

## Exercise Sheet 4

### Exercise 1 (Routers, Layer-3-Switches, Gateways)

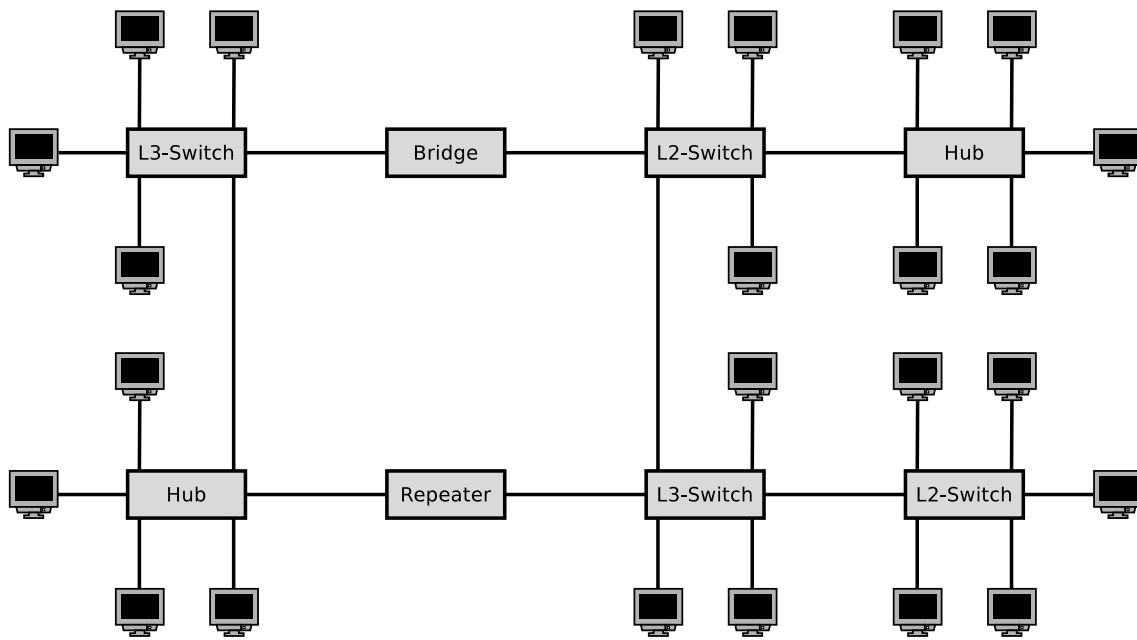
1. What is the purpose of **Routers** in computer networks?  
(Also explain the difference to Layer-3-Switches.)
2. What is the purpose of **Layer-3-Switches** in computer networks?  
(Also explain the difference to Routers.)
3. What is the purpose of **Gateways** in computer networks?
4. Why are **Gateways** in the network layer of computer networks seldom required nowadays?

### Exercise 2 (Collision Domain, Broadcast Domain)

1. Which devices **divide the collision domain**?  

<input type="checkbox"/> Repeater	<input type="checkbox"/> Bridge	<input type="checkbox"/> Router
<input type="checkbox"/> Hub	<input type="checkbox"/> Layer-2-Switch	<input type="checkbox"/> Layer-3-Switch
2. Which devices **divide the broadcast domain**?  

<input type="checkbox"/> Repeater	<input type="checkbox"/> Bridge	<input type="checkbox"/> Router
<input type="checkbox"/> Hub	<input type="checkbox"/> Layer-2-Switch	<input type="checkbox"/> Layer-3-Switch
3. Sketch in the diagram of the network topology all **collision domains** and all **broadcast domains**.



### Exercise 3 (Addressing in the Network Layer)

1. What is the meaning of **Unicast** in the network layer of computer networks?
2. What is the meaning of **Broadcast** in the network layer of computer networks?
3. What is the meaning of **Anycast** in the network layer of computer networks?
4. What is the meaning of **Multicast** in the network layer of computer networks?
5. Why contains the IPv4 **address space** only 4,294,967,296 addresses?
6. Why was **Classless Interdomain Routing (CIDR)** introduced?
7. Describe in simple words the **functioning of CIDR**.

*Focus on the way, IP addresses are treated and subnets are created.*

### Exercise 4 (Addressing in the Network Layer)

Calculate for each subtask of this exercise the **first and last host addresses**, the **network address** and the **broadcast address** of the subnet.

```

IP Address:      151.175.31.100    10010111.10101111.00011111.01100100
Subnet mask:     255.255.254.0     11111111.11111111.11111110.00000000
Network address? ---.---.---.---   -----.-----.-----.-----
First host address? ---.---.---.---   -----.-----.-----.-----
Last host address? ---.---.---.---   -----.-----.-----.-----
Broadcast address? ---.---.---.---   -----.-----.-----.-----

IP Address:      151.175.31.100    10010111.10101111.00011111.01100100
Subnet mask:     255.255.255.240    11111111.11111111.11111111.11110000
Network address? ---.---.---.---   -----.-----.-----.-----
First host address? ---.---.---.---   -----.-----.-----.-----
Last host address? ---.---.---.---   -----.-----.-----.-----
Broadcast address? ---.---.---.---   -----.-----.-----.-----

IP Address:      151.175.31.100    10010111.10101111.00011111.01100100
Subnet mask:     255.255.255.128    11111111.11111111.11111111.10000000
Network address? ---.---.---.---   -----.-----.-----.-----
First host address? ---.---.---.---   -----.-----.-----.-----
Last host address? ---.---.---.---   -----.-----.-----.-----
Broadcast address? ---.---.---.---   -----.-----.-----.-----

