

TINET における ITRON TCP/IP API 拡張機能の実装

苫小牧工業高等専門学校
情報工学科
阿部 司

TINET リリース 1.4 の共同研究

■ 事業名

- 組込みシステム向けネットワーク接続ソフトウェア群の開発

■ 委託元

- 北海道

■ 委託先

- 北海道工場試験場

■ 実施機関

- 1 大学、1 高専、3 企業

■ 開発期間

- 平成 17 年 4 月～18 年 3 月

TINET リリース 1.4 の共同研究

- 組込みシステム向けネットワーク接続ソフトウェア群の開発
- 苫小牧高専担当研究開発テーマ【TINET-1.4】
 - ITRON TCP/IP API 仕様の拡張機能のサポート
 - TCP/UDP 通信端点の生成と削除
 - TCP 受付口の生成と削除
 - TCP 緊急データの送受信
 - TCP/UDP 通信端点オプションの設定と読出し
 - Path MTU
 - IPv6 の断片ヘッダ

TINET の研究の背景

- インターネットが普及し、社会のインフラストラクチャとなった。
- 組み込みシステムもインターネットに接続したいという要求が高まっている。
 - TCP/IP プロトコルスタックが必要である。
 - 組み込みシステムのリソース制約は厳しい。
 - 汎用計算機用の TCP/IP プロトコルスタックをそのまま使うのは現実的ではない。
- 組み込みシステムの特徴に合った TCP/IP プロトコルスタックが必要である。

TINET の特徴

- ベースは FreeBSD である。
 - 枯れたソフトウェアで、他システムの手本である。
 - BSD ライセンスによる配布が可能である。
 - 実は、人的なリソース制約が大きい。
- 組み込みシステムのリソース制約に対応している。
- 対応 RTOS は TOPPERS/JSP を採用した。
- API は ITRON TCP/IP API 仕様を採用した。
- IPv4 と IPv6 に対応している。

TINET が提供する機能

■ TINET-1.1 (2004/2/19)

- 幅広い応用層に対応した。
- TCP のオプションは MSS のみ。
- IP のオプションは分割・再構成機能のみ。
- ルータではないホストに限定した。
- ネットワークの終端ノードに限定した。
- 単一のネットワークインタフェースに限定した。

■ TINET-1.2 (2004/9/13)

- IPv4 と IPv6 に対応した。

TINET が提供する機能

■ TINET-1.3 (2006/2/7)

- タスクからの Time Wait 状態の TCP 通信端点の分離機能を実装した。
- 送受信ウィンドバッファの省コピー機能を実装した。
- 不要な機能を取り外すことを可能にした。
 - TCP/UDP のノンブロッキングコール
 - TCP 受付口
- TCP/UDP API をライブラリ化した。
- 省メモリへの対応を強化した。
 - H8/3069F の内蔵メモリのみで実行可能である。
- TCP ヘッダのトレース出力機能を実装した。

TINET リリース 1.4

■ ITRON TCP/IP API 拡張機能

- TCP 受付口、TCP/UDP 通信端点の動的生成と削除
- TCP/UDP 通信端点オプションの設定と参照
- 緊急データ送受信 API

■ 予約 ID の定義

■ IPv6 関係

- IPv6 アドレス自動設定機能の改良
- IPv6 データグラムの分割・再構成機能
- Path MTU への対応

ITRON TCP/IP API 拡張機能

- コンパイル時コンフィギュレーションパラメータ
 - TCP_CFG_EXTENTIONS
 - ITRON TCP/IP API の TCP の拡張機能を有効にする。
 - UDP_CFG_EXTENTIONS
 - ITRON TCP/IP API の UDP の拡張機能を有効にする。

ITRON TCP/IP API 拡張機能(1)

■ TCP 受付口の動的生成

```
ER ercd = tcp_cre_rep(ID repid,  
                      T_TCP_CREP *pk_crep);
```

```
ER ercd = tcp6_cre_rep(ID repid,  
                       T_TCP6_CREP *pk_crep);
```

■ TCP 受付口の動的削除

```
ER ercd = tcp_del_rep(ID repid);
```

```
ER ercd = tcp6_del_rep(ID repid);
```

ITRON TCP/IP API 拡張機能(2)

■ TCP 通信端点の動的生成

```
ER ercd = tcp_cre_cep(ID cepid,  
                      T_TCP_CCEP *pk_ccep);
```

```
ER ercd = tcp6_cre_cep(ID cepid,  
                       T_TCP6_CCEP *pk_ccep);
```

■ TCP 通信端点の動的削除

```
ER ercd = tcp_del_cep(ID cepid);
```

```
ER ercd = tcp6_del_cep(ID cepid);
```

