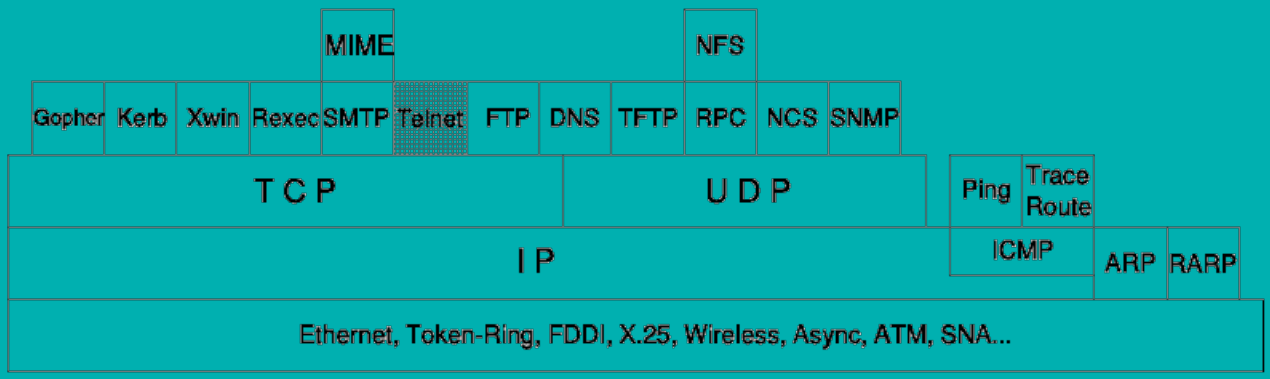
THE TELNET PROTOCOL



- TELNET is a standard protocol. Its status is recommended.
- It is described in RFC 854 - TELNET Protocol Specifications and RFC 855 - TELNET Option Specifications.
- Telnet was the first application demonstrated on the four-IMP (Interface Message Processor) network installed by December 1969. The final edition took 14 more years to develop, culminating in Internet Standard #8 in 1983, three years after the final TCP specification was ratified.
- Telnet even predates internetworking and the modern IP packet and TCP transport layers.
- The TELNET protocol provides a standardized interface, through which a program on one host (the TELNET client) may access the resources of another host (the TELNET server) as though the client were a local terminal connected to the server.
- For example, a user on a workstation on a LAN may connect to a host attached to the LAN as though the workstation were a terminal attached directly to the host. Of course, TELNET may be used across WANs as well as LANs.
- Most TELNET implementations do not provide you with graphics capabilities.

TELNET Overview

- TELNET is a general protocol, meant to support logging in from almost any type of terminal to almost any type of computer.
- It allows a user at one site to establish a TCP connection to a login server or terminal server at another site.
- A TELNET server generally listens on TCP Port 23.

How it works

- A user is logged in to the local system, and invokes a TELNET program (the TELNET client) by typing

  ```
  telnet  xxx.xxx.xxx
  ```

  where xxx.xxx.xxx is either a host name or an IP address.
- The TELNET client is started on the local machine (if it isn't already running). That client establishes a TCP connection with the TELNET server on the destination system.
- Once the connection has been established, the client program accepts keystrokes from the user and relays them, generally **one character at a time**, to the TELNET server.
- The server on the destination machine accepts the characters sent to it by the client, and passes them to a terminal server.
- A "terminal server" is just some facility provided by the operating system for entering keystrokes from a user's keyboard.

