Understanding Computer Networks with Questions (and Answers) Version 0.1

Luca Cittadini Giuseppe Di Battista Maurizio Patrignani

November 4, 2009

Copyright Notice

Preface

The position of the authors is that networking can be fully understood only by means of concrete examples and exercises. This is the origin of the present book, that collects questions, and the corresponding answers, proposed to the students of the networking courses held at Roma Tre University in the last few years.

The questions are grouped into chapters. Each chapter starts with the basic definitions whose knowledge is required to tackle the questions of the chapter and is targeted to a specific networking topic.

The covered topics span more that a single networking course and the contents of the book can be used in several teaching contexts. For example, for a study path that counts two networking courses, one could exploit Exercises **1** to **18** for a course covering basic networking topics. The remaining exercises might be useful for a course on advanced networking topics.

Contents

Chapter 1

Basic Concepts

Propagation delay: It is the time that is needed for the signal to traverse a channel. For all practical uses you can assume that its value is $2/3 \cdot C \cdot d$, where C is the speed of light in empty space, and d is the physical length of the channel. This does not depend on the transmission medium (e.g., copper cable, optical fiber, air, etc.). You cannot decrease the propagation delay unless you bring the source of the message nearer to its destination.

Transmission delay: The time needed to write the message on the transmission channel. It is directly proportional to the size of the message and inversely proportional to the bitrate of the technology.

Circuit switching: It is a transmission strategy based on building a continuous circuit between the source of the message and its destination. The circuit is allocated before transmission, used, and then dismissed. During this time it can not be used by other source-destination pairs.

Packet switching: It is a transmission strategy based on splitting the message into small pieces called *packets*, and sending each packet to an intermediate device towards the destination. This device will in turn forward it to the next intermediate device, and so on, until the destination is reached. This is the paradigm currently used for data transmission in the Internet.

Store and forward: It is the behaviour of intermediate devices in a packet switching network. Intermediate devices receive a packet on an ingress port (*store*), and then *forward* it out of an egress port. Messages are stored completely before forwarding, i.e., the first bit of the forwarded packet is not transmitted until the last bit of the received packet has been stored. In a packet switching network, a packet undergoes a transmission delay for each traversed intermediate device.

Encapsulation: It is the process of adding control information (often in the form of a *packet header*) as information is processed through the protocol stack.

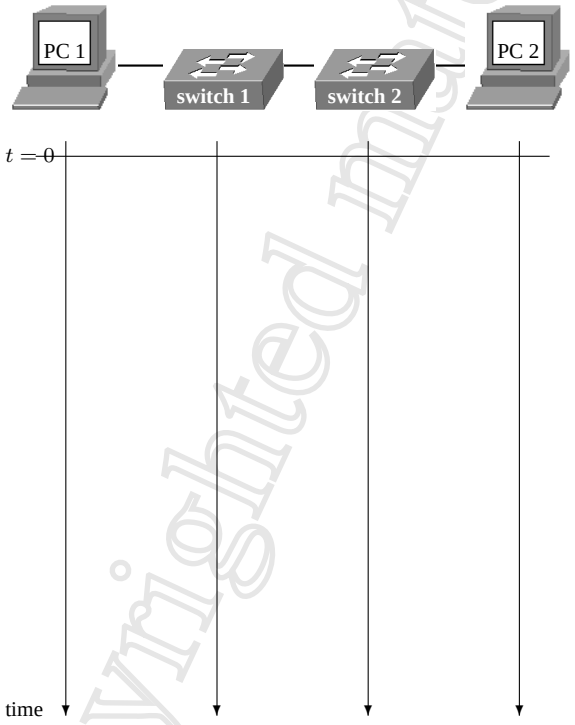
EXERCISE 1: Store and Forward

This exercise is meant to test the understanding of the store and forward paradigm, as well as its role in determining the performance of the network.

In the network represented below, PC 1 and PC 2 are separated by two store-and-forward devices. PC 1 wants to send a $100,000$ bit file to PC 2. In order to do that, the file is split into 100 packets. Suppose PC 1 starts the transmission at time $t = 0$. Further, suppose that:

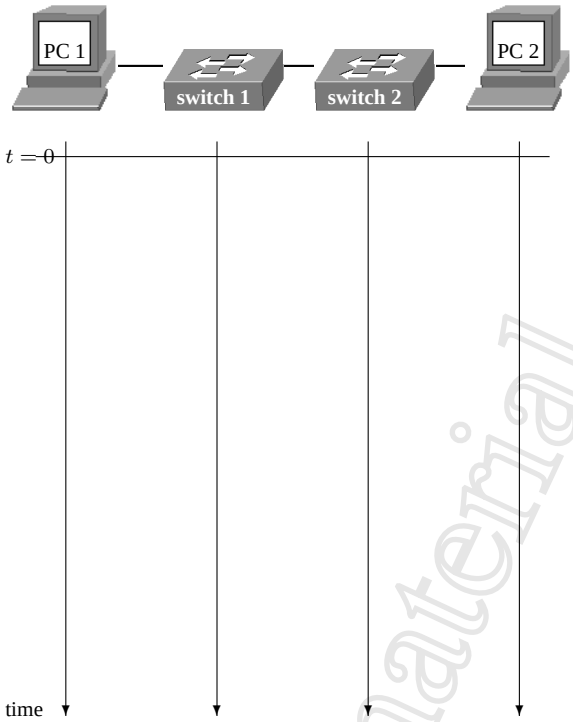
- the links are so short that their propagation delay can be neglected
- you can consider just the payload of the packets, neglecting any headers, trailers, and Inter-Packet Gaps
- the network is currently used only by PC 1 for this transmission
- none of the protocols used to send packets from PC 1 and PC 2 makes use of acknowledgments.

1.1) Assume that all network cards work at 10Mb/sec. Fill in the diagram under the figure of the network, showing for each device the sequence of packets sent over time.



1.2) At what time is the file completely received by PC 2 (i.e., PC 2 gets the last bit of the file)?

1.3) Suppose that all network cards work at 10Mb/sec with the exception of the link between switch 1 and switch 2, where 100Mb/sec is used. Fill in the diagram below, showing for each device the sequence of packets sent over time.



1.4) At what time is the file completely received by PC 2 (i.e., PC 2 gets the last bit of the file)?

EXERCISE 2: As Time Goes By

This exercise is meant to verify the understanding of the interplay between propagation delay and bit rate, and their effect on round-trip time.

Consider two hosts PC 1 and PC 2. Suppose that a PC 1 wants to send a sequence of A packets to PC 2. Each packet consists of B bits, including headers and trailers. Suppose that no acknowledgements are used and that the technology allows a bit rate of R bit/sec.

2.1) Suppose that between PC 1 and PC 2 there is a single router R1 and no other devices. Let P_1 be the propagation delay between PC 1 and R1 and P_2 the propagation delay between R1 and PC 2. Let $P_1 = P_2 = P$. How long does it take to deliver all packets to PC 2?

2.2) Suppose that between PC 1 and PC 2 there are two routers R1 and R2, and no other devices. Let P_1 be the propagation delay between PC 1 and R1, let P_2 be the propagation delay between R1 and R2, and let P_3 be the propagation delay from R2 to PC 2. Let $P_1 = P_2 = P_3 = P$. How long does it take to deliver all packets to PC 2?

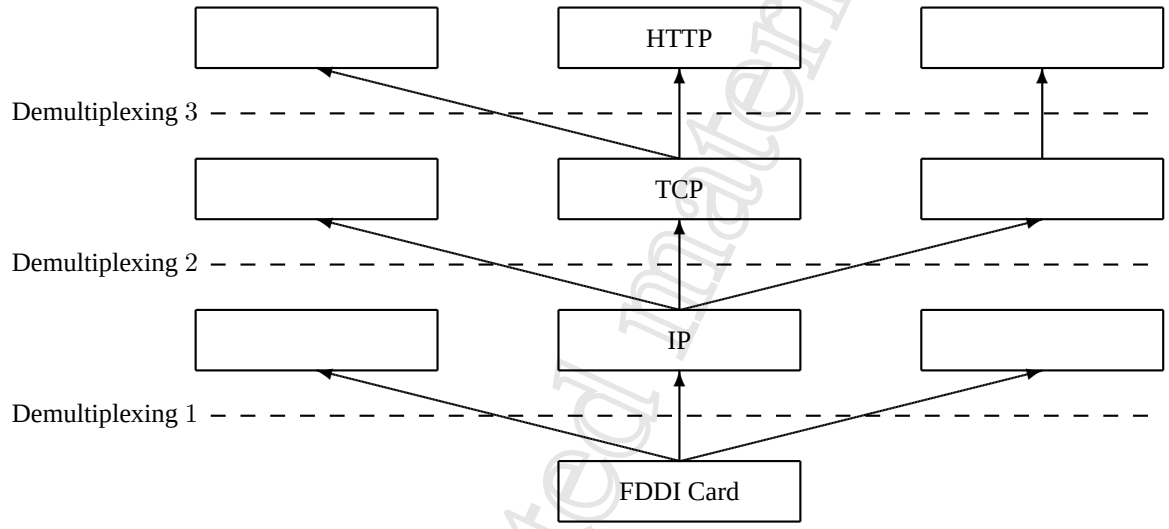
2.3) Suppose that between PC 1 and PC 2 there are $N - 1$ routers and no other devices. Let P be the propagation delay between two consecutive devices. How long does it take to deliver all packets to PC 2?

2.4) Consider the formula of Exercise **2.3** where $A = 1$. Suppose that next years' technological progress will increase the bit rate R. In this case, will the round-trip delay of a packet (measured by the ping command, for example) yield a better or worse approximation of the propagation delay? Argument your answer.

EXERCISE 3: Packets and Encapsulation

This exercise is meant to verify the understanding of the Internet Protocol Suite stack, and its relationship with the ISO-OSI model.

When an FDDI packet is received from a network interface card, it starts travelling up in the protocol stack. When a protocol interpreter in the stack receives the packet, it delivers the payload to the upper layer protocol based on information contained in the packet header. The action of passing the payload to upper layer protocols is called *demultiplexing*. The following diagram shows some protocols of the Internet Protocol Suite, where dashed lines represent demultiplexing actions that are performed from the protocols.



3.1) What information does the FDDI card use to perform demultiplexing?

3.2) What information does the IP layer use to perform demultiplexing?

3.3) What information does the TCP layer use to perform demultiplexing?

3.4) Complete the figure above by filling in the blank cells in the Internet Protocol Suite, according to the demultiplexing hierarchy. (There is more than one way to complete the diagram.)

3.5) Discuss the figure you completed in relation to the ISO-OSI stack.

Chapter 2

CSMA/CD and IEEE 802.3

CSMA/CD: The acronym CSMA/CD stands for Carrier Sense, Multiple Access, Collision Detection, and summarizes the three most important characteristics of protocols adopting this paradigm. CSMA/CD protocols are meant to be used when the transmission medium is shared between several transmitters in such a way that only one at a time can successfully use the medium, and yet the transmission of one station can be detected only with delay by the other stations.

Carrier Sense: A station continuously checks the medium for the presence of a signal (carrier) indicating that some other station is using the medium. If a signal is detected, the station does not transmit. Anytime a station "has a packet to send" but finds the medium used by another station, it will wait until the medium becomes available.

Multiple Access: When no carrier is sensed on the medium, any station could start a new transmission. This implies that two stations could start a transmission without knowing one of the other. The resulting interference between the transmitted signals prevents both transmissions to be successfully completed. This event is called a *collision*.

Collision Detection: When two or more stations start transmitting without knowing of each other, sooner or later their signals interfere, and both stations detect a collision. In this case, since it is not guaranteed that the receivers will be able to read their messages, the collided transmissions have to be aborted. The transmitting stations will close their messages with a *jamming* sequence, that is, a sequence of 96 bits alternating zeros and ones, that will mark the packet as a collision product.

Minimum packet size and maximum network length: A collision is easier to detect when the station is still transmitting. In fact, in this case, it is sufficient to check the differences between the transmitted and the received signal. To ensure that a station will be able to detect a collision, constraint on the minimum packet size must be imposed in relation to the maximum network length (or the other way around). More precisely, you want to be sure to receive the signal of a colliding transmission when you are still transmitting your packet. The worst case occurs when a station A transmits a minimum-length packet and station B, which is located at the maximum distance from A, wants to transmit a packet. In the worst case, the collision happens just a moment before the signal transmitted by A reaches B(after that moment B senses a signal on the medium and is therefore prevented from transmitting, hence it cannot generate collisions). We want A to be able to detect the collision, i.e., it must be guaranteed that the signal transmitted by B reaches A while A has not yet finished transmitting its minimum-length packet. This means that the time needed to transmit a signal from the maximum distance on the network must be less than the time needed to transmit a minimum-length packet.

EXERCISE 4: Good Reasons for Small or Large MTUs

This exercise is meant to understand the issues and tradeoffs related to the choice of the MTU value.

What reasons might have been considered when the 1518 Bytes MTU (Maximum Transmission Unit) for IEEE 802.3 was decided? For each of the following arguments, specify if you think it is a good reason for having a small MTU, a good reason for having a big MTU, or an incorrect argument.

Error rate. Assuming that a frame containing any errors must be discarded by the NIC receiving it, and assuming that the error probability is the same for each bit of the frame, then a longer frame implies a higher probability of errors. As a limit case, a frame of infinite length always contain some wrong bits.	◯ small MTU ◯ big MTU ◯ incorrect
Wait time. If one station sends a very long packet, the other stations will have to wait a long time before having the possibility to use the shared medium. This implies a longer waiting time for users and applications.	◯ small MTU ◯ big MTU ◯ incorrect
Collisions. When the trasmission of a packet ends, all the stations that were not transmitting due to carrier sense will try to transmit at the same time, leading to a collision with very high probability. The longer the transmitted packet, the higher is the probability that when the transmission ends multiple stations will have packets to send.	◯ small MTU ◯ big MTU ◯ incorrect
Efficiency. Since each packet (regardless of its length) contains a header, longer packets imply a more efficient use of the medium.	◯ small MTU ◯ big MTU ◯ incorrect
Memory. Packets must be held into the NIC memory. Longer packets imply NICs with more memory, resulting into more expensive hardware.	◯ small MTU ◯ big MTU ◯ incorrect
Network size. Transmitting stations can recognize a collision only if the transmission lasts enough for the first bit to arrive the farthest station in the network and return back. The bigger the MTU, the larger the network.	◯ small MTU ◯ big MTU ◯ incorrect
Level 3 constraints. There is not advantage at allowing bigger packets at level 2, since level 3 maximum transmission unit is usually too small to take advantage of it.	◯ small MTU ◯ big MTU ◯ incorrect
Router speed. Router throughput is higher if packets are smaller. Bigger packets would imply a slower Internet.	◯ small MTU ◯ big MTU ◯ incorrect

EXERCISE 5: Transmitting Maximum-length Packets

This exercise is meant to verify the understanding of the relationship between packet size, bit rate, and signal transmission speed.

Consider an IEEE 802.3 station that transmits at 10 Mbit/s a packet of maximum size ($1,518$ bytes). Recall that the preamble is 7 bytes and that the start-frame delimiter is one byte.

5.1) How long does it take for the station to transmit the packet?

5.2) Assume that the maximum-length packet is transmitted over a medium of infinite length. How long would the packet be? Put differently, how far would the first bit of the packet be from the last bit?

5.3) What are the analogous values for the minimum packet (64 bytes = 512 bit)?

EXERCISE 6: Efficiency of Channel Usage

This exercise aims at quantifying the protocol overhead of IEEE 802.3*.*

Consider an IEEE 802.3 station transmitting continuously at 10 Mbit/s. Recall that in IEEE 802.3 the preamble is 7 bytes, the start-frame delimiter is one byte, and the inter-packet gap is 12 bytes.

6.1) What is the maximum packet rate? That is, what is the maximum number of packets that a station can transmit per second? (Hint: recall that the minimum packet size is 64 bytes = 512 bits.)

6.2) How many maximum-size packets (1,518 bytes) can a station transmit per second?

6.3) What would be the throughput (bytes per seconds) carried in the packet's payload in the two cases?

6.4) What is the maximum efficiency of channel use? Compute the efficiency as the ratio between the payload throughput and the channel throughput.

EXERCISE 7: A Fake IEEE 802.3

This exercise is meant to verify the understanding of the relationship between packet length, bit rate, and network length.

You are designing a variant of the famous IEEE 802.3 protocol, called FAKE 802.3, having a bit rate of 100 Mbit/sec, a minimum packet size (preamble and CRC included) of 2000 bits, and a maximum packet size (preamble and CRC included) of 2000 bytes. All other characteristics are copied from IEEE 802.3.

7.1) Knowing that a signal propagates on the transmission medium at $200,000$ Km/sec speed, compute how many seconds must 2τ (round-trip delay) be for CSMA/CD to work correctly.

7.2) What is the maximum length of a collision domain in the FAKE 802.3 standard?

Chapter 3

Geographic Links

Stop-and-wait: It is a transmission approach that forces a sender to wait for an ack (or nak) on the previous packet before transmitting the next one. If does not get an acknowledgement within a fixed time called *timeout*, or if a negative acknowledgement is received, then the packet is resent.

Go-back-N: It is a transmission approach that allows a sender to transmit a certain number of packets continuously. If the sender does not get an acknowledgement for packet number N within the timeout, then packet N and all subsequent packets are retransmitted.

Selective repeat: It is a refinement of the go-back-N paradigm where only packets not acknowledged (or those for which a nak is received) are resent.

Stop-and-wait throughput: When no errors occur, this approach guarantees a throughput of $\frac{1}{t_T}$ packets per second, where t_T is the time between the beginning of the transmission of two consecutive packets.

Go-back-N throughput. Assuming that an arbitrary number of bits can be used to identify packets, this approach guarantees, when no errors occur, a throughput of $\frac{1}{t_I}$ packets per second, where t_I is the time needed to write a packet on the transmission medium. More realistically, if you assume to have a finite number of bits to number packets, say b bits, you have $M = 2^b$ different labels. In order to avoid ambiguity, you can send out at most $M - 1$ packets without receiving acks. Let t_p be the propagation delay, let t_s be the time needed to transmit an ack or nak, and let t_I the time needed to transmit a packet. If $2t_p + t_s > (M - 2)t_I$, then you can transmit $M - 1$ packets before receiving an ack (or nak) for the first packet. In this case, you are forced to stop transmitting until you receive the ack. Otherwise, you have continuous transmission. Accordingly, the throughput is different in the two cases, as shown in the following table.

	transmission	
	continuous	**non-continuous**
condition	$2t_p + t_s \leq (M-2)t_I$	$2t_p + t_s > (M-2)t_I$
throughput	$\frac{1}{t_I}$	$\frac{M-1}{2t_p+t_s+t_I}$

When the acknowledgements are piggybacked, you can assume $t_s = t_I$ and the above formulas simplify as follows:

	transmission	
	continuous	**non-continuous**
condition (piggybacking)	$2t_p + t_I \leq (M-2)t_I$	$2t_p + t_I > (M-2)t_I$
throughput (piggybacking)	$\frac{1}{t_I}$	$\frac{M-1}{2t_p+2t_I}$

EXERCISE 8: Is it a Big Deal?

This exercise aims at computing go-back-N throughput in different scenarios.

You are the network administrator of the A.C.M.E & Co. and you want to buy a point-to-point link to connect your local network to the Internet. You collect the following offers:

Offer 1	Offer 2	Offer 3
BIG DEAL A 1 Mbit/s 20 km link with 3 bits to count packets	Yes!! We are giving away transoceanic lines!!! 2000 Km long. 100 Mbit per seconds*. Completely yours for few bucks. * 3 bits to count packets	Joe's links: handmade links for all pockets. The famous Joe's original links: 2 Km long, 100 Kbit per second. You can take advantage of 4 bits to count your packets.

Suppose that the packets that A.C.M.E & Co. wants to send to the Internet are 250 Bytes large. Suppose to use go-back-N and piggy-backing acknowledgements. Answer to the following questions specifying the most relevant formulas you are using to obtain the asked values.

8.1) Compute $2t_p + t_I$ in the three cases.

Offer 1	Offer 2	Offer 3

8.2) How many labels do you have to identify your packets and how many of them you can use simultaneously without receiving any acknowledgement?

Offer 1	Offer 2	Offer 3

8.3) Are the bits sufficient to have a continuous transmission on the line?

Offer 1	Offer 2	Offer 3

8.4) How many packets per second can be transmitted in the three cases?

Offer 1	Offer 2	Offer 3

EXERCISE 9: High-delay link

This exercise aims at understanding the impact of high propagation delays on the throughput for the go-back-N approach.

You have to choose the level 2 features of a satellite link. Suppose the satellite is about $10,000$ km far away, that the bit rate is 1 Mbit/s, that the size of the used packets is $1,000$ Bytes, and that go-back-N with piggybacking acknowledgments is used. Answer to the following questions specifying also the most relevant formulas used to obtain the asked values. Assume that the propagation speed of the signal is $2/3$ of the speed of light in empty space.

9.1) Compute $2t_p + t_I$.

9.2) Compute the minimum number of bits to count packets that is needed to have continuous transmission.

9.3) How many packets per second can be transmitted assuming that the number of bits used count packets guarantees continuous transmission?

Chapter 4

Switches: Forwarding and Filtering

Switches' tasks. IEEE 802.1D-compliant switches (or bridges) are network devices that forward level 2 packets, called frames, between different collision domains. They have two main functions: first, by blocking some of their ports they cut the network in such a way that no loops are present, second they forward/filter traffic in such a way that packets are sent (if possible) only where they are needed.

The first task is the subject of Chapter 8, while this chapter contains exercises where switches are assumed to be connected in a tree-like network and addresses exclusively their forwarding/filtering ability.

Cut-through switches. It is important to keep in mind that switches are store-and-forward devices, that is, they memorize each packet they receive and delete it from their memory only when they have successfully forwarded it towards its destination(s). Commercial switches start sending the first bit of the packets out of the output port(s) only when the whole packet has been received in the input port. *Cut-through* switches, instead, try to forward the packet to the output port(s) before the whole packet has been received, which does not imply that the packet does not need to be kept in memory untill it has been successfully forwarded.

Filtering task. Another concept related with IEEE 802.1D switches is that of transparency: hosts in the network do not need to know if there are also switches and how many of them are there. This is very convenient since it implies that no configuration is needed on the host side. On the other hand, since hosts are not aware of the presence of switches, switches cannot rely on cooperation by hosts to filter the traffic to the right ports. In fact, they use the "experience" they gain while forwarding packets. They start forwarding packets to all ports (with the exception of the input one). When they receive a packet, though, they memorize that the sender is reachable via the input port. All the packets for that destination will be sent to that port only.

EXERCISE 10: Switches Vs Hubs

This exercise aims at highlighting the differences between switches and hubs.

Describe the main differences between a switch and a repeater or hub.

10.1) What is the network level at which they operate and what does this imply with regard to the information they handle?

10.2) When the above devices have multiple ports, where do they forward a received packet?

10.3) What strategy do the two devices apply when they detect a collision while transmitting a packet out of a port?

EXERCISE 11: Malfunctioning Switch

By hypotesizing various malfunctioning scenarios, this exercise verifies the understanding of how the different functions performed by a IEEE 802.1D switch relate to each other.

Consider a IEEE 802.1D-compliant switch S.

11.1) Suppose that S is broken. In particular, suppose that all its functions are properly working except its "learning" ability. Namely, S has a forwarding table that is always empty and each time a frame is received by a port, it is sent out through all other ports. Do you think that the network in which S is placed will work anyway? Do you think that S is able to correctly separate the collision domains? What are the consequences of this fault?

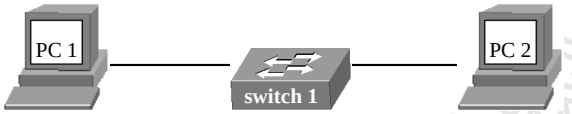
11.2) The vendor repairs S, however, a new malfunctioning turns out after a while. Now, all the functions of S are working as expected, but the output memory buffer has become volatile. In other words, Each time a bit is transmitted over the cable, its value is immediately forgotten. Do you think that the network in which S is placed will work anyway? Is S able to correctly separate the collision domains? What are the consequences of this fault?

