

# **IP Adressen & Subnetzmasken**

Jörn Stuphorn  
stuphorn@rvs.uni-bielefeld.de

# Stand der Veranstaltung

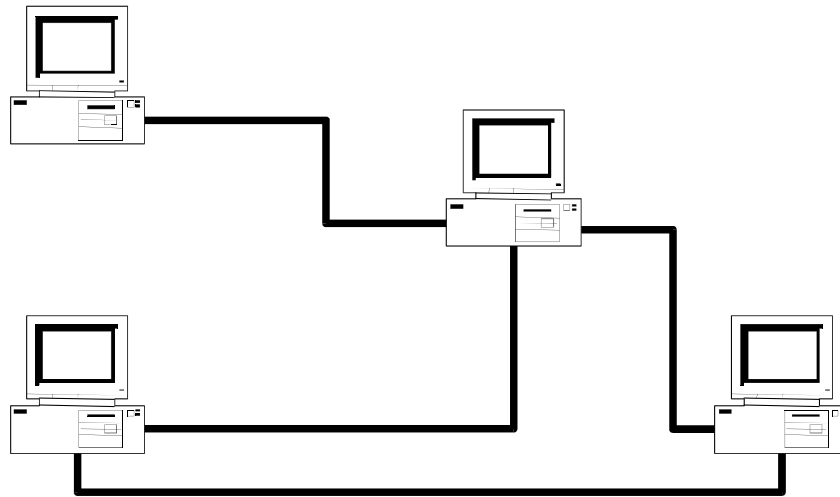
- 13. April 2005 Unix-Umgebung
- 20. April 2005 Unix-Umgebung
- 27. April 2005 Unix-Umgebung
- 4. Mai 2005 ARP, ICMP, ping
- 11. Mai 2005 IP-Adressen & Subnetzmasken**
- 18. Mai 2005 *Bridging*
- 25. Mai 2005 *IOS, Spanning-Tree*
- 1. Juni 2005 *Statisches Routing*
- 8. Juni 2005 *UDP-, MTU- und IP-Fragmentierung*
- 15. Juni 2005 *TCP-Verbindungen und -Datenfluss*
- 22. Juni 2005 *DHCP und NTP*
- 29. Juni 2005 *NAT und Firewalls*  
*Verschlüsselung, Vertraulichkeit,*
- 6. Juli 2005 *Authentisierung*
- 13. Juli 2005 *Sichere Anwendungen*
- 20. Juli 2005 *Wireless LAN*

## **Wiederholung**

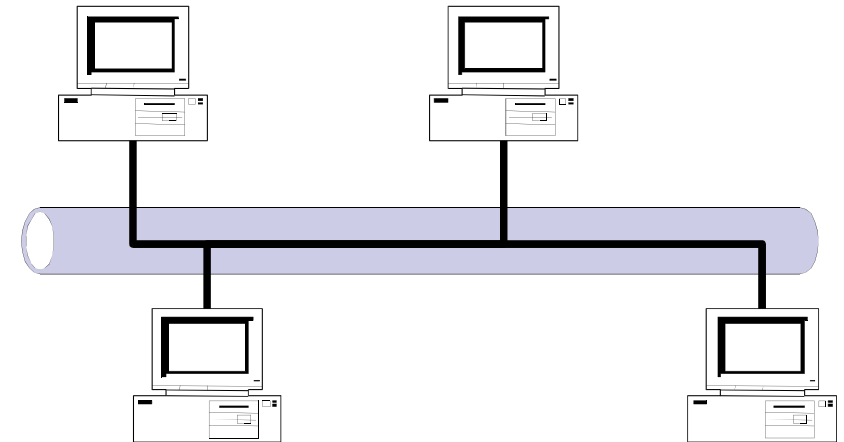
- Netzwerk
- ARP
- ethereal
- tcpdump
- sock – Traffic Generator

