# NOS - TCP/IP

Phil Karn, KA9Q e Gerard Van Der Grinten PA0GRI (Ver. 2.0m)

Versione rielaborata da Pizzichetti Lino IK3NGU

# SISTEMA OPERATIVO DI RETE

Traduzione di Pizzichetti Lino ik3ngu.ampr.org con la collaborazione di Antonio Dimasi iv3ium.ampr.org

Treviso settembre1993
$$versione
Ver. 1.22

***Manuale per l'utente***

***Versione del documento 10***

$$copyright



Phil Karn, KA9Q
7431 Teasdale Ave
San Diego, CA 92122
voce: 610-587-8281
fax: 619-587-1825

$$prefazione

## *Prefazione*

Dopo alcuni anni di attivita’ in ambiente tcp/ip , effettuata soprattutto con il software NOS di KA9Q e PA0GRI , ho sentito l’esigenza di condurre qualche intervento sul codice sorgente. Cio’ e’ stato dettato dal desiderio di operare con un programma piu’ aderente alle esigenze di tutti i giorni , assistito dall’esperienza intanto acquisita.

Come si notera’ nel documento che segue, le aggiunte o le modifiche al programma NOS di PA0GRI sono di lieve entita’ , seppure di grande utilita’. Quindi non risultera’ stravolgimento alcuno ne’ nell’interfaccia uomo-macchina , ne’ nella sostanza.

Il seguente testo sostituisce il manuale operativo in italiano gia' pubblicato e include le novita' introdotte .

Nel complesso , comunque , le caratteristiche descritte nei manuali del NOS non potranno essere tutte presenti nello stesso codice eseguibile per motivi di spazio dettati dai limiti del popolare, ma vecchio sistema operativo DOS.

Tuttavia si tenga presente che non saranno mai necessarie tutte insieme tali caratteristiche presenti nel codice sorgente del NOS . Cio' deriva dal fatto che la telematica moderna va' verso il modello della rete di calcolo distribuita nella quale si attua il decentramento delle risorse verso piu' sistemi. Cio' a dispetto del vecchio modello in cui esisteva un solo elaboratore centrale e tanti terminali piu' o meno remoti.

Inoltre e' importante notare che ogni contesto operativo e' caratterizzato da particolari esigenze di comunicazione. Per esempio in una rete a largo raggio (WAN) saranno tipicamente necessarie tre tipi di configurazione: router o gateway , server di rete, host cliente. Ed infatti e' proprio in questa ottica che verranno generati i codici eseguibili.

Pertanto difficilmente un gateway avra' configurato al suo interno il server POP , o il cliente BOOTP. Oppure difficilmente un host cliente avra' configurato i servizi di accesso IP o la gestione di un link PPP; ed infine pure raro' sara' che un server di rete potra' gestire direttamente dispositivi di comunicazione sincroni tipo SCC.

Dunque, fermamente convinto della strada intrapresa, non mi rimane che divulgare il piu' possibile tutto quanto c'e' da sapere a proposito di NOS TCP/IP . Cio' verra' fatto attraverso questo e altri "manualetti" che di volta verranno aggiornati e pubblicati sperando che possano stimolare il maggior numero di persone ad approfondire il programma NOS e la materia telematica tutta.

Vorrei ricordare che alcune parti del pacchetto NOS KA9Q sono ancora in prova cosi come il clinet/server PPP e BOOTP. Inoltre man mano che si renderanno necessari aggiustamenti ed aggiunte al codice sorgente del NOS si pubblichera' un ulteriore aggiornamento sia software che alla manualistica. Cio' vale anche per il programma di gestione della posta elettronica BMB.

Per il lavoro svolto ringrazio molto le persone a me vicine che in una certa misura hanno contribuito alla realizzazione qui presentata. In particolare ringrazio :

il personale del Gruppo di Lavoro Telematico di Treviso ed in special modo Fabrizio Danovaro che ha diligentemente operato sul codice sorgente . Inoltre Segna Franco, Massimo Casali e Borin Gianpaolo per i consigli e rispettivamentre per gli interventi sui comandi break e sul codice a 8 bit di Telnet. Dimasi Antonio (iv3ium) per i preziosi consigli operativi. Ed infine i vari radioamatori autori italiani e stranieri da cui ho attinto parti di codice e idee.

$$NOS

## 1. Il programma NOS_XXX.EXE

Il file eseguibile NOS_XXX.EXE (abbreviato in NOS) , permette di utilizzare le applicazioni internet (TCP\IP), NET/ROM e AX.25 . Poiche' il NOS possiede un sistema operativo interno multitasking, esso puo' agire simultaneamente da "client", "server" e da "commutatore a pacchetto" per tutti e tre i protocolli.

Cioe', mentre un utilizzatore locale accede a servizi remoti, il sistema puo' anche fornire questi stessi servizi ad altri utilizzatori remoti e nello stesso tempo commutare pacchetti del tipo IP, NET/ROM e AX25 tra altri nodi client e server .

La tastiera ed il monitor vengono usati dall'operatore locale per controllare le funzioni sia a livello "host" che di "gateway" (router), per cio' sono disponibili un gran numero di comandi.

$$avvio

## 1.1 Avvio (Startup)

nos [-b] [-s <#sockets>] [-d </direttorio>] [-v] [<startup file>]

Quando il comando Nos viene eseguito senza opzioni, questo tenta di aprire il file autoexec.nos nel direttorio radice del disco corrente. Se esso esiste lo legge e lo esegue come se il suo contenuto fossero digitato sulla console come comandi. Questo processo e' utile per connettere le interfacce di comunicazione, configurare gli indirizzi di rete ed avviare i vari servizi.

Sono accettate quattro opzioni di comando.

### 1.1.1 -b

L'opzione -b specifica l'uso del BIOS per l'output di console; il valore predefinito e' scrivere direttamente nel buffer del display. Si deve usare questa opzione se si sta utilizzando un pacchetto con gestione a finestra (tipo Windows) e si hanno problemi di output nelle finestre.

### 1.1.2 -s <no of sockets>

L'opzione -s specifica il numero di socket che deve essere allocato all'interno del nos. Questo limita il numero di connessioni di rete che hpossono esistere simultaneamente; il valore predefinito e' 40.

### 1.1.3 -d </direttorio>

L'opzione -d permette all'utente di specificare un direttorio "radice" per la configurazione e l' accodamento dei files; il valore predefinito e' il direttorio radice del sistema.

### 1.1.4 -v

L'opzione -v permette all'utente di visualizzare i comandi eseguiti durante l'avvio del Nos. Permette cioe' la lettura dei comandi (grazie all'echo dell'autoexec.nos) prima che essi vengano eseguiti. Questo e' una funzione utile se il Nos si arresta durante l'inizializzazione.

Dopo tutte le opzioni di comando, si puo' specificare il nome di un file alternativo di avvio. Questo file viene allora aperto e letto al posto dell' autoexec.nos.

## 1.2. VARIABILI di AMBIENTE DOS .

Le seguenti variabili d' ambiente DOS possono essere usate per specificare degli elementi al NOS .

### 1.2.1. TZ

La variabile TZ dovrebbe essere posta al valore della timezone locale. Il valore predefinito e' UTC. Essa viene utilizzata nel campo orario da smtp.

### 1.2.2. MAILER

Il MAILER specifica quale programma deve essere eseguito quando viene digitato il comando mail. Il valore predefinito e' BM.EXE.

#### 1.2.3. COMSPEC

La variabile COMSPEC specifica quale interprete dei comandi sara' usato per l'uscita provvisoria in shell dal Nos.Normalmente all'avvio dell' MS-DOS si utilizza il COMMAND.COM. Il valore predefinito e' COMMAND.COM.

#### 1.2.4. TMP

La variabile TMP e' utilizzata per creare una zona dove vengono posti i files temporanei. Se la variabile TMP non e' specificata i files temporanei verranno creati nel direttorio radice. Per es.: „set TMP=C:\tmp\".

#### 1.2.5. USER

La variabile USER e' utilizzata da ftp e da Rlogin per fissare il nome dell' utente per il demone rlogin sul sistema (*NIX) remoto. Il valore predefinito, se la variabile USER non viene specificata, e' guest . Con ftp il contenuto della variabile USER viene prelevato e proposto come "user name" . Se viene dato solo un "cr" (carriage return) viene utilizzato il nome proposto , altrimenti quello fornito dall'utente.

$$console

## 2. Modi della console

La console puo' trovarsi in uno dei seguenti due modi: modo commandi e modo conversazione . In modo commandi , viene visualizzato il prompt net> e puo' essere inserito uno qualunque dei comandi descritti nel capitolo Commandi. In modo conversazione l'input da tastiera viene eseguito direttamente, secondo la sessione corrente.

Le sessioni sono di molti tipi: Telnet, Ttylink, Rlogin, FTP, AX25, Finger, Command, NETROM, Ping, More, Dial, Dir, PPP PAP, Hopcheck e Tip.

Nelle sessioni Telnet, Ttylink, AX25, NETROM, Rlogin e Tip l'input da tastiera e' inviato al sistema remoto e qualsiasi output del sistema remoto viene visualizzato sulla console.In una sessione FTP, l'input da tastiera e' inizialmente esaminato per vedere se e' un comando locale noto; in questo caso viene eseguito localmente. Se non lo e', esso viene inviato al server remoto FTP. (Si veda il capitolo sui sottocomandi FTP). In una sessione Ping l'utente puo' provare il percorso verso una localita' remota; in una sessione More l'utente puo' esaminare un file locale. Le sessioni Hopcheck possono essere utilizzate per tracciare il cammino di un percorso preso dai pacchetti per raggiungere una destinazione specifica. Una sessione Finger viene usata per conoscere gli utenti di un sistema remoto in quel momento (e quello che stanno facendo , solo su certi sistemi UNIX). Le sessioni PPP PAP sono usate come collegamento fra due sistemi.

La tastiera possiede due stati: cotto e crudo. Nello stato cotto l'input viene dato linea per linea; l'utente puo' usare i caratteri di editor di linea ^U, ^R, ^B, ^W e il backspace rispettivamete per cancellare una linea, rivisualizzare la linea,rivisualizzare il rimanente della precedente linea, cancellare l'ultima parola e cancellare l'ultimo carattere. Premendo il tasto "return" o il "line feed" l'intera linea viene ceduta all'applicazione. In modo crudo , ogni carattere e' passato immediatamente all'applicazione non appena viene digitato. Quando si e' in modo commandi la tastiera e' sempre nello stato cotto. La tastiera si trova nello stato cotto anche quando si sta' lavorando in converse mode in una sessione AX25, FTP o NET/ROM. In una sessione Telnet o

Ttylink lo stato della tastiera cambia in funzione dell' opzione Telnet WILL ECHO che puo' essere caricata dall'utente remoto o meno ( si veda il comando echo).

Sul pc IBM l'utente puo' tornare al modo commandi premendo il tasto "F10" o il tasto definito dal comando "escape". Negli altri sistemi l'utente deve digitare il carattere di escape che come valore predefinito e' control-] (HEX 1d, ASCII GS). (Si noti che questo e' distinto dal carattere ASCII dello stesso nome). Il carattere di escape puo' essere cambiato (si veda il comando escape). Il tasto F10 puo' essere ridefinito con il comando fkey. L' abbinamento , sia di F10 che di escape , con un codice non riproducibile fa si che non si possa uscire dal sistema e l'utente rimanga imprigionato in una sessione.

Nella versione per il pc IBM ogni sessione (incluso la sessione commandi) ha la sua propria pagina video. Quando viene creata una nuova sessione la pagina video della sessione commandi viene salvata in memoria con un conseguente refresh dello schermo. Quando viene premuto il tasto di escape (di solito F10 o ^] ) la schermata della sessione corrente viene salvata e viene ripristinata la pagina video della sessione commandi. Quando si richiama una sessione, la relativa pagina video e' ripristinata esattamente come essa appariva nell'ultimo utilizzo. Il NOS(FP) si aspetta che venga caricato il driver NANSY.SYS per fornire le routines di emulazione dello schermo e del terminale. L' ANSY.SYS del DOS presenta molti problemi e non dovrebbe essere usato. Per avere risultati migliori si deve usare la vesione 24 del NANSI.SYS (nan24hyc.zip presso i bbs). Gli utenti del DeskView non dovrebbero avere problemi con dvansi.sys poiche' e' stato confermato che l'espansione tab e' compatibile con DV.

$$comandi

## 3. Comandi

Questo paragrafo descrive i comandi riconosciuti nel modo commandi , o all'interno di un file di avvio come autoexec.nos. Essi sono espressi con la seguente notazione:

comando
comando parametro-letterale
comando sottocomando <parametro>
comando [<parametro-opzionale>]
comando a | b

Molti comandi richiedono sottocomandi o parametri, che possono essere opzionali o obbligatori. Generalmente se un sottocomando o un parametro richiesto viene omesso, un messaggio di errore riassumera' i sottocomandi disponibili o i parametri richiesti (dando un "?" al posto di un sottocomando si puo' generare tale messaggio). Questo e' utile quando non si conoscono i parametri relativi al comando. Se un comando richiede un valore al parametro, digitando il comando senza il parametro, di solito viene visualizzato il valore corrente della variabile (le eccezzioni a questa regola sono riportate nelle descrizioni relative ai singoli comandi).

Due o piu' parametri separati da barre verticali denotano una scelta tra i valori specificati. I parametri opzionali sono racchiusi tra parentesi quadre, e un parametro racchiuso in <parentesi angolate> deve essere rimpiazzato con un valore reale o con una stringa. Per esempio la notazione <hostid> denota un host o un gateway reale che puo'essere specificato in uno dei due modi seguenti: come un indirizzo numerico IP in notazione punteggiata (es. 44.0.0.1.) o come un nome simbolico elencato nel file domain.txt.

Tutti i comandi ed i sottocomandi possono essere abbreviati. Si devono digitare soltanto i caratteri sufficienti a distingurelo dagli altri comandiche iniziano con la stessa serie di lettere. I parametri invece devono essere digitati completamente. Alcuni comandi FTP (es. put,get,dir, etc.) sono

riconosciuti solo in converse mode con le appropriate sessioni FTP. Essi non sono riconosciuti in modo commandi (si veda il capitolo dei sottocomandi FTP).

## 3.1. <CR>

Inserendo un CR (Enter oppure Invio) mentre si e' in modo commandi si passa in modo conversazione con la sessione corrente. Se non c'e' sessione corrente il Nos rimane in modo commandi e ripristina il prompt net>.

## 3.2. !

E' un'alternativa al comando shell.

## 3.3 #

I comandi che iniziano con il carattere "#" (hash mark) sono ignorati. Questo e' utile principalmente per i commenti nel file autoexec.nos.

$$abort

## 3.4. abort [<session #>]

Interrompe un sottocomando FTP get, put o dir in esecuzione. Se inserito senza un argomento la sessione corrente viene interrotta (questo comando ha effetto solo nella sessione FTP). Quando si sta ricevendo un file, abort azzera semplicemente la connessione dati ; il successivo pacchetto di dati in arrivo generera' un segnale di TCPRST (reset) per azzerare il canale sul server remoto. Quando si sta inviando un file, abort invia un end-of-file anticipato . Si noti che in entrambi i casi abort lasciera' una copia parziale del file sulla macchina di destinazione, che puo' essere rimossa manualmente se non desiderata ( o riutilizzata con l'uso dei comandi resume o rput se implementati).

$$arp

## 3.5. arp

Visualizza la tabella del protocollo di risoluzione degli indirizzi (Address Resolution Protocol) che individua gli indirizzi IP su sottoreti capaci di trasmissione broadcast . Per ogni indirizzo IP viene mostrato il tipo di sottorete (Ethernet, AX.25), l'indirizzo di sottorete ed il tempo di validita'. Se l'indirizzo di collegamento e' al momento sconosciuto, viene mostrato il numero dei datagrammi IP in attesa di risoluzione.

3.5.1. arp add <hostid> ether | ax25 | netrom | arcnet <ether addr> | <ax25 addr>

Aggiunge una voce permanente alla tabella arp (Address Resolution Protocol). Essa non avra' un timeout come accadra' per le voci create automaticamente, ma dovra' essere rimossa con il comando arp drop.

3.5.2. arp drop <hostid> ether | ax25 | netrom | arcnet

Rimuove una voce permanente dalla tabella ARP.

3.5.3. arp flush

Rimuove tutte le voci createsi automaticamente nella tabella ARP; le voci permanenti non vengono toccate.

3.5.4. arp publish <hostid> ether | ax25 | netrom | arcnet <ether addr> | <ax25 addr>

Questo comando e' simile al comando arp add, ma il sistema rispondera' anche a richieste arp qualsiasi che verranno viste sulla rete in cerca dell'indirizzo specificato (usare questo comando con molta attenzione).

$$at_cmd

## 3.6. at yymmddhhmm <command>

at hhmm <command>
at now+hhmm <command>
at k <#>
at [time] "command+"

Il comando "at" e' utile per esecuzioni temporizzate . Senza argomenti mostra la lista degli eventi che stanno per essere eseguiti.

Le possibilita' di impiego sono:

at yymmddhhmm <command>

Esegue <command> alla data specificata, espressa nel formato :

***Anno-Mese-Giorno-Ora-Minnuti***

at hhmm <command>

Esegue <command> all' ora e ai minuti specificati del giorno corrente, nel giorno successivo se il tempo specificato e' passato.

at now+hhmm <command>

Esegue <command> hh ore e mm minuti da adesso. hh e mm possono essere fino a 99.

at k <#>

per cancellare l'evento #,dove # e' il numero mostrato dal comando 'at'.

at [time] "command+"

per indicare recursione. Per es.: dopo che il comando e' stato eseguito, l'esecuzione temporizzata sara' attivata di nuovo.

es.: at now+0030 "mem nibuf 10+" ripristinera' il pool di buffers con le relative statistiche.

Ancora un altro esempio . Si supponga di voler accendere alle 17,00 il server ftp e spegnerlo alle 08,00 di ogni giorno. Sara' allora necessario creare un file, supponiamo di nome "at_cmd" , con all'interno i seguenti comandi "at":

at 0800 "stop ftp"
at 1700 "start ftp"

Infine sra' sufficiente inserire nel file autoexec.nos il comando at usato come segue:

at now+2400 "source at_cmd+"

$$autoroute

## 3.7. autoroute <yes|no>

Mostra o fissa l'opzione di autorouting. Quando essa viena fissata tutti i pacchetti IP AX25 vengono analizzati e registrati.

$$asystat

## 3.8. asystat

Mostra le statistiche relative alle interfacce di comunicazione asincrone collegate (8250 , 16450 o 16550°), se ve ne sono. La visualizzazione per ogni porta consiste in quattro linee. La prima linea da' l' etichetta della porta, e i flags di comunicazione; questi indicano se la porta e' un chip del tipo 16550°, il trigger character, se esiste, se e' abilitato il controllo della linea CTS, se e' abilitato il controllo della linea RLSD (carrier detect) e la velocita' in bit per secondo (la ricezione del trigger character fa si che il driver segnali a livello software superiore che i dati sono pronti; e' posto automaticamente all'appropriato carattere di fine trama per le linee SLIP, PPP e NRS).

La seconda linea visualizzata (MC), mostra lo stato dei pin sulla porta seriale. Essa e' una implementazione software dei visualizzatori a led presenti sul mercato che normalmente si usa interporre tra la porta seriale e il cavo. E' palese l'utilita' di questa riga che consente accelerare le operazioni di verifica.

La terza linea visualizzata, mostra il numero degli eventi del ricevitore (RX): il numero totale delle interruzioni ricevute, dei caratteri ricevuti, gli overrun del ricevitore (caratteri persi) e gli high water mark. L' high water mark e' il massimo numero di caratteri letti dal dispositivo durante un'unica interruzione. Questo e' utile per monitorare gli interrupts di sistema nei margini di ritardo, poiche' mostra quanto la porta hardware e' arrivata vicina all'overflow a causa dell'incapacita' della CPU di rispondere in tempo all'interruzione del ricevitore. I chips 8250 non hanno FIFO, cosi' l'high water mark non puo' assumere valori maggiori di due prima che si abbia un overrun. I chips 16550°, invece, hanno un ricevitore FIFO di 16 bytes che viene programmato via software per interrompere la CPU quando il FIFO e' pieno per un quarto della sua capacita'. L'hight water mark, quando si utilizza un chip 16550°, tipicamente dovrebbe essere pari a 4 o 5; valori piu' elevati indicano che la CPU e' stata, almeno una volta, lenta a rispondere ad una interruzione del ricevitore.

Quando si usa un chip 16550° viene visualizzato un conteggio di timeout FIFO sulle linee dell RX. Questi sono generati automaticamente da 16550° quando tre caratteri di intervallo vanno nel FIFO con piu' di zero caratteri ma meno di quattro. Poiche' i caratteri che costituiscono una trama SLIP o NRS sono normalmente inviati a piena velocita' di linea, questo conteggio sara' di solito un limite inferiore del numero di trame ricevute dalla porta, poiche' solo l'ultimo frammento di una trama generalmente risulta in timeout (e solo quando la trama non e' lunga un multiplo di quattro bytes). In fine vengono mostrati gli overrun del software FIFO e l'high water mark. Questi indicano se il parametro del Bufsize del comando attach necessita di essere modificato (vedere il capitolo relativo all'attach command).

La terza linea mostra i dati di trasmissione (TX) che includono un conteggio del totale delle interruzioni di trasmissione, dei caratteri trasmessi, la lunghezza in bytes della coda di trasmissione, il numero degli stati d'interruzione ed il numero di THRE timeouts. Il numero degli stati di interruzione sara' zero a meno che il controllo della linea CTS o il controllo di linea RLSD siano stati abilitati . Il timeout THRE e' una misura stopgap per prendere gli interrupt persi di trasmissione , che sembrano accadere quando c'e' molta attivita' (in condizioni ideali essi dovrebbero essere uguali a zero).

***Qui di seguito appare un esempio dell' output relativo al comando asystat***

net>
radio1: [trigger 0xc0] [rlsd line control] 19200 bps
MC: int 108394 DTR On  RTS On  CTS On  DSR On  RI Off  CD Off
RX: 73700 int, 73700 chr, 0 hw over, 1 hw hi, 0 sw over, 1635 sw hi
TX: 66784 int, 66784 chr, 0 q, 108394 MS int, 0 THRE TO
net>

$$attach

## 3.9. attach <hw type> ...

Configura e collega un interfaccia hardware al sistema. I dettagli dipendono fortemenete dall'interfaccia e dai flags di configurazione contenuti nel file config.h quando il software e' stato implementato. Puo' accadere che non tutti i drivers elencati qui sotto siano inclusi nella vostra copia di Nos. Le istruzioni dettagliate relative ad ogni driver si trovano nel capitolo Attach Commands. Sono disponibili drivers per i seguenti tipi di hardware:

### 3.9.1. attach 3c500

Non lo si deve usare piu' ! Invece si dovrebbe usare il packet driver. Questo driver e' obsoleto e non verra' piu' supportato.

### 3.9.2. attach asy

E' un interfaccia asincrona standard per PC che utilizzano i chip National 8250 o 16450 o 16550° o equivalenti ad essi compatibili.

### 3.9.3. attach axip

E' un ripetitore digitale (digipeaters) AX25 in modalita' "tunnel internet".

### 3.9.4. attach arcnet

E' un driver Arcnet che funziona tramite un packet driver.

### 3.9.5. attach drsi

E' un driver (N6TTO) per la scheda DRSI PCPA 8530.

### 3.9.6. attach eagle

E' un driver (WA3CVG/NG6Q) per la scheda Eagle Computer (Zilog 8530).

### 3.9.7. attach hapn

E' un driver (KE3Z) per l'adattatore  Hamilton Amateur Packet Network (Intel 8273).

### 3.9.8. attach hs

E' uno speciale driver 8530 ad alta velocita' per il modem (WA4DSY) 56Kb/s.

3.9.9. attach kiss

Esso fa si che si possa utilizzare come secondo canale un tnc “multiplexato “.E’ utilizzato per connetere una seconda porta ad un interfaccia gia’ collegata. Copiera’ la maggior parte dei suoi parametri dalla sua porta primaria (parent).

3.9.10. attach netrom

Essa e’ una pseudo interfaccia che viene utilizzata per abilitare le operazioni NET/ROM (NET/ROM e’ una rete dati amatoriale cosi’ come lo e’ X.25).

3.9.11. attach packet

E’ un driver utilizzato con un software separato (packet drivers) che si adatta alle specifiche del software del FTP, Inc.

3.9.12. attach pc100

E’ un driver per la scheda PACCOMM PC-100 (zilog 8530).

3.9.13. attach pi

E’ una scheda 8530 scc pilotata con tecnica dma (direct memory access) (VE3IFB).

3.9.14. attach scc

E’ un driver (PE1CHL) per le schede 8530 generiche .

3.9.15. attach slfp

E’ un packet driver di tipo Serial Line Faming Protocol.

Tipicamente esso controlla un dispositivo tipo modem in modo esterno al NOS.

A tale scopo e’ necessario avere il file eseguibile “slfp.exe” che
deve essere lanciato con tutti i suoi parametri prima del NOS . Esso e’ molto simile ad un packet driver di tipo Ethernet.

Un modo facile per ottenere un elenco dei parametri richiesti da un dato dispositivo e’ digitare un comando attach parziale (es. attach packet). Questo produce un messaggio che da’ il formato completo del comando.

$$attended

## 3.10. attended [off | on]

Mostra o fissa il segnale “sono presente” nel Nos riferito all’operatore. Questa segnale viene utilizzato nell’intestazione di benvenuto quando ci si connette con il ttylink.

$$ax25

## 3.11. ax25 <subcommand>

Questi comandi vengono utilizzati per le interfacce AX25.

3.11.1. ax25 bc <interface> [on | off]

Il comando bc mostra , abilita o disabilita la trasmissione broadcast attraverso <interface> quando viene dato on | off.

3.11.2. ax25 bcinterval [<seconds>]

Il bcinterval mostra o fissa il tempo, misurato in secondi, fra le trasmissioni bc. Sul video vengono mostrati sia il valore dell'intervallo che del conteggio alla rovescia.

3.11.3. ax25 bckick <interface>

Il comando bckick emette un broadcast direttamente all'interfaccia interface quando e' abilitato dal comando ax25 bc <interval> on.

3.11.4. ax25 bctext ["testo broadcast"]

Il comando bctext mostra o fissa il testo da inviare in un messaggio broadcast emesso ogni bcinterval secondi.

3.11.5. ax25 blimit [<limit>]

Mostra o fissa il limite di backoff delle ritrasmissioni AX25. Normalmente ogni ritrasmissione AX25 successiva e' ritardata di un valore doppio del precedente intervallo; questo e' chiamato binary exponential backoff. Quando il backoff arriva ad essere pari al blimit, si mantiene a quel valore che e' predefinito a 30. Per prevenire la possibilita' di collasso di un canale carico, il blimit deve essere posto ad un valore grande almeno quanto il numero di stazioni che condividono il canale. Si noti che questo e' applicabile su connessioni AX25 reali; gli UI trame non saranno mai ritrasmessi dallo strato AX25 .

3.11.6. ax25 dest

Si veda il comando ax25 hearddest. Esso e' una sua versione abbreviata .

3.11.7. ax25 digipeat [on | off]

Mostra o attivav il flag di abilitazione del digipeater. Se viene usata l'interfaccia axip, il flag DEVE essere attivo (on) altrimenti la funzione digipeat non lavorera' affatto.

3.11.8. ax25 filter <0 | 1 | 2 | 3>

I comandi di filtro abilitano o disabilitano l'accesso alle liste di stazione ascoltate sia sorgente che di destinazione dell'AX25. Questo e' una funzione o bitwise in cui il valore 01 e' per una stazione origine, il valore 02 e' per una stazione di destinazione. Quando il bit si trova nello stato off ilcollegamento e' abilitato, quando e' on e' disabilitato.

3.11.9. ax25 flush

Cancella la lista delle stazioni ascoltate sull'interfaccia AX25 (si veda hax25 heard).

3.11.10 ax25 heard [<interface>]

Visualizza la lista delle stazioni ascoltate sull'interfaccia AX25. Per ogni interfaccia che e' configurata come ax25, viene mostrato un elenco degli indirizzi ax25 ascoltati , assieme ad un

conteggio del numero dei pacchetti ascoltati da ogni stazione e l'intervallo, in ore:minuti:secondi, da quando ogni stazione e' stata ascoltata l'ultima volta. La stazione locale appare per prima nell'elenco; il conteggio dei pacchetti corrisponde al numero dei pacchetti trasmessi. Questa voce e' sempre presente anche se non sono stati inviati pacchetti. Se e' specificata l'interfaccia, viene visualizzato solo l'elenco relativa ad essa. Si noti che il collegamento delle stazioni e' gestito dal comando ax25 filter.

3.11.11 ax25 hearddest [<interface>]

Mostra la lista di destinazione, cioe' la lista delle stazioni a cui stato indirizzato del traffico. Successivamente viene visualizzata l'ultima trasmissione relativa a quella stazione ed il tempo di risposta della stazione stessa. Questo da' un buon riferimento per vedere se la stazione e' raggiungibile ed in grado di rispondere.

3.11.12. ax25 irtt [<milliseconds>]

Mostra o fissa il valore iniziale del tempo di rtt (round trip time) lineare da usare quando viene creata una nuova connessione ax25. Il valore e' espresso in millisecondi. Il valore reale del tempo di andata e di ritorno viene stabilito attraverso una misurazione, una volta che la connessione e' stata realizzata.

3.11.13. ax25 kick <axcd>

Forza una ritrasmissione sullo specifico blocco di controllo dell'ax25. L'indirizzo del blocco di controllo puo' essere trovato con il comando ax25 status.

3.11.14. ax25 maxframe [<count>]

Stabilisce il numero massimo di trame che possono rimanere in attesa di risposta in una sola volta in una nuova connessione ax25. Questo numero non puo' essere maggiore di sette. Senza il parametro count verra' visualizzato il settaggio corrente. Si noti che il conteggio del numero massimo di trame in attesa (outstanding), funziona solo con connessioni virtuali. Le trame UI non sono interessate.

3.11.15. ax25 mycall [<ax25 addr>]

Mostra o fissa l'indirizzo locale predefinito ax25. E' utilizzato il formato standard (es. KA9Q-0 o WB6RQN-5). Questo comando deve essere dato prima di qualsiasi comando attach che utilizzi il modo ax25.

3.11.16. ax25 paclen [<size>]

Limita la grandezza del campo I delle nuove connessioni ax25. Se sono trasmessi datagrammi IP o frammenti piu' grandi di questi essi saranno frammentati a livello ax25, inviati come una serie di trame I , e riassemblati in un datagramma IP completo o in un frammento, all'altra estremita' del collegamento. Per avere dei risultati con i datagrammi IP questo parametro dovrebbe essere minore o uguale al MTU dell'interfaccia associata.

