

```

/*
 * auth.c - PPP authentication and phase control.
 *
 * Copyright (c) 1993 The Australian National University.
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms are permitted
 * provided that the above copyright notice and this paragraph are
 * duplicated in all such forms and that any documentation,
 * advertising materials, and other materials related to such
 * distribution and use acknowledge that the software was developed
 * by the Australian National University. The name of the University
 * may not be used to endorse or promote products derived from this
 * software without specific prior written permission.
 * THIS SOFTWARE IS PROVIDED ``AS IS'' AND WITHOUT ANY EXPRESS OR
 * IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED
 * WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE.
 *
 * Copyright (c) 1989 Carnegie Mellon University.
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms are permitted
 * provided that the above copyright notice and this paragraph are
 * duplicated in all such forms and that any documentation,
 * advertising materials, and other materials related to such
 * distribution and use acknowledge that the software was developed
 * by Carnegie Mellon University. The name of the
 * University may not be used to endorse or promote products derived
 * from this software without specific prior written permission.
 * THIS SOFTWARE IS PROVIDED ``AS IS'' AND WITHOUT ANY EXPRESS OR
 * IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED
 * WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE.
 */

#define RCSID      "$Id: auth.c,v 1.69 2001/03/12 22:50:01 paulus Exp
$"

#include <stdio.h>
#include <stddef.h>
#include <stdlib.h>
#include <unistd.h>
#include <pwd.h>
#include <grp.h>
#include <string.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <sys/socket.h>
#include <utmp.h>
#include <fcntl.h>
#if defined(_PATH_LASTLOG) && defined(_linux_)
#include <lastlog.h>
#endif

#include <netdb.h>
#include <netinet/in.h>
#include <arpa/inet.h>

#ifdef USE_PAM
#include <security/pam_appl.h>
#endif

#ifdef HAS_SHADOW
#include <shadow.h>
#endif
#ifndef PW_PPP

```

```

/*
 * ccp.c - PPP Compression Control Protocol.
 *
 * Copyright (c) 1994 The Australian National University.
 * All rights reserved.
 *
 * Permission to use, copy, modify, and distribute this software and
its
 * documentation is hereby granted, provided that the above copyright
 * notice appears in all copies. This software is provided without
any
 * warranty, express or implied. The Australian National University
 * makes no representations about the suitability of this software
for
 * any purpose.
 *
 * IN NO EVENT SHALL THE AUSTRALIAN NATIONAL UNIVERSITY BE LIABLE TO
ANY
 * PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL
DAMAGES
 * ARISING OUT OF THE USE OF THIS SOFTWARE AND ITS DOCUMENTATION,
EVEN IF
 * THE AUSTRALIAN NATIONAL UNIVERSITY HAVE BEEN ADVISED OF THE
POSSIBILITY
 * OF SUCH DAMAGE.
 *
 * THE AUSTRALIAN NATIONAL UNIVERSITY SPECIFICALLY DISCLAIMS ANY
WARRANTIES,
 * INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY
 * AND FITNESS FOR A PARTICULAR PURPOSE. THE SOFTWARE PROVIDED
HEREUNDER IS
 * ON AN "AS IS" BASIS, AND THE AUSTRALIAN NATIONAL UNIVERSITY HAS NO
 * OBLIGATION TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS,
 * OR MODIFICATIONS.
 */

#define RCSID      "$Id: ccp.c,v 1.32 2001/03/08 05:11:11 paulus Exp
$"

#include <stdlib.h>
#include <string.h>

#include "pppd.h"
#include "fsm.h"
#include "ccp.h"
#include <net/ppp-comp.h>

static const char rcsid[] = RCSID;

/*
 * Unfortunately there is a bug in zlib which means that using a
 * size of 8 (window size = 256) for Deflate compression will cause
 * buffer overruns and kernel crashes in the deflate module.
 * Until this is fixed we only accept sizes in the range 9 .. 15.
 * Thanks to James Carlson for pointing this out.
 */
#define DEFLATE_MIN_WORKS      9

/*
 * Command-line options.
 */
static int setbsdcomp __P((char **));
static int setdeflate __P((char **));
static char bsd_value[8];
static char deflate_value[8];

