

Was Sie schon immer über Netze wissen wollten

Die (fast) komplette Netzwerktechnik in 2 Tagen

Michael Dienert, Peter Maaß

Walther-Rathenau-Gewerbeschule
Freiburg

19. November 2013

Inhalt

Vom LAN zum Internet

Rahmen, Pakete, virtuelle Verbindungen

Versuch: Daten mit nc übertragen und Datenverkehr mitschneiden

IP-Adressen und Subnetze

Address Resolution Protocol

Routing

Entstehung des Internets

- Entstehung in den USA
- Idee: dezentrale Kommunikationsstruktur
- Verbindungen zwischen Computersystemen schaffen
- Staatliche Unterstützung der Entwicklung: Department of Defense, **DOD**
- Arbeitsgruppe innerhalb des DOD: US Defense Advanced Research Projects Agency, **DARPA**
- 1969 (!) erste Experimente mit ARPANET; regulärer Betrieb ab 1975
- 1983 werden TCP und IP zu den Standardprotokollen des **Internets**

Entstehung des Internets

- Entstehung in den USA
- Idee: dezentrale Kommunikationsstruktur
- Verbindungen zwischen Computersystemen schaffen
- Staatliche Unterstützung der Entwicklung: Department of Defense, **DOD**
- Arbeitsgruppe innerhalb des DOD: US Defense Advanced Research Projects Agency, **DARPA**
- 1969 (!) erste Experimente mit ARPANET; regulärer Betrieb ab 1975
- 1983 werden TCP und IP zu den Standardprotokollen des **Internets**

Entstehung des Internets

- Entstehung in den USA
- Idee: dezentrale Kommunikationsstruktur
- Verbindungen zwischen Computersystemen schaffen
- Staatliche Unterstützung der Entwicklung: Department of Defense, **DOD**
- Arbeitsgruppe innerhalb des DOD: US Defense Advanced Research Projects Agency, **DARPA**
- 1969 (!) erste Experimente mit ARPANET; regulärer Betrieb ab 1975
- 1983 werden TCP und IP zu den Standardprotokollen des **Internets**

Entstehung des Internets

- Entstehung in den USA
- Idee: dezentrale Kommunikationsstruktur
- Verbindungen zwischen Computersystemen schaffen
- Staatliche Unterstützung der Entwicklung: Department of Defense, **DOD**
- Arbeitsgruppe innerhalb des DOD: US Defense Advanced Research Projects Agency, **DARPA**
- 1969 (!) erste Experimente mit ARPANET; regulärer Betrieb ab 1975
- 1983 werden TCP und IP zu den Standardprotokollen des **Internets**

Entstehung des Internets

- Entstehung in den USA
- Idee: dezentrale Kommunikationsstruktur
- Verbindungen zwischen Computersystemen schaffen
- Staatliche Unterstützung der Entwicklung: Department of Defense, **DOD**
- Arbeitsgruppe innerhalb des DOD: US Defense Advanced Research Projects Agency, **DARPA**
- 1969 (!) erste Experimente mit ARPANET; regulärer Betrieb ab 1975
- 1983 werden TCP und IP zu den Standardprotokollen des **Internets**

Entstehung des Internets

- Entstehung in den USA
- Idee: dezentrale Kommunikationsstruktur
- Verbindungen zwischen Computersystemen schaffen
- Staatliche Unterstützung der Entwicklung: Department of Defense, **DOD**
- Arbeitsgruppe innerhalb des DOD: US Defense Advanced Research Projects Agency, **DARPA**
- 1969 (!) erste Experimente mit ARPANET; regulärer Betrieb ab 1975
- 1983 werden TCP und IP zu den Standardprotokollen des **Internets**

Entstehung des Internets

- Entstehung in den USA
- Idee: dezentrale Kommunikationsstruktur
- Verbindungen zwischen Computersystemen schaffen
- Staatliche Unterstützung der Entwicklung: Department of Defense, **DOD**
- Arbeitsgruppe innerhalb des DOD: US Defense Advanced Research Projects Agency, **DARPA**
- 1969 (!) erste Experimente mit ARPANET; regulärer Betrieb ab 1975
- 1983 werden TCP und IP zu den Standardprotokollen des **Internets**

Entstehung des Internets

- Entstehung in den USA
- Idee: dezentrale Kommunikationsstruktur
- Verbindungen zwischen Computersystemen schaffen
- Staatliche Unterstützung der Entwicklung: Department of Defense, **DOD**
- Arbeitsgruppe innerhalb des DOD: US Defense Advanced Research Projects Agency, **DARPA**
- 1969 (!) erste Experimente mit ARPANET; regulärer Betrieb ab 1975
- 1983 werden TCP und IP zu den Standardprotokollen des **Internets**

Entstehung des Internets

- Entstehung in den USA
- Idee: dezentrale Kommunikationsstruktur
- Verbindungen zwischen Computersystemen schaffen
- Staatliche Unterstützung der Entwicklung: Department of Defense, **DOD**
- Arbeitsgruppe innerhalb des DOD: US Defense Advanced Research Projects Agency, **DARPA**
- 1969 (!) erste Experimente mit ARPANET; regulärer Betrieb ab 1975
- 1983 werden TCP und IP zu den Standardprotokollen des **Internets**

