX Magazine*, Vol.32, No.1, pp.14-20, 2007.
- 7) 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もつ *AnT* オペレーティングシステム," *情報処理学会研究報告*, Vol.2006-OS-103, pp.71-78 (2006.07).
- 8) 岡本幸大, 谷口秀夫, "*AnT*におけるサーバ間の高速なプログラム間通信機構," 第15回マルチメディア通信と分散処理ワークショップ論文集 (DPSWS2007), pp.61-66 (2007.10).
- 9) 岡本幸大, 谷口秀夫, "*AnT*におけるサーバプログラム間通信機構の評価," *情報処理学会研究報告* 2008-EVA-27, Vol.2008, no.119, pp.19-24 (2008.12).
- 10) 岡本幸大, 谷口秀夫, "*AnT*オペレーティングシステムにおける高速なサーバプログラム間通信機構の実現と評価," *電子情報通信学会論文誌*, Vol.J93-D, No.10, Oct. 2010. (掲載予定)
- 11) 梅本昌典, 田端利宏, 乃村能成, 谷口秀夫, "*AnT*オペレーティングシステムにおけるメモリ領域管理の設計と実現," *情報処理学会研究報告* 2007-OS-104, pp.33-40 (2007.01).