

# Modul Verteilte Systeme

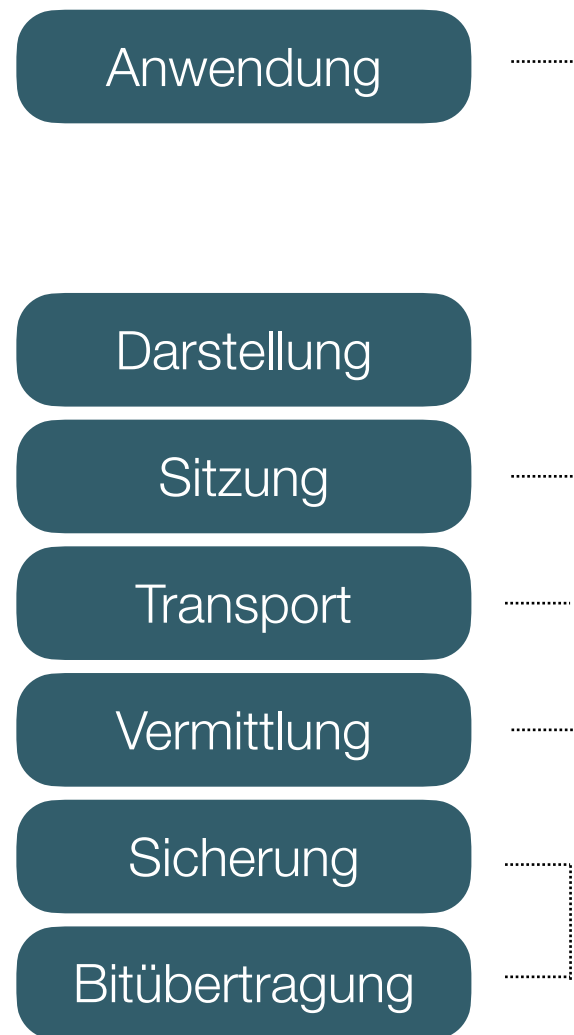
## Mehrpunktübertragung

---

Peter Tröger  
Beuth Hochschule für Technik Berlin  
Sommersemester 2020  
(Version 1)

# Beobachtung?

---

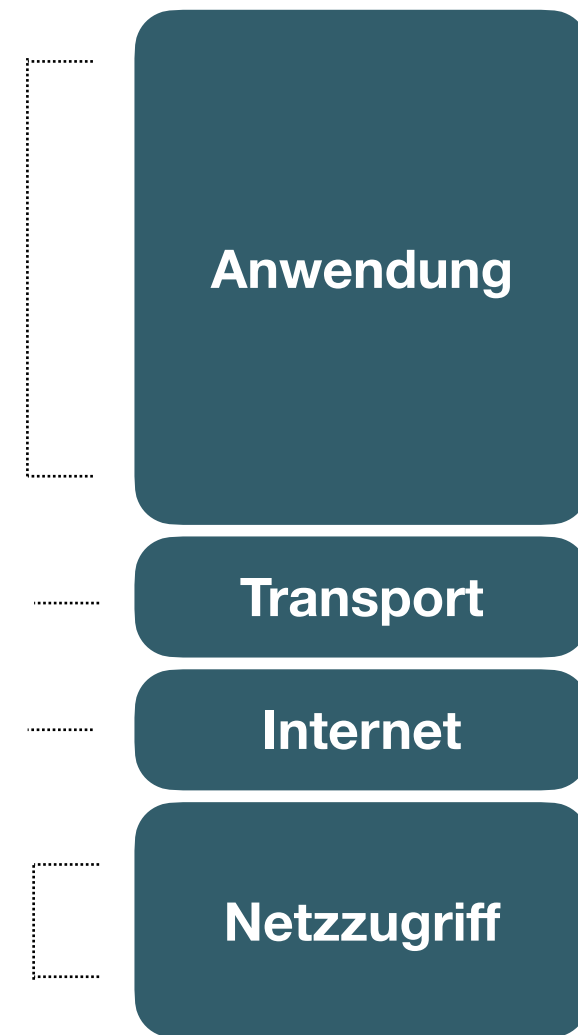


OSI-Modell

DSLP  
Übungen

TCP, UDP  
IPv4, IPv6, ARP

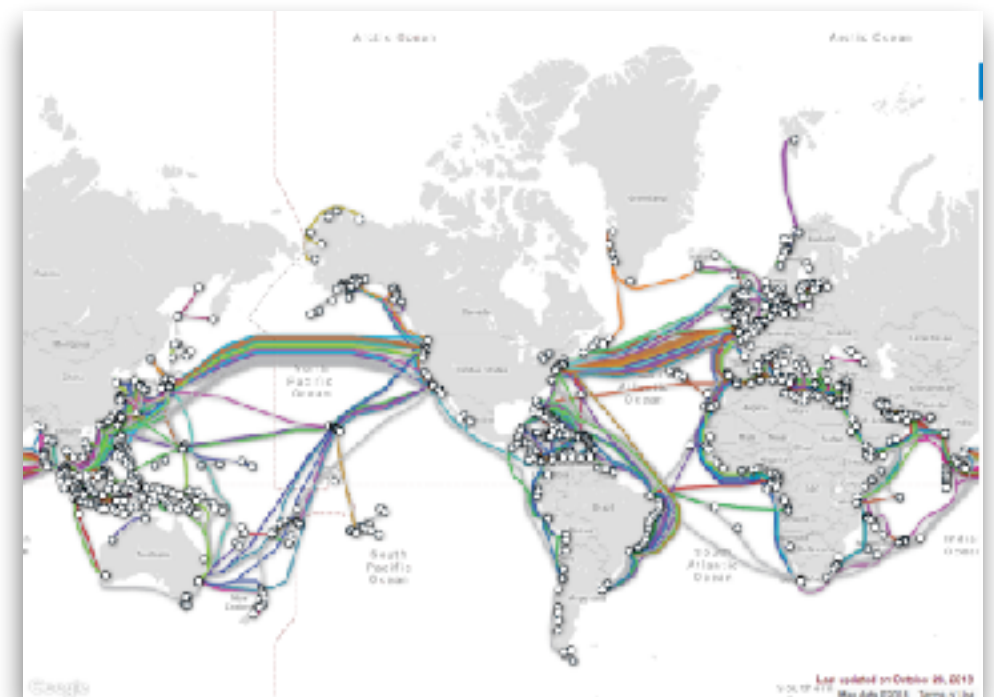
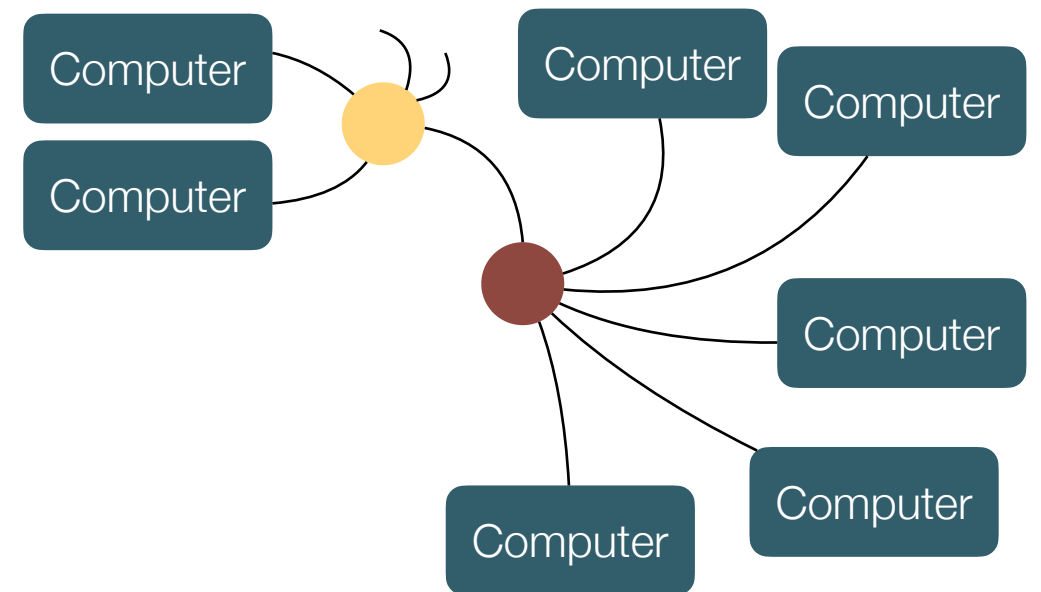
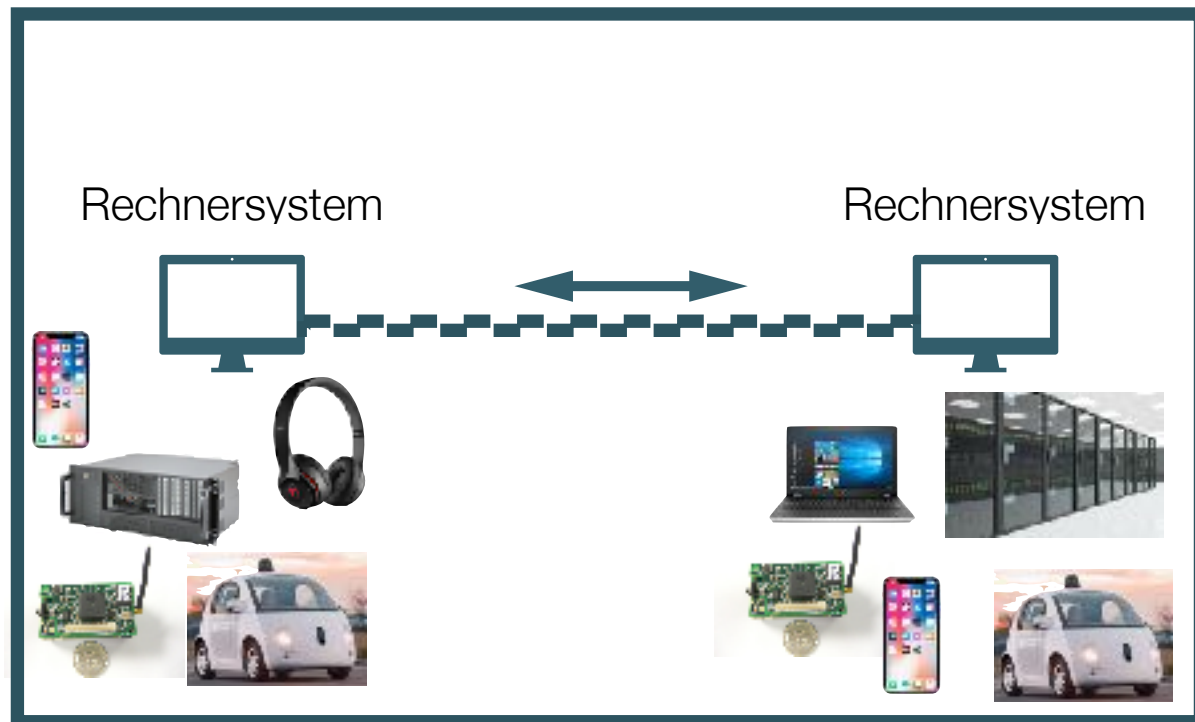
Ethernet



