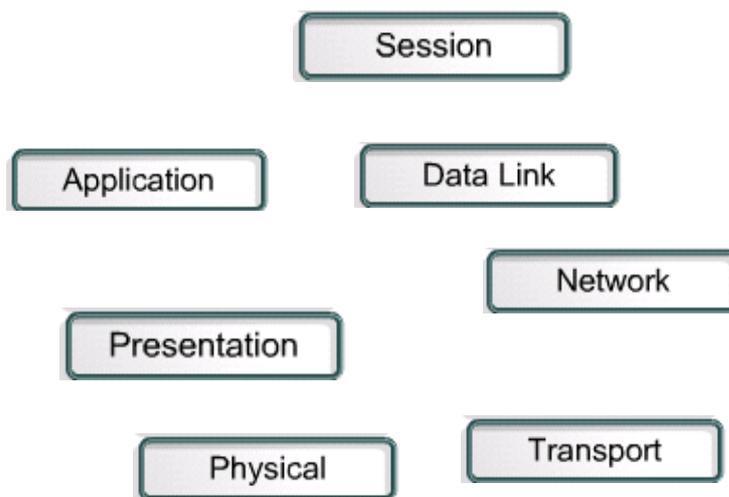


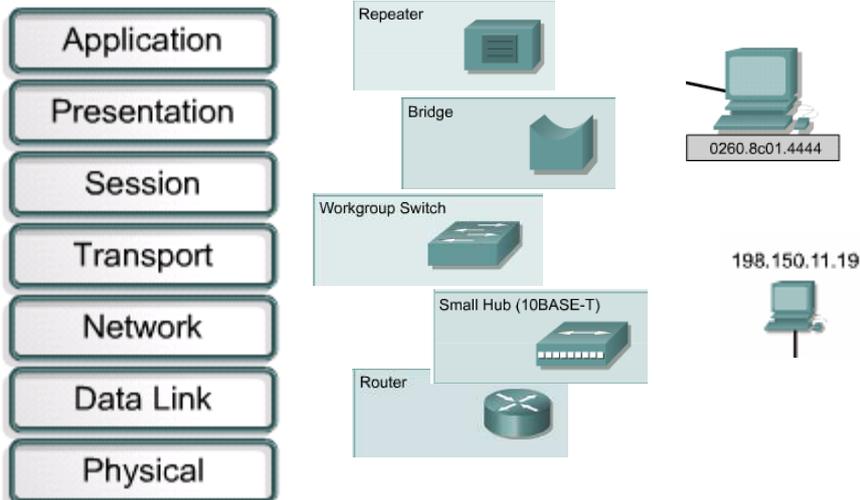
Lecture 2

- Repitition of OSI-model and networking devices
- Binary and hexadecimal numbers
- IP addresses
- Subnetting