    - The terminal server treats the remote user as it would any other user logged in to the system, including relaying commands to other applications.
    - The terminal server passes outputs back to the TELNET server, which relays them to the client, which displays them on the user's screen.
- In general, a TELNET server is implemented as a master server with some number of slave servers. The master server listens for service requests from clients. When it hears one, it spawns a slave server to handle that specific request, while the master goes back to listening for more requests.
- The only thing that makes TELNET hard to implement is the heterogeneity of the terminals and operating systems that must be supported. Not all of them use the same control characters for the same purposes.
- To accomodate this heterogeneity, TELNET defines a Network Virtual Terminal (NVT). Any user TELNETting in to a remote site is deemed to be on an NVT, regardless of the actual terminal type being used.
- It is the responsibility of the client program to translate user keystrokes from the actual terminal type into NVT format, and of the server program to translate NVT characters into the format needed by the destination host. For data sent back from the destination host, the translation is the reverse.
- NVT format defines all characters to be 8 bits (one byte) long. At startup, 7 bit US ASCII is used for data; bytes with the high order bit = 1 are command sequences.
- The 128 7-bit long US ASCII characters are divided into 95 printable characters and 33 control codes. NVT maps the 95 printable characters into their defined values - decimal 65 = "A", decimal 97 = "a", etc.
- The 33 control codes are defined for NVT as:

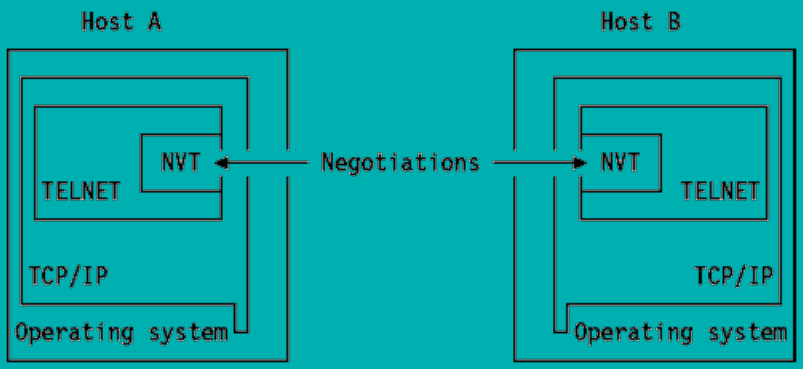
ASCII Code	Decimal value	Meaning
NUL	0	NO - OP
BEL	7	Ring "terminal bell"
BS	8	Backspace; move cursor left
HT	9	Horizontal tab; move cursor right
LF	10	Line feed; move down one line; stay in same column
VT	11	Vertical tab; move cursor down
FF	12	Form Feed
CR	13	Carriage return; move cursor to beginning of current line
all others		NO - OP

- NVT defines end-of-line to be a CR-LF combination - the two-character sequence.
- In addition to the 128 characters mentioned above, there are 128 other possible characters in an 8-bit encoding scheme. NVT uses these 128 (with decimal values 128 through 255, inclusive) to pass control functions from client to server. More on this later.

TELNET Operation

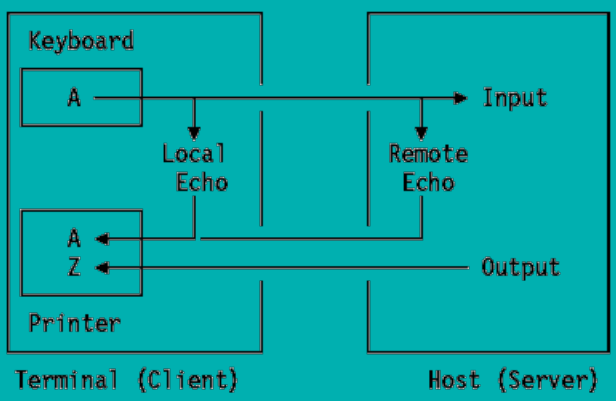
- The TELNET protocol is based on three ideas:
    - The Network Virtual Terminal (NVT) concept. An NVT is an imaginary device having a basic structure common to a wide range of real terminals. Each host maps its own terminal characteristics to those of an NVT, and assumes that every other host will do the same.
    - A symmetric view of terminals and processes .
    - Negotiation of terminal options. The principle of negotiated options is used by the TELNET protocol, because many hosts wish to provide additional services, beyond those available with the NVT. Various options may be negotiated. Server and client use a set of conventions to establish the operational characteristics of their TELNET connection via the ``DO, DON'T, WILL, WON'T'' mechanism discussed later in this document.
- The two hosts begin by verifying their mutual understanding. Once this initial negotiation is complete, they are capable of working on the minimum level implemented by the NVT.
- After this minimum understanding is achieved, they can negotiate additional options to extend the capabilities of the NVT to reflect more accurately the capabilities of the real hardware in use.
- Because of the symmetric model used by TELNET, both the host and the client may propose additional options to be used.
- The set of options is not part of the TELNET protocol, so that new terminal features can be

