

Telnet

10.2.1.1

Протокол Telnet (TErminaL NETwork) (основное RFC -- RFC 854) реализует концепцию NVT (Network Virtual Terminal), уходящую корнями в UNIX-системы.

Первые UNIX-системы были «большими» и строились согласно модели сильносвязанных КС. Физические терминалы в таких системах подключались по выделенным каналам (расстояние могло быть несколько десятков метров) и поэтому их стали называть TTYs (TeleTYpes), хотя по сути они были локальными в современном понимании.

Виртуальные терминалы, то есть NVTs, представляют собой удаленные от хоста программные эмуляторы физических терминалов, связь с которыми «протянута» через СПД.

10.2.1.2

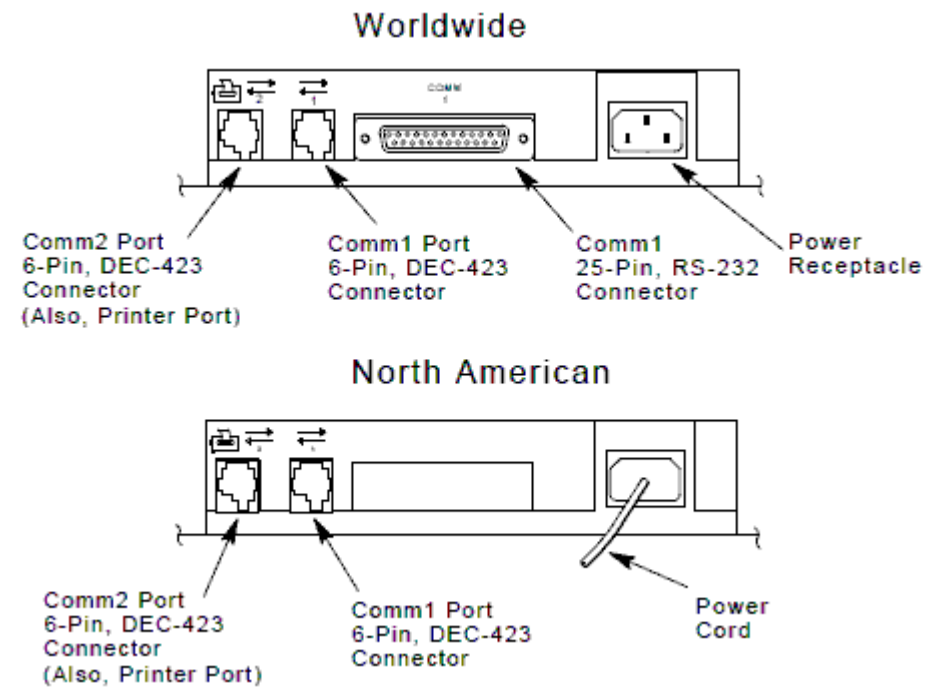
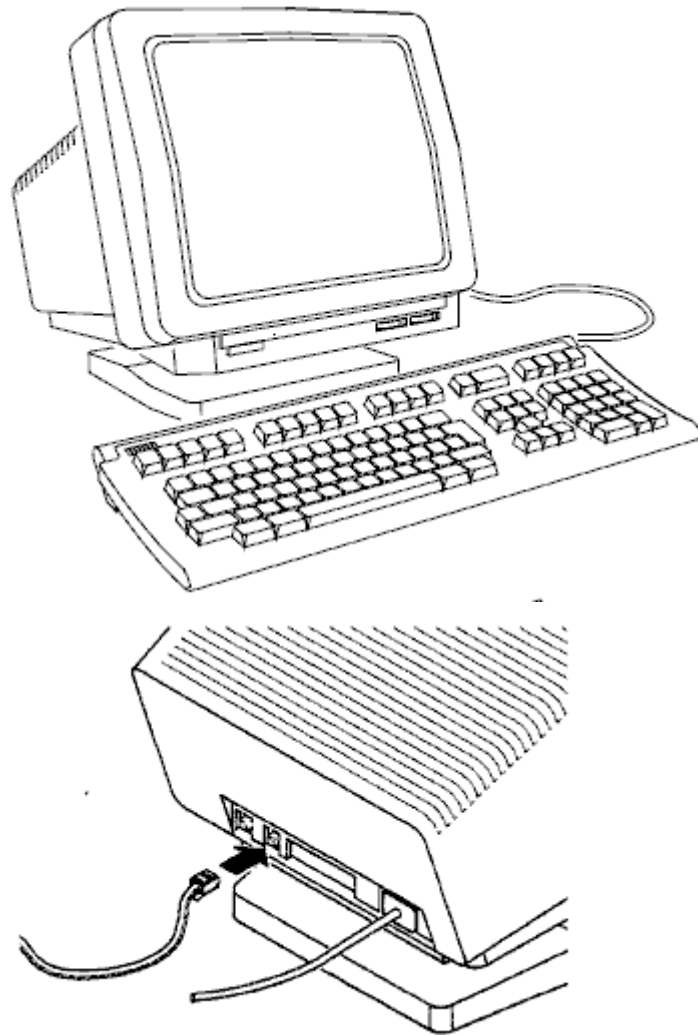
Изначально терминалы были текстовыми.

Текстовые терминалы **различают исходя из** набора поддерживаемых клавиш (основной набор, расширенный и так далее), текстового разрешения (основное -- 80x25) и некоторых других характеристик.

NVTs эмулируют основные виды текстовых терминалов.

Наиболее распространенными были следующие текстовые терминалы (выпускалось множество аналогов): DEC VT100, VT220, VT420, VT520 и IBM TN3270.