3.11.17. ax25 pthresh [<size>]

Mostra o fissa la soglia poll da usarsi per nuove connessioni ax25 in versione 2. La soglia poll controlla il comportamento della ritrasmissione nel modo seguente. Se la piu' vecchia grandezza delle trame I non confermate e' maggiore o uguale alla soglia allora verranno inviate, a seconda dei casi, trame RR oppure RNR, con il poll bit posizionato se avverra' un timeout. L'idea che sta dietro la soglia di poll e' che il tempo ulteriormenete necessario per inviare un "piccola" trama I

invece di un trama di supervisione, quando avviene un polling dopo un timeout, e' minimo, e poiche' vi e' una buona probabilita' che la trama I debba essere inviata in ogni caso (per es.: se fosse stato perduta precedentemente) allora si puo' anche inviarla come poll. Ma se la trama I e' grande, e' necessario inviare un poll di supervisione (RR/RNR) invece di stabilire se ritrasmettere le vecchie trame I in attesa di risposta; il timeout potrebbe essere stato causato da una perdita di conferma. Questo e' ovviamente un compromesso, cosi' e' utile sperimentare nuovi valori della soglia di poll. Il valore predefinito e' 128 bytes, meta' del valore predefinito di paclen.

3.11.18. ax25 reset <axcb>

Cancella il blocco di controllo dell'ax25 all'indirizzo specificato.

3.11.19. ax25 retry [<count>]

Limita il numero di tentativi successivi di ritrasmissione su nuove connessioni ax25. Se questo limite viene superato, si tentera' di ristabilire il collegamento. Se questo fallisce allora la connessione viene abbandonata e tutti i dati della coda saranno cancellati.

3.11.20. ax25 route

Mostra la tabella delle rotte ax25 che specifica i ripetitori che devono essere utilizzati per raggiungere una data stazione.

3.11.20.1. ax25 route add <target> [digits... ]

Aggiunge un elemento alla tabella delle rotte ax25. Viene eseguito automaticamente il comando ax25 route add se i digipeaters sono specificati nel comando ax25 connect, o se viene ricevuta una connessione di una stazione remota per mezzo dei digipeaters. Tali voci automatiche della tabella delle rotte non si sovrapporranno in alcun modo alle voci create localmente.

3.11.20.2. ax25 route drop <target>

Elimina una voce per la data destinazione dalla tabella delle rotte ax25.

3.11.20.3. ax25 route mode <target> [vc | datagram | interface]

Fissa il modo a vc | datagram | interface per la data destinazione. Il parametro interface e' il valore predefinito per questa interfaccia. Vc e' il circuito virtuale (ax25 connected mode) e datagram e' in modo disconnesso, (trame ax25 UI) .

3.11.21. ax25 status [<axcb>]

Senza argomento, mostra una linea riassuntiva per ogni blocco di controllo ax25. Se e' specificato l'indirizzo di un particolare blocco di controllo vengono elencati in maggior dettaglio i contenuti del blocco stesso. Si noti che le unita' "send queue" sono trame, mentre le unita' « receive queue » sono bytes.

3.11.22. ax25 t3 [<milliseconds>]

Mostra o fissa il timer di inattivita' dell'ax25 ("keep alive"). Il valore viene espresso in millisecondi.

3.11.23. ax25 t4 [<seconds>]

Mostra o fissa il timer AX.25 Link (“redundancy”). Il valore viene espresso in secondi. Quando non e’ avvenuto alcuno scambio durante questo tempo il collegamento viene azzerato e chiuso.

3.11.24. ax25 timertype [l | e | o]

Mostra o fissa il tipo di timer usato per la ritrasmissione e per il recupero: linear, exponential, original.

3.11.25. ax25 version [1 | 2]

Mostra o fissa la versione del protocollo ax25 che si tenta di usare su nuove connessioni. Il valore predefinito e’ 1 (la versione che non usa i bits di poll/final).

3.11.26. ax25 window [<size>]

Fissa il numero di bytes che possono essere pendenti su una coda di ricezione dell’ax25 al di la’ del quale le trame genereranno risposte RNR (reciver not ready). Questo in realta’ e’ vero solamente per sessioni ax25 interattive sospese, poiche’ le trame I in arrivo che contengono pacchetti di rete (IP, NET/ROM) vengono elaborate immediatamente e non vengono inserite nella coda di ricezione. Tuttavia, quando una connessione ax25 trasmette sia pacchetti di rete che pacchetti interattivi sara’ generato un RNR a causa di un accodamento del traffico interattivo il quale impedira’ anche l’invio di pacchetti del livello rete.

$$bbs

## 3.12. bbs

Attiva la porta locale bbs (e’ come una sessione telnet con la propria stazione). I comandi sono:

```
help           ?   <command>
area           a   [<area>]
bye            b
connect      c   <node>
                  c   <interface> <ax25_addr> [[<ax25_addr>] .... ]
download    d   <filename>
                  du <filename> (per files non codificati in modo binario)
                  dx <filename> (per attivare la trasmissione di file con il
```

protocollo XMODEM dal mailbox , se attivo il server tip e se configurato)

```
escape      e   [<escape char.>]
finger         f   [<user>]
help           h   [<command>]
info            i
jheard        j
kill              k   <msg number>
list              l   [<msg number>]
mbusers     m
nodes         n   <node>
operator     o
ports          p
read           r   [<msg number>]
send          s[f] <usename[@hostid] [<from_addr>] [$bul_id]>
send repl   sr  [<msg number>]
telnet         t   <hostid>
upload       u   <filename>
verbose      v   <n>
what          w   [<direttorio>]
expert       x
zap           z   <filename>
sysop       @
```

$$bootp

## 3.13. bootp <subcommand>

Questo e' un server/client bootp , incluso nel Nos. Esso e' stato acquisito dall' Universita' del Michigan in Ann Arbour. E' incluso nelle sorgenti del Nos ma chi scrive non lo ha mai usato ne' testato (PA0GRI).

In generale risulta utile per tutte quelle installazioni in cui il numero di rete IP e' sconosciuto e percio' viene fornito dal server Bootp in cambio del nominativo o comunque del numero fisico di scheda ethernet.

Il suo utilizzo e' piuttosto raro . Comunque , in seguito a discussioni avenute presso il tcp-group@ucsd.edu, l'autore di questa versione di Nos lo ha incluso per un possibile uso .

3.13.1. boot start

Avvia il server bootp .

3.13.2. bootp stop

Arresta il server bootp .

3.13.3. bootp dns [<ipaddr>]

Mostra  o fissa la lista dei nomi dei domain serves per il bootp.

3.13.4. bootp dyip [<iface> | <iface> <ipaddr1> <ipaddr2> | <iface> off]

Mostra il campo degli indirizzi di interfaccia. Il campo e' compreso tra ipaddr1 e ippadr2, entrambi in notazione punteggiata.

3.13.5. boot host [<hostaddr> <hardware_addr> <ip_addr> [boot_file]]

Mostra o attiva un processo boot. Hardware type puo' essere netrom, mac_appletalk o  ax25. Hardware_addr  e' il nome dell'interfaccia.  Ip_addr deve essere espresso in notazione punteggiata.

3.13.6. bootp rmhost <ip_addr>

Cancella ip_addr dalla tabella degli hosts.

3.13.7. bootp homedir [<direttorio> | default]

Mostra o fissa il direttorio dove risiedono i files bootp. Il valore predefinito e' il direttorio bpfiles.

3.13.8. bootp defaultfile [<bootfile> | default]

Mostra o fissa il nome del file di bootp. Il valore predefinito e' boot.

3.13.9. bootp logfile [<filename> | default] [on | off]

Avvia o arresta la richiesta di collegamento di bootp al filename o al nome del file di default bootplog.

3.13.10. bootp logscreen [on | off]

Abilita o disabilita il collegamento del bootp a schermo.

3.13.11. start bootp

Avvia il (demone) server per il bootp.

$$break

## 3.14. - break <interfaccia_asincrona>

Il comando break e' presente sia come sottocomando DIALER
che come comando singolo.

Per l'utilizzo all'interno del file /dialer sara' sufficiente dare il comando su una linea a se stante. Mentre per l'utilizzo esterno al dialer bisognera' usare la sintassi:

break <interfaccia_asincrona>

Per es.: il comando "break linea1" invia a sull'interfaccia di nome linea1 un segnale di break.

$$cd

## 3.15. cd [<dirname>]

Cambia il direttorio di lavoro visualizzandone il nuovo. Se non viene dato l'argomento, il comando cd , visualizza il direttorio corrente. Il comando pwd e' un'alternativa a cd.

$$cdrom

## 3.16. - cdrom [<drive>]

Questo comando e' presente solo se si e' compilato il NOS con l'opzione MSCDEX attivata. Esso permette di definire la lettera del drive CD-ROM. A tale scopo e' stata migliorata la gestione dei pathnames che ora accetta come parametro il nome del drive. Cio' vale anche per i comandi: rename, copy, dir, cd, pwd, more, tail, del, mkdir rmdir. Percio' ora la combinazione drive/direttorio puo' essere una qualsiasi valida sul sistema. Il valore predefinito e' F.

Dando il comando "cdrom" senza parametri, viene visualizzata l'assegnazione corrente.

Si fa presente che per gestire un drive su CD-ROM con il NOS di KA9Q, sara' necessario prima attivare lo stesso in DOS .

Questa caratteristica consente di attivare un server FTP dotato appunto di drive su CD-ROM.

$$close

## 3.17. close [<session>]

Chiude la sessione specificata; senza argomento chiude la sessione corrente. In una sessione AX.25, questo comando inizia una disconessione. In una sessione FTP o Telnet questo comando invia un FIN (per es.: inizia un close) sulla connessione TCP della sessione. Questo e' un metodo

alternativo per iniziare una chiusura di un collegamento sul server remoto (QUIT in FTP, o l'appropriato comando di logout per il sistema remoto nel caso di Telnet). Quando i server FTP e Telnet vedono meta' connessione TCP chiusa, risponde automaticamente chiudendo la restante meta'. Il comando close ha un funzionamento migliore del comando "reset" in quanto non lascia il TCP remoto in uno stato semiaperto.

$$cls

## 3.18. cls

Cancella la schermata della sessione corrente (modo commandi) .

$$comm

## 3.19. comm <interface> <text-string>

Il comando comm invia una <text-string> tramite interface. Questo puo' essere utilizzato per inviare dei comandi per es.: a un tnc oppure ad un modem che si trovi in modo comandi durante l' avvio del NOS. Si noti che per mantenere la spaziatura bisogna includere la stringa tra doppi apici.

$$connect

## 3.20. connect <iface> <ax25 addr> [<digipeater>...]

Inizia una sessione ax25 con l'indirizzo ax25_addr usando l'interfaccia specificata. I dati inviati in questa sessione vengono emessi in pacchetti ax25 convenzionali senza protocollo di livello superiore . Viene usato il formato standard, in cui ogni pacchetto occupa una linea di testo che termina con CR. Per le trasmissioni IP e NET/ROM da terminale a terminale si puo' usare una singola connessione AX.25. I tre tipi di dati vengono automaticamente separati dal loro Identificatore di protocolo a livello 3 AX25 .

Possono essere dati fino a sette digipeaters opzionali; si noti che la parola via non e' necessaria. Se sono specificati i digipeaters, essi vengono automaticamente aggiunti alla tabella di intradamento AX.25 come se fosse stato dato il comando ax25 route add prima dell'inserimento del comando connect.

$$copy

## 3.21 copy <file> <newfile>

Utile per operazioni da remoto.

$$dama

3.21.A dama <sottocomandi>

In questa versione e' stata incorporata per la prima volta la gestione del protocollo DAMA. L'implementazione effettuata e' solo un primo tentativo ancora incompleto che si allontana in vari punti dalla definizione 'ufficiale' del protocollo, vuoi per ragioni contingenti, dipendenti dalla struttura del NOS (ad esempio NON viene mantenuta nessuna polling list; nel NOS ogni connessione e' gestita da un processo separato che gira concorrentemente agli altri), vuoi semplicemente perche' tali dettagli non sono stati ancora implementati. Chi volesse sperimentare questo nuovo metodo di accesso al canale, e' pregato di segnalare allo scrivente eventuali bugs o effetti malefici del non aver rispettato pienamente lo standard.

3.21.A.1 dama active on|off default: off

Sceglie se il NOS deve usare l'AX25 normale (off) oppure l'AX25 modificato secondo il protocollo DAMA (on). Come e' ovvio, i tre comandi seguenti hanno effetto SOLO se dama active e' posto ad on.

Dando questo comando senza parametri, viene mostrato lo stato attuale.

3.21.A.2 dama master on|off default: off

Sceglie se il NOS deve comportarsi come stazione MASTER (se posto on) o SLAVE (se posto off). Si ricorda che tutti gli utenti sono SLAVE e che deve esserci UN SOLO master attivo. Dando questo comando senza parametri, viene mostrato lo stato attuale.

3.21.A.3 dama maxpolltime [<tempo in ms>] default: 60000

Stabilisce il massimo intervallo fra un poll ed il successivo verso una stazione che abbia ripetutamente dichiarato di non avere traffico per il master; ad ogni risposta negativa il tempo tra un poll e l'altro viene aumentato linearmente sino a quando non si giunge al valore qui impostato, dopo di che tale tempo resta costante. In caso la stazione abbia traffico, tale tempo viene riportato al valore iniziale impostato in dama polltime.

Dando questo comando senza parametri viene visualizzato il
valore attuale.

3.21.A.4 dama polltime [<tempo in ms>] default: 5000

Stabilisce il tempo intercorrente fra un poll e l'altro per una stazione che abbia traffico da inviare. Questo e' anche il tempo di poll iniziale per una stazione che si sia appena connessa al MASTER. Tale tempo verra' variato in seguito come sopra illustrato.

Anche qui, dando il comando senza parametri, viene visualizzato il valore attualmente impostato.

$$delete

## 3.22. delete <filename>

Il filename viene rimosso dal file system.

$$detach

## 3.23. detach <iface>

Scollega un interfaccia precedentemente collegata al sistema. Tutte le voci della tabella di instradamento IP vengono cancellate e cosi pure vengono rimossi tutti i riferimenti inoltrati da altre interfacce a questa.

$$dialer1

## 3.24. dialer <iface> [<file> [<seconds> [<pings> [<hostid>]]]]

Genera una sessione di chiamata automatica (autodialer) per l'interfaccia. Ogni qualvolta che l'interfaccia e' in uno stato di inattivita' per un intervallo pari a <seconds>, l'autodialer eseguira' un ping dell' <hostid>. Se non si sono avute risposte dopo un numero di tentativi pari a quello espresso in <pings>, l'autodialer eseguira' i comandi speciali contenuti nel file <dialer-file>.

Se l'intervallo espresso dal parametro <seconds> e' zero, sara' rimosso un precedente processo di comandi dialer . Se il numero di pings e' zero il <dialer-file> sara' eseguito senza il pinging del <hostid>.

Il file deve avere un nome valido qualunque e deve essere collocato nel direttorio principale (si veda il paragrafo Installatione). I comandi nel file sono descritti nel capitolo Suttocomandi Dialer.

Dalla versione 1.17 del NOS l'output del dialer in azione viene rediretto in un file di nome fisso "dialer.log". Cio' per evitare eccessivo sovraccarico della console.

I comandi nel file sono :

3.24.1. control <up | down>
3.24.2. send <string> [<milliseconds>]

Invia string all'interfaccia. Se viene dato il parametro milliseconds, l'intervallo di tempo fra i caratteri e' pari a milliseconds millisecondi .

3.24.3. speed <bps>

Mostra o fissa la velocita' dell'interfaccia corrente a bps baud.

3.24.4. status <up | down>
3.24.5. wait <milliseconds> [<string> [speed]]

Attende per un tempo pari a milliseconds. Se viene dato il parametro string i caratteri che giungono dall'interfaccia vengono confrontati con string. Se le stringhe sono comparabili e speed e' la velocita' della stringa , il prossimo numero letto dall'interfaccia e' il nuovo baudrate usato. Il comando wait potrebbbe essere per esempio "wait 10000 CONNECT speed". In questo caso si attende 10 secondi per la risposta CONNECT dal modem.

$$dir

3.23. dir [<dirname>]

Elenca i contenbuti del direttorio specificato sulla console. Se non si specifica l'argomento viene elencato il contenuto del direttorio corrente. Si noti che questo comando lavora dapprima elencando il direttorio in un file temporaneo, e successivamente creando una sessione more per visualizzarlo. Dopo che questo e' stato completato il file temporaneo viene cancellato.

$$domain

## 3.24. domain <subcommand>

Il comando domain controlla e mostra le operazioni effettuate dal software che mappa gli indirizzi internet in nomi di dominio e viceversa. Generalmente il NOS ha un cliente con un semplice file letto dal server locale. Sara' comunque necessario un vero server , per soddisfare le esigenze di una comunita' per le proprie necessita' di sviluppo.

3.24.1. domain addserver <hostid>

Aggiunge un domain server (DNS) alla lista dei nomi dei server. Si noti, che quando viene dato questo comando , nel file autoexec.nos deve essere dato prima il comando ip address (altrimenti il NOS non sapra' come trovare l'indirizzo, e la risposta non sara' mai riconosciuta, o peggio: il sistema puo' andare in crashes).

3.24.2. domain cache <subcommand>

I seguenti comandi agiscono sulla cache domain . Questi sono dei record di risorsa mantenuti nella memoria (si veda RCF 1033/1034).

3.24.2.1. domain cache clean [<yes | no>]

Mostra o fissa la cancellazione per il termine di validita' dei records risorsa. I records scaduti hanno il valore di timeout pari a '0'. Di solito i records risorsa hanno un valore di timeout predefinito pari a 1800 secondi. Dopo questo periodo essi sono considerati "vecchi" e se riconsiderati dovrebbe essere interrogato di nuovo il domain server. Quando clean non e' specificato (cioe' e' off), i records scaduti saranno mantenuti. Se non si puo' ottenere il loro rimpiazzo da un altro domain server, questi records continueranno ad essere utilizzati.

Quando clean e' attivo, i records scaduti saranno rimossi dal file ogni qualvolta che un nuovo record viene aggiunto al file.

3.24.2.2. domain cache list

Questo comando mostra il contenuto attuale della cache in memoria per i records di risorsa.

3.24.2.3. domain cache size [<size>]

Mostra o fissa la grandezza massima nominale del cache di memoria locale. Il valore predefinito e' 20. (Si noti che la cache potrebbe essere temporaneamente piu' grande quando e' in attesa di nuovi records che devono essere scritti nel file domain.txt) .

3.24.2.4. domain cache wait [<seconds>]

Mostra o fissa l'intervallo in secondi che deve attendere per un'ulteriore attivita' prima di aggiornare il file domain.txt. Il valore predefinito e' 300 secondi (5 minuti).

3.24.3. domain dropserver <hostid>

Rimuove un domain name server dall'elenco degli stessi. Si e' avvertiti quando si cancella l'ultimo server.

3.24.4. domain listserves

Elenca i domain name server attualmente configurati assieme ai relativi dati , il numero di interrogazioni e risposte che sono state scambiate con ognuno e i tempi di risposta, etc.

3.24.5. domain maxwait [<timeout>]

Assegna un valore di timeout (da 1 a 256 secondi) ad una interrogazione o ad un domain name server. Esso non e' asseganto ad un server precedentemente definito ma e' usato per un server con il nome appena definito. Il valore e' usato anche per domain nslookups. Si noti che i name server possono presentare dei problemi nel ricercare dei records in una vasta base di dati. Il valore predefinito il valore di timeout e' posto a 30 secondi.

3.24.6. domain retry [<retries>]

Il valore di retry limita il numero di richieste inviate ai domain name resolvers prima di interrompere e comunicare che host xyzzy.ampr.org non e' presente. Il tempo totale perso durante una consultazione si ottiene moltiplicando il valore di retry per il timeout e per il numero di domain server definiti.

$$server_DNS

3.24.7. - domain startdns

Esso avvia il server di dominio DNS . Il server non supporta:

richieste multiple trasportate su una singola trama, autorita' e resource record addizionali nelle risposte. Esso inviera' sempre risposte non autoritative . Il server DNS del NOS e' stato provato per operare con richieste di tipo A,CNAME,PTR,MX,SOA,HINFO,NS. Se sono stati configurati altri servers remoti , se necessario essi saranno interrogati per risolvere delle richieste.

3.24.8. domain suffix [<domain suffix> | none]>

Mostra o specifica il suffisso del nome di dominio predefinito che deve essere aggiunto al nome dell' host quando non contiene punti. Per esempio, se il suffisso e' ampr.org. e l'utilizzatore digita telnet ka9q, il domain resolver cerchera' di trovare ka9q.ampr.org.. Se l'host name su cui si sta' operando contiene uno o piu' punti, allora il suffisso predefinito non viene aggiunto se l'ultima parte del nome e' minore di 5 caratteri e contiene solo lettere; per esempio telnet foo.bar non sara' trasformata in foo.ka9q.ampr.org.. Si noti che e' richiesto un punto di separazione per il suffisso. Se il suffisso e' la stringa none (senza periodo di separazione) il suffisso corrente viene cancellato e dimenticato.

3.24.9. domain trace [on | off]

Mostra o fissa il flag di controllo, l'elenco delle richieste e delle risposte del server di dominio. I messaggi di tracciamento saranno visibili solo se il nome del dominio che si sta' osservando non e' trovato nella cache del file locale, domain.txt.

3.24.10. domain translate [off | on]

Mostra o fissa il flag che controlla la traduzione degli indirizzi ip dalla notazione punteggiata a nomi simbolici. il processo di traduzione usa pesantemente i lookups del reverse domain name. Questo flag non viene fissato se non si ha una connessione veloce al server dei nomi di dominio o un veloce gestore di dominio e se domain.txt contiene tutti i records IN-ADDR.ARPA. di cui si ha bisogno.

Attenzione : Si consiglia di non usare tale opzione poiche' presenta un malfunzionamento non ancora risolto.

3.24.11. domain verbose [off | on]

Mostra o fissa il flag che controlla il ritorno dell' intero nome (vero) o solo il primo nome (falso). Esso serve solo per la traduzione da indirizzi IP in traduzione del nome.

3.24.12. domain xyzzy

Questo e' (quando digitato completamente) una parola magica per abilitare le richieste di dominio verso server dei nomi esterni nel caso avenga la lettura dei comandi dal file di avvio. Questo dovrebbe essere usato solo da quelli che hanno un accesso affidabile verso un server dei nomi di dominio.

$$drsistat

## 3.25. drsistat

Mostra le statistiche relative a tutte le schede drsi configurate.

$$dump

## 3.26. dump <hex-addres | .> [decimal-range]

Il comando dump mostra il contenuto della memoria in codice ASCII e in codice esadecimale. L'indirizzo esadecimale, per un PC, e' un valore a 32 bit diviso in page address e in page offset. Una colonna di divisione non viene usata ne' accettata. Se non viene specificato il campo decimale vengono mostrati 128 bytes. Il comando dump . mostra la memoria a partire dalla fine di un precedente comando dump.

Esso equivale al comando DOS "debug" : dump. Per es.: per scaricare il contenuto della memoria nel segmento 0000:0400 , dare il comando:

dump 0400

Oppure per scaricare il contenuto nel segmento f000:fff0 dare il comando:

dump f000fff0

$$echo

## 3.27. echo [accept | refuse]

Mostra o fissa il flag che controlla la risposta di un cliente telnet ad una proposta di WILL ECHO.

Il protocollo di presentazione di telnet specifica che in assenza di un accordo , nessuna delle parti comunicanti visualizzera' un echo dei dati ricevuti dall'altro. In questo modo, una sessione di un cliente Telnet visualizza localmente gli input da tastiera, e non invia nulla fino a quando non viene dato un CR. E' anche possibile un editing di linea locale: il backspace cancella l'ultimo carattere digitato, mentre il control U ( ^U) cancella l'intera linea.

Quando si comunica da terminale a terminale utilizzando la tastiera, si usa l'echo standard locale, cosi' l'assegnazione di questi parametri non ha effetto. Invece, molti sistemi in timesharing tipo UNIX eseguono un proprio echo dell'input digitato (questo metodo permette un corretto funzionamento degli editor di schermo). Tali sistemi inviano un Telnet WILL ECHO offer immediatamente dopo aver ricevuto una richiesta di connessione Telnet. Se echo accept e' attivato, una sessione di un cliente Telnet inviera' automaticamente una risposta di DO ECHO. In questo modo l'echo e l'editing locale sono disattivati e ogni carattere introdotto viene inviato immediatamente (soggetto all' algoritmo Nagle tinygram in TCP). Mentre questo "modo" e' adatto all' Ethernet esso e' chiaramente insufficiente e inefficiente quando usato in percorsi lenti come su canali packet radio. Se si specifica echo refuse viene risposto un DONT ECHO ad un WILL ECHO. La sessione del cliente Telnet permane nel "local echo mode". Le sessioni che si trovano gia' in modo echo remoto (remote echo mode) non vengono influenzate (si noti che il Berkley Unix ha un errore di funzionamento per cui da' ugualmente un echo anche quando il client ha rifiutato il WILL ECHO. Per evitare questo problema si deve inserire nella shell il comando stty-echo dopo che ci si e' loggati).

$$eol

## 3.28. eol [unix | standard]

Mostra o fissa il comportamento di fine linea con Telnet, quando si e' nel modo echo remoto. In modo standard ogni carattere viene inviato cosi' com'e'. Nel modo UNIX i CR vengono convertiti in LF . Questo comando non e' necessario con tutti i sistemi UNIX: bisogna usarlo solo quando un

particolare sistema risponde ai line feeds ma non ai CR. Solo SunOS versione 3.2 sembra mostrare questo comportamento; le successive versioni non hanno problemi di funzionamento.

$$escape

## 3.29. escape [<char>]

Mostra o fissa il carattere di escape nel modo comandi corrente, in notazione esadecimale. Sul PC il carattere di escape e' il valore predefinito ^]. Il tasto alternativo di escape e' "F10", a meno che "F10" non sia stato ridefinito con l'uso di Fkey.

$$etherstat

## 3.30. etherstat

Mostra le statistiche relative al driver interno al NOS per l'impiego con la scheda Ethernet 3-Com 3c501 Etherlink I , ormai obsoleta .

Di solito si usano i Packet Drivers , pertanto e da considerarsi di raro utilizzo (se configurato).

$$exit

## 3.31. exit !

Permette di uscire dal programma NOS per ritornare alla shell di MS-DOS. In caso di inattivita' da parte del sistema locale verra' accettato semplicemente il comando exit.

Esso e' condizionato pero' alla non esistenza di sockets aperti, al di fuori dei primi 10 , usualmente attivi.

Lo spirito che ha suggerito tale comando consiste nel fatto che bisognerebbe essere sempre sicuri o comunque a conoscenza che il sistema NOS potrebbe essere ancora impegnato in qualche attivita' di client o di server.

Dunque l'estensione e' cosi' descritta: in seguito al comando exit tutto il traffico IP in corso di instradamento verra' interrotto senza rimedio.

Si riporta un esempio, in cui viene dato il comando "exit" durante una sessione cliente ping:

```
net> exit
Close or reset opened session(s) or use "exit !"
   S#  Type    PCB      Remote socket         Owner
   138 Loc St  81a10008                      81cb0008 ping
   139 Loc St  81c40008                      81cb0008 ping
   140 Raw IP  81c20008                      81cb0008 ping
net>
```

Il Nos risponde specificando che per uscire e' necessario o dare il comando "exit !" , oppure chiudere la sessione ping e percio' vengono mostrati i sockets ancora impegnati .

Si riporta ancora un altro esempio, in cui viene dato il comando "exit" durante l'attivita' del server locale ftp , e quindi senza che l'operatore locale avesse generato alcuna sessione cliente:

```
net> exit
Reset active socket(s) or use "exit !"
   S#  Type    PCB      Remote socket         Owner
   141 TCP     80aa0008 151.90.51.200:1032    809f0008 ftpserv
   142 TCP     85640008 151.90.51.200:1034    809f0008 ftpserv
net>
```

Il Nos risponde specificando che per uscire e' necessario o dare il comando "exit !" , oppure azzerare i sockets in uso con il comando " tcp reset <PCB> " e percio' vengono mostrati i sockets ancora impegnati .

3.31.B exit &

Permette di uscire dal programma NOS per ritornare alla shell di MS-DOS solo dopo che sono stati liberati i sockets prima impegnati. E' evidente che una volta inoltrato tale comando non e' piu' consigliabile dare ulteriori comandi pena la fuoriuscita incodizionata dal NOS .

$$finger

## 3.32. finger <user@hostid> | <@hostid>

Emette una richiesta Finger in rete per l'utente user all'host hostid. Questo crea una sessione cliente che puo' essere interrotta, richiamata, azzerata, etc. proprio come una sessione cliente Telnet. Se viene dato solo @hostid, vengono identificati tutti gli utilizzatori su quell'host.

$$fkey

## 3.33. fkey [<number> [<string>]]

Fkey mostra o fissa i valori dei tasti programmabili della tastiera del PC. Fkey da solo da' un elenco di tutti i tasti rimappabili ed il loro numero. Il comando fkey number mostra il valore corrente di quel tasto. Fkey number string assegna la stringa a quel tasto. I caratteri di controllo possono essere creati facendoli precedere dal carattere ^. Un CR e' ^M. Per inserire un ^ in una stringa sono neccessari due caratteri ^ consecutivi. Qui di seguito e' riportata la tabella dei tasti funzione e il relativo numero. F1 e' il tasto funzione 1. Sf1 equivale a shift+tasto funzione 1. Cf1 corrisponde a control+tasto funzione 1. Af1 corrisponde ad alt+tasto funzione 1. La riga piu' a destra rappresenta il tastierino numerico .

key number key number key number key number key number

```
f1   59      sf1   84      cf1   94      af1  104      pgup    73
f2   60      sf2   85      cf2   95      af2  105      pgdn    81
f3   61      sf3   86      cf3   96      af3  106      home    71
f4   62      sf4   87      cf4   97      af4  107      end     79
f5   63      sf5   88      cf5   98      af5  108      arup    72
f6   64      sf6   89      cf6   99      af6  109      ardn    80
f7   65      sf7   90      cf7  100      af7  110      ar l    75
f8   66      sf8   91      cf8  101      af8  111      ar r    77
f9   67      sf9   92      cf9  102      af9  112      ins     82
f10  68      sf10  93      cf10 103      af10 113      del     83
```

La mappatura e' presa in modo tale che assomigli alla tastiera vt100/ansi. Qui di seguito sono riportati i valori delle stringhe assegnate alle chiavi.

```
number    string       key
59       " 330P"     /* F1 */
60       " 330Q"     /* F2 */
61       " 330R"     /* F3 */
62       " 330S"     /* F4 */
71       " 10"       /* home*/
72       " 33[A"     /* up arrow*/
73       " 25"       /* pgup */
75       " 33[D"     /* left arrow*/
```

```
77       " 33[C"     /* right arrow*/
79       " 05"       /* end */
80       " 33[B"     /* down arrow */
81       „ 12"       /* pgdn */
82       „ 01"       /* ins */
83       „ 177"      /* del */
```

$$ftp1

## 3.34. ftp <hostid>

Apre un canale di controllo FTP all'host remoto specificato e entra nel modo conversazione nella nuova sessione. Le risposte del server remoto vengono visualizzate direttamente sullo schermo. Per la descrizione dei comandi disponibili nella sessione FTP si veda il capitolo Sottocomandi FTP .

$$ftype

## 3.35. ftype [ascii | binary | image | logical <size>]

Questo comando mostra o fissa il modo predefinito del file di partenza (ASCII o binario) per i trasferimenti ftp. Se e' dato ftype binary o image la sessione ftp successiva viene attivata in modo binario. Una volta che la sessione e' incominciata non sono piu' necessari i comandi binari. Nel caso di logical, la grandezza della "parola" e' data da size.

$$hdlc

## 3.36. hdlc dcemode <on | off>

Se incluso nell'eseguibile controlla il livello 2 OSI X.25 . Si veda per maggiori informazioni il paragrafo x25.

