

Specificatii Internet Protocol

Acest RFC reprezinta un standard pentru comunitatea ARPA Internet. Hosts de pe ARPA Internet adopta si implementeaza in mod obisnuit acest standard.

Introducere

Scopul Protocolului TELNET este de a furniza facilitati de comunicare compatibile in general, bi-directionale si pe 8 biti bytes. Scopul sau in primul rand este de a permite o metoda standard de realizare reciproca a interfetelor, pentru dispozitivele terminalelor si procesele terminal-orientate. Protocolul poate de asemeni sa fie folosit pentru comunicare terminal-terminal ("linking") si comunicare proces-proces (programare distribuita)

CONSIDERATII GENERALE

O conexiune TELNET este un protocol de control al transmisiei (TCP), conexiune folosita pentru a transmite date intercale cu informatii de control TELNET.

Protocolul TELNET este construit dupa trei idei principale: prima, conceptul de "Retea terminal virtuala "; a doua , principiul optiunilor negociate; si a treia o privire simetrica asupra terminalelor si proceselor.

1. Cand se stabileste pentru prima data o conexiune TELNET, fiecare final este de asteptat sa inceapa si sa se termine la un "terminal virtual de retea" sau NVT. Un NVT este un dispozitiv imaginar care furnizeaza un standard, retea-intinsa, intermediaza reprezentarea unui terminal canonic. Aceasta elimina nevoia de "server" si "user" gazde pentru a pastra informatia despre caracteristicile fiecarui alt terminal si manipularea conventiilor. Toate gazdele, deopotriva userul si serverul, mapeaza caracteristicile si conventiile dispozitivelor lor locale astfel ca apar ca trateaza cu un NVT in afara retelei si fiecare poate sa-si asume o mapare similara de partea cealalta. NVT intentioneaza sa modifice balanta intre a fi supra restrictionat (sa nu furnizeze gazduire pentru un vocabular suficient de bogat pentru maparea in interiorul seturilor lor de caractere locale) fiind peste inclusiv (penalizand uzerii cu terminale modeste).

Nota: "Userul" gazda este gazda la care terminalul fizic este normal atasat, iar "serverul" este gazda care furnizeaza astfel de servicii in mod normal. Ca un alt punct de vedere aplicabil chiar la comunicatiile terminal -la-terminal sau proces-la-proces, "userul" gazda este gazda care initiaza comunicarea.

2. Principiile optiunilor negociate iau cunostinta cu faptul ca multe gazde vor dori sa furnizeze servicii aditionale peste si deasupra celor disponibile fara un NVT, iar multi utilizatori vor avea terminale sofisticate si vor vrea sa aiba mai degraba servicii elegante decat minimale. Independent de el, dar structurat in ce priveste TELNET Protocol exista variate "optiuni" care vor fi santionate si vor fi folosite cu structuri "DO, DON'T, WILL, WON'T" (discutate mai jos) pentru a permite userului si serverului sa cada de acord in ce priveste utilizarea unor conventii mult mai elaborate (sau poate doar diferite) pentru conexiunea lor TELNET. Aceste optiuni pot include schimbari ale setului de caractere, modul echo, etc.

Strategia de baza pentru a seta folosirea optiunilor este de a avea initiat la una din parti (sau la ambele) o cerere pentru care unele optiuni au efect; daca este respinsa aspectul asociat al conexiunii ramane (cel) specificat pentru un NVT. Cu siguranta, o parte poate refuza intotdeauna o cerere de activare (enable) si trebuie sa fie pregatita pentru a sustine(support) NVT-ul.

Sintaxa optiunii de negociere a fost setata astfel incat daca ambele parti fac o cerere simultana pentru o optiune, fiecare va vedea cererea celuilalt ca pe o protocol al uneia proprii.