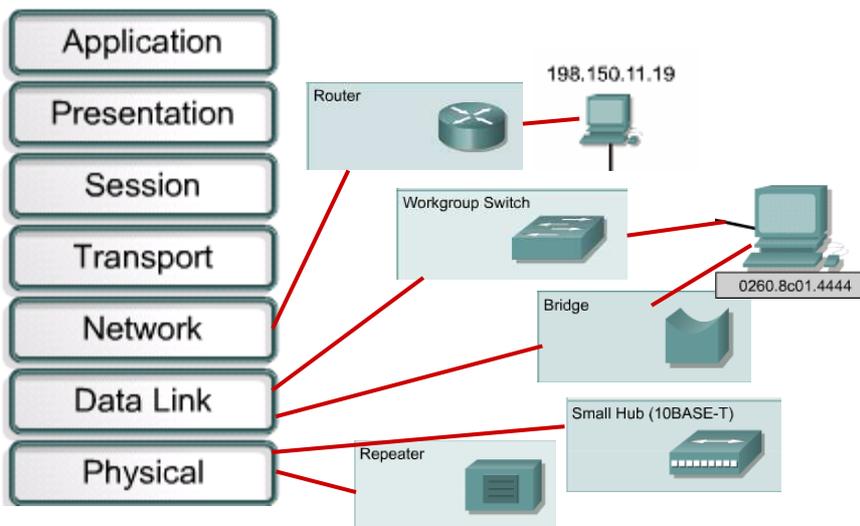
Layers of the OSI-model



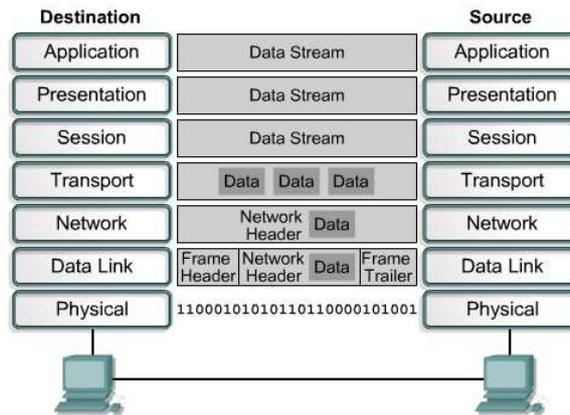
Layers of the OSI-model



Layers of the OSI-model



Data encapsulation



Binary, decimal, hexadecimal

In our number system, the decimal system, we have 10 different symbols to work with, 0-9. The base of the decimal system is therefore 10.

The number 14389, for example, can be divided into different elements:

1 x 10000	or	10^4	(10 x 10 x 10 x 10)
4 x 1000	or	10^3	(10 x 10 x 10)
3 x 100	or	10^2	(10 x 10)
8 x 10	or	10^1	(10)
9 x 1	or	10^0	(1)

A binary number has only 2 symbols and has therefore base 2. Even here each number can be divided into its different parts. For example 110010:

1 x 32	or	2^5	(2 x 2 x 2 x 2 x 2)
1 x 16	or	2^4	(2 x 2 x 2 x 2)
0 x 8	or	2^3	(2 x 2 x 2)
0 x 4	or	2^2	(2 x 2)
1 x 2	or	2^1	(2)
0 x 1	or	2^0	(1)

sum: 50 (110010 in binary becomes 50 in decimal)

Binary, decimal, hexadecimal

The hexadecimal system operates in the same way, but with 16 as its base. As we only know the symbols 0-9 for expressing numbers, but need 16 different symbols for the hexadecimal system, we fill up with letters. 0-9, plus: A for 10, B for 11, C for 12, D for 13, E for 14, F for 15. A hexadecimal number can look like this: 1B 79 CF.

dec	bin	hex
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

In networking, we use binary and hexadecimal numbers mostly for addressing purposes. Binary digits are usually grouped in groups of 8 bits (or 1 byte) and used in IP addresses. With 8 bits of ones or zeros, you have 2^8 (256) different combinations of ones and zeros. Therefore, we seldom handle binary numbers that correspond to larger decimal numbers than 256.

IP address in decimal: 75.60.11.5
IP address in binary octets: 100010110.11110000.10110000.10100000

Hexadecimal digits are used in MAC addresses in groups of 2 hexadecimal symbols.

MAC address in hex: 02-60-8C-01-1C-E7

Conversion binary – decimal - binary

We have already seen a conversion from binary to decimal. Let's take another example.

Question: Convert the binary IP address 11010010.00010011.01100100.10001100 into dotted decimal form.

11010010 can be divided into:

$$1 \times 2^7 = 1 \times 128$$

$$1 \times 2^6 = 1 \times 64$$

$$0 \times 2^5 = 0 \times 32$$

$$1 \times 2^4 = 1 \times 16$$

$$0 \times 2^3 = 0 \times 8$$

$$0 \times 2^2 = 0 \times 4$$

$$1 \times 2^1 = 1 \times 2$$

$$0 \times 2^0 = 0 \times 1$$

This gives:

$$1 \times 128 + 1 \times 64 + 0 \times 32 + 1 \times 16 + 0 \times 8 + 0 \times 4 + 1 \times 2 + 0 \times 1 = 210$$

The other 3 octets of the address are left for you as an exercise

Conversion binary – decimal - binary

Conversion back to binary is done as follows:

210.19.100.140 is the address that should be converted back to binary form, i.e. to base 2. Let's look at the first part, 210.

How many "twos" fit into 210?	Answer: 105.	Rest: 0
How many "twos" fit into 105?	Answer: 52.	Rest: 1
How many "twos" fit into 52?	Answer: 26.	Rest: 0
How many "twos" fit into 26?	Answer: 13.	Rest: 0
How many "twos" fit into 13?	Answer: 6.	Rest: 1
How many "twos" fit into 6?	Answer: 3.	Rest: 0
How many "twos" fit into 3?	Answer: 1.	Rest: 1
How many "twos" fit into 1?	Answer: 0.	Rest: 1

Write down the rests from bottom to top and you have your binary number: 11010010.

