7th Slide Set
Computer Networks

Prof. Dr. Christian Baun

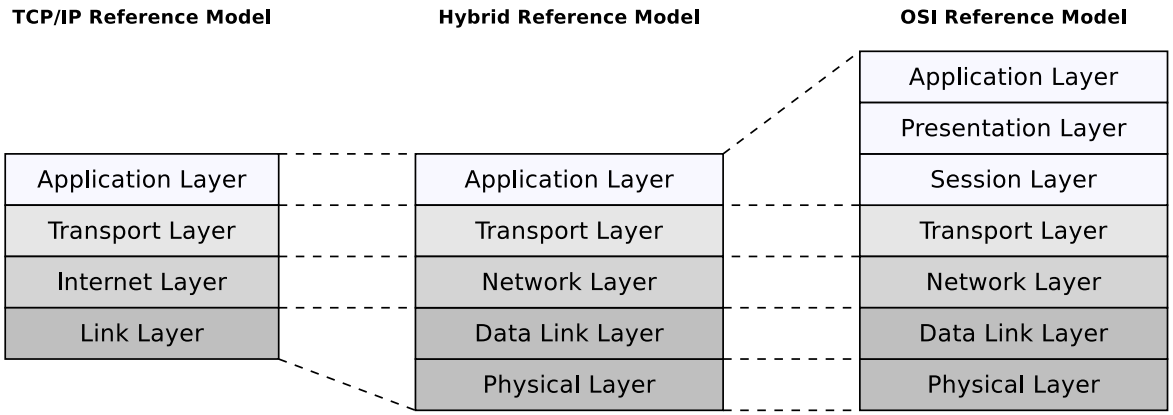
Frankfurt University of Applied Sciences
(1971–2014: Fachhochschule Frankfurt am Main)
Faculty of Computer Science and Engineering
christianbaun@fb2.fra-uas.de

Learning Objectives of this Slide Set

- Network Layer (part 1)
  - Devices of the Network Layer
    - Router
    - Impact on the collision domain
    - Broadcast domain
  - Addressing in the Network Layer
    - Format of IP addresses
    - Address classes, network identifier and host identifier, subnets and subnet mask
    - Private IP addresses
    - Format of IP packets
    - Fragmenting IP packets

Network Layer

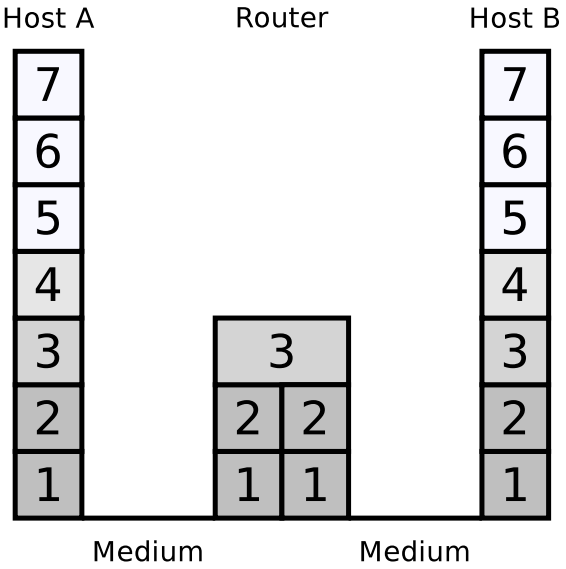
- Functions of the Network Layer
  - Sender: Pack segments of the Transport Layer into pakets
  - Receiver: Identify the packets inside the frames of Data Link Layer
  - Provide logical addresses (IP address)
  - Determinate the best path to the destination = Routing
  - Forward packets between logical networks (across different physical networks)



- Devices: Router, Layer-3-Switch (Router without WAN port)
- Protocols: IPv4, IPv6, ICMP, IPX/SPX, DECnet

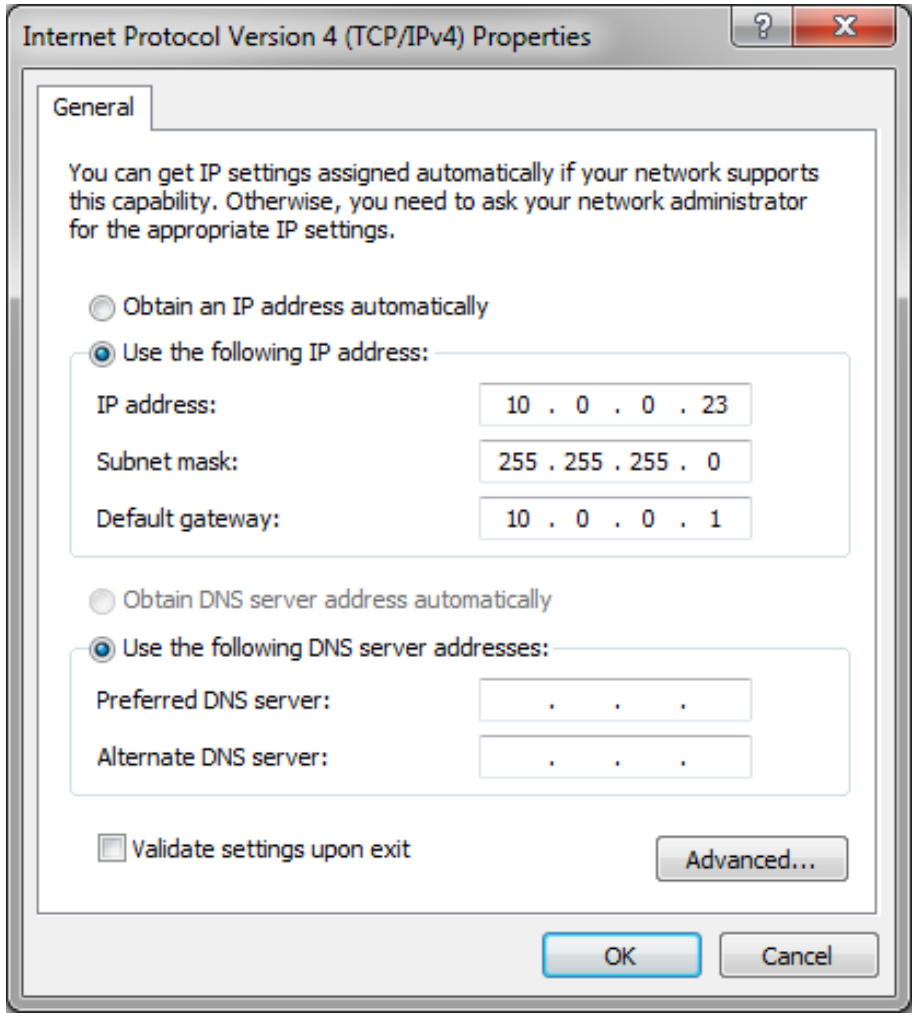
Router, Layer-3-Switch and Gateway

- **Routers** forward packets between networks with different logical address ranges
  - Provide exactly like Hubs and Switches multiple interfaces
  - Enable to connect the local network (LAN) with a WAN (e.g. via DSL or 3G/4G mobile network)
- **Layer-3-Switches** are Routers without a WAN interface
- **Gateways** are protocol converters
  - Enable communication between networks, which base on different protocols
  - A Gateway can in theory operate on all layers
  - Gateways, which operate at the Network Layer, are also called **Multiprotocol Routers**



Gateways (1/2)

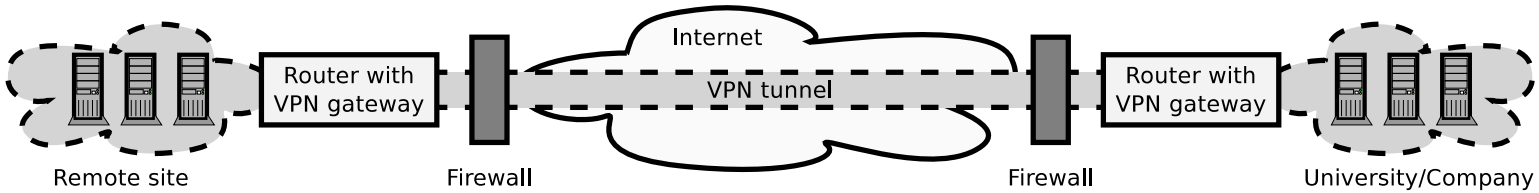
- Modern computer networks operate almost exclusively with the Internet Protocol (IP)
  - For this reason, a protocol conversion at the Network Layer is mostly not necessary
- In the past, in the network preferences of a terminal device, the IP address of the Gateway was specified as **Default Gateway**
  - Today, this field contains the Router address, because a Gateway is usually not required any longer
    - Thus, the term **Default Router** would be suited better



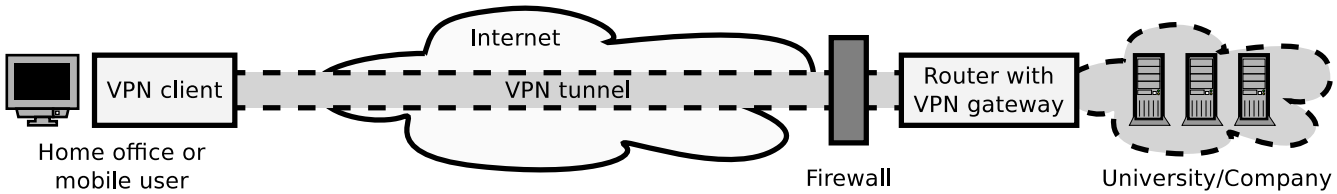
Gateways (2/2)

- VPN-Gateways (Virtual Private Network) may operate on Network Layer
  - They provide secure access to remote protected networks (e.g. intranet of a university or a company) over insecure public networks
  - Services (e.g. Email), which are only available inside the protected network can be used via a tunneled connection