## **IP-Adressen und Subnetzmasken**

- IP-Adresse doppelt vorhanden
- Subnetzmaske falsch gesetzt



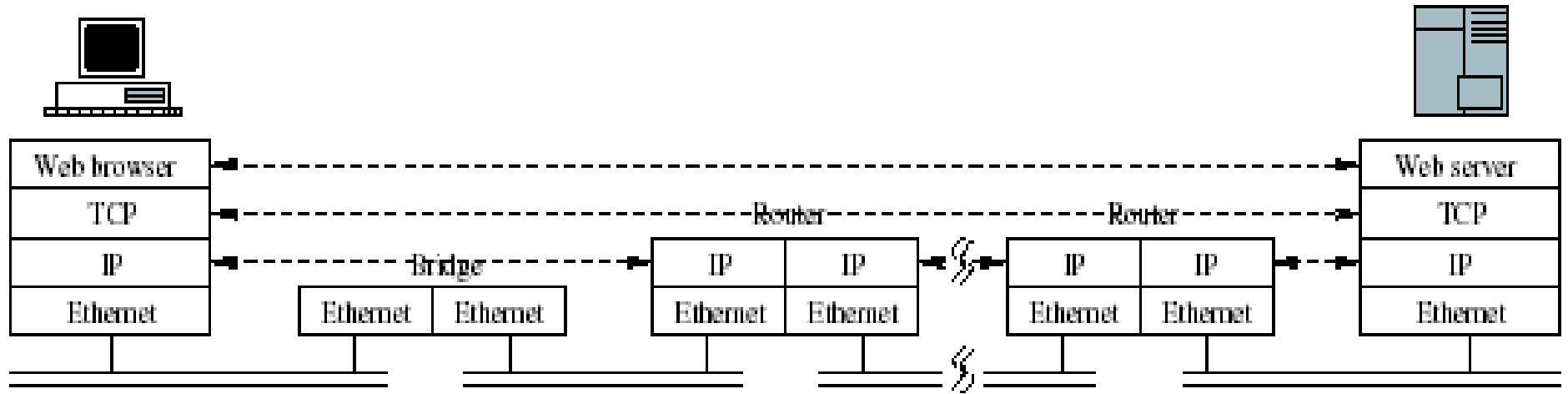
**Point-to-Point Netzwerk**



**Broadcast Netzwerk**

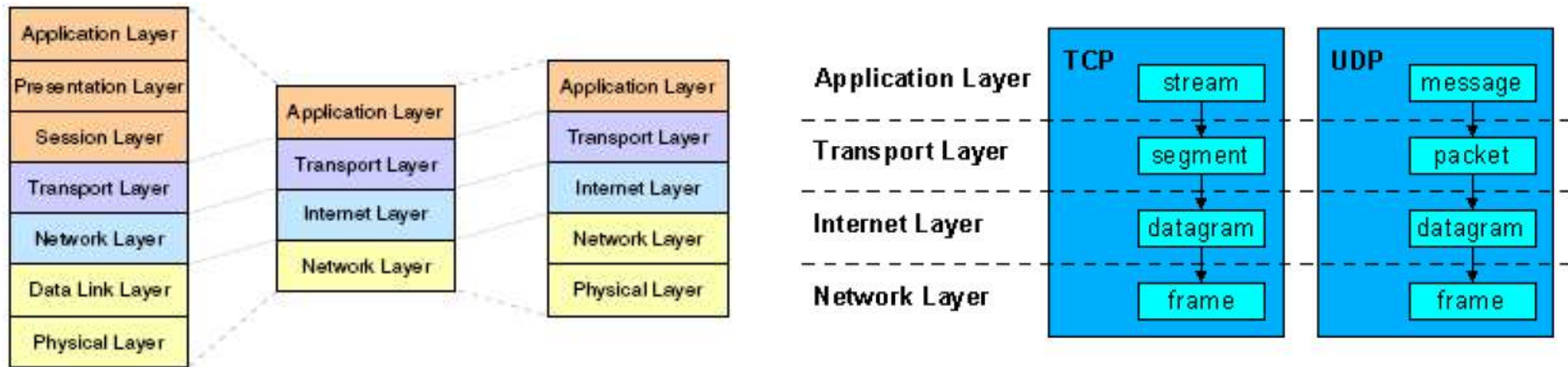
- **Point-to-Point**
  - Verbindung von einzelnen Knoten untereinander, meistens über große Entfernungen (Zugang über Telefonnetz)
  - PPP (Point-to-Point Protocol)
- **Broadcast**
  - Mehrere Knoten teilen sich einen Kanal
  - Zugriff auf Medium muss gesteuert werden