Entstehung des Internets

- Entstehung in den USA
- Idee: dezentrale Kommunikationsstruktur
- Verbindungen zwischen Computersystemen schaffen
- Staatliche Unterstützung der Entwicklung: Department of Defense, **DOD**
- Arbeitsgruppe innerhalb des DOD: US Defense Advanced Research Projects Agency, **DARPA**
- 1969 (!) erste Experimente mit ARPANET; regulärer Betrieb ab 1975
- 1983 werden TCP und IP zu den Standardprotokollen des **Internets**

Entstehung des Internets

- Entstehung in den USA
- Idee: dezentrale Kommunikationsstruktur
- Verbindungen zwischen Computersystemen schaffen
- Staatliche Unterstützung der Entwicklung: Department of Defense, **DOD**
- Arbeitsgruppe innerhalb des DOD: US Defense Advanced Research Projects Agency, **DARPA**
- 1969 (!) erste Experimente mit ARPANET; regulärer Betrieb ab 1975
- 1983 werden TCP und IP zu den Standardprotokollen des **Internets**

Wo kommt das i her



Berkeley
University

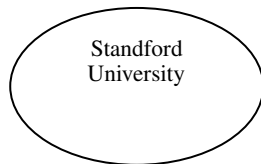
Wo kommt das i her



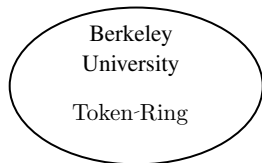
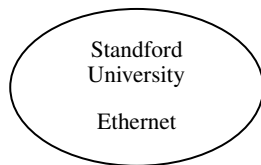
Berkeley
University

Token-Ring

Wo kommt das i her



Wo kommt das i her



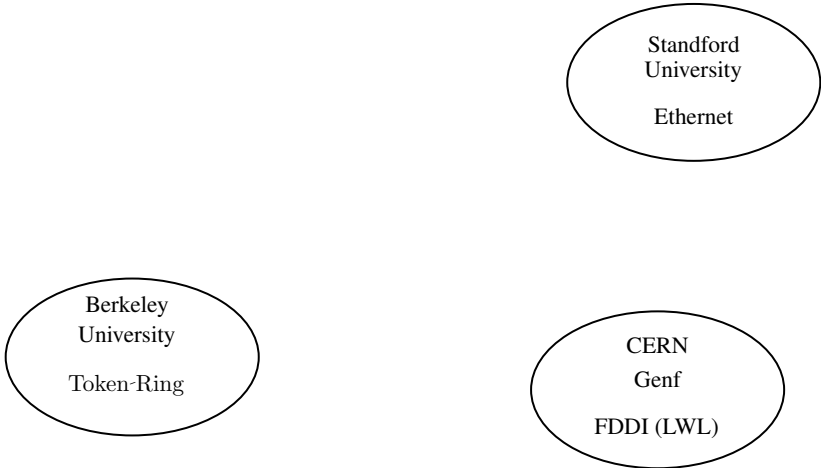
Wo kommt das i her

Stanford
University
Ethernet

Berkeley
University
Token-Ring

CERN
Genf

Wo kommt das i her



Stanford
University

Ethernet

Berkeley
University

Token-Ring

CERN

Genf

FDDI (LWL)

Wo kommt das i her

ARPA
DOD

Stanford
University

Ethernet

Berkeley
University

Token-Ring

CERN
Genf

FDDI (LWL)

