

## 4 Network Layer



arrangement

### 4 Network Layer

#### 4.1 Switching Principles

#### 4.2 Network Layer Addresses

#### 4.3 Internet Protocol

#### 4.4 ICMP - Internet Control Message Protocol

#### 4.5 ARP - Address Resolution Protocol

#### 4.6 DHCP - Dynamic Host Configuration Protocol

#### 4.7 Network Address Translation

#### 4.8 Internet Protocol Version 6 (IPv6)

#### 4.9 Migration IPv6/IPv4

#### 4.10 Routing Algorithms and Protocols

#### 4.11 Exercises - Network Layer

#### 4.12 Summary - Network Layer

The Physical Layer and the Data Link Layer (OSI Layers 1 and 2) determine how data should be transmitted in the local area in order to adapt the transmission to the properties of the medium and protect against transmission errors. If, however, we want to transfer data not only between two neighboring systems but across a large and complex network between two end systems, there are additional tasks that have to be implemented primarily at network nodes. The essential functions that are located in the third layer of the OSI Model are switching and routing. Routing means to find appropriate paths in the network for the data transfer.

In this chapter, the various switching principles will be explained first. Then you will learn about the data transfer protocol IP - the Internet Protocol - used in the Internet in Layer 3. Both versions, IPv4 and IPv6, will be treated extensively. Relevant auxiliary protocols such as ICMP, ARP and DHCP will also be presented. The end of the chapter will provide a glimpse into routing protocols.

### 4.1 Switching Principles



arrangement

## 4.1 Switching Principles

### 4.1.1 Circuit Switching

### 4.1.2 Packet Switching

### 4.1.3 Virtual Circuits

### 4.1.4 Network Convergence

There are some basic possibilities for how communication works in wide area networks. What matters here in particular is the approach according to which the transmission of data units occurs in the network nodes. Two principles can be distinguished.

- **Circuit switching** and
- **Packet switching**

Both switching methods are presented here. There are also ways to combine them.

At the end of this section, we will consider the current situation at many telecommunication companies that provide telephone as well as data communication services for their customers. In the past, there were separate networks for both purposes, but now independent telephone networks are abandoned and telephony is handled via the data network. However, there are also conflicts to resolve here because the telephone network works according to circuit switching and does not fit well with the packet switching used in the data network.

### 4.1.1 Circuit Switching

The circuit switching principle was already introduced with the first **telephone networks**. This principle means that for communication between two participants, transmission capacities are reserved along the communication path. In the past (see info box at the bottom), these were actually wires that were manually plugged to create a physical connection. In modern networks, however, reservations are realized on a logical level. The reservations exist until the communication between the participants is terminated. The transmission capacities can be assigned to other participants only after this connection has ended. This means that it can sometimes happen that all network capacities are already used when an additional participant would like to access the network. In this case, the new participant is rejected. In telephone networks, this means that the participant gets a busy tone. Today this situation is uncommon in everyday life because the networks have a lot of resources.

The components in the network need to be able to manage existing reservations. They do not need buffers to store data because all incoming data can always be forwarded immediately.

If the connection is established on demand, it is called a **switched connection**. If, however, two participants are connected for a long period, they are connected via a **dedicated line**.

### Advantages

- **Guaranteed bit rate:** A bit rate is guaranteed to the participants through the end-to-end reservation. This is also independent of how many participants are currently transmitting. However, the guarantee only applies to a fault-free network.
- **Minimum delay:** The reservation of a fixed path between participants results in the data arriving at the receiver with minimum delay, which is unavoidable due to the signal propagation time. This is particularly important for telephone calls or video conferences because variations in the delay time affect the user experience of these applications.
- **Sequence preservation:** Because the same path must always be taken through the network, the data sequence is preserved.
- **Little overhead:** By managing the reservations in the network, it is possible to know which data units belong to which connections. Corresponding address data does not have to be contained in the data units themselves.

### Disadvantages


- **Poor network use:** Due to guaranteed transmission capacities for the participants, which apply rigidly for a long period, the network capacity is poorly utilized. This means that when a participant does not use the bit rate allocated to him, the capacities cannot be made available for other participants on short notice. This is the case, for example, with telephone calls because in a telephone call usually only one speaker speaks at a time. The return channel remains unused. If you also take into account pauses in the discussion, the use rate is often only 20 to 30 percent. This problem becomes significantly more severe with Internet data traffic, which typically has strongly varying bit rate requirements. For example, when surfing the web sometimes a lot of data is downloaded when someone visits a 3 MB size home page. While the page is subsequently being viewed, no communication is necessary.
- **Delay due to connection setup:** The reservation has to be made before communication is possible. This takes time, which is a disadvantage in particular for short-lived connections.

- **Dealing with line interruptions:** Circuit switching does not perform well when there are errors in the network on the transmission path, i.e. when a network node fails or the transmission path no longer works. The connection is then no longer usable, a new end-to-end path has to be found and resources have to be reserved again. This is noticeable in a classical telephone call so that the conversation ends abruptly and the participants have to dial again.
- **Rejection of participants:** It can happen that all transmission capacity for a link is already assigned to other participants. In this case, further reservation requests that also need this link are rejected by the network.

In conclusion, you could say that circuit switching is appropriate for telephone networks. With telephone calls, you can estimate which bit rates are needed for good call quality and you can make a fixed assignment of this bit rate to the participants (e.g., 64 kbit/s with ISDN). It is, however, not very suitable for Internet data traffic because the bit rates vary considerably here such that a fixed allocation would lead to poor network utilization.



In the online version an video is shown here.

Link to video : <http://www.youtube.com/embed/PpwVu22ij9E> 

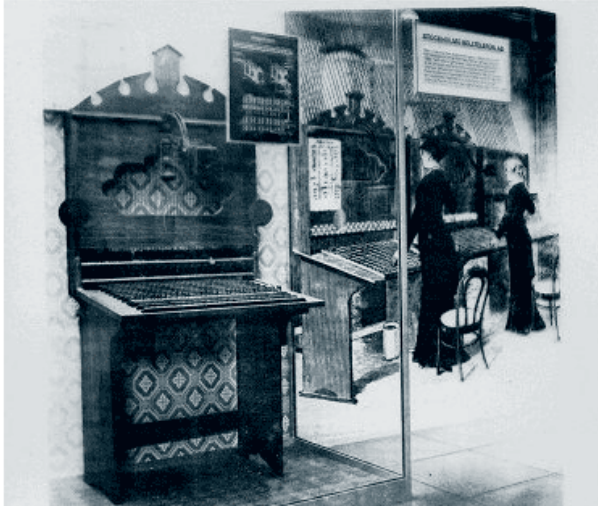
**Circuit switching**



indentation

### The beginnings of circuit switching

You have probably all got an idea of the original circuit-switching technology from old films: The desired connection was indicated to the call center by rotating a crank inductor. An operator would then call the desired participant using a crank inductor and connect both participants with the help of a pair of cables in a connection cabinet. This created a physical connection.



The operator

### 4.1.2 Packet Switching

In contrast to telephone networks, a different approach is used in the **Internet: packet switching**. Here, the data to be transmitted are divided into individual data units so that they can be transmitted separately from one another. The data units in this approach are called **packets**.

The packets are sent from node to node to the destination and have to be stored temporarily at each node. All data packets must therefore contain all information required for their correct forwarding. The routing decisions here are made again at each node for each incoming packet. A particular path is therefore selected for each packet so that the packets can be transmitted along different paths even when they are exchanged between the same participants. This has the major advantage that in the event of problems in the network, new paths can be selected quickly so the communication can continue.

Because **overtaking** of packets may occur with this approach, applications, in which the sequence of the packets is important, have to have mechanisms to recreate the correct packet sequence at the receiver.

No reservations are made for packet switching. This means that the end systems can send as many packets in the network as they wish. The transmission is only limited by the bit rate of the network access. Therefore, within the network the network nodes may sometimes be overloaded. Buffers in the nodes can compensate for this over the short term, but for long-lasting overloads, the packets have to be dropped by the nodes. As a

consequence, there are no guarantees as to when or whether the packets will arrive at the receiver.

### Advantages


- **Flexible bit rates:** With packet switching, the packets can be sent with the bit rate that is needed at the time. This means the network is utilized much better than when providing fixed reservations. It is not guaranteed, however, that the packets will arrive at the destination.
- **Robustness:** When nodes or links in the network fail, alternative paths can be found on short notice. In such cases, only few or no packets are lost so participants do not notice the situation or notice it only very little. This advantage was essential for the development and selection of packet switching as basis for the Internet (more precisely, its predecessor ARPAnet). At the time, the hardware in the switching nodes was unreliable so failures had to be managed well.
- **No connection setup:** Because this approach is not connection-oriented, no connection setup is necessary. This means there is no waiting time, and the first packet can be transmitted immediately.

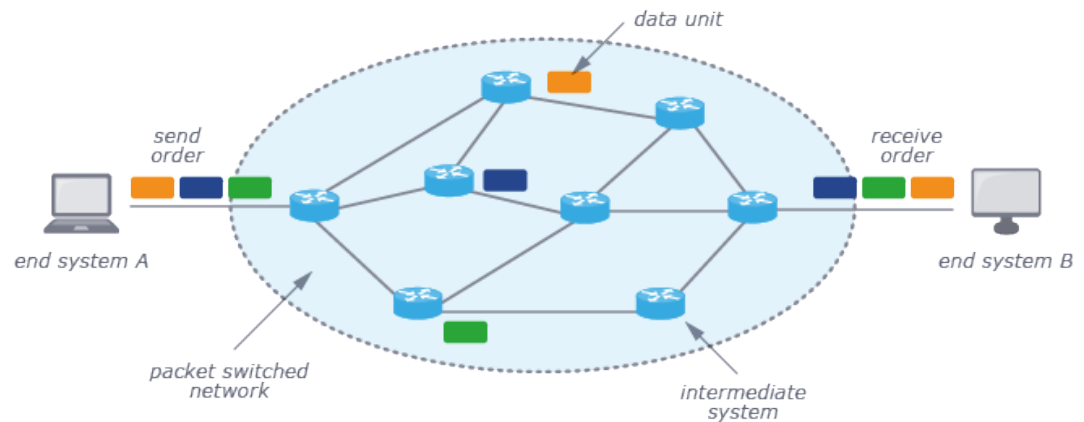
### Disadvantages

- **Packet loss possible:** Because there are no reservations, it can happen that at times too many packets are sent simultaneously. In these cases, the packets are dropped in the network when the network components are overloaded. This means the packets will not reach their destination. Depending on the application, it must be ensured then that appropriate retransmissions are carried out.
- **Overtaking possible:** Because for each packet and each forwarding, a decision must be made about which route is taken, packets can be overtaken by other packets. This has to be dealt with if the right sequence is important for the application.
- **Time variance:** The buffers in the nodes can be temporarily full or empty. This results in packets needing different times to traverse the network. Combined with packet loss and overtaking, this can especially be a problem for voice and video transmission.
- **Overhead:** Because packets are transmitted through the network independent from each other, they have to contain complete destination and source addresses.



In the online version an video is shown here.

Link to video : <http://www.youtube.com/embed/A1E6S302KYY> 

**Packet switching****Datagram switching**

The figure on this page refers to **datagram switching** as a typical type of packet switching. It is possible, however, to emulate properties of circuit switching in a packet switched network. This configuration is called **virtual circuits**.



In the online version an video is shown here.

Link to video : <http://www.youtube.com/embed/MWdiJcyc2jw>

**Datagram switching****4.1.3 Virtual Circuits**

For **virtual circuits**, there are several ways in which circuit switching and packet switching can be combined. There are long-term switched virtual circuits or dynamic ones. The description that follows refers to dynamic circuits.

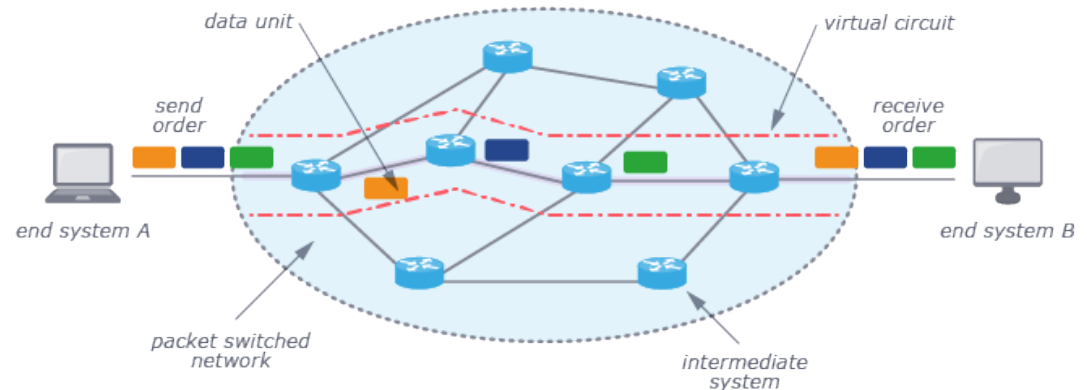


In the online version an video is shown here.

Link to video : <http://www.youtube.com/embed/E6k-aqj0qXc>

**Virtual circuits**

Path selection processes start in every network node through an initial transmitted packet, and the resources needed for this circuit are reserved. The path selected by the nodes will be reserved for all further packets belonging to this logical connection so that, as with circuit switching, all packets are sent along the same path.



#### Virtual circuits in a packet-switched network

It is therefore not possible that packets overtake each other. Since the end users have the impression that they have a dedicated line which is actually not the case, the configuration is called virtual circuits.

In addition, links in the network are used for the simultaneous transmission of packets belonging to different logical connections. This means that the available transmission capacity is better utilized. On a link, several virtual circuits can therefore exist at the same time. If a link fails because of a defect in a node, for example, a new route through the network can be found for all subsequent packets by performing an automated **route change**. In this case, only the few packets that are currently on the link will be lost.

#### Advantages

- **Guaranteed performance:** There can be certain guarantees for performance parameters in such a network. By retaining the sequence and the pre-reserved routes, delays and delay variations are kept within limits. The bit rate needs can also be planned better and the medium can be better utilized. The guarantees are not, however, as strict as with circuit switching. If the data volumes resulting from many participants are very high, these needs may not be met. This is sometimes also referred to as a statistical guarantee.
- **Robustness in case of failures:** In case of failure of links or nodes, other routes can be found dynamically. Because the reservations refer to the original path, the performance may, however, be limited.

#### Disadvantages



- **Effort for reservations:** Reservations have to be made in the network. Scalability has to be considered here because reservations for every individual data stream would overburden the network. That is why such an approach called [IntServ](#) failed. A choice therefore has to be made about what types of data streams require what type of treatment. For reasons of scalability, complex decisions generally have to be made at the edge of the network, whereas the interior of the network, where many data streams are concentrated, has to be kept as simple as possible. This is considered in the successful approaches [DiffServ](#) and [MPLS](#).
- **Time delay due to connection setup:** If a connection first has to be established dynamically, this leads to a delay until the transmission of user data can be started.



summary

The principles of packet switching and circuit switching are combined in the Internet in the interior of the provider networks in the described manner ("virtual circuits"), although the Internet originally used only packet switching. This type of combination is especially relevant for voice and video telephony.

### 4.1.4 Network Convergence

Today's networks allow data exchange for various **services** such as WWW, e-mail, telephone calls and video conferencing. The various services are combined in a common network. Whereas in the past, different networks existed for different purposes (especially the separation of telephone and data networks), there has been a convergence of networks so that one network is used for all services.

In practice, the telecommunication network has been integrated into the data network so that telephony is realized as voice-over-IP (VoIP). This is also referred to as "All IP" because the Internet protocol is the basis of everything. There are some difficulties that also need to be noted here because data networks are designed for **asynchronous** communication where the processing of data units at both communication partners need not happen at certain determined times. This is not a problem for e-mails or web surfing. Telephone or video conferences, however, rely on such (**synchronous**) timing for high-quality transmissions. To ensure that this is possible in a data network that is not really suitable for this purpose, such data traffic has to be preferred over other data traffic.

**Network neutrality** is a political issue that is related to this discussion. This issue is about whether IP packets are treated equally in the network or whether certain data traffic may be prioritized by the provider. This prioritization could happen through payments by third-party service providers or users, or the provider could prioritize its own services. The use of the provider's own services, for example, might not be charged in a data volume-based tariff, or these services could be reachable with better quality than competitor services.

## 4.2 Network Layer Addresses

The assignment of addresses is one of the tasks of the Data Link Layer. As already mentioned in the previous chapter, in practice these are MAC addresses. However, other addresses are introduced at the Network Layer. In practice, these are the IP addresses that belong to the Internet protocol. The question arises therefore why an additional type of addressing is necessary.

It is due to the **scalability problem of MAC addresses**, which are for this reason not suitable for global routing. If you look at a typical network based on switches, you can see that each switch creates an entry in the MAC address table (bridge table) for each MAC address that it encounters. It is not possible to summarize the entries, which means that in large networks there are many entries in the bridge table if a switch is at a central point in the network. Continuous updating of the address assignments is also necessary.

An analogy would be organizing mail delivery in a country on the basis of ID card numbers (at least in Germany every citizen ID card has a unique number). You would then write the ID card number of the appropriate person on the envelope as the recipient. This would indeed be a unique address, but there would be a huge table with all ID card numbers and their assignments to homes, which would need to be searched through each time a letter is forwarded.

Systems such as postal service or telephone numbers function, however, in a **hierarchical fashion**: For the postal service this is done with zip codes, and for telephony it is done by dividing the number according to country code, area code, and access number within this area network. This has the great advantage that, for example, when calling from Hamburg to Munich the Hamburg central office only needs to know that the call should be transferred to Munich. The details of the Munich telephone network only need to be known at the Munich central office. By referring to the area through the addresses, you can distribute the required knowledge, which makes the approach scalable.



In the online version an video is shown here.

Link to video : <http://www.youtube.com/embed/lfvqGWkNHPE>

#### Address spaces

The hierarchical structuring is done in a relatively simple manner for IPv4 addresses. The first bits of the IPv4 address refer to a network area; the bits at the end of the IPv4 address indicate the end system (or more precisely the interface of an end system) within the network area. In IPv6, there is a further subdivision of the network area in order to improve scalability even more.

## 4.3 Internet Protocol



arrangement

### 4.3 Internet Protocol

#### 4.3.1 Introduction to IPv4

#### 4.3.2 IPv4 Header

#### 4.3.3 IPv4 Addresses

#### 4.3.4 Routing

#### 4.3.5 Subnets

#### 4.3.6 CIDR - Classless Inter-Domain Routing

#### 4.3.7 Fragmentation

#### 4.3.8 Path MTU

The most important Network Layer protocol is IP (Internet protocol). It is also the only one that is still relevant in practice today. This was the result of market developments and led to the success of the Internet on the basis of this protocol (in close relation with the Transport Layer protocol TCP). Other Network Layer protocols such as IPX/SPX (from Novell), AppleTalk and the OSI (X.25) network layer were completely displaced by the Internet protocol.

This section first discusses the Internet protocol version 4. After considering auxiliary protocols that belong to IPv4 in the following sections, IPv6 is presented in a separate section.



In the online version an video is shown here.

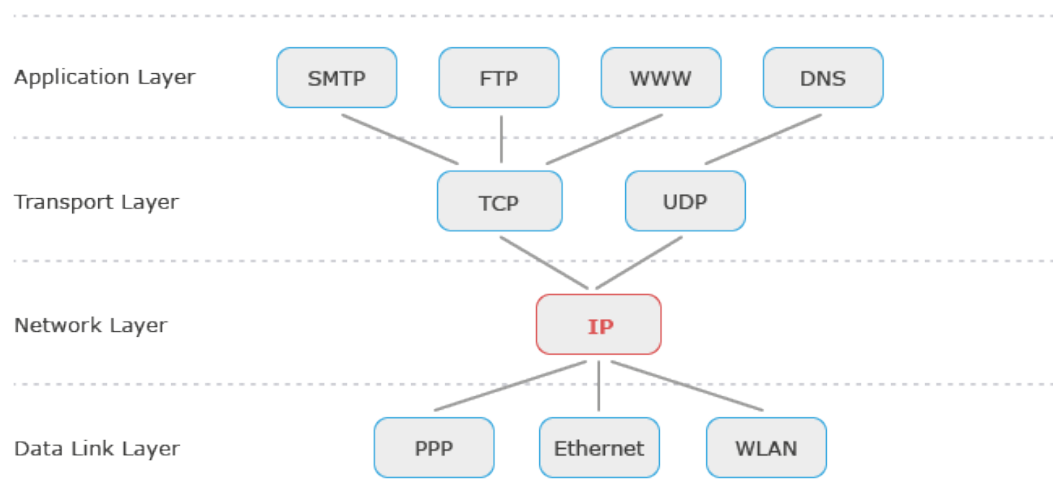
Link to video : <http://www.youtube.com/embed/oBs3N7PEqHE>

**Internet Protocol Version 4**

### 4.3.1 Introduction to IPv4

The **Internet protocol** (RFC 791, September 1981) is used for communication between computers in packet-switched networks. It transmits data units of various lengths between systems that are identified by the protocol addresses (**IP addresses**). The data units are referred to as packets or datagrams. The selection of a route, which generally affects multiple intermediate systems, is called **routing**. The intermediate systems that perform the corresponding routing are called routers. The Internet protocol can also disassemble and reassemble long datagrams when a network only allows small packets. This is referred to as **fragmentation** and reassembly.

The Internet protocol is the only protocol from the entire range of protocols that has been developed in relation to the Internet that must necessarily be installed on every device that wants to use the Internet to communicate. If another protocol is used in its place, the device is no longer part of the Internet. So it is not surprising that the Internet received its name from this central protocol.




**Internet protocol stack**

The Internet protocol treats each individual packet as an independent data unit that does not stand in a relationship with other packets. The routing is done again for each individual packet as provided for by packet switching. There are no fixed connections, neither virtually nor physically. Therefore, if a part of a network fails, a new route can be found easily, and the user generally does not notice it.



practice

Reliability in this regard was the most important criterion for the development of the Internet protocol. The development was financed for the most part by the US Department of Defense. But it is not true that the network was designed for a nuclear war scenario that assumed extensive destruction of the United States. The technology was actually so unreliable at the time that there were frequent hardware failures (see [Doyl06](#) ).

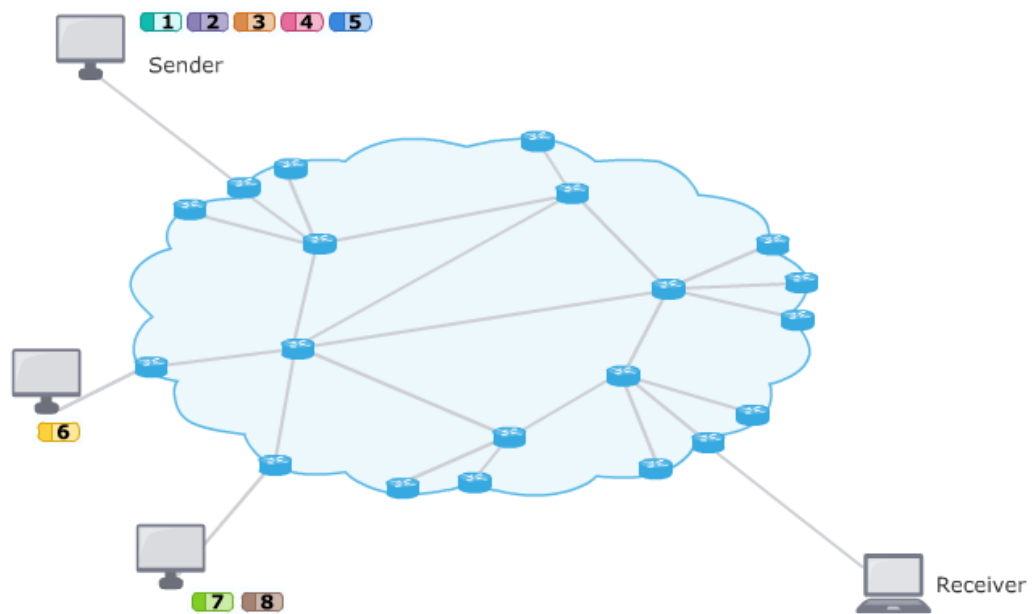
The Internet protocol does not provide mechanisms for reliable communication. There are no confirmations. There is no error monitoring for data transfer, only a header checksum. If a bit error is detected with this checksum, the packet is discarded. The auxiliary protocol ICMP is used for error notification, which however is only used in case of serious error situations. There is no flow control or congestion control to prevent overloading of receivers or the network.



In the online version an animation is shown here.

**Unsafe data transmission with IP**  
**The principle of the Internet**

Begin printversion

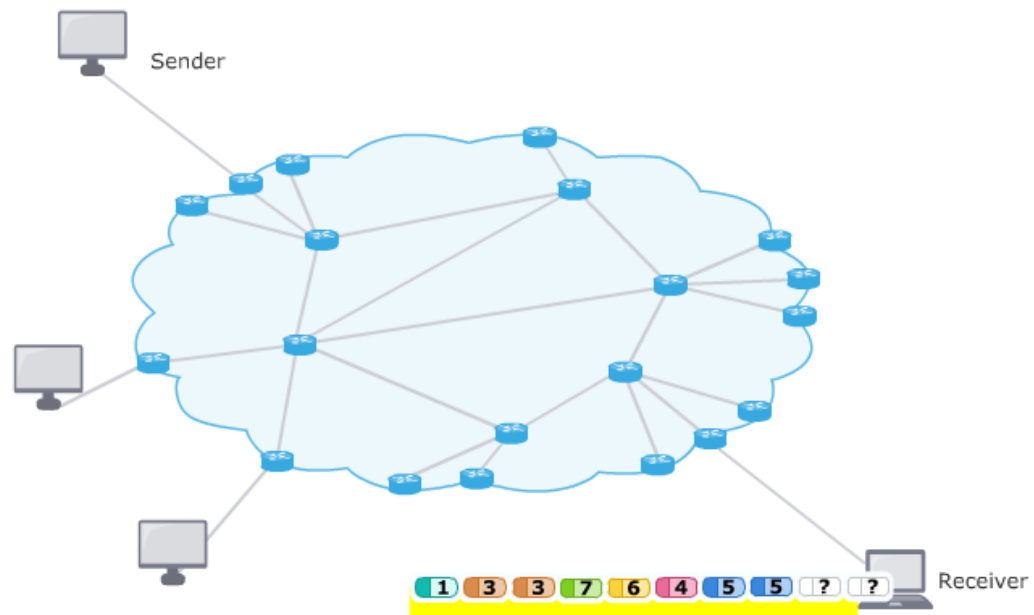


The Internet Protocol is an unreliable datagram protocol. On data transmission via the protocol, it is possible that packets are lost within a router. One reason for this can be bit errors in a packet so that the packet is discarded. This case is rather unlikely nowadays. However, it is much more likely that many data are sent towards a router which is overloaded by them. The router can then no longer accept newly received packets.

Every router decides for each packet about the way to forward it. It can happen that different routes to the destination are used. On these different routes, packets may overtake each other. Consequently, the receive order may be mixed up.

It is also possible that packets are duplicated inside the network.

Due to the explained possible events which are not compensated by the protocol a disordered and incomplete sequence of packets may arrive at the destination which may contain duplicated data.



End printversion

### 4.3.2 IPv4 Header

The following interactive graphic shows the structure of the IPv4 header.



In the online version an rollover element is shown here.

#### Internet Datagram Header

Begin printversion

0		8		16		24		31	
Version	IHL	Type of Service DiffServ ECN		Total Length					
Identification				Flags	Fragment Offset				
TTL		Protocol		Header Checksum					
Source Address									
Destination Address									
Options						Padding			

**Version:** Internet protocol version

**IHL:** IP Header length including options

**Type of Service DiffServ/ECN:** Describes the quality of a service and defines the ECN field

**Total Length:** provides overall IP packet length (header and data)

**Identification:** is used for fragmentation

**Flags:** is used for fragmentation

**Fragment Offset:** is used for fragmentation

**TTL:** Time-to-live, max. number of packet forwards by routers

**Protocol:** specifies the upper-layer protocol (Transport Layer protocol) used in payload part of packet

**Header Checksum:** check for bit errors in IP Header, has to be recalculated by every router

**Source Address:** 32 bit long IP source address

**Destination Address:** 32 bit long IP destination address

**Options:** : The IP Header can contain options

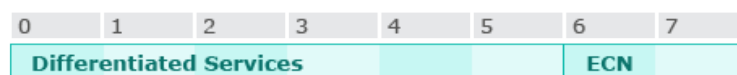
**Padding:** If the options field length is not a multiple of 32 bits, this field adds dummy bytes to it accordingly

End printversion

**Version:** The Internet protocol version is noted here. A binary-coded four is used for IPv4 and a binary-coded six is used for IPv6. If it is IPv6, the rest of the header is structured differently.

**IHL:** This is the length of the IP header including options. It is measured in 32 bit (4 byte) words. The header can have a maximum length of  $15 \cdot 4 \text{ bytes} = 60 \text{ bytes}$ . The minimum length of the IP header is 20 bytes; therefore the smallest possible value for IHL = 5.