```

binary representation	decimal representation	binary representation	decimal representation
10000000	128	11111000	248
11000000	192	11111100	252
11100000	224	11111110	254
11110000	240	11111111	255

## Exercise 5 (Addressing in the Network Layer)

In each subtask of this exercise, a sender transmits an IP packet to a receiver. Calculate for each subtask the **subnet ID of sender and receiver** and specify whether the IP packet **leaves the subnet during transmission** or not.

```

Sender:          11001001.00010100.11011110.00001101    201.20.222.13
Subnet mask:     11111111.11111111.11111111.11110000    255.255.255.240

Receiver:        11001001.00010100.11011110.00010001    201.20.222.17
Subnet mask:     11111111.11111111.11111111.11110000    255.255.255.240

```

Subnet ID of sender?

Subnet ID of receiver?

Does the IP packet leave the subnet [yes/no]?

Sender:           00001111.11001000.01100011.00010111     15.200.99.23  
Subnet mask:     11111111.11000000.00000000.00000000     255.192.0.0

Receiver:        00001111.11101111.00000001.00000001     15.239.1.1  
Subnet mask:     11111111.11000000.00000000.00000000     255.192.0.0

Subnet ID of sender?

Subnet ID of receiver?

Does the IP packet leave the subnet [yes/no]?

## Exercise 6 (Addressing in the Network Layer)

Calculate for each subtask of this exercise the **subnet masks** and answer the **questions**.

1. Split the class C network 195.1.31.0 for implementing 30 subnets.

Network ID:     11000011.00000001.00011111.00000000     195.1.31.0  
Number of bits for subnet IDs?  
Subnet mask:    -----·-----·-----·-----     ---·---·---·---  
Number of bits for host IDs?  
Number of host IDs per subnet?

2. Split the class A network 15.0.0.0 for implementing 333 subnets.

Network ID:     00001111.00000000.00000000.00000000     15.0.0.0  
Number of bits for subnet IDs?  
Subnet mask:    -----·-----·-----·-----     ---·---·---·---  
Number of bits for host IDs?  
Number of host IDs per subnet?

3. Split the class B network 189.23.0.0 for implementing 20 subnets.

Network ID:     10111101.00010111.00000000.00000000     189.23.0.0  
Number of bits for subnet IDs?  
Subnet mask:    -----·-----·-----·-----     ---·---·---·---  
Number of bits for host IDs?  
Number of host IDs per subnet?

4. Split the class C network 195.3.128.0 into subnets, which contain 17 hosts each.

Network ID: 11000011.00000011.10000000.00000000 195.3.128.0  
Number of bits for host IDs?  
Number of bits for subnet IDs?  
Number of possible subnets?  
Subnet mask: . . . . .

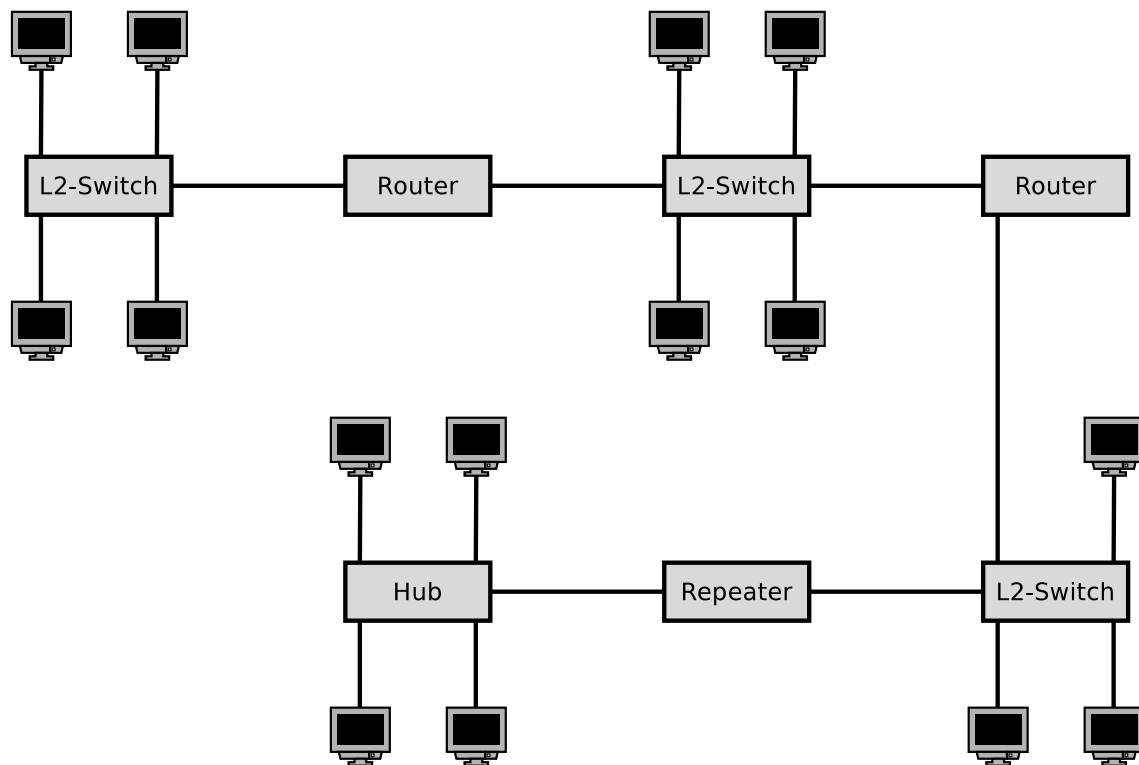
5. Split the class B network 129.15.0.0 into subnets, which contain 10 hosts each.

