

rfc854

Crédits : J. Postel / ISI

Mai 1983

J. Reynolds / ISI

Traduction : Valéry Fremaux / EISTI

Mars 1998

Remplace : NIC 18639

# TELNET

Cette RFC définit un standard pour la communauté ARPA Internet. Les hôtes ARPA Internet sont enjoins à adopter et implémenter ce protocole.

## INTRODUCTION

Le but du protocole TELNET est de permettre une communication bidirectionnelle simple, sur une base d'octets de huit bits. Son objectif premier est de permettre d'interfacer des terminaux et des applications à travers un réseau. Il est envisageable de pouvoir utiliser ce protocole pour une communication de terminal à terminal ("linking") ou d'application à application (applications distribuées).

## CONSIDERATIONS GENERALES

Une connexion TELNET s'appuie sur une connexion TCP pour transmettre des données dans lesquelles s'intercalent des séquences de contrôle TELNET.

Le protocole TELNET est bâti selon trois principes essentiels: premièrement, le concept de terminal réseau virtuel (Network Virtual Terminal); deuxièmement, le principe d'options négociées; et troisièmement, une "vue" symétrique de chaque entité d'extrémité (processus ou terminal).

1. Lorsqu'une connexion TELNET est établie, chaque extrémité est sensée réagir comme un terminal réseau virtuel, (NVT). Un NVT est un composant imaginaire qui permet une représentation intermédiaire standard, appliquée réseau, d'un terminal canonique. Ceci évite aux deux extrémités potentielles d'avoir à mémoriser ou détecter les caractéristiques de l'autre extrémité, apportant une indépendance notable vis à vis des conventions de communication. Tous les hôtes, client ou serveur, devront opérer une conversion de leurs propres caractéristiques et conventions afin de se conformer à cette règle généralisée instaurée par la définition du NVT, et pourront supposer qu'une telle transformation est aussi faite à l'autre bout. La définition du NVT se veut un équilibre entre une définition ultra restrictive (qui conduirait à des problèmes avec certains systèmes du fait d'une trop grande pauvreté dans ses caractéristiques), et une définition trop poussée (pénalisant les utilisateurs de terminaux basiques).

NOTE: L'hôte "utilisateur" ou "client" est celui auquel le terminal physique est rattaché, l'hôte "serveur" est celui qui exécute généralement une application fournissant un service à l'utilisateur. D'un autre point de vue, applicable même dans le cas de communication entre terminaux ou entre processus, le "client" est celui qui aura demandé l'établissement de la connexion.

2. Le principe d'options négociées prend en compte le fait que de nombreux systèmes voudront pouvoir fournir un service amélioré par rapport à ce que permet la définition basique du NVT. D'autre part, de nombreux clients disposant de terminaux sophistiqués désireront pouvoir exploiter le confort de ces derniers, plutôt qu'un service minimaliste. Un certain nombre d'options indépendantes bien que prises en compte par la structure du protocole TELNET pourront être utilisées selon une structure "DO, DON'T, WILL, WON'T" (\*) pour permettre à un client et un serveur de pouvoir se mettre d'accord sur une augmentation de la richesse de l'échange (ou peut être juste un mode légèrement différent) et un ensemble de conventions particulières pour cette connexion TELNET. De telles options pourraient concerner le jeu de caractères, le mode d'écho local, etc...

La stratégie de base pour déterminer la possibilité d'usage d'une option est que l'une des parties (ou les deux) émette une requête demandant l'utilisation de telle ou telle option. L'autre extrémité peut alors accepter ou rejeter cette requête. Si la requête est acceptée, son usage prend immédiatement effet; si elle est rejetée, alors s'applique la règle standard définie pour le NVT. En clair, une extrémité peut toujours refuser d'accorder la mise en œuvre d'un caractère optionnel, mais ne doit jamais refuser une requête de désactivation d'une option, dans la mesure où tous les acteurs doivent au moins implémenter le modèle NVT.

La syntaxe de la négociation d'option a été faite de sorte que deux requêtes émises simultanément pour la même option apparaissent à l'autre bout comme une acceptation de leur propre requête.

*(\*) NdT : Après plusieurs tentatives de traduction de ces primitives, nous avons pensé qu'il était plus judicieux de laisser ces primitives en anglais, dans la mesure où il est assez difficile de trouver une correspondance exacte de ces auxiliaires dans ce contexte. Pour la suite de l'exposé, nous rajoutons ici une partie explicative non contenue dans la norme originale, destinée à fixer le cadre d'utilisation de ces quatre primitives :*

**WILL** Indique le désir d'utiliser, ou confirme le début de l'utilisation de l'option spécifiée. Dans le sens de "Je ferais bien" ou "Je ferai"

**WON'T** Indique le refus d'utiliser ou de continuer à utiliser l'option spécifiée. Dans le sens de "Je ne ferai pas (ou plus)"

**DO** Notifie une requête pour que l'autre extrémité utilise, ou la confirmation que ce côté attend l'utilisation de l'option spécifiée. Dans le sens de "Utilise !" ou "fais !"