Site-to-Site VPN

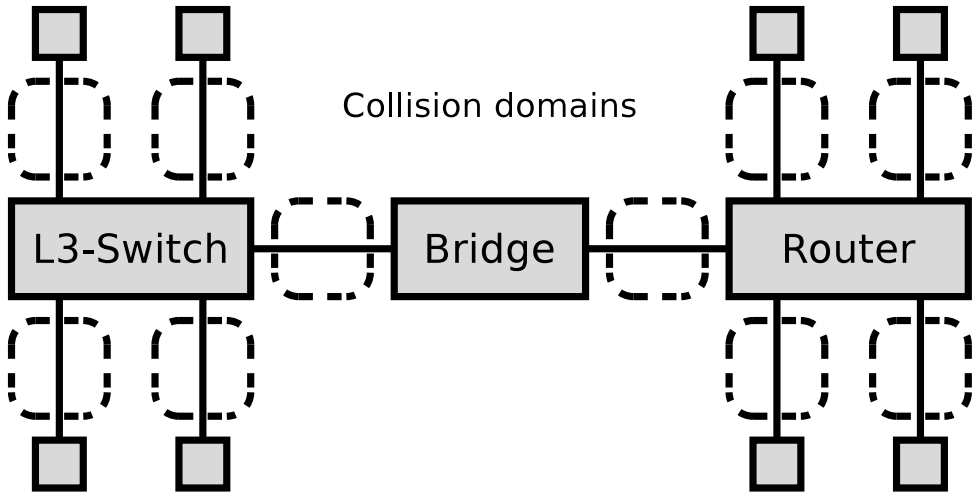


Remote Access VPN = End-to-Site VPN



Collision Domain – Routers and Layer-3-Switches

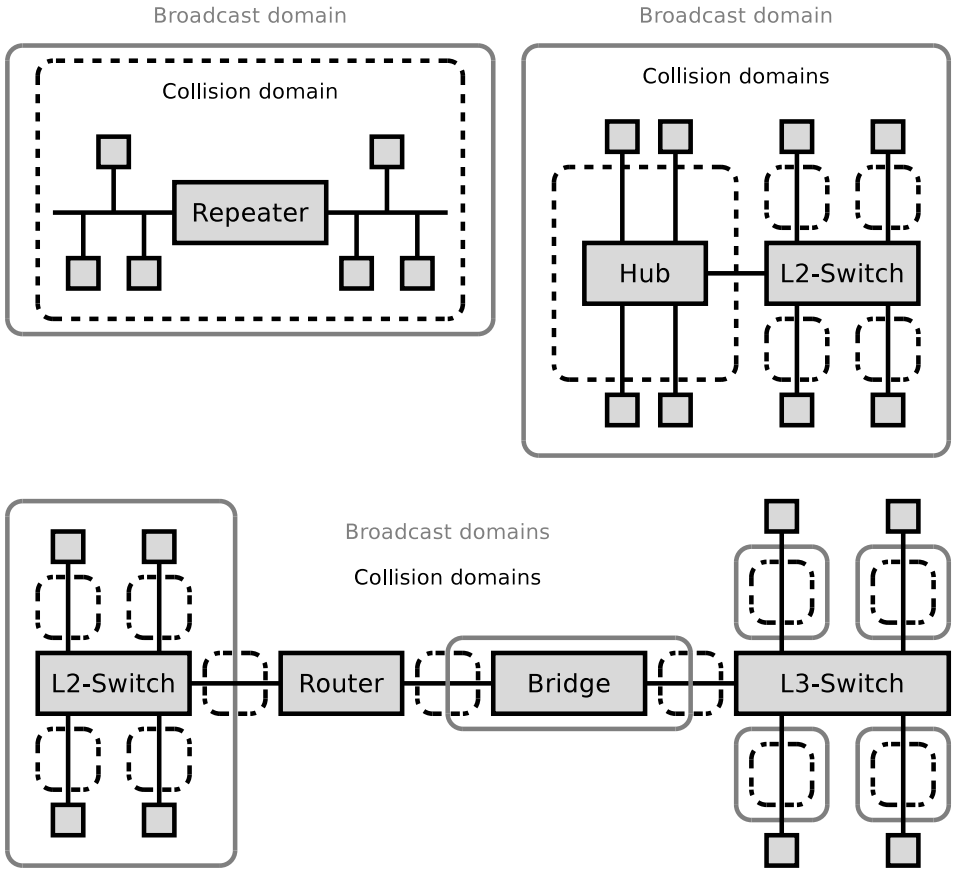
- Routers and Layer-3-Switches divide the collision domain
  - Exactly like Bridges and Layer-2-Switches do



- Devices, which operate on layer 1 (**Repeaters**, **Hubs**) do not divide the collision domain
- Devices, which operate on layer 2 and 3 (**Bridges**, **Layer-2-Switches**, **Routers**, **Layer-3-Switches**) divide the collision domain

Broadcast Domain (1/2)

- Logical part of a computer network, where a broadcast reaches all network devices that belong to that part
  - Devices, which operate on layer 3 (**Routers**) divide the broadcast domain
  - Devices, which operate on layer 1 and 2 (**Repeaters**, **Hubs**, **Bridges**, **Layer-2-Switches**) do not divide it
    - From the perspective of logical networks, they work transparent

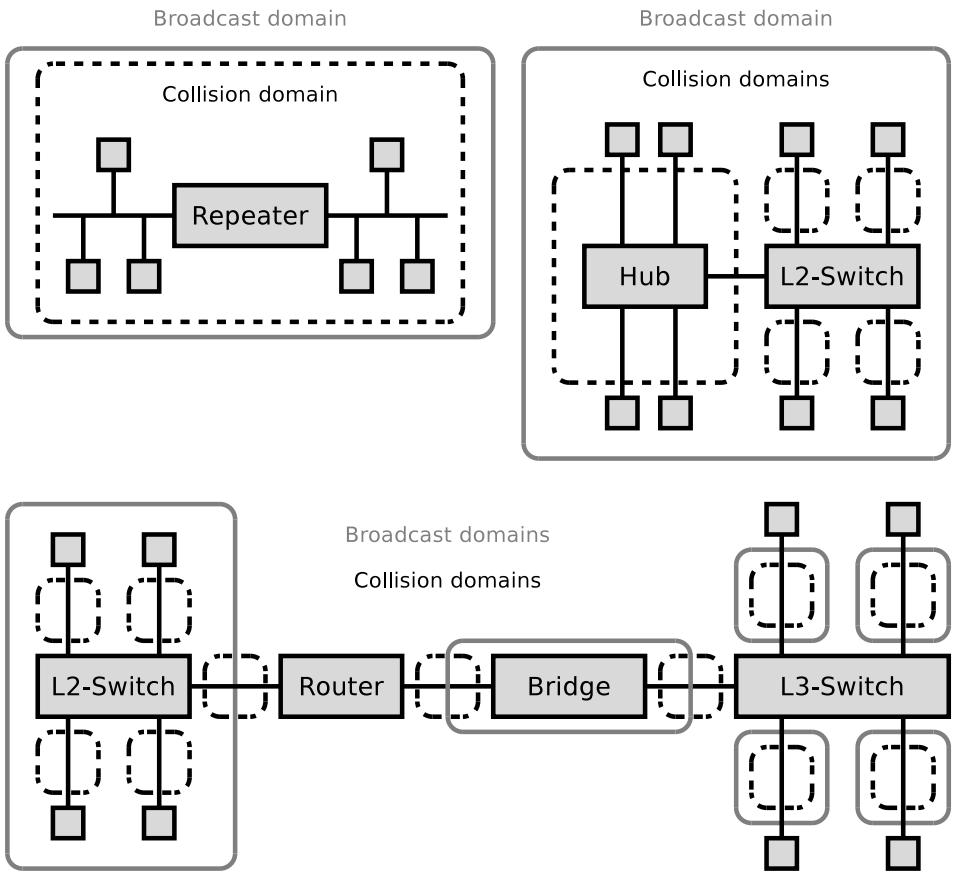


The technical term broadcast domain. . .

always applies to the network layer and never to the data link layer (although broadcasts exist also in the data link layer)

Broadcast Domain (2/2)

- Broadcast domains consist of one or multiple collision domains
- Routers operate at the Network Layer (layer 3)
  - This means, that each port of a Router is connected to a different IP network
    - This information is necessary for the calculation of the required number of subnets
  - Multiple Hubs, Switches, Repeaters or Bridges can operate in the same IP subnet
    - But it is impossible to connect an IP subnet to multiple ports of a Router

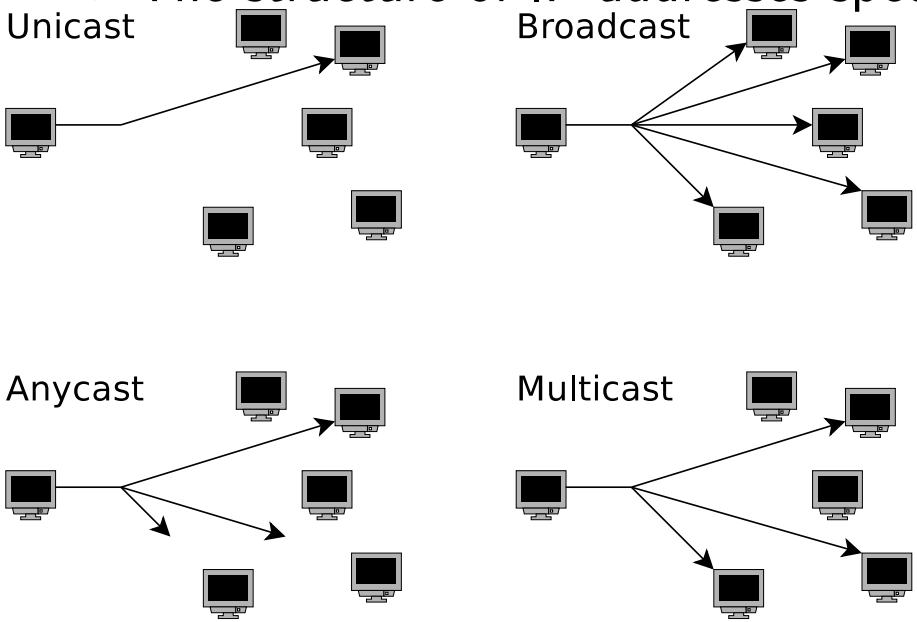


Addressing in the Network Layer (1/2)

- Using only physical addressing via MAC addresses is not useful in large-scale computer networks with possibly global proportions
  - Reason: Maintainability
- **Logical addresses** are required, which are independent from the specific hardware
  - Logical addressing separates the view of humans (logical addresses) from the internal view of computers and software (physical addresses)

Addressing in the Network Layer (2/2)

- Every Network Layer packet contains the IP address of the receiver
  - The structure of IP addresses specifies the Internet Protocol (IP)



- An IP address can be assigned to a single receiver (**unicast**) or a group of receivers (**multicast** or **broadcast**)
- Multiple IP addresses can be assigned to a single network device

- If **Anycast** is used, a single device of a group of devices can be reached via a single address
  - It responds the receiver, which can be accessed via the shortest route

Multicast ist used for example by the routing protocols RIPv2 and OSPF and by Network Time Protocol (NTP) that is used for clock synchronization
Anycast is used for example by some Root Name Servers in the Domain Name System

