

TINET 変更メモ [2009/12/15]

1. リリース 1.0 から 1.1 までの変更

- (1) TINET コンフィグレータの生成で、リンクを不要にした。
- (2) ファイルのインクルードを見直し、`cpu_config.h` から `tinnet_config.h`、`cpu_defs.h` から `tinnet_defs.h` を削除した。
- (3) アプリケーションの Makefile の指定内容を一部変更した。
 - [1] TCP の組み込み、`SUPPORT_TCP` の指定。
 - [2] UDP の組み込み、`SUPPORT_UDP` の指定。
- (4) サンプル `nserv.c` から WWW サーバ機能を `tinnet/netapp/wwws.c` に移した。
- (5) JSP システムコンフィギュレーションファイル `$(UNAME).cfg` では、`tinnet/tinnet.cfg` を指定するだけで、プロトコルスタック内部のカーネルオブジェクトを取り込めるように変更した。
- (6) TINET プロトコルスタック内部で使用するカーネルオブジェクトの指定ファイル `tinnet/tinnet.cfg` での「INCLUDE」静的 API を、各モジュールの JSP システムコンフィギュレーションファイルに移動した。
- (7) `tinnet/include` は廃止し、インクルードファイルを以下のように移動した。
 - [1] `tinnet/include/net` `tinnet/net`。
 - [2] `tinnet/include/netdev` `tinnet/netdev`。
 - [3] `tinnet/include/netinet` `tinnet/netinet`。
 - [4] `tinnet/include/netapp` `tinnet/netapp`。
 - [5] `tinnet/include/tinnet_config.h` `tinnet/tinnet_config.h`。
 - [6] `tinnet/include/tinnet_defs.h` `tinnet/tinnet_defs.h`。
- (8) ネットワークインタフェースのデバイス依存部を独立するための変更を行った。
 - [1] `tinnet/netdev` に、ネットワークインタフェース依存部を配置するディレクトリを新設した。NE2000 互換 NIC では `if_ed` であり、以下この NIC の設定について述べる。
 - [2] 汎用の JSP システムコンフィギュレーションファイル `nic.cfg` を新設した。現在は、`if_ed.cfg` をインクルードするようになっている。
 - [3] 汎用のネットワークインタフェース・コンフィギュレーションファイル `tinnet_nic_config.h` を新設した。現在は、汎用のネットワークインタフェース・別名化ファイル `nic_rename.h` と `if_ed.h` をインクルードするようになっている。
 - [4] 汎用のネットワークインタフェース・別名化ファイル `nic_rename.h` を新設した。
 - [5] `if_ed.c` と `if_ed.cfg` 以外のファイルから `if_ed.h` と `if_edreg.h` を削除した。
 - [6] `tinnet/tinnet_config.h` の変更
#include <tinnet_nic_config.h> を追加し、インクルードの順番を見直した。

#include "tinnet_app_config.h"
#include "tinnet_sys_config.h"
#include "tinnet_cpu_config.h"

```

#ifdef SUPPORT_ETHER
#include "tinet_nic_config.h"
#endif /* of #ifdef SUPPORT_ETHER */
#include "tinet_sys_config.h"
#include "tinet_cpu_config.h"
#include "tinet_app_config.h"

```

[7] tinet/net/ethernet.h の変更

- ・ tinet/netdev/if_ed/if_ed.h の t_ed_softc から、デバイス非依存部を t_if_softc として独立して定義した。

```

struct t_if_softc {
    T_IF_ADDR        ifaddr;
    UH               timer;
    T_IF_ETHER_SOFTC *sc;
    ID               semid_txb_ready;
    ID               semid_rxb_ready;
};
typedef struct t_if_softc T_IF_SOFTC;

```

- ・ メンバ timer は、デバイス非依存部に移動した。
- ・ 「インタフェース選択マクロ」から T_IF_SOFTC を削除した。

[8] tinet/net/ethernet.c、tinet/net/if.c、tinet/netinet/tcp_subr.c の変更

#include <netdev/if_ed.h> を削除し、T_IF_ETHER_SOFTC から、T_IF_SOFTC に変更した。

```

T_IF_ETHER_SOFTC *sc;

T_IF_SOFTC *ic;

```

[9] tinet/netdev/if_ed/if_ed.h の変更

- ・ t_ed_softc 定義を tinet/netdev/if_ed/if_ed.c に移動した。
- ・ 「NIC 選択マクロ」の T_IF_ETHER_SOFTC を T_ED_SOFTC から struct t_ed_softc に変更した。
- ・ ネットワークインタフェースの関数の引数を T_ED_SOFTC から T_IF_SOFTC に変更した。
- ・ 関数 ed_watchdog の引数に、T_IF_SOFTC* を指定するように変更した。

[10] tinet/netdev/if_ed/if_ed.c の変更

- ・ #include <netdev/if_ed.h> を削除した。
- ・ if_edreg.h のインクルード指定の変更

```

#include <netdev/if_edreg.h>

#include "if_edreg.h"

```

- ・ t_ed_softc の定義を、tinet/netdev/if_ed/if_ed.h から移動した。
- ・ t_ed_softc は、デバイス依存部のみ定義するように変更した。
- ・ 以下の局所関数の引数を T_ED_SOFTC から T_IF_SOFTC に変更した。

```

ed_xmit
ed_init_sub

```

- ・関数の引数が T_ED_SOFTC から T_IF_SOFTC に代わったが、

```
T_ED_SOFTC *sc = ic->sc;
```

で、T_ED_SOFTC を取り出すことが出来る。

[11] tinet/netdev/if_ed/if_ed.cfg の変更

```
INCLUDE("<netdev/if_ed.h>");
INCLUDE("<netdev/if_edreg.h>");

INCLUDE("\"if_edreg.h\"");
```

[12] tinet/net_rename.h の変更

if_ed の関数のリネーム定義を、tinnet/netdev/if_ed/nic_rename.h に移動した。

[13] tinet/tinet.cfg の変更

```
#ifdef IF_ETHER_CFG_ED
#include "netdev/if_ed.cfg"
#endif /* of #ifdef IF_ETHER_CFG_ED */

#include "nic.cfg"
```

[14] tinet_cpu_config.h と tinet_sys_config.h の変更

「NIC (NE2000 互換) に関する定義」を tinet_sys_config.h に移動した。

[15] \$(CPU)/\$(SYS)/Makefile.config の変更

- ・「コンパイルオプション」の変更

ネットワークインタフェースのディレクトリを含めるため、INCLUDES の定義の後に追加した。

```
ifdef TINET_ROOT
INCLUDES := $(INCLUDES) -I$(TINET_ROOT)/netdev/if_ed
endif
```