3.Simetria sintaxei de negociere poate conduce in mod potential la o bucla protocol nedeterminate- fiecare parte vazand comenzile care vin nu ca un protocol ci ca o noua cerere care trebuie sa fie recunoscuta. Pentru a preveni astfel de bucle, sunt predominante urmatoarele reguli:

- a. Partile pot numai sa ceara o schimbare in statutul optiunilor; ex. o parte poate sa nu trimita o cerere numai sa anunte in ce mod se afla.
- b. Daca o parte primeste ceva ce pare sa fie o cerere de intrare intr-o stare in care deja de afla, cererea poate sa nu fie recunoscuta (*acknowledged*). Acest non-raspuns este esential pentru a preveni in negociere buclele fara sfarsit . Este necesar ca un raspuns la o cerere sa fie trimis pentru o schimbare de stare -chiar daca starea nu este schimbata.
- c. Ori de cate ori o parte trimite o optiune comanda catre o a doua parte, pentru o cerere sau un protocol si folosesc faptul ca optiunea va avea un efect oarecare asupra procesarii datelor ce au fost trimise de catre prima parte spre a doua, atunci comanda trebuie sa fie inserata in fluxul de date la punctul la care se doreste sa aiba efect. (Se va observa ca uneori exista posibilitatea ca transmisia unei cereri si primirea raspunsului sa dureze mai mult, un timp considerat negativ. Astfel un host poate dori sa salveze datele in buffer, dupa ce a cerut o optiune, pana cand invata daca cererea este acceptata sau respinsa, in masura sa ascunda utilizatorului "perioada de incertitudine").

Cererile optiuni sunt ca un suflu inapoi si inainte atunci cand se stabileste pentru prima data o conexiune TELNET, iar fiecare parte incerca sa obtina cel mai bun serviciu posibil din partea celuilalt. Dincolo de aceasta, optiunile pot fi utilizate pentru a modifica dinamic caracteristicile pentru a se adapta schimbarilor conditiilor locale. De exemplu, NVT, asa cum vom explica ulterior, foloseste o disciplina de transmisie care se adapteaza multor aplicatii "caracter la un moment" cum sunt NLS. Un server poate alege sa-si dedice extra procesorul peste cererea stabilita pentru un "caracter la un moment" atunci cand era adaptat pentru procesorul local si ar fi negociat cea mai potrivita optiune.

Oricum, mai degraba decat sa fie permanent incarcat cu extra procesari, poate sa manevreze (ex. sa negocieze) inapoi cu NVT-ul atunci cand nu mai este necesar un control in detaliu.

Este posibil ca pentru cererile initiale de procese sa stimuleze o bucla de cereri

infinita, daca procesul raspunde la un refuz, pur si simplu cerand din nou aceeasi optiune. Pentru a preveni repetarea unor astfel de bucle, cererile respinse ar trebui sa nu fie repetate pana cand ceva se schimba. Operational, aceasta poate sa insemne ca procesul ruleaza un program diferit, sau ca userul a dat o alta comanda, sau orice altceva poate avea sens in contextul procesului dat si al optiunii date.

Este o regula buna aceea ca o noua cerere a aceleiasi optiuni poate sa revina numai ca un rezultat al unei informatii urmatoare de la celalalt capat al conexiunii sau cand este ceruta de o interventie locala umana.

Optiunile proiectantilor nu ar trebui sa se simta constranse de o oarecare limitare a sintaxei disponibile pentru optiunile de negociere. Intentia unei sintaxe simple este de a usura optiunile- deoarece in mod corespunzator este la fel de usor sa le ignori. Daca unele optiuni partiulare necesita o structura de negociere mai bogata, posibil "DO, DON'T, WILL, WON'T" tinta propriu-zisa este sa foloseasca "DO, DON'T, WILL, WON'T" pentru a stabili ca ambele parti inteleg optiunile, si ca odata ce acest lucru este realizat poate fi folosita liber o sintaxa mult mai exotica. De exemplu, o parte poate sa trimita o cerere catre o alta lungime de linie(stabilita). Daca este acceptata, atunci poate fi folosita o sintaxa diferita pentru negocierea actuala a lungimii de linie- - cum ar fi sub-negocierea care ar include campurile pentru permisie minima, permisie maxima si lungime de linie dorita. Conceptul important este acela care expandeaza negocierile care nu ar incepe vreodata fara unele negocieri anterioare (standard), care au stabilit ca ambele parti sunt capabile sa analizeze sintaxa expandata.

