Remote Controlled Transmission & Echoing TELNET Option

Currently, a terminal in character-at-a-time transmission and
foreign-host echo causes four Network Messages for each character
struck. (The character sent from local to foreign host; its RFNM;
the echoed character sent from the foreign to the local host; and its
RFNM.)

By eliminating most echoing (1/2 as many messages) and packaging the
characters into useful units (assuming an average of five character
per unit; therefore another 80 per cent reduction), it is believed
that almost a 90 per cent reduction in character-mode interactive
Network terminal traffic can be attained.

The packaging of characters and elimination of foreign echoing should
also lessen the load placed on the foreign hosts.

1. Command name and code:

   RCTE

2. Command meanings:

   IAC WILL RCTE

      The sender of this command REQUESTS or AGREES to use the RCTE
      option, and will send instructions for controlling the other
      side's terminal printer.

   IAC WON'T RCTE

      The sender of this option REFUSES to send instructions for
      controlling the other side's terminal printer.

   IAC DO RCTE

      The sender REQUEST or AGREES to have the other side (sender of
      WILL RCTE) issue commands which will control his (sender of the
      DO) output to the terminal printer.

   IAC DON'T RCTE

      The sender of this command REFUSES to allow the other side to
      control his (sender of DON'T) terminal printer.

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

      where:

         <cmd> is one 8-bit byte having the following flags (bits are
         counted from the right):

         Bit   Meaning

         0   0 = Ignore all other bits in this byte and repeat the
                     last <cmd> that was sent. Equals a 'continue what
                     you have been doing'.
                 1 = Perform actions as indicated by other bits in this
                     byte.

         1   0 = Print (echo) Break character
                 1 = Skip (don't echo) Break character

         2   0 = Print (echo) text up to Break character
                 1 = Skip (don't echo) text up to Break character

         3   0 = Continue using same classes of Break characters.
                 1 = The two 8-bit bytes following this byte contain
                     flags for the new Break classes.

         4   0 = Continue using same classes of Transmit characters.
                 1 = Reset Transmit classes according to the two bytes
                     following 1) the Break classes bytes, if the Break
                     classes are also being reset, or 2) this byte, if
                     the Break classes are NOT also being reset.

      Value (decimal) of the <cmd> byte and its meaning:

             0 = Continue what you have been doing

             1 = Print (echo) up to AND INCLUDING Break character

             3 = Print up to Break character and SKIP (don't echo)
                 Break character

             5 = Skip text (don't echo) up to Break character, but
                 PRINT Break character

> 7 = Skip up to and including Break character
>
> Add one of the previous non-zero values to one of the following values, to get the total decimal value for the byte (Note that Classes may not be reset without also resetting the printing action; so an odd number is guaranteed):
>
> 8 = Set Break classes (using the next two bytes [BC1 BC2])
>
> 16 = Set Transmission classes (using the next two bytes [TC1 TC2])
>
> 24 = Set Break classes (using the next two bytes [BC1 BC2]) and the Transmission classes (using the two bytes after that [TC1 TC2]).

Sub-commands (IAC SB RCTE...) are only sent by the Controlling Host and, in addition to other functions, functionally replace the Go-Ahead (IAC GA) Telnet Command.

3. Default:

WON'T RCTE -- DON'T RCTE

Neither host asserts special control over the other host's terminal printer.

4. Motivation for the option:

RFC's 1, 5 and 51 discuss Network and process efficiency and smoothness.

RFC 357, by John Davidson, introduces the problem of echoing delay that occurs when a remote user accesses a full-duplex host, thru a satellite link. In order to save the many thousands of miles of transit time for each echoed character, while still permitting full server responsiveness and clean terminal output, an echo control similar to that used by some Time-sharing systems is suggested for the entire Network.

In effect, the proposed option involves making a user host carefully regulate the local terminal printer according to explicit instructions from the foreign (serving) host.