- ・「カーネルに関する定義」の変更

ネットワークインタフェースのディレクトリとネットワークインタフェースのオブジェクトを含めるため、KERNEL_COBJS の定義の後に追加した。

```
ifdef TINET_ROOT
KERNEL_DIR := $(KERNEL_DIR):$(TINET_ROOT)/netdev/if_ed
KERNEL_COBJS := $(KERNEL_COBJS) if_ed.o
endif
```

[16] 各ソースから IF_ETHER_CFG_ED を削除した。

[17] if_ed.h と if_edreg.h で、SUPPORT_ETHER を削除した。

[18] 各ソースの PROTO_FLG_ETHER_ED を PROTO_FLG_ETHER_NIC に変更した。

[19] 各ソースの net_count_ether_ed 及び net_count_ether_ed* を
net_count_ether_nic 及び net_count_ether_nic* に変更した。

[20] 各ソースの NET_COUNT_ETHER_ED 及び NET_COUNT_ETHER_ED* を
NET_COUNT_ETHER_NIC 及び NET_COUNT_ETHER_NIC* に変更した。

[21] 各ソースの IF_ETHER_CFG_ACCEPT_ALL を ETHER_CFG_ACCEPT_ALL に変更した。

[22] 各ソースの IF_ETHER_* を IF_ETHER_NIC_* に変更した。

(9) tinet/cfg/tinet_parser.cpp の変更

[1] tinet_cfg.c に出力する標準のインクルードファイルに tinet_config.h を追加した。

[2] tinet_cfg.c に出力する標準のインクルードファイルのインクルード順番を以下のように変更した。

```
tinet_defs.h
tinet_config.h
```

[3] tinet_cfg.c に出力する標準のインクルードファイルで、以下のファイル指定を削除した。

```
#include <netinet/in_rename.h>
```

(10) tinet/tinet_defs.h に tinet_nic_defs.h を追加した。

(11) tinet/netdev/if_ed に tinet_nic_defs.h を追加した。

(12) 各ソースファイルに、tinnet_defs.h、tinnet_config.h を追加した。

(13) 各ソースファイルで、tinnet_defs.h、tinnet_config.h の順番にインクルードするように変更した。

(14) tinet_nic_defs.h に IF_ETHER_NIC_HDR_ALIGN を追加した。これは、以下のファイルにあるイーサネットヘッダの定義で、最初にアラインを調整する場合に、調整量を指定する。

```
tinnet/net/ethernet.h
```

(15) TINET プロトコルスタック内部のリネーム指定ファイル tinet/net/net_rename.h と tinet/netinet/in_rename.h のインクルードを、tinnet/tinet_config.h で指定するように変更し、各ソースファイルから以下のファイルを削除した。

```
#include <net/net_rename.h>
#include <netinet/in_rename.h>
```

(16) TINET の標準 Makefile.config を作成し、アプリケーションの Makefile からインクルードするように変更した。

(17) tinet/netdev/if_ed/if_ed.c の以下の関数で、ディスパッチ禁止状態にならないように変更した。

```
ed_pio_readmem
ed_pio_writemem
```

(18) システム時刻 (SYSTIM) の周波数、SYSTIM_HZ を定義した。

(19) cygwin で g++ のバージョンによって、TINET コンフィギュレータの生成時にエラーが発生する問題を修正した。

(20) Makefile.config における指定の変更

[1] tinet/netdev/if_ed/if_ed.c に Makefile.config を作成し、TINET の標準 Makefile.config からインクルードするように変更した。

[2] JSP のシステム依存部の \$(CPU)/\$(SYS)/Makefile.config から、TINET のネットワーク・デバイスの依存部の定義を削除した。

[3] アプリケーションの Makefile には、

```
#   イーサネット・デバイスドライバの選択
NET_DEV = if_ed
```

を指定する必要がある。

(21) tinet/netapp/netapp.h に、コンソールに用いるシリアルポート番号の定義を追加した。

```
/*
 *   コンソールに用いるシリアルポート番号
 */
#ifndef CONSOLE_PORTID
#define CONSOLE_PORTID LOGTASK_PORTID
#endif /* of #ifndef CONSOLE_PORTID */
```

(22) tcp_cls_cep() で、仕様では、FIN を送ってコネクションを切断することになっているが、正しく実装していない問題を修正した。

(23) tinet_cpu_config.h で定義している受信セグメントサイズの最大値 (MAX_TCP_RCV_SEG)、送信セグメントサイズの最大値 (MAX_TCP_SND_SEG)、送信セグメントサイズの規定値 (DEF_TCP_SND_SEG)、受信セグメントサイズの規定値 (DEF_TCP_RCV_SEG) の値に関して、イーサネットを使用する場合、全ての値を 1,460 に変更した。

(24) サンプルアプリケーションとして、TCP ディスカードクライアントを追加した。

(25) TCP の MSS (TCP_MSS) に関して、

```
#define TCP_MSS (IP4_MSS - (IP4_HDR_SIZE + TCP_HDR_SIZE))
```

から、BSD の標準値

```
#define TCP_MSS 512
```

に変更した。

(26) tcp_con_cep で、タイムアウトにノンブロッキング以外を指定した場合、タイムアウト前にエラーになる可能性がある問題を修正した。

(27) ネットワークタイマで、複数のエントリが同時にタイムアウトしても、登録されている全ての関数を呼び出さない問題を修正した。

(28) ブロードキャスト送信を正しく処理していなかった問題を修正した。次の場合は、イーサネットの送信先 MAC にはブロードキャストアドレスを設定する。

- ・ IP アドレスの全ビットが 1
- ・ IP アドレスのホスト部の全ビットが 1 で、ネットワーク部がローカルアドレス

同時に、IP アドレスの全ビットが 1 のブロードキャストに対応するため、静的経路表 (route_cfg.c) には、最後のエントリとして以下の行を追加した。

```
{ 0xffffffff, 0xffffffff, 0 },
```