Wo kommt das i her

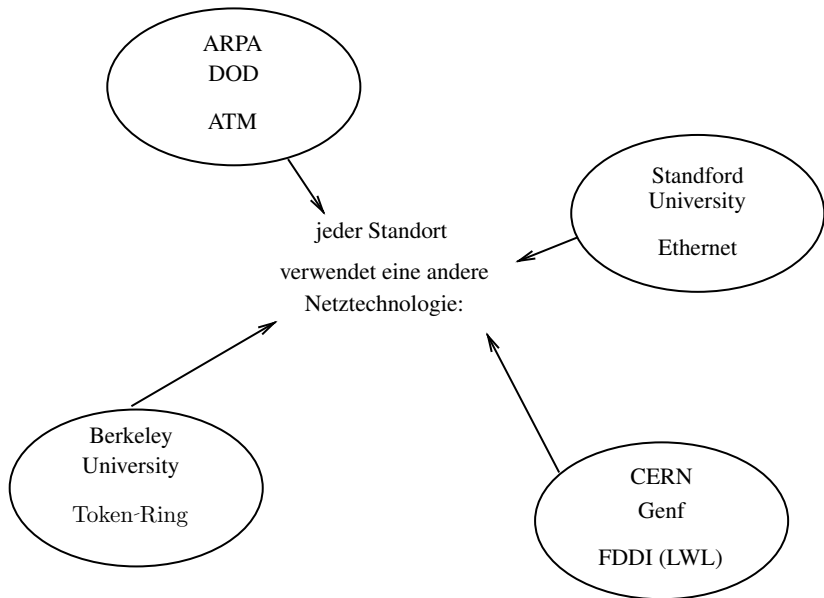
ARPA
DOD
ATM

Stanford
University
Ethernet

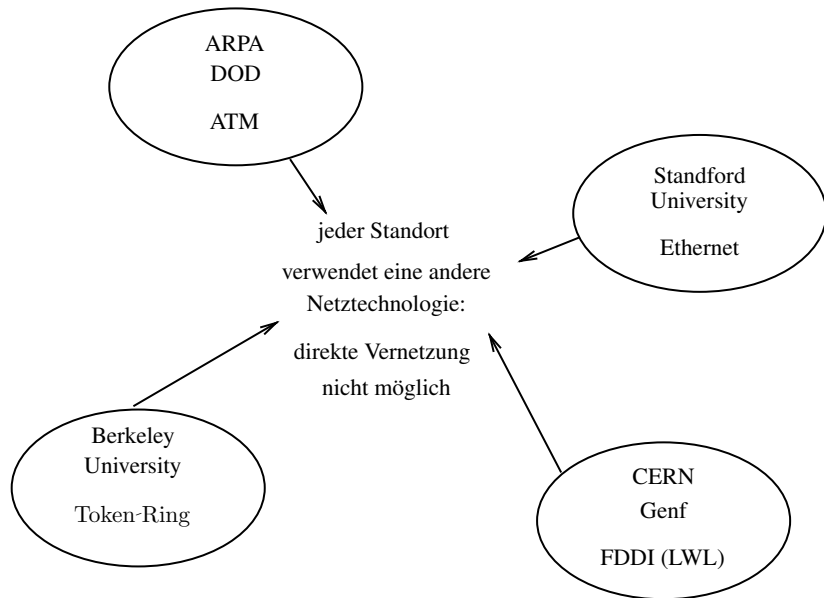
Berkeley
University
Token-Ring

CERN
Genf
FDDI (LWL)

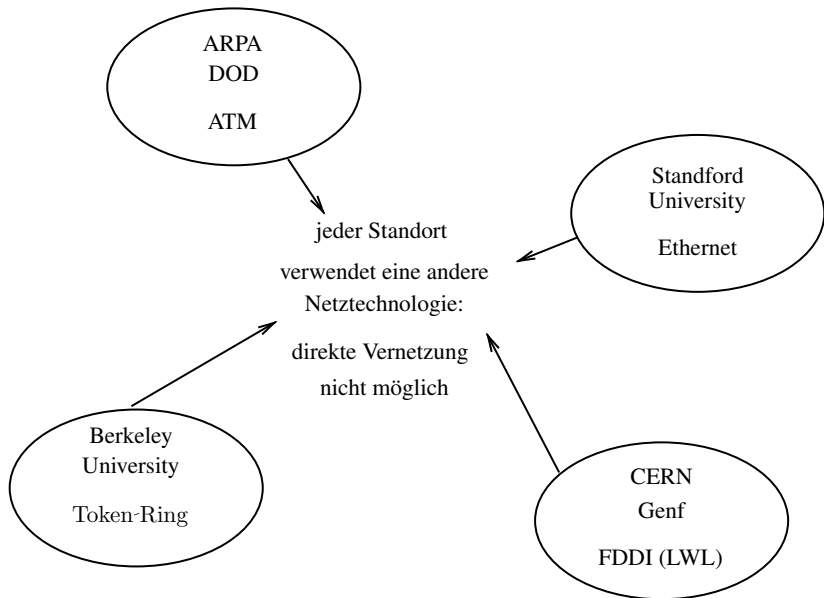
Wo kommt das i her



Wo kommt das i her



Wo kommt das i her



Ein hardwareunabhängiges Adressschema

- Jeder Standort hat eine eigene Netzhardware \Rightarrow Eine direkte Vernetzung ist nicht möglich
- Beispiel:
Ethernet-Adressen 01001000 00101100 011...
Token-Ring-Adressen 00010010 00110100 010...
 \Rightarrow die **Hardware**-Adressen von Ethernet- und Token-Ring-Geräten werden auf dem Netzwerkmedium völlig verschieden übertragen.
- **Lösung des Problems:** Es wird ein **zusätzliches, weltweit einheitliches, hardwareunabhängiges** Adressschema geschaffen
- Vernetzung unterschiedlicher Standorte wird möglich

Ein hardwareunabhängiges Adressschema

- Jeder Standort hat eine eigene Netzhardware ⇒ Eine direkte Vernetzung ist nicht möglich

- Beispiel:

Ethernet-Adressen 01001000 00101100 011...

Token-Ring-Adressen 00010010 00110100 010...

⇒ die **Hardware**-Adressen von Ethernet- und Token-Ring-Geräten werden auf dem Netzwerkmedium völlig verschieden übertragen.