incorporated without changing the TELNET protocol (mouse?).
- All TELNET commands and data flow through the same TCP connection.
- Commands start with a special character called the Interpret as Command escape character (IAC).
- The IAC code is 255.
- If a 255 is sent as data - it must be followed by another 255
- Each receiver must look at each byte that arrives and look for IAC. If IAC is found and the next byte is IAC - a single byte is presented to the application/terminal.
- If IAC is followed by any other code - the TELNET layer interprets this as a command.



The NVT (Network Virtual Terminal) concept

- The NVT has a "printer" (or display) and a "keyboard".
- The keyboard produces outgoing data, which is sent over the TELNET connection. The printer receives the incoming data.
- The basic characteristics of an NVT, unless they are modified by mutually agreed options are:
  - The data representation is 7-bit ASCII transmitted in 8-bit bytes.
  - The NVT is a half-duplex device operating in a line-buffered mode.
  - The NVT provides a local echo function.
- All of these may be negotiated by the two hosts.



- An NVT Printer has an unspecified carriage width and page length. It can handle printable ASCII characters (ASCII code 32 to 126) and understands some ASCII control characters such as:

Command	ASCII	Action
NULL (NUL)	0	No Operation.
Line Feed (LF)	10	Moves the printer to the next print line, keeping the same horizontal position.
Carriage Return (CR)	13	Moves the printer to the left margin.
BELL (BEL)	7	Produces an audible or visible signal.
Back Space (BS)	8	Moves the print head one character position toward the left margin.
Horizontal Tab (HT)	9	Moves the printer to the next horizontal tab stop.
Vertical Tab (VT)	11	Moves the printer to the next vertical tab stop.
Form Feed (FF)	12	Moves the printer to the top of the next page, keeping the same horizontal position.

TELNET Options

- There is an extensive set of TELNET options, and the reader should consult tho Official Internet Protocol Standards for the standardization state and status for each of them.
- At the time of writing, the following options were defined:

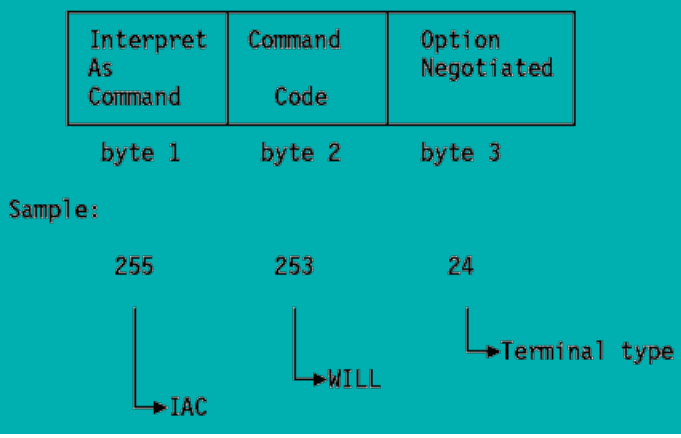
Num	Name	State	RFC	STD
255	Extended-Options-List	Standard	861	32
0	Binary Transmission	Standard	856	27
1	Echo	Standard	857	28
3	Suppress Go Ahead	Standard	858	29
5	Status	Standard	859	30
6	Timing Mark	Standard	860	31
34	Linemode	Draft	1184	
2	Reconnection	Proposed		
4	Approx Message Size Negotiation	Proposed		
7	Remote Controlled Trans and Echo	Proposed	726	
8	Output Line Width	Proposed		
9	Output Page Size	Proposed		
10	Output Carriage-Return Disposition	Proposed	652	
11	Output Horizontal Tabstops	Proposed	653	
12	Output Horizontal Tab Disposition	Proposed	654	
13	Output Formfeed Disposition	Proposed	655	
14	Output Vertical Tabstops	Proposed	656	
15	Output Vertical Tab Disposition	Proposed	657	
16	Output Linefeed Disposition	Proposed	658	
17	Extended ASCII	Proposed	698	

