

オープンソースの組み込みシステム用 TCP/IP プロトコルスタック：TINET

(第6回 IP アワード応募書類 特徴の説明書)

阿部 司 吉村 齋 稲川 清

苫小牧工業高等専門学校 情報工学科

1 TINET の概要

インターネットは、常時接続の普及などにより、生活の中で必要不可欠な媒体である。TINET^{*1} は、組み込みシステムをインターネットに接続するために必要な TCP/IP プロトコルスタックである。組み込みシステムにはメモリ容量、重量、サイズ及び消費電力等の厳しいリソース制約があり [1]、特に厳しいメモリ容量に関して、TINET は RAM のメモリ必要量が約 10K バイト、ROM のメモリ必要量が約 46K バイトと少ないメモリ容量で実装している。また、組み込みシステムのリアルタイム OS におけるリアルタイム性の制約も考慮して設計している。

TINET の開発の当初の目的は、研究・教育における利用であったが、TOPPERS プロジェクト [2] 及び、経済産業省東北経済産業局の委託による「組み込みシステム・オープンプラットフォームの構築とその実用化開発」プロジェクトへの参加により、広く産業界における利用も、現在の開発目的としている。このため、幅広く誰でも自由に利用・変更・配布できるように、オープンソースの形態で開発し配布している。オープンソースの組み込みシステム用 TCP/IP プロトコルスタックとしては、Linux や BSD を応用した例があるが、当初から組み込みシステムを対象とし、ITRON TCP/IP API を実装した例はない。

2 TINET の特徴

TINET は、FreeBSD [3] の TCP/IP プロトコルスタックをベースに開発を行った。その第一の理由は、FreeBSD のベースとなった BSD UNIX の TCP/IP プロトコルスタックに実装されている各種アルゴリズムは、長年のインターネットにおけるプロトコル開発の成果が反映され、枯れたソフトウェアとなっており、事実上、他のシステムの見本となっていることである。第二の理由は、TINET の当初の開発目的である研究・教育における利用において、誰でも自由に利用・変更・配布できるようにするためである。BSD ライセンスで配布されて

いる FreeBSD は、ソースコード及びドキュメント等に BSD ライセンスを明記することで、幅広く、誰でも自由に利用・変更・配布が可能な、制約の少ないライセンス形態となっている。第三の理由は、TINET の開発では、人的リソースに限り、まったくゼロからの開発が困難であったためである。

組み込みシステムには、メモリ容量、重量、サイズ及び消費電力等の厳しいリソース制約がある。この中で、TINET ではメモリ容量の制約、特に RAM のメモリ容量の制約への対応を最重点課題としている。また、OS としてはリアルタイム OS (以降 RTOS) を利用することから、リアルタイム性の制約もある。一方、TINET のベースとした FreeBSD は、動的割当て構造体とリスト構造を積極的に使用しているが、メモリ容量やリアルタイム性の制約から、組み込みシステム用 RTOS では、動的割当て構造体とリスト構造を使用すべきではない。このため、TINET では動的割当て構造体とリスト構造を使用しないように、FreeBSD の TCP/IP プロトコルスタックに変更を加えている。

対応する組み込みシステム用 RTOS は、 μ ITRON 4.0 仕様のスタンダードプロファイル [4] である。具体的には、高い実行性能と小さい RAM 使用量から TOPPERS プロジェクトで開発され、オープンソースのフリーソフトウェアとして配布されている JSP を実装のターゲットとしている。また、応用プログラムとのインタフェースとなる API には、ITRON TCP/IP API [5] を実装している。本仕様は、(社)トロン協会の Embedded TCP/IP 技術委員会により策定され、ITRON 専門委員会により 1998 年 5 月に認定された。仕様では、組み込みシステムのプロトコルスタックに求められる以下に示す性質が挙げられており、TINET も対応している。

- 最小のコピー回数
- 動的メモリ管理の排除
- 非同期インタフェース
- API 毎のエラーの詳細化

組み込みシステムに使用されているプロセッサの規模にも幅があり、メモリ容量が数 K バイトの 8 ビット CPU から、汎用プロセッサ並みの仮想記憶システムを備えた 32 ビットや 64 ビットのプロ

