
Communication Networks 2

Exercise 4 - Interactive Protocols



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Multimedia Communications Lab
TU Darmstadt

Problem 1 Interactive Protocols Setup FTP

Please explain the connection setup of the FTP protocol.

Solution:

- One Connection for control commands (Based on the telnet protocol)
- One Connection for (binary) Data Transfers
- In Active Mode: FTP Server opens the Data Connection to FTP Client
- In Passive Mode: FTP Server announces a Port Number
- Then the client opens the data connection to the port number

Problem 2 Interactive Protocols Distinction Control Data

Why is there a strict distinction between a Control Channel and a Data Channel?

Solution:

- FTP is based on Telnet with a fixed set of commands
- Telnet does not support the transfer of binary data
- Binary Data might be (wrongly) interpreted as Telnet Control Command
- Therefore: Transfer of binary data might interrupt control connection
- Solution: Separate channel for data transfer

Problem 3 Interactive Protocols Ftp Problems

Please name and explain common problems of the FTP Protocol.

Solution:

- FTP is a plain text protocol. FTP was not designed to be a secure protocol.
- Active Mode is (nearly) impossible behind today's NAT Routers
- Passive Mode causes difficulties in the server's firewall configuration
- Alternative: SFTP

Problem 4 Interactive Protocols Telnet SSH Compare

Compare Telnet and SSH. What are the differences between those protocols?

Solution:

- SSH and Telnet commonly serves the same purpose
- SSH is more secure compared to Telnet
- SSH encrypts the data while Telnet sends data in plain text
- SSH uses a public key for authentication while Telnet does not use any authentication
- SSH adds more overhead to the bandwidth compared to Telnet
- Telnet was replaced by SSH in almost all use cases

Source: differencebetween.net; Daniel J. Barrett, Richard E. Silverman, and Robert G. Byrnes, SSH: The Secure Shell (The Definitive Guide), O'Reilly 2005 (2nd edition)

Problem 5 Interactive Protocols Netcat

Read the documentation of netcat. Explain the parameters -l -k and -p

Solution:

- *l*: Use netcat to listen for incoming connections on the specified port
- *k*: Forces nc to stay listening for another connection after its current connection is completed. It is an error to use this option without the -l option.
- *p*: Specifies the source port of outgoing connections.

Problem 6 Interactive Protocols Netcat Client

In this task you will use netcat to build a simple tcp socket client. We provide you the appropriate socket server.

The socket server address is test09.kom.e-technik.tu-darmstadt.de port 1132

Use netcat to connect to the socket server. What does the server output?

Solution:

```

frank@frank-ThinkPad-T430s: ~
frank@frank-ThinkPad-T430s:~$ nc test09.kom.e-technik.tu-darmstadt.de 1132
17.11.2014:
  Bistro:
    Schinkennudeln mit Ei (2) und Tomatensauce S 1,70 €
    Von unserer Suppenbar! Linsen-Orangensuppe mit Baguette V 2,10 €
  Wokpfanne "Thai 7 Spice" mit Putenbrust und Basmatireis G 2,80 € :
    Wokpfanne "Satay" mit Hähnchenbrust und Basmatireis G 2,80 €
    Chicken-Nuggets-Neu 2014 (180 g)
  Marktrest.:
    Flammkuchen-Elsässer Art
  Bolognese N 2014 :
    Knusperschnitzel paniert dazu Jägersauce (2,3,4) mit frischen Pilzen S 2,30 €:
    Flammkuchen-Elsässer Art :
      Flammkuchen-Elsässer Art
  Gabel:
    Hähnchenbrust natur vom Grill G 2,40 €

```

Problem 7 Interactive Protocols Server

Now you have used netcat as a client for a given server. But how could our socket server look like? You will see the server side in the next task:

Start netcat in the listening mode. Chose an arbitrary port number. Then open another terminal session and connect a second netcat instance to the listening instance. Press some chars in both of the terminals. What happens? What happens when you press the enter key?

Solution:

- Terminal 1: `nc -l 11136`
- Terminal 2: `nc localhost 11136`

The chars entered in terminal 1 appears in terminal 2 and vice versa. But the input is buffered until a buffer is full or the user presses "Return"

Problem 8 Interactive Protocols Netcat Audio

You have seen the use of netcat to transfer textual data. Would it also be possible to transfer audio streams or even video streams using netcat? In which cases could this usage pattern cause problems? Assume that the bandwidth is high enough.

Solution:

Generally this would be possible. Netcat could also stream binary data. Together with an audio player (like `aplay`, `mpg123` for audio or `mplayer` or `vlc` for video) one can play the received stream. But the usage of `udp` might cause problems because of the missing buffer management.

Just try it on your own:

```
> nc test09.kom.e-technik.tu-darmstadt.de 1133 | mplayer -
```

Problem 9 Interactive Protocols Audio Server

In the last task, we have seen the client side of a simple audio streaming service. But how does the server look like? Please implement a server which could stream a MP3 file to a remote client using netcat.

Solution:

The solution is piping the output of the linux cat command to a netcat instance set up in listening mode.

Just try it on your own:

```
> cat testvideo.mp4 | nc -l 1133
```

Impressive, isn't it?