**DON'T** Notifie une demande expresse que l'autre partie cesse d'utiliser, ou la confirmation que l'on attend plus l'usage, de l'option spécifiée. Dans le sens de "Ne fais pas !"

3. La symétrie de la syntaxe de négociation peut potentiellement conduire à des situations de bouclage du protocole – chaque extrémité considérant une commande entrante comme une nouvelle commande à acquitter plutôt qu'un acquittement à sa propre commande. Pour éviter de telles situations, les règles suivantes sont admises:

- a. Les acteurs ne doivent émettre des requêtes que pour demander le changement d'une option; c-à-d., éviter d'envoyer une commande pour indiquer dans quel mode on se trouve.

- b. Si un acteur reçoit une requête lui demandant d'entrer dans un mode dans lequel il se trouve déjà, la requête ne doit pas être acquittée. Cette non-réponse est essentielle pour éviter le cas de bouclages protocolaires infinis. Il est important cependant qu'une réponse soit donnée à toute requête de changement de mode "justifiée" – même si c'est par la négative.
- c. Lorsqu'une des deux parties émet une commande d'option vers l'autre, que ce soit une requête ou un acquittement d'une précédente requête, et dans la mesure où l'utilisation de cette option entraîne une modification du traitement effectué sur le représentation des données émises, alors cette commande doit être insérée dans le flux de données, à l'endroit à partir duquel elle doit prendre effet. (Il doit être noté ici qu'une certaine durée peut s'écouler entre l'émission première d'une requête et la réception de la réponse, qui peut d'ailleurs être négative. Pour cela, un hôte souhaitera pouvoir tamponner les données, après avoir requis une option, jusqu'à ce qu'il puisse être informé de l'acceptation ou du refus de cette option. Ceci permet alors de masquer cette période "d'incertitude" pour l'utilisateur).

De nombreuses requêtes d'option vont certainement s'entrecroiser à l'établissement d'une connexion TELNET, chacune des deux parties essayant d'obtenir le meilleur service de la part de la partie adverse. Outre ceci cependant, les options peuvent être utilisées pour modifier dynamiquement les caractéristiques d'une connexion afin de s'adapter à des changements de conditions locales. Par exemple, le NVT, comme il sera expliqué plus loin, utilise une "discipline" de transmission bien adaptée à des applications "interprétées par lignes" comme le BASIC, mais beaucoup moins adaptée à des applications fonctionnant au "caractère" comme pour NLS. Un serveur peut choisir d'affecter le processeur distant à des tâches nécessitant une technique de transmission au "caractère" uniquement lorsque le processeur local en a la nécessité et négocierait à ce moment seulement l'option correspondante. Toutefois, pour ne plus être "dérangé" à chaque caractère émis par le processeur distant, il pourra demander (c-à-d., négocier) le retour au mode NVT lorsqu'un contrôle au caractère n'est plus nécessaire.

Il est possible que des requêtes émises par des processus stimulent une boucle de requête infinie si le processus répond à un rejet par une nouvelle tentative de négociation de la même option. Pour éviter de tels bouclages du protocole, on ne devra jamais requérir une option rejetée tant qu'aucun événement ne suggère un changement dans la situation. Concrètement, cela peut vouloir dire que le processus déroule un code différent, ou une autre commande a été déclenchée par l'utilisateur, ou tout événement ayant une signification pour cette combinaison de processus et d'option. Une règle de sécurité consiste à ne jamais tenter de renégocier une option, sauf après réception d'information substantielle de la part du distant, ou après intervention volontaire de l'opérateur humain local.

Les programmeurs d'options ne doivent pas se sentir contraints par la syntaxe quelque peu limitée de négociation d'option. Le but premier d'une syntaxe aussi simple est de mettre facilement en œuvre le mécanisme d'options – et de ce fait tout aussi facile de les ignorer. Si certaines options demandaient une mécanique de négociation un peu plus complexe qu'il n'est possible de réaliser avec les primitives "DO, DON'T, WILL, WON'T", la méthode acceptable est de s'en tenir à ce mécanisme pour établir si les deux parties connaissent l'option en question, et une fois cette confirmation obtenue, il est possible de mettre en œuvre un mécanisme plus exotique sans crainte d'une possible incompréhension. Par exemple, une des parties pourrait envoyer une requête pour changer la longueur de la ligne. Si celle-ci est acceptée, alors une syntaxe différente peut être mise en application pour

effectivement négocier la longueur de la ligne – telle qu'une négociation à plusieurs champs du type "ligne minimale/ligne maximale/ligne souhaitée". Le concept essentiel à respecter est qu'une négociation complexe ne puisse être effectuée sans le passage par une étape simple (standard) de négociation, établissant au préalable la capacité des deux parties à mener à bien la transaction plus fine.