11.3) The vendor is contacted again, and S gets repaired. Unfortunately, after a while a new type of fault happens (yes, we will blacklist that vendor). This time, the spanning tree protocol is broken, while all other functions are working properly. In particular, S is not able to switch the ports in blocking state when a cycle is detected. Do you think that the network in which S is will work anyway? Do you think that S is able to correctly separate collision domains? What are the consequences of this fault?

EXERCISE 12: Cut-through Switch and Collisions

This exercise is meant to verify the understanding of collisions where multiple Ethernet links are connected by a cut-through switch.

The following 10 Mb/sec network conforms to the IEEE 802.3 standard. PC 1 and PC 2 are connected to switch 1, which is a cut-through switch. Assume that the cut-through switch tries to forward the packet out of the destination port after receiving the preamble (7 byte), sfd (1 byte), and destination address (6 byte) fields.



PC 1 and switch 1 are 88 bit-time apart (1 bit-time is the time needed to transmit a single bit), while switch 1 and PC 2 are 100 bit-time apart. At time $t = 0$, PC 1 starts transmitting an 800 bit packet (preamble and sfd included) addressed to PC 2. At time $t = 250$ bit-time, PC 2 needs to send an 800 bit packet (preamble and sfd included) addressed to PC 1.

12.1) Is PC 2 able to start the transmission at $t = 250$ bit-time? Why?

12.2) What happens in the network at $t = 300$ bit-time?

12.3) What happens in the network at $t = 350$ bit-time?

12.4) Does PC 1's MAC layer force a rentransmission of the packet? Why?

12.5) Does switch 1's MAC layer force a rentransmission of the packet to PC 2? Why?

12.6) If switch 1 were not a cut-through switch, what would have happened instead?

Chapter 5

IP Addressing

IP Address: An IP address is a 4-byte number. It is almost always represented as 4 decimal numbers separated by dots (e.g., $192.168.1.1$). IP addresses are used to identify the end host at the network layer.

IP Prefix: A set of IP addresses that share a number of initial bits is known as an **IP prefix**. An IP prefix represents a network: the common bits (i.e., the prefix) identify the network while the remaining bits are used to identify the hosts. An IP prefix is represented by an IP address and its associated **netmask**.

Netmask: The netmask associated with an IP address allows you to determine what part of the IP address has to be considered the network address (i.e., the prefix) and what part of it is devoted to the hosts. A netmask is a 32-bit number, usually represented in dotted decimal format (the same format used for IP addresses). Bits that have a value of 1 mean that the corresponding position should be considered part of the network address, while bits that have a value of 0 mean that the corresponding position should be considered part of the host address. For this reason, all valid netmasks consist of a sequence of k ones followed by a sequence of $32 - k$ zeroes. Such a netmask means that the first k bits of its associated IP address are considered the network address, while the remaining $32 - k$ bits are considered the host address (hence, we can have 2^{32-k} different hosts address within the same IP prefix).

Two IP addresses belong to the same prefix if they have the same netmask and if their bitwise and with the netmask is the same. For example, assume we have two IP address $10.0.0.1$ and $10.0.0.100$, both with associated netmask $255.255.255.0$. The bitwise and in both cases yields $10.0.0.0$, so the two IP addresses are in the same network prefix. This is the algorithm that each host runs to determine whether the destination of a packet lies in the same network or not.

A brief representation for an IP address and its associated netmask is IP-address/netmask-length, e.g., $10.0.0.11/24$. This means that the IP address is $10.0.0.11$ and the netmask is composed by 24 ones and 8 zeroes, that is, $255.255.255.0$.

Special addresses: Given an IP address and its netmask, two special addresses can be derived: the **network address** is obtained by performing a bitwise and between the address and the netmask; the **broadcast address** is obtained by taking the network address and replacing each bit of the host address with a 1.

Example: from IP $10.1.2.3/24$ we can derive the network address $10.1.2.0$ and the broadcast address $10.1.2.255$.

Subnetting: By incrementing the number of ones of a netmask, you split a network into two subnets. For example $100.100.100.0/24$ becomes $100.100.100.0/25$ and $100.100.128.0/25$, where the first subnet contains the IP addresses from $100.100.100.0$ to $100.100.100.127$ and the second subnet contains the IP addresses from $100.100.100.128$ to $100.100.100.255$.

The following table shows how the 256 host addresses contained in a subnet having a $/24$ netmask ($255.255.255.0$ in dotted decimal notation) can be organized in two $/25$ subnets, four $/26$ subnets, and so on, until we reach sixty-four $/30$ subnets. Observe that a $/30$ subnet cannot be further split: a $/31$ subnet would only contain the network address and the broadcast address, so there would not be any address left to assign to hosts!

255.255.255.0 $1 \times \backslash 24$	255.255.255.128 $2 \times \backslash 25$	255.255.255.192 $4 \times \backslash 26$	255.255.255.224 $8 \times \backslash 27$	255.255.255.240 $16 \times \backslash 28$	255.255.255.248 $32 \times \backslash 29$	255.255.255.252 $64 \times \backslash 30$
x.y.z.0	x.y.z.0	x.y.z.0	x.y.z.0	x.y.z.0	x.y.z.0	x.y.z.0
						x.y.z.3
						x.y.z.4
					x.y.z.7	x.y.z.7
					x.y.z.8	x.y.z.8
						x.y.z.11
						x.y.z.12
				x.y.z.15	x.y.z.15	x.y.z.15
				x.y.z.16	x.y.z.16	x.y.z.16
						x.y.z.19
						x.y.z.20
					x.y.z.23	x.y.z.23
					x.y.z.24	x.y.z.24
						x.y.z.27
						x.y.z.28
			x.y.z.31	x.y.z.31	x.y.z.31	x.y.z.31
			x.y.z.32	x.y.z.32	x.y.z.32	x.y.z.32
						x.y.z.35
						x.y.z.36
					x.y.z.39	x.y.z.39
					x.y.z.40	x.y.z.40
						x.y.z.43
						x.y.z.44
				x.y.z.47	x.y.z.47	x.y.z.47
				x.y.z.48	x.y.z.48	x.y.z.48
						x.y.z.51
						x.y.z.52
					x.y.z.55	x.y.z.55
					x.y.z.56	x.y.z.56
						x.y.z.59
						x.y.z.60
		x.y.z.63	x.y.z.63	x.y.z.63	x.y.z.63	x.y.z.63
		x.y.z.64	x.y.z.64	x.y.z.64	x.y.z.64	x.y.z.64
						x.y.z.67
						x.y.z.68
					x.y.z.71	x.y.z.71
					x.y.z.72	x.y.z.72
						x.y.z.75
						x.y.z.76
				x.y.z.79	x.y.z.79	x.y.z.79
				x.y.z.80	x.y.z.80	x.y.z.80
						x.y.z.83
						x.y.z.84
					x.y.z.87	x.y.z.87
					x.y.z.88	x.y.z.88
						x.y.z.91
						x.y.z.92
			x.y.z.95	x.y.z.95	x.y.z.95	x.y.z.95
			x.y.z.96	x.y.z.96	x.y.z.96	x.y.z.96
						x.y.z.99
						x.y.z.100
					x.y.z.103	x.y.z.103
					x.y.z.104	x.y.z.104
						x.y.z.107
						x.y.z.108
				x.y.z.111	x.y.z.111	x.y.z.111
				x.y.z.112	x.y.z.112	x.y.z.112
						x.y.z.115

					x.y.z.119	x.y.z.116 x.y.z.119
					x.y.z.120	x.y.z.120 x.y.z.123
					x.y.z.127	x.y.z.124 x.y.z.127
					x.y.z.128	x.y.z.128 x.y.z.131
					x.y.z.135	x.y.z.132 x.y.z.135
					x.y.z.136	x.y.z.136 x.y.z.139
					x.y.z.143	x.y.z.140 x.y.z.143
					x.y.z.144	x.y.z.144 x.y.z.147
					x.y.z.151	x.y.z.148 x.y.z.151
					x.y.z.152	x.y.z.152 x.y.z.155
					x.y.z.159	x.y.z.156 x.y.z.159
					x.y.z.160	x.y.z.160 x.y.z.163
					x.y.z.167	x.y.z.164 x.y.z.167
					x.y.z.168	x.y.z.168 x.y.z.171
					x.y.z.175	x.y.z.172 x.y.z.175
					x.y.z.176	x.y.z.176 x.y.z.179
					x.y.z.183	x.y.z.180 x.y.z.183
					x.y.z.184	x.y.z.184 x.y.z.187
					x.y.z.191	x.y.z.188 x.y.z.191
					x.y.z.192	x.y.z.192 x.y.z.195
					x.y.z.199	x.y.z.196 x.y.z.199
					x.y.z.200	x.y.z.200 x.y.z.203
					x.y.z.207	x.y.z.204 x.y.z.207
					x.y.z.208	x.y.z.208 x.y.z.211
					x.y.z.215	x.y.z.212 x.y.z.215
					x.y.z.216	x.y.z.216 x.y.z.219
					x.y.z.223	x.y.z.220 x.y.z.223
					x.y.z.224	x.y.z.224 x.y.z.227
					x.y.z.231	x.y.z.228 x.y.z.231
					x.y.z.232	x.y.z.232 x.y.z.235
						x.y.z.236

				x.y.z.239	x.y.z.239	x.y.z.239
				x.y.z.240	x.y.z.240	x.y.z.240
						x.y.z.243
						x.y.z.244
					x.y.z.247	x.y.z.247
					x.y.z.248	x.y.z.248
						x.y.z.251
						x.y.z.252
x.y.z.255	x.y.z.255	x.y.z.255	x.y.z.255	x.y.z.255	x.y.z.255	x.y.z.255
$1 \times \backslash 24$	$2 \times \backslash 25$	$4 \times \backslash 26$	$8 \times \backslash 27$	$16 \times \backslash 28$	$32 \times \backslash 29$	$64 \times \backslash 30$
255.255.255.0	255.255.255.128	255.255.255.192	255.255.255.224	255.255.255.240	255.255.255.248	255.255.255.252

EXERCISE 13: Subnetting

This exercise aims at designing an IP addressing scheme with given constraints on the number of hosts for each LAN.

You have to assign IP addresses to the hosts of the Euro-Net company. The company has 5 distincs LANs, called A, B, C, D, and E. Each LAN has the number of hosts specified in the following table.

Lan A	Lan B	Lan C	Lan D	Lan E
20	20	100	20	20

Euro-Net asked and obtained from the RIPE European Authority the class C network 193.203.163.0 and now must split it into subnets and assign each subnet to a LAN.

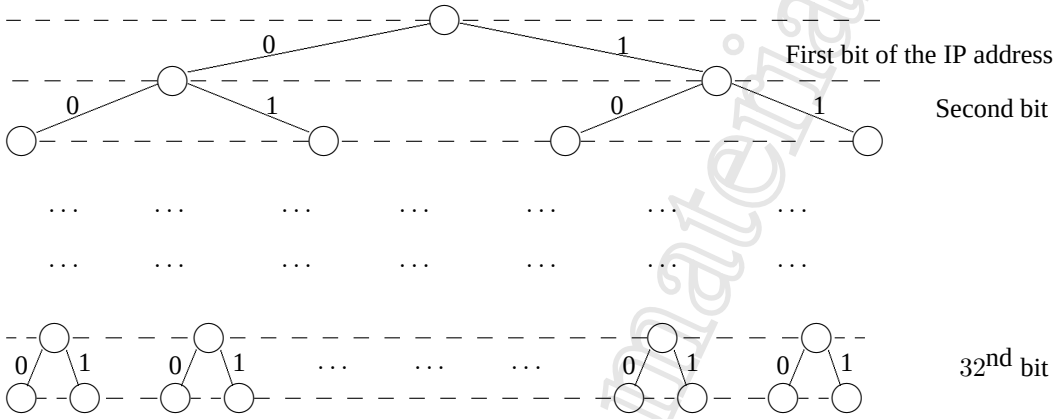
Fill in the following table specifying, for each subnet, the netmask and the corresponding broadcast addresses.

	Subnet	Netmask	Broadcast
Lan A			
Lan B			
Lan C			
Lan D			
Lan E			

EXERCISE 14: Subnets and binary trees

By representing an IP address with a binary tree, this exercise verifies the understanding of key concepts like subnets, prefixes, and overlapping prefixes.

IP address space can be represented as a binary tree whose edges are labeled either 0 or 1 (referred to as *tree representation*). The edge that connects the root of the tree with its left child is labeled 0 and means that the MSB (most significant bit, i.e., the first bit) of the address is 0. Similarly, the edge that connects the root of the tree with its right child is labeled 1 and means that the MSB of the address is 1. More generally, the edge connecting a node at distance d from the root with its left (right) child means that the $d+1^{\text{th}}$ bit of the address is 0 (1). The figure below depicts such a schema.



In the tree representation, there is a biunivocal correspondence between IP addresses and the leaves of the tree. Given a leave x, the corresponding IP address is obtained by concatenating the labels of the edges traversed by the path from the root to x.

14.1) In the tree representation, what corresponds to a net (or subnet)? What corresponds to the IP prefix of a net (or subnet)?

14.2) In the tree representation what corresponds to all the nets (or subnets) having a $/8$ prefix?

14.3) In the tree representation, given a net (or subnet) A and another net (or subnet) B, how is it possible to check whether A and B are overlapping?

14.4) In the tree representation, given a net (or subnet) A and another net (or subnet) B, how is it possible to check whether A and B have been obtained by subnetting a third net (or subnet) C?

14.5) In the tree representation, what corresponds to a routing table?

EXERCISE 15: Guessing Remote Prefixes with Local Netmasks

This exercise aims at verifying the understanding of subnets and netmasks.

Host PC X has IP address $100.100.100.100/16$ (with the netmask $255.255.0.0$) and has to send two packets to PC 1, whose IP address is $100.100.100.124$, and PC 2 whose IP address is $100.200.100.125$.

15.1) First, PC X sends the packet to PC 1 ($100.100.100.124$). PC X computes its own prefix as $100.100.0.0$ and the prefix for PC 1 as $100.100.0.0$. Can the actual prefix of PC 1 be $100.0.0.0$? Comment your answer.

15.2) Can the actual prefix of PC 1 be $100.100.100.0$? Comment your answer.

15.3) Second, PC X sends the packet to PC 2 ($100.200.100.125$). PC X computes its own prefix as $100.100.0.0$ and a prefix for PC 2 as $100.200.0.0$. Can the actual prefix of PC 2 be $100.0.0.0$? Comment your answer.

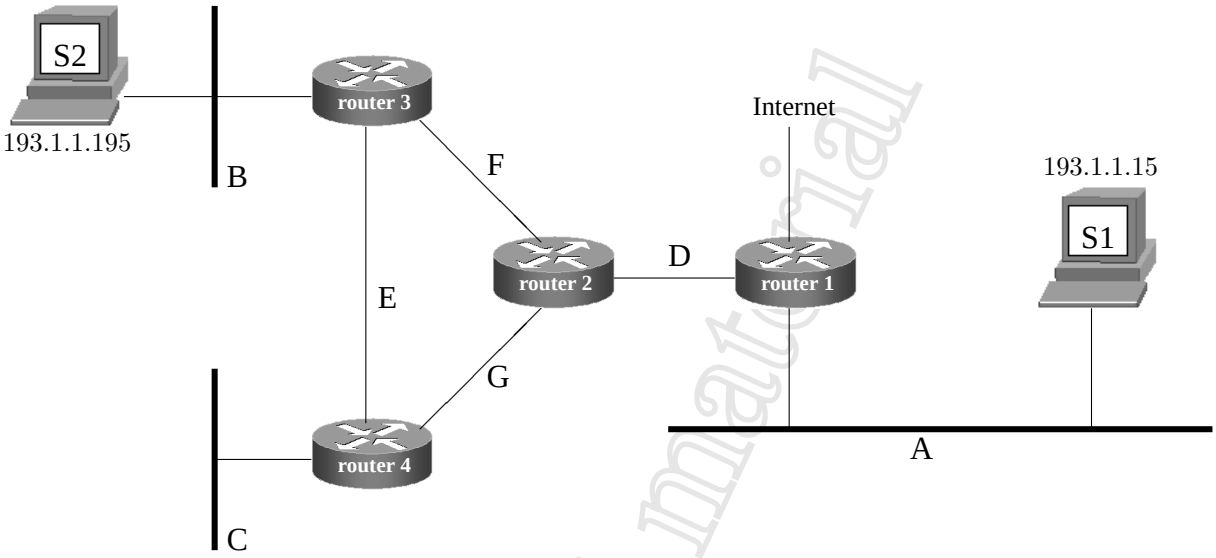
15.4) Can the actual prefix of PC 2 be $100.200.100.0$? Comment your answer.

15.5) Is there a case among the four cases mentioned in the previous questions where PC X risks to send the IP packet to the wrong recipient (for example to the default router instead of to the destination host or vice versa)? Comment your answer.

EXERCISE 16: Constrained Addressing Plan

This exercise is meant to verify the ability to design an IP addressing plan that matches the size of the network while satisfying some given addressing constraints.

YetAnother Corporation has just opened a new department, and installed a network whose topology is represented by the picture below.



The network administrator can use IP addresses from the network $193.1.1.0/24$. An IP address must be assigned to every host in LANs A, B, and C, as well as to point-to-point links D, E, F, and G.

For technical reasons, the following constraints are imposed over the network:

- S1, located in LAN A, is a web server, and its address must be $193.1.1.15$;
- LAN A must support 70 hosts, LAN B must support 40 hosts, LAN C must support 20 hosts;
- S2, located in LAN B, is a web server, and its address must be $193.1.1.195$;
- IP addresses in the range from (and including) $193.1.1.128$ and $193.1.1.143$ are reserved to future use and must not be assigned.

The network administrator cannot modify the network structure. Help him design a feasible IP address plan by answering the following questions.

16.1) Assign an IP prefix to LANs A, B, and C. Fill in the following table and specify for each LAN its subnet, its netmask, and the corresponding broadcast address.

LAN	subnet	netmask	broadcast
A			
B			
C			

16.2) Assign an IP prefix to point-to-point links D, E, F, and G. Fill in the following table and specify for each link its subnet, its netmask, and the corresponding broadcast address.

link	**subnet**	**netmask**	**broadcast**
D			
E			
F			
G			

Chapter 6

IP Routing: Packet Forwarding

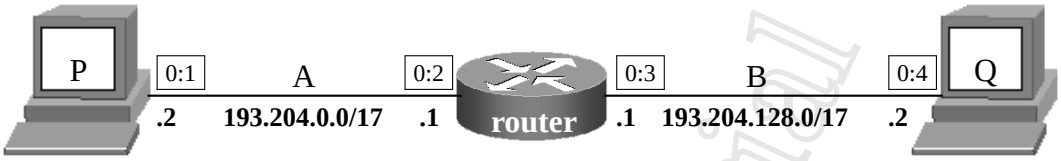
All the excercises of this chapter focus on the forwarding ability of the routers.

Forwarding mechanism: In order to forward packets, a router checks the destination IP address of the packets against each line of its routing table. Given a line of the routing table and the destination address, a bitwise AND operation is performed between the address and the "netmask" field. If the result is exactly what specified in the "network" field, then the packet has to be sent out from the interface specified in the "interface" field. If the value of the "Next hop" field is "Directly connected", then the router will send the IP packet into a level 2 frame sent directly to the destination host. Otherwise it will enclose the IP packet into a level 2 frame sent to the interface having the specified IP address.

EXERCISE 17: Forwarding with wrong subnets

This exercise is meant to understand how packet forwarding is impacted by wrong subnets.

In the network diagram below, dotted numbers (e.g., **.11**) represent the last byte of an IP address, while boxed numbers (e.g., 0:2) represent a MAC address assigned to a network interface. All links are IEEE 802.3, working at 1 Gbit/s. There are no other machines in the network. The routing table of the router is correctly configured.



17.1) At a certain time, after a long period of inactivity, a user on machine P runs the command `ping 193.204.128.2`. For the sake of brevity, assume that the `ping` command only produces one packet. List the packets that a sniffer placed in observation point A sees on the network.

dst MAC	**src MAC**	**src IP** (if any)	**dst IP** (if any)	**packet type** (one of the following: arp request, arp reply, icmp echo-request, icmp echo-reply, icmp error)

Similarly, list the packets that a sniffer placed in observation point B sees on the network.

dst MAC	**src MAC**	**src IP** (if any)	**dst IP** (if any)	**packet type** (one of the following: arp request, arp reply, icmp echo-request, icmp echo-reply, icmp error)

17.2) At a given time, after a long period of inactivity, a user on machine P runs the command `ping 193.204.128.8`. For the sake of brevity, assume that the `ping` command only produces one packet. List the packets that a sniffer placed in observation point B sees on the network.

dst MAC	**src MAC**	**src IP** (if any)	**dst IP** (if any)	**packet type** (one of the following: arp request, arp reply, icmp echo-request, icmp echo-reply, icmp error)

17.3) Now assume that the administrator of machine P made a configuration error: he specified $255.255.0.0$ as the netmask. At a given time, after a long period of inactivity, a user on machine P runs the command `ping 193.204.128.2`. For the sake of brevity, assume that the `ping` command only produces one packet. List the packets that a sniffer placed in observation point A sees on the network.

dst MAC	**src MAC**	**src IP** (if any)	**dst IP** (if any)	**packet type** (one of the following: arp request, arp reply, icmp echo-request, icmp echo-reply, icmp error)

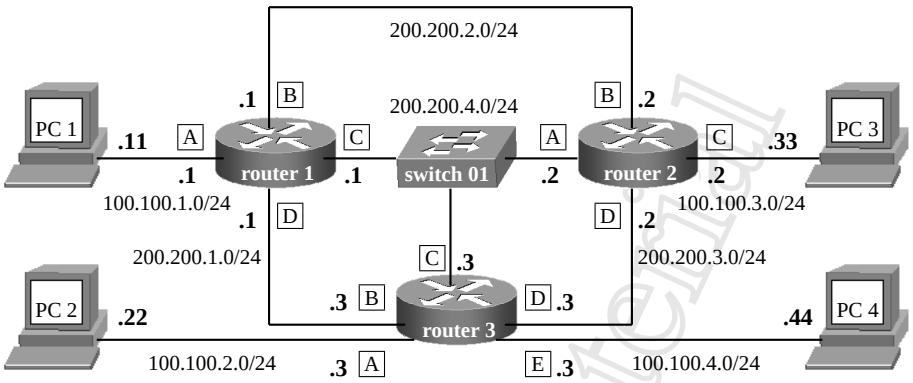
17.4) Assume you are in the same situation as in the previous exercise. At a given time, after a long period of inactivity, a user on machine Q runs the command `ping 193.204.0.2`. For the sake of brevity, assume that the `ping` command only produces one packet. List the packets that a sniffer placed in observation point A sees on the network.

dst MAC	**src MAC**	**src IP** (if any)	**dst IP** (if any)	**packet type** (one of the following: arp request, arp reply, icmp echo-request, icmp echo-reply, icmp error)

EXERCISE 18: Echo request and echo reply

This exercise is meant to verify the understanding of packet forwarding when routing tables are already configured.

In the network below boxed capital letters (e.g., A) denote interfaces, while dotted numbers (e.g., **.1**) denote the decimal value of the last byte of their IP address.