**Type of Service, DiffServ, ECN:** This field was originally intended as a whole to describe the quality of a service and in this way to influence the routing. This field was, however, later redefined as DiffServ field.



**Diff Serv, ECN**  
Diff Serv and ECN instead of TOS

The first 6 bits are defined as **differentiated services** (DiffServ or DS, RFC 2474) and describe the quality of the service. The provider has the option here to define different traffic classes for itself and then handle them differently in its own routers. The last two



bits are defined as the **ECN** field (explicit congestion notification, RFC 3168) and are used to indicate when overload problems occur in a router. The actual use also depends on the ISP.

**Total length:** The 16-bit field specifies the total length (header and data) of the datagram. The maximum length is 65535 bytes. Each host must be able to receive a datagram with a length of 576 bytes or less.

**Identification:** The identification, flags, and fragment offset fields are used when a datagram is too large for the network interface, i.e. is larger than the MTU and therefore has to be fragmented (see [Fragmentation](#)). The individual fragments of a datagram get the same identification so the receiver can reassemble the fragments again.

**Flags:** This field is evaluated bit by bit.

1. The first bit is not used.
2. If the second bit (DF) is set, the packet may not be fragmented (DF = **Don't fragment**). If fragmentation was actually necessary, the packet would be dropped and an error message would be sent via ICMP.
3. If the third bit is set (MF), there is still at least one additional fragment that will follow (MF = **More fragments**). This bit is not set only in the last fragment. If the IP packet is not fragmented, then the bit is not set.

**Fragment offset:** The fragment offset indicates how many bytes of the original IP packet were already contained in the previous fragments. The value is specified in multiples of 8 bytes. The value is 0 for the first fragment or for non-fragmented IP packets.

**Time to live (TTL):** This 8-bit field was originally intended as a maximum time period by which the IP packet must have reached its destination. The intention was to prevent in case of wrong configurations that packets continuously circulate in the network and thereby overload it. The meaning as a time period was then changed so that the value now indicates a maximum number of forwards by routers. Each time the packet is forwarded, the value is reduced by 1. When TTL reaches zero, the packet is discarded and an error message is sent via ICMP. This field is called **hop count** in IPv6, which fits better to the actual usage.

**Protocol:** This field specifies the Transport Layer protocol that is used in the payload part of the datagram. The typical values are ICMP = 1, TCP = 6 and UDP = 17.

**Header checksum:** The header checksum is used to check whether bit errors have occurred in the transmission of the IP header. If there are bit errors, the IP packet is simply discarded and no error message is sent (even not by ICMP).

Because the header is changed by every router (the TTL value is reduced), the checksum must be recalculated in each router. The calculation is simple: The field is calculated as the 16-bit one's complement of the sum of the 16-bit one's complement words of the header. The field itself is set to zero before calculating.

**Source / destination address:** The source and destination IPv4 addresses are contained here, each of which consists of 32 bits.

**Options:** Options can be transmitted in the IP header, but this is usually not done. Without options, the header has a length of 20 bytes. Because the entire header can have a maximum length of 60 bytes, there are only 40 bytes available for options. This means the use of some options is very limited. The “record route” option is used so routers can store their IP addresses in the options. This allows the receiver to identify which route the packet used. However, only less than ten addresses can be stored in the option field. This is not enough in today's Internet, for example, to record a route between Germany and the USA completely. A total of more than 20 different options have been defined, some of which are obsolete however (see RFC 6814). The [detailed list](#) can be accessed at IANA.

### 4.3.3 IPv4 Addresses

The 32-bit long IPv4 addresses are used globally for **unique identification** of a device's network **interface**. This means that devices with multiple interfaces have an IP address for each interface. This is particularly true for routers, but may also occur on end systems. IP addresses consist of a network identification and a host identification. In the context of the Internet, host is the usual name for a server or end system. This older term comes from the fact that such a system is host for the programs running on it.

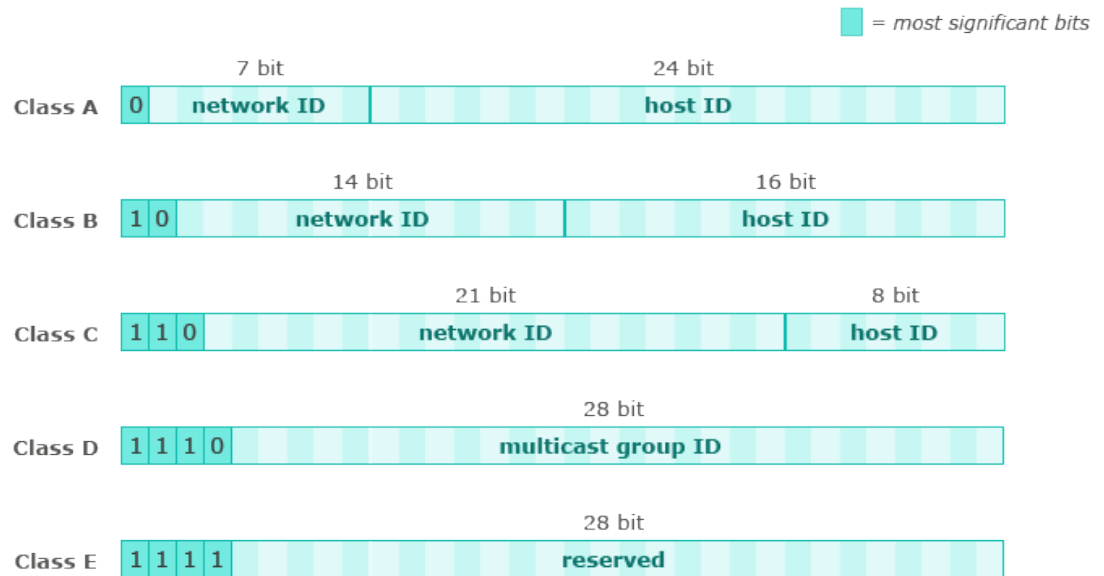


In the online version an video is shown here.

Link to video : <http://www.youtube.com/embed/faTq8LwCPDw>

**IPv4 addresses**

Until 1993, the so-called class-based classification of IP addresses was used. The assignment of classes was dependent on the highest-value bits at the beginning of the address, as depicted in the following figure. Through the assignment of classes, the division of areas within an IP address into a network ID and host ID was clear.



#### The five classes of Internet addresses

Beginning in 1993, this class-based division was replaced by CIDR (see [CIDR - Classless Inter-Domain Routing](#)). Because, however, in practice people often talk about "class B" and "class C" networks, the old class-based division is still presented here.



#### example

#### Typical class C address

This is a typical class C address: 192.175.123.34

In the binary representation, the address looks like this:

11000000.10101111.01111011.00100010

The usual way to represent IPv4 addresses is the **dotted decimal notation**. Each byte is given as a decimal value and the bytes are separated by dots. The highest decimal value which is possible in an IP address is therefore 255.

The host IDs, in which all bits are set to 0 or 1, have a special meaning and can therefore not be used for individual network interfaces. If the host ID consists only of zeros, the network itself is specified and not an individual host in the network. All hosts in a network are reached with a host ID that consists of only ones. This address is called a **broadcast address**.  $256 - 2 = 254$  hosts can therefore exist in a class C network.



example

**All hosts in the network: Broadcast**

Example: All hosts in network 192.175.123.0 are addressed with 192.175.123.255.

Not all possible addresses can be found in the Internet. Some address ranges are defined as **private addresses** - they may not be routed in the Internet. They can be used in a private network, for example, within a company. A network operator can assign these addresses in an arbitrary way to its own devices without the need to consult in advance with other network operators or registration authorities. All other addresses must be registered publicly, which is a key job of IANA and its subordinate organizations.

The following address ranges are reserved for private networks:

Dotted notation	Private networks
10.0.0.0	1 Class A network
172.16.0.0 to 172.31.255.255	16 Class B networks
192.168.0.0 to 192.168.255.255	256 Class C networks

**Internet address ranges that are reserved for private networks**

The address **127.0.0.1**, which is also called **localhost**, has a special meaning: Packets sent to this address do not leave the computer but are delivered to other processes that also run on the computer. This is often referred to as using the loopback interface.



practice

127.0.0.1 is used in the development of network applications.

This is important, for example, if a complex website is developed. A WWW server must be installed for this purpose. If the computer used for development and the WWW server run on different computers, the development is complicated because the sites to be tested must be transmitted from the development computer to the WWW server. It is easier if the WWW server is also installed on the development system. In this case, it is possible that the web server can be reached using this special IP address. Sites that are under testing can then be accessed via the WWW browser for example with "http://127.0.0.1/index.html".

### 4.3.4 Routing

The most important role of the Internet protocol is routing, i.e. finding a path through the Internet for a packet. This is the primary task of routers, which have to decide which interface is used to forward a packet. However, end systems also make routing decisions when sending a packet.

To perform routing decisions, a **routing table** is needed. It stores information about which routers allow to reach certain IP addresses.

A typical entry in the routing table might be:

Receiver address	Next router	Interface
210.10.20.0	192.173.123.1	192.173.123.5



Typical entry in a routing table

This means that hosts in network 210.10.20.0 can be reached via the router 192.173.123.1; the interface used for this has the IP address 192.173.123.5.

For routing decisions within end systems, it is necessary to distinguish between two cases.

- If the receiver is located in the connected network or is directly connected (e.g. over PPP), the packet is sent directly (i.e. forwarded to the receiver using only the Data Link Layer),
- otherwise the packet is sent to a router in its own network.

Most hosts, very likely also the computer you are using right now, can only reach a single router. All packets that are not sent to hosts in one's own network must therefore be sent to this router. It is therefore called the **default router** (for historical reasons also **standard gateway** or **default gateway**). It is obviously not necessary to provide a special address in the routing table because all packets that do not remain in one's own network are sent to this router. If the entry for the default router is not present, the Internet cannot be reached.

The entry for the default router is the following:

Receiver address	Router	Interface
------------------	--------	-----------

0.0.0.0	192.173.123.1	192.175.123.5
---------	---------------	---------------



Entry for the default router




annotation

When naming the default router, there are several possible ways of writing it. The IP address **0.0.0.0** is given here. It can also be called **default** or **standard gateway**. All of these are common and refer to the default router.



practice

In Windows systems, you can see the routing table of your computer using the command “netstat -rn” on the command line. The default router is also shown using “ipconfig -all”. For Linux/Mac, the similar command is “ifconfig”.

Not only network addresses are allowed as receiver addresses; individual host addresses are also routable. This is, however, rare because the routing tables become very large when too many host addresses are included. Instead, all efforts are aimed at keeping the routing tables as small as possible so searching the table is as quick as possible and the packets only need to be kept on the router for as short a time as possible. Nevertheless, routing tables at central routers in the Internet have several hundred thousand entries (see [statistics for AS 6500](#) .



example

### Routing example in a network

The following animation shows

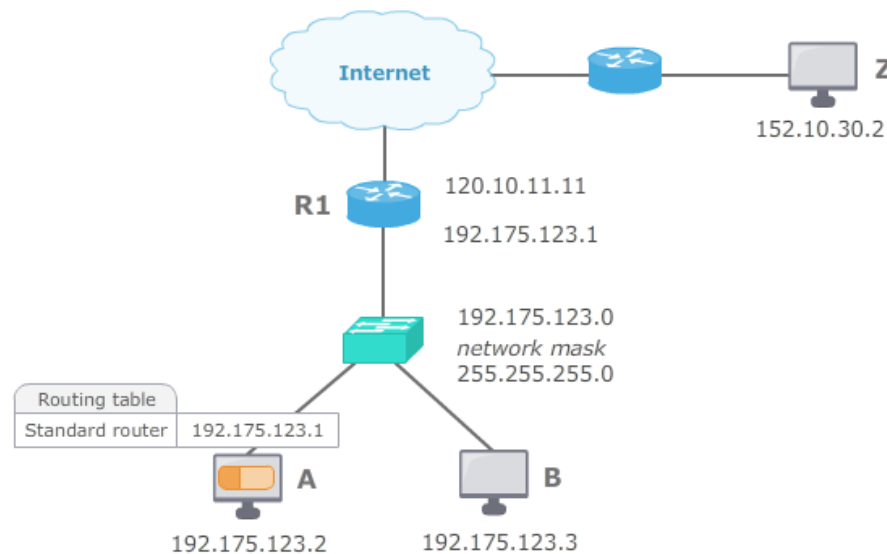
1. The routing tables of the host and the router in the depicted network,
2. The routing of a datagram from host A to host B
3. The routing of a datagram from host A to the Internet
4. The routing of a datagram from host B to the Internet



In the online version an animation is shown here.

**Routing example in a network**

Begin printversion



We see a simple network with the network ID 192.175.123.0 and the network mask 255.255.255.0. This implies that the last 8 bits are used in this network to distinguish between hosts. In the network there is router R1 which is connected to the Internet. Its host ID is 1. Computer A has the host ID 2 and computer 3 has the host ID 3. We let us take a look at the routing tables. In the routing table of host A, the router with the IP address 192.175.123.1, that is router 1, is specified as standard router. The routing table of router B has no entry.

Now let us take a look at the first case. A datagram should be sent from A to B. The network ID of source and destination are the same. Therefore, the datagram is sent directly without accessing the routing table.

In the second case, a datagram of A should be sent to the Internet, for example to the IP address 152.10.30.2, that is computer Z. The network IDs are different. Now there is a lookup for the destination address in the routing table, but no entry is found. Afterwards, a lookup for the standard router is performed. The datagram is sent to the IP address 192.175.123.1, that is router R1. A has no knowledge how the router forwards the datagram.

In the last case a datagram of B should be sent to the Internet, for instance to the computer Z. The network IDs are different. Now there is a lookup of the destination address or a standard router in the routing table, but nothing is found. Therefore, the Internet is not reachable from B. There is the message "Network or host unreachable". Host B discards the datagram. This host can only communicate with hosts in its own network. It cannot reach other hosts in the Internet.

End printversion



example

### Routing example with two networks

The following animation shows

1. the routing tables of the hosts and routers in both depicted networks,
2. the routing of a datagram from host C to host Z

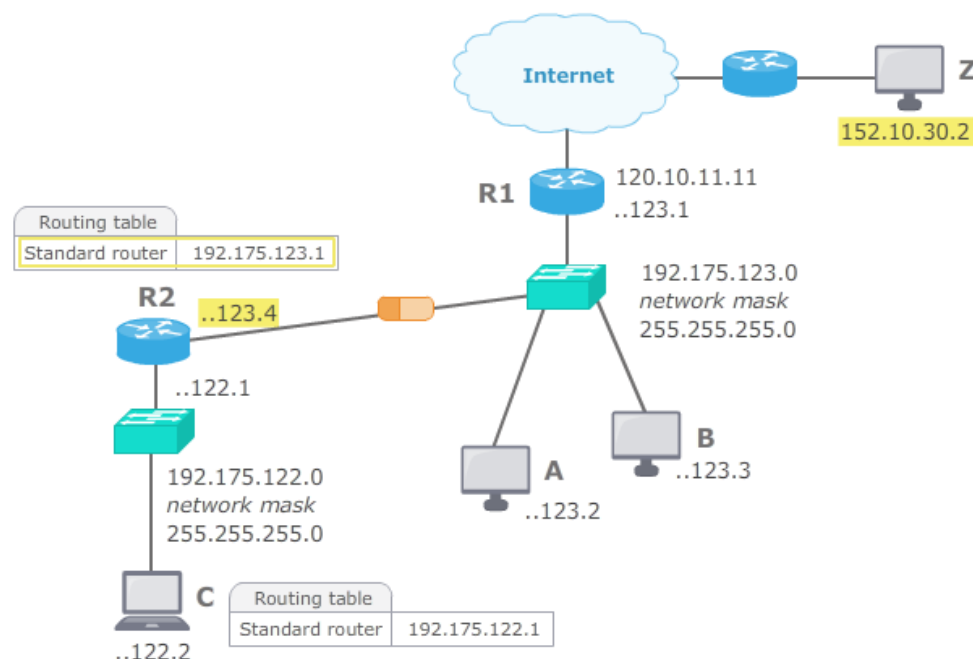
The animation contains a large number of IP addresses almost all of which start with 192.175... . To facilitate reading and listening, the first two bytes are left out of the figure and the audio and only the last two bytes are considered.



In the online version an animation is shown here.

### Routing example with two networks

Begin printversion



The network from the previous example is connected to a second network via router R2. Therefore, we have the networks 192.175.123.0 and 122.0. Router R2 has two interfaces which are connected to one of these networks each. Which routing table entries are necessary so that all hosts can communicate with each other and with the Internet?



Let us start with host A. It can directly communicate with host B, router R1 and router R2 since all of them are attached to the network 123.0. Therefore, no entry in the routing table is necessary. If host A wants to send data to host C, which is located in the network 122.0, it has to send the data to router R2. The routing table needs the following entry: All hosts in the network 122.0 are reached via the router 123.4, that is router R2. If host A wants to communicate with the Internet, the data has to be sent via router R1. The standard router entry therefore is 123.1. It is responsible for all destinations outside the own network except of the network 122.0. For host B the same holds as for host A.

Host C can only communicate with the other hosts or the Internet via router R2. Therefore, its routing table entry is that router 122.1 is the standard router.

The router R1 is directly connected to the Internet so that a router of the ISP is its standard router. It can directly reach host A, host B and router R2 via the network 123.0. To be able to communicate with host C there is the following entry in its routing table: The router for it is the router 123.4, that is R2.

Router R2 can directly communicate with the hosts A, B, C and with router R1. To get to the Internet it has to send data via R1. Therefore, its routing table says that the standard router is R1. Now all devices can communicate with each other and with the Internet.

Now we take a look at an example communication. If e.g. host C wants to send a datagram to the Internet, e.g. to host Z, the following steps are taken. It compares its network ID with the destination's network ID. They are not equal. Therefore, an entry for the destination address is looked up in the routing table, but is not found. Afterwards, the entry for the standard router is used. The datagram is sent to 122.1, that is, router R2. R2 compares the network IDs. Since they are not equal, a search for an entry for the destination address is performed. Since there is no entry, the datagram is sent to the standard router R1 with the IP address 123.1. Host C has no knowledge how router R1 forwards the datagram.

End printversion

### 4.3.5 Subnets

We have seen that a router connects different networks. In the examples, there were only a few hosts present in the network, but up to 254 hosts can be connected in a class C network. The unused host addresses can obviously also not be used at another location, e.g. in another company, because (in the second example) all datagrams for the networks 192.175.123.0 and 192.175.122.0 must be sent to router R1 and not to the router of the other company. The fact that this would be a huge waste of host addresses is clear, and

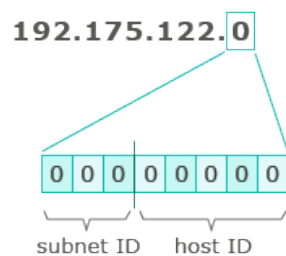
this was also one of the reasons that led to the introduction of subnets. With the help of subnets, the size of the networks to be routed can be adapted better because the subnets can also be routed.



practice

Better adaptation to different network structures is possible by using subnets.

The host ID of an Internet address is split into a **subnet ID** and a (truncated) host ID. Let us take the class C address 192.175.122.0 as an example. The host ID consists of the last byte (8 bits). We now define, for example, the first 3 bits of the host ID as a subnet ID, then 5 bits remain for the host ID.



#### Distribution of the host ID into subnet ID and host ID

##### The subnetting principle

From this we can construct  $2^3 = 8$  subnets with  $2^5 - 2 = 30$  hosts each. Together we get 240 possible addresses. If we compare this with the 254 possible addresses in a class C network without subnets, we see that some addresses are lost – but that is still more economical than without using subnets.

A **subnet mask** is used to define which parts of the entire address should be interpreted by the router as network ID or subnet ID. The subnet mask is 32 bits long, as with the Internet address. All set bits define the length of the network ID/ subnet ID.

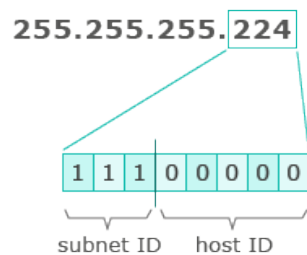


example

A typical subnet mask is, for example, 255.255.255.224. In the binary representation, this becomes:

11111111.11111111.11111111.11100000

In the example above, the first 24 bits define the network ID – i.e. the first 24 bits are set in the subnet mask. Of the last 8 bits, the first 3 should be used as the subnet ID – so in the subnet mask, these 3 bits must also be set. Thus  $24 + 3 \text{ bits} = 27 \text{ bits}$  are used for the network ID/ subnet ID.



#### Subnet mask

The router now compares the first 27 bits of the IP address of a packet that is to be routed with the first 27 bits of the receiver address from the routing table. If both addresses are the same in the first 27 bits, the packet is forwarded to the appropriate router from the routing table.



#### example

#### Routing example with subnets

The following two animations show

1. The network with the two subnets
2. The routing table of the host and the router in the depicted networks
3. The routing of a datagram from host D to host C
4. The routing of a datagram from host B to host C

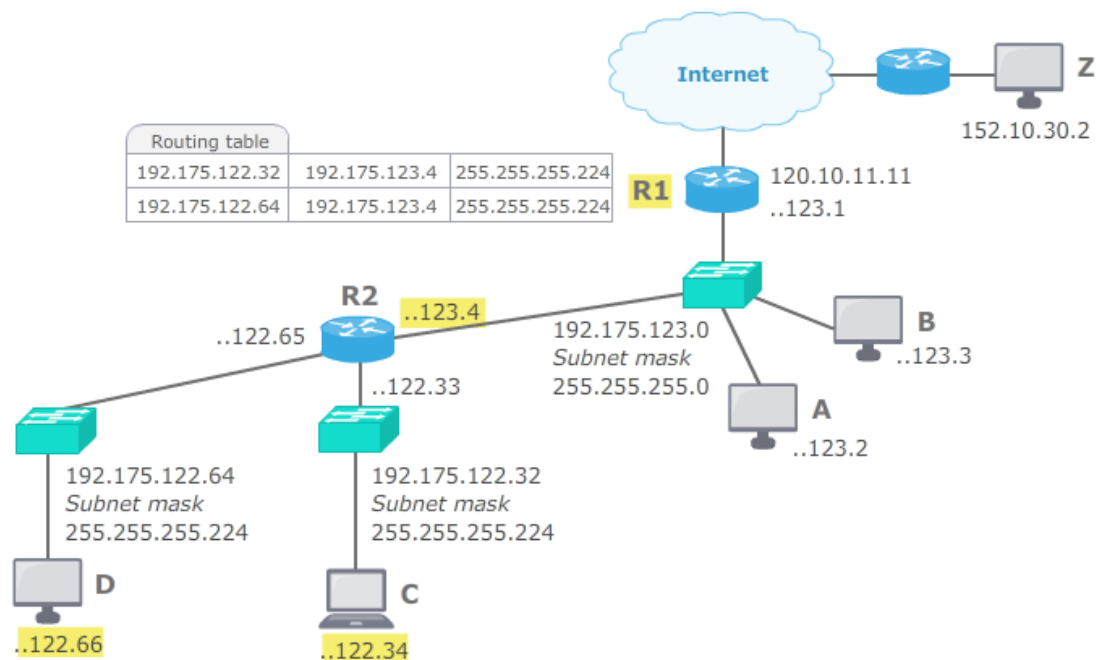
The animations contain a large number of IP addresses, all of which begin with 192.175... To facilitate reading and listening, the first two bytes are left out of the figures and the audio and only the last two bytes are considered.



In the online version an animation is shown here.

#### Network with two subnets

Begin printversion



Two subnets are attached to the network 192.175.123.0 via router R2. The network mask for the subnets, where we find the hosts C and D, is 255.255.255.224. This means that the first three bits in the last byte are the subnet ID. The network address of the first subnet is 122.32. The last byte looks as follows in the binary representation. The network address of the second subnet is 122.64. The last byte looks as follows in the binary representation.

Which entries are required for the routing tables so that all hosts can communicate with each other and also with the Internet?

Let us start again with host A. Host A can directly reach host B, router R1 and router R2. It is necessary to note in the routing table that router R2 is used to send packets to the hosts in the subnets of 122.0. This is possible via the interface of R2 with the IP address 123.4. The subnet mask is provided as 255.255.255.224. It is therefore necessary to compare the first 27 bits. To get to the Internet, the packets have to be sent to 123.1, that is router R1. This is the standard router for host A. Host B gets the same routing table entries.

Host C needs the following entry in its routing table so that it can communicate with other computers: The standard router is router R2 which is reachable via the interface with the IP address 122.33. The subnet mask is not considered for a standard router. All packets, which cannot be delivered directly in the attached network, are sent to the standard router independent from the subnet mask. Therefore, the subnet mask does not have to be provided.

The standard router for host D is also the router R2. However, it is reached from host D via the interface 122.65.

Router R1 can directly reach host A, B and router R2. To be able to communicate with the host C and D, it has to send packets to the router R2. The routing table entries therefore are: All packets for hosts in the networks 122.32 and 122.64 are provided to router R2 which can be reached from R1 via the interface 123.4. The subnet mask determines that on data transmission the first 27 bits are checked whether they are the same. The router R2 can directly communicate with the hosts A, B, C, and D as well as router 1. If it wants to send data to the Internet, it has to send them to router R1. This is the standard router in the routing table.

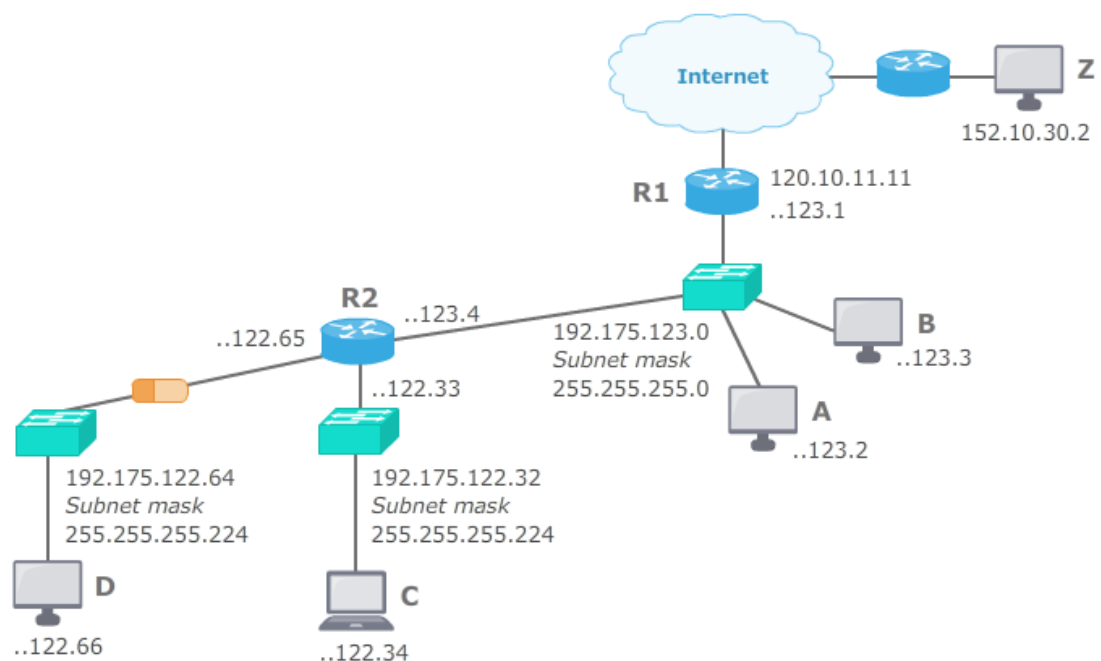
End printversion



In the online version an animation is shown here.

#### Two routing examples

Begin printversion



In the network, which has been presented in the previous video, we want to consider two cases. In the first case, host D in the subnet 122.64 wants to send data to host C in the subnet 122.32. Host D knows its subnet mask 255.255.255.224 and compares its IP address with the one of host C on the basis of this length value. It notices that the subnet IDs are not the same. Therefore, it takes a look at the routing table if there is an

entry for the subnet 122.32. However, there is only an entry for a standard router. The datagram is sent to the known standard router R2. R2 determines that the destination is in an attached network and can directly provide the datagram.

In the second case, host B wants to send a datagram to host C. At first, B notices that the datagram has to be forwarded to another network. B uses the first 27 bits for its routing table lookup and determines the router which can be used to reach the network 122.32. The datagram is sent to the interface of R2 with the IP address 123.4. R2 can then directly provide the datagram to host C.

End printversion



important

Two important points can be learned from the examples.

- In the routing table, a new subnet mask needs to be defined for every entry (except for the default router).
- The external routers in the Internet have no information about whether subnets are present in the network structure behind router R1. This inner structure of the company network remains hidden to external routers.



summary

Let us sum up the most important points again:

1. The subnet mask specifies how many bits of the destination IP address are interpreted by the router.
2. The number of the addressable hosts in a subnet depends on the length of the subnet mask.