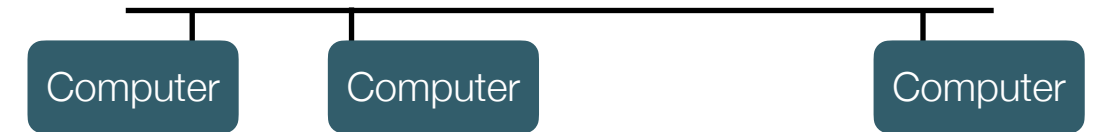
TCP/IP-Modell

Demo: Wireshark, Traceroute - Übungsaufgabe

## Verteiltes System



# Netzwerktopologie



Startseite > Wörterbuch > Topologie

## Topologie, die

Wortart INFO **Substantiv, feminin**

Gebrauch INFO **Mathematik**

Häufigkeit INFO

---

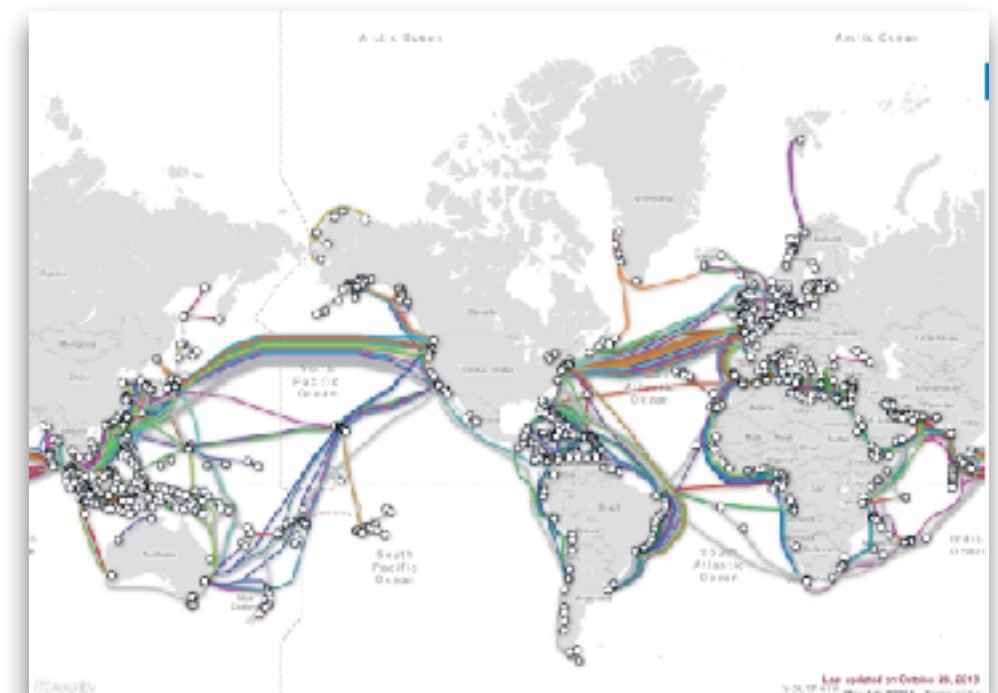
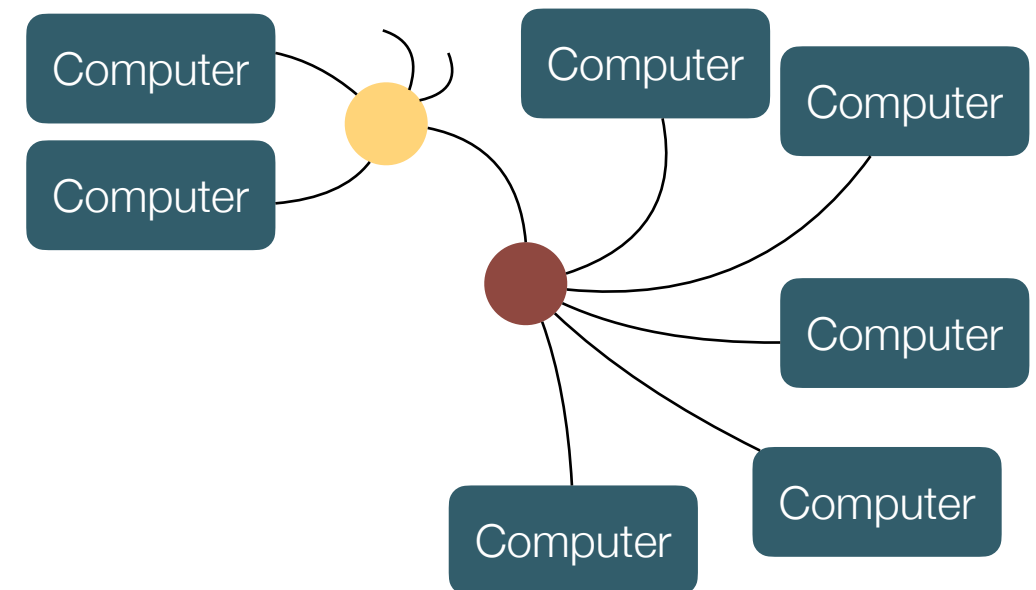
**Rechtschreibung** INFO

Worttrennung **To|po|lo|gie**

---

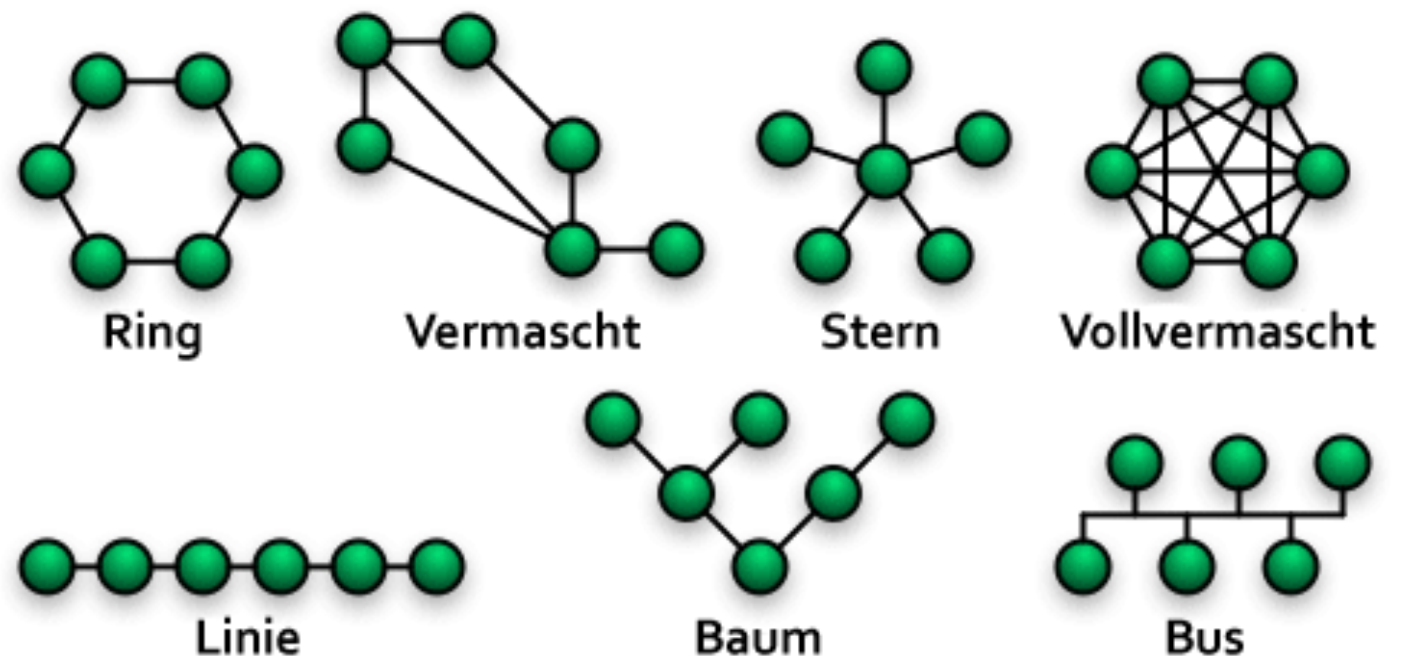
**Bedeutung** INFO

Lehre von der Lage und Anordnung geometrischer Gebilde im Raum

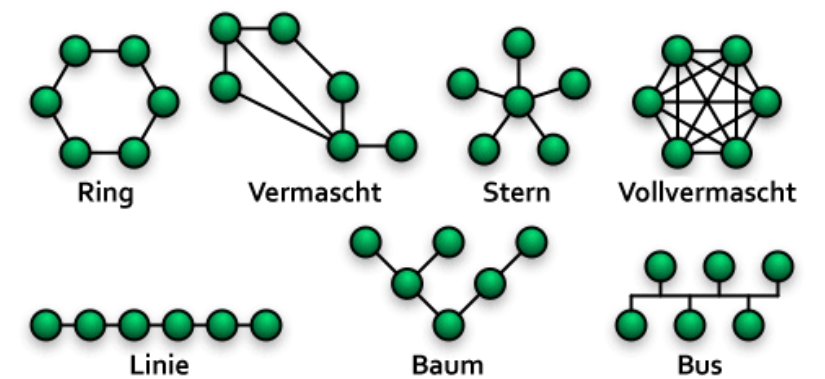


# Netzwerktopologie

- **Physische Topologie** - Netzwerkkarten, Kabel, Wellenausbreitung
- **Logische Topologie** - Datenfluss zwischen Endgeräten zur Laufzeit
- **Grad** - Anzahl der Kanten an einem Knoten
- **Durchmesser** - Größter Abstand zwischen zwei Knoten
  - Anzahl der Hops
  - Beispiel Linie:  $N-1$

