Technische Hochschule Rosenheim

Modul - IT Systems (IT)

Bachelor Programme AAI

08 - Lecture: IP Adresses, Sockets, ...

Prof. Dr. Marcel Tilly

Faculty of Computer Science, Cloud Computing

© Technische Hochschule Rosenheim

Agenda

Technische Hochschule Rosenheim

The menu for today:

- OSI Model: Network Layer
- IPv4 and IPv6
- Sockets and Sockets in Python
- Telnet

Bon apetit!

Learning objectives

Technische Hochschule Rosenheim

Students will be able to ...

- ... explain IP, IPv4 and IPv6
- ... understand and explain an IP address
- ... understand how routing and forwarding based on IP works
- ... can program a simple socket connection between server and client in Python
- ... can connect with telnet against a service via IP address and port

Technische Hochschule Rosenheim

Bit rate vs. baud rate

- Bitrate = Bit/s:
  - Number of bits transmitted per second
- Baud
  - Number of characters/symbols transmitted per second. How many bits/s a baud is depends on the encoding.

So if I transmit 2 characters (01, 10, 00, 11), my

```
bitrate = 2 * baudrate
```

> A 9600 bit-per-second modem works with 2400 baud, but it transmits 4 bits per baud (so 4 x 2400 = 9600 bits per second).

© Technische Hochschule Rosenheim

OSI layers

Technische Hochschule Rosenheim

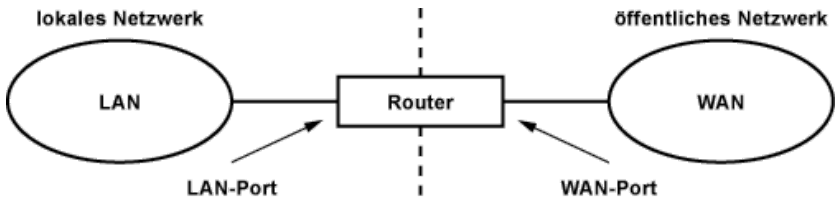
Layers and meaning

#	Layer	Meaning	Units
7	Application Layer	Application Layer	
6	Presentation Layer	Data	
5	Session Layer	Communication Layer	
4	Transport Layer	TCP, UDP	
3	Network Layer	Network Layer	Packets
2	Data Link Layer	Link Layer	Frame
1	Physical Layer	Physical Layer	Bits, Symbols, Packets

Technische Hochschule Rosenheim

Router

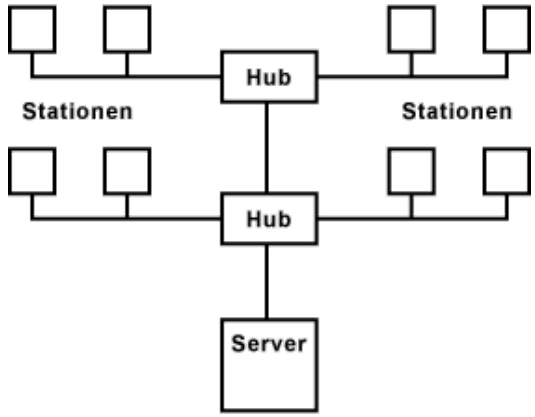
- *Routers* connect networks using different protocols and architectures.
- *Routers* are often found at the outer boundaries of a network.
  - This is where the connection to other networks and the Internet is created.
- A router uses the routing table to decide which path a data packet takes.
  - This is a dynamic process that can take failures and bottlenecks into account without the intervention of an administrator.
- A router has at least two network ports.
- It operates on the network layer (layer 3) of the OSI layer model.



Network element

Hub

- A hub is a coupling element that connects multiple hosts in a network.
- Hubs operate at the *physical layer (layer 1)* of the OSI layer model and are thus limited to the pure distribution function.
- A hub receives a data packet and forwards it to all other ports.
  - This means: "it broadcasts".
  - This means that not only all ports are occupied, but also all hosts.

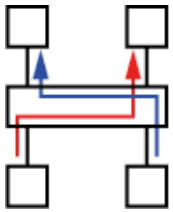


- Hubs receive all data packets, even if they are not the recipients. For the hosts, this also means that they can only transmit when the hub is not currently sending data packets. Otherwise, collisions will occur.

Technische Hochschule Rosenheim

Switch

- A switch is a coupling element that connects multiple stations in a network.
- A switch operates at the *link layer (layer 2)* of the OSI model.
- Its function is similar to a hub, with the difference that a switch can switch direct connections between the connected devices, provided that the ports of the data packet receivers are known to it.
- If not, the switch broadcasts the data packets to all ports.
- When the reply packets come back from the receivers, then the switch remembers the MAC addresses of the data packets and the corresponding port and then sends the data packets only there.