### 4.3.6 CIDR - Classless Inter-Domain Routing


As we have seen, only the host part of the Internet address is used for the creation of subnets and subnet masks. Beginning in 1993, this limitation was dropped and masks of any length could be defined. Since then, there is therefore a classless technique for the creation of masks, which shows how many bits of an IP address should be interpreted by the router. This technique is called CIDR ("Classless Inter-Domain Routing", RFC 4632). CIDR is just a generalization of the use of subnet masks.

The properties of subnets can be transferred unchanged to CIDR.

1. The masks specify how many bits of the destination IP address are interpreted by the router.
2. The number of the addressable hosts depends on the length of the mask.
3. The available host addresses are also dependent on the mask and cannot be freely selected.



In the online version an video is shown here.

Link to video : <http://www.youtube.com/embed/KHg3j4dRosY> 

**Classless Inter-Domain Routing**

The practical significance of CIDR is illustrated in the following example:



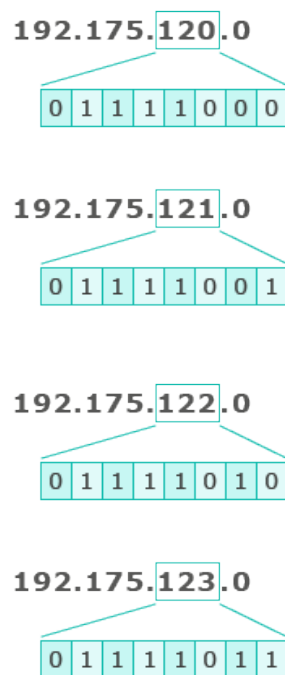
**example**

#### **A company needs approximately 1,000 IP addresses**

As we have seen, IP addresses were assigned on the basis of classes: Class A, B or C. Suppose that a company needs about 1,000 IP addresses. It does not make sense to assign this company a class B network because in a class B network more than 65,000 hosts can be addressed. Therefore several class C networks are assigned to the company. Four class C networks are needed here. This means that  $4 \cdot 256 = 1024$  IP addresses can be used. The four class C networks 192.175.120.0, ...121.0, ...122.0 and ...123.0 are assigned to the company. To reach the company via the Internet, four entries need to be present on an upstream router – one entry for each network.

However, with CIDR all four networks can be summarized in a single entry in a routing table on a router. So, the huge growth of routing tables can largely be avoided.

Let us look now at the third byte of the above network addresses more closely to examine the entry that is now necessary.



The third byte of the above network addresses

We need to tell the upstream router that all IP addresses where the first 6 bits within the third byte are identical should be sent to the company router (the first and second byte must of course also be identical). Here we have exactly the same problem as in the definition of a subnet mask. We need to tell the router how many bits of an IP address should be considered in the routing. In this case, the first 22 bits need to be compared. In the routing table for the external router, the following entry must therefore be present:

192.175.120.0	R1	255.255.252.0
---------------	----	---------------



Entry in the routing table of an external router (with subnet mask)

All packets for IP addresses that match to the IP address 192.175.120.0 in the first 22 bits should be sent to router R1. With this single entry in the routing table, all four class C networks with over 1000 possible hosts can be reached.

The routing decision is no longer made on the basis of addresses classes but on the basis of a mask. Instead of specifying all the bits in a mask, often the number of bits that need to be considered in the routing decision are specified. This number is also called a **prefix**.

The above entry in the routing table is therefore:



192.175.120.0/22

R1



Entry in the routing table of an external router (with prefix)

The length of the **prefix** (the mask) also determines the number of IP addresses. The following table shows which prefixes must be used for a certain number of desired IP addresses.



Prefix for desired IP addresses

Begin printversion

Addresses	Prefix	Mask	Number of Class-C-Nets
1	/32	255.255.255.255	
2	/31	255.255.255.254	
4	/30	255.255.255.252	
8	/29	255.255.255.248	
16	/28	255.255.255.240	
32	/27	255.255.255.224	
64	/26	255.255.255.192	
128	/25	255.255.255.128	
256	/24	255.255.255.0	1
512	/23	255.255.254.0	2
1024	/22	255.255.252.0	4
2048	/21	255.255.248.0	8
4096	/20	255.255.240.0	16
8192	/19	255.255.224.0	32

End printversion



Prefix /32 =  
single host

The **Prefix /32** is used when a single host should be addressed.



Präfix /31 = PPP  
connections

With **prefix /31**, there is one network with only two addresses: 0 and 1, which actually should not be used. In this case, however, an exception is made because there are many networks that consist of only two hosts. Where point-to-point connections are used, obviously only two hosts are used. Because private Internet users are usually connected via PPP (point-to-point protocol), there are a lot of these networks. Previously networks with prefix /30 had to be used here, i.e. only two of four IP addresses were used. In this case, prefix /31 may be used as an exception. For a provider with, for example, 500 PPP connections, this can save 1000 IP addresses that can be used elsewhere.



example

The following animations show

1. The routing table of the host and the router in the depicted networks
2. The routing of a datagram from the Internet to host C
3. The routing of a datagram from the Internet to host E

The summary of the entries in the routing table requires that routes are selected according to the **longest prefix matching rule**: A router must select the entry with the largest prefix when the same network is present in the routing table with prefixes of different lengths. The following animations illustrate this principle.

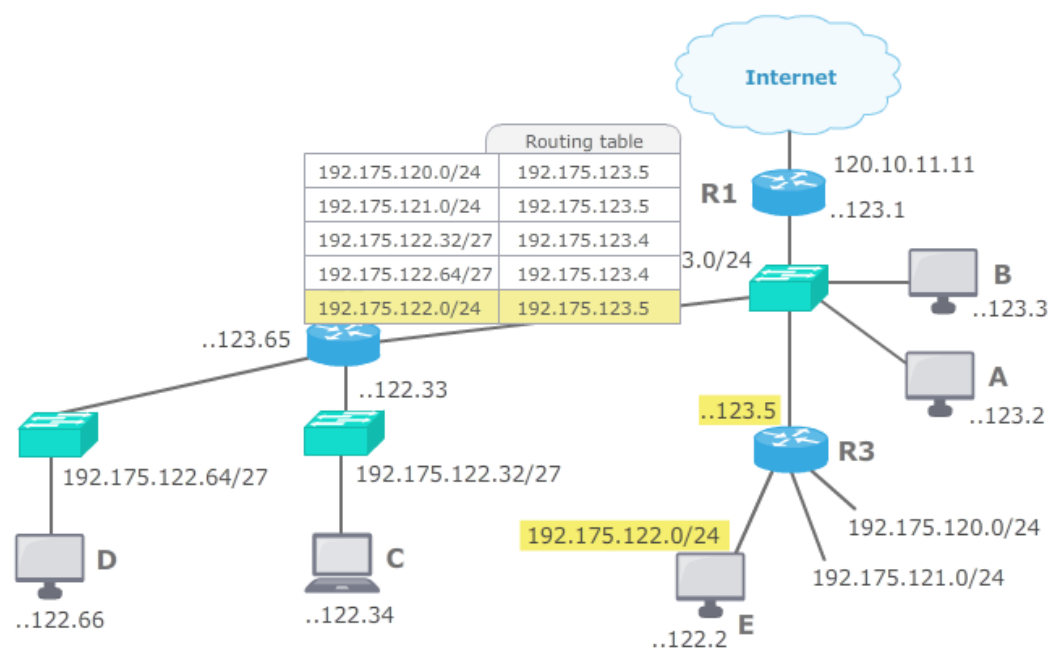
The animations contain a large number of IP addresses, all of which begin with 192.175... To facilitate reading and listening, the first two bytes are left out of the figures and the audio and only the last two bytes are considered.



In the online version an animation is shown here.

**Routing table of host and router in the depicted networks**

Begin printversion



Four slash 24 networks have been assigned to a company, that is, the networks 192.175.120.0 until 123.0. Router R1 is connected to the Internet. Two subnets are attached to router R2: 122.32 and 122.64. The networks 120.0 and 121.0 are connected via router R3 as well as all subnets in the network 122.0 which are not attached to R2. The routing table of routers in the Internet, which send datagrams directly to the ingress router R1, have to contain the following entry: 192.175.120.0 with prefix 22 is reachable via router R1. At this point, you see the aggregation of the four slash 24 networks to one slash 22 network.

How do the entries in the routing table on R1 look like? The network 120.0 is reachable via router R3. The entry in the routing table therefore is 192.175.120.0 combined with prefix 24 for the network ID.

The IP address of router R3 is 192.175.123.5. The network 121.0 is also reachable via R3. The subnets 122.32 and 122.64 are accessible via R2. The prefix has 27 bits. The remaining hosts in the network 122.0 are reachable via R3. The prefix for the network ID has 24 bits in this case.

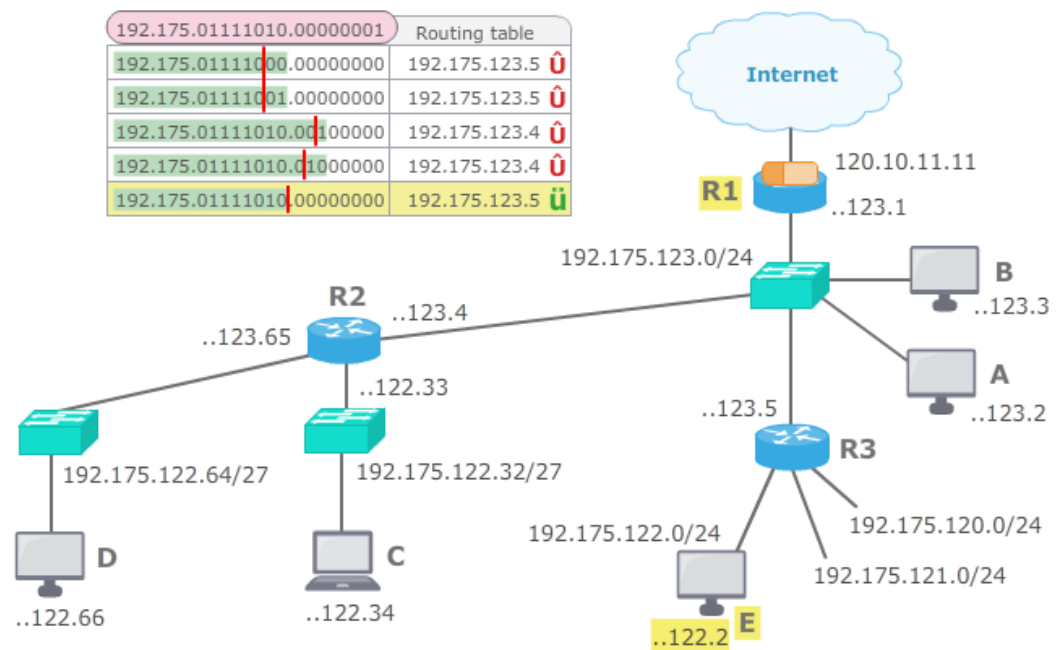
End printversion



In the online version an animation is shown here.

**Routing examples with CIDR**

Begin printversion



Now we take a look at two examples how datagrams reach their destination in the network configuration from the previous video.

The first example is a datagram for host C, that is, for the IP address 122.34, which has arrived at R1. R1 now has to find the router which has to be used to forward the datagram. For doing so, a bitwise comparison of the routing table entries in relation to the destination address is performed. To make it easier for you to understand how each bit is compared, we have chosen a binary representation for the third and fourth byte of the addresses. As has been mentioned before, the prefix shows how many bits have to be compared. In this example, 24 and 27 bits are regarded, respectively. Now the addresses are checked bitwise for matching.

The first entry does not match to the destination address of the IP packet for the required length, which is given by the prefix. Now the second entry is checked. This entry does also not match to the destination address for the required length. For the third entry, the first 27 bits are identical. The matching for the fourth entry is shorter than the requirement given by the prefix. For the fifth entry, the first 24 bits are identical. Two entries provide matchings. The result is therefore ambiguous.

In such a case, the result with the highest number of matching bits is used. This is the so-called longest prefix matching rule. It is the third entry in the table. The datagram is sent to router R2 with the IP address 192.175.123.4 and is provided by the router to the destination host C.

Let us take a look at another case. A datagram with the destination host E and the destination IP address 192.175.122.2 has arrived at router R1. Similar to the previous case, the datagram's IP address is compared to entries in the routing table. For easier observation of the bitwise address matching, we choose again a binary representation of the last two bytes. The check provides the following result. Only the fifth entry matches to the destination address for the required length. The result is unambiguous. The datagram is sent to router R3 with the IP address 192.175.123.5 and is provided to the destination host E via the router.

End printversion

We see that the technique of forming subnets has evolved into the more general CIDR technique. Instead of referring to a subnet mask, today people refer to a mask or prefix.



notice

You can find a “CIDR calculator” at [subnet-calculator.de](https://subnet-calculator.de) and elsewhere on the Internet. This facilitates the determination of CIDR address ranges.



summary

Let us sum up the main characteristics of route selection that is undertaken by every host or router in the Internet:

IP routing supports route selection with masks or prefixes.

- The length of the masks or prefixes is arbitrary.
- Multiple entries with different masks are possible for the same destination.
- A router selects from the routing table the entry with the longest match between the desired destination address and the entries in the routing table (**longest prefix matching rule**).
- Entries can be summed up in the routing table with CIDR. Nevertheless, routing tables can still be very large.

### 4.3.7 Fragmentation

An IP packet is divided into **fragments** if the IP packet is larger than the MTU on the Data Link Layer permits. The MTU is the maximum size of the payload from the perspective of the Data Link Layer (see [Ethernet Frames](#)).



In the online version an video is shown here.

Link to video : <http://www.youtube.com/embed/UGV7ioLaZ94>

#### Fragmentation

All fragments of an IP packet contain the same **identification** to ensure that fragments from different IP packets are not mixed up. The “more fragments” (**MF**) bit is set in the flag field for all fragments except the last. The **fragment offset** indicates the position of the payload in the original IP packet. The offset of the first fragment is 0. The payload of the IP packet are copied into the fragments using units of 8 bytes.



example

The following example shows a datagram with 2000 bytes of data divided into two fragments.

#### First fragment:

Internet protocol

Header length: 20 bytes

Total length: 1500 bytes

Identification: 0x0e71

Flags: 0x02

.0.. = Don't fragment: Not set

..1. = More fragments: Set

Fragment offset: 0

Protocol: UDP

User datagram protocol, Src port: 1029, Dst port: 7

Data (1472 bytes)

#### Second fragment:

Internet protocol

Header length: 20 bytes

Total length: 548 bytes

Identification: 0x0e71

Flags: 0x00

.0.. = Don't fragment: Not set

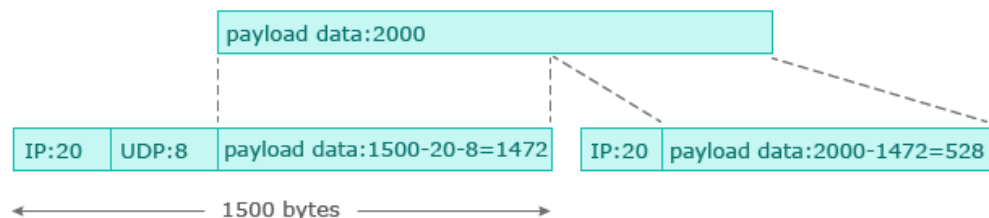
..0. = More fragments: Not set

Fragment offset: 1480

Protocol: UDP

Data (528 bytes)

The figure shows how the data transferred by the IP packet are distributed across both fragments.



#### Distribution of data into two fragments

The UDP header is present only in the first fragment. The total length of the first fragment is 1500 bytes (regarded from IP's point of view). The offset of the second fragment is 8 bytes (UDP header) + 1472 bytes (data) = 1480. The MF bit is not set in the second fragment because it is the final fragment. The identification is the same in both fragments.

If the “Don’t Fragment” (DF) bit is set in a packet, it must not be fragmented. If the datagram actually needs to be fragmented, it will be dropped and an ICMP error message will be generated (“packet too big”) and returned to the source.

A router should minimize the number of fragments and transmit them in the correct order. Because all fragments have their own IP header, they can be delivered along different routes to the receiver. Therefore, they can only be reassembled after reaching the receiver.



**important**

If a fragment is lost, it is not possible to send only the lost fragment again. Instead, all fragments must be sent again.

Overall, one tries to avoid fragmentation because it is relatively inefficient. Therefore, in practice relatively few packets are fragmented.

### 4.3.8 Path MTU

To avoid or minimize fragmentation it is useful to know the smallest MTU, the **path MTU**, between the sender and receiver. Then the sender can fragment the packet so that it no longer needs to be fragmented in the routers through which it is sent.

The following animation shows:

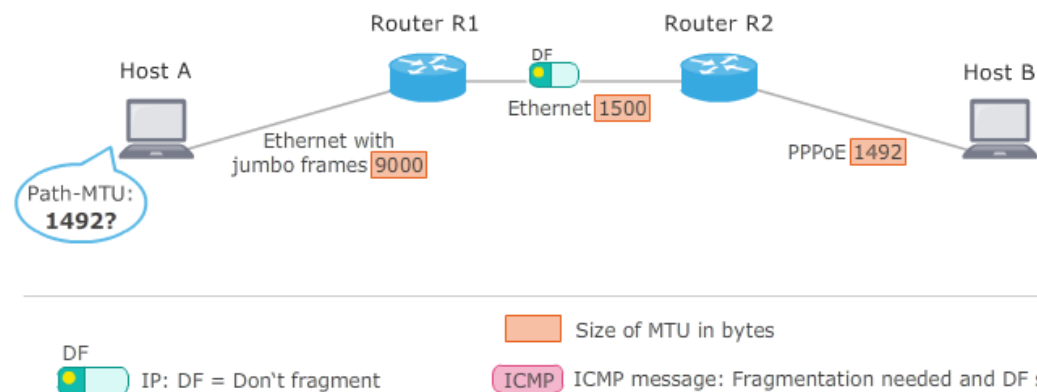
1. The sending of data from host A to host B with fragmentation
2. The sending of data from host A to host B with “**path MTU discovery**”



In the online version an animation is shown here.

#### Path MTU Discovery

Begin printversion



Host A wants to send data to host B. At first we see what happens if the path MTU, that is the smallest MTU on the way from A to B, is unknown to the sender. The Ethernet network to which host A is attached has an MTU of 9000 bytes because jumbo frames are allowed. Host A sends a data packet with 9000 bytes payload data to router R1.

The MTU of the Ethernet link between R1 and R2 is smaller than the size of the data packet because it is 1500 bytes. Therefore, the packet is fragmented in router R1. The PPPoE connection of R2 to host B has an MTU of 1492 bytes. The fragmented packets are again fragmented in router R2, prior to being forwarded to the destination, host B.



In the next example the procedure to determine the smallest MTU is used, the so-called “path MTU discovery”. The sender sets the “don’t fragment bit” in the IP Header, abbreviated as DF. Host A sends again a packet with 9000 bytes via the Ethernet connection to router R1. The router would be required to fragment the packet because the MTU on the Ethernet link to router R2 is just 1500 bytes. Since the DF bit is set, the packet is dropped by the router. The router generates an ICMP message “Fragmentation needed and DF set” and provides this message back to the source. In the message, it indicates the MTU of the Ethernet link as 1500 bytes. As a consequence, the source reduces the packet length to 1500 bytes, sets again the DF bit and sends the packet to router R1. R1 forwards the packet to R2.

Since the MTU of the PPPoE connection is smaller than 1500 bytes, R2 would be required to fragment the packet. Because the DF bit is set, the router drops the packet and sends an ICMP message “Fragmentation needed and DF set” back to the source. It indicates the MTU size of the PPPoE connection as 1492 bytes.

In the following, host A sends a packet with a length of 1492 bytes. The packet is now small enough to reach the destination without the need to be fragmented. Since the source does not receive further ICMP messages, it has determined the path MTU of the connection to B and sends the remaining data to host B in packets of this size.

End printversion

This technique sounds simple. But in reality different problems can occur to such an extent that an existing connection is terminated (see RFC 2923, TCP problems with path MTU discovery).

## 4.4 ICMP - Internet Control Message Protocol



arrangement

### 4.4 ICMP - Internet Control Message Protocol

#### 4.4.1 ICMP Error Messages


#### 4.4.2 Ping

#### 4.4.3 Traceroute

**ICMP (Internet control message protocol)** (RFC 792) is an auxiliary protocol that provides diagnostic and error handling functionality in addition to IP. ICMP is handled like a Transport Layer protocol (the protocol field in IP header has the value 1) although it is an integral part of IP. Every device that uses IP must also implement ICMP.



In the online version an video is shown here.

Link to video : <http://www.youtube.com/embed/TB95SIROP7k> 

**Internet Control Message Protocol**

ICMP messages can be divided into two groups:

**Error messages**, such as when

- the receiver cannot be reached,
- the requested port number does not exist at the receiver,
- a router detects that it is better to send a datagram to another router, or
- a datagram has exceeded the maximum number of redirects,

**Requests** and associated responses, e.g. for

- an echo,
- an address mask, or
- a time stamp.

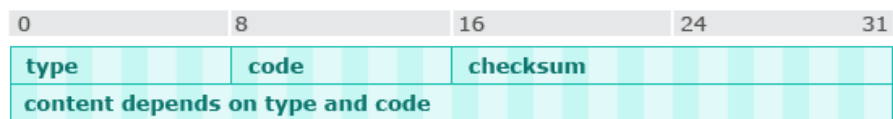
An ICMP message should return as much data as possible from the received IP packet that caused the error or response - up to a length of 576 bytes. This allows the packet sender to best recognize the situation that led to the ICMP message. The port numbers in particular must be sent back when using TCP or UDP so the process that triggered the ICMP message can be notified.

ICMP messages are not sent,

- if an ICMP message is received,
- if errors have occurred when checking the IP header checksum,
- if broadcast or multicast packets have been received, or
- for fragments except the first fragment.

### 4.4.1 ICMP Error Messages

The following figure shows the general structure of the ICMP header.



#### ""Structure of the ICMP header""

The most important fields in the ICMP header are type field and code field. The structure and content of the message depends on these two fields.




#### ""Some ICMP messages""

Begin printversion

Type	Code	Meaning
0	0	Echo Reply
3	0	Network Unreachable
3	1	Host Unreachable
3	2	Protocol Unreachable
3	3	Port Unreachable
3	4	Fragmentation Needed and Don't Fragment was Set
3	5	Source Route Failed
3	6	Destination Network Unknown
3	7	Destination Host Unknown
3	8	Source Host Isolated
3	9	Communication with Destination Network is Administratively Prohibited
3	10	Communication with Destination Host is Administratively Prohibited
3	11	Destination Network Unreachable for Type of Service
3	12	Destination Host Unreachable for Type of Service
3	13	Communication Administratively Prohibited
3	14	Host Precedence Violation
3	15	Precedence cutoff in effect
4	0	Source Quench (no longer valid, see <a href="#">RFC 6633</a> )
5	0	Redirect Datagram for the Network (or subnet)
5	1	Redirect Datagram for the Host
5	2	Redirect Datagram for the Type of Service and Network
5	3	Redirect Datagram for the Type of Service and Host
6	0	Alternate Address for Host
8	0	Echo Request
9	0	Normal Router Advertisement
10	0	Router Selection
11	0	Time to Live exceeded in Transit
11	1	Fragment Reassembly Time Exceeded
12	0	Parameter Problem: Pointer indicates the Error
12	1	Parameter Problem: Missing a Required Option
12	2	Parameter Problem: Bad Length
13	0	Timestamp Request
14	0	Timestamp Reply
17	0	Address Mask Request
18	0	Address Mask Reply
30	0	Traceroute (is in general not used by the traceroute tool)
31	0	Datagram Conversion Error
32	0	Mobile Host Redirect

---

End printversion

The complete list of ICMP messages can be found at [IANA](#) .

Some selected messages are explained in the following.

- **Destination unreachable:** When a router cannot forward an IP packet further because it has no corresponding route and no default router is specified, it must generate this message.
- **Redirect:** This message informs a host that another route is more suitable for forwarding the IP packet.
- **Time exceeded:** When the value of the TTL field reaches 0 in a router, the IP packet cannot be forwarded further. An ICMP error message must be generated in addition.
- **Parameter problem:** A router must generate this message when an error occurs that cannot be described by any other ICMP message. An indicator on the parameter that generated this message is also sent.
- **Echo request/reply:** These messages are implemented on both hosts and routers. Responses to broadcast or multicast requests do not need to be answered. Management systems can send broadcast echo requests in order to determine with a single command which hosts and routers are active in the network.
- **Address mask request/reply:** A router must return the requested address mask in the response.
- **Router advertisement/selection:** Both of these messages are used to find routers in networks.

### 4.4.2 Ping

Ping is a command-line tool that is available for popular operating systems such as Windows and Linux. Ping allows to determine whether a host is reachable. Ping uses ICMP messages for this purpose: An “**echo request**” message (“Hello host! Are you there?”) is sent. An “**echo reply**” message is expected (“I’m here!”). Ping also provides the round trip time of the ICMP messages (i.e. the time to host and back) and packet loss based on the test packets.

Network management systems use ping to determine which computers are active in the network. To do this, an “echo request” must be sent to each computer. If a network broadcast address (host part of the address consists only of ones) is specified, only a single “echo request” packet is generated after which all active computers return an “echo reply” packet.

The example shows a trace of this behavior.



example

No. Source		Destination	Protocol Info	
1	192.168.0.35	192.168.0.255	ICMP	Echo (ping) request
2	192.168.0.188	192.168.0.35	ICMP	Echo (ping) reply
3	192.168.0.40	192.168.0.35	ICMP	Echo (ping) reply
4	192.168.0.189	192.168.0.35	ICMP	Echo (ping) reply
5	192.168.0.52	192.168.0.35	ICMP	Echo (ping) reply
6	192.168.0.50	192.168.0.35	ICMP	Echo (ping) reply
7	192.168.0.5	192.168.0.35	ICMP	Echo (ping) reply
8	192.168.0.9	192.168.0.35	ICMP	Echo (ping) reply
9	192.168.0.3	192.168.0.35	ICMP	Echo (ping) reply



Trace: Pinging a broadcast address

With pinging, potential attackers from the Internet can find out which hosts are active in a network. For this reason, firewalls are often configured so that they reject ICMP packets with “echo request.”

### 4.4.3 Traceroute

Traceroute, which is called `tracert` in Windows, is also a standard command line tool that is available by default in current operating systems. The first implementation was by Van Jacobson in 1987. Traceroute is used to determine the route by which a packet is transmitted from the sender to the receiver.

Traceroute exploits the fact that a router reduces the **TTL value in the IP header** by one. In the first packet that traceroute sends, the TTL is set to 1. The first router sets TTL to 0 and discards the packet. It also generates an ICMP message ("TTL expired in transit"), enters its IP address as sender and sends the ICMP message back to the sender. This allows the sender to detect the IP address of the first router. Such a test is usually repeated two times and the round trip times are recorded. In the next packet, TTL is then set to 2, which results in the second router returning a "TTL expired" ICMP message, and so on. This process is repeated until the receiver is reached.

Traceroute sends UDP packets with an invalid port number (usually port 33434). Because no UDP service with this port number is active at the receiver, a "destination unreachable" ICMP message is generated and returned to the sender. This tells traceroute that the receiver was reached and no further packets have to be sent. Alternatively, this can also be implemented in a different way by simply sending echo request packets, as with pinging, and waiting to receive an echo reply message when the destination is reached.

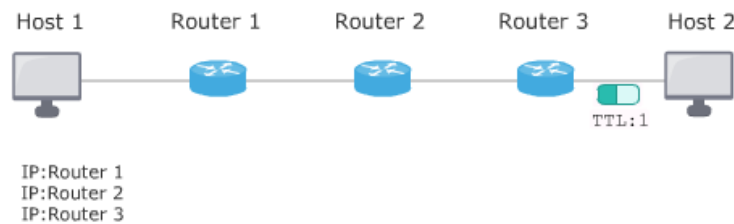
This does not mean that successive packets always use the same route. In fact, a route selection is made for each individual packet. But it can be assumed that in the short time that traceroute is active generally no route change takes place – but this is not certain! You also have to consider that the time is influenced by the current load situation in the network.



In the online version an animation is shown here.

**Traceroute**  
**Traceroute animation**

Begin printversion



In the first packet, which is sent by traceroute, TTL is set to 1. The first router sets TTL to 0, drops the packet and generates an ICMP time exceeded message. In the message it puts its own IP address as source address and provides the ICMP message back to the source. Therefore, the source gets to know the IP address of the first router. In the next packet TTL is set to 2. As a consequence, the second router sends an ICMP time exceeded message back. And so on.

This process is continued until the destination is reached. Traceroute sends UDP packets with an invalid port number. The standard port is 33434. Since there is no UDP service on this port number on the destination, an ICMP destination unreachable message is generated and is sent back to the source. Due to this reason traceroute learns that the destination has been reached and that it is not necessary to send additional test packets. End printversion



annotation

If you look at the possible options for the IP header, you will find the option called “record route.” Every router that receives a packet with this option enters its IP address in the header both on the way to the receiver and on the way back. Unfortunately the “option” in the IP header is not long enough to show the entire route that, for example, a packet has taken from Germany to Australia. The maximum length of the option is 40 bytes. This means a maximum of 10 IP addresses with 4 bytes and overhead can be stored.