The conversion of the remaining 3 octets is left as an exercise.

Conversion binary – decimal – hexadecimal – and back

Conversion from binary to hexadecimal is done by separating the binary number into groups of 4, converting the into decimal (as seen before) and then into hexadecimal.

Example: 00101110

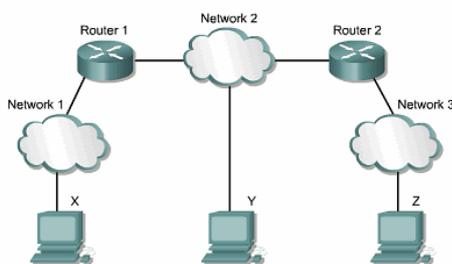
0010 becomes $0 \times 8 + 0 \times 4 + 1 \times 2 + 0 \times 1 = 2$

1110 becomes $1 \times 8 + 1 \times 4 + 1 \times 2 + 0 \times 1 = 14$, which is E in hexadecimal.

Even the conversion back is done through the decimal system as intermediate step.

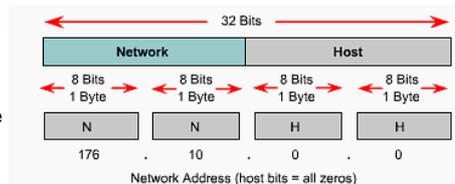
Network addresses

To find the best path through the network, a router has to work with IP-addresses (the destination address added to the packet during the encapsulation process). We prefer IP-addresses in dotted decimal form, while a router handles them in binary. Lets take a look at a routing table of Router 1: By identifying the network that a destination address belongs to, the router can easily find the next hop in its routing table.



Dest. Network	Distance	Next hop
Netw. 1	1	direct
Netw. 2	1	direct
Netw. 3	2	Router 2

A network address identifies all hosts within a certain network (or part of a network). Network addresses have all zeros in the host part. If the network part of network address and host address match (e.g. when a router checks a destination address of a packet against the networks in its routing table), the host is positively identified as part of this network.



Address Classes

Depending on the number of bits that are used for the network part or the host part, the IP-addresses are classified in different classes. The important ones right now are class A, B and C with 8, 16 and 14 bits for the network part, respectively. You can easily distinguish between the classes by looking at the first octet of an address.

IP Address Class	High Order Bits	First Octet Address Range	Number of Bits in the Network Address
Class A	0	0 - 127 *	8
Class B	10	128 - 191	16
Class C	110	192 - 223	24
Class D	1110	224 - 239	28

Class A	Network			Host
Octet	1	2	3	4

Class B	Network		Host	
Octet	1	2	3	4

Class C	Network			Host
Octet	1	2	3	4

Class D	Host			
Octet	1	2	3	4

Address Classes

Few bits in the network part means few bits to address the network, but many bits to address hosts in this network. Large organizations get class A addresses and can address more than 16 million hosts. If you want to e.g. build a LAN for a small company, you will be assigned a class C address and can have a maximum of 254 host.

Address Class	Number of Networks	Number of Host per Network
A	126 *	16,777,216
B	16,384	65,535
C	2,097,152	254
D (Multicast)	N/A	N/A

Class A	Network			Host
Octet	1	2	3	4

Class B	Network		Host	
Octet	1	2	3	4

Class C	Network			Host
Octet	1	2	3	4

Class D	Host			
Octet	1	2	3	4

Network Address

The network address (i.e. the address used to denote the whole network) cannot be used to address a host. In the example figure, 198.150.11.0 is the class C network address and we have all zeros in the host part:

111000110.10010110.00001011.00000000

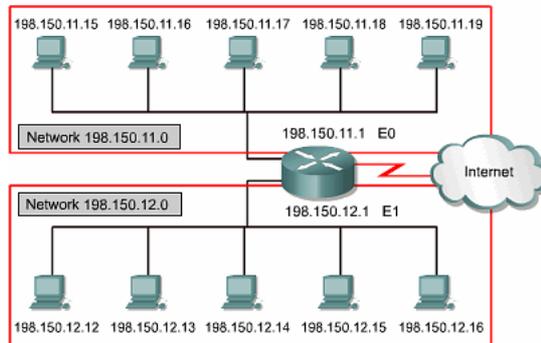
The first address that can be used for host addressing is therefore 198.150.11.1. In binary:

111000110.10010110.00001011.00000001 198.150.11.1

111000110.10010110.00001011.00000010 198.150.11.2

111000110.10010110.00001011.00000011 198.150.11.3

...