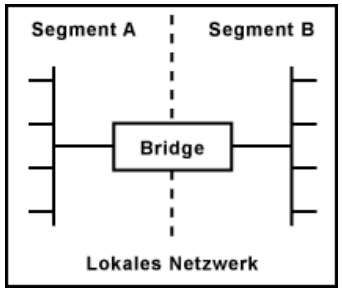


© Technische Hochschule Rosenheim

Technische Hochschule Rosenheim

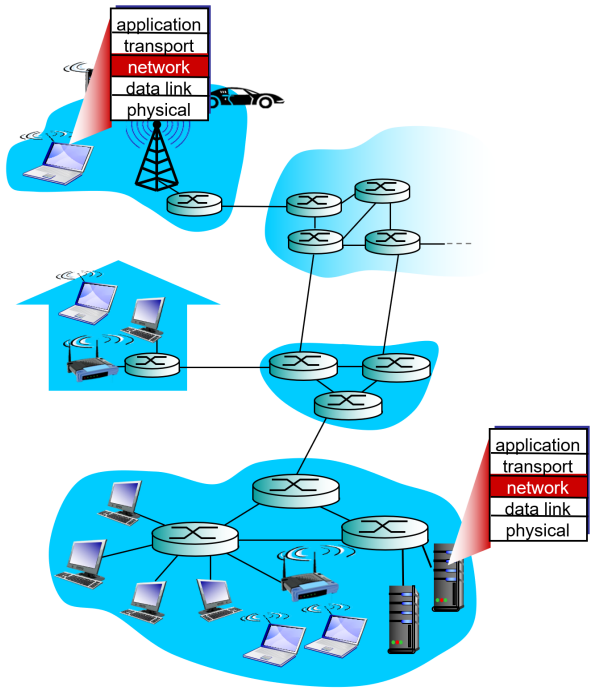
Bridge

- A bridge is a coupling element that divides a local network into two segments.
- This compensates for the disadvantages of Ethernet, which occur particularly in large networks.
- As a coupling element, the bridge is rather a typical.
- Today, the limitations of Ethernet are rather avoided by switches.
- A bridge works on the *link layer (layer 2)* of the OSI layer model.
- It transmits all protocols running on the Ethernet.



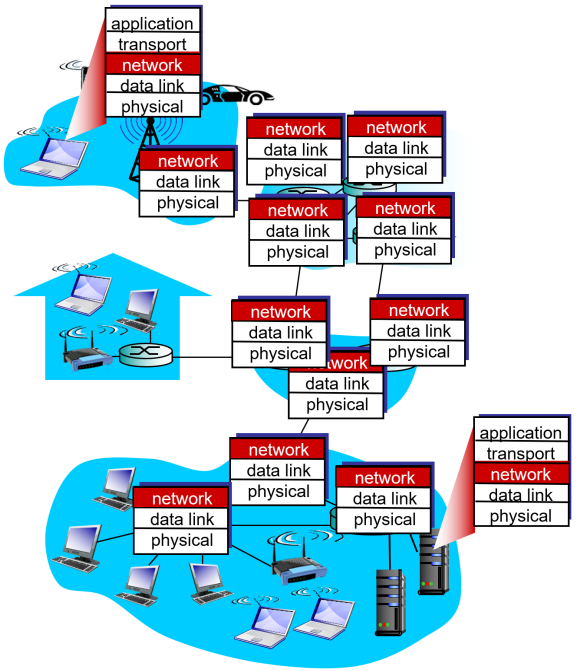
Network Layer

- **End-to-End** connection between sender and receiver
- Sender
  - Packing a transport layer (layer 4) segment into datagrams
- Receiver
  - Delivering the datagram to transport layer
- Router
  - not interested in layer 4/5
  - only care about forwarding to destination host.



Network Layer

- **End-to-End** connection between sender and receiver
- Sender
  - Packing a transport layer segment into datagrams
- Receiver
  - Delivering the datagram to transport layer
- Router
  - not interested in layer 4/5
  - only care about forwarding to destination host.



Tasks of the Network Layer

Technische Hochschule Rosenheim

- Addressing : **IP addresses**
- Forwarding
  - To which outgoing interface must a datagram be forwarded?
  - For routers: Often implemented in HW.
- Routing
  - Calculation of the paths
  - Entering the results in forwarding tables.
  - Routing protocols, mostly implemented in SW.

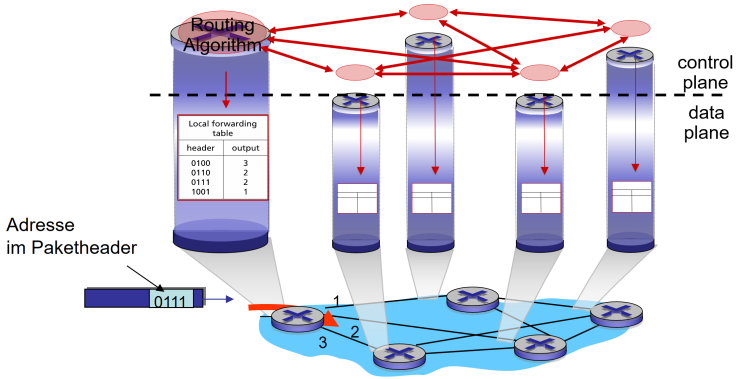
IP is connectionless

Analogy:

- Forwarding:
  - Navigation tells driver at an intersection whether to turn left or right.
- Routing:
  - Navigation system calculates travel route.