## 4.5 ARP - Address Resolution Protocol



arrangement

### 4.5 ARP - Address Resolution Protocol

#### 4.5.1 ARP Table



### 4.5.2 Gratuitous ARP

As we have already seen, two situations arise when an IP packet is to be sent to a receiver. Either the receiver is located in a different network so the IP packet needs to be sent to a router for forwarding, or the receiver is in the same network. In the latter case, no router is needed, but usually only the IP address of the receiver is known. However, because LANs are usually built on the basis of switches, this information is not sufficient since switches can only make decisions about forwarding on the basis of MAC addresses.

**ARP (Address Resolution Protocol)** is therefore used to find the MAC address that belongs to the given IP address (the protocol also has more general options for other types of addresses and technologies, which are not considered here). These assignments are stored in the **ARP table**, which is also called the ARP cache. ARP is designed for physical broadcast media such as Ethernet.



In the online version an video is shown here.

Link to video : <http://www.youtube.com/embed/aKOLfYBXBGg>

**Address Resolution Protocol**

#### Basic process

The MAC address corresponding to the IP address is first searched for in one's own ARP table. If the mapping is known, the Ethernet header can be built, and the packet can be sent. If the mapping is unknown, an ARP request is sent as an Ethernet frame to all hosts and routers in one's own network. The IP address being searched for is recorded in the ARP request. The addressed host or router then sends its MAC address in an ARP reply. Then the mapping of MAC address and IP address is stored in the ARP table. Afterwards the Ethernet header can be created, and the packet can be sent.



example

#### ARP Example

Both packets - ARP request and ARP reply - can be seen in the following trace.

##### ARP request:

<span style="font-family: Courier New"> Ethernet

Destination: ff:ff:ff:ff:ff:ff

Source: 00:50:da:63:9c:16

Type: ARP (0x0806)

<span style="color:#ff0000;"> Address resolution protocol

Hardware type: Ethernet (0x0001)

Protocol type: IP (0x0800)

Hardware size: 6

Protocol size: 4

Opcode: request (0x0001)

Sender MAC address: 00:50:da:63:9c:16

Sender IP address: 193.175.122.94

Target MAC address: 00:00:00:00:00:00

Target IP address: 193.175.122.2

The ARP request (Opcode: request) is sent to the Ethernet broadcast address so that switches will forward to every device in the network. Routers in contrast do not forward an Ethernet broadcast. An identifier for ARP is put into the Ethernet type field. The known IP address (target IP address) is contained in the ARP header. The MAC address (target MAC address) in contrast is not available because it is supposed to be identified. The sender also includes its own IP address and MAC address so that another ARP request is not necessary for a later data frame from the receiver to the sender.

#### **ARP reply:**

<span style="font-family: Courier New"> Ethernet

Destination: 00:50:da:63:9c:16

Source: 00:50:e2:6e:30:1c

Type: ARP (0x0806)

<span style="color:#ff0000;"> Address resolution protocol

Hardware type: Ethernet (0x0001)

Protocol type: IP (0x0800)

Hardware size: 6

Protocol size: 4

Opcode: reply (0x0002)

Sender MAC address: 00:50:e2:6e:30:1c

Sender IP address: 193.175.122.2

Target MAC address: 00:50:da:63:9c:16

Target IP address: 193.175.122.94

The ARP reply (Opcode: reply), which is structured identically in terms of fields, is selectively sent to the requesting MAC address. The desired MAC address is recorded.

The format of an ARP message depends on the format of the hardware address and the network address. The addresses for Ethernet (hardware type 1) are 6 bytes long (hardware size). The IP addresses (protocol type 0x0800) are 4 bytes long (protocol size).

In addition to the operation codes (request/reply) described here, other functions are also defined such as reverse ARP and inverse ARP. A complete overview of [all types of hardware and operating codes](#) [↗](#) can be found at IANA.

### 4.5.1 ARP Table

If an ARP request and reply are exchanged, both sides record the mapping of the MAC address to the IP address in their **ARP tables** (also called ARP cache). These assignments are only retained for a few minutes. Then they are deleted. This allows to avoid false entries if the mappings are changed. A end system can actually be reconfigured so that it uses a different IP address. In addition, the ARP tables should also not become too large.

It is possible (in exceptional cases) to include static assignments in the ARP table. These are kept permanently in the ARP table.



example

#### The program arp

The ARP table can be viewed with the program arp (DOS window in Windows) and changed

**arp -a** displays all entries

**arp -d IP\_ADR** removes an entry

**arp -s IP\_ADR Ethernet\_ADR** sets a static entry

### 4.5.2 Gratuitous ARP

An ARP request is normally only generated if an IP packet is to be sent. With **gratuitous ARP**, an ARP request is sent without the presence of an IP packet. Its own IP address is also given as the destination in such a request.



annotation

The following animation shows

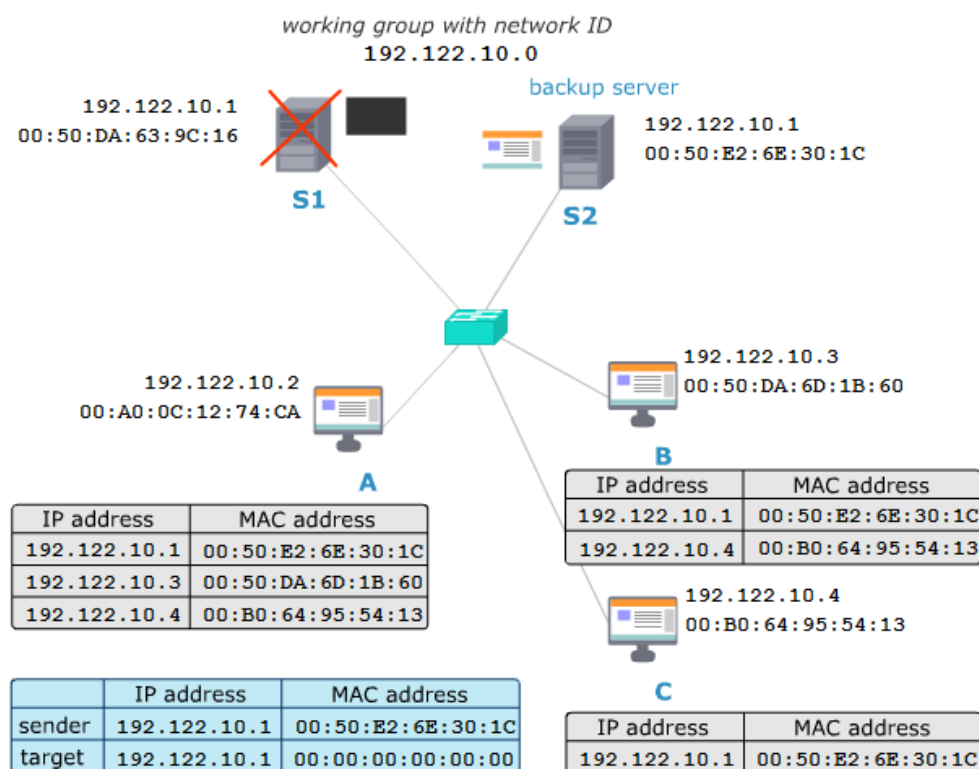
1. the normal flow of an ARP request,
2. the normal flow of a gratuitous ARP request,
3. the process of a gratuitous ARP request, if its own IP address is already used in the network,
4. and finally the possibility of making two servers fail-safe with gratuitous ARP.



In the online version an animation is shown here.

### ARP

Begin printversion



We find here the network of a working group which has the network ID 192.122.10.0. The server S1 is part of the network. Server S2 is its backup server. S2 is currently not active. Both servers have the same IP address, that is, 192.122.10.1, but they have different MAC addresses. The hosts A and B are connected to the network as well. Only host A is switched on at the moment. The figure shows the IP addresses and MAC addresses of both hosts.

Host A now wants to communicate with server S1. In reality this could for instance be the case if an FTP program has been started on host A. It may want to retrieve files from FTP server S1. For doing so, host A sends an ARP request to all computers in the network. S1's IP address is indicated as target. Host A's IP address and MAC address are provided as source. Server S1 inserts the addresses of host A into its ARP table. Then it sends an ARP reply with its own MAC address to host A. Host A inserts the address of S1 into its ARP table.

An employee who has access to host B has arrived and boots the computer. During this process a gratuitous ARP is sent to all hosts. In a gratuitous ARP the own IP address is set as target. The target MAC address is set to 0. The connected computers now know the MAC address of B and insert it into their ARP tables. If host B now establishes a connection to S1, for example a TCP connection for data exchange, an ARP request is posed. The sender is host B. The target is S1. S1 sends its MAC address as ARP reply to host B. Host B now knows the address of S1 and inserts it into its table.

Computer C is attached to the network and sends a gratuitous ARP containing A's IP address because it has been configured in a wrong manner. A sends an ARP reply with its MAC address. Once this reply is received by C, C learns that there is obviously another computer which already uses this IP address. It shows an error message on its screen with the text "duplicate IP address". Afterwards, the IP address of C has to be changed. It now gets the IP address 192.122.10.4 and sends a new, gratuitous ARP. Now all computers in the network know the MAC address of C and insert it into their ARP tables.

Now there is an incident so that server S1 fails. The backup server S2 takes over the functionality of S1 and at first sends a gratuitous ARP request. As a consequence, all computer in the network know the MAC address of S2 and overwrite the old entry of S1. As you can see, communication may not be interrupted even though server S1 is broken.

End printversion

## 4.6 DHCP - Dynamic Host Configuration Protocol



arrangement

### 4.6 DHCP - Dynamic Host Configuration Protocol

#### 4.6.1 DHCP Configuration Options

#### 4.6.2 DHCP Procedure

The configuration of IP addresses can be done manually. For example, you can set your own IP address in Windows by accessing the properties of network adapters and opening the IPv4 or IPv6 configuration. There you can enter the IP address as well as other information such as subnet mask, default router and DNS servers.

However, the alternative is to obtain these configuration data automatically, which is possible via **DHCP (Dynamic Host Configuration Protocol)**. DHCP is very helpful from the perspective of administrators. If for example, pools are managed with a lot of PCs, it is better to activate the protocol on the devices, rather than to configure them manually. A DHCP server can then be used to organize the distribution of configuration data in a centralized manner. Changes of address ranges can be implemented much easier in this way as well.

Another scenario where the benefit of DHCP is obvious is the use for mobile devices: When, for example, students come to the university in the morning, their devices have to receive IP addresses and other configuration data that belong to the university network, in particular the IP address of the default router. When the students use their DSL connections at home in the evening, then another configuration is necessary. So it is very desirable that these changes occur automatically.



In the online version an video is shown here.

Link to video : <http://www.youtube.com/embed/Ewu1rxszwdg> 

**Dynamic Host Configuration Protocol**

### 4.6.1 DHCP Configuration Options

DHCP (RFC 2131, March 1997) is configured on the DHCP server side so that certain address ranges are intended for dynamic allocation. You can specify for example that IP addresses are allocated dynamically from the network range 192.168.2.0/24. This is called a **DHCP address pool**. As already mentioned, DHCP does not only allow to assign IP addresses dynamically, but also subnet masks, default routers and DNS servers are assigned. Usually two DNS servers are specified. If the client knows two DNS servers, then it can send its requests to the other DNS server if one fails.

When configuring the DHCP address pool, you have to consider that the default router needs an IP address that matches the network. So related to the previous example the router may receive the address 192.168.2.1 (this is also actually a setting in common DSL routers). This address then cannot be allocated dynamically to other devices.

When configuring the DHCP server, you can also specify a **lease time**. This is the time that an address assignment is valid if it is not renewed. This should be made dependent on the usual usage behavior. If, for example, students are attending a lecture, the time can be set to 105 minutes so that it is sufficient for the duration of the lecture. You must also consider how large the address pool is.

You can also make static assignments via DHCP. This is done in conjunction with the MAC address. This means that if a DHCP request is made using a certain MAC address, the computer with this MAC address will always get the same IP address.

### 4.6.2 DHCP Procedure

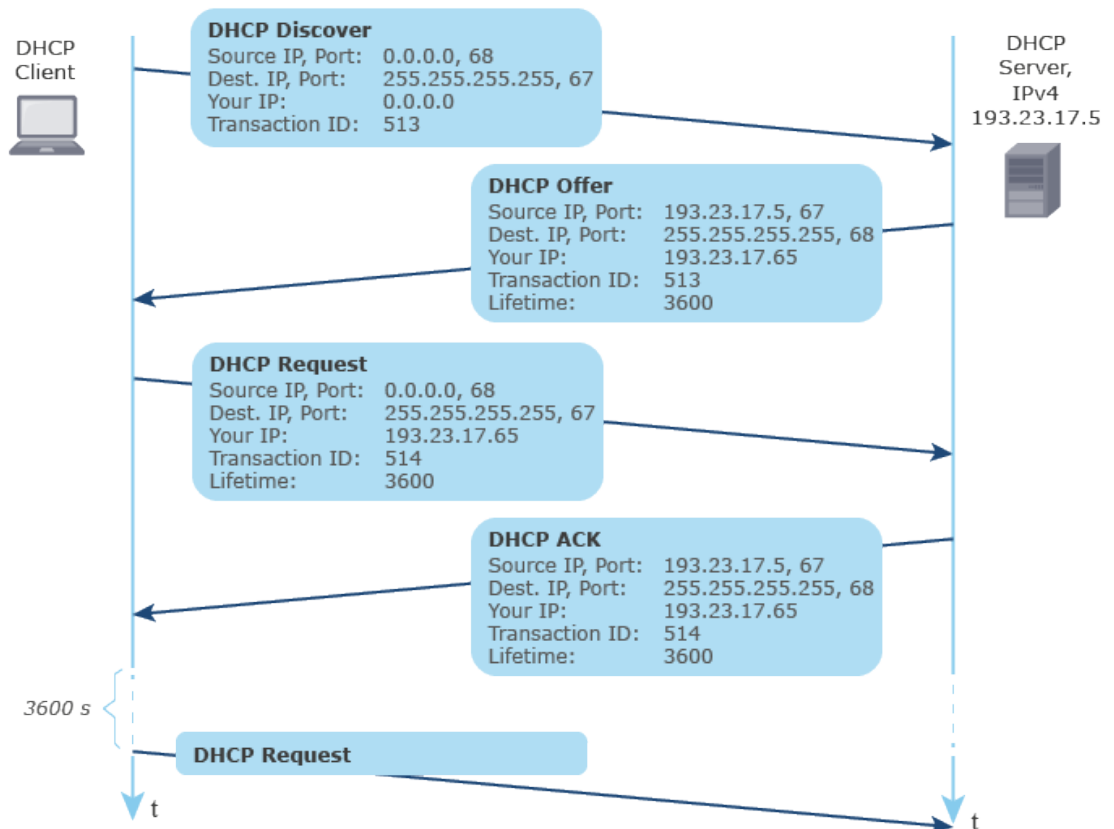
The dynamic address allocation via DHCP consists of four steps. The UDP port number 67 is used for client requests, and the UDP port number 68 is used for responses. The relationship between requests and responses can be seen from the transaction number in the DHCP data.

**DHCP discover:** First, the client searches for a DHCP server. This request must be a broadcast; but it must be kept in mind that the network mask is also still unknown. Therefore all bits of the destination IP address are set to one here. The source address consists only of zeros.

**DHCP offer:** A DHCP server then offers an IP address to the client and also indicates how long it will be valid.

**DHCP request:** Here the client indicates that it would like to use offered the IP address. In this request, the client does not use the IP address immediately; it uses the address as it does with DHCP discover.

**DHCP ack:** With the acknowledgment, the server confirms that the client may now use the IP address.



#### DHCP process

In the context of this procedure, it must be kept in mind that the reliable assignment of IP addresses is very important for the network to work properly. A client that does not receive an IP address cannot communicate in a reasonable manner. Therefore, it is useful to run several DHCP servers in a network. So a DHCP offer would not be provided by a single DHCP server but from several. However, the client only answers to one DHCP server. The other DHCP servers can use the other IP addresses they have offered for other end systems later.

## 4.7 Network Address Translation



arrangement

### 4.7 Network Address Translation

#### 4.7.1 Dynamic NAT/PAT

#### 4.7.2 Example of Dynamic NAT/PAT

#### 4.7.3 Static NAT/PAT




One of the biggest problems with IPv4 from today's point of view is the lack of IPv4 addresses. In addition to using CIDR, the use of private IP address ranges is an important method for dealing with this problem. However, private addresses may not appear on the public Internet as agreed. Therefore, terminals that have such addresses can only communicate locally. But since this is not desirable, **network address translation** (NAT) (RFC 2663) was introduced. This technique is used to translate private IP addresses into public IP addresses.

Because in typical scenarios far more private IP addresses are in operation than public IP addresses, a 1:1 mapping between the IP addresses is not possible. Therefore, the port numbers from the Transport Layer are included; this means the technique should more precisely be called **network address translation / port address translation (NAT/PAT)**. The port numbers basically serve to distinguish between applications, but are used here to distinguish between terminals.

Whether this technique is regarded as an improper use of port number is a matter of perspective. In any case, it is undisputed that NAT / PAT is used very often in practice and the lack of IPv4 addresses has since been significantly mitigated. However, it can also be argued that its use has delayed the fundamental improvement of using IPv6.



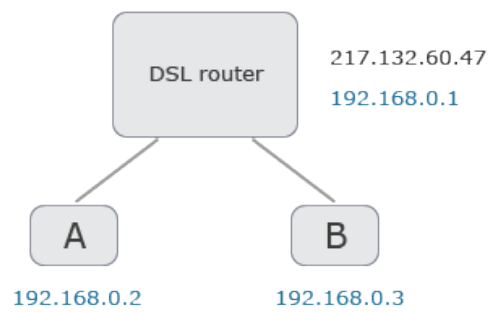
In the online version an video is shown here.

Link to video : <http://www.youtube.com/embed/qyHelWi26ak> 

**Network Address Translation**

### 4.7.1 Dynamic NAT/PAT

NAT/PAT is found especially in homes with DSL connections and other Internet access options. Here, the ISP provides only one public IPv4 address but multiple end systems are supposed to be used at the same time. However, they have to be distinguishable and therefore receive different private IP addresses.



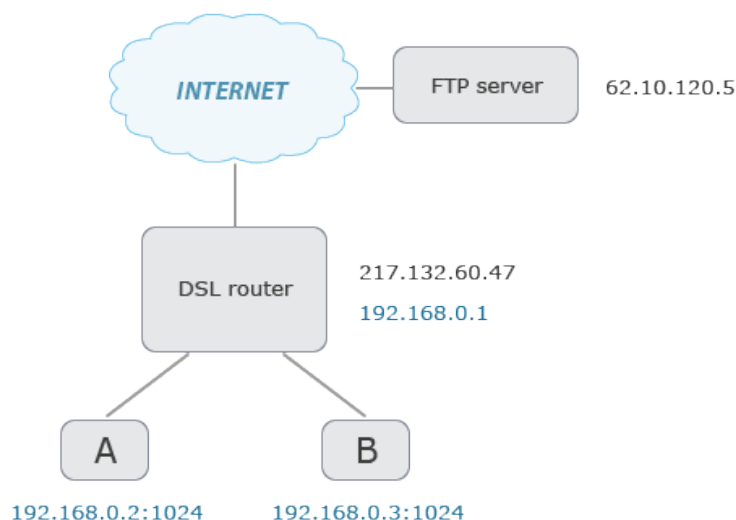
**DSL router with integrated NAT/PAT functionality**

The figure above shows a router that also has the ability to perform NAT/PAT. It is connected to a private home network. The connection to the Internet is made via a DSL modem. It should be noted here that DSL modems and routers are usually integrated in hardware so they are referred to as DSL routers. These devices usually have additional functionality (e.g. WLAN access point, telephone system, etc.).

In such a scenario, IP addresses from the range of the private IP addresses are applied; the range 192.168.0.0/16 is often used. In the example, the default router 192.168.0.1 must be known on hosts A and B. Hosts A and B use the private addresses 192.168.0.2 and 192.168.0.3. Because private IP addresses may not appear on the Internet, they must be appropriately mapped by the router to the public IP address provided by the ISP (in this example 217.132.60.47) using NAT/PAT. This is done automatically and with assignments that are valid only for a limited time. That is why it is called dynamic NAT/PAT.

### 4.7.2 Example of Dynamic NAT/PAT

For the purpose of illustration, we assume that the host A would like to use an FTP connection to a remote FTP server. Host B wants to communicate with the same FTP server.



### Dynamic NAT/PAT

#### What steps are necessary now?

Host A starts the FTP program, which is dynamically allocated to port 1024 (the port number is selected by random). In the IP packet that is sent to the router, the following IP addresses and port numbers are set:

Source (private) IP: Port	Destination IP: Port
192.168.0.2:1024	62.10.120.5:21



### IP addresses and port numbers in the original IP packet sent by host A

The router creates the following entry in its NAT/PAT table:

Private IP: Port	Public IP: Port	Destination IP: Port
192.168.0.2:1024	217.132.60.47:1100	62.10.120.5:21



### NAT table entry after processing the packet from host A

The router sends the IP packet to the destination using the public address 217.132.60.47 and the dynamically assigned port 1100. This address and port number has therefore replaced the original address and port number. The destination could then assume that an FTP program was started on the router. The router receives IP packets from the FTP

server, which it forwards to the private IP address 192.168.0.2 and port number 1024. To do this, the translation is reversed on the router.

Host B now also starts the FTP program, which is randomly allocated to port 1024 dynamically. This is just an example to show that it works even with the same port number on different hosts. In reality in most situations different port numbers occur. In the IP packet that is sent to the router, the following IP addresses and port numbers are set:

Source (private) IP: Port	Destination IP: Port
192.168.0.3:1024	62.10.120.5:21



IP addresses and port numbers in the original IP packet sent by host B

The router creates the following entry in its NAT/PAT table:

Private IP: Port	Public IP: Port	Destination IP: Port
192.168.0.2:1024	217.132.60.47:1100	62.10.120.5:21
192.168.0.3:1024	217.132.60.47:1101	62.10.120.5:21



NAT table entries after processing the packet from host B

The router sends the changed IP packets to the FTP server and forwards the FTP server's IP packets, which were received as a response, to the host 192.168.0.3 (port 1024).

Seeing it from an outside perspective from the Internet the only difference between the two hosts A and B is the port number. The used IP address 217.132.60.47 is identical.

The operations that have to be performed by the router require significant efforts. Every datagram is changed.

- The IP address is exchanged, which requires a recalculation of the checksum in the IP header.
- The port numbers are exchanged. Therefore the TCP/UDP checksum must be calculated again for all data.



annotation

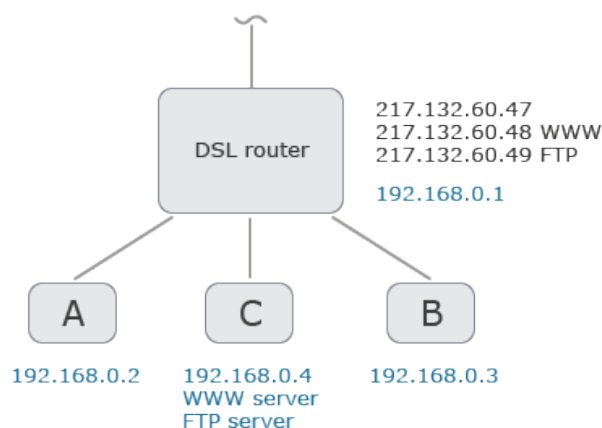
In the example, the destination IP address and port number are also recorded in the NAT table. This is not necessary for NAT/PAT to work. This information can, however, be included to better control or reduce the communication options for security reasons.

### 4.7.3 Static NAT/PAT

Dynamic NAT/PAT generates entries in the router NAT table in a completely automated manner. However, internal computers - for example an own FTP server - cannot be reached via the Internet because the port number, e.g. 21, is exchanged by the NAT/PAT router.

If internal servers need to be reachable via the Internet, static entries have to be generated in the NAT table. This procedure is called **static NAT/PAT**.

Suppose that a WWW server and an FTP server are installed in an internal network. They should be reachable from the Internet under different IP addresses, e.g. under the address 217.132.60.48 and 217.132.60.49, respectively. The IP address of the server in the internal network where both services are offered should be 192.168.0.4.



Static NAT/PAT

The static entries in the router NAT table in this example are as follows:

Private IP: Port	Public IP: Port	Destination IP: Port
192.168.0.4:80	217.132.60.48:80	
192.168.0.4:21	217.132.60.49:21	

**NAT table containing static entries**

The static entries must be set when booting the router. They are not changed again. A sender in the Internet reaches the WWW server under the address 217.132.60.48 – it does not know that it is actually the NAT router's address. The router records the sender address in the NAT table and forwards the packet to the address 192.168.0.4:80. Again this recording is not necessary for NAT/PAT to work, but can be useful for security reasons.

This technique can also be used if the internal servers have different IP addresses or when only a single public address was assigned.

## 4.8 Internet Protocol Version 6 (IPv6)

**arrangement**

### 4.8 Internet Protocol Version 6 (IPv6)

#### 4.8.1 Base Header

#### 4.8.2 Extension Headers

#### 4.8.3 IPv6 Addresses

#### 4.8.4 ICMPv6

#### 4.8.5 Automatic Address Configuration

#### 4.8.6 IPv6 Fragmentation

#### 4.8.7 Jumbograms

#### 4.8.8 Mobile IPv6

#### 4.8.9 Summary - IPv6


In the development of the Internet protocol version 4 in the 1970s, an address length of 32 bits was defined. At the time, it was not foreseeable that the Internet would develop so rapidly in terms of user numbers in the subsequent decades. In addition, there were a number of requirements, e.g. for security and data throughput, which were not provided for in the protocol. The Internet protocol was therefore fundamentally revised starting from 1995 to meet the new requirements. By the way, the version number 5 was intended for a special stream transport, which is however no longer relevant.

- The most important change from IPv4 to IPv6 is the expansion of the address range from 32 to 128 bits to eliminate the shortage of IP addresses. Compared to this change, the other changes are much less important.

- Encryption and authentication techniques are supported by the new Internet protocol.
- The structure of the header has been simplified compared to IPv4. There is an always-present base header with a few fields as well as a set of extension headers.
- The emerging, increased transmission of audio and video data is taken into account in the improved support of parameters for determining quality of service.



In the online version an video is shown here.

Link to video : <http://www.youtube.com/embed/-dRcAeqUjOs> 

Internet Protocol Version 6

### 4.8.1 Base Header

The structure of the base header and extension header were described in RFC 2460 (Dec. 1998) as the following rollover shows.



In the online version an rollover element is shown here.

IPv6 base header

Begin printversion

0	8	16	24	31
Version	Traffic Class		Flow Label	
	Payload Length		Next Header	Hop Limit
		Source Address		
		Destination Address		

End printversion

The fields in the IPv6 header are the following:

- **Version:** This field indicates that it is an IPv6 header. For this purpose, a binary-coded 6 is included. This field is the same for IPv4, except that a binary-coded 4 is contained in the IPv4 header.

- **Traffic class:** This field is used to distinguish between different types of data traffic in order to treat them differently. It is similar to the “type of service” or “DiffServ code point” field in the IPv4 header.
- **Flow label:** The “flow label” field was introduced in IPv6. It can be used to mark individual traffic streams in the network in order to be able to handle them in a specific way. This means that a stream can be preferred so that other packets have to wait or are even rejected. However, such an approach can lead to scaling problems if you want to carry out special treatments for many data streams. This field is also in competition with [MPLS technology](#) , which is often used in provider networks in addition to IP.
- **Payload length:** This length field indicates the length of the payload. Such a field is also contained in the IPv4 header.
- **Next header:** Options are handled differently in IPv6 in comparison to IPv4. So-called “extension headers” are used for options so this field refers to the first extension header. Many IP packets, however, do not contain any options so no extension header is present either. In this case, the field refers to the header of the Transport Layer protocol and thus corresponds to the “upper layer protocol” field of the IPv4 header.
- **Hop limit:** This field corresponds to the TTL field of IPv4. In IPv4, this field does not show a maximum time, contrary to its name, but a maximum number of forwards by routers. If you regard a “hop” as a forward by a router, the naming fits to the actual use.
- **Source address/Destination address:** The source and destination IPv6 addresses, each with 128 bits, are found at the end of the header. They occupy significantly more space than the 32-bit IPv4 addresses, which leads to the length of the header in IPv6 without options being 40 bytes. The IPv4 header has only 20 bytes if there are no options.

Compared to the IPv4 header, some fields have been eliminated from the IPv6 base header.


- **IHL:** The base header for IPv6 has a constant length of 40 bytes because the options have been moved to the extension headers. A constant length is an advantage for the efficient processing the header. You have to consider here that routers at central points of the Internet today have  $n$  times interfaces with 100 Gigabit/s where routing decisions have to be made for every IP packet. Efficient processing in hardware is therefore very important.