Pour résumer, WILL XXX est émis par les deux côtés, pour indiquer que les deux parties désirent (offrent) utiliser l'option XXX, DO XXX et DON'T XXX représentant un acquittement positif ou négatif; de façon similaire, DO XXX est envoyé pour indiquer une demande à utiliser l'option XXX (par l'autre partie), WILL XXX et WON'T XXX étant alors les acquittements positifs et négatifs. Dans la mesure où le NVT est tout ce qui reste lorsqu'aucune option n'est active, les réponses DON'T et WON'T garantissent le passage à un état dans lequel les deux côtés peuvent se mettre en attente. Ainsi, tous les hôtes devront implémenter leur processus TELNET de sorte à ne pas reconnaître implicitement toute option non connue, en simplement renvoyant un refus à toute requête d'option qui ne peut être comprise.

Pour autant que possible, le protocole TELNET a rendu la communication serveur-utilisateur symétrique de sorte à pouvoir gérer aussi bien les connexions utilisateur-utilisateur (intercommunication) que serveur-serveur (applications distribuées). Il est souhaitable, mais non imposé, que les options respectent ce principe. Dans tous les cas, la symétrie reste un des principes généralement reconnus.

Le document associé, "TELNET Option Specifications," pourra être consulté pour connaître la procédure à suivre pour l'établissement de nouvelles options.

## **Le Terminal Réseau Virtuel (Network Virtual Terminal)**

Le Network Virtual Terminal (NVT) est un composant réseau bidirectionnel en mode caractères. Le NVT dispose d'une sortie "d'impression" et d'un clavier. L'impression (au sens le plus large) reçoit le flux d'entrée de données et le clavier produit un flux sortant de données émis sur la connexion TELNET et, si un "écho local" est souhaité, également "imprimé" sur le NVT. Les "Echos" ne sont pas sensés traverser le réseau (bien que certaines options autorisent la génération de l'écho par le distant ("remote echoing"), aucun hôte n'est "obligé" de fournir cette option). Le jeu de caractères utilisé est l'ASCII-US sept bits dans un octet de huit bits, sauf cas particuliers décrits ici. Toute conversion de code et considérations temporelles sont des problématiques locales et n'affectent pas le NVT.

## **TRANSMISSION DES DONNEES**

Bien qu'une connexion TELNET à travers le réseau soit intrinsèquement bidirectionnelle, le NVT doit être considéré comme un appareil unidirectionnel alterné (half-duplex) fonctionnant en mode tampon de ligne. C'est-à-dire, sauf et jusqu'à ce que des options ne soient négociées pour signifier le contraire, les conditions suivantes constituent la transmission de données par défaut sur une liaison TELNET :

- 1) Pour autant que la taille du tampon local le permet, les données doivent être accumulées dans l'hôte d'où elles sont émises tant qu'une ligne complète de caractères n'est pas prête pour la transmission, ou qu'un signal local invitant explicitement à

transmettre ne soit reçu. Ce dernier pouvant être émis aussi bien par un processus que par un utilisateur humain.

Cette règle est motivée par le "coût" élevé, dans certains hôtes, du traitement des interruptions d'arrivée réseau, et correspond mieux à la spécification du NVT dans laquelle les échos ne traversent pas le réseau. De ce fait, il est raisonnable d'accumuler une certaine quantité de données à la source. De nombreux systèmes effectuent un traitement à la fin de chaque ligne (de nombreuses imprimantes en ligne ou lecteurs de cartes fonctionnent de cette façon), et la transmission sera donc déclenchée à la fin de la ligne. A l'inverse, un utilisateur ou un processus pourra souhaiter envoyer les données sans pour autant que la fin de ligne ne soit explicitement marquée; pour cela, les implémenteurs seront tenus de mettre à disposition un moyen de signaler que toutes les données rémanentes dans le tampon doivent être transmises.

2) Lorsqu'un processus a terminé l'envoi de données à une imprimante NVT et n'a aucune entrée de donnée en attente de traitement en provenance du clavier NVT (c'est-à-dire, lorsqu'un processus à l'une des extrémités d'une communication TELNET ne peut continuer son travail sans l'entrée de données à l'autre extrémité), , il devra émettre la commande TELNET Go Ahead (GA).

Cette règle n'impose pas la transmission d'une commande TELNET GA à partir d'un terminal pour chaque fin de ligne, dans la mesure où les hôtes serveurs ne nécessitent pas, en général, de signal supplémentaire (en plus du caractère habituel fin-de-ligne ou tout autre caractère défini localement dans ce but) pour commencer leur exécution. Plutôt, la commande TELNET GA est destinée à aider un hôte local (celui de l'utilisateur) à piloter un terminal half-duplex équipé d'un clavier "verrouillable" tel qu'un IBM 2741. Une description de ce type de terminal et de son fonctionnement permettra de mieux comprendre l'utilité de la commande GA.