- **Lösung des Problems:** Es wird ein **zusätzliches, weltweit einheitliches, hardwareunabhängiges Adressschema** geschaffen
- Vernetzung unterschiedlicher Standorte wird möglich

Ein hardwareunabhängiges Adressschema

- Jeder Standort hat eine eigene Netzhardware \Rightarrow Eine direkte Vernetzung ist nicht möglich

- Beispiel:

Ethernet-Adressen 01001000 00101100 011...

Token-Ring-Adressen 00010010 00110100 010...

\Rightarrow die **Hardware**-Adressen von Ethernet- und Token-Ring-Geräten werden auf dem Netzwerkmedium völlig verschieden übertragen.

- **Lösung des Problems:** Es wird ein **zusätzliches, weltweit einheitliches, hardwareunabhängiges Adressschema** geschaffen
- Vernetzung unterschiedlicher Standorte wird möglich

Ein hardwareunabhängiges Adressschema

- Jeder Standort hat eine eigene Netzhardware \Rightarrow Eine direkte Vernetzung ist nicht möglich

- Beispiel:

Ethernet-Adressen 01001000 00101100 011...

Token-Ring-Adressen 00010010 00110100 010...

\Rightarrow die **Hardware**-Adressen von Ethernet- und Token-Ring-Geräten werden auf dem Netzwerkmedium völlig verschieden übertragen.

- **Lösung des Problems:** Es wird ein **zusätzliches, weltweit einheitliches, hardwareunabhängiges Adressschema** geschaffen
- Vernetzung unterschiedlicher Standorte wird möglich

Ein hardwareunabhängiges Adressschema

- Jeder Standort hat eine eigene Netzhardware \Rightarrow Eine direkte Vernetzung ist nicht möglich

- Beispiel:

Ethernet-Adressen 01001000 00101100 011...

Token-Ring-Adressen 00010010 00110100 010...

\Rightarrow die **Hardware**-Adressen von Ethernet- und Token-Ring-Geräten werden auf dem Netzwerkmedium völlig verschieden übertragen.

- **Lösung des Problems:** Es wird ein **zusätzliches, weltweit einheitliches, hardwareunabhängiges Adressschema** geschaffen
- Vernetzung unterschiedlicher Standorte wird möglich

Ein hardwareunabhängiges Adressschema

- Jeder Standort hat eine eigene Netzhardware \Rightarrow Eine direkte Vernetzung ist nicht möglich

- Beispiel:

Ethernet-Adressen 01001000 00101100 011...

Token-Ring-Adressen 00010010 00110100 010...

\Rightarrow die **Hardware**-Adressen von Ethernet- und Token-Ring-Geräten werden auf dem Netzwerkmedium völlig verschieden übertragen.

- **Lösung des Problems:** Es wird ein **zusätzliches, weltweit einheitliches, hardwareunabhängiges Adressschema** geschaffen
- Vernetzung unterschiedlicher Standorte wird möglich

Ein hardwareunabhängiges Adressschema

- Jeder Standort hat eine eigene Netzhardware \Rightarrow Eine direkte Vernetzung ist nicht möglich

- Beispiel:

Ethernet-Adressen 01001000 00101100 011...

Token-Ring-Adressen 00010010 00110100 010...

\Rightarrow die **Hardware**-Adressen von Ethernet- und Token-Ring-Geräten werden auf dem Netzwerkmedium völlig verschieden übertragen.

- **Lösung des Problems:** Es wird ein **zusätzliches, weltweit einheitliches, hardwareunabhängiges Adressschema** geschaffen
- Vernetzung unterschiedlicher Standorte wird möglich

Ein hardwareunabhängiges Adressschema

- Jeder Standort hat eine eigene Netzhardware \Rightarrow Eine direkte Vernetzung ist nicht möglich

- Beispiel:

Ethernet-Adressen 01001000 00101100 011...

Token-Ring-Adressen 00010010 00110100 010...

\Rightarrow die **Hardware**-Adressen von Ethernet- und Token-Ring-Geräten werden auf dem Netzwerkmedium völlig verschieden übertragen.

- **Lösung des Problems:** Es wird ein **zusätzliches, weltweit einheitliches, hardwareunabhängiges Adressschema** geschaffen
- Vernetzung unterschiedlicher Standorte wird möglich

Ein hardwareunabhängiges Adressschema

- Jeder Standort hat eine eigene Netzhardware \Rightarrow Eine direkte Vernetzung ist nicht möglich

- Beispiel:

Ethernet-Adressen 01001000 00101100 011...

Token-Ring-Adressen 00010010 00110100 010...

\Rightarrow die **Hardware**-Adressen von Ethernet- und Token-Ring-Geräten werden auf dem Netzwerkmedium völlig verschieden übertragen.

- **Lösung des Problems:** Es wird ein **zusätzliches, weltweit einheitliches, hardwareunabhängiges Adressschema** geschaffen
- Vernetzung unterschiedlicher Standorte wird möglich

Ein hardwareunabhängiges Adressschema

- Jeder Standort hat eine eigene Netzhardware \Rightarrow Eine direkte Vernetzung ist nicht möglich

- Beispiel:

Ethernet-Adressen 01001000 00101100 011...