ITRON TCP/IP API 拡張機能(3)

■ UDP 通信端点の動的生成

```
ER ercd = udp_cre_cep(ID cepid,  
                      T_UCP_CCEP *pk_ccep);
```

```
ER ercd = udp6_cre_cep(ID cepid,  
                       T_UDP6_CCEP *pk_ccep);
```

■ UDP 通信端点の動的削除

```
ER ercd = udp_del_cep(ID cepid);
```

```
ER ercd = udp6_del_cep(ID cepid);
```

通信管理オブジェクトの動的生成と削除の課題

- 通信管理オブジェクトに用いるメモリ領域の確保が必要である。
 - 組込みシステムの特徴から動的メモリ管理は出来るだけ避けたい。
 - TINET が対応する RTOS である TOPPERS/JSP で提供されているメモリ管理サービスは、固定メモリプールのみである。

通信管理オブジェクトの動的生成と削除の課題

■ 通信管理オブジェクトの指定方法

- 通信管理オブジェクトの ID を定義し、アプリケーションから参照できる必要がある。
- 各 API で、静的 API で指定された通信管理オブジェクトと区別することなく扱える必要がある。

通信管理オブジェクトの予約 ID の定義

- 通信管理オブジェクトの予約 ID を定義する静的 API を実装する。
- メモリ管理に関しては、予約 ID 分の通信管理オブジェクトのメモリを静的に確保する。
 - 静的 API で指定される通信管理オブジェクトと同じメモリ領域内に確保する(固定メモリプールは使用しない)。
 - 各 API で、一部を除き区別することなく扱うことが可能になる。
 - 例えば、tcp_del_rep では、静的 API の TCP_CRE_REP で生成した TCP 受付口は削除できない。

予約 ID の定義

■ TCP 受付口の予約 ID の定義

VRID_TCP_REP(ID repid);

VRID_TCP6_REP(ID repid);

■ TCP 通信端点の予約 ID の定義

VRID_TCP_CEP(ID cepid);

VRID_TCP6_CEP(ID cepid);

■ UDP 通信端点の予約 ID の定義

VRID_UDP_CEP(ID cepid);

VRID_UDP6_CEP(ID cepid);

TCP 通信端点の動的生成と削除の使用例[1]

- TCP 通信端点の予約 API により、TCP 通信端点を予約する。

```
VRID_TCP_CEP(TCP_RSV_CEPID1);
```

- TINET のオブジェクト ID 自動割付結果ファイル `tinet_id.h` に、対応するマクロ定義が出力される。

```
#define TCP_RSV_CEPID1 1
```

TCP 通信端点の動的生成と削除の使用例[2]

- TCP 通信端点生成情報構造体に情報を設定する。

```
T_TCP_CCEP ccep;  
ccep.cepatr = 0;  
ccep.sbufsz = TCP_ECHO_SRV_SWBUF_SIZE;  
ccep.rbufsz = TCP_ECHO_SRV_RWBUF_SIZE;  
ccep.sbuf = tcp_echo_srv_swbuf;  
ccep.rbuf = tcp_echo_srv_rwbuf;  
ccep.callback = (FP)callback_nblk_tcp_echo_srv;
```

TCP 通信端点の動的生成と削除の使用例[3]

- TCP 通信端点の生成 API により、TCP 通信端点を動的に生成する。

```
ercd = tcp_cre_cep(TCP_RSV_CEPID1, &ccep);
```

- これ以降は、TCP_RSV_CEPID1 を、静的 API で生成したTCP 通信端点の cepid と区別することなく TCP の API で使用することが出来る。

TCP 通信端点の動的生成と削除の使用例[4]

- TCP 通信端点の削除 API により、TCP 通信端点を動的に削除する。

```
ercd = tcp_del_cep(TCP_RSV_CEPID1);
```

- これ以降は、TCP_RSV_CEPID1 を、他の目的で使うことができる。

ITRON TCP/IP API 拡張機能(4)

■ TCP 通信端点オプションの設定と読出し

```
ER ercd = tcp_set_opt(ID cepid, INT optname,  
                      VP optval, INT optlen);
```

```
ER ercd = tcp_get_opt(ID cepid, INT optname,  
                      VP optval, INT optlen);
```

■ UDP 通信端点オプションの設定と読出し

```
ER ercd = udp_set_opt(ID cepid, INT optname,  
                      VP optval, INT optlen);
```

```
ER ercd = udp_get_opt(ID cepid, INT optname,  
                      VP optval, INT optlen);
```