また、コンパイル時コンフィギュレーションパラメータ NUM_ROUTE_ENTRY を 1 増加した。

(29) 関数 ntohs() を使うべきところで関数 ntohl() を使っていた問題を修正した。

- (30) IP 及び UDP のチェックサムの値を関数 `htons()` を使って、代入していた問題を修正した。
- (31) `(SIL_ENDIAN == SIL_ENDIAN_LITTLE の、htonl() htons() の定義誤りを修正した。`
- (32) UDP の入力関数 `udp_input` で、宛先アドレスとして正しいかチェックする関数 `dstaddr_accept` を呼出す前に、ネットワークバイトオーダーからホストバイトオーダーに変換していない問題を修正した。
- (33) NE2000 互換 NIC の送受信バッファページサイズの定義誤りを修正した。
- (34) NE2000 互換 NIC で、NIC の内蔵 SRAM へのデータの書込み終了後、不要のため書込み完了を待たないように変更した。
- (35) UDP で、ブロードキャスト・データグラムも受信するように修正した。
- (36) コンパイル時コンフィギュレーションパラメータ `DHCP_CFG` を新設した。これは、DHCP メッセージを受信する場合に指定する。ただし、TINET は、DHCP を実装していない。応用プログラムで、DHCP メッセージを受信するための定義である。`DHCP_CFG` を指定すると、`IPV4_ADDR_LOCAL` が未定義（全て 0）でも、UDP に限り、DHCP メッセージを受信することが可能になる。
- (37) コンパイル時コンフィギュレーションパラメータ `IPV4_ADDR_LOCAL_BC` を削除し、自分の IP アドレスのサブネットマスクを指定する `IPV4_ADDR_LOCAL_MASK` を新設した。
- (38) ネットワークインタフェースに IPv4 アドレスを追加する関数 `in4_add_ifaddr` を新設した。ただし、この関数は ITRON TCP/IP API 非互換であり、ネットワークインタフェースがイーサネットのときのみ有効である。
- (39) 経路表にエントリを追加する関数 `in4_add_route` を新設した。ただし、この関数は ITRON TCP/IP API 非互換であり、ネットワークインタフェースがイーサネットのときのみ有効である。
- (40) ARP キャッシュで、空きがなかった場合、タイムアウトまで時間が最短のエントリを探索する部分の問題を修正した。
- (41) コンパイル時コンフィギュレーションパラメータ `ETHER_NIC_CFG_RELEASE_NET_BUF` を新設した。これはイーサネット出力時に、NIC で `net_buf` を開放する場合に指定する。
- (42) ARP 入力で、IP アドレスの重複検出を行うように変更した。検出した場合は、重複相手の MAC アドレスを `syslog` に出力し、重複相手にも重複したことを伝える。
- (43) コンパイル時コンフィギュレーションパラメータ `ARP_CFG_CALLBACK_DUPLICATED` を新設した。これを指定すると、ARP 入力で IP アドレスの重複を検出した時に、応用プログラムで用意したコールバック関数 `arp_callback_duplicated` を呼び出す。
- (44) IP 出力時に、ARP キャッシュのタイムアウト値を更新していたが、FreeBSD の実装にあわせて、更新しないように変更した。ARP キャッシュのタイムアウト値は ARP 入力でのみ更新される。

2. リリース 1.1.x から 1.2 までの変更

- (1) TINET コンフィギュレーションファイルで、TCP 受付口を指定しなかったときの問題は、`tnet/cfg/tnet_parser.cpp` で、`tnet_cfg.c` にダミーの TCP 受付口変数を出力するように修正した。
- (2) UDP のみ使用するときの `kernel_cfg.c` で、コンパイルエラーが発生する問題は、`tnet/netinet/tcp.cfg` から

```
INCLUDE("<netinet/in.h>");
```

を削除し、`tnet/netinet/ip.cfg` に

```
INCLUDE("<netinet/in.h>");
```

を追加した。
- (3) UDP のみ使用するときの `ip_icmp.c` で、`_tnet_tcp_notify` の参照エラーが発生する問題は、`tnet/netinet/ip_icmp.c` で、ICMP UNREACH メッセージを無視するように修正した。
- (4) ループバック・インタフェースの IP アドレスマスク、ローカルブロードキャストアドレスの定義誤りを修正した。
- (5) `tnet/netdev/if_ed/if_ed.c` で、レジスタを SIL 経由でアクセスするように修正した。
- (6) IPv6 に対応した。
- (7) コンパイル時コンフィギュレーションパラメータに、デフォルト・ゲートウェイの IP アドレスを指定する `IPV4_ADDR_LOCAL_DEFAULT_GW` を新設した。これにより、デフォルト・ゲートウェイのみのネットワークでは、静的ルーティング情報設定ファイル `route_cfg.c` の変更を不要にした。
- (8) UDP の受信時、受信ポート番号が未定義の場合、ネットワークバッファを正しく解放しない問題と、ICMP のエラー通知を正しく行わない問題を修正した。
- (9) TCP の受信時、受信ポート番号が未定義の場合、リセットセグメントを正しく返送しない問題を修正した。
- (10) ネットワーク統計情報 (`net_count`) の各変数を `_tnet_` でリネームしないように変更した。
- (11) バイトオーダ変換の定義を `netinet/in.h` から、`net/net.h` に集約し、リトルエンディアン用のマクロを追加した。
- (12) コンパイル時コンフィギュレーション・ファイルのいずれかで、マクロ `SUPPORT_MIB` を定義することにより、SNMP 用管理情報ベース (MIB) に準拠したネットワーク統計の取得が可能となった。ただし、TINET 自体は、管理情報ベース (MIB) に準拠したネットワーク統計を提供するだけで、SNMP をサポートしていない。また、RFC1213、RFC2465、RFC2466 に定義されている全ての情報が取得できるわけではない。取得できる情報は、関係するインクルードファイルの構造体の定義を参照すること。
- (13) PPP の HDLC 入力における問題を修正した。

- (14) 認識できない上位プロトコルまたは拡張ヘッダを受信したとき、コード番号 ICMP6_PARAMPROB_NEXTHEADER (1) の ICMP6_PARAM_PROB (4) メッセージを返送するよう変更した。

