Computer Networking
Exercises

Antonio Carzaniga
Faculty of Informatics
USI
(Università della Svizzera italiana)

Edition 2.2
October 2018

▸**Exercise 1.** Consider a DNS query of type *A* within a DNS system containing *IN* class information. Using boxes to represent servers and lines with labels to represent packets, diagram an *iterative request* for "www.ban.mcyds.net". The answer must be authoritative. Label any DNS servers you need to contact using a descriptive label. Label every packet with the essential information. For example, a box may be labeled "authority for .com," while a packet might be labeled "answer, www.ban.mcyds.net, 192.168.23.45" *(10')*

Consider the same DNS request specified above. Now create a new diagram showing a *recursive request*. Once again, the answer must be authoritative. *(10')*

▸**Exercise 2.** Given the utility functions listed below, write the pseudo-code to perform an *iterative* DNS lookup.

---

dns_query_pkt `make_dns_packet`(*type*, *class*, *flags*)
Creates a new DNS query packet. Flags can be combined via the '`|`' operator. So for a query that is both authoritative and recursive, one would write: `(DNS_AUTH | DNS_RECURSE)`. Only the `DNS_AUTH` and `DNS_RECURSE` flags are valid. Type can be *A*, *MX*, *NS*, or any other valid DNS type.

value `get_dns_answer`(*dns_answer_packet*, *n*)
Return the value in the n^{th} answer of a *dns_answer_pkt* packet. For example, in reply to a *MX* lookup for inf.unisi.ch, `get_dns_answer(pkt,1)` would return the SMTP mail server for the inf.unisi.ch domain. In reply to a *NS* query it would return the authoritative name server.

dns_answer_packet `send_and_wait`(*dns_query_packet*, *server*)
Send the given *dns_query_packet* and wait for a replay from the given DNS server. Returns a *dns_answer_packet*.

By "pseudo-code" here we mean a simplification of an actual program that shows only the essential operations. The syntax we expect is that of Java, or C if you prefer. Do not worry about the details of the program. For example, if you need to output something, just write something like "print(...)." Insert comments in your code to describe your ideas. What counts here is that the procedure you implement be clear at a high-level. (Hint: the operation corresponds to the execution of *dig* with the *+trace* option.) *(20')*

// Implement your code here.
void `ns_trace`(*server_name*) {

Given the same functions listed on the previous page, write the pseudo code to perform a *recursive* DNS lookup. *(10')*

// Implement your code here.
void `ns_recurse_lookup`(*server_name*) {

▸**Exercise 3.** Design an extension to HTTP to support *ratings* for documents. Every document may be rated by users on a scale of 1-100. An HTTP server computes and stores the average ratings of each document. An HTTP server should also return the rating of a document when that is requested by a client.

Which HTTP *method* would you use to send a new rating to a server? Give an example of an HTTP request containing a rating. *(5')*

How would you modify a *GET* request in order to include a rating query from a client? Show a corresponding HTTP request as an example. What if the client is interested only in the ratings? Show another HTTP request that returns the rating without returning the content. *(5')*

▸**Exercise 4.** The following is an SMTP conversation containing only the responses from the server. Fill in the blanks with the messages sent by the client. *(15')*

Server: 220 mx.duder.org ESMTP Sendmail 8.12.10; Mon, 9 May 2005 12:26:27 -0600 (PST)

Client:

Server: 250 mx.duder.org. Hello mail.bowlingalley.net [171.33.22.6], pleased to meet you

Client:

Server: 250 2.1.0 donnie@bowlingalley.net... Sender ok

Client:

Server: 550 5.1.1 dude@mx.duder.org... User unknown

Client:

Server: 250 2.1.5 thedude@mx.duder.org... Recipient ok

Client:

Server: 354 Enter mail, end with "." on a line by itself

Client:

Server: 250 2.0.0 j49IQRp8013851 Message accepted for delivery

Client:

Server: 221 2.0.0 mx5.Colorado.EDU closing connection

►**Exercise 5.** Consider the following SMTP message. Separate the headers added by transport agents from those added by the user agent or delivery agents. Do that by drawing a line between the two sets of headers. Write a label above and below the line marking which set of headers are from the transport-agent and which set are from other agents. You may need to draw more than one line. *(5')*

```
Delivered-To:  hallc@lu.unisi.ch
Received:  from spamfilter.usilu.net by campus9.usilu.net
        with Microsoft SMTPSVC; Fri, 11 Mar 2005 23:07:56 +0100
Received:  from localhost (spamfilter.usilu.net]) by
        spamfilter.usilu.net (Postfix) with ESMTP id C2AC07C17D for
        <hallc@lu.unisi.ch>; Fri, 11 Mar 2005 23:07:55 +0100 (CET)
Received:  from spamfilter.usilu.net by localhost
        (spamfilter.usilu.net) (amavisd-new, port 10024) with ESMTP id
        02298-08 for <hallc@lu.unisi.ch>;
        Fri, 11 Mar 2005 23:07:53 +0100 (CET)
Received:  from mail-fs.sunrise.ch (mta-fs-be-01.sunrise.ch)
        by spamfilter.usilu.net (Postfix) with ESMTP id
        68A397C017 for <hallc@lu.unisi.ch>;
        Fri, 11 Mar 2005 23:07:52 +0100 (CET)
Received:  from [187.22.132.87] by mail-fs.sunrise.ch
        id 422D8BF5001729DB for hallc@lu.unisi.ch; Fri, 11 Mar 2005
        23:07:52 +0100
Mime-Version:  1.0 (Apple Message framework v619.2)
Content-Transfer-Encoding:  7bit
Message-Id:  <7d7e7812ec273c6e2e26842e834324ef@lu.unisi.ch>
Content-Type:  text/plain; charset=US-ASCII; format=flowed
To:  Cyrus Hall <hallc@lu.unisi.ch>
From:  The Dude <thedude@duder.org>
Subject:  my rug is gone
Date:  Fri, 11 Mar 2005 23:07:41 +0100
X-Mailer:  Apple Mail (2.619.2)
X-Virus-Scanned:  by amavisd-new at usilu.net
Return-Path:  thedude@duder.org
X-OriginalArrivalTime:  11 Mar 2005 22:07:56.0089 (UTC)
```

Could the message of the previous page be valid? Justify your answer. *(5')*

►**Exercise 6.** Below is an HTTP response from the server www.unisi.ch.

```
HTTP/1.0 302 Found
Location:  http://www.unisi.ch/
Content-Type:  text/html
Server:  Apache/1.3.29 (Win32)
Last-Modified:  Thu, 03 May 2001 16:00:38 GMT
Content-Length:  1494
Date:  Mon, 09 May 2005 20:37:25 GMT
Connection:  close
Expires:  Mon, 09 May 2005 20:45:02 GMT
```

Give a HTTP request that could generate such a response. *(5')*

Below is an HTTP request.

```
PUT /files/uploads/private/tbl.avi HTTP/1.1
Host: www.personalpage.net
Content-Type: video/x-msvideo
Content-Length: 1822990

...a bunch of binary data goes here...
```

Enumerate at least four responses that an HTTP server could give. Only show the response line of each response. *(10')*

►**Exercise 7.** An HTTP document is made of the following three objects: x.html (1Kb), x.jpg (50Kb), and x.png (100Kb). Assuming that the underlying transport layer has a constant throughput T = 100KB/s and that the latency between the client and the origin server is L = 500ms, compute the total time $\Delta_{1.1}$ that it would take an HTTP/1.1 browser using pipelining to retrieve the entire document. Assume the server is able and willing to serve (HTTP/1.1) pipelined requests. Also compute the total download time $\Delta_{1.0}$ in the case of an HTTP/1.0 client that does not use pipelining. For the purpose of your calculations, you may assume that the size of HTTP headers (both requests and responses) are negligible. *(10')*

Now assume the client is connected to the origin server through a caching proxy. Under the same assumption of a constant throughput at the transport level, given that the latency with the proxy is L_p = 100ms, that the throughput with the proxy is T_p = 1000KB/s, that the latency between the proxy and the origin server is $L_{p\to o}$ = 500ms, and that the throughput there is $T_{p\to o}$ = 100KB/s, compute the total download time for an HTTP/1.1 client using pipelining, assuming that the two images (x.jpg and x.png) are cached, while x.html is not. *(10')*

►**Exercise 8.** Consider a reliable transport layer implemented through a Go-Back-N protocol, with maximum segment size MSS = 1KB, and with a fixed window size W = 10. Suppose the sender transmits at the maximum speed allowed by the protocol and that the network has plenty of bandwidth and no congestion, suppose also that the underlying network looses, on average, one segment out of 1000, and suppose that each segment, whether a data segment or an acknowledgment, has a 2-byte header, with acknowledgment segments having no content.

On average, what is the total number of bytes sent into the network by both the sender and the receiver to transfer a 10MB file? Briefly justify your answer. (Assume 1MB = 1000KB and 1KB = 1000B.) *(15')*

►**Exercise 9.** Briefly describe the algorithm used by TCP to control the size of its congestion window. Complete your description with a diagram showing how the window size might vary over time, in response to every protocol event. *(10')*

►**Exercise 10.** Briefly explain the functionality of the *SYN* flag in the TCP header. *(5')*

►**Exercise 11.** A TCP segment with sequence number 1234 carries the following HTTP request:

```
HTTP/1.0 404 Not Found
Content-Type: text/html
Content-Length: 200

<html><body>Not Found!
...more html text here...
</body></html>
```

What is the sequence number of the next segment? Briefly justify your answer. (Remember that all HTTP header lines end with a CRLF sequence.) *(10')*

►**Exercise 12.** Outline the UDP header format. What kind of transport-level features does UDP provide? Say whether each feature relates to any header field, and if so, describe how. *(10')*

►**Exercise 13.** Describe the high-level architecture of a router. Explain where a router queues packets. For each packet queue, explain why that queue is necessary and what circumstances may cause that queue to fill up. *(10')*

►**Exercise 14.** Consider the forwarding function of a router within (1) an IPv4 network, (2) a generic packet-switched network, and (3) a generic virtual-circuit network. For each case, write the "signature" of the forwarding function in terms of its domain (input set) and range (output set) for a router (e.g., the signature of the logarithm function is $\textsc{Log} : R \to R^+$, where R^+ represents positive real numbers and R represents all real numbers). Comparing (2) and (3), tell which function would be simpler to compute. Justify your answer. *(5')*

►**Exercise 15.** A router receives a non-fragmented 1400-bytes IPv4 packet from interface 1, and decides to forward it to interface 2, which has an MTU of 512 bytes. Explain in detail how the router compiles the output packet(s). Write your explanation in the form of pseudo-code. Refer to the input packet as X and to the output packet(s) as Y_1, Y_2, etc. Refer to header fields using a Java-like dot notation. E.g., refer to the source address of the input packet as $X.source$ and to the first 100 bytes of payload as $Y.data[0 \ldots 99]$. Assume the input packet has no options. (Hint: the first step could be something like this: "$Y_1.source \leftarrow X.source$") *(15')*

►**Exercise 16.** Express the following address ranges using the subnet prefix notation. If a range can not be represented with the prefix notation, write "N.A."

range	*subnet prefix-address/prefix-length*
67.56.34.64–67.56.34.79	
121.232.111.128–121.232.111.255	
121.34.56.64–121.34.56.128	
128.131.9.0–128.131.9.192	
108.47.200.192–108.47.200.223	
93.20.10.0–93.20.11.0	
128.242.138.0–128.242.139.127	
200.220.76.0–200.220.79.255	
200.220.0.0–200.223.255.255	

For each valid prefix you wrote above, write the corresponding address/mask expression *(15')*

►**Exercise 17.** Given the following set of prefixes, write an equivalent minimal set of prefixes by simplifying the list (i.e., by applying "supernetting"). *(10')*

128.138.242.0/24	
128.138.243.0/24	
128.138.0.0/16	
108.47.128.0/22	
108.47.136.0/21	
108.47.132.0/22	
128.138.128.0/22	
128.138.136.0/21	
138.138.132.0/22	

►**Exercise 18.** Outline the structure of an entire IPv6 packet containing a UDP packet containing a simple HTTP 1.0 request. Referring to each field using a symbolic identifier, fill in as many header fields and payload data as possible. Use sensible data. For each field, explain very briefly your choice of value. For example, you might write "*IP.source* = 123456789: value chosen at random." Make sure to mention any relation between fields. *(10')*

►**Exercise 19.** Compare IPv4 with IPv6. List the major differences between the two protocols. Briefly explain the design rationale for each difference. *(10')*

►**Exercise 20.** Consider a router in an IPv4 network using longest-prefix matching. The router has the following forwarding table:

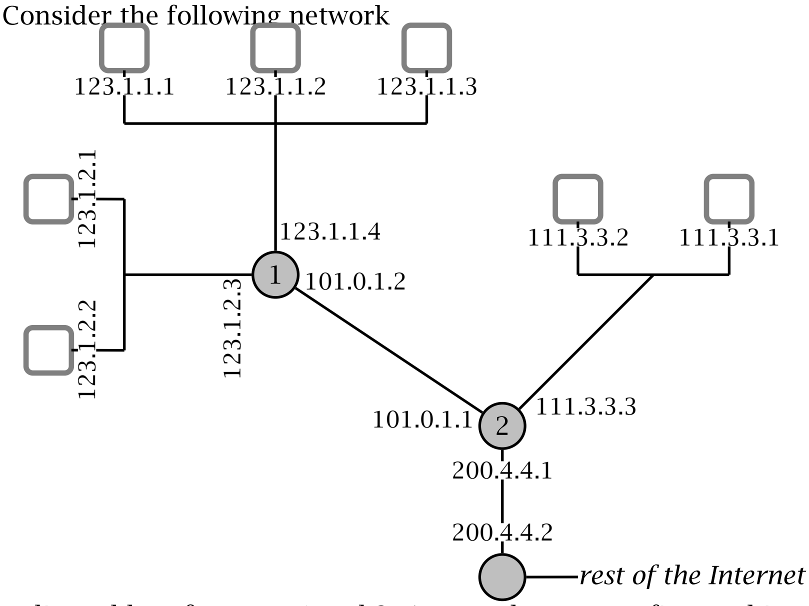
entry	*destination*	*interface*
1	98.7.1.0/16	1
2	211.57.20.0/24	1
3	40.120.0.0/16	2
4	160.0.0.0/2	3
5	111.11.0.0/16	3
6	211.57.0.0/16	4
7	0.0.0.0/2	4
8	0.0.0.0/0	5

For each destination addresses below, write the output port and the list all the matching table entries.

address	*output port*	*matching entries*
211.57.1.69		
10.142.226.44		
98.7.2.71		
200.100.2.1		
40.120.207.167		
211.57.20.11		
211.57.21.10		

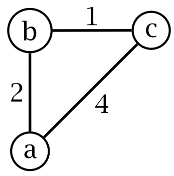
(10')

►**Exercise 21.** Consider the following network



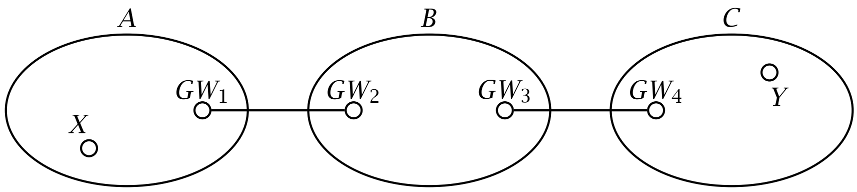
Write the forwarding tables of routers 1 and 2. Assume longest-prefix matching. Assume also that each subnet holds a block of 256 addresses. In writing the forwarding tables, identify each output port directly with their IP address. *(15')*

►**Exercise 22.** Consider the following simple network topology where routers use a distance-vector routing protocols



For simplicity, assume all routers start at time 0, that routing messages (i.e., distance vectors) are sent out by routers every 10 seconds, and that they are received by neighbor routers after one second. Write the first iterations of the distance-vector routing algorithm, at times 0, 10, ..., until the protocol converges to a stable state. For each iteration, list the routing tables of each router. *(15')*

►**Exercise 23.** Consider three autonomous systems A, B, and C, connected in the following AS-level topology



Illustrate the idea of hierarchical routing in the particular case of the Internet by describing the process by which an IP packet would be routed from a source router X in A to a destination router Y in C. For each step, briefly discuss which routing information is used, which protocol determines that routing information, and how that routing information is transmitted to the router. *(10')*

►**Exercise 24.** Briefly describe what kind of routing information is exchanged by two BGP peers. In particular, explain what information allows BGP to detect and avoid routing loops. *(5')*

►**Exercise 25.** Alice and Bob plan to communicate privately using a one-time pad encryption scheme, and therefore agree on a secret key k = 10100100111100111010011010001101. The first message that Alice wants to send is "CIAO". Alice and Bob use an insecure channel (e.g., a simple TCP connection). Therefore, Eve can intercepts every transmission. Assuming that Alice uses the ASCII encoding, and that the codes for the letters C, I, A, and O are 67, 73, 65, and 79, respectively, what does Eve see when Alice sends her first message? How many more bytes can Alice and Bob exchange securely? Justify your answers. *(10')*

►**Exercise 26.** You are sending an e-mail message to *dude@duder.org*. The mail exchange server for the *duder.org* domain is *mail.duder.org*. Detail the SMTP conversation between your mailer and *mail.duder.org*. Write every SMTP message exchanged by your mailer and the server. Assume that *dude@duder.org* is a valid address for *mail.duder.org* and that no communication or server errors occur. *(20')*

►**Exercise 27.** Answer the following questions.
Briefly explain the functionality of the *ACK* flag in the TCP header. *(5')*
Do IPv4 headers and IPv6 headers have any fields in common? If any, describe the function of the common fields? *(5')*
Do TCP headers and UDP headers have any fields in common? If any, describe the function of the common fields? *(5')*

►**Exercise 28.** Answer the following questions.
Consider a TCP connection between host X and host Y. Suppose host X sends two TCP segments, one after the other, to host Y, with sequence numbers 200 and 500, respectively. What happens if the first segment is lost but the second segment arrives at Y? Does Y send an acknowledgment? If so, what is the sequence number of the acknowledgment? How many data bytes are in the first datagram? Briefly justify your answers. *(5')*
Briefly describe the idea of *classless interdomain routing (CIDR)*. Give an example of a "classless" subnet address. Why is this addressing scheme important in Internet routing? *(5')*
Suppose an IP datagram goes from its source X to its destination Y through four routers. How many interfaces does the datagram go through? How many times does the datagram cause a router to look up its forwarding tables? Justify your answers. Assume no fragmentation. *(5')*

►**Exercise 29.** List three HTTP methods. For each method, briefly explain its main purpose and give a valid server response. *(10')*

►**Exercise 30.** Consider a non-recursive DNS lookup for the A record of *www.elet.polimi.it* executed from within the *unisi.ch* domain. Give a list of plausible DNS servers contacted by your local DNS resolver. You do not need to list the local DNS server. For each server, write a plausible query and response (in English—don't bother using the exact DNS format). *(10')*

►**Exercise 31.** A TCP segment with sequence number 2345 carries the following HTTP request:

```
HTTP/1.0 404 Not Found
Content-Type:  text/html
Content-Length:  200

<html><body>Not Found!
...more html text here...
</body></html>
```

What is the sequence number of the next segment? Briefly justify your answer. (Remember that all HTTP header lines end with a CRLF sequence.) *(10')*

►**Exercise 32.** A Web browser makes an HTTP/1.0 request over TCP to a web server to retrieve a 2Kb object (e.g., a web page). The network between the browser and the server has an MTU of 1500 bytes, a latency of 100ms, and a throughput of 100KB/s. List all IP datagrams for the connection. For each datagram, describe the most important header fields of each protocol (IP, TCP, and HTTP). Assume no congestion, no network errors, and no fragmentation. Also, how long does it take for the browser to retrieve the entire object? Justify your answer. *(20')*

►**Exercise 33.** Consider a router in an IPv4 network using longest-prefix matching. The router has the following forwarding table:

entry	*destination*	*interface*
1	98.7.1.0/16	1
2	139.57.20.0/24	1
3	40.120.0.0/16	2
4	160.0.0.0/2	3
5	111.11.0.0/16	3
6	139.57.0.0/16	4
7	0.0.0.0/2	4
8	0.0.0.0/0	5

For each destination addresses below, write the output port and the list all the matching table entries.

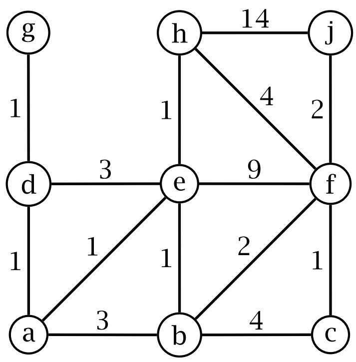
address	*output port*	*matching entries*
139.57.1.69		
10.142.226.44		
98.7.2.71		
162.100.2.1		
40.120.207.167		
139.57.20.11		
139.57.21.10		

(10')

►**Exercise 34.** Consider the following networks: (1) a packet-switched network with 16-bit addresses and a total of 1000 nodes, and (2) a virtual-circuit network with 8-bit virtual-circuit identifiers and 30 active virtual circuits. For each network, write and describe the "signature" of the forwarding function. That is, describe its domain (input set) and range (output set). Assuming that the forwarding function is implemented through a simple table containing one entry per input value. For each network, tell how many entries are in the forwarding table. Briefly justify your answer. *(10')*

►**Exercise 35.** Briefly describe a *block cipher*. In particular, describe the parameters that define the signature of a block cipher. *(5')*

►**Exercise 36.** Given the following network topology, specify the forwarding function of node *a* in the form of a simple forwarding table. Specify the output interfaces using the identifier of the next-hop router. Does the same forwarding table for node *f* have more, less, or the same number of entries? Briefly justify your answer.



(15')

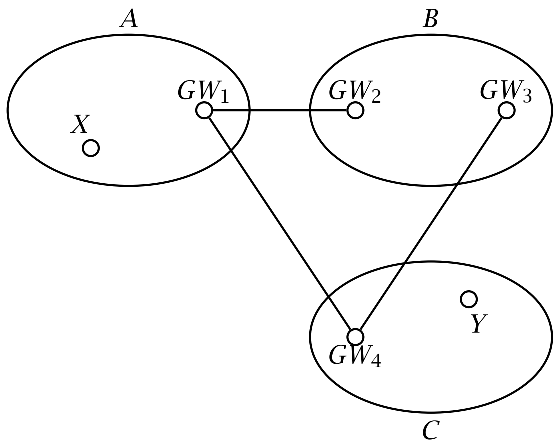
►**Exercise 37.** Compare and contrast *distance-vector* and *link-state* routing. In particular, describe the information that routers exchange, and the kind of processing they perform locally. *(10')*

►**Exercise 38.** For each one of the following subnet addresses, give an example of an IP address that can be assigned to that subnet, and one that can not.

subnet	*IP address in subnet*	*IP address outside subnet*
67.56.0.0/16		
121.232.111.128/25		
121.34.56.64/30		
128.131.9.0/16		
128.131.10.0/16		
128.0.0.0/8		
93.20.10.0/28		
128.242.138.0/18		
200.220.76.0/24		
200.220.0.0/12		

(10')

►**Exercise 39.** Consider three autonomous systems *A*, *B*, and *C*, connected in the following AS-level topology



Illustrate the idea of hierarchical routing in the particular case of the Internet by describing the process by which an IP packet would be routed from a source router X in A to a destination router Y in C. For each step, briefly discuss which routing information is used, which protocol determines that routing information, and how that routing information is transmitted to the router. *(15')*

►**Exercise 40.** The web object located at *http://exacttime.ch/now* gives the time of day.

Question 1: Write an HTTP 1.0 request to read the current time, and a plausible response from the server. Write the request and the response as (ASCII) text. If you need to express non-text bytes, then write a backslash '\' and then either their numeric value or a short textual description of the character. *(5')*

Question 2: Write an HTTP 1.1 request for the same object *http://exacttime.ch/now* where the client attempts to keep the connection open. Again write both the client request and the server response. What kind of cache-control should the server specify in its reply? Justify your answer. *(5')*

►**Exercise 41.** Briefly describe at least four types of *resource records* in the Domain Name System. *(10')*

►**Exercise 42.** Your computer uses a DNS resolver connected on your local-area subnet where the latency is L_{in} = 5ms. Your subnet is connected to the rest of the Internet through a single link that introduces a latency of 200ms. Suppose you click on a link to *http://www.gnu.org/*. What is the minimum amount of time before your browser can start to display the web page? Ignore the rendering time and other CPU latencies. Assume that the address of *www.gnu.org* is not in the cache of your host or your local DNS. Justify your answer. *(10')*

►**Exercise 43.** An e-commerce web site needs to keep track of per-user shopping sessions. How can that be done using HTTP 1.0? Describe this mechanism through an example in which two different users are active at the same time. Write every user request and every server response in your example. *(10')*

►**Exercise 44.** Can user sessions be implemented in HTTP 1.1 with persistent connections? If so, show how through an example, writing at least two requests/replies for the same session. If not, justify your answer through a counter-example in which a persistent connection could represent more than one user. *(10')*

►**Exercise 45.** Design the high-level interface of a web-mail system. That is, a mail system where the client communicates with the server using HTTP. In particular, specify how you would you implement a *send* function, to send an e-mail message, a *read-folders* function, to get a list of folders, a *read-folder-summary* function to get the list of messages in a folder, a *read* function, to read a given message within a given folder. Also, how can you implement a *notification* function that tells you you have received a message. *(20')*

►**Exercise 46.** You are setting up the Internet resources of an organization called *Addio Lugano Bella*.