The router forwarding tables are as follows:

router 1			
Subnet	**Netmask**	**Interface**	**Next hop**
100.100.1.0	255.255.255.0	A	Directly connected
100.100.2.0	255.255.255.0	C	200.200.4.3
200.200.1.0	255.255.255.0	D	Directly connected
200.200.2.0	255.255.255.0	B	Directly connected
200.200.4.0	255.255.255.0	C	Directly connected
0.0.0.0	0.0.0.0	B	200.200.2.2

router 2			
Subnet	**Netmask**	**Interface**	**Next hop**
100.100.3.0	255.255.255.0	C	Directly connected
100.100.4.0	255.255.255.0	D	200.200.3.3
200.200.2.0	255.255.255.0	B	Directly connected
200.200.3.0	255.255.255.0	D	Directly connected
200.200.4.0	255.255.255.0	A	Directly connected
0.0.0.0	0.0.0.0	B	200.200.2.1

router 3			
Subnet	**Netmask**	**Interface**	**Next hop**
100.100.1.0	255.255.255.0	C	200.200.4.1
100.100.2.0	255.255.255.0	A	Directly connected
100.100.3.0	255.255.255.0	C	200.200.4.2
100.100.4.0	255.255.255.0	E	Directly connected
200.200.1.0	255.255.255.0	B	Directly connected
200.200.3.0	255.255.255.0	D	Directly connected
200.200.4.0	255.255.255.0	C	Directly connected

18.1) Suppose to run a `ping` on PC 1 towards PC 2. Which routers are traversed (and in what order) by echo request packets and which routers are traversed by echo reply packets?

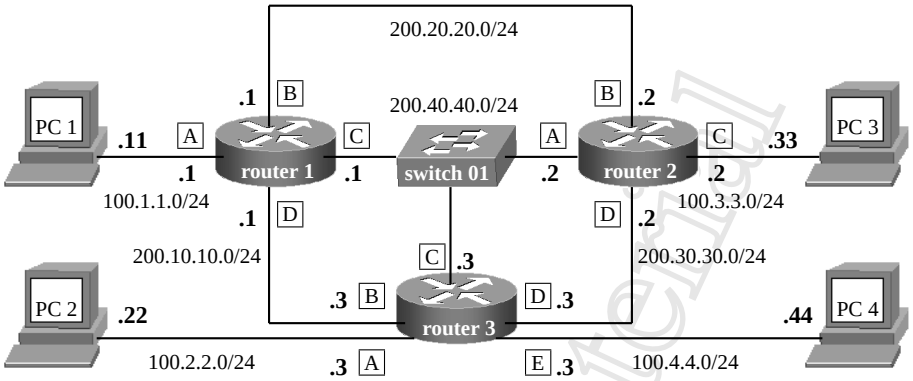
18.2) Suppose to run a `ping` on PC 3 towards PC 2. Which routers are traversed (and in what order) by echo request packets and which routers are traversed by echo reply packets?

18.3) Suppose to run a `ping` on PC 4 towards PC 1. Which routers are traversed (and in what order) by echo request packets and which routers are traversed by echo reply packets?

EXERCISE 19: Traceroute

This exercise is a variant of the previous one. Here, the student is asked to simulate traceroutes in the network.

In the network below boxed capital letters (e.g., A) denote interfaces, while dotted numbers (e.g., **.1**) denote the decimal value of the last byte of their IP address.



The routing tables of the three routers are as follows (directly connected subnet are not shown since their routing is trivial).

router 1

Network	Netmask	Interface	Next hop
100.2.2.0	255.255.255.0	C	200.40.40.2
0.0.0.0	0.0.0.0	D	200.10.10.3

router 2

Network	Netmask	Interface	Next hop
100.1.1.0	255.255.255.0	A	200.40.40.3
100.4.4.0	255.255.255.0	B	200.30.30.3
0.0.0.0	0.0.0.0	D	200.20.20.1

router 3

Network	Netmask	Interface	Next hop
100.1.1.0	255.255.255.0	B	200.10.10.1
100.3.3.0	255.255.255.0	C	200.40.40.2
0.0.0.0	0.0.0.0	D	200.30.30.2

19.1) What is the output of the traceroute command from PC 4 to PC 2?

```
user@PC4 ~> traceroute 100.2.2.22
```

19.2) What is the output of the traceroute command from PC 3 to PC 4?

user@PC3 ∼> traceroute 100.4.4.44

19.3) What is the output of the traceroute command from PC 3 to PC 1?

user@PC3 ∼> traceroute 100.1.1.11

19.4) What is the output of the traceroute command from PC 3 to PC 2?

user@PC3 ∼> traceroute 100.2.2.22

19.5) What is the output of the traceroute command from PC 1 to a non-existent host (for example 100.5.5.55)?

user@PC1 ∼> traceroute 100.5.5.55

Chapter 7

Distance Vector Routing Protocols

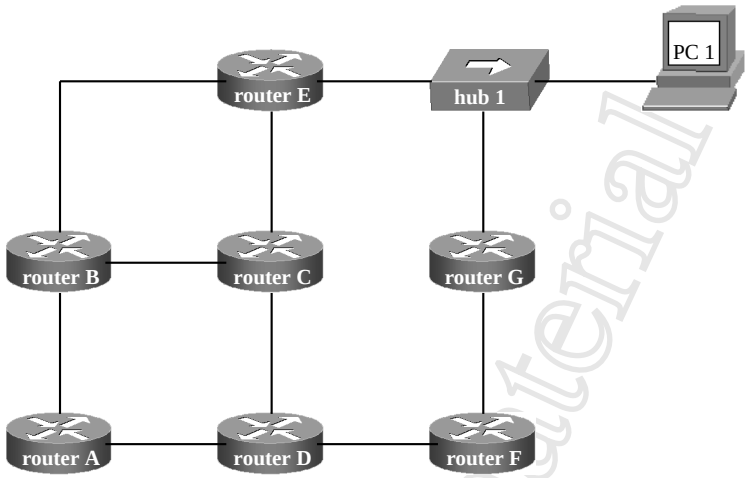
Routing Protocol: For correctly delivering packets, routers in a network need to build and maintain routing tables that are consistent with the network topology. Although manually configuring routing tables is possible and is sometimes done when the topology is trivial, networks events are usually too rapid for human intervention to be in the loop. It is more desirable to have cooperating routers that are able to keep their routing tables updated. The set of information that routers exchange and the technique they use to update their routing tables form a **routing protocol**.

Distance Vector Routing Protocol: It is a routing protocol where each router periodically exchanges **distance vectors** with every neighboring router. A distance vector is essentially a two-column table that maps each destination (e.g., an IP prefix) to a distance metric (e.g., the number of hops to reach the destination). By collecting distance vectors from every neighbor, each router is able to build its routing table using the Bellman-Ford algorithm for distributed shortest path computation.

EXERCISE 20: Sniffing Distance-Vectors

This exercise aims at verifying the understanding of the relationship between distance-vector routing protocols and the network topology.

The diagram below represents an operational network. Some days ago, some links of the network were accidentally cut.



The network uses a hop-based distance-vector routing protocol. The network administrator is running a sniffer on PC 1, trying to understand what is going on. The sniffer captures the following two distance-vectors.

from E to G

target	hops
A	3
B	2
C	1
D	3
E	0
F	2
G	1

from G to E

target	hops
A	3
B	3
C	2
D	2
E	1
F	1
G	0

Based on the sniffed distance-vectors, try to understand which links were cut and which are working properly. Then, update the diagram by putting a cross on the broken links.

Chapter 8

Switches: Spanning Tree Algorithm

STA input: Switches run a spanning tree algorithm (STA) in order to prevent loops in the topology. The algorithm is based on the following inputs:

- A topology consisting of switches whose ports are connected to collision domains (sometimes improperly called "LANs"). Hosts possibly connected to the collision domains are generally not described since they are not involved in the computation.
- An 8-bytes *bridge-id* for each switch. The bridge-id is the concatenation of the bridge priority (2 bytes, specified by the administrator) and mac address of its first port (6 bytes). The default value for the priority is 8000 (which is the intermediate value between 0000 and $FFFF$).
- A 2-bytes *port-id* for each port of each switch. The port-id is the concatenation of the port priority (1 byte, specified by the administrator) and the port number (1 byte). The default value for the port priority is 80 (the intermediate value between 00 and FF).
- A *cost* for each port. The cost is meant to represent the cost of the *incoming* traffic. Default values are obtained from the inverse of the bit rate.

STA steps: Although the spanning tree algorithm actually is not subdivided in phases, in order to understand its behaviour it is convenient to pretend that its computation is carried out in parallel by all the switches of the network by traversing, simultaneously across the network, four successive steps.

(i) **Root-bridge election:** a single switch is selected to be the root of the tree

(ii) **Root-port identification:** each switch, with the exception of the root-bridge, chooses the port "nearest" to the root-bridge.

(iii) **Designated-port selection:** for each collision domain, one of the ports connecting it to the switches is chosen as the *designated-port*, that is, the port that will inject into the domain the packets coming from the root bridge.

(iv) **Blocking of redundant ports:** ports that are not root-ports nor designated-ports are put in blocking state by the switches.

This steps are detailed in the following.

Root-bridge election:

(i) each switch sends a configuration BPDU (Bridge PDU) where its own bridge-id is proposed as root-identifier

(ii) when a switch receives a configuration BPDU with a lower value of bridge-id, it stops producing configuration BPDU with its own bridge-id, and it starts propagating the new configuration BPDU out of all its ports

(iii) the root-bridge is the switch that never stops producing configuration BPDUs with its bridge-id in the root-identifier field

Root-port identification: Eac non-root bridge selects a *root-port* by running a deterministic decision process based on lowest cost and a number of ways to break ties. The root-port of a (non-root) bridge is the port that receives the configuration BPDUs such that:

(i) the sum of the cost of the receiving port and the root-path-cost advertised in the BPDU is the lowest

(ii) in case of a tie, the bridge-identifier specified in the BPDU is the lowest

(iii) in case of a tie, the port-identifier specified in the BPDU is the lowest

(iv) in case of a tie, the port-identifier of the receiving port is the lowest

Designated-port selection: STA must ensure that each collision domain is served by exactly one port of one switch. In order to do so, switches run a deterministic decision process based on lowest cost and a number of ways to break ties. All ports of all switches that are connected to the same collision domain send (and receive) BPDUs. The designated-port of a domain is the one having

(i) lowest root-path-cost

(ii) in case of a tie, lowest bridge-identifier

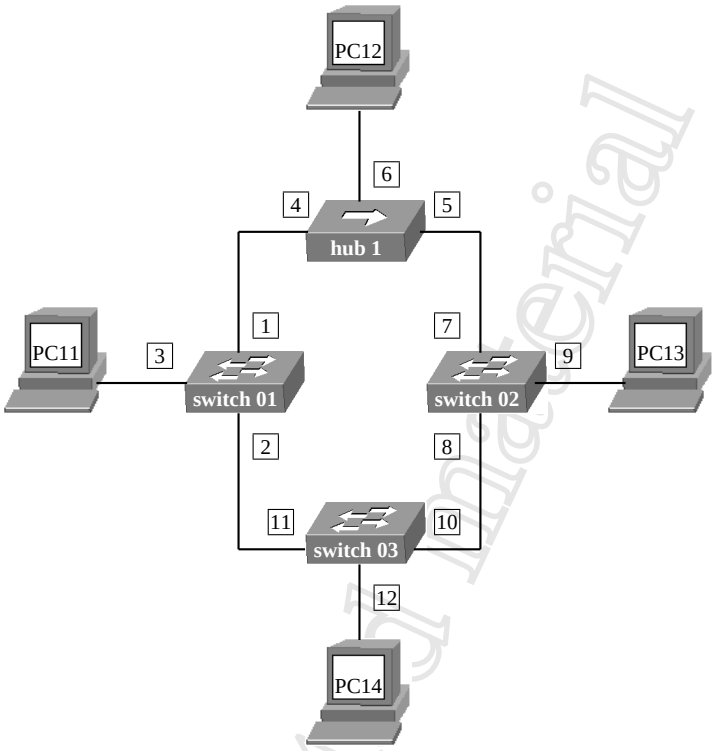
(iii) in case of a tie, lowest port-identifier

Blocking of redundant ports. All non-root-ports and non-designated-ports are put in blocking state. This ensures that active ports can never result in a cycle.

EXERCISE 21: Switches and Hubs

This exercise aims at verifying the understanding of how the spanning tree algorithm works and how forwardin tables of the switches are maintained.

Consider the following network with 3 802.1D switches, a hub, and 4 PCs.



Encoded in the name of each PC is the MAC address of its NIC (only the last two digits of it, for example "PC11" has MAC address "11"). Analogously, the name of each switch encodes the MAC address of its first port (for example, "switch 01" has MAC address "01"). Boxed numbers near the ports of the switches represent port numbers (for example, 12). Bridge priority and port priority is the same for all the switches and all the ports. All switch ports have cost 100.

21.1) Which switch is the root bridge and why?

21.2) Compute the spanning tree. Complete the figure above writing a small "d" near designated ports, circling root ports, and crossing ports in blocking state.

21.3) What role does hub 1 play in the spanning tree computation?

21.4) Supposing that all PCs constantly exchange packets, fill in the forwarding tables of the four switches (ignore BPDU traffic):

Switch 1		
port 1	port 2	port 3

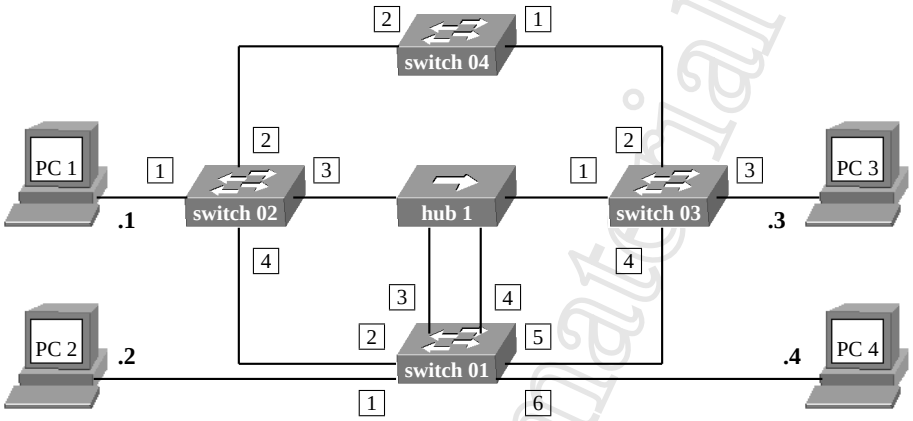
Switch 2		
port 7	port 8	port 9

Switch 3		
port 10	port 11	port 12

EXERCISE 22: Switches, Hosts and Hubs

This exercise is a variant of the previous one. Its purpose is to verify the understanding of how the spanning tree algorithm works.

In the network diagram below, the name of each switch represents its bridge-id (e.g., switch 01 has bridge-id "01"), while port numbers are represented by boxed numbers (e.g., [1]). Dotted numbers (e.g., **.1**) represent the last byte of the IP address assigned to a network interface. All bridge priorities are administratively set to zero. Also, all costs and port priorities are set to zero.



22.1) Which one is the root-bridge and why? By the way: is it true that the root-bridge never has a port in blocking state?

22.2) Which is the root-port of switch 02 and why?

22.3) Which is the root-port of switch 04 and why?

22.4) Cross the ports blocked by the spanning tree algorithm.

22.5) Which is the designated-port of the collision domain represented by hub 1? Why?

Chapter 9

Routing: Classless Inter-Domain Routing

Supernetting. Classless Inter-Domain Routing (CIDR) is a technique used to compact routing tables. The principle on which it is based is very similar to that allowing subnetting. In fact, the operation performed by CIDR is exactly the opposite of that performed bu subnetting, and is sometimes called "supernetting". By incrementing the number of ones of a netmask, a subnetting operation splits a network into two subnets. For example $100.100.100.0/24$ can be split into two subnets $100.100.100.0/25$ and $100.100.128.0/25$. In contrast, supernetting decrements the number of ones of a netmask, merging two subnets. For example: if you have the two networks $100.100.100.0/25$ and $100.100.128.0/25$ having the same interface I and next hop NH in your routing table, they can be replaced by a single line having $100.100.100.0/24$ as the prefix, I as the interface, and NH as the next hop. This operation reduces the routing table size.

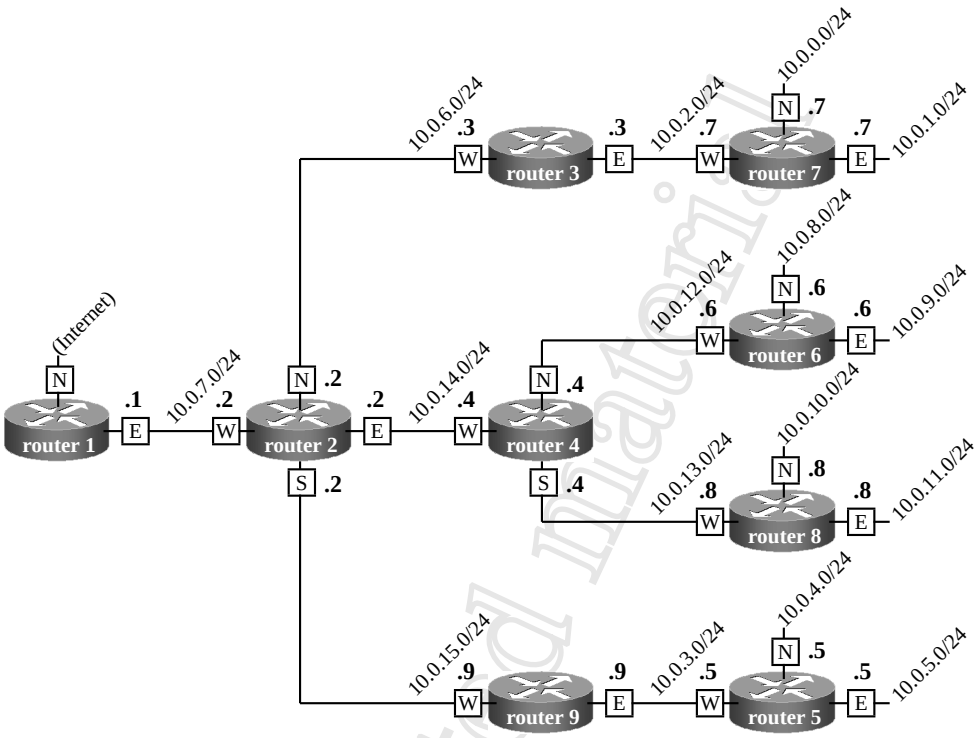
CIDR constraints. A legitimate CIDR compression takes a routing table, and produces as output a smaller routing table. To ensure that packet forwarding is not affected by the compression, it must be possible to suitably perform subnetting on the resulting routing table and obtain the original routing table. For this to be true, the following constraints must be satisfied:

- Only consecutive prefixes can be contracted, that is, CIDR cannot compact a set of prefixes which do not span a legitimate prefix. E.g., it is not possible to compress $100.100.0.0/26$, $100.100.64.0.0/26$, and $100.100.192.0/26$ into $100.100.0.0/24$, since the three input prefixes do not span the output prefix (there is a "hole")
- Only lines with the same interface and next-hop values can be contracted in a single line
- Only 2^k lines can be contracted into a single line ($k \in \{1, 2, \ldots\}$)
- If you contract the 2^k lines X.Y.Z_1.0, X.Y.Z_2.0, …, X.Y.Z_{2^k}.0 into X.Y.Z.0, then Z must be a multiple of 2^k.

EXERCISE 23: CIDR Scenario 1

This exercise aims at building routing tables given a network topology, and then compressing them with CIDR.

In the network below, the last byte of the IP addresses of routers' interfaces is represented by a dotted number (e.g., **.1**), while routers' interfaces are labeled with boxed letters (e.g., N).



23.1) Fill in the routing table of router 7 once compressed with CIDR.

Subnet Address	Netmask	Interface	Next-hop

23.2) Fill in the routing table of router 1 once compressed with CIDR.

Subnet Address	Netmask	Interface	Next-hop

23.3) Fill in the routing table of router 4 once compressed with CIDR.

Subnet Address	Netmask	Interface	Next-hop

23.4) Can router 1 announce all the networks in the diagram with a single routing line? How?

EXERCISE 24: CIDR Compaction 1

This exercise and the following ones are meant to verify the ability of the student to compress routing tables with CIDR, without knowing the network topology.

Show how the following routing table may be compressed by using CIDR:

Subnet Address	Netmask	Interface
130.30.0.0	255.255.0.0	int 1
130.31.0.0	255.255.0.0	int 1
130.32.0.0	255.255.0.0	int 1
130.33.0.0	255.255.0.0	int 1
130.34.0.0	255.255.0.0	int 1
130.35.0.0	255.255.0.0	int 1
130.36.0.0	255.255.0.0	int 1
130.37.0.0	255.255.0.0	int 1
130.38.0.0	255.255.0.0	int 1
130.39.0.0	255.255.0.0	int 1
130.40.0.0	255.255.0.0	int 1
130.41.0.0	255.255.0.0	int 2

Subnet Address	Netmask	Interface

EXERCISE 25: CIDR Compaction 2

Show how the following routing table may be compressed by using CIDR:

Subnet Address	Netmask	Interface
194.100.0.0	255.255.255.0	int 1
194.100.1.0	255.255.255.0	int 1
194.100.2.0	255.255.254.0	int 1
194.100.4.0	255.255.252.0	int 1
194.100.8.0	255.255.248.0	int 1
194.100.48.0	255.255.240.0	int 1
194.100.64.0	255.255.240.0	int 1

Subnet Address	Netmask	Interface

EXERCISE 26: CIDR Compaction 3

Show how the following routing table may be compressed by using CIDR:

Subnet Address	Netmask	Interface
194.38.40.0	255.255.255.0	int 1
194.38.41.0	255.255.255.0	int 1
194.38.42.0	255.255.254.0	int 1
194.38.44.0	255.255.252.0	int 1
194.38.48.0	255.255.254.0	int 1
194.38.50.0	255.255.255.0	int 1
194.38.51.0	255.255.255.0	int 1

Subnet Address	Netmask	Interface

EXERCISE 27: CIDR Compaction 4

Show how the following routing table may be compressed by using CIDR:

Subnet Address	Netmask	Interface
193.205.3.0	255.255.255.0	int 2
193.205.4.0	255.255.255.0	int 2
193.205.5.0	255.255.255.0	int 4
130.200.0.0	255.254.0.0	int 3
130.202.0.0	255.254.0.0	int 3
190.204.118.0	255.255.254.0	int 1
190.204.120.0	255.255.252.0	int 1
190.204.124.0	255.255.252.0	int 1

Subnet Address	Netmask	Interface

EXERCISE 28: CIDR Compaction 5

Show how the following routing table may be compressed by using CIDR:

Subnet Address	Netmask	Interface
140.38.0.0	255.255.0.0	int 1
140.39.0.0	255.255.0.0	int 1
140.40.0.0	255.255.0.0	int 1
140.41.0.0	255.255.0.0	int 1
140.42.0.0	255.255.0.0	int 1
140.43.0.0	255.255.0.0	int 1
140.44.0.0	255.255.0.0	int 1
140.45.0.0	255.255.0.0	int 1
140.46.0.0	255.255.0.0	int 2
140.47.0.0	255.255. 0.0	int 1
140.48.0.0	255.255. 0.0	int 1

Subnet Address	Netmask	Interface

EXERCISE 29: CIDR Compaction 6

Show how the following routing table may be compressed by using CIDR:

Subnet Address	Netmask	Interface
194.39.0.0	255.255.0.0	int 2
194.40.0.0	255.255.128.0	int 2
194.40.128.0	255.255.128.0	int 2
194.41.0.0	255.255.192.0	int 2
194.41.64.0	255.255.192.0	int 2
194.41.128.0	255.255.192.0	int 2
194.41.192.0	255.255.192.0	int 2
194.42.0.0	255.254.0.0	int 2
194.44.0.0	255.255.0.0	int 2

Subnet Address	Netmask	Interface

EXERCISE 30: CIDR Compaction 7

Show how the following routing table may be compressed by using CIDR:

Subnet Address	Netmask	Interface
20.0.1.0	255.255.255.128	int 1
20.0.1.128	255.255.255.128	int 1
20.0.2.0	255.255.255.0	int 1
20.0.3.0	255.255.255.0	int 1
20.0.4.0	255.255.255.0	int 1
20.0.5.0	255.255.255.0	int 1
20.0.6.0	255.255.255.0	int 1
20.0.7.0	255.255.255.0	int 1
20.0.8.0	255.255.255.0	int 1

Subnet Address	Netmask	Interface

Chapter 10

Traffic Shaping with Leaky and Token Buckets

Congestion: As more and more packets are sent over a network, routers need to store more and more data. Since routers only have a limited amount of available memory, the queue made up by packets waiting to be delivered on a link might exceed the memory capacity of the router, forcing some packets to be dropped. As a consequence, the average time needed to deliver a packet increases, while the percentage of delivered packets out of sent ones decreases. When this performance degradation is caused by the presence of too many packets in the network, we speak about *congestion*.