An important additional issue is efficient Network transmission.
Implementation of the Davidson Echoing Scheme will eliminate almost
all server-to-user echoing.

The proposed option also requests using hosts to buffer a
terminal's input to the foreign host until it forms a useful unit
(with "useful unit" delimited by Break or Transmission characters
as described below). Therefore, fewer messages are sent on the
user-to-server path.

N.B.: This option is only intended for use with full-duplex hosts.
The Go-Ahead Telnet feature is completely adequate for HALF-duplex
server hosts.

5. Explicit description of control mechanism:

   A. Overview of Interaction

      (1) Agree to use RCTE option
      (2) User holds echo printing until instructed by server to do
          otherwise
      (3) Server may send output to terminal printer.
      (4) Network output is printed up to an RCTE command
      (5) Server sends IAC SB RCTE <cmd>
      (6) User acts upon the command up to a Break character or until
          receipt of output from the server host.
      (7) Go to (2)

   Note: Output from the server host may occur at any time, in which
         case, the flow of control switches to (2) and then proceeds to
         (3), (4), etc.

   B. Explanation:

      (1) Both Hosts agree to use the RCTE option. After that, the
          using host (IAC DO RCTE) merely acts upon the Controlling
          (serving) host's commands and does not issue any RCTE commands
          unless and until it (using host) decides to stop allowing use
          of the option (by sending IAC DON'T RCTE).

      (2) User host begins synchronization between the serving host and
          itself by suspending terminal echo printing until directed to
          do otherwise by the controlling host, thru an IAC SB RCTE
          <cmd>.

(3) The server may send output to the terminal printer, either in
    response to input from the user (in which case it is already
    synchronized with the terminal input) or spontaneously. In
    the latter case, flow of control automatically switches to (2)
    and continues from there. Output from the server is defined
    as completed when step (5) occurs. That is, text from the
    Server to the terminal printer MUST end with an RCTE command.

(4) Any output from the server is printed on the terminal
    IMMEDIATELY. Again note that the end of such output is
    defined to be the occurrence of an IAC SB RCTE <cmd> command.

(5) Server sends an RCTE command. The command may redefine Break
    and Transmission classes, Action to be performed on Break
    characters, and action to be performed on text. Each of these
    independent functions is controlled by separate bits in the
    <cmd> byte.

    a. A Transmission character is one which REQUIRES the User
       Host to transmit all text accumulated up to and including
       its occurrence. (For Net efficiency, User hosts are
       DISCOURAGED from sending before the occurrence of a
       Transmission character).

       If the Transmission Classes bit (Bit 4) is on, the two
          bytes following the two Break Classes bytes (or
          immediately following the <cmd> byte, if the Break
          Classes bit is not on) will indicate what classes are to
          be enabled.

       If the Bit is OFF, the Transmission classes remain
          unchanged. When the RCTE option is first initiated, NO
          CLASSES are in effect. That is, no character will be
          considered a Transmission character. (As if both TC1
          and TC2 are zero.)

    b. A Break character has the effect of a Transmission
       character, but also causes the User host to stop its
       print/discard action upon the User's input text, until
       directed to do otherwise by another IAC SB RCTE <cmd>
       command from the Serving host. Break characters therefore
       define printing units. "Break character" as used in this
       document does NOT mean Telnet Break character.

       If the Break Classes bit (Bit 3) is on, the two bytes
          following <cmd> will indicate what classes are to be
          enabled. There are currently nine (9) classes defined,
          with room for expansion.

If the bit is OFF, the Break classes remain unchanged.
When the RCTE option is initiated, CLASSES 4, 5, and 9
are to be in effect. That is, Format Effectors, Non-
format effector Control Characters and DEL, and
Punctuation characters are to be Break characters.

c. The list of the character classes, used to define Break and
Transmission classes are listed at the end of this
document, in the "Tables" Section.

d. Because Break characters are special, the print/discard
action that should be performed upon them is not always the
same as should be performed upon the rest of the input
text.