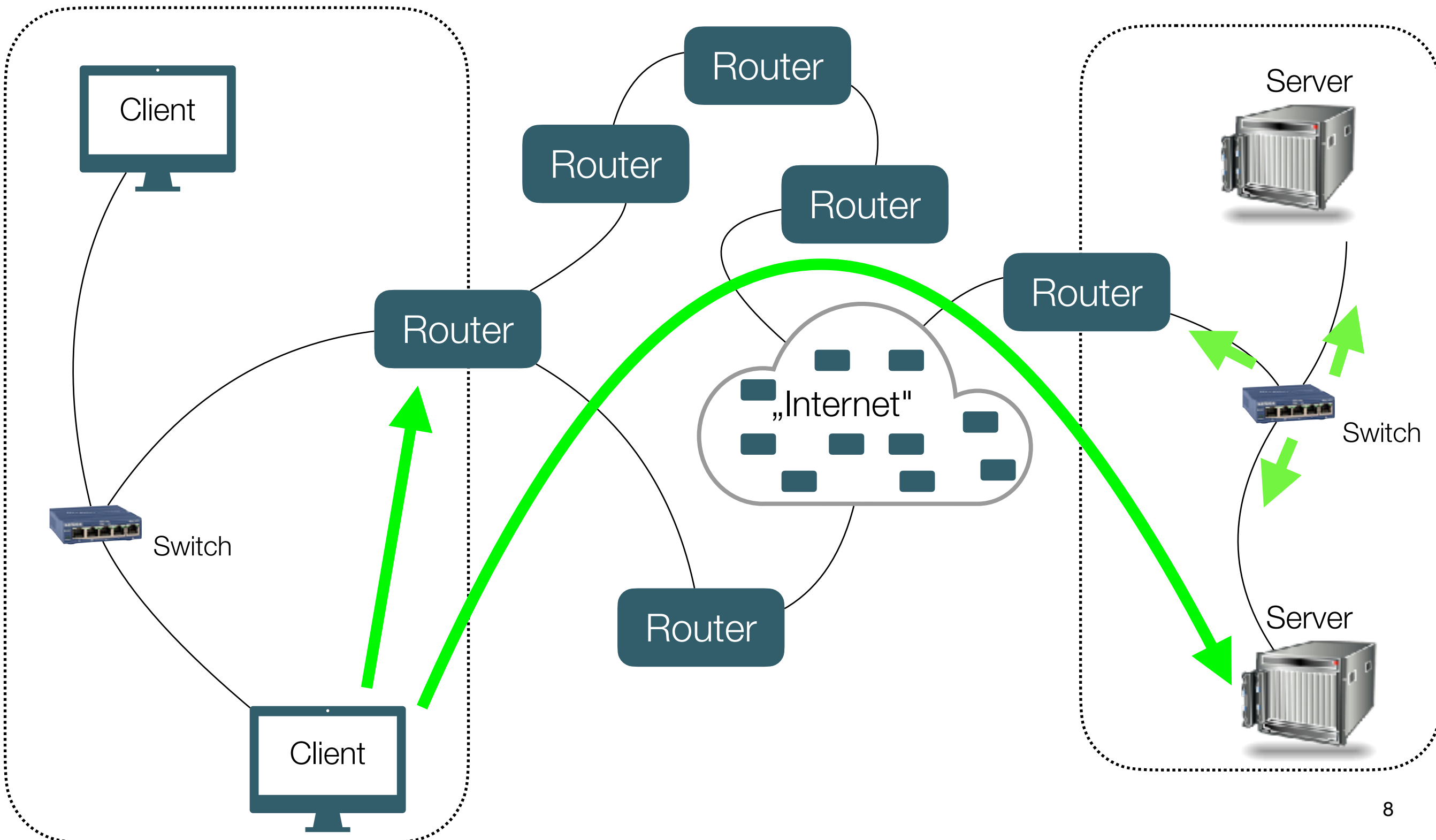


# Netzwerktopologie



	Durchmesser	Grad	Anmerkung
<b>Bus</b>	<b>1</b>	<b>1</b>	Konfliktbehandlung nötig
Linie	N-1	1,2	Lange Verzögerungen möglich
Ring	N/2	2	Für bidirektionalen Ring
Binärer Baum	2(Höhe-1)	<b>1,2,3</b>	Effizientes Routing
<b>Stern</b>	<b>2</b>	<b>1,</b> Zentrum N-1	Ähnlich zum Bus, aber bessere Fehlertoleranz
Voll-vermascht	1	N-1	Hoher Grad hat Auswirkung auf Kosten

# Nachrichtenübertragung vs. Topologie





# Übertragungsarten

---

- **Punkt-zu-Punkt** Übertragung (**unicast, point-to-point**)
  - Ein Sender, ein Empfänger, kein Knoten dazwischen
- **Ende-zu-Ende** Übertragung (**unicast, end-to-end**)
  - Ein Sender, ein Empfänger, potentiell Knoten dazwischen
- **Mehrpunktübertragung** (**broadcast, multicast, anycast**)
  - Ein Sender, mehrere Empfänger, potentiell Knoten dazwischen
- Kombinationen aus Topologie und Übertragungsart auf jeder OSI-Schicht

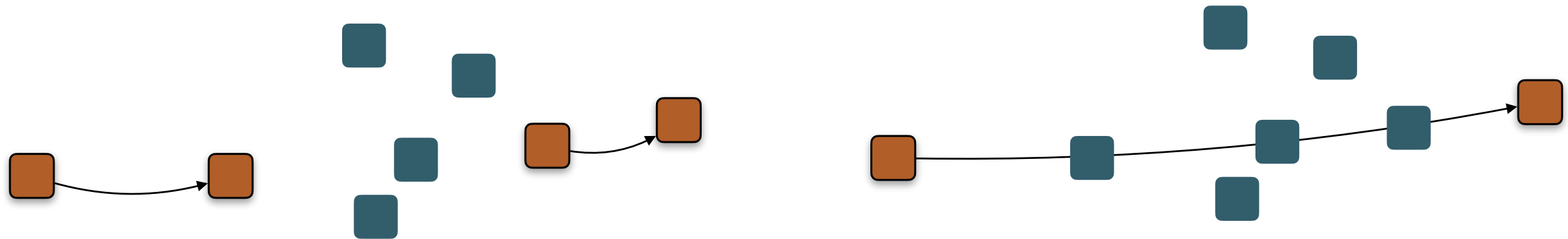
# Beispiel: DSLP Client

OSI - Schicht	Beispiel	Topologie	Übertragungsart
1/2	WLAN Access Points in der Nähe senden ihre SSID	Bus (Wellenausbreitung)	Mehrpunkt
1/2	Laptop im WLAN kommuniziert mit Router	Bus (Wellenausbreitung)	Punkt-zu-Punkt
1/2	Laptop mit Kabel am Switch, kommuniziert mit Router	Stern	Punkt-zu-Punkt
3	IP-Pakete zum DSLP-Server	Baum (logische Topologie)	Ende-zu-Ende
4	TCP-Verbindung zum DSLP-Server	vollvermascht (logische Topologie)	Punkt-zu-Punkt
7	DSLIP <i>group notify</i>	vollvermascht (logische Topologie)	Punkt-zu-Punkt, Mehrpunkt

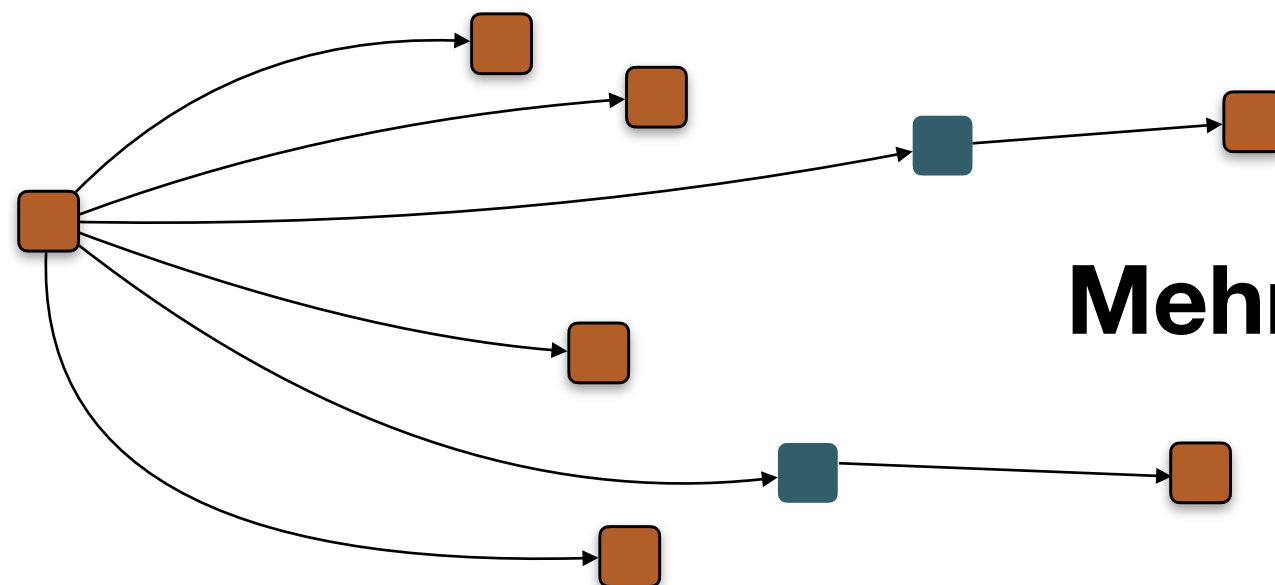
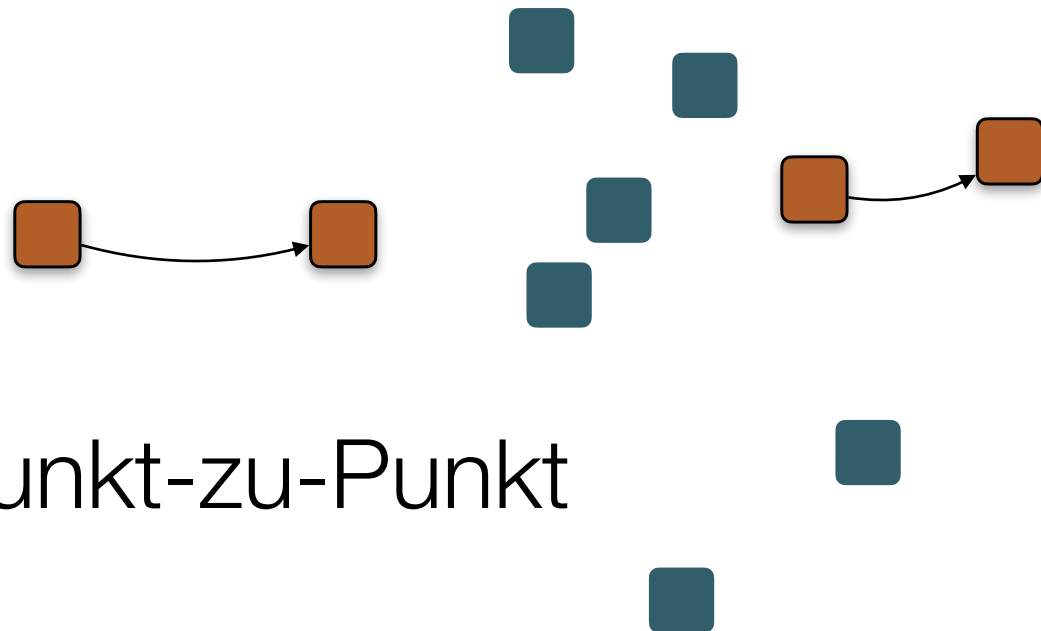
# Übertragungsarten