Traffic Shaping: The most common behavior that results in a congestion is the bursty nature of network traffic. Hosts usually produce bursts of data at the maximum speed allowed by the network interface card, and then sit back for a relatively large amount of time before sending additional traffic. This behavior causes a high **maximum** packet rate despite a moderate **average** packet rate, hence exacerbating network congestion. Regulating the average rate and burstiness of traffic is an activity known as *traffic shaping*.

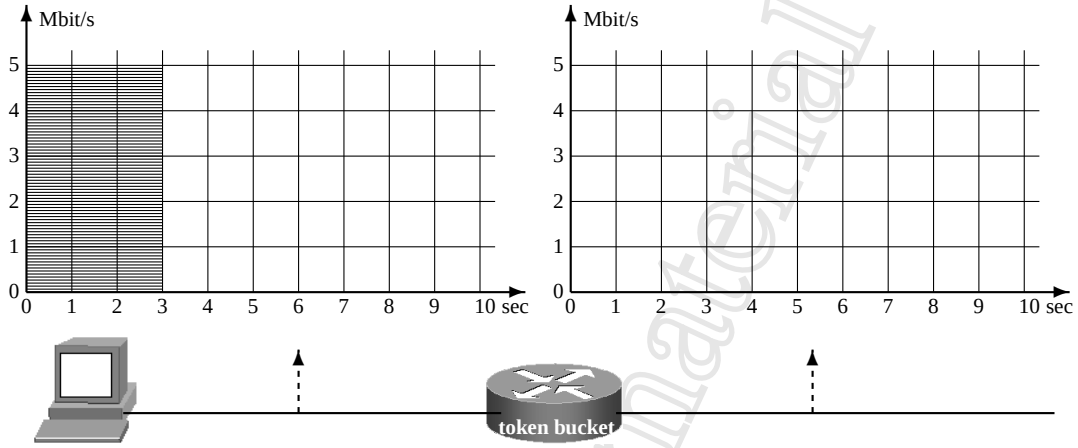
Leaky Bucket Algorithm: A *leaky bucket* has the property that water drips out of the bucket at a constant rate, no matter the rate at which water enters the bucket. In computer networks, a leaky bucket is simply a device that is able to buffer received packets in a queue, and then send them at a constant rate. The main characteristics of a leaky bucket are the output data rate M (expressed in bits/s) and the memory capacity C (expressed in bits). When the capacity of the leaky bucket is exceeded, packets are immediately dropped.

Token Bucket Algorithm: In contrast to a leaky bucket, a *token bucket* allows bursts to pass through the device, but only for a controlled maximum burst lenght. The main characteristics of a token bucket are the maximum output data rate M, the token capacity C, and the *token* rate ρ, expressed in bits/s ($\rho < M$). The bucket collects ρ tokens per second up to a maximum of C bits. Each bit of a token can be "spent" to send a bit in output at a rate of M bits/s. When the bucket has no tokens, the output data rate coincides with the token rate ρ. In contrast, if the bucket has some tokens, the output data rate is M, until the bucket runs out of tokens. The maximum time in which a bucket runs out of tokens coincides with the maximum length of a transmission burst at the maximum output rate, and it can be computed as $S = \frac{C}{M-\rho}$ (the result is expressed in seconds). Essentially, a token bucket transmits bursts at rate M for a maximum of S seconds, and then transmits data at a constant rate ρ.

EXERCISE 31: Token and Leaky Bucket

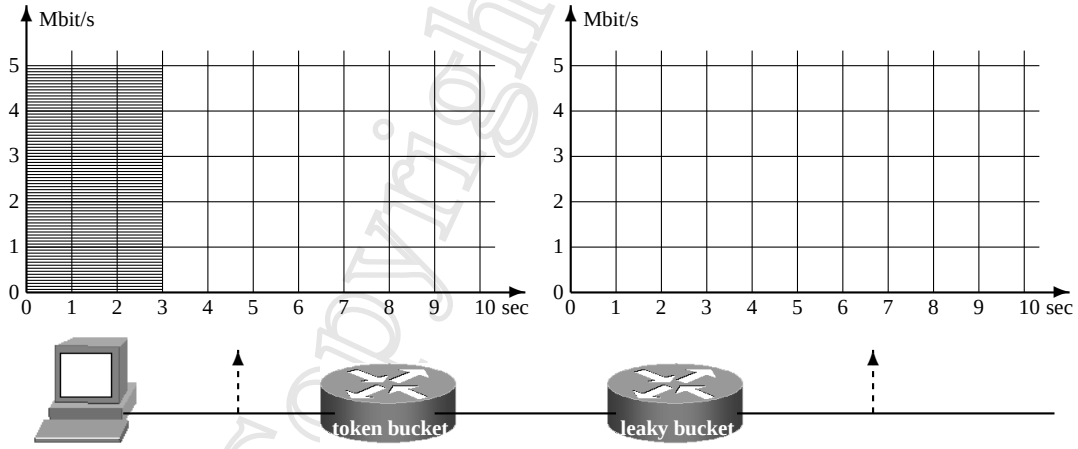
This exercise and the following ones are meant to verify the understanding of the traffic shaping actions that leaky and token buckets perform in a network, and their effect when used in combination.

A user buys a 5 Mbit/s line from a provider who puts a bit-oriented token bucket in the middle. The token bucket has a token frequency of 1 Mbit/s and a token capacity of 5 Mbit. The first plot below represents the traffic produced by the user. **31.1)** Draw the plot of the traffic exiting the token bucket.



Output freq.: 5 Mbit/s
Token freq.: 1 Mbit/s
Token capacity: 5 Mbit

31.2) Suppose that the provider changes the configuration putting a bit-oriented leaky bucket with capacity 9 Mbit and output frequency 2 Mbit/s immediately after the token bucket. Draw the plot of the traffic exiting the leaky bucket.



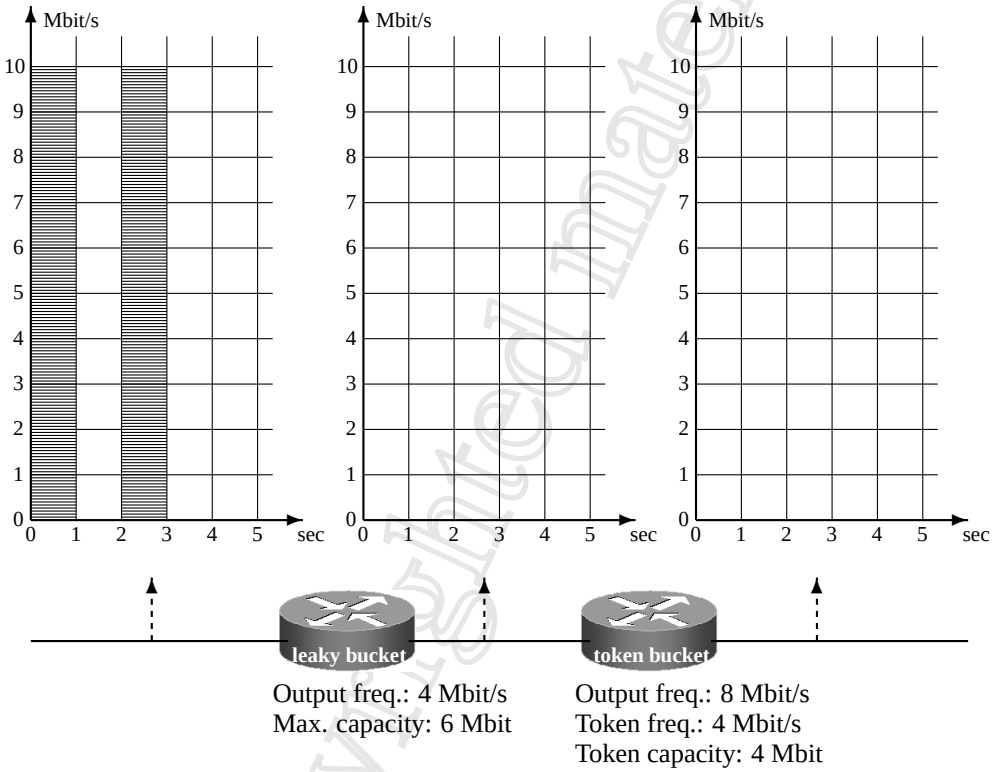
Output freq.: 5 Mbit/s
Token freq.: 1 Mbit/s
Token capacity: 5 Mbit

Output freq.: 2 Mbit/s
Max. capacity: 9 Mbit

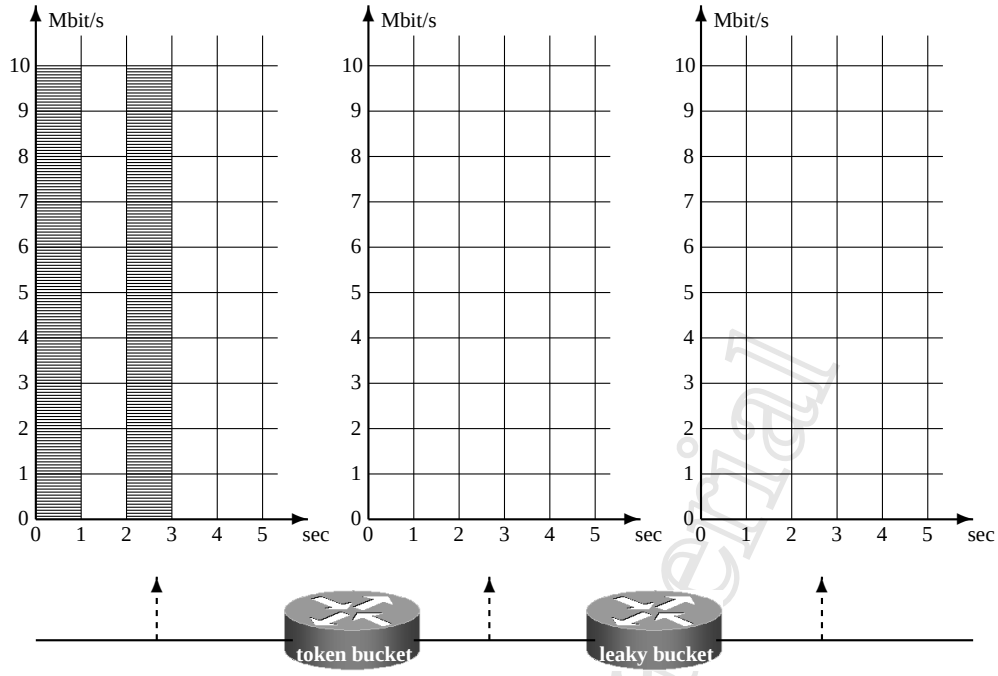
EXERCISE 32: Permutation of Leaky and Token Bucket

The first plot below represents the traffic produced by an application. Such a traffic is first filtered by a bit-oriented leaky bucket with an output frequency of 4 Mbit/s and maximum capacity 6 Mbit. Then, it is filtered by a bit-oriented token bucket with a token frequency of 4 Mbit/s, an output frequency of 8 Mbit/s and a token capacity of 4 Mbit.

32.1) Draw the plots of the traffic exiting the leaky bucket and the token bucket.



32.2) Suppose to swap the positions of the leaky and token buckets. Draw the plots of the traffic exiting the token bucket and the leaky bucket.



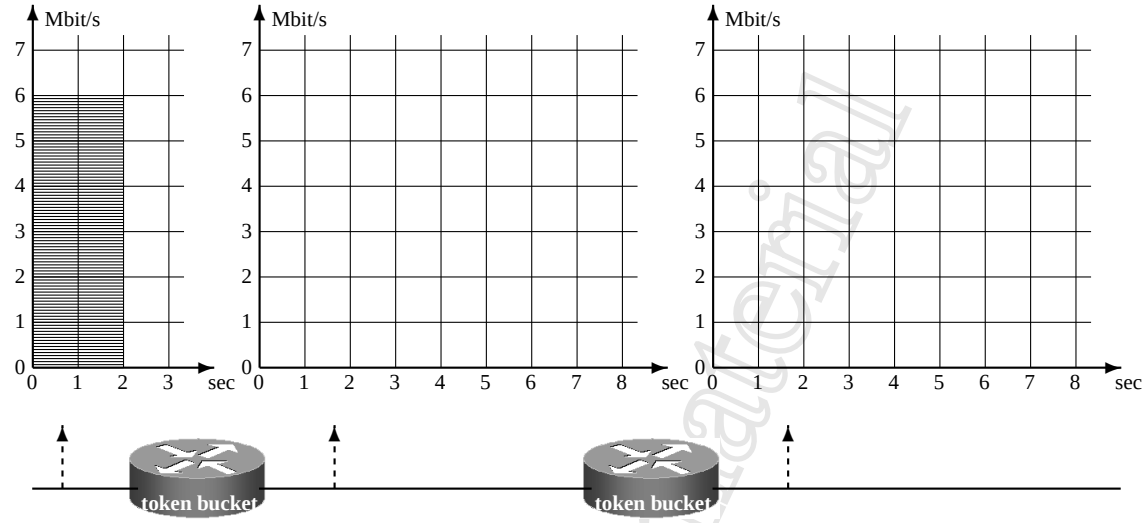
Output freq.: 8 Mbit/s
Token freq.: 4 Mbit/s
Token capacity: 4 Mbit

Output freq.: 4 Mbit/s
Max. capacity: 6 Mbit

EXERCISE 33: Two Token Buckets

The first plot below represents the traffic produced by an application. Such a traffic is first filtered by a bit-oriented token bucket with a token frequency of 1 Mbit/s, an output frequency of 5 Mbit/s, and a token capacity of 4 Mbit. Then, it is filtered by another bit-oriented token bucket with a token frequency of 1 Mbit/s, an output frequency of 3 Mbit/s, and a token capacity of 6 Mbit.

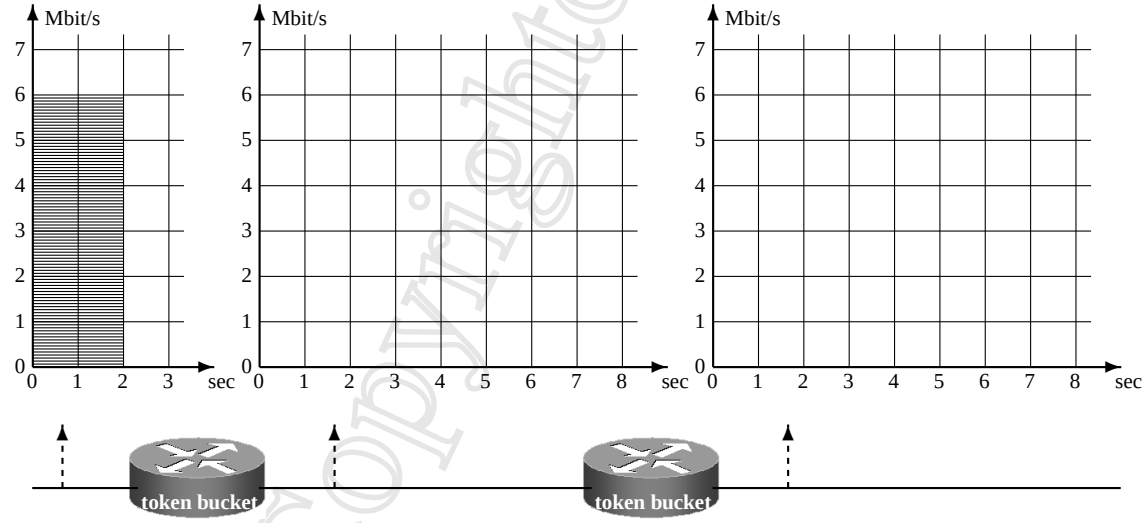
33.1) Draw the plots of the traffic exiting the first and second token bucket.



Output freq.: 5 Mbit/s
Token freq.: 1 Mbit/s
Token capacity: 4 Mbit

Output freq.: 3 Mbit/s
Token freq.: 1 Mbit/s
Token capacity: 6 Mbit

33.2) Suppose to swap the positions of the two token buckets. Draw the graphs of the traffic exiting the first and second token bucket.



Output freq.: 3 Mbit/s
Token freq.: 1 Mbit/s
Token capacity: 6 Mbit

Output freq.: 5 Mbit/s
Token freq.: 1 Mbit/s
Token capacity: 4 Mbit

Chapter 11

Transmission Control Protocol

Congestion Window (*cwnd*): During a TCP session, both endpoints use a congestion window (*cwnd*) to estimate the available bandwidth and adjust the data rate accordingly. In order to set *cwnd* to an appropriate value, TCP monitors the number of received acknowledgements. Typical TCP implementation use the **slow-start** algorithm if *cwnd* is below a certain threshold called `ssthresh`, and use the **congestion avoidance** algorithm otherwise. The threshold value changes dynamically during the TCP session.

Slow-start: Slow start increments *cwnd* by one Maximum Segment Size (mss) for each received acknowledegement. This results in an exponential growth of *cwnd*.

Congestion Avoidance: Congestion avoidance increments *cwnd* by $\frac{mss^2}{cwnd}$. Since TCP can send at most $\frac{cwnd}{mss}$ packets without receiving acknowledgements, when all the acknowledgments are received *cwnd* will be incremented by $\frac{mss^2}{cwnd} \cdot \frac{cwnd}{mss}$, that is, mss. Hence, congestion avoidance induces a linear growth of *cwnd*.

EXERCISE 34: TCP transmission with packet loss

This exercise aims at understanding how slow-start reacts to lost data packets and lost acknowledgements.

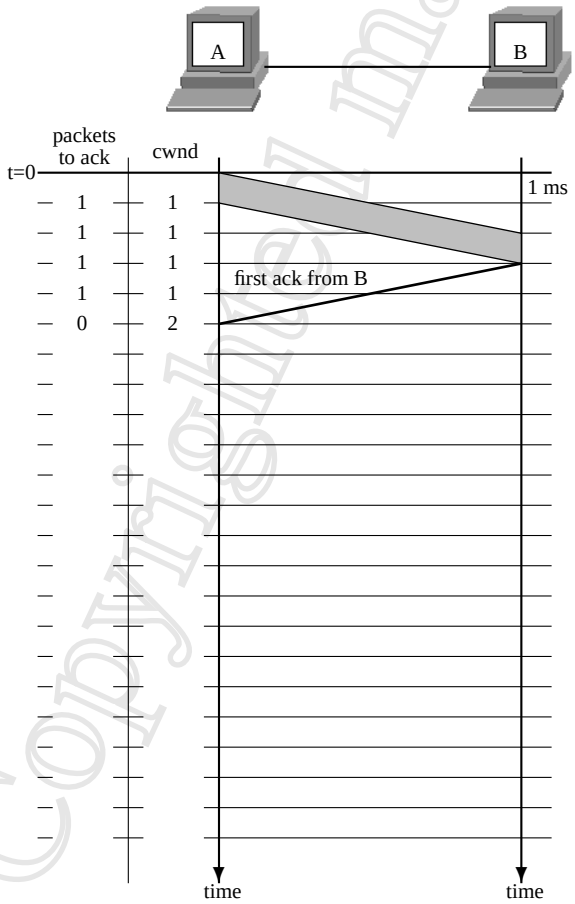
Host A uses TCP with slow-start to transmit 10 packets to host B. Suppose that:

- transmitting one packet takes 1 ms, and the propagation delay between the two hosts is 2 ms
- acknowledgments are sent immediately and have a negligible size
- after 8 ms a packet is assumed lost unless an acknowledgement is received (time-out)
- the control flow window of B and the slow-start threshold of A are high enough to be negligible
- pure slow-start is used (no fast recovery, no fast retransmit)

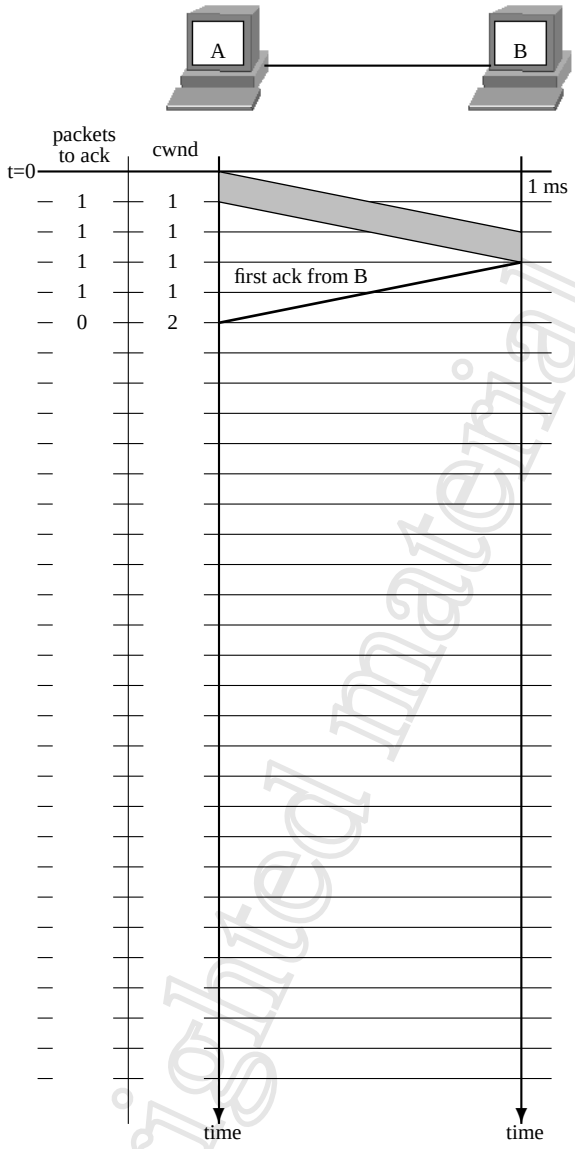
Fill in the following two diagrams. Write in the first column the number of unacknowledged packets sent by A. Write in the second column the value of A's congestion window ($cwnd$).

Suggestion: recall that a TCP acknowledgement indicates the correct reception of the packet to which it refers and of all the previous packets, and that the congestion window is increased by a single unit even when receiving a a cumulative acknowledgement of more than one packet.

34.1) Case 1: the 5^{th} ack from B does not arrive to A.



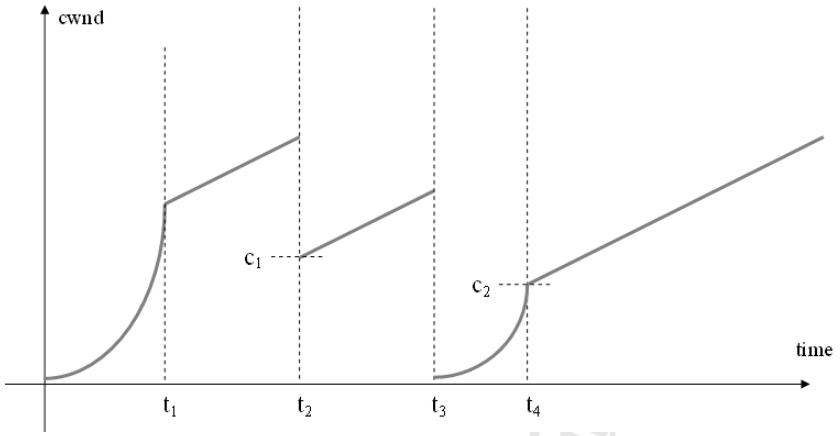
34.2) Case 2: the 5^{th} packet from A does not arrive to B.



EXERCISE 35: Interpreting a *cwnd* Graph

The purpose of this exercise is understanding what network events have happened during a TCP session by looking at the evolution of the TCP congestion window.

The following plot shows how $cwnd$ changes during a TCP conversation.



35.1) What is happening between $t = 0$ and $t = t_1$? What is the formula that TCP uses to increment $cwnd$?