# Ethernet und TCP/IP



Quelle: Panwar, Mao, Ryoo, Li, „TCP/IP Essentials“, Lecture Slides

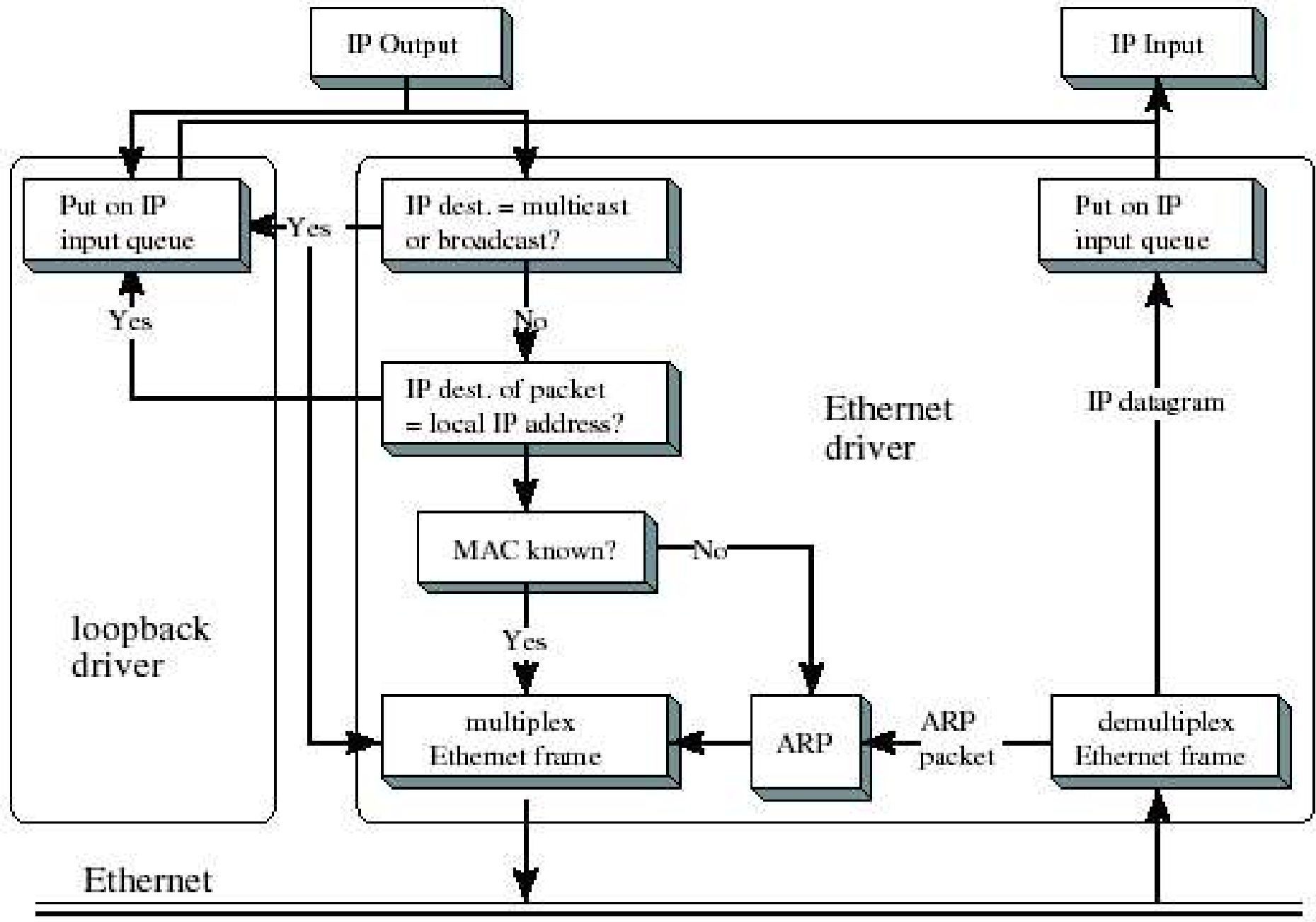
- Ethernet ist Broadcast Netzwerk
  - Knoten werden über Ethernet-Adresse adressierbar  
MAC Adresse: 6 Byte lang
  - Länge eines Strangs ca. 100m (Cu) bis 5000m (Glas)
- Verbindung von mehreren Strängen über Bridge
- Mehrere Netzwerke werden über Router verbunden
  - Protokoll: IP (Internet Protocol)
- Anwendungen kommunizieren mittels Socket (TCP) bzw. Message (UDP)