```

```

/*
 * chap.c - Challenge Handshake Authentication Protocol.
 *
 * Copyright (c) 1993 The Australian National University.
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms are permitted
 * provided that the above copyright notice and this paragraph are
 * duplicated in all such forms and that any documentation,
 * advertising materials, and other materials related to such
 * distribution and use acknowledge that the software was developed
 * by the Australian National University. The name of the University
 * may not be used to endorse or promote products derived from this
 * software without specific prior written permission.
 * THIS SOFTWARE IS PROVIDED ``AS IS'' AND WITHOUT ANY EXPRESS OR
 * IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED
 * WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE.
 *
 * Copyright (c) 1991 Gregory M. Christy.
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms are permitted
 * provided that the above copyright notice and this paragraph are
 * duplicated in all such forms and that any documentation,
 * advertising materials, and other materials related to such
 * distribution and use acknowledge that the software was developed
 * by Gregory M. Christy. The name of the author may not be used to
 * endorse or promote products derived from this software without
 * specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED ``AS IS'' AND WITHOUT ANY EXPRESS OR
 * IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED
 * WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE.
 */

#define RCSID      "$Id: chap.c,v 1.25 2001/03/08 05:11:11 paulus Exp
$"

/*
 * TODO:
 */

#include <stdio.h>
#include <string.h>
#include <sys/types.h>
#include <sys/time.h>

#include "pppd.h"
#include "chap.h"
#include "md5.h"
#ifdef CHAPMS
#include "chap_ms.h"
#endif

static const char rcsid[] = RCSID;

/*
 * Command-line options.
 */
static option_t chap_option_list[] = {
    { "chap-restart", o_int, &chap[0].timeouttime,
      "Set timeout for CHAP", OPT_PRIO },
    { "chap-max-challenge", o_int, &chap[0].max_transmits,
      "Set max #xmits for challenge", OPT_PRIO },

```

```

/*
 * fsm.c - {Link, IP} Control Protocol Finite State Machine.
 *
 * Copyright (c) 1989 Carnegie Mellon University.
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms are permitted
 * provided that the above copyright notice and this paragraph are
 * duplicated in all such forms and that any documentation,
 * advertising materials, and other materials related to such
 * distribution and use acknowledge that the software was developed
 * by Carnegie Mellon University. The name of the
 * University may not be used to endorse or promote products derived
 * from this software without specific prior written permission.
 * THIS SOFTWARE IS PROVIDED `AS IS' AND WITHOUT ANY EXPRESS OR
 * IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED
 * WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE.
 */

#define RCSID      "$Id: fsm.c,v 1.17 1999/08/13 06:46:12 paulus Exp
$"

/*
 * TODO:
 * Randomize fsm id on link/init.
 * Deal with variable outgoing MTU.
 */

#include <stdio.h>
#include <string.h>
#include <sys/types.h>

#include "pppd.h"
#include "fsm.h"

static const char rcsid[] = RCSID;

static void fsm_timeout __P((void *));
static void fsm_rconfreq __P((fsm *, int, u_char *, int));
static void fsm_rconfack __P((fsm *, int, u_char *, int));
static void fsm_rconfnakrej __P((fsm *, int, int, u_char *, int));
static void fsm_rtermreq __P((fsm *, int, u_char *, int));
static void fsm_rtermack __P((fsm *));
static void fsm_rcoderej __P((fsm *, u_char *, int));
static void fsm_sconfreq __P((fsm *, int));

#define PROTO_NAME(f) ((f)->callbacks->proto_name)

int peer_mru[NUM_PPP];

/*
 * fsm_init - Initialize fsm.
 *
 * Initialize fsm state.
 */
void
fsm_init(f)
    fsm *f;
{
    f->state = INITIAL;
    f->flags = 0;
    f->id = 0; /* XXX Start with random id? */
    f->timeouttime = DEFTIMEOUT;
    f->maxconfreqtransmits = DEFMAXCONFREQS;
}