^{*1} 名前の由来は、Tomakomai の T に、インターネットの略称である INET を組合わせている。

セッサまで存在している。また、あまりにリソース制約の厳しいプロセッサに、TCP/IP プロトコルスタックを実装することは現実的ではない。このため、TINET では、以下に示す規模の組込みシステムを対象としている。

- (1) メモリ容量は、ROM が 128K バイト、RAM が 32K バイト程度で、メモリ管理ユニットなし。
- (2) TCP/IP の各ヘッダは、32 ビット単位で処理すると効率が良いため、C 言語の int が 32 ビットで、レジスタと ALU も 32 ビットが望ましい。

組込みシステム用のプロセッサの現状では、ROM メモリ容量には比較的余裕があるが、RAM メモリ容量はあまり余裕がなく、RAM メモリ容量の制約への対応が重要である。これらを考慮して、TINET では、対象とした組込みシステムの規模に合わせて通信機能を、次のように設定している。

- (1) TCP のオプションは MSS のみ実装する。
- (2) IP での転送、データグラム分割・再構成、及びオプションは実装しない。
- (3) ネットワークでの配置は、単一リンクの終端ノードを想定している。
- (4) データリンク・物理層は、単一のネットワークインタフェースのみ実装する。

3 動作環境と使用したツール

研究・開発の初期より現在まで、実装ターゲットに関して、プロセッサには、ルネサステクノロジ製 H8/300H シリーズの H8/3048F、H8/3068F 及び H8/3069F を使用している。H8/300H シリーズは、16 ビット CPU であるが、レジスタと ALU が 32 ビットである。内蔵 ROM 容量に余裕はあるが、内蔵 RAM は H8/3068F 及び H8/3069F でも 16K バイトと小容量のため、外部 RAM を増設している。

実装ターゲットのネットワークインタフェースを以下に示す。

- 内部ループバック
- シリアルインタフェースを利用した PPP
- NE2000 互換のイーサネット

これ以外にも「組込みシステム・オープンプラットフォームの構築とその実用化開発」プロジェクトにおいて、プロセッサとしてルネサステクノロジ製 SH2、XILINX 製 MicroBlaze。イーサネットインタフェースとしてはワイ・デー・ケー製 MicroBlaze 用 MAC IP の MAC_YG0130 へも実装されている。

TINET 自体の開発に関して、われわれの開発環境では、Windows 2000/XP 上で動作する cygwin と、クロス開発用ツールである GNU C コンパイラと GNU バイナリツール等を使用しているが、JSP カーネルと同じ開発環境であり、Linux 等の開発環境で開発を行うことも可能である。

4 開発の経緯

われわれは、2000 年 7 月より、研究・教育に使用する目的で組込みシステム用の TCP/IP プロトコルスタックの研究を開始した。当初は、FreeBSD の TCP/IP プロトコルスタックを使用し、API としてソケットインタフェースを実装していた。しかし、メモリ必要量と FreeBSD で多用されている動的割当て構造体とリスト構造等の問題から、FreeBSD を OS として組込みシステムに実装することは困難であった。同様に、ソケットインタフェースに関しても、多様なネットワークインタフェースに対応するため、TCP/IP だけ必要な場合は、オーバーヘッドが大きく、TINET が対象とする規模の組込みシステムに実装することは困難であった [6]。

このため、2001 年 11 月より、組込みシステム用 RTOS として TOPPERS プロジェクトの JSP カーネル、TINET の API として ITRON TCP/IP API を採用し、実装ターゲットプロセッサには H8/300H シリーズの H8/3048F、ネットワークインタフェースはシリアルインタフェースを利用した PPP として、改めて TINET の研究・開発を開始し、2002 年 3 月には、簡易 WWW サーバを動作させることができた [7]。また、同年 7 月には、ITRON TCP/IP API 仕様に規定されているノンブロッキングコール機能を除いた基本機能の実装を完了した [8]。