- Informationen werden in unterschiedlichen Einheiten transportiert:
  - *Network Layer*  
ethernet: Rahmen (frames) werden übertragen
  - *Internet Layer*  
IP: Datagramme werden übertragen
  - *Transport Layer*  
TCP: Segmente werden übertragen  
UDP: Pakete werden übertragen
  - *Schnittstelle*  
TCP bietet Programmen stream zur Kommunikation an  
UDP bietet Programmen message zur Kommunikation an

- Die meisten TCP Implementationen bieten ein Loopback-Device an
  - IP-Adresse: 127.0.0.1
  - Name: localhost
- Verhält sich wie eigenständiges Interface
- Wird zum Debuggen genutzt
  
- Ein an das Loopback Interface geschicktes Paket wird durch den Protokoll Stack geleitet und an den Software-Treiber des „Device“ zurückgegeben
  
- Pakete an das Loopback Interface werden nicht über das Netzwerk geleitet.

# Abläufe im Netzwerkinterface





- Die Größe eines Rahmens ist begrenzt.
- Diese Größe wird als *MTU* oder *Maximum Transmission Unit* bezeichnet.
  
- Gängige Größen der MTU:
  - SLIP/PPP            296 Byte
  - PPPoE              ≤ 1492 Byte
  - IEEE 802.2        1492 Byte
  - Ethernet            1500 Byte
  - FDDI                4352 Byte
  - TokenRing(4)      4464 Byte
  - TokenRing(16)    17914 Byte
  - Hyperchannel      65535 Byte

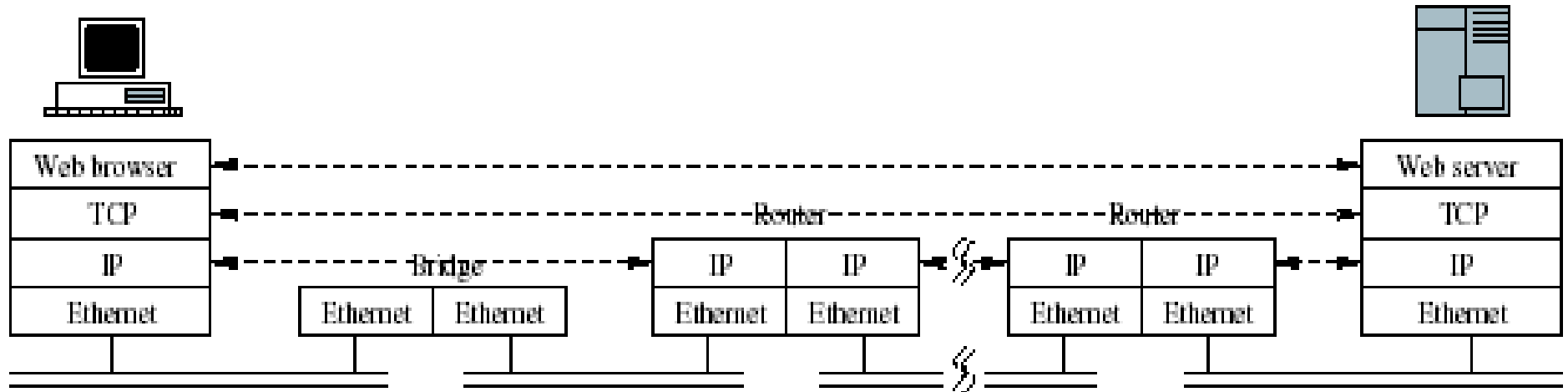
- IP-Adresse:
  - Class A [0][7bit NetworkID][24bit Host ID]
  - Class B [10][14bit NetworkID][16bit Host ID]
  - Class C [110][21bit NetworkID][8bit Host ID]
  - Class D [1110][28bit Multicast group ID]
  - Class E [11110][27bit ID *für zukünftige Nutzung reserviert*]
  