Network Layer

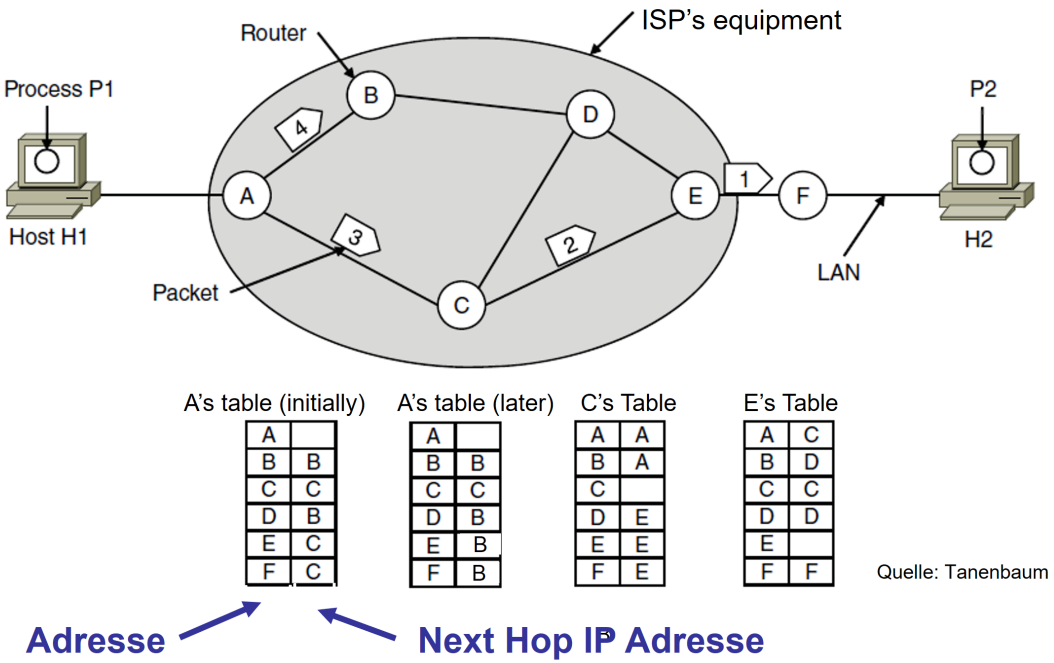
Forwarding and routing



- **Forwarding / "Data Plane "**
  - fForwarding: local function of each router
  - Implementation often in HW
- **Routing / "Control Plane "**
  - Automatic routing: Network-wide function
  - Routing messages between routers

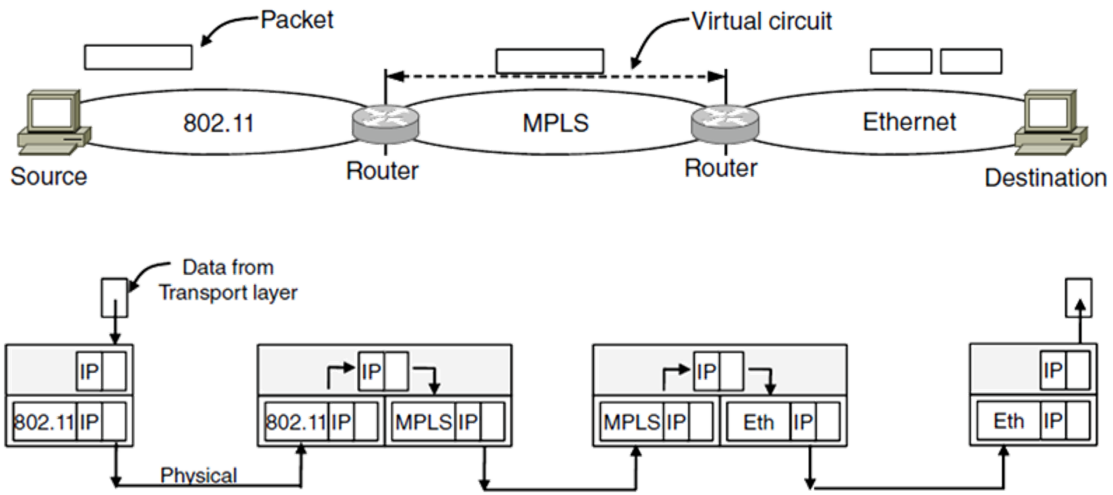
IP: Connectionless Forwarding

- Forwarding of the packet only based on the destination IP address.
- When changing tables, packets can follow different paths.