Une connexion depuis un terminal sur un hôte local est toujours sous contrôle soit de l'utilisateur, soit de l'hôte. Aucune des deux extrémités ne peut unilatéralement "prendre la main" sur l'autre; il faudra que la partie contrôlante laisse la main explicitement. Côté terminal, le matériel est conçu de sorte à laisser la main à l'hôte à chaque fois qu'une ligne est terminée (c'est-à-dire, lorsque la touche "Entrée" est frappée par l'utilisateur). Dès que c'est le cas, l'ordinateur (local) rattaché traite la ligne d'entrée, décide des sorties à générer, en l'absence desquelles la main est rendue au terminal. Si une sortie doit être faite, l'ordinateur gardera le contrôle de la connexion jusqu'à ce que toutes les données soient émises.

La difficulté d'utiliser ce type de liaison à travers un réseau est évidente. L'hôte local n'est plus en position pour décider s'il doit garder le contrôle de la liaison sur le terminal après réception d'une fin-de-ligne ou au contraire rendre la main; cette décision ne peut être prise que par l'hôte "distant" qui exécute le processus de traitement. La commande TELNET GA institue un mécanisme par lequel le processus "distant" (serveur) peut signaler à l'hôte local qu'il est temps de redonner la main au terminal (donc à l'utilisateur). Elle doit être utilisée au moment (et seulement au moment) où la main doit être redonnée à l'utilisateur. Notez que la transmission prématurée d'une commande GA peut provoquer le blocage de "l'impression" de donnée, dans la mesure où il sera en droit de considérer que la voie de transmission s'est libérée, et échouera dans sa tentative d'inverser le sens de communication (on rappelle ici que nous sommes typiquement dans le cas d'un terminal à liaison unidirectionnelle alternée ou "half-duplex").

Ce qui précède, bien sûr, ne s'applique pas pour la direction de communication dans le sens utilisateur vers serveur. Dans cette direction, des commandes GA peuvent être émises à tout moment, mais ce n'est à la rigueur même pas nécessaire. Dans le même ordre d'idée, si la connexion TELNET est utilisée pour une communication de processus à processus, les commandes GA ne sont utiles dans aucune des deux directions. Enfin, pour une communication de terminal à terminal, des commandes GA peuvent être nécessaires dans aucune, une seule, ou les deux directions. Lorsqu'un hôte prévoit d'autoriser la communication de terminal à terminal, il est suggéré que l'hôte donne à l'utilisateur un moyen de signaler manuellement qu'il est temps d'envoyer une commande GA vers l'autre extrémité de la connexion TELNET; ceci, cependant, n'est pas une obligation pour les programmeurs d'applications TELNET.

Notez dans ce cas qu'étant donné la symétrie du modèle TELNET, il est supposé que l'on a affaire à un NVT à chaque extrémité de la connexion TELNET, du moins conceptuellement.

## **REPRESENTATION STANDARD DES FONCTIONS DE CONTROLE**

Comme il a été dit dans l'introduction de ce document, l'objectif premier du protocole TELNET est de fournir une interface standard de terminaux et de processus "orientés terminaux" à travers une liaison réseau. Des expériences précédentes de ce type d'interconnexion ont montré que ces fonctionnalités sont implémentées dans de nombreux serveurs, mais que les méthodes d'invocation de ces fonctions sont très diversifiées. Pour un utilisateur humain qui doit travailler simultanément sur plusieurs serveurs de natures différentes, cette hétérogénéité est assez frustrante. TELNET, pour cela, définit une représentation standardisée pour cinq de ces fonctionnalités, telles que décrites ci-après. Ces représentations standard ont des significations elles aussi standard, bien que non obligatoires (à l'exception que la fonction Interrupt Process (IP) peut être imposée par d'autres protocoles s'appuyant sur TELNET); en somme, un système qui ne proposerait pas une fonction identique à ses utilisateurs locaux n'est nullement obligé de proposer une telle fonction à ses utilisateurs distants et pourra interpréter l'appel d'une telle commande comme s'il s'agissait d'une No-operation. A l'inverse, un système qui propose la commande à ses utilisateurs locaux est tenu de reconnaître et d'exécuter cette commande pour des utilisateurs distants qui utiliseraient la représentation standard appropriée.

### ***Interrupt Process (IP)***

De nombreux systèmes procurent une commande qui permet de suspendre, interrompre, arrêter brutalement, ou provoquer la fin normale d'un processus utilisateur. Cette fonction est fréquemment utilisée lorsque l'utilisateur pense que son programme est parti dans une boucle infinie, ou lorsqu'un programme a été lancé par inadvertance. IP est la représentation standardisée de cette fonctionnalité. Les implémenteurs devront noter qu'IP peut être nécessaire à d'autres protocoles se basant sur TELNET, et que cette fonctionnalité devra être implémentée si l'on prévoit d'utiliser ces protocoles par la suite.