Question 1: Can you have your web site at *http://addio.luganobella.org/* and, at the same time, have an e-mail address such as *michael@addio.luganobella.org*? Justify your answer. *(5')*

Question 2: Can you do that with two different hosts, one running a web server and one running a mail server? If so, explain how. If not, give a counter-example. *(5')*

►**Exercise 47.** Someone composes the following e-mail message

From:	The Dude <duder@bowlingalley.net>
Subject:	the story of my rug
To:	Jeffrey Lebowsky <jeffrey@lebowsky.com>
Cc:	Walter <walter@bowlingalley.net>
Bcc:	Maude <maude@lebowsky.com>

Just so you know, I think I saw Bunny last night...
Dude.

Write the SMTP commands that the user agent issues to send this message. *(10')*

►**Exercise 48.** The network connection between Host A and Host B has a latency of $L = 200\text{ms}$, a throughput $T = 500\text{KB/s}$, and a maximum segment size $MSS = 1460b$. How long does it take for TCP to transfer a 10Kb file from A to B? Assume no errors, no packet loss, and no packet duplication. Justify your answer. *(10')*

►**Exercise 49.** The network connection between Host A and Host B has a latency of $L = 500\text{ms}$, a throughput $T = 100\text{KB/s}$, and a maximum segment size $MSS = 1460b$. How long does it take for TCP to reach the maximum throughput for that connection? Assume no errors, no packet loss, and no packet duplication. Justify your answer. *(10')*

►**Exercise 50.** The network connection between Host A and Host B admits TCP segments that can carry up to 512 bytes of data. Host A connects to Host B, transfers 2000 bytes, and closes the connection.

Question 1: List all the TCP segments exchanged between A and B in the presence of a perfectly reliable network (no errors, no losses, no duplicates). For each segment, write source and destination port (choose some values), sequence number, ack number, and all the active flags. *(10')*

Question 2: List all the TCP segments exchanged between A and B in case the network looses two segments. Choose which segments are dropped among the ones that carry data. Again, for each segment, write source and destination port (choose some values), sequence number, ack number, and all the active flags. Also, clearly mark the two dropped segments. *(10')*

►**Exercise 51.** Briefly describe the state maintained by a sender *Go-Back-N*. How does the sender update its local state in case (1) a timeout occurs, and (2) a packet is received correctly. *(10')*

►**Exercise 52.** A router has output interfaces with a buffer (queue) that can hold up to 64 packets and with links each capable of transmitting up to 10000 packets per second. The router has 4 input interfaces, each one receiving an average of 6000 packets per second. Suppose that at some point, and for a long period of time, all traffic happens to be forwarded to the same output interface.

Question 1: Assuming that both input ports and switch fabric are capable of handling the input flow, is the router congested during this period? If so, what is the probability that a packet be dropped? Assume the router uses a "drop-tail" policy. Justify your answer. *(5')*

Question 2: Assume that each input port is capable of receiving and processing 10000 packets per second, and that the switching fabric can process 20000 packets per second. Assume also that the router processes packets in a first-come first-served manner. What is he expected latency during this period? Justify your answer. *(5')*

►**Exercise 53.** A router has four output interfaces each with a buffer that can hold up to 64 packets and with a link capable of transmitting up to 6000 packets per second. Suppose that at time $t = 0$, when all buffers are empty, the router starts receiving a steady flow of 20000 packets per second from its input interfaces. Assume that both input ports and switch fabric are capable of handling the input flow. Of all the input packets, 40% go to output interface 1, 30% to interface 2, 20% to interface 3, and 10% to interface 4. Can the router sustain this kind of traffic without dropping packets? If so, what is the expected length of the packet queue on each output interface? If not, what is the expected time t' when the router will start dropping packets? Justify your answer. *(10')*

►**Exercise 54.** Describe the IPv6 packet format. Briefly explain the function of each header field. (Hint: if you don't remember all the fields, at least try to focus on the most important ones.) *(5')*

►**Exercise 55.** A router receives an IPv4 datagram with the following header fields: *datagram-length*=1500, *header-length*=20, *identifier*=1234, *fragmentation-offset*=300, *more-fragments*=1. The router decides that it must forward this datagram through an interface with an MTU of 512 bytes. To do that, the router must fragment the datagram. Explain how the router does that. In particular, for each fragment, write all the relevant header fields. *(10')*

►**Exercise 56.** For each one of the following network (prefix) addresses, write the corresponding range of addresses. *(10')*

subnet prefix-address/prefix-length	*range*
34.254.21.128/25	
128.129.242.160/30	
192.168.0.0/16	
231.111.10.160/27	
68.103.128.0/24	
68.103.128.0/22	
127.0.0.1/8	
230.1.0.192/27	
224.0.0.0/4	
0.0.0.0/0	

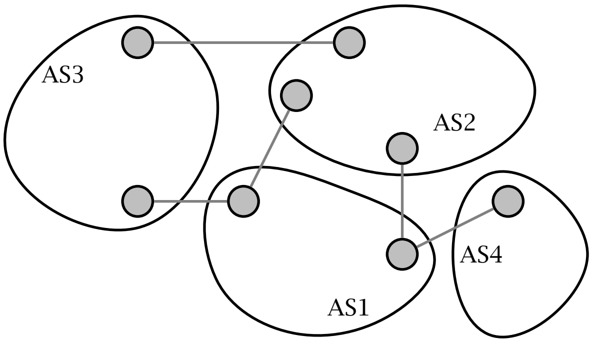
►**Exercise 57.** For each one of the following network (prefix) addresses, write the corresponding address and mask (i.e., address/mask) *(5')*

subnet prefix-address/prefix-length	*range*
34.254.21.128/25	
128.129.242.160/30	
192.168.0.0/16	
231.111.10.160/27	
68.103.128.0/24	
68.103.128.0/22	
127.0.0.1/8	
230.1.0.192/27	
224.0.0.0/4	
0.0.0.0/0	

►**Exercise 58.** Consider the following AS-level topology and the given allocation of addresses

AS3	128.138.0.0/16
	200.23.128.0/18

AS1	200.23.192.0/18
	39.81.36.0/22
	39.81.40.0/22
	39.81.44.0/22



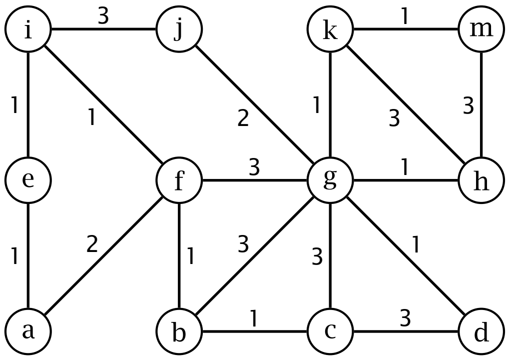
AS2	68.10.20.0/24
	87.13.192.0/27
	128.30.0.0/16

AS4	87.13.128.0/27
	100.91.45.0/24
	39.81.32.0/22

Write all the BGP route advertisements produced by each one of the autonomous systems. For each advertisement write only destination and AS path. Assume that autonomous systems are always willing to accept and forward route advertisements. (Hint: remember that addresses should be combined in router advertisements.) *(20')*

►**Exercise 59.** Give an example in which two ISPs advertise address prefixes that define overlapping ranges of addresses (in fact subsets). Briefly explain how longest-prefix matching is used to disambiguate the forwarding function. Draw the network topology, the address allocation, and the route advertisements. Briefly explain why doing this is useful. *(10')*

►**Exercise 60.** Consider the following network where routers use a distance-vector routing protocols



Question 1: Write all the routing information maintained by router f at a time when the routing protocol has converged to a stable state. (Hint: this includes router f's distance vector as well as the distance vectors received from its neighbors.) *(15')*

Question 2: At a later time, the cost (e.g., latency) of the (f, g) link raises to 10. Write the distance vector that router f sends out after this link-cost change. *(5')*

►**Exercise 61.** Describe a *block cipher* and a *stream cipher*.

Question 1: Write the mathematical definition of a block cipher (i.e., its domain and range), and briefly describe its desired properties. *(5')*

Question 2: Write the mathematical definition of a stream cipher, and briefly describe its desired properties. *(5')*

►**Exercise 62.** Alice wants to send Bob a private message m of $N = |m|$ bits. Alice and Bob share a secret 128-bit key K. Briefly explain how Alice would encrypt the message and how Bob would recover the original message from the ciphertext. Specify which cryptographic primitive(s) they would use, and how. *(10')*

►**Exercise 63.** Below are four TCP packets captured on the network at more or less the same time.

(1)

src-address: 128.138.242.241	
dest-address: 66.78.102.32	
src-port: 3241	*dest-port:* 5432
seq-num: 2000	
ack-num: 0	
...	
...	

(2)

src-address: 66.78.132.200	
dest-address: 128.138.242.241	
src-port: 5432	*dest-port:* 3241
seq-num: 2000	
ack-num: 0	
...	
...	

(3)

src-address: 128.138.242.241	
dest-address: 66.78.102.32	
src-port: 5432	*dest-port:* 3241
seq-num: 2001	
ack-num: 0	
...	
...	

(4)

src-address: 128.138.242.241	
dest-address: 66.78.132.200	
src-port: 3241	*dest-port:* 5432
seq-num: 1	
ack-num: 2300	
...	
...	

Which ones belong to the same TCP connection? Briefly justify your answer. Also, write another plausible packet belonging to the same connection. *(10')*

►**Exercise 64.** What is the theoretical maximum number of TCP connections allowable between two given hosts at the same time? Briefly justify your answer. *(5')*

►**Exercise 65.** List and briefly describe the primary attributes of a BGP advertisement. *(5')*

►**Exercise 66.** An IPv4 header contains a 16-bit packet identifier. What is the purpose of this identifier? Is there an equivalent header field in IPv6? If so, which one is it? If not, explain why not. *(5')*

►**Exercise 67.** The HTTP 1.1 protocol requires that clients specify at least one header in their requests. Which header is it? Explain its purpose and the reason why HTTP 1.1 requires it. *(5')*

►**Exercise 68.** Answer "yes" or "no" to the following questions.

Question 1: Is it possible for an IPv6 datagram to contain a TCP segment?

Question 2: Does the UDP header contain a flag called "more-fragments"?

Question 3: Does UDP provide any ordered-delivery guarantee?

Question 4: Is there any flags in the TCP header that signals network congestion?

Question 5: Does TCP support bi-directional communication? *(10')*

►**Exercise 69.** TCP is used in a network with $MTU = 512$B to transfer a 1000-bytes file. Briefly list all the segments necessary to open the connection, send the 1000-bytes file, and close the connection. For each segment, write source and destination port (choose some values), sequence number, ack number, and all the active flags. You may assume that the network is perfectly reliable. *(20')*

►**Exercise 70.** A network link has a latency $L = 600$ms and is perfectly reliable. What is the minimum throughput T (in bytes per second) necessary to transmit a 70KB file in less than two seconds? Briefly justify your answer. *(5')*

►**Exercise 71.** A datagram network link has a latency $L = 600$ms, throughput $T = 50$KB/s, and a maximum segment size $MTU = 1$KB. How long does it take to transmit a 50KB file using a stop-and-wait transport protocol in the absence of errors? Briefly justify your answer. *(5')*

►**Exercise 72.** A router in an IPv4 network using longest-prefix matching has the following forwarding table:

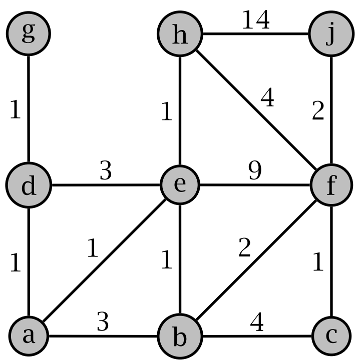
entry	*destination*	*interface*
1	39.129.0.0/16	1
2	139.57.20.128/25	1
3	39.129.128.0/18	2
4	66.160.0.0/11	3
5	222.22.0.0/16	3
6	139.57.0.0/16	4
7	66.192.0.0/10	4
8	0.0.0.0/0	5
9	66.224.0.0/11	6

Question 1: For each destination address below, write the output port and the list all the matching table entries. *(10')*

address	*output port*	*matching entries*
139.57.1.69		
66.250.226.44		
98.7.2.71		
162.100.2.1		
40.120.207.167		
139.57.20.11		
139.57.21.10		

Question 2: Write all the *inclusion* relations between table entries. (An entry x *includes* another entry y if x matches a *superset* of the addresses matched by y. *(10')*

►**Exercise 73.** Consider the following network topology.



Question 1: Assuming that routers use a link-state routing protocol, write the *link-state advertisement* announced by router b. Which nodes receive this link-state advertisement? Explain and justify your answer. *(5')*

Question 2: Now assume a distance-vector protocol. Write the *distance vector* announced by router b. Which nodes receive this distance vector? Explain and justify your answer. *(5')*

►**Exercise 74.** Your computer uses a local, iterative DNS resolver. How many DNS packets does your computer send out to resolve the address of *www.cs.colorado.edu*? Specifically, list every DNS request and plausible replies. Assume that your computer is outside the *colorado.edu* domain, that the network is reliable, and that the given name is not cached by any name server. *(10')*

►**Exercise 75.** Alex (*alex@buonasera.com*) writes the message "ciao, come stai?" to *antonio@ciao.com* and also sends a "blind carbon-copy" (Bcc) to *alberto@arrivederci.com*. Write the SMTP exchange between Alex's user agent and his mail server. *(10')*

►**Exercise 76.** *archie@carygrant.com* opens his MUA and sends an email to *grace@principaute.mc*. Grace opens her MUA and retrieves the new mail in her inbox, thus finding Archie's message.

Question 1: Describe all the the communication at the application level taking place between:

- Archie's MUA running SMTP on host host-151-100-17-8.dialup.aol.com
- the DNS resolver library that is part of the OS on the same host
- the local DNS on local-dns.dialup.aol.com that serves queries for hosts in the domain dialup.aol.com
- The mail server smtp.carygrant.com

Draw a box for each of these end-systems; describe the requests and replies as arrows, and label them with a number to describe the sequence of different communications, and with a quick description of what's exchanged (e.g. what *resource records* are sent in case of a DNS reply, and so on). *(10')*

Question 2: Do the same for the following hosts:

- Grace's MUA running POP3 on host majesty.principaute.mc
- the DNS resolver library that is part of the OS on the same host
- the local DNS on dns.principaute.mc that serves queries for hosts in the domain principaute.mc.
- The mail server pop.principaute.mc

Assume that Grace's MUA runs in retrieve-and-delete mode. *(10')*

►**Exercise 77.** A cracker gains control of the host running the local DNS system of your ISP, and runs a modified version of the DNS server, that artificially keeps forever the following Resource Record:

(ebanking.ubs.com, 108.20.213.23, IN, A)

(The IP 108.20.213.23 is the address of a web server administered by the cracker that shows a fake UBS login page, made for the purpose of stealing passwords for bank accounts.)

Question 1: In what case this situation may be harmful? How can a client avoid to be directed to the fake page? Motivate your answer. *(5')*

Question 2: Consider the situation in which the fake record is

(ubs.com, evildns.piratedomain.net, IN, NS)

where evildns.blackhat.net is a DNS server containing the previous record. Does the solution to avoid the problem given in the previous exercise still work? Motivate your answer. *(5')*

►**Exercise 78.** Consider a web page consisting of 4 JPEG images of 10KB each. The size of the HTML code is 4KB. The average latency induced by a single TCP 3-way handshake is 10ms, the average latency due to closing the connection is 5ms, and the average latency induced by the DNS query to get the IP address of the web server is 200ms. Assuming that the web server is accessible through a corporate point-to-point link that allows a throughput of 1MB/s, and assuming that the rendering time is negligible, compute the average time it takes for the browser to display the page in the following cases.

Question 1: The web server does not support persistent connections. *(10')*

Question 2: The web server supports persistent connections without pipelining. *(10')*

Question 3: The web server supports persistent connection with pipelining. *(10')*

►**Exercise 79.** Below are four TCP packets captured on the network at more or less the same time.

(1)

src-address: 34.198.10.3	
dest-address: 101.124.102.32	
src-port: 3241	*dest-port:* 5432
seq-num: 2000	
ack-num: 0	
...	
...	

(2)

src-address: 34.198.10.3	
dest-address: 101.124.102.32	
src-port: 5432	*dest-port:* 3241
seq-num: 2001	
ack-num: 0	
...	
...	

(3)

src-address: 34.198.10.3	
dest-address: 101.124.132.200	
src-port: 3241	*dest-port:* 5432
seq-num: 1	
ack-num: 2300	
...	
...	

(4)

src-address: 101.124.132.200	
dest-address: 34.198.10.3	
src-port: 5432	*dest-port:* 3241
seq-num: 2000	
ack-num: 0	
...	
...	

Which ones belong to the same TCP connection? Briefly justify your answer. Also, write another plausible packet belonging to the same connection. *(10')*

►**Exercise 80.** A network link has a latency $L = 800$ms and is perfectly reliable. What is the minimum throughput T (in bytes per second) necessary to transmit a 60KB file in less than two seconds? Briefly justify your answer. *(5')*

►**Exercise 81.** A datagram network link has a latency $L = 500$ms, throughput $T = 50$KB/s, and a maximum segment size $MSS = 1$KB. How long does it take to transmit a 50KB file using a stop-and-wait transport protocol in the absence of errors? Briefly justify your answer. *(5')*

►**Exercise 82.** Describe how *congestion control* is implemented within TCP. In particular, describe (a) how TCP detects congestion, (b) what mechanism it uses to control the sender rate, and (c) how it modulates the sender rate. *(20')*

►**Exercise 83.** List all the headers of a UDP datagram. Briefly describe the functionality of each header, specifically referring to their role as transport-level features. *(5')*

►**Exercise 84.** Describe the IPv4 packet format. Briefly explain the function of each header field. If you don't remember all the fields, at least try to focus on the most important ones. *(10')*

►**Exercise 85.** Consider the high-level architecture of a router.

Question 1: Describe the architecture of an input port and an output port of a router. Briefly describe the function of each component. *(5')*

Question 2: Describe at least two different scenarios in which a router drops packets. For each scenario, list the performance characteristics of the router (i.e., throughput of each component) and the characteristics of the input/output traffic. *(5')*

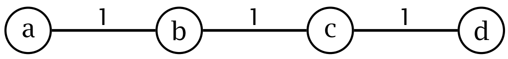
►**Exercise 86.** A router has 4 input interfaces and n output interfaces. The input lines have a maximum individual speed of 20000 packets per second. Specify the throughput of the other components of the router in such a way that no packet queues are needed. Do the results depend on the number of output interfaces? If so, say how. If not, say why. *(5')*

►**Exercise 87.** A router has one input interface and two output interfaces. The input port can receive and process $\lambda = 10000$ packets per second. The transmission rate of the two output ports are λ' and λ'', respectively. The router's manufacturer designed the router so that $\lambda = \lambda' + \lambda''$. Compute the values of λ' and λ'', assuming that:

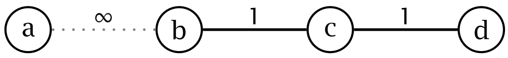
- the port with packet throughput λ' is connected to a link with throughput $t' = 10$MB/s, with $MTU = 1$KB (Assume 1K = 1024)
- the port with packet throughput λ'' is connected to a link with throughput $t' = 5$MB/s, with MTU = 512B

(10')

►**Exercise 88.** Consider the following network topology at time $t = 0$:



The four routers compute their routing tables using the distance-vector algorithm. At time $t' > 0$, after the algorithm has converged, the link connecting a to b breaks down, resulting in this topology:



Question 1: Write the distance $D_x(a)$ from router x to router a, for $x \in \{b, c, d\}$, computed by the distance-vector algorithm at each router, for at least six iterations of the algorithm. *(10')*

Iteration	$D_b(a)$	$D_c(a)$	$D_d(a)$

Question 2: Compare and contrast the way distance-vector and link-state routing deal with these cases. In particular, say which algorithm would have a faster reaction time to a link failure? Justify your answer. *(5')*

►**Exercise 89.** Express the following address ranges using the subnet prefix notation. If a range can not be represented with the prefix notation, write "N.A."

range	*subnet prefix-address/prefix-length*
88.99.100.128–88.99.100.191	
171.220.142.64–171.220.142.255	
128.138.50.0–128.138.51.255	
204.88.0.0–204.90.255.255	
108.80.0.0–108.87.255.255	
128.128.0.0–128.159.255.255	

For each valid prefix you wrote above, write the corresponding address/mask expression *(10')*

►**Exercise 90.** A small ISP administers the IPv4 address range defined by the prefix 41.195.32.0/24. The ISP has three clients. Client A requires 125 addresses, clients B and C require up to 60 addresses each. Allocate the address range of the ISP to the three clients. In particular, write the network address (address prefix) of the subnet of each client. *(10')*

►**Exercise 91.** Autonomous system AS7 has a single gateway router R, and receives the following BGP advertisements.

Network prefix	AS-PATH
81.128.242.0/24	AS2, AS1
81.128.243.0/24	AS2, AS1
81.128.0.0/16	AS2, AS1
199.203.128.0/22	AS5, AS3, AS1
199.203.136.0/21	AS4, AS3, AS1
199.203.132.0/22	AS4, AS3, AS1
81.128.128.0/22	AS6, AS2, AS1
81.128.136.0/21	AS6, AS2, AS1
138.138.132.0/22	AS6, AS2, AS1

Question 1: Write a possible AS-level topology *(5')*

Question 2: Write the minimal set of entries in the forwarding table of router R for its external interface (i.e., the interface that connects R to the outside of AS7). *(10')*

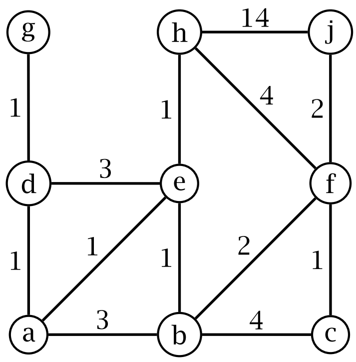
▸**Exercise 92.** Consider a block cipher $E_1 : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ with a key size of $k = 32$ bits and a block size of $n = 128$ bits. Would you use E_1 to secure your most secret communications? Briefly justify your answer. *(5')*

▸**Exercise 93.** Consider a block cipher $E_2 : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ with a key size of $k = 128$ key bits and a block size of $n = 2$ bits.

Question 1: Would you use E_2 to secure your most secret communications? Briefly justify your answer. *(5')*

Question 2: An ℓ-bit plaintext message m is encrypted with E_2 in cipher-block chaining (CBC) mode. You are given the ciphertext x but not the encryption key. Could you recover the plaintext message in a reasonable amount of time? How? How many possible plaintext messages exist that would result in the given ciphertext x? Briefly justify your answers. *(15')*

▸**Exercise 94.** Given the following network topology where link costs represent latencies in seconds. At time $t = 0$, node a sends a broadcast datagram d. Assuming that the network implements a controlled-flood broadcast, write all the packets transmitting d across the network. For each packet, write the start time, arrival time, start node, arrival node. Assume that processing time at each node is negligible.



(20')

▸**Exercise 95.** A router x issues the following *link-state advertisement* $\text{LSA}_x = \{(a,1),(b,1),(d,2)\}$ and receives the following other advertisements, where letters $(a, b, \ldots)$ represent router addresses.

$\text{LSA}_g = \{(d,2),(h,5),(f,1)\}$
$\text{LSA}_e = \{(h,2),(f,4),(a,1)\}$
$\text{LSA}_f = \{(g,1),(d,3),(b,2),(e,4),(h,4)\}$
$\text{LSA}_b = \{(a,2),(x,1),(d,1),(f,2)\}$
$\text{LSA}_d = \{(x,2),(b,1),(f,3),(g,2)\}$
$\text{LSA}_a = \{(e,1),(b,2),(x,1)\}$
$\text{LSA}_h = \{(e,2),(f,4),(g,5)\}$

Write the forwarding table of router x. Justify your answer by explaining briefly how link-state routing works. *(20')*

▸**Exercise 96.** A guy who calls himself "The Dude" sends a message to his friend Donny, sending also a "carbon-copy" (Cc) to his other friend Walter, and also sending a "blind" carbon copy (Bcc) to his lady friend, Maude. The message is the following

> **From:** the.dude@elduderino.org
> **To:** donny@bowling-alley.net
> **Cc:** walter@bowling-alley.net
> **Bcc:** maude@v-artist.org
>
> Hello Donny,
> You were throwing rocks last night!
> See you tomorrow,
> The Dude

Write all the SMTP sessions necessary to deliver the message, including the session between the Dude's user agent and the Dude's SMTP server (mail.elduderino.org). *(20')*

▸**Exercise 97.** You point your web browser to a site called *www.pizzaefichi.ch*. Your computer uses a local DNS resolver to find the address of *www.pizzaefichi.ch*. Suppose your resolver does not know the address of *www.pizzaefichi.ch* but it has in cache the address of a name server for the *ch* domain. Describe all the DNS requests, and plausible responses, issued by your resolver and by external name servers, respectively. *(20')*

▸**Exercise 98.** Describe the function of the congestion window used by TCP. Also describe the mechanisms by which TCP controls the congestion window. *(20')*

▸**Exercise 99.** Consider a link with round-trip time $2L$ = 200ms and throughput T = 200KB/s.