Network ID: 10000001.00001111.00000000.00000000 129.15.0.0  
Number of bits for host IDs?  
Number of bits for subnet IDs?  
Number of possible subnets?  
Subnet mask: . . . . .

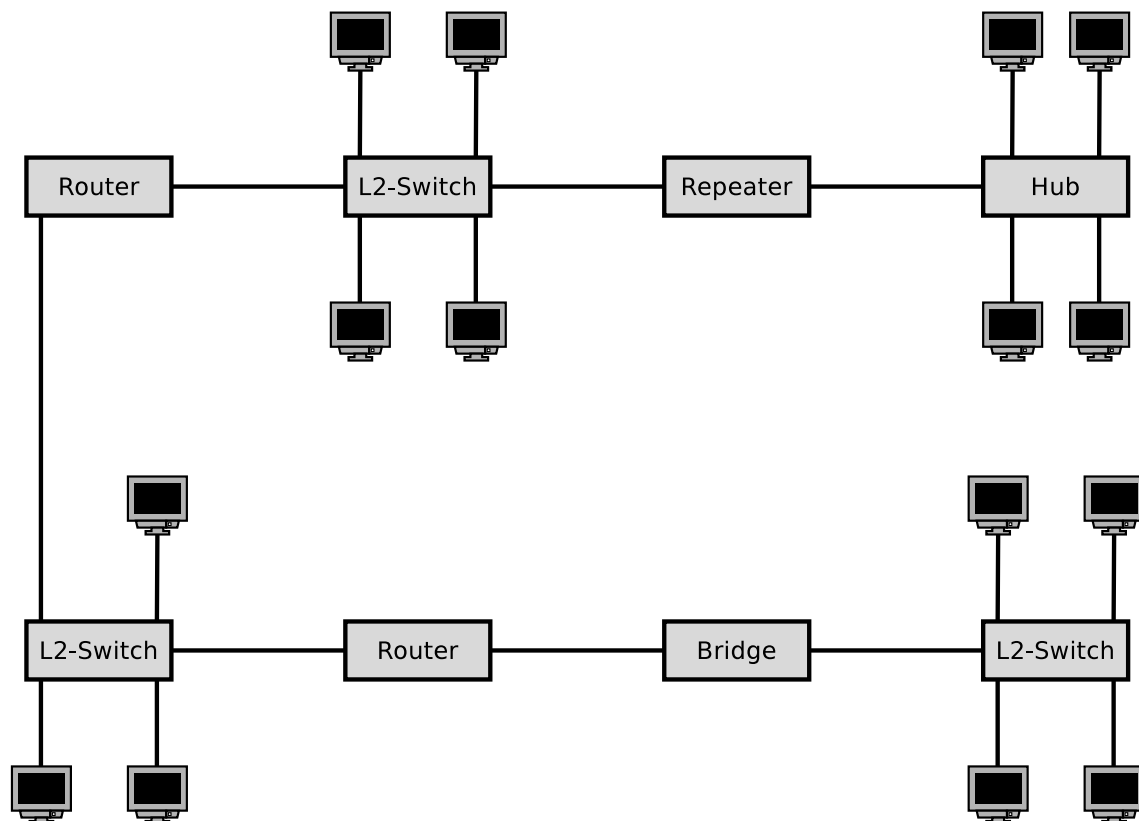
binary representation	decimal representation	binary representation	decimal representation
10000000	128	11111000	248
11000000	192	11111100	252
11100000	224	11111110	254
11110000	240	11111111	255