Broadcast Address

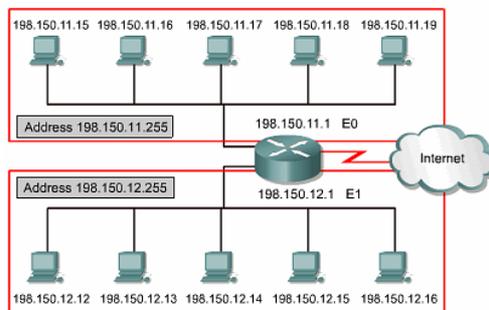
...

111000110.10010110.00001011.11111101 198.150.11.253

111000110.10010110.00001011.11111110 198.150.11.254

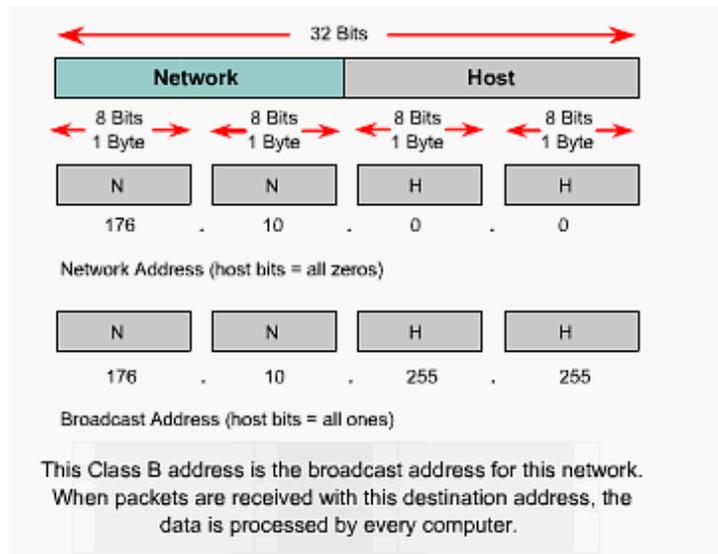
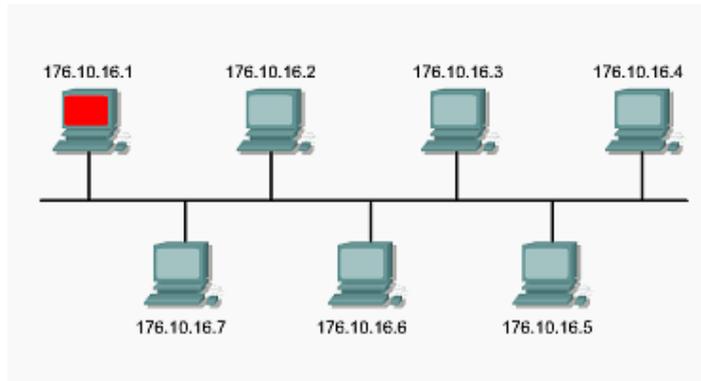
What about 198.150.11.255? The very last address in a network (the address with only ones in the host part) is called broadcast address and cannot be used for host addressing. So, the first (networking address) and the last (broadcast address) are not available when you count the number of addressable hosts in your network.

Broadcast address: 111000110.10010110.00001011.11111111 198.150.11.255



Broadcast

So what is broadcast? If you address a single host, it is called **Unicast**. Even unicast packets are received by each and every host, but after unpacking it up to the network level, the host sees if it was the intended recipient or not. If not, the packet is simply thrown away. Sometimes you want to reach everybody in a certain network without specifying every single address. Here you use the broadcast address. Broadcast is very useful if you really intend to reach everyone (or simply don't know the specific address of a host, only what network it is in), but used excessively and without purpose, it floods the network with meaningless data traffic. This slows the network down and leads to congestion, which should be avoided.