```

```

/*
 * ipcp.c - PPP IP Control Protocol.
 *
 * Copyright (c) 1989 Carnegie Mellon University.
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms are permitted
 * provided that the above copyright notice and this paragraph are
 * duplicated in all such forms and that any documentation,
 * advertising materials, and other materials related to such
 * distribution and use acknowledge that the software was developed
 * by Carnegie Mellon University. The name of the
 * University may not be used to endorse or promote products derived
 * from this software without specific prior written permission.
 * THIS SOFTWARE IS PROVIDED ``AS IS'' AND WITHOUT ANY EXPRESS OR
 * IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED
 * WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE.
 */

#define RCSID      "$Id: ipcp.c,v 1.57 2001/03/08 05:11:12 paulus Exp
$"

/*
 * TODO:
 */

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <netdb.h>
#include <sys/param.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>

#include "pppd.h"
#include "fsm.h"
#include "ipcp.h"
#include "pathnames.h"

static const char rcsid[] = RCSID;

/* global vars */
ipcp_options ipcp_wantoptions[NUM_PPP]; /* Options that we want to
request */
ipcp_options ipcp_gotoptions[NUM_PPP]; /* Options that peer ack'd
*/
ipcp_options ipcp_allowoptions[NUM_PPP]; /* Options we allow peer to
request */
ipcp_options ipcp_hisoptions[NUM_PPP]; /* Options that we ack'd */

u_int32_t netmask = 0; /* IP netmask to set on interface */

bool disable_defaultip = 0; /* Don't use hostname for default IP
adrs */

/* Hook for a plugin to know when IP protocol has come up */
void (*ip_up_hook) __P((void)) = NULL;

/* Hook for a plugin to know when IP protocol has come down */
void (*ip_down_hook) __P((void)) = NULL;

/* Hook for a plugin to choose the remote IP address */
void (*ip_choose_hook) __P((u_int32_t *)) = NULL;

```

```

/*
 * lcp.c - PPP Link Control Protocol.
 *
 * Copyright (c) 1989 Carnegie Mellon University.
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms are permitted
 * provided that the above copyright notice and this paragraph are
 * duplicated in all such forms and that any documentation,
 * advertising materials, and other materials related to such
 * distribution and use acknowledge that the software was developed
 * by Carnegie Mellon University. The name of the
 * University may not be used to endorse or promote products derived
 * from this software without specific prior written permission.
 * THIS SOFTWARE IS PROVIDED ``AS IS'' AND WITHOUT ANY EXPRESS OR
 * IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED
 * WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE.
 */

#define RCSID      "$Id: lcp.c,v 1.57 2001/03/08 05:11:14 paulus Exp
$"

/*
 * TODO:
 */

#include <stdio.h>
#include <string.h>
#include <stdlib.h>

#include "pppd.h"
#include "fsm.h"
#include "lcp.h"
#include "chap.h"
#include "magic.h"

static const char rcsid[] = RCSID;

/*
 * When the link comes up we want to be able to wait for a short
while,
 * or until seeing some input from the peer, before starting to send
 * configure-requests. We do this by delaying the fsm_lowerup call.
 */
/* steal a bit in fsm flags word */
#define DELAYED_UP      0x100

static void lcp_delayed_up __P((void *));

/*
 * LCP-related command-line options.
 */
int    lcp_echo_interval = 0; /* Interval between LCP echo-requests
 */
int    lcp_echo_fails = 0;    /* Tolerance to unanswered echo-
requests */
bool   lax_recv = 0;          /* accept control chars in asyncmap */
bool   noendpoint = 0;       /* don't send/accept endpoint
discriminator */

static int noopt __P((char **));

#ifdef HAVE_MULTILINK
static int setendpoint __P((char **));
static void printendpoint __P((option_t *, void (*)(void *, char *,