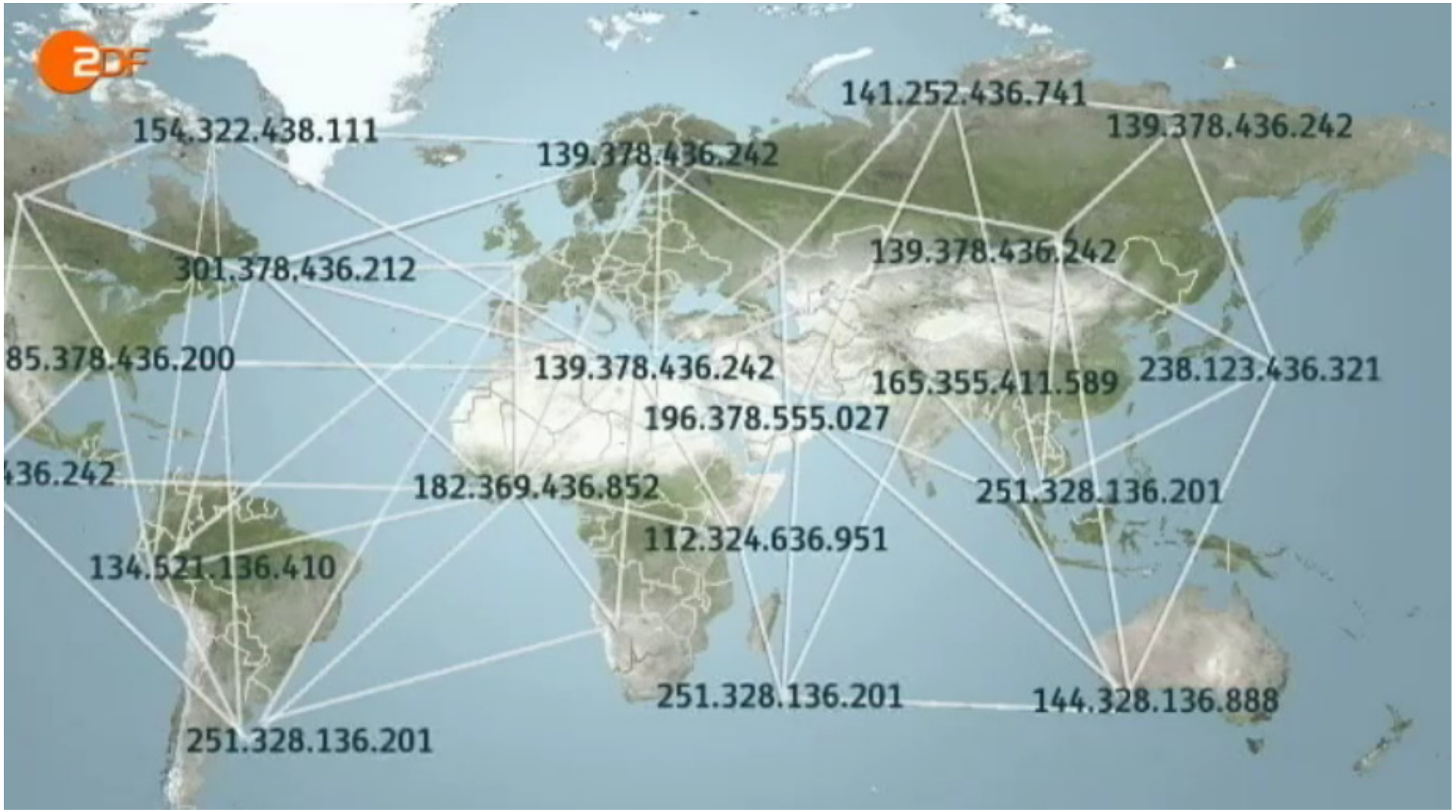
Format of IP Addresses

- IPv4 addresses have a length of 32 bits (4 bytes)
  - Thus, the address space contains $2^{32} = 4,294,967,296$ possible addresses

Address space = amount of all valid network identifiers

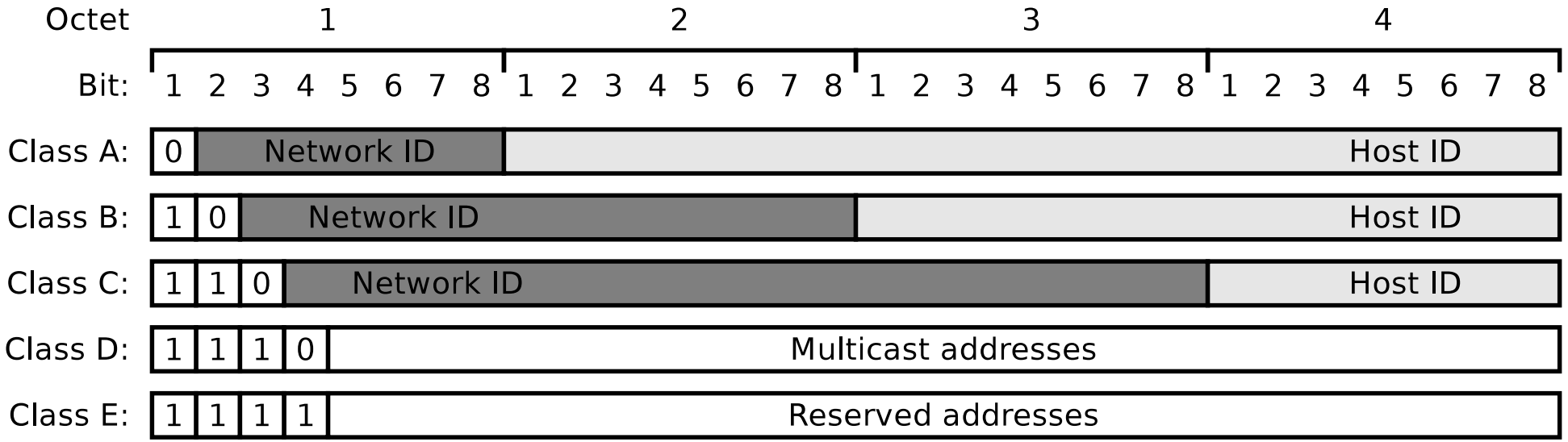
- The usual representation in the so-called **dotted decimal notation**
  - The 4 octets are written as decimal integers in the range from 0 to 255, which are separated from each other by points
    Example: 141.52.166.25

ZDF heute from February 3rd 2011 – What is wrong here?



Address Classes, Network Identifier and Host Identifier

- Originally, IPv4 addresses were categorized into classes from A to C
  - Additionally, the classes D and E for special purposes existed
- A 32 bits long IPv4 address consists of 2 fields:
  - **Network identifier** (network ID)
  - **Host identifier** (host ID)
  - Class A: 7 bits for the network ID and 24 bits for the host ID
  - Class B: 14 bits for the network ID and 16 bits for the host ID
  - Class C: 21 bits for the network ID and 8 bits for the host ID



Address Classes (1/2)

- The prefixes specify the address classes and their address ranges

Class	Prefix	Address range	Network ID	Host ID
A	0	0.0.0.0 - 127.255.255.255	7 bits	24 bits
B	10	128.0.0.0 - 191.255.255.255	14 bits	16 bits
C	110	192.0.0.0 - 223.255.255.255	21 bits	8 bits
D	1110	224.0.0.0 - 239.255.255.255	—	—
E	1111	240.0.0.0 - 255.255.255.255	—	—

- $2^7 = 128$ class A networks with a maximum of $2^{24} = 16,777,216$ host addresses each
- $2^{14} = 16,384$ class B networks with a maximum of $2^{16} = 65,536$ host addresses each
- $2^{21} = 2,097,152$ class C networks with a maximum of $2^8 = 256$ host addresses each
- Class D contains multicast addresses (e.g. for IPTV)
- Class E is reserved for future (?!) purposes and experiments

Why is the class E address space of IPv4 not used?

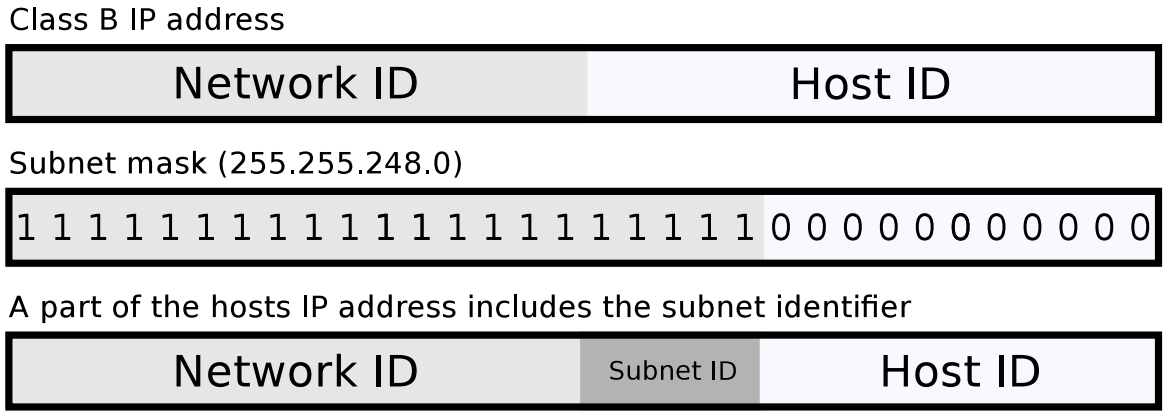
"The class E space has 268 million addresses and would give us in the order of 18 months worth of IPv4 address use. However, many TCP/IP stacks, such as the one in Windows, do not accept addresses from class E space and will not even communicate with correspondents holding those addresses. It is probably too late now to change this behavior on the installed base before the address space would be needed."

Source: `http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_10-3/103_addr-cons.html`

Address Classes (2/2)

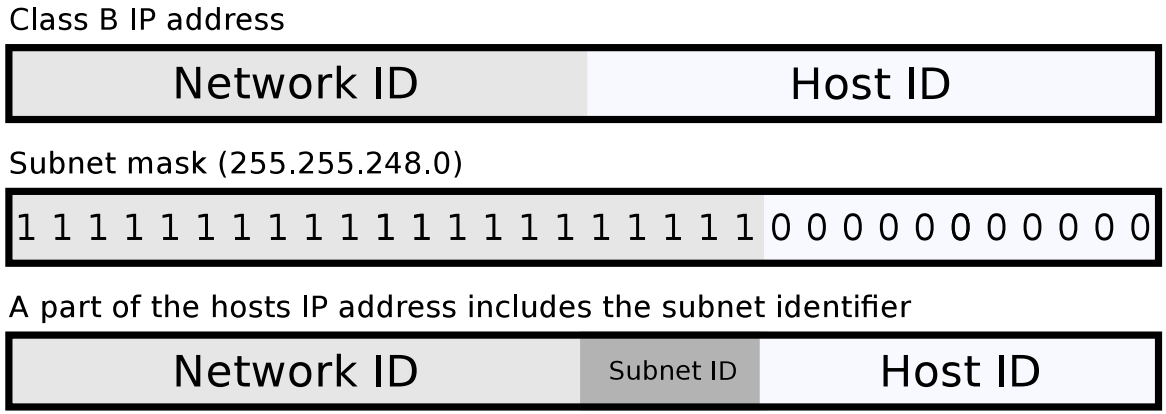
- Only the classes A, B and C are relevant in practice
- The original intention was to identify physical networks in an unique way via the network ID
  - This approach causes some drawbacks
- **Drawbacks of Address Classes:**
  - It is impossible to dynamically adjust them
  - Many addresses are wasted
    - A class C network with 2 devices wastes 253 addresses
    - The address space of class C networks is quite small
    - A class B network with 256 devices wastes $>$ 64,000 addresses
    - Only 128 class A networks exist
    - Migrating multiple devices to a different network class is complex task
- Solution: Logical networks are divided into **subnets**
  - 1993: Introduction of the **Classless Interdomain Routing** (CIDR)

Subnet Mask (1/2)



- For creating subnets, a **(sub-)netmask** is required
  - All hosts in a network have a subnet mask assigned
    - Length: 32 bits (4 bytes)
    - It is used to specify the number of subnets and hosts
- The subnet mask splits the host ID of an IP address into **subnet ID** and **host ID**
  - The network ID remains unchanged
  - The network mask adds another level of hierarchy into the IP address

Subnet Mask (2/2)



- Structure of the subnet mask:
  - 1-bits indicate, which part of the address space is used for subnet IDs
  - 0-bits indicate, which part of the address space is used for host IDs
- Example: Splitting a class B network into 20 subnets requires 5 bits
  - Each subnet requires its own subnet ID and it must be represented in binary form
  - If 5 bits are used for the representation of the subnet IDs, 11 bits remain for host IDs

Syntax of the Classless Interdomain Routing (CIDR)

- Since **CIDR** was introduced in 1993, IP address ranges are assigned in this notation: `First address/mask bits`
  - The number of mask bits indicates the number of 1-bits (prefix) in the subnet mask
- The table shows the possible splits of a class C network into subnets