TCP/UDP 通信端点オプションの実装

- ITRON TCP/IP API 仕様では、TCP/UDP 通信端点オプションの種類と機能は実装依存である。
- 現在 TINET では、定義していないので、設定 API、読出し API とも、戻り値はエラーとなる。

ITRON TCP/IP API 拡張機能(5)

■ 緊急データの送信

```
ER_UINT ercd = tcp_snd_oob(ID cepid, VP data,  
                             INT len, TMO tmout);
```

■ 緊急データの受信

```
ER_UINT ercd = tcp_rcv_oob(ID cepid, VP data,  
                             INT len);
```

■ tcp_rcv_oob は、緊急データ受信のコールバック関数内で呼出すことが想定されている。

■ 事象の種類は TEV_TCP_RCV_OOB である。

緊急データ送受信 API

■ 緊急データ送受信に関する制約 (実装にもよりますが)

- 緊急データだからといって、すでに送信ウィンドバッファにある通常データより先に送信されるわけではない。
- 緊急データ送信 API で、複数バイトのデータを送信しても、受信側で受信できるのは、送信されたデータの最後の 1 バイトのみである。
 - これより前のデータは通常データとして受信される。

緊急データ送受信 API

■ コンパイル時コンフィギュレーションパラメータ

- 緊急データの最後のバイトのオフセットに関して、RFC1122 と TINET が参考実装としている BSD で異なっている (BSD のバク?)。
- パラメータ TCP_CFG_URG_OFFSET で、オフセット値を指定できる。
 - 1 BSD の実装と同じで、緊急ポインタは、緊急データの最後のバイトの次のバイトを指す。
 - 0 RFC1122 の規程と同じで、緊急ポインタは、緊急データの最後のバイトを指す。

ITRON TCP/IP API 拡張機能のメモリ必要量

■ IPv4、RAM

機 能	RAM			
	標準	拡張	増減数	増減率
TCP	1,914	1,974	60	3.13%
UDP	114	122	8	7.02%
ICMPv4	0	0	0	0.00%
IPv4	2	2	0	0.00%
ARP	160	160	0	0.00%
TCP/IP 共通機能	100	100	0	0.00%
汎用通信機能	1,170	1,170	0	0.00%
イーサネット	2,120	2,120	0	0.00%
ネットワークバッファ	11,296	11,296	0	0.00%
小 計	16,876	16,944	68	0.40%
応用プログラム	13,570	13,602	32	0.24%
カーネル	5,526	5,606	80	1.45%
合 計	35,972	36,152	180	0.50%

ITRON TCP/IP API 拡張機能のメモリ必要量

■ IPv4、ROM

機 能	ROM			
	標準	拡張	増減数	増減率
TCP	25,886	28,340	2,454	9.48%
UDP	4,002	4,814	812	20.29%
ICMPv4	1,400	1,400	0	0.00%
IPv4	998	998	0	0.00%
ARP	2,504	2,504	0	0.00%
TCP/IP 共通機能	3,570	3,570	0	0.00%
汎用通信機能	2,800	2,802	2	0.07%
イーサネット	4,014	4,014	0	0.00%
ネットワークバッファ	1,684	1,684	0	0.00%
小 計	46,858	50,126	3,268	6.97%
応用プログラム	24,228	28,482	4,254	17.56%
カーネル	32,760	33,042	282	0.86%
合 計	103,846	111,650	7,804	7.51%

ITRON TCP/IP API 拡張機能のメモリ必要量

■ IPv6、RAM

機 能	RAM			
	標準	拡張	増減数	増減率
TCP	2,250	2,302	52	2.31%
UDP	138	146	8	5.80%
近隣探索	680	680	0	0.00%
ICMP	0	0	0	0.00%
IP	4	4	0	0.00%
TCP/IP 共通機能	172	172	0	0.00%
汎用通信機能	1,194	1,194	0	0.00%
イーサネット	2,356	2,356	0	0.00%
ネットワークバッファ	11,312	11,312	0	0.00%
小 計	18,106	18,166	60	0.33%
応用プログラム	13,622	13,654	32	0.23%
カーネル	5,558	5,638	80	1.44%
合 計	37,286	37,458	172	0.46%

ITRON TCP/IP API 拡張機能のメモリ必要量

■ IPv6、ROM

機 能	ROM			
	標準	拡張	増減数	増減率
TCP	26,190	28,492	2,302	8.79%
UDP	4,010	4,834	824	20.55%
近隣探索	17,186	17,186	0	0.00%
ICMP	2,806	2,806	0	0.00%
IP	3,886	3,886	0	0.00%
TCP/IP 共通機能	6,888	6,888	0	0.00%
汎用通信機能	3,000	2,998	-2	-0.07%
イーサネット	4,528	4,528	0	0.00%
ネットワークバッファ	1,684	1,684	0	0.00%
小 計	70,178	73,302	3,124	4.45%
応用プログラム	26,646	31,150	4,504	16.90%
カーネル	33,006	33,066	60	0.18%
合 計	129,830	137,518	7,688	5.92%