- Subnetz:
  - Netz ID der IP-Adresse bezieht sich auf ein lokales Netz
  - Adressraum kann nicht auf mehrere Netze verteilt werden
  - Mittels Subnetzen kann Adressraum weiter unterteilt werden

- Übung:
  - Rufe *ifconfig -a* auf
  
- Fragen:
  - Wie viele Interfaces hat der Rechner?
  - Welche MTUs benutzen die Interfaces?
  - Existiert ein Subnetz?

- ifconfig dient zur Konfiguration des Interfaces
- ifconfig eth0 up/down startet/stoppt das Interface
- ifconfig eth0 address setzt die IP-Adresse
- ifconfig eth0 netmask setzt die Subnetzmaske
- ifconfig eth0 mtu setzt die MTU
  
- Übungen:
  - stoppe eth0, kontrolliere ifconfig -a Ausgabe
  - starte eth0
  - änder die IP-Adresse von eth0
  - änder die Netzmaske von eth0
  - kontrolliere ifconfig -a Ausgabe
  - mache Änderungen rückgängig

- *Starteinstellungen für Interface sind in /etc/network/interfaces angegeben*
  - *Kontrolliere die Starteinstellungen*
- *Startskript /etc/init.d/networking initialisiert Interfaces*
  - *änder Standardeinstellungen*
  - *stoppe Netzwerk*
  - *starte Netzwerk*
- *Stelle Ursprungszustand wieder her.*

# Address Resolution Protocol



- Allgemeinste Schicht bei TCP/IP Übertragung: IP
- physikalische Adresse des Ziels unbekannt
- ARP löst die MAC Adresse gegen eine IP Adresse auf
- RARP (reverse ARP) funktioniert entgegengesetzt
- Um nicht für jedes IP-Paket 2 ARP-Pakete zu verschicken:  
lokaler ARP-Cache auf jeder Maschine
- Cache enthält nur aktuelle Informationen, veraltete  
Informationen werden gelöscht

# ARP Paket

Dest. Addr	Src. Addr	Type	Data	CRC
6	6	2	46-1500	4

Hardware Type	Protocol Type	Hardware Size	Protocol Size	Operation Field	Sender Eth. Addr.	Sender IP Addr.	Target Eth. Addr.	Target IP Addr.
2	2	1	1	2	6	4	6	4 bytes

Type 0806	ARP request/reply	PAD
2	28	18

Type 8035	RARP request/reply	PAD
2	28	18

---

Type 0800	IP datagram
2	46-1500

- Test, ob anderer Host mit gleicher IP im Netz vorhanden ist
- Host fragt mit ARP-Paket nach MAC-Adresse der eigenen IP
- Nebeneffekt:
  - Host verteilt seine IP- und MAC-Adresse im Netz
  - Andere Hosts übernehmen Daten in lokalen ARP Cache



- Übung:
  - Rufe ***arp -a*** auf
  - Wenn Server vorhanden ist
    - Lösche mit ***arp -d [remoteHost]*** ARP-Eintrag
  - Notiere ARP-Tabelle
  - Protokolliere Netzwerktraffic mit *ethereal*
  - Rufe ***ping [remoteHost]*** auf.
  