2002 年 8 月からは、「組込みシステム・オープンプラットフォームの構築とその実用化開発」プロジェクトに参加するとともに、実装ターゲットプロセッサには H8/300H シリーズの H8/3068F 及び H8/3069F、ネットワークインタフェースは NE2000 互換のイーサネットインタフェースとして、ITRON TCP/IP API 仕様に規定されている基本機能の全ての実装を開始した。その実装が完了した 2003 年 6 月に、TINET リリース 1.0 の一般への公開を開始しており、2003 年 11 月末までに、約 450 のサイトにダウンロードされている。また、複数の企業が採用予定であり、「組込みシステム・オープンプラットフォームの構築とその実用化開発」プロジェクトにおいても、4 企業、1 公設試で TINET を採用しており、その波及効果が期待されている。

5 TINET の実装

5.1 TINET のネットワーク階層構成

ネットワークインタフェースとしてイーサネットを組み込んだ場合の TINET のネットワーク階層構成を図 1 に示す。TINET は、機種依存性をできるだけ排除し、多様な組込みシステム用 RTOS に対応するため、組込みシステム用 RTOS カーネル内部に組込むのではなく、RTOS カーネルと応用プログラムの中間のミドルウェアとして位置付けており、TINET 内でも、RTOS カーネルが提供する各種サービスを利用して動作する。以下に、各ネッ

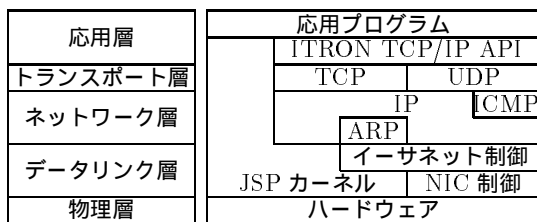


図 1: ネットワーク階層構成

ネットワーク階層の概要を示す。

- (1) ITRON TCP/IP API において、機能レベルでは基本機能を実装しており、組込みシステム特有のノンブロッキングコール、省コピー API、及びコールバック関数も実装している。
- (2) TCP では、性能向上目的の拡張機能は未実装である。しかし、以下の BSD の基本制御アルゴリズムは実装している。
 - 遅延確認応答
 - 再送制御
 - 輻輳回避
 - 高速再転送と高速リカバリ
 TCP のオプションは、最大セグメントサイズ (MSS) オプションのみ実装している。現在の TINET では、IP データグラムの分割・再構成機能は未実装であるが、TCP では MSS オプションにより、IP データグラムの分割を防ぐことができる。また、TCP の再送制御と組込みシステム特有のノンブロッキングコールの処理のため、応用プログラムとは独立に動作する TCP 出力タスクを実装している。
- (3) UDP は、基本的なプロトコル処理は単純であるが、組込みシステム特有のノンブロッキングコール処理のため、応用プログラムとは独立に動作する UDP 出力タスクを実装している。
- (4) IP では、ネットワークでの配置は終端ノードのため、IP データグラムの転送機能は実装していない。IP データグラムの分割・再構成機能は実装する予定であるが、現在は未実装である。TCP に関しては MSS オプションにより分割は回避可能であり、UDP に関しても、512 オクテット以上のデータグラムを送信する応用プログラムは稀である。したがって、IP データグラムの分割・再構成機能が未実装でも問題は少ない。また、経路選択に関して、IP からイーサネットへの出力の前に、配送すべきノードの IP アドレスを決定しなければならない。TINET では、エントリがターゲットノードの IP アドレス、サブネットマスク及びルータから構成される単純な静的経路表を検索するように簡素化している。
- (5) ICMP では、単一リンクの終端ノードに必要と考えられる ICMP エコー機能、配送エラーの受信と報告機能のみ実装している。

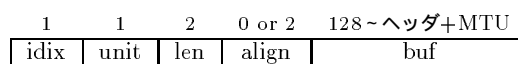


図 2: ネットワークバッファ構造体 (net_buf)