TINET-1.4 の IPv6 アドレス自動設定機能

- IPv6 アドレス数は、コンパイル時コンフィギュレーションパラメータにより設定可能とした。
- デフォルトルータ・リストの管理を実装した。
- プレフィックス・リストの管理を実装した。
- コンパイル時コンフィギュレーションパラメータ
 - NUM_ND6_DEF_RTR_ENTRY
デフォルトルータ・リストのエントリ数
 - NUM_ND6_PREFIX_ENTRY
プレフィックス・リストのエントリ数

IPv6 データグラムの分割・再構成機能

- 拡張ヘッダの断片ヘッダの実装した。
 - IPv4 データグラムの分割・再構成機能の実装と基本的に同じ。
- コンパイル時コンフィギュレーションパラメータ
 - IP6_CFG_FRAGMENT この機能を組み込む
 - NUM_IP6_FRAG_QUEUE 再構成キューのサイズ
 - TMO_IP6_FRAG_GET_NET_BUF
ネットワークバッファ獲得タイムアウト
 - IP6_CFG_FRAG_REASSM_SIZE
再構成用ネットワークバッファのサイズ

Path MTU への対応

■ ホストキャッシュを実装した。

■ キャッシュエントリの内容

- 目的ネットワークアドレス

- 有効時間

- Path MTU

- IP のみサポートで、TCP などへの対応は今後の課題

■ コンパイル時コンフィギュレーションパラメータ

- NUM_IN6_HOSTCACHE_ENTRY

ホストキャッシュのエントリ数

IPv6 Ready Logo Program 対応のメモリ必要量

■ 断片ヘッダとホストキャッシュの有効化、RAM

機能	RAM			
	標準	RL対応	増減数	増減率
TCP	2,250	2,250	0	0.00%
UDP	138	138	0	0.00%
近隣探索	680	680	0	0.00%
ICMP	0	0	0	0.00%
IP	4	56	52	1300.00%
TCP/IP 共通機能	172	272	100	58.14%
汎用通信機能	1,194	1,194	0	0.00%
イーサネット	2,356	2,356	0	0.00%
ネットワークバッファ	11,312	19,528	8,216	72.63%
小計	18,106	26,474	8,368	46.22%
応用プログラム	13,622	13,622	0	0.00%
カーネル	5,558	5,606	48	0.86%
合計	37,286	45,702	8,416	22.57%

IPv6 Ready Logo Program 対応のメモリ必要量

■ 断片ヘッダとホストキャッシュの有効化、ROM

機 能	ROM			
	標準	RL対応	増減数	増減率
TCP	26,190	26,190	0	0.00%
UDP	4,010	4,010	0	0.00%
近隣探索	17,186	17,186	0	0.00%
ICMP	2,806	3,334	528	18.82%
IP	3,886	7,262	3,376	86.88%
TCP/IP 共通機能	6,888	7,500	612	8.89%
汎用通信機能	3,000	3,000	0	0.00%
イーサネット	4,528	4,528	0	0.00%
ネットワークバッファ	1,684	1,684	0	0.00%
小 計	70,178	74,694	4,516	6.44%
応用プログラム	26,646	26,646	0	0.00%
カーネル	33,006	33,034	28	0.08%
合 計	129,830	134,374	4,544	3.50%

各オブジェクトのメモリ容量

オブジェクト	IPv4	IPv6
TCP受付口	12	24
TCP通信端点	212	244
UDP通信端点	40	52
TW用TCP通信端点	36	72
分割・再構成	12	24
ホストキャッシュ		24
デフォルトルータ		28
プレフィックス		36
近隣探索キャッシュ		40
ARPキャッシュ	16	

今後の課題

■ IPv6 関連

- PPP (IPCPv6)
- IPv6 IPSec 仕様
- Mobile IPv6 仕様
- IPv4/IPv6 デュアルスタック

■ IPSec 仕様

■ ソケットインタフェースへの対応

TINETの配布状況

- TOPPERS ライセンス (BSD ライセンス制約)
- 配布開始は、2003 年 7 月
- ダウンロード数は、2007/10末現在で、約10,100 件
- 最新リリースは、1.4
- 配布場所
 - 宮城県産業技術総合センター様
 - <http://www.mit.pref.miyagi.jp/embedded/consortium/>
 - TOPPERSプロジェクト
 - <http://www.toppers.jp/>