3. リリース 1.2.x からリリース 1.3 までの変更

- (1) IPv6 で、ネットワークインタフェースにループバックインタフェースを指定したとき、コンパイルエラーが発生する問題を修正した。
- (2) チェックサム計算でエラーになる問題を修正した (TOPPERS/TINET バグトラックスystem ID:1)。
- (3) echos で、TCP_ECHO_SRV_STACK_SIZE の二重定義警告が出る問題を修正した (TOPPERS/TINET バグトラックスystem ID:2)。
- (4) ネットワークバッファの取得失敗の時、常に警告を出力するように変更した (TOPPERS/TINET バグトラックスystem ID:3)。
- (5) echos/echos の問題を修正した (toppers-users 1724)。
- (6) IP4_CFG_FRAGMENT を指定したとき、tinet/netapp/dbg_cons.c と tinet/netinet/ip_output.c で、コンパイルエラーになる問題を修正した。
- (7) 拡張ヘッダの付いた ICMPv6 エコーリクエストに対する応答の問題を修正した。
- (8) 近隣探索の到達可能性の確認でタイムアウトした後、アドレス解決が出来ない問題を修正した。
- (9) TCP の入力セグメントの再構成関数の潜在的な問題点を修正した。
- (10) UDP のみ使用するときの icmp6.c で、_tinnet_tcp_notify の参照エラーが発生する問題を修正した。
- (11) UDP の送信で、タイムアウトしたとき、ネットワークバッファを開放しない問題を修正した。
- (12) TCP の応答で、IP のペイロードサイズを誤る問題を修正した (TOPPERS/TINET バグトラックスystem ID:5)。
- (13) udp_rcv_dat で、タイムアウトに時間を指定したとき、タスクの実行がブロックされる問題を修正した。
- (14) ネットワーク統計情報の計測の変数サイズを、_int64_ が定義されている場合のみ UD となるように変更した。
- (15) JSP リリース 1.4.2 に暫定対応した。変更内容を以下に示す。
 - [1] H8 依存部の割込みレベル導入に伴い、if_ed イーサネット・デバイスドライバの割込み処理を実装する config/h8/akih8_3069f/tinet_sys_config.c を追加した。
 - [2] ed_ena_inter、ed_dis_inter の引数と戻り値を変更した。
 - [3] ed_ena_inter、ed_dis_inter の引数と戻り値の変更に伴い、tinnet/netdev/if_ed/if_ed.c も変更した。
 - [4] ed_ena_inter、ed_dis_inter の定義を、tinnet/netdev/if_ed/if_edreg.h から、config/h8/akih8_3069f/tinet_sys_config.c に変更した。
- (16) tcp_cls_cep で、タイムアウトした後、tcp_acp_cep がエラーになる問題を修正した。
- (17) 誤った IP SDU 長の TCP リセットセグメントを送信する問題を修正した。
- (18) TCP のシーケンス番号がランダムに初期化されない問題を修正した。

(19) リトルエンディアンのプロセッサで、IPv6 ヘッダのバージョン、トラフィッククラス、フローラベルフィールドを誤って参照・変更する問題を修正した。

(20) リトルエンディアンのプロセッサにおける問題に対応するための変更内容を以下に示す。

[1] `netinet/net/net.h` で、コンフィギュレーション時にエラーが発生する問題を修正した。

[2] ルータ通知の受信で、有効時間を誤る問題を修正した。

(21) JSP リリース 1.4.2 に対応した。変更内容を以下に示す。

[1] 割込みベクタの生成でエラーとなる問題を修正した。

[2] NE2000 互換 NIC のデバイスドライバ `if_ed` において、システム依存でバスの初期化を行う `ed_bus_init` と、割込みの初期化を行う `ed_inter_init` を新設した。

[3] 品川通信計装サービス製 NKEV-010H8 (H8/3069F) をサポートした。

(22) TCP において、送信側で再構成され、前後と重なりがあるセグメントの再構成が正しく行われない問題を修正した (TOPPERS/TINET バグトラックスystem ID:8)。

(23) 以下のコンパイル時コンフィギュレーションパラメータを変更した。

`NUM_MPF_NET_BUF_4096`

`NUM_MPF_NET_BUF_REASSM`

ただし、`netinet_config.h` で、リリース 1.2 との互換性を確保してある。

(24) ネットワークバッファの各長さの指定で、無指定または 0 を指定した場合は、対応する固定メモリプールを割当てないように変更した。

(25) コンパイル時コンフィギュレーションパラメータに、再構成バッファサイズを指定する `IP4_CFG_FRAG_REASSM_SIZE` を新設した。

(26) ネットワークバッファに新しいサイズを追加した。

[1] `T_NET_BUF_IPV6_CSEG`、サイズはネットワークインタフェースのヘッダサイズ + IP ヘッダサイズ + TCP ヘッダサイズ

[2] `T_NET_BUF_IP_MSS` で、サイズは `IP_MSS` (ネットワークインタフェースのヘッダサイズ + 576 オクテット)。

[3] `T_NET_BUF_IPV6_MMTU`、サイズは `IPV6_MMTU` (ネットワークインタフェースのヘッダサイズ + 1280 オクテット)。

(27) TINET 内部のタイムアウト値、タスクのスタックサイズ、内部のタスクの優先度、データキューのエントリ数、及び以下のコンパイル時コンフィギュレーションパラメータは、`netinet/tinet_config.h` で定義するように変更した。

`MAX_TCP_RCV_SEG`

`DEF_TCP_SND_SEG`

`ARP_CACHE_KEEP`

`LOMTU`

(28) TCP の受動オープンをサポートする場合に指定する `SUPPORT_TCP_POPEN` を追加した。

- (29) コンパイル時コンフィギュレーションパラメータに、NUM_TCP_TW_CEP_ENTRY を追加した。
これは、タスクから Time Wait 状態の TCP 通信端点を分離する機能で、Time Wait 状態の TCP 通信端点のエントリ数を指定する。指定しないか、0 を指定するとタスクから Time Wait 状態の TCP 通信端点を分離する機能を組込まない。
- (30) TCP の 受 動 オ ー プ ン を サ ポ ー ト す る 場 合 に 指 定 す る SUPPORT_TCP_POPEN は TCP_CFG_PASSIVE_OPEN のドキュメント誤り。
- (31) tinet コンフィギュレータが生成する tinet_cfg.c の TCP_CRE_CEP に対応する構造体 T_TCP_CCEP のフィールド sbuf と rbuf の値の前に (VP) を、フィールド callback の値の前に (FP) を出力するように変更した。
- (32) tinet コンフィギュレータが生成する tinet_cfg.c の TCP6_CRE_CEP に対応する構造体 T_TCP6_CCEP のフィールド sbuf と rbuf の値の前に (VP) を、フィールド callback の値の前に (FP) を出力するように変更した。
- (33) tinet コンフィギュレータが生成する tinet_cfg.c に
- ```
#include <netinet/in_itron.h>
```
- を出力するように変更した。
- (34) TCP 通信端点で、送受信の二つのセマフォをイベントフラグにより実装した。
- (35) TCP 通信端点の送受信ウィンドバッファの省コピー機能を実装した。
- (36) TCP 通信端点の送受信ウィンドバッファの省コピー機能用に、以下に示すコンパイル時コンフィギュレーションパラメータを追加した。
- [1] TCP\_CFG\_RWBUF\_CSAVE\_ONLY、TCP 通信端点の受信ウィンドバッファの省コピー機能を組み込み、この機能のみ使用する。
  - [2] TCP\_CFG\_RWBUF\_CSAVE、TCP 通信端点の受信ウィンドバッファの省コピー機能を組込む。
  - [3] TCP\_CFG\_SWBUF\_CSAVE\_ONLY、TCP 通信端点の送信ウィンドバッファの省コピー機能を組み込み、この機能のみ使用する。
  - [4] TCP\_CFG\_SWBUF\_CSAVE、TCP 通信端点の送信ウィンドバッファの省コピー機能を組込む。
  - [5] TCP\_CFG\_RWBUF\_CSAVE\_MAX\_QUEUES、TCP 通信端点の受信ウィンドバッファの省コピー機能の、受信ウィンドバッファキューの最大エントリ数。ただし、正常に受信したセグメントも破棄するため、再送回数が増加する。また、指定しないと制限しない。標準値は 2 である。
  - [6] TCP\_CFG\_SWBUF\_CSAVE\_MAX\_SIZE、TCP 通信端点の送信ウィンドバッファの省コピー機能で、送信ウィンドバッファに使用するネットワークバッファの最大サイズ。標準値は IF\_PDU\_SIZE である。
  - [7] TCP\_CFG\_SWBUF\_CSAVE\_MIN\_SIZE、TCP 通信端点の送信ウィンドバッファの省コピー機能で、送信ウィンドバッファに使用するネットワークバッファの最大サイズ。標準値は 0 である。
- (37) TCP と UDP のノンブロッキングコール機能の組み込みを指定するコンパイル時コンフィギュレーションパラメータを追加した。
- [1] TCP\_CFG\_NON\_BLOCKING、TCP のノンブロッキングコール機能を組込む。