Interworking of heterogeneous networks

- IP is the link
- The link layer can be different



Network Layer

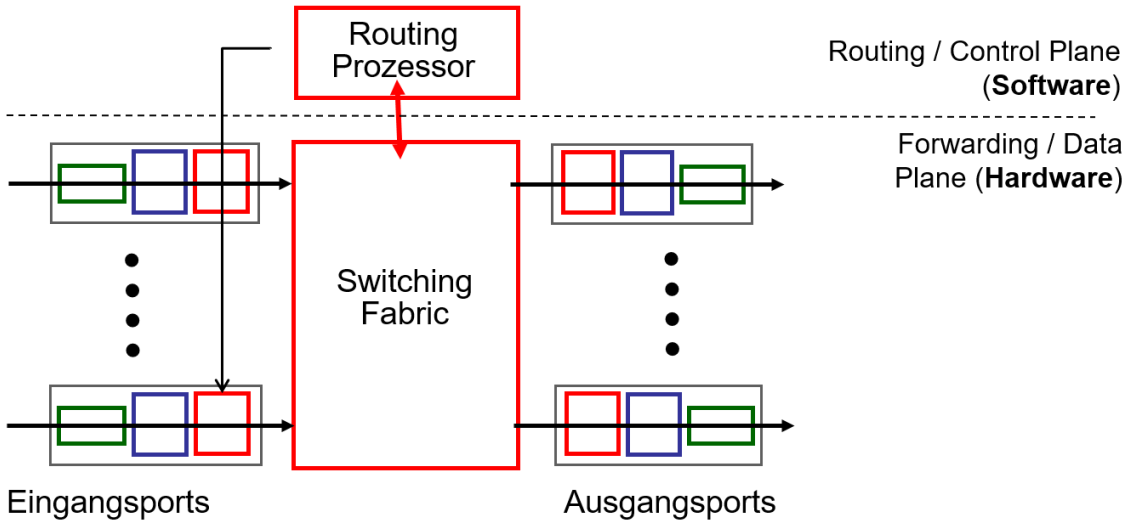
Technische Hochschule Rosenheim

Best Effort

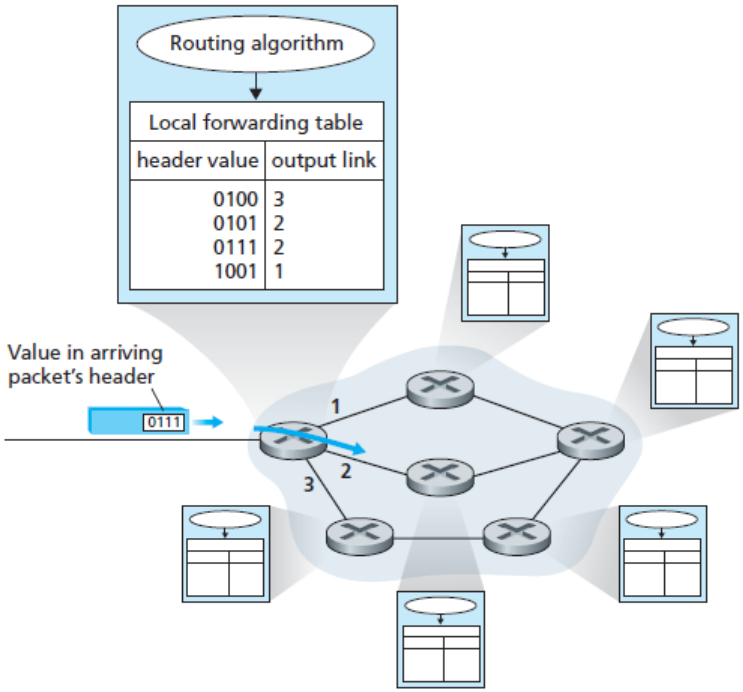
- Network Layer provides Best Effort service
- No preference for certain packets
- Each router *does its best*
- Sequence, bandwidth, etc. not guaranteed!
    - Routers do not store any information about currently running end-to-end relationships.

Architecture of a Router

- **Input ports**: Buffering of incoming IP packets, terminates Link Layer.
- **Fabric**: "Network within router", forwarding to appropriate output port.
- **Output ports**: Buffering until transmission possible, Link/PHY functionality.
- **Routing processor**: Execute routing protocols



Forwarding Table



Forwarding by Destination Address

- Router forwards by **range** (see table).
- Advantage: scalability, as not every single address occupies table space.

Target address	
11001000 00010111 00010000 00000000 to 11001000 00010111 00010111 11111111	
11001000 00010111 00011000 00000000 to 11001000 00010111 00011000 11111111	
11001000 00010111 00011001 00000000 to 11001000 00010111 00011111 11111111	
otherwise	

Longest Prefix Matching

Technische Hochschule Rosenheim

- Instead of specifying address ranges, prefixes are used in practice.
- **Longest Prefix Matching** - Definition:
  - Lookup a destination IP (32 bit) in forwarding table of router.
  - Look up longest address prefix that matches the destination address.
- Examples: Which output port?
  - Destination IP: 11001000 00010111 00010110 10100001
  - Destination IP: 11001000 00010111 00011000 10101010

Target address	Port
11001000 00010111 00010*** *********	0
11001000 00010111 00011000 *********	1
11001000 00010111 00011*** *********	2
otherwise	3

Longest Prefix Matching

Technische Hochschule Rosenheim

- Instead of specifying address ranges, prefixes are used in practice.
- **Longest Prefix Matching** - Definition:
  - Lookup a destination IP (32 bit) in forwarding table of router.
  - Look up longest address prefix that matches the destination address.
- Examples: Which output port?
  - Destination IP: 11001000 00010111 00010110 10100001 -- port 0
  - Destination IP: 11001000 00010111 00011000 10101010 -- port 1

Target address	Port
11001000 00010111 00010*** *********	0
11001000 00010111 00011000 *********	1
11001000 00010111 00011*** *********	2
otherwise	3

What is valid here?

Technische Hochschule Rosenheim

Given is the forwarding table of an IP router. To which port does it forward the packet with the following destination IP?