Mask bits (prefix)	/24	/25	/26	/27	/28	/29	/30	/31	/32
Subnet mask	0	128	192	224	240	248	252	254	255
Subnet bits	0	1	2	3	4	5	6	7	8
Subnets IDs	1	2	4	8	16	32	64	128	256
Host bits	8	7	6	5	4	3	2	1	0
Host IDs	256	128	64	32	16	8	4	2	—
Hosts (maximum)	254	126	62	30	14	6	2	0	—

Not all Addresses can or should be used

Mask bits (prefix)	/24	/25	/26	/27	/28	/29	/30	/31	/32
Subnet mask	0	128	192	224	240	248	252	254	255
Subnet bits	0	1	2	3	4	5	6	7	8
Subnets IDs	1	2	4	8	16	32	64	128	256
Host bits	8	7	6	5	4	3	2	1	0
Host IDs	256	128	64	32	16	8	4	2	—
Hosts (maximum)	254	126	62	30	14	6	2	0	—

2 Host IDs cannot be assigned to network devices, because each (sub-)network requires. . .

- an address for the network itself (all host ID bits are 0 bits)
- a broadcast address to address all devices in network (all bits of the host ID are 1 bits)

2 subnet IDs should not be used

- The subnet IDs, consisting exclusively of 0 bits and 1 bits should not be used
  $\Longrightarrow$ This rule is obsolete, but still often followed
- Modern Routers and network software have no problem, when all possible subnet IDs are assigned to subnets

Determining the necessary Subnets Bits

- By using the table, it is simple to determine the required bits for subnets

Mask bits (prefix)	/24	/25	/26	/27	/28	/29	/30	/31	/32
Subnet mask	0	128	192	224	240	248	252	254	255
Subnet bits	0	1	2	3	4	5	6	7	8
Subnets IDs	1	2	4	8	16	32	64	128	256
Host bits	8	7	6	5	4	3	2	1	0
Host IDs	256	128	64	32	16	8	4	2	—
Hosts (maximum)	254	126	62	30	14	6	2	0	—

- Example: Subdivide a class C network into 5 subnets, each with a maximum of 25 hosts
  - Each subnet requires a subnet address
  - For representing 5 subnets, 3 subnet bits are required
  - The remaining 5 bits are used for representing the host IDs and they allow the addressing of $32 - 2 = 30$ hosts per subnet
  - Thus, the subnet mask with the prefix /27 is well suited for this use case

Calculation example for Subnetting

- Example: 172.21.240.90/27 is a class B address ($\Longrightarrow$ see prefix)
  - The number behind the slash is the number of 1 bits in the subnet mask
- **IP address `AND` subnet mask = subnet address**

```
1 AND 1 = 1,  1 AND 0 = 0,  0 AND 1 = 0,  0 AND 0 = 0
```

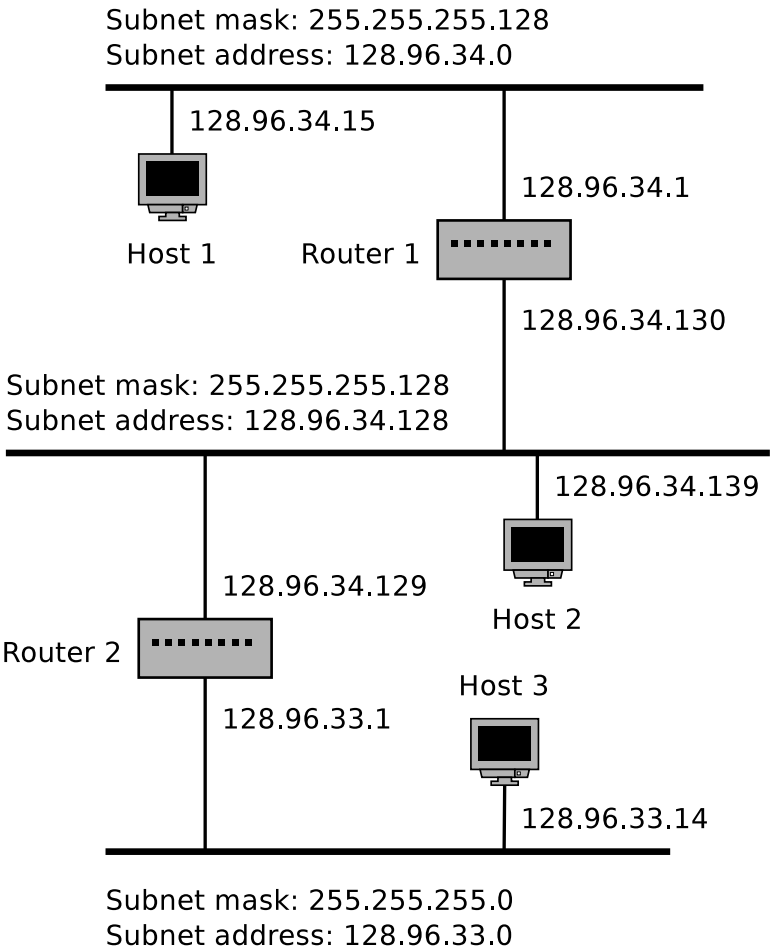
IP address	172.21.240.90	10101100 00010101 11110000 01011010
Subnet mask	255.255.255.224	11111111 11111111 11111111 11100000
Subnet address	172.21.240.64	~~10101100 00010101~~ 11110000 01~~000000~~
Subnet ID	1922	~~10101100 00010101~~ 11110000 01~~000000~~

- **IP address `AND` (`NOT` subnet mask) = host ID**

IP address	172.21.240.90	10101100 00010101 11110000 01011010
Subnet mask	255.255.255.224	11111111 11111111 11111111 11100000
Inverse subnet mask	000.000.000.31	~~00000000 00000000 00000000 000~~11111
Host ID	26	~~00000000 00000000 00000000 000~~11010

- /27 and class B prefix $\Longrightarrow$ 11 bits for the subnet ID
  - 5 bits and therefore $2^5 = 32$ addresses remain for the host IDs
    - 30 of these addresses can be assigned to network devices

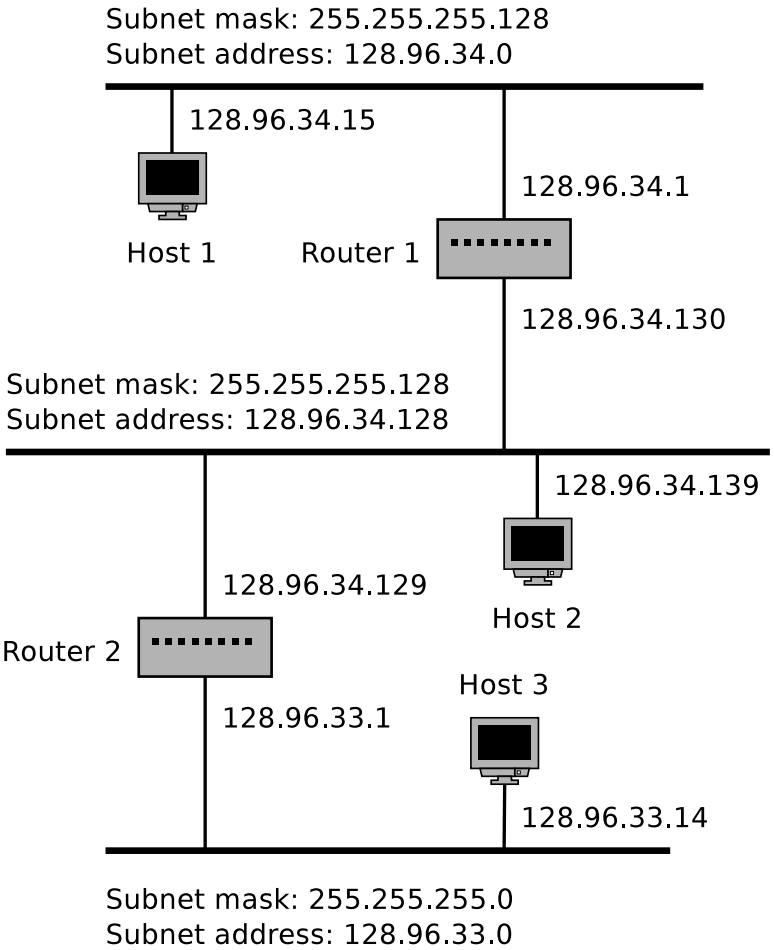
Example (1/4)



Source: Computernetzwerke. Peterson and Davie. dpunkt (2000)

- All hosts inside the same subnet have the same subnet mask
- IP address `AND` subnet mask = subnet address
- If a host wants to transmit a packet, it calculates the `AND` of its own subnet mask and the destination IP address
  - If the result es equal to the subnet address of the sender, the sender learns that the destination is inside the same subnet
  - If the result does not match the subnet address of the sender, the packet must be transmitted to a Router, which forwards it to another subnet

Example (2/4)



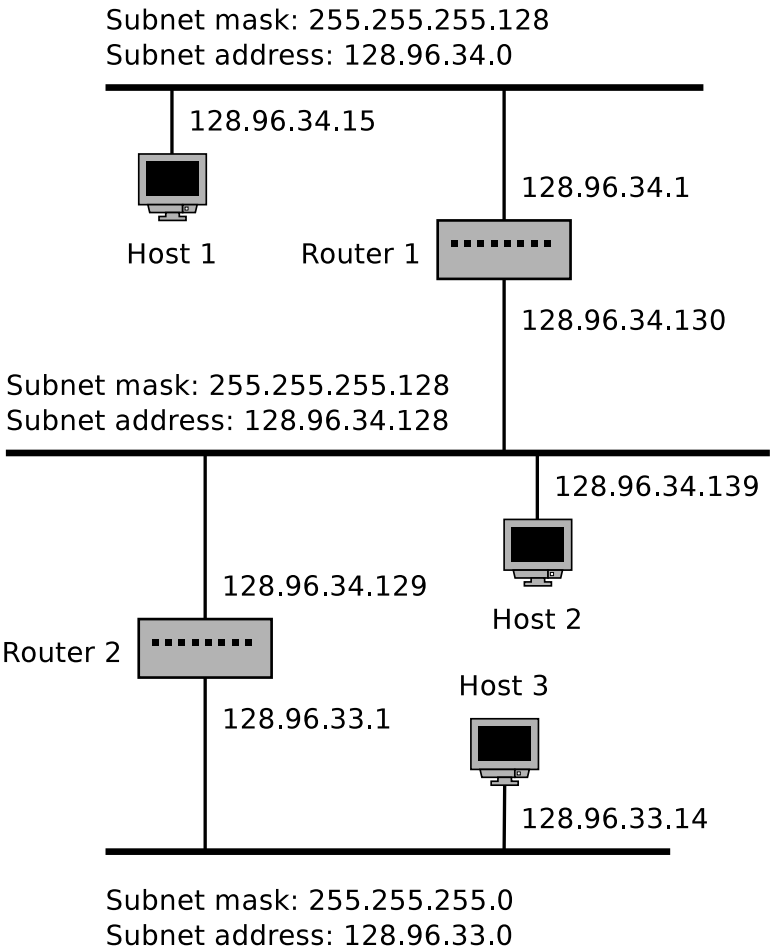
Source: Computernetzwerke. Peterson and Davie. dpunkt (2000)

- Example: Host 1 transmits a packet to host 2 (128.96.34.139)
- Host 1 calculates subnet mask (255.255.255.128) AND destination address (128.96.34.139). Result: 128.96.34.128
- This is not the subnet of host 1 $\Longrightarrow$ Host 2 is in a different subnet
- Host 1 transmits the packet to its default Router (128.96.34.1)
- Entries in the routing table of Router 1

