

Groupe de travail Réseau  
**Request for Comments : 726**  
 NIC : 39237  
 Traduction Claude Broère de L'Isle

Jon Postel & Dave Crocker  
 SRI-ARC, UC Irvine  
 8 mars 1977

## Option Telnet de transmission et d'écho contrôlés à distance

### 1. Nom et code de la commande

RCTE 7

### 2. Signification de la commande

IAC WILL RCTE

L'envoyeur de cette commande DEMANDE ou ACCEPTE d'utiliser l'option RCTE, et va envoyer des instructions pour contrôler l'imprimante du terminal de l'autre côté.

IAC WON'T RCTE

L'envoyeur de cette option REFUSE d'envoyer des instructions pour contrôler l'imprimante du terminal de l'autre côté.

IAC DO RCTE

L'envoyeur DEMANDE ou ACCEPTE que l'autre côté (l'envoyeur de WILL RCTE) produise des commandes qui vont contrôler les résultats de son (l'envoyeur du DO) imprimante de terminal.

IAC DON'T RCTE

L'envoyeur de cette commande REFUSE de permettre à l'autre côté de contrôler l'imprimante de son (l'envoyeur du DON'T) terminal.

IAC SB RCTE <cmd> [BC1 BC2] [TC1 TC2] IAC SE

où :

<cmd> est un octet qui a les fanions (les bits sont comptés de droite à gauche) suivants :

#### Bit Signification

- |   |   |
|---|---|
| 0 | 0 = Ignorer tous les autres bits de cet octet et répéter la dernière <cmd> envoyée. Équivaut à 'continuer ce que vous êtes en train de faire'.<br>1 = Effectuer les actions indiquées par les autres bits de cet octet.   |
| 1 | 0 = Imprimer (faire écho) les caractères de coupure.<br>1 = Sauter (ne pas faire écho) les caractères de coupure.   |
| 2 | 0 = Imprimer (faire écho) le texte jusqu'au caractère de coupure.<br>1 = Sauter (ne pas faire écho) le texte jusqu'au caractère de coupure.   |
| 3 | 0 = Continuer d'utiliser les mêmes classes de caractères de coupure.<br>1 = Les deux octets qui suivent celui-ci contiennent des fanions pour les nouvelles classes de coupure.   |
| 4 | 0 = Continuer d'utiliser les mêmes classes de caractères de transmission.<br>1 = Rétablir les classes de transmission conformément aux deux octets qui suivent 1) les octets de classes de coupure, si les classes de coupure sont aussi rétablies, ou 2) cet octet, si les classes de coupure NE SONT PAS aussi rétablies. |

Valeur (en décimal) de l'octet <cmd> et sa signification :

0 = Continuer ce que vous êtes en train de faire.

Les nombres pairs positifs (c'est-à-dire les nombres dont le bit de poids fort est à zéro) sont erronés et devraient être interprétés comme égaux à zéro. Lorsque la commande <cmd> est un nombre pair positif, les octets de classes TC1 & TC2 et/ou BC1 & BC2 ne doivent pas être envoyés.

1 = Imprimer (faire écho) jusque ET Y COMPRIS au caractère de coupure.

3 = Imprimer jusqu'au caractère de coupure et SAUTER (ne pas faire écho) le caractère de coupure.

5 = Sauter le texte (ne pas faire écho) jusqu'au caractère de coupure, mais IMPRIMER le caractère de coupure.

7 = Sauter jusque et y compris le caractère de coupure.

Ajouter une des valeurs non zéro précédentes à une des valeurs suivantes pour obtenir la valeur décimale totale pour l'octet. (Noter que les classes ne peuvent pas être rétablies sans rétablir aussi l'action d'impression, de sorte qu'un nombre impair est garanti).

8 = Établir les classes de coupure (en utilisant les deux octets suivants [BC1 BC2]).

16 = Établir les classes de transmission (en utilisant les deux octets suivants [TC1 TC2]).