### ***Abort Output (AO)***

De nombreux systèmes procurent une fonction qui, lorsqu'un processus "imprime" des données à l'écran, en permet l'exécution normale (ou du moins, le déroulement jusqu'au même point que si l'exécution est faite sans utilisation de cette commande) mais sans imprimer les sorties sur le terminal utilisateur. En plus, cette fonction videra tout tampon intermédiaire des données déjà sorties par le processus, mais non encore "imprimées" (ou affichées) à l'écran. AO est la représentation standardisée pour cette fonctionnalité. Par exemple, certains systèmes d'exploitation (ou sous systèmes, par exemple un Shell fils), suite à une commande utilisateur, renvoient en réponse une longue chaîne de caractères sur le terminal de l'utilisateur, puis signalent qu'ils sont prêt à recevoir une nouvelle commande en affichant un caractère de "prompt" (précédé d'un <CR><LF>). Si la commande AO est reçue pendant la transmission de la chaîne de texte, une implémentation "sensée" supprimerait ce qu'il reste à afficher de la chaîne, mais laisserait au minimum passer le "prompt" et la séquence <CR><LF> qui le précède. (Ceci peut se distinguer de l'action qui pourrait être menée dans le cas d'une commande IP; cette dernière provoquerait la suppression de la fin de la ligne de texte, mais en plus la sortie du sous-système, suivant le cas).

Il doit être rappelé, pour les serveurs qui implémentent cette fonctionnalité, qu'il peut exister des tampons de données extérieurs au système lui-même (dans le réseau, et sur l'hôte local où est raccordé l'utilisateur) qui devraient aussi être vidés; la manière de le faire est d'envoyer le signal "Synch" (décrit ci-après) à l'hôte local.

### ***Are You There (AYT)***

De nombreux systèmes proposent à l'utilisateur une méthode pour tester d'une façon explicite (ex, à l'écran) si le système est toujours opérationnel. Cette fonction peut être invoquée lorsque le système reste "silencieux" pendant un temps apparemment long, peut être parce que le traitement des données est subjectivement plus long que ce qu'avait prévu l'utilisateur, ou à cause d'une surcharge temporaire, etc. AYT est la représentation standard de cette fonctionnalité.

### ***Erase Character (EC)***

La plupart des systèmes proposent une fonction pour effacer la dernière position de caractère "imprimé"\* dans le flux de données entré par l'utilisateur. Cette fonction est typiquement utilisée pour éditer et corriger la ligne d'entrée lorsque des erreurs y ont été commises. EC est la représentation standard pour cette fonctionnalité.

\*NOTE: Une position de caractère "imprimé" peut contenir plusieurs caractères comme le résultat d'une surcharge, notamment des séquences de type <char1> BS <char2> ou BS est le caractère de retour arrière et <char2> le caractère "corrigé".

### ***Erase Line (EL)***

La plupart des systèmes proposent une fonction qui permet l'effacement complet de la "ligne" d'entrée. Cette fonction est typiquement utilisée pour taper une nouvelle

ligne de commande à la place d'une autre qui est abandonnée. EL est la représentation standard pour cette fonctionnalité.

## **LE SIGNAL "SYNCH" TELNET**

La plupart des systèmes en temps partagé proposent des mécanismes qui permet à l'utilisateur d'un terminal de reprendre la main sur un processus "en rideau"; les fonctions IP et AO sont des exemples de ces mécanismes. De tels systèmes, pour ce qui est des utilisateurs locaux, ont accès à tous les signaux émis par ces derniers, soit sous forme de caractères normaux ou signaux "hors bande" implémentés par le matériel telle que ceux produits par activation de la touche "BREAK" d'un Télétype ou de la touche "ATTN" de l'IBM 2741. Ceci n'est évidemment pas nécessairement vrai lorsque le terminal est raccordé via un réseau; les mécanismes de contrôle de flux du réseau peuvent obliger à une rétention de ces données, notamment au niveau de l'hôte local associé à l'utilisateur.

Pour contourner ce problème, le mécanisme du signal "Synch" a été introduit dans TELNET. Un signal Synch utilise le mécanisme d'urgence de TCP, couplé à une commande DATA MARK de TELNET. L'introduction de données urgentes dans un flux TCP, lesquelles outrepassent les règles de contrôle de flux établies pour la connexion TELNET normale, est utilisée pour indiquer au processus récepteur d'entrer dans un mode particulier de traitement des données émises sur le réseau. Dans ce mode, le flux d'arrivée réseau est inspecté en permanence dans l'attente de signaux "significatifs" décrits ci-après, toute autre donnée étant ignorée. La commande DATA MARK (DM) de TELNET est la marque de synchronisation dans le flux de données qui indique que la commande spéciale a été d'ores et déjà envoyée et que le récepteur peut repasser dans un mode normal de traitement du flux réseau.