```

```

/*
 * magic.c - PPP Magic Number routines.
 *
 * Copyright (c) 1989 Carnegie Mellon University.
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms are permitted
 * provided that the above copyright notice and this paragraph are
 * duplicated in all such forms and that any documentation,
 * advertising materials, and other materials related to such
 * distribution and use acknowledge that the software was developed
 * by Carnegie Mellon University. The name of the
 * University may not be used to endorse or promote products derived
 * from this software without specific prior written permission.
 * THIS SOFTWARE IS PROVIDED `AS IS' AND WITHOUT ANY EXPRESS OR
 * IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED
 * WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
 * PURPOSE.
 */

#define RCSID      "$Id: magic.c,v 1.9 1999/08/13 06:46:15 paulus Exp
$"

#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/time.h>

#include "pppd.h"
#include "magic.h"

static const char rcsid[] = RCSID;

extern long mrand48 __P((void));
extern void srand48 __P((long));

/*
 * magic_init - Initialize the magic number generator.
 *
 * Attempts to compute a random number seed which will not repeat.
 * The current method uses the current hostid, current process ID
 * and current time, currently.
 */
void
magic_init()
{
    long seed;
    struct timeval t;

    gettimeofday(&t, NULL);
    seed = get_host_seed() ^ t.tv_sec ^ t.tv_usec ^ getpid();
    srand48(seed);
}

/*
 * magic - Returns the next magic number.
 */
u_int32_t
magic()
{
    return (u_int32_t) mrand48();
}

#ifdef NO_DRAND48
/*
 * Substitute procedures for those systems which don't have

```

```
/*
 * main.c - Point-to-Point Protocol main module
 *
 * Copyright (c) 1989 Carnegie Mellon University.
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms are permitted
 * provided that the above copyright notice and this paragraph are
 * duplicated in all such forms and that any documentation,
 * advertising materials, and other materials related to such
 * distribution and use acknowledge that the software was developed
 * by Carnegie Mellon University. The name of the
 * University may not be used to endorse or promote products derived
 * from this software without specific prior written permission.
 * THIS SOFTWARE IS PROVIDED ``AS IS'' AND WITHOUT ANY EXPRESS OR
 * IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED
 * WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE.
 */
```

```
#define RCSID "$Id: main.c,v 1.105 2001/03/12 22:58:59 paulus Exp $"
```

```
#include <stdio.h>
#include <ctype.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <signal.h>
#include <errno.h>
#include <fcntl.h>
#include <syslog.h>
#include <netdb.h>
#include <utmp.h>
#include <pwd.h>
#include <setjmp.h>
#include <sys/param.h>
#include <sys/types.h>
#include <sys/wait.h>
#include <sys/time.h>
#include <sys/resource.h>
#include <sys/stat.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
```

```
#include "pppd.h"
#include "magic.h"
#include "fsm.h"
#include "lcp.h"
#include "ipcp.h"
#ifdef INET6
#include "ipv6cp.h"
#endif
#include "upap.h"
#include "chap.h"
#include "ccp.h"
#include "pathnames.h"
#include "tdb.h"
```

```
#ifdef CBCP_SUPPORT
#include "cbcp.h"
#endif
```

```
#ifdef IPX_CHANGE
#include "ipxcp.h"
```

/\*

```
*****
**
** md5.c -- the source code for MD5 routines
**
** RSA Data Security, Inc. MD5 Message-Digest Algorithm
**
** Created: 2/17/90 RLR
**
** Revised: 1/91 SRD,AJ,BSK,JT Reference C ver., 7/10 constant corr.
**
```

```
*****
**
** /
```