- **Identification, flags, fragment offset:** The fields necessary for fragmentation were moved to an extension header. As mentioned in relation to IPv4, fragmentation is relatively inefficient so you try to avoid it. This can be done successfully in many cases so only a small part of the IP packet is fragmented. Therefore, it is useful to move the fragmentation control into an extension header.
- **Checksum:** The checksum field is no longer available. With IPv4, the checksum has to be recalculated in every router because the TTL value changes in the header at each forwarding. So it was omitted for efficiency reasons. However, this raises the question of whether more bit errors could go undetected. You have to consider though that there are also error detection mechanisms at layers 2 and 4 that are still present. For example, the CRC checksum at the end of the Ethernet frame is much more reliable than the checksum in the IP header. Moreover, transmission in fixed-line networks has become significantly more reliable over time so only few bit errors still occur.

## 4.8.2 Extension Headers

The options that are known from IPv4 as well as additional functions have been implemented in the IPv6 **extension headers**. If no extension header is present, a header from the Transport Layer follows the base header as part of the payload, which is a TCP or UDP header in most cases.

Some extension headers are presented here. A list of extension headers with possible parameters is available at the [IANA](#) .

- **Hop-by-hop options header**

This header contains options that have to be evaluated at each router. It must directly follow the base header in order to shorten the processing time in the router. This extension header is used, for example, for jumbograms (RFC 2675).

- **Routing header**

This header was introduced so it is possible to influence the route of an IP packet. You can specify routers through which the packet must be routed. This application was flagged as obsolete in RFC 5095 because it can be abused by attackers. In addition, it raises the question of why one would want to influence the routing in this way (apart from for test purposes) rather than changing the configuration of the routing protocol. Currently there is only an application for this header for Mobile IPv6 (RFC 6275). There is also the special mobility header for this purpose, which is also defined in this RFC.

- **Fragment header**

In contrast to IPv4, routers are not permitted in IPv6 to fragment packets on the way to the destination. Instead, the sender must determine the maximum packet size on the route to the destination, which is called **path MTU discovery**, and if necessary fragment the packets on its own. Using the fragment header (RFC 2460), it is possible for the destination host to reassemble the packets correctly. IPv6 assumes that at least 1280 bytes can be transmitted without the need to fragment.

- **Authentication header**

This extension header (RFC 4302) allows a receiver to determine whether the packet is actually from the indicated source or whether it has been modified during transmission. In other words, the integrity and authenticity are secured. The authentication header is a part of [IPsec](#), which is an extension of IP introduced in order to later add security aspects that were not considered in the original design. This extension was initially developed for IPv6 but was also carried back to IPv4 after the delayed introduction of IPv6.

- **Encapsulating security payload header**

This extension header (RFC 4303) is used in order to secure confidentiality in addition to integrity and authenticity (as with the authentication header). Encryption algorithms can be selected for this purpose. The encapsulating security payload header is also a part of IPsec and is also offered as an extension of IPv4.

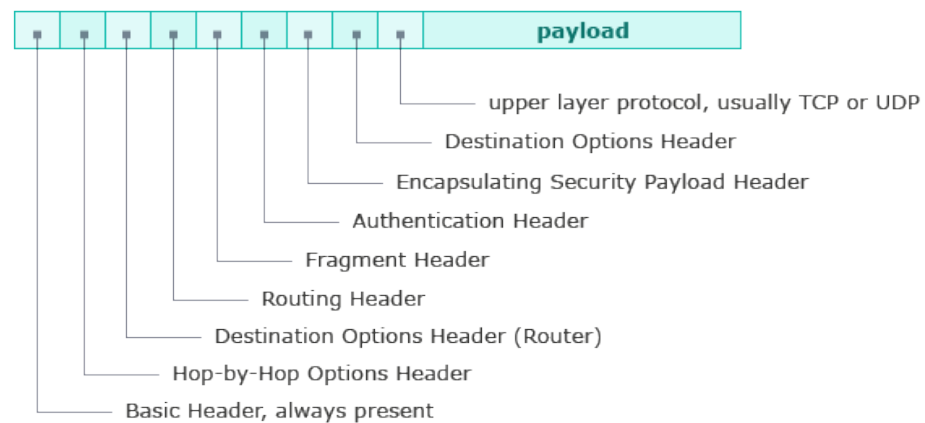
- **Destination options header**

This extension header (RFC 2460) is used to transmit optional information which is evaluated exclusively by the destination host. This is the only header that can occur twice. If the information refers to the router, it is transmitted directly after the hop-by-hop options header. If the information is related to the destination system, it is transmitted as the last extension header.

There is also the headers host identity protocol (RFC 7401), Shim6 protocol (RFC 5533) and no text header (RFC 2460).


The length of the base header, as already mentioned, is always 40 bytes. Most extension headers can vary in length; therefore they have a length field that specifies the length of the extension header.

The order proposed in RFC 2460 with the maximum number of IPv6 headers possible is:



### Order of the IPv6 extension headers

However, the order does not have to be followed in any case.

Every IPv6 header has a **next header field** in which the type of the next header is specified. Some possible values are given in the following (see [IANA list](#) 



### Values in the next header field and their meaning

Source: IANA

Begin printversion

value in next header field	following header type	explanation
0	HOPOPT	IPv6 Hop-by-hop option
6	TCP	Transmission control
17	UDP	User datagram
43	IPv6–Route	Routing header for IPv6
44	IPv6–Frag	Fragment header for IPv6
50	ESP	Encapsulating security payload
51	AH	Authentication header
58	IPv6–ICMP	ICMP for IPv6
59	IPv6–NoNxt	No-next-header for IPv6
60	IPv6–Opts	Destination options for IPv6
135	Mobility	Mobile IPv6

End printversion



example

Two examples are provided here to show which headers can appear and which corresponding values can be present in the next header field.

- IPv6 packet with TCP data:



In the online version an rollover element is shown here.

**IPv6 packet with TCP data**

Begin printversion

IPv6	TCP	payload
<i>field</i>	<i>description</i>	
IPv6	Next Header = 6	

End printversion

- IPv6 packet with routing and fragmentation headers as well as UDP data:



In the online version an rollover element is shown here.

**IPv6 packet with routing and fragmentation headers as well as UDP data**

Begin printversion

IPv6	Routing	Fragment	UDP	Payload
<i>field</i>	<i>description</i>			
IPv6	Next Header = 43			
Routing	Next Header = 44			
Fragment	Next Header = 17			

End printversion

### 4.8.3 IPv6 Addresses



arrangement

### 4.8.3 IPv6 Addresses

#### 4.8.3.1 Address Representation

#### 4.8.3.2 Unicast, Multicast and Anycast Addresses

#### 4.8.3.3 Scopes of Unicast and Multicast Addresses

#### 4.8.3.4 Aggregatable Global Unicast Addresses

#### 4.8.3.5 Construction of Interface IDs

#### 4.8.3.6 Unique Local Unicast Addresses

#### 4.8.3.7 Link Local Unicast Addresses

#### 4.8.3.8 Multicast Addresses

#### 4.8.3.9 Node Addresses

#### 4.8.3.10 Privacy Protection

In IPv6, there are unicast and multicast addresses with the same meaning as in IPv4. Newly added are anycast addresses, which are assigned to a group of interfaces; an IP packet with such a destination address is only forwarded to the next interface belonging to the group. In contrast, there are no longer broadcast addresses because a broadcast is considered a special case of multicast where all interfaces should receive the IP packet.

An important innovation related to IPv6 addresses is that an interface can have several IPv6 addresses. This is actually typically the case because IPv6 addresses have different areas where they are valid.



In the online version an video is shown here.

Link to video : <http://www.youtube.com/embed/5cZBmIX9Wg8>

**IPv6 Addresses**

A global division of the entire IPv6 address space can be retrieved at [IANA](#) . You can see that a large part of the address space has not yet been allocated for specific purposes.



task

### **Task: IP addresses per square meter of Earth's surface**

How many IP addresses are there per m<sup>2</sup> of the Earth's surface with IPv6 and IPv4?

**Solution**

Assume that the earth is a ball with a radius of 6378 km and that with IPv6  $2^{128} = 3.4 \cdot 10^{38}$  addresses are available and with IPv4  $2^{32} = 4,3 \cdot 10^9$  addresses are available.

**As a reminder**, the surface of a ball is calculated as:

$$O = 4 \pi r^2.$$

**Solution:**

$$\text{Radius } r = 6378 \text{ km} = 6.378 \cdot 10^3 \text{ km} = 6.378 \cdot 10^6 \text{ m}$$

$$O = 4 \cdot \pi \cdot 40.67 \cdot 10^{12} \text{ m}^2 = 511 \cdot 10^{12} \text{ m}^2 = 5.11 \cdot 10^{14} \text{ m}^2$$

- IPv6 addresses / surface =  $3.4 \cdot 10^{38} / 5.11 \cdot 10^{14} = 0.66 \cdot 10^{24}$  addresses/m<sup>2</sup>
- IPv4 addresses / surface =  $4.3 \cdot 10^9 / 5.11 \cdot 10^{14} = 0.78 \cdot 10^{-6}$  addresses/m<sup>2</sup>

### 4.8.3.1 Address Representation

The **128 bits** of an IPv6 address are divided into eight blocks of 16 bits each (according to RFC 4291). These blocks are separated by colons. The 16 bits of a block are represented by four hexadecimal digits (0...9, A...F):

xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx



example

2030:000F:0014:0000:0000:33AF:EE01:1234

To simplify writing, abbreviation rules were defined in RFC 5952. According to the RFC, leading zeros may be omitted. If a whole block, or directly adjoining blocks, consist only of zeros, they can be abbreviated as "::".



example

2030:F:14:0:0:33AF:EE01:1234

2030:F:14::33AF:EE01:1234

This abbreviation is, however, only allowed once in the address so it is clearly recognizable how many zeros were omitted in which position. Within the two colons "::" as many zeros must be filled in as are needed to make the address 128 bits long again.

The IPv4 notation (“**dotted decimal notation**”) can be used for addresses in mixed IPv4/IPv6 environments.



example

```
xxxx:xxxx:xxxx:xxxx:xxxx:xxx:ddd.ddd.ddd.ddd  
0000:0000:0000:0000:0000:0000:193.175.120.62  
::193.175.120.62  
::C1AF:783E
```

If an application program uses the IPv6 address instead of the DNS name, the IPv6 address must be placed in square brackets.



example

```
http://[2030:F:14::33AF:EE01:1234]:80/abc/default.php
```

The identification of the part of an address to be evaluated by routers is done via the **address prefix** - just as with IPv4 addresses in the CIDR notation.



example

```
2030:F:14::33AF:EE01:1234/56
```

The first 56 bits of the address are evaluated by the router in this example.

The **loopback address** is ::1. It corresponds to the localhost address of IPv4 with the IPv4 address 127.0.0.1.

### 4.8.3.2 Unicast, Multicast and Anycast Addresses

**Unicast addresses** are required for point-to-point communication. Such an address can only be used by a single interface.

All receivers with the same **multicast address** form a multicast group. The same multicast address can be used by different systems (interfaces) in different networks. Typical multicast applications could be video conferences or communication between routers (with routing protocol OSPF). The routers must be multicast-capable. There is a special format for multicast addresses.

A group of interfaces can be addressed with **anycast addresses**; the IP packet here is only sent to one interface. Generally, it is the nearest one according to the routing protocol. Anycast addresses can be used to submit requests to a group of servers that all offer the same services without needing to configure the different addresses for the servers manually. For example, an anycast address can be used in order to submit a request to a group of DNS servers; in this case, only the nearest server will respond. All anycast addresses have the same format as unicast addresses and cannot be distinguished from them. Therefore, the interfaces have to be explicitly configured with anycast addresses. IPv6 end systems may not use anycast addresses.

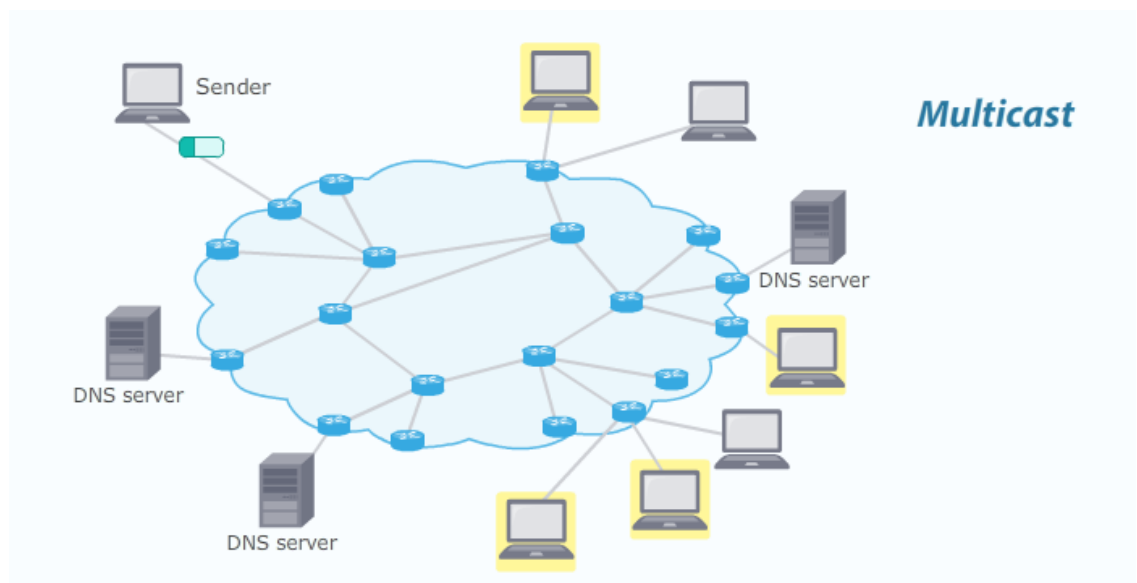
The following animation shows how unicast, multicast and anycast addresses can be used on the Internet.



In the online version an animation is shown here.

#### IPv6 address types

Begin printversion



A unicast address is assigned to a single interface. It is used for point-to-point communication.

All stations that have the same multicast address form a multicast group. A data packet to this address is provided to all group members.

If several stations, in this example DNS servers, are reachable via the same anycast address, the data packet is provided to the nearest station according to the routing protocol.



End printversion

### 4.8.3.3 Scopes of Unicast and Multicast Addresses

A key concept of IPv6 is the introduction of different areas where IPv6 addresses are valid. They are called **scopes**.

- With unicast addresses, the “link local” scope is smaller than the “site local” scope, and the “site local” scope is smaller than the global scope.
- With multicast addresses, a scope with a small value in the scope field is smaller than one with a large value.

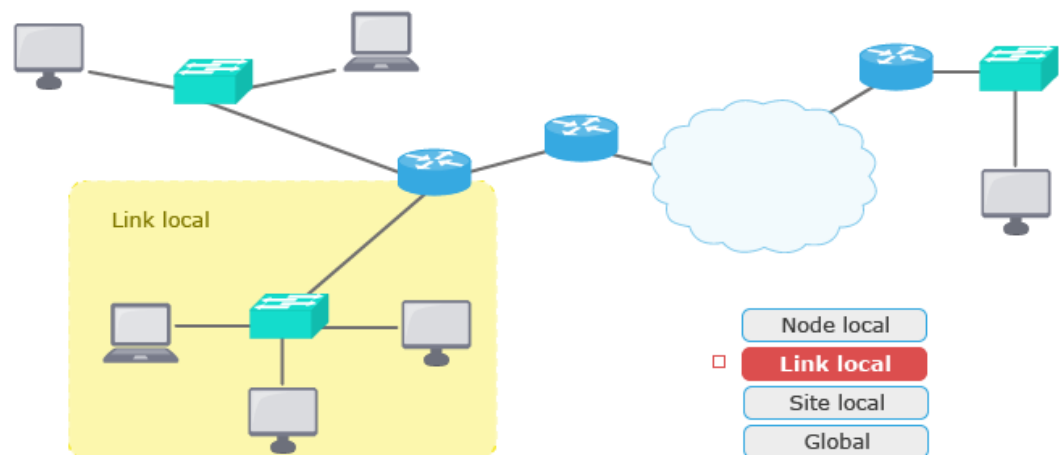
The following interactive rollover shows the scopes of IPv6 addresses.



In the online version an rollover element is shown here.

#### Scopes of IPv6 addresses

Begin printversion



End printversion

### 4.8.3.4 Aggregatable Global Unicast Addresses

There are globally valid IPv6 addresses as part of the unicast addresses which correspond to the globally valid IPv4 addresses. In the specification of IPv6 the possibility to define different hierarchy levels was an important point. This possibility

allows to aggregate address ranges in the routing tables to limit the table sizes better than with IPv4.

The format that the aggregatable global unicast addresses have is shown in the following interactive rollover:



In the online version an rollover element is shown here.

#### Format for aggregatable, global unicast addresses

Begin printversion



<i>field</i>	<i>description</i>
<b>Global Routing Prefix</b>	First n bits are used for routing between organizations.
<b>Subnet ID</b>	Additional 64-n bits are available to form internal subnets.
<b>Interface ID</b>	Theses 64 bits are used to distinguish between network interfaces within a network.

End printversion

The addresses consist of three logical parts:

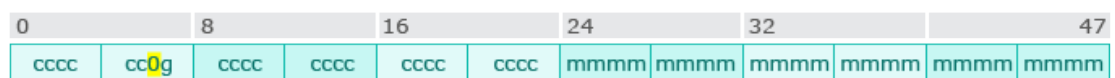
1. The global routing prefix consisting of n bits is distributed hierarchically, as it is already known from IPv4 addresses. The entire address space is available to IANA. It distributes it to the Regional Internet Registries, who then distribute parts to local registration authorities. These are Internet service providers. For example, RIPE has assigned the address space 2001:0638::/32 to DFN which is a part of the address space 2001:0600::/23 managed by RIPE. The n in this example is 32 and 23, respectively. It is also possible for universities to receive address spaces directly from RIPE (so called provider independent address spaces [\[4\]](#)). The University of Kaiserslautern for example uses such the PI address space 2a03:63c0::/32. FH Lübeck in contrast uses the address space provided by DFN, i.e., 2001:0638:0707::/48.
2. The other 64-n bits specify a **subnet ID** and define the **private part** of the IPv6 address. This means an organization can create networks on its own; often /64 networks are used within an organization. The network 2001:0638:0707:0001::/64 is for example a network at FH Lübeck.

3. The **EUI-64 (extended unique identifier)** format, which is standardized by IEEE is often used as **interface ID** (64 bits). In this case, the interface ID is derived from the MAC address. This is in particular useful for automated address configuration.

In the first IPv6 specifications, stricter rules on how to form address spaces were foreseen. However, they were declared as no longer valid in RFC 3587, and the more flexible model as described above was introduced.

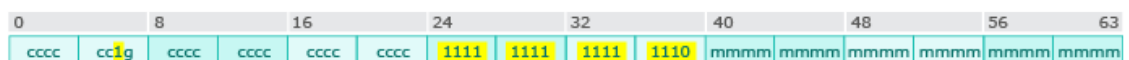
### 4.8.3.5 Construction of Interface IDs

As already mentioned, the interface ID part of the IPv6 address can be derived automatically from the MAC address. To expand the 48 bits of the MAC address to 64 bits, two bytes are inserted with the values 0xFF and 0xFE between the company and manufacturer bits. The “u” bit is also inverted.



IEEE 802 48-bit address

This results in:



IPv6 Interface ID (64 bit)



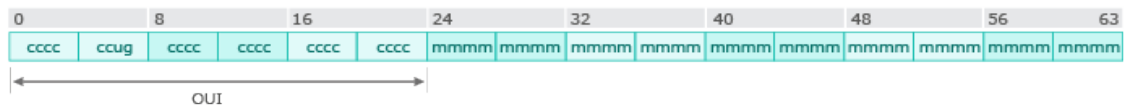
example

**MAC address with 48 bits :** 00-80-25-00-3A-3B

**IPv6 Interface ID:** 0280:25FF:FE00:3A3B

The format achieved before the inversion of the “u” bit is called **EUI-64** and is composed of two parts:

1. The 24-bit long company identifier (OUI, Organizationally Unique Identifier), which is assigned by IEEE;
2. And the 40-bit long identifier that is assigned independently by the company.



#### EUI-64-ID

c	company bits
u	0 = universal, 1 = local
g	0 = unicast, 1 = multicast
m	manufacturer bits



#### EUI-64 ID

Begin printversion

c	company bits
u	0 = universal, 1 = local
g	0 = unicast, 1 = multicast
m	manufacturer bits

End printversion

### 4.8.3.6 Unique Local Unicast Addresses

The concept of “**site local**” addresses, which are only valid in a closed network, has been revised over time and redefined in RFC 4193. “Site local” addresses were originally intended for the same tasks as private addresses in IPv4 (see [IPv4 Addresses](#)), but they now have different features. In particular, they have been designed so that they are (most likely) globally unique, which is not the case for private IPv4 addresses. This avoids

certain difficulties if, for example, it cannot be clearly defined what belongs to a local area. The addresses are now called **unique local unicast** addresses.

The definitions for these local IPv6 addresses have existed since 2005. The unique local unicast IPv6 addresses have the following features:

- They have a (very likely) globally unique prefix.
- They can be filtered at site boundaries based on the prefix.
- Different sites can be combined or privately connected without address conflicts.
- They are independent from Internet service providers and can be used within a site.
- If they appear in the global Internet, there are no conflicts with other addresses.
- They can be treated in applications the same way as global addresses.

The following interactive rollover shows the format of the unique local unicast IPv6 addresses.



In the online version an rollover element is shown here.

#### Format of unique local unicast IPv6 addresses

Begin printversion

0		64		127
Pre-fix	L	Global ID	Subnet ID	Interface ID
7	1	40	16	64 Bit

field	description
Prefix	FC00::/7 prefix for local IPv6 addresses
L	1: locally assigned address, 0: for future use
Global ID	40 bit global ID to generate a globally unique prefix
Subnet ID	16 bit subnet ID
Interface ID	64 bit interface ID

End printversion

#### Generation of global ID

In the area FC00::/7, two different address spaces have to be differentiated.

Begin printversion

FC00::/8	future use not yet decided, central administration is likely
FD00::/8	uncoordinated use by organizations

End printversion

In the area FD00::/8, the global ID is to be created with a pseudo-random number generator by the respective network operator. This is done **without consultation** or assignment by a higher-level instance such as RIPE. Sequential IDs or fixed IDs should not exist, so it is clear that these IDs cannot be aggregated as with CIDR and that they cannot be used on the Internet instead of global addresses. The following algorithm (RFC 4193) should be used for **prefix generation**:

1. Determine the time of day in the 64-bit NTP format.
2. Determine the EUI-64 ID possibly from the 48-bit MAC (Ethernet) address.
3. Merge both values in one key.
4. Determine the 160-bit long SHA1 checksum from this key.
5. The least significant 40 bits are used as the global ID.
6. FD00::/8 is used as the prefix.

This algorithm is used once for a local site to set the prefix for the site. The probability of generating duplicate IDs is so small that the resulting addresses can be assumed to be unique globally.



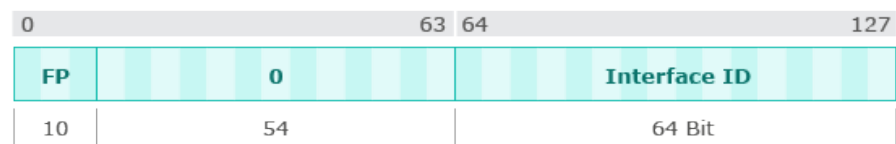
annotation

In practice, you should consider the purpose for using these addresses. Private IP addresses were introduced because of a lack of IPv4 addresses, which is not the case with IPv6. So there are enough global IPv6 addresses available. The use of such addresses could be considered for security reasons because they can be filtered easily in the router to the Internet. Some internal services would then not be accessible from the outside.

### 4.8.3.7 Link Local Unicast Addresses

Another type of IPv6 address with even more limited validity are **link local unicast** addresses. They are valid from an end system to the next router. A “link” in this sense is therefore more than just a link from an end system to a switch.

They are used for the **autoconfiguration** of IPv6 addresses, for **neighbor discovery**, or for communication without routers. Routers are not allowed to forward these addresses.



FP (format prefix) = 1111 1110 10



#### Format of link local unicast addresses

The addresses consist of 10 bits as format prefix followed by 54 zeros and the interface ID.

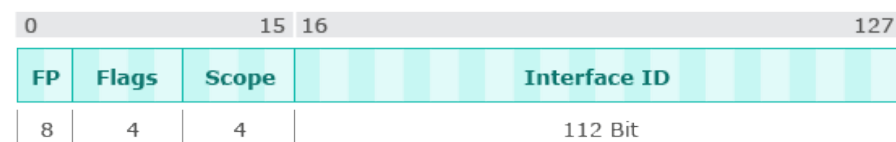


example

FE80::0280:25FF:FE00:3A3B

### 4.8.3.8 Multicast Addresses

An **IPv6 multicast address** identifies a group of interfaces. An interface can have any number of multicast addresses.



#### Format of multicast addresses



#### Format of multicast addresses


Begin printversion

FP	Format prefix: 1111 1111	
Flags	The first 3 bits are 0. The fourth bit indicates a temporary multicast address (1) or a permanently defined multicast address (0) which is administrated by IANA.	
Scope	Areas where the multicast address is valid	
	1	node local
	2	link local
	5	site local
	8	organization local
	E	global

End printversion

Unlike IPv4 multicast addresses, IPv6 multicast addresses are structured. They refer to certain areas (scopes). Therefore, they can be used to reproduce the broadcast functionality of IPv4, e.g. for use during automatic address configuration or to discover IPv6 routers. In contrast to IPv4, all IPv6 routers must support multicast mechanisms.

- **“Node Local”** multicast addresses denote an interface within a node and are only used for multicast loopback tasks, e.g. as a form of interprocess communication within a host.
- **“Link local”** and **“site local”** multicast addresses refer to the same area as the corresponding unicast addresses, namely to an individual link (no forwarding by router), or a private site (no forwarding to the Internet).

Permanent multicast addresses can be defined in certain areas or in all areas (see [list at IANA](#) ) . For example, the permanent multicast address for the “Network Time Protocol” server (computers can use this protocol to ask for the time) is defined in all scopes:



#### Permanently defined multicast addresses for NTP

Begin printversion

FF01:0:0:0:0:0:0:101	all NTP servers on the same node (node local)
FF02:0:0:0:0:0:0:101	all NTP servers on the same link (link local)
FF05:0:0:0:0:0:0:101	all NTP servers in the same site (site local)
FF0E:0:0:0:0:0:0:101	all NTP servers in the Internet (global)

End printversion



Some other permanently defined multicast addresses are:



#### Permanently defined multicast addresses

Begin printversion

FF01:0:0:0:0:0:0:1	<b>All nodes:</b> just defined for node local and link local scopes, i.e., these addresses are only valid for a node or a link and are not forwarded by routers.
FF02:0:0:0:0:0:0:1	
FF01:0:0:0:0:0:0:2	<b>All routers:</b> just defined for node local, link local and site local scopes, i.e., it is not possible to reach all routers in the Internet via a single address, but at most all routers within a site.
FF02:0:0:0:0:0:0:2	
FF05:0:0:0:0:0:0:2	

End printversion



task

#### Task: Multicast addresses

Are multimedia conferences possible on the Internet?

##### Solution

Yes, multimedia conferences are defined in all scopes:

FF0X:0:0:0:0:2:0000 – FF0X:0:0:0:0:2:7FFD

Instead of X, any scope can be specified.

### 4.8.3.9 Node Addresses

Which addresses must an IPv6 node know?

A **host** must know the following addresses:

- The “link local” address of every interface
- The unicast addresses assigned to the interfaces
- The loopback address
- The “solicited node multicast” address
- The “all nodes” multicast address

- The multicast addresses of all groups to which the host belongs

A **router** must also know the following addresses:

- The anycast addresses that are specified for the router
- The “all routers” multicast address

### 4.8.3.10 Privacy Protection

When using IPv6, new questions arise regarding user privacy.

A constant 64-bit long interface ID, which can be created by the MAC address or in other ways, can be misused to track the behavior of a user. This problem is particularly serious when mobile devices such as laptops, smartphones, etc. are used. Although the prefix can change since different access points are used, the interface ID remains constant. This means a **movement profile** can be created and the use of a mobile device can be tracked. To make spying more difficult, RFC 3041 (**privacy extensions**), proposes using a randomly generated interface ID instead of a constant interface ID. Its use should be limited to a certain number of hours. The random generation is based on the MD5 hash algorithm.

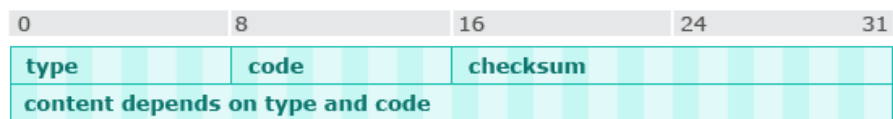
These dynamically generated interface IDs can be used when connections are initiated by clients. Servers continue to have a fixed interface ID. However, the dynamic allocation makes it more difficult to locate network problems because it is unclear whether the problems are caused by one computer or by various computers.