## Exercise 7 (Collision Domain, Broadcast Domain)

1. Sketch in the diagram of the network topology all **collision domains** and all **broadcast domains**.

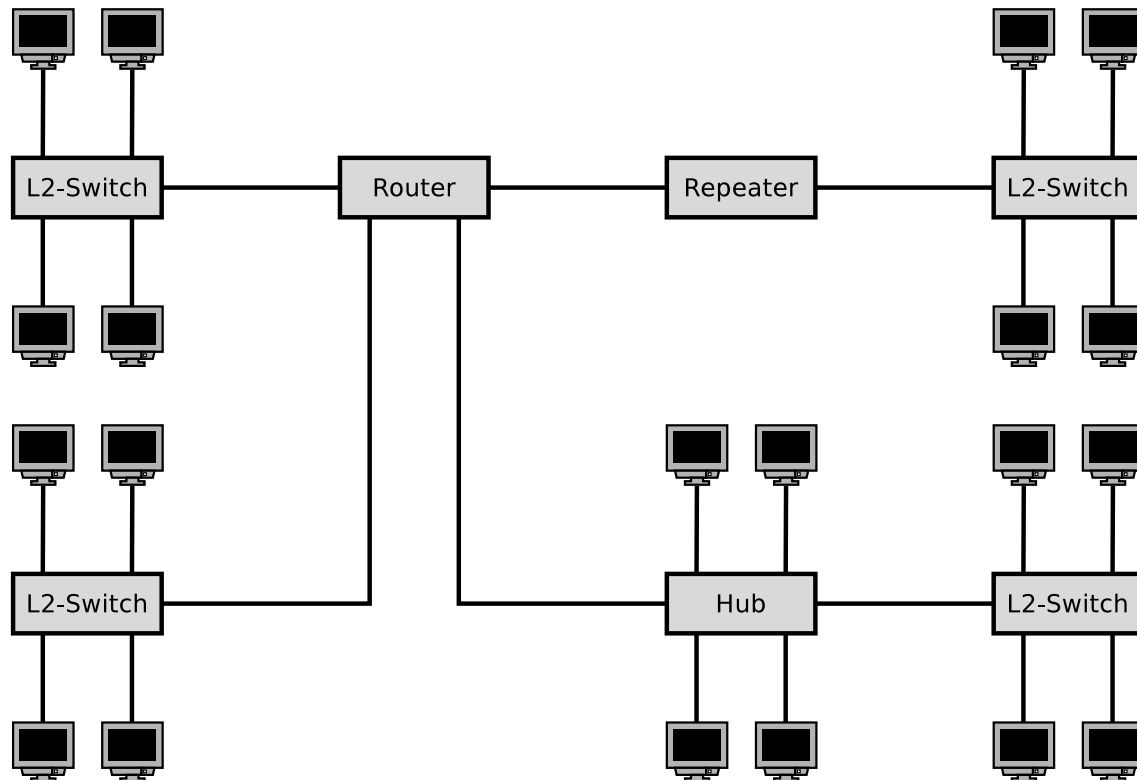


2. Sketch in the diagram of the network topology all **collision domains** and all **broadcast domains**.



## Exercise 8 (Broadcast Domain)

1. Sketch in the diagram of the network topology all **broadcast domains**.
2. What is the **required number of subnets** for this network topology?



## Exercise 9 (Private IP Address Spaces)

Name the three private IPv4 address spaces.

## Exercise 10 (Addressing in the Network Layer)

Calculate for each network configuration in the table whether an IP packet, which is send from the given IP address to the destination address, **leaves the subnet during transmission** or not.

IP address	Subnet mask	Destination address	Leaves the subnet [yes/no]
201.20.222.13	255.255.255.240	201.20.222.17	
15.200.99.23	255.192.0.0	15.239.1.1	
172.21.23.14	255.255.255.0	172.21.24.14	
210.5.16.198	255.255.255.252	210.5.16.197	
210.5.16.198	255.255.255.252	210.5.16.201	
5.5.5.5	255.254.0.0	5.6.6.6	

(A part of the solution is the calculations performed. Where no calculation is required, you need to give a reason for your answer. Answering the question with just „yes“ or „no“ is not sufficient!)

Source: Jörg Roth. Prüfungstrainer Rechnernetze. Vieweg (2010)

## Exercise 11 (Addressing in the Network Layer)

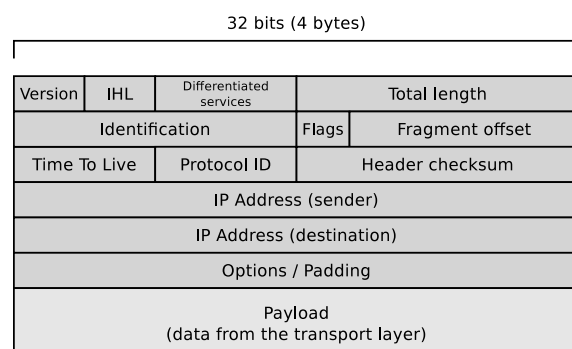
Specify for each subtask of this exercise the correct **subnet mask**.

1. A maximum number of subnets with 5 hosts each in a class B network.
2. 50 subnets with 999 hosts each in a class B network.
3. 12 subnets with 12 hosts each in a class C network.

Source: Jörg Roth. Prüfungstrainer Rechnernetze. Vieweg (2010)

## Exercise 12 (Checksums in IP Packets)

The figure shows the structure of IPv4 packets as discussed in the computer networks course.



The given data in hexadecimal notation is a truncated excerpt of an IP packet:

4500 0034 B612 4000 4006 6F80 0A00 008B 5BC6 AEE0

The data contains the values of the fields of the IP packet header.