Question 1: Suppose the link is used by a *stop-and-wait* transport protocol with maximum segment size S = 4KB. What is the maximum utilization factor for the link? Justify your answer. (*Hint:* the utilization factor is the portion of total time in which the link transmits data.) *(5')*

Question 2: Suppose the link is used by a *go-back-n* transport protocol with maximum segment size S = 4KB. What is the (sender) window size that maximizes the utilization factor for the link? Justify your answer. *(5')*

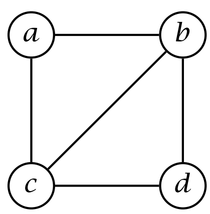
▸**Exercise 100.** A sender host wants to transmit a 20000 bytes file using TCP/IP. The host is connected through a link with a maximum packet size of S_p = 1500 bytes. What is the minimum number of bytes that the sender must push through the network. Assume that an IPv4 header uses 20, and that a TCP header uses 20 bytes. Justify your answer. *(5')*

▸**Exercise 101.** Consider a link with round-trip time $2L$ = 100ms, throughput T = 1MB/s, and maximum packet size S = 1KB. Suppose this link is used by a transport protocol with a fixed window size of W bytes. What is minimum possible latency to transmit a 100Kb file? What is the minimum window size W that achieves this minimum latency? Justify your answer. *(5')*

▸**Exercise 102.** Consider a Web document that contains very sensitive information (e.g., an on-line medical report). How can the server ask the client *not* to store a copy of the document locally? Explain this method by writing an HTTP request for that document and the corresponding server response. *(5')*

▸**Exercise 103.** Briefly describe the most important attributes of a BGP route-advertisement. *(5')*

▸**Exercise 104.** Consider the following network where each link has a total throughput of 1MB/s.



Question 1: What is the absolute maximum throughput between two applications running on hosts a and b, respectively? Briefly justify your answer. *(5')*

Question 2: Assuming the network is circuit-switched and each link supports up to 2 circuits at the same time, how main pairs of applications can communicate simultaneously from a to d? From a to b? From b to c? Briefly justify your answers. *(5')*

Question 3: In the same circuit-switched network described above in problem 2, assume that applications transmit and receive at 200 Kbps. What is the maximum throughput between any two hosts? Briefly justify your answer. *(5')*

►**Exercise 105.** Imagine a Web-based weather system where *http://www.chetempofa.ch/lugano* shows the weather for Lugano. For each city, the systems shows a page with the current temperature and weather conditions, plus a satellite image of the area. The temperature readings are updated every 2 minutes, while the satellite image is updated every 30 minutes.

Question 1: Write the *first* HTTP 1.1 conversation between a client and the server to access the Lugano page. Assume that both client and server support persistent connections. Clearly write all the requests and replies, but omit the content of the replies for binary objects (images). Clearly specify all the headers that control the properties of the connection as well as the caching policies and parameters. *(10')*

Question 2: Write a second HTTP 1.1 conversation (requests and replies) by the same client. Assume that this second request occurs three minutes after the first one. Explain how each object is handled with respect to caching. *(10')*

►**Exercise 106.** Answer the following questions about electronic mail.

Question 1: What is the difference between `MAIL FROM:` in SMTP and and the `From:` header in the mail message? *(5')*

Question 2: What is the purpose of the `Received:` header? Which component(s) of the mail systems produces the `Received:` header? *(5')*

Question 3: What is a *mail relay server*? Explain the term with an example. *(5')*

►**Exercise 107.** You compose the following message for *joe@usi.ch* with a blind carbon copy to *bob@usilu.net*.

From:	ciccio@mail.ch
Subject:	meeting
To:	joe@usi.ch
Bcc:	bob@usilu.net
I'll see you tomorrow at 3PM. Ciccio	

The MX (DNS) record for both *usi.ch* and *usilu.net* points to *mg1.ti-edu.ch*. Write the full SMTP conversation between your user agent and the server at *mg1.ti-edu.ch*. *(10')*

►**Exercise 108.** Consider a sender A and a receiver B connected by link with rate $R = 1\text{MB/s}$ and latency $d = 60\text{ms}$. Sender and receiver use the *Go-Back-N* protocol with a segment size of 8000B, a window size $W = 20$, and a timeout $T = 500\text{ms}$. How long does it take to transfer a file of $S = 80000\text{B}$ in case the link drops exactly every fourth packet in both directions? Justify your answer by writing the complete exchange between sender and receiver. *(20')*

►**Exercise 109.** A Web browser from address 100.200.12.34 connects to a Web server at address 30.40.50.60 and requests object *http://server.com/xyz*, which can not be found on the server. Write *all* the TCP packets of this connection. For each packet, specify the values of all the important headers. Fill all the unspecified headers with reasonable values. *(20')*

►**Exercise 110.** A network link has a throughput $T = 2\text{MB/s}$ and a latency $L = 2\text{ms}$. Suppose this link is used with a *stop-and-wait* protocol.

Question 1: What is the minimal segment size S that, in the absence of errors, would guarantee a link utilization of 50% or more? What is the effective throughput in this case? Justify your answers. *(5')*

Question 2: Using the same segment size S, what is the effective throughput in the presence of an error probability $p_e = 0.25$ (i.e., one in four packets gets lost)? Justify your answer. *(5')*

►**Exercise 111.** The TCP protocol is designed to adjust itself to links of varying latency. Explain how TCP does that. Explain what parameters of the protocols are relevant and how they are dynamically adjusted. Illustrate this process with an example. *(10')*

►**Exercise 112.** A router x issues the following *link-state advertisement* $\text{LSA}_x = \{(d,1),(f,1),(a,2)\}$ and receives the following other advertisements, where letters $(a,b,c,\ldots)$ represent router addresses.

$\text{LSA}_a = \{(x,2),(f,1),(c,3),(e,2)\}$
$\text{LSA}_b = \{(g,2),(c,1),(e,5)\}$
$\text{LSA}_c = \{(e,1),(a,3),(f,2),(g,4),(b,1)\}$
$\text{LSA}_d = \{(g,1),(f,2),(x,1)\}$
$\text{LSA}_e = \{(a,2),(b,5),(c,1)\}$
$\text{LSA}_f = \{(d,2),(x,1),(a,1),(c,2)\}$
$\text{LSA}_g = \{(b,2),(c,4),(d,1)\}$

Write the forwarding table of router x. Justify your answer by explaining briefly how link-state routing works and by illustrating a few steps of the Dijkstra algorithm. *(25')*

►**Exercise 113.** Answer the following questions.

Question 1: Do IPv4 headers and IPv6 headers have any fields in common? If any, describe the function of the common fields? *(10')*

Question 2: Do TCP headers and UDP headers have any fields in common? If any, describe the function of the common fields? *(10')*

►**Exercise 114.** Consider the following forwarding table

network	*port*
64.0.0.0/8	3
192.0.0.0/2	1
98.7.0.0/16	2
128.0.0.0/12	2
208.0.0.0/10	3
130.0.0.0/6	3
128.138.0.0/16	4
0.0.0.0/0	4

For each of the following destination addresses write the output port.

address	*port*
128.208.31.5	
75.21.40.22	
220.138.152.10	
130.21.86.66	
6.21.86.66	
34.60.120.159	
96.100.1.242	
128.138.241.69	
75.128.40.22	
208.71.49.43	

►**Exercise 115.** Consider a router with eight input ports, each one with a maximum throughput of 200000 packets per second, and eight output ports, four with a throughput of 240000 packets per second, and four with a throughput of 100000 packets per second. The switch fabric of the router has a throughput of 1.5 million packets per second. *(10')*

Question 1: Assuming that traffic spreads uniformly across input/output ports, is the router congested under a total input traffic of 1.4 million packets per second? If so, which queues are full at steady state? Briefly justify your answer. *(5')*

Question 2: Describe the behavior of the router under its maximum input traffic, with all its input ports running at maximum throughput. Is the router congested? If so, which queues are full at steady state? Briefly justify your answer. *(5')*

Question 3: Consider once again the behavior of the router under its maximum input traffic, and again assume that traffic spreads uniformly across input/output ports. Let d be the packet-drop rate (i.e., the number of packets dropped by the router per time unit). Assuming that every packet queue has a capacity of 1000 packets, and that all queues are empty at time $t = 0$, plot d as a function of time. Justify the result. *(10')*

►**Exercise 116.** A TCP segment with sequence number 3001 carries the following SMTP command:

```
MAIL FROM: <dude@elduderino.org>
```

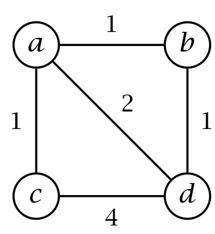
What is the sequence number of the next TCP segment going from the client to the server? Can you also determine the sequence number of the next TCP segment going from the server to the client? If so, what is that sequence number? Briefly justify your answers. *(10')*

►**Exercise 117.** For each one of the following subnet addresses, give an example of an IP address that can be assigned to that subnet, and one that can not.

subnet	*IP address in subnet*	*IP address outside subnet*
192.0.0.0/4		
230.208.32.0/28		
88.68.124.132/30		
103.124.20.128/26		
128.129.0.0/16		
128.131.64.0/18		
53.220.211.0/24		
100.0.0.0/6		
203.242.138.0/18		
184.180.0.0/12		

Also, do any of these network addresses overlap? If so, which ones? *(10')*

►**Exercise 118.** Consider the following network where routers use a distance-vector routing protocol.



Assuming that routers exchange routing information synchronously (at the same time), illustrate the state of each router until the routing protocol stabilizes. *(25')*

►**Exercise 119.** Consider the server of a mailing list called *studentparties@lists.unisi.ch*. Suppose *joe@unisi.ch*, *mario@unisi.ch*, and *pippo@mail.ch* are subscribed to the list, and that *bart@blabla.com* sends the following message to the list:

> **From:** bart@blabla.com
> **To:** studentparties@lists.unisi.ch
>
> Let's all go to the beach today!
> See you there,
> Bart

Write *all* SMTP interactions necessary to send and deliver the message, between Bart's user agent and the list server, and then between the list server and all the mail servers of the subscribers. For simplicity, you may assume that server responses are always positive, and therefore you may ignore them. *(20')*

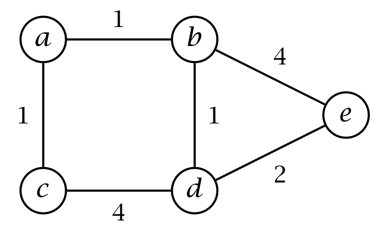
►**Exercise 120.** Explain the difference between routing and forwarding. *(10')*

►**Exercise 121.** Describe the circumstances in which a router drops packets at output ports. Give an example in which this situation can occur, specifying the traffic conditions as well as maximum throughput of each component of the router. *(10')*

►**Exercise 122.** Describe the connection-setup phase of TCP. Describe each packet sent by client and server highlighting the relevant headers, the information they carry, and how it relates to the state of the connection maintained by client and server. Give three examples: one where the connection phase is completed successfully, one where the connection is refused, and one where the connection times out. *(20')*

►**Exercise 123.** Consider a *stop-and-wait* reliable transport protocol. Specify a variant of this protocol in which the receiver sends a second (repeated) acknowledgement, after having received and immediately acknowledged a data segment, if it does not receive another data segment within 100ms. Write the finite-state machine that specifies the receiver. Also argue whether this receiver behavior can help to reduce transmission time. Support your argument by drawing one or more diagrams of a example session between a sender and a receiver. (*Hint:* the diagrams show the exchange of data along two vertical time lines representing the sender and receiver, respectively.) *(30')*

►**Exercise 124.** Consider the following network where routers use a distance-vector routing protocol.



Write the routing information transmitted by every router until the protocol stabilizes. Assume that routers exchange routing information synchronously once per second. Therefore, identify each message with a progressive time-stamp (1, 2, ...), a source, and one or more destinations. *(30')*

►**Exercise 125.** Three important properties of a communication channel are cost of setup and maintenance, ability to share the channel among multiple senders/receivers, and quality of service. Compare and contrast circuit switching and packet switching especially with regard to these three factors. *(10')*

►**Exercise 126.** A web server is serving requests for the *www.speedopizza.com* web site. The site consists of a single page containing the logo of the restaurant and the names and images of 4 types of pizza. The web site is implemented with static files placed in a given directory. This is the content of this directory:

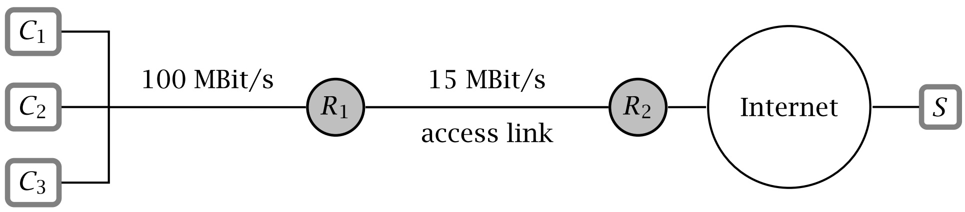
```
-r--r--r--  apache apache   4123 2009-04-24 14:01 index.html
-r--r--r--  apache apache 214123 2009-04-24 14:01 logo.jpg
-r--r--r--  apache apache  70534 2009-04-24 14:01 porcini.jpg
-r--r--r--  apache apache  55912 2009-04-24 14:01 sicilia.jpg
-r--r--r--  apache apache  65109 2009-04-24 14:01 peperoni.jpg
-r--r--r--  apache apache  94388 2009-04-24 14:01 diavola.jpg
```

Question 1: Write the HTTP replies corresponding to the following requests received by the web server. Be as specific as you can, including, the appropriate headers. If the reply has a body, then just write the single line "...BODY..." *(10')*

```
HEAD /index.html HTTP/1.1
Host: www.speedopizza.com
Connection: close

GET /logo.jpg HTTP/1.1
Host: www.speedopizza.com
Connection: close

GET /porcini.jpg HTTP/1.1
Host: www.speedopizza.com
Connection: close
If-Modified-Since: Mon, 27 Apr 2009 12:01:00 GMT

GET /napoli.jpg HTTP/1.1
Host: www.speedopizza.com
Connection: close
If-Modified-Since: Mon, 12 Jan 2009 14:28:24 GMT
```

Question 2: The restaurant offers a special *pizza of the day* that is advertised on the web site. However, the owner of the restaurant notices that some of the customers ask for the *pizza of the day* that was offered two days ago, even though he has updated the web page since then. Briefly explain what could cause the problem and suggest how the problem could be solved by configuring the server to send an appropriate HTTP header with the index page. *(5')*

►**Exercise 127.** The schematic diagram below shows three clients C_1, C_2 and C_3 connected to the Internet through a 15MBits/s access link. Using HTTP, clients fetch objects of average size of 900,000 bits, at a total rate of 10 requests per second (for all three clients). Suppose that the amount of time that it takes from when the router R_2 forwards an HTTP request until it receives the response (*Internet delay*) is 2 seconds on average. Model the total average response time as the sum of the *Internet delay* and the average *access delay*, which is the delay between R_2 and R_1. For the average access delay use the formula $\Delta/(1-\Delta\beta)$, where Δ is the average time required to send an object over the access link and β is the arrival rate of objects to the access link. (Is this formula *dimensionally* correct?)



Question 1: Find the total average response time. Briefly justify your answer. *(10')*

Question 2: Now suppose a cache is installed in the same local network with the clients. assuming that the hit rate is 60%, find the total response time. Briefly justify your answer. *(5')*

Question 3: Explain the meaning of the formula $\Delta/(1-\Delta\beta)$ that computes the average transmission delay of multiple objects arriving at a constant rate at the same link. The throughput of the link is T (bits/second), the average size of the objects is S (bits), $\Delta = S/T$, and the objects arrive at a rate β (objects per second). *(10')*

▶**Exercise 128.** Answer the following questions about Internet electronic mail.

Question 1: You send an e-mail to *name@eecs.berkeley.edu*. How does your mail user agent (or your mail transport server) find the IP address of the mail server responsible for these mailboxes? *(5')*

Question 2: What is *MIME* and how is it used to extend Internet mail? Write an example message detailing the relevant headers and the relevant content fragments. *(10')*

▶**Exercise 129.** Compare and contrast the *Selective Repeat* and *Go-Back-N* protocols. Describe the advantage of *Selective Repeat* with an example. *(10')*

▶**Exercise 130.** An implementation of the *Selective Repeat* protocol uses sequence-numbers from 1 to 4 (i.e., the packets are numbered $1 \to 2 \to 3 \to 4 \to 1 \to 2 \ldots$). The window size, on both the sender and the receiver, is set to 3. The underlying network is unreliable and packets might get lost or delivered out of order, but a checksum guarantees that their content is error-free.

Question 1: This setting is problematic because in some cases the receiver can not decide whether a packet is new or it is a retransmission of a previous packet. Show a scenario that illustrates this problem. Use arrows to represent packets and label them with their sequence numbers. *(10')*

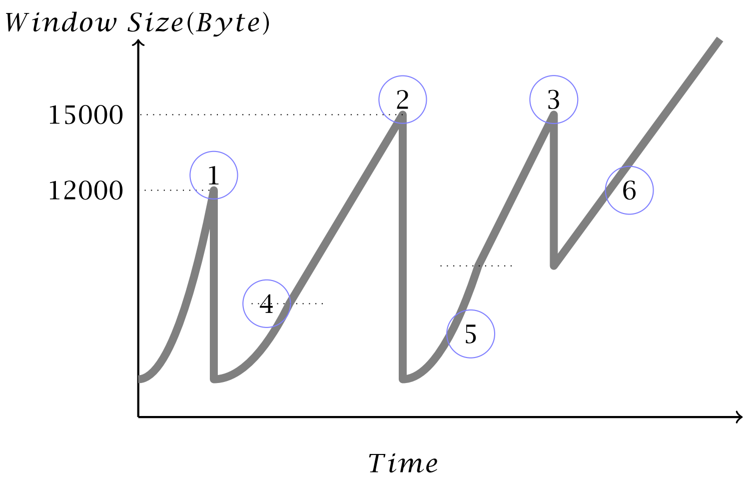
Question 2: How can you correct this problem with a minimal change in the code? *(5')*

▶**Exercise 131.** A and B are communicating over a TCP connection. B sends a packet with ACK number 490, which is received by A. Suppose A then sends two segments to B, one immediately after the other. The first and second segments contain 50 and 70 bytes of data, respectively. In the first segment, the source port number is 1028 and the destination port number is 21. B sends an ACK after receiving each packet.

Question 1: What are the sequence number, source port, and destination port of the second segments sent from A to B? Briefly justify your answer. *(5')*

Question 2: Assume the first segment arrives before the second segment, so B sends an acknowledgment after each segments it receives. What is the sequence number in each acknowledgment? Briefly justify your answer. *(5')*

▶**Exercise 132.** Host A is sending a file to Host B over a TCP connection. The diagram below plots the size of the congestion window over time, in the presence of events labeled 1 through 6.



Question 1: What is the state of the TCP state machine at host A when events 1, 2, and 3 occur? *(5')*

Question 2: What is the window size of the sender at event 4? *(5')*

Question 3: Briefly explain the behavior of TCP during events 5 and 6, and also the purpose of those behaviors. *(10')*

▶**Exercise 133.** Answer the following questions on IP addressing.

Question 1: Write the number of IPv4 addresses in each of the following network addresses. Briefly justify your answers explaining the meaning of the prefix notation.
142.11.240.0/22
127.0.0.0/8
192.168.0.0/16 *(5')*
128.138.242.0/24
0.0.0.0/0

Question 2: Explain the concept of *supernetting* giving an example in which three subnet addresses are combined. *(5')*

▶**Exercise 134.** Answer the following questions on forwarding.

Question 1: An IPv4 router has 16 physical interfaces, which function as both input and output interfaces. The network in which the router lives has a total of N addresses. Considering the forwarding table as a mathematical function, write the domain and range of the forwarding function. At most, how many entries does the forwarding table contain? How many addresses does the router use? Briefly justify your answers. *(10')*

Question 2: Does a router in a virtual-circuit network have a forwarding table? If so, how is that different from the forwarding table in a datagram network? Briefly justify your answers writing also the domain and range of the forwarding function in a virtual-circuit network. *(10')*

▶**Exercise 135.** Answer the following questions on routing.

Question 1: Briefly explain, using an example, the notion of *hierarchical routing*, and how that is realized in today's Internet. In particular, explain the role of inter-domain and intra-domain routing and explain how each contributes to building the forwarding tables. *(10')*

Question 2: A router has 5 input interfaces and 5 output interfaces, each with the same throughput x. The router also has a switch fabric operating at maximum throughput y.

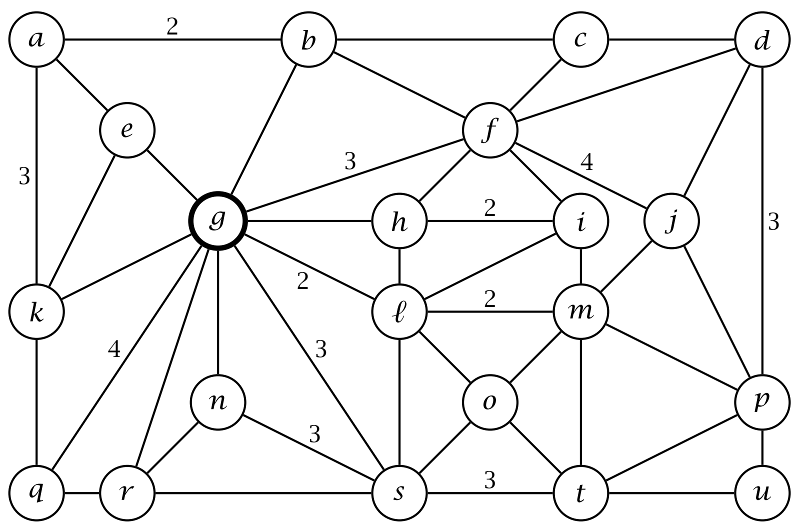
a. Are there values of x and y for which the router can operate without input queues? If not, explain why. If so, explain how.

b. Are there values of x and y for which the router can operate without output queues? If not, explain why. Is so, explain how.

c. Are there values of x and y for which the router can operate without input and output queues? If not, explain why. Is so, explain how.

(10')

▶**Exercise 136.** Consider the following network. Link costs are 1 except where otherwise indicated. Use Dijkstra's algorithm to compute the forwarding table of router g. Write the result in the first table below. Also, show the state of the algorithm at every step using the second table below and if necessary in the next page. (*Hint:* the state of the algorithm consists of a "distance" vector and a "previous" vector, so for each destination and each step, write the distance followed by the previous.) *(30')*



Forwarding Table

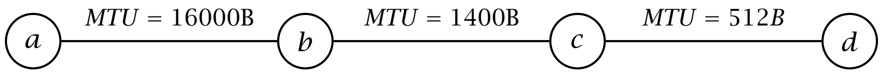
a	*b*	*c*	*d*	*e*	*f*	*g*	*h*	*i*	*j*	*k*	*ℓ*	*m*	*n*	*o*	*p*	*q*	*r*	*s*	*t*	*u*

Execution of Dijkstra's Algorithm

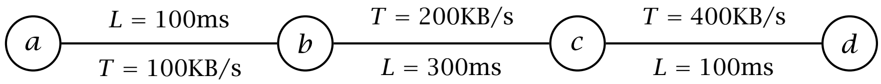
step	*a*	*b*	*c*	*d*	*e*	*f*	*g*	*h*	*i*	*j*	*k*	*ℓ*	*m*	*n*	*o*	*p*	*q*	*r*	*s*	*t*	*u*
1																					
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…

step	*a*	*b*	*c*	*d*	*e*	*f*	*g*	*h*	*i*	*j*	*k*	*ℓ*	*m*	*n*	*o*	*p*	*q*	*r*	*s*	*t*	*u*
1																					
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…

►**Exercise 137.** Consider the following network path from *a* to *d*, where each link has the noted MTU. Router *a* sends an IPv4 packet to *d* containing a UDP datagram with 4000 bytes of payload. The packet is fragmented along the way. Write all the fragments received by *d*, specifying the relevant fragmentation information. *(20')*



►**Exercise 138.** Consider the following network path from *a* to *d*, where each link has the noted maximum throughput *T* and latency *L*. Suppose host *a* runs a Web browser that accesses, through HTTP, a document on server *d*. The document consists of 4 objects of 1000B, 5000B, 6000B, and 100KB, respectively.



Question 1: How long does it take for the browser to receive all the web objects with and without pipelining? Ignore the behavior of the underlying TCP connection, so assume that each host can sends packets back-to-back, and that they are all received correctly and in order. Justify your answers. *(10')*

Question 2: Now assume that host *b* runs a (transparent) caching proxy. What is the total delivery time if only the first object is in cache? What is the total delivery time if only the second object is in the cache? What is the total delivery time if only the fourth object is in the cache? Justify your answers. *(10')*

►**Exercise 139.** Hosts *A*, *B*, and *C* are connected to the Internet through asymmetric access links. The following table lists the maximum upload and download rates for the three hosts.

host	*max upload*	*max download*
A	100KB/s	500KB/s
B	50KB/s	100KB/s
C	60KB/s	400KB/s

Assume that the core of the network does not further reduce transfer rates, and does not introduce significant latencies. What is the best way to transfer a 500MB file from host *A* to hosts *B* and *C* so as to minimize each of their respective transfer time? What are the transfer times in this case? How low can *B*'s upload rate be without incurring any increase in transfer time? What are the transfer times if *B*'s upload rate is reduced to 30KB/s? What if *B*'s upload rate remains at 50KB/s, but its download rate is reduced to 80KB/s? Answer each question in turn. Briefly justify your answers. *(20')*

►**Exercise 140.** Explain the longest-prefix matching algorithm used in Internet forwarding. Explain how it works and why forwarding is carried out that way. Show an example in which address ranges are assigned to subnets in a non-trivial way, in particular in a way that justifies longest-prefix matching. With this assignment, show a few examples of how addresses in the various ranges are forwarded. *(20')*

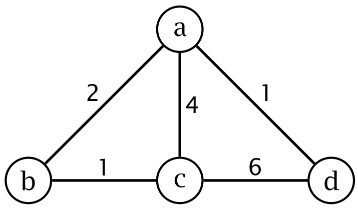
►**Exercise 141.** You are sending an e-mail message to *friend@someschool.edu*. Describe every step, including DNS and SMTP, of the process undertaken by either your user agent or your mail server to deliver the message to the mail server of the recipient. Assume the address is valid and that no communication or server errors occur. *(20')*

►**Exercise 142.** Consider an HTTP 1.0 connection from a user agent to a server to retrieve a 4Kb object called *index.html*.