- [2] UDP\_CFG\_NON\_BLOCKING、UDP のノンブロッキングコール機能を組込む。
- (38) netinet/in.h に、TINET のバージョン情報 TINET\_PRVER を定義した。
- (39) netinet/in.h に、8 ビット毎に指定した IPv4 アドレスを 32 ビットにするマクロ MAKE\_IPV4\_ADDR を追加した。
- (40) TCP と UDP の一部をライブラリ化した。
- (41) アプリケーションの Makefile にインクルードする TINET の Makefile.config を Makefile.tinet に変更した。
- (42) int のサイズが 16 ビットの処理系に対応した（動作確認は、（株）ルネサス製コンパイラ H8C で行った）。変更内容を以下に示す。
  - [1] tcp\_seq.h のインライン関数の記述を変更した。
  - [2] 関数 timeout と関数 untimeout の第 1 引数を (FP) 型に明示的にキャストした。
  - [3] マクロ内でシフト演算のオペランドとして使用されている引数の方を明示的にキャストした。
  - [4] マクロ GET\_TCP\_CREPID、GET\_TCP\_CCEPID、GET\_TCP\_TW\_CCEPID、GET\_UDP\_CCEPID の戻り値を ID 型に明示的にキャストした。
  - [5] tinet/net/net\_buf.h、ネットワークバッファのテンプレートの定義 struct t\_net\_buf で、バッファ本体の大きさを buf[4] に変更した。
  - [6] 乱数関数 rand() の関数名を net\_rand() に変更し、戻り値の型を UW に変更した。
  - [7] 乱数初期化関数 srand() の関数名を net\_srand() に変更し、引数の型を UW に変更した。
  - [8] tcp\_read\_rwbuf\_cs() のコンパイルエラー（(UB\*)data += blen）の問題を修正した。
  - [9] MAKE\_IPV4\_ADDR の各引数へのキャストを UB から UW に変更した。
  - [10] SYSTIM\_HZ の定義で、定数にキャスト（SYSTIM）を追加した。
  - [11] IS\_PTR\_DEFINED の定義で、(a) にキャスト（VP）を追加した。
  - [12] UB 型変数、UH 型変数への代入での警告の問題を修正した。
  - [13] 32 ビット幅が必要な変数の型には、明示的に W または UW を使用した。
  - [14] syslog の引数の型を明確にした。
  - [15] 警告が出る関数の引数をキャストした。
  - [16] 警告が出る関数の戻り値をキャストした。
  - [17] 精度に関する警告が出る代入の右辺をキャストした。
- (43) マクロ ntohs と htons の定義で、ビッグエンディアンのプロセッサでは、UH の戻り値を返すように変更した。
- (44) マクロ ntohl と htonl の定義で、ビッグエンディアンのプロセッサでは、UW の戻り値を返すように変更した。
- (45) TCP ヘッダのトレース出力機能の組込みを指定するコンパイル時コンフィギュレーションパラメータを追加した。
  - [1] TCP\_CFG\_TRACE、TCP ヘッダのトレース出力機能を組込む。
  - [2] TCP\_CFG\_TRACE\_IPV4\_RADDR、トレース出力対象のリモートホストの IPv4 アドレスを指定する。IPV4\_ADDRANY を指定すると、全てのホストを対象とする。

- [3] TCP\_CFG\_TRACE\_LPORTNO、トレース出力対象のローカルホストのポート番号を指定する。  
TCP\_PORTANY を指定すると、全てのポート番号を対象にする。
- [4] TCP\_CFG\_TRACE\_RPORTNO、トレース出力対象のリモートホストのポート番号を指定する。  
TCP\_PORTANY を指定すると、全てのポート番号を対象にする。
- (46) サポート関数 ip2str で、第 1 引数に NULL を指定すると、TINET 内部で確保してあるバッファに文字列を書き込み、そのアドレスを返すように変更した。これに伴い、バッファ数を指定するコンパイル時コンフィギュレーションパラメータ NUM\_IPV4ADDR\_STR\_BUFF を追加した。
- (47) サポート関数 ipv62str で、第 1 引数に NULL を指定すると、TINET 内部で確保してあるバッファに文字列を書き込み、そのアドレスを返すように変更した。これに伴い、バッファ数を指定するコンパイル時コンフィギュレーションパラメータ NUM\_IPV6ADDR\_STR\_BUFF を追加した。
- (48) サポート関数 mac2str で、第 1 引数に NULL を指定すると、TINET 内部で確保してあるバッファに文字列を書き込み、そのアドレスを返すように変更した。これに伴い、バッファ数を指定するコンパイル時コンフィギュレーションパラメータ NUM\_MACADDR\_STR\_BUFF を追加した。
- (49) tinet と tinet/netdev/if\_ed における Makefile のインクルードファイルを Makefile.tinet に変更した。
- (50) 送信ウィンドバッファの省コピー機能で、TCP\_CFG\_SWBUF\_SAVE を指定し、TCP 通信端点を生成する静的 API においても、送信ウィンドバッファを指定した場合、誤ったデータを送信する問題を修正した。
- (51) tinet/Makefile.tinet における TINET\_KERNEL\_CFG の定義を、tinetcfg.c から、tinetcfg\_kern.c、tinetcfg.c、TINET 内部で使用するカーネルオブジェクトを定義しているコンフィギュレーションファイルに変更した。
- (52) 8 ビット毎に指定した IPv4 アドレスを 32 ビットにするマクロ MAKE\_IPV4\_ADDR を追加した (TOPPERS/TINET バグトラックスシステム ID:4)。
- (53) 各ソースコードのコピーライトを修正した (TOPPERS/TINET バグトラックスシステム ID:6)。