In concluzie, este trimis WILL XXX, catre partea cealalta, pentru a indica dorinta acesteia (oferta) pentru a incepe executia optiunii XXX, DO XXX si DON'T XXX in acord cu confirmarile pozitive si negative; in mod similar DO XXX este trimis sa indice o dorinta (cerere) catre cealalta parte (ex. destinatarul lui DO) incepe executia optiunii XXX, WILL XXX si WON'T XXX in concordanta cu confirmarile pozitive si negative. Deoarece NVT inseamna ca este permis atunci cand nici o optiune nu este activa, raspunsurile DON'T si WON'T garanteaza ca parasesc conexiunea intr-o stare care ambele parti o pot trata. Astfel toate gazdele isi pot implementa procesele lor TELNET pentru a fi total nepregatite pentru optiunile care nu sunt suportate, prin simpla respingere(refuz) a oricarei cereri optiuni care nu poate fi inteleasa.

Atat cat este posibil, protocolul TELNET a fost facut simetric client-server, astfel ca sunt usor si natural acoperit cazurile (legatura) client-client si server-server(procese de cooperare). Este de dorit dar nu neaparat necesar ca optiunile sa urmeze mai departe acest plan in orice caz, este confirmat in mod explicit ca simetria este un principu de operare mai degraba, decat o regula greu de incalcat.

Ca document ajutorator, "Specificatiile optiunii TELNET", ar trebui consultat pentru informatii asupra procedurii de stabilire de noi optiuni.

TERMINALUL RETEA VIRTUALA

Terminalul Retea Virtuala (NVT) este un dispozitiv cu caracter bi-directional. NVT

are un printer si o tastatura. Printerul raspunde la datele care intra iar tastatura produce datele de iesire care sunt transmise prin conexiunea TELNET si daca se doreste un "ecou", catre printer-ul NVT. Nu este de asteptat ca "ecourile" sa traverseze reseaua (de altfel optiunile exista pentru a permite "o comunicare" pentru modul ecou al operatiei, nu este necesara nici o gazda pentru a implementa aceasta optiune). Setul de coduri este un USASCII pe 7 biti, intr-un camp de 8 biti, exceptand pe cel modificat aici.Orice conversie de cod si consideratii de timp sunt probleme locale si nu afecteaza NVT.

TRANSMISIA DE DATE

Cu toate ca conexiunea TELNET prin retea este intrinsec full duplex, NVT este vazut ca un dispozitiv half-duplex operand in modul linie-buffer. Aceasta este in afara de cazul si pana cand sunt negociate optiunile din contra, urmatoarele conditii implicite se refera la transmitia datelor prin conxiunea TELNET:

1) Astfel cum permite disponibilitatea unui buffer local, datele ar putea fi acumulate in host, acolo unde este generata, pana cand este gata pentru transmisie, o linie completa de date, sau pana cand intervine un semnal explicit definit local. Aceste semnal poate fi generat deopotriva de un proces sau de un utilizator uman.

Motivatia pentru aceasta regula este costul ridicat, pentru unele gazde, sau intreruperile de processare a inputurilor retelei, cuplate cu specificatiile NVT implicite astfel ca "ecourile" nu traverseaza reseaua. Astfel este ceva normal sa stocam in buffer unele date si sursele lor. Multe sisteme incep procesarea la sfarsitul fiecarei linii de intrare (chiar si printerele sau cartelele perforate lucreaza in acest fel) astfel transmitia ar putea fi oprita la sfarsitul unei linii. Pe de alta parte, un utilizator sau un proces poate considera uneori ca este necesar sau dezirabil sa furnizeze date care nu se termina la sarsitul unei linii; de aceea cei care implementeaza sunt avertizati sa furnizeze metode care sa semnalizeze local, ca toate datele din buffer pot fi transmise imediat.

2) Cand un proces a terminat de transmis date catre un printer NVT si nu are nici un input verificat din partea tastaturii NVT pentrtru procesele urmatoare (ex., cand un proces la sfarsitul unei conexiuni TELNET nu poate incepe fara inputul celuilalt capat), procesul trebuie sa transmita catre TELNET comanda Go Ahead (GA).

Aceasta regula nu intentioneaza sa ceara unei comenzi TELNET GA sa fie trimisa de la un terminal la sfarsitul fiecarei linii, daca serverul gazda nu cere in mod normal un semnal special (in plus la sfarsit-de -linie sau alte caractere definite local) pentru a incepe procesarea.

Mai degraba, TELNET GA este proiectata pentru a ajuta utilizatorul local sa opereze fizic cu o jumătate a unui terminal duplex, care are o tastatura "blocabila" asa cum este IBM 2741. O descriere a acestui tip de terminal poate ajuta la explicarea modului de utilizare al comenzii GA.