4           = Version  
5           = IHL = IP Header Length ( $\Rightarrow 5 * 4$  Byte words = 20 bytes)  
00          = Differentiated services  
0034       = Total length ( $\Rightarrow 52$  bytes)  
B612       = Identification  
4000       = Flags + Fragment offset  
40          = Time To live ( $\Rightarrow 62$  hops)  
06          = Protocol ID ( $\Rightarrow$  TCP)  
6F80       = Header Checksum  
0A00 008B = IP address (sender)  
5BC6 AEE0 = IP address (destination)

### Example: Calculate checksum

RFC 791 says on page 14: „*The checksum field is the 16 bit one's complement of the one's complement sum of all 16 bit words in the header.*“.

To calculate the checksum of the packet, the sum of each 2 byte word inside the header must be calculated. The checksum field itself is skipped here!

$4500 + 0034 + B612 + 4000 + 4006 + 0A00 + 008B + 5BC6 + AEE0 = 2907D$

Next, the result of the calculation is converted to binary:

$2907D \Rightarrow 10\ 1001\ 0000\ 0111\ 1101$

The first two bits are the carry and need to be added to the rest of the value:

$10 + 1001\ 0000\ 0111\ 1101 = 1001\ 0000\ 0111\ 1111$

Next, every bit of the result is flipped to obtain the checksum:

1001 0000 0111 1111  
 $\Rightarrow 0110\ 1111\ 1000\ 0000$

The result 0110 1111 1000 0000 is equal to the value 6F80 in hexadecimal notation, as already shown in the original IP packet header.

### Example: Verify checksum

RFC 791 says on page 14: „*For purposes of computing the checksum, the value of the checksum field is zero.*“.

To verify a checksum, the same procedure is used as above, with a single exception: The original header checksum is not omitted.

$4500 + 0034 + B612 + 4000 + 4006 + 6F80 + 0A00 + 008B + 5BC6 + AEE0 = 2FFFD$

Next, the result of the calculation is converted to binary:

$2\text{FFFD} \Rightarrow 10\ 1111\ 1111\ 1111\ 1101$

The first two bits are the carry and need to be added to the rest of the value:

$10 + 1111\ 1111\ 1111\ 1101 = 1111\ 1111\ 1111\ 1111$

Next, every bit of the result is flipped:

$1111\ 1111\ 1111\ 1111$   
 $\Rightarrow 0000\ 0000\ 0000\ 0000$

This indicates: No error detected! Any result, which is  $\neq 0$  indicates: Error!

*Source: RFC 791 and Wikipedia*

1. **Calculate** the checksum for each IP header:

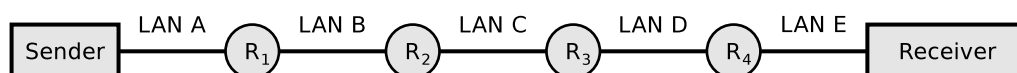
- 4500 0034 4C22 4000 F706 ???? C163 9055 0A00 008B
- 4500 0034 671E 4000 4006 ???? 0A00 008b C163 9055
- 4500 00F2 0000 4000 4011 ???? 0A00 008b 0A00 00FF

2. **Verify** the checksum of each IP header:

- 4500 0034 02FD 4000 3606 276C 6CA0 A330 0A00 008B
- 4500 00E7 02FC 4000 3606 37BC 6CA0 A330 0A00 008B
- 4500 0034 A9D5 4000 4006 814E 0A00 008B adC2 4613

## Exercise 13 (Fragmenting IP Packets)

4,000 bytes payload need to be transmitted via the IP protocol. The payload must be fragmented, because it is transmitted over multiple physical networks, whose MTU is  $< 4,000$  bytes.



	LAN A	LAN B	LAN C	LAN D	LAN E
Network technology	Ethernet	PPPoE	ISDN	Ethernet	WLAN
MTU [bytes]	1,500	1,492	576	1,400	2,312
IP-Header [bytes]	20	20	20	20	20
maximum payload [bytes]	1,480	1,472	556	1,380	2,292

Display graphically the way, the payload is fragmented, and how many bytes of payload each fragment contains.

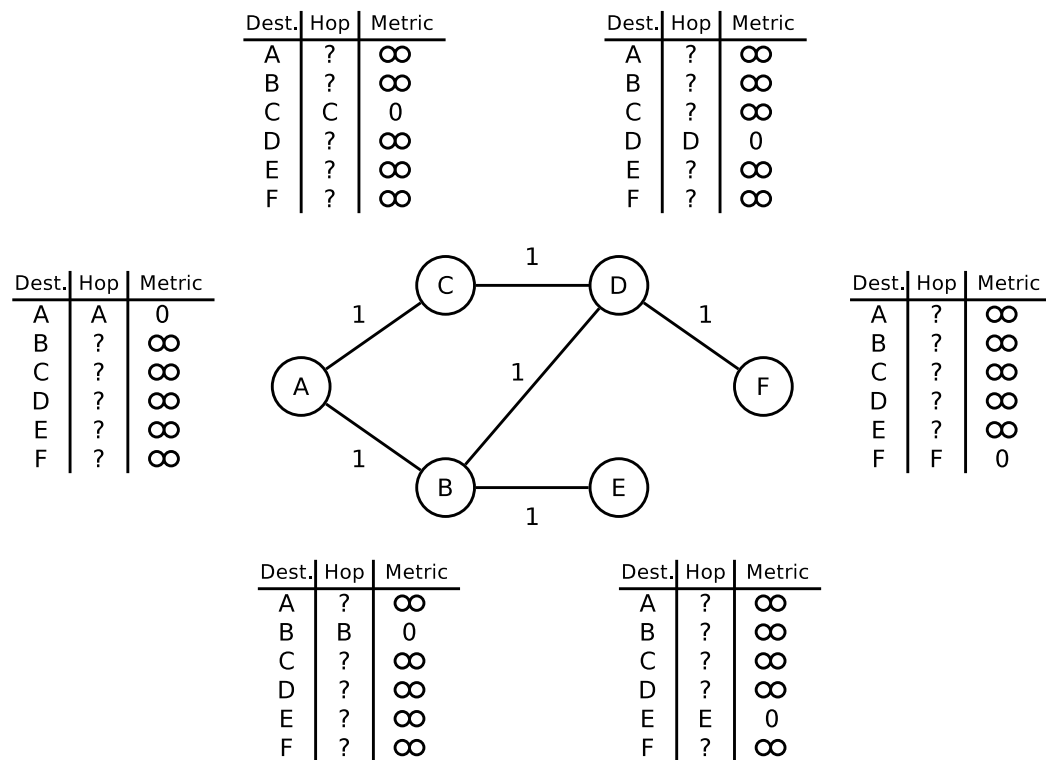
## Exercise 14 (Forwarding and Path Calculation)