24 = Établir les classes de coupure (en utilisant les deux octets suivants [BC1 BC2]) et les classes de transmission (en utilisant les deux octets qui suivent de [TC1 TC2]).

Les sous-commandes (IAC SB RCTE...) ne sont envoyées que par l'hôte contrôleur et, en plus de leurs autres fonctions, remplacent fonctionnellement le dispositif Telnet Go-Ahead (IAC GA). RCTE remplace aussi fonctionnellement l'option Telnet Écho (IAC ECHO). C'est-à-dire que l'option Suppress Go-Ahead devrait s'appliquer et l'option Écho ne devrait pas s'appliquer lorsque l'option RCTE est en activité. Le mode écho devrait être l'état par défaut à la fin de l'utilisation de l'option RCTE, c'est-à-dire DON'T ECHO, WON'T ECHO.

Les classes de coupure et de transmission (le bit de poids fort du second octet (TC2 ou BC2) représentent la classe 1 ; le bit le plus à gauche du premier octet (TC1 ou BC1) représente la classe 16 actuellement non définie :

1 : Lettres majuscules (A-Z)

2 : Lettres minuscules (a-z)

3 : Chiffres (0-9)

4 : Opérateurs de format (<BS> <CR> <LF> <FF> <HT> <VT>)

La séquence <CR><LF> compte pour un caractère lorsque traitée comme fin de ligne Telnet, et est un seul caractère de coupure lorsque la classe 4 est établie. La séquence <CR><NUL> compte comme un caractère et est un caractère de coupure si et seulement si <CR> est un caractère de coupure (c'est-à-dire si la classe 4 est établie).

5 : Caractères de contrôle d'opérateurs non de formatage incluant <DEL> et <ESC>

6 : . , ; : ? !

7 : { [ ( < > ] }

8 : ' " / \ % @ \$ & # + - \* = ^ \_ | ~

9 : <Espace>

Et les commandes Telnet (IAC . . .) envoyées par l'utilisateur ont toujours l'effet d'un caractère de coupure. C'est-à-dire que chaque instance d'un IAC doit être traitée comme un caractère de coupure, sauf la séquence IAC IAC.

La représentation à afficher lorsque l'impression est invoquée est celle qui est évidente pour les caractères visibles (classes 1, 2, 3, 6, 7, et 8). Les espaces (classe 9) sont représentées par un blanc. Les opérateurs de format (classe 4) par leur effet de format. Les contrôles d'opérateur de non formatage (classe 5) n'impriment rien (pas d'espace).

Au départ, aucune classe de coupure ni de transmission n'est en activité.

Prière de noter que si tous les bits sont établis dans un octet d'argument de sous-commande Telnet tel que TC2 ou BC2, cet octet doit alors être précédé par un octet de fanion <IAC>. C'est la convention courante de doublement du caractère d'échappement pour utiliser sa valeur comme données.

On appelle les sous-commandes (IAC SB RCTE...) les "commandes de rétablissement de coupure".

### 3. État par défaut

WON'T RCTE -- DON'T RCTE

Ni l'un, ni l'autre hôte n'affirme un contrôle particulier sur l'imprimante du terminal de l'autre hôte.

## 4. Motifs de l'option

Les RFC 1, 5 et 51 discutent de l'efficacité et de souplesse du traitement du réseau.

La RFC357, de John Davidson, introduit le problème du délai d'écho qui survient lorsque un utilisateur distant accède à un hôte en bidirectionnel, à travers une liaison par satellite. Afin d'économiser des milliers de kilomètres de temps de transit pour chaque écho de caractère, tout en permettant quand même une parfaite réactivité du serveur et un résultat propre du terminal, un contrôle d'écho similaire à celui utilisé par certains systèmes en temps partagé est suggéré pour le réseau tout entier.