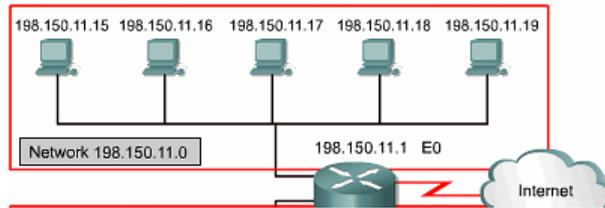
Net mask

If a router checks a destination address, how does it know how many bits are the network part and how many are the host part? An IP address is often given together with a net mask. A netmask has only one purpose: It shows which bits are network part – by marking them with ones, and which bits are host bits – by marking them with zeros.

Example:

address: 198.150.11.17, net mask: 11111111.11111111.11111111.00000000

Alternative way of writing: 198.150.11.14/24 (24 ones in the net mask).



Class A:	11111111.00000000.00000000.00000000	or	/8
Class B:	11111111.11111111.00000000.00000000	or	/16
Class C:	11111111.11111111.11111111.00000000	or	/24

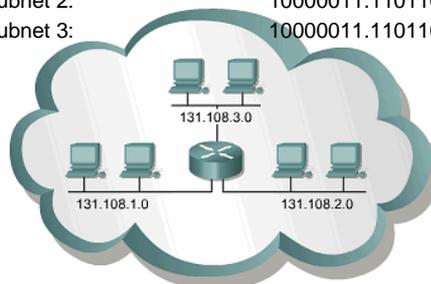
Addressing with Subnets

In this figure, we see a class B address. The network address is 131.108.0.0. This gives you one network with 65 535 different hosts. Broadcasting within this network would reach all these host, simply by giving the broadcast address 131.108.255.255.

As we mentioned before, broadcasting should be limited to reduce the burden on the network. Partitioning the network into a number of subnetworks with their own (sub-)network addresses and (sub-) broadcast addresses, makes the network easier to handle. In the figure, 3 subnets were created by borrowing 8 bits from the host part to address the subnets.

network address (without subnetting):	10000011.1101100.00000000.00000000
network address subnet 1:	10000011.1101100.00000001.00000000
network address subnet 2:	10000011.1101100.00000010.00000000
network address subnet 3:	10000011.1101100.00000011.00000000

and so on.



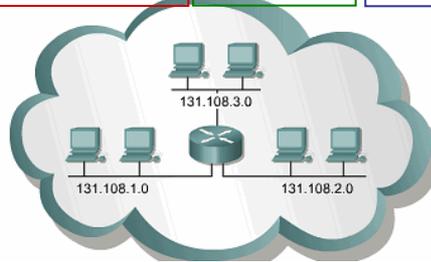
Addressing with Subnets

Without subnetting, 131.108.3.0 would specify a host in the class B network 131.108.0.0. A netmask makes sure there is no confusion: 131.108.3.0/16.

10000011 . 1101100 . 00000011 . 00000000

If the address, on the other hand, is written 131.108.3.0/24, it still is a class B address with 16 bits for the network part, but here an additional 8 bits are used for the subnet and only 8 are left to address the hosts. The net mask (here called subnet mask because it shows the size of the subnet) is all we need to understand what the network looks like.

10000011 . 1101100 . 00000011 . 00000000



Without subnetting, a class B address, gives us 2^{16} possible hosts. With subnetting, this number is reduced by borrowing a certain number of bits for the subnet part of the address. Borrowing 8 bits for subnetting, means that 2^8 (256) subnets can be created. This leaves, on the other hand, only 2^8 (256) hosts per subnet. Borrowing 3 bits for subnetting means 2^3 (8) subnets and 2^{13} (8192) hosts per subnet. (These numbers are not really true, you will soon see why.)

Questions 1:

You have a class C network.

a) How many usable host addresses are there?

Now you want to partition this network into subnets with at 30 hosts per subnet.

b) How many bits do you have to borrow from the host part?

c) How many subnets are there? And how many host can be addresses in this subnetted network in total? Compare with a)

d) What does the subnet mask look like?

Questions 2:

The network of host 201.19.100.140 shall be reorganized into at least 7 subnets.

a) What does the subnet mask for this network look like?

b) Give the network address, broadcast address, as well as the first and the last usable host address in each of the subnets?