Conexiunea terminal-computer este intotdeauna sub controlul ori al utilizatorului ori

al computerului. Nici unul nu poate prelua controlul asupra celuilalt; mai degraba sfarsitul controlului trebuie sa abandoneze controlul sau explicit (categoric). La sfarsitul terminalului, hardul este construit astfel incat sa abandoneze(predea) controlul de fiecare data cand o "linie " este terminata (ex., atunci utilizatorul tasteaza cheia "noii linii"). Cand aceasta intervine, computerul (local) proceseaza datele input, decide daca outp-utul poate fi generat, iar daca nu returneaza controlul terminalului. Daca output-ul poate fi generat, controlul este retinut de computer pana cand sunt transmise toate output-urile.

Este evidenta dificultatea folosirii acestui tip de terminal pentru retea. Computerul "local" nu mai este capabil sa decida daca sa retina controlul dupa ce primeste un semnal de sfarsit de linie sau nu; aceasta decizie poate fi luata numai de computerul "la distanta" care proceseaza datele. De aceea, comanda TELNET GA furnizeaza un mecanism pentru care (serverul) computerul "la distanta" poate semnaliza (userul) computerul "local" astfel ca este timp sa treaca controlul catre userul terminalului. Ar putea fi transmis in acesti timpi si numai in acestia atunci cand userul poate avea controlul asupra terminalului.Observam ca transmitia prematura a comenzii GA poate avea ca rezultat blocarea output-ului, pana cand userul este gata sa accepte ca sistemul de transmisie s-a oprit si de aceea nu va putea intoarce linia.

Anteriorul desigur nu aplica directia de comunicare de la user la server. In aceasta directie, GA poate fi transmisa in orice moment, dar nu mai este necesar. De asemeni, daca conexiunea TELNET este folosita si de comunicarea proces-proces, nu este necesar ca GA sa fie transmisa in ambele directii. Daca o gazda vrea sa sustina o comunicatie terminal-terminal se sugereaza ca gazda sa furnizeze userului impreuna cu semnificatiile semnalizarii manuale ca este timpul ca GA sa fie trimisa prin conexiunea TELNET; aceasta, oricum, nu este o cerere de implementare a procesului TELNET.

Observam ca simetria modelului TELNET necesita un NVT la fiecare capat al conexiunii TELNET, cel putin conceptual.

PREZENTAREA STANDARD A FUNCTIILOR DE CONTROL

Asa cum am afirmat la inceputul acestui document, scopul de baza al protocolului TELNET este furnizarea unei interfete standard pentru dispozitivul terminalului si a proceselor orientate-terminal prin retea. Primele experiente cu acest tip de interconexiune au aratat ca cele mai multe servere implementeaza anumite functii, dar metodele invocate de aceste functii difera considerabil. Pentru un utilizator uman care interactioneaza cu cateva sisteme de servere, aceste diferente aduc o puternica confuzie. De aceea, TELNET, defineste o reprezentare standard pentru cinci din aceste functii, dupa cum este descris mai jos. Aceste reprezentari standard au standard, dar nu si cerinte, semnificatii (cu exceptia functiei de Intrerupere a Procesului(IP) care poate fi ceruta de alte protocoale care folosesc TELNET); de aceea, un sistem care nu furnizeaza aceasta functie utilizatorilor locali, nu trebuie sa o furnizeze utilizatorilor din retea si pot sa trateze reprezentarea standard pentru functie ca fiind fara-operatii. Pe de alta parte, un sistem care furnizeaza functia catre un

utilizator local este obligat sa furnizeze aceeasi functie catre un utilizator din retea care transmite reprezentarea standard pentru functie.

Intreruperea procesului (IP)

Multe sisteme furnizeaza o functie care suspenda, intrerupe, renunta, sau termina operarea unui proces utilizator. Aceasta functie este utilizata frecvent atunci cand userul crede ca procesul sau este o bucla fara sfarsit sau cand un proces este activat inadvertent. IP este reprezentarea standard pentru a invoca aceasta functie. Poate fi caracterizata de catre cei care o implementeaza, ca IP poate fi ceruta de catre alte protocoale care folosesc TELNET si de aceea ar trebui implementata daca sunt necesare si alte protocoale.