En effet, l'option décrite dans le présent document implique de faire qu'un hôte utilisateur régule soigneusement l'imprimante locale conformément aux instructions explicites provenant de l'hôte (serveur) distant.

Une question supplémentaire importante est l'efficacité de la transmission par le réseau. La mise en œuvre du schéma d'écho de Davidson va éliminer presque tous les échos de serveur à usager.

L'option décrite dans le présent document exige aussi d'utiliser les hôtes pour mettre en mémoire tampon le résultat d'un terminal chez l'hôte serveur jusqu'à ce qu'il forme une unité utile (une "unité utile" étant délimitée par des caractères de coupure ou de transmission comme on le décrit ci-dessous). Donc, moins de messages sont envoyés sur le chemin entre l'usager et le serveur.

Note : Cette option est destinée à n'être utilisée qu'avec des hôtes reliés en bidirectionnel. Le dispositif Go-Ahead est parfaitement adéquat pour les liaisons serveur-hôte en unidirectionnel. Aussi, RCTE devrait être utilisé à la place de l'option Telnet ECHO. C'est-à-dire que l'option Suppress Go-Ahead devrait être activée et que l'option Écho ne devrait pas être activée lorsque l'option RCTE est utilisée.

## 5. Description explicite du mécanisme de contrôle

### Procédure d'action et de contrôle d'impression de terminal d'utilisateur

Négocier l'utilisation de l'option RCTE. Une fois que l'option est activée, l'usager Telnet suit la procédure ci-après.

- 1) Lire un élément provenant du réseau.  
Si l'élément est des données, les imprimer et passer à 1.  
Si l'élément est une commande, établir alors les classes et passer à 2.
- 2) Si la mémoire tampon d'entrée du terminal est vide, passer alors en 3, sinon aller en 4.
- 3) Attendre qu'un élément apparaisse provenant du terminal ou du réseau.  
Si un élément provient du terminal, passer alors en 4.  
Si un élément de données provient du réseau, l'imprimer et passer alors en 3.  
Si une commande provient du réseau, une erreur s'est alors produite.
- 4) Lire un élément de la mémoire tampon d'entrée du terminal.  
Si l'élément n'est pas une coupure, l'imprimer/le sauter et passer en 2.  
Si l'élément est une coupure, l'imprimer/la sauter, et passer en 1.

Note : Le résultat provenant de l'hôte serveur peut survenir à tout moment ; un tel "résultat spontané" est imprimé à l'étape 3.

### Explication :

Les deux hôtes se mettent d'accord pour utiliser l'option RCTE. Après cela, l'hôte utilisateur (IAC DO RCTE) agit simplement sur les commandes de l'hôte contrôleur (serveur) et ne produit plus aucune commande RCTE sauf si, et jusqu'à ce qu'il (l'hôte utilisateur) décide de cesser de permettre l'utilisation de l'option (par l'envoi de IAC DON'T RCTE).

- 1) L'hôte utilisateur est synchronisé avec le serveur à l'initialisation et chaque fois qu'il retourne à l'étape 1 en suspendant l'impression d'écho de terminal jusqu'à ce qu'il reçoive une commande du serveur.

Le serveur peut envoyer soit un résultat à l'imprimante du terminal, soit une commande, et il envoie habituellement les deux.

Le serveur peut envoyer un résultat à l'imprimante du terminal soit en réponse à une entrée de l'utilisateur, soit spontanément. Dans le premier cas, l'entrée est traitée à l'étape 1. Dans le second cas, l'entrée est traitée dans l'étape 3.

Le serveur envoie une commande RCTE. La commande peut redéfinir des classes de coupure et de transmission, l'action à effectuer sur les caractères de coupure, et l'action à effectuer sur du texte. Chacune de ces fonctions indépendantes est contrôlée par des bits distincts dans l'octet <cmd>.