- (6) ARP では、静的経路表から決定したノードに IP データグラムを配送するのに必要な MAC アドレス解決のための ARP 要求の送信、ARP 応答の受信、及び MAC アドレス情報のキャッシュを実装している。また、他のノードからの MAC アドレス解決の ARP 要求の受信と、それに対する ARP 応答の送信も実装しているが、ARP 代理応答機能は実装していない。
- (7) イーサネット制御は、NIC に依存しないイーサネットの制御で、イーサネット出力タスクとイーサネット入力タスクで構成している。RTOS で要求されるリアルタイム性を損なわないために、セマフォにより NIC 制御との同期を取り、実際の入出力はイーサネット出力タスクとイーサネット入力タスクが行うよう実装している。
- (8) NIC 制御は、NIC に依存するイーサネットの制御で、現在の TINET の実装においては、実装ターゲットとして NE2000 互換 NIC を使用した。リアルタイム性への対応を考慮し、入出力割込みに要する時間を短縮するため、セマフォによりイーサネット制御と同期を取るだけで割り込みを終了する。

5.2 ネットワークバッファ (net_buf)

FreeBSD に実装されているプロトコルスタックでは、プロトコルスタック内における各層間のデータの受け渡しに使用されるメモリブロックは、mbuf と呼ばれる固定的なメモリブロックによるリスト構造で実装している。この mbuf は、プロトコルスタック内での処理中に動的メモリ操作が行われる等の問題がある。それに対し、TINET では、ITRON TCP/IP API 仕様で求められている動的メモリ管理の排除を行うため、図 2 に示すように、固定長メモリプールから割り当てる net_buf を用いてメモリブロックを実装している。

図 2 で、buf にはネットワークインタフェースのヘッダとペイロードが入る。buf のサイズの種類は 128, 256, 512, 1,024 及び物理ネットワーク層の MTU にヘッダを加えた長さである。buf のサイズの種類と各種類の個数は、応用プログラムの構成やネットワーク負荷によりコンパイル時に決定するよう工夫を加えた。

ネットワークインタフェースとしてイーサネットを組込んだ場合、イーサネットインタフェース入力では、NIC 制御 LSI の受信バッファに記憶されているフレーム長から net_buf を割当てる。イーサネットインタフェースへの出力では、TCP 出力で、TCP 通信端点から算出したサイズの net_buf を割当てる。入出力いずれも、割当て前にデータサイズ

表 1: TINET のメモリ必要量

機能	RAM [バイト]	ROM [バイト]	計 [バイト]
TCP	1,328	28,652	29,980
UDP	1,104	4,746	5,850
ICMP	20	1,404	1,424
IP	0	736	736
ARP	160	2,314	2,474
TCP/IP 共通機能	0	1,572	1,572
汎用通信機能	1,184	2,288	3,472
イーサネット	2,140	3,408	5,548
net_buf 管理	3,524	256	3,780
小計	9,460	45,376	54,836
応用プログラム	13,404	4,614	18,018
JSP カーネル	4,364	30,276	34,640
合計	27,228	80,266	107,494

は確定しているため、これ以降、net_buf の動的メモリ操作は行わない。

5.3 TINET のメモリ必要量

表 1 は、ネットワークインタフェースとして NE2000 互換のイーサネットを組み込み、応用プログラムとして TCP エコーサーバと UDP エコーサーバを実装した時の TINET のメモリ必要量であり、応用プログラムと JSP カーネルを含めて、対象とする規模の組み込みシステムのメモリ容量の制約を満たしていることがわかる。この中で、TINET が関係するネットワーク階層のトランスポート層から net_buf 管理までのメモリ必要量は、RAM が 9,460 バイト、ROM が 45,376 バイトである。RAM メモリ必要量には、以下に示すように、TINET 内部のタスク用のスタックと、応用プログラムの構成やネットワーク負荷に依存する変動分が含まれている。この RAM メモリ必要量の変動分は、ターゲットの組み込みシステムの RAM 容量に適合するように調整し、削減することが可能である。

- (1) TCP, UDP 及び汎用通信機能には独立したタスクを実装しており、スタック領域として各 1,024 バイトの RAM を割り当てている。
- (2) ソケットインタフェースのソケット、プロトコル制御ブロック、及び TCP 制御ブロックに対応する制御構造体が ITRON TCP/IP API 仕様により規定されており、その数は、応用プログラムの構成により異なる。この例では、TCP と UDP の RAM メモリ必要量に、この制御構造体である TCP 受付口 (12 バイト)、TCP 通信端点 (130 バイト × 2)、及び UDP 通信端点 (64 バイト) が含まれている。
- (3) ARP の RAM メモリ必要量は、全て ARP キャッシュであり、RAM メモリ必要量の固定分は 0 バイトである。ARP キャッシュは 1 エントリ当たり 16 バイトで、コンパイル時に個数を変更可能である。この例では、10 エントリ分を割り当てている。
- (4) イーサネットには、出力タスクと入力タスク