Token-Ring-Adressen 00010010 00110100 010...

\Rightarrow die **Hardware**-Adressen von Ethernet- und Token-Ring-Geräten werden auf dem Netzwerkmedium völlig verschieden übertragen.

- **Lösung des Problems:** Es wird ein **zusätzliches, weltweit einheitliches, hardwareunabhängiges Adressschema** geschaffen
- Vernetzung unterschiedlicher Standorte wird möglich

Ein hardwareunabhängiges Adressschema

- Jeder Standort hat eine eigene Netzhardware \Rightarrow Eine direkte Vernetzung ist nicht möglich

- Beispiel:

Ethernet-Adressen 01001000 00101100 011...

Token-Ring-Adressen 00010010 00110100 010...

\Rightarrow die **Hardware**-Adressen von Ethernet- und Token-Ring-Geräten werden auf dem Netzwerkmedium völlig verschieden übertragen.

- **Lösung des Problems:** Es wird ein **zusätzliches, weltweit einheitliches, hardwareunabhängiges Adressschema** geschaffen
- Vernetzung unterschiedlicher Standorte wird möglich

Ein hardwareunabhängiges Adressschema

- Jeder Standort hat eine eigene Netzhardware \Rightarrow Eine direkte Vernetzung ist nicht möglich

- Beispiel:

Ethernet-Adressen 01001000 00101100 011...

Token-Ring-Adressen 00010010 00110100 010...

\Rightarrow die **Hardware**-Adressen von Ethernet- und Token-Ring-Geräten werden auf dem Netzwerkmedium völlig verschieden übertragen.

- **Lösung des Problems:** Es wird ein **zusätzliches, weltweit einheitliches, hardwareunabhängiges** Adressschema geschaffen
- Vernetzung unterschiedlicher Standorte wird möglich

Ein hardwareunabhängiges Adressschema

- Jeder Standort hat eine eigene Netzhardware \Rightarrow Eine direkte Vernetzung ist nicht möglich

- Beispiel:

Ethernet-Adressen 01001000 00101100 011...

Token-Ring-Adressen 00010010 00110100 010...

\Rightarrow die **Hardware**-Adressen von Ethernet- und Token-Ring-Geräten werden auf dem Netzwerkmedium völlig verschieden übertragen.

- **Lösung des Problems:** Es wird ein **zusätzliches, weltweit einheitliches, hardwareunabhängiges Adressschema** geschaffen
- Vernetzung unterschiedlicher Standorte wird möglich

Ein hardwareunabhängiges Adressschema

- Jeder Standort hat eine eigene Netzhardware \Rightarrow Eine direkte Vernetzung ist nicht möglich

- Beispiel:

Ethernet-Adressen 01001000 00101100 011...

Token-Ring-Adressen 00010010 00110100 010...

\Rightarrow die **Hardware**-Adressen von Ethernet- und Token-Ring-Geräten werden auf dem Netzwerkmedium völlig verschieden übertragen.

- **Lösung des Problems:** Es wird ein **zusätzliches, weltweit einheitliches, hardwareunabhängiges Adressschema** geschaffen
- Vernetzung unterschiedlicher Standorte wird möglich

Ein hardwareunabhängiges Adressschema

- Jeder Standort hat eine eigene Netzhardware \Rightarrow Eine direkte Vernetzung ist nicht möglich

- Beispiel:

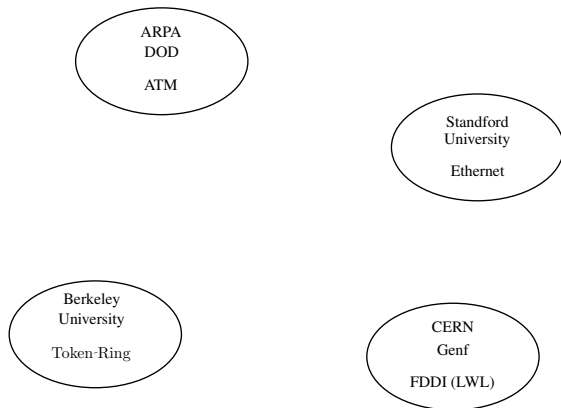
Ethernet-Adressen 01001000 00101100 011...

Token-Ring-Adressen 00010010 00110100 010...

\Rightarrow die **Hardware**-Adressen von Ethernet- und Token-Ring-Geräten werden auf dem Netzwerkmedium völlig verschieden übertragen.

