

TINET 変更メモ [2006/5/22]

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 tinet_config.h、cpu_defs.h から tinet_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 の組込み、SUPPORT_TCP の指定。
 - [2] UDP の組込み、SUPPORT_UDP の指定。
- (4) サンプル nserv.c から WWW サーバ機能を tinet/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/tinet.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 tinet/tinet.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) tinet/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] tinet/include/net tinet/net。
 - [2] tinet/include/netdev tinet/netdev。
 - [3] tinet/include/netinet tinet/netinet。
 - [4] tinet/include/netapp tinet/netapp。
 - [5] tinet/include/tinet_config.h tinet/tinet_config.h。
 - [6] tinet/include/tinet_defs.h tinet/tinet_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] tinet/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 tinet_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] tinet/tinet_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、tinnet/net/if.c、tinnet/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、tinnet/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) `tinnet/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
 */
#ifdef CONSOLE_PORTID
#define CONSOLE_PORTID LOGTASK_PORTID
#endif /* of #ifdef 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) `tinnet_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 `ntohl()` を使うべきところで関数 `ntohs()` 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 へのデータの書き込み終了後、不要のため書き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 から 1.1.1 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、`tinet/cfg/tinet_parser.cpp` で、`tinet_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、`tinet/netinet/tcp.cfg` から

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

を削除し、`tinet/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、`_tinnet_tcp_notify` の参照エラーが発生する問題は、`tinet/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) `tinet/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。

3. リリース 1.1.1 から 1.2 までの変更

- (1) IPv6 に対応した。
- (2) 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。
- (3) 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。
- (4) 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。
- (5) ネットワーク統計情報 (`net_count`) の各変数を `_tinnet_` でリネームしないように変更した。
- (6) 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。
- (7) 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。
- (8) PPP の HDLC 入力における問題を修正した。

- (9) 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。

4. リリース 1.2 からリリース 1.2.1 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)。

5. リリース 1.2.1 からリリース 1.2.2 までの変更

- (1) 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。
- (2) 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。
- (3) 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。
- (4) 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。
- (5) 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。
- (6) 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。
- (7) 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)。

6. リリース 1.2.2 からリリース 1.2.3 までの変更

- (1) 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。
- (2) 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。

7. リリース 1.2.3 からリリース 1.2.4 までの変更

- (1) 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、`tinet/netdev/if_ed/if_ed.c` も変更した。
- [4] `ed_ena_inter`、`ed_dis_inter` の定義を、`tinet/netdev/if_ed/if_edreg.h` から、`config/h8/akih8_3069f/tinet_sys_config.c` に変更した。

- (2) `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。
- (3) 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。
- (4) 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。
- (5) 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。

8. リリース 1.2.4 からリリース 1.2.5 までの変更

- (1) 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] `tinet/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。
- (2) 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。

9. リリース 1.2.5 からリリース 1.2.6 までの変更

- (1) 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う問題を修正した (TOPPERS/TINET バグトラックスシステム ID:8)。

10. リリース 1.2.x から 1.3 までの変更

- (1) 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、`tinet_config.h` で、リリース 1.2 との互換性を確保してある。

- (2) 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。
- (3) 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。
- (4) 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)。

- (5) 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、`tinnet/tinnet_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
```

- (6) 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。
- (7) 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。
- (8) 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。
- (9) tinnet コンフィギュレータが生成する `tinnet_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。
- (10) tinnet コンフィギュレータが生成する `tinnet_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。
- (11) tinnet コンフィギュレータが生成する `tinnet_cfg.c` に

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

を出力するように変更した。

- (12) 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。
- (13) 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。
- (14) 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

- とも破棄するため、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。
- (15) 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。
- (16) netinet/in.h に、TINET のバージョン情報 TINET_PRVER を定義した。
- (17) 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。
- (18) TCP と UDP の一部をライブラリ化した。
- (19) 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。
- (20) 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。

- (21) 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。
- (22) 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。
- (23) 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。
- (24) 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。
- (25) 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。
- (26) 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。
- (27) `tinnet` と `tinnet/netdev/if_ed` における `Makefile` のインクルードファイルを `Makefile.tinnet` に変更した。
- (28) 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。
- (29) `tinnet/Makefile.tinnet` における `TINET_KERNEL_CFG` の定義を、`tinnet_cfg.c` から、`tinnet_kern.cfg`、`tinnet.cfg`、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。
- (30) 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)。
- (31) 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)。

11. リリース 1.3から 1.3.1までの変更

- (1) コネクションを切断する API の `tcp_cls_cep` で例外が発生する問題を修正した。
- (2) `int` のサイズが 16 ビットのプロセッサで、TCP のコネクション切断の時に、65,535 オクテットのセグメントが送信される問題を修正した。
- (3) TCP のオプションで、誤ったサイズが指定されても正常に動作するように修正した。
- (4) `tinnet` コンフィギュレータが生成する `tinnet_cfg.c` の `UDP_CRE_CEP` に対応する構造体 `T_UDP_CCEP` のフィールド `callback` の値の前に (FP) を出力するように変更した。
- (5) `tinnet` コンフィギュレータが生成する `tinnet_cfg.c` の `UDP6_CRE_CEP` に対応する構造体 `T_UDP6_CCEP` のフィールド `callback` の値の前に (FP) を出力するように変更した。
- (6) H8C で、文字列関係のライブラリ関数を使用しているにもかかわらず、`#include <string.h>` を定義していない場合発生する問題に対応した。
- (7) 「タスクからの Time Wait 状態の TCP 通信端点分離機能」を有効にしても、コネクションの同時切断のタイミングによっては、分離されない場合があることを、TINET ユーザズマニュアルに明記した。
- (8) リトルエンディアンのプロセッサで、IPv4 の分割・再構成が正しく行われない問題を修正した。
- (9) IP ヘッダの Identification フィールドの値を、データグラムを送信する毎にインクリメントするように変更した。
- (10) 「タスクからの Time Wait 状態の TCP 通信端点分離機能」を有効した場合、TCP 通信端点が Time Wait の時、FIN フラグが付いていないセグメントに対する応答の問題を修正した。
- (11) 「タスクからの Time Wait 状態の TCP 通信端点分離機能」を有効した場合、TCP 通信端点が Time Wait の時、RST フラグが付いたセグメントに対する応答の問題を修正した。
- (12) 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。
- (13) 「タスクからの Time Wait 状態の TCP 通信端点分離機能」を有効した場合、TCP 通信端点が Time Wait に移行するとき、誤ってリセットセグメントを送信する場合がある問題を修正した。
- (14) UDP で、ノンブロッキングコールを組み込み、`udp_snd_dat` を呼び出した時、ノンブロッキングコールの引数 `p_parblk` に、送信したデータ数が設定されない問題を修正した。
- (15) UDP の通信端点の `myportno` に `UDP_PORTANY` を指定できないことを、TINET ユーザズマニュアルに明記した。
- (16) コールバックの UDP パケットの受信で、事象の種類が誤っていた問題を修正した。
- (17) `int` のサイズが 16 ビットの処理系で、TCP の初期シーケンス番号の更新間隔が規定より短くなる事がある問題を修正した。