Le signal Synch est émis par appel à la primitive d'émission de TCP en marquant le bit Urgent et en terminant le message urgent par la commande DM (dans le flux de données urgentes).

Lorsque plusieurs signaux Synch sont émis successivement à cadence rapide, le marquage de l'Urgence peut être fait une fois pour les multiples signaux Synch. Il n'est donc pas possible de se limiter à compter le nombre de passage en mode Urgent dans la mesure où il sera reçu moins de signaux urgent qu'il n'est émis de signaux Synch. En mode normal, une commande DM équivaut à une commande No-operation; en mode urgent, elle signale la fin de la séquence d'urgence.

Si TCP notifie la fin d'un état d'urgence avant que la commande DM ne soit détectée, TELNET devra continuer à traiter les données dans ce mode spécial jusqu'à réception explicite de cette commande.

TCP ne notifie pas toujours la fin d'un état d'urgence après réception de la commande DM. On détecte alors le chaînage de plusieurs signaux Synch consécutifs dans la même séquence d'urgence TCP. TELNET doit alors continuer à procéder au traitement spécial des données jusqu'à réception d'une nouvelle commande DM. Les signaux "significatifs" sont sensé être : les représentations standard TELNET des commandes IP, AO, et AYT (mais pas EC ou EL) ; les analogues locales de ces commandes standard (si elles existent) ; toute autre



commande TELNET ; d'autres signaux propriétaires qui doivent pouvoir être transmis sans retard à travers le flux réseau.

Du fait qu'un des effets du mécanisme SYNCH est d'ignorer pratiquement tous les caractères (sauf dans une commande TELNET) tamponnés entre l'émetteur du Synch et le processus destinataire, ce mécanisme peut être employé à chaque fois qu'une liaison doit être "nettoyée" des données rémanentes dans les tampons locaux. Par exemple, si l'utilisateur d'un terminal provoque l'émission d'une commande AO, le serveur recevant la commande AO (s'il gère cette fonctionnalité) devra retourner un signal Synch vers l'utilisateur.

Enfin, tout comme l'utilisation du mécanisme d'urgence de TCP sert à la couche TELNET comme méthode pour émettre des signaux "hors bande", d'autres protocoles au dessus de TELNET peuvent nécessiter l'emploi de commandes TELNET vues comme des signaux "hors bande" du point de vue de ce protocole.

Par convention, la séquence [IP, Synch] doit être utilisée pour générer un tel signal. Par exemple, supposons qu'un autre protocole, s'appuyant sur TELNET, définisse une commande écrite "STOP" (un bouton par exemple) pour proposer la même fonctionnalité que la commande AO de TELNET. Imaginons que l'utilisateur de cet agent appuie sur ce bouton "STOP", et que la connexion est actuellement bloquée par un traitement en cours d'exécution. L'utilisateur devrait ordonner à son système de:

1. Envoyer le caractère IP de TELNET;
2. Envoyer la séquence SYNCH de TELNET, à savoir:

Envoyer la Data Mark (DM) comme caractère unique d'une séquence urgente de TCP.

3. Emettre la traduction haut niveau de l'action sur le bouton STOP (par exemple la commande "STOP" sous forme d'une chaîne de caractères ; et

4. Envoyer l'équivalent haut niveau de la commande DM de TELNET (à l'attention du protocole de même niveau distant), si elle existe.

L'utilisateur (ou le processus agissant sous son contrôle, encore appelé "agent") doit transmettre la séquence SYNCH TELNET à l'étape 2 ci dessus pour s'assurer que le caractère IP TELNET traverse effectivement toute la liaison jusqu'à l'interpréteur TELNET distant.

Le déclenchement du mode urgent de TCP devra réveiller le processus TELNET dès réception ; le caractère IP devra réveiller le processus protocolaire de niveau supérieur.

## **L'IMPRIMANTE ET LE CLAVIER NVT**

L'imprimante NVT présente un nombre de colonnes et de lignes indéterminés peut reproduire la représentation des 95 caractères graphiques de l'ASCII-US (codes 32 à 126). Parmi les 33 caractères de contrôle de USASCII (0 à 31 et 127), et les 128 codes de l'ASCII étendu (128 à 255), les codes suivants ont été retenus comme ayant une signification particulière pour l'imprimante NVT :

NOM	CODE	SIGNIFICATION
-----	------	---------------

NULL (NUL)	0	No-operation
Line Feed (LF)	10	Déplace le curseur d'impression à la ligne suivante, en conservant sa position horizontale
Carriage Return (CR)	13	Déplace le curseur d'impression à l'extrême gauche de la ligne courante