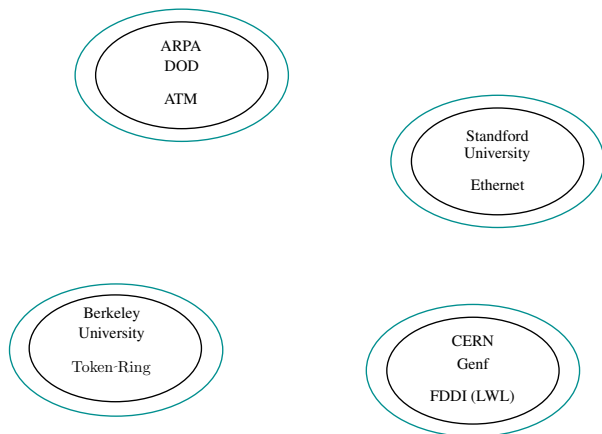
- **Lösung des Problems:** Es wird ein **zusätzliches, weltweit einheitliches, hardwareunabhängiges** Adressschema geschaffen
- Vernetzung unterschiedlicher Standorte wird möglich

Ein hardwareunabhängiges Adressschema



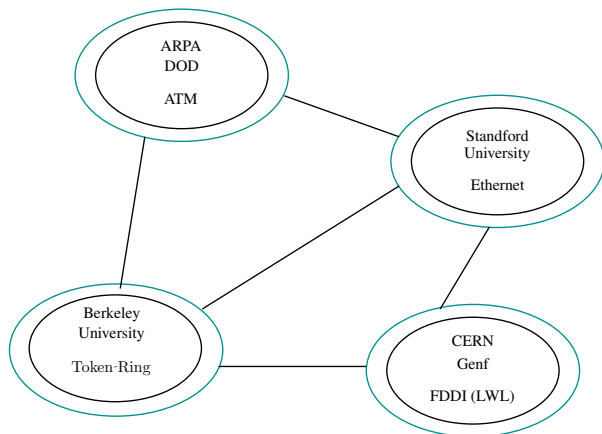
Ein hardwareunabhängiges Adressschema

Weltweit einheitliches, hardwareunabhängiges Adressschema



Ein hardwareunabhängiges Adressschema

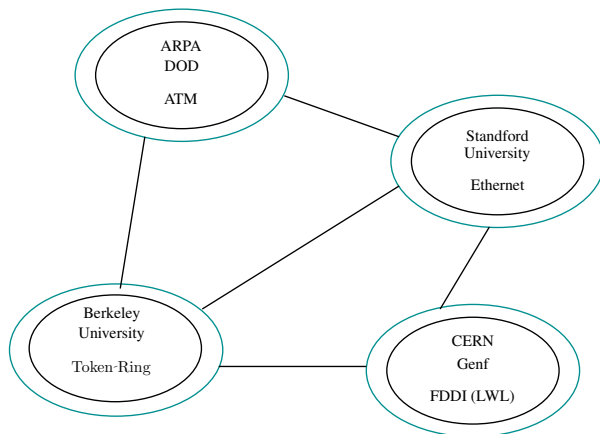
Weltweit einheitliches, hardwareunabhängiges Adressschema



Vernetzung zwischen den lokalen Netzen wird möglich

Ein hardwareunabhängiges Adressschema

Weltweit einheitliches, hardwareunabhängiges Adressschema

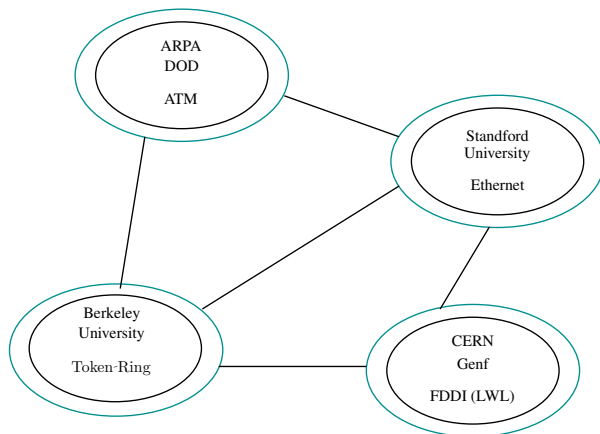


Vernetzung zwischen den lokalen Netzen wird möglich

Es entsteht ein Inter-LAN-Netzwerk: Internet

Ein hardwareunabhängiges Adressschema

Weltweit einheitliches, hardwareunabhängiges Adressschema



Vernetzung zwischen den lokalen Netzen wird möglich

Es entsteht ein Inter-LAN-Netzwerk: Internet

Das Internet-Protokoll Version 4 und seine Adressen

- Protokoll für den Datenaustausch im Internet:
Internet-Protokoll
- 1980: DoD Standard Internet Protocol **IPv3**
- IPv3 verwendete **32-Bit Internet-Protokoll-Adressen**
- 1981 wird **IPv4** eingeführt, ebenfalls mit 32-Bit IP-Adressen.
- Adressraum ist ca. 4.3 Milliarden Adressen gross, seit 2011 erschöpft
- Seit 1998: **IPv6**, 128bit Adressen.
- Da IPv6 rasant an Bedeutung gewinnt, sollte man nun das IP-Protokoll exakt bezeichnen: IPv4 / IPv6

Das Internet-Protokoll Version 4 und seine Adressen

- Protokoll für den Datenaustausch im Internet:
Internet-Protokoll
- 1980: DoD Standard Internet Protocol **IPv3**
- IPv3 verwendete **32-Bit Internet-Protokoll-Adressen**
- 1981 wird **IPv4** eingeführt, ebenfalls mit 32-Bit IP-Adressen.
- Adressraum ist ca. 4.3 Milliarden Adressen gross, seit 2011 erschöpft
- Seit 1998: **IPv6**, 128bit Adressen.
- Da IPv6 rasant an Bedeutung gewinnt, sollte man nun das IP-Protokoll exakt bezeichnen: IPv4 / IPv6