Another issue arises from the allocation of /64 address spaces to users by the provider. This allows to maintain user profiles regarding Internet sites visited if the address space is permanently assigned. The privacy extensions do not provide a protection against this possibility.

### 4.8.4 ICMPv6

ICMPv6 (Internet Control Message Protocol version 6) is similar to ICMP for IPv4. A part of the functions have been adopted from ICMP and some functions have been added (see RFC 4443 and RFC 4861).

The structure of the ICMPv6 header was adopted unchanged from ICMP as follows:



#### Structure of the ICMPv6 header

Error messages have a type value of 0 to 127; information messages use a type value between 128 and 255.



#### ICMPv6 messages

Begin printversion

Typ	Code	Meaning
1	Destination unreachable	
	0	No route to destination
	1	Communication with destination administratively prohibited
	3	Address unreachable
	4	Port unreachable
2	0	Packet too big
3	Time exceeded	
	0	Hop limit exceeded in transit
	1	Fragment reassembly time exceeded
4	Parameter problem	
	0	Erroneous header field encountered
	1	Unrecognized next header type encountered
	2	Unrecognized IPv6 option encountered
	16	Does not route common traffic
128	0	Echo request
129	0	Echo reply
130	0	Multicast listener query
131	0	Multicast listener report
132	0	Multicast listener done
133	0	ND: Router solicitation
134	0	ND: Router advertisement
135	0	ND: Neighbor solicitation
136	0	ND: Neighbor advertisement
137	0	ND: Redirect message
138	Router renumbering	
	0	Router renumbering command
	1	Router renumbering result
	255	Sequence number reset
139	0	ICMP node information query
140	0	ICMP node information response
141	0	Inverse neighbor discovery solicitation message
142	0	Inverse neighbor discovery advertising message

End printversion

Some message types are explained in more detail below.

- **Packet too big:** Routers are not permitted to fragment IPv6 packets (in contrast to IPv4). If the MTU of the interface selected by the router is too small for the desired packet length, this message is generated. The MTU of the affected interface is then

sent back in the payload. This mechanism is used by IPv6 in order to determine the smallest MTU for a path between sender and receiver (Path MTU Discovery).

- **Multicast listener messages:** These messages are used when routers want to determine whether there are hosts that want to receive specific multicast packets. These functions are performed by the IGMP protocol in IPv4.
- **Type 133 to 137:** Neighbor discovery: These messages are part of the **Neighbor Discovery Protocol** (ND) and are required for automated address configuration. The following options may be present in ND messages:
  1. Link address of the sender
  2. Link address of the destination
  3. Prefix information
  4. Header of an IP value that triggered a redirect message
  5. MTU
- **Router renumbering:** These messages can be used to automatically change address prefixes on routers. This is necessary, for example, if all addresses have to be changed when a provider is changed.
- **ICMP node information:** Query of network information of an IPv6 node such as a host message or domain message.
- **Inverse neighbor discovery:** Query of the IPv6 address of a node if only the link address is known.

#### 4.8.4.1 Neighbor Discovery Protocol

In IPv4, the determination of MAC addresses is done by ARP using Ethernet broadcast messages (if Ethernet is used on Layer 2). For IPv6, there is the **Neighbor Discovery Protocol**, which works similarly and uses ICMPv6 messages.

Let us assume that an interface with the IPv6 address

2030::14:0280:25FF:FE00:3A3B

and the MAC address

00-80-25-00-3A-3B

wants to send a packet to an interface with the IP address

2030::14:0280:25FF:FE11:2233

and the MAC address

00-80-25-11-22-33.

But the sender does not yet know the MAC address of the destination.

An ICMP “neighbor solicitation” message is sent to the “**solicited node multicast**” address, which is derived from the IP address of the destination. Because the multicast address can be ambiguous, the ICMP message must specify the IP address of the destination. The sender's own address is used as the source IP address. The ICMP message also specifies the sender's MAC address:

Destination address	“solicited node multicast” address: FF02::1:FE11:2233
Source address	2030::14:0280:25FF:FE00:3A3B
Hop limit	255
Next header	58: ICMPv6 message



IPv6 header of the ICMPv6 neighbor solicitation message

Begin printversion

Destination address	“solicited node multicast” address: FF02::1:FE11:2233
Source address	2030::14:0280:25FF:FE00:3A3B
Hop limit	255
Next header	58: ICMPv6 message

End printversion

Type	135: Neighbor solicitation
Option	Destination IP address: 2030::14:0280:25FF:FE11:2233 Source link address: 00-80-25-00-3A-3B



ICMPv6 neighbor solicitation message

Begin printversion

Type	135: Neighbor solicitation
Option	Destination IP address: 2030::14:0280:25FF:FE11:2233 Source link address: 00-80-25-00-3A-3B

End printversion

All recipients of this message can now update the mapping of the IP address to MAC address in their “**neighbor cache**.” The receiver with the IP address being searched for subsequently generates a “neighbor advertisement” message in which its own MAC address is entered. This message is sent directly to the sender of the request:

Destination address	2030::14:0280:25FF:FE00:3A3B
Source address	2030::14:0280:25FF:FE11:2233
Hop limit	255
Next header	58: ICMPv6 message



IPv6 header of the ICMPv6 neighbor advertisement message

Begin printversion

Destination address	2030::14:0280:25FF:FE00:3A3B
Source address	2030::14:0280:25FF:FE11:2233
Hop limit	255
Next header	58: ICMPv6 message

End printversion

Type	136: Neighbor advertisement
Option	Destination link address: 00-80-25-11-22-33



ICMPv6 neighbor advertisement message

Begin printversion

Type	136: Neighbor advertisement
Option	Destination link address: 00–80–25–11–22–33

End printversion

Once the original source receives this message, it can enter the mapping of the IP address to MAC address in its “neighbor cache.” The actual message can then be sent.



annotation

The “neighbor cache” corresponds to the ARP table for IPv4. In Windows, it can be viewed with the command line “netsh interface ipv6 show neighbors level=verbose”.

IPv6 does not use Ethernet broadcast when using multicast addresses. Rather, the last four bytes of the multicast address are copied into the last four bytes of the Ethernet address. The first bytes of the Ethernet address contain the hexadecimal value “3333” (see RFC 2464).

If, for example, a “neighbor solicitation” message is to be sent to the multicast address FF02::1:FE11:2233 to determine a MAC address, the Ethernet address 33:33:FE:11:22:33 is formed and the message is sent to this Ethernet address. This Ethernet address is referred to as the “**IPv6 neighbor discovery**” address.



annotation

Why are Ethernet broadcast messages not sent as with ARP?

Ethernet broadcast messages must always be read completely by the Ethernet hardware from the transmission medium, copied into the computer, and passed to the higher layer protocol, which then decides what to do with the message. However, if an Ethernet address starts with “3333,” the Ethernet hardware checks whether the last three bytes (here “112233”) match the last three bytes of its own Ethernet address.

- If they match, the message will be read completely and passed to the higher layer protocol.
- If they do not match, the message will not be read from the transmission medium. The computer then does not have to perform any unnecessary actions.



## 4.8.5 Automatic Address Configuration



arrangement

### 4.8.5 Automatic Address Configuration

#### 4.8.5.1 Basic Procedure

#### 4.8.5.2 Addresses for Autoconfiguration

The goal of **automatic address configuration** is to avoid manual configuration when a computer is connected to a network. In other words: An IPv6 host can be connected to a network without any manual configuration – all necessary addresses and settings are automatically configured (routers cannot automatically configure themselves; some parameters must be set by an administrator). Consequently there must be a procedure to assign a unique IP address to each interface. Interface IDs are used for this purpose.

**Small sites** with only a single link (without router) use the “link local” addresses that consist of the fixed prefix and the interface ID, e.g.

FE80::0280:25FF:FE00:3A3B

**Large sites** with multiple networks and routers must use “unique local unicast” or global unicast addresses. For this purpose, hosts must receive information about the prefix of the subnets to which they belong. Routers periodically generate ICMP messages (“router advertisements”) that distribute the current prefixes. From this, the hosts can construct their desired addresses by assembling the prefix and the interface ID into a valid “unique local unicast” or global address, e.g. the “unique local unicast” address:

FD00::0012:0280:25FF:FE00:3A3B

or the global address:

2030::0012:0280:25FF:FE00:3A3B


It is possible through this automatic address assignment to change all addresses at a site easily (“**renumbering**”). The prefix is changed; the interface IDs (and possibly the subnet number) can remain the same. This makes it possible to easily perform a provider change – today provider changes at large sites are avoided as much as possible because of the immense effort involved in allocating new IP addresses. It is necessary for this procedure to be able to assign a lifetime to the IP addresses and assign multiple (generally two) IP addresses to the interfaces. When assigning a new, second IP address, during a temporary transition phase, the old and the new IP address must be able to be used at the same time. Newly launched applications will only use the new address. On the other hand, applications that are already running cannot easily change their IP address

and therefore must continue to use the old IP address so, for example, when using TCP the IP address change does not lead to a termination of the connection.

If in the automatic address configuration, no other server is needed, this is called a “stateless” configuration; if a server is needed, it is called a “**stateful**” configuration. For the “**stateful**” configuration, mechanisms are used that are already known as DHCP from IPv4 networks and implemented in IPv6 as DHCPv6. DHCPv6 works similar to DHCP; however, for the identification of hosts so-called DUIDs (see RFC 6355) are used instead of MAC addresses. When administering a network, you can decide whether a “stateless” or “stateful” configuration or combination of the two should be used.



In the online version an video is shown here.

Link to video : <http://www.youtube.com/embed/qLxOWaLmoSs> 

IPv6 Management

### 4.8.5.1 Basic Procedure

The following steps must be performed during automatic configuration.

1. Generation of a “link local” address (hosts and routers)
2. Checking whether this address is already in use (hosts and routers)
3. Transmission of a prefix and other parameters from the router for generating the “unique local unicast” and the global address (hosts only)

First, a “local link” address is tentatively determined for an interface. Before it can be used by the interface, it must first be checked whether another interface is already using this address. To do this, an ICMP “**neighbor solicitation**” message is sent with an individual “link local” address. If another interface is using this address, an ICMP “**neighbor advertisement**” message is returned. This means that this “link local” address is already in use. Therefore, the automatic configuration cannot be carried out. In this case, the administrator must manually assign a new “link local” address. Usually there will be no response so the “link local” address can be assigned to the interface and used. The node can now communicate with all other nodes in its local network.



annotation

“Link local” addresses are valid for an indefinite period.

Routers regularly send information about prefixes and other parameters in ICMP messages of the “**router advertisement**” type. If a host does not want to wait for this message, it can prompt the router to send the “**router advertisement**” immediately by sending an ICMP “router solicitation” message. The router message specifies whether the prefixes transmitted in the message should be used for automatic address configuration (“stateless” configuration) or a DHCPv6 server must be addressed (“stateful” configuration). It also specifies whether additional information should be obtained, such as the MTU or the maximum hop limit. The host can construct its unique local unicast address and its global address from the prefixes.

There is also the option of stateless DHCPv6. Here the DHCPv6 server only provides additional information (DNS servers) but does not perform address management.



annotation

In practice, it is not so easy to choose and implement an appropriate configuration of IPv6 addressing; there are two pages with advice on this topic (see [🔗](#), [🔗](#)).

### 4.8.5.2 Addresses for Autoconfiguration

Let us take a closer look at which addresses are used in automatic configuration.

The MAC address of an interface is:

00-80-25-00-3A-3B

As we have seen from in the section Construction of Interface IDs, the following IPv6 interface ID is derived from this address:

0280:25FF:FE00:3A3B

The tentative “link local” address is formed by adding the prefix:

FE80::0280:25FF:FE00:3A3B

Now an ICMP “neighbor solicitation” message is sent to determine if the address is already in use. This message must be sent to all eligible nodes. IPv6 uses a special multicast address for this, which is formed using the last four bytes of the “link local” address (see Neighbor Discovery Protocol).

FF02::1:FE00:3A3B

This is a “link local” multicast address. It is not routed and can only be used in one's own local network. This address is called a “**solicited node**” **multicast address** because it is used by all nodes that have to respond to an ICMP “neighbor solicitation” message.

The ICMP “neighbor solicitation” message is sent to all hosts whose last four bytes are identical. The unspecified address ::0 is used as the source address. The tentative “link local” address is transmitted in the ICMP message.

Destination Address	“Solicited node” multicast address: FF02::1:FE00:3A3B
Source address	0
Hop limit	255
Next header	58: ICMPv6 message



IPv6 header of the ICMPv6 neighbor solicitation message

Begin printversion

Destination Address	“Solicited node” multicast address: FF02::1:FE00:3A3B
Source address	::0
Hop limit	255
Next header	58: ICMPv6 message

End printversion

Type	135: Neighbor solicitation
Destination address	Link local address to be checked: FE80::0280:25FF:FE00:3A3B



ICMPv6 neighbor solicitation message

Begin printversion

Type	135: Neighbor solicitation
Destination address	Link local address to be checked: FE80::0280:25FF:FE00:3A3B

End printversion

We now need to distinguish between what happens if the “link local” address already exists and what happens if the “link local” address is not yet in use.

#### “Link local” address already exists:

If the tentative “link local” address is already in use by another node, this node generates a “neighbor advertisement” message with the following content:

Destination address	All nodes multicast address: FF02:0:0:0:0:0:1
Source address	Link local address of source node
Hop limit	255
Next header	58: ICMPv6 message



IPv6 header for the ICMPv6 neighbor advertisement message

Begin printversion

Destination address	All nodes multicast address: FF02:0:0:0:0:0:1
Source address	Link local address of source node
Hop limit	255
Next header	58: ICMPv6 message

End printversion

Type	136: Neighbor advertisement
Destination address	link local address of source node



ICMPv6 neighbor advertisement message

Begin printversion

Type	136: Neighbor advertisement
Destination address	link local address of source node

End printversion

The tentative “link local” address may not be used because it is already in use by another interface. The automatic address configuration is aborted at this point. A new “link local” address must be assigned manually to the interface.



task

### Task: Identical IPv6 interface IDs

How can it happen that another end system in the network already has the link local address, which was tentatively formed from an Ethernet address that is globally unique?

#### Solution

The link local address, as we already know, is formed from the format prefix FE80 and the IPv6 interface ID, which is generated either from an Ethernet address or from an EUI-64 ID. The manufacturer code of an EUI-64 ID consists of five bytes, and that of an Ethernet address consists of three bytes, which are filled by the bytes FF and FE. If the company code (the first 3 bytes) and the bytes FF and FE happen to occur in the OUI address and the last three bytes of the OUI and Ethernet address are identical, there can be two identical IPv6 interface IDs within a network.



annotation

Why is the hop limit set to the maximum value of 255?

This is to prevent ICMP messages coming from another network. If this were to happen, they would have had to pass a router, which would have reduced the hop limit by 1. If the hop limit for the received packet is 255, the receiver can be sure that the packet came from its own local network.

**“Link local” address does not exist:**

If no “neighbor advertisement” message is received, it can be concluded that the link local address is not in use yet. This is the **usual case**. Now the tentative link local address may be used and assigned to the interface. Communication with all other nodes in the local network is possible from this point on.

Next the unique local unicast and/or the global address has to be formed. An ICMP “router solicitation” message is sent for this purpose:

Destination address	All routers multicast address: FF02:0:0:0:0:0:2
Source address	Link local address: FE80::0280:25FF:FE00:3A3B
Hop limit	255
Next header	58: ICMPv6 message



IPv6 header of the ICMPv6 router solicitation message

Begin printversion

Destination address	All routers multicast address: FF02:0:0:0:0:0:2
Source address	Link local address: FE80::0280:25FF:FE00:3A3B
Hop limit	255
Next header	58: ICMPv6 message

End printversion

Type	133: Router solicitation
------	--------------------------



ICMPv6 router solicitation message

Begin printversion

Type	133: Router solicitation
------	--------------------------

End printversion

A router immediately responds to a “router solicitation” message with an ICMP “router advertisement” message sent directly to the sender of the “router solicitation” message:

Destination address	Link local address: FE80::0280:25FF:FE00:3A3B
Source address	Router's link local address
Hop limit	255
Next header	58: ICMPv6 message



IPv6 header of the ICMPv6 router advertisement message

Begin printversion

Destination address	Link local address: FE80::0280:25FF:FE00:3A3B
Source address	Router's link local address
Hop limit	255
Next header	58: ICMPv6 message

End printversion

Type	134: Router advertisement
Information	including: <ul style="list-style-type: none"> <li>• Hop limit</li> <li>• Lifespan of the default router</li> <li>• MTU</li> <li>• Prefixes</li> </ul>



ICMPv6 router advertisement message



Begin printversion

Type	134: Router advertisement
Information	including: <ul style="list-style-type: none"><li>• Hop limit</li><li>• Lifespan of the default router</li><li>• MTU</li><li>• Prefixes</li></ul>

End printversion

The transmitted hop limit should be used when sending IPv6 datagrams (these can be sent from the start with values less than the maximum value 255). The lifetime indicates how long a router should be used as default router. The maximum lifetime is about 18 hours. If the lifetime is 0, the router is not used as the default router. The MTU of the connected interface is transmitted. The unique local unicast and/or the global address can be formed based on the prefixes.



annotation

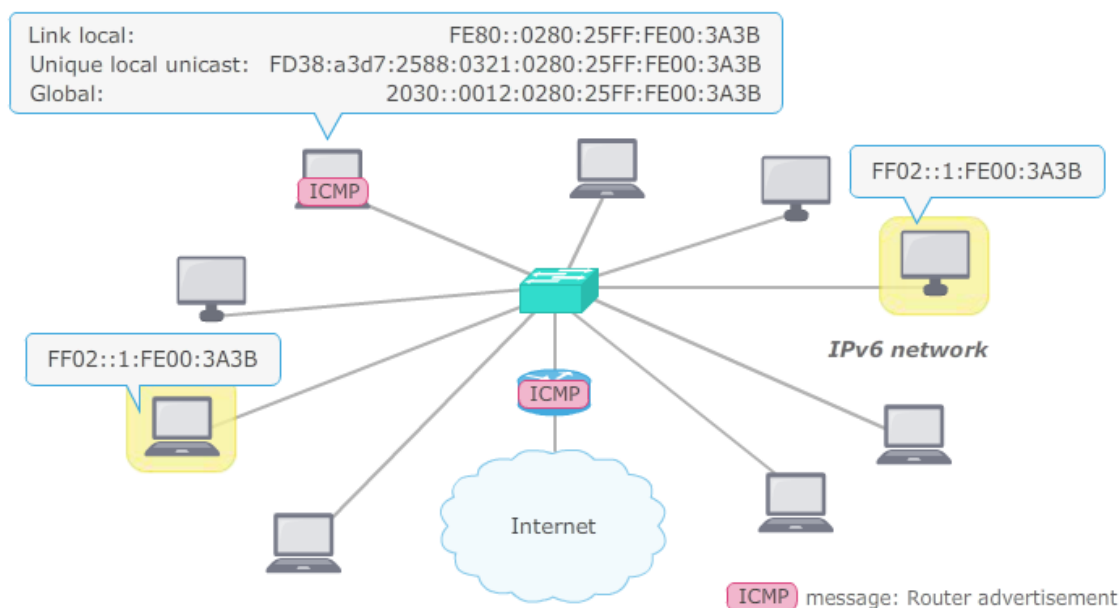
The following animation shows each step of the automatic IPv6 address configuration.



In the online version an animation is shown here.

#### Automatic address configuration

Begin printversion



A new host should be integrated into an IPv6 network. Its addresses are configured automatically in this process. For doing so, there is an attempt to generate a link local address for it based on the MAC address of its network interface. To be sure that no other interface in the network uses this address, an ICMP “neighbor solicitation” message is sent to all interfaces with the “solicited node” multicast address.

Since there is no “neighbor advertisement” reply in this example case, the host can use the link local address and can therefore communicate with all hosts in the local network.

The host needs network prefix information from the router to be able to generate the unique local unicast address and the global address. It therefore sends an ICMP “router solicitation” message to the router.

It replies with an ICMP “router advertisement” message.

Based on the prefix information, it is possible to construct the unique local unicast address and the global address. Now the host can communicate with all hosts in the site and globally.

End printversion

In addition to the requested messages, the router automatically generates “router advertisement” messages at certain intervals (e.g. every 600 seconds), which are sent to the “all nodes” multicast address. Thus, for example, when changing providers, changed prefixes are delivered so each node can construct new IP addresses (“renumbering”).

## 4.8.6 IPv6 Fragmentation

Unlike in IPv4, routers may not fragment IPv6 datagrams. Only the sender may fragment packets whose length is longer than the MTU of the interface to be used. The smallest MTU is set to **1280 bytes**. If the packet needs to be fragmented, necessary information for this is set in the fragment header, as can be seen in the following rollover element.



In the online version an rollover element is shown here.

### Fragment header

Begin printversion

0	8	16	24	31
Next Header	Reserved	Fragment Offset	Res	M
Identification				

field	description
Next Header	Kind of next header
Reserved	Reserved field
Fragment Offset	13 bits. The offset of the following data in relation to the start of the fragmentable part of the datagram (base unit: 8 bytes)
Res	Reserved field
M	M flag, 1 bit. 1 = more fragments, 0 = last fragment
Identification	32 bits. All fragments belonging together have the same identification

End printversion

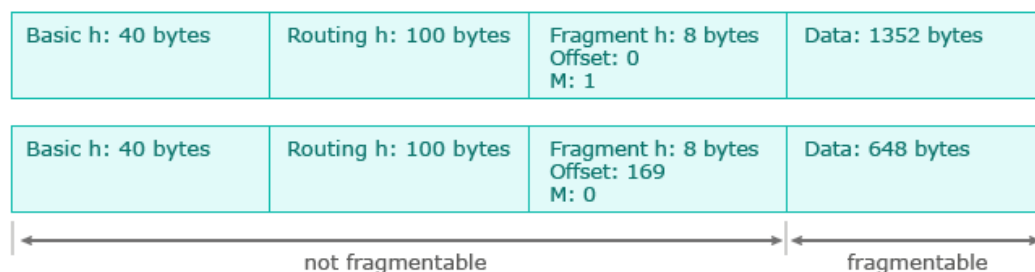
The original packet is divided into a non-fragmentable and a fragmentable part. The non-fragmentable part consists of the IPv6 base header, all expansion headers that must be interpreted by the routers, i.e., hop-by-hop options header, the destinations options header, and the routing header, as well as of course the **fragment header**. These headers are contained in every fragment.



example

### Fragmentation

2000 bytes need to be sent. The routing header has a length of 100 bytes. The fragment header is always 8 bytes long. With an MTU of 1500 bytes Ethernet), two fragments will be created:



Fragmentation example

### 4.8.7 Jumbograms

The payload length in the IPv6 base header specifies the size of the packet without IPv6 base header in bytes (including the expansion header). Because the field is only 16 bits long, a maximum payload of  $2^{16}-1 = 65,535$  bytes can be indicated.

However, IPv6 also allows to send larger packets. The length must then be specified in the hop-by-hop options header. The length of the payload is set to 0 in the base header. The length of the **jumbogram** (see RFC 2675) is specified in the hop-by-hop options header in a 32-bit long field; jumbograms can therefore have length between 65,536 and  $2^{32}-1 = 4,294,967,295$  bytes (without IPv6 base header).

If jumbograms are to be used, this involves the following changes with UDP and TCP:


**UDP** packets (including the UDP header) can have a maximum length of 65,535 bytes because the length field in the UDP header is a 16-bit field. For jumbograms, this length field is simply set to 0, and the actual length of the payload is taken over from IPv6.

There is no length specification in the **TCP** header; there is therefore no limit to the length of a single TCP packet. However, when establishing a connection the largest TCP packet to be sent is negotiated with the 16-bit MSS value, which therefore cannot be greater than 65,535. If the MTU of the interface is greater or equal to 65,535 bytes – 40 bytes (IPv6 base header) – 20 bytes (TCP header) = 65,475 bytes, the MSS is always set to 65,535. The actual MTU is set to the value determined by the path MTU discovery mechanism, which is then reduced by 60 bytes (length of the IPv6 base header and TCP header).

Jumbograms are designed for communication between supercomputers when there are interfaces with a very high bit rate. The MTU of the interfaces must be greater than 65,575 bytes (65,535 plus 40-byte IPv6 header length).



annotation

Although this definition of a jumbogram is still valid, the question of its practical relevance can be raised. The current MTU sizes at the Data Link Layer are significantly less than 10,000 bytes. Even so-called jumbo frames  (a non-standardized extension of Ethernet) have only an MTU size of 9000 bytes.

### 4.8.8 Mobile IPv6

The IP addresses must be known so that two nodes can communicate over the Internet. The IP addresses may not change as long as the nodes are exchanging data. If, however, nodes are used in a mobile way, it must also be possible to create connections when the mobile node is located in foreign networks and thus has a different IP address. This issue is even more serious if a mobile node enters a foreign network during data transmission and receives a different IP address. The transmission should not be disrupted in this case and the user should also not notice the network change.



annotation

The idea behind the technique presented in the following can be thought of as similar to forwarding mail at the post office. If you move, you can submit a request to the post office so they will forward your mail to your new address for a certain time period. Therefore, you can still be reached via your old, well-known address.

With **Mobile IPv6** (RFC 6275), mobile nodes can move to different networks while being addressed through the home address that was assigned during configuration. Data are thus routed to the mobile node regardless of the current network (and the current IPv6 address) of the mobile node. Communication is not interrupted then if the mobile node moves to another network. This technique is completely transparent for layers above IP and for applications – they do not notice anything.

If a mobile node is in its home network, packets are sent through the router using the usual routing mechanisms. However, if the mobile node is in another network under a different IPv6 address, the router must know this IPv6 address; otherwise, the packet cannot be forwarded to the mobile node. Therefore, the mobile node must tell

its new IPv6 address to the router in the home network when it connects to a foreign network. The new IPv6 address in the foreign network is called the **care-of address**. The mapping of the care-of address to the home address is called binding. The router in the home network, which knows the binding and through which the mobile node is always reachable is called the **home agent**.

Packets that are sent to the home address of the mobile node, are tunneled by the home agent (wrapped in IP packets) and forwarded to the mobile node. The mobile node can get the IP address of the correspondent node from the tunneled IP header and takes the IP address of the home agent from the header. If the mobile node has received the IP address of the correspondent node from the tunneled packet, the mobile node can tell its current care-of address to the correspondent node. This allows the mobile node and correspondent node to communicate directly with each other without needing to pass through the home agent. The mobile node transmits its home address in a destination options header. The correspondent node transmits the desired home address of the mobile node in a routing header. Direct communication bypassing the home agent reduces the network load and also increases security.

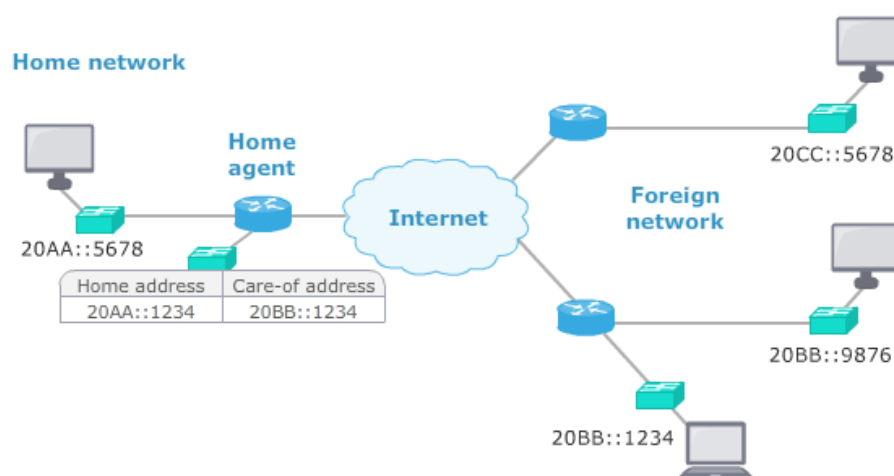
The following interactive element shows what happens if a mobile node leaves its home network while communicating with another node.



In the online version an animation is shown here.

#### Mobile IPv6

Begin printversion



A mobile end system, let us say a notebook, is currently part of its home network. It has an IP address from this network and is treated as a usual host. It can send data to other systems in the Internet via the router in its network. This router is the home agent in this Mobile IP scenario.

If the mobile end system moves to another network, it receives another IP address from it, the so-called “care-of address”.

This address is announced to the home agent via a binding update. The home agent then maps the current care-of address to the home address of the end system.

This procedure is called binding. The home agent sends a binding acknowledgement back to the mobile end system.


If the home agent receives packets for the mobile end system afterwards, it does not forward them to the local network. Instead, it sends them to the foreign network by using tunneling.

Once the mobile end system receives the first tunneled packet, it learns the IP address of the corresponding end system. Then it can send its care-of address to the corresponding end system via a binding update. The corresponding end system can react with a binding acknowledgement. In the following both end systems can directly communicate with each other without using the home agent.

