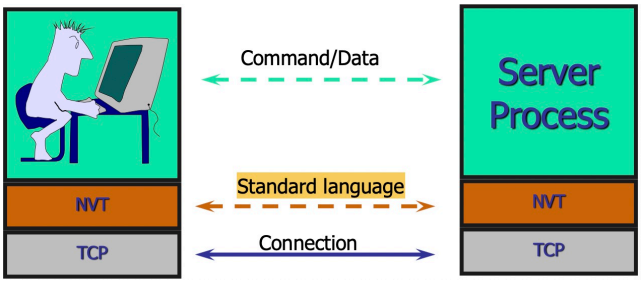
TELNET

TELNET and telnet
TELNET
TELNET: a protocol used to establish a dumb terminal session to another computer on the Internet (dumb terminal: 功能较有限的终端)
Purpose of TELNET: provide a general, bi-directional, byte oriented communications facility

TELNET — protocol (provide a general, bi-directional, byte oriented communications facility)
telnet — program (supports TELNET protocol over TCP)

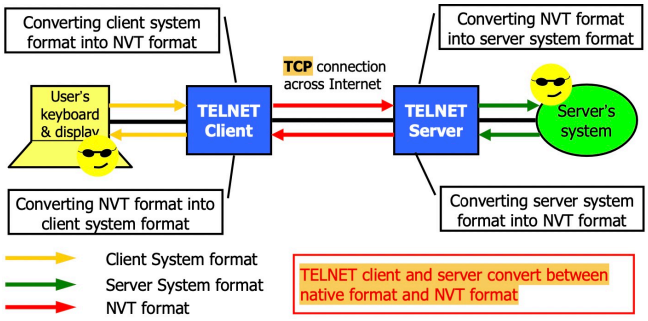
Motivation and problem
Motivation: remote access to interactive system
Problem: lack of common language between the terminal and the remote host

NVT
Network Virtual Terminal (NVT)
作用: transform local characteristics into standardized form
- Imaginary device
- Both sides generate data and control signals in native language but translates them to NVT form



Functions:
- The sending side translates native data and control signal —> NVT form before send out
- The receiving side gets the data and control signal in NVT form —> native form

NVT Operation
- Accommodating heterogeneity



TELNET Protocol
TELNET Features
- TCP connection: server **port 23**
- Data and control multiplexed over the same connection (混在一起传输)
- **Negotiated options** — Enabling Telnet to evolve to meet new demands without endless new version of basic protocol (advantage)
- A symmetric view of terminals and processes

- Transmission of data
- Telnet control functions

Transmission of Data
- Underlying TCP full duplex
- Data sent half duplex (communication one direction at a time)
- Data sent as stream of 8-bit bytes
- Control signal and other non-data information sent as Telnet commands (byte stream embedded in data stream)

Telnet control functions
- Extra virtual keys in the NVT keyboard
- IAC, DONT, DO, WONT, WILL

Command	Decimal Codes	Description
IAC	255	Interpret next octet as command
DONT	254	Denial of request to perform specific option
DO	253	Approval to allow specific option
WONT	252	Refusal to perform specific option
WILL	251	Agreement to perform specific option

Control Functions — IAC
TELNET command structure
At least a two byte sequence: IAC followed by the code for the command
IAC code is 255, (Look for command):
1. If IAC is found and the next byte is "IAC" - **a single data byte** (value 255) is presented to the application/ terminal (IAC + 255 —> 1个255data)
2. If IAC is followed by any other code — the TELNET layer interprets this as a command

Control Functions — DO, DONT, WILL, WONT

Sender	Receiver	Meaning
WILL	DO	Sender wants to active an option, and receiver agrees
WILL	DONT	Sender wants to active an option, and receiver refuses
DO	WILL	Sender wants receiver to active an option, and receiver agrees
DO	WONT	Sender wants receiver to active an option, and receiver refuses

TELNET Options Negotiation
Motivation: all NVTs support a minimal set of capabilities

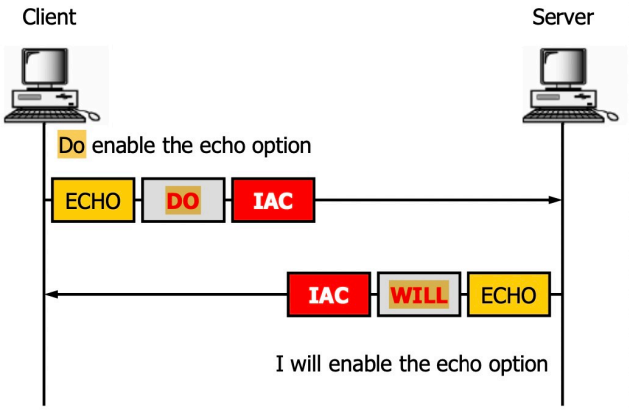
Some terminals have more capabilities than the minimal set, the two endpoints negotiate a set of mutually acceptable options (character set, echo mode, etc). The set of options is not part of the TELNET protocol, so that new terminal features can be incorporated without changing the TELNET protocol