---

Ende-zu-Ende (= Vermittlung)



Punkt-zu-Punkt

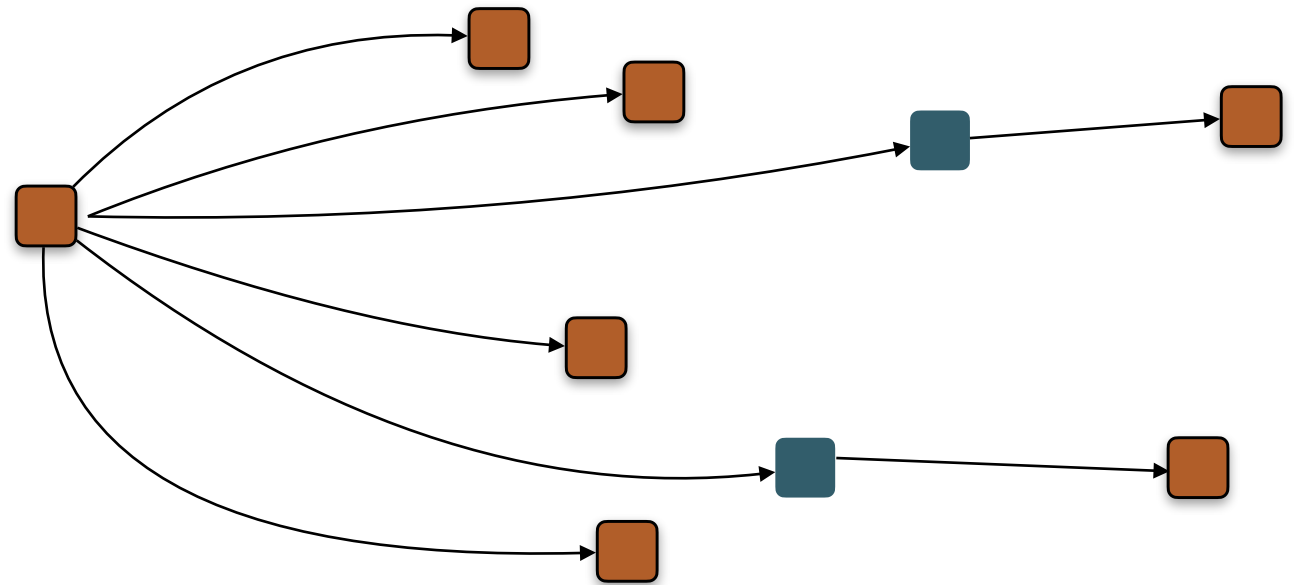


**Mehrpunkt?**

# Mehrpunktübertragung

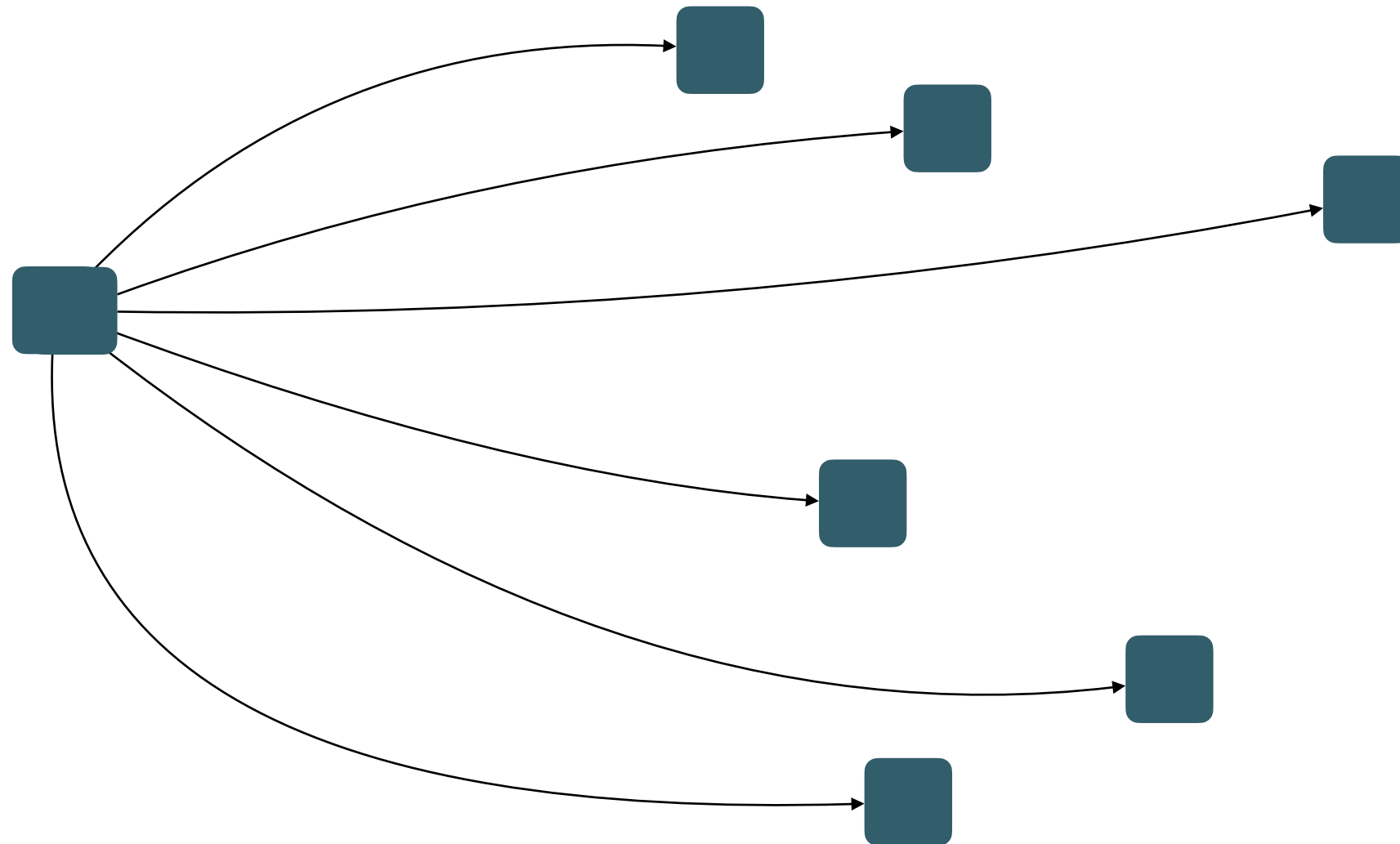
---

- Layer 1-4 sollen auch Mehrpunktübertragung ermöglichen
- Grundidee: Spezielle Empfängeradressen, um Übertragungsmodus festzulegen
- Keine Punkt-zu-Punkt Kommunikation, Konzept einer „Verbindung“ sinnlos
  - Teilnehmende Knoten wechseln beliebig, für den Sender nicht relevant
  - Effiziente Skalierung als primäre Aufgabe
- Mehrpunktübertragung = Versand und Verteilung einzelner Nachrichten



# Mehrpunktübertragung - Broadcast

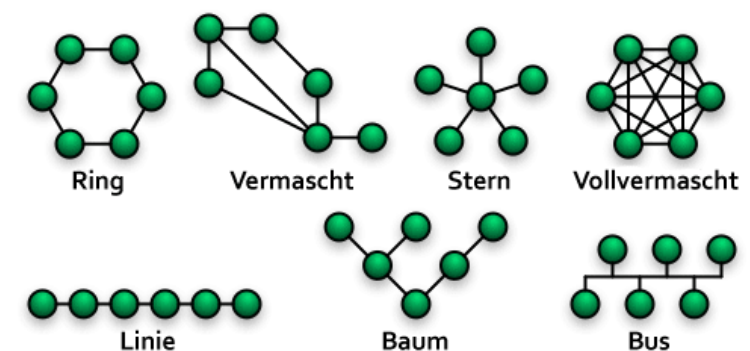
---



# Mehrpunktübertragung - Broadcast

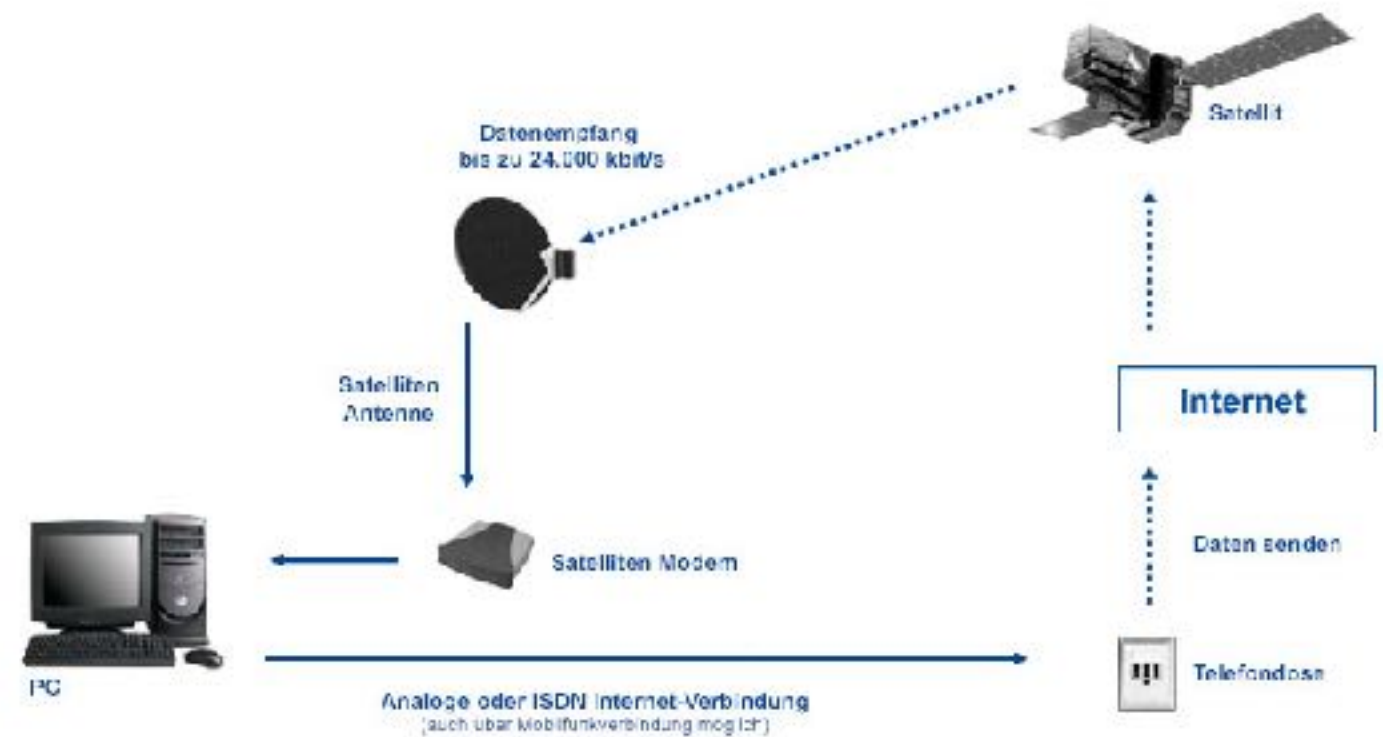
---

- Rundruf innerhalb der jeweiligen Topologie
  - **Alle direkt erreichbaren** Knoten werden angesprochen -> **kein Routing**
  - Pseudo-Adresse als Empfänger, gleicher Nachrichtenkopf
- Umsetzung hängt von der Topologie ab
  - Unterliegende Schicht beherrscht kein Broadcast -> **Flooding**
  - Ansonsten direkter gleichzeitiger Versand an alle
- Beispiel: Bus vs. Ring