Esso commuta il modo di operare del livello 2 (indirizzi di trame) in DCE. Dando il comando "hdlc ?" si ottiene la lista dei sottocomandi possibili che sono simili a quelli dell' AX.25.

$$help

## 3.37. help

Mostra un breve riassunto dei comandi .

$$history

## 3.38. - history [<numero_comandi_buffers>]

I comandi vengono salvati in un buffer circolare di una grandezza che puo' essere fissata appunto con il comando "history #", per es.: history 10 , che e' anche il valore predefinito. Digitato da solo il comando history mostrera' tutto il buffer e quindi tutta lista dei comandi mantenuta. "history 0" disabilitera' questa caratteristica.

Le frecce su e giu' scorreranno ciclicamente attraverso i vecchi comandi quando si e' nel modo comandi del NOS. Per ora non e' ancora possibile rieditare il buffer, ma solo richiamarlo.

Inoltre e' stato introdotta la possibilita' di utilizzare le parentesi quadre come sinonimo delle frecce per quelle situazioni in cui e' scomodo usarle ( tipicamente sulle tastiere 88 tasti).

( Con “[” o “freccia in alto” : richiamo comando precedente; con “]” o “freccia in basso” : richiamo comando successivo. )

$$hop

## 3.39. hop <subcommands>

Questi comandi vengono usati per testare i collegamenti di rete.

3.39.1. hop check <hostid>

Inizia una sessione hopcheck all’host specificato. Esso usa una serie di pacchetti “sonda” UDP con campi crescenti IP TTL per determinare la sequenza dei gateways nel path verso la destinazione specificata. Questa funzione e’ simile al dispositivo UNIX traceroute.

La visualizzazione dei messaggi ICMP dovrebbe essere disabilitata prima di eseguire questo comando (si veda il comando icmp trace).

3.39.2. hop maxttl [<hops>]

Mostra o fissa il valore massimo di TTL che deve essere usato nelle sessioni hop check. Questo di fatto limita il raggio della ricerca.

3.39.3. hop maxwait [<seconds>]

Mostra o fissa il massimo intervallo espresso in secondi per cui la sessione di hopcheck attendera’ per la risposta ad ogni fase del tracciamento. Il valore predefinito di tale valore e’ di 5 secondi.

3.39.4. hop queries [<count>]

Mostra o fissa il numero di pacchetti sonda UDP che saranno inviati ad ogni fase del tracciamento. Il valore predefinito di tale numero e’ 3.

3.39.5. hop trace [on | off]

Mostra o fissa il flag che controlla la visualizzazione delle informazioni aggiuntive durante la sessione di hop check.

$$hostname

## 3.40. hostname [<name>]

Mostra o fissa il nome dell’ host locale. Per convenzione esso dovrebbe coincidere con il nome del dominio primario dell’ host. Questa stringa e’ usata solo nei messaggi di benvenuto dei vari server di rete; si noti che esso non fissa l’ indirizzo IP del sistema. Se <name> e’ lo stesso di <iface> (si veda il capitolo Attach commands), questo comando cerchera’ un record CNAME domain resource che corrisponda all’ indirizzo IP del <iface>.

$$hs

## 3.41. hs

Mostra i dati relativi al driver HDLC ad alta velocita’ (se configurato e attivo).

$$icmp

## 3.42. icmp <subcommand>

Questi comandi vengono usati per il servizio Internet Control Message Protocol.

3.43.1. icmp echo [on | off]

Mostra o fissa il flag che controlla la visualizzazione asincrona dei pacchetti ICMP Echo Reply. Questa flag deve essere attivo (on) affinche' i pings singoli funzionino correttamente (si veda il comando ping) .

3.43.2. icmp status

Mostra le statistiche relative a ICMP che includono il numero di messaggi ICMP di ogni tipo inviati o ricevuti.

3.43.3. icmp trace [on | off]

Mostra o fissa il flag che controlla la visualizzazione dei messaggi di errore ICMP. Questi messaggi informativi vengono generati dai routers Internet in risposta a problemi di protocollo o congestione. Questa opzione dovrebbe essere disabilitata prima di usare il comando hop check poiche' essa dipende da messaggi ICMP Time Exceeded, e la visualizzazione asincrona di questi messaggi sara' mischiata con l'output del comando hop check.

$$ifconfig

3.41. ifconfig

Mostra un elenco delle interfacce, con una breve descrizione dello stato in cui si trovano.

3.44.1. ifconfig [<iface> [[[ <subcommand> <param>]
<subcommand> <param>]...]

Quando viene specificato soltanto il parametro <iface>, viene mostrata una descrizione estesa dello stato delle interfacce. Su una stessa linea possono essere posti sottocomandi o parametri multipli.

3.44.2. ifconfig <iface> broadcast <addr>

Assegna all' indirizzo di trasmissione dell' interfaccia <iface> il contenuto di <addr>. <addr> puo' essere o un indirizzo ax25 o un indirizzo ethernet, a seconda del tipo di interfaccia, con degli "1" nella parte di indirizzo che riguarda l' host. Esso e' legato al sottocomando netmask. Si veda anche il comando arp.

3.44.3. ifconfig <iface> description [ »description »]

Questo comando pone la descrizione dell' interfaccia uguale alla stringa specificata. Se non viene data alcuna stringa la descrizione attuale viene cancellata. La descrizione viene mostrata usando il comando ifconfig <iface> (senza parametri) e con i comandi di mailbox.

3.44.4. ifconfig <iface> encapsulation <slip | ax25 | ether | encap | ppp>

Pone l'incapsulamento dell' interfaccia iface in uno dei modi:

slip/ax25/ether/encap/ppp.

3.44.5. ifconfig <iface> forward <iface-2>

Quando e' definito un forward, tutti gli output per l'interfaccia <iface> vengono reindirizzati all' interfaccia <iface-2>. Per rimuovere il forward si deve porre <iface-2>=<iface>.

3.44.6. ifconfig <iface> ipaddress <addr>

Pone l' indirizzo IP per questa interfaccia uguale a <addr>. Questo potrebbe essere necessario quando un sistema funziona da gateway. Come un sistema con indirizzo IP 44.137.1.8 che ha un accesso Internet tramite ethernet. L'indirizzo IP Internet potrebbe essere 129.179.122.10. Il comando ifconfig ec0 ipaddress 129.179.122.10 fissa l'indirizzo IP al fine di un corretto instradamento (si noti che l'indirizzo 44.x.x.x non vviene connesso all' Internet). Si vedano anche i comandi hostname e ip address.

3.44.7. ifconfig <iface> linkaddress <hardware-dependant>

Fissa l'indirizzo dipendente dall' hardware per questa interfaccia. Per AX.25 questo puo' essere il nominativo, per l'ethernet un nuovo indirizzo ethernet.

3.44.8. ifconfig <iface> mtu <param>

Fissa la massima unita' di trasferimento ad un valore di bytes pari a param. Si veda il capitolo setting ... MTU, MSS e Window per ulteriori informazioni.

3.44.9. ifconfig <iface> netmask <address>

Fissa la maschera di sottorete per questa interfaccia. Il parametro <address> assume la forma di un indirizzo IP con degli "1" nella parte relativa all'indirizzo di rete e sottorete, e degli "0" nella parte relativa all'host. Esempio: ifconfig ec0 netmask 0xffffff00 per una rete di classe C (24 bits). Esso e' in relazione al sottocomando broadcast. Si veda anche il comando route.

3.44.10. ifconfig <iface> rxbuf <size>

Fissa la grandezza del buffer di ricezione.

$$info1

3.45. info

Il comando info da' informazioni sulla versione di Nos in uso e la sua configurazione. Le informazioni relative alla configurazione di quel dato eseguibile, si ottengono dalle vavrie linee "define" nel file config.h nel codice sorgente. In questo modo viene data automaticamente una corretta informazione riguardante la configurazione.

$$ip

3.46. ip <subcommand>

Questi comandi vengono usati per il servizio Internet Protocol.

3.46.1. ip access <permit|deny|delete> <dest addr>[/<bits>] <ifname> [lowport [highport]]

Mostra o fissa i controlli d'accesso per il funzionamento delle funzioni IP. Questo comando implementa le "router access functions" del Nos(Fp). Il parametro <permit> abilita i pacchetti verso <dest-addr> all'inoltro tramite <ifname>. Il parametro <deny> li disabilita. Se il parametro <lowport> non viene specificato sono considerate tutte le porte. Se viene dato solo

il parametro lowport viene considerata solo quella porta. Se vengono specificati i parametri <lowport> ed <highport> viene fornita una lista delle porte che si possono utilizzare o delle porte vietate. Il parametro <dest-addr> puo' essere la parola "all" per tutti gli indirizzi possibili. Il parametro <lowport> puo' essere la parola "none" per tutte le porte. La cancellazione di un accesso IP deve verificare un precedente permesso o divieto per poter eliminare quella definizione. Si Riportano ora alcuni esempi:

ip access permit 44/8 ax0
ip access deny all ax0 1 1023
ip access permit all ax0

Se non viene creata una lista d'accesso , tutte le interfacce potrebbero portare tutti i tipi. Se viene definito un controllo d'accesso per un interfaccia deve anche essere stato definito un permesso per questo interfaccia in modo tale che sia consentita la comunicazione. Cosi' un'interdizione parziale senza un permesso corrisponde ad una interdizione totale.

3.46.2. ip address [<hostid>]

Mostra o fissa l'indirizzo locale predefinito di IP. Questo comando deve essere dato prima del comando attach se deve essere usato come indirizzo predefinito per l'interfaccia.

3.46.3. ip rtimer [<seconds>]

Mostra o fissa il timeout per gli reassemblaggio dei datagrammi IP . Il valore predefinito e' di trenta secondi.

3.46.4. ip status

Mostra le statistiche relative all' Internet Protocol (IP), come il numero di pacchetti totali e i contatori dei vari tipi di errore.

3.46.5. ip ttl [<hops>]

Mostra o fissa il valore predefinito del time-to-live (ttl) riposto in ogni datagramma IP emesso. Questo limita il numero di salti che il datagramma potra' fare. L'idea e' di limitare la vita del pacchetto nel caso dovesse finire in un loop di instradamento, percio' bisogna rendere tale valore maggiore del numero di salti che ci si aspetta che i pacchetti facciano attraverso la rete. Il valore predefinito e' posto, al momento della compilazione, al valore ufficiale raccomandato per l'Internet che e' 255.

$$isat

3.47. isat [on | off]

Mostra o attiva il flag per PC AT. Di solito questo segnale viene attivato quando viene collegata un'interfaccia con un canale d'interruzione (IRQ) maggiore o uguale ad otto. Questo serve per segnalare che il secondo hcontrollore d'interruzione in un AT necessita di un ritorno di segnale. Se si sta usando il segnale di clock di un AT questo comando permettera' di misurare il tempo in millisecondi piuttosto che in clock ticks (55 millisecondi per clock ticks). Durante l'inizializzazione dell' I/O questo segnale viene attivato se la prom del monitor ha il byte standard 0xfc all'indirizzo f000:fffe.

$$kick

3.48. kick [<session>]

Forza tutti i sockets associati ad una sessione; se non e' specificato alcun argomento viene forzata la sessione attuale. Compie la stessa funzione del comando ax25 kick e tcp kick, ma e' piu' facile da digitare.

$$lock

3.49. lock [password <"passeod string"]

Blocca la tastiera o definisce la stringa come password. Se la password e' specificata allora viene salvata come stringa di accesso. Se non viene specificato nessun parametro allora la tastiera viene disabilitata se in precedenza era stata specificata una password. Se la tastiera e' disabilitata, allora viene richiesta la password. Se viene fornita la password corretta la tastiera viene sbloccata. La definizione della password e il disinserimento della tastiera puo' essere effettuato soltanto per mezzo della tastiera della console di sistema o del file autoexec.nos. La password non puo' essere mostrata.

$$log

3.50. log [stop | <filename>]

Mostra il nome del file di log attuale o fissa il parametro filename per la registrazione dell'attivita' dei servers. Se come argomento viene dato stop, la registraione viene terminata (i servers non vengono comunque influenzati). Se come argomento viene dato il nome di un file, le annotazioni di sessione del server verranno appese ad esso.

$$lpq

3.51. lpq -S <nome_server> -P <nome_printer>

Lpq visualizza lo stato dei job di stampa.

Dato che piu' persone possono voler utilizzare contemporaneamente la stessa stampante, un job puo' non essere stampato immediatamente; deve cioe' attendere in una coda fino a che la stampante non risultera' disponibile.

Con il comando 'lpq' viene interrogato il server di stampa e viene mostrato lo stato dei job di stampa processati a quell'istante.

Anche in questo caso se non vengono specificati il nome_server ed il nome_stampante vengono utilizzati i valori predefiniti, che sono rispettivamente :"127.0.0.1" ed "lp".

Anche il comando 'lpq' ha a disposizione alcune opzioni:

- S nome del server
- P nome della stampante
- l visualizzazione completa della coda
- c chiamata da lpc (comando presente solo sul server)

Come valore predefinito e' stata attivata l'opzione -l che visualizza lo stato dei job in forma piu' dettagliata.

$$lpr

3.52. lpr -S <nome_server> -P <nome_stampante> <nomefile>

Il comando 'lpr' esegue la stampa di un file in rete . Se non sono specificati il nome_server ed il nome_stampante vengono utilizzati i valori predefiniti (che per il nome_server e' l'indirizzo stesso del client e per nome_stampante e' "lp").

Il comando ‘lpr’ ha a disposizione un insieme di opzioni che introducono una serie di possibilita’ e permettono una gestione piu’ flessibile della stampante.

Si riporta di seguito la lista delle opzioni a disposizione:

- S nome del server
- P nome della stampante
- C classe del job
- J nome del job
- T titolo del job
- # numero di copie
- w larghezza di pagina
- i indentazione di x spazi
- f file di tipo ordinario
- p utilizzo del formattatore di stampa
- c file di tipo ‘cifplot’
- d file di tipo testo
- g file di tipo plot
- l abilita i caratteri di controllo
- n file di tipo ditroff
- t file di tipo troff
- v file di tipo raster image
- h elimina la pagina di burst

Per l’impiego del cliente LPR con file documento contenenti caratteri non in formato ASCII , cioe’ con caratteri non stampabili come i caratteri di controllo , si ricorda che e’ necessario stampare prima su file . A tale scopo si dovra’ utilizzare il driver della stampante specifica all’interno del programma di scrittura dei testi.

Per esempio si supponga di impiegare regolarmente il noto programma Word-Star Professional per la scrittura di testi ed inoltre di avere a disposizione sul server di stampa LP una stampante ad aghi con la classica emulazione IBM Proprinter . Le operazioni necessarie per stampare in rete il proprio documento riguardano la scelta della stampante che in questo caso sara’ “> nome_file”. A questo punto il file di nome “nome_file” potra essere inviato al print server con il comando:

lpr -S <nome_print_server> -P <nome_stampante> -l nome_file

Si noti l’opzione ‘-l’ per comunicare al server di stampa di abilitare i caratteri di controllo contenuti del file.

Si consideri un altro esempio, forse piu’ vicino alla realta’ dell’ufficio moderno. Si supponga di operare regolarmente all’interno del popolare ambiente Windows della Microsoft. Per stampare in rete non si dovranno usare i comandi dei menu predisposti a questo scopo , poiche’ non verra’ usato alcun dispositivo attaccato sul sistema locale, come LPT1, o COM1. Invece si dovra’ collegare il driver o i drivers delle stampanti installate dentro Windows a FILE . Per realizzare cio’ si dovra’ utilizzare il Pannello di controllo , opzione “stampanti” e poi “collega”. A questo punto una volta elaborato il proprio documento all’interno del programma di scrittura dei testi, per esempio Write o WinWord, si potra’ scegliere di stampare come sempre. Windows chiedera’ il nome del file da utilizzare per la stampa. Solo a questo punto si sara’ pronti ad inviare il proprio documento in rete verso la stampante. Supponendo che all’interno di Windows fosse installato il driver per una stampante laser Canon LPB-4 Plus, e sul server sia disponibile la stessa con il nome “laser”, useremo la sintassi seguente :

lpr -S <nome_print_server> -P laser -l nome_file

Si noti di nuovo l'opzione '-l', sempre necessaria per queste operazioni.

Infine si tenga presente che i file cosi' preparati per l'invio al print server sono da considerare binari perche' contenenti i caratteri di controllo per la stampante.

$$PRINT

Da notare che in mancanza del cliente LPR all'interno della propria versione NOS e' possibile comunque inviare stampe in rete con il comando FTP come di seguito spiegato. Unico limite di questa modalita' e' l' impossibilita' di inviare al server ftp piu' stampe in contemporanea a causa della mancanza di un processo spooler remoto, presente invece nel print_server (lpd).

Si supponga di voler stampare il file "pippo.txt" presso la stampante collegata su LPT1 della macchina di nome ik3rpc. A tale scopo e' sufficiente aprire una sessione con il comando "ftp ik3rpc" . Una volta che si e' guadagnato l'accesso al disco di tale macchina si effettui un "put pippo.txt lpt1". Se la stampante e' accesa Il risultato del comando sara' prodotto su carta. Naturalmente se la stampante e' collegata su LPT2 il comando essenziale diventa "put pippo.txt lpt2".

$$lprm

3.53. lprm -S <nome_server> -P <nome_stampante> <num_job>

Lprm rimuove dalla coda i job in attesa di essere stampati.

Se si cambiasse idea dopo aver inviato al server un file per la stampa e se il file non e' ancora stato stampato, e' possibile rimuoverlo dalla coda con il comando lprm.

Il parametro <num_job> (numero del job che si vuole cancellare) vieneassegnato dal server nel momento in cui viene lanciata la stampa e puo'essere ottenuto con il comando lpq descritto in precedenza.

La cancellazione di un job dalla coda di stampa puo' essere effettuata solo da parte del proprietario del file.

A tale regola non e' assoggettato il server di stampa che puo' cancellare dalla coda un file, qualunque sia il proprietario del file stesso.

Opzioni possibili:

- S nome del server
- P nome della stampante
- c chiamata da lpc (comando presente solo sul server)
- a listing in formato esteso

$$lzw

3.54. lzw [<subcommand>]

Il comando lzw ha la capacita' di comprimere i dati di alcuni sockets. Questo comando definisce o cambia le loro definizioni. Solitamente si lascia il valore predefinito.

3.54.1. lzw mode <fast|compact>

Mostra o fissa il metodo utilizzato per comprimere i dati dei sockets specifici. Di solito SMTP puo' utilizzare la compressione.

3.54.2. lzw bits <number>

Mostra o fissa il numero di bits usati per definire l'entita' della compressione.Piu' e' grande il numero di bits piu' sara' grande lo spazio necessario. Il campo varia da 9 a 16.

$$mail

3.55. mail

Questo comando attivera' un comando di uscita in shell DOS. Il mailer usato viene definito con la variabile d' ambiente DOS che ha come valore predefinito BM.EXE. DA notare che se si usa BMB.exe saranno necessari almeno 110 Kb liberi perche' questo si avvii. Controllare prima quanta memoria si ha a disposizione con il comando "memory status" osservando il campo"coreleft".

Al termine del processo di ricezione della posta da parte del server SMTP si osservera' alla console uno o piu' messaggi "new mail arrived for <nome_utente>" accompagnato da un beep.

$$Messaggio_vocale

Dalla versione 1.15 del NOS, all'atto della ricezione della posta oltre al messaggio e al beep , si udira' un messaggio vocale sintetizzato. Cio' avvera' se presente nel direttorio corrente il file "speech.com" e se ci sara' abbastanza memoria (vedi campo 'coreleft' del comando "memory status") per eseguire tale comando in shell. Il messaggio vocale sintetizzato e': "NEW MAIL ARRIVED" .

Cio' e' utile per richiamare maggiormente l'attenzione dell'operatore distante dalla macchina oltre che per dimostrare l'applicazione di messaggi vocali ad eventi sul sistema.

$$man

3.56. man all | <stringa> [<output_file>]

Se presente il file "ka9q_doc.man" nel direttorio ~\usr\doc, e

se all'interno di tale file sono stati inseriti degli indici , il comando "man"

consentira' di consultare a video l'intero testo contenuto nel paragrafo associato attraverso il comando "more".

La paragrafatura si ottiene inserendo nel testo un indice composto dal doppio simbolo '$' seguito dall'argomento . Il programma cerchera' e mostrera' il testo tra un indice e l'altro.

Se verra' data la parola "all" come argomento , verranno mostrati tutti gli argomenti presenti nel manuale in ordine alfabetico.

Se verra' dato un nome di file come <output_file> , allora l'argomento richiesto verra' scritto in quel file.

Da notare che il testo e'in formato ASCII. Inoltre e' necessario infondo al file un ultimo doppio carattere '$' per segnalare la fine del testo al programma man.

L'introduzione di questo comando aggiunge flessibilita' all'ambiente NOS. Esso infatti consente di avere un contatto piu' stretto con la documentazione tecnica relativa e quindi di sviluppare prima e con piu' comodita' quella abilita' necessaria alle operazioni TCP/IP.

$$mbox

3.57. mbox [<subcommand>]

Mostra lo stato del server mailbox del sistema (se configurato).

3.57.1. mbox attend [yes | no]

Mostra o fissa il segnale di “operatore presente”. Esso viene usato per avvisare l’utente del mailbox se l’operatore della stazione e’ disposto a colloquiare conquesti (chat).

3.57.2. mbox expert <on|off>

Mostra o fissa il livello presunto di un utente mailbox. Quando e’ attivato appare il seguente prompt: (un breve prompt) “>”. Quando non e’ attivato appare un prompt lungo con la prima lettera di ogni comando possibile sul mailbox. L’utente puo’ passare al modo esperto digitando X per avere il prompt di tipo breve.

3.57.3. mbox fwdinfo [”forward info”]

Mostra o fissa l’info del mailbox forward da includere nella linea R:

per i bollettini BBS inoltrati. Una stringa vuota (“”) cancella l’info field.

Esempio: netrom fwdinfo “HNLNET BBS”.

Apparira’ ‘[HNLNET BBS]’ nella linea R: .

3.57.4. mbox haddress [”home-address”]

Mostra o fissa il proprio indirizzo da includere nella linea R: per i bollettini BBS inoltrati. Una stringa vuota (“”) cancella il campo.

Esempio: netrom haddress “#CRW.OR.USA”

Apparira’ ‘@WG7J#CRV.OR.USA’ nella linea R:

(quando ax25 mycall e’WG7J).

3.57.5. mbox jumpstart <on|off>

Mostra o fissa il codice del mailbox jumpstart. Quando attivato ed un nodo conosciuto si connette al mailbox non e’ necessario inviare linee aggiuntive per attivare il mailbox dato che il prompt viene inviato direttamente. Attenzione: quando viene specificato on e’ necessario un certo tempo per riconoscere tutti i nodi e la connessione AX25 potrebbe avere un avvio herrato poiche’ non si attende il controllo del protocollo di livello 3. Specialmente se RSPF e’ in modo virtuale, si potrebbe incorrere in seri problemi.

3.57.6. mbox kick

Riattiva il mailbox dopo l’intervento dei timeouts.

3.57.7. mbox maxmsgs

Mostra o fissa il numero massimo di messaggi per area quando viene mostrato all’utente un messaggio. Questo riserva molte locazioni di memoria per ogni sezione del mailbox.

3.57.8. mbox motd [”message string”]

Mostra o fissa il messaggio di benvenuto del giorno del mailbox.

3.57.9. mbox nrid <on|off>

Mostra o fissa il segnale “id” del nodo Netrom. Quando viene attivato il modo,l”id” del nodo viene visualizzato sulla linea di prompt.

3.57.10. mbox operator [<address>]

Il comando Operator mostra o fissa un indirizzo alternativo per l'operatore di controllo. Quando esso e' fissato e "mbox attended" e' off , oppure l'operatore e' in shell e l'utente del mailbox richiede il comando Operator, allora la sessione ttylink generata viene rediretta all'indirizzo specificato. Esso lavora come prima che all'utente del mailbox venga notificato il messaggio "Unattended". Se il sistema <address> risultasse anch'esso senza operatore (mbox attended off), allora verra' replicato un messaggio "Unattended" allo stesso modo , pero' generato dal sistema (<address>) remoto. Questo e' un modo per avere un sistema "senza operatore con operatore!".

3.57.11. mbox password <"password string">

Fissa una stringa come password che deve essere presentata al Sysop quando si passa in questo modo dal mailbox (si deve usare il comando "@" ed avere il privilegio richiesto dal login name/password in f/ftpusers). Quando viene definita una password (al massimo di trenta caratteri) viene visualizzata con cinque numeri prima che sia consentito l'accesso. I cinque numeri rappresentano le cinque locazioni di carattere nella stringa data, per cui il primo carattere corrisponde al numero 0. Righe multiple di cinque caratteri possono essere inviate agli stupidi ficcanaso. La fine dell'invio della password e' segnalata con una linea vuota. Se cio' da' un buon risultato si entra nel modo sysop. La definizione della password puo' essere fatta solo tramite la tastiera della console o dal file di startup autoexec.nos. La password non puo' essere mostrata.

3.57.12. mbox gth ["gth info"]

Mostra o fissa l'info per il qth nella linea R: per i messaggi BBS

inoltrati.Esempio: netrom qth "Driebruggen, NL"

3.57.13. mbox secure <yes|no>

Mostra o fissa le opzioni di sicurezza per gli utenti del mailbox gateway. Se fissata, non e' permesso agli utenti collegati alla BBS tramite il Telnet di usare il gateway. Se non e' fissata chiunque puo' usare il gateway (si noti: non vi e' il controllo di Bozo). Anche il comando "Send" del mailbox viene disabilitato ad eccezione che per le connessioni ax25 e netrom.

3.57.14. mbox smtptoo <yes|no>

Mostra o fissa il segnale che serve ad includere le testate SMTP nei messaggi BBS. Quando e' attivato, le testate SMTP vengono incluse nei messaggi. Quando non e' fissato le testate non vengono incluse.

3.57.15. mbox status

E' il comando alternativo a "mbox" dato al prompt net>.

3.57.16. mbox timer [<seconds>]

Mostra l'intervallo corrente ed il tempo rimanente oppure fissa il timer di forwarding del mailbox.

3.57.17. mbox tiptimeout

Mostra o fissa il valore di timeout per una connectione tip. Dopo timeout secondi di inattivita' la connessione viene chiusa.

3.57.18. mbox trace [yes|no]

Mostra o fissa il segnale per il tracciamento del mailbox forward . Attualmente il tracciamento e' ridotto al minimo, ma chiunque stia lavorando al codice del mailbox/forward ha una flag Mtrace in comune.

3.57.19. mbox utc <offset>

Mostra o fissa il fuso orario in cui ci si trova secondo l'ora ZULU . Sono accettati sia numeri positivi che negativi, inoltre sono effettuati i calcoli ad intervalli di tempo di anni o mesi.

3.57.20. mbox zipcode zip

Mostra o fissa l'info dei campi zip (zip code e' per noi il C.A.P.) per la linea R:. Questo campo e' lungo al massimo sette caratteri. Negli USA lo zip ha una lunghezza di solo sei numeri. In Olanda utilizzano 4 cifre uno spazio e due lettere. Ogni amministrazione PPTT vuole inventare qualcosa di proprio!

$$memory

3.58. memory <subcommand>

Questi comandi vengono usati per l'allocazione di memoria.

3.58.1. memory debug [on|off]

Mostra o attiva il segnale di debug dell' allocatore di memoria. Se attivato l'informazione di debug viene scritta nel file di log contenente la maggior parte dei flag e dei parametri provenienti dalle routine di allocazione di memoria.

3.58.2. memory efficient [yes | no]

Mostra o setta l'algoritmo di ricerca del buffer di memoria. Quando specificato, la ricerca parte sempre dall'inizio della lista vuota. Questo e' piu' lento ma consente una minima frammentazione di memoria. Quando non e' specificato la ricerca parte dalla fine della lista, il che comporta una maggiore frammentazione della memoria ma una maggiore velocita'.

Si dovrebbe includere questo comando come prima linea
nell'autoexec.nos.

3.58.3. memory freelist

Mostra la lista dei segmenti di memoria libera dello storage allocator . Ogni voce consiste in un undirizzo di partenza , in esadecimale, e la grandezza in bytes decimali.

3.58.4. memory ibufsize [<size>]

Mostra o fissa l'ampiezza dei buffer per l'insieme dei buffers di interrupt . L'ampiezza dovrebbe essere fissata pari al tipo piu' grande di buffer piu' un header di 22. Per esempio: se l'ax25 e' la sola interfaccia ed e' definito un pacchetto di lunghezza 256, il bufsize dovrebbe essere pari a 256+10*6+22. L'addendo 10*6 e' l'header dell'ax25 (sorgente,destinazione e 8 digipeaters).

3.58.5. memory minheap [<number>]

Mostra o fissa l'ampiezza minima dell'heap che deve essere allocato prima che si passi alla shell del DOS. Questo assicura un heap libero cosicche' il Nos possa girare senza mai rimanere a corto di memoria anche solo per un momento.

3.58.6. memory nibufs [<number>]

Mostra o fissa il numero di buffers di interrupt buffer pool. Se il numero di buffers e' fissato, i dati nella memoria memory status vengono azzerati per il numero di fallimenti dell'interrupt buffer. Una buona regola per il numero di buffers e' osservare le statistiche e tenere un minimo di due buffers liberi, aumentandolo o diminuendolo a seconda dei casi.

3.58.7. memory sizes

Mostra un istogramma delle grandezze richieste degli allocatori. Ogni histogram bin e' un ordine di grandezza binario (cioe' un fattore di 2).

3.58.8. memory status

Di seguito viene mostrato un esempio di output dita le comando :

```
net> mem st
heap size 111712, avail 47632 (42%), morecores 179, coreleft 39512
allocs 3425, frees 3222 (diff 203), alloc fails 0, invalid frees 0, overused 0
garbage collections yellow 0, red 0
interrupts-off calls to malloc 0, free 0
Intqlen 15, Ibufsize 2048, Iminfree 15, Ibuffail 0
net>
```

Mostra un riassunto dei dati relativi agli allocatori.

La prima linea mostra l'heap con la sua grandezza totale, la quantita' di memoria disponibile espressa in bytes con a fianco una percentuale della grandezza totale dell' heap , infine la quantita' di memoria lasciata libera (cioe' non piazzata sull'heap all' avvio) e percio' disponibile per i sottocomandi di shell DOS.

La seconda linea mostra il numero totale di chiamate necessarie per allocarsi ed i blocchi di memoria liberi, la differenza di questi due valori (ossia il numero di blocchi allocati outstanding), il numero di allocazioni richieste che sono state negate a causa della mancanza di memoria ed il numero di chiamate che tentano di liberare due volte lo stesso blocco od un puntatore troncato.

