Chapitre IV Protocoles De La Couche Application - TELNET

Cours Réseaux Avancés M1 SIC-IA

Ilyas Bambrik

2024

Table des matières

I - **TELNET (TELetype NETwork)** 3

II - **Fonctionnement** 4

III - **NVT (Network Virtual Terminal)** 5

IV - **Négociation d'options** 7

Références 8

TELNET (TELetype NETwork)

I

- Lors des débuts d'internet, l'accès à distance (*remote login*) été l'une des premières applications souhaitées pour permettre aux utilisateurs de contrôler des machines à distance. Par exemple, les serveurs peuvent exécuter plusieurs programmes de divers utilisateurs simultanément. Mais, un seul utilisateur peut être présent physiquement devant le serveur;
- L'idée de base est simple : concevoir un protocole permettant à l'utilisateur de connecter à une machine distante et de contrôler celle-ci. Comme si l'utilisateur été physiquement assis devant la machine pour taper ses commandes ;
- Le problème rencontré à cette époque été l'hétérogénéité entre l'encodage des fabricants de logiciels / matériel (surtout pour les systèmes d'exploitation). Par exemple, une interruption de processus sur Windows est représentée par la séquence *CTRL+ALT+SUPPR*, alors que interruption de processus sur Linux est représentée par *CTRL+C*;
- TELNET est un simple protocole applicatif qui a été développé pour résoudre ce problème ;

Figure IIX.1. Telnet

Fonctionnement

II

TELNET fonctionne avec l'architecture client /serveur et utilise TCP pour assurer la livraison des données entre les deux parties (commandes transmis du client vers le serveur, résultats d'exécution transmis du serveur vers le client). Généralement, ce protocole utilise le port TCP 23 ;

Le protocole TELNET repose sur deux éléments :

- *NVT* (*Network Virtual Terminal*) et commandes de contrôle unifiés;
- Négociation des options ;

NVT (Network Virtual Terminal)

III

- Vu l'hétérogénéité des systèmes d'exploitations, TELNET utilise un clavier virtuel standard pour encoder les échanges entre le client et le serveur. Cette norme est appelée NVT ;
- Lorsque le client saisie ça commande, cette commande est convertie vers l'équivalent NVT avant d'être transmis vers le serveur. Ensuite, le serveur converti les données reçus sous format NVT vers son format spécifique pour les traiter;
- Ceci évite que touts les clients et serveurs TELNET ne soit conscients des encodages possibles de d'autres systèmes d'exploitation(ce qui complique énormément l'implémentation de ce protocole);

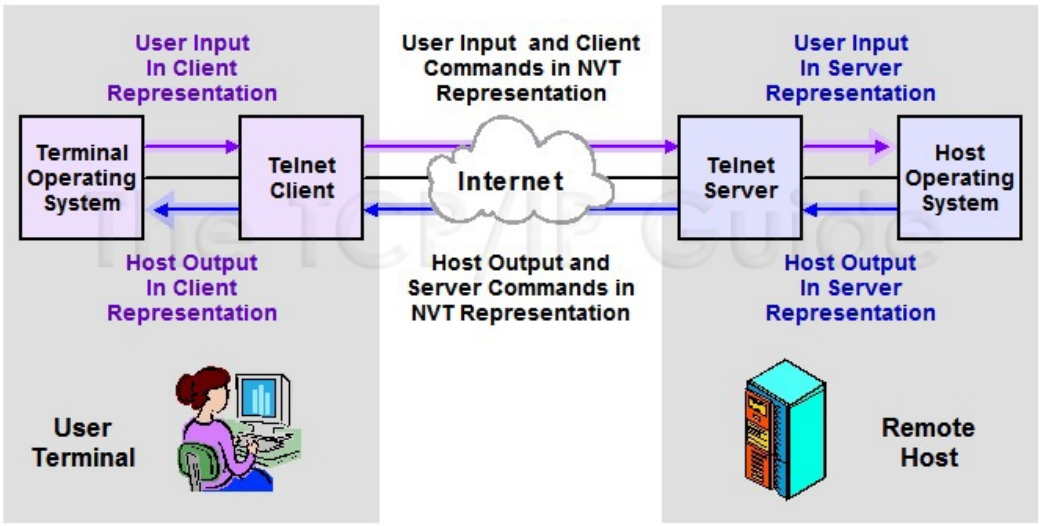

Figure IIX.2. NVT[3*]