10010100 10010001 01000010 01100001

Port A v Port B v Port C v Port D

Target address	Port
1001001* ******** ******** *********	0
1001**** ******** ******** *********	1
10010*** ******** ******** *********	2
******** ******** ******** *********	3

VOTE! SessionID: **097124**

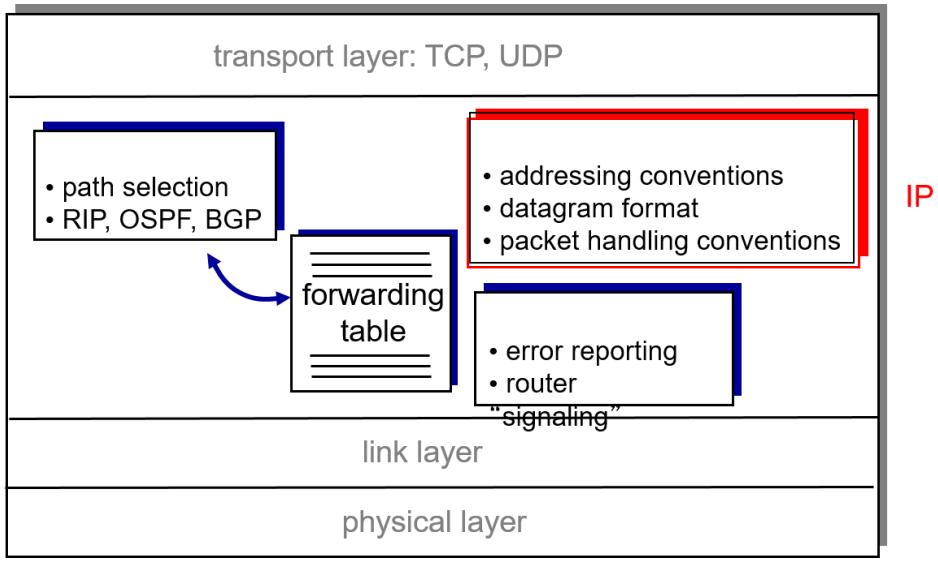
https://pingo.coactum.de/097124

Network Layer

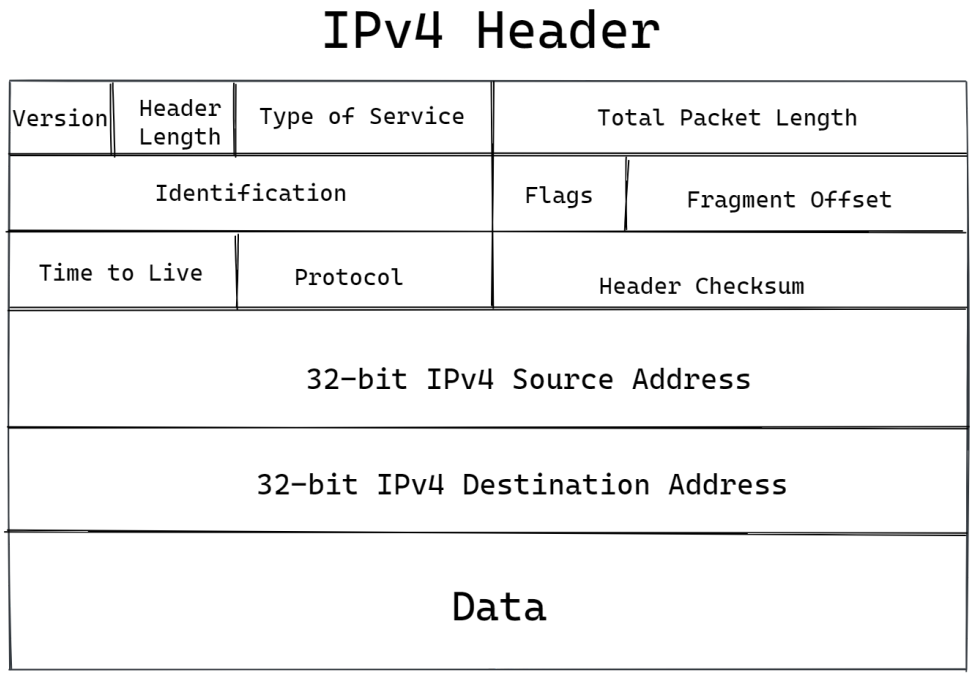
Technische Hochschule Rosenheim

Protocols

- Actually, Network Layer consists of several protocols.
- But most important is the Internet Protocol (IP)
  - versions IPv4 and IPv6