表 2: TINET のメモリ必要量の固定分

機能	RAM [バイト]	ROM [バイト]	計 [バイト]
TCP	32	28,652	28,684
UDP	16	4,746	4,762
ICMP	20	1,404	1,424
IP	0	736	736
ARP	0	2,314	2,314
TCP/IP 共通機能	0	1,572	1,572
汎用通信機能	160	2,288	2,448
小計	228	41,712	41,940
イーサネット	92	3,408	3,500
net_buf 管理	20	256	276
合計	340	45,376	45,716

表 3: FreeBSD のメモリ必要量の固定分

機能	RAM [バイト]	ROM [バイト]	計 [バイト]
TCP	3,492	39,986	43,478
UDP	536	5,196	5,732
ICMP	648	4,914	5,562
IP	3,804	28,944	32,748
ARP	488	6,780	7,268
TCP/IP 共通機能	1,528	14,354	15,882
汎用通信機能	1,368	37,498	38,866
合計	11,864	137,672	149,536

の 2 個のタスクを実装しており、スタックの 2,048 バイトを除いた RAM メモリ必要量の固定分は 92 バイトである。

- (5) net_buf 管理の RAM メモリ必要量中 3,504 バイトは、固定メモリプールに割り当てた net_buf であり、RAM メモリ必要量の固定分は 20 バイトである。net_buf の割り当て量は、応用プログラムの構成やネットワーク負荷により調整する必要があるが、表 1 では全種類を 1 個ずつ割り当てている。

表 2 は、TINET が関係するネットワーク階層のトランスポート層から net_buf 管理までのメモリ必要量の固定分を表しており、RAM メモリ必要量の固定分は 340 バイトである。

5.4 FreeBSD とのメモリ必要量の比較

TINET との比較対象として、TCP から汎用通信機能までに関して、FreeBSD リリース 4.8 TCP/IP プロトコルスタックのメモリ必要量の固定分の調査を行った。対象とした機能は、TINET と同等の基本機能であるトランスポート層の TCP, UDP 及び ICMP, ネットワーク層の IP と ARP, 及び TCP/IP 共通機能と汎用通信機能である。調査方法は、TINET で実装ターゲットとした H8/300H 用の gcc コンパイラを用いて、TCP/IP プロトコルスタックに関するソースコードをコンパイルし、メモリ割り当てを解析した。

結果を表 3 に示す。この表と表 2 に示す TINET のメモリ必要量の小計までの固定分を比較すると、FreeBSD の 11,864 バイトの RAM メモリ必要量

表 4: FreeBSD の TCP 制御用構造体

構造体	サイズ [バイト]
ファイル記述子	28
ソケット	160
TCP 制御ブロック	188
プロトコル制御ブロック	96
合計	472

に対して、TINET の RAM メモリ必要量は 228 バイトとなった。FreeBSD に対して約 1/50 の RAM メモリ必要量となり、制約がより厳しい RAM のメモリ容量に十分対応できる結果が得られた。同様に、ROM メモリ必要量に関しても、FreeBSD の 137,672 バイトの ROM メモリ必要量に対して、TINET は 41,712 バイトであり、約 1/3 の ROM メモリ必要量となった。

5.5 FreeBSD との TCP 制御用構造体の比較

FreeBSD の TCP では、表 4 に示す構造体によりコネクションの状態を管理している。各階層での独立性と汎用性を目的として、ポインタが多用されており、TCP/IP のみの単一のネットワークシステムには不要な項目も多い。

TINET で対応する TCP 制御用構造体は、TCP 受付口と TCP 通信端点である。最適化することにより、これらのメモリ必要量は、1 コネクション当り TCP 受付口が 12 バイト、TCP 通信端点が 130 バイトの合計 142 バイトで、FreeBSD の合計 472 バイトの約 1/3 となっている。