Output-ul Abort (AO)

Multe sisteme furnizeaza o functie care permite unui proces ce genereaza un output, sa ruleze pana la capat (sau sa atinga acelasi punct de stop pe care l-ar atinge daca ar rula pana la capat) insa fara sa transmita output-ul catre userul terminalului. In afara de aceasta, aceasta functie, in mod obisnuit, sterge orice output deja produs dar care nu a fost inca printat (sau afisat) pe terminalul userului. AO este reprezentarea standard pentru invocarea acestei functii. De exemplu, unele subsisteme pot accepta in mod normal o comanda din partea userului, trimit ca raspuns un long text string la terminalul userului, iar in final semnalul ca este gata sa accepte urmatoarea comanda prin trimiterea unui caracter "prompt" (precedat de <CR><LF>) catre terminalul userului. daca AO a fost primit in timpul transmisiei sirului text, o implementare buna ar trebui sa suprima remainder-ul sirului text, dar sa transmita caracterul prompt si precedentele <CR><LF>. (Acest lucru este posibil sa distinga actiunea care ar fi putut fi facuta daca era primita de IP; IP ar putea cauza suprimarea unui string text si o iesire din subsistem.)

Ar trebui luat in considerare de catre sistemele server care furnizeaza aceasta functie, ca ar putea exista buffere externe ale sistemului (in retea si gazda utilizatorului local) care ar putea fi sterse; calea cea mai potrivita de a face acest lucru este sa transmita semnalul "Synch" (descrie mai jos) catre sistemul user.

Are You There (AYT)

Multe sisteme au o functie care furnizeaza utilizatorului o evidenta vizibila (ex. printabila) ca sistemul este inca activ si ruleaza. Aceasta functie poate fi invocata de catre utilizator atunci cand sistemul este neasteptat de "tacut" pentru o perioada mare de timp, din cauza unei lungimi de calcul care nu a fost anticipata (de catre utilizator) sau executia neasteptat de incheata a unui sistem, etc. AYT este reprezentarea standard pentru invocarea acestei functii.

Caracterul de stergere (EC)

Multe sisteme furnizeaza o functie ce sterge ultimul caracter precedent nesters sau "pozitia de print" * de la fluxul de date furnizat de utilizator. Aceasta functie este folosita in mod obisnuit pentru a edita un input de la tastatura cand se fac greseli de

tastare. EC este reprezentarea standard pentru invocarea acestei functii.

*NOTA: O "pozitie de print" poate contine cateva caractere care sunt rezultatul traversarilor (overstrike), sau a unor secvente cum sunt <char1> BS <char2>...

Linia de stergere (EL)

Multe sistem furnizeaza o functie care sterge toate datele in "linia" curenta de input. EL este reprezentarea standard pentru invocarea acestei functii.

SEMNALUL TELNET "SYNCH"

Cele mai multe sisteme de partajare a timpului furnizeaza mecanisme care permit unui user terminal sa recastige controlul asupra unui proces "dezertor" (care nu mai poate fi controlat); functiile IP si AO descrise mai sus sunt exemple ale acestor mecanisme.

Astfel de sisteme, cand sunt utilizate local, au acces la toate semnalele furnizate de utilizator, daca acestea sunt caractere normale sau semnale speciale "out of band" cu sunt cele furnizate de cheia teletip "BREAK" sau de cheia IBM 2741 "ATTN". Acest lucru nu este in mod necesar adevarat cand terminalul este conectat la sistemul din retea; mecanismul de flow control al retelei poate determina ca un astfel de semnal sa fie depozitat oriunde in buffer, de exemplu in host-ul utilizatorului.

Pentru a contacta aceasta problema, a fost introdus mecanismul TELNET "Synch". Un semnal Synch consta intr-o notificare TCP urgenta, cuplata cu comanda TELNET, DATA MARK. Notificarea urgenta, care este utilizata pentru a invoca manipularea speciala a fluxului de date care il primeste. In acest mod, fluxul de date este scanat imediat pentru semnale "interesante" cum sunt definite mai jos, si resping datele care intervin. Comanda TELNET, DATA MARK(DM) este marca de sincronizare in fluxul de date care arata ca orice semnal special a intervenit deja si destinatarul poate sa se intoarca la procesarea normala a fluxului de date.