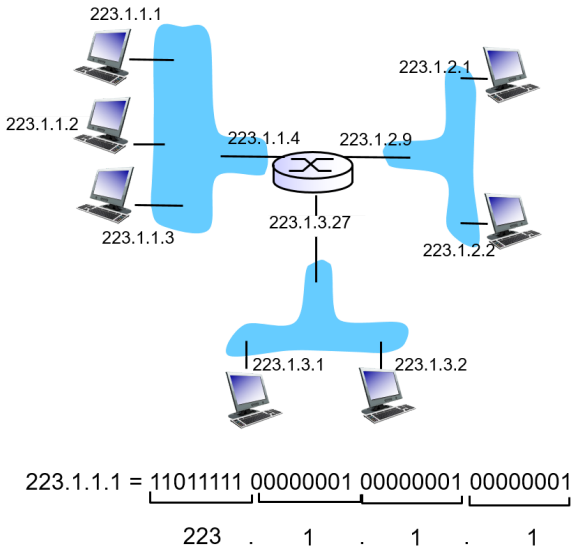


IPv4 Datagram



IP Addressing

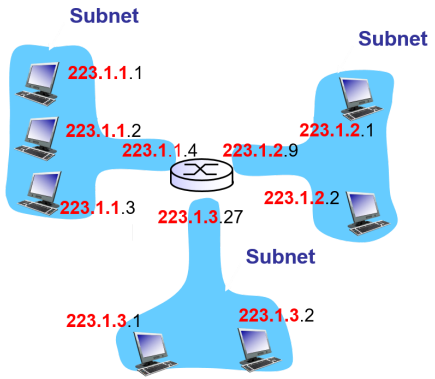
- IP address
  - has **32 bit**
  - Identifies host on the Internet
  - But logically belongs to Interface.
- Interface
  - Connection between host/router and link
  - Routers have several interfaces.
  - Each interface needs 1 IP address.



Writing IPv4 address: Decimal numbers separated by dots

Subnets

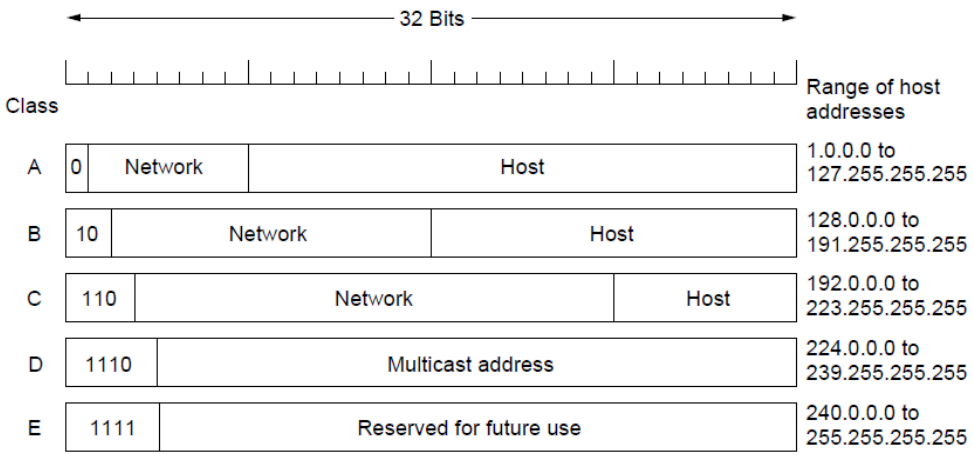
- What is an IP subnet?
  - Hosts share the same IP address prefix
  - Hosts can reach each other without router.
  - Ex: Ethernet, WLAN, etc.
- Address of a subnet
  - Subnet is addressable via common prefix!
  - Subnet mask (red): Length of the common prefix (e.g. /24)
  - Host part: Bits of the IP address which are different for each host.



- Notation, example
  - 223.1.3.0/24
  - The first 24 bits are the same for all hosts of the subnet.
- Advantage: You only have to keep subnet addresses in the routing tables.

Class Addressing

- Address ranges are passed to companies, universities, etc.
- Previously: Subnet prefixes must have mandatory length /8, /16 or /24.
- How many hosts can a Class /24 or Class /16 network have?