#### 4. リリース 1.3.x から 1.4 までの変更

- (1) tinet コンフィギュレーションに TCP/IPv4 受付口の予約 ID を定義する VRID\_TCP\_REP を追加した。
- (2) tinet コンフィギュレーションに TCP/IPv4 通信端点の予約 ID を定義する VRID\_TCP\_CEP を追加した。
- (3) tinet コンフィギュレーションに UDP/IPv4 通信端点の予約 ID を定義する VRID\_UDP\_CEP を追加した。
- (4) tinet コンフィギュレーションに TCP/IPv6 受付口の予約 ID を定義する VRID\_TCP6\_REP を追加した。
- (5) tinet コンフィギュレーションに TCP/IPv6 通信端点の予約 ID を定義する VRID\_TCP6\_CEP を追加した。
- (6) tinet コンフィギュレーションに UDP/IPv6 通信端点の予約 ID を定義する VRID\_UDP6\_CEP を追加した。
- (7) ITRON TCP/IP API の TCP の拡張機能を有効にする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 を追加した。
- (8) ITRON TCP/IP API の UDP の拡張機能を有効にするコンパイル時コンフィギュレーションパラメータ UDP\_CFG\_EXTENTIONS を追加した。
- (9) 一般定数マクロ TCP\_REP\_NONE を追加した。値は (0)。
- (10) 一般定数マクロ TCP\_CEP\_NONE を追加した。値は (0)。
- (11) 一般定数マクロ UDP\_CEP\_NONE を追加した。値は (0)。
- (12) 全域変数として、ipv6\_addrany をマクロで定義した。実体は、in6\_addr\_unspecified であり、これに伴って、プロトコルスタック内部名のリネームから、in6\_addr\_unspecified を外した。
- (13) UDP の通信端点の myportno に UDP\_PORTANY を指定できない問題を修正した。
- (14) ITRON TCP/IP API の TCP/IPv4 受付口を動的に生成する API の cre\_tcp\_rep を追加した。
- (15) ITRON TCP/IP API の TCP 通信端点を動的に生成する API の cre\_tcp\_cep を追加した。
- (16) ITRON TCP/IP API の UDP/IPv4 通信端点を動的に生成する API の cre\_udp\_cep を追加した。
- (17) ITRON TCP/IP API の TCP 受付口を動的に削除する API の del\_tcp\_rep を追加した。
- (18) ITRON TCP/IP API の TCP 通信端点を動的に削除する API の del\_tcp\_cep を追加した。
- (19) ITRON TCP/IP API の UDP 通信端点を動的に削除する API の del\_udp\_cep を追加した。
- (20) ITRON TCP/IP API の TCP/IPv6 受付口を動的に生成する API の cre\_tcp6\_rep を追加した。
- (21) ITRON TCP/IP API の UDP/IPv6 通信端点を動的に生成する API の cre\_udp6\_cep を追加した。

- (22) ITRON TCP/IP API の TCP 通信端点オプションを設定する API の `tcp_set_opt` を追加した。
- (23) ITRON TCP/IP API の TCP 通信端点オプションを読む API の `tcp_get_opt` を追加した。
- (24) ITRON TCP/IP API の UDP 通信端点オプションを設定する API の `udp_set_opt` を追加した。
- (25) ITRON TCP/IP API の UDP 通信端点オプションを読む API の `udp_get_opt` を追加した。
- (26) 内部的に使用している構造体名を変更した。

```
[1] struct t_tcp4_ccep struct t_tcp4_cep
[2] struct t_tcp6_ccep struct t_tcp6_cep
[3] T_TCP_CCEP T_TCP_IN_CCEP T_TCP_CEP
[4] T_TCP_TW_CCEP T_TCP_TWCEP
[5] T_TCP4_CCEP T_TCP4_CEP
[6] T_TCP4_TW_CCEP T_TCP4_TWCEP
[7] T_TCP6_CCEP T_TCP6_CEP
[8] T_TCP4_TW_CCEP T_TCP4_TWCEP
[9] T_TCP_CREP T_TCP_IN_CREP T_TCP_REP
[10] T_TCP4_CREP T_TCP4_REP
[11] T_TCP6_CREP T_TCP6_REP
[12] T_UDP_CCEP T_UDP_IN_CCEP T_UDP_CEP
[13] T_UDP4_CCEP T_UDP4_CEP
[14] T_UDP6_CCEP T_UDP6_CEP
```

- (27) 内部的に使用している変数・配列名を変更した。

```
[1] tcp_crep tcp_rep
[2] tcp_ccep tcp_cep
[3] tcp_tw_ccep tcp_twcep
[4] udp_ccep udp_cep
[5] tmax_tcp_crepid tmax_tcp_repid
[6] tmax_tcp_ccepid tmax_tcp_cepid
[7] tmax_udp_ccepid tmax_udp_cepid
```

- (28) 内部的に使用しているマクロ名を変更した。

```
[1] VOID_TCP_CREPID VOID_TCP_REPID
[2] VOID_TCP_CCEPID VOID_TCP_CEPID
[3] VOID_UDP_CCEPID VOID_UDP_CEPID
[4] INDEX_TCP_CREP INDEX_TCP_REP
[5] INDEX_TCP_CCEP INDEX_TCP_CEP
[6] INDEX_UDP_CCEP INDEX_UDP_CEP
[7] GET_TCP_CREP GET_TCP_REP
[8] GET_TCP_CCEP GET_TCP_CEP
[9] GET_UDP_CCEP GET_UDP_CEP
[10] GET_TCP_CREPID GET_TCP_REPID
[11] GET_TCP_CCEPID GET_TCP_CEPID
```

```

[12] GET_UDP_CCEPID GET_UDP_CEPID
[13] GET_TCP_TW_CCEPID GET_TCP_TWCEPID
[14] TMIN_TCP_RCEPID TMIN_TCP_REPID
[15] TMIN_TCP_CCEPID TMIN_TCP_CEPID
[16] TMIN_UDP_CCEPID TMIN_UDP_CEPID
[17] TMIN_TCP_TW_CCEPID TMIN_TCP_TWCEPID
[18] TNUM_TCP_RCEPID TNUM_TCP_REPID
[19] TNUM_TCP_CCEPID TNUM_TCP_CEPID
[20] TNUM_UDP_CCEPID TNUM_UDP_CEPID