La terza linea mostra lo stato del processo "garbage collection". La garbage collection e' appunto un processo separato che gira ogni secondo. Infatti se la somma totale della memoria rimasta per la shell (coreleft) e quella disponibile nell'heap (avail) scendono al di sotto di un certo valore (memory thresh), viene incrementato il contatore "yellow alert". Se invece la memoria scende al disotto della meta' del valore "memory thresh" viene incrementato il contatore "red alert".

La quarta linea mostra il numero di chiamate alla funzione malloc che avvengono con gli interrupts off. In una situazione normale questi valori dovrebbero essere zero. La quarta linea mostra inoltre le statistiche per l'insieme ( pool ) speciale dei buffers di grandezza fissa utilizzati per le richieste di memoria al momento in cui avviene un interruzione. Le variabili mostrate sono: il numero di buffer al momento nel pool, la loro grandezza ed il numero di richieste che sono fallite a causa dell'esaurimento del pool.

La quinta linea mostra il settaggio attuale del buffer del pool di interruzione, il suo valore minimo ed il numero di buffers non disponibili. Questi dati vengono azzerati quando viene dato il comando memory nibufs <number>.

3.58.9. memory thresh [<size>]

La grandezza di soglia della memoria espressa in bytes. Se la memoria scende sotto questo valore non sono piu' accettate nuove sessioni.

$$mkdir

3.59. mkdir <dirname>

Crea una sotto-direttorio nel direttorio attuale di lavoro.

$$mode

3.60. mode <iface> [vc | datagram]

Controlla il modo di trasmissione predefinito sull'interfaccia ax25 specificata.

In modo datagram i pacchetti IP vengono incapsulati in trame AX.25 UI e trasmessi senza altri meccanismi di legame come le connessioni o le conferme.

In modo vc (virtual circuit) i pacchetti IP vengono incapsulati in trame AX.25 I e viene data loro risposta a livello link secondo il protocollo AX.25. Se necessario vengono aperte connessioni a questo livello (link).

In entrambi i modi, per mappare gli indirizzi IP su AX.25 viene

usato l'ARP. hI valori predefiniti possono essere ignorati con i bits TOS

(type-of-service) nelle teste IP . L'attivazione del bit "reliability" fa si' che

vengano usate trame I, mentre con l'abilitazione del bit "low delay" vengono utilizzate le trame UI (l'effetto dell'abilitazione di entrambi i bits non e' definito ed e' soggetto a cambiamenti).

In entrambi i modi se il datagramma e' piu' grande del MTU dell'interfaccia viene effettuata una frammentazione del livello IP. In modo vc, tuttavia, il datagramma risultante, se e' ancora piu' grande del parametro AX.25 paclen, viene ulteriormente frammentato a livello AX.25. Nella frammentazione AX.25 i datagrammi vengono spezzati in numerose trame che vengono ricomposte al momento della ricezione prima di essere fatte passare all'IP. Questo metodo e' sempre preferibile alla frammentazione Ip a causa della diminuzione dell'overhead (l'IP header non e' ripetuto in ogni frammento) e a causa dell'aumentata robustezza (un frammento pereduto e' immediatamente ritrasmesso dallo livello datalink .

$$more

3.61. - more <filename> [<searchstring>]

Sono state apportate radicali modifiche al codice del more (usato sia come comando a se' che nel comando dir), che ora permette di spostare il testo avanti e indietro ed effettuare la ricerca di una stringa di testo.

Fare attenzione al fatto che i tasti "freccia in su", "freccia in giu'", "pagina su" e "pagina giu'" vengono riconosciuti correttamente SOLO se definiti con la stringa ANSI di default del NOS; ridefinendoli diversamente si perda' la funzione ad essi associata e si dovranno usare i comandi alternativi sotto indicati.

Comandi disponibili:

```
PRINCIPALI                          ALTERNATIVI
avanti di una linea                 +,CR  come
indietro di una linea               -     come
PgDn  avanti di una pagina          ^J    come PgDn
```

```
PgUp  indietro di una pagina          ^U    come PgUp
q(uit)     chiude la sessione “more”
n(ext)     ripete  ricerca in avanti iniziata con la  stringa
           opzionale [<searchstring>].
```

p(revious) ripete ricerca indietro iniziata con la stringa

opzionale [<searchstring>].

```
1G(o)      mostra l’inizio del file
G(o)       mostra la fine del file.
```

$$motd

3.62. motd [<”message of the day”>]

Mostra il messaggio corrente o fissa come messaggio di benvenuto il contenuto della stringa definita.

$$multitask

3.63. multitask [on | off]

Mostra o attiva il segnale di multitask. Se attivo , il Nos continua a fornire le interruzioni e gestire le normali operazioni, anche quando e’ stato eseguito un comando di uscita in shell DOS. Alcuni programmi non gradiscono di perdere tempo prezioso cosi’ in quanto, in questo modo, puo’ risultare una cattiva elaborazione dei dati. La cancellazione del flag ferma tutte le attivita’ quando il Nos gira in background.

$$netrom

3.64. netrom <subcommand>

3.64.1. netrom acktime [<milliseconds>

Mostra o fissa il timer che regola il ritardo della risposta in modo simile all’ ax25 t2.

3.64.2. netrom alias <alias>

Questo comando fissa il netrom alias per questa stazione. Altre stazioni possono essere connesse al nominativo ax25 e all’alias netrom . L’alias viene trasmesso con una emissione broadcast Netrom .

3.64.3. netrom bcnodes <iface>

Avvia una trasmissione immediata della lista dei nodi su <iface>.

3.64.4. netrom call <call>

Mostra o fissa il nominativo che deve essere usato dall’interfaccia netrom. Questo e’ un metodo per abbreviare il comando “ifconfig netrom linkaddress call”.

3.64.5. netrom connect <node>

Stabilisce una connessione al nodo.

3.64.6. netrom choketime [<milliseconds>]

Mostra o fissa il tempo di breaking di un choke in trasmissione . Il choke e' il termine netrom che si usa per le condizioni di controllo del flusso.

3.64.7. netrom derate [on | off]

Mostra o attiva il derating automatico delle routines di netrom in caso di mancato collegamento.

3.64.8 netrom interface <iface> <quality>

Definisce un interfaccia netrom iface. Il parametro quality e' compreso tra 1 e 255. Solitamente per motivi di compatibilita' viene fissato a 192. Viene effettuato un controllo per verificare che l'interfaccia sia del tipo CL_AX25 ,cioe' del tipo di mezzo capace di supportare Netrom. Si noti che alias non e' piu' specificato su questa linea. Si deve usare il comando netrom alias.

3.64.9. netrom irtt [<milliseconds>]

Mostra o fissa l'irtt (initial round trip time).

3.64.10. netrom kick <&nrcb>

Forza il blocco di controllo affinche' l'attivita' possa proseguire.

3.64.11. netrom load [<filename>]

Quando e' pienamente implementato esso ricarica una lista salvata di nodi netrom per continuare dal punto in cui ci si trovava al momento in cui si e' salvata la lista. Tutte le letture della lista vengono diminuite temporalmente proprio come se fosse trascorso il tempo normale. Se questo ha richiesto del tempo la vostra lista potrebbe essere vuota come se tutti le voci fossero scadute.

3.64.12. netrom minquality [<value>]

Mostra o fissa il valore minimo del parametro quality per riconoscere un "node entry". Le voci (entry) al di sotto di questo valore non vengono considerate utili.

3.64.13. netrom nodefilter <subcommand>

Esegue operazioni di filtro dei nodi.

3.64.13.1. netrom nodefilter add <neighbor> <iface>

Si accettano da <neighbor> (nodi vicini o adiacenti) gli aggiornamenti sui nodi.

3.64.13.2. netrom nodefilter drop <neighbor> <iface>

Non si accettano da <neighbor> gli aggiornamenti sui nodi.

3.64.13.3. netrom nodefilter mode [node | accept | reject]

Mostra o fissa lo schema iniziale del filtro. Il parametro none fa si' che venga accettato tutto. Il parametro accept accetta solo dai nodi definiti nelle liste netrom nodefilter add. Reject non accetta dai nodi definiti nelle liste netrom nodefilter add.

3.64.14. netrom nodetimer [<seconds>]

Mostra o fissa l’intervallo di tempo con cui viene trasmessa la lista dei nodi locale.

3.64.15. netrom obsotimer [<seconds>]

Mostra o fissa il tempo che icnide sulla qualita’ di un nodo netrom.

3.64.16. netrom promiscious [on | off]

Abilita i nodi con una qualita’ maggiore di quella definita con il parametro minquality. Se il suo valore e’ on sono ricevuti tutti i nodi indipendentemente dal nodefilter.

3.64.17. netrom qlimit [<bytes>]

Mostra o fissa il limite massimo della coda di ricezione. E’ simile al comando ax25 window.

3.64.18. netrom reset <&nrcb>

Rimuove il blocco di controllo. Il blocco di controllo puo’ essere trovato con il comando socket.

3.64.19. netrom retries [<value>]

Mostra o fissa il numero massimo di tentativi di connessioni o disconnessioni o i dati.

3.64.20. netrom route <subcommand>

Comandi netrom di instradamento.

3.64.20.1. netrom route add <alias> <destination> <iface> <quality> <neighbor>

Aggiunge una rotta netrom .

3.64.20.2. netrom route drop <destination > <neighbor> <iface>

Rimuove una rotta netrom .

3.64.20.3. netrom route info <destination>

Visualizza la rotta che porterebbe a destinazione i dati di una connessione.

3.64.21. netrom status

Visualizza tutte le connessioni netrom.

3.64.22. netrom save [<filename>]

Quando pienamente implemento, salva in memoria la lista dei nodi netrom attuale nel file netrom.sav o nel file specificato dal parametro <filename>, se specificato.

3.64.23. netrom timertype [exponential | linear]

Mostra o fissa il tipo di contatore di “backoff”.

3.64.24. netrom ttl [<hope>]

Mostra o fissa il numero massimo di salti fatti da un trama prima di essere scartata, se non e' riuscita a trovare la sua destinazione.

3.64.25. netrom user [<username>]

Mostra o fissa il proprio username di netrom. Esso e' utilizzato nelle connessioni "esterne".

3.64.26. netrom verbose [off | on]

Mostra o attiva il segnale verbose. Se fissata, tutti i nodi conosciuti vengono trasmessi ogni volta che il nodetimer scade. Se e' nello stato off, viene inviato solo l'identificatore del nodo locale.

3.64.28 netrom window [<trame>]

Mostra o fissa la dimensione della finestra mobile. Questo e' il valore piu' grande di ricezione e trasmissione che si possa negoziare.

$$nntp

3.65. nntp < sottocomandi>

Network News Transfer Protocol ha i seguenti sottocomandi:

3.65.1. nttp addserver <nntpserver> <secondi> [<campo>] [<gruppi>]

Aggiunge un server nntp a cui chiedere ogni intervallo di secondi per i nuovi articoli specificati dal comando nntp gruppi. Il campo puo' essere un limite di tempo per la richiesta (query) come nntp addserver qualcheserver 600 22:00-23:00, per esempio per chiedere solo tra le 22:00 el 23:00. Possono essere usati piu comandi "nntp add" per concatenare gruppi (fino a 512 byte). L'intervallo secondi puo' essere fissato a 0, cosi che normalmente il cliente nntp non effettui alcuna richiesta regolarmente ; tuttavia il comando nntp kick nntpserver puo' essere usato per attivare una sessione.

3.65.2. nntp direttorio <direttorio>

Mostra o fissa il direttorio predefinito per accodare i bollettini (news).

3.65.3. nntp dropserver <nntpserver>

3.65.4. nntp groups <group> [<group> ...]

Mostra o fissa i gruppi news correntemente installati.

3.65.5. nntp kick <nntpserver>

Attiva il cliente allo scopo di contattare il server installato con il comando "nntp addserver" .

3.65.6. nntp listservers

Elenca i servers correntemente definiti.

3.65.7. nntp quiet [yes | no]

Mostra o fissa il segnale che controlla la visualizzazione dei nuovi messaggi nntp ricevuti. Il segnale "smtp quiet" controlla il beep che segue il messaggio. Se entrambi sono disattivati , non viene mostrato nulla, evitando cosi di avere come output sul monitor molti -more- specialmente quando si opera senza operatore (unattended).

3.65.8. nntp trace <level>

Mostra o fissa il livello corrente di tracciamento per il cliente nntp. 0 equivale a tracciamento disabilitato, 1 (predefinito) mostra errori seri. 2 mostra come 1 e gli errori transienti. 3 mostra come 2 e la progressione della sessione. 4 mostra come 3 e gli articoli effettivamente ricevuti. 5 mostra gli errori.

$$nrstat

3.66. nrstat

Mostra le statistiche dell'interfaccia netrom , se configurata.

$$param

3.67. param <iface> [<param> ...]

Invoca una procedura di controllo specifica per un dato dispositivo. Su di una interfaccia KISS TNC, param invia pacchetti di controllo per il TNC. I bytes di dati vengono trattati come decimali. Per esempio, il comando param ax0 1 255, fissera' il timer di attivazione del trasmettitore (keyup) (campo tipo = 1) sull'interfaccia configurata come ax0 a 2,55 secondi (255 x .01 sec). Su di una interfaccia SLIP , il comando param consente di leggere o di fissare la velocita' (baud rate). Su linee asincrone possono essere (e possibilimente dovrebbero esserlo) attivati i segnali DTR e RTS con il comando: "param iface dtr 1 "e "param iface rts 1". L'implementazione di questo comando per i vari drivers d' interfaccia e' incompleta e soggetta a cambiamento. I comandi attualmente definiti sono:

```
  0    Dati           - Dati normali KISS
  1    Txdelay        - fissa il ritardo di trasmissione per il TNC
                        (txdelay).
  2    Persist        - fissa il ritardo di  persistenza per il TNC.
  3    Slottime       - fissa il ritardo di slottime per il TNC.
  4    TxTail         - fissa il rittardo di TX tail per il TNC.
  5    Fulldup        - fissa il full-duplex per il TNC.
  6    Hardware       - dipende dall'hardware.
  7    TxMute         -
  8    DTR             - 0 = giu', 1 = su.
  9    RTS             - 0 = giu', 1 = su.
 10    Speed         - baud rate.
 11    Enddelay     -
 12    Gruppo        -
 13    Idle             -
 14    Min             -
 15    Maxkey       -
 16    Wait           -
 17    Parity         - 0 = none, 1 = even, 2 = odd parity.
129    Down         -
130    Up             -
        Blind          - Ignora i segnali di handshake (DTR, DSR ecc..)
254    Return2      - Alcuni tnc ne hanno bisogno.
255    Return        - Riporta un TNC dal modo KISS al modo
        comandi.
```

$$ping

3.68. ping <hostid> [<len> [<interval> [<incflag>]]]

Effettua un ping ( invia pacchetti ICMP per richiesta di eco a <hostid>) all'host specificato. Il valore predefinito del campo dati contiene solo un indicazione sull'ora (timestamp) per

consentire la determinazione del tempo di andata e ritorno (rtt); se viene dato l'argomento opzionale di lunghezza (length), viene aggiunto al pacchetto ping l'appropriato numero di bytes di dati (che consiste nell' esadecimale 55) .

Se viene specificato l'interval , i pings verranno ripetuti indefinitivamente all'intervallo specificato, in secondi; altrimenti viene effettuato un singolo ping, a colpo unico . Le risposte per pings a colpo singolo appaiono in modo asincrono sullo schermo in modo comandi, hmentre per pings ripetuti si crea una sessione che puo' essere sospesa e ripresa. L'azione di ping continuera' finche' non verra' azzerata manualmente.

L'opzione incflag provoca pings ripetuti con l'incremento dell'indirizzo IP di destinazione per ogni ping; questa e' una caratteristica sperimentale per la ricerca di blocchi di indirizzi IP per hosts attivi.

$$pop

3.69. Sottocomandi pop

Il cliente pop fornisce un'interfaccia automatica verso il pop server che e' totalmente trasparente all'utente. Cio' che e' necessario fare e' fissare alcuni parametri chiave cosicche' il cliente li utilizzera' sia per la registrazione con il server che per lo scambio dei dati ( si veda a tale scopo il comando "pop userdata"). Comunque per saperne di piu' si legga rfc 937. Il seguente paragrafo descrive i sottocomandi pop.

3.69.1. pop mailbox [<name>]

Mostra o fissa il nome del file per la raccolta dei messaggi ricevuti a name . Di solito coincide con il nome dell'utente . Il server tiene la posta nel mailbox name situato nel direttorio ~/spool/mail/name.txt. Questo e' un parametro obbligatorio.

3.69.2. pop mailhost [<ipaddr>]

Mostra o fissa il nome dell'host o l'indirizzo IP del pop server a ipaddr per connessione POP. Questo e' un parametro obbligatorio.

3.69.3 pop kick

Inizia subito una connessione POP per controllare se ci sono
nuovi messaggi.

3.69.4 pop timer [<seconds>]

Mostra o fissa il timer, in secondi, tra connessioni POP (successive) verso il computer remoto (mailhost) e il tempo che rimane per la connessione successiva. Quando il timer non viene impostato il cliente pop viene attivato solo con il comando pop kick.

3.69.5. pop userdata [<username> <password>]

Senza argomenti, esso mostra solo il nome dell'utente

(username). Altrimenti installa il nome dell'utente e parola chiave da inviare al computer remoto per la convalida.

Da ricordare: il nome e la password dovrebbero essere anche
definite nel file popusers presso il server pop.

$$popmail

3.70. popmail <sottocomandi>

popmail e' una implementazione piu' nuova di client/server. Esso puo' gestire entrambi i client/server di tipo 2 e 3. Per la funzionalita' vedi pop , ma i sottocomandi sono differenti.

3.70.1. popmail addserver <host> [<seconds>] [hh:mm-hh:mm] <protocol> <mailbox> <username> <password>

Aggiunge host come pop server. Quando viene dato il parametro seconds, viene attivato un timer allo scopo di effettuare le connessioni verso il serve per la posta con quell'intervallo . Se seconds non viene specificato non verra' attivata alcuna richiesta al pop server. Sara' dunque' necessario attivare manualmente il cliente con il sottocomando kick. Quando viene dato hh:mm-hh:mm allora solo a quell'esatto intervallo di tempo verra' effettuata la richiesta. Il protocollo puo' essere sia POP2 che POP3 (se configurati ; si veda il comando "info"), dipende dal servizio di posta che il server fornisce. Da notare che POP2 e' stato sorpassato da POP3. Mailbox e' il nome del file nel quale viene scritta la posta presso il server e da dove quindi verra' prelevata. Username e password sono i parametri di convalida per questo host. Da notare che all'immissione di questo comando il nome di host viene controllato. Se quest'ultimo risultera' inesistente verra' mostrato un messaggio di errore.

3.70.2. popmail dropserver <host>

Toglie <host> dalla lista dei pop server da interrogare. Tutti i riferimenti alla voce saranno eliminati dal sistema .

3.70.3. popmail kick <host>

Inizia subito una connessione POP con l'host per ritirare la posta.
Questo comando e' necessario quando nessun intervallo viene
specificato con il comando "popmail addserver" .

3.70.4. popmail list

Mostra la tabella dei server popmail corrente.

3.70.5. popmail quiet <yes|no>

Mostra o fissa la notifica di "new mail arriverd for ..." (nuova posta in arrivo per ... ) via pop.

3.70.6. popmail trace <level>

Mostra o fissa il livello di tracciamento delle sessioni pop. I livelli di trace correnti sono:

0 - Tracing disattivato

1 - Riferisce sugli errori seri
2 - Riferisce sugli errori transienti
3 - Riferisce sulla progressione della sessione.

Da notare che il tracciamento e' diretto solo nel file di log.

$$ppp

3.66. ppp <sottocomandi>

I seguenti comandi vengono usati per le interfacce Point to Point Protocol.

Questa implementazione di PPP e' stata progettata per essere la piu' completa possibile . Percio' il numero delle opzioni potra' sembrare molto alto tanto da scoraggiarne l'impiego. Ma a tale scopo viene fornita qualche facilitazione completa di qualche esempio. Comunque, una tipica configurazione PPP puo' includere i seguenti comandi:

```
attach asy 0x3f8 4 ppp linea1 4096 1500 9600
dial linea1 dialer.pp0 30 3 <hostid>
            #
ppp pp0 quick
ppp pp0 lcp open
            #
route add default linea1
```

3.71.1. ppp <iface>

Mostra lo stato dell'interfaccia PPP.

3.71.2. ppp <iface> lcp ...

Questi comandi vengono usati per la configurazione di LCP (Link Control Protocol) .

3.71.2.1. ppp <iface> lcp close

Disattiva l'interfaccia PPP.

3.71.2.2. ppp <iface> lcp local ...

Questi comandi controllano il lato locale del collegamento. Se viene specificata un'opzione, i parametri saranno usati come valori iniziali nelle richieste di configurazione. Come valore predefinito , tutte le opzioni ono concesse.

3.71.2.2.1. ppp <iface> lcp local accm [<bitmap> | allow [on | off]]

Mostra o fissa la Mappa dei Caretteri di Controllo Asincroni (Accm). Il valore predefinito e' 0xffffffff.

3.71.2.2.2 ppp <iface> lcp local authenticate [pap | none | allow [on | off]]

Mostra o fissa il protocollo di autenticazione. Il valore predefinito e' nullo (none).

3.71.2.2.3 ppp <iface> lcp local compress address/control [on | off | allow [on | off]]

Mostra o fissa l' opzione per comprimere l'indirizzo ed i campi di controllo della testata PPP (la quale e' simile a HDLC). Cio' e' generalmente hdesiderabile per collegamenti asincoroni lenti, e non desiderabili per collegamenti sincroni veloci. Il valore predefinito e' off.

3.71.2.2.4. ppp <iface> lcp local compress protocol [on | off | allow [on | off]]

Mostra o fissa l'opzione per comprimere il campo protocollo della testata PPP simile a HDLC. Cio' e' generalmente desiderabile per collegamenti asincoroni lenti, e non desiderabili per collegamenti sincroni veloci. Il valore predefinito e' off.

3.71.2.2.5 ppp <iface> lcp local magic [on | off | <value> | allow [on |off]]

Mostra o fissa il Numero Magico iniziale. Il valore predefinito e' zero.

3.71.2.2.6 ppp <iface> lcp local mru [<size> | allow [on |off]]

Mostra o fissa la Massima Unita' di Ricezione (MRU). Il valore predefinito e' 1500.

3.71.2.2.7 ppp <iface> lcp local default

Riporta le opzioni al loro valore iniziale.

3.71.2.3 ppp <iface> lcp [open | listen]

Attende che il livello fisico divenga attivo e se viene dato "open" inizia la negoziazione della configurazione. Se "listen", attende per la negoziazionedella configurazione dal remoto.

3.71.2.4 ppp <iface> lcp remote ...

Questi comandi controllano la configurazione del lato remoto del collegamento. Le opzione sono identiche a quelle del lato locale. Se viene specificata una opzione, i parametri saranno usati in risposta alle richieste di configurazione del remoto. Se non specificato, quell'opzione sara' accettata , se concesso.

Per ogniuna di queste opzioni, il parametro allow permettera' al remoto di specificare quell'opzione nella sua richiesta. Come valore predefinito , tutte le opzioni sono permesse.

3.71.2.5. ppp <iface> lcp timeout [<seconds>]

Mostra o fissa l'intervallo da attendere tra i tentativi di configurazione o termine. Il valore predefinito e' 3 secondi.

3.71.2.6. ppp <iface> lcp try ...

Questi comandi vengono usati per i vari contatori.

3.71.2.6.1. ppp <iface> lcp try configure [<count>]

Mostra o fissa il numero di richieste di configurazioni da inviare. Il valore predefinito e' 10.

3.66.2.6.2. ppp <iface> lcp try failure [<count>]

Mostra o fissa il numero di richieste di "cattiva configurazione" concesse al remoto. Il valore predefinito e' 5.

3.71.2.6.3. ppp <iface> lcp try terminate [<count>]

Mostra o fissa il numero di richieste di fine collegamento prima dello shutdown (termine). Il valore predefinito e' 2.

3.71.3. ppp <iface> ipcp ...

Questi comandi vengono usati per la configurazione di (Internet Protocol Control Protocol) IPCP. I sottocomandi close, open, timeout e try sono identici a quelli di LCP (descritti poco sopra).

3.71.3.1. ppp <iface> ipcp local ...

Questi comandi controllano la configurazione del lato locale del collegamento. Se viene specificata una opzione, i parametri saranno usati in risposta alle richieste di configurazione . Se non specificato, quell'opzione non sara' richiesta.

Per ogniuna di queste opzioni, il parametro allow permettera' al remoto di includere quell'opzione nella sua risposta, anche quando l'opzione non e' inclusa nella richiesta. Come valore predefinito , tutte le opzioni sono permesse.

3.71.3.1.1. ppp <iface> ipcp local address [<hostid> | allow [on |off]]

Mostra o fissa l'indirizzo locale per scopi di negoziazione. Se verra' specificato un indirizzo uguale a 0, allora l'altro capo del collegamento fornira' l'indirizzo. Il valore predefinito e': "nessun indirizzo sara' negoziato".

3.71.3.1.2. ppp <iface> ipcp local compress [tcp <slots> [<flag>] | none | allow [on | off]]

Mostra o fissa il protocollo di compressione. Il valore predefinito e' none.

tcp <slots> specifica il numero di slots di " conversazione" che deve essere nel campo da 1 a 255 (cio' puo' essere limitato al momento della compilazione ad un numero piu' piccolo). Una buona scelta e' nel campo da 4 a 16.

tcp <flag> e' 0 (non comprime il numero di slot) oppure 1 ( va bene per comprimere il numero di slot). Il Nos di ka9q puo' gestire numeri di slots compressi, percio' il valore predefinito e' 1.

3.71.3.2. ppp <iface> ipcp remote ...

Questi comandi controllano la configurazione del lato remoto del collegamento. Le opzioni sono identiche a quelle del lato locale. Se viene specificata una opzione, i parametri saranno usati in risposta alle richieste di configurazione del remoto. Se non specificato, quell'opzione sara' accettata , se concesso.

3.71.4. ppp <iface> pap ...

Questi comandi vengono usati per la configurazione del PAP (Password Authentication Protocol) .

I sottocomandi timeout e try sono identici ad LCP descritto prima.

Comunque il contatore di termine non viene utilizzato.

3.71.4.1. ppp <iface> pap user [<username> [<password>]]

Mostra o fissa lo <username> ( la password puo ' essere installata, ma non visualizzata). Quando viene specificato lo <username>, ma non viene fornita alcuna password, viene cercato il file ftpusers . Quando la coppia username/password e' sconosciuta o rifiutata, la sessione apparira' alla console richiedendo username/password.

3.71.5. ppp <iface> quick

Nonostante non siano necessari valori diversi da quelli predefiniti ,questo comando risulta molto utile per l"inizializzazione veloce del linkcon valori raccomandabili.In seguito alla popolare richiesta di semplificazione della procedura di attivazione di un link ppp , questo comando rappresenta una scorciatoiaper i seguenti comandi:

ppp pp0 ipcp local compress tcp 16 1
ppp pp0 ipcp open
ppp pp0 lcp local accm 0
ppp pp0 lcp local acfc on
ppp pp0 lcp local pfc on
ppp pp0 lcp local magic on

3.71.6. ppp <iface> trace [<flags>]

Mostra o fissa i segnali (flags) che controllano la fase di logging durante la configurazione del collegamento PPP.

Il valore del segnale e' 0 per none, 1 per basic, 2 per general. Valori piu' grandi di 2 non vengono generalmente compilati, e vengono descritti nei files sorgenti specifici dove essi vengono definiti.

Da notare che nel caso di attivazione del tracing , l'output andra' nel filespecificato dal comando NOS "log" e quindi mai sul video. E a tal propositosi consiglia di tenere d'occhio il file di log poiche' ppp scrive molte informazioni se attivato il trace con segnale 2.

3.71.7. Modificato il codice "ppp" affinche' aggiunga la rotta per l'hostcorrispondente e , se attivato il server rip, aggiunga anche i comandi:

rip accept <corrispondente>

rip request <corrispondente>
rip add <corrispondente> 200 1

Tale modifica consente a due stazioni dotate della corrente versione di NOSdi effettuare il collegamento in modo completamente automatico. Inoltre in caso le due stazioni siano due routers e' possibile avvalersi del rip pergli instradamenti automatici.

$$ps

3.72. ps

Mostra tutti i processi correnti nel sistema. I campi sono i seguenti :

PID - Identificatore di Processo (indirizzo del descrittore del processo).

SP - Il valore corrente del puntatore del processo nello stack.

stksize - La grandezza dello stack allocato al processo.

maxstk - Il picco apparente di utilizzazione nello stack di

questo processo. Questo viene fatto in uno stile piuttosto euristico

percio' i numeri dovrebbero essere considerati in modo approssimativo. Se questo numero raggiunge o eccede la cifra di "stksize", il sistema quasi certamente andra' allo sfascio; percio' il programma dovrebbe essere ricompilato per dare ai processi una allocazione maggiore all'avvio (startup).

event - L'evento che questo task sta' aspettando, se non e' attivabile.

fl - Flags di stato dei processi. Ce ne sono tre : " I" (interrupts abilitati), "W" (in attesa di evento) ed "S" (sospeso-non correntemente usato). Il segnale I viene attivato ogni qualvolta un task ha eseguito una chiamata "pwait ()" (aspetta un evento) senza prima disabilitare gli interrupts

hardware. Solo i tasks che attendono gli eventi di interrupts hardware disabiliteranno questo segnale; questo viene fatto per evitare sezioni critiche ed interrupts mancati. Il segnale W indica che il processo sta aspettando un evento; la colonna event non sara' vuota . Da notare che sebbene ci possano essere parecchi processi attivi alla volta (mostrati nella lista ps come quelli senza il flag W e con i campi event vuoti ) solo un processo e' effettivamente attivo in ogni istante (The Refrigerator Light Effect dice che il comando ps e' sempre quello attivo quando viene generata questa schermata ).

$$pwd

3.73. pwd [<dirname>]

Un sinonimo per il comando cd.

$$rarp

3.74. rarp <sottocomandi>

Questo comando usa il protocollo di risoluzione inversa RARP.

3.74.1. rarp query <iface> <callsign>

Questo comando avvia una richiesta di risoluzione inversa via iface per trovare l'indirizzo IP del callsign. Esso conta alla rovescia per 10 secondi prima di cessare di attendere una risposta.

$$record

3.75. record [<filename>|off]

Appende al filename tutti i dati ricevuti sulla sessione corrente. I dati inviati sulla sessione corrente vengono anche scritti nel file ad eccezione delle sessioni Telnet nel modo eco remoto. Il comando record off ferma la registrazione e chiude il file.

$$remote

3.76. remote [-p <port>] [-k <chiave>] [-a <kickaddr>] <hostid> hexit|reset|kick

Invia un pacchetto UDP all'host specificato comandandogli di uscire dal programma nos, azzerare il processore, oppure forzare una trasmissione su connessioni TCP. Perche' questo comando venga accettato, il sistema remoto deve aver attivato il server remote ed il numero della porta specificato nel comando remote deve coincidere con il numero della porta dato quando e' stato attivato sul sistema remoto. Se i numeri di porta non corrispondono o se il server remote non e' attivo, sul sistema destinatario, il pacchetto di comando viene ignorato. Anche se il comando viene accettato dal server non viene data alcuna conferma.