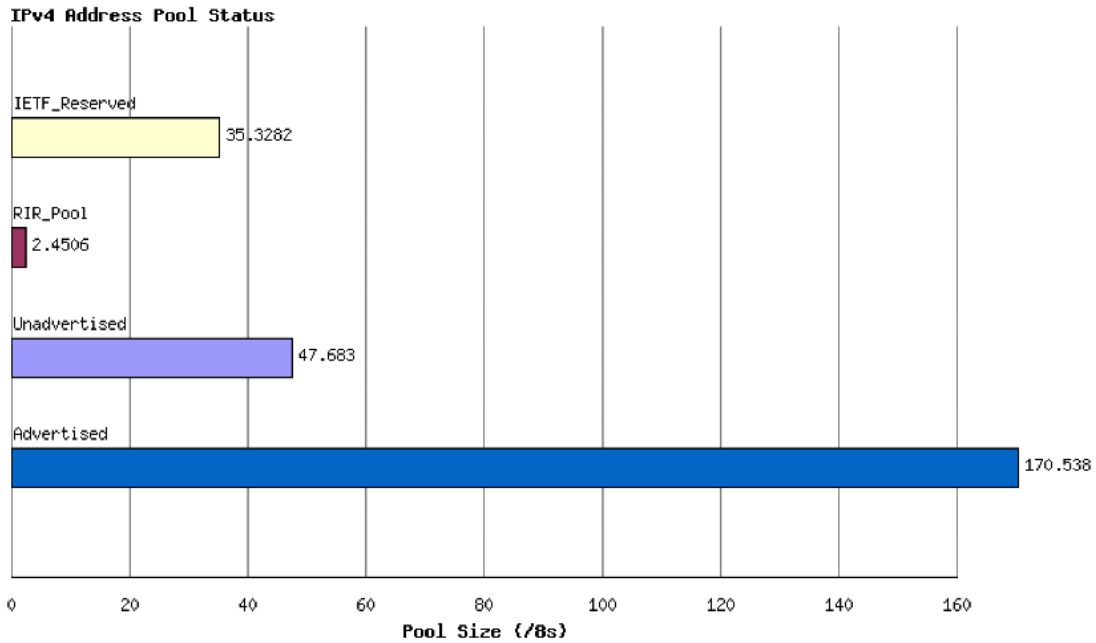
Special IPv4 addresses

Technische Hochschule Rosenheim

- Localhost, own PC
  - 127.0.0.1
- Private IPv4 addresses
  - globally not visible, to be used only locally in own administrative network.
  - 10.0.0.0/8
  - 172.16.0.0/12
  - 192.168.0.0/16
- Special addresses in a subnet
  - Example: 192.168.0.0/16 (netmask: 255.255.0.0)
  - Broadcast address: 192.168.255.255
    - For messages to all hosts of the subnet
    - All bits of the host part are set to 1
  - Network address identifies the subnet: 192.168.0.0
    - All bits of the host part are set to 0
    - Cannot be configured to interface.

IPv4 addresses are scarce!

- Almost no free IPv4 addresses left!
- However, there are address ranges that have been allocated but are not (yet) announced in public routing tables



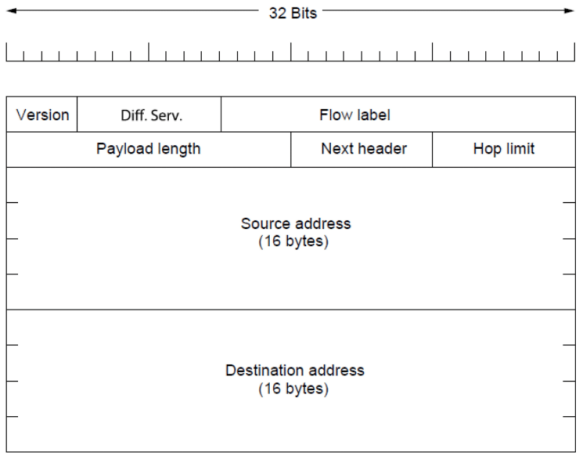
Quelle:
http://www.potaroo.net/tools/ipv4/
(abgerufen am
25.05.2018)

Goals for the development of IPv6

- Support for many hosts!
- Small, compact routing tables
- Simplification of the protocol, e.g. fast processing
- Flexibility: allow future extensions.
- Migration and coexistence of IPv4 and IPv6 during transition.
- Better support for multicasting, mobility, Quality of Service (QoS).

IPv6 Header Format

- Diff.server
  - "Priority" of the packet or flow
- Flow Label
  - Packages with the same label form a group (= flow)
  - Should be treated the same.
  - Rarely used.
- Next Header
  - Specifies if Extension Header follows or which Transport
- Layer Protocol (TCP/UDP)
  - Multiple Extension Header possible; each header points to the next ("chain")
- Hop Limit =TTL



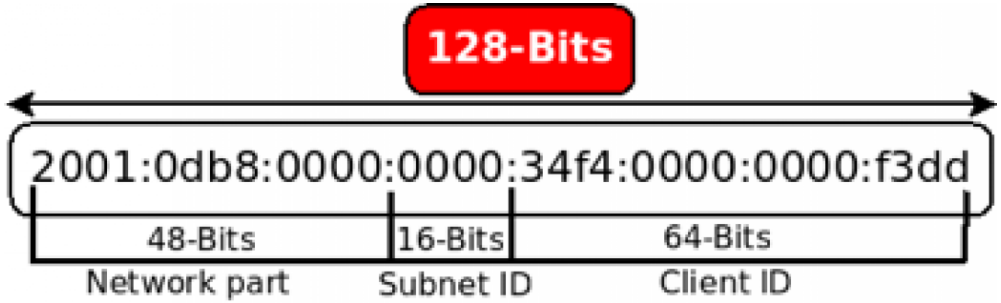
IPv6: Notation

Technische Hochschule Rosenheim

- Full notation
  - 128 bits are divided into 8 blocks of 16 bits each (4 hexadecimal digits)
  - blocks are separated by ":".
  - Example: 2001:0db8:85a3:08d3:1319:8a2e:0370:7344
- Abbreviated notation
  - Leading zeros can be omitted
  - Only once (!) one or more consecutive blocks with the value 0000 may be omitted and replaced by `::`.
    - 2001:0db8:0:0:0:1428:57ab becomes 2001:db8::1428:57ab
  - IPv4 addresses can be written as follows:
  - ::192.31.20.46
- URL notation of IPv6 addresses with square brackets
  - http://[2001:0db8:85a3:08d3:1319:8a2e:0370:7344]:8080/
- Address ranges
  - ::1/128 Loopback
  - 2000::/3 Global Unicast: Globally reachable address
  - oFE80::/10 Link-Local: Valid only in local subnet