# Klassisches Beispiel: Satelliten

- Geostationäre Umlaufbahn
- Verschiedene Radio- und Mikrowellenfrequenzen
- Latenz ca. 270ms
- Störungen durch Regen



- Für Telefonie, Internet, Ortsbestimmung, Wetteranalyse, Geoanalyse, ...
- Immer *broadcast*, daher ggf. Ende-zu-Ende Verschlüsselung notwendig
- Internet über Satellit: Nur für Download, Upload per Telefonleitung

# Ethernet Broadcast

---

- Empfängeradresse `FF : FF : FF : FF : FF : FF`
  - Bus: Alle Stationen fühlen sich angesprochen und empfangen
  - Switch: Empfangenes Paket wird an alle Ports weitergeleitet
  - Gut beobachtbar mit Wireshark (`eth.dst==FF : FF : FF : FF : FF : FF`)
- Menge aller erreichbaren Knoten: **Broadcast - Domäne**
  - Signifikante Belastung für das physische Netzwerk
  - Einschränkung der Domäne durch VLANs und Router

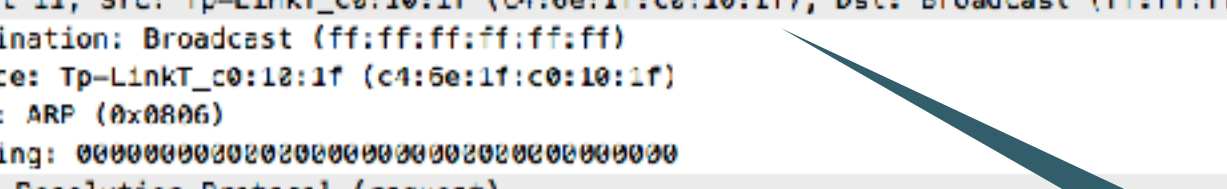


# Beispiel: Address Resolution Protocol (ARP)

---

- Problem: Finden der MAC - Adresse zu einer IP-Adresse **im eigenen LAN**
  - Computer startet zum ersten Mal nach der Netzwerkeinrichtung
  - Anwendung möchte IP-Paket in die Welt verschicken
  - Betriebssystem muss MAC des Routers bestimmen, aber nur seine IP bekannt
- Lösung: ARP-Anfrage per Ethernet Broadcast
  - Enthält die IP-Adresse der gesuchten Maschine als Teil der Anfrage
  - Antwort von Maschine mit Unicast, gesuchte MAC-Adresse als Absender
- Betriebssystem verwaltet bekannte Liste der Übersetzungen (siehe `man arp`)

No.	Time	Source	Destination	Protocol	Length	Info
7699	8.652573	AvmAudio_6...	Apple_3a:e...	ARP	60	Who has 192.168.178.177? Tell 192.168.178.1
7700	8.652617	Apple_3a:e...	AvmAudio_6...	ARP	42	192.168.178.177 is at a8:20:56:3a:e4:8e
30510	55.392683	AvmAudio_6...	Apple_3a:e...	ARP	60	Who has 192.168.178.177? Tell 192.168.178.1
30511	55.392733	Apple_3a:e...	AvmAudio_6...	ARP	42	192.168.178.177 is at a8:20:56:3a:e4:8e
43519	71.861786	Tp-LinkT_c...	Broadcast	ARP	60	Who has 192.168.178.1? Tell 192.168.178.43
43782	72.133789	Tp-LinkT_c...	Broadcast	ARP	60	Who has 192.168.178.43? Tell 0.0.0.0
44242	72.711325	Tp-LinkT_c...	Broadcast	ARP	60	Who has 192.168.178.177? Tell 192.168.178.43
44243	72.711371	Apple_3a:e...	Tp-LinkT_c...	ARP	42	192.168.178.177 is at a8:20:56:3a:e4:8e
44556	73.132922	Tp-LinkT_c...	Broadcast	ARP	60	Who has 192.168.178.43? Tell 0.0.0.0



```
▶ Frame 44242: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
▼ Ethernet II, Src: Tp-LinkT_c0:10:1f (c4:6e:1f:c0:10:1f), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  ▶ Destination: Broadcast (ff:ff:ff:ff:ff:ff)
  ▶ Source: Tp-LinkT_c0:10:1f (c4:6e:1f:c0:10:1f)
  Type: ARP (0x0806)
  Padding: 0000000002020000000000020200000000
▼ Address Resolution Protocol (request)
  Hardware type: Ethernet (1)
  Protocol type: IPv4 (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: request (1)
  Sender MAC address: Tp-LinkT_c0:10:1f (c4:6e:1f:c0:10:1f)
  Sender IP address: 192.168.178.43
  Target MAC address: 02:00:00_00:00:00 (00:00:00:00:00:00)
  Target IP address: 192.168.178.177
```

# Layer 2 Protokoll

## Ethernet-Empfänger

# Ethernet-Sender

# Layer 3 Protokoll

0000	ff	ff	ff	ff	ff	ff	c4	5e	1f	c0	10	1f	00	26	00	01	.....	n	.....
0010	28	00	05	04	00	01	c4	5e	1f	c0	10	1f	c0	a8	b2	2b	.....	n	.....+
0020	20	00	02	00	00	00	c0	38	b2	b1	00	02	00	20	00	02	.....		.....
0030	20	00	02	00	00	00	00	20	00	20	00	02					.....		.....

Ethernet: en0

arp

No.	Time	Source	Destination	Protocol	Length	Info
7699	8.652573	AvmAudio_6...	Apple_3a:e...	ARP	60	Who has 192.168.178.177? Tell 192.168.178.1
7700	8.652617	Apple_3a:e...	AvmAudio_6...	ARP	42	192.168.178.177 is at a8:20:66:3a:e4:8e
30510	55.392683	AvmAudio_6...	Apple_3a:e...	ARP	60	Who has 192.168.178.177? Tell 192.168.178.1
30511	55.392733	Apple_3a:e...	AvmAudio_6...	ARP	42	192.168.178.177 is at a8:20:66:3a:e4:8e
43519	71.861786	Tp-LinkT_c...	Broadcast	ARP	60	Who has 192.168.178.1? Tell 192.168.178.43
43782	72.133789	Tp-LinkT_c...	Broadcast	ARP	60	Who has 192.168.178.43? Tell 0.0.0.0
44242	72.711325	Tp-LinkT_c...	Broadcast	ARP	60	Who has 192.168.178.177? Tell 192.168.178.43
44243	72.711371	Apple_3a:e...	Tp-LinkT_c...	ARP	42	192.168.178.177 is at a8:20:66:3a:e4:8e
44556	73.132922	Tp-LinkT_c...	Broadcast	ARP	60	Who has 192.168.178.43? Tell 0.0.0.0

▶ Frame 44243: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0

▼ Ethernet II, Src: Apple\_3a:e4:8e (a8:20:66:3a:e4:8e), Dst: Tp-LinkT\_c0:10:1f (c4:6e:1f:c0:10:1f)

- ▶ Destination: Tp-LinkT\_c0:10:1f (c4:6e:1f:c0:10:1f)
- ▶ Source: Apple\_3a:e4:8e (a8:20:66:3a:e4:8e)
- Type: ARP (0x0806)

▼ Address Resolution Protocol (reply)

- Hardware type: Ethernet (1)
- Protocol type: IPv4 (0x0800)
- Hardware size: 6
- Protocol size: 4
- Opcode: reply (2)
- Sender MAC address: Apple\_3a:e4:8e (a8:20:66:3a:e4:8e)
- Sender IP address: 192.168.178.177
- Target MAC address: Tp-LinkT\_c0:10:1f (c4:6e:1f:c0:10:1f)
- Target IP address: 192.168.178.43

0000 c4 6e 1f c0 10 1f a8 20 65 3a e4 8e 00 06 00 01 -n..... f:.....

0010 00 00 05 04 00 02 a8 20 65 3a e4 8e c0 a8 b2 b1 ..... f:.....