Das Internet-Protokoll Version 4 und seine Adressen

- Protokoll für den Datenaustausch im Internet:
Internet-Protokoll
- 1980: DoD Standard Internet Protocol **IPv3**
- IPv3 verwendete **32-Bit Internet-Protokoll-Adressen**
- 1981 wird **IPv4** eingeführt, ebenfalls mit 32-Bit IP-Adressen.
- Adressraum ist ca. 4.3 Milliarden Adressen gross, seit 2011 erschöpft
- Seit 1998: **IPv6**, 128bit Adressen.
- Da IPv6 rasant an Bedeutung gewinnt, sollte man nun das IP-Protokoll exakt bezeichnen: IPv4 / IPv6

Das Internet-Protokoll Version 4 und seine Adressen

- Protokoll für den Datenaustausch im Internet:
Internet-Protokoll
- 1980: DoD Standard Internet Protocol **IPv3**
- IPv3 verwendete **32-Bit Internet-Protokoll-Adressen**
- 1981 wird **IPv4** eingeführt, ebenfalls mit 32-Bit IP-Adressen.
- Adressraum ist ca. 4.3 Milliarden Adressen gross, seit 2011 erschöpft
- Seit 1998: **IPv6**, 128bit Adressen.
- Da IPv6 rasant an Bedeutung gewinnt, sollte man nun das IP-Protokoll exakt bezeichnen: IPv4 / IPv6

Das Internet-Protokoll Version 4 und seine Adressen

- Protokoll für den Datenaustausch im Internet:
Internet-Protokoll
- 1980: DoD Standard Internet Protocol **IPv3**
- IPv3 verwendete **32-Bit Internet-Protokoll-Adressen**
- 1981 wird **IPv4** eingeführt, ebenfalls mit 32-Bit IP-Adressen.
- Adressraum ist ca. 4.3 Milliarden Adressen gross, seit 2011 erschöpft
- Seit 1998: **IPv6**, 128bit Adressen.
- Da IPv6 rasant an Bedeutung gewinnt, sollte man nun das IP-Protokoll exakt bezeichnen: IPv4 / IPv6

Das Internet-Protokoll Version 4 und seine Adressen

- Protokoll für den Datenaustausch im Internet:
Internet-Protokoll
- 1980: DoD Standard Internet Protocol **IPv3**
- IPv3 verwendete **32-Bit Internet-Protokoll-Adressen**
- 1981 wird **IPv4** eingeführt, ebenfalls mit 32-Bit IP-Adressen.
- Adressraum ist ca. 4.3 Milliarden Adressen gross, seit 2011 erschöpft
- Seit 1998: **IPv6**, 128bit Adressen.
- Da IPv6 rasant an Bedeutung gewinnt, sollte man nun das IP-Protokoll exakt bezeichnen: IPv4 / IPv6

Das Internet-Protokoll Version 4 und seine Adressen

- Protokoll für den Datenaustausch im Internet:
Internet-Protokoll
- 1980: DoD Standard Internet Protocol **IPv3**
- IPv3 verwendete **32-Bit Internet-Protokoll-Adressen**
- 1981 wird **IPv4** eingeführt, ebenfalls mit 32-Bit IP-Adressen.
- Adressraum ist ca. 4.3 Milliarden Adressen gross, seit 2011 erschöpft
- Seit 1998: **IPv6**, 128bit Adressen.
- Da IPv6 rasant an Bedeutung gewinnt, sollte man nun das IP-Protokoll exakt bezeichnen: IPv4 / IPv6

Das Internet-Protokoll Version 4 und seine Adressen

- Protokoll für den Datenaustausch im Internet:
Internet-Protokoll
- 1980: DoD Standard Internet Protocol **IPv3**
- IPv3 verwendete **32-Bit Internet-Protokoll-Adressen**
- 1981 wird **IPv4** eingeführt, ebenfalls mit 32-Bit IP-Adressen.
- Adressraum ist ca. 4.3 Milliarden Adressen gross, seit 2011 erschöpft
- Seit 1998: **IPv6**, 128bit Adressen.
- Da IPv6 rasant an Bedeutung gewinnt, sollte man nun das IP-Protokoll exakt bezeichnen: IPv4 / IPv6

Das Internet-Protokoll Version 4 und seine Adressen

- Protokoll für den Datenaustausch im Internet:
Internet-Protokoll
- 1980: DoD Standard Internet Protocol **IPv3**
- IPv3 verwendete **32-Bit Internet-Protokoll-Adressen**
- 1981 wird **IPv4** eingeführt, ebenfalls mit 32-Bit IP-Adressen.
- Adressraum ist ca. 4.3 Milliarden Adressen gross, seit 2011 erschöpft
- Seit 1998: **IPv6**, 128bit Adressen.
- Da IPv6 rasant an Bedeutung gewinnt, sollte man nun das IP-Protokoll exakt bezeichnen: IPv4 / IPv6