Il sottocomando "kick" forza una ritrasmissione per timeout su tutte le connessioni che il nodo remoto puo' avere con il nodo locale incluso quelle SMTP , se ci sono messaggi che attendono la trasmissione . Se viene usato l'opzione "-a" vengono forzate , invece, le connessioni all'host specificato. Nessuna parola chiave viene richiesta per il sottocomando kick.

I sottocomandi "exit" e "reset" sono principalmente utili per riavviare il programma nos su di un sistema incustodito dopo che il file di configurazione e' stato aggiornato. Il sistema remoto dovrebbe richiamare il programma nos automaticamente al boot, preferibilmente in un loop infinito. Per esempio, sotto MS-DOS il disco di boot dovrebbe contenere le seguenti linee in autoexec.bat :

```
:loop
```

nos
goto :loop

3.76.1. remote -s <key>

I sottocomandi “exit” e “reset” di remote richiedono una parola chiave. Questa viene fissata su di un dato sistema con l’opzione -s, e viene specificata in un comando al sistema remoto con l’opzione -k. Se non viene fissata nessuna parola chiave con l’opzione -s, allora i sottocomandi “exit” e “reset” vengono disabilitati. Da notare che remote e’ una caratteristica sperimentale nel NOS e non e’ ancora supportato da nessun altra implementazione TCP/IP.

$$rename

3.77. rename <file1> <file2>

Rinomina il file1 in file2

$$reset

3.78. reset [<sessione>]

Azzera la sessione specificata; se non viene dato argomento, azzera la sessione corrente. Questo comando dovrebbe essere usato con cautela poiche’ non informa in modo affidabile il capo remoto che la connessione non esiste piu’ (in TCP un messaggio di “reset” (RST) verra’ generato automaticamente in modo tale che il corripondente non invii nient’altro dopo cio’. In AX.25 il messaggio DM fa un’ azione simile. Entrambi vengono husati per sbarazzarsi di una connessione lenta, semi-aperta, dopo che un sistema remoto si e’ “inchiodato” ).

$$rip

3.74. rip sottocomandi

Questi comandi vengono usati per il servizio RIP.

3.79.1. rip accept <gateway>

Rimuove il gateway specificato dalla tabella di filtro RIP, in modo da accettare futuri broadcasts da quel gateway.

3.79.2. rip add <hostid> <intervallo> [<flags>]

Aggiunge una voce alla tabella di emissioni broadcast RIP. La tabella di instradamento IP verra’ inviata a <hostid> ogni <intervallo> secondi. Seviene specificato un <flags> di 1, allora verra’ attuato il processo “split horizon” per questa destinazione. Cioe’ qualsiasi voce della tabella di instradamento IP che punta verso l’ interfaccia che sara’ usata per inviare questo aggiornamento sara’ rimossa dall’aggiornamento stesso.Se non viene specificato il processo “split horizon”, allora tutte le vocidella tabella di instradamento eccetto quelle marcate “private” vengono inviate in ogni aggiornamento (le rotte private non vengono mai inviate nei pacchetti RIP).

Vengono sempre effettuati aggiornamenti “provocati”. Cioe’ qualsiasi cambiamento nella tabella di instradamento che causa una destinazione precedentemente raggiungibile divenire irraggiungibile , provoca un aggiornamento che pubblicizzera’ la destinazione con “metric 15”, cosi definita per significare “infinita”.

Da notare che affinche’ i pacchetti RIP vengano emessi in modo appropriato all’indirizzo broadcast, devono esistere gli instradamenti IP corretti e le voci nella tabella ARP; questi prima

guideranno il broadcast all'interfaccia corretta, e poi porranno il corretto indirizzo broadcast del livello link nel campo destinazione relativo. Se viene usata la convenzione standard degli indirizzi IP broadcast (ad es.:128.96.0.0 oppure 128.96.255.255) allora c'e' la possibilita' che si abbia gia' la voce necessaria nella tabella di instradamento IP; ma reti fuori dalla norma o reti con indirizzamento cluster possono richiedere speciali accorgimenti. Comunque, sara' necessario un comando arp add per tradurre tale indirizzo all'appropriato indirizzo broadcast del livello link, es.:

arp add 151.90.1.255 ethernet ff:ff:ff:ff:ff:ff

per una rete ethernet, e

arp add 44.134.176.255 ax25 qst-0

per canale packet radio ax25. In presenza di interfacce ax25 multiple, si crei un indirizzo unico per ognuna.

Inoltre si dovra' fissare l'indirizzo broadcast con il comando ifconfig, per es.:

ifconfig ether broadcast 151.90.1.255

3.79.3. rip drop <dest>

Rimuove una voce dalla tabella broadcast RIP

3.79.4. rip merge [on|off]

Questo segnale controlla una caratteristica sperimentale per il consolidamento delle voci ridondanti nella tabella di instradamento IP. Quando il "rip merging" e' abilitato, viene analizzata la tabella dopo aver processato ogni aggiornamento RIP. Una voce e' considerata ridondante se i(l) destinatari(o) che essa copre verrebbe(ro) instradato(i) in modo identico da una voce meno specifica gia' in tabella.

Cioe' gli indirizzi destinazione specificati dalla voce in questione devono anche combaciare con gli indirizzi destinazione di voci meno specifiche e le due voci devono avere gli stessi campi interfaccia e gateway. Per esempio, se la tabella di instradamento contiene

```
Dest          Len Interface    Gateway       Metric  P Timer  Use
1.2.3.4       32  ethernet0    128.96.1.2    1       0 0      0
1.2.3          24  ethernet0    128.96.1.2    1       0 0      0
```

allora la prima voce dovrebbe essere cancellata come ridondante poiche' i pacchetti inviati a 1.2.3.4 saranno instradati correttamente anche dalla seconda voce. Da notare che i relativi dati metrici delle voci vengono ignorati.

3.79.5. rip refuse <gateway>

Rifiuta di accettare gli aggiornamenti RIP dai gateways specificati aggiungendoli nella tabella di filtro RIP. Puo' essere successivamente rimosso con il comando rip accept.

3.79.6. rip request <gateway>

Invia un pacchetto di richiesta al gateway specificato, portandolo a rispondere con un pacchetto di risposta RIP contenente la sua tabella di instradamento.

3.79.7. rip status

Mostra lo stato RIP, includendo una somma del numero dei pacchetti inviati e ricevuti, il numero delle richieste e delle risposte, il numero dei tipi di pacchetti RIP sconosciuti, ed il numero degli aggiornamenti RIP rifiutati dagli hosts nella tabella filtro. Viene anche mostrato un elenco degli indirizzi e degli intervalli ai quali vengono inviati periodici aggiornamenti RIP, insieme con il contenuto della tabella filtro.

3.79.8. rip trace [0|1|2]

Questa variabile controlla la visualizzazione dei pacchetti RIP in arrivo ed in uscita. Mettendola a “0” disabilita tutte le visualizzazioni RIP. Il hvalore di “1” porta alla visuavisualizzazione dei cambiamenti della tabella di instradamento , mentre i pacchetti che non causano cambiamenti non vengono mostrati. Mettendo la variabile a “2” produrra’ il massimo output, inclusa la visualizzazione dei pacchetti che non causano cambiamenti nella tabella di instradamento.

3.79.9. rip ttl [<seconds>]

Mostra o fissa il tempo di sopravvivenza (Time-To-Live) RIP a <seconds>. Un valore normale di tempo massimo (timeouts) e’ 240 secondi. Quest’ultimo non e’ il ttl contenuto in un rip broadcast (16 = infinito). Si fissi questo valore prima di attivare rip e lo si cambi solo in cooperazione con i nodi adiacenti. Il valore predefinito e’ 240 secondi.

$$rlogin

3.80. rlogin host

Realizza una sessione rlogin attreverso il socket 513 di una workstation *NIX (UNIX , XENIX , ecc..) compatibile . Il terminale predefinito e’ un terminale compatibile ansi (come definitao con fkeys). Il nome dell’utente predefinito e’”guest” (ridefinibile attraverso la variabile DOS di ambiente “USER”).

$$rmdir

3.81. rmdir <dirname>

Rimuove il sottodirettorio specificato, se risulta vuoto.

$$route

3.82. route

Senza argomenti, route mostra la tabella di instradamento IP.

3.82.1. route add <dest_hostid> [/bits] | default <iface>
[<gateway_hostid> | direct [<metric>]]

Questo comando aggiunge una voce alla tabella di instradamento. Esso necessita di almeno due argomenti ulteriori, l’host_id della destinazione finale ed il nome dell’interfaccia alla quale verrebbero inviati i pacchetti. Se la destinazione non e’ locale, dovrebbe essere specificato anche host_id del gateway (se l’interfaccia e’ una connessione punto_a_punto,allora gateway_hostid puo’ essere omesso anche se il destinatario non e’ locale perche’ questo campo viene solamente utilizzato per determinare l’indirizzo del livello link (2) del gateway. Se la destinazione e’ direttamente raggiungibile , il gateway_hostid non e’ necessario poiche’ l’indirizzo di destinazione viene utilizzato per determinare l’indirizzo dell’interfaccia link). Se viene usato rspf ed il sistema e’ un router, per moltiplicare le rotte puo’ essere utilizzata la parola chiave direct oppure gateway_hostid per fissare il valore di metric piu’ grande del valore 1 predefinito. In questa maniera le rotte

pubblicizzate da altre stazioni rspf possono essere meno costose ed essere quindi selezionate. Se viene dato direct , ma non metric , viene usato un nuovo algoritmo per fissare il valore metric dipendente dal numero di bits della maschera di sottorete.Il suffisso opzionale /bits all'hostid destinazione specifica quanti bits fondamentali nell'host_id devono essere considerati significativi nei confronti degli instradamenti. Se non viene specificato, vengono assunti 32 bits (significato pieno). Con questa opzione, una singola voce della tabella di instradamento puo' fare riferimento a diversi hosts tutti aventi in comune un bit di prefisso di stringa nei loro indirizzi IP. Per esempio, le reti ARPA di classe A, B e C potrebbero utilizzare i suffissi di /8, /16 e /24 rispettivamente;

il comando :

route add 151/8 linea1 151.90.220.50

porta qualsiasi indirizzo IP iniziante con "151" nei primi 8 bits ad essere instradato verso 151.90.220.50; i restanti 24 bits non sono significativi.

Quando un indirizzo IP da instradare combacia con piu' di una voce nella tabella di instradamento, viene utilizzata la voce con il parametro con maggior numero di bits (il confronto migliore). Questo permette che hosts individuali o blocchi di hosts siano eccezioni per una regola piu' generale riguardanti blocchi di hosts piu' grandi.

La destinazione speciale default viene utilizzata per instradare i datagrammi ad indirizzi che non sono presenti nella tabella di instradamento; questo equivale a specificare un suffisso /bits di /O per ogni destinazione hostid. Bisogna fare attenzione con le voci default poiche'due nodi con voci default indicanti l'uno verso l'altro instraderanno pacchetti verso indirizzi sconosciuti avanti ed indietro in un loop finche' iloro campi time_to_live (TTL) scadranno (i loops di instradamento per indirizzi specifici possono anche essere creati, ma questo e' meno probabile che accada accidentalmente).

Ci sono due interfacce entrocontenute : loopback ed encap.

Loopback e' solo per scopi interni. La encap e' un'encapsulatore d'interfaccia IP. Questo viene usato per incapsulare un datagramma completo IP in un altro datagramma IP cosicche' questo venga "portato in spalla". Esso viene spesso usato per portare datagrammi ampr.org (rete 44) attraverso l'Internet. Da notare che l'Internet e' completamente interconnessa mentre la rete ampr.org e' alquanto dispersiva . In questo modo, per esempio, due localita' possono scambiarsi l'un l'altro datagrammi della rete 44 . Alcune note aggiuntive: Si supponga che un gateway Internet abbia 2 indirizzi IP: uno sulla rete 44 ampr.org e uno sull'Internet. Si dovrebbe assicurare che l'interfaccia connessa sull'Internet abbia dato il comando "ifconfig ipaddr" in modo corretto. Da notare che questo gateway opera solo come gateway per altre stazioni. Ci sono state alcune supposizioni nel prendere un indirizzo IP quando l'interfaccia encap viene usata localmente. La supposizione prende la peggiore delle ipotesi (sempre errata con un fattore di Murphy di 2.7). Il codice ora impiega il numero IP locale come origine quando la rotta parte presso la stazione locale. Se cio' non e' quello che si desiderava, esso si puo' annullare fissando l'indirizzo IP della pseudo-interfaccia encap a cio' che si vuole che sia.

Ancora altre note su encap. Si supponga di avere 3 sistemi su di
una rete ethernet con la rete 129.179.122.128/25. In un altro luogo si
abbia un'altra hrete collegata all'Internet. Gli indirizzi sono nel campo
129.179.122.0/25. Adesso si supponga di avere un collegamento radio con gli indirizzi 44.137.0.2 e 44.137.1.2 sulle rispettive localita'. Sul sistema 44.137.0.2 si abbia : route add 44.0.0.0/8 encap 44.137.1.1 99 .
Sul sistema successivo sulla rete locale ethernet si abbia 44.137.0.1 /

129.179.122.129. Per arrivare da quel sistema , diciamo 44.62.0.1 si dovra' aggiungere un encap presso il gateway locale con il comando: route add 44.0.0.0/8 encap 129.179.122.130 . Il comando route add default ether 129.179.122.130 dara' accesso all'Internet. Altrimenti esso osservera' l'indirizzo per l'interfaccia da usare per raggiungere 129.179.122.130 e usera' 129.179.122.129. Ora 44.62.0.1non sapra' MAI da dove esso provenga. Percio' l'aggiunta di encap sul secondo sistema risolve il problema.

Il parametro <metric> e' un'indicazione della qualita' della rotta data. La migliore sara' 1 e 16 significhera' infinita o destinazione irraggiungibile. Un instradamento aggiunto manualmente ottiene un metric di default di 1, mentre le rotte ottenute via broadcast RIP ottengono un metric di 2 o piu', dipende da quanti gateways ci sono per una data destinazione.

Questi sono alcuni esempi dei comandi route:

#Instrada datagrammi all'indirizzo IP 44.0.0.3 verso la l'interfaccia sl0.
#Non e'necessario nessun gateway perche' SLIP e' punto_a_punto.
route add default 44.0.0.3 sl0
#Instrada tutto il traffico di default verso il gateway sulla rete
#ethernet locale con indirizzo IP 44.0.0.1
route add default ec0 44.0.0.1
#L 'ethernet locale ha un assegnazione di indirizzo ARPA Class-C;
#instrada tutti gli indirizzi IP inizianti con 192.4.8 verso di essa. route add 192.4.8/24 ec0 #La stazione con indirizzo di IP 44.0.0.10 e' sul canale locale AX.25 route add 44.0.0.10 ax0.

# Un collegamento via encap verso 192.4.8.12 dove la sottorete

```
# 44.64.0.0 sia accessibile . L'Internet non sapra' da dove la   rete
# 44 provienga sebbene la si utilizzi con cio' che e' noto   ad essa.
```

route add 44.64.0.0/16 encap 192.4.8.12 4

3.82.2. route addprivate <dest hostid>[/bits]|default <iface> [<gateway hostid> [<metric>]].

Questo comando e' identico a route add eccetto che marca la nuova voce come privata; questa non verra' mai inclusa nelle emissioni per gli aggiornamenti RIP .

3.82.3. route drop <dest hostid>

route drop cancella una voce dalla tabella. Se arriva un pacchetto per l'indirizzo cancellato ed e' operativo un instradamento di default, verra' utilizzato quest'ultimo.

3.82.4 route lookup <dest>

Mostra la voce d'instradamento per una data destinazione. Se non esistono rotte la destinazione non e' raggiungibile.

$$rspf

3.83. rspf sottocomandi

RSPF e' il protocollo Radio Shortest Path First. Ogni stazione ascolta i messaggi RRH (Router To Router Hello). Quando tale messaggio RRH viene ricevuto, il Nos calcolera' se il collegamento e' bidirezionale effettaundo alcuni ping all'altra stazione. Il protocollo e' descritto nelle specifiche RSPF 2.1.

3.83.1. rspf interface <interface> <quality> <horizon>

<interface> e' l'interfaccia che rspf dovrebbe usare. <quality> e' nel campo da 1 a 127, <horizon> e' tra 1 e 255. I nodi dovrebbero avere la qualita' fissata a 1. I nodi adiacenti dovrebbero avere la qualita' fissata a 8. Il valore normalmente usato per <horizon> e' 32.

3.83.2. rspf mode [vc | datagram | none]

Senza argomenti, mostra il modo preferito da RSPF. I modi sono VC (Virtual Circuit ) e Datagram. none riporta al modo preferito.

3.83.3. rspf rrhtimer [seconds]

Senza argomenti, mostra il valore del timer rrh.

3.83.4. rspf suspecttimer [seconds]

Senza argomenti, mostra il valore del timer suspect .

3.83.5. rspf timer [seconds]

Senza argomenti, mostra il valore del timer update .

Per attivare RSPF, effettuare le seguenti operazioni. Fissare l'indirizzo broadcast corretto per l'interfaccia destinazione, per es.: ax0.

ifconfig ax0 broadcast 44.134.176.255

Questo automaticamente crea una voce nella tabella delle rotte per 44.134.176.255 . Se si intende usare RSPF su piu' di una interfaccia, queste dovranno avere un'indirizzo broadcast differente. Anche le voci di instradamento in tabella verranno sovrascritte dalle definizioni successive.

Si configuri ax0 come interfaccia RSPF con <horizon> 32 e

<quality> a 1 (salti). Situazione tipica di un nodo terminale. Si sostituisca 1 con 8 per i nodi adiacenti.

rspf interface ax0 1 32

Si installi l'intervallo tra messaggi RRH.

rspf rrhtimer 900

Si definisca quanto e' necessario perche' un collegamento sia definito sospetto.

rspf suspecttimer 2000

Si fissi l'intervallo tra aggiornamenti successivi.

rspf timer 900

$$sccstat

3.84. sccstat

Mostra le statistiche del drivers scc di PE1CHL.

$$session

3.85. session [<session #>] [flowmode [on | off]]

Senza argomenti, mostra l'elenco delle sessioni correnti, incluso il numero di sessione, l'indirizzo del TCP remoto o AX.25 e l'indirizzo del blocco di controllo TCP o AX.25. Un asterisco (*) e' posto vicino alla sessione corrente; immettendo un carriage return a questo punto vi pone in modo conversazione con quella sessione. Inserendo un numero di sessione

come argomento per il comando vi porra' in modo conversazione con quella sessione. Se il server Telnet e' abilitato, l'utente viene avvisato dell'arrivo di una richiesta e viene cosi' automaticamente assegnato un numero di sessione. L'utente puo' quindi selezionare normalmente la sessione per conversare con l'utente remoto come se fosse stata iniziata localmente . Aggiungendo il sottocomando flowmode o si abilita/disabilita l'attivazione del output paginato con **more** per quella sessione. A tale scopo sara' necessario tornare al modo comandi , prima di dare il comando dir al server ( per es.: ftp) , e immettere "session # flowmode on" per avere un output paginato. Quando lo si desidera si potra' poi tornare al modo comandi e tornare in modalita flowmode off. Da notare che le sessioni ftp hanno gia' il loro comando flow entrocontenuto. Si veda il comando FTP a tale scopo piu' avanti .

$$shell

3.86. shell

Esegue una sottoshell, che normalmente e' COMMAND.COM sotto MS-DOS, ma potrebbe essere qualsiasi altra cosa specificato dalla variabile d'ambiente COMSPEC. Se si vuole eseguire un comando DOS, (es. passando argomenti al COMMAND.COM) si deve battere /c come primo argomento. Se non c'e' memoria a sufficienza per il COMMAND.COM o altri programmi, come indicato da "coreleft" nella visualizzazione dello stato della memoria (memory status) , il comando shell non avra' alcun effetto. Quando c'e' attivita' in background, del tipo server ftp o altro, questa viene sospesa mentre si esegue la sottoshell dos. Il messaggio di "operatore assente " verra' inviato se verra' scelta l'opzione Operator dal Mailbox locale e hquindi sara' rifiutata la connessione al server ttylink locale.

Affinche' si possa godere del multitasking mentre si esegue la sottoshell del dos , e' necessario dare il comando "multitask on". Il NOS rimarra' dunque residente e continuera' a girare. Questa e' un'importante caratteristica ed avra' futuri sviluppi, poiche' sara' possibile a programmi esterni usare risorse NOS.

Per esempio, se si vuole copiare un file si digiti:

shell /c copy <file1> <file2>

se si vuole vedere il contenuto di un archivio, si potra' digitare:

shell /c pkzip -v archive.zip

$$skick

3.87. skick #socket

Questo e' un modo breve per accedere ai vari sottocomandi kick. Esso cerca il socket per il tipo corretto e ed effettua il kick al livello trasporto.

$$smtp

3.88. smtp.

Questi comandi vengono usati per il servizio Simple Mail Transfer Protocol (cioe' per la posta elettronica).

3.88.1. smtp batch [yes | no]

Se esso e' yes smtp dara' i comandi in un solo datagramma . Se esso e' no solo un comando alla volta viene inviato al server ed una sola risposta viene atteso. Alcuni vecchi e poco flessibili server non accettano di gestire piu' di un comando alla volta. Nos puo' gestire piu'

comandi alla volta. Se non si hanno problemi di collegamento con vecchi server, si metta l'opzione batch attiva poiche' salvera' larghezza di banda.

3.88.2. smtp gateway [<hostid>]

Mostra o fissa l'host da usare come "intelligente" ripetitore per la posta. Tutta la posta inviata ad un host non presente nella tabella delle rotte, verra' istradata invece ad un gateway per il successivo inoltro.

3.88.3. smtp kick

Esamina la posta da evadere in coda e tenta di consegnare ogni messaggio in sospeso. Questo comando viene periodicamente richiamato da un smtp timer mentre il nos sta operando; In definitiva "smtp kick" consente quindi all'utente di azzerare il timer e attivare subito il cliente smtp .

3.88.4. smtp kill <jobid>

Rimuove un messaggio SMTP (job) dalla coda di invio perche'
esso non venga piu' consegnato.

3.84.5. smtp list

Mostra la coda di messaggi corrente. Se e' presente una "L" in prima colonna, significa che quel job (messaggio) e' bloccato da smtp, cioe' in processo. E' consigliabile aggiungere nel file autoexec.bat un comando "del ~/spool/mqueue/*.lck" allo scopo di rimuovere il blocco e ripristinare la coda di invio alla successiva attivazione del nos.

3.88.6. smtp maxclients [<count>]

Mostra o fissa il massimo numero di sessioni SMTP simultanee in uscita ammissibili. Il valore predefinito e' 10; lo si riduca se la congestione della rete e' un problema.

3.88.7. smtp mode [queue |route]

Fissa il modo di consegna di smtp. Se il modo e' queue, tutti i messaggi vengono lasciati in ~/spool/rqueue per la gestione e l'inoltro esterno. Se il modo e' route, i messaggi vengono gestiti, se destinati per l'host locale , appesi al file di mailbox, se destinati a host remoti, verranno inoltrati.

3.88.8. smtp mxlookup [yes |no]

Mostra o attiva il segnale che abilita o disabilita la consultazione dei record MX (Mail Exchanger) .Questo puo' essere abilitato se un server di dominio e' disponibile nelle vicinanze (e quindi raggiungibile). Dovrebbe invece essere disabilitato (valore predefinito) se il server di dominio non e' in grado di soddisfare la richiesta per il record MX. Da notare che la gestione del record MX e' molto limitata nel nos. Se si riceve una risposta dal server di dominio , essa verra' presa per la destinazione.

3.88.9. smtp quiet [yes |no]

Abilita o disabilita il messaggio che e' arrivata nuova posta presso il sistema locale.

3.88.10. smtp timer [<val>]

Mostra o fissa l'intervallo, in secondi , che intercorre tra due esplorazioninella coda della posta da evadere. Per esempio smtp timer 600 portera' il sistema a controllare la posta in partenza ogni 10 minuti e tentare di recapitare ogni cosa esso trovi, soggetto naturalmente al limite smtp maxclients. Fissando come valore zero si disabilita la scansione della coda,si noti che questo e' il valore predefinito!. Questo valore viene raccomandato per gateways IP isolati che non trattano mai posta, poiche' dotati solo di drivers per floppy disk.

3.88.11. smtp trace [<value>]

Mostra o fissa il segnale che regola la visualizzazione del colloquio tra client e server SMTP. Esso consente quindi di osservare la conversazione del proprio clinete SMTP mano a mano che recapita la posta. Lo zero (valore predefinito) disabilita la visualizzazione.

$$socket

3.89. socket [[<socket #>] [flowmode [yes|no]]]

Senza argomenti mostra tutte le porte logiche ( sockets) attive ,
dandone l'indice ed il tipo , l'indirizzo del blocco di controllo del
protocollo associato, il possessore del Processo ID ed il nome. Se viene
fornito l'indice di un socket attivo, viene richiamata la visualizzazione
dello statoper l'appropriato protocollo . Per esempio, se il socket si
riferisce ad una connessione TCP, la visualizzazione sara' quella data dal comando tcp status con l'indirizzo del blocco di controllo del protocollo.

Aggiungendo il sottocomando flowmode si abilita/disabilita l'attivazione del output paginato con -more- per quella sessione. Questo comando torna utile per esempio quando viene mostrato un direttorio pieno di files in ftp. A tale scopo sara' necessario tornare al modo comandi , prima di dare il comando dir al server ( per es.: ftp) , e immettere "socket # flowmode on" per avere un output paginato. Quando lo si desidera si potra' poi tornare al modo comandi e tornare in modalita "flowmode off".

$$source

3.90. source <filename>

Esegue una sequenza di comandi da <file>, come se fosse digitato alla tastiera. Quindi riprende a leggere i comandi dal flusso precedente. Questo puo' essere utile per tenere per esempio le dichiarazioni di rotte in un file separato, che altrimenti potrebbe trovarsi in qualche punto nel autoexec.nos.

$$start

3.91. start ax25 | convers | discard | echo | finger | ftp | lpd | netrom
| nntp | pop | pop2 | pop3 | remote | rip | smtp | telnet | tip | ttylink

Avvia il server Internet specificato , accettando cosi' richieste successive di connessioni da remoto.

$$status

3.92. status

Mostra le informazioni caricate dal nos. Quando e' stato avviato, da quanto tempo e' in attivita', files aperti, messaggio corrente del giorno, ecc.

Da notare che sara' necessario dare il comando DOS "setver
nos.exe 5.0" per poter vedere lo stato dei files aperti.

$$stop

3.93. stop ax25 | convers | discard | echo | finger | ftp | lpd | netrom | nntp | pop | pop2 | pop3 | remote | rip | smtp | telnet | tip | ttylink

Ferma il server Internet specificato , trascurando ogni ulteriore richiesta di connessione remota. Le connessioni gia' esistenti termineranno normalmente.

$$tcp

3.94. tcp <sottocomandi>

Questi comandi vengono usati per il servizio Trasmission Control Protocol.

3.94.1. tcp irtt [<milliseconds>]

Mostra o fissa il tempo della stima del percorso iniziale di andata e ritornodei datagrammi (irtt) , in millisecondi, da usare per nuove connessioni in TCP finche' i corrispondenti non potranno misurare e adattare loro stessi l' attuale valore. Il valore predefinito e' 5000 millisecondi (5 secondi). Aumentandolo , quando si opera su canali molto lenti, evitera' la sequenza di ritrasmissioni che accadrebbero altrimenti mamo mano che la stima normalizzata scende al valore corretto . Si noti che questo comando dovrebbe essere dato prima che il server ftp venga avviato con lo scopo di avere effetto sulle connessioni successive .

Il TCP mantiene inoltre traccia del tempo misurato di andata e ritorno e delle deviazioni medie (MDEV) per destinazioni correnti e recenti. Quando si apre una nuova connessione TCP , il sistema prima osserva tali tracce . Se viene trovata la destinazione , vengono utilizzati i valori registrati nelle tracce IRTT e MDEV . Altrimenti viene utilizzato il valore predefinito nominato precedentemente, insieme con MDEV a 0 . Questa caratteristica e' completamente automatica, e puo' incrementare molto le prestazioni , quando vengono aperte e chiuse una serie di connessioni per una data destinazione (es.: una serie di trasferimenti in FTP o un elenco di un sottodirettorio).

3.94.2. tcp kick <tcb addr>

Se ci sono dati non confermati sulla coda di invio (send queue) dei TCB specificati , questo comando forza una immediata ritrasmissione .

3.94.3. tcp mss [<size>]

Mostra o fissa la massima lunghezza del segmento TCP in bytes ( TCP Maximum Segment Size ) che verra' inviato su tutte le richieste di connessione TCP in uscita in (segmenti SYN). Questo comunica al lato remoto la lunghezza massima del segmento (pacchetto) che esso puo' inviare. Cambiando MSS si ha effetto solamente sulle connessioni future ; quelle gia' esistenti non vengono influenzate.

3.94.4. tcp reset <tcb addr>

Cancella il blocco di controllo TCP all'indirizzo specificato.

3.94.5. tcp rtt <tcb addr> <milliseconds>

Sostituisce il tempo di andata e ritorno valutato automaticamente nel TCB specificato con il rttval in millisecondi. Questo comando e' utile per incrementare la velocita' di recupero da

una serie di pacchetti persi poiche' esso fornisce una scorciatoia manuale attorno ai normali meccanismi di temporizzazione della ritrasmissione backoff.

3.94.6. tcp status [tcb addr>]

Senza argomenti , mostra diverse statistiche del livello TCP,
piu' un riassunto di tutte le connessioni esistenti in TCP, incluso indirizziTCB, la lunghezza delle code di invio e di ricezione , sockets locali e remoti, e lo stato della connessione . Se viene specificato tcb_addr , viene generato un elenco piu' dettagliato del TCB specificato, incluso i numeri di sequenza di invio e di ricezione ed informazioni sul timer.

3.94.7. tcp syndata [yes | no]

Mostra o attiv il segnale di "trasporto in spalla" di tcp syn + dati. Alcuni sistemi tcp non sono in grado di trasportare syn + dei dati insieme.

3.94.8. tcp timertype [linear | exponential]

Mostra o fissa il tipo di timer usato da tcp per il retry (backoff). Il modo "exponential" e' quello normale , dove rtt viene raddoppiato ad ogni retry. Nel modo "linear", rtt viene moltiplicato da una costante dopo ogni retry.

3.94.9. tcp trace [yes | no]

Mostra o attiva il modo di tracciamento (tracing) tcp .

3.94.10. tcp view

Mostra in sintesi lo stato delle connessioni aperte a livello TCP . Qui di seguito viene riportato un esempio di output prodotto da tale comando.

Esso comunque risulta piu' comodo del comando "tcp status" il quale produce un output molto piu' articolato e ricco di informazioni, spesso poco leggibili all'operatore poco esperto.