Subnet address	Subnet mask	Next hop
128.96.34.0	255.255.255.128	Port 0
128.96.34.128	255.255.255.128	Port 1
128.96.33.0	255.255.255.0	Router 2

- Routing protocol/algorithms ($\Longrightarrow$ see slide set 8) create and maintain the entries in the routing tables inside the Routers

Example (3/4)



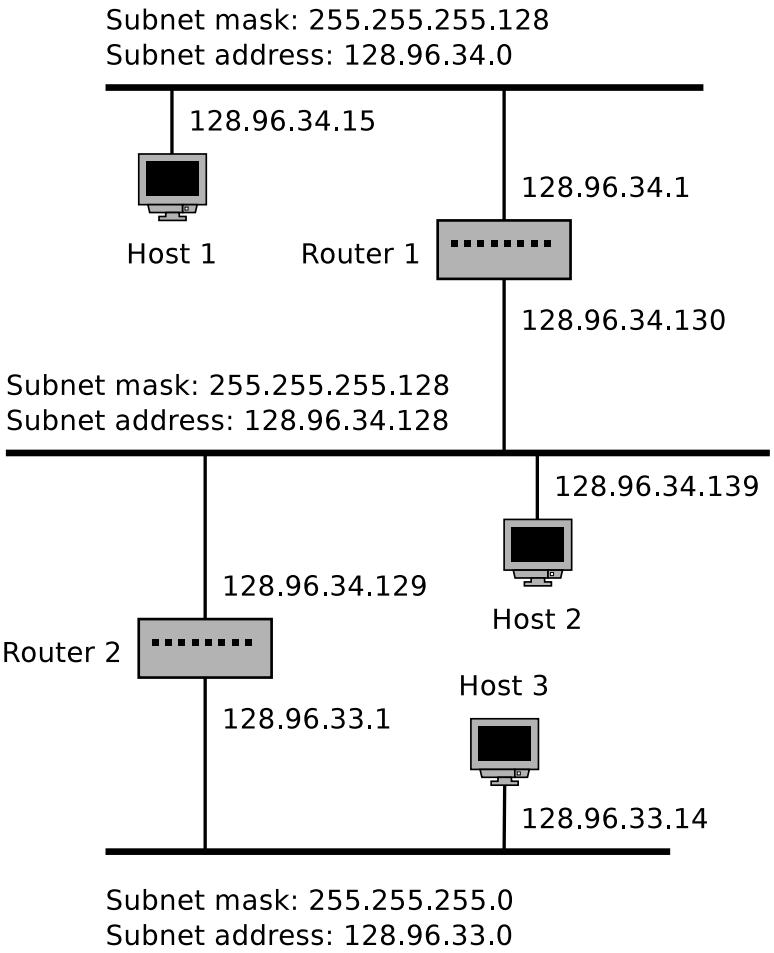
Source: Computernetzwerke. Peterson and Davie. dpunkt (2000)

- Entries in the routing table of Router 1

Subnet address	Subnet mask	Next hop
128.96.34.0	255.255.255.128	Port 0
128.96.34.128	255.255.255.128	Port 1
128.96.33.0	255.255.255.0	Router 2

- The Router calculates the destination address `AND` subnet mask for each entry (row)
- If the result is equal with the subnet address of one entry, the Router forwards the packet the corresponding Router or port
- Router 1 calculates for the 1st row: Host 2 (128.96.34.139) `AND` subnet mask (255.255.255.128) $\Longrightarrow$ 128.92.34.128
- This result does not match the subnet address (128.96.34.0) inside the routing table

Example (4/4)



Source: Computernetzwerke. Peterson and Davie. dpunkt (2000)

- Entries in the routing table of Router 1

Subnet address	Subnet mask	Next hop
128.96.34.0	255.255.255.128	Port 0
128.96.34.128	255.255.255.128	Port 1
128.96.33.0	255.255.255.0	Router 2

- Router 1 calculates for the 2nd row: Host 2 (128.96.34.139) AND subnet mask (255.255.255.128) $\Longrightarrow$ 128.96.34.128
- This result is equal with the subnet address entry in the forwarding table
  $\Longrightarrow$ The 2nd row is a hit
- Router 1 transmits the packet via port 1 to host 2, because this port is connected to the same network as host 2

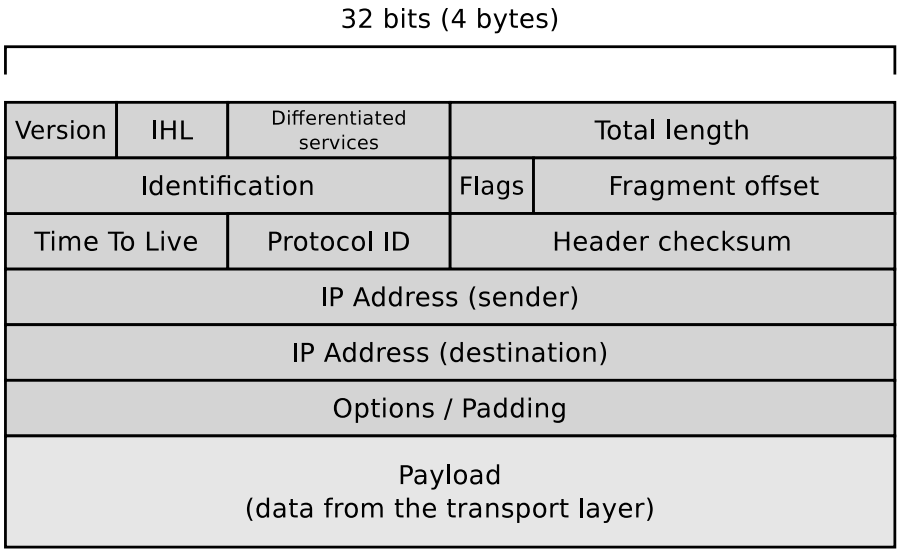
Private Networks – Private IP Address Spaces

- In private networks, it is also required to assign IPs to network devices
  - These addresses are not allowed to interfere with global accessible internet services
- Several address spaces exist, containing private IP addresses
  - These address spaces are **not routed** in the internet

Address space:	10.0.0.0 to 10.255.255.255
CIDR notation:	10.0.0.0/8
Number of addresses:	$2^{24} = 16{,}777{,}216$
Address class:	Class A. 1 private network with 16,777,216 addresses
Address space:	172.16.0.0 to 172.31.255.255
CIDR notation:	172.16.0.0/12
Number of addresses:	$2^{20} = 1{,}048{,}576$
Address class:	Class B. 16 private networks with 65,536 addresses each
Address space:	192.168.0.0 to 192.168.255.255
CIDR notation:	192.168.0.0/16
Number of addresses:	$2^{16} = 65{,}536$
Address class:	Class C. 256 private networks with 256 addresses each

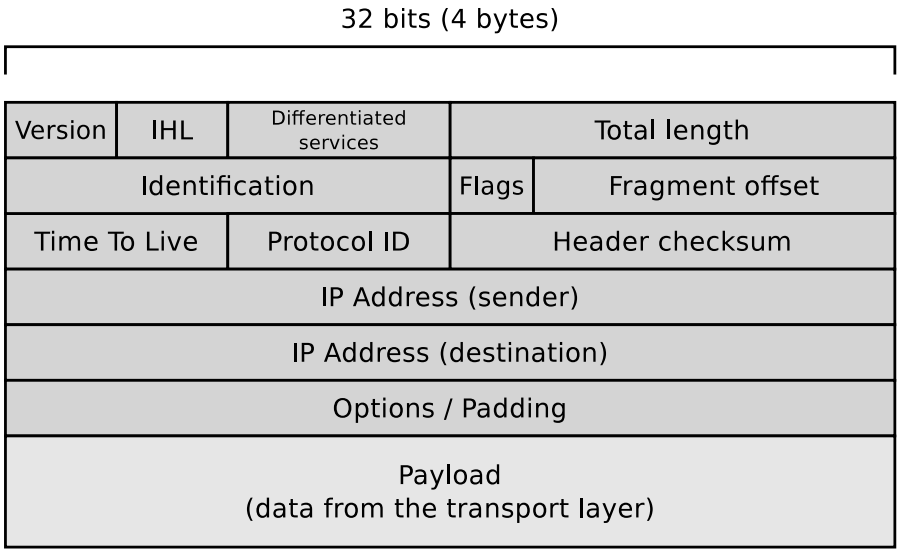
Structure of IPv4 Packets (1/7)

- **Version** (4 bits)
  - Protocol version
    - Version $= 4 \Longrightarrow$ IPv4
    - Version $= 6 \Longrightarrow$ IPv6



- **IHL** = IP Header Length (4 bits)
  - Header length, represented as the number of 4 byte words
    - Example: IHL $= 5 \Longrightarrow 5 * 4$ bytes $= 20$ bytes
  - Indicates where the payload begins
- **Differentiated services** (8 bits)
  - Prioritization of IP packets is possible with this field (Quality of Service)
  - The field slightly changed over the years (RFC 791, RFC 2474, RFC 3168)

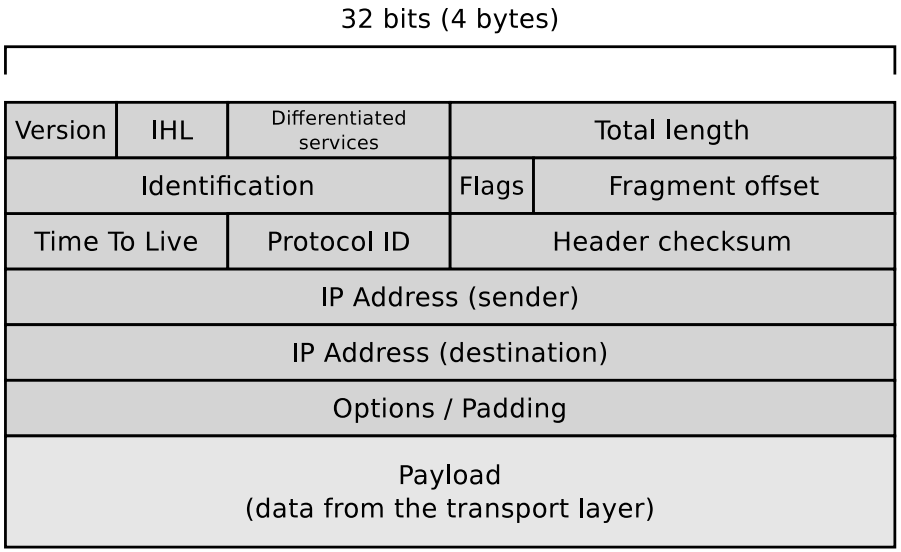
Structure of IPv4 Packets (2/7)



- **Total length** (16 bits)
  - This field defines the entire packet size (header and payload)
  - This length of the field is 16 bits and therefore the maximum possible IPv4 packet length is 65,535 bytes

Structure of IPv4 Packets (3/7)