6 今後の計画

今後のインターネットでは、実験段階を終え、普及段階に入ろうとしている IP バージョン 6 (以下 IPv6) への対応が不可欠であるが、IPv6 のアドレスは IP バージョン 4 (以下 IPv4) の 4 倍の 128 ビットと大幅に拡大したほか、アドレスの自動設定機能などのため、メモリ必要量への影響が大きい。

組込みシステムの IPv6 への対応に関しては、TAHI プロジェクト [9] において、非 PC 系デジタル機器への適用に向けた IPv6 最小要求仕様の検討 [10] が行われ、IPv6 最小ホスト仕様が規定されている。現在われわれは、この IPv6 最小ホスト仕様をターゲットとし、「組込みシステム・オープンプラットフォームの構築とその実用化開発」プロジェクトのテーマとして、組込みシステムと RTOS に適合するよう IPv6 用 TINET の実装を行っている。

実装方法に関して、KAME プロジェクト [11] による IPv6 ソフトウェアは、フィールドで十分に検証されており、IPv6 用 TINET の実装のベースとしても最適である。しかし、BSD を実装ターゲットとしているため、動的割当て構造体とリスト構造を積極的に使用しており、組込みシステムを対象とする TINET に、そのまま適用することは困難である。このため、われわれは、動的割当て構造体とリ

スト構造を排除し、メモリ必要量の増加を抑えた IPv6 用の TINET の実装を行っており、IPv4 に対して、全体のメモリ必要量の増加率は約 15%、特に、制約の厳しい RAM のメモリ必要量の増加率は約 2% で実装することができている。必要とされる機能はほぼ完成しているため、今後は品質の向上を図り、これまでと同様に、幅広く、誰でも自由に利用・変更・配布できるように、オープンソースの形態で配布する予定である。

謝辞 TINET の研究・開発は、次に示す各機関からの支援を頂いた。この場を借りて、各関係機関の皆様へ感謝する。

- 財団法人道央産業技術振興機構様、テーマ「組込み型制御システム用 TCP/IP プロトコルスタックの開発」
- 株式会社 NTT ドコモ北海道苫小牧支店様
- 経済産業省東北経済産業局様 (委託先管理法人: 財団法人みやぎ産業振興機構様)、テーマ「組込みシステム・オープンプラットフォームの構築とその実用化開発」

参考文献

- [1] 高田広章: 組込みシステム開発技術の現状と展望, 情報処理学会論文誌, Vol. 42, pp. 930-938 (2001).
- [2] TOPPERS/JSP プロジェクトホームページ: <http://www.toppers.jp/>.
- [3] FreeBSD プロジェクトホームページ: <http://www.freebsd.org/>.
- [4] 坂村健 (監修), 高田広章 (編): μ ITRON 4.0 仕様 4.01.00, トロン協会 (2001).
- [5] 高田広章 (編): ITRON TCP/IP API 仕様 1.00.01, トロン協会 (1998).
- [6] 阿部司, 吉村斎, 久保洋: 組込みシステム用 TCP/IP プロトコルスタックの実装と評価, 情報処理学会論文誌, Vol. 44, pp. 1583-1592 (2003).
- [7] 阿部司: 群知能機械組込み用基本通信ソフトウェア, 2002 年度精密工学会春季大会論文集, p. 233 (2002).
- [8] 阿部司: 組込みシステム用ネットワークインタフェースに関する研究, SWEST4 第 4 回組込みシステム技術に関するサマワークショップ予稿集, pp. 77-84 (2002).
- [9] TAHI プロジェクトホームページ: <http://www.tahi.org/>.
- [10] 岡部宣夫, 石山政浩, 井上淳, 箆島雅之, 坂根昌一, 左治木次郎, 野口敬, 宮田宏: 非 PC 系デジタル機器への適用に向けた IPv6 最小要求仕様の検討, 情報処理, Vol. 42, pp. 920-925 (2001).
- [11] KAME プロジェクトホームページ: <http://www.kame.net/>.