Un caractère de transmission est celui qui RECOMMANDE que l'hôte utilisateur transmette tout le texte accumulé jusque et y compris le moment où il survient. (Pour l'efficacité du réseau, il est DÉCONSEILLÉ (mais pas interdit) aux hôtes utilisateurs d'envoyer avant que survienne un caractère de transmission, comme défini au moment où le caractère est frappé).

Si le bit des classes de transmission (bit 4) est établi (*à 1*), les deux octets qui suivent les deux octets de classe de coupure (ou qui suivent immédiatement l'octet <cmd> si la classe de coupure n'est pas établie) vont indiquer quelles classes sont à activer.

Si le bit est OFF (*à 0*), la classe de transmission reste inchangée. Lorsque l'option RCTE est initialisée, AUCUNE CLASSE n'est activée. C'est-à-dire qu'aucun caractère ne sera considéré comme étant un caractère de transmission. (Comme si TC1 et TC2 étaient tous les deux à zéro.)

Un caractère de coupure EXIGE que l'hôte utilisateur transmette tout le texte accumulé jusque à et y compris sa survenance et cause aussi l'arrêt par l'hôte utilisateur de son action d'impression/élimination sur le texte d'entrée de l'utilisateur, jusqu'à ce qu'il ait pour instruction de faire autrement par une autre commande IAC SB RCTE <cmd> IAC SE provenant de l'hôte serveur. Les caractères de coupure définissent donc les unités d'impression. Les "caractères de coupure" tels qu'utilisés dans le présent document NE signifient PAS des caractères de coupure Telnet.

Si le bit des classes de coupure (bit 3) est établi (*à 1*), les deux octets qui suivent <cmd> vont indiquer quelles classes sont à activer. Il y a actuellement neuf (9) classes définies, avec de la place pour de l'expansion.

Si le bit est OFF (*à 0*), les classes de coupure restent inchangées. Lorsque l'option RCTE est initialisée, AUCUNE CLASSE n'est activée. C'est-à-dire qu'aucune transmission n'aura lieu dans la direction usager-serveur jusqu'à ce que soit reçue par l'utilisateur la première commande de rétablissement de coupure de la part du serveur.

La liste des classes de caractères, utilisée pour définir les classes de coupure et de transmission figure à la fin du présent document, à la Section Tableaux.

Parce que les caractères de coupure sont particuliers, l'action d'impression/élimination qui devrait être effectuée sur eux n'est pas toujours la même que celle qui devrait être effectuée sur le reste du texte d'entrée.

Par exemple, lors de la frappe d'un nom de fichier de TENEX, je veux que le texte du nom de fichier soit imprimé (qu'il en soit fait écho) mais je ne veux pas que le <échappement> (si j'utilise le dispositif d'achèvement de nom) soit imprimé.

Si le bit 1 est à 1 (*ON*) le caractère de coupure N'EST PAS à imprimer.

Un bit distinct (le bit 2) signale si le texte lui-même devrait être ou non imprimé (y être fait écho) au terminal. Si le bit 2 = 0, le texte DOIT être imprimé.

Encore un autre bit (le bit 0 – bit de droite) signale si les autres bits de la commande devraient ou non être vérifiés. Si ce bit est à 0 (*OFF*), la commande devrait alors être interprétée comme signifiant "continuer quelle que soit la stratégie d'écho que vous suivez, en utilisant les mêmes classes de coupure et de transmission".

- 2) Le Telnet utilisateur vérifie maintenant la mémoire tampon d'entrée du terminal, pour savoir si elle contient les données qui sont traitées à l'étape 4, autrement, le Telnet utilisateur atteint l'étape 3 pour d'autres développements.
- 3) Le Telnet utilisateur attend que l'utilisateur humain entre des données, auquel cas Telnet passe à l'étape 4, ou qu'un élément soit reçu du réseau. Si l'élément venant du réseau est des données, elles sont spontanément sorties et sont imprimées ; Telnet continue alors d'attendre. Si l'élément venant du réseau est une commande, une erreur s'est alors produite. Dans ce cas, le Telnet utilisateur peut tenter de resynchroniser l'utilisation de RCTE comme indiqué ci-dessous.