End printversion



annotation

The practical relevance of Mobile IPv6 is not as high as you might assume. Although many mobile devices are used today, the applications on them are used as clients. In this case, it is not important that the IPv6 address is retained, even when the TCP connection might be aborted with a change. Imagine that a user is currently surfing in the Internet using WLAN and then moves outside of the building with her mobile device making a change of the wireless network necessary. The user can then continue to surf after a short interruption and will receive a new address to do this. The use of mobile IP also requires the operation of a home agent functionality on the router in the home network (see [instructions for Cisco routers](#) .

### 4.8.9 Summary - IPv6

Although the introduction of IPv6 according to specifications in the 1990s initially moved slowly, this has been changing in many areas towards the end of the 2010s. This can e.g. be seen in the [statistics for accesses to Google services](#) (see [also website for the world IPv6 launch](#)). For some topics, such as address assignment, it is not yet clear which methods will be regarded as best practice.



annotation

The add-ons [IPvFoo](#) for Google Chrome and [IPvFox](#) for Firefox show in the browser line whether the communication with a website is via IPv4 or IPv6. If you click on the red four or green six, you will also see the server IP addresses from which a website has been loaded.

More detailed information about IPv6 can be found in a [freely accessible course from Hurricane Electric](#).

## 4.9 Migration IPv6/IPv4



arrangement

### 4.9 Migration IPv6/IPv4

#### 4.9.1 Parallel Operation

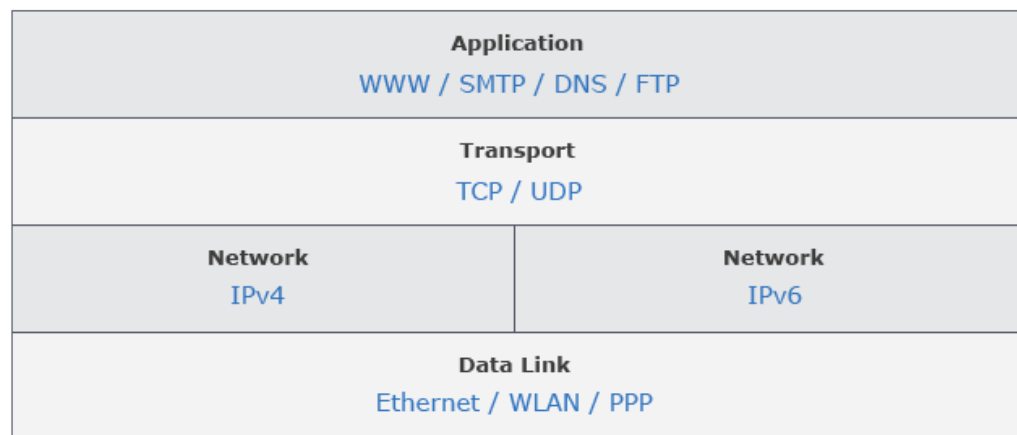
#### 4.9.2 Tunnel Configurations

#### 4.9.3 Translation Methods

There are several ways to migrate from IPv4 to IPv6. In this context, it is important to ensure that communication with other networks outside one's own network continues to be possible, even if they only use IPv4 or IPv6.

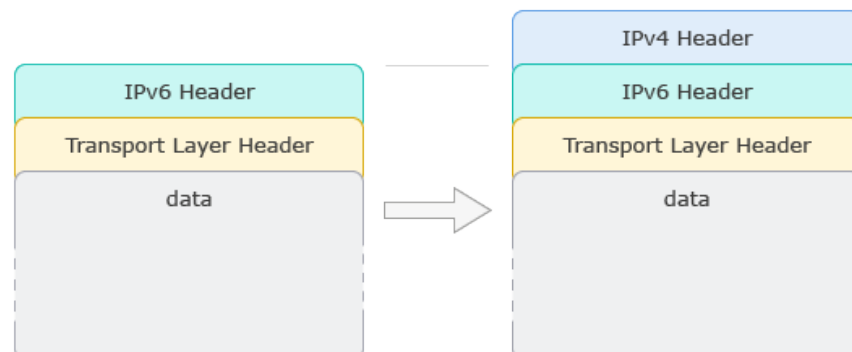
1. One obvious option is to continue to support IPv4 in addition to IPv6. This is referred to as a **dual stack** solution. Devices that are configured in this way initially examine an IP packet to determine whether it uses IPv4 or IPv6 and then process it accordingly.





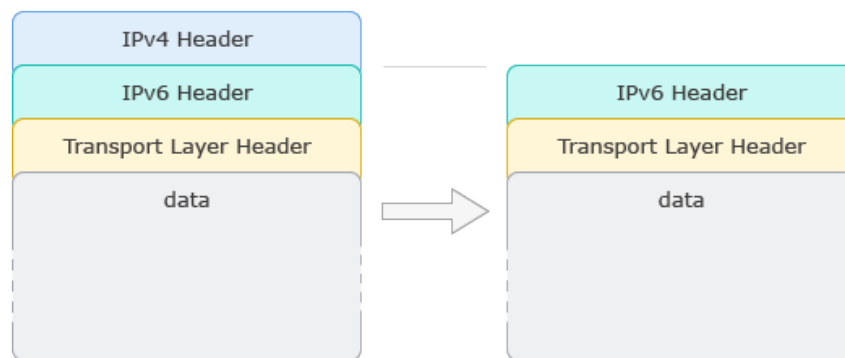
#### Host with dual IP stack

2. If an IPv6 host would like to communicate with another IPv6 host using IPv4 infrastructure, tunneling techniques can be used. With such a method, IPv6 packets are packed in IPv4 packets, routed over the IPv4 infrastructure and then unpacked.



#### Encapsulation of an IPv6 packet

The value 41 (IPv6) is set in the protocol field of the IPv4 header. When an IPv6 / IPv4 host or router receives a tunneled IPv4 packet addressed to IPv4 address, it removes IPv4 header. The packet is processed further as a normal IPv6 packet:



#### Decapsulation of an IPv6 packet

3. If an IPv6-only host wants to communicate with an IPv4-only host, a protocol translation must be performed. The IPv4 packets must be converted to IPv6 packets and vice versa.



notice

By the way, if you are looking for excuses for not carrying out a migration to IPv6, visit the [IPv6 bingo](#) website.

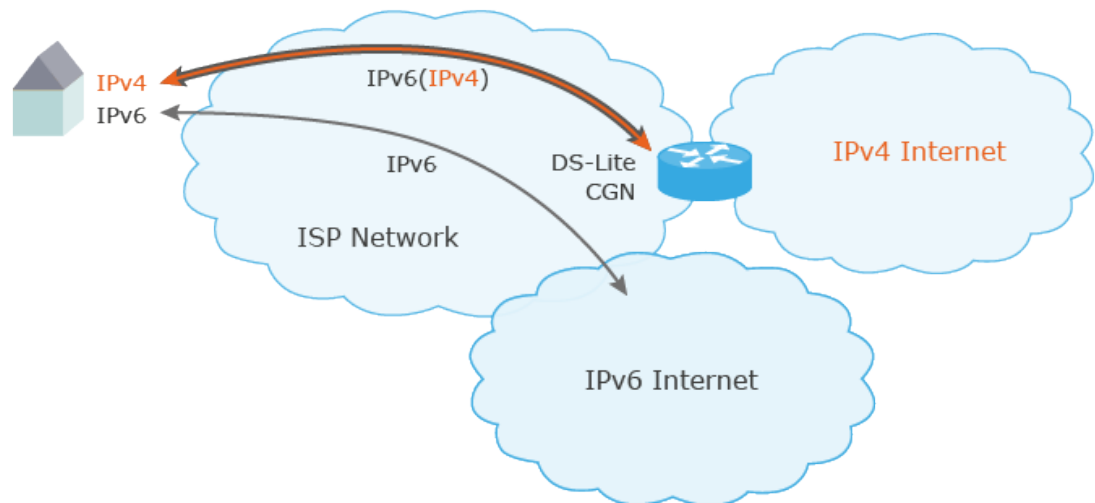
### 4.9.1 Parallel Operation

Parallel operation of IPv4 and IPv6 requires a lot of effort because both must be configured accordingly. It is also important to consider whether address management should be done for both versions in a similar way, which, however, is not in line with the new addressing design of IPv6 (for example, see a [blog entry by Jeff Doyle](#) on this).

As already mentioned, **dual stack** refers to operating IPv4 and IPv6 in parallel. Dual stack has been used on routers for many years (e.g. on the routers in the [Wissenschaftsnetz](#)) and is also common for servers. For clients, both IP versions can be implemented simultaneously, but this is not yet very common.

When connecting private customers to the networks of providers, the so-called **dual stack lite** configuration is often used today. This means that users are given a globally valid IPv6 address prefix, which allows them to communicate with the IPv6 Internet. However, the user is not assigned a globally valid IPv4 address because the provider does not have many of these addresses available. IPv4 packets from this customer network, which must use private IPv4 addresses, are tunneled via the DSL router or cable modem

over IPv6 into the provider network. The customer's device therefore does not perform NAT. The implementation instead occurs in the transition from the provider network to IPv4 networks and is called carrier-grade NAT. This results in restrictions for the user, who cannot easily offer server services because the implementation via carrier-grade NAT is not foreseeable for the user.

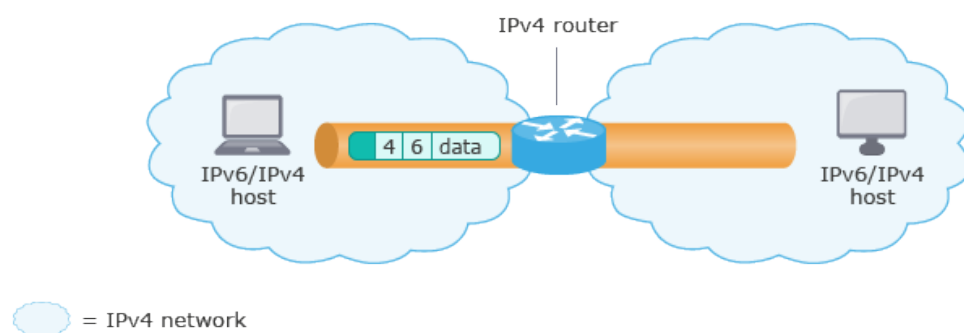


Dual stack lite

## 4.9.2 Tunnel Configurations

Various tunnel configurations are possible:

- A single IPv6/IPv4 host wants to communicate with another IPv6/IPv4 host over IPv6. Both hosts are operated in IPv4 networks:

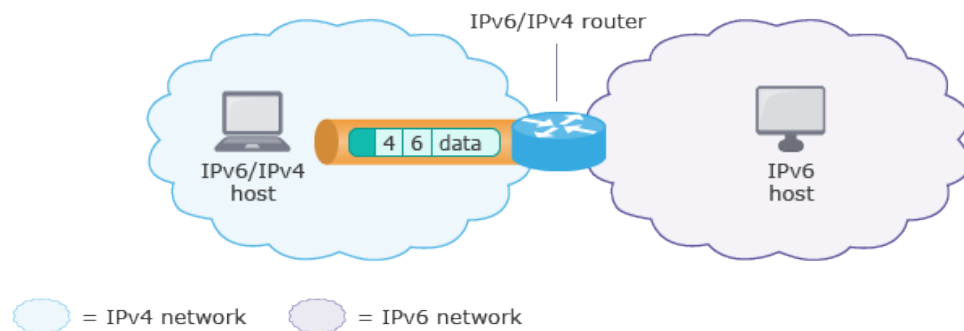




#### Tunnel between two IPv6/IPv4 hosts in two IPv4 networks

The sender packs its IPv6 packets into IPv4 packets and sends them to the IPv4 router, which in turn delivers them to the receiver. The tunnel starts at the transmitter and ends at the receiver.

- A single IPv6/IPv4 host wants to communicate with another IPv6 host that is operated in an IPv6 network:



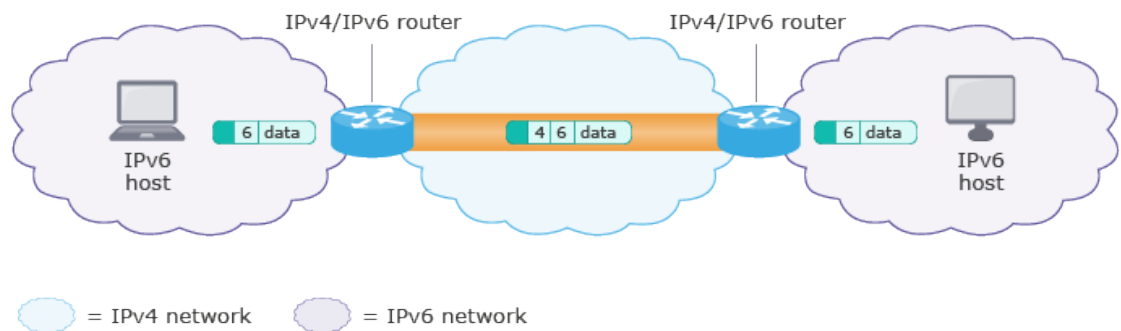
#### Tunnel between IPv6/IPv4 host and IPv6/IPv4 router in and IPv4 network

The sender packs its IPv6 packets into IPv4 packets and sends them to the IPv6/IPv4 router, which removes the IPv4 header and delivers the IPv6 packets to the receiver via IPv6. The tunnel starts at the sender and ends at the router.

So-called **tunnel brokers** (RFC 3053) are relevant for this scenario. Special servers are contacted instead of the depicted router; such servers can automatically accommodate tunnel requests that have been initiated by the user. Tunnel brokers are well-suited for individual IPv6 hosts as well as small IPv6 networks who want to make a connection to IPv6 networks via an IPv4 infrastructure.

ISPs often offer tunnel brokers. A [list of tunnel brokers](#)  is maintained on Wikipedia.

- Two IPv6 networks are connected via IPv4 transit networks:



#### Tunnel between two IPv6 networks over an IPv4 infrastructure

The transmitter sends IPv6 packets to the IPv6/IPv4 router, which encapsulates them into IPv4 packets and sends them to the IPv4 transit network. When leaving the transit network, the IPv4 header is removed and the IPv6 packets are sent to the receiver. The tunnel is built between both routers.

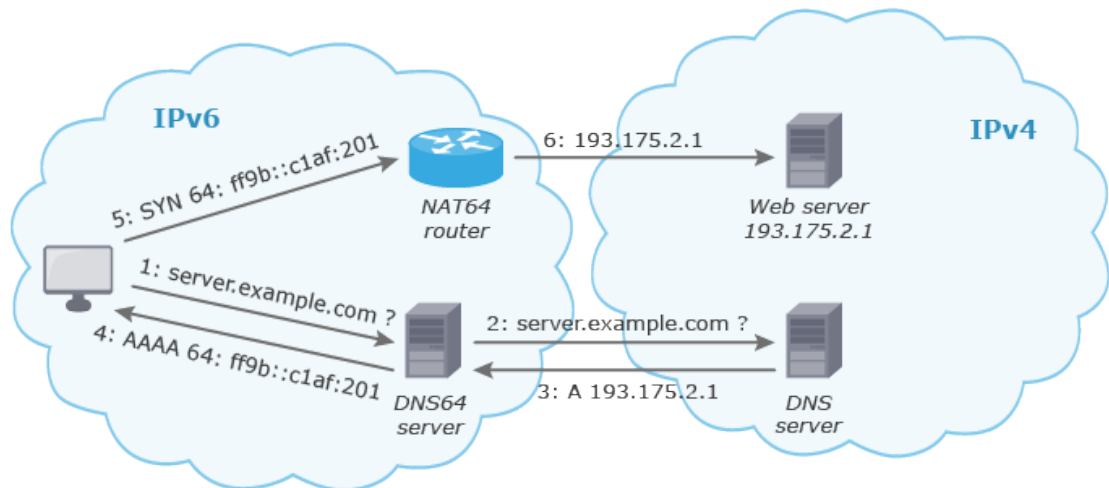
A large number of techniques and configuration options for tunneling has been developed in the last years (see [list on Wikipedia](#) [↗](#)). There are also other scenarios such as 464XLAT (RFC 6877), where IPv4 applications are operated over an IPv6 network (see [report on T-Mobile US](#) [↗](#)). This is also now true for Spotify (see [Spotify blog](#) [↗](#)).

### 4.9.3 Translation Methods

There are various options for communication between IPv6-only and IPv4-only hosts. We consider a scenario here where the aim is to make it possible for IPv6-only clients to access servers in an IPv4-only network. This allows you to shut down your own IPv4 network and no longer have to support parallel operation, but you can continue to communicate with IPv4 networks.

The configuration is called **NAT64/DNS64** and works as follows (see figure below). An IPv6-only client would like to communicate with a server that operates on the basis of IPv4. But the client does not know this, so the client must first perform a DNS (see [Domain Name System](#)) request to determine the IP address of the server. The DNS request is directed to a DNS server in the local network and thus occurs via IPv6; the client also asks for an IPv6 address. The DNS server used is a special DNS server that implements the DNS64 procedure. This means that it contacts the responsible DNS server but in doing so it notices that it is in an IPv4 network. It converts the request so that now only an IPv4 address for the server is being requested. After the IPv4 address is delivered, the DNS64 server translates it into a special IPv6 address. The IPv6 address space 64:ff9b::/96 is reserved for this purpose; the IPv6 address is

generated by inserting the IPv4 address in the last 32 bits behind the network prefix. The client receives this address as a response. Assume, for example that the IP address is 193.175.2.1, then the resulting IPv6 address would be 64:ff9b::c1af:0201.



Example of using NAT64/DNS64

The client now uses the IPv6 address to contact the server. The IPv6 packet then reaches a router that is now a special NAT64 router. This router recognizes that the special address prefix is being used and consequently converts the IPv6 address into the associated IPv4 address. To do so, it must omit the network prefix. In this way an IPv4 packet is created, which can be sent in the direction of the server and can therefore reach the server. The response from the server is then translated back by the router.

Public DNS64 servers [🔗](#) are operated by Google under the addresses 2001:4860:4860::6464 and 2001:4860:4860::64.

## 4.10 Routing Algorithms and Protocols



arrangement

### 4.10 Routing Algorithms and Protocols

#### 4.10.1 Static and Dynamic Routing

#### 4.10.2 Route Selection

#### 4.10.3 Overview of Routing Protocols

#### 4.10.4 Routing Information Protocol

4.10.5 [Open Shortest Path First](#)

4.10.6 [Border Gateway Protocol](#)

4.10.7 [Summary - Routing Algorithms and Protocols](#)

For the Internet to work, it is not sufficient to implement only the Internet Protocol. The routing tables in the routers must be maintained so that useful route selections can be made for IP packets.

Routing tables can be configured statically, but this is too inflexible for large networks. **Routing protocols** are therefore used. These enable the routers to exchange information with each other about the best paths. They can, for example, then react to network failures and find alternative paths.

In the context of routing protocols, you must take into account that the Internet consists of many independently managed networks. Such a network is called an **Autonomous System (AS)**. A distinction is then made between routing within an AS and routing between ASs. A provider can independently define within an AS how the routing should be configured. There are various routing protocols available for this purpose. However, the providers have to collaborate for the routing between ASs. Contracts with other providers have to be considered, and the exchange of routing information must be done in a standardized way. The BGP (Border Gateway Protocol) is used for this purpose in practice.

### 4.10.1 Static and Dynamic Routing

The routing tables in small, easy-to-handle networks with few routers can be maintained by the administrator. The routing information will rarely change. Often it remains the same over months or even years. The routing tables are loaded when starting a router and are no longer changed. This is called **static routing**. There are, however, some limitations with static routing:

- It cannot handle node or link failures
- It cannot take new link or devices into account
- It can only set the cost of a route once

A DSL router, for example, belongs to static routing especially since it only has the option to send the outbound packets to the provider.

As the networks become larger, other methods have to be used, which are able to react dynamically to changes in the network. Mechanisms have to be implemented here, which are realized in routing protocols. Routers use routing protocols to exchange information with each other automatically regarding the reachability of their attached networks and the resulting optimal paths. This entire field is called **dynamic routing**. It has the following characteristics:

- Automatically determines the best route - the shortest path - out of several alternatives
- Can adapt to changed conditions - such as new or failed links
- Can often distribute the load across multiple routes
- Saves costs due to less effort in administration

Before providing an overview of the common dynamic routing protocols, this section will present principles that are applied by all routing protocols.

## 4.10.2 Route Selection



arrangement

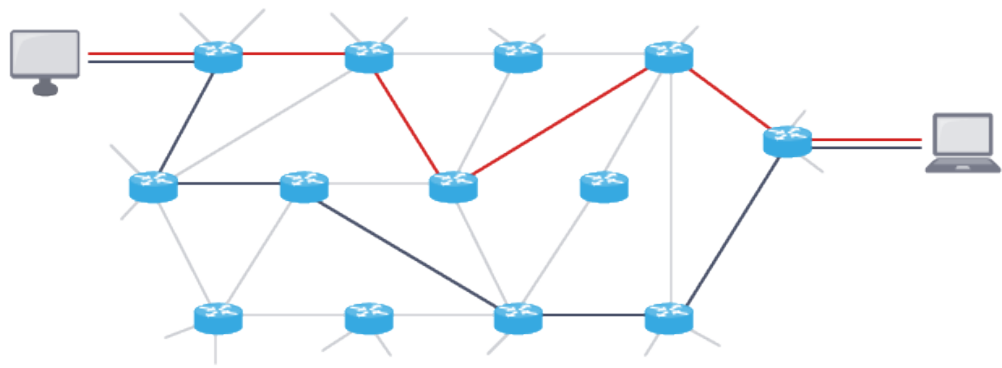
### 4.10.2 Route Selection

#### 4.10.2.1 Route Selecting Principle

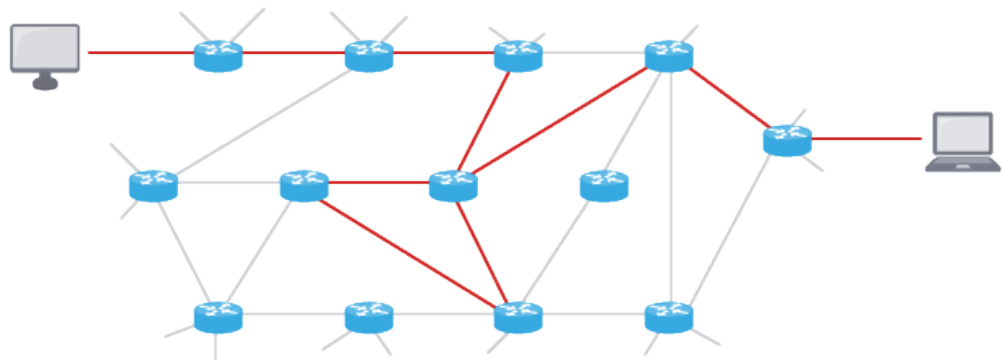
For the efficient transmission of packets from the source to the destination, the network nodes must select a suitable path from the multitude of possibilities. To do this, they need both information, which is used to get evaluation criteria for the selection of the route, as well as suitable algorithms for the selection process itself.

If we look at network part depicted in the following figure, several paths can easily be selected between the two shown end systems A and B. An example can be found in the following figure, which depicts two possible routes from end system A to end system B:

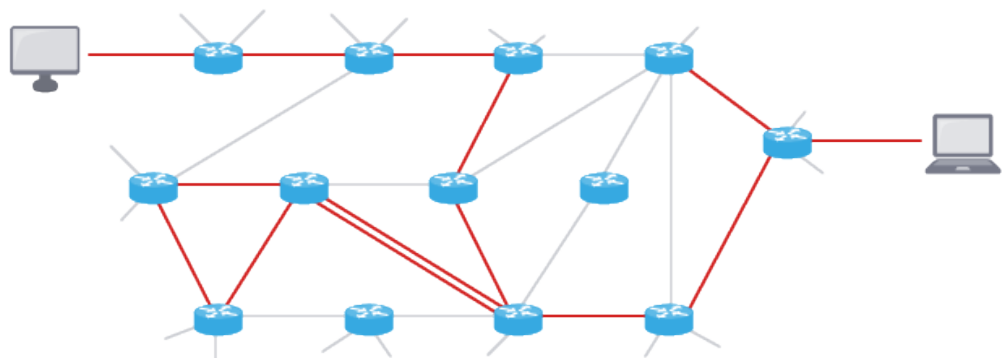


**Route selection in a network – with two examples**

Intuitively, some possibilities will be excluded from the candidate set when it comes to finding an optimum path. Hardly anyone would describe the following path as an efficient solution:

**Route selection in the network – with nodes visited twice**

In example 2, a node is passed through twice. In example 3, links are used several times.





#### Route selection in the network – with links visited twice

### 4.10.2.1 Route Selecting Principle

As we will see in the following, routing protocols are based on algorithms for graphs. The networks are modeled as graphs for this purpose.

The depicted network thus represents an “**undirected graph**”  $G = (V, E)$  which consists of a set of vertices  $V$  (or nodes) and a set of edges  $E$ , which can be depicted as unordered pairs  $(x, y)$  where  $x$  and  $y$  are nodes of a network.

An “edge sequence” from  $x_1$  to  $x_n$  is a sequence of edges  $(x_1, x_2) (x_2, \dots) \dots (\dots, x_n)$ . An “edge chain” is an edge sequence in which all edges are distinct from each other. A “path” is an edge chain in which all vertices are distinct from each other. If we consider the part of a communication network, an edge represents a physical connection between two nodes. If it is possible to realize the connection via a “path” for the communication between two nodes, the problem using edges twice would be solved.



annotation

How can we make a selection if several alternative routes are available?

What do we do in our daily lives when we choose a route with the help of a road map (if still want to get along without a navigation system)? We apply a valuation factor that enables a direct comparison between multiple routes. This factor can, for example, be distance, the type of road connection (interstate, highway, country road, etc.), or the anticipated gasoline consumption. In general, we look at costs (time, money,...) as a way of evaluating the effort associated with a particular path. Often, such a cost estimate can be made for the effort expended for communication via an edge.

A “**weighted graph**” is thus an unordered graph including a cost function. The job of optimization is accordingly to minimize the costs expended between the two nodes. If you look at the costs as the distance between the nodes, optimization just means **calculating the shortest path**; here the “length of the path” from  $x_1$  to  $x_n$  is understood as the sum of the weighted edges along a given path.

## 4.10.3 Overview of Routing Protocols



arrangement

### 4.10.3 Overview of Routing Protocols

#### 4.10.3.1 Routing Table

In order to make routing decisions, routers periodically exchange routing information as needed with other routers. This information must be used to calculate which paths are the best.

Some general considerations should be fulfilled by the routing protocols:

- It is desirable that a router supports different types of costs (**metrics**) for path selection such as **distance**, **pecuniary cost**, **throughput**, **delay** and connection **security**.
- If there are several best routes to choose from, these paths should be able to be used at the same time to distribute the load across different paths (load balancing).
- Path changes should be announced as soon as possible to all affected routers.
- Finally, it is desirable that the network load caused by the communication between the routers is as low as possible.

These requirements are met in different ways by the routing protocols.

Several protocols are available as alternatives within autonomous systems.

- The **Routing Information Protocol (RIP)** is based on the basic principle of distance vector routing. This means that the routers only know the costs for reaching destination networks and the respective next router to be used. This limited knowledge where the complete paths are not known can lead to difficulties if paths need to be recalculated. Because how well this protocol works depends on the situation, it is not suitable for use in large networks.
- The **Open Shortest Path First (OSPF)** protocol works on the basis of the link-state method. When using OSPF, each router knows the complete network. Distributing this knowledge is more complex than with the distance vector method, but it allows to get a consistent view on the network again in the event of changes significantly faster than with the other method. The problematic situations that occur during the use of the distance vector method can thus be avoided. OSPF is therefore a good choice for the task.

- **Intermediate System to Intermediate System (IS-IS)** is a routing protocol that is also based on the link-state method. Overall it is very similar to OSPF. The existence of both protocols is due to the history of two working groups working on two protocols in parallel. IS-IS has the special feature that it comes from the OSI world. Therefore, routers exchange information with each other on the basis of OSI protocols and with OSI addresses. Compared to OSPF, IS-IS is less “chatty”, which means that less data are exchanged over the protocol. Because IS-IS is used in many provider networks, vendors often implement new features are first for IS-IS.
- The **Enhanced Interior Gateway Routing Protocol (EIGRP)** is a combination of the distance-vector method and the link-state method. For many years the protocol was Cisco-proprietary so that you were dependent on this manufacturer when using it. However, in 2013 the protocol was disclosed in RFC 7868.

The **Border Gateway Protocol (BGP)** must be used for routing between AS. It is based on the path-vector method which is related to the distance-vector method. However, here not only the costs to the destination are known, but the respective complete path to the destination is also known. It is also important to note that the paths are considerably abstracted. They are not described as a sequence of routers but as a sequence of AS. Which router is used in an AS is therefore not considered.

### 4.10.3.1 Routing Table

As we have already learned, all routers have a routing table. The route that is to be used to forward a data packet is selected on the basis of the entries in the routing table. To select the next router, only the destination address/prefix and the router address are used: A packet is sent to the next router on the basis of its destination address.