Question 1: Write the HTTP request and a successful reply (omitting the content of the reply). *(5')*

Question 2: Write all the TCP segments exchanged for the HTTP request and reply. Assume an MTU of 1400 bytes. For each segment, write all the relevant information, including port numbers, sequence number, ack number, flags, size, etc. *(15')*

►**Exercise 143.** Consider the following network where routers use a distance-vector routing protocol.



Assume that all routers start at time 0, that routing messages are sent periodically every 10 seconds. Assume also that links have a fixed latency of 1 second. Write the iterations of the distance-vector routing algorithm, at times 0, 10, ..., until the protocol converges to a stable state. For each iteration, specify the time and list the routing tables of each router. *(20')*

►**Exercise 144.** How and why does TCP estimate the network-level round-trip time for its connection? How is the estimated round-trip time used in the protocol? Describe and explain the estimation algorithm using an example. Also, discuss the goal of this algorithm, showing again by example what would happen if the round-trip time is underestimated or overestimated. *(20')*

►**Exercise 145.** A file of size $S = 1\text{GB}$ ($1\text{ GB} = 10^9\text{B}$) available from host a is downloaded by n hosts, $b_1, b_2, \ldots, b_n$. Host a has a maximum upload (output) throughput $U_a = 2$ MB/s and a maximum download (input) throughput $D_a = 1$ MB/s. Hosts b_1 through b_n each have a maximum upload (output) throughput $U_b = 100$KB/s and a maximum download (input) throughput $D_b = 500$ KB/s. All receiver hosts start their download at the same time $t = 0$ and the download finishes at time T when all hosts $b_1, b_2, \ldots, b_n$ have obtained a copy of the entire file.

Question 1: What is the best total download time T_{cs} achievable with a "client-server" protocol with $n = 10$ receivers? Show and briefly justify your calculations. *(10')*

Question 2: What is the best total download time T_{p2p} achievable with a "peer-to-peer" protocol with $n = 10$ receivers? Show and briefly justify your calculations. *(10')*

Question 3: What is largest number of receivers n for which a client-server protocol is not worse than a peer-to-peer protocol? Show and briefly justify your calculations. *(10')*

►**Exercise 146.** Consider the following HTTP request and the corresponding reply:

```
PUT /some/file/called/blah HTTP/1.0
Host: www.example.edu
Content-Type: text/plain
Content-Length: 1000

blah blah blah...
```

```
HTTP/1.0 405 Method Not Allowed
Content-Length: 400
Content-Type: text/html
Connection: close

<html><head>
<title>Error: Method Not Allowed</title>
</head><body>...
```

This request and reply are sent over a TCP connection between client c on port 1234 and server s on port 80. Write every TCP packet sent over this connection, including those for the opening and closing of the connection. For each packet, write (1) the source address (c or s), (2) the destination address (c or s), (3) the source port, (4) the destination port, (5) the sequence number, (6) the acknowledgment number, (7) the main flags (ACK, SYN, FIN), and (8) the first few bytes of its data (if any). Assume a maximum packet size of 1500 bytes, which means that each TCP packet can carry at most 1440 bytes of data. Use the template below to write each packet. *(30')*

src IP	*dst IP*	*src port*	*dst port*	*seq. number*	*ack. number*	*flags*
content:						

►**Exercise 147.** You send an e-mail message to your friend *amir@amici.ch* with a copy to your other friend *marco@amici.ch*, and with a "blind" carbon copy to *antonio@usi.ch*. The subject of the message is "Pizza," and the text of the message is "Let's go out for a pizza tonight." Assuming your e-mail address is *student@usi.ch* and that your mail user-agent is configured to use the mail server *mail.usi.ch*, write the complete SMTP exchange between your user agent and the mail server, including the whole body of the message. *(10')*

►**Exercise 148.** A sender transfers a 200 MB file to a receiver using a "stop-and-wait" reliable transport protocol with a maximum segment size of 4 KB and timeout of $\Delta = 500$ ms, through a network link with latency $L = 40$ ms and throughput $T = 200$ KB/s.

Question 1: How long does the transfer take in the absence of errors or losses in the network? Show and briefly justify your calculations. *(10')*

Question 2: What is the expected transfer time in case the network looses on average one out of 1000 packets? Show and briefly justify your calculations. *(10')*

►**Exercise 149.** Answer the following questions. *(10')*

Question 1: Does UDP provide any support for reliability? If so, in what way and through which header fields?

Question 2: Does TCP provide any support for reliability? If so, in what way and through which header fields?

Question 3: What is the effect of the header *Connection: close* in an HTTP request?

Question 4: Compare packet switching and circuit switching in terms of the efficiency in the usage of network links? Which one is better? Why?

►**Exercise 150.** Your computer uses a DNS resolver connected through a local-area network with latency $L_l = 1$ ms. The DNS resolver is connected to the Internet through an access link with $L_h = 50$ ms. Assuming no other latencies in the whole network, and considering that DNS packets have negligible sizes, how long does it take for your computer to resolve the address *www.chomsky.info*. Assume also that neither the requested name nor its domain are in the DNS cache (local or remote). Show and briefly justify your calculations. In particular, list all the packets sent and received by your local resolver. *(20')*

►**Exercise 151.** A user accesses a Web document consisting of a 10KB HTML page plus a 100 KB image and a 10MB video clip, all stored at their origin server S. The user's browser C is configured to access the Web through a caching proxy P. The connection between the browser and the proxy has latency $L_{CP} = 1$ms and throughput $T_{CP} = 1$MB/s. The connection between the proxy and the origin server has latency $L_{PS} = 100$ ms and throughput $T_{PS} = 100$ KB/s. *(20')*

Question 1: What is the total transfer time when the image is already in cache? Show and justify your calculations.

Question 2: What is the total transfer time when the video clip is in cache? Show and justify your calculations.

►**Exercise 152.** An application running on host A opens a TCP connection with an application running on host B, and immediately starts transferring a large file. The latency between A and B is $L = 200$ms, the maximum throughput between A and B is $T = 500$KB/s, and the maximum segment size is $MSS = 1000$B. The receiver has plenty of capacity, and therefore you should assume that the receiver window is always larger than the congestion window.

Question 1: How long does it take for the TCP connection to reach the maximum throughput? Justify your answer by showing and briefly describing your calculation. (*Hint:* recall that, in the initial "slow-start" phase, the sender opens the congestion window exponentially, increasing its size by one segment for each acknowledged segment. Remember also to consider the initial handshake.) *(10')*

Question 2: Assume that a packet is dropped only when the sender rate goes over the maximum throughput T. Also assume that only sender segments are dropped, and therefore that acknowledgments are always received correctly. In this case, what is the effective throughput of the TCP connection over a long period of time? Justify your answer by showing and briefly describing your calculation. (*Hint:* show the throughput controlled by TCP over time; model the network as a kind of conveyor belt whose *capacity*—the maximum amount of data it can contain at any given time—determines the maximum amount of data that can be sent without losses.) *(10')*

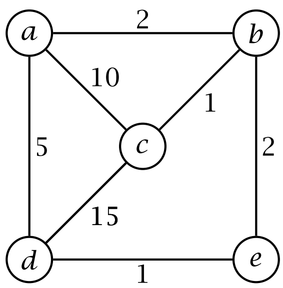
►**Exercise 153.** A router with two interfaces is configured with the following forwarding table:

network address	*interface*
24.36.0.0/16	1
138.128.0.0/16	2
125.200.192.0/18	2
31.0.0.0/8	1
138.89.0.0/16	2
31.98.7.0/24	2
138.128.10.0/24	2
31.80.66.0/24	2
0.0.0.0/0	1
125.200.128.0/18	2
125.201.0.0/16	2
31.99.0.0/16	2

Question 1: Where would the router forward a datagram addressed to 31.99.100.101? Justify your answer by describing at a high-level forwarding algorithm used in IP routers. *(10')*

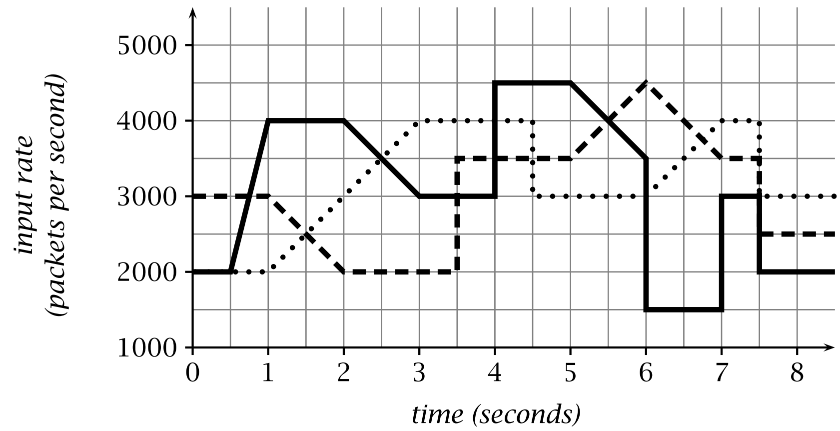
Question 2: Can the router compress its forwarding tables? (I.e., reduce the number of table entries.) If so, show the minimal set of entries that are exactly equivalent to the given table. *(10')*

►**Exercise 154.** Consider the network shown below in which routers use a distance-vector routing algorithm.



In the initial state each router knows the costs of the links to its neighbors. Assuming that routers exchange distance vectors in a synchronous way (i.e., all at the same time) show the state of router a after each update to its routing tables until the protocol converges. *(20')*

►**Exercise 155.** A router has three input ports, a switch fabric capable of sustaining a maximum throughput of 9000 packets per second, and a number of output ports each capable of sending 9000 packets per second. Each input port has a queue of size q, and starts with an empty queue at time 0. The three input ports receive the traffic described by the three graphs below, respectively.



Assume that the switch fabric processes packets in a round-robin fashion (i.e., one packet from each input port in turn; if no packets are available from one port, the switch fabric skips to the next port, and so on). What is the minimum queue size $\overline{q}$ that would allow the router to process all input packets? Assuming that $q = \overline{q}/2$, at what time does the router drop the first packet? Justify your answers by showing and briefly explaining your calculations. *(20')*

►**Exercise 156.** Answer the following questions.

Question 1: Explain how IPv4 fragmentation works using an example in which a datagram is split up in at least three fragments. *(5')*

Question 2: Describe the IPv6 datagram format. Discuss at least two headers other than the version and the source and destination address, explaining their purpose. *(5')*

Question 3: Explain how TCP closes a connection. Draw a diagram of the various segments sent by the two sides. Explain what happens if any segment is lost. *(5')*

Question 4: Describe at a high level a symmetric encryption scheme to protect the privacy of a message m sent by sender S to receiver R. Also explain briefly what it means for this scheme to be perfectly secure. *(5')*

►**Exercise 157.** Consider a reliable connection between a sender and a receiver implemented with a Go-Back-N protocol with maximum segment size MSS = 2KB and with a fixed window size $W = 8$. The network latency between the sender and receiver is L = 200ms, the access link of the sender allows a maximum transmission throughput T_S = 100KB/s, and both sender and receiver have very high reception throughput.

Question 1: How long does it take to reliably transmit a file of size S = 100KB in the best case, without any loss of packets? Justify your answer. *(10')*

Question 2: Assuming the timeout is set to 1s, how long does it take to reliably transmit the same file (S = 100KB) in the presence of losses when one out of 20 packets gets lost in both directions? Justify your answer. *(10')*

►**Exercise 158.** A router has 16 input interfaces capable of receiving up to 10000 packets per second, and 16 output interfaces capable of sending 20000 packets per second.

Question 1: Let T_S be the maximum throughput of the switch fabric. What is the maximum (total, best-case) throughput of the router? In particular, are there values of T_S for which the total throughput does not depend on T_S? Justify your answers. *(10')*

Question 2: Consider the previous question in the case of a specific distribution of traffic. Let T_S be the maximum throughput of the switch fabric, and suppose that 20% of the input traffic goes to output interface 1, 10% to output interface 2, and the rest is distributed uniformly onto all other interfaces. What is the maximum throughput of the router in this case? Also, for what values of T_S (if any) does the router drop packets on its input and output queues, respectively? Are there any values of T_S for which neither queues would ever be full? Justify your answers. *(10')*

►**Exercise 159.** An HTTP connection is carried by a TCP connection with maximum segment size of 1400 bytes.

Question 1: Within this connection, a TCP segment with sequence number 3344 carries the following HTTP request:

```
HTTP/1.0 404 Not Found
Content-Type: text/html
Content-Length: 41

<html><body>Page Not Found!</body></html>
```

What is the sequence number of the next segment going from the Web server to the client? Justify your answer. (Remember that all HTTP header lines end with a CRLF sequence.) *(10')*

Question 2: Within the same connection, a TCP segment with sequence number 6677 carries the following HTTP request:

```
HTTP/1.0 200 Ok
Content-Type: image/jpg
Content-Length: 25000

a binary image...
```

What is the sequence number of the next segment sent by the server in this case? Justify your answer. *(10')*

►**Exercise 160.** Antonio (*antonio@usi.ch*) sends a message to a mailing list dedicated to teaching and related discussions. The list is served by *lists.org*. Thus, the destination of the message is *teachers@lists.org*, and the subject line is "retake exam." Among the subscribers is Cyrus (*cyrus@usi.ch*) who will therefore receive a copy of Antonio's message. Describe exactly what happens in terms of application-level protocols for each of these two message exchanges, starting from the necessary DNS queries, and then proceeding with the SMTP sessions. Simply describe the DNS requests and replies without detailing the DNS resolution processes. Then show the entire SMTP session for each exchange. You may skip the server replies in the SMTP sessions, but make sure you clearly specify the relevant information, including the sending server, the receiving server, the envelope destination and sender, and the message destination and sender. *(30')*

►**Exercise 161.** Describe an IPv6 datagram containing a TCP segment. Describe both the IP and TCP headers. Describe as many header fields as you can remember. For each field, briefly describe the purpose of the field and the allowable values. *(10')*

►**Exercise 162.** Router x issues the *link-state advertisement* $\mathrm{LSA}_x = \{(g,1),(f,3),(a,2),(e,4),(b,1)\}$ and receives the following other advertisements, where letters $(a,b,c,\ldots)$ represent router addresses.

$\mathrm{LSA}_a = \{(d,2),(c,1),(f,1),(x,2)\}$
$\mathrm{LSA}_b = \{(e,2),(x,1),(g,5)\}$
$\mathrm{LSA}_c = \{(d,1),(a,1),(f,2)\}$
$\mathrm{LSA}_d = \{(e,1),(a,2),(x,1)\}$
$\mathrm{LSA}_e = \{(b,2),(x,4),(d,1)\}$
$\mathrm{LSA}_f = \{(c,2),(a,1),(x,3),(g,2)\}$
$\mathrm{LSA}_g = \{(f,2),(b,5),(x,1)\}$

Write the forwarding table of router x. Identify the output interfaces with the corresponding neighbor router. Justify your answer by explaining briefly how link-state routing works. *(20')*

►**Exercise 163.** A mailing-list server called *lists.inf.usi.ch* receives the following message posted to the *jokes* list:

From:	Antonio Carzaniga <antonio.carzaniga@usi.ch>
Subject:	a good one by Yogi Berra
To:	Jokes Mailing List <jokes@lists.inf.usi.ch>

You can observe a lot by just watching.

The *jokes* list has four subscribers: *koorosh@usi.ch*, *amir@usi.ch*, *gino@colorado.edu*, and *antonio.carzaniga@usi.ch*. Write all the SMTP sessions that the server uses to distribute the message to the members of the list. Do not worry about remembering the exact numeric codes sent by the receiving server, but be precise in listing *everything* the sender writes in each session. Assume all sessions are successful and without errors. *(20')*

►**Exercise 164.** You open your web browser and go to the url *http://www.usi.ch/slogan.jpg*. Your browser then fetches and displays the page (an image). Write every network packet that your computer sends and receives to accomplish this task. For each packet, write the important transport-level headers as well as the relevant application-level content (abbreviate the content of the image with "...")). The important transport-level headers are the port numbers (for both TCP and UDP) and sequence and acknowledgment numbers, and flags (for TCP). For example, for an HTTP request, write the TCP headers as well as the HTTP request. Assume the maximal transmission unit of your network is 1500 bytes, and the size of the *slogan.jpg* image is 4000 bytes. *(20')*

►**Exercise 165.** Twenty users download a large file through a peer-to-peer system. The size of the file is 2GB. The whole file is available from two other "seed" users S_1 and S_2 connected to the network with an access links of maximal upload rates U_1 = 200KB/s and U_2 = 500KB/s, respectively. Of the 20 users, 5 have a "fast" access link and 15 have a slow access link. A "fast" access links has maximal upload and download rates of U_{fast} = 100KB/s and D_{fast} = 500KB/s, respectively. A "slow" access links has maximal upload and download rates of U_{slow} = 50KB/s and D_{slow} = 200KB/s, respectively. What is the (theoretical) best total download time? This is the time it takes for all twenty users to obtain the whole file. Justify your answer by showing and briefly explaining your calculations. (For simplicity, let 1GB and 1Kb represent 10^9 and 10^3 bytes, respectively.) *(20')*

►**Exercise 166.** A sender sends a file to a receiver using TCP. The sender application simply connects to the receiver and sends the file as fast as the TCP connection permits, and then closes the connection; the receiver accepts the connection, reads from it as fast as data comes in, and then closes the connection. The network between the sender and the receiver has a transmission delay d = 500ms, a maximum transmission rate R = 200KB/s, and a maximum segment size MSS = 1KB. The network introduces no transmission errors or packet losses when the transmission rate is less than R. Notice that R is the maximum rate of the entire path between sender and receiver. However, the sender may try to send at a higher rate, in which case those packets will be dropped by the network. More specifically, every packet that causes the instantaneous transmission rate to exceed R is dropped. (Hint: the instantaneous transmission rate induced by a packet p is the size of p divided by the time between the transmission of p and the transmission of the immediately preceding packet p' *that was not itself dropped.*)

Question 1: Exactly how long does it take for the sender (and receiver) to complete the transmission of an 80Kb file? *(10')*

Question 2: Does the network ever drop any packets in the transmission of the 80Kb file? If so, exactly when does that happen for the first time? If not, what is the minimum file size that would causes the network to drop a packet? Justify your answers by showing and briefly explaining your calculations. *(10')*

Question 3: What is the effective transmission rate of the TCP connection for a continuous stream of data? That is, the rate available to the application (as opposed to the network-level rate available at the transport layer) computed by excluding the initial "slow start" phase of the TCP transmission. Justify your answers by showing and briefly explaining your calculations. *(10')*

►**Exercise 167.** A user U connects to the Web through a caching proxy P. U goes to a Web page consisting of an HTML file *one.html*, size 20Kb, and two image files, *two.jpg* and *three.jpg*, sizes 100Kb and 80Kb, respectively, all of which are from the same origin server S and that are requested by U in the given order (one, two, three). The link between U and P has transmission delay d_1 = 10ms and rate R_1 = 1000KB/s; the link between P and S has transmission delay d_2 = 100ms and rate R_2 = 100KB/s. Suppose all three objects are in P's cache, where they were retrieved at times T_1, T_2, and T_3, respectively. Feel free to assume that requests are pipelined. In any case, state your assumptions explicitly.

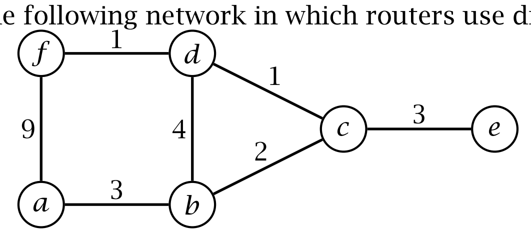
Question 1: Write all HTTP requests and their corresponding replies involved in this interaction, indicating for each request/reply which is the client-side and which is the server-side. Assume in this case that the cached copies of *one.html* and *three.jpg* are valid, and instead *two.jpg* was modified after T_2. Abbreviate the object content in the replies (if any) by writing "..." *(10')*

Question 2: Exactly how long does it take for the whole page to be transmitted to the user in the case described above in exercise 1? Justify your answers by showing and briefly explaining your calculations. *(10')*

Question 3: Exactly how long does it take for the whole page to be transmitted to the user in the case where the cached copies of *one.html* and *two.jpg* are valid, and instead *three.jpg* was modified after T_3? *(10')*

►**Exercise 168.** Compare and contrast distance-vector and link-state routing. List and briefly explain their differences. *(10')*

►**Exercise 169.** Consider the following network in which routers use distance-vector routing.



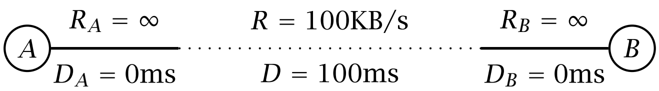
Question 1: Show the forwarding table of router a from the initialization until the state of the table converges. Assume that all routers initialize at the same time, and that, at each round of the algorithm, each router receives distance vectors from all its neighbors simultaneously. *(20')*

Question 2: After convergence, what happens if the cost of the link between c and d raises from 1 to 4? Again, show the evolution of the forwarding table of router a. *(10')*

►**Exercise 170.** Compare and contrast client/server (e.g., with HTTP) and peer-to-peer (e.g., with BitTorrent) file transfer. In particular, describe a situation in which peer-to-peer is better in terms of total transfer time. *(10')*

►**Exercise 171.** Compare and contrast IPv4 and IPv6. In particular, briefly describe the most important header fields in the two protocols. *(10')*

►**Exercise 172.** Consider two hosts A and B connected through a network as in the following diagram:



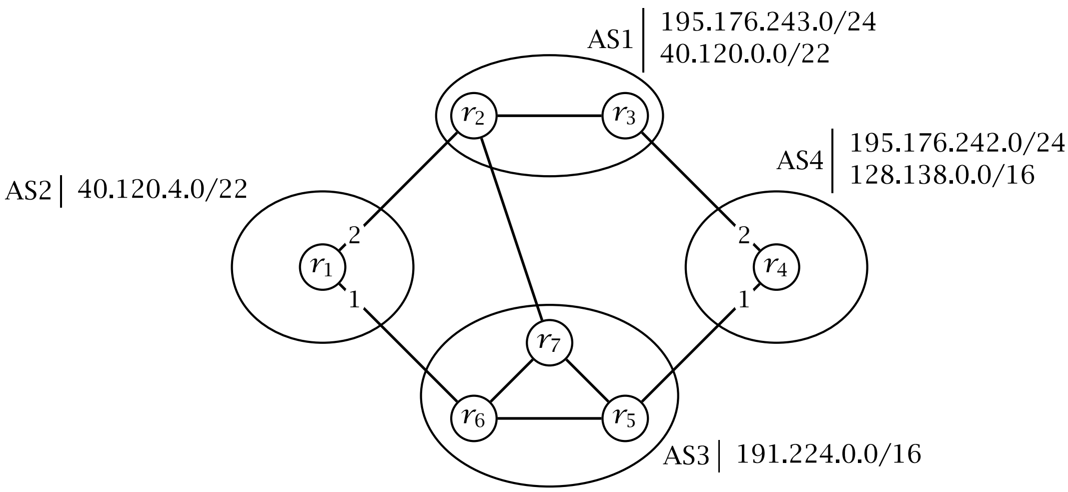
A and B are connected to the network with access links with infinite transmission rate and zero transmission delay. The network path that connects A and B has a maximum transmission rate $R = 100$KB/s, a total transmission delay $D = 100$ms, and maximum segment size $MSS = 1$KB. The network introduces no transmission errors or packet losses when the transmission rate computed over a period of D is less than R. Formally, every packet sent at time t_0 is successfully transmitted if and only if the average transmission rate of successfully transmitted packets in the interval between $t_0 - D$ and t_0 was more than R.

Question 1: How long does it take for A to transfer a 500KB file to B using the go-back-N protocol with a fixed window of $W = 10$ segments? Justify your answer. *(10')*

Question 2: How long does it take for A to transfer a 500KB file to B using the go-back-N protocol with a fixed window of $W = 20$ segments? Justify your answer. *(10')*

Question 3: How long does it take for A to transfer a 500KB file to B using TCP? Consider *all* packets transmitted by TCP. Justify your answer. *(20')*

►**Exercise 173.** Consider the network of autonomous systems depicted below.



Question 1: Indicate (on the figure) which connections are eBGP and which ones are iBGP. *(5')*

Question 2: Assuming that every AS is willing to forward traffic of every other AS, write the forwarding table of routers r_1 and r_4 after the convergence of the protocol. You must also consider the supernetting feature of BGP. (Write the forwarding tables referring to the interface numbers given by the labels on the links of r_1 and r_4.) *(15')*

►**Exercise 174.** A sender A sends a file to a receiver B using a stop-and-wait transport protocol over a link with maximum segment size $MSS = 1$KB, transmission rate $R = 1000$KB/s, and delay $D = 100$ms. The sender detects errors with a fixed timeout $T = 1$s. Acknowledgment packets can be considered to have zero length.

Question 1: How long would it take to transmit a 300KB file in the best case? Justify your answer. *(5')*

Question 2: What is the expected transmission time for a 300KB file when each packet is dropped with probability $p = 0.01$ (i.e., one every 100 packets is dropped)? Justify your answer. *(15')*

►**Exercise 175.** Consider a peer-to-peer system in which host A holds a 100MB file and three other hosts B_1, B_2, and B_3 want to obtain that file. The access link of A has a maximum upload speed of $U_A = 200$KB/s while all other hosts have an access link with maximum upload and download speeds of $U_B = 60$KB/s and $D_B = 500$KB/s, respectively.