```

(29) 内部的に使用しているカーネルオブジェクト名を変更した。

```

[1] SEM_TCP_CCEP SEM_TCP_CEP
[2] SEM_UDP_CCEP SEM_UDP_CEP

```

(30) TW 用 TCP 通信端点で、不要なメンバを削除した。

(31) 緊急データの最後のバイトのオフセットを規定するコンパイル時コンフィギュレーションパラメータ `TCP_CFG_URG_OFFSET` を追加した。

(32) ITRON TCP/IP API の緊急データ受信 API の `tcp_rcv_oob` を追加した。

(33) ITRON TCP/IP API の緊急データ送信 API の `tcp_snd_oob` を追加した。

(34) UDP のコールバック用受信データキューを使用しないようにした。

(35) ノンブロッキングコールを組込む場合にリンクされる関数の定義を、`udp_usrreq.c` から分離し、`udp_usrreq_nblk.c` に移動した。`udp_usrreq_nblk.c` は、`udp_usrreq.c` にインクルードする。

(36) TCP で、再送を 12 回行なっても、通信が回復しないとき、コネクションを破棄するリセットセグメントが送信されない問題を修正した。

(37) TCP で、ACK を正常に送信しないことがある問題を修正した。

(38) TCP で、高速再転送と高速リカバリが正常に行なわれないことがある問題を修正した。

(39) 受信ウィンドバッファの省コピー機能を使用したとき、TCP のハーフクローズ状態で、データを全て受信できないことがある問題を修正した。

(40) ビッグエンディアンのプロセッサにおいて、TCP ヘッダのトレース出力で、正しい値を表示しない問題を修正した。

(41) TCP ヘッダのトレース出力で、送受信が混ざる場合がある問題を修正した。

(42) コールバック用 UDP 受信キューサイズを指定する `NUM_DTQ_UDP_CB_RCVQ` は、アルゴリズムの変更により不要になった。

(43) TCP 受付口が `tinnet_${UNAME}.cfg` に指定されていない場合、`TCP_CFG_PASSIVE_OPEN` を未定義にしなければ、リンク時に未定義エラーになる問題を修正した。



- (44) T\_NET\_BUF\_IF\_PDU で、フレームの終わりの境界の調整量を指定するイーサネットインタフェース依存パラメータ定義 IF\_PDU\_HDR\_PADDING を追加した。
- (45) 内部的に使用している構造体名を変更した。
- ```
[1] t_net_buf_reasm    t_net_buf4_reasm
[2] T_NET_BUF_REASSM  T_NET_BUF4_REASSM
```
- (46) 再構成バッファサイズのネットワークバッファ数を指定するコンパイル時コンフィギュレーションパラメータ NUM_MPF_NET_BUF_REASSM を NUM_MPF_NET_BUF4_REASSM に変更し、IPv4 用の再構成バッファサイズのネットワークバッファ数を指定するパラメータとした。
- (47) ICMP6 エラーの出力時に、IPv6 MMTU (1,280 オクテット) を超えるデータグラムを出力する問題を修正した。
- (48) IPv6 データグラム再構成用ネットワークバッファ獲得タイムアウトを指定するコンパイル時コンフィギュレーションパラメータ TMO_IP6_FRAG_GET_NET_BUF を追加した。
- (49) IPv6 データグラムの分割・再構成を行う場合に指定する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 を追加した。
- (50) IPv6 データグラム再構成キューサイズを指定するコンパイル時コンフィギュレーションパラメータ NUM_IP6_FRAG_QUEUE を追加した。標準値は 2 である。
- (51) IPv6 データグラム再構成用ネットワークバッファのサイズを指定するコンパイル時コンフィギュレーションパラメータ IP6_CFG_FRAG_REASSM_SIZE を追加した。標準値は 4096 である。
- (52) ICMPv6 で、受信したチェックサムの計算を誤ることがある問題を修正した。
- (53) IPv6 で、認識できない次ヘッダコードを入力したとき、送信する ICMPv6 エラーメッセージに誤りがある問題を修正した。
- (54) IPv6 の Traffic Class と Flow Label をサポートしないが、これらに 0 以外の値を設定して送信することがある問題を修正した。
- (55) IPv6 で、ホップリミットが 0 の ICMPv6 エコー要求パケットを受信したとき、送信する ICMPv6 エコーメッセージに誤りがある問題を修正した。
- (56) IPv6 で、重複アドレス検出時に、誤ったあて先アドレスのデータグラムを受信したとき、誤って自アドレスが重複していると認識してしまうことがある問題を修正した。
- (57) IPv6 で、生存期間が切れたプレフィックスのアドレスが有効になる事がある問題を修正した。
- (58) IPv6 で、近隣探索キャッシュの状態遷移に一部誤りがあった問題を修正した。
- (59) IPv6 で、近隣探索要請マルチキャストデータグラムの送信元アドレスとして誤ったアドレスを設定する事がある問題を修正した。
- (60) IPv6 で、ルータ要請の再送時間間隔が、誤っていた問題 (3 秒 4 秒) を修正した。
- (61) セグメントの異なるホストに送信するとき、デフォルトルータの検索に失敗することがある問題を修正した。

- (62) 認識できない近隣探索オプションの処理に関する問題を修正した。
- (63) IPv6 で、ルータから通知されたホップリミットを正しく処理していない問題を修正した。
- (64) IPv6 で、重複アドレス検出中に E_QOVR エラーが発生することがある問題を修正した。
- (65) IPv6 の近隣通知で、誤ったフラグを設定していた問題を修正した。
- (66) IPv6 で、次ヘッダが 0 の IPv6 データグラムを受信したとき、送信する ICMPv6 エコーメッセージに誤りがある問題を修正した。
- (67) IPv6 で、近隣通知のルータフラグを正しく処理しない問題を修正した。
- (68) IPv6 で、近隣要請の再送時間間隔が、標準より長くなることがある問題を修正した。
- (69) プレフィックスリストのエントリ数を指定するコンパイル時コンフィギュレーションパラメータ NUM_ND6_PREFIX_ENTRY を追加した。
- (70) インタフェースのアドレスリスト (IPv6) のエントリ数を指定するコンパイル時コンフィギュレーションパラメータ NUM_IN6_IFADDR_ENTRY を追加した。
- (71) 次ヘッダ無しの IP データグラムを受信したとき、誤った ICMPv6 エラーメッセージを出力していた問題を修正した。
- (72) デフォルトルータの有効時間が切れた時に、別のデフォルトルータに接続できない問題を修正した。
- (73) IPv6 で、ルータから通知された到達可能状態の有効時間を正しく処理していない問題を修正した。
- (74) IPv6 で、ルータから通知されたルータ有効時間を正しく処理していない問題を修正した。
- (75) マルチキャスト宛の ICMPv6 エコーリクエストへの応答で、送信元アドレスを正しく設定していない問題を修正した。
- (76) Path MTU とホスト情報キャッシュを実装した。
- (77) IPv6 用ホストキャッシュのエントリ数を指定する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 を追加した。
- (78) ネットワークインタフェースに PPP を選択したとき、コンパイルエラーになる問題を修正した。
- (79) リトルエンディアンのプロセッサで、ルータ通知を正しく処理しない問題を修正した。
- (80) リトルエンディアンのプロセッサで、断片ヘッダを正しく処理しない問題を修正した。
- (81) ESP 用として、ネットワークバッファのフラグに NB_FLG_DECRYPTED を追加した。ただし、TINET は、IPSec を直接サポートしない。
- (82) if_ed.c で、IF_ETHER_NIC_HDR_ALIGN 分のアドレスを調節するように変更した。
- (83) if_ed.c の NIC 割り込みハンドラ if_ed_handler で、ターゲット依存部の割り込みクリア関数 ed_inter_clear を呼出すように変更した。