- 4) Les éléments provenant du terminal sont contrôlés à l'impression par le réglage de la plus récente commande de rétablissement de coupure. Lorsque un caractère de coupure est traité, le cycle de contrôle est achevé et l'action recommence à l'étape 1.

Les entrées provenant du terminal sont (on l'espère) mises en mémoire tampon dans des unités qui se terminent par un caractère de transmission ou de coupure, et l'écho du texte d'entrée est suspendu après la survenance d'un caractère de coupure jusqu'à réception d'une commande de rétablissement de coupure provenant de l'hôte serveur. La plus récente commande de rétablissement de coupure détermine les actions de coupure.

En résumé, ce qui est exigé est que pour chaque caractère de coupure envoyé dans la direction usager à serveur, il y ait une commande de rétablissement de coupure envoyée dans la direction serveur à usager. L'hôte utilisateur n'a initialement aucune connaissance des caractères qui sont de coupure et commence ainsi dans un état qui suppose qu'il n'y a pas de caractère de coupure, et qu'aussi aucun écho n'est à fournir. L'hôte serveur est supposé envoyer une commande de rétablissement de coupure pour établir les classes de coupure et le mode d'écho avant qu'il reçoive aucune donnée de l'utilisateur.

### **Synchronisation et resynchronisation :**

Les hôtes serveur et utilisateur doivent synchroniser soigneusement les commandes de rétablissement de coupure avec la transmission des caractères de coupure. Sauf au commencement d'une interaction, l'hôte serveur peut seulement envoyer une commande de rétablissement de coupure en réponse à l'envoi par l'hôte utilisateur d'un caractère de coupure comme défini à ce moment là. Cela devrait établir une correspondance biunivoque entre eux. (Une valeur <cmd> de zéro, dans ce contexte, est interprétée comme un rétablissement des classes de coupure à la ou les mêmes classes qu'auparavant.) La commande de rétablissement de coupure peut être précédée d'une sortie du terminal.

La resynchronisation des caractères de coupure et des commandes de rétablissement de coupure est faite via l'échange du signal Telnet Abort Output (AO, *interrompre la sortie*) dans la direction serveur à utilisateur et du signal SYNCH dans la direction utilisateur à serveur.

- Supposons que le serveur veuille resynchroniser les caractères de coupure et les commandes de rétablissement de coupure.
- Le serveur devrait être sûr que tous les résultats au terminal ont été imprimés en utilisant, par exemple, l'option Timing Mark.
  - Le serveur envoie le signal AO.
  - L'utilisateur reçoit le signal AO. L'utilisateur purge toutes les données d'utilisateur à serveur qu'il en ait été fait écho ou pas. L'utilisateur envoie un SYNCH au serveur. (Le SYNCH consiste en la Marque de données (DM, *Data Mark*) Telnet et en l'interruption (INS) d'hôte à hôte.) L'utilisateur entre maintenant dans l'état initial à l'étape 1.
  - Le serveur reçoit le SYNCH et purge toutes les données précédant le DM (comme toujours). Le serveur envoie maintenant une commande de rétablissement de coupure. (En fait, la commande de rétablissement de coupure pourrait être envoyée à tout moment après le AO.)

Supposons que l'utilisateur veuille resynchroniser les caractères de coupure et les commandes de rétablissement de coupure.

- L'utilisateur devrait éliminer toutes les données d'utilisateur à serveur qu'il en ait été fait ou non écho.
- L'utilisateur envoie le signal AO. L'utilisateur entre maintenant dans l'algorithme à l'étape 1.
- Le serveur reçoit le signal AO. Le serveur élimine toutes les données en mémoire tampon mais non encore envoyées à l'utilisateur. Le serveur envoie un SYNCH à l'utilisateur. Le serveur envoie une commande de rétablissement de coupure à l'utilisateur.