Semnalul Synch este trimis via TCP cu un flag cu setarea Urgent si cu DM ca ultimul (sau singurul) octet de date.

Cand sunt trimise succesiv cateva semnale Synchs, notificarile urgente pot fi unite. Nu este posibil sa numaram urgentele, cand numarul primirilor va fi mai mic sau egal cu numarul trimiterilor. In mod normal DM este fara operatii; in modul urgent, semnalizeaza sfarsitul procesului urgent.

Daca TCP indica sfarsitul datelor urgente inainte sa fie gasit DM, TELNET poate continua manipularea speciala a fluxului de date pana cand este gasit DM.

Daca TCP indica mai multe date urgente dupa ce este gasit DM, poate fi din cauza unei subsecvente Synch. TELNET poate continua manipularea speciala a fluxului de date pana cand este gasit DM.

Sunt definite ca semnale "interesante": reprezentarile TELNET standard pentru IP,

AO si AYT (dar nu si EC si EL); reprezentarile standard analoage locale (daca exista); toate celelalte comenzi TELNET; alte semnale definite care pot fi actionate fara a intarzia scanarea fluxului de date.

Cand un efect al mecanismului SYNCH este respingerea tuturor caracterelor esentiale (exceptand comenzile TELNET) intre emitatorul semnalului Synch si destinatarii lui, acest mecanism este specificat ca o cale standard pentru a sterge calea datelor atunci cand se doreste. De exemplu, daca un utilizator al unui terminal cauzeaza un AO pentru a fi transmis, serverul care primeste AO (daca furnizeaza functia tuturor) ar putea returna semnalul Synch utilizatorului.

In sfarsit, daca este necesare o notificare TCP urgenta la nivelul TELNET, ca un semnal out-of-band, alte protocoale care ar putea folosi TELNET-ul, ar putea necesita o comanda TELNET care poate fi vazuta ca un semnal out-of-band la un nivel diferit.

Prin conventie, secventa [IP, Synch] poate fi folosita ca un semnal. De exemplu, presupunem ca un alt protocol, care utilizeaza TELNET, defineste un sir de caractere STOP in mod analog cu comanda TELNET , AO. Ne putem imagina ca dorinta utilizatorului acestui protocol este un server care sa proceseze sirul STOP, dar conexiunea este blocata deoarece serverul proceseaza alta comanda. Userul poate sa instruiasca sistemul sa:

1. Trimite in caracter TELNET IP;
2. Trimite o secventa TELNET SYNC, care sa:
Trimite Data Mark (DM) ca singurul caracter intr-o operatie de transmisie in modul TCP urgent.
3. Trimite sirul de caractere STOP; si
4. Trimite celalalt protocol analog TELNET DM, daca exista.

Utilizatorul (sau procesul care ruleaza in partea lui) trebuie sa transmita secventa TELNET SYNCH de la pasul 2 de mai sus, pentru a se asigura ca TELNET IP trece prin interpretarea serverului TELNET.

Mesajul urgent poate activa procesul TELNET; IP-ul va activa urmatorul nivel superior al procesului.

PRINTERUL NVT SI TASTATURA

Printerul NVT are o latime nespecificata a carriage-ului si a lungimii paginii si poate produce reprezentari ale tuturor celor 95 de semne grafice USASCII (coduri de la 32 la 126). Peste 33 coduri de control USASCII (de la 0 la 31 si 127), si cele 128 coduri neacoperite (128 pana la 255), urmatoarele au rolul special al printerului NVT:

NUME	COD	SEMNICIFICATIE
NULL (NUL)	0	Nici o operatie
avanseaza linie de printare, Line Feed (LF)	10	Muta printorul catre urmatoarea linie pastrand aceeasi pozitie orizontala.
return de car (CR)	13	Muta printorul la marginea stanga a liniei curente.