  - Kontrolliere die ARP-Tabelle
  - Werte Traffic mit *ethereal* aus.
  
- Fragen:
  - Beschreibe, wie ARP funktioniert anhand der *ethereal* Ausgabe

- Übung:
  - Protokolliere Netzwerktraffic mit tcpdump
  - Rufe ***telnet 192.168.254.100*** auf
  
- Fragen:
  - Werte tcpdump Ausgabe aus
  - Beschreibe, wie ARP-Timeout und Retransmission ausgeführt wurden
  - Wie viele Versuche wurden unternommen, die nicht vorhandene IP-Adresse aufzulösen?

- Übung:
  - Protokolliere Netzwerktraffic mit tcpdump
  - Starte Server neu
  
- Fragen:
  - Werte tcpdump Ausgabe aus
  - Beschreibe, gratuitous ARP

- IP-Adresse:
  - Class A [0][7bit NetworkID][24bit Host ID]
  - Class B [10][14bit NetworkID][16bit Host ID]
  - Class C [110][21bit NetworkID][8bit Host ID]
  - Class D [1110][28bit Multicast group ID]
  - Class E [11110][27bit ID *für zukünftige Nutzung reserviert*]
  
- Subnetz:
  - Netz ID der IP-Adresse bezieht sich auf ein lokales Netz
  - Adressraum kann nicht auf mehrere Netze verteilt werden
  - Mittels Subnetzen kann Adressraum weiter unterteilt werden
  
- Fragen:
  - Was passiert, wenn 2 Hosts die gleiche IP-Adresse haben?
  - Wie verhalten sich Hosts mit inkorrektter Subnetzmaske?

# Doppelte IP-Adressen

- *Setze die IP-Adressen wie folgt:*
  - *yoda*      *192.168.254.100, Subnetz 255.255.255.0*
  - *windu*     *192.168.254.100, Subnetz 255.255.255.0*
  
  - *client1*    *192.168.254.20, Subnetz 255.255.255.0*
  - *client2*    *192.168.254.21, Subnetz 255.255.255.0*
- *Protokolliere Traffic mit tcpdump auf allen Hosts*
  
- *Versuche:*
  - 1. telnet von yoda oder windu auf client*
    - *Was passiert?*
  - 2. telnet von client1 nach 192.168.254.100*
    - *Welcher Host stellt telnet Verbindung?*
  - 3. telnet von client1 nach 192.168.254.100*
    - *Welcher Host stellt telnet Verbindung?*
  
- *Werte tcpdump-Ausgaben mit ethereal aus*

- *Setze die IP-Adressen wie folgt:*
  - *yoda      192.168.254.100,      Subnetz 255.255.255.240*
  - *windu     192.168.254.101,      Subnetz 255.255.255.0*
  
  - *client1    192.168.254.102,      Subnetz 255.255.255.0*
  - *client2    192.168.254.120,      Subnetz 255.255.255.240*
- *Protokolliere Traffic mit ethereal auf allen Hosts*
  
- *Versuche:*
  - 1. ping von yoda nach windu oder client1*
  - 2. ping von client2 nach windu oder client1*
  - 3. ping von windu oder client1 nach yoda*
  - 4. ping von windu oder client1 nach client2*
  
- *Werte ethereal-Ausgaben aus*

- *yoda:*
  - *IP:* 11000000.10101000.11110101.01100010
  - *SubNet:* 11111111.11111111.11111111.11110000
  - *Netz:* 11000000.10101000.11110101.01100000  
192.168.254.96 - 192.168.254.111
  
- *windu:*
  - *IP:* 11000000.10101000.11110101.01100011
  - *SubNet:* 11111111.11111111.11111111.00000000
  - *Netz:* 11000000.10101000.11110101.01100000  
192.168.254.0
  
- *client2:*
  - *IP:* 11000000.10101000.11110101.01111000
  - *SubNet:* 11111111.11111111.11111111.11110000
  - *Netz:* 11000000.10101000.11110101.01110000  
192.168.254.112 - 192.168.254.127