0020 c4 6e 1f c0 10 1f c0 a8 b2 2b -n..... +

Address Resolution Protocol: Protocol

Pakete: 194538 · Angezeigt: 124 (0.1%) · Profil: Default

Ethernet-Empfänger

Ethernet-Sender

Layer 2 Protokoll

Layer 3 Protokoll

# Beispiel: ARP

---

- ARP entstand zusammen mit IPv4, keine Security berücksichtigt
- Problem des *ARP spoofing*
  - Angreifer bringt Maschine in Ethernet-Segment ein
  - Antwortet auf ARP-Anfragen mit eigener MAC-Adresse  
—> ***man-in-the-middle*** Attacke
- Technische Lösungen überwachen Netzwerk und prüfen ARP-Paket auf Plausibilität (zeitlicher Abstand, feste MAC-Liste, vergebene IP-Adressen)

# IPv4 Broadcast

---

- **Limited broadcast** - Zieladresse **255.255.255.255**
  - Direkte Übersetzung in Layer 1/2 - Broadcast
  - Bei Ethernet entsprechend FF:FF:FF:FF:FF:FF als Empfänger
- **Directed broadcast** - an alle Knoten in einem bestimmten IP-Subnetz
  - Bits der Geräte-ID in der Adresse werden auf 1 gesetzt
  - Beispiel 192.168.0.0/24 -> Broadcast-Adresse 192.168.0.255 für Subnetz
  - Routing in's Zielnetz mittlerweile untersagt (RFC 2644)
- Programmierung mit Sockets als **UDP Broadcast**

# Beispiel: Schlumpf-Attacke

---

- Identifikation eines Netzwerks, welches eingehende Broadcast-Pakete am Router akzeptiert
- Senden eines Broadcast-Pakets mit **gefälschter Absender-IP-Adresse**
- Beispiel Ping-Paket (*ICMP Echo Request*):  
Alle (!) Knoten antworten dem Absender
- Gewähltes Opfer wird mit Antwortpaketen überflutet
- Typische ***Distributed Denial of Service*** Attacke
- *Directed Broadcast* deshalb mittlerweile an Routern geblockt



[https://gabrieleswings.deviantart.com/art/evil-smurf-134146400]

# Beispiel:

## Dynamic Host Configuration Protocol (DHCP)

---

- Standard in allen modernen Netzen, besonders bei WLAN
- Automatische Zuweisung von IP-Adresse, Router-Adresse, DNS-Server, ...
- Falls DHCP-Server noch nicht bekannt:
  - *DHCPDISCOVER*: Client sendet UDP-Broadcast an 255.255.255.255:67
  - *DHCPOFFER*: Server antwortet mit Angebot an Port 68
  - *DHCPREQUEST*: Client fordert (erneute) Reservierung einer IP-Adresse
  - *DHCPACK*: Server bestätigt Reservierung



Ethernet: en0

(bootp.option.type == 53)

No.	Time	Source	Destination	Protocol	Length	Info
292...	40.017018	0.0.0.0	255.255.255.255	DHCP	363	DHCP Discover - Transaction ID 0x7d213e3a
292...	40.018013	192.168.178.1	255.255.255.255	DHCP	590	DHCP Offer - Transaction ID 0x7d213e3a
334...	41.146121	0.0.0.0	255.255.255.255	DHCP	363	DHCP Request - Transaction ID 0x7d213e3a
334...	41.147557	192.168.178.1	255.255.255.255	DHCP	590	DHCP ACK - Transaction ID 0x7d213e3a

Frame 29200: 363 bytes on wire (2904 bits), 363 bytes captured (2904 bits) on interface 0

Ethernet II, Src: Apple\_29:56:c5 (88:53:95:29:56:c5), Dst: Broadcast (ff:ff:ff:ff:ff:ff)

Internet Protocol Version 4, Src: 0.0.0.0, Dst: 255.255.255.255

User Datagram Protocol, Src Port: 68, Dst Port: 67

Bootstrap Protocol (Discover)

- Message type: Boot Request (1)
- Hardware type: Ethernet (0x01)
- Hardware address length: 6
- Hops: 0
- Transaction ID: 0x7d213e3a
- Seconds elapsed: 2
- Bootp flags: 0x0000, Broadcast flag (Broadcast)
- Client IP address: 0.0.0.0
- Your (client) IP address: 0.0.0.0
- Next server IP address: 0.0.0.0
- Relay agent IP address: 0.0.0.0
- Client MAC address: Apple\_29:56:c5 (88:53:95:29:56:c5)
- Client hardware address padding: 00000000000000000000
- Server host name not given
- Boot file name not given
- Magic cookies: DHCP
- Option: (53) DHCP Message Type (Discover)
- Option: (55) Parameter Request List
  - Length: 10
  - Parameter Request List Items: (1) Subnet Mask
  - Parameter Request List Items: (121) Classless Static Route
  - Parameter Request List Items: (3) Router
  - Parameter Request List Items: (6) Domain Name Server
  - Parameter Request List Items: (15) Domain Name
  - Parameter Request List Item: (119) Domain Search
  - Parameter Request List Item: (252) Private/Proxy autodiscovery
  - Parameter Request List Items: (95) LDAP [TODO:RFC3679]
  - Parameter Request List Items: (44) NetBIOS over TCP/IP Name Server
  - Parameter Request List Items: (46) NetBIOS over TCP/IP Node Type
- Option: (57) Maximum DHCP Message Size
- Option: (61) Client identifier
- Option: (51) IP Address Lease Time
- Option: (12) Host Name
- Option: (82) Agent Information Option
- Option: (255) End
- Padding: 200000002000000000000000000000000000

Ethernet-Empfänger

IPv4-Empfänger

angefragte Parameter

Bootstrap Protocol (bootp), 321 Bytes

Pakete: 55137 - Angezeigt: 10 (0.0%) - Verworfen: 0 (0.0%) - Profil: Default



IPv4-Sender

IPv4-Empfänger

Zugewiesene Adresse

Gültigkeitsdauer der Zuweisung

Zugewiesene Subnetz-Maske

Zugewiesener Router

Zugewiesener DNS-Server

Zugewiesener Domainen-Name

Ethernet: en0

(bootp.option.type == 53)

No.	Time	Source	Destination	Protocol	Length	Info
292...	40.017018	0.0.0.0	255.255.255.255	DHCP	363	DHCP Discover - Transaction ID 0x7d213e3a
292...	40.018013	192.168.178.1	255.255.255.255	DHCP	590	DHCP Offer - Transaction ID 0x7d213e3a
334...	41.146121	0.0.0.0	255.255.255.255	DHCP	363	DHCP Request - Transaction ID 0x7d213e3a
334...	41.147557	192.168.178.1	255.255.255.255	DHCP	590	DHCP ACK - Transaction ID 0x7d213e3a

Frame 33488: 363 bytes on wire (2904 bits), 363 bytes captured (2904 bits) on interface 0

Ethernet II, Src: Apple\_29:56:c5 (88:53:95:29:56:c5), Dst: Broadcast (ff:ff:ff:ff:ff:ff)

Internet Protocol Version 4, Src: 0.0.0.0, Dst: 255.255.255.255

User Datagram Protocol, Src Port: 68, Dst Port: 67

Bootstrap Protocol (Request)

- Message type: Boot Request (1)
- Hardware type: Ethernet (0x01)
- Hardware address length: 6
- Hops: 0
- Transaction ID: 0x7d213e3a
- Seconds elapsed: 3
- Bootp flags: 0x0000, Broadcast flag (Broadcast)
- Client IP address: 0.0.0.0
- Your (client) IP address: 0.0.0.0
- Next server IP address: 0.0.0.0
- Relay agent IP address: 0.0.0.0
- Client MAC address: Apple\_29:56:c5 (88:53:95:29:56:c5)
- Client hardware address padding: 00000000000000000000
- Server host name not given
- Boot file name not given
- Magic cookie: DHCP
- Option: (53) DHCP Message Type (Request)
- Option: (55) Parameter Request List
- Option: (57) Maximum DHCP Message Size
- Option: (61) Client identifier
- Option: (50) Requested IP Address
  - Length: 4
  - Requested IP Address: 192.168.178.41
- Option: (54) DHCP Server Identifier
  - Length: 4
  - DHCP Server Identifier: 192.168.178.1
- Option: (12) Host Name
  - Length: 7
  - Host Name: macmini
- Option: (82) Agent Information Option
- Option: (255) End
- Padding: 00000000000000000000

Ethernet-Empfänger

IPv4-Empfänger

angefragte Parameter

angefragte Adresse

Gewählter Host-Name

Bootstrap Protocol (bootp), 321 Bytes

Pakete: 55137 - Angezeigt: 10 (0.0%) - Verworfen: 0 (0.0%) - Profil: Default

IPv4-Sender

Sender	Source	Destination	Protocol	Length	Info
292...	192.168.178.1	255.255.255.255	DHCP	590	DHCP Offer - Transaction ID 0x7d213e3a
334...	41.146...	255.255.255.255	DHCP	363	DHCP Request - Transaction ID 0x7d213e3a
334...	41.147557	255.255.255.255	DHCP	590	DHCP ACK - Transaction ID 0x7d213e3a

[illegible]

IPv4-Empfänger

Zugewiesene Adresse

## Gültigkeitsdauer der Zuweisung

## Zugewiesene Subnetz-Maske

## Zugewiesener Router

## Zugewiesener DNS-Server

Zugewiesener Domainen-Name

# Beispiel: DHCP Server

---

```
# DHCP-Server ist die Autorität im Subnetz
authoritative;

# Definition des ersten Subnetzes
subnet 192.168.2.0 netmask 255.255.255.0 {
    range 192.168.2.10 192.168.2.40;
    default-lease-time 600;
    max-lease-time 7200;
    option domain-name "mein-tolles-netzwerk.de";
    option domain-name-servers 192.168.2.1;
    option broadcast-address 192.168.2.255;
    option subnet-mask 255.255.255.0;
    option routers 192.168.2.1;
}

# Zuweisung einer festen IP-Adresse, basierend auf MAC
gameserver {
    hardware ethernet 00:00:0e:d2:da:be;
    fixed-address 192.168.2.5;
    option host-name "gameserver";
}
```

# Beispiel: DHCP Server

---

- Erneuerung der Reservierung (*renew*): Client sendet periodisch DHCPREQUEST als Unicast-Nachricht nach Hälfte der *Lease-Time*
- Clients können freiwillig ihre Adresse aufgeben (*DHCPRELEASE*)
- Router kann als *DHCP Relay* arbeiten
  - Nimmt Broadcast-Pakete von Clients entgegen
  - Weiterleitung per Unicast an DHCP-Server in anderem Subnetz
- DHCP Clients akzeptieren jede Antwort —> Security-Problem
- *Authoritative Server*: Kennt alle IP-Adressen, lehnt falsche Reservierungen ab



# IPv4 vs. IPv6 Broadcast

---

- Bei Erfindung von IPv4 existierte noch keine Alternative zu Broadcast
  - Viele klassische Protokolle enthalten „Rundruf“ als essentiellen Teil
  - Bei größeren Netzen viel periodischer Datenverkehr
- IPv6 unterstützt kein direktes Broadcast mehr
  - Keine „Schnatterei“ mehr auf dem Übertragungsmedium
  - Vollständig ersetzt durch Multicast
  - Multicast an Gruppe mit allen Knoten = Broadcast