10.2.1.3



DEC VT420 [DEC]

10.2.1.4

Почти все современные терминалы -- графические.

Соответственно, графические терминалы **различают** в основном **исходя из** графического разрешения.

Удаленные графические терминалы UNIX-систем **подключают** по протоколу XDMCP (X Display Manager Control Protocol).

UNIX-программа Xterm -- стандартный эмулятор текстового терминала на графическом (на одном графическом терминале можно эмулировать сразу несколько текстовых).

10.2.1.5

Следует отметить, что хост-терминальные системы **не «вымерли»**, а **их** до сих пор достаточно широко **применяют**, особенно где беспроводные подключения не подходят по соображениям безопасности.

Усовершенствованные, значительно более компактные физические терминалы сейчас **называют** тонкими клиентами (thin clients).

10.2.1.6



HP t520 [HP]

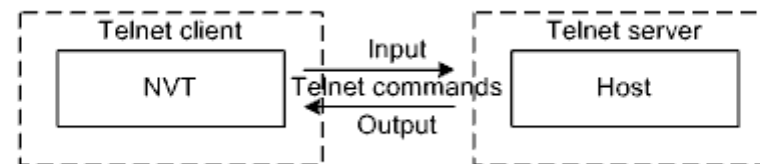
10.2.1.7

Telnet базируется на клиент-серверной модели и использует транспорт TCP.

Задействуется одно соединение.

Стандартный номер программного порта Telnet-сервера -- 23.

10.2.1.8



Структура системы Telnet

10.2.1.9

Основная задача протокола Telnet заключается в обеспечении корректной транспортировки символов потока ввода-вывода между NVT и хостом.

Используется буферизация, в том числе чтобы излишне не загружать СПД.

В режиме по умолчанию «набранные» символы отсылаются незамедлительно.

В режиме linemode (RFC 1184) символы отсылаются после нажатия Enter.

10.2.2.1

Команды самого протокола Telnet «накладываются» на основной поток ввода-вывода путем вставки в него управляющих метасимволов.

Команды могут исходить как от Telnet-клиента, так и от Telnet-сервера.

Часть команд предназначена для выполнения и на Telnet-сервере, и на Telnet-клиенте, часть -- только для выполнения на Telnet-сервере.

Некоторые команды являются запросами и поэтому зависят от команд, являющихся подтверждениями.

Признаком Telnet-команды является метасимвол <IAC> (байт со значением 255). Далее следуют код команды (один байт) и аргументы. Если необходимо переслать равный <IAC> байт, то <IAC> повторяется два раза (байт-стаффинг).

10.2.2.2

Важной частью протокола Telnet является возможность согласования параметров NVT (например, текстового разрешения).

Эти параметры выражаются в нескольких десятках Telnet-опций (под код опции так же отведен один байт).

Изучать Telnet-опции в настоящее время смысла не имеет.

10.2.2.3a

| Код | Название | Описание |
|-----|-------------------|--|
| 240 | SE | Конец согласуемой Telnet-опции |
| 241 | NOP | Холостая Telnet-команда |
| 242 | Data Mark | Принудительно синхронизировать NVT с хостом с помощью экстренных TCP-данных (выполнить буферизированные Telnet-команды) |
| 243 | Break | Прервать текущий процесс (альтернатива <code>ctrl-c</code>) |
| 244 | Interrupt Process | Прервать текущий Telnet-процесс на Telnet-сервере (завершить сеанс) |
| 245 | Abort Output | Прервать поток вывода на NVT |
| 246 | Are You There | Послать уведомление о получении данной команды («Вы на связи?») |
| 247 | Erase Character | Удалить предыдущий символ (альтернатива <code>Backspace</code> и <code>Delete</code>) |
| 248 | Erase Line | Удалить предыдущую строку |
| 249 | Go Ahead | Ожидается следующая Telnet-команда |
| 250 | SB | Начало согласуемой Telnet-опции (далее должны следовать код опции и аргументы) |
| 251 | WILL | Два варианта. Предложение начать согласование Telnet-опции. Подтверждение начала согласования Telnet-опции. (Далее должен следовать код Telnet-опции) |
| 252 | WON'T | Два варианта. Предложение не начинать согласование Telnet-опции. Отказ начать согласование Telnet-опции. (Далее должен следовать код Telnet-опции) |
| 253 | DO | Два варианта. Начать согласование Telnet-опции. Ожидается согласование Telnet-опции. (Далее должен следовать код Telnet-опции) |
| 254 | DON'T | Два варианта. Не начинать согласование Telnet-опции. Не ожидается согласование Telnet-опции. (Далее должен следовать код Telnet-опции) |
| 255 | — | Экранированный символ 255 в потоке |

Telnet-команды

10.2.3.1

Наиболее серьезным из недостатков Telnet является полная незащищенность соединения от несанкционированного доступа.

Данные, в том числе и пароли, пересылаются в виде открытого текста (plain text). В современных условиях это не может устраивать любую организацию, даже некоммерческую.

Стандарты в области защиты информации фактически запрещают применение Telnet.

Поэтому на смену Telnet пришел SSH (Secure SHell) -- идея та же, но соединение полноценно защищено.

В свое время предпринимались попытки доработать Telnet, но это направление оказалось тупиковым.

10.2.4.1

C<-S: <IAC> <DO> <NAWS>

C->S: <IAC> <WILL> <NAWS>

C->S: <IAC> <SB> <NAWS> 0 80 0 24 <IAC> <SE>

Пример согласования текстового разрешения (Telnet-опция NAWS)