Num	Name	State	RFC	STD
18	Logout	Proposed	727	
19	Byte Macro	Proposed	735	
20	Data Entry Terminal	Proposed	1043	
21	SUPDUP	Proposed	736	
22	SUPDUP Output	Proposed	749	
23	Send Location	Proposed	779	
24	Terminal Type	Proposed	1091	
25	End of Record	Proposed	885	
26	TACACS User Identification	Proposed	927	
27	Output Marking	Proposed	933	
28	Terminal Location Number	Proposed	946	
29	TELNET 3270 Regime	Proposed	1041	
30	X.3 PAD	Proposed	1053	
31	Negotiate About Window Size	Proposed	1073	
32	Terminal Speed	Proposed	1079	
33	Remote Flow Control	Proposed	1372	
35	X Display Location	Proposed	1096	
39	TELNET Environment Option	Proposed	1572	
37	TELNET Authentication Option	Experimental	1416	

- All of the standard options have a status of recommended and the remainder have a status of elective.

TELNET Command Structure

- The communication between client and server is handled with internal commands, *which are not accessible by users*.
- All internal TELNET commands consist of 2 or 3-byte sequences, depending on the command type.
- The Interpret As Command (IAC) character is followed by a command code. If this command deals with option negotiation, the command will have a third byte to show the code for the referenced option.



- The key point is that, in order for any byte to be accepted as a command, it must be preceded by a byte with value 255 - "Interpret as Command" (IAC). Otherwise, the server program assumes that that byte is simply data that will be understood by the application program.
- This command proposes negotiation about terminal type.

Command name	Code	Comments
SE	240	End of sub-negotiation parameters.
NOP	241	No operation.
Data Mark	242	The data stream portion of a synch. This should always be accompanied by a TCP urgent notification.
Break	243	NVT character BRK.
Go ahead	249	The GA signal.
SB	250	Indicates that what follows is sub-negotiation of the option indicated by the immediately following code.
WILL	251	Shows the desire to use, or confirmation that you are now using, the option indicated by the code immediately following.
WON'T	252	Shows the refusal to use, or to continue to use, the option indicated by the code immediately following.
DO	253	Requests that the other party uses, or confirms that you are expecting the other party to use, the option indicated by the code immediately following.
DON'T	254	Demands that the other party stop using, or confirms that you are no longer expecting the other party to use, the option indicated by the code immediately following.
IAC	255	Interpret As Command. Indicates that what follows is a TELNET command, not data.

Option Negotiation

- Using internal commands, TELNET in each host is able to negotiate options.
- The starting base of negotiation is the NVT capability: each host to be connected must agree to this minimum.
- Every option can be negotiated by the use of the four command codes WILL, WON'T, DO, DON'T described above.
- In addition, some options have sub-options: if both parties agree to the option, they use the SB and SE commands to manage the sub-negotiation.
- Here is a simplified example of how option negotiation works.

Send	Reply	Meaning
DO transmit binary	WILL transmit binary	
DO window size	WILL window size	Can we negotiate window size?
SB Window Size 0 80 0 24 SE		Specify window size
DO terminal type	WILL terminal type	Can we negotiate terminal type?
SB terminal type SE		Send me your terminal characteristics.
	SB terminal type IBM-3278-2 SE	My terminal is a 3278-2
DO echo	WONT echo	