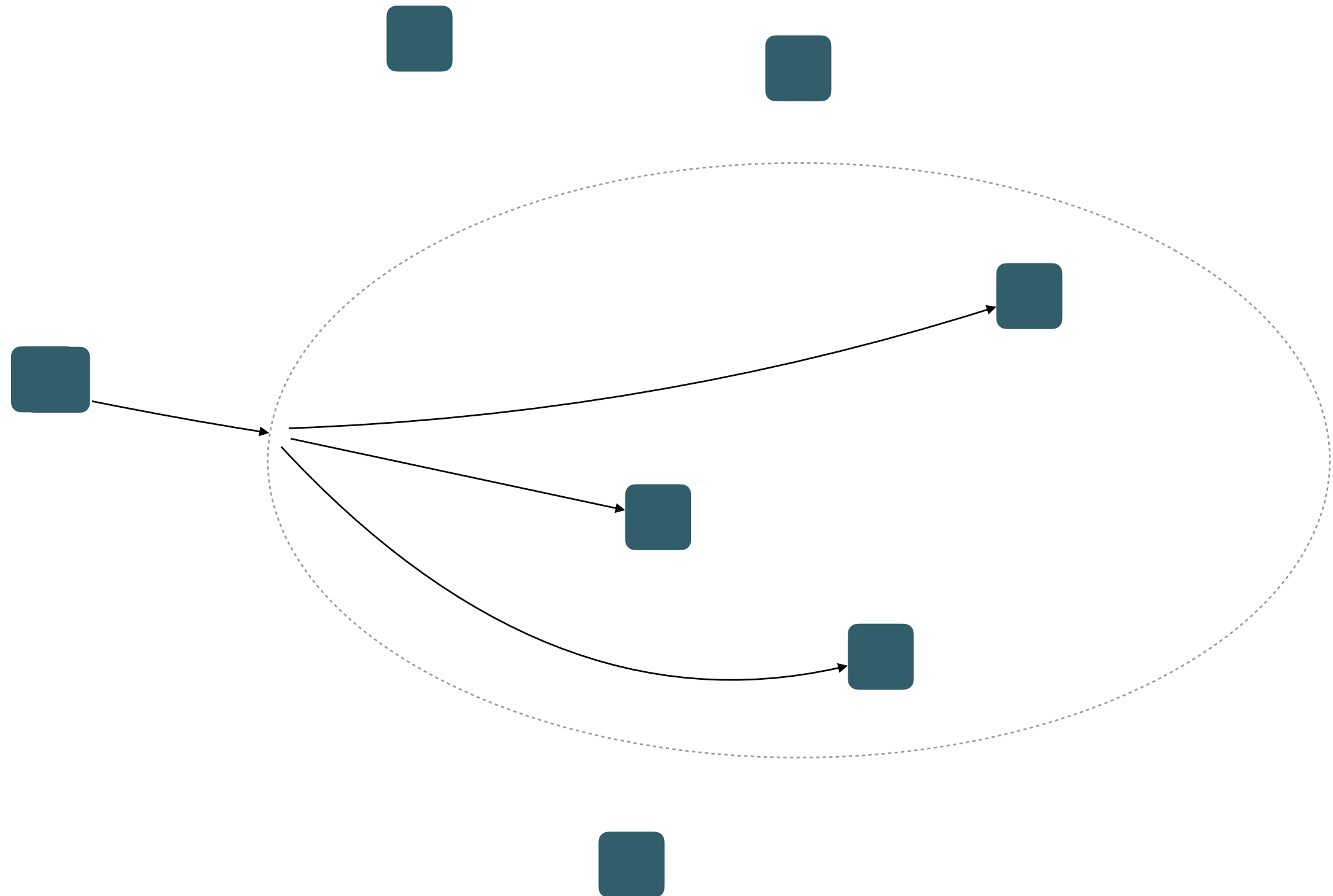
# Broadcast auf höheren Schichten

---

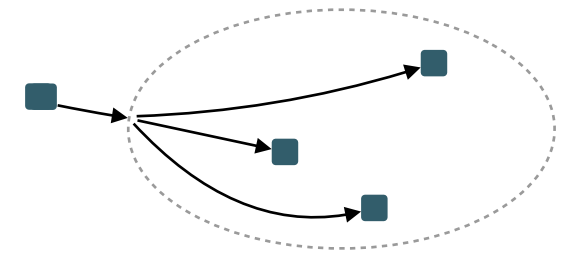
- IPv4 Broadcast nur im eigenen Subnetz
- Layer 7 IPv4 - Anwendungen nutzen deshalb **Flooding**
- Bsp. DSLP-Server
  - Server empfängt *group notify* - Nachricht auf TCP/IP-Verbindung
  - Schleife über Liste aller offenen TCP/IP-Verbindungen, jeweils *group notify* - Nachricht wieder versenden
  - Ergebnis: Broadcast auf Layer 7, Unicast auf Layer  $\leq 4$
- Effizientere globale Mehrpunktübertragung nur mit nativen IPv6 Multicast

# Mehrpunktübertragung - Multicast

---





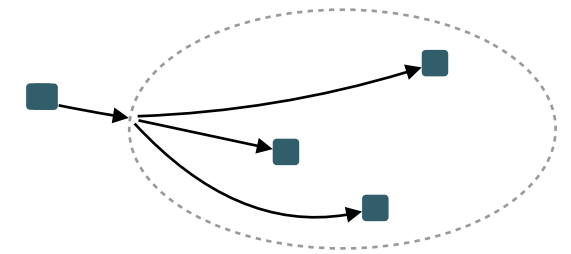


# Mehrpunktübertragung - IP Multicast

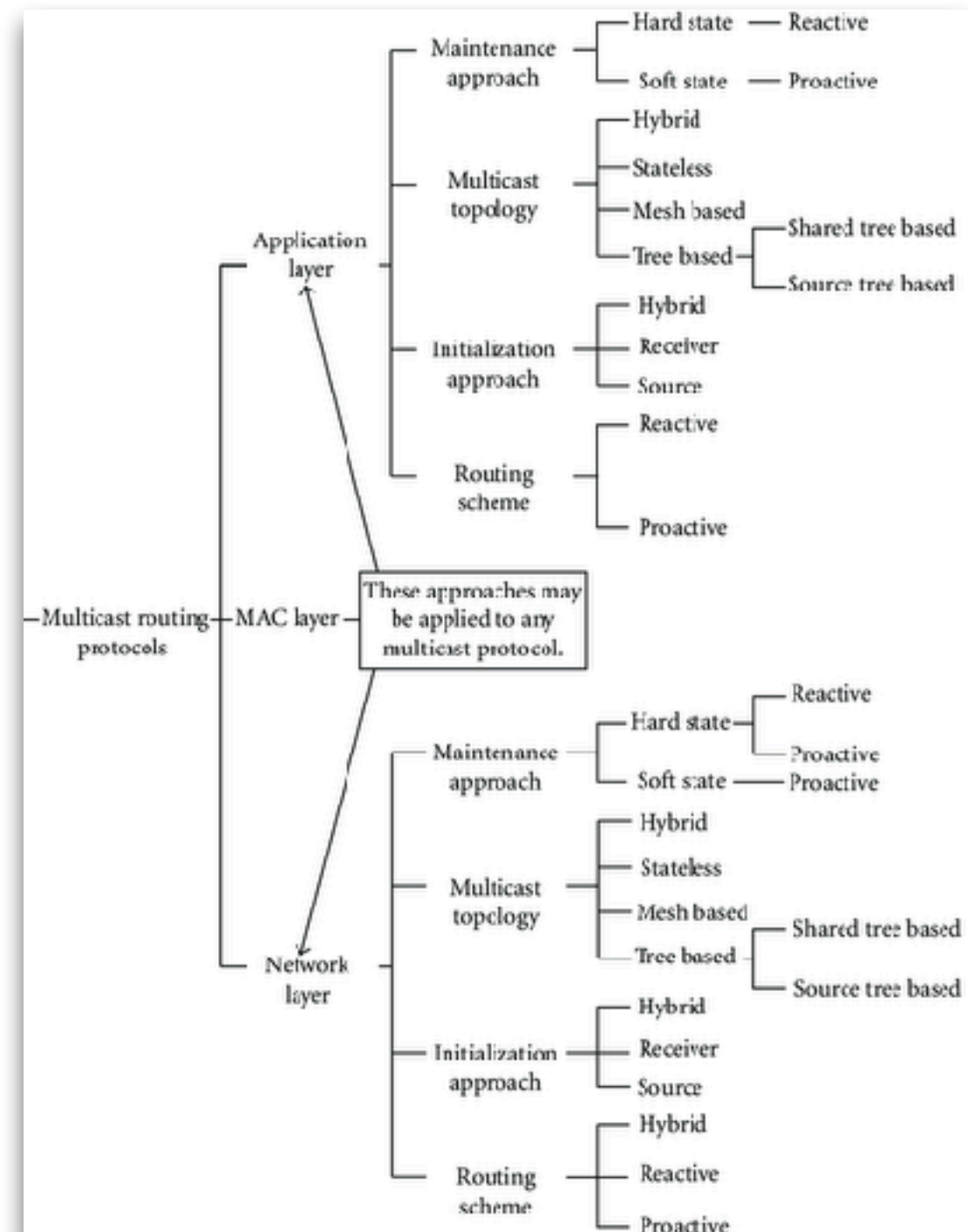
---

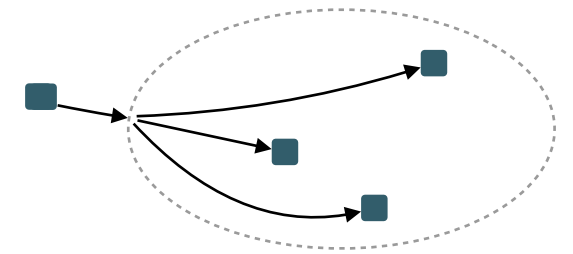
- Generischer Begriff aus der Nachrichtenübertragung
- Versand an eine **definierte** Gruppe von Knoten
- Wieder spezielle Zieladressen
  - IPv4: 224.0.0.0 bis 239.255.255.255
  - IPv6: ff::/8
- Knoten muss einer Multicast-Gruppe explizit beitreten, um Daten zu erhalten
- Senden von Datenpaketen ist auch ohne Beitritt möglich (Analogie Radio)

# Mehrpunktübertragung - IP Multicast



- Protokoll zwischen Routern für Informationsaustausch über Gruppenmitgliedschaft
- Ebenfalls Protokoll zur Ermittlung optimaler Routen nötig
- Wenn ein Router nicht mitspielt, funktioniert der Ansatz nicht mehr
- Viele Forschungsprojekte zur globalen Umsetzung in IPv4 -> als gescheitert betrachtet

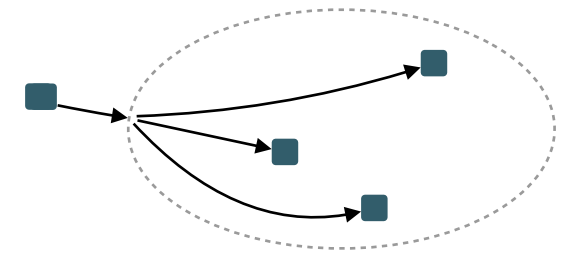




# Mehrpunktübertragung - IP Multicast

---

- *Internet Group Management Protocol (IGMP)*
  - Router sendet Nachfrage an „All Systems Group“ (224.0.0.1)
  - Alle 60-120 Sekunden
  - Ein Knoten antwortet mit „Host Membership Report“, Bericht über Gruppen und deren Mitglieder
  - Nachrichten von Mitgliedern beim Betreten und Verlassen einer Gruppe
  - Ausschliesslich Regelung der Mitgliedschaft, keine Routing-Entscheidungen