In plus, urmatoarele coduri ar putea fi definite, dar nu in mod obligatoriu, au efect asupra printorului NVT. Nici chiar la sfarsitul conexiunii TELNET nu putem garanta ca cealalta parte va face, sau va fi facut deja, orice fel de actiune particulara de primire sau transmitere ale acestora:

BELL (BEL) (care NU va muta capul printorului).	7	Produce un semnal vizual sau audio
Back Space (BS) fata de marginea din stanga.	8	Muta capul printorului cu un caracter
Tab Orizontal (HT) stop tab orizontal. Ramane nespecificat cum cealalta sunt localizate aceste stopuri de tab.	9	Muta printorul la urmatorul parte determina sau stabileste unde
Tab Vertical (VT) vertical. Ramane nespecificat cum cealalta sunt localizate aceste stopuri de tab.	11	Muta printorul la urmatorul stop tab parte determina sau stabileste unde
Form Feed (FF) urmatoare, pastrand aceeasi pozitie orizontala.	12	Muta printorul la inceputul paginii

Toate codurile care raman nu determina nici o actiune din partea printorului NVT .

Secventa "CR LF", dupa cum este definita, va determina NVT sa fie positionat la marginea din stanga al urmatoarei linii de printare (asa cum face de exemplu, secventa "LF CR"). Ori de cate ori, multe sisteme si terminale nu trateaza CR si LF independent si vor trebui sa depuna efort pentru a simula efectul lor. (de ex. unele terminale nu au CR independent de LF , dar la astfel de terminale este posibil sa simulam CR prin backspacing.)

De aceea, secventa "CR LF" trebuie sa fie tratata ca un singur caracter "linie noua" si folosita cand este ceruta combinatia lor; secventa "CR NUL" trebuie sa fie utilizata

cand carul se doreste de fapt returnul singur al carului; si caracterul CR trebuie evitat in alte contexte.

Aceasta regula asigura sistemele care trebuie sa decida asupra folosirii unei functii "linie noua" sau a unui multiplu backspace, daca fluxul TELNET contine un caracter care urmeaza un CR ce va permite o decizie rationala.

Observati ca "CR LF" sau "CR LF" sunt necesare in ambele directii, (in modul ASCII implicit), pentru a conserva simetria modelului NVT. Chiar daca este cunoscut in unele situatii (ex., cu ecou la distanta si suprimarea optiunilor) despre caractere ca nu au fost trimise de fapt la un printer, in nici un caz de dragul consistentei, protocolul necesita un NUL care sa fie inserat dupa un CR (in absenta optiunilor de negociere care explicitarea specifica de altfel) ar putea fi scoase ca prioritate prin aplicarea unui set de mapare a caracterelor locale.

Tastatura NVT are taste sau combinatii de taste, sau secvente de chei, pentru generarea tuturor celor 128 coduri USASCII. Observati ca de fapt multe nu au nici un efect asupra printerului NVT, tastatura NVT este capabila sa le genereze.

In plus pentru aceste coduri, tastatura NVT poate fi capabila sa genereze urmatoarele coduri aditionale care, cu exceptiile observate, au sensuri definite, dar nu neaparat necesare. Semnificatiile codului existent pentru aceste "caractere" sunt in sectiunea comenzilor TELNET, deoarece ele sunt vazute ca fiind in anumit sens, generic si disponibile cand fluxul de date este interpretat ca fiind alt set caracter.

Synch

Aceasta cheie permite utilizatorului sa stearga calea datelor sale catre partea cealalta. Activarea acestei chei cauzeaza un DM (vezi sectiunea comenzi) pentru a fi trimisa in fluxul de date si avand asociata cu ea o notificare TCP urgenta. Perechea DM-Urgent are semnificatia definita anterior.

Break (BRK)

Acest cod este furnizat deoarece exista un semnal din afara setului USASCII care are in mod obisnuit o semnificatie locala in mai multe procese. Se intentioneaza sa se indice ca Break Key sau Attention Key a fost atinsa. Observam oricum, ca se intentioneaza sa se furnizeze al 129-lea cod pentru sistemele care necesita asta, nu ca un sinonim pentru reprezentarea standard IP.

Intreruperea Procesului (IP)

Suspenda, intrerupe, renunta sau termina procesul pentru care este conectat NVT. De altfel, o parte din semnalul out-of-band pentru alte protocoale care folosesc TELNET.

Abort Output (AO)

Permite procesului curent sa (apara ca) ruleaza pana la capat, dar nu trimite output-ul sau catre user. De asemeni, trimite un Synch utilizatorului.

Are You There (AYT)

Trimite inapoi catre NVT o evidenta vizibila (ex. printabila) cum ca AYT a fost primit.