/\*

```
*****
**
** Copyright (C) 1990, RSA Data Security, Inc. All rights reserved.
**
**
** License to copy and use this software is granted provided that
**
** it is identified as the "RSA Data Security, Inc. MD5 Message-
**
** Digest Algorithm" in all material mentioning or referencing this
**
** software or this function.
**
**
** License is also granted to make and use derivative works
**
** provided that such works are identified as "derived from the RSA
**
** Data Security, Inc. MD5 Message-Digest Algorithm" in all
**
** material mentioning or referencing the derived work.
**
**
** RSA Data Security, Inc. makes no representations concerning
**
** either the merchantability of this software or the suitability
**
** of this software for any particular purpose. It is provided "as
**
** is" without express or implied warranty of any kind.
**
**
** These notices must be retained in any copies of any part of this
**
** documentation and/or software.
**
```

```
*****
**
** /
```

```

/*
 * options.c - handles option processing for PPP.
 *
 * Copyright (c) 1989 Carnegie Mellon University.
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms are permitted
 * provided that the above copyright notice and this paragraph are
 * duplicated in all such forms and that any documentation,
 * advertising materials, and other materials related to such
 * distribution and use acknowledge that the software was developed
 * by Carnegie Mellon University. The name of the
 * University may not be used to endorse or promote products derived
 * from this software without specific prior written permission.
 * THIS SOFTWARE IS PROVIDED ``AS IS'' AND WITHOUT ANY EXPRESS OR
 * IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED
 * WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE.
 */

#define RCSID      "$Id: options.c,v 1.80 2001/03/12 22:56:12 paulus
Exp $"

#include <ctype.h>
#include <stdio.h>
#include <errno.h>
#include <unistd.h>
#include <fcntl.h>
#include <stdlib.h>
#include <syslog.h>
#include <string.h>
#include <pwd.h>
#ifdef PLUGIN
#include <dlfcn.h>
#endif
#ifdef PPP_FILTER
#include <pcap.h>
#include <pcap-int.h> /* XXX: To get struct pcap */
#endif

#include "pppd.h"
#include "pathnames.h"

#if defined(ultrix) || defined(NeXT)
char *strdup __P((char *));
#endif

static const char rcsid[] = RCSID;

struct option_value {
    struct option_value *next;
    const char *source;
    char value[1];
};

/*
 * Option variables and default values.
 */
#ifdef PPP_FILTER
int  dflag = 0; /* Tell libpcap we want debugging */
#endif
int  debug = 0; /* Debug flag */
int  kdebugflag = 0; /* Tell kernel to print debug messages */
/*
int  default_device = 1; /* Using /dev/tty or equivalent */
char devnam[MAXPATHLEN]; /* Device name */

```

```

/*
 * sys-linux.c - System-dependent procedures for setting up
 * PPP interfaces on Linux systems
 *
 * Copyright (c) 1989 Carnegie Mellon University.
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms are permitted
 * provided that the above copyright notice and this paragraph are
 * duplicated in all such forms and that any documentation,
 * advertising materials, and other materials related to such
 * distribution and use acknowledge that the software was developed
 * by Carnegie Mellon University. The name of the
 * University may not be used to endorse or promote products derived
 * from this software without specific prior written permission.
 * THIS SOFTWARE IS PROVIDED ``AS IS'' AND WITHOUT ANY EXPRESS OR
 * IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED
 * WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE.
 */

#include <sys/ioctl.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <sys/time.h>
#include <sys/errno.h>
#include <sys/file.h>
#include <sys/stat.h>
#include <sys/utsname.h>
#include <sys/sysmacros.h>

#include <stdio.h>
#include <stdlib.h>
#include <syslog.h>
#include <string.h>
#include <time.h>
#include <memory.h>
#include <utmp.h>
#include <mntent.h>
#include <signal.h>
#include <fcntl.h>
#include <ctype.h>
#include <termios.h>
#include <unistd.h>

/* This is in netdevice.h. However, this compile will fail miserably
if
you attempt to include netdevice.h because it has so many
references
to __memcpy functions which it should not attempt to do. So, since
I
really don't use it, but it must be defined, define it now. */

#ifndef MAX_ADDR_LEN
#define MAX_ADDR_LEN 7
#endif

#if __GLIBC__ >= 2
#include <asm/types.h> /* glibc 2 conflicts with linux/types.h
 */
#include <net/if.h>
#include <net/if_arp.h>
#include <net/route.h>
#include <netinet/if_ether.h>
#else
#include <linux/types.h>