NVT, qui doit être utilisé par toute application TELNET est basé sur le code ASCII :

- 95 caractères affichables [du 32 jusqu'à 126] sur écran (dits *"printables"*) ;
- NVT spécifie comment trois caractères de contrôle qui doivent être interprétés de manière unifiée par n'importe quel Serveur / Client TELNET [*NULL, LF, CR*];
- Cinq autres caractères sont optionnels, et si implémentés, ils doivent être conformes à la spécification de NVT ;

Pour se déplacer au début de la ligne suivante, la séquence *CR +LF* est utilisée ;

ASCII Value (Decimal)	ASCII Character Code	ASCII Character	Description	Support Optional / Mandatory
0	NUL	Null	No operation (no effect on output).	Mandatory
7	BEL	Bell	Produces an audible or visible signal on the output without moving the print head. This notification may be used to get the user's attention, as in the case of an error.	Optional
8	BS	Back Space	Moves the print position one character to the left.	Optional
9	HT	Horizontal Tab	Moves the printer to the next horizontal tab stop. The standard does not specify how devices agree on tab stop positions; this can be negotiated using Telnet options.	Optional
10	LF	Line Feed	Moves the printer to the next line, keeping the print position the same.	Mandatory
11	VT	Vertical Tab	Moves the print line to the next vertical tab stop. As with the "HT" character, devices must use an option to come to an agreement on vertical tab stop positions.	Optional
12	FF	Form Feed	Moves the printer to the top of the next page (or on a display, clears the screen and positions the cursor at the top.)	Optional
13	CR	Carriage Return	Moves the printer to the left margin of the current print line.	Mandatory

Figure IIX.3. Caractères normalisés[3*]

- Pour contrôler le déroulement de la session TELNET, NVT défini des caractères virtuels qui représentent des commandes ;
- Pour qu'un octet soit interprété comme une commande, celui-ci doit être précédé par un octet IAC (*Interpret As Command == Code FF ou 255*) ;
- Pour transmettre un octet avec la valeur FF sans qu'il soit interprété comme IAC, deux octets de valeur Fl doivent être transmis ;

Command Byte Value (Decimal)	Command Code	Command	Description
240	*SE*	Subnegotiation End	Marks the end of a Telnet option subnegotiation, used with the *SB* code to specify more specific option parameters. See the topic on Telnet options for details.
241	*NOP*	No Operation	Null command; does nothing.
242	*DM*	Data Mark	Used to mark the end of a sequence of data that the recipient should scan for urgent Telnet commands. See the topic on Telnet interrupt handling for more information.
243	*BRK*	Break	Represents the pressing of the "break" or "attention" key on the terminal.
244	*IP*	Interrupt Process	Tells the recipient to interrupt, abort, suspend or terminate the process currently in use.
245	*AO*	Abort Output	Instructs the remote host to continue running the current process, but discard all remaining output from it. This may be needed if a program starts to send unexpectedly large amounts of data to the user.
246	*AYT*	Are You There	May be used to check that the remote host is still "alive". When this character is sent the remote host returns some type of output to indicate that it is still functioning.
247	*EC*	Erase Character	Instructs the recipient to delete the last undeleted character from the data stream. Used to "undo" the sending of a character.
248	*EL*	Erase Line	Tells the recipient to delete all characters from the data stream back to (but not including) the last end of line ("CR"+"LF") sequence.
249	*GA*	Go Ahead	Used in Telnet half-duplex mode to signal the other device that it may transmit.
250	*SB*	Subnegotiation	Marks the beginning of a Telnet option subnegotiation, used when an option requires the client and server to exchange parameters. See the topic on Telnet options for a full description.
251	*WILL*	Will Perform	In Telnet option negotiation, indicates that the device sending this code is willing to perform or continue performing a particular option.
252	*WON'T*	Won't Perform	In Telnet option negotiation, indicates that the device sending this code is either not willing to perform a particular option, or is now refusing to continue to perform it.
253	*DO*	Do Perform	In Telnet option negotiation, requests that the other device perform a particular option or confirms the expectation that the other device will perform that option.
254	*DON'T*	Don't Perform	In Telnet option negotiation, specifies that the other party not perform an option, or confirms a device's expectation that the other party not perform an option.
255	*IAC*	Interpret As Command	Precedes command values 240 through 254 as described above. A pair of IAC bytes in a row represents the data value 255.