For example, while typing a filename to TENEX, I want the
text of the filename to be printed (echoed); but I do
not want the <escape> (if I use the name completion
feature) to be printed.

If Bit 1 is ON The Break character is NOT to be printed.

e. A separate bit (Bit 2) signals whether or not the text
itself should be printed (echoed) to the terminal. If Bit
2 = 0, then the text IS to be printed.

f. Yet another bit (Bit 0 - right-most bit) signals whether or
not any of the other bits of the command should be checked.
If this bit is OFF, then the command should be interpreted
to mean "continue whatever echoing strategy you have been
following, using the same Break and Transmission classes."

This is particularly useful for the <cmd> command that
follows spontaneously generated output from the Serving
host (such as "System Going Down") which needs to signal
End-of-Message, but does not usually want to reset any
other conditions.

The server may, however, alter user action after a
spontaneous message, but it is possible that text will
be lost, or printed when it should not be, since there
is no guarantee that the RCTE <cmd> from the serving
host will be properly synchronized with the terminal
input.

(6) Input from the terminal is (hopefully) buffered up to the occurrence of a Transmission or Break character; and the input text is echoed or not echoed, up to the occurrence of a Break Character. The most recent RCTE command determines the echo, Transmission and Break actions.

(7) When a Break character is typed, the cycle of control is complete and action re-commences at (2). Action also automatically switches to (2) upon receipt of any text from the Server host.

C. Notes, Comments, Etc.:

(1) Even-Numbered Commands, greater than zero, are in error, since they will have the low-order bit off. The command should be interpreted as equal to zero, which means that any Classes Reset bytes ([TC1 TC2] [BC1 BC2]) will be in error.

(2) Servers will generally instruct Users NOT to echo Break Characters, even though it might be alright to echo most Break characters. For example, <cr> is usually a safe character to echo but <esc> is not. TENEX Exec is willing to accept either, during filename specification. Therefore, the user must be instructed NOT to echo ANY Break Characters.

This is generally a tolerable problem, since the server has to send an RCTE command at this point, anyhow. Adding the Break character to the message (so that it appears to be echoed) will not cause any extra Network traffic.

(3) The RCTE Option entails a rather large overhead. In a true character-at-a-time situation, this overhead is not justified. But on the average, it should result in significant savings, both in Network traffic and Host wake-ups.

(4) A severe (User) site-dependent problem will be buffering type-ahead input from the terminal. It is possible, especially in the case of TIPS, that the input buffer will overflow often. If the receiving (serving) host will permit, the accumulated text should be transmitted at this point. If the text cannot be transmitted and further typing by the user will result in lost text, the user should be notified.

   D. Sample Interaction:

      "S:" is sent from serving (WILL RCTE) host to Using host.
      "U:" is sent from Using (DO RCTE) host to Serving host.
      "T:" is entered by the terminal user.
      "P:" is printed on the terminal.

      Text surrounded by square brackets ([]) is commentary.
      Text surrounded by angle brackets (<>) is to be taken as a single
         unit. E.G., carriage return is <cr>, and the decimal value 27
         is represented <27>.

      The following interaction shows a Logon to a Tenex, initiation of
      the DED editor, insertion of some text and return to the Exec
      level. A Telnet connection has already been opened, but the TENEX
      prompt has not yet been issued. The hosts first discuss using the
      RCTE option:

      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>

         [Print the Herald and echo input text upto a Break character,
         but do not echo the Break Character. Classes 4 (Format
         Effectors), 5 (Non-format effector Controls and <DEL>), and 9
         (<space>) act as Break Characters.]