1. Which two major classes of **routing protocols** exist?
2. Which **algorithms for best path calculation** implement the routing protocol classes from subtask 1?
3. What is an **autonomous system**?
4. The **Border Gateway Protocol** (BGP) is a protocol for...  
☐ Intra-AS routing      ☐ Inter-AS routing
5. Which **routing protocol class** from subtask 1 implements the BGP?
6. **Open Shortest Path First** (OSPF) is a protocol for...  
☐ Intra-AS routing      ☐ Inter-AS routing
7. Which **routing protocol class** from subtask 1 implements OSPF?
8. The **Routing Information Protocol** (RIP) is a protocol for...  
☐ Intra-AS routing      ☐ Inter-AS routing
9. Which **routing protocol class** from subtask 1 implements the RIP?
10. When RIP is used, each Router communicates only with its **direct neighbors**. What are the **advantages** and **drawbacks** of method?
11. When RIP is used, the path cost (metric) depend only on the number of Routers (**hops**), which need to be passed on the way to the destination network. What is the **drawback** of this method?
12. When OSPF is used, **all Routers** communicate with each other. What are the **advantages** and **drawbacks** of method?

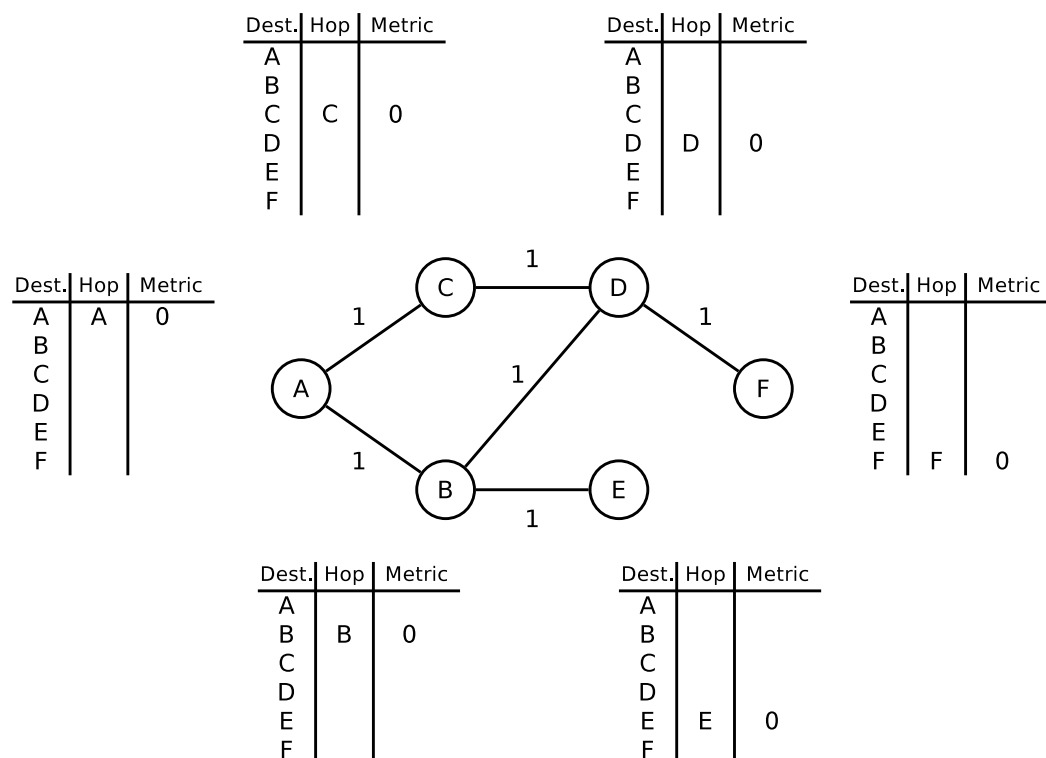
## Exercise 15 (Bellman-Ford Algorithm)

1. **Calculate** the entries of the routing tables for each advertisement round of the Routing Information Protocol (RIP).  
(The *hop metric* is used.)