35.2) What happened at $t = t_1$?

35.3) What is happening between $t = t_1$ and $t = t_2$? What is the formula that TCP uses to increment $cwnd$?

35.4) What happened at $t = t_2$?

35.5) What formula is used by TCP to compute c_1?

35.6) What happened at $t = t_3$?

Chapter 12

Domain Name System and the World Wide Web

Domain Name System (DNS) namespace: Internet names are organized in a tree of **domains**, further divided into **subdomains**, and so on.

Nameserver (NS): The DNS namespace is partitioned into non-overlapping subtrees called **zones**. A Nameserver (NS) for a given zone is responsible for knowing the IP addresses of all hosts whose name falls in that zone.

Authoritative NS: It is a NS that has always the correct and most up-to-date database for its assigned zone.

HTTP: The HyperText Transfer Protocol is used to access Internet web pages over a network. In its simplest flavour, HTTP mandates a *web browser*, i.e., an application, to initiate a TCP connection on destination po rt 80. The other end of the TCP connection is a *web server*, i.e., an application running on a remote computer. Once the TCP connection has been set up, the browser sends an **HTTP request** message, and the web server sends an **HTTP response** back.

HTTP request: There are various type of messages (*methods*) that an HTTP request can carry. The most common is the **GET** method, which is used to ask the web server for a single *resource*, identified by its Unique Resource Locator (**URL**). A resource is essentially a file that the web server can access.

EXERCISE 36: DNS resolution

The purpose of this exercise is verifying the understanding of the DNS resolution process both with and without cached records.

Consider a network connecting the following machines

DNS name	IP address	Role
`root.inter.net`	100.100.100.100	Authoritative nameserver for `.` (root)
`dns.it`	200.200.200.200	Authoritative nameserver for `.it`
`dns.pippo.it`	100.0.0.1	Authoritative nameserver for `pippo.it`
`pc.pippo.it`	100.0.0.2	User PC
`dns.pluto.it`	200.0.0.1	Authoritative nameserver for `pluto.it`
`www.pippo.it`	200.0.0.2	Web server
`ftp.pippo.it`	200.0.0.2	FTP server

36.1) What packets are exchanged on the network after `pc.pippo.it` asks `dns.pippo.it` to resolve the name `www.pluto.it` ?

Source IP	Destination IP	Packet type (Iterative/Recursive DNS query, DNS answer)
100.0.0.2	100.0.0.1	

36.2) Immediately after the DNS query of the previous exercise, `pc.pippo.it` asks `dns.pippo.it` to resolve `ftp.pluto.it` . What packets are exchanged on the network?

Source IP	Destination IP	Packet type (Iterative/Recursive DNS query, DNS answer)

EXERCISE 37: Browsers and DNS

This exercise aims at understanding how URLs in a web page trigger multiple DNS resolution processes.

A user on machine **pc.dia.edu** points his browser to **www.firm.com** , and downloads the following web page, which also includes some images

```
<html>
<head><title>Web Page</title></head>
<body>
    <h1>Sample Images</h1>
    <img src="http://www.com/1.gif"/><br/>
    <img src="http://www.buy.firm.com/2.gif"/><br/>
    <img src="http://www.firm.com/3.gif"/><br/>
    <img src="http://www.sell.firm.com/4.gif"/><br/>
    <img src="http://web.sell.firm.com/5.gif"/><br/>
</body>
</html>
```

Assume the following network setting:

name	IP address	role
root.net	100.0.0.1	DNS root server
com	100.0.0.2	authority for **com**
firm.com	100.0.0.3	authority for **firm.com**
buy.firm.com	100.0.0.4	authority for **buy.firm.com**
sell.firm.com	100.0.0.5	authority for **sell.firm.com**
www.firm.com	100.0.0.6	Web server
www.com	100.0.0.7	Web server
www.buy.firm.com	100.0.0.8	Web server
www.sell.firm.com	100.0.0.9	Web server
web.sell.firm.com	100.0.0.10	Web server
pc.dia.edu	100.0.0.11	PC
ns.dia.edu	100.0.0.12	default nameserver for **pc.dia.edu**

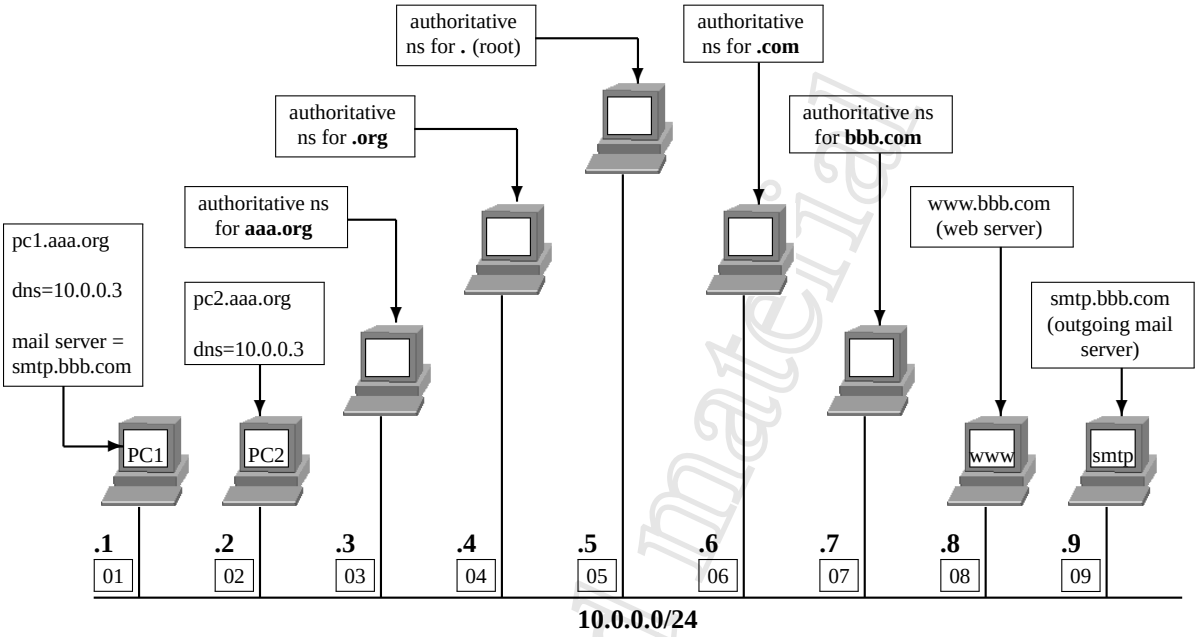
Fill in the sequence of DNS resource records that are inserted in the cache of **ns.dia.edu** from the first request till **pc.dia.edu** has finished downloading the web page and all the included images.

	Domain name	Time to live	Class	Type	Value (IP address)
1		172800	IN		
2		172800	IN		
3		172800	IN		
4		172800	IN		
5		172800	IN		
6		172800	IN		
7		172800	IN		
8		172800	IN		
9		172800	IN		

EXERCISE 38: Emails and DNS

The purpose of this exercise is understanding how HTTP and the email system rely on the DNS resolution process.

In the network diagram below each interface is assigned an IP address, whose last byte is represented by a bold dotted number (e.g., **.1**), and a MAC address, whose last byte is represented by a boxed number (e.g., 01).



Computer **`pc1.aaa.org`** configured **`10.0.0.3`** as its default nameserver, and **`smtp.bbb.com`** as its outgoing mail server. The network has been free from traffic for a very long time.

38.1) What messages are exchanged at the application layer when a user on **`pc1.aaa.org`** writes an email and his mail user agent delivers it to the outgoing mail server?

Source IP	Destination IP	Packet type (mark the right cell)			
		Recursive DNS query	Iterative DNS query	DNS answer	SMTP mail sending

38.2) Few instants after the events in the previous exercise, **`pc2.aaa.org`** issues an HTTP GET request to **`www.bbb.com`** , which returns an HTTP response. What messages are exchanged at the application layer?

Source IP	Destination IP	Packet type (mark the right cell)			
		Recursive DNS query	Iterative DNS query	DNS answer	HTTP request or response

38.3) What packets are exchanged in the network at layer 2 and 3 during the events described in the previous exercise? List them all up to the first TCP SYN packet of the HTTP connection.

Layer 2		Layer 3 (if any)		ARP request	ARP reply	Recursive DNS query	Iterative DNS query	DNS answer	TCP SYN
Dest MAC	Source MAC	Source IP	Destination IP						

EXERCISE 39: Digging into DNS

This exercise shows the trace log of a real-world DNS resolver. The purpose is being able to fully understand what steps the resolver takes to resolve a DNS name.

A user wants to understand exactly what happens during the DNS resolution of **`www.cs.brown.edu`** . To this end, he runs the command `dig +trace www.cs.brown.edu`. `dig` is a DNS resolver. When run with the `+trace` option, the DNS name is resolved through multiple iterative queries, exactly as a name server would behave. The user gets the following output.

```
1.  <gdb@omega ~>dig +trace www.cs.brown.edu
2.  ; <<>> DiG 9.2.4 <<>> +trace www.cs.brown.edu
3.  ;; global options:  printcmd
4.  .                   359695  IN      NS      K.ROOT-SERVERS.NET.
5.  .                   359695  IN      NS      L.ROOT-SERVERS.NET.
6.  .                   359695  IN      NS      M.ROOT-SERVERS.NET.
7.  .                   359695  IN      NS      A.ROOT-SERVERS.NET.
8.  .                   359695  IN      NS      B.ROOT-SERVERS.NET.
9.  .                   359695  IN      NS      C.ROOT-SERVERS.NET.
10. .                   359695  IN      NS      D.ROOT-SERVERS.NET.
11. .                   359695  IN      NS      E.ROOT-SERVERS.NET.
12. .                   359695  IN      NS      F.ROOT-SERVERS.NET.
13. .                   359695  IN      NS      G.ROOT-SERVERS.NET.
14. .                   359695  IN      NS      H.ROOT-SERVERS.NET.
15. .                   359695  IN      NS      I.ROOT-SERVERS.NET.
16. .                   359695  IN      NS      J.ROOT-SERVERS.NET.
17. ;; Received 260 bytes from 193.204.161.85 in 2 ms

18. edu.                172800  IN      NS      a.gtld-servers.net.
19. edu.                172800  IN      NS      c.gtld-servers.net.
20. edu.                172800  IN      NS      d.gtld-servers.net.
21. edu.                172800  IN      NS      e.gtld-servers.net.
22. edu.                172800  IN      NS      f.gtld-servers.net.
23. edu.                172800  IN      NS      g.gtld-servers.net.
24. edu.                172800  IN      NS      l.gtld-servers.net.
25. ;; Received 302 bytes from K.ROOT-SERVERS.NET in 13 ms

26. brown.edu.          172800  IN      NS      dark.brown.edu.
27. brown.edu.          172800  IN      NS      knot.brown.edu.
28. brown.edu.          172800  IN      NS      ns1.ucsb.edu.
29. ;; Received 171 bytes from a.gtld-servers.net in 185 ms

30. www.cs.brown.edu.   86400   IN      A       128.148.32.110
31. cs.brown.edu.       86400   IN      NS      dns.cs.brown.edu.
32. cs.brown.edu.       86400   IN      NS      ns1.ucsb.edu.
33. cs.brown.edu.       86400   IN      NS      knot.brown.edu.
34. ;; Received 179 bytes from dark.brown.edu in 131 ms
```

39.1) Who is **`K.ROOT-SERVERS.NET`** , and what role does it play in the query?

39.2) Who are **`a.gtld-servers.net`** and **`dark.brown.edu`** , and what role do they play in the query?

39.3) Explain line 30 of the response in detail.

39.4) What can be concluded by analyzing lines 31, 32, and 33 of the response?

39.5) Explain the meaning of lines 4 through 17.

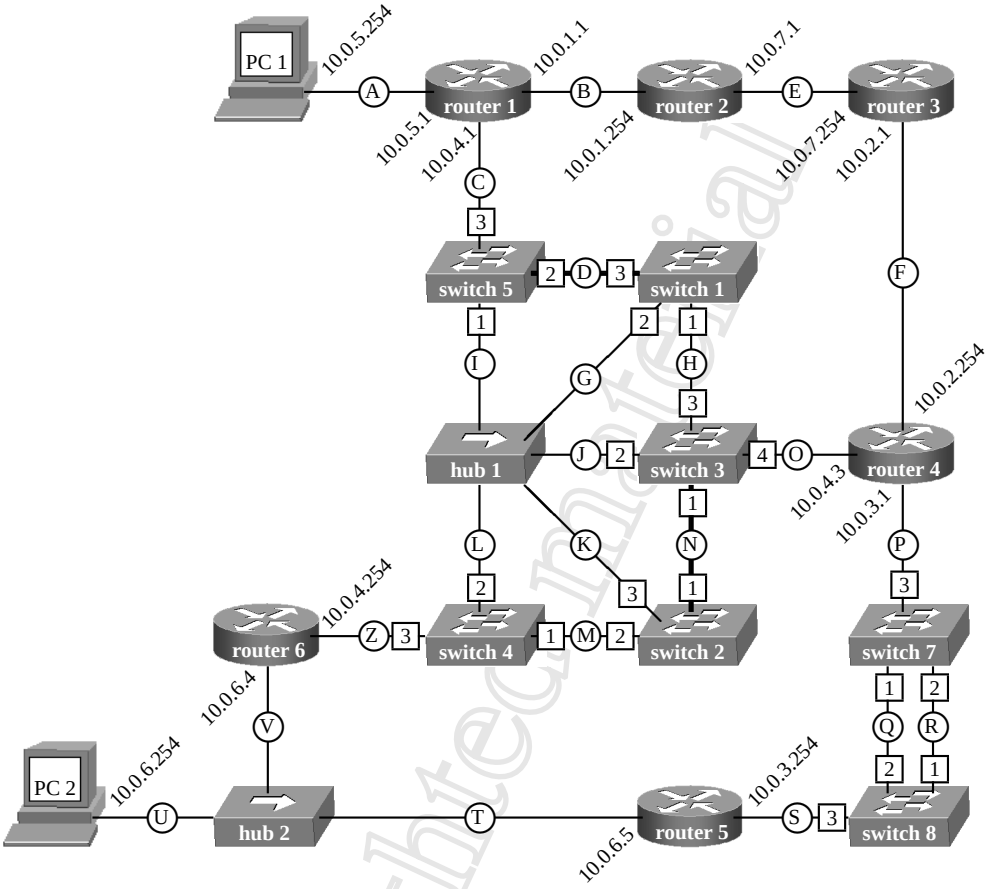
Chapter 13

Summary Exercises

Putting it all together. Summary exercises do not refer to a specific network layer or protocol. They deliberately mix information from different layers. In the same scenario hubs, switches and routers are connected together with PCs and servers. Layer two protocols run simultaneously with layer three routing. Physical accidents impact connectivity one layer after the other. All information needs to be classified and the events that occur at different levels must be reconstructed.

EXERCISE 40: A Network with Routers, Switches, and Hubs

In the network below each link is labeled with a letter inside a circle (as in Ⓐ). IP addresses are specified near the interfaces. As for the switches: (1) port numbers are represented into squares (as in 1); (2) thick links are associated with a cost 10, while thinner links have a cost 100; (3) the switch names encode the bridge-ids (for example "switch 01" has bridge-id equal to 01). Routers use the RIP distance-vector routing protocol.



40.1) Fill in the routing table of router 1.

router 1				
Network	**Netmask**	**Interface**	**Next hop**	**Cost**

40.2) Which links are traversed by a packet sent by PC 1 to PC 2 (specify the sequence of link labels)?

A										

40.3) Suppose that the administrator turns off hub 1 for maintenance. Show the new stable configuration reached by the routing table of router 1.

router 1				
Network	**Netmask**	**Interface**	**Next hop**	**Cost**

40.4) With the same hypotheses of question **40.3** (hub 1 is turned off), which links are traversed by a packet sent by PC 1 towards PC 2 (specify the sequence of link labels)?

A										

40.5) Hub 1 turns out to be definitely out of order. Furthermore, the network administrator disconnects by error link N. In these new conditions, which links are traversed in the stable stete by a packet sent by PC 1 to PC 2 (specify the sequence of link labels)?

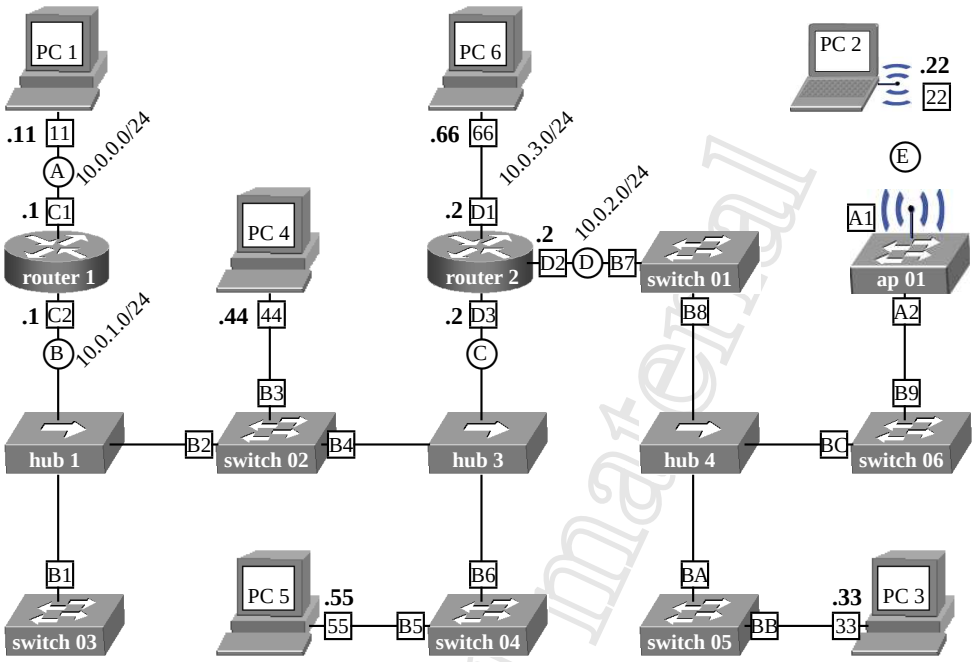
A										

40.6) In addition to the disasters described above (hub 1 turned off and link N disconnected), link H is cut by error. In these conditions, which links are traversed by a packet sent by PC 1 towards PC 2 (specify the sequence of link labels)?

A										

EXERCISE 41: Scenario with WiFi

In the network below MAC addresses are represented into squares (as in C1) while IP addresses are specified in bold near the interfaces (only the last byte of them, as in: **.11**). Some link are labeled with a letter inside a circle (as in Ⓐ).



41.1) Fill in the routing tables of router 1 and router 2, without using default routes, such that there is full connectivity between all devices of the network.

router 1			
Network	**Netmask**	**Interface**	**Next hop**

router 2			
Network	**Netmask**	**Interface**	**Next hop**

41.2) List the devices that will receive the bits of a broadcast packet (e.g., an arp request) sent by one of the following hosts.

	host sending the broadcast packet		
	PC 1	**PC 3**	**PC 4**
Devices receiving the bits			

41.3) Host PC 1 sends an echo request packet to host PC 2. How does the packet appear on the following links of the network?

41.3.1) IEEE 802.3 packet on link Ⓐ

MAC dst	**MAC src**	**IP src**	**IP dst**	**payload**
				echo request

41.3.2) IEEE 802.3 packet on link Ⓑ

MAC dst	**MAC src**	**IP src**	**IP dst**	**payload**
				echo request

41.3.3) IEEE 802.3 packet on link Ⓒ

MAC dst	**MAC src**	**IP src**	**IP dst**	**payload**
				echo request

41.3.4) IEEE 802.3 packet on link Ⓓ

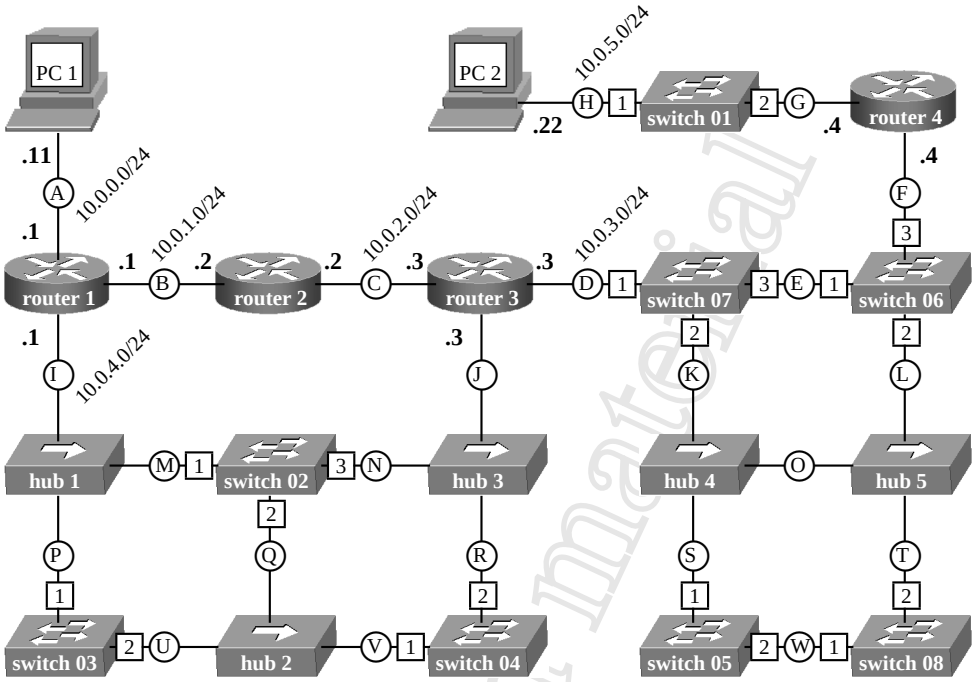
MAC dst	**MAC src**	**IP src**	**IP dst**	**payload**
				echo request

41.3.5) IEEE 802.11 packet on link Ⓔ

Address				**DS**		**IP**		**pay**
1	**2**	**3**	**4**	**To**	**From**	**Src**	**Dst**	**load**
								echo request

EXERCISE 42: Scenario with Distance Vector

In the network below each link is labeled with a letter inside a circle (as in Ⓐ). IP addresses are specified near the interfaces. As for the switches: (1) port numbers are represented into squares (as in 1); (2) all ports have the same cost; (3) the switch names encode the bridge-ids (for example "switch 01" has bridge-id equal to 01). Routers use the RIP distance-vector routing protocol.