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

      T: LOGIN ARPA <cr>

      P: LOGIN

      U: LOGIN <space>

      S: <space><IAC><SB><RCTE><0>

      P: <space>ARPA

      U: ARPA <cr>

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

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

T: WASHINGTON 1000<cr>

   [The password "WASHINGTON" is not echoed. Action on "1000<cr>"
   is withheld]

U: WASHINGTON <space>

S: <space><IAC><SB><RCTE><3>

P: <space> 1000

U: 1000<cr>

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

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>

P: .SAV;1

U: <cr>

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

   [The program is started and the DED prompt ":" is sent. At the
   command level, DED responds to every character.]

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

T: IThis is a text line.<cr> This is another test line.<^Z> Q

   ["I" means Insert Text. The text follows, terminated by a
   Control-Z. The "Q" instructs DED to Quit.]

U: I

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

[DED prompts the user, during text input, with an asterisk at
the beginning of every line.]

P: I<cr><lf> *This is a test line.

U: This is a test line.<cr>

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

P: <cr><lf>* This is another test line.

U: This is another test line.<^Z>

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

[The returned "^Z" is two characters, not the ASCII Control-Z.]

U: Q

[Note that the "Q" is not yet printed on the terminal, since it
is a Break character.]

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

P: Q<cr><lf> @

And the user is returned to the Exec level.

E. Tables:

(1) <cmd> is one 8-bit byte having the following flags (bits are
    counted from the right):

    Bit Meaning

    0   0 = Ignore all other bits in this byte and repeat the last
            <cmd> that was sent. Equals a 'continue what you have
            been doing'.

        1 = Perform actions as indicated by other bits in this
            byte.

    1   0 = Print (echo) Break character
        1 = Skip (don't echo) Break character

2   0 = Print (echo) text up to Break character
     1 = Skip (don't echo) text up to Break character

3   0 = Continue using same classes of Break characters.
     1 = The two 8-bit bytes following this byte contain flags
         for the new Break classes.

4   0 = Continue using same classes of Transmit characters

     1 = Reset Transmit classes according two the two bytes
         following 1) the Break classes bytes, if the Break
         classes are also being reset, or 2) this byte, if the
         Break classes are NOT also being reset.

Byte Value (decimal) and its meaning:

0 = Continue what you have been doing

Even numbers greater than zero (i.e., numbers with the right-
    most bit off) are in error and should be interpreted as
    equal to zero. When the <cmd> is an even number greater
    than zero, Classes bytes TC1 & TC2 and/or BC1 & BC2 must
    not be sent.

1 = Print (echo) up to AND INCLUDING Break character

3 = Print up to Break character and SKIP (don't echo) Break
    character

5 = Skip text (don't echo) up to Break character, but PRINT
    Break character

7 = Skip up to and including Break character

Add one of the previous non-zero values to one of the following
    values, to get the total decimal value for the byte (Note
    that Classes may not be reset, without also resetting the
    printing action; so an odd number is guaranteed):

8 = Set Break classes (using the next two bytes [BC1 BC2])

16 = Set Transmission classes (using the next two bytes [TC1
     TC2])

24 = Set Break classes (using the next two bytes [BC1 BC2]) and
     the Transmission classes (using the two bytes after that
     [TC1 TC2]).

(2) Classes for Break and Transmission (The right-most bit of the
    second byte (TC2 or BC2) represents Class 1; the left-most bit
    of the first byte (TC1 or BC1) represents the currently
    undefined Class 16):

    1: Upper-Case Letter (A-Z)

    2: Lower-case letters (a-z)

    3: Numbers (0-9)

    4: Format Effectors (<BS> <CR> <LF> <FF> <HT> <VT>)

    5: Non-format effectors Control Characters, <DEL> and <ESC>

    6: . , ; : ? !

    7; - [(< >)] |

    8: ' " / \ % @ $ # + - * = ^ <- _ (square box symbol)

    9: <space>

    And Telnet commands (IAC...) are ALWAYS to have the effect of
    a Break character.

   [This RFC was put into machine readable form for entry]
        [into the online RFC archives by Via Genie]