Question 1: Is it possible to transfer the file from A to all other hosts in 10 minutes or less? If so, explain how. If not, explain why not. *(5')*

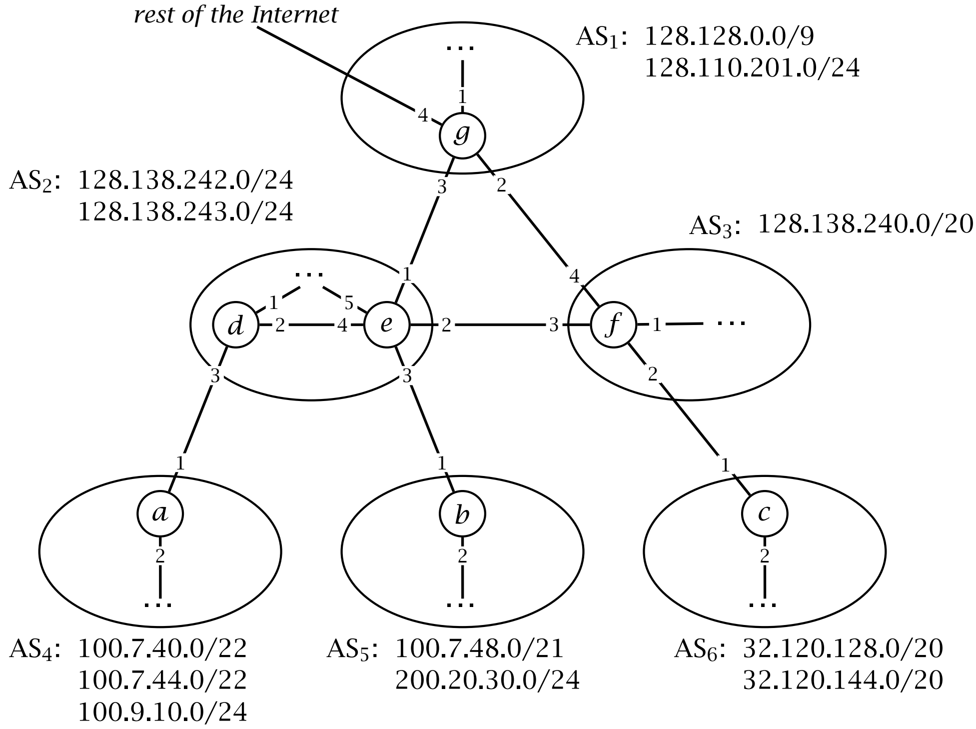
Question 2: In order to reduce the transfer time (from A to all other hosts) you may choose one of the following improvements: (1) double A's upload speed U_A, (2) double the other hosts' upload speed U_B, or (3) double their download speed D_B. Which one would you choose? In that case, what would be the best way to transfer the file, and how long would it take? *(15')*

►**Exercise 176.** A router g issues the *link-state advertisement* $\mathrm{LSA}_g = \{(a,5),(b,1),(d,2),(e,5)\}$ and receives the following other advertisements, where letters $(a,b,c,\ldots)$ represent router addresses.

$\mathrm{LSA}_a = \{(b,3),(c,1),(e,4),(f,1)\}$
$\mathrm{LSA}_b = \{(a,3),(c,6),(d,4),(g,1)\}$
$\mathrm{LSA}_c = \{(a,1),(b,6),(d,6)\}$
$\mathrm{LSA}_d = \{(b,4),(c,6),(g,2)\}$
$\mathrm{LSA}_e = \{(a,4),(f,2),(g,5)\}$
$\mathrm{LSA}_f = \{(a,1),(e,2)\}$

Based on these advertisements, write the forwarding tables of all routers. For destination addresses, use the symbolic labels $a,\ldots,g$; also, identify each interface by the label of the corresponding adjacent router. *(20')*

►**Exercise 177.** Consider the following network of autonomous systems each with the assigned prefixes. (These are the ranges of addresses held within each AS.)



Write the forwarding tables of all the seven routers, using the actual IP prefixes and the interface numbers annotated in the figure. Routing must be follow shortest paths, and addresses must be combined ("supernetting") whenever possible. Make sure all addresses are correctly reachable, including all addresses in the rest of the network. *(30')*

►**Exercise 178.** The network between hosts A and B has a maximum segment size MSS = 1KB, a total delay of D = 200ms, and an infinite transmission rate $R = \infty$. Consider the transmission of a 20KB file from A to B using TCP. Assume that A initiates the connection at time $t = 0$ and that the CPUs of A and B are infinitely fast such that the processing time is always zero.

Question 1: Assuming the network is perfectly reliable, write all the TCP packets exchanged by A and B. For each packet, write the time the packet is sent, the SYN and ACK flags, if present, and the sequence number and the acknowledgment number, if meaningful. For example, you should write "t = 123ms, $A \longrightarrow B$, ACK, seq = 2345, ack = 3456" for a packet sent at time t = 123ms from A to B carrying the ACK flag, the sequence number 2345 and the acknowledgment number 3456. *(15')*

Question 2: Now consider the transfer of a 200KB file, and in this case assume that the network looses the 100th packet sent by A. Exactly how long does it take for A to transmit the entire file? Justify your answer by showing a synthetic trace of the packets exchanged by A and B. In this case, do not write every single packet but instead write the initial time and the initial and final sequence number of every sequence of consecutive packets sent by A (e.g., "t = 123ms, $A \longrightarrow B$, 10pkts, seq = 1000...11000"). Show and briefly explain exactly what happens after the loss of the 100th packet. *(15')*

►**Exercise 179.** A 1GB file is held by 4 "seeder" peers in a peer-to-peer file sharing group (torrent). Imagine now that 10 more peers join the group to download that file. Assume that all peers are connected to the network through an access link with maximum download and upload rates of 500KB/s and 100KB/s, respectively, and that the core of the network has infinite bandwidth.

Question 1: How long does it take for an ideal peer-to-peer protocol to complete the file transfer? Justify your answer. *(10')*

Question 2: After the first file transfer is complete, how long does it take for an ideal peer-to-peer protocol to transfer the same file to 10 more peers, assuming the first 10 would also share the file? Justify your answer. *(10')*

►**Exercise 180.** A robotic probe is on Mars at a time when the distance from Earth to Mars is 300 million kilometers. The radio communication between Earth and Mars is on a frequency that allows for an error-free throughput of 1KB/s. Also, recall that the speed of light, which is the propagation speed of radio waves, is 300'000 kilometers per second.

Question 1: How long does it take to directly transmit a 1MB image from the probe to Earth? Justify your answer. *(5')*

Question 2: How long does it take to download five images of 1MB each using HTTP with and without pipelining? Justify your answer. *(10')*

Question 3: How long does it take to upload a 2Kb text file from Earth to the probe using SMTP? Assume the client is on Earth and the probe runs an SMTP server. Justify your answer. *(15')*

►**Exercise 181.** A university e-mail server supports a mailing list *cn@inf.usi.ch* for the computer networking class. Suppose that the mailing list includes *joe@email.ch*, *jane@gmail.com*, *mario@email.ch*, and *luigi@gmail.com*. Suppose now that *antonio@usi.ch* sends a message to the computer networking mailing list. Describe in detail all the network communications between the university server and the rest of the network. In particular, describe all SMTP sessions and all DNS messages. *(20')*

►**Exercise 182.** A client at address C downloads a 10Kb image from a server at address S using HTTP over a TCP connection requesting that the connection be immediately closed. Assume a maximum segment size of 1500 bytes. Write all the TCP segments exchanged between the client and the server. For each segment, write the source and destination addresses, source and destination ports, sequence number, ack number, relevant flags, and also a summary of the content. *(20')*

►**Exercise 183.** A company would like to distribute a 600MB file on-line. The file is downloaded many times per day by many different users. The company initially makes the file available through an HTTP server connected to the network through a network link with an upload rate of 5MB/s. All users have links with an upload rate of 100KB/s and a download rate of 3.14MB/s. Assume that the core of the network has infinite bandwidth.

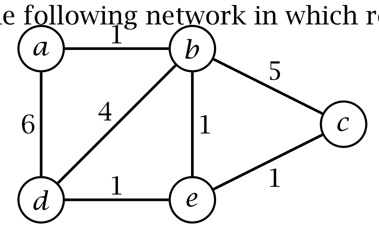
Question 1: On average, how many users per hour can download the whole file? Justify your answer. *(5')*

Question 2: Suppose now that the company decides to support more users with a peer-to-peer protocol in which users that completed their download are encouraged to make the file available to other users. Assuming that on average a user makes their copy available for 10 minutes, ideally, how many more users per hour can download the file? Justify your answer. *(15')*

Question 3: Can the peer-to-peer protocol support an unlimited number of users? If so, how? *(10')*

►**Exercise 184.** A client host A opens a TCP connection with a server host B, sends 1MB of data as fast as possible, and then closes the connection. The path between A and B has a total (one way) delay $D = 100\text{ms}$, a maximum throughput rate $R = 100\text{KB/s}$, and a maximal segment size $MSS = 1\text{KB}$. How long does it take to complete the transmission? Assume there are no packet losses when the transmission rate is below R. Justify your answer. *(20')*

►**Exercise 185.** Consider the following network in which routers use distance-vector routing.



Question 1: Show the forwarding tables of routers a and b from the initialization until the state of the tables converges. Assume that all routers initialize at the same time, and that, at each round of the algorithm, each router receives distance vectors from all its neighbors simultaneously. *(15')*

Question 2: After convergence, the cost of the link between b and e raises from 1 to 10? Again, show the evolution of the forwarding tables of routers a and b. *(15')*

►**Exercise 186.** Consider using the Go-Back-N protocol over an unreliable link with transmission delay $D = 100\text{ms}$, transmission rate $R = 200\text{KB/s}$, maximum segment size $MSS = 500\text{B}$, and per-packet error probability $p = 0.01$. Define a good timeout T and a good window size W for the protocol, and based on those values, compute the expected total transmission time for a file of size $S = 600\text{MB}$. *(20')*

►**Exercise 187.** For each of the following ranges of IPv4 addresses, write an address prefix that defines the range exactly. If it is not possible to express the range exactly with a prefix, then write "N.E." (meaning "not exact") followed by a *minimal* approximate prefix, meaning a prefix that defines a minimal (smallest possible) range that contains the given range. *(20')*

range	*subnet prefix-address/prefix-length*
34.112.126.0–34.112.127.255	
89.54.131.160–89.54.131.175	
128.138.0.0–128.138.2.255	
191.203.111.128–191.203.111.192	
241.37.144.0–241.37.151.255	
62.252.0.128–62.252.1.128	
127.0.0.0–127.255.255.255	
59.127.0.0–59.128.255.255	
0.0.0.0–255.255.255.255	
179.240.0.0–179.243.255.255	

►**Exercise 188.** A router has 10 input ports and 10 output ports. All input and output ports have the same maximum throughput of 1GB/s.

Question 1: Is it possible to design the router so that packets are never queued? If so, explain why and in particular specify the throughput of the switch fabric? If not, explain why not. *(5')*

Question 2: Suppose that the switch fabric has a maximum throughput of 5GB/s, and that the input traffic is such that half of it goes to output ports 1 and 2, where it is distributed evenly, and the rest goes to the other output ports, also evenly distributed. What is the expected output of each output port? Does the router drop packets? If so, specify where and for each point specify the loss rate. If not, explain why not. *(5')*

►**Exercise 189.** Consider a router with four interfaces that uses longest-prefix matching in a datagram network using 8-bit host addresses.

Question 1: Given the following forwarding table, compute how many addresses would be routed through each interface. *(10')*

prefix	*port*
0/0	1
64/2	2
128/3	2
240/4	3
16/4	4
104/5	4
8/5	3

Question 2: Assuming now that the router connects subnets *A*, *B*, *C*, and *D*, and that *A* and *B* must each support at least 80 interfaces (i.e., addresses), and *C* and *D* must each support at least 40 interfaces. Assign the necessary addresses to each network and write the corresponding forwarding table for the router. *(10')*

►**Exercise 190.** A group of 30 users download a 3GB file through a peer-to-peer system. The whole file is available from 3 other "seed" users S_1, S_2, S_3 connected to the network with access links of maximal upload rates U_1 = 400KB/s and U_2 = 700KB/s, and U_3 = 1MB/s respectively. Of the 30 users, 10 have a fast access link and 20 have a slow access link. A fast access link has maximal upload and download rates of U_{fast} = 200KB/s and D_{fast} = 1MB/s, respectively. A slow access links has maximal upload and download rates of U_{slow} = 100KB/s and D_{slow} = 400KB/s, respectively. What is the theoretical best total download time? This is the time it takes for all thirty users to obtain the whole file. Justify your answer by showing and briefly explaining your calculations. (For simplicity, let 1GB and 1Kb represent 10^9 and 10^3 bytes, respectively.) *(20')*

►**Exercise 191.** An HTTP connection is established over a TCP connection with maximum segment size of 1400 bytes.

Question 1: Within this connection, a TCP segment with sequence number 2300 carries the following HTTP request:

```
HTTP/1.0 404 Not Found
Content-Type:  text/html
Content-Length:  41

<html><body>Page Not Found!</body></html>
```

What is the sequence number of the next segment going from the Web server to the client? Justify your answer. (Remember that all HTTP header lines end with a CRLF sequence.) *(10')*

Question 2: Within the same connection, a TCP segment with sequence number 6500 carries the following HTTP request:

```
HTTP/1.0 200 Ok
Content-Type:  image/jpg
Content-Length:  25000

a binary image...
```

What is the sequence number of the next segment sent by the server in this case? Justify your answer. *(10')*

►**Exercise 192.** A router has 16 input interfaces capable of receiving up to 10000 packets per second, and 16 output interfaces capable of sending 20000 packets per second.

Question 1: Let T_S be the maximum throughput of the switch fabric. What is the maximum total throughput of the router in the best case? Are there values of T_S for which the total throughput would not depend on T_S? Justify your answers. *(10')*

Question 2: Consider the previous question in the case of a specific distribution of traffic. Let T_S be the maximum throughput of the switch fabric, and suppose that 20% of the input traffic goes to output interface 1, 10% to output interface 2, and the rest is distributed uniformly onto all other interfaces. What is the maximum throughput of the router in this case? Also, for what values of T_S (if any) does the router drop packets on its input and output queues, respectively? Are there any values of T_S for which neither queues would ever be full? Justify your answers. *(10')*

►**Exercise 193.** Consider a reliable connection between a sender and a receiver implemented with a Go-Back-N protocol with maximum segment size MSS = 2KB and with a fixed window size W = 8. The network latency between the sender and receiver is L = 200ms, the access link of the sender allows a maximum transmission throughput T_S = 100KB/s, and both sender and receiver have very high reception throughput.

Question 1: How long does it take to reliably transmit a file of size S = 100KB in the best case, without any loss of packets? Justify your answer. *(10')*

Question 2: Assuming the timeout is set to 1s, how long does it take to reliably transmit the same file (S = 100KB) in the presence of losses when one out of 20 packets gets lost in both directions? Justify your answer. *(10')*

►**Exercise 194.** Consider IP forwarding with longest-prefix matching.

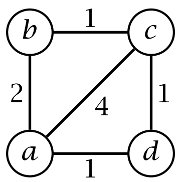
Question 1: Briefly describe and motivate longest-prefix matching with an example network. How does it work, and why it useful? *(10')*

Question 2: A router in an IPv4 network using longest-prefix matching has the forwarding table shown below on the left. For each destination addresses in the table on the right, write the output port and the list all the matching table entries. *(10')*

entry	*destination*	*port*
1	98.7.1.0/16	1
2	211.57.20.0/24	1
3	40.120.0.0/16	2
4	211.57.21.0/24	2
5	160.0.0.0/2	3
6	111.11.0.0/16	3
7	211.57.20.0/22	4
8	211.57.0.0/16	4
9	0.0.0.0/2	4
10	0.0.0.0/0	5

address	*port*	*matching entries*
211.57.1.69		
10.142.226.44		
98.7.2.71		
200.100.2.1		
40.120.207.167		
211.57.20.11		
211.57.21.10		

►**Exercise 195.** Consider the following simple network topology where routers use a distance-vector routing protocols



For simplicity, assume all routers start at time 0, that routing messages (i.e., distance vectors) are sent out by routers every 10 seconds, and that they are received by neighbor routers after one second. Write the first iterations of the distance-vector routing algorithm, at times 0, 10, ..., until the protocol converges to a stable state. For each iteration, list the routing tables of each router. *(20')*

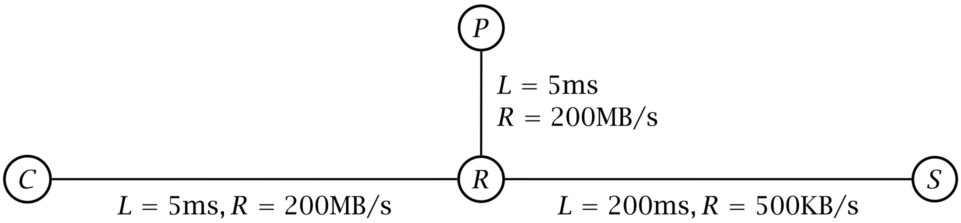
►**Exercise 196.** An HTTP server issues the following replies.

```
HTTP/1.1 304 Not Modified
Date:  Wed, 13 Nov 2013 10:30:00 GMT
ETag:  "978140-db4-4e3bb1239a5c0"
```

```
HTTP/1.1 200 OK
Date:  Wed, 13 Nov 2013 10:31:00 GMT
Last-Modified:  Mon, 12 Aug 2013 07:28:31 GMT
ETag:  "978140-db4-4e3bb1239a5c0"
Content-Length:  3508
Content-Type:  text/html; charset=UTF-8
Connection:  close
```

Write two plausible requests that could have generated such replies. Notice that the body of both replies is empty. *(10')*

►**Exercise 197.** Consider an HTTP client C, an HTTP caching proxy P, and an HTTP server S connected through a router R in the network depicted below, where each link is characterized by the given latency L and transmission rate R in each direction. Assume the router itself has no bandwidth limitations and introduces no delay.

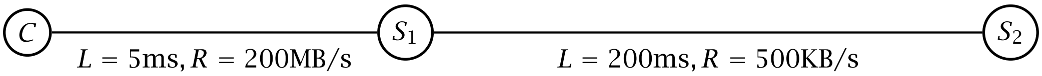


Client C accesses the Web through proxy P and issues a sequence of HTTP requests for objects whose origin server is S.

Question 1: How long does it take for client C to obtain two images *a.jpg* and *b.jpg* of 200KB and 100KB, respectively, assuming that neither is cached by the proxy. Justify your answer. *(10')*

Question 2: How long does it take for client C to obtain three images *a.jpg*, *b.jpg*, and *c.jpg* of 200KB, 100KB, and 200KB, respectively, assuming that valid copies of *a.jpg* and *b.jpg* are cached by the proxy while *c.jpg* must be fetched from the origin server. Justify your answer. *(10')*

►**Exercise 198.** Consider the following network between an e-mail client C, an SMTP server S_1 serving domain *usi.ch*, and another SMTP server S_2 serving and handling all the e-mail for domain *democracynow.org*.



The sender using client C composes a message of 250KB (including all headers and attachments) addressed to *amy@democracynow.org* and *juan@democracynow.org*. How long would it take in the best case to deliver the message to all the intended receivers from the time the sender hits the "send" button? Justify your answer. *(20')*

►**Exercise 199.** Consider a sender A and a receiver B connected by a link with transmission rate R = 1MB/s and latency L = 100ms. Sender and receiver use the *go-back-N* protocol with a segment size of 1KB and a window of W segments. Knowing that in the best case (without errors) it takes 4.02s for sender A to transfer a file of size S = 1MB to B, what is the value of W used by A? Could A improve the transfer time by choosing another window size? If so, what would be an optimal value for W and the resulting transfer time? Justify your answers. *(20')*

►**Exercise 200.** A file of size 300MB is published using bittorrent and immediately 10 peers join the torrent to download the file. The publisher has an upload rate of 500KB/s and the other peers contribute on average an upload rate of 350KB/s. Assume that the download rate of all the peers is high enough to be irrelevant.

Question 1: How long would it take in the best case to transfer the file to all peers? Justify your answer. *(10')*

Question 2: Now assume that exactly 5 minutes after the publication, another 10 peers join the torrent. These new peers share at an average upload rate of 100KB/s and have plenty of download bandwidth. In the best case, when would all the peers complete their download? Justify your answer. *(20')*

►**Exercise 201.** An application A transmits a continuous stream of data to another application B using a reliable transport layer on top of a link with latency L = 40ms, transmission rate R = 1MB/s, maximum segment size S = 1KB, and per-packet loss probability P_e.

Question 1: What is the effective transmission rate (as seen by the applications) when the transport layer uses a *stop-and-wait* protocol with a timeout T = 100ms? Compute the effective transmission rate for error probabilities $P_e = 1/10$, $P_e = 1/50$, $P_e = 1/100$, and $P_e = 0$, that is, if a packet is lost on average every 10, 50, 100 packets, or never. *(10')*

Question 2: What is the effective transmission rate when the transport layer uses a *go-back-N* protocol with a timeout T = 100ms and window size W = 100? Again compute the effective transmission rate for error probabilities $P_e = 1/10$, $P_e = 1/50$, $P_e = 1/100$, and $P_e = 0$, that is, if a packet is lost on average every 10, 50, 100 packets, or never. *(10')*

►**Exercise 202.** An application A transfers a file to another application B using a TCP connection over a communication link that has a maximum throughput of 1MB/s and no additional packet losses. In other words, the link drops a packet *only* when that exceeds the maximum throughput over a certain short period. How long does it take for B to receive a 1.5GB file? Justify your answer by explaining the behavior of TCP at steady state. (**Hint:** ignore the initial and final phases of the TCP connection. Consider the overall behavior of TCP, not the individual packets.) *(10')*

►**Exercise 203.** A Web client connects to the Web server on host *example.com* to verify that the URL *http://example.com/image.jpg* is valid, but without transferring the object.

Question 1: Write the full HTTP exchange between client and server. *(5')*

Question 2: Write all the TCP packets of the connection between client and server. For each packet, specify all the important information in the IP and TCP headers (addresses, ports, flags, sequence numbers, etc.) as well as the content. *(15')*

►**Exercise 204.** Consider the convergence of distance-vector routing under the assumption that all routers execute the protocol in synchronous steps.

Question 1: Write a network of at least five nodes in which the routing protocol would converge to a stable state after exactly four steps. Also, write the routing tables at each step for one of the nodes whose tables converge last (at the fourth step). *(10')*

Question 2: After convergence, change the cost of one link so that the routing protocol would again converge to a stable state after two or more steps. Also, for each step after the change, write the routing tables of one of the nodes whose tables are affected by the change. *(10')*

►**Exercise 205.** Consider a router with four input and four output ports. The output ports have a maximum transmission rate of 5000 packets per second. The switch fabric has a maximum throughput of 10000 packets per second. All packet queues in the router can contain up to 1000 packets. Suppose the input traffic starts to ramp up linearly and uniformly on all input ports, from 0 packets per second at time $t = 0$ to the maximum input rate of 5000 packets per second for each input port at time $t = 10$s. The input traffic is such that it spreads uniformly over the first three output ports, so one third of the traffic goes to ports 1–3 each, and no traffic goes to port 4.

Question 1: At what time and for which queues does the router start to drop packets? Would the situation change if the switch fabric were twice as fast? Justify your answers. *(10')*

Question 2: Suppose the traffic goes back to zero in the same way it ramps up, so starting at time $t = 10$s it declines uniformly and linearly reaching zero for all input ports at time $t = 20$s. When does the router stop dropping packets? How many packets does the router drop in total between $t = 0$ and $t = 20$s? Justify your answers. *(10')*

►**Exercise 206.** A university owns the IPv4 addresses represented by prefixes 195.176.180.0/22 and 128.138.240.0/20.

Question 1: How many addresses does the university own in total? Justify your answer. *(5')*

Question 2: The university has one campus connected to the internet through ISP A and now opens a new campus in a different city connected through ISP B. The university assigns 320 of its IP addresses to the new campus. Write a plausible network topology that shows the two campuses and their two access ISPs, and for each ISP write the entries in the forwarding tables that are relevant to the addresses owned by the university. *(15')*

►**Exercise 207.** Describe the “slow start” in TCP. How does it work and what is its purpose? *(10')*

►**Exercise 208.** A router x issues the following link-state advertisement $\text{LSA}_x = \{(d,1),(e,2),(b,4)\}$ and receives the following other advertisements, where letters represent router addresses.

$\text{LSA}_g = \{(d,1),(h,1)\}$
$\text{LSA}_h = \{(e,2),(f,4),(j,14),(g,1)\}$
$\text{LSA}_d = \{(g,1),(e,3),(x,1)\}$
$\text{LSA}_e = \{(d,3),(x,2),(b,1),(h,2)\}$
$\text{LSA}_f = \{(h,4),(j,2),(b,2),(c,1)\}$
$\text{LSA}_b = \{(x,4),(e,1),(f,2),(c,4)\}$
$\text{LSA}_c = \{(b,4),(f,1)\}$

Write the forwarding table of router x. Justify your answer by explaining how link-state routing works. *(20')*

►**Exercise 209.** A person (*bill@somewhere.net*) sends an ordinary e-mail message to *joe@coolplace.org* through a local SMTP server. Describe every network operation, at the application level, performed by your SMTP server to deliver the message. *(20')*

►**Exercise 210.** An application downloads 10 objects (HTML, images, and other content objects) of 50KB each using HTTP over an access link with an effective transmission rate R = 500KB/s. Assume that the rest of the network has unlimited bandwidth and ignore the latency introduced by DNS and the initial latency and dynamics of TCP. The transmission delay with the origin server is L = 200ms. Write how long it would take for the downloads to complete in each of the following cases. Justify your answers.

Question 1: The web server does not support persistent connections. *(5')*

Question 2: The web server supports persistent connections without pipelining. *(5')*

Question 3: The web server supports persistent connections with pipelining. *(5')*

Question 4: The web server supports persistent connection with pipelining, and the last 5 objects are in a transparent caching proxy close to the client, with a transmission delay L = 10ms. *(5')*

►**Exercise 211.** Two computers are connected through a link with transmission rate R = 1000KB/s and latency L = 100ms.

