Université Nice Sophia Antipolis
International Master in Computer Science

Duration : 2h

Networking
Guillaume Urvoy-Keller & Dino Lopez

The utilization of any document or electronic device is forbidden. Your arguments must be clear and well written
Question with two stars (**) weigh 2 points while others weigh one point.

Exercise 1: Basic knowledge

1. Give the precise definition of a protocol.
2. What are the two key differences between the old dial-up modem access and modern ADSL access to the Internet ?
3. Contrast the circuit switching and the packet switching techniques of data transmission.
4. What are the four sources of delay in a packet switched network ?
5. What are the root DNS servers and why are they deduplicated in several locations around the world ?
6. What is the role of a DNS local resolver and of an authoritative server ?
7. What are the services that can/cannot be guaranteed in the Internet to the applications in terms of delay, loss rate and throughput ?
8. Assume you want to connect your company to the Internet with a Web and a mail server. Which piece of information must you provide to the DNS registrar you contact ?

Exercise 2: YouTube Buffering

Video and audio streaming heavily relies on buffering at the client side. Let us a consider a toy example to illustrate the complexity of picking the right value for the buffering delay (time before playing the video to the user). Assume that :
– YouTube transfers a video that must be read at R=300 kps.
– The access capacity of the client is R0=200 kbps
– The duration of the video is T=10 minutes
Let us call :
– t=0 the time instant when the YouTube server starts streaming the content.
– t=t_0 the time at which the video is being played to the user.
Questions :

1. (**) Which quantity of data is buffered during the buffering period [0,t_0] ?
2. (**) To have no image freeze, it is sufficient that the quantity of information received during the buffering period + the playout duration be smaller than the quantity of information consumed by the player during time T, i.e., t_0+T . Write the corresponding equation.
3. (**) What is the minimum buffering time ?
4. Why is it more complex in practice ?
5. Why does YouTube rely on TCP rather that sending at a fixed rate using UDP ?

Exercise 3: TCP/IP Layers

1. Define the multiplexing/de-multiplexing function of the transport layer.
2. What is the crucial difference between TCP and UDP ?
3. Which transport layer does HTTP use and why ?
4. Which transport layer does the DNS use and why ?
5. Does the network layer route between processes or machines ? Justify.

Exercise 4: First connection

Assume a machine with an IP address (a network mask) and a default gateway. The user types the following command in the terminal :

```
telnet 87.248.120.148 80
```

Figure 1 displays the set of frames generated by this command.

Time	Source	Destination	Src Port	Dst port	Info
19.556675000	Apple_16:ab:e5	Broadcast			Who has 192.168.0.254? Tell 192.168.0.3
19.562012000	FreeboxS_42:4d:8e	Apple_16:ab:e5			192.168.0.254 is at 00:07:cb:42:4d:8e
19.562081000	192.168.0.3	87.248.120.148	56607	80	56607 > http [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=8 TSval=41725281
19.652939000	87.248.120.148	192.168.0.3	80	56607	http > 56607 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM=
19.653066000	192.168.0.3	87.248.120.148	56607	80	56607 > http [ACK] Seq=1 Ack=1 Win=524280 Len=0 TSval=417252819 TSecr=

FIGURE 1 – Telnet Connection

1. Let us consider the first two frames.
   (a) What is their purpose ?
   (b) Which type of address do we have in the source/destination columns ?
   (c) Why does the first frame have a broadcast destination address ?
2. Prove that the first 3 frames belong to the same transfer between 2 application process.
3. Figure 2 depicts the details of the protocol layers from the third frame. Prove that the Ethernet and IP destination addresses do not correspond to the same machine.

```
▷ Frame 10: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface 0
▷ Ethernet II, Src: f0:b4:79:16:ab:e5 (f0:b4:79:16:ab:e5), Dst: 00:07:cb:42:4d:8e (00:07:cb:42:4d:8e)
▷ Internet Protocol Version 4, Src: 192.168.0.3 (192.168.0.3), Dst: 87.248.120.148 (87.248.120.148)
▷ Transmission Control Protocol, Src Port: 56607 (56607), Dst Port: http (80), Seq: 0, Len: 0
```

FIGURE 2 – Telnet connection - frame 3

Exercise 5: TCP

Consider a TCP transfer between a client and a server with one data packet of 10 bytes transfered from the client to the server and 10 packets of 1000 bytes sent in response to the first data packet by the server.

1. (**) Make a drawing with the establishment and transfer phase and the sequence and ack numbers (assume initial sequence number of 0)

2. What is the difference between congestion control and flow control in TCP ?
3. (**) Make a drawing of the evolution of the TCP sequence number of the server assuming an initial congestion window of 1 MSS (1500 bytes) and assuming that the advertised window from the client is equal to 3 MSS.

Exercise 6: Socket

1. What are the three types of sockets in the IF_INET domain and to which transport protocol are they associated ?
2. Define the concept of endianness and why it is crucial to account for it during data transfers between machines ?
3. Define the different styles of access (iterating, forking and concurrent single server) and compare them in terms of performance.
4. Which privilege is needed to bind to well-know ports, i.e., ports below 1024 ?

Exercise 7: DHCP

Questions :

1. What is the purpose of DHCP ?
2. Draw a diagram of the different phases of a DHCP exchange (discovery, offer, request, ack) with details of the addresses at the MAC and IP layer used each time.

Exercise 8: BSD Sockets

Assume we have two machines connected by local network. Machine C has IP address 192.168.1.1 and S has IP address 192.168.1.250. We run a server program (python code below) on machine S and a client program on machine C (python code below).
Server code :

```
#!/usr/bin/python

import socket

s = socket.socket(IF_INET, SOCK_STREAM)
host = socket.gethostname()
port = 12345
s.bind((host, port))

s.listen(5)
while True:
   c, addr = s.accept()
   print 'Got connection from', addr
   c.send('Thank you for connecting')
   c.close()
```

Client code :

```
#!/usr/bin/python

import socket

s = socket.socket(IF_INET, SOCK_STREAM)
host = socket.gethostname()
port = 12345

s.connect((host, port))
print s.recv(8)
s.close
```

1. Which transport protocol is going to be used for this client to server communication ?
2. Let us focus on the client code. Why does it fail when one starts one client ? How do you fix the problem ?
3. Assuming you fixed the (above) problem and the client can now reach the server. What is going to be output on the screen on the client side ?
4. What happens if 10 clients are connecting simultaneously to the server ?