Inoltre tale comando e' utile ad osservare la bonta' dei parametri "tcp mss" e "tcp window" specialmente durante i trasferimenti di files. La configurazione migliore dara' come risultato un valore vicino allo zero sotto le colonne "Retries".

esempio:

net> tcp view

```
Send   Send    Receive Receive
&TCB Remote Socket:Port:Local Port/State  Bytes Retries  Bytes  Retries
7°99 151.90.51.200:ftp:1025/Established      57        0     228        1
8231 151.90.51.200:telnet:1024/Established   32        0   3640        1
```

3.94.11. tcp window [<size>]

Mostra o fissa l'ampiezza predefinita della finestra di ricezione in bytes che TCP usa quando si creano delle nuove connessioni. Le connessioni esistenti non vengono interessate.

$$telnet

3.95. telnet <hostid> [<socket #>]

Crea una sessione telnet verso l'host specificato ed entra nel modo conversazione. Se si immette il numero di socket opzionale, la connessione sara' diretta al socket remoto specificato; es.: 87 per TTYLINK, 110 per POP2, 513 per rlogin ,ecc.

Comunque per una buona emulazione di terminale VT100 con telnet si consiglia , inanzitutto l'uso del driver "NANSI.SYS" al posto di quello meno preciso del DOS "ANSI.SYS". Inoltre di dare al server UNIX , soprattutto quello della SCO, il tipo "dosansi" alla relativa richiesta "term=". Le freccie di direzione comunque non funzioneranno.

$$test

3.96. test

Attiva un test interno per problemi di overflow che possono comparire nella funzione clock di alcuni computer AT. In condizioni normali non produce nessun output questo comando.

$$thirth-party

3.97. thirth-party [yes | no]

Questo comando consente di restringere la gestione di posta
elettronica per terze parti.

$$ttylink

3.98. ttylink <hostid> [<port>]

Crea una sessione telnet all'host specificato ed entra in modo conversazione (chat). Se viene dato <port> verra' usato quel numero di socket. Il valore predefinito e' 87. Questo comando usa una interfaccia a schermo diviso per consentire una facile conversazione.

$$tip

3.99. tip <iface>

Crea una sessione tip che si connette all'interfaccia specificata nel modo "terminale stupido". L'interfaccia deve essere gia' attaccata con il comando attach. Tutto il traffico a pacchetti (datagrammi IP, ecc.) instradato verso <iface> mentre e' attiva questa sessione verra' scartato. Per chiudere una sessione tip, si usi il comando reset. Essa ritornera' quindi verso i normali modi di operazione slip , nrs o kiss .

Questa caratteristica e' principalmente utile per realizzare manualmente connessioni SLIP. Attualmente , solamente le porte entro contenute "com" possono essere utilizzate con questo comando.

$$xmodem

3.100. start tip

Avviamento della modalita' di trasferimento "xmodem" per il server TIP.

Estesa la funzionalita' del server TIP con la modalita' di trasferimento di file binari XMODEM.

In proposito si ricorda l'uso del server TIP. A differenza del noto cliente TIP , che consente di mettersi in collegamento con un dispositivo qualsiasi sull'interfaccia asincrona specificata, dando il comando "start tip <intface>" inanzitutto si sospende il traffico tcp/ip su detta

interfaccia. Poi si mette il NOS di KA9Q in modo host su quella interfaccia, cioe' verra' generato un prompt di login al terminale che si collega su quella interfaccia.

Da cio' si puo' comprendere che una volta effettuato il login , si puo' accedere ai servizi internet tipici del mailbox. In pratica il server TIP fornisce la possibilita' di interfacciare e quindi far comunicare
sistemi non dotati di intelligenza , come i terminali stupidi, oppure
sistemi non dotati di protocollo di comunicazione tcp/ip , come i PC
con i soliti programmi di comunicazione/emulazione .

La novita' introdotta con l'estenzione del server TIP, consiste nel poter selezionare il comando D(ownload) oltre che nel mode UUencode (DU, che ricordo trattasi di un metodo di codifica a 7 bit di file binari per la relativa trasmissione tipicamente via posta elettronica ), nel modo XMODEM , quindi con il comando mailbox DX. Ovviamente valgono i corrispondenti U(pload), UU, UX.

Pertanto volendo fare un esempio pratico: si supponga di possedere un PC con il noto programma di comunicazione QMODEM e volendo copiare alcuni file presenti sul disco del PC con il NOS di KA9Q , sara' sufficiente, una volta effettuato il login ed ottenuto il prompt del mailbox, dare il comando DX <nome_file> al server, e poi selezionare l'opzione di download binario XMODEM sul programma QMODEM (Pg dn).

sintassi : start tip <interface> <terminal|modem> [timeout seconds]

nuovi comandi sul mailbox: dx <filename> , ux <filename>

Esso lavora con terminali o modem a soli tre fili sulla linea seriale (TXD, RXD , SGND). La perdita del DCD verra' rilevata incondizionalmente.

Il valore predefinito di timeout TIP e' di 180 secondi.

Le connessione telefoniche e i caricamenti e scaricamenti avvenuti con successo di file con protocollo XMODEM verrano registrati nel file di log. Naturalmente i comandi XMODEM saranno disponibili solo su sessioni tip. Sono consentiti solo pacchetti xmodem standard da 128 byte alla volta che viene supportata dalla maggior parte dei programmi.

$$trace

3.101. trace [<iface> [off | <btio> [<tracefile>]]]

Regola la visualizzazione dei pacchetti attraverso i drivers dell'interfacce. Specifici bits abilitano la visualizzazione delle varie interfacce e l'entita' delle informazioni prodotte. Il tracciamento viene controllato sulla base del tipo di interfaccia; senza argomenti trace fornisce un elenco di tutte le interfacce definite ed il loro stato di tracing. La visualizzazione puo' essere limitata ad una singola interfaccia specificandola; anche i segnali di controllo possono cambiare specificandoli.I segnali vengono dati sotto forma di numero esadecimale che viene interpretato nel modo seguente:

BTIO

||||--- se 1 abilita la visualizzazione dei pacchetti in uscita , ||| se 0 la disabilita .

|||---- se 1 abilita la visualizzazione dei pacchetti in entrata, || se 0 la disabilita .

||----- controlla il tipo di visualizzazione:

| 0 - vengono decifrate le testate dei protocolli ,ma i
| dati non vengono visualizzati.| 1 - vengono decifrate
le testate dei protocolli , ed i

| dati (ma non le testate stesse ) vengono

```
|             visualizzati in   caratteri ASCII, 64 caratteri per
|             linea.   I   caratteri   non   stampabili   vengono
|             visualizzati  come punti.
|       2  -  Le  testate dei protocolli  vengono  decifrate,   e
|             l'intero  pacchetto  (testate  e  dati)       viene
|             visualizzato  in esadecimale e ASCII, 16  caratteri
|             per linea.
```

|------ Segnale di filtro dei broadcasts. Se si fissa 1, verranno

visualizzati solo i pacchetti indirizzati a questo nodo ed i pacchetti di broadcast non verranno visualizzati.

Se tracefile non viene specificato la visualizzazione sara' fatta alla console.

$$type

3.102. Aggiunta del comando "type" utile al sysop da remoto.

sintassi : type <nome_file> [<+-linee>]

Mostra i contenuti del file. Se viene fornita un numero di linea,

vengono mostrate le prime N linee, se <linee> e' positivo, oppure le ultime N <linee> se e' negativo. Esso differisce dal comando "more" perche' non crea una sessione separata e percio' lavora anche nel modo sysop remoto. A seguito dell'introduzione di questo comando e' stato tolto il comando "tail" poiche' divenuto superfluo.

$$udp

3.103. udp status

Mostra le statistiche del protocollo UDP a livello quattro .

Tipicamente esse riguardano il traffico effettuato da applicazioni quali Remote , Rip e Domain.

$$upload

3.104. upload [<filename>]

Apre <filename > e lo invia sulla sessione corrente come se fosse stato digitato sul terminale.

$$watch

3.105. watch

Mostra i valori correnti dei tempi intermedi di esecuzione del software , leggendo per ciascuno il valore minimo e massimo. Questa utilita' permette al programmatore di valutare il tempo di esecuzione delle sezioni critiche del programma con una risoluzione dell'ordine del microsecondo. Questo comando viene supportato sul PC IBM, ed il significato di ciascun valore di stopwatch dipende da dove sono state inserite le chiamate con propositi di test all' interno del codice sorgente; la copia in distribuzione del nos normalmente non ha chiamate di stopwatch.

$$watchdog

3.106. watchdog [on | off]

Abilita o disabilita il timer di whatchdog. Se le operazioni interne al ka9q cessano per 300 secondi e whatchdog e' abilitato, verra' effettuato un reset del sistema.

$$reboot

Dalla versione 1.20 del NOS e' stata inserito un meccanismo ulteriore di sicurezza contro i malfunzionamenti. Infatti nel caso il NOS non venga soddisfatto nelle richieste di memoria (morecores) per dieci volte consecutive, verra' effettuato un reboot del sistema.

Si veda il comando remote per la messa a punto del autoexec.bat.

$$x25

3.107. x25

Introdotto il codice per operazioni su X.25. Essa e' una implementazione di un DTE X.25 che sulla parte HDLC puo' agire sia da DCE che da DTE. A tale scopo e' stato modificato il KISS TNC . L' hardware necessario consiste in una parte di TNC normalmente usato per operazioni AX.25 via radio. Piu' precisamente sara' necessaria la parte digitale , quindi senza il modem interno.

Naturalmente se e' disponibile una porta PAD asincrona sul nodo X25 non e'necessario usare questa parte di programma per accedere alla rete. Infatti essa e' un tipo di connessione sincrona. Pertanto normalmente per basse velocita' sara' sufficiente usare una porta seriale asincrona standard.

Nota bene tale parte di codice non e' stato provato e comunque necessita ancora di qualche intervento prima di essere impiegato.

Adesso una breve descrizione dei comandi di livello 3 OSI:

3.107.1. x25 address <address>

Fissa l'indirizzo del DTE locale.

3.107.2. x25 channels <canali>

Massimo e minimo numero di canali . I valori predefiniti sono 1 e 100 rispettivamente.

3.107.3. x25 idle < time>

Fissa il tempo di inattivita' del timer . Il valore predefinito e' 30 secondi.

3.107.4. x25 table add <ip-address> <x25-address>

Aggiunge una voce alla tabella degli indirizzi .

3.107.5. drop <ip-address>

Cancella una voce alla tabella degli indirizzi .

3.107.6. Il significato degli altri comandi dovrebbe essere chiaro.

Dare il comando "x25 ?" per ottenerne una lista.

$$?

3.108. ?

E' lo stesso del comando help.

$$attach2

4. Sottocomandi attach

Questa sezione tratta nel dettaglio i comandi attach per i vari drivers delle interfacce hardware. Tutti questi drivers non possono essere configurati nell' eseguibile nos.exe; un elenco dei tipi disponibili si puo'ottenere immettendo il comando attach ?.

Alcuni parametri vengono accettati da parecchi drivers. Essi sono:

4.0.1. <bufsize>

Per dispositivi asincroni (es.: porte COM operanti nel modo SLIP

- NRS) questo parametro specifica la grandezza del ring-buffer del ricevitore. Esso dovrebbe essere grande abbastanza per mantenere i dati in arrivo a piena velocita' di linea per il massimo tempo che il sistema puo' essere occupato in MS-DOS o per il BIOS a fare una lenta operazione di I/O (es.: verso un floppy disk). Un kilobyte e' abitualmente piu' che sufficiente.

Su dispositivi sincroni (per es.: interfacce scc, hs, pc100, hapn e drsi operanti nel HDLC), il parametro bufsize specifica il pacchetto piu' grande che puo' essere ricevuto dall'interfaccia. Questo dovrebbe essere fissato in seguito ad accordi mutui tra stazioni che condividono il singolo canale fisico. Per operazioni standard AX.25 con la grandezza massima della trama I di 256 bytes, un valore di 325 dovrebbe fornire un adeguato margine di sicurezza. Su canali a velocita' piu' elevata (per es.: 56Kb/s) valori piu' grandi (per es.: 2Kbytes) forniranno prestazioni molto migliori e consentiranno il trasporto di pacchetti Ethernet alla massima grandezza senza frammentazione.

4.0.2. <ioaddr>

L'indirizzo base dei registri di controllo dell'interfaccia . Questo puo 'essere specificato in esadecimale come 0xnnn o decimale (dove nnn e' il numero in formato esadecimale).

4.0.3. <vector>

Il vettore di interruzione (interrupt IRQ) hardware dell'interfaccia in decimale. Quando un vettore viene seguito dal carattere 'c' allora il vettore viene aggiunto alla catena di interruzione. In questo modo dispositivi multipli possono condividere la linea di interruzione (tuttavia saranno necessarie modifiche hardware). un esempio e' l'uso di 4 porte com che dividono lo stesso vettore. Il primo comando attach ha un vettore piano ed il secondo avra' la 'c' appesa. Da notare che la porta con la velocita' piu' alta dovrebbe essere l'ultima ( poiche' verra' servita per prima dalla catena). NON si specifichi la 'c' con la prima dichiarazione attach in quel gruppo poiche' risultati accadranno imprevedibili .

```
Es.:     attach asy 0x3f8 4 144 ax25 2048 256 1200
           attach asy 0x3f0 4c 430 ax25 2048 256 9600
```

4.0.4. <iface>

Il nome (una stringa di caratteri qualsiasi) da assegnare a questa interfaccia. Esso viene usato per fare riferimento all'interfaccia nei comandi iface e route e nella visualizzazione di trace.

4.0.5. <mtu>

La dimensione della massima unita' di trasmissione (Maximum Trasmission Unit), in bytes. I datagrammi piu' grandi di questo limite verranno frammentati allo strato IP in pezzi piu' piccoli. Per trame AX.25 UI , comunque, e' rilevante anche il parametro ax25 paclen. Se il datagramma o frammento e' ancora piu' largo di paclen, viene frammentato anche al livello AX.25 (diversamente dal livello IP) prima della trasmissione(vedere il comando ax25 paclen per ulteriori informazioni ).

4.0.6. <speed>

La velocita' in bit per secondo (per es.: 2400).

4.1. attach 3c500 <ioaddr> <vector> arpa <iface> <qlen> <mtu> [<ip add>]

Attacca una interfaccia Ethernet 3Com 3C501 . qlen e' la lunghezza massima di trasmissione della coda. Se il parametro ip_addr non viene dato, verra' usato il valore associato al primo comando ip address .

L'uso di questo driver non e' raccomandato; si usi invece l'interfaccia packet driver con il relativo driver 3c501 .

4.2. attach asy <ioaddr> <vettore> slip | ax25 | nrs | ppp | crudo <iface> <bufsize> <mtu> <speed> [<vf>]

Attacca una "porta com" (porta seriale asincrona ) che utilizzi un chip National 8250 , 16450 o 16550°. I valori standard su IBM PC e compatibili di ioaddr e vettore sono 0x3f8 e 4 per COM1, e 0x2f8 e 3 per COM2. Se la porta usa un chip 16550°, sara' individuato automaticamente e i FIFO (First In First Out) saranno abilitati.

Sono disponibili i seguenti modi di operare :

4.2.1. ax25

Simile a slip, eccetto che vengono aggiunte una testata AX.25 ed una testata di controllo per KISS TNC all'inizio del datagramma prima della codifica SLIP. Possono essere usate entrambe le trame:

UI (senza connessione) oppure I (con connessione); si veda il comando "mode" per i dettagli.

4.2.3. nrs

Usa la tecnica della trama asincrona NET/ROM per comunicazioni con un TNC NET/ROM locale .

4.2.4. ppp

Point-to-Poin-Protocol. Incapsula datagrammi IP in trame tipo HDLC. Questo e' il nuovo standard Internet per comunicazioni punto a punto, compatibile con gli standards CCITT.

4.2.5. slip

Serial Line Internet Protocol. Incapsula datagrammi IP direttamente in trame SLIP senza una testata di livello 2 (link ). Questo serve per operazioni su linee punto_a_punto ed e' compatibile con lo SLIP 4.2BSD UNIX.

4.2.6. raw

Linea seriale raw (cruda) , speciale per server lpd (server di stampa).

4.2.6. <vf>

Essi sono dei segnali opzionali. v abilita la compressione delle testate TCP/IP secondo Van Jacobson, ed e' valido solo per SLIP. f forza il FIFO sui chip compatibili 16550AFN non impediti dal "baco progettuale " originale e non abbisognevoli di particolari attenzioni. Sfortunatamente questi ottimi chip non danno il loro FIFO abilitato. Specificando f sulla linea attach si forza l'uso

del FIFO (si otterranno invece risultati imprevedibili quando cio' sara'fatto su chips non del tipo 16550 !).

E' anche possibile regolare il livello di soglia FIFO con i chips UART 16550° puo' essere fissata a piacere sulla linea di comando attach. Il valore predefinito e' 4 byte. Valori validi sono 1,4,8,14. Nel tal caso l'opzione "f" diventa obbligatoria!

per es.: per una soglia di 8 byte si abbia:

"attach asy 3f8 4 slip linea1 1500 1006 9600 f8" oppure
"attach asy 3f8 4 slip linea1 1500 1006 9600 f 8"

4.3. attach axip <iface> <mtu> <ip addr> <callsign>

Questo comando crea un incapsulatore di trama AX.25 compatibile con RFC1226 appunto per trasmissioni di trame AX.25 sull'Internet. Iface sara' il nome dell'interfaccia, ip_Addr l'indirizzo del sistema remoto e callsign il nominativo AX.25 (indirizzo fisico) che questa stazione ascoltera' per la ripetizione delle trame. Da notare che ogni interfaccia axip attaccato dovrebbe avere un differente nominativo da ascoltare e questo dovra' essere anche differente da altri nominativi usati su questa stazione.

4.4. attach drsi <ioaddr> <vector> ax25 <iface> <buffsize> <mtu>
<ch a speed> <ch b speed>

Esso usa il driver di N6TTO per la scheda PCPA 8530 della Digital Radio System. Poiche' ci sono due canali sulla scheda, saranno attaccate due interfacce. Esse verranno chiamate iface con 'a' e 'b' appeso. bufsize e' la grandezza del buffer del ricevitore. ch_a_speed e ch_b_speed sono le velocita', in bits/sec, per i canali A e B, rispettivamente.

4.5. attach eagle <ioaddr> <vector> ax25 <iface> <buffsize>
<mtu>

Esso usa il driver di WA3CVG/NG6Q per la scheda Eagle
Computer (Zilog 8530).

4.6. attach hapn <ioaddr> <vector> ax25 <iface> <buffsize>
<mtu> csma | full

Esso usa il driver di KE3Z per la scheda Hamilton Amateur Packet Network (Intel 8273). I parametri csma | full specificano se la porta deve operare nel modo Carrier Sense Multiple Access (CSMA) o in full duplex.

4.7. attach hs <ioaddr> <vector> ax25 <iface> <buffsize> <mtu>
<keyup delay> <p>

Esso attacca una interfaccia per scheda DRSI PCPA oppure

Eagle Computer usando uno speciale driver per 8530 ad alta velocita'. Questo driver usa dei loops "busy-wait" per inviare e ricevere ogni byte invece degli interrupt, rendendolo adatto per modem ad alta velocita' (come quello di WA4DSY a 56 kb/s) su sistemi lenti. Questo pero' ha l'effetto di congelare il sistema ogniqualvolta il trasmettitore o il ricevitore del modem diviene attivo. Il driver puo' operare solo nella modalita' CSMA, e si raccomanda che nessuna altra interfaccia che richieda un piccolo ritardo da interrupt venga attaccata sulla stessa macchina.

4.8. attach packet <intvec> <iface> <txqlen> <mtu>

Driver per l'impiego con software separato "packet drivers" conforme alle specifiche della FTP Software, Inc . Il driver deve essere gia' lanciato in DOS prima che venga dato il comando attach. Vengono supportati i packet drivers per le classi Ethernet, ARCNET, SLIP, KISS/AX25 e SLFP.

intvec e' il vettore interrupt software usato per la comunicazione con il packet driver, e txqlen e' il massimo numero di pacchetti che sara' permesso nella coda di trasmissione.

4.9. attach pc100 <ioaddr> <vector> ax25 <iface> <buffsize> <speed>

Esso attacca una interfaccia per scheda Paccomm PC100 (Zilog 8530). Sono supportate solo operazioni AX.25 .

4.10. attach scc <devices> init <addr> <spacing> <Aoff> <Boff> <Dataoff> <intack> <vector> [p | r] <clock> [<hdwe>] [<param>]

Driver di PE1CHL per inizializzare una scheda di interfaccia SCC (8530) generica prima di attaccarla effettivamente. I parametri sono come segue:

4.10.1. <devices>

Numero di chip SCC da supportare.

4.10.2. <addr>

L'indirizzo base del primo chip SCC in esadecimale.

4.10.3. <spacing>

La spaziatura tra gli indirizzi base dei chip SCC .

4.10.4. <Aoff>

L'offset dall'indirizzo base del chip al suo registro di controllo del canale B.

4.10.5. <Boff>

L'offset dall'indirizzo base del chip al suo registro dati .

4.10.6. <Dataoff>

L'offset da ogni registro di controllo del canale al suo registro dati.

4.10.7. <intack>

L'indirizzo della porta Vettore INTACK/Read. Se e' nulla , si specifichi zero affinche' si possa leggere da RR3A/RR2B. Se invece si specifica, esso vale <addr> + 8 .

4.10.8. <vector>

Il vettore di interruzione hardware della CPU per tutti gli SCC connessi.

4.10.9. <clock>

La frequenza di clock (PCLK/RTXC) di tutti gli SCC in hertz. Si prefissi con 'p' per PCLK, 'r' per RTXC clock (per generazione clock).

4.10.10. <hdwe>

Tipo opzionale di hardware. I seguenti valori vengono attualmente supportati: 1 - Scheda Eagle, 2- Paccomm PC-100, 4 - PRIMUS PC card (DG9BL), 8 - scheda DRSI PCPA.

4.10.11. <param>

Parametro opzionale ulteriore. Al momento , cio' viene usato solo con le schede PC-100 e PRIMUS-PC per fissare la modalita' del modem. Il valore 0x22 viene usato con la PC-100 e 0x2 con la scheda PRIMUS-PC.

Il comando attach scc ... init deve essere dato prima che l'interfaccia venga effettivamente attaccata con i seguenti comandi.

4.11. attach scc <chan> slip | kiss | nrs | ax25 <iface> <mtu> <speed> < bufsize> [<call>]

Attacca una porta SCC inizializzata al sistema. I parametri sono come segue:

4.11.1. <chan>

Il numero di canale SCC da attaccare, 0 o 1 per la porta A o B del primo chip, 2 o 3 per la porta del secondo chip, ecc.

4.11.2. slip | kiss | nrs | ax25

I modi operativi dell'interfaccia. slip, kiss e nrs tutti operano la porta hardware nel modo asincrono; slip e' il modo standard per le linee seriali asincrone; kiss genera trame SLIP contenenti comandi KISS TNC e pacchetti AX.25; e nrs usa le convenzioni per linee seriali con trama NET/ROM per portare pacchetti NET/ROM. Selezionando il modo ax25 si mette l'interfaccia nel modo sincrono HDLC il quale e' adatto per connessioni dirette con modem radio in semi duplice per esempio.

4.11.3. <speed>

La velocita' dell'interfaccia in bit per secondo (per es.:1200). Si prefissi con 'd' quando e' disponibile un divisore esterno per generare il clock di TX. Quando la sorgente di clock e' PCLK, la velocita' di TX puo' essere un divisore per 32 tra TRxC e TRxC. Cio' e' necessario solo per operazioni full duplex sincrone. Quando questo argomento viene dato come 'ext' , i clocks di ricezione e di trasmissione sono esterni, e il generatore interno di baud rate (BRG) e il digital phase lockked loop /DPLL) non vengono usati.

4.12. Esempi di attach

Eccovi alcuni esempi di comandi attach:

```
#Attacca la scheda asincrona per PC, normalmente conosciuta come
#"com1", per operare punto_a_punto nel modo slip a 9600 baud,
#chiamandola "sl0". Viene allocato un ring-buffer al ricevitore di
#1024 bytes. I pacchetti da trasmettere piu' larghi di 256 bytes
#vengono frammentati.
attach asy 0x3f8 4 slip sl0 1024 256 9600
#Attacca la seconda porta asincrona per PC ("com2") per operare nel
#modo AX.25 con un MTU di 576 bytes a 9600 baud con un TNC KISS,
# e la chiama "ax0". Come modalita' predefinita, i $datagrammi IP
#vengono inviati come trame UI.
attach asy 0x2f8 3 ax25 ax0 1024 576 9600
#Attacca il packet driver caricato all'interrupt 0x7e. Il packet driver
```

#e’ adatto per un interfaccia ethernet, per esempio.
attach packet 0x7e ether 8 1500
#Attacca un’ interfaccia axip, cioe’ collegata all’interno di un’altra.
attach axip ai0 256 129.179.122.10 pa0gri-11
#e dall’altra parte del collegamento axip
attach axip ai0 256 129.179.122.130 pa0gri-12
#ora si supponga che l’host con interfaccia 129.179.122.10 abbia #un’interfaccia AX.25 con nominativo pa0gri-10 e 129.179.122.130 un #interfaccia con nominativo pa0gri-8.

#Ora una trama AX.25 sembra: pe1chl->pa0gri-11->pa0gri-8->pe1dna #[dati] Ricevuta da pa0gri-11, ammesso che sia stato ripetuto #(digipeated) , cio’ cambia il nominativo al dell’interfaccia con quello che #realmente sta entrando e lo incapsula in una trama IP di tipo 93 e lo #spedisce al numero IP 129.179.122.130 #pe1chl->pa0gri-10*->pa0gri-8*->pe1dna [dati]

#Arrivato a 129.179.122.130 viene cercato il ripetitore successivo #e dunque trovato. La trama viene cambiata in : pe1chl->pa0gri-10*-#>pa0gri-12*->pe1dna [dati]

#sicche’ sulla via del ritorno la trama trovera’ l’interfaccia corretta.

$$ftp2

5. Sottocomandi FTP

Durante una conversazione con un altro server, tutto cio’ che viene digitato alla console viene prima esaminato per vedere se e’ un comando locale conosciuto. Altrimenti , il comando viene passato intatto al server remoto sul canale di controllo. Se cio’ che si digita comunque e’ uno dei seguenti comandi, viene eseguito localmente (da notare che questo generalmente implica che vengano inviati altri comandi al server remoto sul canale di controllo).

5.1. ascii

Fissa il tipo di file da trasferire ad ASCII (valore predefinito). Si veda il comando Ftype per cambiare il valore predefinito.

5.2. binary

Fissa il tipo di file da trasferire in IMAGE.

5.3. dele

Questo comando viene usato per cancellare un file sul sistema remoto. Da ricordare che per l’uso di tale comando e’ necessario avere il permesso di cancellazione e sovrascrittura sul sistema remoto.

5.4. dir [<file> | <direttorio> [<file locale>]]

Senza argomenti, dir chiede che un elenco completo del direttorio corrente del server remoto venga inviato al terminale. Se viene dato un argomento, questo viene passato sul canale di controllo attraverso il comando LIST; questo puo’ essere un file specifico o direttorio che sia significativo sul file system remoto. Se vengono dati due argomenti, il secondo viene preso per il file locale nel quale si dovrebbe mettere l’elenco del direttorio (invece dell’invio sulla console ) (Il comando PORT viene usato prima di quello LIST sul canale di controllo ) .

5.5. flow [off | on]

Mostra o fissa la modalita’ di processo da parte di -more- sulla sessione corrente. Quando e’ attivato, viene proposto un -more- dopo ogni schermo pieno di dati. Si ottiene lo stesso

risultato utilizzando il comando sessione # flow . Da notare che cio' e' una estensione locale all'insieme di comandi standard ftp.

5.6. get <file remoto> [<file locale>]

Chiede al server remoto di inviare il file specificato nel primo argomento. Il secondo argomento, se dato, sara' il nome del file sulla macchina locale ; altrimenti avra' lo stesso nome come sulla macchina remota (verranno inviati i comandi PORT e RETR sul canale di controllo).

5.7. hash

Un sinonimo per il comando verbose 3.

5.8. ls [<file>|direttorio> [<file locale>]]

Ls e' identico a comando dir eccetto che sul canale di controllo viene inviatoil comando NLST al server invece del comando LIST. Da questo risultera' un elenco abbreviato del direttorio, per esempio mostrando solo i nomi dei files senza nessun'altra informazione.

5.9. mget <file> [<file>....]

Copia un insieme di files dal server. I nomi di files possono includere caratteri jolly; essi verranno interpretati ed espansi in una lista di files dal sistema remoto usando il comando di controllo NSLT. I files sulla macchina locale prenderanno gli stessi nomi di quelli sul server.

5.10. mkdir <direttorio remoto>

Crea un direttorio sul disco della macchina remota.

5.11. mput <file> [<file>....]

Invia un insieme di files al server. I nomi di files possono includere caratteri jolly; essi saranno espansi localmente in una serie da inviare. Questi avranno lo stesso nome del sistema locale sul server.

5.12. put <file locale> [<file remoto>]

Chiede al server remoto di accettare dati, creando il file nominato nel primo argomento. Il secondo argomento, se dato, sara' il nome del file sulla macchina remota; altrimenti avra' lo stesso nome come quello sulla macchina locale (vengono inviati i comandi PORT e STOR sul canale di controllo ).

5.13. resume <remote file> [<local file>]

Possibilita' di "resume" e di "rput" per FTP.
Se il server ftp remoto possiede questa caratteristica , e' possibile riprendere un trasferimento di files interrotto.
Resume e' simile al comando get. Se il file locale esiste gia', la sua lunghezza viene passata al server remoto ed i rimanenti bytes vengono inviati. Se il file locale non esiste, viene assunto di lunghezza zero e l'intero file viene reinviato. Da notare per ragioni di sicurezza, il comando "resume" lavora solo nel modo ricezione. Non e' possibile riprendere un trasferimento interrotto verso un server remoto. In tale caso, si dovra' usare il sotocomando "rput", se implementato .

5.14. rput <local file> [<remote file>]

Rput e' simile al comando put, e viene usato per riprendere i

trasferimenti interrotti o incompleti di file. Se il file remoto esiste gia', la

sua lunghezza viene passata al cliente locale ed i rimanenti bytes vengono inviati. Se il file remoto non esiste, viene assunto di lunghezza zero e l'intero file viene reinviato.

5.15. rmdir <direttorio remoto>

Cancella un direttorio sul disco della macchina remota.

5.16. type [a | i | l <grandezza_byte>]

Comunica al cliente locale ed al server remoto il tipo di file che

sta' per essere trasferito. Il valore predefinito e' "a", che significa ASCII

(es.: un file di testo ). Il tipo "i" significa imagine (image), es.: binario. Nel modo ASCII, i files vengono inviati con la lunghezza delle linee di testo in ASCII variabile separate da sequenze cr/lf ; in modo IMAGE, i files vengono inviati esattamente come appaiono nel file system. Il modo ASCII si dovrebbe usare ogni qualvolta che si trasferiscono testi tra sistemi diversi (es.:UNIX e MS-DOS) poisi seguono diverse convenzioni di fine-linea e/o fine del file. Quando questi si scambiano files di testo macchine dello stessotipo, lavoreranno entrambi i modi , ma il modo IMAGE e' abitualmente piu' veloce. Il tipo 'l' (grandezza del byte logico) viene usato nello scambiodi files binari come con servers remoti che trattano parole di bytes diverse (es.: DECSYSTEM-10s e 20s). Localmente lavora esattamente come IMAGE, eccetto che esso notifica al sistema remoto quanto grande e' il byte. grandezza_byte tipicamente e' 8. Il comando type fissa il modo il trasferimento locale e genera il comando TYPE sul canale di controllo.