Question 1: What is the effective throughput with a stop-and-wait transport protocol with maximum segment size MSS = 1KB? Briefly justify your answer. *(5')*

Question 2: What is the optimal window size in KB? Briefly justify your answer. *(5')*

Question 3: What is the effective throughput in a transmission using TCP over that link? Briefly justify your answer. *(5')*

Question 4: With a maximum segment size MSS = 1KB, how long does it take for TCP to achieve its maximum throughput? Briefly justify your answer. *(5')*

►**Exercise 212.** Answer the following questions regarding IP prefixes and longest-prefix matching. Briefly justify your answers.

Question 1: Is it possible to have a prefix that represents exactly 100 IP addresses? *(5')*

Question 2: What is the best way to represent 100 IP addresses using IP prefixes. Write a minimal set of prefixes that represents exactly 100 IP addresses. *(5')*

Question 3: A small network provider A owns 100 IP addresses that it sells to two customers B and C on two separate subnets. Write the forwarding tables of A's IP router. *(10')*

►**Exercise 213.** Consider three applications, A, B, and C, each connected to the network through an independent access link with maximum download rate R_d = 1000KB/s and maximum upload rate R_u = 200KB/s. What is the fastest way to send 1GB of data from A to both B and C? Describe a transfer scheme. Detail who transfers what to whom, and when. Also, assuming no errors and perfect coordination between A, B, and C, compute the total transfer time, meaning the time it takes for both B and C to download the 1GB file? Justify your answer. *(20')*

►**Exercise 214.** How and why does TCP estimate the network-level round-trip time for its connection? How does TCP use the estimated round-trip time? Describe and explain the estimation algorithm using an example. Also, discuss the goal of this algorithm, showing again by example what would happen if the round-trip time is underestimated ors overestimated. *(20')*

►**Exercise 215.** Compare the HTTP and BitTorrent protocols from the perspective of a single user downloading a large file.

Question 1: Describe a case in which the user obtains a faster download time with HTTP. Precisely specify all the quantities that characterize the case (file size, rates for all parties involved, etc.). *(10')*

Question 2: Describe a case in which the user obtains a faster download time with BitTorrent. Precisely specify all the quantities that characterize the case (file size, rates for all parties involved, etc.). *(10')*

►**Exercise 216.** A web browser downloads a web page consisting of four objects: an HTML document *index.html* of 50KB, plus three image referenced in *index.html*, *A.jpg* of 100KB, *B.jpg* of 200KB, and *C.jpg* of 100KB. All objects are from the same origin server. The propagation delay (latency) and transmission rate between the browser and the server are $L = 150$ms and $R = 500$KB/s, respectively. In each of the following cases, compute the total time to download the whole page. Assume that there is no content caching. Assume that, when multiple parallel connections, each connection uses a perfectly fair share of the available transmission rate. Justify your answers.

Question 1: The browser opens one connection with the server and does not use pipelining. *(5')*

Question 2: The browser opens one connection with the server and uses pipelining. *(5')*

Question 3: The browser opens up to three connections with the server and uses pipelining. *(10')*

►**Exercise 217.** Consider an HTTP request and reply over a TCP connection between a client and the server *www.example.com*.

Question 1: Write an example of an HTTP request and the corresponding reply. Be as clear and precise as possible. You may omit the body of the request. *(5')*

Question 2: Write the important TCP headers of every packet of the connection defined in exercise 1. At a minimum, for each packet write the port numbers, and the sequence and acknowledgment numbers. *(15')*

►**Exercise 218.** A sender transfers a file to a receiver using the Go-Back-N protocol with a window size of $W = 200$ segments, a maximum segment size MSS = 1KB, and a timeout $T = 2$s, over a network connection with latency $L = 400$ms and transmission rate $R = 800$KB/s.

Question 1: How long does it take to transfer a large file of size S in the absence of errors? Justify your answer. *(5')*

Question 2: How long does it take to transfer a large file of size S when the network drops packets with probability $p = 10^{-3}$? Justify your answer. *(15')*

►**Exercise 219.** Answer the following questions concisely but clearly.

Question 1: What is the role of the domain name system in the exchange of electronic mail? *(5')*

Question 2: Is e-mail (SMTP) secure against forgery? If so, briefly describe the mechanisms that make it secure. If not, briefly describe a potential forgery attack. *(5')*

Question 3: Is the Web secure against forgery? A forgery in this context means that a user goes to a site (e.g., *www.inf.usi.ch*) but instead gets a page forged by the attacker. In particular, consider the case of an attacker on the same local-area network as the user. If you consider the Web secure, then briefly describe the mechanisms that make it secure. Otherwise briefly describe a potential forgery attack. *(10')*

►**Exercise 220.** Briefly explain the basics of caching in HTTP. Describe a case in which caching provides faster access to an object. Clearly specify each HTTP request and reply, as well as all the quantities that characterize the case. *(20')*

►**Exercise 221.** Is it possible to design a router with 10 input and output ports (10 of each) that does not require any queuing? If so, describe the router architecture and give a specification of the throughput of its input ports, output ports, and switch fabric. If not, explain why not. *(10')*

►**Exercise 222.** A space probe approaches a comet when the comet is 150 million kilometers away from Earth. The probe sends data to Earth using a radio signal and a corresponding encoding that allows for a maximum transmission rate of 50KB/s. Assume that the link-level communication uses an encoding that guarantees error-free communication. Recall that radio signals propagate at about 300'000 kilometers per second.

Question 1: How long would it take to transmit a data file of 100KB using the link-level transmission directly? Justify your answer. *(5')*

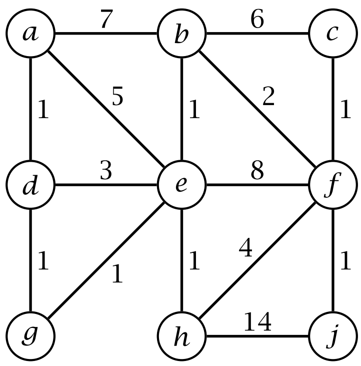
Question 2: How long would it take to transmit a data file of 100KB by opening a TCP connection, sending the whole file, and then closing the connection. The maximum segment size is 10KB. Justify your answer. *(15')*

▶**Exercise 223.** Describe all the network operations involved in delivering an e-mail message, from the time the sender composes the message to the time the receiver reads it.

Question 1: Briefly describe the components of the e-mail systems of the sender and receiver and those of the network in general that have any meaningful role in the e-mail communication. For each interaction between components, specify the protocol and the essential elements of the interaction. You may use a diagram to explain the interactions and roles of the components. *(10')*

Question 2: Detail at least one interaction, writing the sequence of messages or datagrams exchanged in the interaction. What is essential here is that you identify the important information exchanged in the interaction rather than the exact bytes exchanged. *(10')*

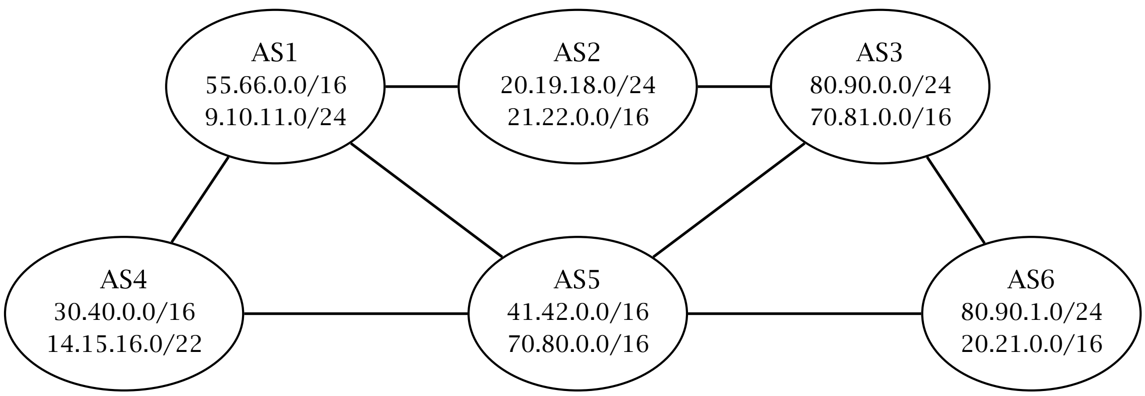
▶**Exercise 224.** Consider the behavior of a synchronous distance-vector routing protocol on the following network. The protocol is synchronous in the sense that all routers exchange their distance vectors at the same regular times $t_0, t_1, t_2, \ldots$ and therefore the protocol proceeds in steps $i = 0, 1, 2, \ldots$ for the entire network.



Question 1: Write the distance vector held by router a at times $t_0, t_1, t_2, \ldots$, from the initialization step at t_0 until the distance vector stabilizes completely. (Hint: you do not have to simulate the distance-vector protocol on every router. There is a simpler way to figure out the distance vector at a router.) *(20')*

Question 2: At time t_{100}, well after the distance-vector protocol has stabilized, the cost of the link between h and j changes from 14 to 1. Write the distance vector held by router a from time t_{100} for each step until the distance vector stabilizes again. *(10')*

▶**Exercise 225.** Consider the following AS-level network topology with the given network prefixes assigned to each AS.



Question 1: Write the BGP advertisements issued by every AS once BGP has converged to a stable state, considering also prefix aggregation, and assuming that all ASes always choose the shortest AS path to a destination. *(10')*

Question 2: Write the BGP advertisements issued by every AS once BGP has converged to a stable state, considering prefix aggregation, and assuming that AS1 and AS3 adopt a routing policy never to send traffic through AS5 (except for traffic destined to AS5) while all other ASes always choose the shortest AS path to a destination. *(10')*

►**Exercise 226.** Briefly describe the longest-prefix matching algorithm used for IP forwarding. Also, show an example network, complete with subnet address assignments, in which longest-prefix matching is essential. *(10')*

►**Exercise 227.** Decrypt the following ciphertext encrypted using a shift cipher. Notice that space that separates words is treated like another letter.

vbjn oalngw sfn hfgngvsnqoaabanpozzfntzlnpstbesngvslnoesntbesisenpoaasr

(10')

►**Exercise 228.** A router has 10 input ports and 10 output ports. All input and output ports have the same maximum throughput of 2GB/s.

Question 1: Is it possible to design the router so that packets are never queued? If so, explain why and in particular specify the throughput of the switch fabric? If not, explain why not. *(10')*

Question 2: Suppose that the switch fabric has a maximum throughput of 10GB/s, and that the input traffic is such that half of it goes to output ports 1 and 2, where it is distributed evenly, and the rest goes to the other output ports, also evenly distributed. What is the expected output of each output port? Does the router drop packets? If so, specify where and for each point specify the loss rate. If not, explain why not. *(10')*

►**Exercise 229.** A sender A sends a file to a receiver B using a stop-and-wait transport protocol over a link with maximum segment size MSS = 1KB, transmission rate R = 1000KB/s, and delay D = 100ms. The sender detects errors with a fixed timeout T = 1s. Acknowledgment packets can be considered to have zero length.

Question 1: How long would it take to transmit a 300KB file in the best case? Justify your answer. *(5')*

Question 2: What is the expected transmission time for a 300KB file when each packet is dropped with probability $p = 0.01$ (i.e., one every 100 packets is dropped)? Justify your answer. *(15')*

►**Exercise 230.** Consider a peer-to-peer system in which host A holds a 100MB file and three other hosts B_1, B_2, and B_3 want to obtain that file. The access link of A has a maximum upload speed of U_A = 200KB/s while all other hosts have an access link with maximum upload and download speeds of U_B = 60KB/s and D_B = 500KB/s, respectively.

Question 1: Is it possible to transfer the file from A to all other hosts in 10 minutes or less? If so, explain how. If not, explain why not. *(5')*

Question 2: In order to reduce the transfer time (from A to all other hosts) you may choose one of the following improvements: (1) double A's upload speed U_A, (2) double the other hosts' upload speed U_B, or (3) double their download speed D_B. Which one would you choose? In that case, what would be the best way to transfer the file, and how long would it take? *(15')*

►**Exercise 231.** A router g issues *link-state advertisement* $\text{LSA}_g = \{(a,5),(b,1),(d,2),(e,5)\}$ and receives the following other advertisements, where letters $(a,b,c,\ldots)$ represent router addresses.

$$\begin{aligned}
\text{LSA}_a &= \{(b,3),(c,1),(e,4),(f,1)\}\\
\text{LSA}_b &= \{(a,3),(c,6),(d,4),(g,1)\}\\
\text{LSA}_c &= \{(a,1),(b,6),(d,6)\}\\
\text{LSA}_d &= \{(b,4),(c,6),(g,2)\}\\
\text{LSA}_e &= \{(a,4),(f,2),(g,5)\}\\
\text{LSA}_f &= \{(a,1),(e,2)\}
\end{aligned}$$

Based on these advertisements, write the forwarding tables of all routers. For destination addresses, use the symbolic labels $a,\ldots,g$; also, identify each interface by the label of the corresponding adjacent router. *(20')*

►**Exercise 232.** A university e-mail system serves a mailing list *cn@inf.usi.ch* for the computer networking class. The mailing list includes *joe@email.ch*, *jane@gmail.com*, *mario@email.ch*, and *luigi@gmail.com*. Suppose now that *antonio@usi.ch* sends a message to the computer networking mailing list. Describe in detail all the network communications between the university server and the rest of the network. In particular, describe all SMTP sessions and all DNS messages. *(20')*

►**Exercise 233.** The network between hosts A and B has a maximum segment size MSS = 1KB, a total delay of D = 200ms, and an infinite transmission rate $R = \infty$. Consider the transmission of a 20KB file from A to B using TCP. Assume that A initiates the connection at time $t = 0$ and that the CPUs of A and B are infinitely fast such that the processing time is always zero.

Question 1: Assuming the network is perfectly reliable, write all the TCP packets exchanged by A and B. For each packet, write the time the packet is sent, the SYN and ACK flags, if present, and the sequence number and the acknowledgment number, if meaningful. For example, you should write "t = 123ms, $A \longrightarrow B$, ACK, seq = 2345, ack = 3456" for a packet sent at time t = 123ms from A to B carrying the ACK flag, the sequence number 2345 and the acknowledgment number 3456. *(10')*

Question 2: Now consider the transfer of a 200KB file, and in this case assume that the network looses the 100th packet sent by A. Exactly how long does it take for A to transmit the entire file? Justify your answer by showing a synthetic trace of the packets exchanged by A and B. In this case, do not write every single packet but instead write the initial time and the initial and final sequence number of every sequence of consecutive packets sent by A (e.g., "t = 123ms, $A \longrightarrow B$, 10pkts, seq = 1000...11000"). Show and briefly explain exactly what happens after the loss of the 100th packet. *(10')*

►**Exercise 234.** Write an HTTP request and a plausible reply, abbreviating the body of the reply as needed. Briefly describe the headers and each part of the request and the reply. *(10')*

►**Exercise 235.** Consider a reliable stream implemented with the Go-Back-N protocol with a window of $W = 20$ packets with maximum segment size $MSS = 1000B$, with a timeout of $T = 1s$.

Question 1: Compute the time needed to transfer a file of $S = 1800KB$ over a link with propagation delay $d = 300ms$ and transmission rate $R = 50KB/s$. ($1KB = 1000B$.) *(10')*

Question 2: Compute also the expected transfer time when the link drops packets with probability $p = 0.005$, which means that on average one every 200 packets is dropped. *(20')*

►**Exercise 236.** Answer each of these questions.

Question 1: Is it true that the Internet is better than a virtual-circuit network because it can guarantee better quality of service, for example for voice over IP? Justify your answer. *(5')*

Question 2: Assuming your computer has one IP address, is it true that there can be at most 65536 applications simultaneously accepting TCP connections on your computer? Justify your answer. *(5')*

Question 3: An attacker compromises and disables your local DNS server. Describe what happens when you try to access your web-mail system? Would your web browser return a 404 error? Would the web-mail system stop working? Would other people from other universities still be able to send you e-mail? Justify your answers. *(10')*

►**Exercise 237.** A client gets four objects from a server using HTTP over a link with propagation delay $d = 50ms$ and transmission rate $R = 1MB/s$. The sizes of the objects are $S_1 = 5KB$, $S_2 = 100KB$, $S_3 = S_4 = 2MB$.

Question 1: How long does it take for the client to obtain all four objects (a) with sequential, non-persistent connections, (b) with one persistent connection with pipelining, and (c) with four parallel connections. Assume that at any given time, n parallel connections get each $1/n$ of the link bandwidth, that is, the rate for each of the n connections is R/n. Also, ignore the time needed to open and close connections. *(15')*

Question 2: Consider now the case in which the HTTP requests go through a proxy close to the client, such that the propagation delay and rate between client and proxy are $d_p = 5ms$ and $R = 20MB/s$, respectively. The total delay and the transmission rate between the client and the server through the proxy remain $d = 50ms$ and $R_p = 1MB/s$, respectively. In this case, object 3 is available at the proxy. So, does the total download time differ with (a) with sequential, non-persistent connections, (b) one persistent connection and pipelining, or (c) four parallel connections? Describe the case with the minimal total download time. Notice that in this case multiple connections may not share the fast local connection evenly, since some of them might be limited by the bottle-neck remote links. *(15')*

►**Exercise 238.** Two peer nodes hold a file of size $S = 600MB$ in a peer-to-peer system when 10 more peers join the peer-to-peer system to download the file. All peers have the same maximum upload rate $R_u = 500KB/s$ and the same maximum download rate $R_d = 2MB/s$.

Question 1: What is the minimum time it would take for all the 10 new peers to download the file? Justify your answer. *(10')*

Question 2: After the 10 new peers have completed their download, another 8 peers join the peer-to-peer system to download the file. What is the total download time for them? How many other peers need to have the file for the download time to stay at a constant minimum as more peers join one at a time? Justify your answer. *(10')*

►**Exercise 239.** Write an exchange between a mail user agent and the SMTP server of the *usi.ch* domain in which the user agent sends an e-mail message to *bob@usi.ch* and *charlie@usi.ch*. *(10')*

►**Exercise 240.** Consider the process by which a local DNS server resolves a DNS query for IPv4 address of *www.inf.usi.ch*. Describe this process and in particular describe every DNS datagram that the server transmits or receives, specifying source, destination, and the relevant DNS data for each datagram. Assume that none of the name components are in the DNS cache. Also assume that no server would perform a recursive query. *(20')*

►**Exercise 241.** Consider a router with n input/output ports with transmission rate R (i.e., n interfaces, each with a full-duplex link with transmission and reception rates $R_{TX} = R_{RX} = R$). The router also has a switch fabric with maximum throughput T_{SWITCH}.

Question 1: Does the router need output queues? Does the answer depend on R, T, and n? How so? If there is a case in which output queues are needed, describe a scenario in which those queues would fill up (and therefore the router would have to drop packets). Describe this scenario by specifying the length Q of those queues, the concrete values of R, T, and n, and the maximum time t the router could cope with traffic without dropping packets. *(10')*

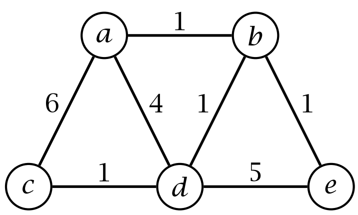
Question 2: Does the router need any other queue? Does the answer depend on R, T, and n? How so? If there is a case in which other queues are needed, describe a scenario in which those queues would fill up (and the router would have to drop packets). Describe this scenario by specifying the length Q of those queues, the concrete values of R, T, and n, and the maximum time t the router could cope with traffic without dropping packets. *(10')*

►**Exercise 242.** Answer the following questions about congestion control in TCP.

Question 1: What is the mechanism by which TCP controls the output rate of a sender? Describe a concrete example in which the sending rate is $R_{send} = 1MB/s$. In describing the example, specify concrete values for every relevant network parameter. *(10')*

Question 2: What is the effective transmission rate R_{TCP} of a TCP connection over a link with maximum transmission rate R_{link} in the absence of errors? Absence of errors means that packets are lost only when they are sent at a rate that exceeds the maximum transmission rate R_{link}. In this case, the link randomly drops some packets so that the effective rate remains under R_{link}. Explain your answer by describing the behavior of TCP in the given case. *(10')*

►**Exercise 243.** Consider the following network where routers use a distance-vector routing protocol.



Write the first iterations of the distance-vector routing algorithm until the protocol converges to a stable state. For each iteration, list the routing tables of each router. *(20')*

Extra page for Exercise 243

▸**Exercise 244.** Write a minimal set of prefixes that represent exactly the same IPv4 addresses represented by the following prefixes. Justify your answer by showing the relations between the given prefixes and the prefixes in your solution.

127.0.0.0/8
10.20.30.0/20
172.16.254.0/24
172.17.254.0/24
10.20.128.0/17
192.168.242.0/24
192.168.241.0/24
10.20.0.0/17
172.16.255.0/24
172.17.255.0/24
203.0.113.192/26
203.0.113.224/27

(20')

▸**Exercise 245.** The following link-state advertisements are broadcast within a network that uses a link-state routing protocol.

$$\begin{aligned}
\text{LSA}_a &= \{(d,2),(f,6)\}\\
\text{LSA}_b &= \{(d,7),(e,3),(f,1)\}\\
\text{LSA}_c &= \{(e,2),(f,1)\}\\
\text{LSA}_d &= \{(a,2),(b,7),(e,1)\}\\
\text{LSA}_e &= \{(b,3),(c,2),(d,1)\}\\
\text{LSA}_f &= \{(a,6),(b,1),(c,1)\}
\end{aligned}$$

Compute the *diameter* of the network. Recall that the diameter of a graph is the maximal distance between any two vertices, where the distance between two vertices u and v is the minimal length of a path connecting u and v. Justify your answer. *(20')*

▸**Exercise 246.** A router g issues *link-state advertisement* $\text{LSA}_g = \{(a,5),(b,1),(d,2),(e,5)\}$ and receives the following other advertisements, where letters $(a,b,c,\ldots)$ represent router addresses.

$$\begin{aligned}
\text{LSA}_a &= \{(b,3),(c,1),(e,4),(f,1)\}\\
\text{LSA}_b &= \{(a,3),(c,6),(d,4),(g,1)\}\\
\text{LSA}_c &= \{(a,1),(b,6),(d,6)\}\\
\text{LSA}_d &= \{(b,4),(c,6),(g,2)\}\\
\text{LSA}_e &= \{(a,4),(f,2),(g,5)\}\\
\text{LSA}_f &= \{(a,1),(e,2)\}
\end{aligned}$$

Based on these advertisements, write the forwarding tables of all routers. For destination addresses, use the symbolic labels $a,\ldots,g$; also, identify each interface by the label of the corresponding adjacent router. *(20')*

▸**Exercise 247.** A sender A sends a file to a receiver B using a stop-and-wait transport protocol over a link with maximum segment size MSS = 1KB, transmission rate R = 1000KB/s, and delay D = 50ms. The sender detects errors with a fixed timeout T = 0.5s. Acknowledgment packets can be considered to have zero length.

Question 1: How long would it take to transmit a 400KB file in the best case? Justify your answer. *(5')*

Question 2: What is the expected transmission time for a 400KB file when each packet is dropped with probability $p = 0.01$ (i.e., one every 100 on average)? Justify your answer. *(15')*

▸**Exercise 248.** A router has 10 input ports and 10 output ports. All input and output ports have the same maximum throughput of 1GB/s.

Question 1: Suppose that the switch fabric has a maximum throughput of 5GB/s, and that the input traffic is such that half of it goes evenly distributed to output ports 1 and 2, and the rest goes to the remaining output ports (3–10), also evenly distributed. What is the expected output of each output port? Does the router drop packets? If so, specify where and for each point specify the loss rate. If not, explain why not. *(10')*

Question 2: Would it be possible to design the router so that packets are never queued? If so, explain why and in particular specify the necessary throughput of the switch fabric? If not, explain why not. *(10')*

►**Exercise 249.** Consider a peer-to-peer system in which host A holds a 100MB file and three other hosts B_1, B_2, and B_3 want to obtain that file. The access link of A has a maximum upload speed of U_A = 200KB/s while all other hosts have an access link with maximum upload and download speeds of U_B = 60KB/s and D_B = 500KB/s, respectively.

Question 1: Is it possible to transfer the file from A to all other hosts in 10 minutes or less? If so, explain how. If not, explain why not. *(5')*

Question 2: In order to reduce the transfer time (from A to all other hosts) you may choose one of the following improvements: (1) double A's upload speed U_A, (2) double the other hosts' upload speed U_B, or (3) double their download speed D_B. Which one would you choose? In that case, what would be the best way to transfer the file, and how long would it take? *(15')*

►**Exercise 250.** A university e-mail system serves a mailing list *cn@inf.usi.ch* for the computer networking class. The mailing list includes *joe@email.ch*, *jane@gmail.com*, *mario@email.ch*, and *luigi@gmail.com*. Suppose now that *antonio@usi.ch* sends a message to the computer networking mailing list. Describe in detail all the network communications between the university server and the rest of the network. In particular, describe all SMTP sessions and all DNS messages. *(20')*

(This page is intentionally left blank)

►**Exercise 251.** The network between hosts A and B has a maximum segment size MSS = 1KB, a total delay of D = 200ms, and an infinite transmission rate $R = \infty$. Consider the transmission of a 20KB file from A to B using TCP. Assume that A initiates the connection at time $t = 0$ and that the CPUs of A and B are infinitely fast such that the processing time is always zero.