# Mehrpunktübertragung - IP Multicast

---

- Komplexe Protokolle, damit Router Multicast-Übertragung miteinander regeln
  - Fragestellung: Hier schon verteilen, oder erst weiterleiten?
  - Broadcast an benachbarte Router (*dense mode*)
  - Rendezvous - Router zur Verteilung (*sparse mode*)
- IGMP Snooping
  - Intelligenter Switch versteht IGMP - Pakete
  - Ordnet Multicast - Mitglieder entsprechend den Ports zu

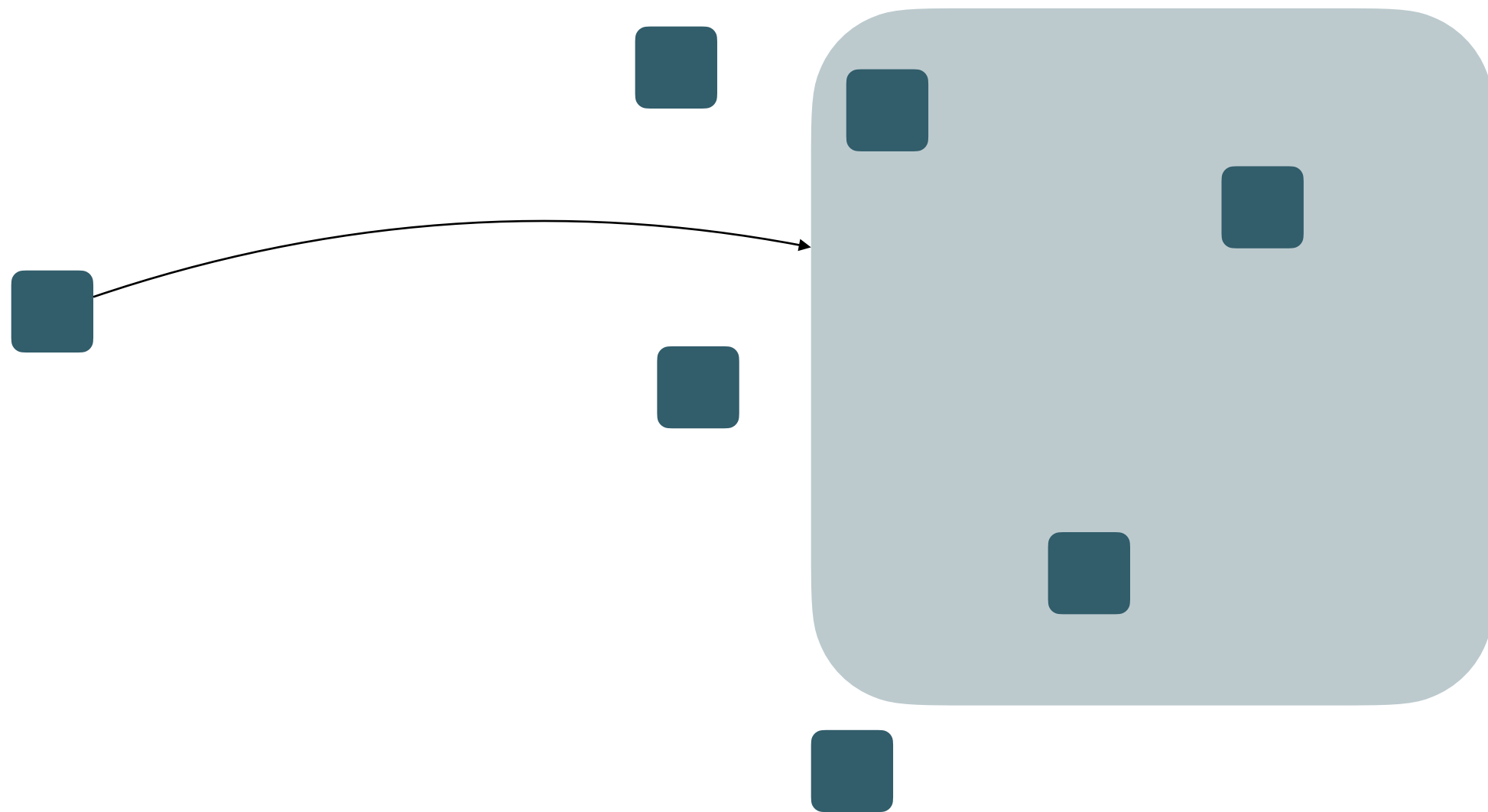
# Mehrpunktübertragung - IP Multicast

---

- Multicast in IPv4 - Netzen funktioniert nur verlässlich innerhalb des Subnetzes
  - Grundlage für einige Layer 7 - Protokolle (*ZeroConf, Bonjour, ...*)
  - Interessante Ausnahme: TV Streaming im Netz der Telekom
- Alternative: *Automated Multicast Tunneling (AMT)*
- IPv6
  - Broadcast im Standard nicht mehr unterstützt
  - Stabile Multicast Unterstützung - „aus den Fehlern gelernt“

# Mehrpunktübertragung - Anycast

---



# Mehrpunktübertragung - Anycast

---

- Spezielle Lösung beim Routing
- „Normale“ IP-Zieladresse hat mehrere Routen und mehrere Endpunkte
- Kriterien: Anzahl der Hops, Distanz zum Ziel, Latenz, Auslastung
- Nicht speziell in IPv4 vorgesehen, aber in IPv6 (RFC 4291)
- IPv4: Kreative Nutzung des **Border Gateway Protocol (BGP)**
  - Dient sowieso zum Austausch von Information zwischen IPv4 - Routern
  - Verschiedene Router bekommen verschiedene Routen zum gleichen Ziel

# Mehrpunktübertragung - Anycast

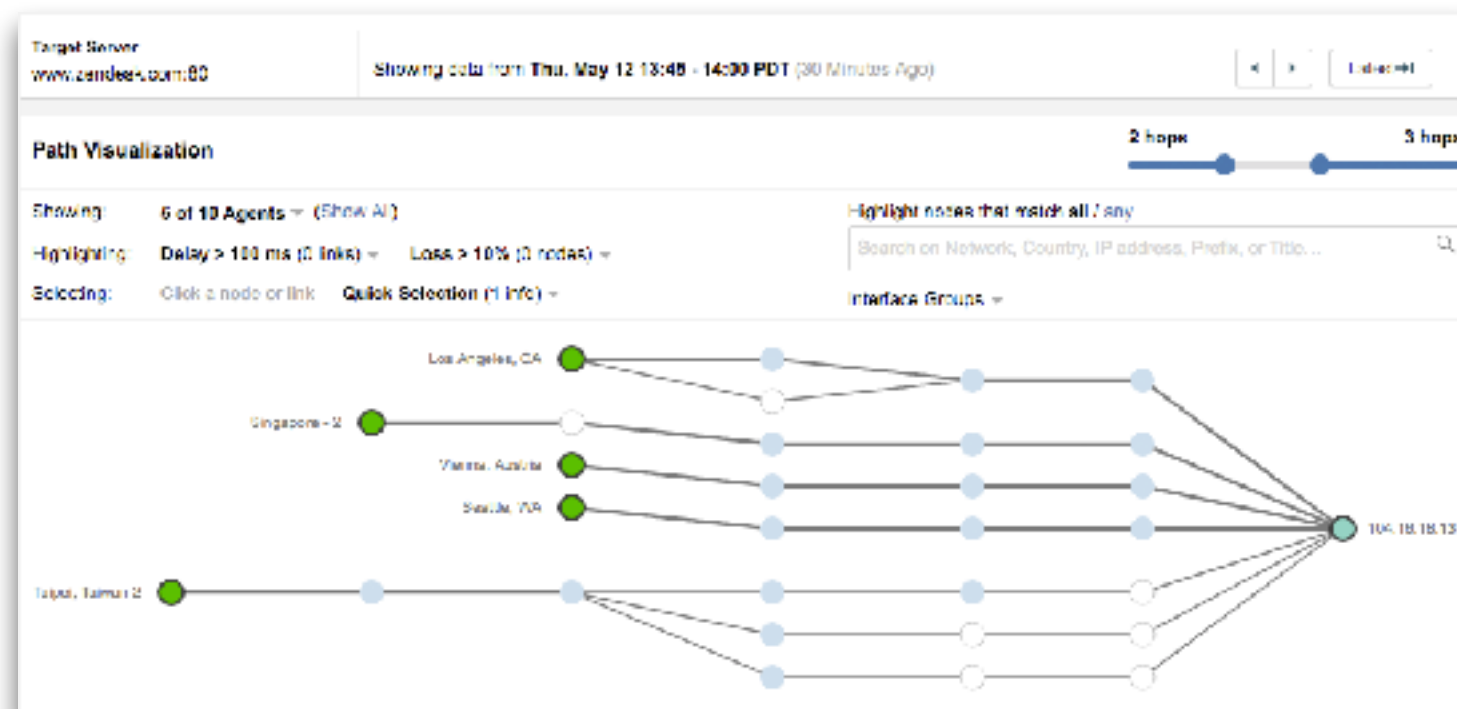
---

- Grundlage für ausfallsichere skalierbare Serverdienste (z.B. YouTube)
  - Server wird an mehreren Standorten repliziert
  - Anycast - Routing verteilt die Anfragen an Server-IP über mehrere Kopien
  - Beispiel DDoS-Angriff: Nur der naheliegende Server wird angegriffen
- Physischer Endpunkt kann sich mit jedem IP-Datagramm ändern, trotz gleichbleibender Zieladresse
  - Eigentlich nur sinnvoll mit UDP
  - Funktioniert trotzdem für kurzlebige TCP-Verbindungen (HTTP!)



# Beispiel: Content Delivery Network (CDN)

- Anbieter mit tausenden Servern, weltweit verteilt
- Kunden lagern statische Inhalte (Bilder, Downloads etc.) im CDN, gleichbleibende IP-Adresse via Anycast
- Anfragen werden gleichmäßig verteilt, je nach Standort des Client
- Bsp: Cloudflare, Edgecast



# Zusammenfassung

---

- Klassische Netzwerkprogrammierung nimmt *Unicast*-Übertragung an
- Alternativen: *Broadcast*, *Multicast*
  - Gruppen von Empfängern
  - Setzt Mitarbeit des Routers voraus, für Switch kein Problem
  - Übertragung einzelner Nachrichten (UDP-Stil), mehr Arbeit für die Anwendung
  - IPv4 mit Fokus auf *Broadcast* (ARP, DHCP)
  - IPv6 mit Fokus auf *Multicast*
- Anycast: Moderner Routing-Ansatz für Lastenverteilung und Fehlertoleranz