42.1) Suppose the network reaches equilibrium. Fill in the distance vector sent by router 2 to its neighbors, the output of the traceroute command from PC 1 to PC 2, and the link traversed by a packet sent from PC 1 to PC 2.

router 2 distance vector		
Network	**Netmask**	**Distance**

traceroute from PC 1 to PC 2

Links traversed from PC 1 to PC 2:

A													

42.2) You turn off switch 02 and switch 05 and let the network reach equilibrium. Fill in the distance vector sent by router 2 to its neighbors, the output of the traceroute command from PC 1 to PC 2, and the link traversed by a packet sent from PC 1 to PC 2.

router 2 distance vector		
Network	**Netmask**	**Distance**

traceroute
from PC 1 to PC 2

Links traversed from PC 1 to PC 2:

A													

42.3) In addition to switch 02 and switch 05 you turn off also switch 03 and let the network reach equilibrium. Fill in the distance vector sent by router 2 to its neighbors, the output of the traceroute command from PC 1 to PC 2, and the link traversed by a packet sent from PC 1 to PC 2.

router 2 distance vector		
Network	**Netmask**	**Distance**

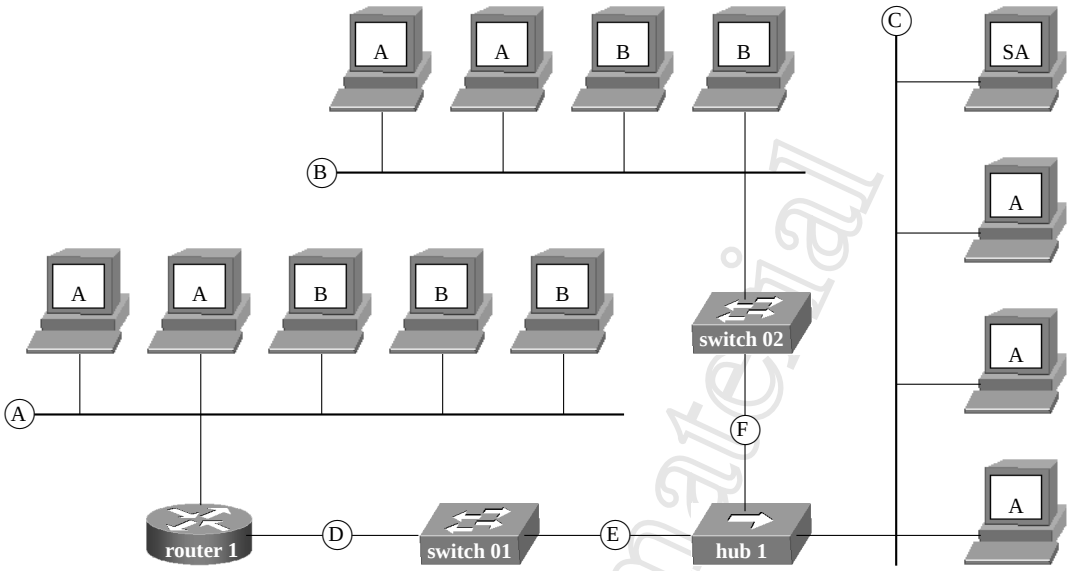
traceroute
from PC 1 to PC 2

Links traversed from PC 1 to PC 2:

A													

EXERCISE 43: Scenario with Link Load

In the network below all links are 10 Mbit/s IEEE 802.3 links, the host labeled SA is a server and the host labeled "A" and "B" are PCs. Suppose that router 1 is correctly configured and that bridge forwarding tables are up to date.



43.1) What lines are reached by a broadcast packet (for example, an ARP request) sent by server SA?

43.2) What lines are reached by a unicast packet coming from lan A and directed to server SA?

43.3) Suppose that each IP interfaces (assume bridges have no IP interface) produces 0.01 Mbit/s of brodacast traffic. Fill in the table of the broadcast load (Mbit/s) that can be expected on the various lines.

Line	Broadcast Load
A	
B	
C	

Line	Broadcast Load
D	
E	
F	

43.4) Suppose that server SA sends to each host labeled "A" 0.1 Mbit/s of unicast traffic. Fill in the table of the unicast load (Mbit/s) that can be expected on the various lines.

Line	Unicast Load
A	
B	
C	

Line	Unicast Load
D	
E	
F	

43.5) You have to add a server SB that will send to each host labeled "B" 1 Mbit/s of unicast traffic.

Fill in the table of the total (unicast + broadcast) load (Mbit/s) that can be expected on the various lines when server SB is added to lan Ⓐ.

Line	Total Load
A	
B	
C	

Line	Total Load
D	
E	
F	

Fill in the table of the total (unicast + broadcast) load (Mbit/s) that can be expected on the various lines when server SB is added to lan Ⓑ.

Line	Total Load
A	
B	
C	

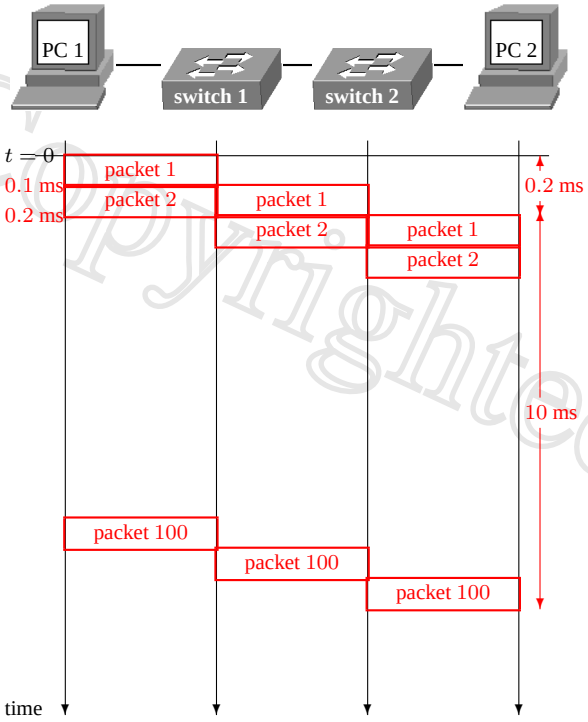
Line	Total Load
D	
E	
F	

43.6) What of the two options above is preferable, if you want to keep the total load on the links below the threshold of 1 Mbit/s?

Chapter 14

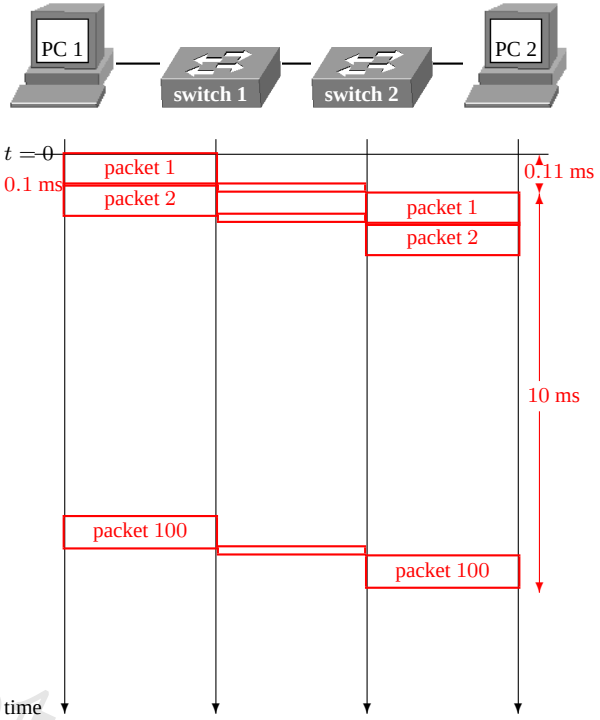
Solutions

SOLUTION TO EX. 1: Store and Forward



1.2) At what time is the file completely received by PC 2 (i.e., PC 2 gets the last bit of the file)?

10.2 ms



1.4) At what time is the file completely received by PC 2 (i.e., PC 2 gets the last bit of the file)?

10.11 ms

SOLUTION TO EX. 2: As Time Goes By

2.1) Suppose that between PC 1 and PC 2 there is a single router R1 and no other devices. Let P_1 be the propagation delay between PC 1 and R1 and P_2 the propagation delay between R1 and PC 2. Let $P_1 = P_2 = P$. How long does it take to deliver all packets to PC 2?

$2P + (A + 1)B/R$ sec.

2.2) Suppose that between PC 1 and PC 2 there are two routers R1 and R2, and no other devices. Let P_1 be the propagation delay between PC 1 and R1, let P_2 be the propagation delay between R1 and R2, and let P_3 be the propagation delay from R2 to PC 2. Let $P_1 = P_2 = P_3 = P$. How long does it take to deliver all packets to PC 2?

$3P + (A + 2)B/R$ sec.

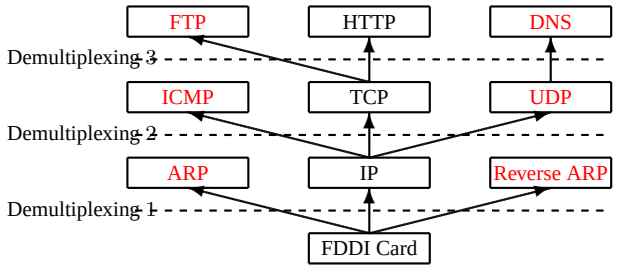
2.3) Suppose that between PC 1 and PC 2 there are $N - 1$ routers and no other devices. Let P be the propagation delay between two consecutive devices. How long does it take to deliver all packets to PC 2?

$NP + (A + N - 1)B/R$ sec.

2.4) Consider the formula of Exercise **2.3** where $A = 1$. Suppose that next years' technological progress will increase the bit rate R. In this case, will the round-trip delay of a packet (measured by the ping command, for example) yield a better or worse approximation of the propagation delay? Argument your answer.

As technology improves, round-trip delay will yield a better approximation of the propagation delay. As the bitrate R increases, the second term (due to store-and-forward behaviour) tends to be negligible with respect to the first term (due to the propagation delay).

SOLUTION TO EX. 3: Packets and Encapsulation



3.1) What information does the FDDI card use to perform demultiplexing?

The Ethertype field in the LLC header embedded in the FDDI frame.

3.2) What information does the IP layer use to perform demultiplexing?

The "protocol" field in the IP header.

3.3) What information does the TCP layer use to perform demultiplexing?

TCP uses the destination port number and the source port number to identify the application to which the payload must be forwarded.

3.4) Complete the figure above by filling in the blank cells in the Internet Protocol Suite, according to the demultiplexing hierarchy. (There is more than one way to complete the diagram.)

3.5) Discuss the figure you completed in relation to the ISO-OSI stack.

The ISO-OSI stack consists of 7 different layers, while the Internet Protocol Suite only have 5 of them. Most notably, there are no session and presentation layers: the data payload extracted from TCP is passed directly to the application. Moreover, the ICMP protocol is a layering violation: it is clearly a network protocol, nevertheless it is encapsulated within an IP packet.

SOLUTION TO EX. 4: Good Reasons for Small or Large MTUs

Error rate.	⊗ small MTU ◯ big MTU ◯ incorrect
Wait time.	⊗ small MTU ◯ big MTU ◯ incorrect
Collisions.	⊗ small MTU ◯ big MTU ◯ incorrect
Efficiency.	◯ small MTU ⊗ big MTU ◯ incorrect
Memory.	⊗ small MTU ◯ big MTU ◯ incorrect
Network size.	◯ small MTU ◯ big MTU ⊗ incorrect
Level 3 constraints.	◯ small MTU ◯ big MTU ⊗ incorrect
Router speed.	◯ small MTU ◯ big MTU ⊗ incorrect

SOLUTION TO EX. 5: Transmitting Maximum-length Packets

5.1) How long does it take for the station to transmit the packet?

$t_I = (1,518 + 8)$ [bytes] $\cdot\ 8$ [bit/byte] $\cdot\ 10^{-7}$ [s/bit] $=$
$= 12,208 \cdot 10^{-7}$ [s] $=$
$= 1.2208$ [ms]

5.2) Assume that the maximum-length packet is transmitted over a medium of infinite length. How long would the packet be? Put differently, how far would the first bit of the packet be from the last bit?

The last bit is transmitted t_I seconds after the first bit. From **5.1** we have $t_I = 1.2208$ [ms]. Assuming that the speed of the signal is $2/3$ the speed of light in empty space, we have:
$1.2208 \cdot 10^{-3}$ [s] $\cdot\ 2 \cdot 10^5$ [Km/s] $=$
$= 244.16$ [Km]

5.3) What are the analogous values for the minimum packet (64 bytes = 512 bit)?

In this case, the station would transmit for $t_I = (512 + 64)$ [bits] $\cdot\ 10^{-7}$ [s/bit] $=$
$= 576 \cdot 10^{-7}$ [s] $=$
$= 0.0576$ [ms]
The length of the minimum packet would be:
$0.0576 \cdot 10^{-3}$ [s] $\cdot\ 2 \cdot 10^5$ [Km/s] $=$
$= 11.52$ [Km]

SOLUTION TO EX. 6: Efficiency of Channel Usage

6.1) What is the maximum packet rate? That is, what is the maximum number of packets that a station can transmit per second? (Hint: recall that the minimum packet size is 64 bytes = 512 bits.)

Since the bigger the packet, the longer the transmission time, we focus on minimum-size packets of 64 bytes. Each packet transmission requires *(i)* transmitting the preamble and start-frame delimiter (8 bytes = 64 bits); *(ii)* transmitting the packet itself (64 bytes = 512 bits); and *(iii)* waiting the inter-packet gap (12 bytes = 96 bits). Then, transmitting a single packet takes a time
$T = (512 + 64 + 96)$ [bits/pack.] $\cdot\ 10^{-7}$ [s/bit] $=$
$= 672 \cdot 10^{-7}$ [s/pack.].
The packet rate is therefore
$1/T = 14,880.9$ [pack/s]

6.2) How many maximum-size packets (1,518 bytes) can a station transmit per second?

The period T is:
$T = (1,518 + 8 + 12)$ [bytes/pack.] $\cdot\ 8$ [bit/byte] $\cdot\ 10^{-7}$ [s/bit] $=$
$= 12,304 \cdot 10^{-7}$ [s/pack.].
The packet rate is
$1/T = 812.74$ [pack/s]

6.3) What would be the throughput (bytes per seconds) carried in the packet's payload in the two cases?

For minimum-size packets, from **6.1** and from the fact that each packet carries 46 bytes, we have:
$14,880.9$ [packets/s] $\cdot$ 46 [bytes/packet] $= 684,480$ [bytes/s]
For maximum-size packets, from **6.2** and from the fact that each packet carries 1,500 bytes, we have:
812.74 [packets/s] $\cdot\ 1,500$ [bytes/packet] $= 1,219,110$ [bytes/s]

6.4) What is the maximum efficiency of channel use? Compute the efficiency as the ratio between the payload throughput and the channel throughput.

We have the maximum efficiency when sending maximum-size packets. From **6.3** we have a payload throughput of $P = 1,219,110$ [bytes/s] $\cdot$ 8 [bits/byte] $= 9.75288 \cdot 10^6$ [bits/s].
The channel throughput is 10^7 [bits/s]. Hence, the efficiency is $P/10^7 = 0.975288$. That is, 2.5% of the bandwidth is used by the protocol in the best case.

SOLUTION TO EX. 7: A Fake IEEE 802.3

7.1) Knowing that a signal propagates on the transmission medium at $200,000$ Km/sec speed, compute how many seconds must 2τ (round-trip delay) be for CSMA/CD to work correctly.

A collision must be detected during transmission. In the worst case, it will be detected during the transmission of the last bit of the packet of the minimum size. Hence, $2,000$ [bit-time] $= 2\tau$, where we have to find τ. $\tau = 1,000$ [bit-time] $= \frac{10^3 \text{ [bit]}}{10^8 \text{ [bit/sec]}} = 10^{-5} sec$

7.2) What is the maximum length of a collision domain in the FAKE 802.3 standard?

$L_{max} = \tau \cdot \frac{2}{3}c = 10^{-5}$ [sec] $\cdot\ 2 \cdot 10^8$ [m/sec] $= 2 \cdot 10^3$ [m] $= 2$ [Km]

Note: the fact that the MTU is 2000 bytes is irrelevant.

SOLUTION TO EX. 8: Is it a Big Deal?

8.1) Compute $2t_p + t_I$ in the three cases.

Offer 1	Offer 2	Offer 3
$t_p = \frac{20 \text{ [Km]}}{2 \cdot 10^8 \text{ [Km/sec]}} = 0.0001$ [sec]	$t_p = \frac{2000 \text{ [Km]}}{2 \cdot 10^8 \text{ [Km/sec]}} = 0.01$ [sec]	$t_p = \frac{2 \text{ [Km]}}{2 \cdot 10^8 \text{ [Km/sec]}} = 0.00001$ [sec]
$t_I = \frac{250 \cdot 8 \text{ [bit/pack.]}}{10^6 \text{ [bit/sec]}} = 0.002$ [sec]	$t_I = \frac{250 \cdot 8 \text{ [bit/pack.]}}{10^8 \text{ [bit/sec]}} = 0.00002$ [sec]	$t_I = \frac{250 \cdot 8 \text{ [bit/pack.]}}{10^5 \text{ [bit/sec]}} = 0.02$ [sec]
$2t_p + t_I = 0.0022$ [sec]	$2t_p + t_I = 0.02002$ [sec]	$2t_p + t_I = 0.02002$ [sec]

8.2) How many labels do you have to identify your packets and how many of them you can use simultaneously without receiving any acknowledgement?

Offer 1	Offer 2	Offer 3
You have $M = 2^3 = 8$ labels. You can use $M - 1 = 7$ labels.	You have $M = 2^3 = 8$ labels. You can use $M - 1 = 7$ labels.	You have $M = 2^4 = 16$ labels. You can use $M - 1 = 15$ labels.

8.3) Are the bits sufficient to have a continuous transmission on the line?

Offer 1	Offer 2	Offer 3
Yes, because $2t_p + t_I < (M-2)t_I$, in fact: $0.0022 < 0.012$	No, because $2t_p + t_I > (M-2)t_I$, in fact: $0.02002 > 0.00012$	Yes, because $2t_p + t_I < (M-2)t_I$, in fact: $0.02002 < 0.28$

8.4) How many packets per second can be transmitted in the three cases?

Offer 1	Offer 2	Offer 3
$\frac{1}{t_I} = 500$ [pack./sec]	$\frac{M-1}{2t_p + 2t_I} = 349$ [pack./sec]	$\frac{1}{t_I} = 50$ [pack./sec]

SOLUTION TO EX. 9: High-delay link

9.1) Compute $2t_p + t_I$.

$t_p = \frac{10{,}000 \text{ [km]}}{200{,}000 \text{ [km/sec.]}} = 0.05[sec.]$
$t_I = \frac{8{,}000 \text{ [bit]}}{1 \text{ [Mbit/sec]}} = 0,008 \text{ [sec.]}$
$2t_p + t_I = 0.108 \text{ [sec.]}$

9.2) Compute the minimum number of bits to count packets that is needed to have continuous transmission.

We want $(M-2)t_I \geq 2t_p + t_I$, i.e., $M \geq 15.5$
Hence, the minimum number of bits is 4.

9.3) How many packets per second can be transmitted assuming that the number of bits used count packets guarantees continuous transmission?

The throughput is $1/t_I = 125$ [pack./sec]

SOLUTION TO EX. 10: Switches Vs Hubs

10.1) What is the network level at which they operate and what does this imply with regard to the information they handle?

A switch is a layer 2 network device, while a hub is a layer 1 network device. Hence, the switch decodes MAC packets, while the hub only decodes bits.

10.2) When the above devices have multiple ports, where do they forward a received packet?

If the destination address is unicast, the switch forwards the packet on a single port, which is the one through which the destination can be reached (filtering) .The hub forwards the packet towards all ports except the one that received the packet.

10.3) What strategy do the two devices apply when they detect a collision while transmitting a packet out of a port?

When a collision is detected during transmission, the switch outputs the jamming sequence out of the transmission port only, launches the binary exponential backoff algorithm, and prepares to send again the packet after that. This is possible because the switch stores the packet in memory until it has been successfully transmitted. The hub, instead, outputs the jamming sequence out of all its ports.

SOLUTION TO EX. 11: Malfunctioning Switch

11.1) Suppose that S is broken. In particular, suppose that all its functions are properly working except its "learning" ability. Namely, S has a forwarding table that is always empty and each time a frame is received by a port, it is sent out through all other ports. Do you think that the network in which S is placed will work anyway? Do you think that S is able to correctly separate the collision domains? What are the consequences of this fault?

The network in which S is placed will work anyway. Collision domains are separated by S. Only the load on the links will change since S is sending frames to all ports, while a properly functioning switch would send them only on the single port where they are needed.

11.2) The vendor repairs S, however, a new malfunctioning turns out after a while. Now, all the functions of S are working as expected, but the output memory buffer has become volatile. In other words, Each time a bit is transmitted over the cable, its value is immediately forgotten. Do you think that the network in which S is placed will work anyway? Is S able to correctly separate the collision domains? What are the consequences of this fault?

The network will not work anymore. S is now not able to correctly separate the collision domains. In particular, each time a transmission collision occurs, S is able to detect the collision, but cannot retransmit the frame.

11.3) The vendor is contacted again, and S gets repaired. Unfortunately, after a while a new type of fault happens (yes, we will blacklist that vendor). This time, the spanning tree protocol is broken, while all other functions are working properly. In particular, S is not able to switch the ports in blocking state when a cycle is detected. Do you think that the network in which S is will work anyway? Do you think that S is able to correctly separate collision domains? What are the consequences of this fault?

The network will not work properly. S is sending packets in such a way to create cycle. Hosts will receive multiple copies of the same packet.

SOLUTION TO EX. 12: Cut-through Switch and Collisions

12.1) Is PC 2 able to start the transmission at $t = 250$ bit-time? Why?

Yes, it is. Switch 1 receives the first bit of the packet sent by PC 1 at $t = 88$ bit-time. The delay introduced by switch 1 is $(7 + 1 + 6)$[bytes] $\cdot$ 8[bit/byte] $= 112$ bit-time. Then, at $t = 88 + 112 = 200$ bit-time switch 1 starts to forward the packet towards PC 2. At $t = 200 + 100 = 300$ bit-time PC 2 receives the first bit of the packet. Hence, at $t = 250$ bit-time, PC 2 carrier-sense circuit does not forbid the transmission of a packet.

12.2) What happens in the network at $t = 300$ bit-time?

PC 2 detects a collision with the packet forwarded by switch 1 (see above).

12.3) What happens in the network at $t = 350$ bit-time?

Switch 1 detects a collision with the packet transmitted by PC 2.

12.4) Does PC 1's MAC layer force a rentransmission of the packet? Why?

No, it does not. PC 1's MAC layer is not affected by the collision that happens in a different collision domain.

12.5) Does switch 1's MAC layer force a rentransmission of the packet to PC 2? Why?

Yes, it does. Switch 1 is responsible for forwarding the packet towards PC 2.

12.6) If switch 1 were not a cut-through switch, what would have happened instead?

If switch 1 were not a cut-through switch, then it would have forwarded the packet from PC 1 only after complete reception (store-and-forward behavior). In this case, no collisions occur: switch 1 receives the last bit of the packet from PC 1 at time $t = 888$ bit-time. At that time, switch 1 is also receiving the packet from PC 2. Hence, switch 1 is able to detect the transmission from PC 2 (carrier-sense).

SOLUTION TO EX. 13: Subnetting

Either

	Subnet	Netmask	Broadcast
Lan A	193.203.163.0	255.255.255.224	193.203.163.31
Lan B	193.203.163.32	255.255.255.224	193.203.163.63
Lan C	193.203.163.128	255.255.255.128	193.203.163.255
Lan D	193.203.163.64	255.255.255.224	193.203.163.95
Lan E	193.203.163.96	255.255.255.224	193.203.163.127

Or

	Subnet	Netmask	Broadcast
Lan A	193.203.163.128	255.255.255.224	193.203.163.159
Lan B	193.203.163.160	255.255.255.224	193.203.163.191
Lan C	193.203.163.0	255.255.255.128	193.203.163.127
Lan D	193.203.163.192	255.255.255.224	193.203.163.224
Lan E	193.203.163.223	255.255.255.224	193.203.163.255

Lans A, B, D, and E can be permutated.

SOLUTION TO EX. 14: Subnets and binary trees

14.1) In the tree representation, what corresponds to a net (or subnet)? What corresponds to the IP prefix of a net (or subnet)?

A subnet corresponds to a non-leave node x and the subtree rooted at x. The leaves that belong to the subtree rooted at x correspond to all the IP addresses contained in the subnet. The prefix of the subnet associated with node x is obtained by concatenating the labels of the edges traversed by the path from the root to x.

14.2) In the tree representation what corresponds to all the nets (or subnets) having a $/8$ prefix?

Nets having a $/8$ prefix correspond to all nodes at distance 8 from the root.

14.3) In the tree representation, given a net (or subnet) A and another net (or subnet) B, how is it possible to check whether A and B are overlapping?

Let a be the node in the tree representing A, and let b be the node representing B. If a is on the path from the root to b, then A contains all addresses in B. If b is on the path from the root to a, then B contains all addresses in B. Otherwise, A and B do not overlap.

14.4) In the tree representation, given a net (or subnet) A and another net (or subnet) B, how is it possible to check whether A and B have been obtained by subnetting a third net (or subnet) C?

Let a be the node in the tree representing A, b the node representing B. If a and b have a common ancestor c, that is, c is in the path from the root to a and in the path from the root to b, then A and B have been obtained by subnetting a third net C corresponding to node c.