- (84) config/h8 に、`tinet_cpu_config.c` を新設し、ターゲット依存部の割込みクリア関数 `ed_inter_clear` を追加した。ただし、H8 では、明示的にプロセッサの割り込みをクリアする必要はないため、何もしない関数になっている。
- (85) config/h8 の `tinet_cpu_config.h` に、ターゲット依存部の割込みクリア関数 `ed_inter_clear` のプロトタイプ宣言を追加した。
- (86) config/h8 に、`tinet_cpu_config.c` を新設したことに伴い、`Makefile.tinet` も追加した。
- (87) IPv6 で、ルータから通知された到達可能状態の有効時間 (Reachable Time) の再計算は、仕様上は、0.5 ~ 1.5 倍の範囲でランダムな値に設定するが、タイマーの起動間隔を考慮して、0.5 ~ 1.45 の範囲になるよう調整した。
- (88) イーサネットデバイスドライバにより、ネットワークバッファで、アライン調整を行う `IF_ETHER_NIC_NET_BUF_ALIGN` を追加した。
- (89) IPv6 で、ルータから通知されたプレフィックスの有効時間が過ぎた直後に、有効なアドレスとして処理することがある問題を修正した。
- (90) リトルエンディアンのプロセッサのチェックサム計算の処理効率を改善した。

5. リリース 1.4.x から 1.5 までの変更

- (1) データグラム再構成用ネットワークバッファの定義に、`IF_ETHER_NIC_NET_BUF_ALIGN` が不足していた問題を修正した。
- (2) TINET プロトコルスタックの記述を、TOPPERS/ASP カーネルで規定されている TOPPERS 共通データ型に変更した。
- (3) UDP で、ノンブロッキングコール機能を組み込まず、UDP 通信端点に `UDP_PORTANY` を指定したとき、ポート番号が自動的に割り当てられない問題を修正した。
- (4) コンパイル時コンフィギュレーションパラメータ `SUPPORT_IPSEC` を新設した。これは、IPSEC を処理するための指定である。ただし、TINET は、IPSEC を処理するためのフックのみを `ip_input.c` と `ip_output.c` 用意しているだけで、IPSEC 自体は実装していないので、別に組み込む必要がある。
- (5) TOPPERS/ASP に対応した。
- (6) ネットワークインタフェースの PPP とループバックを参考実装とした。
- (7) IPv4/IPv4 データグラムの設定の、4 オクテット境界までパディングで埋める処理で、ネットワークバッファの領域外にパディングを書き込むことがある問題を修正した。
- (8) PPP で、サーバからの IP アドレス自動取得ができない問題を修正した。
- (9) IPv4 の入力処理で、上位層が ICMP 以外の場合に行うオプションの消去をヘッダチェックサムの検査の後に変更した。
- (10) IPv4 の入力処理で、分割のチェックから RF ビットを外した。
- (11) IPv6 で、通信帯域を食いつぶす攻撃への対応のため、タイプ 0 経路制御ヘッダの処理を行わないように変更した。
- (12) TCP の出力で、送信サイズを通信相手の最大セグメントサイズに調整する場合、誤ったサイズに調整することがある問題を修正した。
- (13) 通信相手の TCP の MSS オプションの処理で、記録する値を、`TCP_MINMSS` (216) から `MAX_TCP_SND_SEG` (標準は、IPv4 が 512、IPv6 が 1024) に制限するように変更した。
- (14) IPv6 で、リンクローカルアドレスの重複を検出した場合、それ以降の近隣探索パケットを出力しなように変更した。
- (15) IPv6 で、ルータ通知のプレフィックス長が不正でも処理を継続し、プレフィックスリストに登録する。また、オンリンク・フラグが有効であれば、同一 LAN 内のアドレスとして有効となるように変更した。
- (16) ITRON TCP/IP API の仕様に定義されているコールバック関数の引数 `p_parblk` に関して、アドレス渡しがいしいが、値渡しとしていた実装上の誤りを修正した。また、TINET リリース 1.4 以前と互換性を維持するために値渡しにするコンパイル時コンフィギュレーションパラメータ `TCP_CFG_NON_BLOCKING_COMPAT14` と `UDP_CFG_NON_BLOCKING_COMPAT14` を新設した。
- (17) 能動オープン (`TCP_CON_CEP` 呼出し) でノンブロッキングコールを行った時に、同時オープン

(TCP の状態遷移が「SYN 送信」「SYN 受信」「コネクション確立」)が発生した場合、コールバック関数が呼出されない問題を修正した。

- (18) タスクからの Time Wait 状態の TCP 通信単点分離機能を組込んだ場合に、ACK セグメントを再送する時、ポート番号のバイトオーダーの変換が誤っていた問題を修正した。
- (19) TCP で Keep Alive を有効にした時、ウィンドサイズが 0 のセグメントが送信される問題を修正した。
- (20) Makefile に NO_USE_TINET_LIBRARY = true を指定して、ライブラリ化させない方法における問題を修正した。
- (21) TOPPERS/ASP の実行環境、GCC 3.x 系の開発環境でアプリケーションプログラムを構築した時、ethernet ヘッダ構造体のサイズが不一致になり、動作が不安定になることがある問題を修正した。
- (22) H8 依存部で、tinet_sys_config.h に指定されている ED_IPM の設定を、IPM_LEVEL1 から IPM_LEVEL0 に変更した。