De plus, les codes suivants peuvent avoir un effet défini, mais non nécessairement implémenté, sur l'imprimante NVT. Aucune des extrémités d'une connexion TELNET ne peut être sûre que l'autre extrémité agit, ou va agir, sur réception d'un des caractères suivants :

BELL (BEL)	7	Produit un signal audible ou visible sans déplacer la position du curseur d'impression.
Back Space (BS)	8	Déplace le curseur d'impression d'un caractère vers la marge gauche.
Horizontal Tab (HT)	9	Déplace le curseur d'impression jusqu'à la marque de tabulation horizontale suivante. La façon dont ces tabulations sont placées ou créées reste à discrétion du récepteur.
Vertical Tab (VT)	11	Déplace le curseur d'impression jusqu'à la marque de tabulation verticale suivante. La façon dont ces tabulations sont placées ou créées reste à discrétion du récepteur.
Form Feed (FF)	12	Déplace le curseur d'impression en tête de la page suivante sans changer de position horizontale.

Tous les autres codes sont sensés ne provoquer aucune réaction de la part d'un NVT.

La séquence "CR LF", ainsi définie, provoquera un déplacement du curseur d'impression du NVT à la marge gauche de la ligne suivante (de même, la séquence inverse "LF CR"). Cependant, de nombreux systèmes et terminaux ne traitent pas les caractères CR et LF indépendamment, mais peuvent simuler leur effet au prix de certains efforts. (Par exemple, certains terminaux ne savent pas traiter l'effet du CR sans y adjoindre celui du LF, mais il reste possible de simuler l'effet du CR seul par une suite de BackSpace). C'est pourquoi la séquence "CR LF" devra être traitée comme un caractère unique de signification "nouvelle ligne" à utiliser chaque fois que l'action combinée des deux caractères de base est souhaitée ; la séquence "CR NUL" devra être utilisée lorsque seul l'effet du CR est désiré ; l'usage du CR seul devenant de ce fait déconseillé. Cette règle permet d'assurer à un système devant faire le choix de l'effet "nouvelle ligne" ou "backspace multiple" la présence systématique d'un deuxième caractère après le CR qui permet de lever le doute dans tous les cas.

Notez que l'usage des séquences "CR LF" ou "CR NUL" est nécessaire dans les deux directions (pour le mode ASCII par défaut), afin de préserver la symétrie du modèle du NVT. Même dans certaines situations bien identifiées (ex, avec les options écho local et Go Ahead supprimé en fonction) dans laquelle il est connu qu'aucun caractère n'est envoyé à "l'imprimante" locale, et ce pour des raisons de

cohérence, le protocole demandera qu'un caractère NUL soit placé après un CR si celui-ci n'est pas suivi immédiatement d'un caractère LF. La contrepartie de ceci est qu'un caractère NUL reçu dans un flux de données après un CR (sauf si une option négociée en décide autrement) devra être retiré du flux de données avant de soumettre celui-ci à la conversion locale des caractères pour le NVT.

Le clavier NVT utilise des touches, des combinaisons, ou séquences de touches, pour générer les 128 codes de l'ASCII US. Notez que bien que de nombreux caractères de cet ensemble n'aient aucun effet sur une "imprimante" NVT, le clavier NVT demeure capable de les émettre.

En plus de ces codes, le clavier NVT peut générer les codes spéciaux suivants qui, sauf mention contraire, ont une signification définie (mais pour lesquels la fonctionnalité n'est pas obligatoirement implémentée). Les codes assignés pour ces "caractères" sont donnés dans la section Commandes TELNET. Dans la mesure où ces fonctions sont génériques, d'un certain point de vue, ils devront rester indépendant du jeu de caractère utilisé pour coder le flux de données.

### ***Synch***

Cette touche permet à l'utilisateur de vider tous les tampons intermédiaires entre le NVT et le destinataire final. L'activation de cette touche provoque l'émission d'une commande DM (cf. la section Commandes TELNET) dans le flux de données, associée à l'activation du dispositif d'urgence de TCP. La paire DM-Urgent a une signification normalisée telle que définie précédemment.

### ***Break (BRK)***

Cette touche est fournie pour l'émission d'un code hors du jeu de caractères ASCII US auquel est donnée une signification locale par de nombreux systèmes. Il est d'usage de signaler une pression de la touche Break (ou Attention sur IBM). Notez que cela suppose l'utilisation d'un 129ème code pour les systèmes qui utilisent cette fonction, et n'est pas un synonyme de la fonctionnalité IP.

### ***Interrupt Process (IP)***

Le processus auquel est connecté le NVT est suspendu, interrompu, arrêté de façon normale ou abrupte. Est utilisé comme signal "hors-bande" par d'autres protocoles s'appuyant sur TELNET.