5.17. verbose [0 | 1 | 2 | 3 | 4]

Mostra o fissa il livello di output nei messaggi durante i trasferimenti di file . Verbose 0 da' l'output minore, e verbose 4 da' il maggiore, cosi come segue:

0 -Mostra solo i messaggi di errore.

1 -Mostra i messaggi di errore piu' un sommario su di una linea dopo ogni trasferimento.

2 -Mostra gli errori e i messaggi sommari piu' i messaggi
generati progresivamente dal server FTP remoto (questo e' il valore predefinito).

3 -Mostra tutti i messaggi . Inoltre, viene mostrato un carattere di cancelletto (#) per ogni 1000 byte trasmesso o ricevuto.

4 -Mostra tutti i messaggi come 3. Al posto del carattere, (#)
viene mostrato un contatore numerico per i bytes trasmessi o ricevuti. Esso e' il valore predefinito.

Se un comando viene inviato al server remoto perche' non e' stato riconosciuto localmente, la risposta sara' sempre mostrata, senza riguardo del valore fissato dal sottocomando verbose. Cio' e' necessario per sottocomandi ftp tipo pwd (che mostra il direttorio corrente), il quale altrimenti non produrrebbe alcun messaggio se verbose fosse fissato a 0 oppure 1.

$$dialer2

6. Sottocomandi dialer

Ogni comando dialer puo' (dovrebbe) avere un file dialer diverso. Il file risiede nel direttorio di configurazione , come specificato nella sezione di installazione (si veda il capitolo 1). Un tipico file dialer potrebbe essere:

6.1. control down | up

Controlla l'interfaccia asincrona. L'opzione down lascia cadere il DTR e il RTS. L'opzione up attiva il DTR e l'RTS.

6.2. break

Se implementato manda un segnale di break su una linea
asincrona.

6.3. send "string"

Questo comando dialer scrivera' la stringa specificata sull'interfaccia. Gli apici sono necessari ed inoltre la stringa non puo' contenere caratteri di controllo al suo interno. Comunque , le sequenze di escape standard C di stringa saranno riconosciute ad eccezione di \0.

6.4. speed [9600 | 4800 | 2400 | 1200 | 300 ]

Questo comando dialer fissera' la velocita' dell'interfaccia ad una delle velocita' disponibili. Se la velocita' e' mancante, questa sara' mostrata nella finestra di sessione dialer.

6.5. wait <millisecondi> ["stringa di testo"] [velocita']

Se viene specificato solo il tempo, il dialer fara' una pausa per il numero desiderato di millisecondi.

Altrimenti, il dialer leggera' finche' la stringa di testo verra' rilevata sull'interfaccia. Se questa non viene rilevata invece entro il tempo desiderato, la funzione di autodialer si azzerera'. Gli apici di stringa sono richiesti, inoltre la stringa puo' non contenere caratteri di controllo incorporati. Comunque , le sequenze di escape standard C di stringa saranno riconosciute ad eccezione di \0.

In fine , se viene specificato il parametro speed, il dialer continuera' a leggere caratteri finche' non sara' rilevato un carattere non numerico. La stringa letta viene convertita in un intero, e usata per fissare la velocita' dell'interfaccia. Se il carattere non numerico in fine non viene rilevato entro il tempo desiderato, o il valore intero non e' una velocita' valida, l'autodialer si fermera'.

$$Installazione

7. Installazione

Il nos usa la seguente struttura di files e direttori:

~/alias
~/autoexec.nos
~/dialer
~/domain.txt
~/ftpusers
~/net.rc
~/netrom.sav
~/popusers

~/finger/
~/etc/printcap
~/etc/lpdperms
~/etc/log
~/spool/areas
~/spool/mail.log
~/spool/net.log
~/spool/forward.bbs
~/spool/history
~/spool/rewrite
~/spool/help
~/spool/mail
~/spool/mqueue
~/spool/news
~/spool/news/active
~/spool/news/pointer
~/spool/news/info
~/spool/news/help
~/spool/news/history
~/spool/news/forward
~/spool/news/poll
~/spool/rqueue
~/spool/signatur/
~/spool/lpd/
~/usr/doc/ka9q_doc.man

Il carattere ~ davanti a tutti i files e' un direttorio definibile con l'opzione -d sulla linea di comando del NOS. Puo' essere scelto qualsiasi nome; inoltre il valore predefinito e' senza direttorio, cioe' nella radice del disco corrente per es.: /. Se per esempio viene dato il comando nos.exe -d/net, la struttura di direttori si sposta su /net/... I files di configurazione alias, autoexec.nos, dialer, domain.txt , net.rc, popusers e ftpusers saranno posizionati nel suo interno. Il file netrom.sav sara' creato li.

Il direttorio "/spool/ e suoi sottodirettori saranno usati dai servizi bbs, SMTP e NNTP. I files di configurazione areas, forward.bbs, history, mail.log e rewrite saranno posizionati qui. Il direttorio /spool/news puo' avere molti sottodirettori ed ogniuna puo' averne ancora molte altre. I gruppi News verranno divisi in strutture gerarchiche di direttori. Un articolo news in newsgroup.rec.amateur.radio.packet finira' in /spool/news/rec/amateur/radio/packet.txt.

## 7.1. Il file /ftpusers

Poiche' MS-DOS e' un sistema operativo mono-utente (qualcuno potrebbe anche dire che e' solo un avviatore di sistema operativo glorificato ) , esso non fornisce controllo sull'acesso; tutti i files possono essere letti, scritti e cancellati dall'utente locale. Non e' desiderabile dare tale accesso libero al sistema agli utenti della rete. Nos percio' fornisce da se' i meccanismi di controllo sull'accesso al disco.

Il file ftpusers controlla l'accesso remoto FTP e mailbox. Il valore predefinito e' nessun accesso; se questo file non esiste, il server FTP e' inutilizzabile. Un utente remoto deve prima entrare ("login") nel sistema con i comandi USER e PASS, dando un nome valido e una parola chiave elencata in ftpusers, prima che egli possa trasferire files.

Ogni voce in ftpusers consiste di una singola linea della forma

username password /path permissions ip_address

Ci devono essere almeno quattro campi, ed esattamente uno spazio tra ogni campo. Possono essere aggiunti commenti dopo l'ultimo campo. Le linee di commento iniziano con "#" in prima colonna.

username e' il nome di login dell'utente.

password e' la parola chiave richiesta. Da notare che questa e' un testo in chiaro, percio' non e' una buona idea dare il permesso generale di lettura nel direttorio radice (root). Una parola chiave "*" (un asterisco singolo), significa che qualsiasi parola verra' accettata.

/path e' il direttorio in cui e' concesso operare sui files . Prima di qualsiasi operazione su files o direttori, il direttorio corrente ed il nome del file specificato dall'utente vengono associati per formare un indirizzario assoluto nella forma 'canonica' (es.: un'indirizzo completo partendo dalla radice, con riferimenti "./" e "../", come pure i ridondanti /, riconoscuti e rimossi). Il risultato DEVE iniziare con il prefisso della path ammissibile; altrimenti, l'operazione viene negata. Questo campo deve sempre iniziare con "/", es.: nel direttorio radice. Concessioni su in-dirizzari multipli possono essere specificati separati dal carattere ";",senza spazi bianchi intorno ad esso.

Permissions e' un numero decimale per la concessione dei permessi delle operazioni "read"," create" e "write". Se viene fissato il bit meno significativo (0x1), viene concesso all'utente di leggere un file soggetto comunque alla restrizione della path . Se viene impostato il bit successivo (0x2), viene concesso all'utente di creare un nuovo file se non sovrascrive un file preesistente. Se si imposta il terzo bit (0x4) viene concesso all'utente di scrivere un file anche se ne sovrascrive uno preesistente, ed in aggiunta egli puo' cancellare files. Di nuovo, tutte le operazioni vengono permesse soggette pero' alle restrizioni della path. I permessi possono essere combinati aggiungendo i bits, per esempio, 0x3 (= 0x2 + 0x1 ) significa che viene dato all'utente il permesso create e read, ma non il permesso di sovrascrittura/cancellazione (write).

Altri bits di permesso sono stati definiti come segue (i numeri vengono specificati in decimale) per l'erogazione di servizi mailbox e PPP:

$$Permessi

Permessi FTP

1 - Solo lettura.
2 - Scrive e crea sottodirettori, ma non sovrascrive.
4 - Sovrascrive e cancella files e sottodirettori.

Permessi Mailbox

8 - Accesso al gateway AX.25.
16 - Accesso Telnet.
32 - Accesso al nodo NET/ROM se attivato.
64 - Accesso dell'operatore remoto dopo l'invio del carattere "@".
128 - Questo utente viene rifiutato dal sistema.
256 - Bit di privilegio per connessioni PPP
512 - Bit di privilegio per la consultazione della coppia peerID/pass
1024 - Non consente l'utilizzo del comando send del mailbox (eccetto per il SYSOP).
2048 - Non consente l'utilizzo del comando read del mailbox . 4096 - Non consente la gestione di posta per terze parti dal mailbox.
8192 - Questa stazione e' un bbs conosciuto (funzione bbs forward, valido se configurato).

ip_address viene usato solo per PPP ed e' l'indirizzo IP remoto del sistema connesso.

Lo username univperm ha un significato speciale nel meccanismo di convalida. Se univperm viene incluso come utente valido nel file ftpusers , allora qualsiasi utente sconosciuto, cioe' non presente in ftpusers , sara' tradotto in univperm e ricevera' i suoi bits di permesso e la path dei files. Se univperm non viene incluso in ftpusers agli utenti sconosciuti non verra' consentito l'accesso.

Per esempio si supponga che sulla macchina pc.ka9q.ampr.org il file ftpusers contenga la linea:

```
friendly test /testdir 7
```

Una sessione che utilizzi questo account assomiglierebbe a questa:

```
net> ftp pc.ka9q.ampr.org
resolving pc.ka9q.ampr.org ... Trying 128.96.160.1...
FTP session 1 connected to pc.ka9q.ampr.org
220 pc.ka9q.ampr.org FTP version 900418 ready at Mon May
7
16:27:18 1990
Enter user name: friendly
331 Enter PASS command
Password: test [non visualizzata]
230 Logged in
ftp>
```

L'utente ora ha i privilegi per la lettura, scrittura, sovrascrittura e cancellazione per qualsiasi file sotto /testdir; egli non puo' accedere a nessun altro file.

Ecco alcuni altre voci d'esempio nel file ftpusers:

```
karn foobar / 7      # L'utente "karn" con parola chiave
        # "foobar" puo' leggere, scrivere,
        # sovrascrivere e cancellare qualsiasi                    # file sul sistema.
guest bletch /g/bogus;/public 3

   # L'utente "guest" con la parola chiave                                  # "bletch" puo'
   leggere qualsiasi file                                  # sotto /g/bogus e i suoi sottodirettori,
   # e /public e suoi sottodirettori e puo'                  # creare un nuovo file finche' non ne
   # sovrascrive gli esistenti. Egli non puo'                # cancellare nessun file.

anonymous * /public 1       # L'utente "anonymous"      (qualsiasi
                            #  parola chiave) puo' leggere files
                            # sotto /public e suoi sottodirettori;
                            # egli non puo' creare, sovrascrivere,   o
   # cancellare alcun file.
```

Questa ultima voce segue la convenzione standard per la tenuta di un "contenitore" di files pubblici; in particolare, l'username "anonymous" segue una convenzione stabilita da ARPA.

Gli utenti che accedono al mailbox da telnet devono dare il loro username e la loro parola chiave e percio' sono soggetti ai permessi dati nel file ftpusers. Gli utenti che accedono al mailbox da AX.25 o NET/ROM vengono identificati dal loro nominativo. Se questo e' presente nel file ftpusers, senza parola chiave (*), viene assunta quella voce ; altrimenti, l'utente sconosciuto avra' i permessi specificati da univperm (se presente). Questo sara' la voce predefinita per gli utenti non registrati.

$$popusers

7.2. Il file /popusers

Ecco la combinazione username/password definite per gli utenti POP. Essa ha una convenzione semplice:

user:password:

per ogni utente POP deve essere aggiunta una riga cosi fatta. I campi user e password dovrebbero coincidere con quelli dichiarati nel cliente POP remoto con il comando pop userdata . Entrambi user e password devono essere delimitati dal carattere ":".

$$net.rc

7.3. Il file net.rc

Il file net.rc consente un rapido accesso ai server ftp conosciuti. Ogni linea inizia con il nome del server ftp. Di seguito ci sono le dichiarazioni user e password da inviare al server ftp remoto. Il nome, lo user e la password sono separati da uno spazio , quindi non e' ammesso alcun carattere tab o piu' di uno spazio. Qui di seguito c'e' un file net.rc di esempio:

ucsd.edu anonymous ik3ngu@osi.iunet.it
ka9q.ampr.org guest pa0gri

$$domain.txt

7.4. Il file /domain.txt

Nos traduce i nomi di dominio (es.: "pc.ka9q.ampr.org") in indirizzi IP (es.: 128.96.160.3) attraverso l'uso di un dispositivo di risoluzione (resolver) dei nomi di dominio Internet e di un file "cache" locale, domain.txt. Ogni qualvolta l'utente specifica un nome di dominio, la voce desiderata viene ricercata nella cache locale . Se essa e' presente viene usata, altrimenti , se sono stati configurati server(s) di nomi di hdominio , viene inviata una richiesta sulla rete al server corrente. Se il server risponde, la risposta viene appesa al file domain.txt per uso futuro. Se il server non risponde, qualsiasi server aggiuntivo nella lista viene tentato senza fine in successione finche' non ne risponde uno, oppure non viene raggiunto il limite di retry . Se domain.txt non contiene la voce desiderata e non sono configurati servers di nomi di dominio, allora la richiesta fallisce immediatamente.

Se e' disponibile un server di nomi di dominio, e se tutti i riferimenti ad hostid nel proprio file autoexec.nos sono nel formato indirizzo IP, allora e' possibile iniziare con un file domain.txt completamente vuoto e lasciare che il nos lo crei automaticamente. Comunque, si puo' desiderare di aggiungere da soli le voci in domain.txt , perche' ognuno preferisce usare nomi simbolici di dominio nel proprio file autoexec.nos oppure perche' non si ha accesso a server di dominio e si preferisce creare le voci per tutti gli hosts a cui si desidera accedere. Ogni voce occupa una linea, ed i campi sono separati da tabulazioni . Per esempio:

pc.ka9q.ampr.org. IN A 128.96.160.3

IN e' la classe del record. Significa Internet, e sara' trovato in

tutte le voci. A e' il tipo di record, e significa che questo e' un record indirizzo. Percio' il nome di dominio pc.ka9q.ampr.org ha l'indirizzo Internet 128.96.160.3.

Un'altra possibile voce e' il record CNAME (Canonical Name), per es.:

```
ka9q.ampr.org.          IN   CNAME     pc.ka9q.ampr.org.
```

Questa linea dice che il nome di dominio "ka9q.ampr.org." e'

effettivamente un alias per il sistema con il nome di dominio (primario , o canonico) “pc.ka9q.ampr.org”. Quando viene dato al nos un nome di dominio che ha un record CNAME, il sistema automaticamente segue il riferimento al nome canonico ed immette l’indirizzo IP associato con tale voce.

Le voci aggiunte automaticamente dal nos avranno un campo addizionale tra il nome di dominio ed il campo classe (IN), per es.:

pc.ka9q.ampr.org. 3600 IN A 128.96.160.3

Questo e’ il valore time-to-live, in secondi, associato con il record ricevuto dal server. I Clienti mantengono (caching) questi records che si suppone vengano cancellati dopo che l’intervallo time-to-live e’ scaduto, ammettendo la possibilita’ che l’informazione nel record possa divenire vecchia.

Questa implementazione del nos decrementera’ il TTL a zero, ma non cancellera’ il record a meno che non sia stato dato il comando “domain cache clean on”. Quando non e’ disponibile un server remoto, verra’ usata la vecchia voce . Quando il valore TTL manca, come nell’esempio qui di sopra, il record non scadra’ mai e percio’ dovra’ essere gestito a mano editando il filedomain.txt. Poiche’ domain.txt e’ un file di testo in chiaro, puo’ essere facilmente editato dall’utente per aggiungere , cambiare o cancellare records.Possono comparire nel file domain.txt altri tipi di records, incluso NS (name server) e SOA (Start Of Authority) provenienti da risposte del server DNS remoto . Questi non vengono correntemente usati dal nos, ma vengono ritenuti di sviluppo futuro (come l’incorporazione stessa del server dei nomi nel nos).

Un altro tipo di voce e’ costituita dalla parola “$origin”. Essa viene usata per specificare un suffisso da appendere a tutte le voci di seguito nel file domain.txt fino al successivo $origin o la fine del file.

Per esempio :

```
$origin  ampr.org.
                iw0cnb     IN    A      44.134.0.3
                ir0rmt     IN    A      44.134.0.200
```

trattera’ automaticamente iw0cnb come iw0cnb.ampr.org ed ir0rmt come ir0rmt.ampr.org. Questo e’ un metodo migliore di quello usato attraverso il comando “domain suffix”.

$$alias

## 7.5. Il file /alias

Esso e’ il file di ALIAS del server SMTP. Questo serve per la risoluzione di un dato indirizzo di destinazione in una lista singola o multipla di voci di posta.

Formato:

mail_list_name call_1@host_1 [call_2@host_2].....# commenti

pa0gri gvdg@fridley.cdh.cdc.com
kelvin g1emm@g1emm.ampr.org
#
lino persico@persico.glt.esercito.it

#
glt maurici@maurici.glt.esercito.it
damico@damico.glt.esercito.it
greselin@greselin.glt.esercito.it
persico@persico.glt.esercito.it

bellucci@bellucci.glt.esercito.it
depietri@depietri.glt.esercito.it
Da notare che e' ragionevole e talvolta desiderabile avere i records di alias nella forma:

area area dest1 dest2

Poiche' il file alias viene scandito una sola volta, non ne risultera' una recursione infinita.

$$Areas

7.6. il file /spool/areas

Questo file e' la testa di un file mostrata all'utente del mailbox quando viene selezionato il comando "a". Esso dovrebbe mostrare tutti i mailbox public da leggere. Ecco un esempio:

---------- Public ---- Mail --- Area ----------

Generale -- Qualsiasi vecchio chit-chat che sia pulito
```
Tcp/ip     --  Messaggi tcp/ip generali. Nos ecc.
Bugs       --  Dove riportare bachi sul ka9q nos
```
Aggiornam. -- Info - revisioni del software nos

---

$$forward.bbs

7.7. Il file /spool/forward.bbs

Il mailbox legge un file di inoltro :/spool/forward.bbs .Ecco un file di esempio:

wb0ttw 0006
ax25 ax0 wb0ttw
wb0ttw
w0tn
mspbul
all

---

wb0gdb
netrom #msparth
..c msparth
all

---

La prima parola sulla prima linea nel record di inoltro e' il nome di BBS al quale verra' effettuato l'inoltro. Esso dovrebbe essere lo stesso tipo di nome che viene mostrato dal comando mbox status. La seconda parola e' opzionale. Esso specifica il tempo in cui potrebbe avvenire l'inoltro. 0006 significa che ci sara' solo inoltro a questa stazione tra mezzanote e le sei.

La seconda linea specifica come realizzare la connessione. Essa dovrebbe inziare con il protocollo (ax25, connect, tcp, telnet, o netrom) e sarebbe seguita da tutti i parametri che sono necessari quando il nos deve realizzare una connessione.

Direttamente dopo la seconda linea, ci possono essere linee che partono con un punto. Cio' che segue dopo il punto sara' inviato al bbs remoto appena possibile man mano che la connessione sara' realizzata.

Poi seguono i nomi di un numero di messaggi di area, pubblici e privati. In fine, ci dovrebbe essere una coppia di segni '-' per separare un record di inoltro dall'altro. Si termini anche il file con almeno una linea di trattini.

7.8. Il file /spool/rewrite.

Legge il file rewrite per le linee dove la prima parola e' un'espressione regolare e la seconda parola sono regole di riscrittura. Un terzo campo opzionale , contenente appena la lettera 'r' , quando presente, instrutisce il nos di riaviare il file di rewrite, usando il nuovo indirizzo di destinazione. Il carattere speciale '$' seguito da un numero denota la stringa che corrisponde ad un carattere '*' . I caratteri '*' sono numerati da 1 a 9.

Per es.: la linea "*@*.* $2@$1.ampr.org" riscriverebbe l'indirizzo "foo@bar.xxx" in "bar@foo.ampr.org".

```
#
*@g1emm.ampr.org $1
*@g1emm.ampr $1
*@g1emm $1
#
!!*!*!*!*!* $7%$6@$5@$4@$3@$2@$1
!!*!*!*!* $6@$5@$4@$3@$2@$1
!!*!*!*!* $5@$4@$3@$2@$1
!!*!*!* $4@$3@$2@$1
!!*!* $4@$3@$2@$1
!!* $3@$2@$1
! $2@$1
!* $1 r
#
#
```

$$ppp_note

### 7.8. Operazioni PPP

Brevi note sulle operazioni PPP (Tratto dalla RFC 1331 )

- Perche' PPP ?

Negli ultimi anni , la rete Internet ha visto una crescita esplosiva nelnumero degli host che supportano il protocollo TCP/IP. La gran parte di questihosts sono connessi a LAN di vari tipi tra cui Ethernet che sembra essere la piu' comune. La maggior parte degli altri host sono connessi attraversohWAN come le reti pubbliche dati tipo X.25. Relativamente pochi di questihosts sono connessi con semplici collegamenti punto a punto , per esempio seriali. Tuttavia i collegamenti punto a punto rappresentano il piu' vecchiosistema di comunicazione . Infatti le interfacce asincrone RS-232 sono praticamente dovunque.

- Incapsulazione

A differenza delle LAN dove esiste uno schema di incapsulazione standardsui collegamenti punto a punto non esiste. Percio' e' stato progettato PPP. Infatti PPP fornisce un protocollo di incapsulazione su entrambi i collegamentti : sincroni orientati al bit e asincroni con 8 bit di dati senza parita'. Questi collegamenti sono full-duplex , ma possono esseresia dedicati che a commutazione di circuito. PPP usa HDLC come base per l'incapsulazione.

PPP e' stato progettato attentamente per mantenere la compatibilita' conla maggior parte dell' hardware comunemente usato. Inoltre viene utilizzatoun meccanismo di "escape" per consentire il controllo dei dati, per es.affinche' la sequenza XON/XOFF venga trasmessa trasparentemente sulcollegamento , e quindi vengano rimossi dati di controllo spuri che potrebbero esservi inseriti dal sofware o dall'hardware.

L’incapsulazione PPP provvede anche al multiplexing di differenti protocollidello strato 3 rete simultaneamente sulla stessa rete, ma cio’ non si applica al KA9Q poiche’ esiste un solo protocollo a quel livello e cioe’ IP.

PPP usa il Frame Check Sequence HDLC per la rilevazione degli errori.

Come valore predefinito , solo 8 ottetti addizzionali sono necessari per formare l’incapsulazione. In ambienti dove la larghezza di banda e’ preziosa,l’incapsulazione puo’ essere ridotta ad appena 2 ottetti. Allo scopo di fornire supporto alle implementazioni hardware ad alta velocita’ PPP provvedea che il valore predefinito dell’incapsulazione della testate ed i campi di informazione cadano entro i 32 bit .

- Link Control Protocol (LCP)

Ancora piu’ importante , PPP definisce piu’ che uno schema di incapsulazione.Allo scopo di essere sufficientemente versatile per essere portabile versouna larga varieta’ di ambienti, PPP fornisce un protocollo di controllo allivello link (LCP). LCP viene usato per negoziare automaticamente sulle opzioni del formato di incapsulazione, gestire vari limiti circa la grandezzadei pacchetti, autenticare l’identita’ del proprio corrispondente dall’altraparte del collegamento , determinare quando un collegamento sta’ funzionando correttamente e quando questo e’ inattivo (defunto o morto), rilevare un loop ed altri comuni errori di configurazione, ed infine terminare ilcollegamento.

- Network Control Protocol (NCP)

I collegamenti PPP tendono ad eliminare molti problemi con l’attuale famiglia di protocolli di rete. Per esempio , l’assegnazione e gestione di indirizzi IP,che rappresenta un problema anche in ambiente LAN, e’ specialmente difficoltosasu collegamenti a circuito commutato ( vedi i server dotati di molti modems a selezione ). Questi problemi vengono gestiti da una famiglia di NCP i qualirisolvono ognuno le necessita’ specifiche dei loro rispettivi protocolli dellostrato rete. Come prima specificato il NOS di KA9Q ha solo un NCP cioe’ IPCP.

- Configurazione

Per definizione PPP deve essere facile da configurare . Infattiper progetto i valori predefiniti standards dovrebbero gestire tutte le configurazioni comuni. Gli implementatori possono specificare miglioramenti alla configurazione base predefinita, che vengano automaticamente comunicati al corrispondente sul link senza l’intervento dell’operatore.In fine, l’operatore puo’ esplicitamente configurare opzioni per ilcollegamento che abiliti ad operare in ambienti dove sarebbe altrimenti impossibile.

Questa auto-configurazione e’ implementata attraverso un meccanismo di opzione estensibile di negoziazione, dove ogni capo del collegamento descrive all’ altro le proprie abilita’ ed esigenze. Sebbene il meccanismo di negoziazione dell’opzione descritto in questo documento sia specificato in termini di Link Control Protocol (LCP), le stesse facilita’ possono essere usate da Internet Protocol Control Protocol (IPCP) ed altri della famiglia di NCP.

- In pratica

Dunque PPP per definizione rende i collegamenti seriali piu’ efficienti e sicuri attraverso operazioni semplici. Ma ora vediamo come questo si realizza in pratica con il NOS di KA9Q.

Inanzitutto e’ necessario cablare il cavo seriale RS-232 con tutti gli otto fili standard, e quindi senza usare ponticelli sul connettore lato computer (DTE). Cio’ si rende necessario a causa dell’ aumentata interazione tra il livello fisico e il livello collegamenteo dati (1 e 2 del modello ISO/OSI

) introdotta dal PPP. Si ricordi che dalla versione del NOS 1.20 e' possibile controllare lo stato di questi pin attraverso il comando "asystat" alla linea MC.

Inoltre e' necessario essere in possesso dell'eseguibile che contengail codice PPP che tipicamente si chiama: "nos_ppp.exe". Una volta configurato alla solita maniera l'interfaccia asincrona , eccetto che per il protocollo che stavolta dovra' essere "ppp" anziche' "slip", sono necessarie solo quattro righe per operare in modo automatico e con opzioni avanzate.Qui di seguito si riporta una configurazione di esempio:

```
attach asy 0x3f8 4 ppp linea1 4096 1500 4800
        #
dial linea1 link.ppp 30 3 <numero_IP_del_corrispondente>
        #
  ppp linea1 quick
  ppp linea1 lcp open
  ppp linea1 ipcp local address <mio_numero_IP_linea1>
  ppp linea1 ipcp open
```

Una volta lanciato il NOS , con la seconda linea qui di sopra descritta, si attiva il collegamento fisico . Per inciso va detto che il corrispondente dovra' essere nelle stesse condizioni operative per consentire un collegamento PPP. Non appena sara' stabilito que-st' ultimo si attiveranno i vari strati software PPP , e cioe' :

NCP, LCP,IPCP.

Comunque, dopo alcuni istanti il link PPP sara' attivo e quindi sara' noto il corrispondente con il suo numero IP. Da quel momento in poi , se si opera come gateway , verra ' attivato lo scambio delle tabelle degli instradamenti con RIP verso il corrispondente. Altrimenti sara' sufficiente consultare dette tabelle per scoprire se una data rete o host sono raggiungibili . Per chiudere un link PPP sara' sufficiente abbattere il collegamento fisico o dare il comando " ppp linea1 lcp close". Il corrispondete sara' cancellato dalla tabella delle rotte e gli eventuali instradamenti rimasti potranno essere rimossi o con il comando " route flush" o aspettando il tempo di ttl rimasto.

Naturalmente durante le operazioni PPP e' possibile osservare tutta una serie di valori , prima assenti con il modo "slip" , sui vari contatori, timers , ecc. circa il buon funzionamento del link attraverso il comando : "ifconfig linea1". Purtroppo , come al solito, tali statistiche non sono di facile lettura , ma consentono comunque di rilevare la bonta' del collegamento seriale verso il corrispondente.

Concludendo dunque , i collegamenti PPP consentono , oltre che una maggiore efficienza , una certa automaticita' rispetto a prima e al tipico lavoro dell'operatore. Inoltre si ha la sicurezza di operare con un protocollo standard, come quello per le reti locali su cavo coassiale Ethernet.

$$smtpuser

## 7.9. File smtpuser

L'opzione -s al comando "start smtp" consente , oltre ad effettuare una operazione di selezione di seguit descritta, di risolvere un problema di loop infinito sul server alla ricezione di messaggi senza il campo utente (<>@<host>).

Se presente tale opzione verra effettuato un controllo sulla posta in arrivo. Essa consente di ricevere messaggi da un cliente smtp solo se l' utente destinatario e presente nel file "smtpuser" .

Per esempio dovendo ricevere un messaggio per pippo presso questo host (per es.: pippo@ntt3.glt.esercito.it), se il server e stato attivato con "start smtp -s" verra controllato il

file “smtpuser”. Se tale file contiene il nome pippo allora verra ricevuto il relativo messaggio come sempre. Altrimenti ne verra generato un altro per il mittente da MAILER-DAEMON con l’oggetto “Failed mail” ; all’interno vi sara’ un messaggio di errore smtp “utente pippo sconosciuto” e quindi il messaggio non recapitato .

Il file “smtpuser” si dovra’ posizionare nel direttorio di lavoro del NOS di KA9Q cosi come gli altri file di dati tipo ~\ftpusers eccetera.

Tale caratteristica e’ utile nella gestione di un server POP dove i messaggi recapitati sono tipicamente destinati a piu’ utenti. Infatti puo’ capitare di compilare il nome dell’ utente in modo errato e non accorgersi di cio’ . Come e’ noto il server smtp del NOS, prima dell’ introduzione di questa carattestica accettava tutti i nomi utente . Per cui poteva facilmente risultare un mancato recapito del messaggio all’ utente giusto.

Inoltre , come accennato nel titolo , e’ stato risolto un baco sul server smtp per il fatto che ora se l’ utente non viene incluso nel file “smtpuser” non viene accettata posta per questi anche nel caso di mancata compilazione del nome di utente.

$$parametri

8. Installazione di Buffsize, Paclen, Maxframe, MTU, MSS e Window.

Molti utenti nos sono confusi da questi parametri e non sanno come installarli adeguatamente. Questa sezione passera’ in rivista prima questi parametri e poi discutera’ come sceglierne i valori. Speciale enfasi viene data nell’evitare problemi di interoperabilita’ che possono apparire quando si comunica con implemetazioni AX.25 non-nos.