Question 1: Assuming the network is perfectly reliable, write all the TCP packets exchanged by A and B. For each packet, write the time the packet is sent, the SYN and ACK flags, if present, and the sequence number and the acknowledgment number, if meaningful. For example, you should write "t = 123ms, $A \longrightarrow B$, ACK, seq = 2345, ack = 3456" for a packet sent at time t = 123ms from A to B carrying the ACK flag, the sequence number 2345 and the acknowledgment number 3456. *(10')*

Question 2: Now consider the transfer of a 200KB file, and in this case assume that the network looses the 100th packet sent by A. Exactly how long does it take for A to transmit the entire file? Justify your answer by showing a synthetic trace of the packets exchanged by A and B. In this case, do not write every single packet but instead write the initial time and the initial and final sequence number of every sequence of consecutive packets sent by A (e.g., "t = 123ms, $A \longrightarrow B$, 10pkts, seq = 1000...11000"). Show and briefly explain exactly what happens after the loss of the 100th packet. *(10')*

►**Exercise 252.** Consider a web browser connected to the internet through an access link with transmission delay d = 60ms and a very high transmission rate $R = \infty$. What is the absolute minimum time necessary for the browser to display a web page when (a) the browser must open a new TCP connection to an unknown server; (b) the browser must open a new TCP connection to a server whose address is in the local DNS cache; (c) the browser reuses an open TCP connection? Justify your answers. *(10')*

►**Exercise 253.** Consider an image-processing service running on host *process.images.ch* accessible through HTTP. The service processes images by applying algorithms such as edge detection or color balancing.

Question 1: Explain how you would implement such a service by showing an HTTP exchange for the edge-detection processing of a 2MB jpeg image resulting in a 1MB image. Show all the relevant headers. *(10')*

Question 2: Suppose a client requests the service described in Question 1 through a network link with propagation delay $d = 150$ms and transmission rate $R = 2$MB/s. How long would it take for the client to invoke the service and obtain the result when the server processing time is $T_{\text{proc}} = 50$ms? Justify your answer. *(10')*

►**Exercise 254.** Consider users connected to the internet with the same type of access link with maximum upload rate R_u and maximum download rate R_d. Consider now that $s > 0$ users hold a file and that $c > 2$ users want to obtain that file.

Question 1: Describe a case in which the total download time of a client-server system would be the same as that of a peer-to-peer system. Give the specific conditions on the various network and file parameters R_u, R_d, s, c. Also, illustrate the case with a concrete example. *(10')*

Question 2: Describe a case in which the total download time of a client-server system would be ten times that of a peer-to-peer system. Give the specific conditions on the various network and file parameters R_u, R_d, s, c. Also, illustrate the case with a concrete example. *(20')*

►**Exercise 255.** A sender sends a file of size $F = 2$MB over a link with maximal segment size $MSS = 1$KB, propagation delay $d = 40$ms, and transmission rate $R = 500$KB/s, using the *stop-and-wait* protocol with a timeout set to $t = 200$ms.

Question 1: What is the utilization factor—that is, the proportion of time in which the link is actually in use—in the best case (no errors)? Justify your answer. *(10')*

Question 2: What is the total transfer time in the best case, that is without network errors? Justify your answer. *(10')*

Question 3: What is the total transfer time when packets are randomly dropped by the network with probability $p_e = 1\%$ (error probability)? Justify your answer. *(10')*

►**Exercise 256.** A sender sends a file of size $F = 2$MB over a link with maximal segment size $MSS = 1$KB, propagation delay $d = 40$ms, and transmission rate $R = 500$KB/s, using the *go-back-N* protocol with a timeout set to $t = 200$ms and a window size of W packets.

Question 1: What is the utilization factor—that is, the proportion of time in which the link is actually in use—in the best case (no errors) when $W = 10$? Justify your answer. *(10')*

Question 2: What is the minimal window size that would guarantee a best-case (no errors) utilization factor of 100% for the link? Justify your answer. *(10')*

Question 3: What is the total transfer time when $W = 100$ and when packets are randomly dropped by the network with probability $p_e = 1\%$ (error probability)? Justify your answer. *(10')*

►**Exercise 257.** A computer networking class uses a network reserved for student projects. The network is defined by the prefix 73.90.80.0/24. The class is then divided into three groups: group A needs at least 100 IP addresses, group B needs at least 50 addresses, and group C needs at least 90 addresses. The three groups have their private routers, R_A, R_B, and R_C. R_A and R_C are connected to the Internet through an access router R_X, R_B connects indirectly through R_A. Draw a diagram of the network, assign numbers to each router interface, including the interfaces to the local network, and write the forwarding tables of each router. *(20')*

►**Exercise 258.** A router has 10 full duplex ports, so 10 input and 10 output ports. The input and output links have the same transmission rate $R = 1$GB/s with packets that have a minimum size $S_{\min} = 50$B and a maximum size $S_{\max} = 2000$B. The input and output ports each have a queue of 20000 packets. The switch fabric has a maximum throughput $T_{SW} = 10$Mpackets/s.

Question 1: The router processes intense but short bursts of traffic, whereby there is no traffic for a long period of time, then there is a burst of packets of size 500B that saturates every input port for 15ms, and then the traffic goes back to zero. The traffic is such that it is forwarded uniformly onto 5 output ports. Does the router drop any packet? If so, in which queues? What is the maximal delay of the packets that go through? *(10')*

Question 2: What happens if the packets in the traffic bursts of Question 1 double in size but still saturate the input links? Does the router drop packets? If so, in which queues? What is the maximal delay of the packets that go through? *(10')*

▶**Exercise 259.** Consider a network link with propagation delay $d = 50$ms, and transmission rate $R = 400$KB/s, used by the *go-back-N* protocol with maximal segment size $MSS = 1$KB, timeout set to $t = 100$ms and a window size of $W = 20$ packets.

Question 1: What is the transmission time for a 40MB file in the best case? *(10')*

Question 2: Would it be possible to speed up the download by (a) doubling the transmission rate to $R' = 2R$, or by (b) doubling the window size $W' = 2W$? Which option would be better? Justify your answer. *(10')*

▶**Exercise 260.** The SYN-ACK packet of a TCP connection has the following headers: source port 1234, destination port 5678, sequence number 1800, ACK number 9000. The connection carries an HTTP GET request that results in a 404 error, after which the client decides to close the connection.

Question 1: Write the HTTP request and the corresponding reply exchanged by client and server over the TCP connection (application level). Make the request and the reply simple but plausible. *(5')*

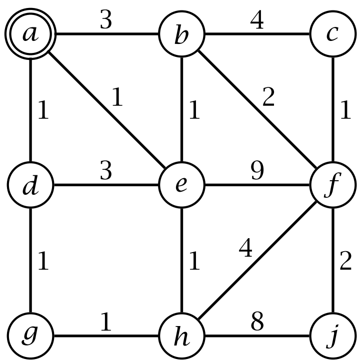
Question 2: Assuming the network is perfectly reliable, write all TCP packets transmitted between client and server after the specified SYN-ACK segment. For each packet, clearly specify the values of all the relevant TCP headers. Justify your answer by briefly describing the relations between header values. *(15')*

▶**Exercise 261.** Answer the following questions about TCP.

Question 1: A triple duplicate acknowledgment in TCP triggers a retransmission. Briefly explain the rationale of this design. Why duplicate acknowledgments? Why wait for three and not retransmit immediately after one? *(10')*

Question 2: Consider the additive increase of the congestion window in TCP. Suppose the congestion window is at $10MSS$ on a connection with round-trip time RTT. When and how much does TCP change the congestion window? How long does it take for the congestion window to reach $18MSS$? In that period, what is the average throughput as a function of MSS and RTT? Assume no packet losses or errors. *(10')*

▶**Exercise 262.** Explain how a controlled flooding works. Consider a controlled flood of message m from node a in the network illustrated below in which edge weights represent propagation delays. Assuming that the network is perfectly reliable (no packet losses) and that the transmission rates are infinite, write all the packets exchanged in the flood of a message m issued by node a at time $t = 0$. In particular, for each packet p departing from node v at time t_1 and arriving at node w at time t_2, write "$v : t_1 \to w : t_2$".



(20')

▶**Exercise 263.** A server S holds a file of F that n clients C_i want to download. The server has a network access link with maximum upload and download rate $U_S = D_S$. The clients all have the same type of access link with a maximum download rate that is four times the maximum download rate, so $D_c = 4U_C$. Consider the total download times of an ideal client-server download or an ideal peer-to-peer download. The total download time is the minimal time necessary for all the clients to obtain the entire file. Ideal here means that there are no errors, and the transmission scheduling is as good as it can be, so transmission times are only limited by transmission rates.

Question 1: Is there a case in which the peer-to-peer download is slower than the client-server download. If so, characterize this case in terms of the upload and download rates. If not, explain why. *(10')*

Question 2: Is there a case in which the client-server download is as fast as the peer-to-peer download. If so, characterize this case in terms of the upload and download rates. If not, explain why. *(10')*

Question 3: In the case of the peer-to-peer download, the server stops serving the file after all the n peers have downloaded their copy of the file. At that point, m new peers with the same access link (same rates D_C and U_C) join the group to download the file. What is the total download time for these new m peers as a function of F, U_C, n, and m? *(10')*

►**Exercise 264.** A router exchanges continuous messages with its four neighbors, connected to ports 1, 2, 3, and 4, as part of a distance-vector routing protocol. The messages allow the router to compute the costs of each connection as follows:

port	1	2	3	4
cost	15	8	5	10

The router then receives the following distance vectors from its neighbors. Each vector associates IP addresses or prefixes with a corresponding distance. Notice that not all addresses are present in all the vectors.

port	*distance vector*
1	$(a_1:7)$ $(a_2:30)$ $(a_3:24)$ $(a_4:40)$ $(a_6:5)$ $(a_7:0)$ $(a_9:20)$ $(a_{10}:36)$
2	$(a_2:35)$ $(a_3:18)$ $(a_4:32)$ $(a_6:8)$ $(a_7:8)$ $(a_8:17)$ $(a_9:42)$ $(a_{10}:14)$
3	$(a_1:13)$ $(a_2:30)$ $(a_4:40)$ $(a_5:19)$ $(a_6:5)$ $(a_7:9)$ $(a_8:12)$ $(a_9:36)$ $(a_{10}:20)$
4	$(a_1:5)$ $(a_2:28)$ $(a_4:10)$ $(a_5:23)$ $(a_6:7)$ $(a_7:9)$ $(a_8:0)$ $(a_9:27)$ $(a_{10}:25)$

Question 1: Write the forwarding table that the router computes. Briefly justify your answer by explaining the essence of distance-vector routing. *(15')*

Question 2: Later measurements lead the router to update the costs of the connections with its neighbors as follows:

port	1	2	3	4
cost	6	8	5	10

Would the router compute an updated forwarding table? If so, write the new table. If not, explain why. Also, write the distance vector that the router would transmit to its neighbors. *(15')*

►**Exercise 265.** A Web user agent needs to render an image located at *http://usi.ch/logo.jpg*. The Web user agent connects to the *usi.ch* server to obtain the object. However, the user agent also has a cached copy of that object retrieved on Mon, 03 Sep 2018 12:30:00 GMT that is still the most current version of the object on the origin server.

Question 1: Write the full HTTP exchange between client and server. Briefly explain your answer. *(10')*

Question 2: Write all the TCP packets of the connection between client and server. For each packet, specify all the important information in the IP and TCP headers (addresses, ports, flags, sequence numbers, etc.) as well as the content. *(20')*

►**Exercise 266.** A sender A sends a file to a receiver B using a stop-and-wait transport protocol over a link with maximum segment size $MSS = 1\text{KB}$, transmission rate $R = 1000\text{KB/s}$, and delay $D = 25\text{ms}$. The sender detects errors with a fixed timeout $T = 0.2\text{s}$. Acknowledgment packets can be considered to have zero length.

Question 1: How long would it take to transmit a 800KB file in the best case? Justify your answer. *(10')*

Question 2: What is the expected transmission time for a 800KB file when each packet is dropped with probability $p = 0.05$ (i.e., 5% on average)? Justify your answer. *(20')*

Solutions

WARNING: solutions are very sparse, meaning that many are missing, most of the solutions are only sketched at a high level, and many may be incorrect! Please, consider contributing your solutions, including alternative solutions, and please report any error you might find to the author (Antonio Carzaniga <antonio.carzaniga@usi.ch>).

▷ *Solution 235.1*
We start by figuring out the basic, error-free behavior of the protocol on the given link. We first need to know whether the window size is large enough to saturate the link capacity, or otherwise whether the transmission stalls for some time waiting for acknowledgments even in the absence of errors. The question is whether the first acknowledgment arrives before or after the sender has filled the window? Filling the window means transmitting W packets, which means $W \cdot MSS = 20KB$. At a transmission $R = 50KB/s$, the sender transmits the whole window in

$$T_W = \frac{W \cdot MSS}{R} = \frac{20KB}{50KB/s} = 0.4s$$

which is less than the time needed for the first acknowledgment to arrive, which is

$$T_{ACK1} = \frac{MSS}{R} + 2d = 0.02s + 0.6s = 0.62s$$

This means that even in the absence of errors, the transmission stalls for $0.1s$ after each full window is sent. In other words, each window of $W = 20$ takes $2d = 0.62s$, and with a file size $S = 200KB$, we need to send $S/(WMSS) = 90$ full windows, so the error-free transfer time is:

$$\Delta = T_{ACK1}\frac{S}{WMSS} = 55.8s$$

.

▷ *Solution 235.2*
Now, in the presence of errors, that will take longer because every error causes the sender to stall for one timeout. More precisely, errors in acknowledgment packets do not cause trouble because acknowledgments are cumulative, so one lost ack would be quickly superseded by the next one. On the other hand, errors in data packets will stall the sender for one timeout interval. The question then how many errors will occur on average when we need to transmit n segments, where

$$n = \frac{S}{MSS} = 1800$$

In expectation, meaning on average, the number of errors is simply the probability of errors multiplied by the total number of packets sent. More specifically, again, we only care about *data* packets. Thus we need to compute the total number of packets we need to send, which we can do by considering that each error causes the retransmission of W packets. So, let x be the total number of packets sent in order to transmit n segments, then on average there will be px errors, and therefore pxW additional packets. This gives us this equation to compute x:

$$x = \frac{n}{1 - pW} = 2000$$

Thus we need to transmit

$$x/W = 100$$

full windows, which will take $100T_{ACK1} = 62s$, and we will incur $px = 10$ errors (in expectation) and therefore 10 timeouts, $T = 1s$, for a total transmission time of

$$\Delta = \frac{x}{W}T_{ACK1} + 10T = 72s.$$

▷ *Solution 236.1*

False. The Internet is a datagram network. Each packet is treated independently and is afforded a "best-effort" transmission service. This also means that the network does not reserve resources, and therefore can not guarantee even a minimal quality of service. Instead, a virtual-circuit network can and does provide exactly that kind of service.

▷ *Solution 236.2*

True. No more than one application can hold the same port number on a host. In fact, port numbers are the way applications are addressed within a host. TCP (and UDP) use 16 bits for port numbers, so there are at most $2^{16} = 65536$ open ports at the same time.

▷ *Solution 236.3*

The browser would not even be able to find the network address of the web-mail system, so it would not even try to connect to the web-mail system, let alone return a 404 error. The web-mail system would keep functioning perfectly. It would still be possible

▷ *Solution 237.1*

Case a: for each of the four objects, the client opens a connection, requests the object, gets the object, closes the connection. So, assuming the size of the requests is negligible, for each object i, the time is $T_i = 2d + S_i/R$. And everything is sequential, to we must add up each T_i:

$$T = T_1 + T_2 + T_3 + T_4 = 4 \cdot 2d + \frac{S_1 + S_2 + S_3 + S_4}{R} = 0.4s + \frac{4.105MB}{1MB/s} = 4.505s$$

Case b: the client opens one persistent connection, then makes four requests back-to-back for the four objects, then gets the four objects, and closes the connection. The total time is:

$$T = 2d + \frac{S_1 + S_2 + S_3 + S_4}{R} = 0.1s + \frac{4.105MB}{1MB/s} = 4.205s$$

Case c: the client opens four connections, then on each connection it requests one of the objects, gets that object, and then closes that connection. Intuitively, this case should not be substantially different from case b because parallel connections share the connection evenly. To see that, consider n parallel connections sending x bytes each. The total transfer time is the same as the transfer time on each connection, which is:

$$T_{parallel}(n, x) = 2d + \frac{x}{R/n} = 2d + \frac{nx}{R}$$

but this is also the same as the total time for n pipelined requests on one connection:

$$T_{pipeline}(n, x) = 2d + n\frac{x}{R}$$

.

Yet another way to compute the time in this specific case is to consider the three phases of the parallel download. Initially, the four downloads proceed in parallel, until the first connection terminates. This initial period lasts until time $T_1 = 2d + 4S_1/R$, and in this period all four connections download S_1 bytes. Then, the remaining three connection continue until the second object is downloaded, which happens at time $T_2 = T_1 + 3(S_2 - S_1)/R$, at which point the remaining two connections will also have downloaded S_2 bytes, and will then continute to finish downloading until time $T_4 = T_2 + 2(S_3 - S_2)/R$. The total time is

$$T = 2d + \frac{4S_1}{R} + \frac{3(S_2 - S_1)}{R} + \frac{2(S_3 - S_2)}{R} = 2d + \frac{S_1 + S_2 + 2S_3}{R} = 4.205s$$

▷ *Solution 237.2*

Case b: we have one persistent connection, four sequential back-to-back requests for the four objects, and then the four objects coming sequentiallyin that order. If the first object is in the proxy, then the total download is bounded by the download of the remaining three (larger) objects over the slow link:

$$T_1 = 2d + \frac{S_2 + S_3 + S_4}{R} = 0.1s + 4.1s = 4.2$$

If the second object object is in the proxy, then the total download is bounded by the download of the remaining three objects over the slow link:

$$T_2 = 2d + \frac{S_1 + S_3 + S_4}{R} = 0.1s + 4.01s = 4.012$$

▷ *Solution 238.1*

Each new peer would have to download the whole file. At a rate of R_d, that would take $S/R_d = 300s$. At the same time, the two initial sources must push at least one copy of the file into the network, which is equivalent to uploading the file at twice the upload rate R_u, therefore the time is at least $S/2R_u = 600s$. Then, considering the aggregate in/out flow, the network must transmit 10 copies of the file, which mean that the 12 peers must output $10S$ bytes into the network, at their aggregate rate $12R_u$. Thus the time is at least $10S/12R_u = 1000s$, which is also the overall best possible download time.

▷ *Solution 238.2*

We now have 12 peers with the file, which means that time to upload one copy at their aggregate rate is now very small $S/12R_u = 100s$, which also means that the single download time dominates $S/R_d = 300s$. But what about the aggregate upload flow? We now have a total of 20 peers, and we need to upload a total of $8S$ bytes (the new copies) into the network. Therefore the time is at least $8S/20R_u = 480s$, which is the dominating time for the download. We reach the constant minimum given by the download rate, $S/R_d = 300s$, when 32 or more peers are in the system.

▷ *Solution 240*

It is reasonable to assume that the name space of the *inf.usi.ch* domain is managed by one DNS server $S_{inf.usi.ch}$. (It is also reasonable to assume that there is a single server for the entire *usi.ch* domain—in fact, that is precisely what we have in reality—but just to make things a bit more fun here...) So, local DNS would reach this authoritative DNS server with an iterative process. The server would first contact one of the *root* DNS servers, S_{root}, from which it would get the address of the authoritative server for the ".ch" domain, S_{ch} (one or more of them); then the local DNS would contact S_{ch}, from which it would get the address of the authoritative server for the "usi.ch" domain, $S_{usi.ch}$ (one or more of them); then the local DNS would contact $S_{usi.ch}$, from which it would get the address of the authoritative server for the "inf.usi.ch" domain, $S_{inf.usi.ch}$ (one or more of them); then the local DNS would contact $S_{inf.usi.ch}$, from which it would get the address of the host *www.inf.usi.ch*.

Thus these are the DNS packets involved in this process:

1. From local DNS to S_{root}, query: id=1, name="www.inf.usi.ch", type=A.

2. From S_{root} to local DNS, answer: id=1, name="ch", type=NS, value=S_{ch}; additional info: name=S_{ch}, type=A, value=... (IPv4 address of S_{ch}).

3. From local DNS to S_{ch}, query: id=2, name="www.inf.usi.ch", type=A.

4. From S_{ch} to local DNS, answer: id=2, name="usi.ch", type=NS, value=$S_{usi.ch}$; additional info: name=$S_{usi.ch}$, type=A, value=... (IPv4 address of $S_{usi.ch}$).

   ...

5. From local DNS to $S_{inf.usi.ch}$, query: id=5, name="www.inf.usi.ch", type=A.

6. From $S_{inf.usi.ch}$ to local DNS, answer: id=5, name="www.inf.usi.ch", type=A, value=... (IPv4 address of *www.inf.usi.ch*).

▷ *Solution 241.1*

The router would need to have output queues when the total traffic reaching an output port can exceed the transmission rate of that port (R). The total traffic reaching an output port is at most T, since that is what can get through the switch fabric. Also, the total *input* rate would have to be more than R, which means that the router must have $n > 1$ ports. Therefore the answer is yes—the router needs output queues—if and only if $T \geq R$ and $n > 1$.

In this case, an output queue of length Q packets would fill up when the traffic from two or more input ports with total input rate $\lambda > R$ would have to be forwarded through output port 1 for some period of time. Since the queue would accumulate packets at a rate $\lambda - R$, then the router would have to drop packets after time

$$t = \frac{Q}{\lambda - R}$$

▷ *Solution 241.2*

The router would need to queue packets before the switch fabric if the total input rate, meaning the rate of traffic reaching the switch fabric, can exceed the throughput of the switch fabric. The total input rate is nR, therefore the answer is yes—the router needs a queue before the switch fabric—if and only if $T < nR$.

In this case, a queue of length Q packets would fill when the total input rate is $\lambda > T$ for some period of time. Since the queue would accumulate packets at a rate $\lambda - T$, then the router would have to drop packets after time:

$$t = \frac{Q}{\lambda - T}$$

▷ *Solution 242.1*

The mechanism by which TCP controls the sending rate is the congestion window W. The congestion window is the amount of data (bytes) the sender is willing to send into the network before receiving an acknowledgment. It corresponds to the window size in Go-Back-N (multiplied by the maximum segment size MSS, since the window in Go-Back-N typically counts the number of packets, not bytes). The sender sends W bytes, and then stops waiting for acknowledgments. So, effectively, assuming no packet loss, the sender sends one full congestion window for every round-trip-time. Therefore, the sending rate is $R = W/RTT$.

For example, considering a TCP connection between two hosts where the round-trip time is $100ms$, then TCP must have a congestion window $W = 100KB$ in order for the sending rate to be $R = 1MB/s$.

▷ *Solution 242.2*

At steady state, TCP ramps up the sender rate linearly. The sending rate is controlled by the congestion window. When the sending rate exceeds the maximum link rate, some of the packets within one congestion window will be dropped, while others will still reach the receiver. The receiver then notices the hole in the sequence number, and therefore sends back a repeated acknowledgment (equivalent to a NACK). The sender then reacts by cutting the sending rate in half, and then starting again its linear increase.

As a result, the TCP sender sends at a rate R_{TCP}. that varies over time linearly between $R_{link}/2$ and R_{link}. Thus, on average, TCP sends at a rate $R_{TCP} = 0.75R_{link}$.

▷ *Solution 243*

The routing table at router x contain x's own distance vector D_y, which is the vector of the currently best-known distances between x and every other router, the vector n_x of next-hop routers on the currently shortest path to every other router, and then a set of distance vectors D_y for every neighbor y that has shared its distance vector with x. The distances that are updated at each step are highlighted in bold font.