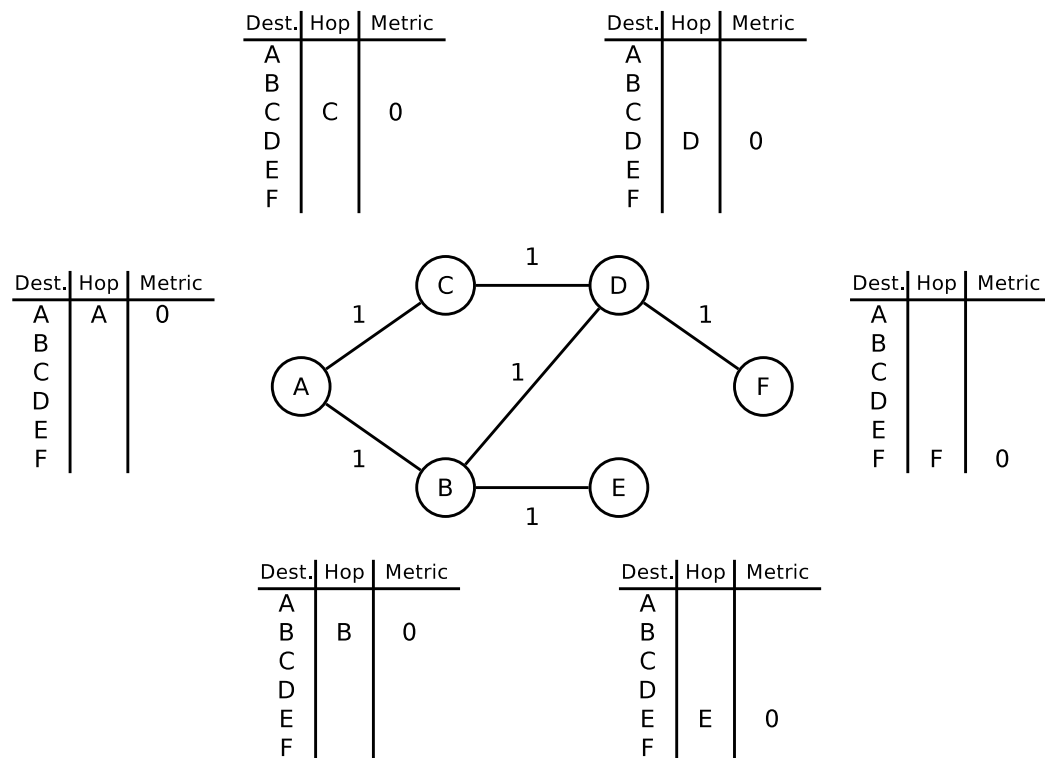
## Step 1



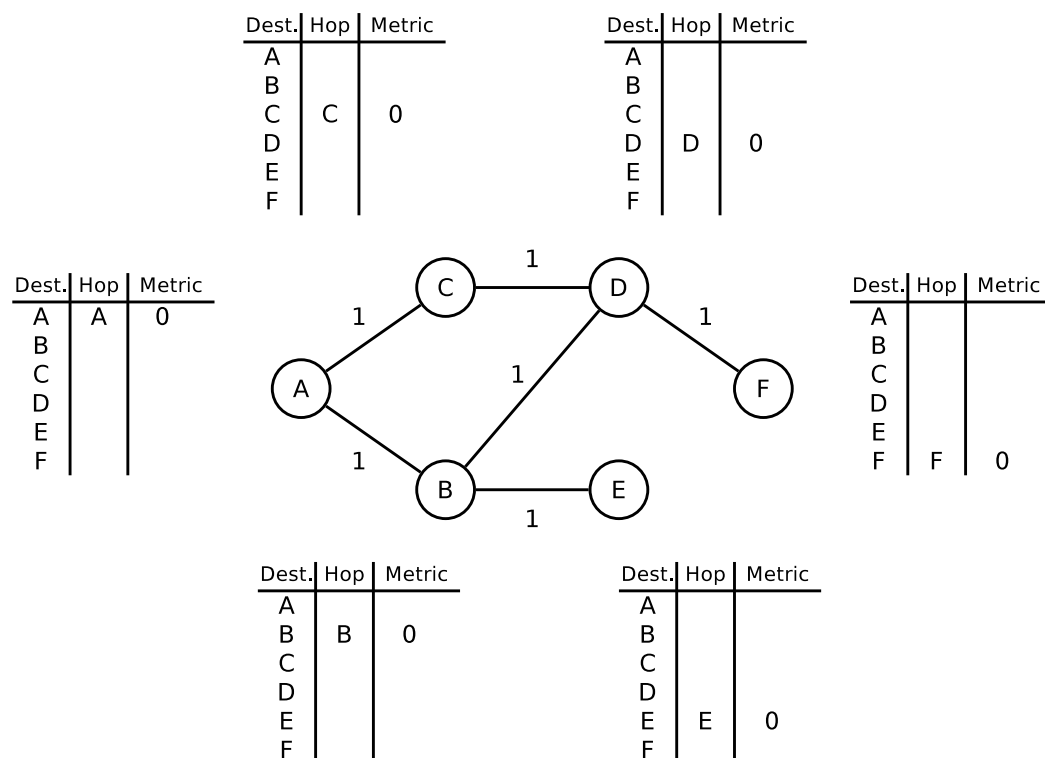
## Step 2



### Step 3



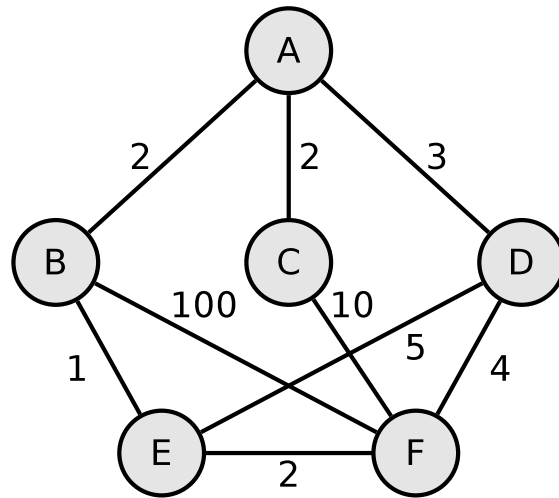
### Step 4



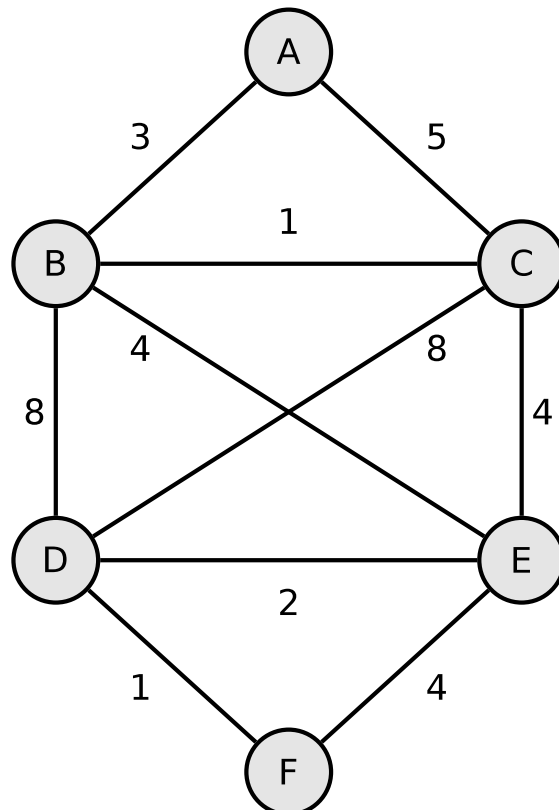
## Exercise 16 (Dijkstra's Algorithm)

1. Calculate the shortest path from node A to all other nodes using Dijkstra's algorithm.

Source: Jörg Roth. *Prüfungstrainer Rechnernetze*. Vieweg (2010)



2. Calculate the shortest path from node A to all other nodes using Dijkstra's algorithm.



## Exercise 17 (Internet Control Message Protocol)

1. What is the **function** of the Internet Control Message Protocol (ICMP)?
2. Give two examples for **command line tools**, which use ICMP.

## Exercise 18 (IPv6)

1. Simplify these IPv6 addresses:

- 1080:0000:0000:0000:0007:0700:0003:316b

*Solution:* \_\_\_\_\_

- 2001:0db8:0000:0000:f065:00ff:0000:03ec