- To use an option, the client and server must negotiate and agree to use it. The tools for negotiation are the commands we've already talked about. One side - usually, but not always, the client - sends a "WILL X" packet (WILL is decimal value 251), where X is the option it wants to use (numeric values for X are given on p. 373). The other side will respond with a DO X or a DON'T X, depending on whether it is willing to support the option.
- Alternatively, the first side could send a "DO X" packet, in which case the response is either "WILL X" or "WON'T X".
- TELNET is one of the programs that requires the TCP Urgent Data function, because buffers may fill up (for example, if a program being executed is in an infinite loop), and the server's program will stop reading data - including the "IP" command the user sent after he realized what was happening. The packet with the "Terminate" command can be sent as "urgent data" at the TCP level; that will bypass the standard TCP flow controls and enable an out-of-control process to be stopped.

TELNET Basic Commands

- The primary goal of the TELNET protocol is the provision of a standard interface for hosts over a network.
- To allow the connection to start, the TELNET protocol defines a standard representation for some functions:

```
IP     Interrupt Process
AO     Abort Output
AYT    Are You There
EC     Erase Character
EL     Erase Line
SYNCH  Synchronize
QUIT   quit session
```

Implementation on DOS

- TCP/IP for DOS provides three TELNET client implementations:
  - DOS TELNET: To be used from the DOS command line.
  - DOS TN3270: To be used from the DOS command line.
  - Windows TELNET: To be used through the Windows interface.
- TCP/IP for DOS does not provide a TELNET Server implementation.
- The SETTERM command allows the configuration of a DOS TELNET client. This program is menu driven and allows the following settings:
  - Key assignment: Specifies how keys are interpreted for the currently defined session. For example, PF8 can be assigned to the Down function.
  - Character translations: Specifies how a particular character is translated. The decimal value (in the range of 0 to 255) for the byte to be translated must be entered. After this value is supplied, the system prompts for the replacement value (in the range of 0 to 255).
  - Video attributes: Specifies foreground, background, and highlighting attributes.
  - Terminal choices: Specifies the order of preference for terminals (five emulators are available) to use during a TELNET session.
  - Session hot-key definitions: TELNET can open up to eight different sessions. You can assign a hot-key to each session. The session hot key is used to quickly switch among active sessions.
- The modifications are stored in an ASCII file which can be referenced when the TELNET command is issued.
- It is possible to have different configuration files for different settings.
- TELNET offers four terminal emulators:
  - VT220
  - VT100
  - IBM-3278-2
  - ANSITERM
- Windows TELNET offers six terminal emulators:
  - VT220

  - VT100
  - IBM-3278-2 through 3278-5
  - ANSI
  - TTY
  - 5250
- The terminal preference is generally handled through the SETTERM command but it can be specified with the TELNET command.
- All the TELNET sessions must be opened from a full-screen window.
- TELNET provides a menu-driven interface that makes it easy to supply the information needed to begin a TELNET session.

Yet another review of TELNET

- During the connection, enhanced characteristics other than those offered by the NVT may be negotiated either by the user or the application. This task is accomplished by embedded commands in the data stream. TELNET command codes are one or more octets in length and are preceded by an interpret as command (IAC) character, which is an octet with each bit set equal to one (FF hex). The following are the TELNET command codes:

Commands	Code No. Dec	Hex	Description
data			All terminal input/output data.
End subNeg	240	FO	End of option subnegotiation command.
No Operation	241	F1	No operation command.
Data Mark	242	F2	End of urgent data stream.
Break	243	F3	Operator pressed the Break key or the Attention key.
Int process	244	F4	Interrupt current process.
Abort output	245	F5	Cancel output from current process.
You there?	246	F6	Request acknowledgment.
Erase char	247	F7	Request that operator erase the previous character.
Erase line	248	F8	Request that operator erase the previous line.
Go ahead!	249	F9	End of input for half-duplex connections.
SubNegotiate	250	FA	Begin option subnegotiation.
Will Use	251	FB	Agreement to use the specified option.
Won't Use	252	FC	Reject the proposed option.
Start use	253	FD	Request to start using specified option.
Stop Use	254	FE	Demand to stop using specified option.
IAC	255	FF	Interpret as command.