14.5) In the tree representation, what corresponds to a routing table?

A routing table corresponds to a set of nodes in the tree, where each node is associated with a network interface.

SOLUTION TO EX. 15: Guessing Remote Prefixes with Local Netmasks

15.1) First, PC X sends the packet to PC 1 (100.100.100.124). PC X computes its own prefix as 100.100.0.0 and the prefix for PC 1 as 100.100.0.0. Can the actual prefix of PC 1 be 100.0.0.0? Comment your answer.

No, it cannot. If the actual prefix of PC 1 were 100.0.0.0, then the prefix of PC X would be contained into the prefix of PC 1, which would lead to wrong subnetting.

15.2) Can the actual prefix of PC 1 be 100.100.100.0? Comment your answer.

No, it cannot. If the actual prefix of PC 1 were 100.100.100.0, then the prefix of PC 1 would be contained into the prefix of PC X, which would lead to wrong subnetting.

15.3) Second, PC X sends the packet to PC 2 (100.200.100.125). PC X computes its own prefix as 100.100.0.0 and a prefix for PC 2 as 100.200.0.0. Can the actual prefix of PC 2 be 100.0.0.0? Comment your answer.

No, it cannot. If the actual prefix of PC 2 were 100.0.0.0, then the prefix of PC X would be contained into the prefix of PC 2, which would lead to wrong subnetting.

15.4) Can the actual prefix of PC 2 be 100.200.100.0? Comment your answer.

Yes, it can. The remote LAN where PC 2 lies can have an arbitrarily long prefix, provided that does not overlap with the prefix of PC X.

15.5) Is there a case among the four cases mentioned in the previous questions where PC X risks to send the IP packet to the wrong recipient (for example to the default router instead of to the destination host or vice versa)? Comment your answer.

No, such a risk does not exist. In any case, PC X can only be wrong when it guesses a prefix that is different from its own. However, even if the guessed prefix is not the correct one, the packet will be delivered to the default router.

SOLUTION TO EX. 16: Constrained Addressing Plan

16.1) Assign an IP prefix to LANs A, B, and C. Fill in the following table and specify for each LAN its subnet, its netmask, and the corresponding broadcast address.

LAN	subnet	netmask	broadcast
A	193.1.1.0	255.255.255.128	193.1.1.127
B	193.1.1.192	255.255.255.192	193.1.1.255
C	193.1.1.160	255.255.255.224	193.1.1.191

16.2) Assign an IP prefix to point-to-point links D, E, F, and G. Fill in the following table and specify for each link its subnet, its netmask, and the corresponding broadcast address.

link	subnet	netmask	broadcast
D	193.1.1.144	255.255.255.252	193.1.1.147
E	193.1.1.148	255.255.255.252	193.1.1.151
F	193.1.1.152	255.255.255.252	193.1.1.155
G	193.1.1.156	255.255.255.252	193.1.1.159

Any permutation of links D, E, F, and G is also a solution.

SOLUTION TO EX. 17: Forwarding with wrong subnets

17.1) At a certain time, after a long period of inactivity, a user on machine P runs the command `ping 193.204.128.2`. For the sake of brevity, assume that the `ping` command only produces one packet. List the packets that a sniffer placed in observation point A sees on the network.

dst MAC	**src MAC**	**src IP** (if any)	**dst IP** (if any)	**packet type** (one of the following: arp request, arp reply, icmp echo-request, icmp echo-reply, icmp error)
bcast	0:1			arp request
0:1	0:2			arp reply
0:2	0:1	193.204.0.2	193.204.128.2	icmp echo-request
0:1	0:2	193.204.128.2	193.204.0.2	icmp echo-reply

Similarly, list the packets that a sniffer placed in observation point B sees on the network.

dst MAC	**src MAC**	**src IP** (if any)	**dst IP** (if any)	**packet type** (one of the following: arp request, arp reply, icmp echo-request, icmp echo-reply, icmp error)
bcast	0:3			arp request
0:3	0:4			arp reply
0:4	0:3	193.204.0.2	193.204.128.2	icmp echo-request
0:3	0:4	193.204.128.2	193.204.0.2	icmp echo-reply

17.2) At a given time, after a long period of inactivity, a user on machine P runs the command `ping 193.204.128.8`. For the sake of brevity, assume that the `ping` command only produces one packet. List the packets that a sniffer placed in observation point B sees on the network.

dst MAC	**src MAC**	**src IP** (if any)	**dst IP** (if any)	**packet type** (one of the following: arp request, arp reply, icmp echo-request, icmp echo-reply, icmp error)
bcast	0:3			arp request
bcast	0:3			arp request
bcast	0:3			arp request

Arp requests are never answered since there is no machine with IP address 193.204.128.8. The router will try an arbitrary number of requests (the exact number depends on the software running on the router) and then send an ICMP error "Destination Unreachable" back to P. This ICMP error, however, cannot be seen by observation point B.

17.3) Now assume that the administrator of machine P made a configuration error: he specified 255.255.0.0 as the netmask. At a given time, after a long period of inactivity, a user on machine P runs the command `ping 193.204.128.2`. For the sake of brevity, assume that the `ping` command only produces one packet. List the packets that a sniffer placed in observation point A sees on the network.

dst MAC	**src MAC**	**src IP** (if any)	**dst IP** (if any)	**packet type** (one of the following: arp request, arp reply, icmp echo-request, icmp echo-reply, icmp error)
bcast	0:1			arp request
bcast	0:1			arp request
bcast	0:1			arp request

17.4) Assume you are in the same situation as in the previous exercise. At a given time, after a long period of inactivity, a user on machine Q runs the command `ping 193.204.0.2`. For the sake of brevity, assume that the `ping` command only produces one packet. List the packets that a sniffer placed in observation point A sees on the network.

dst MAC	**src MAC**	**src IP** (if any)	**dst IP** (if any)	**packet type** (one of the following: arp request, arp reply, icmp echo-request, icmp echo-reply, icmp error)
bcast	0:2			arp request
0:2	0:1			arp reply
0:1	0:2	193.204.128.2	193.204.0.2	icmp echo-request
bcast	0:1			arp request
bcast	0:1			arp request
bcast	0:1			arp request

SOLUTION TO EX. 18: Echo request and echo reply

18.1) Suppose to run a `ping` on PC 1 towards PC 2. Which routers are traversed (and in what order) by echo request packets and which routers are traversed by echo reply packets?

Each echo request packet traverses: router 1 and router 3.
Each echo reply packet traverses: router 3 and router 1.

18.2) Suppose to run a `ping` on PC 3 towards PC 2. Which routers are traversed (and in what order) by echo request packets and which routers are traversed by echo reply packets?

Each echo request packet traverses: router 2, router 1, and router 3.
Each echo reply packet traverses: router 3 and router 2.

18.3) Suppose to run a `ping` on PC 4 towards PC 1. Which routers are traversed (and in what order) by echo request packets and which routers are traversed by echo reply packets?

Each echo request packet traverses: router 3 and router 1.
Each echo reply packet traverses: router 1, router 2, and router 3.

SOLUTION TO EX. 19: Traceroute

19.1) What is the output of the traceroute command from PC 4 to PC 2?

user@PC4 ∼> traceroute 100.2.2.22
100.4.4.3
100.2.2.22

19.2) What is the output of the traceroute command from PC 3 to PC 4?

user@PC3 ∼> traceroute 100.4.4.44
100.3.3.2
200.30.30.3
100.4.4.44

19.3) What is the output of the traceroute command from PC 3 to PC 1?

user@PC3 ∼> traceroute 100.1.1.11
100.3.3.2
200.40.40.1
100.1.1.11

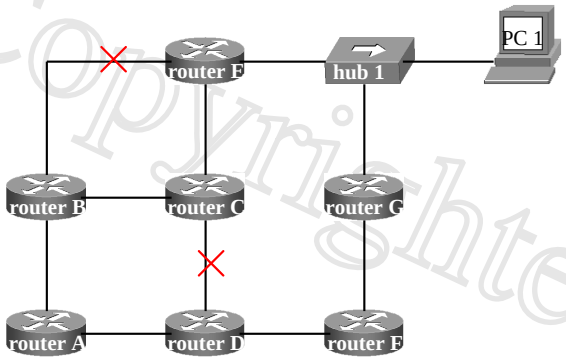
19.4) What is the output of the traceroute command from PC 3 to PC 2?

user@PC3 ∼> traceroute 100.2.2.22
100.3.3.2
200.20.20.1 (more correctly 200.40.40.1)
200.40.40.2 (more correctly 100.3.3.2)
200.20.20.1 (more correctly 200.40.40.1)
200.40.40.2 (more correctly 100.3.3.2)
...

19.5) What is the output of the traceroute command from PC 1 to a non-existent host (for example 100.5.5.55)?

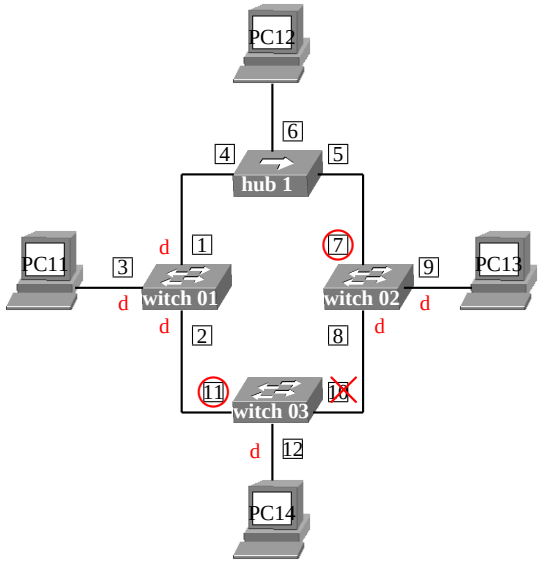
user@PC1 ∼> traceroute 100.5.5.55
100.1.1.1
200.10.10.3
200.30.30.2 (more correctly 200.40.40.2)
200.20.20.1 (more correctly 100.1.1.1)
200.10.10.3
....

SOLUTION TO EX. 20: Sniffing Distance-Vectors



SOLUTION TO EX. 21: Switches and Hubs

Consider the following network with 3 802.1D switches, a hub, and 4 PCs.



21.1) Which switch is the root bridge and why?

switch 01 is the root bridge as it has the lowest bridge-id.

21.2) Compute the spanning tree. Complete the figure above writing a small "d" near designated ports, circling root ports, and crossing ports in blocking state.

21.3) What role does hub 1 play in the spanning tree computation?

None. It merely forwards BPDUs from one port to all the others.

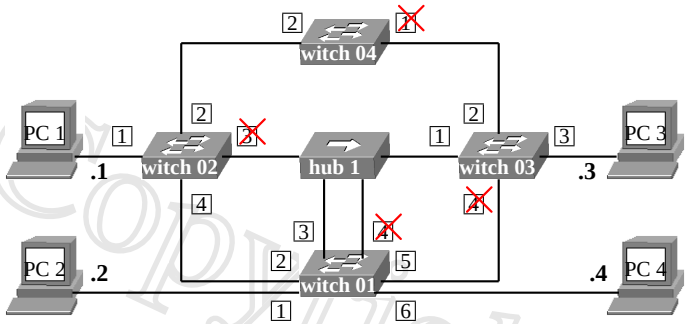
21.4) Supposing that all PCs constantly exchange packets, fill in the forwarding tables of the four switches (ignore BPDU traffic):

Switch 1		
port 1	port 2	port 3
12	14	11
13		

Switch 2		
port 7	port 8	port 9
12		13
11		
14		

Switch 3		
port 10	port 11	port 12
	11	14
	12	
	13	

SOLUTION TO EX. 22: Switches, Hosts and Hubs



22.1) Which one is the root-bridge and why? By the way: is it true that the root-bridge never has a port in blocking state?

Switch 01 is the root bridge since it has the lowest bridge-id. It is not true that the root bridge never has a port in blocking state, because there could be a loop between two (or more) of its ports. This scenario is an example.

22.2) Which is the root-port of switch 02 and why?

The root-port of switch 02 is port 4. In fact, ports 2, 3, and 4 receive configuration BPDUs with the same cost (0). Among these three, ports 3 and 4 receive BPDUs with lower sender bridge-id (01). Finally, port 4 receives BPDUs with lower sender port (2).

22.3) Which is the root-port of switch 04 and why?

The root-port of switch 04 is port 2. In fact, ports 1 and 2 receive configuration BPDUs with the same cost (0). Among these two, port 2 receives BPDUs with lower sender bridge-id (02).

22.4) Cross the ports blocked by the spanning tree algorithm.

22.5) Which is the designated-port of the collision domain represented by hub 1? Why?

All ports attached to hub 1 send configuration BPDUs with zero path cost. Port 3 and 4 of switch 01 send BPDUs with lowest sender bridge-id. Port 3 sends BPDUs with lowest sender port. Hence, the designated-port of the collision domain is port 3 of switch 01.

SOLUTION TO EX. 23: CIDR Scenario 1

23.1) Fill in the routing table of router 7 once compressed with CIDR.

Subnet Address	Netmask	Interface	Next-hop
10.0.0.0	255.255.255.0	N	d.c.
10.0.1.0	255.255.255.0	E	d.c.
10.0.2.0	255.255.255.0	W	d.c.
10.0.3.0	255.255.255.0	W	10.0.2.3
10.0.4.0	255.255.252.0	W	10.0.2.3
10.0.8.0	255.255.248.0	W	10.0.2.3

23.2) Fill in the routing table of router 1 once compressed with CIDR.

Subnet Address	Netmask	Interface	Next-hop
10.0.0.0	255.255.252.0	E	10.0.7.2
10.0.4.0	255.255.254.0	E	10.0.7.2
10.0.6.0	255.255.255.0	E	10.0.7.2
10.0.7.0	255.255.255.0	E	d.c.
10.0.8.0	255.255.248.0	E	10.0.7.2

23.3) Fill in the routing table of router 4 once compressed with CIDR.

Subnet Address	Netmask	Interface	Next-hop
10.0.0.0	255.255.248.0	W	10.0.14.2
10.0.8.0	255.255.254.0	N	10.0.12.6
10.0.10.0	255.255.254.0	S	10.0.13.8
10.0.12.0	255.255.255.0	N	d.c.
10.0.13.0	255.255.255.0	S	d.c.
10.0.14.0	255.255.255.0	W	d.c.
10.0.15.0	255.255.255.0	E	10.0.14.2

23.4) Can router 1 announce all the networks in the diagram with a single routing line? How?

Router 1 could announce 10.0.0.0 with netmask 255.255.240.0 (or /20).

SOLUTION TO EX. 24: CIDR Compaction 1

Subnet Address	Netmask	Interface
130.30.0.0	255.255.0.0	int 1
130.31.0.0	255.255.0.0	int 1
130.32.0.0	255.255.0.0	int 1
130.33.0.0	255.255.0.0	int 1
130.34.0.0	255.255.0.0	int 1
130.35.0.0	255.255.0.0	int 1
130.36.0.0	255.255.0.0	int 1
130.37.0.0	255.255.0.0	int 1
130.38.0.0	255.255.0.0	int 1
130.39.0.0	255.255.0.0	int 1
130.40.0.0	255.255.0.0	int 1
130.41.0.0	255.255.0.0	int 2

Subnet Address	Netmask	Interface
130.30.0.0	255.254.0.0	int 1
130.32.0.0	255.248.0.0	int 1
130.40.0.0	255.255.0.0	int 1
130.41.0.0	255.255.0.0	int 2

SOLUTION TO EX. 25: CIDR Compaction 2

Subnet Address	Netmask	Interface
194.100.0.0	255.255.255.0	int 1
194.100.1.0	255.255.255.0	int 1
194.100.2.0	255.255.254.0	int 1
194.100.4.0	255.255.252.0	int 1
194.100.8.0	255.255.248.0	int 1
194.100.48.0	255.255.240.0	int 1
194.100.64.0	255.255.240.0	int 1

Subnet Address	Netmask	Interface
194.100.0.0	255.255.240.0	int 1
194.100.48.0	255.255.240.0	int 1
194.100.64.0	255.255.240.0	int 1

SOLUTION TO EX. 26: CIDR Compaction 3

Subnet Address	Netmask	Interface
194.38.40.0	255.255.255.0	int 1
194.38.41.0	255.255.255.0	int 1
194.38.42.0	255.255.254.0	int 1
194.38.44.0	255.255.252.0	int 1
194.38.48.0	255.255.254.0	int 1
194.38.50.0	255.255.255.0	int 1
194.38.51.0	255.255.255.0	int 1

Subnet Address	Netmask	Interface
194.38.40.0	255.255.248.0	int 1
194.38.48.0	255.255.252.0	int 1

SOLUTION TO EX. 27: CIDR Compaction 4

Subnet Address	Netmask	Interface
193.205.3.0	255.255.255.0	int 2
193.205.4.0	255.255.255.0	int 2
193.205.5.0	255.255.255.0	int 4
130.200.0.0	255.254.0.0	int 3
130.202.0.0	255.254.0.0	int 3
190.204.118.0	255.255.254.0	int 1
190.204.120.0	255.255.252.0	int 1
190.204.124.0	255.255.252.0	int 1

Subnet Address	Netmask	Interface
193.205.3.0	255.255.255.0	int 2
193.205.4.0	255.255.255.0	int 2
193.205.5.0	255.255.255.0	int 4
130.200.0.0	255.252.0.0	int 3
190.204.118.0	255.255.254.0	int 1
190.204.120.0	255.255.248.0	int 1

SOLUTION TO EX. 28: CIDR Compaction 5

Subnet Address	Netmask	Interface
140.38.0.0	255.255.0.0	int 1
140.39.0.0	255.255.0.0	int 1
140.40.0.0	255.255.0.0	int 1
140.41.0.0	255.255.0.0	int 1
140.42.0.0	255.255.0.0	int 1
140.43.0.0	255.255.0.0	int 1
140.44.0.0	255.255.0.0	int 1
140.45.0.0	255.255.0.0	int 1
140.46.0.0	255.255.0.0	int 2
140.47.0.0	255.255. 0.0	int 1
140.48.0.0	255.255. 0.0	int 1

Subnet Address	Netmask	Interface
140.38.0.0	255.254.0.0	int 1
140.40.0.0	255.252.0.0	int 1
140.44.0.0	255.254.0.0	int 1
140.46.0.0	255.255.0.0	int 2
140.47.0.0	255.255.0.0	int 1
140.48.0.0	255.255.0.0	int 1

SOLUTION TO EX. 29: CIDR Compaction 6

Subnet Address	Netmask	Interface
194.39.0.0	255.255.0.0	int 2
194.40.0.0	255.255.128.0	int 2
194.40.128.0	255.255.128.0	int 2
194.41.0.0	255.255.192.0	int 2
194.41.64.0	255.255.192.0	int 2
194.41.128.0	255.255.192.0	int 2
194.41.192.0	255.255.192.0	int 2
194.42.0.0	255.254.0.0	int 2
194.44.0.0	255.255.0.0	int 2

Subnet Address	Netmask	Interface
194.39.0.0	255.255.0.0	int 2
194.40.0.0	255.252.0.0	int 2
194.44.0.0	255.255.0.0	int 2

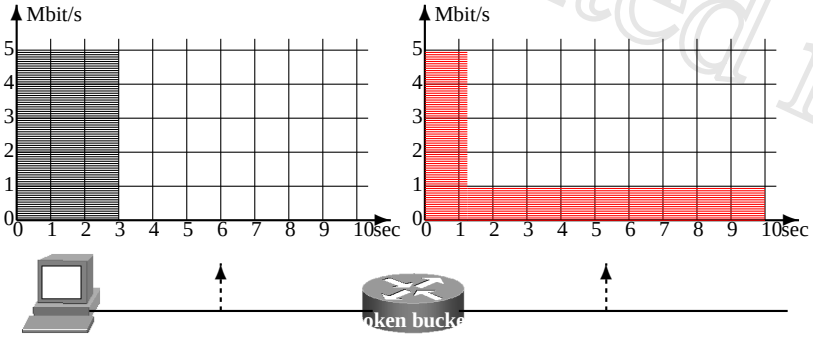
SOLUTION TO EX. 30: CIDR Compaction 7

Subnet Address	Netmask	Interface
20.0.1.0	255.255.255.128	int 1
20.0.1.128	255.255.255.128	int 1
20.0.2.0	255.255.255.0	int 1
20.0.3.0	255.255.255.0	int 1
20.0.4.0	255.255.255.0	int 1
20.0.5.0	255.255.255.0	int 1
20.0.6.0	255.255.255.0	int 1
20.0.7.0	255.255.255.0	int 1
20.0.8.0	255.255.255.0	int 1

Subnet Address	Netmask	Interface
20.0.1.0	255.255.255.0 or /24	Int 1
20.0.2.0	255.255.254.0 or /23	Int 1
20.0.4.0	255.255.252.0 or /22	Int 1
20.0.8.0	255.255.255.0 or /24	Int 1

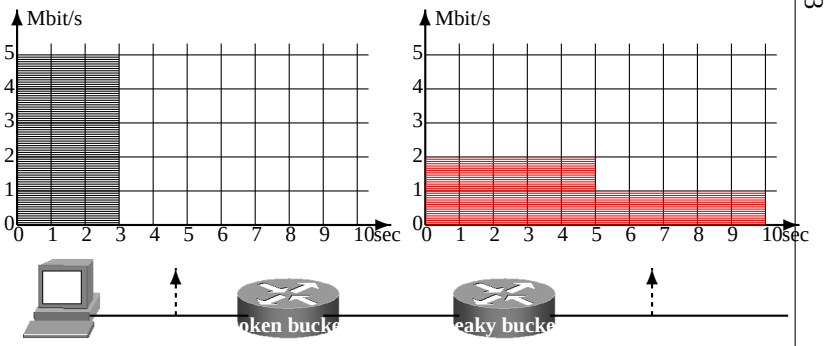
SOLUTION TO EX. 31: Token and Leaky Bucket

31.1) Draw the plot of the traffic exiting the token bucket.



Output freq.: 5 Mbit/s
Token freq.: 1 Mbit/s
Token capacity: 5 Mbit

31.2) Suppose that the provider changes the configuration putting a bit-oriented leaky bucket with capacity 9 Mbit and output frequency 2 Mbit/s immediately after the token bucket. Draw the plot of the traffic exiting the leaky bucket.

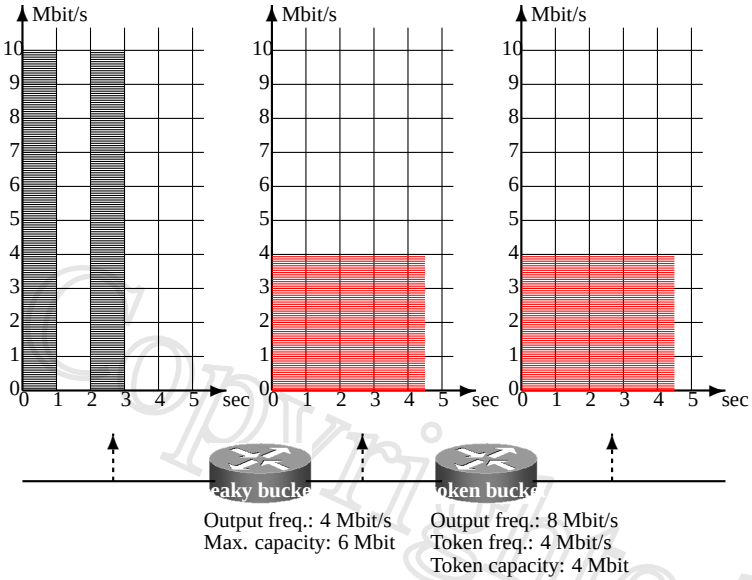


Output freq.: 5 Mbit/s
Token freq.: 1 Mbit/s
Token capacity: 5 Mbit

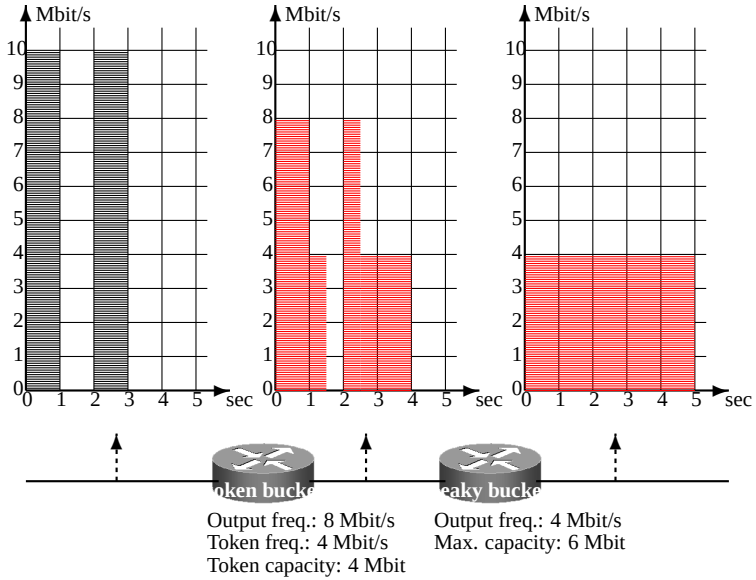
Output freq.: 2 Mbit/s
Max. capacity: 9 Mbit

SOLUTION TO EX. 32: Permutation of Leaky and Token Bucket

32.1) Draw the plots of the traffic exiting the leaky bucket and the token bucket.