*Solution:* \_\_\_\_\_

- 2001:0db8:3c4d:0016:0000:0000:2a3f:2a4d

*Solution:* \_\_\_\_\_

- 2001:0c60:f0a1:0000:0000:0000:0000:0001

*Solution:* \_\_\_\_\_

- 2111:00ab:0000:0004:0000:0000:0000:1234

*Solution:* \_\_\_\_\_

2. Provide all positions of these simplified IPv6 addresses:

- 2001::2:0:0:1

*Solution:* \_\_\_\_:\_\_\_\_:\_\_\_\_:\_\_\_\_:\_\_\_\_:\_\_\_\_:\_\_\_\_:\_\_\_\_

- 2001:db8:0:c::1c

*Solution:* \_\_\_\_:\_\_\_\_:\_\_\_\_:\_\_\_\_:\_\_\_\_:\_\_\_\_:\_\_\_\_:\_\_\_\_

- 1080::9956:0:0:234

*Solution:* \_\_\_\_:\_\_\_\_:\_\_\_\_:\_\_\_\_:\_\_\_\_:\_\_\_\_:\_\_\_\_:\_\_\_\_

- 2001:638:208:ef34::91ff:0:5424

*Solution:* \_\_\_\_:\_\_\_\_:\_\_\_\_:\_\_\_\_:\_\_\_\_:\_\_\_\_:\_\_\_\_:\_\_\_\_

- 2001:0:85a4::4a1e:370:7112

*Solution:* \_\_\_\_:\_\_\_\_:\_\_\_\_:\_\_\_\_:\_\_\_\_:\_\_\_\_:\_\_\_\_:\_\_\_\_