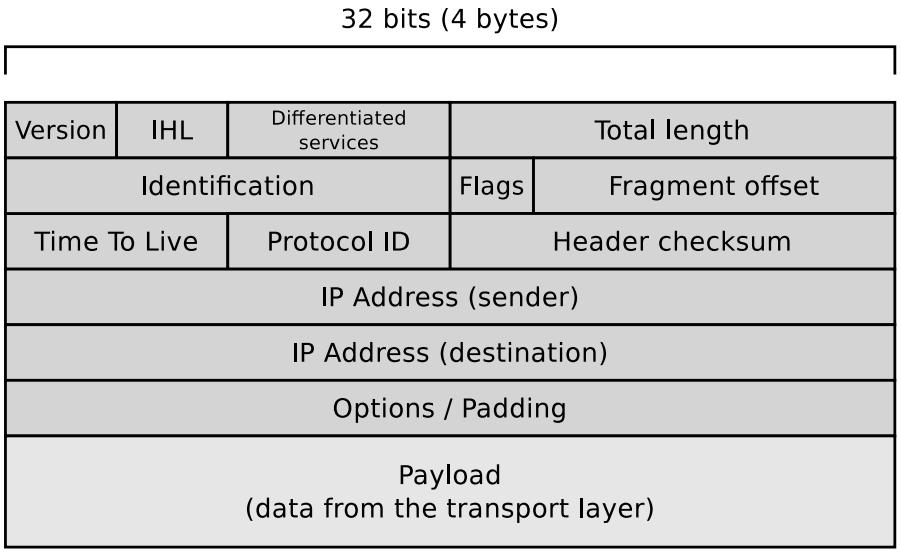
- The fields **Identification**, **Flags** and **Fragment offset** control the assembly of fragmented IP packets
- **Identification** (16 bits)
  - Contains a unique identifier of the IP packet



- **Flags** (3 bits)
  - Here the sender informs whether the packet can be fragmented and the receiver is informed whether more fragments follow
- **Fragment Offset** (13 bits)
  - Contains a number which states for fragmented packets, from which position of the unfragmented packet the fragment begins

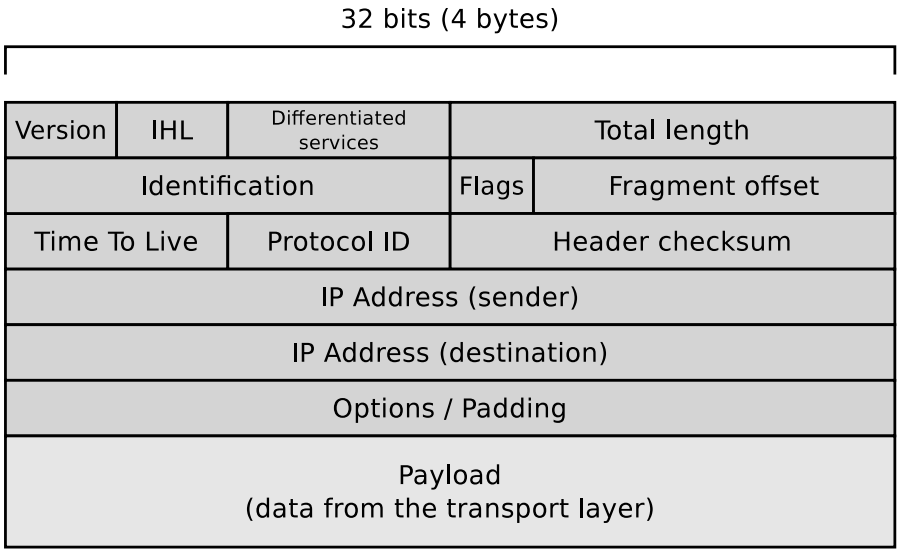
More information about the fragmentation of IP packages provide the slides 35 + 36

Structure of IPv4 Packets (4/7)



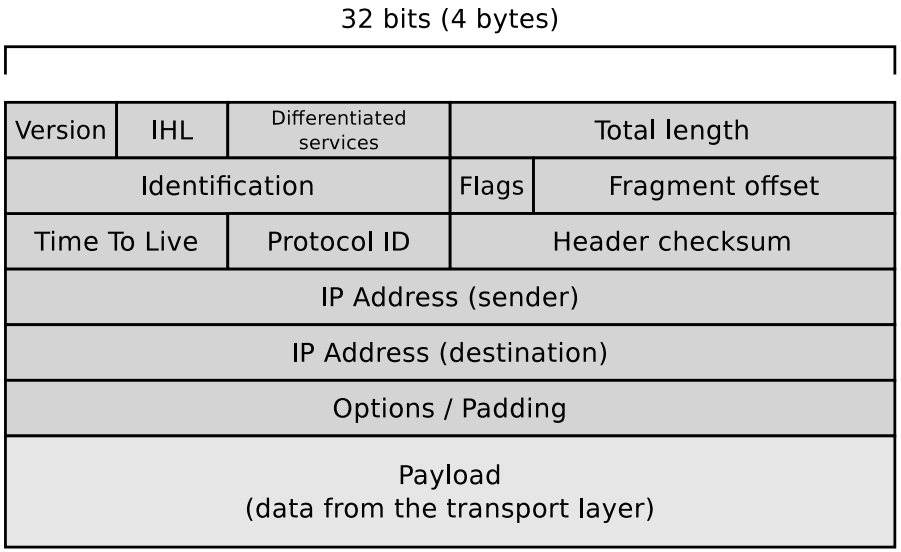
- **Time To Live** (8 bits)
  - Specifies the maximum lifetime of an IP packet during transmission in seconds
  - If the value is zero, the packet is discarded by the Router
  - In practice, the field is used as a hop count and each Router on the route to the destination decrements the TTL field by one
  - Prevents that undeliverable IP packets endlessly go in cycles on the network

Structure of IPv4 Packets (5/7)



- **Protokoll ID** (8 bits)
  - Contains the number of the Transport Layer protocol used
  - For TCP segments, the value is 6
  - For UDP segments, the value is 17
  - If the payload contains an ICMP message, this field contains the value 1
  - If the payload contains an OSPF message, this field contains the value 89

Structure of IPv4 Packets (6/7)

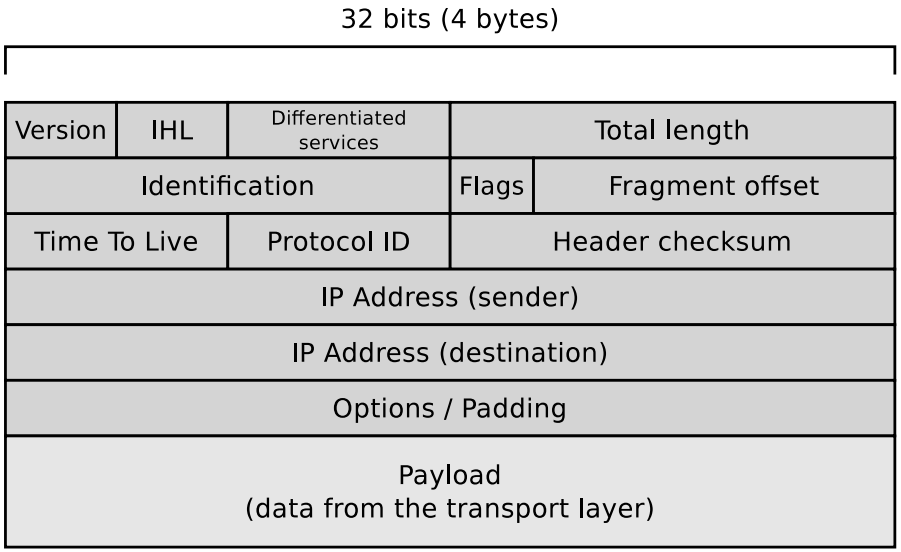


- Each IPv4 packet contains a checksum (16 bits) of the header
  - Because at each Router on the way to the destination, the content of the field **Time To Live** changes, each Router need to verify the checksum, recalculate and insert it into the header

Routers usually ignore the checksum to speedup the packet forwarding

Therefore, IPv6 packets contain no checksum field

Structure of IPv4 Packets (7/7)



- The field **IP address (sender)** (32 bits) contains the source address and **IP address (destination)** contains the destination address
- The field **Options / Padding** can contain additional information such as a time stamp
  - This last field before the payload area is filled with padding bits (0 bits) if necessary, to ensure that the header size is an integer number of 32 bit words
- The last field contains the data from the Transport Layer

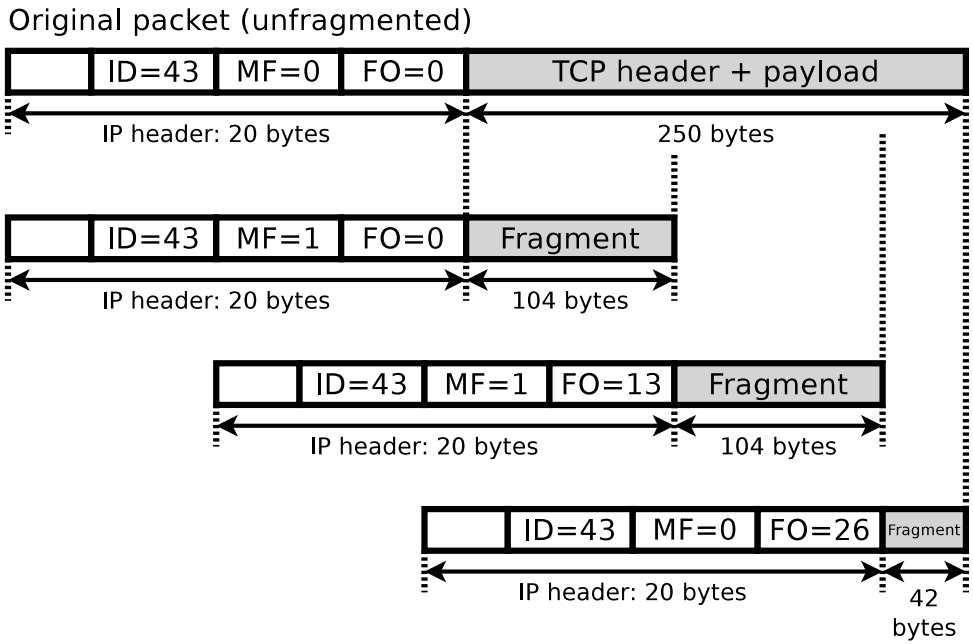
Packet Fragmentation (1/2)

- The split up (and reassembling) of IP packets into smaller packets (**fragments**) is called **Packet fragmentation**
  - Is usually done by Routers
    - Packet fragmentation can also be carried out by the sender
- Reason for packet fragmentation:
  - The maximum packet length depends on the network technology used
- The **Maximum Transmission Unit** (MTU) specifies the maximum payload of a frame (and thus the maximum size of an IP packet too)
  - MTU of Ethernet: usually 1,500 bytes
    - For Gigabit Ethernet, *Jumboframes* exist with a size of up to 9,000 bytes
  - MTU of WLAN (IEEE 802.11): 2,312 bytes
  - MTU of Token Ring with 4 Mbit/s (IEEE 802.5): 4,464 bytes
  - MTU of Token Ring with 16 Mbit/s: 17,914 bytes
  - MTU of PPPoE (e.g. DSL): $\leq$ 1,492 bytes
  - MTU of ISDN: 576 bytes
  - MTU of FDDI: 4,352 bytes

Packet Fragmentation (2/2)

- IP packets contain a flag which can be used to prohibit fragmentation
  - If a Router needs to fragment a packet because it is too large to forward, but the fragmentation is prohibited in the packet, the Router discards the packet because he cannot forward it
- If a network device does not receive all fragments of an IP packet within a certain period of time (a few seconds), the network device discards all received fragments
- Routers can split IP packets into smaller fragments, if the MTU makes this necessary and it is not prohibited in the packets
  - **But no Router can assemble fragments of a packet to create a larger fragment**
    - Only the receiver can assemble fragments