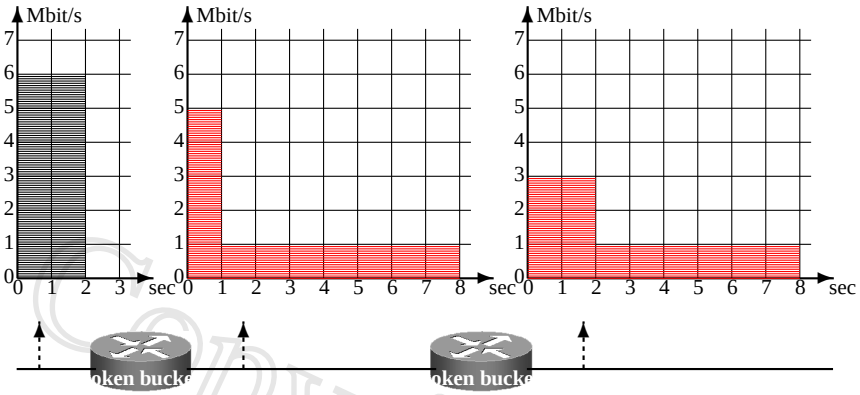


32.2) Suppose to swap the positions of the leaky and token buckets. Draw the plots of the traffic exiting the token bucket and the leaky bucket.

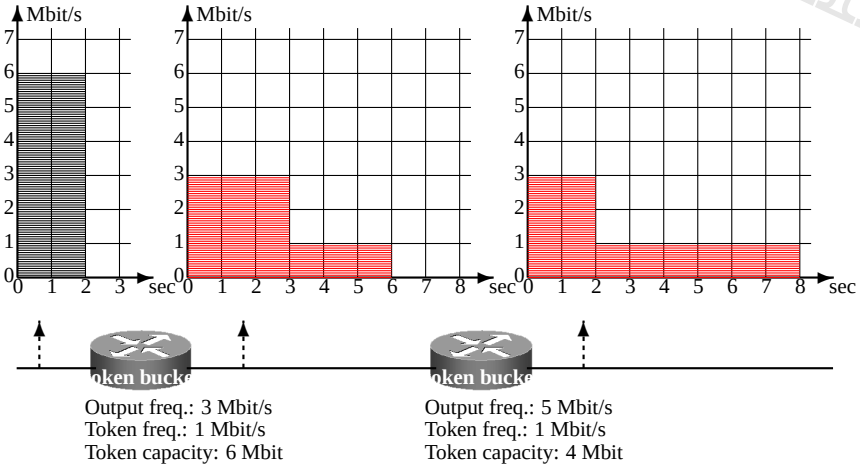


SOLUTION TO EX. 33: Two Token Buckets

33.1) Draw the plots of the traffic exiting the first and second token bucket.

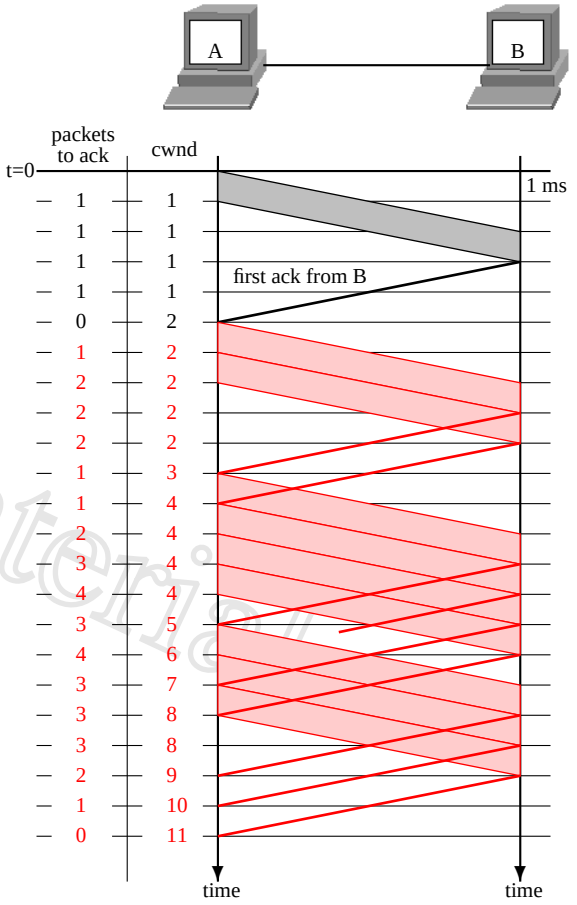


33.2) Suppose to swap the positions of the two token buckets. Draw the graphs of the traffic exiting the first and second token bucket.

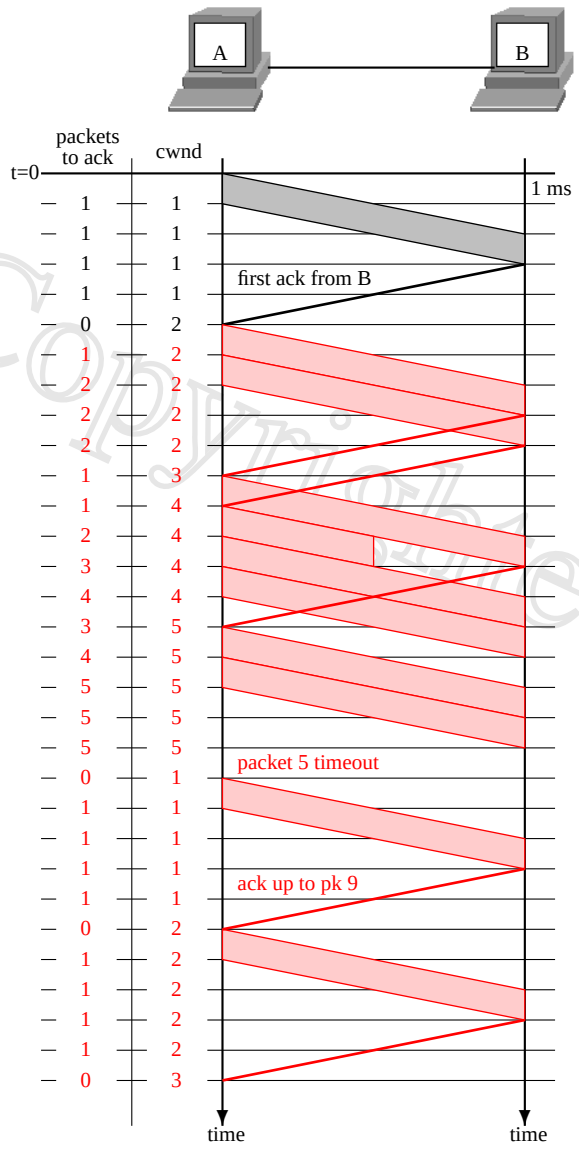


SOLUTION TO EX. 34: TCP transmission with packet loss

34.1) Case 1: the 5^{th} ack from B does not arrive to A.



34.2) Case 2: the 5^{th} packet from A does not arrive to B.



SOLUTION TO EX. 35: Interpreting a *cwnd* Graph

35.1) What is happening between $t = 0$ and $t = t_1$? What is the formula that TCP uses to increment $cwnd$?

Slow start: exponential growth.
For each received ack $cwnd = cwnd + mss$.

35.2) What happened at $t = t_1$?

The threshold `ssthresh` has been reached: TCP changes from slow start to congestion avoidance.

35.3) What is happening between $t = t_1$ and $t = t_2$? What is the formula that TCP uses to increment $cwnd$?

Congestion avoidance: linear growth.
For each received ack $cwnd = cwnd + mss^2/cwnd$.

35.4) What happened at $t = t_2$?

TCP received the third duplicate ack for the same byte. This is interpreted by TCP as an isolated packet loss: fast recovery and fast retransmit.

35.5) What formula is used by TCP to compute c_1?

TCP resets the `ssthresh` at half the minimum of the current `ssthresh` and the flow window.

35.6) What happened at $t = t_3$?

No ack received within the timeout. Slow-start starts all over again.

SOLUTION TO EX. 36: DNS resolution

36.1) What packets are exchanged on the network after **`pc.pippo.it`** asks **`dns.pippo.it`** to resolve the name **`www.pluto.it`** ?

Source IP	Destination IP	Packet type (Iterative/Recursive DNS query, DNS answer)
100.0.0.2	100.0.0.1	Recursive DNS query
100.0.0.1	100.100.100.100	Iterative DNS query
100.100.100.100	100.0.0.1	DNS answer (NS record)
100.0.0.1	200.200.200.200	Iterative DNS query
200.200.200.200	100.0.0.1	DNS answer (NS record)
100.0.0.1	200.0.0.1	Iterative DNS query
200.0.0.1	100.0.0.1	DNS answer (A record)
100.0.0.1	100.0.0.2	DNS answer (A record)

36.2) Immediately after the DNS query of the previous exercise, `pc.pippo.it` asks `dns.pippo.it` to resolve `ftp.pluto.it`. What packets are exchanged on the network?

Source IP	Destination IP	Packet type (Iterative/Recursive DNS query, DNS answer)
100.0.0.2	100.0.0.1	Recursive DNS query
100.0.0.1	200.0.0.1	Iterative DNS query
200.0.0.1	100.0.0.1	DNS answer (A record)
100.0.0.1	100.0.0.2	DNS answer (A record)

SOLUTION TO EX. 37: Browsers and DNS

	Domain name	Time to live	Class	Type	Value (IP address)
1	com	172800	IN	NS	100.0.0.2
2	firm.com	172800	IN	NS	100.0.0.3
3	www.firm.com	172800	IN	A	100.0.0.6
4	www.com	172800	IN	A	100.0.0.7
5	buy.firm.com	172800	IN	NS	100.0.0.4
6	www.buy.firm.com	172800	IN	A	100.0.0.8
7	sell.firm.com	172800	IN	NS	100.0.0.5
8	www.sell.firm.com	172800	IN	A	100.0.0.9
9	web.sell.firm.com	172800	IN	A	100.0.0.10

SOLUTION TO EX. 38: Emails and DNS

38.1) What messages are exchanged at the application layer when a user on `pc1.aaa.org` writes an email and his mail user agent delivers it to the outgoing mail server?

Source IP	Destination IP	Packet type (mark the right cell)			
		Recursive DNS query	Iterative DNS query	DNS answer	SMTP mail sending
10.0.0.1	10.0.0.3	X			
10.0.0.3	10.0.0.5		X		
10.0.0.5	10.0.0.3			X	
10.0.0.3	10.0.0.6		X		
10.0.0.6	10.0.0.3			X	
10.0.0.3	10.0.0.7		X		
10.0.0.7	10.0.0.3			X	
10.0.0.3	10.0.0.1			X	
10.0.0.1	10.0.0.9				X

38.2) Few instants after the events in the previous exercise, `pc2.aaa.org` issues an HTTP GET request to `www.bbb.com`, which returns an HTTP response. What messages are exchanged at the application layer?

Source IP	Destination IP	Packet type (mark the right cell)			
		Recursive DNS query	Iterative DNS query	DNS answer	HTTP request or response
10.0.0.2	10.0.0.3	X			
10.0.0.3	10.0.0.7		X		
10.0.0.7	10.0.0.3			X	
10.0.0.3	10.0.0.2			X	
10.0.0.2	10.0.0.8				X
10.0.0.8	10.0.0.2				X

38.3) What packets are exchanged in the network at layer 2 and 3 during the events described in the previous exercise? List them all up to the first TCP SYN packet of the HTTP connection.

Layer 2		Layer 3 (if any)		ARP request	ARP reply	Recursive DNS query	Iterative DNS query	DNS answer	TCP SYN
Dest MAC	Source MAC	Source IP	Destination IP						
bcast	02			X					
02	03				X				
03	02	10.0.0.2	10.0.0.3			X			
07	03	10.0.0.3	10.0.0.7				X		
03	07	10.0.0.7	10.0.0.3					X	
02	03	10.0.0.3	10.0.0.2					X	
bcast	02			X					
02	08				X				
08	02	10.0.0.2	10.0.0.8						X

SOLUTION TO EX. 39: Digging into DNS

39.1) Who is `K.ROOT-SERVERS.NET`, and what role does it play in the query?

`K.ROOT-SERVERS.NET` is the DNS root server to which `dig` asks to resolve the `edu` Top Level Domain (TLD), as seen in line 25 of the response.

39.2) Who are `a.gtld-servers.net` and `dark.brown.edu`, and what role do they play in the query?

`a.gtld-servers.net` is the name server for the `edu` domain to which `dig` asks to resolve the `brown.edu` domain, as seen in line 29 of the response. Similarly, `dark.brown.edu` is the name server for the `brown.edu` domain which is asked to resolve `www.cs.brown.edu` (line 34).

39.3) Explain line 30 of the response in detail.

Line 30 contains the final DNS reply which maps `www.cs.brown.edu` to its IP address. The resource record is of class IN (standing for Internet) and type A (meaning that the record contains an address). The resulting IP address is 128.148.32.110, and the resource record is set to be cached for 86400 seconds.

39.4) What can be concluded by analyzing lines 31, 32, and 33 of the response?

Besides the A record, `dark.brown.edu` also provided additional NS records to the resolver, listing the authoritative name servers for the `cs.brown.edu` subdomain.

39.5) Explain the meaning of lines 4 through 17.

The DNS server 193.204.161.85 is asked to provide a NS record for the root domain. It replies with a list of all the DNS root servers that he knows. There are multiple DNS root servers so that the load of processing DNS queries can be shared among them.

SOLUTION TO EX. 40: A Network with Routers, Switches, and Hubs

40.1) Fill in the routing table of router 1.

router 1				
Network	**Netmask**	**Interface**	**Next hop**	**Cost**
10.0.1.0	255.255.255.0	B	d.c.	0
10.0.2.0	255.255.255.0	C	10.0.4.3	1
10.0.3.0	255.255.255.0	C	10.0.4.3	1
10.0.4.0	255.255.255.0	C	d.c.	0
10.0.5.0	255.255.255.0	A	d.c.	0
10.0.6.0	255.255.255.0	C	10.0.4.254	1
10.0.7.0	255.255.255.0	B	10.0.1.254	1

40.2) Which links are traversed by a packet sent by PC 1 to PC 2 (specify the sequence of link labels)?

A	C	D	G	L	Z	V	U			

40.3) Suppose that the administrator turns off hub 1 for maintenance. Show the new stable configuration reached by the routing table of router 1.

The routing table does not change. The solution is the same of exercise **40.1**.

40.4) With the same hypotheses of question **40.3** (hub 1 is turned off), which links are traversed by a packet sent by PC 1 towards PC 2 (specify the sequence of link labels)?

A	C	D	H	N	M	Z	V	U		

40.5) Hub 1 turns out to be definitely out of order. Furthermore, the network administrator disconnects by error link N. In these new conditions, which links are traversed in the stable stete by a packet sent by PC 1 to PC 2 (specify the sequence of link labels)?

A	C	D	H	O	P	Q	S	T	U	

40.6) In addition to the disasters described above (hub 1 turned off and link N disconnected), link H is cut by error. In these conditions, which links are traversed by a packet sent by PC 1 towards PC 2 (specify the sequence of link labels)?

A	B	E	F	P	Q	S	T	U		

SOLUTION TO EX. 41: Scenario with WiFi

41.1) Fill in the routing tables of router 1 and router 2, without using default routes, such that there is full connectivity between all devices of the network.

router 1			
Network	**Netmask**	**Interface**	**Next hop**
10.0.0.0	255.255.255.0	C1	d.c.
10.0.1.0	255.255.255.0	C2	d.c.
10.0.2.0	255.255.255.0	C2	10.0.1.2
10.0.3.0	255.255.255.0	C2	10.0.1.2

router 2			
Network	**Netmask**	**Interface**	**Next hop**
10.0.0.0	255.255.255.0	D3	10.0.1.1
10.0.1.0	255.255.255.0	D3	d.c.
10.0.2.0	255.255.255.0	D2	d.c.
10.0.3.0	255.255.255.0	D1	d.c.

41.2) List the devices that will receive the bits of a broadcast packet (e.g., an arp request) sent by one of the following hosts.

	host sending the broadcast packet		
	PC 1	**PC 3**	**PC 4**
Devices receiving the bits	Router 1	switch 1, switch 5, switch 6, hub 4, ap 1, PC 2, router 2.	switch 2, switch 3, switch 4, router 1, router 2, hub 1, hub 4, PC 5.

41.3) Host PC 1 sends an echo request packet to host PC 2. How does the packet appear on the following links of the network?

41.3.1) IEEE 802.3 packet on link (A)

MAC dst	**MAC src**	**IP src**	**IP dst**	**payload**
C1	11	10.0.0.11	10.0.3.22	echo request

41.3.2) IEEE 802.3 packet on link (B)

MAC dst	MAC src	IP src	IP dst	payload
D3	C2	10.0.0.11	10.0.3.22	echo request

41.3.3) IEEE 802.3 packet on link (C)

The packet is the same of question **41.3.2**.

41.3.4) IEEE 802.3 packet on link (D)

MAC dst	MAC src	IP src	IP dst	payload
22	D2	10.0.0.11	10.0.3.22	echo request

41.3.5) IEEE 802.11 packet on link (E)

Address				DS		IP		pay load
1	2	3	4	To	From	Src	Dst	
22	A1	D2		0	1	10.0.0.11	10.0.3.22	echo re-quest

SOLUTION TO EX. 42: Scenario with Distance Vector

42.1) Suppose the network reaches equilibrium. Fill in the distance vector sent by router 2 to its neighbors, the output of the traceroute command from PC 1 to PC 2, and the link traversed by a packet sent from PC 1 to PC 2.

router 2 distance vector		
Network	Netmask	Distance
10.0.0.0	255.255.255.0	1
10.0.1.0	255.255.255.0	0
10.0.2.0	255.255.255.0	0
10.0.3.0	255.255.255.0	1
10.0.4.0	255.255.255.0	1
10.0.5.0	255.255.255.0	2

traceroute from PC 1 to PC 2
10.0.0.1
10.0.4.3
10.0.3.4
10.0.5.22

Links traversed from PC 1 to PC 2:

A	I	M	N	J	D	K	O	L	F	G	H		

42.2) You turn off switch 02 and switch 05 and let the network reach equilibrium. Fill in the distance vector sent by router 2 to its neighbors, the output of the traceroute command from PC 1 to PC 2, and the link traversed by a packet sent from PC 1 to PC 2.

Both the distance vector sent by router 2 and the output of the traceroute command from PC 1 to PC 2 stay unchanged. The solution is the same of that of exercise **42.1**.

Links traversed from PC 1 to PC 2:

A	I	P	U	V	R	J	D	E	F	G	H		

42.3) In addition to switch 02 and switch 05 you turn off also switch 03 and let the network reach equilibrium. Fill in the distance vector sent by router 2 to its neighbors, the output of the traceroute command from PC 1 to PC 2, and the link traversed by a packet sent from PC 1 to PC 2.

The distance vector sent by router 2 stays unchanged.

router 2 distance vector		
Network	**Netmask**	**Distance**
10.0.0.0	255.255.255.0	1
10.0.1.0	255.255.255.0	0
10.0.2.0	255.255.255.0	0
10.0.3.0	255.255.255.0	1
10.0.4.0	255.255.255.0	1
10.0.5.0	255.255.255.0	2

traceroute
from PC 1 to PC 2
10.0.0.1
10.0.1.23
10.0.2.3
10.0.3.4
10.0.5.22

Links traversed from PC 1 to PC 2:

A	B	C	D	E	F	G	H						

SOLUTION TO EX. 43: Scenario with Link Load

43.1) What lines are reached by a broadcast packet (for example, an ARP request) sent by server SA?

All the lines but line (A). Lines (B), (C), (D), (E), and (F).

43.2) What lines are reached by a unicast packet coming from lan A and directed to server SA?

All the lines but line (B). Lines (A), (D), (E), (F), and (C).

43.3) Suppose that each IP interfaces (assume bridges have no IP interface) produces 0.01 Mbit/s of brodacast traffic. Fill in the table of the broadcast load (Mbit/s) that can be expected on the various lines.

Line	Broadcast Load
A	0.06 Mbit/s
B	0.09 Mbit/s
C	0.09 Mbit/s

Line	Broadcast Load
D	0.09 Mbit/s
E	0.09 Mbit/s
F	0.09 Mbit/s

43.4) Suppose that server SA sends to each host labeled "A" 0.1 Mbit/s of unicast traffic. Fill in the table of the unicast load (Mbit/s) that can be expected on the various lines.

Line	Unicast Load
A	0.2 Mbit/s
B	0.2 Mbit/s
C	0.5 Mbit/s

Line	Unicast Load
D	0.2 Mbit/s
E	0.5 Mbit/s
F	0.5 Mbit/s

43.5) You have to add a server SB that will send to each host labeled "B" 1 Mbit/s of unicast traffic.

Fill in the table of the total (unicast + broadcast) load (Mbit/s) that can be expected on the various lines when server SB is added to lan (A).

Line	Total Load
A	0.96 Mbit/s
B	0.49 Mbit/s
C	0.99 Mbit/s

Line	Total Load
D	0.69 Mbit/s
E	0.99 Mbit/s
F	0.99 Mbit/s

Fill in the table of the total (unicast + broadcast) load (Mbit/s) that can be expected on the various lines when server SB is added to lan (B).

Line	Total Load
A	0.56 Mbit/s
B	0.99 Mbit/s
C	1.09 Mbit/s

Line	Total Load
D	0.59 Mbit/s
E	1.09 Mbit/s
F	1.09 Mbit/s

43.6) What of the two options above is preferable, if you want to keep the total load on the links below the threshold of 1 Mbit/s?

Add server SB to lan (A).

Chapter 15

Glossary

BPDU	Bridge Protocol Data Unit: the packets sent by IEEE 802.1D bridges to compute the Spanning Tree Algorithm (STA).
CIDR	Classless Inter-Domain Routing
CSMA/CD	Carrier Sense, Multiple Access, Collision Detection, a well known paradigm for asyncronous transmission on a shared channel
LAN	Local Area Network: a network whose hosts assume to be able to directly exchange packets. This would imply that each pair of hosts of the LAN shares some transmission medium. Generally, the latter condition is relaxed to include scenarios where hosts can reach each other through some transparent devices as hubs or switches. The IEEE 802.1D standards calls LAN each collision domain of a Local Area Network.
MAC	Medium Access Control: a sublayer of layer two of the protocol stack. This sublayer is in charge of managing the delivery of a packet to an adjacent station.
MTU	Maximum Transmission Unit: the maximum size that a packet of a specific protocol can have.
NIC	Network Interface Card: the hardware card that attaches an host to a network medium.
PDU	Protocol Data Unit: a packet (data unit) of a specific protocol
STA	Spanning Tree Algorithm
UTP	Unshielded Twisted Pair