1. More capabilities
2. Negotiate mutually acceptable options
3. These options are not part of the TELNET protocol
4. Result: add new features without change the TELNET protocol

Option Examples
- Echo modes (1)
- Binary transmission (0)
- Line mode v.s. character mode (34)
- Character set

Echo modes: keyboard input be echoed on the terminal side or not
Line mode v.s. character mode: one line or one character per transmission
Character set (EBCDIC v.s. ASCII):
- EBCDIC — Extended Binary-Coded Decimal Interchange Code
- ASCII — American Standard Code for Information Interchange

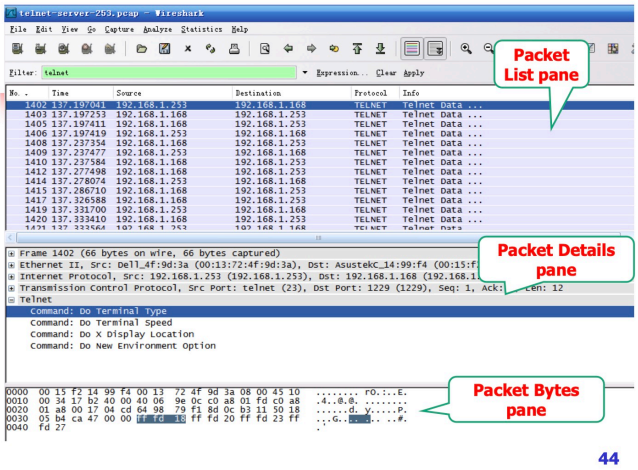
Option Negotiation (过程)
- Each option is assigned a byte value
- Commands are used to negotiate options (DO, DONT, WILL, WONT)
- Subnegotiations: used when more information is needed, such as when negotiating terminal type (24), window size (31) ...



解释: Client wants server to echo, server agrees.

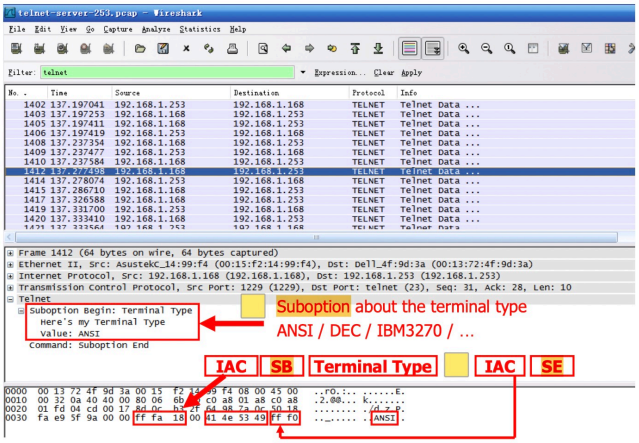
Common option codes
- Binary Transmission — 0
- Echo — 1
- Line mode — 34
- Terminal Type — 24
- Window Size — 31

Example 1 (Wireshark)



解析: (关注点)

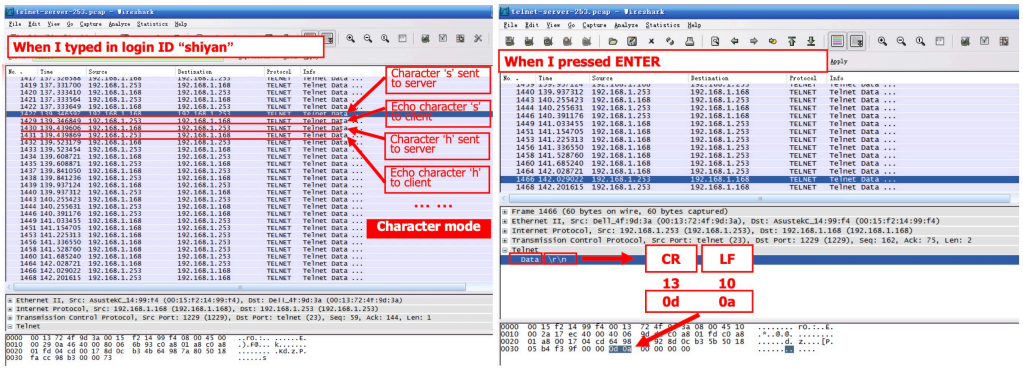
1. Transmission Control Protocol, src port: 23 —> well-known port, 从server端发出的 packet
2. DO Terminal Type —> ff fd 18
3. 讲16进制转为10进制, ff —> 255 (IAC), fd —> 253 (DO), 18 —> 24 (Terminal type)
4. 解释：server端想讨论Terminal Type (active an option)



解析: (关注点)

1. 子协商从client端发出，server端接收 (Dst: port 23)
2. 子协商SB开始，SE结束
3. 41 43 53 49 —> ANSI (terminal type)
4. Explain: The terminal type is ANSI

Example 2 (Wireshark)



Type Login ID 的过程: client send one character to server, server echos back one character (Character mode)

od oa —> 13 10 —> CR LF —> "\r\n" (回车， 换行)