### **Notes et commentaires :**

Les commandes de numéro pair, supérieur à zéro, sont erronées, car elles vont avoir le bit de moindre poids à zéro. La commande devrait être interprétée comme égale à zéro, ce qui signifie que tous les octets de rétablissement de classes ([TC1 TC2] [BC1 BC2]) seront erronés. (Le IAC SE, à la fin de la commande, élimine tous les problèmes d'analyse dus à cette erreur.)

Les hôtes serveurs vont généralement donner pour instruction aux hôtes utilisateurs de ne pas faire écho aux caractères de coupure, même si il n'y aurait pas de problème à faire écho à la plupart des caractères de coupure. Par exemple, il n'y a généralement pas de problème à faire écho au caractère <cr> mais ce n'est pas le cas de <esc>. TENEX Exec est d'accord pour accepter l'un et l'autre, durant la spécification de nom de fichier. Donc, l'hôte utilisateur doit avoir pour instruction de ne faire écho à aucun caractère de coupure.

C'est généralement un problème tolérable, car de toutes façons, l'hôte serveur doit envoyer une commande RCTE à ce moment là. Ajouter au message un écho pour le caractère de coupure ne causera aucun trafic réseau supplémentaire.

L'option RCTE entraîne une redondance assez importante. Dans une vraie situation de caractère par caractère, cette redondance n'est pas justifiée. Mais en moyenne, elle devrait se traduire par des économies significatives, à la fois en trafic réseau et en réveils d'hôtes.

### **Problèmes de mise en mémoire tampon et de transmission contre contraintes d'impression :**

Il n'y a pas de contraintes de transmission obligatoires. Il est permis à l'hôte utilisateur d'envoyer un caractère à la fois, bien que ce soit un gaspillage du RCTE. Les commandes de classes de transmission sont des LIGNES DIRECTRICES, de sorte qu'il est permis de s'en écarter, comme lorsque la mémoire tampon de l'utilisateur est pleine.

De plus, l'hôte utilisateur peut envoyer un caractère de classe de coupure, sans savoir que c'en est un (comme avec la frappe continue).

Si la mise en œuvre d'utilisateur est habile, elle peut envoyer les données entrées par l'usager au serveur avant qu'elles ne soient réellement nécessaires. Ces données en frappe continue peuvent contenir des caractères de coupure.

Supposons que seule une espace soit un caractère de coupure (c'est-à-dire que c'est la dernière commande de rétablissement de coupure spécifiée imprimée jusques et y compris les caractères de coupure et réglons les classes de coupure à la classe 9). Supposons que l'usager a frappé "abc<espace>def<esc>ghi<cr>". Le RCTE côté usager peut envoyer tout cela au serveur, mais il pourrait imprimer seulement "abc<espace>", et aurait en mémoire tampon "def<esc>ghi<cr>" au moins jusqu'à ce que soit reçue une commande de rétablissement de coupure de la part du serveur. Cette commande de rétablissement de coupure pourrait changer les classes de coupure en exigeant de réexaminer la chaîne mise en mémoire tampon.

Par exemple, supposons que la commande de rétablissement de coupure règle les caractères de coupure à la classe 5 et l'action d'imprimer jusqu'au caractère de coupure non inclus. Le RCTE utilisateur pourrait alors imprimer "def" et éliminer le <esc>, mais devrait continuer de garder en mémoire tampon le "ghi<cr>".

Le problème avec la mise en mémoire tampon se produit lorsque l'impression sur le terminal de l'utilisateur doit être suspendue, après que l'usager a tapé un caractère de coupure actuellement valide et jusqu'à ce qu'une commande de rétablissement de coupure soit reçue de l'hôte serveur. Durant cet intervalle de temps, l'usager peut simplement continuer la frappe. Le texte tapé peut être ENVOYÉ, mais ne peut pas encore être IMPRIMÉ.