IPv6 addresses

- Longest Prefix Matching like IPv4!
- Network and host part
  - Client-ID: Practically always exactly 64 bit (right part)
  - So there are no /80 subnets
- Practice:
  - Network-ID: ISP assigns e.g. /48 or/56 IP prefix to private customers.
  - Subnet-ID: Everyone has 8 or 16 bits (subnet-ID) to divide his own network into further subnets.
  - Client-ID: 64 bits



Further differences to IPv4

Technische Hochschule Rosenheim

- No fragmentation
  - Router informs sender via ICMPv6 that message too large.
  - Each IPv6 host automatically has a link localIPv6 address
    - e.g. derived from the MAC address.
    - Only valid in local LAN.
- No ARP
  - Implemented via other protocol (Neighbor Discovery)
- Stateless Autoconfiguration
  - Stateless DHCP server part of IPv6 standard
  - But there is also DHCPv6

© Technische Hochschule Rosenheim

Own IP address

Technische Hochschule Rosenheim

How can I get my own IP address?

Linux/Mac:

```
$ ip addr
```

```
$ hostname -I
```

```
$ ifconfig
```

Windows:

```
> ipconfig
```

What is your IP address (IPv4 and IPv6)?

What is the subnet mask?

Sockets

Technische Hochschule Rosenheim

Sockets add so many layers to the complexity of network connections that they can be thought of as a stream, analogous to a file operation.

A socket can also be thought of as the access port of an IP connection channel between a server (*host*) and a client.

For a client, the most important socket operations are:

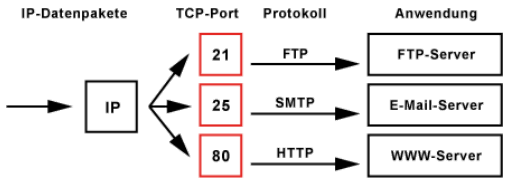
- Create connection (`open`)
- send data (`send`)
- receive data (`read`)
- close connection (`close`)

For a server the following operations are added:

- Bind an IP port (`bind`)
- accept connection request (`accept`)

Ports

- A port is a logical connection interface
- A port allows your computer to communicate with other computers and devices, as well as with the Internet.
- With ports, data packets on the Internet can always be assigned to a suitable application.
- Ports allow access to a computer
- The possible port numbers range from 1 to 65535 (16-bit unsigned).
  - The first 1-1023 ports were defined by the *Internet Assigned Numbers Authority* (IANA) for specific services and purposes.
  - Port 80, for example, is used for surfing (HTTP), and port 25 is always used for sending mail (SMTP).
  - The ports between 1024 and 49151 can be assigned quite freely by programs and applications.



Unix Commands

Technische Hochschule Rosenheim

Showing socket connections with `ss` (show socket statistics)

- Show all socket connections:

```
$ ss -s
```

- Show open ports

```
$ ss -l
```

- and much more ...

```
$ ss -h
```

server - client

Technische Hochschule Rosenheim

Netcat (`nc`) is a simple tool to transport data from standard input or output over network connections.

server:

```
$ nc -l localhost 3000
```

client:

```
$ nc localhost 3000
```

- Server and client read from standard input
- Server and client write to standard output

Sockets in Python

Technische Hochschule Rosenheim

In Python, sockets are supported by the module socket.

- `socket.socket()`: creates a socket object
- `socket.bind(port)`: binds it to a port
- `socket.accept()` : Blocks further processing until a connection is requested

```
# first of all import the socket library
import socket
# next create a socket object
s = socket.socket()
# reserve a port on your computer in our case it is 12345 but it can be anything
port = 12345
# Next bind to the port
s.bind(('', port))
print ("socket binded to %s" %(port))
# put the socket into listening mode
s.listen(5)
print ("socket is listening")
# a forever loop until we interrupt it or an error occurs
while True:
  # Establish connection with client.
  c, addr = s.accept()
  print ('Got connection from', addr )
  # send a thank you message to the client. encoding to send byte type.
  c.send('Thank you for connecting'.encode())
  # Close the connection with the client
  c.close()
  # Breaking once connection closed
  break
```

Sockets with the shell

Telnet

- `Telnet` is originally an abbreviation for *"Telecommunication Network"*.
- denotes a protocol from the TCP/IP world that can be used to execute text-based commands on remote computers.
- The client starts the connection setup for a session with the command 'telnet' followed by the computer name or the target IP address (the connection uses port 23 by default!).
- A typical application is remote control of computers with text-based input and output.
  - For example, resources on a server located at a certain location can be made accessible to others.
  - Also many network devices like routers, switches or access points can be controlled and managed with the text based sessions.

```
$ telnet 127.0.0.1 7000
```

Socket Client in Python

Technische Hochschule Rosenheim

- Requires only the `socket` module
- Connects with `connect` method against an IP address and port

```
# Import socket module
import socket

# Create a socket object
s = socket.socket()

# Define the port on which you want to connect
port = 12345

# connect to the server on local computer
s.connect(('127.0.0.1', port))

# receive data from the server and decoding to get the string.
print (s.recv(1024).decode())
# close the connection
s.close()
```

Summary

Technische Hochschule Rosenheim

Lessons learned today:

- IP is an important (if not the most important protocol) in networking
- Socket programming is sooo easy and a lot of fun with Python!
- Telnet is really a cool tool ... I will tell my grandchildren about it :-)