Packet Fragmentation Example (1/2)



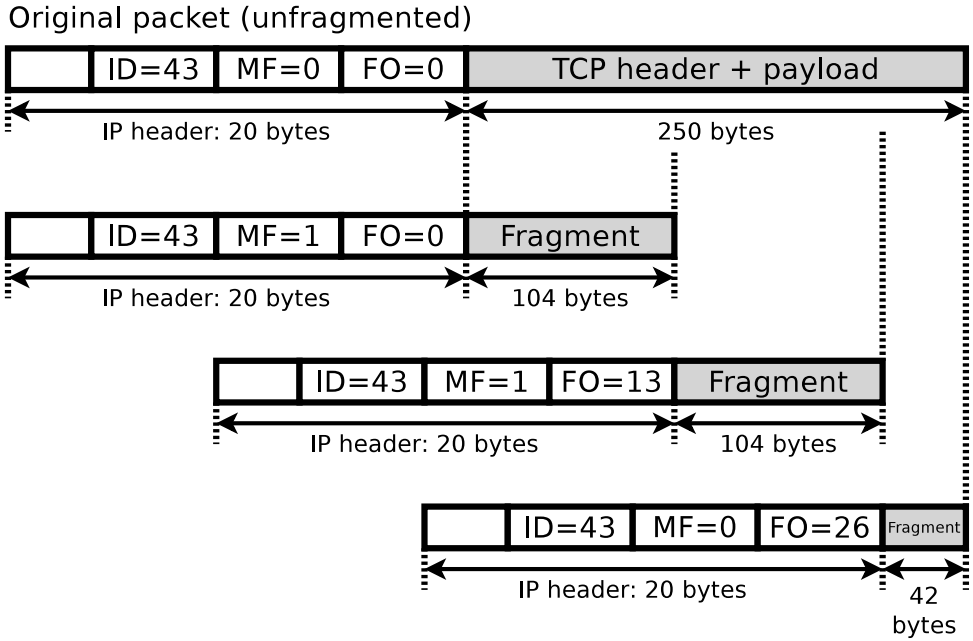
Source

`http://www.netzmafia.de/skripten/netze/netz8.html`

- A TCP segment of 250 bytes length is transmitted via IP
- Maximum packet length: 124 bytes
- IP header length: 20 bytes
- Packet ID: 43
- The fragment offset is counted in 8-byte word increments
- The fragment must therefore be a multiple of 8

- Because all fragments belong to the same packet, the ID is equal for all fragments

Packet Fragmentation Example (2/2)



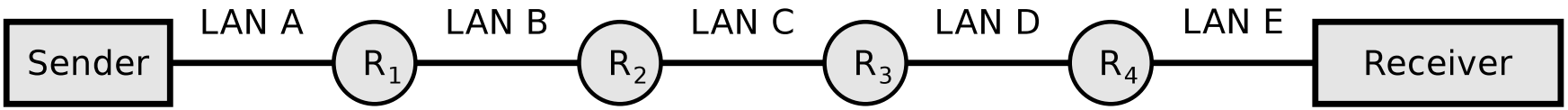
Source

`http://www.netzmafia.de/skripten/netze/netz8.html`

- In the 1st fragment, fragment offset has value 0
- The MF flag has value 1, which indicates that more fragments will follow
- In the 2nd fragment, the fragment offset has value 13 ($104/8 = 13$), which indicates the position of the fragment in the unfragmented packet
- The MF flag has still value 1, because another fragment will follow
- In the header of last fragment, the MF flag has value 0, because it is the final fragment of packet 43
- The fragment offset has value 26, because 208 bytes ($8 * 26 = 208$) have already been sent

Another Fragmentation Example (1/2)

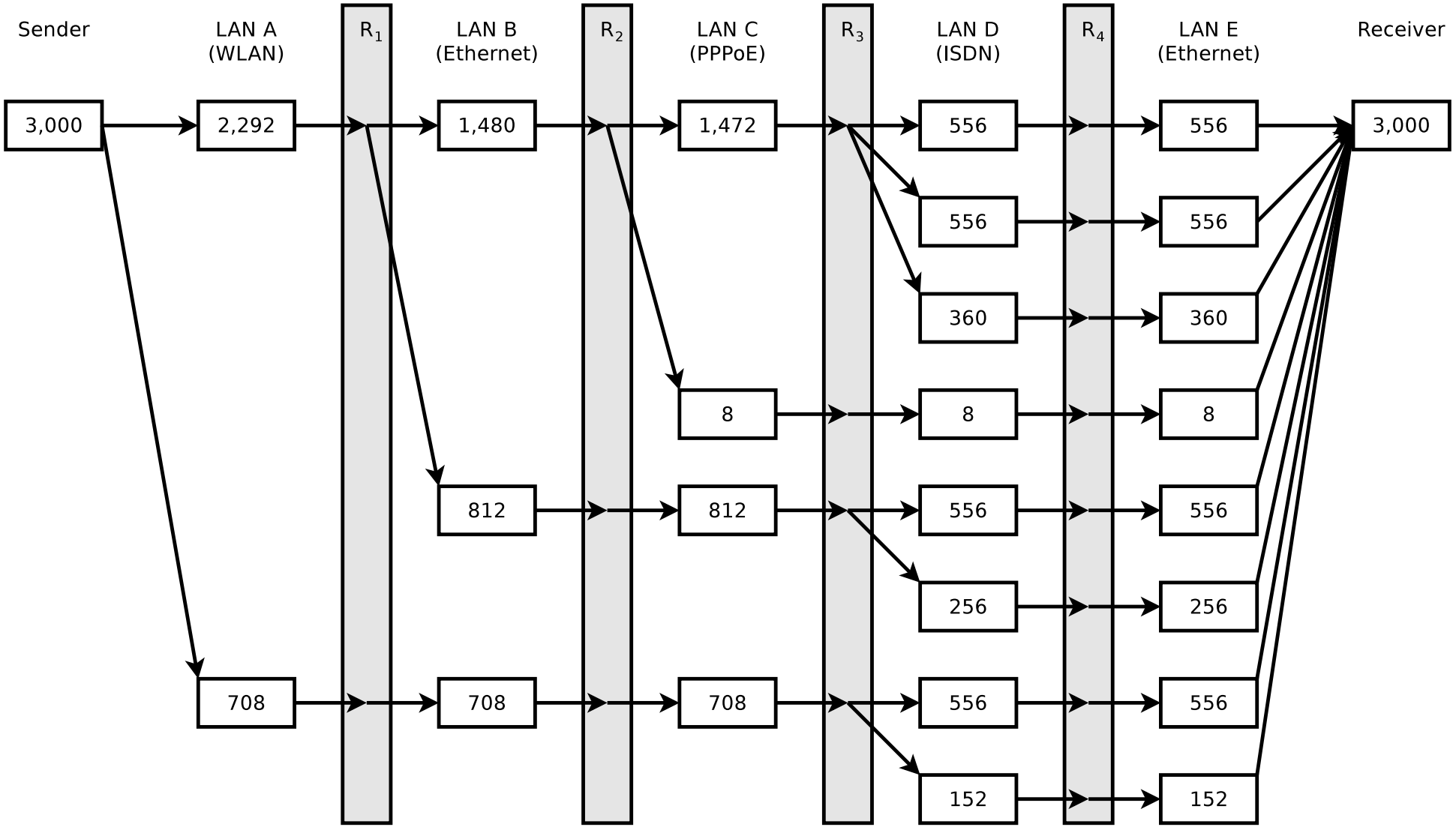
- 3,000 bytes payload need to be transmitted via the IP protocol
- The resulting packets must be fragmented because they are transmitted over multiple physical networks, whose MTU is $< 3,000$ bytes



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

- Show in a graphical way how the packet is fragmented, and how many bytes of payload, each fragment contains

Another Fragmentation Example (2/2)



Status of IPv4

ZEIT ONLINE | INTERNET

INTERNET PROTOKOLL

Bye, bye IPv4

Die letzten Adressblöcke des alten Internet Protokolls Version vier sind vergeben. Die Umstellung auf IPv6, die seit Jahren nicht vorankommt, wird nun beginnen müssen.

VON: Monika Ermert | 2.2.2011 - 16:36 Uhr

Im Netz hat eine neue Zeitrechnung begonnen: In der Nacht zum Dienstag hat die Internet Assigned Numbers Authority (IANA) die letzten freien IPv4-Adressen verteilt. Wer künftig IP-Adressen an Nutzer vergeben möchte, sei es für Mobiltelefone, PCs oder internetfähige Autos, muss sich mit der nächsten Generation von "Rufnummern" befassen, mit der Internet-Protokoll Version 6 – IPv6.

Das Internet-Protokoll ist Teil der komplexen Struktur, die notwendig ist, damit Computer miteinander Daten austauschen können. Es sorgt darin für die korrekte Vermittlung der transportierten Informationen. IPv4 nutzt Adressen mit einer Länge von 32 Bit, was die Zahl der insgesamt verfügbaren IPs auf 4.294.967.296 oder 4,2 Milliarden Stück beschränkte.

Das klingt viel. Aber bei 6,5 Milliarden Menschen weltweit und angesichts des Trends, mehr und mehr Geräte internetfähig zu machen, ist seit Jahren klar, dass die IPv4-Adressen knapp werden. Netzanbieter nutzten daher dynamische Adressen, vergaben also keine festen für jedes einzelne Gerät. Doch auch diese Technik ist begrenzt, weswegen seit vielen Jahren an einem neuen Internet-Protokoll gearbeitet wurde.

IPv6 basiert auf längeren Nummern und bietet damit für die Zukunft die nicht mehr so richtig vorstellbare Zahl von 340 Sextillionen eindeutiger Internetadressen. Jedes Sandkorn könnte damit künftig eine IP-Adresse bekommen.

Bis heute allerdings kam die technische Umstellung nur langsam voran. Nun sind jedoch die letzten freien IPv4-Blöcke an den für Asien zuständigen regionalen IP-Adressverwalter vergeben worden. Bis diese an die einzelnen Netzbetreiber und deren Kunden verteilt sind, wird es noch eine Weile dauern. Außerdem bekommt jede der weltweit fünf Verwaltungen in den kommenden Tagen noch eine Reserve von 16 Millionen IPv4-Adressen, doch der Zeitraum ist absehbar.