Step 1

(*a*)	*a*	*b*	*c*	*d*	*e*
D_a	0	1	6	4	∞
n_a	*a*	*b*	*c*	*d*	–

(*b*)	*a*	*b*	*c*	*d*	*e*
D_b	1	0	∞	1	1
n_b	*a*	*b*	–	*d*	*e*

(*c*)	*a*	*b*	*c*	*d*	*e*
D_c	6	∞	0	1	∞
n_c	*a*	–	*c*	*d*	–

(*d*)	*a*	*b*	*c*	*d*	*e*
D_d	4	1	1	0	5
n_d	*a*	*b*	*c*	*d*	*e*

(*e*)	*a*	*b*	*c*	*d*	*e*
D_e	∞	1	∞	5	0
n_e	–	*b*	–	*d*	*e*

Step 2

(*a*)	*a*	*b*	*c*	*d*	*e*
D_a	0	1	**5**	**2**	**2**
n_a	*a*	*d*	*c*	*b*	*b*
D_b	1	0	∞	1	1
D_c	6	∞	0	1	∞
D_d	4	1	1	0	5

(*b*)	*a*	*b*	*c*	*d*	*e*
D_b	1	0	**2**	1	1
n_b	*a*	*b*	*d*	*d*	*e*
D_a	0	1	6	4	∞
D_d	4	1	1	0	5
D_e	∞	1	∞	5	0

(*c*)	*a*	*b*	*c*	*d*	*e*
D_c	**5**	**2**	0	1	**6**
n_c	*d*	*d*	*c*	*d*	*d*
D_a	0	1	6	4	∞
D_d	4	1	1	0	5

(*d*)	*a*	*b*	*c*	*d*	*e*
D_d	**2**	1	1	0	**2**
n_d	*b*	*b*	*c*	*d*	*b*
D_a	0	1	6	4	∞
D_b	1	0	∞	1	1
D_c	6	∞	0	1	∞
D_e	∞	1	∞	5	0

(*e*)	*a*	*b*	*c*	*d*	*e*
D_e	**2**	1	**6**	**2**	0
n_e	*b*	*b*	*d*	*b*	*e*
D_b	1	0	∞	1	1
D_d	4	1	1	0	5

Step 3

(*a*)	*a*	*b*	*c*	*d*	*e*
D_a	0	1	**3**	2	2
n_a	*a*	*d*	*b*	*b*	*b*
D_b	1	0	2	1	1
D_c	5	2	0	1	6
D_d	2	1	1	0	2

(*b*)	*a*	*b*	*c*	*d*	*e*
D_b	1	0	2	1	1
n_b	*a*	*b*	*d*	*d*	*e*
D_a	0	1	5	2	2
D_d	2	1	1	0	2
D_e	2	1	6	2	0

(*c*)	*a*	*b*	*c*	*d*	*e*
D_c	**3**	2	0	1	**3**
n_c	*d*	*d*	*c*	*d*	*d*
D_a	0	1	5	2	2
D_d	2	1	1	0	2

(*d*)	*a*	*b*	*c*	*d*	*e*
D_d	2	1	1	0	2
n_d	*b*	*b*	*c*	*d*	*b*
D_a	0	1	5	2	2
D_b	1	0	2	1	1
D_c	5	2	0	1	6
D_e	2	1	6	2	0

(*e*)	*a*	*b*	*c*	*d*	*e*
D_e	2	1	**3**	2	0
n_e	*b*	*b*	*b*	*b*	*e*
D_b	1	0	2	1	1
D_d	2	1	1	0	2

Step 4

(*a*)	*a*	*b*	*c*	*d*	*e*
D_a	0	1	3	2	2
n_a	*a*	*d*	*b*	*b*	*b*
D_b	1	0	2	1	1
D_c	3	2	0	1	3
D_d	2	1	1	0	2

(*b*)	*a*	*b*	*c*	*d*	*e*
D_b	1	0	2	1	1
n_b	*a*	*b*	*d*	*d*	*e*
D_a	0	1	3	2	2
D_d	2	1	1	0	2
D_e	2	1	3	2	0

(*c*)	*a*	*b*	*c*	*d*	*e*
D_c	3	2	0	1	3
n_c	*d*	*d*	*c*	*d*	*d*
D_a	0	1	3	2	2
D_d	2	1	1	0	2

(*d*)	*a*	*b*	*c*	*d*	*e*
D_d	2	1	1	0	2
n_d	*b*	*b*	*c*	*d*	*b*
D_a	0	1	3	2	2
D_b	1	0	2	1	1
D_c	3	2	0	1	3
D_e	2	1	3	2	0

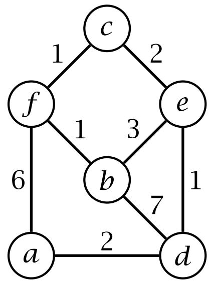
(*e*)	*a*	*b*	*c*	*d*	*e*
D_e	2	1	3	2	0
n_e	*b*	*b*	*b*	*b*	*e*
D_b	1	0	2	1	1
D_d	2	1	1	0	2

▷ *Solution 244*

original	*combinations…*	*minimal*
127.0.0.0/8		127.0.0.0/8
10.20.128.0/17 10.20.0.0/17	10.20.0.0/16	10.20.0.0/16
10.20.30.0/20	10.20.0.0/16	
172.16.254.0/24 172.16.255.0/24	172.16.254.0/23	172.16.254.0/23
172.17.254.0/24 172.17.255.0/24	172.17.254.0/23	172.17.254.0/23
192.168.242.0/24		192.168.242.0/24
192.168.241.0/24		192.168.241.0/24
203.0.113.192/26 203.0.113.224/27	203.0.113.192/26	203.0.113.192/26

▷ *Solution 245*

The link-state advertisements allow us to reconstruct the network graph, as follows:



It is then easy to compute all the shortest paths and therefore the distances between all vertices:

	b	*c*	*d*	*e*	*f*
a	6	5	2	3	5
b		2	4	3	1
c			3	2	1
d				1	4
e					3

From which we see that the diameter is 6.

▷ *Solution 247.1*

Stop-and-wait means that each segment must be acknowledged before the following one is sent. Therefore, since acknowledgments have negligible size, each error-free transmission of a segment and the corresponding acknowledgment takes

$$T_{segment} = \frac{MSS}{R} + 2D = \frac{1\text{KB}}{1000\text{KB/s}} + 0.1\text{s} = 0.101\text{s}$$

So, to send a 400KB file of with a maximum segment size of MSS = 1KB, the sender has to send $n = 400\text{KB}/MSS = 400$ segments. In the best case, every segment and the corresponding acknowledgment will be received correctly, so the total time is

$$nT_{segment} = 400 \times 0.101\text{s} = 40.4\text{s}$$

▷ *Solution 247.2*

Let's first consider the most intuitive solution. We need to send $n = 400\text{KB}/MSS = 400$ segments, this means a total of 800 segments (data packet plus acknowledgment). If the error probability is $p = 0.01$, then we can expect that $800 \times p = 8$ packets will be dropped. So, each of those lost packets will introduce a delay equivalent to the timeout $T = 0.5$s. So, in total, A will have $n = 400$ successful segment transmissions, each lasting

$$T_{segment} = \frac{MSS}{R} + 2D = \frac{1\text{KB}}{1000\text{KB/s}} + 0.1\text{s} = 0.101\text{s}$$

plus 8 timeout delays. The total time is therefore

$$T = 400 \times 0.101\text{s} + 8 \times 0.5\text{s} = 44.4$$

Now, this is not completely correct, since the *additional* packets that are resent due to errors are also subject to errors...
So, let x be the total number of packets necessary to transmit n segments and n acknowledgments in the presence of an error probability p. We can write

$$x = n + px$$

▷ *Solution 252*
Case a: the browser must first query the DNS, then open a TCP connection, then send the HTTP request and get a reply. DNS costs at least one round-trip time; the TCP handshake costs at least another round-trip time; the HTTP request costs at least one round-trip time. Answer: $T = 2d + 2d + 2d = 360$ms.
Case b: the browser knows the IP address, and can directly open the TCP connection, then send the HTTP request and get a reply. the TCP handshake costs at least another round-trip time; the HTTP request costs at least one round-trip time. Answer: $T = 2d + 2d = 240$ms.
Case c: the browser just sends the HTTP request and gets a reply. The HTTP request costs at least one round-trip time. Answer: $T = 2d = 120$ms.

▷ *Solution 253.1*
request:

```
POST /edge-detection HTTP/1.1
Host:  process.images.ch
Connection:  close
Content-type:  image/jpg
Content-size:  2097152

...
```

reply:

```
HTTP/1.1 200 okay
Connection:  close
Content-type:  image/jpg
Content-size:  1048578

...
```

▷ *Solution 253.2*
The client must first submit the image for processing with its request, then the server processes the image, then the server ships the image back to the client in its reply. The total time T is therefore

$$T = d + \frac{S_{\text{in}}}{R} + T_{\text{proc}} + \frac{S_{\text{out}}}{R} + d = 1.9\text{s}$$

▷ *Solution 254.1*
Any case in which the bottleneck is the download rate of the receivers. In other words, any case in which the upload rate dominates the download rate. In particular, when $R_d \leq \frac{s}{c} R_u$. For example, when $s = 1$, $c = 2$, and $R_d = 1$MB/s and $R_u = 2$MB/s.

▷ *Solution 254.2*

This is a case in which the bottleneck for the client-server system is not the client download rate, which would be the same for the peer-to-peer system anyway.

Therefore, the client-server system is limited by the aggregate server upload rate, sR_u. In particular, for a file of size F, the minimal client-server time would be

$$T_{\mathrm{CS}} \geq \frac{cF}{sR_u}.$$

Correspondingly, assuming that the download rate is very high, the peer-to-peer system would also be limited by the aggregate upload rate, so

$$T_{\mathrm{P2P}} \geq \frac{cF}{(s+c)R_u}.$$

Therefore, if we want the client-server limit to be ten times higher than the peer-to-peer limit, we set

$$\frac{cF}{sR_u} \geq 10\frac{cF}{(s+c)R_u} \quad \text{that is, } c \geq 9s.$$

If instead the limiting factor for the peer-to-peer system is instead the download rate, that is, if $\frac{cF}{(s+c)R_u} \leq \frac{F}{R_d}$ that is, if $R_d \leq \frac{s+c}{c}R_u$, then the condition becomes

$$\frac{cF}{sR_u} \geq 10\frac{F}{R_d} \quad \text{that is, } cR_d \geq 10sR_U.$$

Putting everything together the condition is:

$$\begin{array}{ll} c\frac{R_d}{R_u} \geq 10s & \text{if } \frac{R_d}{R_u} \leq \frac{s+c}{c} \\ c \geq 9s & \text{otherwise} \end{array}$$

▷ *Solution 255.1*

Stop-and-wait sends one segment at a time, then wait for the corresponding acknowledgment. Therefore, the total time to send one segment is $2d + MSS/R = 82\text{ms}$, while the actual time in which the link is in use is $MSS/R = 2\text{ms}$. The utilization factor is therefore $2/82 \approx 2.4\%$.

▷ *Solution 255.2*

There are $n = F/MSS = 2000$ segments. In the absence of errors, each segment requires $2d + MSS/R = 82\text{ms}$, so the total time is 164s.

▷ *Solution 255.3*

Let n' be the total number of packets that are necessary to transfer $n = F/MSS = 2000$ segments. A simple way to compute n' is to compute the probability $P(\text{segment okay})$ that a segment will be transferred successfully. That happens when the both the data segment and its acknowledgment are transferred successfully. These are independent events, so that is the product of the probabilities of each event: $P(\text{segment okay}) = (1-p_e)(1-p_e)$.

With this success probability, we can easily calculate how many segments are needed (in expectation) to transfer $n = F/MSS = 2000$ of them successfully. Since $n = n'P(\text{segment okay})$, then $n' = 2000/(1-0.01)^2 \approx 2040$. Therefore we have $n = 2000$ segments delivered correctly, each in $2d + MSS/R = 82\text{ms}$, plus $n' - n \approx 40$ packets that fail, for which the time cost is simply the cost of the timeout $t = 200\text{ms}$. So, the total time is $T = 2000 \times 82\text{ms} + 40 \times 200\text{ms} = 172s$.

▷ *Solution 256.1*

Go-back-N sends W segments before stopping to wait for the first acknowledgment. Therefore, there are two cases: it the first acknowledgment arrives at the sender before the sender has managed to transmit a full window, then the sender simply slides the windows without ever blocking. Otherwise, the sender will block for some time. The first acknowledgment arrives after $T_{\mathrm{ACK}} = 2d + MSS/R = 82\text{ms}$, while the sender sends the whole window in time $T_W = W \times MSS/R = 20\text{ms}$. Therefore, in this case, the sender blocks, then waits, then slides another window, then waits, and so on. The utilization ratio is therefore $T_{\mathrm{ACK}}/T_W \approx 1/4 = 25\%$.

▷ *Solution 256.2*

We use the link fully when the window is large enough that the sender can send continuously before the first acknowledgment returns. The first acknowledgment arrives after $T_{\text{ACK}} = 2d + MSS/R =$ 82ms, so sending at a rate rate R means that we must send at least $W_{\text{OPT}} \times MSS \geq T_{\text{ACK}}R$. So W must be at least $_{\text{OPT}} = T_{\text{ACK}}R/MSS = 41$.

▷ *Solution 256.3*

From the previous exercise we can see that $W = 100$ gives us a full utilization factor. So, in the absence of errors, we start with a delay, then transfer all $n = F/MSS = 2000$, and then again we pay a delay for the last ack. So, the absolute best transfer time is $2d + F/R = 0.04\text{s} + 4\text{s} = 4.04\text{s}$.

When the network drops a packet there are two cases. If the packet is an acknowledgment, then there is no problem, since acknowledgments are cumulative, so the next acknowledgment will cause the window to slide normally. If the packet is a data segment, then the receiver will stop sending acknowledgments, and therefore the sender will fill the window or in any case wait for a time t (timeout) before going back and re-sending the whole window. Therefore, each dropped data segment will introduce a delay in the packet pipeline equal to the timeout t plus one transmission delay d. So, all we need to do is compute the number of dropped data segments.

Again, we can do that by computing the probability that a data segment is sent correctly. In this case, since dropped acks are not a problem, we have $P(\text{segment okay}) = (1 - p_e)$. So, we compute the number of segments n' needed (in expectation) to transfer $n = F/MSS = 2000$ segments successfully as follows: since $n = n'P(\text{segment okay})$, then $n' = 2000/0.99 \approx 2020$. Therefore we have 20 error events, which means that we have to add $20 \times (t + d) = 4.8\text{s}$ to the best transfer time. So, the total time is 8.81s.

▷ *Solution 257*

The following assignment of addresses works well. The table shows the binary prefix of the last byte.

....00—	A	64
....010—	A	32
....0110—	A	8
	total:	104
....10—	C	64
....110—	C	32
	total:	96
....111—	B	32
....0111—	B	16
....01101—	B	8
	total:	56

These are the router interfaces:

router	**interface**	**connects**
R_X	1	Internet
	2	R_A
	3	R_C
R_A	1	R_X
	2	R_B
	3	A's local network
R_B	1	R_A
	2	B's local network
R_C	1	R_X
	2	C's local network

And these are the forwarding tables:

router	prefix	interface
R_X	0.0.0.0/0	1
	73.80.90.0/25	2
	73.80.90.224/27	2
	73.80.90.128/26	3
	73.80.90.192/27	3
R_A	0.0.0.0/0	1
	73.80.90.224/27	2
	73.80.90.112/28	2
	73.80.90.104/29	2
	73.80.90.0/26	3
	73.80.90.64/27	3
	73.80.90.96/28	3
R_B	0.0.0.0/0	1
	73.80.90.224/27	2
	73.80.90.112/28	2
	73.80.90.104/29	2
R_C	0.0.0.0/0	1
	73.80.90.128/26	2
	73.80.90.192/27	2

▷ *Solution 258.1*

For each input port, the burst carries $S = 500$B packets at the maximal input rate $R = 1$GB/s. This means an input rate $\lambda_i = R/S = 2$Mpackets/s for each input port, which means a total of $\lambda_{tot} = 20$Mpackets/s. The switch fabric can only process $T_{SW} = 10$Mpackets/s, so the queues start to fill at a rate $\lambda_{tot} - T_{SW} = 10$Mpackets/s, or 1Mpackets/s per individual queue. The burst lasts for 15ms, so each queue fills up to 15000 packets. So, the router does not drop packets in its input queues.

Then the total traffic coming out of the switch fabric, 10Mpackets/s, goes to five output ports. So, each port gets a total of $\lambda_{out} = 2$Mpackets/s corresponding to $S\lambda_{out} = 500\text{B/packet}\times 2\text{Mpackets/s} = 1\text{GB/s}$, which is just about the maximal transmission rate. So, there is no queuing in the output ports either, which means that the router does not drop packets.

The maximal delay is that of the packets that get queued up with the maximal queue length. The delay is $d_{\text{tot}} = d_{\text{CPU}} + d_{Q_{\text{in}}} + d_{\text{SW}} + d_{Q_{\text{out}}} + d_{\text{TX}}$, where $d_{\text{CPU}} = R/S_{\text{min}}$ is the processing delay in each input port, $d_{Q_{\text{in}}} = |Q_{\text{in}}|/T_{SW}$ is the queuing delay with $|Q_{\text{in}}| = 15000$packets, $d_{\text{SW}} = 1/T_{SW}$ is the delay within the switch fabric, $d_{Q_{\text{out}}} = 0$ is the queuing delay in the output ports (the queues are empty), and $d_{\text{TX}} = R/S$ is the transmission delay.

▷ *Solution 258.2*

In this case, the input rate in terms of packets per second drops by a factor of 2: $\lambda_i = R/S = 1$Mpackets/s for each input port, which means a total of $\lambda_{tot} = 10$Mpackets/s, which is what the switch fabric can process. So, there is no queuing in the input ports. The output throughput in terms of packets per second is the same as in the previous question, $\lambda_{out} = 2$Mpackets/s. However, now the required transmission rate on the output ports doubles, since packets are twice as large. So, each output port can take only half of its share of traffic, meaning that the output queues now grow at a rate of 1Mpackets/s for 15ms. So, same story as in the input. So, no packets are dropped, and the delay is similar.

▷ *Solution 259.1*

We must first figure out whether $W = 20$ saturates the link. Transmitting one full window takes

$$T_W = \frac{W \cdot MSS}{R} = 50\text{ms}$$

which is less than the time it takes for the first ACK to arrive, which is

$$T_{ACK} = 2d + \frac{MSS}{R} = 102.5\text{ms}.$$

Therefore, each full window costs $T_{ACK} = 102.5$ms. We need to send a total of $n = 40\text{MB}/1MSS = 40000$ segments and therefore $n/W = 2000$ full windows. So, the total time is

$$T = T_{ACK}\frac{n}{W} = 205\text{s}.$$

▷ *Solution 259.2*
Doubling the transmission rate would reduce the total transfer time by reducing the transfer time for a single window, but leaving the same number of windows. So, the total time would be lower. Conversely, doubling the window size would not change the transfer time for a single window but would cut the number of windows in half. This is because the transmission of $W' = 2W$ packets would take $T_{W'} = 100$ms, which is still less than the ACK time $T_{ACK} = 2d + \frac{MSS}{R} = 102.5$ms.
So, both option would reduce the time. However, the second option is much better, since it reduces the number of windows and therefore the total time in half, while the first option would save us very little, bringing the transfer time for a single window from $T_{ACK} = 2d + \frac{MSS}{R} = 102.5$ms to $T_{ACK} = 2d + \frac{MSS}{2R} = 101.25$ms, which is a minimal improvement.

▷ *Solution 260.1*
request:

```
GET /carzaniga/antonio.jpg HTTP/1.1
Host:  www.inf.usi.ch

```

reply:

```
HTTP/1.1 404 Not found
Content-type:  text/plain
Content-size:  43

<html><body>Object not found.<body></html>
```

▷ *Solution 260.2*
1. (*client→server*): src-port=5678; dst-port=1234; seq-num=9001; flags={}.
2. (*server→client*): src-port=1234; dst-port=5678; seq-num=1801; ack-num=1862; flags={ACK}
3. (*client→server*): src-port=5678; dst-port=1234; seq-num=9062; ack-num=1914; flags={ACK}
4. (*client→server*): src-port=5678; dst-port=1234; seq-num=9062; flags={FIN}
5. (*server→client*): src-port=1234; dst-port=5678; seq-num=1914; ack-num=9063; flags={FIN,ACK}
6. (*client→server*): src-port=5678; dst-port=1234; seq-num=9063; ack-num=1915; flags={ACK}

▷ *Solution 261.1*
A duplicate acknowledgment is semantically equivalent to a negative acknowledgment. The receiver receives a segment with sequence number higher than expected, which indicates that the missing segment might have been lost. So, the receiver re-acknowledges the missing sequence number—that is, it *requests* that sequence number for the second time.
Retransmitting immediately might be too aggressive. Two consecutive segments could easily swap order, with the second being received right before the first one. Upon receiving the second packet, the receiver would send a duplicate acknowledgment, but then it would immediately receive the missing packet. So, a retransmission in that case would be wasteful.

▷ *Solution 261.2*
The congestion window increases by 1MSS every RTT, so if the congestion window is currently w, TCP increases w by $1/w$ every time it receives an ACK. This is because, within that RTT, TCP would have sent w segments, so it is expected to receive w ACKs, therefore increasing w by $1/w$ for each of the w ACKs leads to an increase of $1MSS$ per RTT.
Thus the congestion window grows from 10 to 18 MSS in 8 RTT. During that time, the increase is linear, therefore the average throughput corresponds to the average window size, namely 14 MSS. So, the throughput is $T = 14MSS/RTT$.

▷ *Solution 263.1*

An ideal download, the time is limited only by the transmission rates. In the case of a client-server download, the server must transmit nF bytes at a maximum rate U_S while each client must receive F bytes at a maximum rate D_C. Therefore, the total download time is at least $T_{\text{cs}} > \frac{nF}{U_S}$ and at least $T_{\text{cs}} > \frac{F}{D_C}$.

In the case of a peer-to-peer download, the server must upload the entire content of the file at least once, but not necessarily n times, since the peers can also exchange information between themselves. So the limit on the side of the server is $T_{\text{p2p}} > \frac{F}{U_S}$ bytes, each client must still download the whole file, so $T_{\text{p2p}} > \frac{F}{D_C}$. Also, all the data exchanged in the peer-to-peer download, which is at least nF bytes since that is what the clients download, must be transmitted into the network at a maximum upload rate $U_{\text{tot}} = U_S + nU_C$ that is the total upload rate of all the participants in the download. So, the total download time is also limited by this transmission: $T_{\text{p2p}} > \frac{nF}{U_S+nU_C}$.

Thus the two download times are limited by different factors, The limit that are due to the server upload rate always favor the peer-to-peer download, since $T_{\text{p2p}} > \frac{F}{U_S}$ is always better than $T_{\text{cs}} > \frac{nF}{U_S}$, assuming $n > 1$. The same is true of the limit that is due to the total upload time in the peer-to-peer case, since $T_{\text{p2p}} > \frac{nF}{U_S+nU_c}$ is also always better than $T_{\text{cs}} > \frac{nF}{U_S}$.

Both client-server and peer-to-peer downloads are limited by the individual download time of the clients $T > \frac{F}{D_C}$, but this limit is the same for both downloads.

In conclusion, a client-server download can never be better than a peer-to-peer download, at least in an ideal model.

▷ *Solution 263.2*

As discussed for the previous question, both client-server and peer-to-peer downloads are limited by the individual download time of the clients $T > \frac{F}{D_C}$, so if this is the dominating limit, then client-server would be just as fast as peer-to-peer.

In particular, this means that

$$\frac{F}{D_C} > \frac{nF}{U_S + nU_C}$$

and since in our case we have $D_C = 4U_C$ that in turn means that

$$U_C < \frac{U_S}{3n}$$

▷ *Solution 263.3*

The limits are exactly the same as before, except that now the n initial peers act as the server, so the time is still limited by the download rate $T > \frac{F}{D_C}$ or $T > \frac{F}{4U_C}$ in terms of the upload rate. Time is also limited by the "server" upload rate, meaning the collective rate of the n peers that are "seeding" the file, so $T > \frac{F}{nU_C}$. Finally, time is limited by the total peer-to-peer transfer $T > \frac{nF}{(n+m)U_C}$. Therefore, $T = \frac{1}{\alpha}\frac{F}{U_C}$, where α is the minimum between 4, n, and $1 + \frac{m}{n}$.

▷ *Solution 264.1*

The essence of distance-vector routing is the Bellman-Ford equation, which effectively states that the next-hop associated with an address a is the port $p = \mathit{fwd}(a)$ such that $\mathit{cost}_p + D_p(a)$. The resulting forwarding table is as follows:

address	*port*
a_1	4
a_2	3
a_3	2
a_4	4
a_5	3
a_6	3
a_7	3
a_8	4
a_9	1
a_{10}	2

▷ *Solution 264.2*
The table would be updated as follows: follows:

address	*port*
a_1	1
a_2	3
a_3	2
a_4	4
a_5	3
a_6	3
a_7	1
a_8	4
a_9	1
a_{10}	2

At this point the router would send the following distance vector to its neighbors:

$(a_1 : 13)$ $(a_2 : 35)$ $(a_3 : 26)$ $(a_4 : 20)$ $(a_5 : 24)$ $(a_6 : 10)$ $(a_7 : 6)$ $(a_8 : 10)$ $(a_9 : 26)$ $(a_{10} : 22)$

▷ *Solution 265.1*
In this case the user agent would send a conditional GET request, to which the server would reply with only a header (no body) that essentially says that the object has not changed. The HTTP exchange might look like this:

client → server

```
GET /logo.jpg HTTP/1.1
Host:  usi.ch
If-Modified-Since:  Mon, 03 Sep 2018 12:30:00 GMT
Connection:  close

```

client ← server

```
HTTP/1.1 304 Not Modified
Connection:  close

```

▷ *Solution 266.1*
Stop-and-wait means that each segment must be acknowledged before the following one is sent. Therefore, since acknowledgments have negligible size, each error-free transmission of a segment and the corresponding acknowledgment takes

$$T_{segment} = \frac{MSS}{R} + 2D = \frac{1\text{KB}}{1000\text{KB/s}} + 0.05\text{s} = 0.051\text{s}$$

So, to send a 800KB file with a maximum segment size of $MSS = 1\text{KB}$, the sender has to send $n = 800\text{KB}/MSS = 800$ segments. In the best case, every segment and the corresponding acknowledgment will be received correctly, so the total time is

$$nT_{segment} = 800 \times 0.051\text{s} = 40.8\text{s}$$

▷ *Solution 266.2*
Let's first consider the most intuitive solution. We need to send $n = 800\text{KB}/MSS = 800$ segments plus the corresponding acknowledgments. This means a total of 1600 segments. If the error probability is $p = 0.05$, then we can expect that $1600 \times p = 80$ packets will be dropped. So, each of those lost packets will introduce a delay equivalent to the timeout $T = 0.5\text{s}$. In total, A will have $n = 800$ successful segment transmissions, each lasting

$$T_{segment} = \frac{MSS}{R} + 2D = \frac{1\text{KB}}{1000\text{KB/s}} + 0.05\text{s} = 0.051\text{s}$$

plus 80 timeout delays. The total time is therefore

$$T_{\text{total}} = 800 \times 0.051\text{s} + 80 \times 0.2\text{s} = 56.8$$

However, this is not completely correct, since the additional packets that are resent due to errors are also subject to errors. So, let x be the total number of packets that are necessary, in expectation, to transmit n segments and n acknowledgments in the presence of an error probability p. Thus we can write $x = 2n + px$ from which we find

$$x = \frac{2n}{1-p}$$

And from that we get, again in expectation, the number of errors $x_e = 2n\frac{p}{1-p} = 84.21$. Thus, again in expectation, the time is

$$T_{\text{total}} = 800 \times 0.051\text{s} + 84.21 \times 0.2\text{s} = 57.64$$