### ***Abort Output (AO)***

Permet au processus courant de terminer son exécution, mais sans plus envoyer de données en sortie vers l'utilisateur. Sur réception, un code Synch est émis dans le sens processus vers utilisateur.

### ***Are You There (AYT)***

Renvoie au NVT un signal visible (c'est-à-dire imprimable) montrant que ce contrôle AYT a été reçu (sous entendu : le système est toujours opérationnel).

### ***Erase Character (EC)***

Le récepteur effacera le dernier caractère non effacé (ou la dernière "position imprimée").

### ***Erase Line (EL)***

Le récepteur effacera toutes les données situées entre la fin du flux et la dernière séquence "CR LF" émise (cette séquence est cependant conservée à la fin du flux restant).

L'esprit dans lequel on a été conçues ces touches, ainsi que les contrôles de format de l'impression, est qu'elles doivent être prises comme une extension naturelle du transcodage qui est déjà effectué pour passer du modèle NVT à l'implémentation locale. Tout comme le code 68 du NVT (104 en octal) devra être transcodé dans son équivalent local pour le caractère "D", le code EC devra être transcodé en son équivalent associé à la fonctionnalité "Erase Character" locale. De même, tout comme le transcodé du code 124 (174 en octal) peut apparaître comme un caractère quelque peu arbitraire sur un système n'utilisant pas le caractère "vertical bar", le caractère EL sera transcodé tout aussi arbitrairement (ou pas du tout) si la fonction "Erase Line" n'existe pas sur le système cible. Il en est de même pour les codes de contrôle de format: Si le terminal utilisé dispose de la fonction de Tabulation Verticale, alors le transcodage du code VT devient évident, et son effet ne sera imprévisible que sur les terminaux qui ne connaissent pas cette fonction.

## **STRUCTURE DES COMMANDES TELNET**

Toutes les commandes TELNET consistent au minimum en une séquence de deux octets : un premier caractère d'échappement IAC (pour) "Interpret as Command" suivi immédiatement du code numérique de la commande sur un octet. Les commandes qui implémentent le processus de négociation d'options ont un octet supplémentaire, ce troisième octet reçoit le code de l'option concernée par la négociation. Ce format a été choisi de sorte à minimiser les "collisions" ou fausses interprétations qui conduiraient à confondre données et commandes (dans la forme basique de la négociation NVT, du moins). Dans ce contexte, seul le caractère IAC doit être répété dans le flux de données pour être reconnu en tant que donnée, les 255 autres codes de caractères pouvant "passer" de façon totalement transparente.

Ce qui suit sont les commandes prédéfinies de TELNET. Notez que ces codes et séquences de codes n'ont la signification mentionnée que lorsqu'ils sont immédiatement précédés d'un caractère IAC.

NAME	CODE	SIGNIFICATION
SE	240	Fin de négociation de paramètre.
NOP	241	No-operation.
Data Mark	242	La partie données du signal Synch. Doit toujours être associé à un marquage du bit Urgent dans TCP.
Break	243	Caractère BRK du NVT.
Interrupt Process	244	Représente la fonction IP.
Abort output	245	Représente la fonction AO.

Are You There	246	Représente la fonction AYT.
Erase character	247	Représente la fonction EC.
Erase Line	248	Représente la fonction EL.
Go ahead	249	Représente le signal GA.
SB	250	Indique que ce qui suit est une négociation de l'option précédente.
WILL (code d'option)	251	Indique le désir d'utiliser, ou confirme le début de l'utilisation de l'option spécifiée. Dans le sens de "Je ferais bien" ou "Je ferai"
WON'T (code d'option)	252	Indique le refus d'utiliser ou de continuer à utiliser l'option spécifiée. Dans le sens de "Je ne ferai pas (ou plus)"
DO (code d'option)	253	Notifie une requête pour que l'autre extrémité utilise, ou la confirmation que ce côté attend l'utilisation de l'option spécifiée. Dans le sens de "Utilise !" ou "fais !"
DON'T (code d'option)	254	Notifie une demande expresse que l'autre partie cesse d'utiliser, ou la confirmation que l'on attend plus l'usage, de l'option spécifiée. Dans le sens de "Ne fais pas !".
IAC	255	Octet de données 255.

## **ETABLISSEMENT DE LA CONNEXION**

La connexion TCP de TELNET entre le port U du système utilisateur et le port L du système serveur. Le serveur passe en écoute sur le port prédéfini L dans l'attente de telles connexions. Dans la mesure où une connexion TCP est bidirectionnelle et est identifiée par une paire de sockets impliquant deux numéros de ports, le serveur peut accepter plusieurs connexions simultanées sur son même port L tant que le socket utilisateur est à chaque fois distinct.

### ***Assignment du port TELNET***

Lorsque ce protocole est utilisé pour la connexion distante d'un terminal utilisateur aux services d'un hôte, ce protocole est assigné au port prédéfini 23 (27 en octal). Soit, L=23.