Figure IIX.4. Commandes Telnet[3*]

Négociation d'options

IV

- Le client et serveur peuvent négocier des options supplémentaires au NVT standard ;
- Pour qu'une option soit utilisée, les deux parties doivent être d'accord sur celle-ci ;
- Par exemple : TELNET est un protocole half-duplex où le client et le serveur doivent attendre un signal *Go Ahead* pour transmettre vers l'autre. Il existe une option définie appelée *SUPPRESS-GO-AHEAD* (code == 3) pour que la communication soit simultanée. Alors si le client souhaite utiliser cette option il envoie la séquence suivante :
  IAC WILL SUPPRESS-GO-AHEAD (suite de codes == 255 251 3) ;
  Si le serveur accepte cette option, celui-ci répondra avec : *IAC DO SUPPRESS-GO-AHEAD* (suite de codes == 255 253 3). Sinon, si l'option est refusée :*IAC DON'T SUPPRESS-GO-AHEAD* (suite de codes == 255 254 3) ;

Option Number	Option Code	Option Name	Description	Defining RFC
0	*TRANSMIT-BINARY*	Binary Transmission	Allows devices to send data in 8-bit binary form instead of 7-bit ASCII.	856
1	*ECHO*	Echo	When you press a key on a terminal, you also expect to see the character you entered appear on the terminal screen as output; this is called *echoing* the input. Echoing is a significant issue in terminal applications because it can be implemented in a number of different ways. This option allows devices to negotiate any of a variety of different echo modes.	857
3	*SUPPRESS-GO-AHEAD*	Suppress Go Ahead	Allows devices not operating in half-duplex mode to no longer need to end transmissions using the Telnet *Go Ahead* command.	858
24	*TERMINAL-TYPE*	Terminal Type	Allows the client and server to negotiate the use of a specific terminal type. If they agree, this allows the output from the server to be ideally customized to the needs of the particular terminal the user is working on.	1091
34	*LINEMODE*	Linemode	Allows the client to send data one line at a time instead of one character at a time. This improves performance by replacing a large number of tiny TCP transmissions with a smaller number of larger ones.	1184

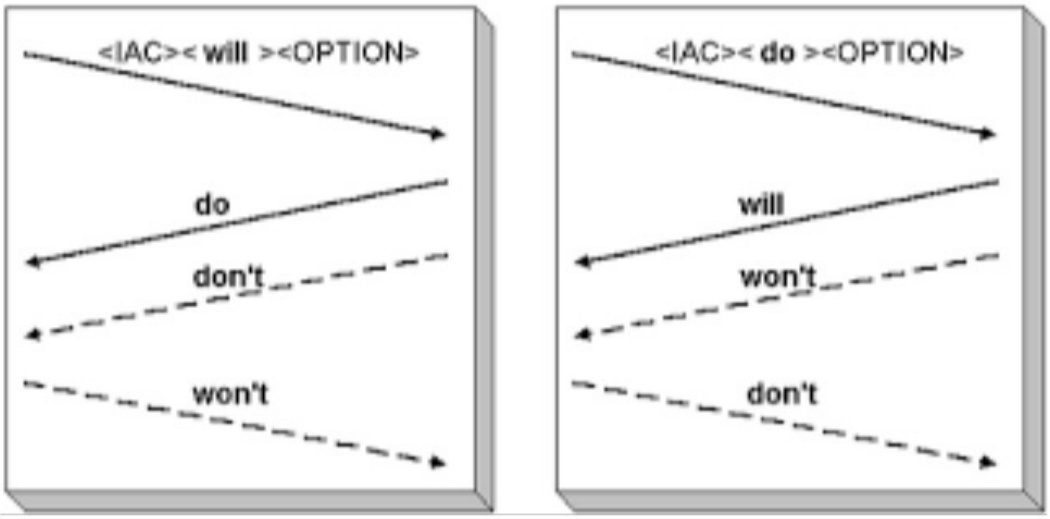

Figure IIX.5. Exemple d'échanges Telnet[3*]

Références

[03] The TCP/IP Guide: A Comprehensive, Illustrated Internet Protocols Reference, 2017 http://www.tcpipguide.com/free/index.htm