Caracter de Stergere (EC)

Destinatarii ar trebui sa stearga ultimul caracter care il precede pe cel nesters sau "pozitia de print" din fluxul de date.

Linia de stergere (EL)

Destinatarii trebuie sa stearga caracterele din fluxul de date inapoi, dar fara sa includa, ultima secventa "CR LF" trimisa prin conexiunea TELNET.

Spiritul acestei "extra" chei si de asemeni efectele formatarii printerului, este ca ele ar trebui sa reprezinte o extensie naturala a maparii care deja a fost realizata dintr-una "NVT" in una "locala". La fel cu byte-ul 68 de date NVT (104 octal) ar putea fi mapati oriunde exista codul local pentru litera mare "D", astfel caracterul EC ar putea fi mapat oriunde exista functia locala "Sterge Caracter". Mai departe, asa cum maparea pentru 124 (174 octal) este oarecum arbitrara intr-un mediu in care nu exista caracterul "bara verticala", caracterul EL poate avea o mapare oarecum arbitrara (sau deloc) daca nu exista nici o facilitate "Linie de Stergere". In mod similar pentru formatare: daca terminalul de fapt are un "Tab Vertical", atunci maparea pentru VT este evidenta si numai cand terminalul nu are un tab terminal ar putea avea un efect imprezibil asupra lui VT.

STRUCTURA DE COMANDA A TELNETULUI

Toate comenzile TELNET constau din cel putin o secventa de bytes: "Interpretorul de comenzi"(IAC) secventa de caractere urmata de codul pentru comanda. Comenzile care trateaza negocierea optiunilor sunt de fapt secvente de 3 biti, bitul al treilea fiind codul pentru optiunea referita. Acest format a fost ales astfel incat sa utilizeze mult mai comprehensiv "spatiul de date"- prin negociere din NVT de baza-coliziuni de biti de date cu valorile rezervate ale comenzilor fiind minimizate, toate coliziunile de acest fel reclamand inconvenientul si ineficienta "secventierii" byte-ilor de date in fluxuri. Cu setarea curenta, numai IAC are nevoie sa fie dublat pentru a fi trimis ca date, iar celelalte 255 de coduri pot trece in mod transparent.

Comenzile TELNET sunt cele ce urmeaza.Observati ca aceste coduri si secvente de cod au rolul indicat cand sunt precedate imediat de un IAC.

NUME	COD	SEMNIFICATIE
SE	240	Sfarsitul parametrilor de subnegociere.
NOP	241	Nici o operatie.
Data Mark	242	Fluxul unei portii de date a unui Synch.
		Acesta poate fi intotdeauna insotita de o notificare TCP urgenta.

Break	243	Caracter NVT, BRK.
Intreruperea Procesului	244	Functia IP.
Abort Output	245	Functia AO.
Are You There	246	Functia AYT.
Sterge caracter	247	Functia EC.
Sterge linie	248	Functia EL.
Go ahead	249	Semnalul GA.
SB	250	Indica ca ceea ce urmeaza este o
subnegociere a optiunii indicate.		
WILL(optiune de cod)	251	Indica dorinta de a incepe executia
sau confirmarea ca esti in timpul executiei		
		optiunii indicate.
WON'T (optiune de cod)	252	Indica refuzul de a incepe sau
continua executia optiunii indicate.		
DO (optiune de cod)	253	Indica cererea celeilalte parti de a
intra in executie sau confirmarea pe care o astepti		
		din partea celuilalt cum ca executa
optiunea indicata.		
DON'T (optiune de cod)	254	Indica cererea celeilalte parti pentru
a opri executia sau confirmarea ca nu trebuie sa		
		astepti ca cealalta parte sa mai
execute optiunea indicata.		
IAC	255	Byte de date 255.

STABILIREA CONEXIUNII

Conexiunea TELNET TCP se stabileste intre portul U al utilizatorului si portul L al serverului. Serverul asteapta conexiunile la binecunoscutul sau port L. Daca conexiunea TCP este full duplex si identificata de catre perechea de porturi, serverul se poate angaja in mai multe conexiuni simultane implicand portul sau L si un port diferit U pentru user.

Asignarea portului

Cand este folosit pentru accesul unui user aflat la distanta de serviciile hostului (ex., accesul la un terminal aflat la distanta) acest protocol este portul 23 asignat serverului (27 in octal). acesta este L=23.