```

```

/*
 * upap.c - User/Password Authentication Protocol.
 *
 * Copyright (c) 1989 Carnegie Mellon University.
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms are permitted
 * provided that the above copyright notice and this paragraph are
 * duplicated in all such forms and that any documentation,
 * advertising materials, and other materials related to such
 * distribution and use acknowledge that the software was developed
 * by Carnegie Mellon University. The name of the
 * University may not be used to endorse or promote products derived
 * from this software without specific prior written permission.
 * THIS SOFTWARE IS PROVIDED ``AS IS'' AND WITHOUT ANY EXPRESS OR
 * IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED
 * WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE.
 */

#define RCSID      "$Id: upap.c,v 1.24 2001/03/08 05:11:16 paulus Exp
$"

/*
 * TODO:
 */

#include <stdio.h>
#include <string.h>

#include "pppd.h"
#include "upap.h"

static const char rcsid[] = RCSID;

static bool hide_password = 1;

/*
 * Command-line options.
 */
static option_t pap_option_list[] = {
    { "hide-password", o_bool, &hide_password,
      "Don't output passwords to log", OPT_PRIO | 1 },
    { "show-password", o_bool, &hide_password,
      "Show password string in debug log messages", OPT_PRIOSUB | 0
    },

    { "pap-restart", o_int, &upap[0].us_timeouttime,
      "Set retransmit timeout for PAP", OPT_PRIO },
    { "pap-max-authreq", o_int, &upap[0].us_maxtransmits,
      "Set max number of transmissions for auth-reqs", OPT_PRIO },
    { "pap-timeout", o_int, &upap[0].us_reqtimeout,
      "Set time limit for peer PAP authentication", OPT_PRIO },

    { NULL }
};

/*
 * Protocol entry points.
 */
static void upap_init __P((int));
static void upap_lowerup __P((int));
static void upap_lowerdown __P((int));
static void upap_input __P((int, u_char *, int));
static void upap_protrej __P((int));
static int upap_printpkt __P((u_char *, int,

```

```

/*
 * utils.c - various utility functions used in pppd.
 *
 * Copyright (c) 1999 The Australian National University.
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms are permitted
 * provided that the above copyright notice and this paragraph are
 * duplicated in all such forms and that any documentation,
 * advertising materials, and other materials related to such
 * distribution and use acknowledge that the software was developed
 * by the Australian National University. The name of the University
 * may not be used to endorse or promote products derived from this
 * software without specific prior written permission.
 * THIS SOFTWARE IS PROVIDED `AS IS' AND WITHOUT ANY EXPRESS OR
 * IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED
 * WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE.
 */

#define RCSID      "$Id: utils.c,v 1.13 2001/03/16 02:08:13 paulus Exp
$"

#include <stdio.h>
#include <ctype.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <signal.h>
#include <errno.h>
#include <fcntl.h>
#include <syslog.h>
#include <netdb.h>
#include <utmp.h>
#include <pwd.h>
#include <sys/param.h>
#include <sys/types.h>
#include <sys/wait.h>
#include <sys/time.h>
#include <sys/resource.h>
#include <sys/stat.h>
#include <sys/socket.h>
#include <netinet/in.h>
#ifdef SVR4
#include <sys/mkdev.h>
#endif

#include "pppd.h"

static const char rcsid[] = RCSID;

#ifdef SUNOS4
extern char *strerror();
#endif

static void logit __P((int, char *, va_list));
static void log_write __P((int, char *));
static void vslp_printer __P((void *, char *, ...));
static void format_packet __P((u_char *, int, void (*) (void *, char
*, ...),
                               void *));

struct buffer_info {
    char *ptr;
    int len;
};

```