- Each negotiable option has an ID, which immediately follows the command for option negotiation, that is, IAC: command, option code. Following is a list of TELNET option codes:

Option ID Dec	Hex	Option Codes	Description
0	0	Binary Xmit	Allows transmission of binary data.
1	1	Echo Data	Causes server to echo back all keystrokes.
2	2	Reconnect	Reconnects to another TELNET host.
3	3	Suppress GA	Disables Go Ahead! command.
4	4	Message Sz	Conveys approximate message size.
5	5	Opt Status	Lists status of options.
6	6	Timing Mark	Marks a data stream position for reference.
7	7	R/C XmtEcho	Allows remote control of terminal printers.
8	8	Line Width	Sets output line width.
9	9	Page Length	Sets page length in lines.
10	A	CR Use	Determines handling of carriage returns.
11	B	Horiz Tabs	Sets horizontal tabs.
12	C	Hor Tab Use	Determines handling of horizontal tabs.
13	D	FF Use	Determines handling of form feeds.
14	E	Vert Tabs	Sets vertical tabs.
15	F	Ver Tab Use	Determines handling of vertical tabs.
16	10	Lf Use	Determines handling of line feeds.
17	11	Ext ASCII	Defines extended ASCII characters.
18	12	Logout	Allows for forced log-off.
19	13	Byte Macro	Defines byte macros.
20	14	Data Term	Allows subcommands for Data Entry to

			be sent.
21	15	SUPDUP	Allows use of SUPDUP display protocol.
22	16	SUPDUP Outp	Allows sending of SUPDUP output.
23	17	Send Locate	Allows terminal location to be sent.
24	18	Term Type	Allows exchange of terminal type information.
25	19	End Record	Allows use of the End of record code (0xEF).
26	1A	TACACS ID	User ID exchange used to avoid more than 1 log-in.
27	1B	Output Mark	Allows banner markings to be sent on output.
28	1C	Term Loc#	A numeric ID used to identify terminals.
29	1D	3270 Regime	Allows emulation of 3270 family terminals.
30	1E	X.3 PAD	Allows use of X.3 protocol emulation.
31	1F	Window Size	Conveys window size for emulation screen.
32	20	Term Speed	Conveys baud rate information.
33	21	Remote Flow	Provides flow control (XON, XOFF).
34	22	Linemode	Provides linemode bulk character transactions.
255	FF	Extended	options list Extended options list.

TELNET vs. telnet

- **TELNET** is a protocol that provides “a general, bi-directional, eight-bit byte oriented communications facility”.
- **telnet** is a program that supports the TELNET protocol over TCP. Put in other words: telnet is a TELNET client.
- Many application protocols are built upon the TELNET protocol

rlogin and rsh

- rlogin is a UNIX protocol that provides some enhancements to TELNET. Among other things, rlogin allows "trusted hosts", from which users are not required to enter passwords when logging in. (One can get into interesting discussions at network security conferences about whether that's a good idea or not.)
- There are variants of rlogin - for example, rsh - run a shell program on the remote system. The general syntax is

```
rsh   machine_to_run_command_on  command_to_be_run
```

- The point is that rlogin, rsh, etc. understand the operating system, file system, etc. on both the client and server machines, and can take advantage of them in ways that TELNET cannot. For example, rlogin understands UNIX standard input, standard output, and standard error. Thus, you can do things like

```
rsh machine  command > filename
```

  and the behavior will be pretty much what you expect.
- **rlogin** also understands things like Ctrl-Q, Ctrl-S, etc. It can take advantage of this understanding by doing things on the client machine, without waiting for the delay induced by sending the traffic across the network.
- For example: Ctrl-S stops printing of output to the screen. When you are rlogin-ed to another host, and hit Ctrl-S, your local machine will stop printing - you won't have to wait for the command to be sent to the distant machine, processed, and then take effect.
- Also, rlogin does not restrict you to a simple terminal type like NVT. It exports part of your environment - including your terminal type - to the remote machine, which can take advantage of it.