The most important entries in the routing table are:

- Destination address/prefix
- Router
- Interface
- Metric

The **netmask** can be used instead of the **prefix**: Both provide the same information; namely, how many bits of the destination address should be evaluated by the router.

For each destination, the **router** through which it is to be reached is specified (for historical reasons, routers are sometimes also called gateways).

For each entry in the routing table, an **interface** is defined through which the specified router can be reached. An IP address is assigned to each interface.

There may be several paths to reach the destination address: A path is selected based on the **metric**.

#### 4.10.4 Routing Information Protocol

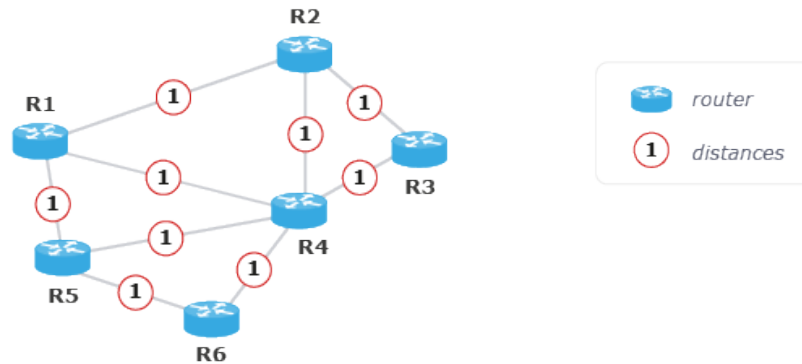
RIP uses an algorithm that falls under the **distance-vector** algorithm class. The theoretical considerations were carried out at the beginning of the 1960s, long before routers were used, and are known as the **Bellman-Ford algorithm**.

The metric in the routing table can generally be viewed as the **distance** of a destination from the router. Distance-vector algorithms received this name because they allow to determine optimal paths only on the basis of exchanges about distances information. In addition, this information need only be exchanged between **adjacent routers**.

Messages are exchanged between adjacent routers, which include information about the current distances of each router to the destinations. Pairs of “**destination: distance**” values are thus exchanged between the adjacent routers, which are referred to as **distance vectors**. Optimal paths can be determined for the entire network based on these data.

It can be shown that the algorithm converges and yields the optimum result in a finite time. No assumptions are made when the distance information is exchanged; each router can send the information on the basis of its own clock. Information may also be lost – but not all of it may be lost. The algorithm makes no assumptions about initial conditions, which means that the algorithm is suitable for dealing with changes in the network. When a change has occurred, the algorithm finds a new optimal state within a finite time.

The algorithm is simple to implement: a router gets distances to a destination from all its neighboring routers. The router adds the distance to the neighboring router to these distances and thus gets its own distance to the destination. If several possible paths are available, the path with the smallest distance is accepted. It may, however, be the case that the initial distance is too small. There must therefore also be a way to increase the metric: if a distance is reported by a neighboring router, which is already in the routing table for the destination address, the distance will always be accepted - whether it is small or large.



Example of a network with distances (all distances are one in compliance with the hop count metric)

RIP always simply uses a counter as the distance metric; it indicates how many routers a packet must be routed through - this is why the metric is also called “**hop count**”.

The figure above shows an example network where all costs are one which matches to the hop count metric of RIP. The corresponding routing table for router R1 is:

Destination	Next router	Distance
R2	R2	1
R3	R2	2
R4	R4	1
R5	R5	1
R6	R5	2



Routing table in router R1

Begin printversion

Destination	Next router	Distance
R2	R2	1
R3	R2	2
R4	R4	1
R5	R5	1
R6	R5	2

End printversion

For RIP, the distance vectors of a router are sent to the adjacent routers every 30 seconds.

RIP is a very simple protocol, which however has some disadvantages:

- Only the hop count can be used as the metric.
- The convergence time of the algorithm after a router change is relatively long depending on the situation and can take several minutes.
- No load balancing is possible because exact paths to a destination are always selected.
- All distance vectors are always sent.

## 4.10.5 Open Shortest Path First



arrangement

### 4.10.5 Open Shortest Path First

#### 4.10.5.1 OSPF Example

OSPF (RFC 2328, version 3 for IPv6 in RFC 5340) is a so-called **link-state routing** protocol designed for use within an AS. Each OSPF router receives the same data base that describes the topology - the link states - of the **entire autonomous system**. From this data base, the shortest paths are determined using the Dijkstra algorithm and the routing tables are constructed. The link-state data base contains the local status of each router – which neighbors can be reached at what cost. Because each router requires the same data base, this information must be shared between all routers. The messages about the state of the connections used for this are called **link-state advertisements (LSA)**.

OSPF calculates new routes very quickly, minimizing traffic. The same algorithm runs on all routers: From the data base, a **shortest path first (SPF) tree** is constructed, starting from the current router. The SPF tree specifies the paths to each destination within the AS. If there are several paths with the same costs, the network traffic is evenly spread over these paths (**load balancing**).

Large OSPF networks can be divided into independent **OSPF domains**. The topology of a domain is not known in the rest of the AS. This can significantly reduce routing protocol traffic. The shortest paths are only determined within an area. Possible errors cannot spread then throughout the entire AS, but from an global point of view there are no optimum routes across the areas. Nowadays such a division into areas is in many cases no longer necessary since the calculations and previously necessary data exchanges are also possible in networks with more than 100 routers.



In the online version an video is shown here.

Link to video : <http://www.youtube.com/embed/JOW2FSpAZqk>

Link State Routing

### 4.10.5.1 OSPF Example

The following example shows how the shortest paths are determined by OSPF.



example

#### Determination of the shortest path with OSPF

We will use this example to explain the steps of this process. Let us start with the representation of the network as a weighted graph, as shown in the following animation. The network has eight routers connected several links. The digits on the links represent the costs, i.e. the distance between the routers.

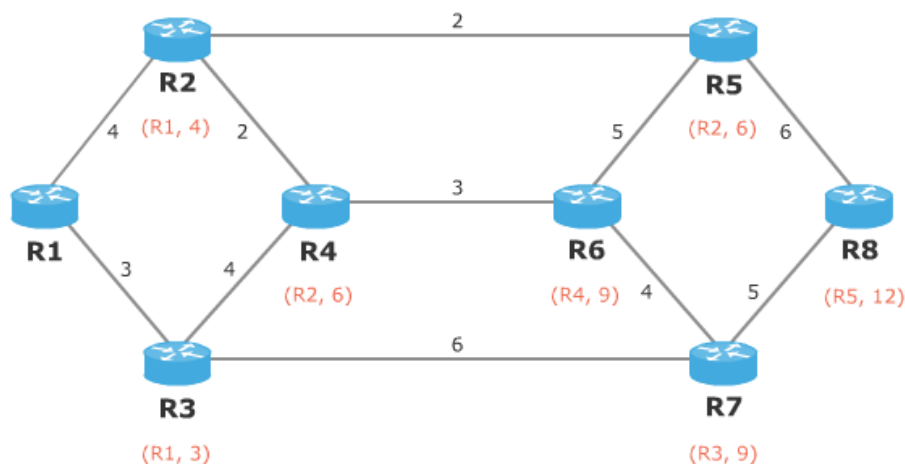
Now the shortest path to all other routers is determined for router 1 of this network:



In the online version an animation is shown here.

#### Determination of the shortest paths

Begin printversion





The shortest paths from R1 to all other routers in the network shall be determined.

We start with router R1 and go to the neighbor routers R2 and R3. We take note of their distances to R1: If we go from R2 to R1, we have the distance 4; if we go from R3 to R1, we have the distance 3.

The router with the lowest distance is selected and marked, that is R3. If routers have been marked, they are not considered anymore in the following because optimum paths for them have been found and cannot be improved.

Now we go from the last marked router R3 to its neighbor routers R4 and R7 and determine their ways to R1. All not yet marked routers, that is R4, R7, but also R2 are considered.

The least distance router is selected and marked, this is R2.

We go again from the last marked router R2 to its neighbor routers R4 and R5 and take note of their distances to R1. The new way from R4 via R2 is better than the old way from R4 via R3. Therefore, the cheaper way is noted for R4.

The router with the lowest distance is selected and marked. At this point, there are two possibilities: R4 or R5 can both be selected because they have the same distances. We select R4.

We now go from the marked router R4 to the only neighbor router R6 and take note of its distance to R1.

The router with the lowest distance is selected and marked. At this time, it is R5.

We go from router R5 to its neighbor routers R6 and R8 and take note of their distances to R1. Since the new way from R6 via R5 with the distance 11 is worse than the old way via R4 with the distance 9, the entry for R6 is not changed. The router R6 with the lowest distance is selected and marked.

We go from R6 to the only neighbor router R7 and take note of its distance to R1. Since the way from R7 via R6 with the distance 13 is worse than the way via R3 with the distance 9, the entry for R7 is not changed. Router R7 has the way with the lowest distance and is marked.

We go from router R7 to its only neighbor router R8 and take note of the distance to R1. Since the way from R8 via R7 has the distance 14 and is not better than the way via R5 with distance 12, the entry for R8 is not changed. And again a least cost router is selected and marked, this time R8.

Now all routers have been marked and therefore the shortest paths to R1 have been determined.

End printversion

This allows us to create the tree of the shortest paths - the SPF tree. The SPF tree can be displayed in a tabular or graphical form. First it will be displayed in tabular form. The results determined by the optimization algorithm are used for this.

<loop\_noprint>

Router	Predecessor	Distance
R1	-	0
R2	R1	4
R3	R1	3
R4	R2	6
R5	R2	6
R6	R4	9
R7	R3	9
R8	R5	12



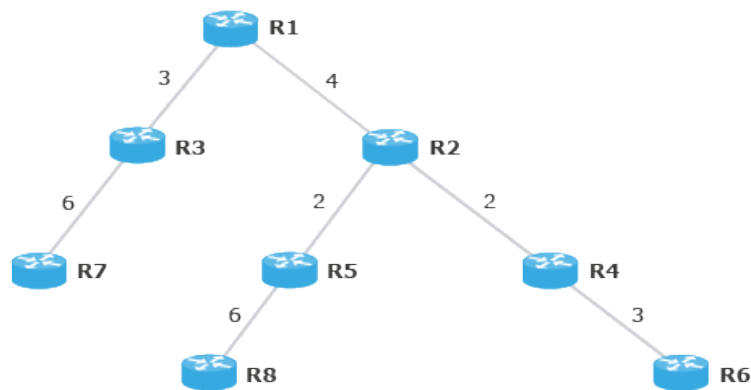
SPF tree for R1

Begin printversion

Router	Predecessor	Distance
R1	-	0
R2	R1	4
R3	R1	3
R4	R2	6
R5	R2	6
R6	R4	9
R7	R3	9
R8	R5	12

End printversion

Graphically, the SPF tree looks as follows, where the distance between the individual routers are provided and not the total distances from the root router as in the table:



SPF tree for R1

The routing table for R1 can be determined easily based on the SPF tree:

<loop\_noprint>

Destination	Router	Distance
R2	R2	4
R3	R3	3
R4	R2	6
R5	R2	6
R6	R2	9
R7	R3	9
R8	R2	12



Routing table for R1

Begin printversion

Destination	Router	Distance
R2	R2	4
R3	R3	3
R4	R2	6
R5	R2	6
R6	R2	9
R7	R3	9
R8	R2	12

End printversion

The example shows the SPF algorithm principle. The actual algorithm used by OSPF is more complicated and has to consider not only routers but also the networks, OSPF areas and external networks.



task

### Comparison of SPF algorithm and spanning-tree protocol

Is there a relation between the SPF algorithm and the spanning-tree protocol?

#### Solution

In both methods, you will end up with a minimum spanning tree which reaches all destinations with minimal costs.

However, the tree in the spanning tree is valid throughout the entire network, so there is a unique root bridge. An SPF tree, on the other hand, is calculated internally by each router, with itself as a root node.

## 4.10.6 Border Gateway Protocol



arrangement

### 4.10.6 Border Gateway Protocol


#### 4.10.6.1 Definition of Autonomous Systems

#### 4.10.6.2 Autonomous Systems Scenarios

#### 4.10.6.3 Hierarchy of Routing Protocols

The **Border Gateway Protocol** is used to exchange routing information between routers in different **Autonomous Systems**. To do this it uses a path-vector algorithm where the paths to the destination networks are known. These are described as sequences of ASs.

BGP is used by every provider to exchange route information with other providers. These routes are initially known only on one BGP router but not on the other routers within the AS. BGP is also used within ASs to forward this information. In other words, the routers implement BGP in addition to the internal routing protocol.

Some interesting statistics on BGP are provided on a [webpage by Geoff Huston](#)  (chief scientist at the Asia Pacific Network Information Center).

### 4.10.6.1 Definition of Autonomous Systems



annotation

What is an **Autonomous System**?

Originally an AS referred to a group of routers

1. that were administered together
2. that used the same routing protocol (e.g. RIP or OSPf) and
3. a common metric (e.g. hop count) and
4. a special protocol (BGP) to reach other AS.


Today, different routing protocols with different metrics can be used in an AS, so the definition of an AS is more generalized (see RFC 1930):



definition

#### **Autonomous System**

An autonomous system is a group of one or more connected IP prefixes for which one or more organizations is responsible and for which there is a single and clearly defined routing policy. A routing policy defines how the routing decisions are made.

**Autonomous systems numbers** are assigned to distinguish the ASs. They are managed like the IP address areas in a hierarchical structure with IANA at the top. Previously ASNs had 16 bits, but some 32 bit numbers have already been defined (see [list at IANA](#) 

because the 16-bit ASNs are already used quite a lot like IPv4 addresses. Similar to IP addresses, there are also private ASNs that can only be used within one's own AS.

A view on AS in reality is provided by a [map from CAIDA.org](#) .




### 4.10.6.2 Autonomous Systems Scenarios

An AS can be used to forward data for another AS; this is referred to as transit traffic. Depending on how an AS deals with transit traffic, it can be characterized as the one of following.

- **Stub AS**, which has only a single link to another AS and only has local traffic (i.e., traffic that has a source and/or destination in this network)
- **Multihomed AS**, which has several connections to other ASs but does not permit transit traffic and therefore has only local traffic
- **Transit AS**, which has several connections to other ASs and can have transit and local traffic

This explanation is somewhat simplified: In the case of a “multihomed AS,” it is possible to decide whether traffic should be passed between certain ASs but not between all ASs; each AS can also decide which other AS to send its own data through.

The networks of ISPs are usually transit AS, while stub AS and multihomed AS are networks of their customers. Customers, for whom the connection to the Internet is very important, are increasingly opting for multihoming, so their Internet connection is realized independently by two providers. In this way, the customer network remains connected to the Internet, even if there are problems with the connection to one provider.

In the context of these scenarios, it is also important to understand that economic considerations are the primary basis of decisions. Customers have to pay money for the use of their Internet access, and there are also different ISP sizes, with the so-called [Tier 1 ISPs](#)  at the top. The Tier 1 ISPs operate world-wide networks and manage the global routing table over which any network can be accessed. So-called **peering** agreements are often concluded between ISPs, which regard each other as equivalent. This means that the ISPs send data traffic to each other while respecting contractual boundary conditions but do not pay for it. The networks of ISPs are often connected to large Internet exchange points (e.g., [DE-CIX](#) , [AMS-IX](#) ) so the data exchange between networks occurs there.



annotation

The following exercise consists of two parts. Read about the exercise and place the appropriate checkmarks on the networks. You can then check your results with the help of the evaluation function on the bottom right. If the results are correct, you can move on to the second part of the exercise by pressing the “next” button. You also have the option of starting the exercise again from the beginning by pressing “retry.” It should take about 5 minutes to complete.

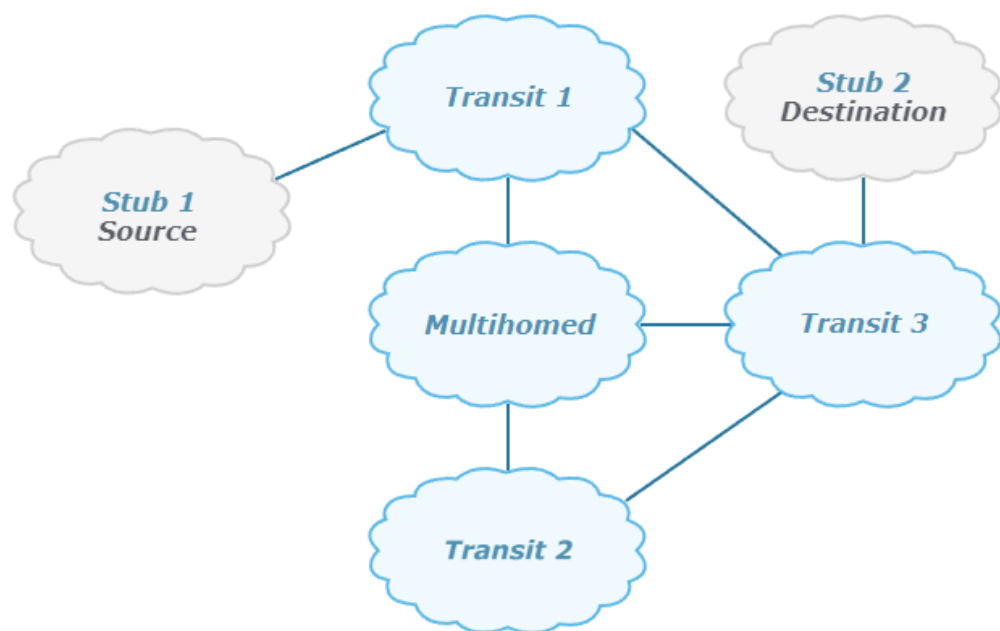


In the online version an click interaction is shown here.

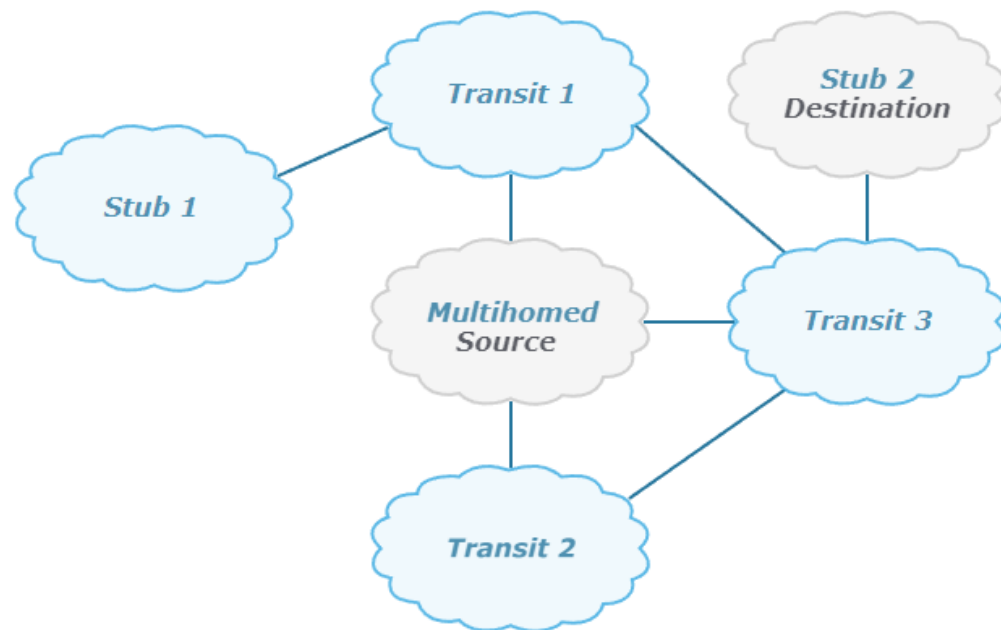
#### Difference between multihomed and transit autonomous systems

Begin printversion

**TASK 1:** Mark those Autonomous Systems which can be used for data transfer if the data source is in AS Stub 1 and the destination in AS Stub 2.



**TASK 2:** Mark those Autonomous Systems which can be used for data transfer if the data source is located in the Multihomed AS and the destination is part of AS Stub 2.



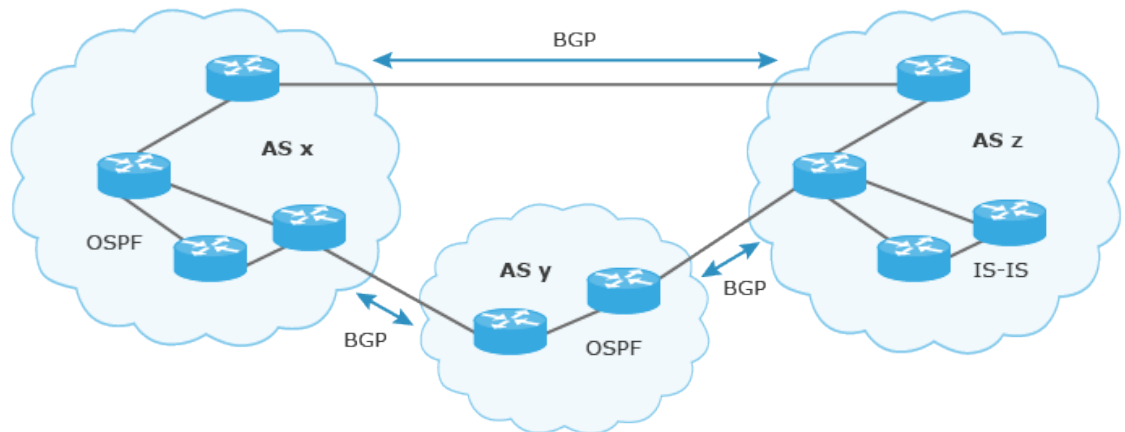
End printversion

### 4.10.6.3 Hierarchy of Routing Protocols

The Internet can be viewed as an uncontrolled composition of Autonomous Systems, each using its own routing protocols, between which the exchange of routing information takes place in a generally known way (specifically, with BGP). This gives us a **hierarchy of routing protocols**. Communication between the ASs is therefore the upper level of the hierarchy; routing within the AS is below it. We have also seen by considering the OSPF areas that further subdivisions are also possible within the AS.

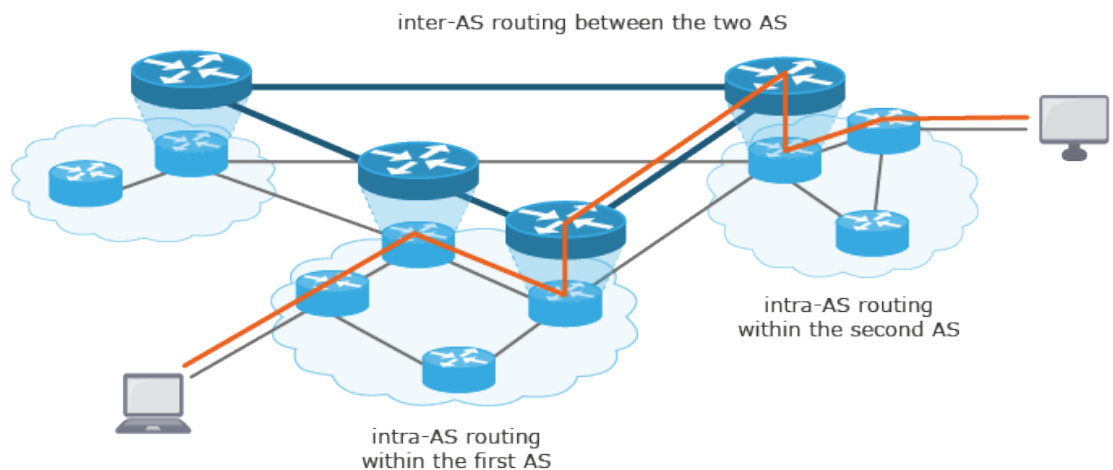
The figure shows three ASs. BGP is used between them. It has been decided to use OSPF within AS x and AS y. In contrast, AS z uses IS-IS internally.





#### Use of routing protocols in and between ASs

If you consider how the resulting path in the Internet look like (see figure below), you can see that there is a mixture of technical and financial optimization. Within an AS, the paths are selected according to technical considerations (e.g. low latency or minimum utilization). The routing between ASs on the other hand is done primarily on the basis of the financial considerations of the providers.



#### Overall path as a result of routing in and between ASs

### 4.10.7 Summary - Routing Algorithms and Protocols

This chapter has explained the following important terms and concepts that are used in today's routing protocols:


- The difference between static and dynamic routing

- The principles for determining an optimal path
- An overview of requirements for routing protocols
- The principles of RIP, OSPF and BGP
- Dijkstra's shortest path algorithm
- The definition of autonomous systems

## 4.11 Exercises - Network Layer



task

In the following it is assumed that your Internet connection is based on IPv4. You can verify this by accessing the test page [WhatIsMyIPv6](#) .

### *Tasks for Beginners*

#### **Task 1:**

Use the command line tool ping to check the availability of [www.fh-luebeck.de](http://www.fh-luebeck.de). Use Wireshark to record the packets that are generated.

#### **Task 2:**

Use the command line tool traceroute (it is called "tracert" in Windows) to examine the path to [www.fh-luebeck.de](http://www.fh-luebeck.de).

#### **Task 3:**

ARP requests are contained in the provided Wireshark file. After applying a filter just to get the ARP requests, investigate the mappings of IP addresses to MAC addresses.

#### **Task 4:**



The Wireshark file shows assignments with the DHCP protocol. Use this to determine which IP addresses, netmasks, default gateway and DNS servers have been assigned.

### *Tasks for Advanced Learners*

#### **Task 1:**

Fragmented IP packets are contained in the Wireshark file. Investigate the fragmentation details.



#### **Task 2:**

Network providers such as [Hurricane Electric](#)  offer so called looking glasses which allow to retrieve some information from network components and to execute tests. Investigate which possibilities are offered by the looking glass of the European research backbone [GEANT](#) .




**Task 3:**

The routing protocol OSPF is active in the network provided via an eNSP file. Investigate the routing tables as well as the OSPF data units which can be visualized by Wireshark.


**Task 4:**

You can get information about autonomous systems with the [ASnumber](#)  Firefox AddOn. It shows the AS that the web page server belongs to. Since the AddOn is a bit older, it is necessary to install the [Classic Theme Restorer](#)  AddOn so that ASnumber is shown on the right side of the bottom line.

**Task 5:**

Information about Autonomous Systems, BGP use and the global routing table are available on various sites in the Internet. [RIPE](#)  provides a tool named BGPlay to visualize the neighborhood relations of ASs. Information about ASs and their peerings is available at [PeeringDB](#) . [Potaroo.net](#)  provides statistical data about the global routing table.



**Special Tasks about IPv6****Task 1:**

Google provides [statistics](#)  whether the access to its services is based on IPv6. Take a look at these statistics. You can also see the statistics on a per-country basis.

**Task 2:**

In this task a small IPv6 network which is configured via the eNSP tool should be investigated. Copy/paste the contents of the router configuration file to the router CLI. The switch does not have to be configured. Perform ping tests in the network (the command is just ping, not ping6) and check the client configuration via ipconfig. As an alternative, a Wireshark capture and several screenshots are provided if you cannot run eNSP.

**Task 3:**

As mentioned before, it is assumed that you communicate from home via IPv4. To examine the practical use of IPv6 it is therefore necessary to configure a tunnel. You can do it by using the [tunnelbroker.net](#)  service. Verify whether you are using the tunnel for communication by checking a test page (e.g. [ipv6-test.com](#) ) or by using IPv6 versions of ping and traceroute.

**Task 4:**

Check whether IPv6 privacy extensions are enabled on your computer. Under Windows use the command "netsh interface ipv6 show privacy".

**Special Task: TV Film Reality Check**

Read the following text that summarizes part of a plot from a TV film. Then answer several questions as to whether the technical explanations are given correctly.

In the television crime series “Irene Huss - Kripo Göteborg,” the following technical explanations were presented in the episode entitled “Deadly Network” (the TV series was not translated into English). The perpetrator stole a laptop from the murder victim. The investigators can determine the MAC address of the laptop because the person who sold the computer knew it. Since the other members of the investigation team are not very knowledgeable about computers, their computer expert explains: “Every computer has a network interface card. It is required to surf in the Internet. On the network interface card there is a unique number: the MAC address. You leave traces of it in the network. It is like a fingerprint, absolutely unique.” The computer expert says that he has provided the MAC address already to the police IT department. They have contacts to all the surfing areas. Later the police receives a hint that a user is using this MAC address. A police team arrests him at his home as long as he continues to use the computer. He is interrogated by the police and claims that he had only used the MAC address for spoofing. The police computer expert explains that the computer did not have this MAC address in the first place, but that the suspect has stolen it. The suspect explains that he has taken the MAC address from the network on the train between Göteborg and Malmö. The police is later able to determine that someone with the MAC address uses the WLAN in the train to surf in the Internet. However, there are several passengers on the train so the inspector cannot determine who uses it when she passes through the train (she doesn't want to identify herself as an inspector). She limits herself to photographing passengers working on notebooks. The perpetrator is caught later using other methods.

## 4.12 Summary - Network Layer

This extensive chapter presented the basic switching methods, the Internet protocol version 4 and version 6, as well as the related auxiliary protocols such as ICMP, ARP, DHCP and the NAT method. Some insight into routing was also provided at the end.