8.1. Parametri hardware

$$bufsize

8.1.1. Bufsize

Esso specifica la grandezza del buffer da allocare per ogni porta di ricezione.

Non c’e’ bufsize predefinito , inoltre esso deve essere specificato nel comando attach per l’interfaccia e puo’ essere cambiato con il comando “ifconfig”.

8.2. Parametri AX.25

$$paclen

8.2.1. Paclen

Paclen limita la grandezza del campo dati in una trama AX.25 I . Questo valore non include la testata del protocollo (sorgente, destinazione e indirizzi dei ripetitori).

Poiche’ nel modo sconneso (datagram) si usano trame UI, questo parametro non ha effetto in tale modo. Il valore paclen di default e’ di 256 bytes.

$$maxframe

8.2.2. Maxframe

Questo parametro controlla il numero delle trame I che il nos puo’ inviare su una connessione AX.25 prima che si debba fermare ed aspettare una conferma. Poiche’ il campo di sequenza AX.25/LAPB e’ grande 3 bits, questo numero non puo’ essere piu’ grande di 7.

Poiche' nel modo sconnesso (datagram) AX.25 usa trame UI che non hanno numeri di sequenza, questo parametro non influisce in tale modo operativo.

Il valore maxframe predefinito nel nos e' 1 trama.

## 8.3. Parametri IP e TCP

$$mtu

### 8.3.1. MTU

L'MTU (maximum transmission unit) e' un parametro d'interfaccia che limita l'ampiezza del datagramma IP piu' grande che questa possa gestire. I datagrammi IP instradati su di una interfaccia, che sono piu' grandi della sua MTU vengono spezzati in tre o piu' frammenti. Ogni frammento ha la sua testata IP e viene gestito dalla rete come se fosse un datagramma IP distinto, ma quando arriva a destinazione viene trattenuto dallo strato IPfinche' tutti gli altri frammenti appartenenti al datagramma originale non saranno arrivati. Allora essi saranno riassemblati nel datagramma IP originale completo. L'MTU minimo accettabile di interfaccia e' di 28 bytes: 20 bytes per la testata IP (frammento), piu' 8 bytes di dati.

Non c'e' MTU predefinito nel NOS; esso deve essere esplicitamente dichiarato per ogni interfaccia come parte del comando attach o con il comando "ifconfig".

$$mss

### 8.3.2. MSS

MSS (Maximum Segment Size) e' un parametro del livello TCP che limita la quantita' dei dati che il TCP remoto inviera' in un singolo pacchetto. I valori MSS vengono scambiati nei pacchetti SYN (richiesta di connessione) che aprono una connessione TCP.

Nell'implementazione nos del TCP, l'MSS effettivamente usato e' ulteriormente ridotto allo scopo di evitare frammentazioni sull'interfaccia IP locale. Cioe', il TCP locale chiede ad IP per l'MTU dell'interfaccia che sara' usata per raggiungere la destinazione. Esso allora sottrae 40 bytes dal valore MTU tenendo conto delle testate TCP ed IP che sono sopra. Se il risultato e' minore dell' MSS ricevuto dal TCP remoto, allora viene usato questo ultimo.

Il valore predefintio di MSS e'di 512 bytes.

$$window

### 8.3.3. Window

Questo e' un parametro TCP che controlla quanti dati il TCP locale permettera' al TCP remoto di inviare prima che esso debba fermarsi ed aspettare per una conferma. Il valore effettivo di "window" (finestra) usato dal TCP nel decidere quanti dati in piu' inviare si chiama finestra effettiva. Questo e' il piu' piccolo di due valori : la finestra proposta dal TCP remoto meno i dati non confermati in viaggio , e la finestra di congestione, che e' una stima variabile nel tempo, di quanti dati la rete puo' gestire, calcolata automaticamente.

Il valore predefinito di Window e' 2048 bytes.

$$frammentazione

## 8.4. Discussione

### 8.4.1. Frammentazione IP e Segmentazione AX.25

La frammentazione al livello IP spesso rende possibile interconnettere due diverse reti, ma e' meglio evitarla ogni qualvolta e'possibile. Una delle ragioni e' che quando un singolo frammento IP viene perso, tutti gli altri frammenti appartenenti allo stesso datagramma vengono effettivamente persi e l'intero datagramma deve essere ritrasmesso dal nodo origine. Anche senza perdita, i frammenti richiedono l'allocazione di buffers di memoria temporanei alla destinazionee non e' mai facile decidere quanto tempo aspettare per i frammenti mancanti prima di rinunciare e scartare quelli che sono gia' arrivati. Un timer di riassemblaggio controlla questo processo. Nel nos esso viene (re)inizializzato con il parametro ip rtimer (valore predefinito 30 secondi) ogni qualvolta e' in progresso il riassemblaggio di un datagramma (es.: viene ricevuto un frammento nuovo ). Non e' necessario che tutti i frammenti appartenenti al datagramma arrivino entro un singolo intervallo di timeout, solo quell'intervallo tra segmenti dovrebbe essere inferiore del timeout.

La maggior parte delle sottoreti che portano IP hanno MTU di 576

bytes o piu', cosi' interconettendole con sottoreti che hanno valori

piu' piccoline puo' risultare una considerevole frammentazione. Per questa ragione, gli implementatori IP che stanno lavorando su collegamenti o sottoreti che hanno abitualmente piccoli limiti di ampiezza dei pacchetti sono incoraggiati ad usare la frammentazione trasparente, cioe', ad escogitare degli schemi per spezzare grossi datagrammi IP in sequenze di trame al livello link o sulle sottoreti che vengono immediatamente riassemblati all'altro capo del collegamento o della sottorete nel datagramma IP originale, senza usare la frammentazione al livello rete (IP). Tale schema e' fornito nella versione AX.25 2.1. Esso puo' spezzare un grosso datagramma IP o NET/ROM in una serie di segmenti (da non confondere con i segmenti TCP) di grandezza paclen AX.25, uno per trama AX.25 I, per trasmetterle e riassemblarle in un singolo datagramma all'altro capo del collegamento prima di portarlo su' al modulo IP o NET/ROM.

Sfortunatamente, la procedura di segmentazione e' una nuova caratteristica in AX.25 e non e' ancora largamente implementata; infatti il nos e' fino ad ora la sola implementazione conosciuta. Questo crea alcuni hproblemi di interoperabilita' tra il NOS e nodi non-nos, in particolare con i nodi a standard NET/ROM che vengono usati per portare datagrammi IP. Questo problema e' discusso oltre nella sezione d'installazione dell'MTU.

### 8.4.2. Installazione di paclen e bufsize

Piu' dati si mettono nella trama AX.25 I, piu' piccole sono le testate AX.25 in relazione alla grandezza totale della trama. In altre parole incrementando paclen, si abbassa cio' che e' sopra (overhead) al protocollo AX.25. Anche, grossi pacchetti di dati riducono l'overhead nell'attivare il trasmettitore, e questo puo' essere un fattore importante con i modems ad alta velocita'. Dall'altra parte trame grandi aumentano le difficolta' coni destinatari in termini di rumore e di interferenze . Ogni collegamento ha un valore ottimale di paclen che viene scoperto al meglio sperimentando.

Un'altra cosa da ricordare quando si da' il valore paclen, e' che la specifica della versione AX.25 2.0 la limita a 256 bytes. Sebbene il nos possa gestire valori molto piu' grandi, alcune altre implementazione AX.25 (incluso i digipeaters) non possono e questo puo' causare problemi di interoperabilita'. Anche il nos puo' avere problemi con certi KISS-

TNC poiche' hanno la grandezza dei buffers fissi. Il programma originale del KISS-TNC per il TNC-2 di K3MC puo' gestire trame limitate in ingresso solo dalla RAM nel TNC, ma alcuni altri KISS-TNC non possono.

I drivers HDLC entro contenuti del nos allocano buffers di ricezione secondo la massima grandezza della trama aspettata , sicche' diviene importante la corretta configurazione di questi dispositivi con il corretto bufsize. Per far questo, si deve conoscere la grandezza della trama piu' grossa possibile che puo' essere ricevuta. Il parametro paclen controlla solo la grandezza del campo dati in una trama I e non la grandezza totale della trama come appare in aria. Le specifiche AX.25 permettono fino ad 8 digipeaters, cosi'la trama piu' grande possibile e' paclen + 72 bytes. Percio' si dovrebbe rendere bufsize almeno di questa grandezza.

Un'altra importante considerazione e' che le versioni piu' recenti del nos migliorano la risposta di interrupt mantenendo uno speciale fondo comune (pool) di buffers per l'utilizzo da parte delle procedure di ricezione. Questi buffers sono correntemente fissati nella grandezza di 2048 bytes e tale valore puo' essere cambiato solo editando config.h e ricompilando il nos. Cio' limiterebbe bufsize; infatti, il tentativo di mettere un valore piu' grande puo' portare a disattivare il driver . Questa situazione puo' essere rivelata immettendo il comando memory status ed osservando un conteggio diverso da zero per gli eventi Ibuffail, sebbene questi possano accadere occasionalmente durante le normale operazioni .

Uno degli svantaggi dell'AX.25 e' che non c'e' modo per una stazione di comunicare all'altra quanto grande e' il pacchetto che e' disposta ad accettare. Questo richiede alle stazioni che stanno dividendo un canale diaccordarsi in anticipo sulla massima grandezza del pacchetto. TCP e' differente, come vedremo.

### 8.4.3. Installare Maxframe

Per una migliore prestazione su di un canale radio in semiduplice (half- duplex), maxframe dovrebbe essere sempre impostato ad 1. Le ragioni sono spiegate nel documento Link Level Protocols rivisto da Brian Lloyd e Phil Karn, comparso durante lo svolgimento della 5 Conferenza ARRL su Reti di Computers nel 1986.

### 8.4.4. Installare MTU

Nella scelta di MTU , per il complesso della testata TCP/IP , si applicano considerazioni simili a quelle dello strato AX.25 quando si installa paclen. Comunque, certi tipi di sottoreti supportate dal nos hanno un MTU prestabilito, percio', a meno che si sappia cosa si sta facendo, bisognerebbe sempre usare questi:1500 bytes per ETHERNET e 508 bytes per ARCNET. Altri tipi di sottoreti, incluso SLIP e AX.25, non sono ben standardizzate. SLIP non ha un MTU ufficiale, ma le implementazioni piu' comuni (per UNIX-BSD) usano un MTU di 1006 bytes. Sebbene il nos non abbia un limite su filo fisico per quanto riguarda la grandezza di trama SLIP ricevuta, questo non e' vero per altri sistemi. Problemi di interoperabilita' possono percio' risultare, se vengono impiegati MTU piu' grandi nel nos.

Scegliere un MTU per una interfaccia AX.25 e' piu' complesso. Quando l'interfaccia opera nel modo "datagram " (trama UI), il parametro paclen non ha effetto. L'MTU effettivamente diventa il paclen del (link) collegamento. Comunque, come menzionato prima, i grossi pacchetti inviati su connessioni AX.25 vengono automaticamente segmentati in trame I non piu' grandi dei bytes in paclen. Sfortunatamente il nos e' finora la sola implementazione conosciuta della nuova procedura di segmentazione AX.25. Questo va' bene finche' tutti i nodi NET/ROM lungo il percorso sono nos, ma

poiche' la ragione principale per cui nos supporta NET/ROM e' permettere l'uso delle esistenti reti NET/ROM, cio' e' improbabile.

Percio' ,di solito , e' importante evitare la segmentazione AX.25 quando IP viaggia su NET/ROM. Il modo per attuare questo e' assicurare che pacchetti piu' grandi di paclen non vengano mai gestiti da AX.25. Una testata di trasporto NET/ROM e' lunga 5 bytes ed una testata rete NET/ROM consta di 15 bytes, sicche' devono essere aggiunti 20 bytes all'ampiezza di un datagramma IP quando si calcola l'ampiezza del campo dati della trama I AX.25. Se paclen e' 256, esso lascia 236 bytes per il datagramma IP. Questo e' l'MTU default della pseudo-interfaccia netrom, cosi' finche' pa-clen e' almeno 256 bytes, la segmentazione AX.25 non puo' avvenire. Ma se vengono usati valori piu' piccoli di paclen anche l'MTU netrom deve essere ridotto con il comando ifconfig.Dall'altra parte se IP gira in cima ad AX.25, c'e' la possibilita' che tutti i nodi stiano girando con nos e supportino la segmentazione AX.25. In questo caso non c'e' motivo di non usare MTU piu' grandi e lasciare che la segmentazione AX.25 avvenga. Se si sceglie un MTU dell'ordine di 1000-1500 bytes, si puo' largamente evitare la frammentazione al livello IP e ridurre "il peso " della testata del livello TCP/IP su trasferimenti di files a livelli molto bassi. Si e' sempre liberi di scegliere qualsiasi valore di paclen sia appropriato per il collegamento.

### 8.4.5. Installare MSS

L'installazione di questo parametro del livello TCP e' qualcosa di meno critico dei parametri del livello IP e AX.25 gia' discussi, principalmente perche' esso viene abbassato automaticamente secondo l'MTU dell'interfaccia locale quando viene creata una connessione. Sebbene questa sia, strettamente parlando , una violazione dei protocolli a strati (Layering; si suppone che TCP non sia a conoscenza delle operazioni degli strati piu' bassi) questa tecnica lavora bene in pratica. Comunque, essa puo' essere ingannata; per esempio se avviene un cambiamento di rotta dopo che la connessione e' stata aperta e la nuova interfaccia locale ha un MTU piu' piccolo di quello precedente, puo' occorrere la frammentazione IP nel sistema locale.

Il solo svantaggio di fissare un grosso MSS e' che puo' causare

frammentazioni evitabili in qualche altro punto entro il percorso di rete se esso include una sottorete "collo di bottiglia" con un MTU piu' piccolo di quello dell'interfaccia locale (sfortunatamente, non c'e' modo attualmente di riconoscere quando questo sia il caso. E' in corso un lavoro nell'ambito della Task Force Engineering Internet su di una procedura "Alla scoperta dell'MTU" per determinare il piu' grosso datagramma che puo' essere inviato su di un dato percorso senza frammentazione, ma non e' ancora completo). Inoltre, poiche' l'MSS che si specifica viene inviato al sistema remoto, e non tutti gli altri TCP effettuano ancora la procedura di abbassamento dell'MSS, questo puo' causare al sistema remoto la generazione di frammenti IP non necessaria.

Dall'altra parte, un MSS troppo piccolo puo' produrre considerevole perdita di prestazione, specialmente quando opera su reti locali veloci e su reti che possono gestire pacchetti piu' grandi. Cosi' il miglior valore di MSS e' probabilmente 40 di meno dell'MTU piu' grande sul proprio sistema, con un margine di 40 bytes che permette le testate IP e TCP. Per esempio, se si ha una interfaccia SLIP con un MTU di 1006 bytes e una interfaccia Ethernet con un MTU di 1500 bytes, si fissi MSS a 1460 bytes. Questo permette di ricevere pacchetti Ethernet alla massima grandezza, supponendo che il percorso per il proprio sistema non abbia sottoreti a "collo di bottiglia" con MTU piu' piccoli.

### 8.4.6. Installazione di Window

Un protocollo a finestra mobile come TCP non puo' trasferire dati per piu' di una finestra per intervallo di tempo di andata e ritorno. Cosi' questo parametro del livello TCP controlla la capacita' del TCP remoto a mantenere un lungo canale di dati (pipe) pieno. Cioe', quando si opera su di un percorso con molti salti, offrendo una grossa finestra TCP si aiutera' a tenere tutti (que)i nodi occupati mentre si ricevono dati. D'altra parte, offrire una finestra troppo grande puo' congestionare la retese non puo' bufferizzare tutti i dati . Fortunatamente , da pochi anni sono stati sviluppati nuovi algoritmi per il controllo della finestra in modo dinamico dell'effettivo flusso TCP ed ora sono largamente impiegati. Nos li include , e si possono osservare in azione con i comandi tcp status <tcb> o socket <sockno>. Si osservi il valore "cwind" (congestion window).

Nella maggior parte dei casi, e' sicuro mettere TCP window al multiplo intero piu' piccolo di MSS, es.: 4x, o piu' largo se necessario all'utilizzo pieno di una larghezza di banda elevata moltiplicato per il ritardo introdotto dal percorso . Una cosa da tenere in mente, comunque e' che proponendo un certo valore di finestra TCP si dichiara che il sistema ha quel tanto di spazio di buffer per i dati in arrivo. Nos effettivamente non alloca prima questo spazio ; lo tiene in un fondo comune (pool) che hpuo' comunque utilizzare, sfruttando il fatto che molte connessioni TCP sono inattive per lunghi periodi e scommettendo sul fatto che la maggior parte delle applicazioni leggera' i dati in ingresso sulle connessioni attivenon appena questi saranno arrivati , liberando percio' rapidamente il buffer di memoria. Comunque, e' possibile far girare il nos fuori memoria se vengono proposte ampiezze eccessive di finestra TCP lasciando che le applicazioni "vadano a dormire" per sempre (es: sessioni Telnet sospese) ; oppure se arrivano molti dati fuori sequenza. E' consigliabile tenere d'occhio la quantita' di memoria disponibile e decrementare l'ampiezza della finestra TCP (o limitare il numero delle connessioni simultanee) se questa scende troppo in basso.

L'uso di valori Window che eccedono MSS possono causare un incremento delle collisioni sul canale, dipende dal metodo d'accesso al canale e dal protocollo del livello link . In particolare , collisioni tra pacchetti di dati e conferme di ritorno durante un certo volume di trasferimenti di files possono divenire comuni. Sebbene questo non sia, strettamente parlando , un difetto del TCP, e' possibile lavorare intorno al problema al livello TCP, diminuendo la finestra in modo che il protocollo operi nel modo stop_and_wait. Questo avviene impostando il valore della finestra uguale ad MSS.

8.5. Conclusione

In molti casi i valori predefiniti forniti dal nos per ognuno di questi parametri lavoreranno correttamente e daranno prestazioni ragionevoli. Solo in circostanze speciali , come con operazioni su collegamenti molto poveri o nella sperimentazione di modems ad alta velocita', dovrebbe essere necessario cambiarli.

$$MIB

9. Variabili MIB

Le variabili MIB sono oggetti collocati in un database chiamato appunto Management Information Base cioe' Base di Informazioni per la Gestione delle comunicazionioni. Le reti vengono gestite attraverso la continua osservazione e l' aggiornamento dei contatori, timers, descrizioni di stringhe e insiemi che costituiscono questo database. La gestione di tali oggetti viene definita dal documento "Abstract Syntax Notation One" (ASN.1). Ogni oggetto ha un nome , una sintassi e un codice.

L’ insieme delle variabili standard MIB-1 sono state specificate con il documento RFC 1156 (1989).

Gruppo Internet Protocol

Oggetti in relazione con lo strato del protocollo IP (3 OSI)

Lista ottenuta con il comando ip stat del pacchetto ka9q nos tcp/ip.

net> ip stat

```
( 1)ipForwarding                1      ( 2)ipDefaultTTL                10
( 3)ipInReceives                0      ( 4)ipInHdrErrors                0
( 5)ipInAddrErrors              0      ( 6)ipForwDatagrams              0
( 7)ipInUnknownProtos           0      ( 8)ipInDiscards                 0
( 9)ipInDelivers                0      (10)ipOutRequests                0
(11)ipOutDiscards               0      (12)ipOutNoRoutes                0
(13)ipReasmTimeout             40      (14)ipReasmReqds                 0
(15)ipReasmOKs                  0     (16)ipReasmFails                  0
(17)ipFragOKs                   0     (18)ipFragFails                   0
(19)ipFragCreates               0
```

ipForwarding Indica se l’ entita’ sta agendo come un gateway IP in relazione all’ inoltro dei datagrammi ricevuti (ma non indirizzati a ) da un’ altra entita’.

ipDefaultTTL Valore predefinito inserito nel campo Time-To-Live della testata IP del datagramma generato dall’ entita’ ne fornito dallo strato del protocollo Trasporto (4 OSI).

ipInReceives Numero di datagrammi ricevuti in ingresso dalle interfacce, incluso quelli ricevuti con errore.

ipInHdrErrors Numero di datagrammi in ingresso scartati in seguito ad errori nelle loro testate IP, incluso i cattivi checksum ,discordanza di versione, altri errori di formato, Time-To-Live scaduto, errori scoperti durante il processo delle opzioni IP.

ipInAddressErrors Numero di datagrammi in ingresso scartati perche’ l’indirizzo IP della testata nel campo destinazione non era valido. Il conteggio include indirizzi non validi e classi di indirizzo non supportati. Per entita’ che non sono gateway IP (datagrammi non inoltratati), questo contatore include i datagrammi scartati perche’ l’ indirizzo di destinazione non era locale.

ipForwDatagrams Numero di datagrammi in ingresso per i quali questa entita’ non era la destinazIone finale. Nelle entita’ che non agiscono come gateways IP, questo contatore include solo quei pacchetti che sono stati generati attraverso questa entita’ (e il relativo processo e’ stato concluso con successo).

ipInUnknownProtos Numero di datagrammi indirizzati localmente ricevuti con successo, ma scartati a causa di protocollo sconosciuto o non supportato.

ipInDiscards Numero di datagrammi IP in ingresso scartati. Questo contatore non include i datagrammi scartati durante il processo di riassemblaggio.

ipInDelivers Numero di datagrammi in ingresso consegnati con successo al protocollo IP utente, inculso ICMP.

ipOutRequests Numero di datagrammi IP che il locale protocollo IP utente (incluso ICMP) ha fornito ad IP in seguito ad una richiesta di trasmissione. Questo contatore non include i datagrammi conteggiati in ipForwDatagram.

ipOutDiscards Numero di datagrammi IP in uscita scartati. Questo contatore non include i datagrammi scartati in attesa di riassemblaggio. Tuttavia questo contatore include i datagrammi conteggiati in ipForwDatagram se tali pacchetti incontrano questo criterio di eliminazione.

ipOutNoRoutes Numero di datagrammi IP scartati a causa di mancanza di una rotta e quindi di instradamento per la trasmissione alla loro destinazione di assemblaggio. Questo contatore include i pacchetti conteggiati in ipForwDatagram se tali pacchetti incontrano questo criterio di non-rotta.

ipReasmTimeout Tempo massimo in secondi per il quale i frammenti vengono tenuti in attesa di riassemblaggio da questa entita'.

ipReasmReqds Numero di frammenti IP ricevuti che necessitano di essere riassemblati presso questa entita'.

ipReasmOKs Numero di datagrammi IP riassemblati con successo.

ipReasmFails Numero di guasti rilevati dall' algoritmo di riassemblaggio IP. Questo non e' necessariamente un contatore di frammenti IP scartati perche' alcuni algoritmi (es. RFC- 815s) possono perdere traccia del numero di frammenti combinandoli man mano che vengono ricevuti.

ipFragOKs Numero di datagrammi IP frammentati con successo presso questa entita'.

ipFragFails Numero di datagrammi IP che non sono stati frammentati con successo presso questa entita'.

ipFragCreates Numero di frammenti di datagrammi IP generati presso questa entita'.

Gruppo Internet Control Messages Protocol

Statistiche di ingresso e di uscita

Lista ottenuta con il comando icmp stat del pacchetto ka9q nos

tcp/ip

net> icmp stat

```
( 1)icmpInMsgs                    0     (14)icmpOutMsgs                0
( 2)icmpInErrors                   0     (15)icmpOutErrors              0
( 3)icmpInDestUnreachs         0     (16)icmpOutDestUnreachs        0
( 4)icmpInTimeExcds             0     (17)icmpOutTimeExcds           0
( 5)icmpInParmProbs             0     (18)icmpOutParmProbs           0
( 6)icmpInSrcQuenchs           0     (19)icmpOutSrcQuenchs          0
( 7)icmpInRedirects               0     (20)icmpOutRedirects           0
( 8)icmpInEchos                   0     (21)icmpOutEchos               0
( 9)icmpInEchoReps              0     (22)icmpOutEchoReps            0
(10)icmpInTimestamps           0     (23)icmpOutTimestamps          0
(11)icmpInTimestampReps     0     (24)icmpOutTimestampReps       0
(12)icmpInAddrMasks             0     (25)icmpOutAddrMasks           0
(13)icmpInAddrMaskReps      0     (26)icmpOutAddrMaskReps        0
```

icmpInMsgs Numero di messaggi ICMP ricevuti dall' entita'. Il contatore include tutti quelli conteggiati da icmpInErrors.

icmpInErrors Numero di messaggi ICMP con errori tipo cattivo checksums ICMP, lunghezza sbagliata, ecc.

icmpInDestUnreachs Numero di messaggi ICMP destinazione irragiungibile ricevuti.

icmpInTimeExcds Numero di messaggi ICMP tempo scaduto ricevuti.

icmpParmProbs Numero di messaggi ICMP problemi di parametri ricevuti.

icmpInSrcQuenchs Numero di messaggi ICMP source quenchs ricevuti (mancanza di buffers).

icmpInRedirects Numero di messaggi ICMP redirect ricevuti (cambiamento di rotta verso gateway).

icmpInEchos Numero di messaggi ICMP richiesta di eco ricevuti.

icmpInEchoReps Numero di messaggi ICMP risposta ad eco ricevuti.

icmpInTimestamps Numero di messaggi ICMP richiesta dell'ora ricevuti.

icmpInTimestampReps Numero di messaggi ICMP risposta con ora ricevuti.

icmpInAddrMasks Numero di messaggi ICMP richiesta di maschera di indirizzo ricevuti.

icmpInAddrmaskReps Numero di messaggi ICMP risposta di maschera di indirizzo ricevuti.

icmpOutMsgs Numero di messaggi ICMP che questa entita' ha tentato di inviare. Il contatore include quelli conteggiati da icmpOutErrors.

icmpOutErrors Numero di messaggi ICMP che questa entita' non ha inviato a causa di problemi ICMP come la mancanza di buffers. Il valore non include gli errori causati dall' esterno di ICMP come l' incapacita' di IP di instradare il datagramma.

icmpOut DestUnreachs Numero di messaggi ICMP destinazione Irraggiungibile inviati.

icmpOutTimeExcds Numero di messaggi ICMP tempo scaduto inviati.

icmpOutParmProbs Numero di messaggi ICMP problemi di parametri inviati.

icmpOutSrcQuenches Numero di messaggi ICMP source quenchs (mancanza di buffers) inviati.

icmpOutRedirects Numero di messaggi ICMP redirezione di instradamento verso gateway inviati.

icmpOutEchos Numero di messaggi ICMP richiesta di eco inviati.

icmpOutEchoReps Numero di messaggi ICMP risposta ad eco inviati.

icmpOut TimeStamps Numero di messaggi ICMP richiesta dell'ora inviati.

icmpOutTimestampReps Numero di messaggi ICMP di risposta di ora inviati.

icmpOutAddrMasks Numero di messaggi ICMP richiesta di maschera di indirizzo inviati.

icmpOutAddrMaskReps Numero di messaggi ICMP di risposta di maschera di indirizzo inviati.

Gruppo Trasmission Control Protocol

Oggetti relativi ai valori di connessione TCP

Lista ottenuta con il comando tcp status del pacchetto ka9q nos

tcp/ip

net> tcp status

```
( 1)tcpRtoAlgorithm            4 ( 2)tcpRtoMin            0
( 3)tcpRtoMax         4294967295 ( 4)tcpMaxConn   294967295
( 5)tcpActiveOpens             0 ( 6)tcpPassiveOpens      0
( 7)tcpAttemptFails            0 ( 8)tcpEstabResets       0
( 9)tcpCurrEstab               0 (10)tcpInSegs            0
(11)tcpOutSegs                 0 (12)tcpRetransSegs       0
(14)tcpInErrs                  0 (15)tcpOutRsts           0
```

tcpRtoAlgorithm Algoritmo usato per determinare il valore di timeout uttilizzato per la ritrasmissione degli ottetti non confermati.

tcpRtoMin Valore minimo permesso da un' implementazione TCP per il valore di timeout misurato in millisecondi. La semantica piu' raffinata per gli oggetti di questo tipo dipende dall' algoritmo usato per determinare il timeout per la ritrasmissione.

tcpRtoMax Massimo valore permesso da una implementazione TCP per il timeout di ristrasmissione misurato in millisecondi. La semantica piu' raffinata per gli oggetti di questo tipo dipende dall' algoritmo usato per determinare il timeout per la ritrasmissione.

tcpMaxConn Limite sul numero totale di connessioni TCP che l' entita' puo' sopportare. Le entita' con il numero massimo di connessioni dinamico conterranno il valore -1.

tcpActiveOpens Numero di volte che le connessioni TCP hanno avuto transizioni dirette dallo stato CLOSED a SYN-SENT .

tcpPassiveOpens Numero di volte che le connessioni TCP hanno avuto transizioni dirette dallo stato LISTEN a SYN-RCVD.

tcpAttemptFails Numero di volte che le connessioni TCP hanno avuto transizioni dirette dallo stato SYN-SENT o SYN-RCVD allo stato di LOSED ed il numero di volte che le connessioni TCP hanno avuto transizioni dirette dallo stato SYN-RCV a LISTEN.

tcpEstatbResets Numero di volte che le connessioni TCP hanno avuto transizioni dirette dallo stato ESTABILISHED o CLOSED-WAIT allo stato di CLOSED .

tcpCurrEstab Numero di connessioni TCP allo stato di ESTABILSHED o CLOSED-WAIT .

tcpInSegs Numero di segmenti ricevuti, inclusi quelli ricevuti in errore. Il conteggio include i segmenti ricevuti sulle connessioni correntemente realizzate.

tcpOutSegs Numero di segmenti inviati, inclusi quelli delle sessioni correnti, ma esclusi quelli contententi solo gli ottetti ritrasmessi.

tcpRetransSegs Numero di segmenti ritrasmessi, cioe', quei segmenti TCP trasmessi che contengono uno o piu' ottetti precedentemente trasmessi.

tcpInErrs Il numero totale di segmenti ricevuti con errore ( ad esempio , cattivo TCP checksum).

tcpOutRsts Il numero totale di segmenti TCP inviati contenenti il segnale RST.

Gruppo User Datagram Protocol

Oggetti in relazione con lo strato UDP (4 OSI)

Lista ottenuta con il comando udp status del pacchetto ka9q nos tcp/ip

net> udp status

```
( 1)udpInDatagrams           0     ( 2)udpNoPorts                  0
( 3)udpInErrors              0     ( 4)udpOutDatagrams             1

udpInDatagrgams   Numero  di datagrammi consegnati dagli  utenti UDP.

udpNoPorts        Numero  di datagrammi UDP ricevuti per i quali non
```
esiste nessuna applicazione alla porta di destinazione.

udpInErrors Numero di datagrammi UDP ricevuti che non si sono potuti consegnare per ragioni diverse dalla mancanza di applicazioni alla porta di destinazione.

udpOutDatagrams Numero di datagrammi UDP inviati da questa entita'.

```
$$
```

FINE