Structure of IPv6 Addresses and Networks (1/5)

- IPv6 addresses have a length of 128 bits (16 bytes)
  - Therefore, $2^{128} \approx 3,4 * 10^{38}$ addresses can be represented
- The introduction is useful because of the limited address space of IPv4
- Problem: The decimal notation is confusing
  - For this reason, IPv6 addresses are represented in hexadecimal format
    - Groups of 4 bits are represented as a hexadecimal number
  - Goups of 4 hexadecimal numbers are merged into blocks
  - The blocks are separated by colons
    Example: `2001:0db8:85a3:08d3:1319:8a2e:0370:7344`

- The last 4 bytes (32 bits) of an IPv6 address may also be written in decimal notation
- This is useful to embed the IPv4 address space into the IPv6 address space
  $\Longrightarrow$ see slide 47

RFC 4291 (2006) *IP Version 6 Addressing Architecture*

Structure of IPv6 Addresses and Networks (2/5)

- Rules for simplification (RFC 5952):
  - Leading zeros within a block may be omitted
  - Successive blocks with value 0 (= 0000), may be omitted **exactly 1 time within an IPv6 address**
    - If blocks are omitted, this is indicated by 2 consecutive colons
  - If several groups of null blocks exist, it is recommended to shorten the group with the most null blocks
- Example:
  - The IPv6 address of `j.root-servers.net` is:
    2001:0503:0c27:0000:0000:0000:0002:0030
    $\Longrightarrow$ 2001:503:c27::2:30

Notation of IPv6 addresses (URLs)

- IPv6 addresses are enclosed in square brackets
- Port numbers are appended outside the brackets
  `http://[2001:500:1::803f:235]:8080/`
- This prevents the port number from being interpreted as part of the IPv6 address

Structure of IPv6 Addresses and Networks (3/5)

- IPv6 addresses consist of 2 parts

64 Bits	64 Bits
Network Prefix	Interface Identifier
2001:638:208:ef34	:0:ff:fe00:65

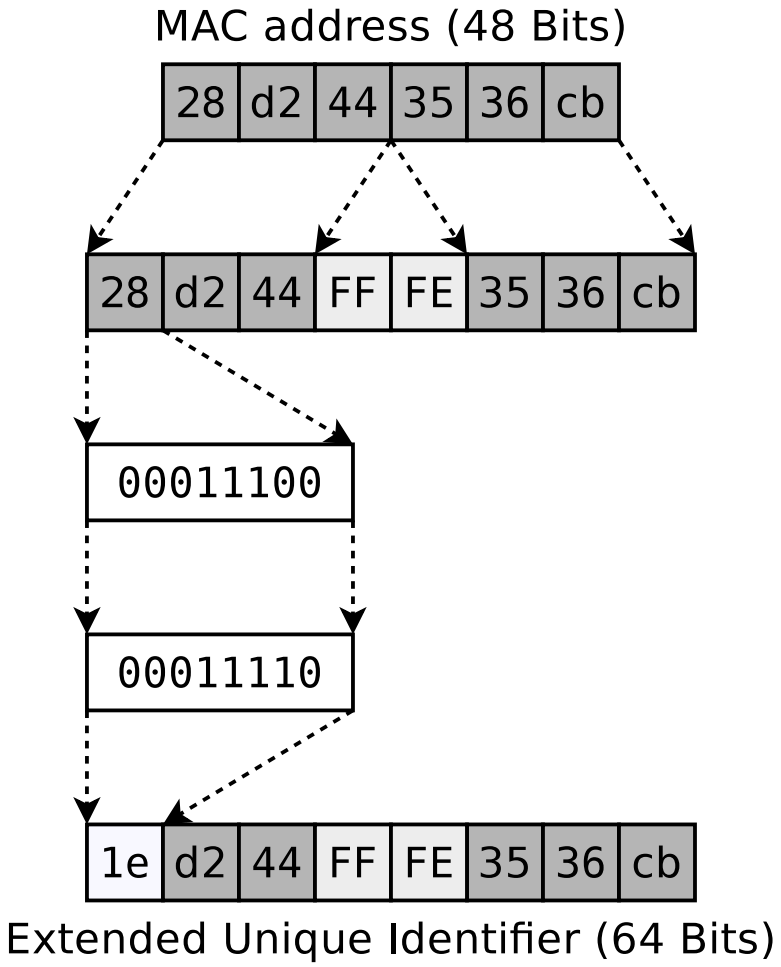
1. **Prefix** (Network Prefix)
   - Identifies the network
2. **Interface identifier** (Interface ID)
   - Identifies a network device in a network
   - Can be manually set, assigned via DHCPv6 or calculated from the MAC address of the network interface
   - If the interface identifier is calculated from the MAC address, it is called **Extended Unique Identifier** (EUI)
     - When this is done, the MAC address (48 bits) is converted into a 64-bit address $\Longrightarrow$ **modified EUI-64 address format** (see slide 45)

Some address spaces

`fe80::/10` $\Longrightarrow$ Link local addresses. They are only valid in the local network and are therefore not forwarded by Routers
`2000::/3` $\Longrightarrow$ (`2000...` until `3fff...`) Global unicast addresses. Routers forward them
`ff00::/8` $\Longrightarrow$ All addresses `ff...` are multicast addresses. Since IPv6 has no broadcast addresses, multicast addresses implement the broadcast functionality. The addresses `ff01::1` and `ff02::1` address all nodes in the local network and the addresses `ff01::2`, `ff02::2` and `ff05::2` address all local Routers
`2001:db8::/32` $\Longrightarrow$ Addresses only for documentation purposes

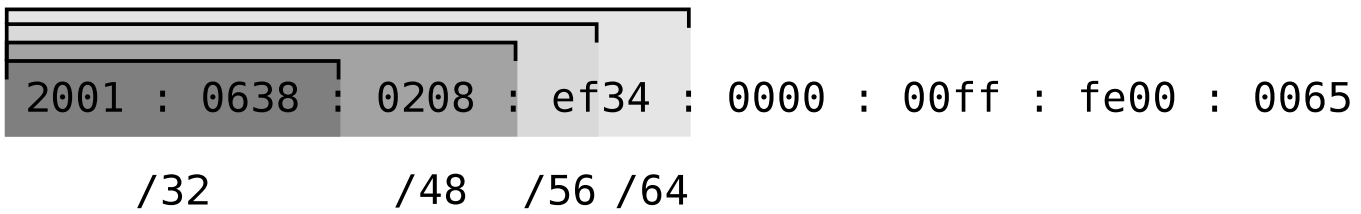
Structure of IPv6 Addresses and Networks (4/5)

- Converting a MAC address in the modified EUI-64 address format
  1. The MAC address is split into 2 parts of 24 bits
     - The 1st part becomes the first 24 bits
     - The 2nd part becomes the final 24 bits of the modified EUI-64 address
  2. The free 16 bits in the middle of the EUI-64 address have the following bit pattern: `1111 1111 1111 1110` (hex: `FFFE`)
  3. Finally, the value of the seventh bit from the left is inverted



Structure of IPv6 Addresses and Networks (5/5)

- (Sub-)netmasks do not exist in IPv6
  - The subdivision of address ranges into subnets is done by specifying the prefix length
- IPv6 networks are specified in CIDR notation
  - The address of a single device sometimes has `/128` attached
  - An example is the loopback address of IPv6: `::1/128`
    - All bits – except the last one – have value `0`
      (For IPv4, the loopback address is: `127.0.0.1`)
  - Internet Providers (ISPs) or operators of large networks get the first 32 or 48 bits assigned from a Regional Internet Registry (RIR)
    - The ISPs or network operators split this address space into subnets
    - End users usually get a `/64` or even a `/56` network assigned



- If a user gets a /56 network assigned, the 8 Bits between the Prefix and the Interface Identifier are the **Subnet Prefix**

Embedd IPv4 Addresses into IPv6 (*IPv4 mapped*)

- A globally routed (unicast) IPv4 address can be represented as an IPv6 address and thus integrated into the IPv6 address space
  - In literature, this approach is called *IPv4 mapped*
- The IPv4 address gets a 96 bytes long prefix:
  `0:0:0:0:0:FFFF::/96`

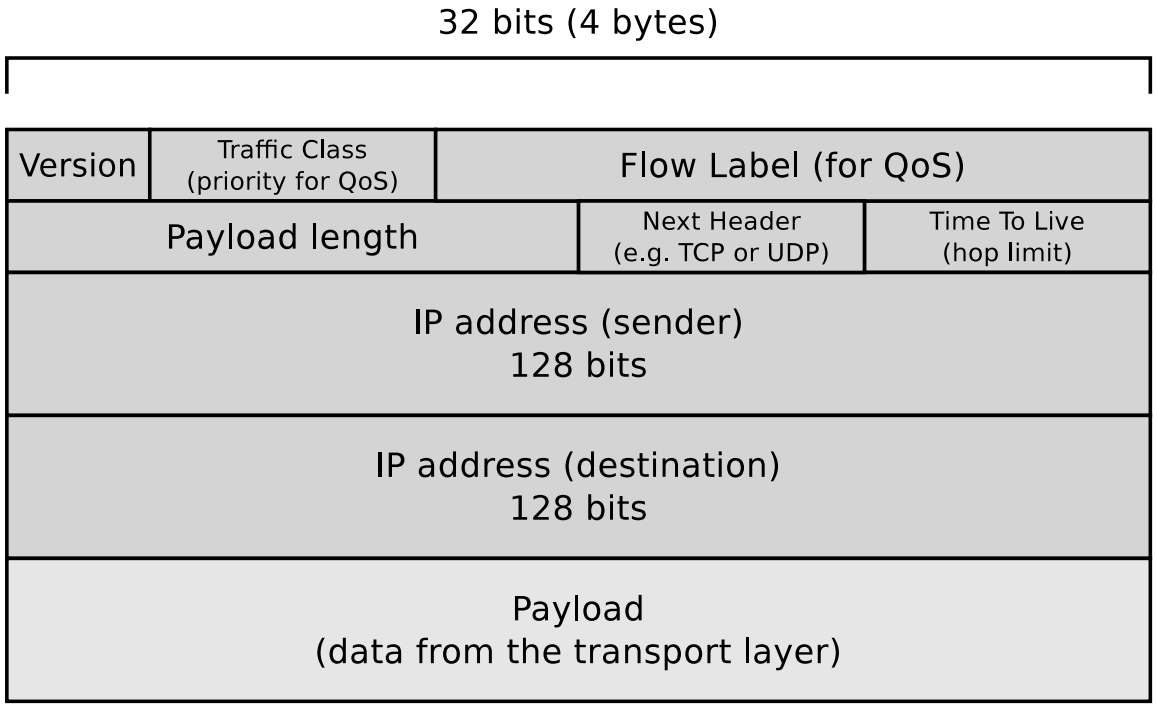
80 Bits					16 Bits	32 Bits
0000	0000	0000	0000	0000	FFFF	IPv4 address

- The IPv4 address may be represented in hexadecimal or decimal notation
- Example

IPv4 address:	`131.246.107.35`
IPv6 address:	`0:0:0:0:0:FFFF:83F6:6B23`
Shorter notation:	`::FFFF:83F6:6B23`
	`::FFFF:131.246.107.35`

Structure of IPv6 Packets

- The size of the IPv6 header is fixed (320 bits $\Longrightarrow$ 40 bytes)



- The field **next header** points to an extension header field or identifies the Transport Layer protocol (e.g. TCP = type 6 or UCP = type 17) which is carried in the payload of the packet

Concept: Simplified (reduced) package structure, but simple option to add additional (new) features with a chain of extension headers

Extension Headers

- **Hop-By-Hop Options** (type 0, RFC 2460)
  - Contains optional information that must be examined by every node along the delivery path of the packet
- **Routing** (type 43, RFC 2460)
  - Can be used to specify the route of the packet though the network
- **Fragment** (type 44, RFC 2460)
  - Contains parameters for the fragmentation of packets
- **Encapsulating Security Payload** (type 50, RFC 4303)
  - Contains encrypted data for secure communication
- **Authentication Header** (type 51, RFC 4302)
  - Contains information used to verify the authenticity parts of the packet
- **No Next Header** (type 59, RFC 2460)
  - Placeholder to indicates that that there is nothing following that header
- **Destination Options** (type 60, RFC 2460)
  - Options that need to be examined only by the destination network device of the packet