Le problème le plus courant du remplissage de la mémoire tampon de transmission, tout en attendant une allocation d'hôte à hôte de la part de l'hôte serveur, peut aussi se produire, mais ce problème est bien connu des mises en œuvre et n'est en aucune façon propre à RCTE.

Dans tous les cas, lorsque la mémoire tampon se remplit et que du nouveau texte tapé par l'usager va être perdu, cela devrait être notifié à l'usager (peut-être par une sonnerie sur le terminal).

Le texte devrait être mis en mémoire tampon par l'hôte utilisateur jusqu'à ce que l'usager tape un caractère qui appartient à la classe de transmission en vigueur au moment de la frappe du caractère.

Les commandes de rétablissement de classe de transmission peuvent être envoyées par l'hôte serveur à tout moment. Si elles sont fréquemment envoyées séparément des commandes de rétablissement de classe de coupure, il sera probablement préférable de sortir de RCTE et d'entrer un caractère régulier au moment de la transmission.

Ce que l'hôte utilisateur devrait faire avec le texte actuellement en mémoire tampon n'apparaît pas clairement dans l'immédiat, lorsque est reçue une commande de rétablissement des classes de transmission. La mise en mémoire tampon dépend du schéma de classes de transmission précédent.

L'hôte utilisateur ne devrait pas simplement attendre jusqu'à ce qu'un caractère de transmission (selon le nouveau schéma) soit tapé.

Le texte en mémoire tampon devrait être réexaminé, sous le nouveau schéma, ou il devrait simplement être envoyé comme un groupe. C'est la plus simple approche, et probablement assez adéquat.

Il est possible de définir AUCUN CARACTÈRE DE COUPURE sauf les commandes Telnet (IAC ...). Cela semble indésirable et ne devrait pas être fait.

Si cette situation devait se produire, l'hôte utilisateur devrait envoyer une commande Telnet pour permettre au serveur de savoir lorsque il peut rétablir les classes de coupure, mais le mécanisme est étrange et ce cas devrait être évité.

## 6. Exemple d'interaction

"S:" est envoyé de l'hôte serveur (WILL RCTE) à l'hôte utilisateur.

"U:" est envoyé de l'hôte utilisateur (DO RCTE) à l'hôte serveur.

"T:" est entré par l'utilisateur terminal.

"P:" est imprimé sur le terminal.

Le texte entouré de crochets ([ ]) est un commentaire.

Le texte entouré de crochets angulaires (< >) est à considérer comme une seule unité. Par exemple, le retour chariot est <cr>, et la valeur décimale 27 est représentée par <27>.

L'interaction suivante montre une connexion à un Tenex, l'initiation de l'éditeur DED, l'insertion de texte et le retour au niveau Exec.

On a tenté de donner une idée de l'asynchronisme de l'entrée/sortie réseau et de l'entrée du terminal de l'utilisateur. De nombreuses autres combinaisons possibles, utilisant le même ensemble d'actions que celles énumérées ci-dessous, pourrait être envisagées. L'ordre réel des événements va dépendre de la charge du réseau et des hôtes et de la vitesse de frappe de l'utilisateur.

On suppose que le Telnet de l'utilisateur est aussi un mode "insertion de saut de ligne". C'est-à-dire que chaque fois que l'usager tape un retour chariot <cr>, le Telnet utilisateur envoie à la fois le retour chariot et le saut à la ligne <cr><lf> (le signal de fin de ligne Telnet). Lorsque un caractère espace survient à la fin de la ligne dans la description de l'exemple, cela est montré explicitement par <sp> pour éviter les confusions. Les autres utilisations des caractères d'espace ne sont pas marquées ainsi pour éviter d'affecter la lisibilité de l'exemple.

Une connexion Telnet a déjà été ouverte, mais l'invite TENEX (*prompt*) n'a pas encore été produite. Les hôtes discutent d'abord de l'utilisation de l'option RCTE :

S: <IAC><WILL><RCTE>

U: <IAC><DO><RCTE>

S: TENEX 1.31.18, TENEX EXEC 1.50.2<cr><lf>@<IAC><SB><RCTE><11><1><24><IAC><SE>

[Imprimer l'en-tête et faire écho au texte entré jusqu'à un caractère de coupure, mais ne pas faire écho au caractère de coupure. Les classes 4 (opérateurs de format), 5 (commandes d'opérateurs non format et <DEL>), et 9 (<sp>) agissent comme des caractères de coupure.]

P: TENEX 1.31.18, TENEX EXEC 1.50.2<cr><lf>@

T: LOGIN ARPA<cr>

P: LOGIN

U: LOGIN<sp>

U: ARPA<cr><lf>

S: <sp><IAC><SB><RCTE><0><IAC><SE>

P: <sp>ARPA

S: <cr><lf>(PASSWORD): <IAC><SB><RCTE><7><IAC><SE>

P: <cr><lf>(PASSWORD):<sp>

T: WASHINGTON 1000<cr>

[Il n'est pas fait écho au mot de passe "WASHINGTON". On s'abstient d'imprimer "1000<cr>".]

U: WASHINGTON<sp>

U: 1000<cr><lf>

S: <sp><IAC><SB><RCTE><3><IAC><SE>

S: <cr><lf>JOB 17 ON TTY41 7-JUN-73 14:13<cr><lf>@ <IAC><SB><RCTE><0><IAC><SE>

P: <sp>1000

[L'impression est lente à ce moment, de sorte que le numéro de compte n'est pas imprimé tant que la commande du serveur n'est pas reçue pour lui.]

P: <cr><lf>JOB 17 ON TTY41 7-JUN-73 14:13<cr><lf>@

T: DED<esc><cr>

P: DED

U: DED<esc>

S: .SAV;1<IAC><SB><RCTE><0><IAC><SE>

P: .SAV;1

U: <cr><lf>

S: <cr><lf><lf>DED 3/14/73 DRO,KRK<cr><lf>: <IAC><SB><RCTE><15><1><IAC><255><IAC><SE>

[Le programme est lancé et l'invite DED ":" est envoyée. Au niveau de la commande, DED répond à chaque caractère. Le serveur établit les classes de coupure à toutes les classes.]

P: <cr><lf><lf>DED 3/14/73 DRO,KRK<cr><lf>:

T: IC'est une ligne d'essai.<cr>C'est une autre ligne d'essai.<^Z>Q

["I" signifie Insérer le texte. Le texte suit, terminé par un Control-Z. Le "Q" ordonne à DED de quitter.]

U: I

U: C'est une ligne d'essai.<cr><lf>

S: I<cr><lf>\*<IAC><SB><RCTE><11><0><24><IAC><SE>

[DED invite l'utilisateur, durant l'entrée du texte, avec un astérisque au début de chaque ligne. Le serveur règle les classes de coupure aux classes 4 et 5, et aux opérateurs de format et aux commandes d'opérateur de non format.]

P: I<cr><lf>\*C'est une ligne d'essai.

S: <cr><lf>\*<IAC><SB><RCTE><0><IAC><SE>

P: <cr><lf>\*C'est une autre ligne d'essai.

U: C'est une autre ligne d'essai.<^Z>

U: Q

[Noter que le "Q" ne va pas être immédiatement imprimé sur le terminal, car il doit attendre l'autorisation.]

S: ^Z<cr><lf>:<IAC><SB><RCTE><15><1><IAC><255><IAC><SE>

[Le "^Z" retourné fait deux caractères, et non le Control-Z ou <sub> ASCII.]

S: Q<cr><lf>@<IAC><SB><RCTE><11><1><24><IAC><SE>

P: Q<cr><lf>@

Et l'utilisateur retourne au niveau Exec.