Ein Modell des Internets

Behutsame Hinführung an die berühmt-berüchtigten OSI-Schichten

- Das Internet ist historisch über Jahrzehnte gewachsen
- Verschiedene Institutionen waren an der Entwicklung beteiligt (DOD/ARPA, IEEE, IETF)
 - ⇒ Bei der Datenübertragung zwischen zwei Rechnern im Internet sind eine Vielzahl an Programmen, Protokollen und Geräten beteiligt.
 - ⇒ Gesamtüberblick ist nur durch Einführung eines Modells möglich
- Problem für Anfänger: das Modell ist zwar sehr mächtig, aber auch sehr abstrakt und zu Beginn (mir) unverständlich
 - ⇒ Annäherung an das Modell mit einem **Beispiel** von der praktischen Seite her.

Ein Modell des Internets

Behutsame Hinführung an die berühmt-berüchtigten OSI-Schichten

- Das Internet ist historisch über Jahrzehnte gewachsen
- Verschiedene Institutionen waren an der Entwicklung beteiligt (DOD/ARPA, IEEE, IETF)
 - ⇒ Bei der Datenübertragung zwischen zwei Rechnern im Internet sind eine Vielzahl an Programmen, Protokollen und Geräten beteiligt.
 - ⇒ Gesamtüberblick ist nur durch Einführung eines Modells möglich
- Problem für Anfänger: das Modell ist zwar sehr mächtig, aber auch sehr abstrakt und zu Beginn (mir) unverständlich
 - ⇒ Annäherung an das Modell mit einem **Beispiel** von der praktischen Seite her.

Ein Modell des Internets

Behutsame Hinführung an die berühmt-berüchtigten OSI-Schichten

- Das Internet ist historisch über Jahrzehnte gewachsen
- Verschiedene Institutionen waren an der Entwicklung beteiligt (DOD/ARPA, IEEE, IETF)
 - ⇒ Bei der Datenübertragung zwischen zwei Rechnern im Internet sind eine Vielzahl an Programmen, Protokollen und Geräten beteiligt.
 - ⇒ Gesamtüberblick ist nur durch Einführung eines Modells möglich
- Problem für Anfänger: das Modell ist zwar sehr mächtig, aber auch sehr abstrakt und zu Beginn (mir) unverständlich
 - ⇒ Annäherung an das Modell mit einem **Beispiel** von der praktischen Seite her.

Ein Modell des Internets

Behutsame Hinführung an die berühmt-berüchtigten OSI-Schichten

- Das Internet ist historisch über Jahrzehnte gewachsen
- Verschiedene Institutionen waren an der Entwicklung beteiligt (DOD/ARPA, IEEE, IETF)
 - ⇒ Bei der Datenübertragung zwischen zwei Rechnern im Internet sind eine Vielzahl an Programmen, Protokollen und Geräten beteiligt.
 - ⇒ Gesamtüberblick ist nur durch Einführung eines Modells möglich
- Problem für Anfänger: das Modell ist zwar sehr mächtig, aber auch sehr abstrakt und zu Beginn (mir) unverständlich
 - ⇒ Annäherung an das Modell mit einem **Beispiel** von der praktischen Seite her.

Ein Modell des Internets

Behutsame Hinführung an die berühmt-berüchtigten OSI-Schichten

- Das Internet ist historisch über Jahrzehnte gewachsen
- Verschiedene Institutionen waren an der Entwicklung beteiligt (DOD/ARPA, IEEE, IETF)
 - ⇒ Bei der Datenübertragung zwischen zwei Rechnern im Internet sind eine Vielzahl an Programmen, Protokollen und Geräten beteiligt.
 - ⇒ Gesamtüberblick ist nur durch Einführung eines Modells möglich
- Problem für Anfänger: das Modell ist zwar sehr mächtig, aber auch sehr abstrakt und zu Beginn (mir) unverständlich
 - ⇒ Annäherung an das Modell mit einem **Beispiel** von der praktischen Seite her.

Ein Modell des Internets

Behutsame Hinführung an die berühmt-berüchtigten OSI-Schichten

- Das Internet ist historisch über Jahrzehnte gewachsen
- Verschiedene Institutionen waren an der Entwicklung beteiligt (DOD/ARPA, IEEE, IETF)
 - ⇒ Bei der Datenübertragung zwischen zwei Rechnern im Internet sind eine Vielzahl an Programmen, Protokollen und Geräten beteiligt.
 - ⇒ Gesamtüberblick ist nur durch Einführung eines Modells möglich
- Problem für Anfänger: das Modell ist zwar sehr mächtig, aber auch sehr abstrakt und zu Beginn (mir) unverständlich
 - ⇒ Annäherung an das Modell mit einem **Beispiel** von der praktischen Seite her.

Ein Modell des Internets

Behutsame Hinführung an die berühmt-berüchtigten OSI-Schichten

- Das Internet ist historisch über Jahrzehnte gewachsen
- Verschiedene Institutionen waren an der Entwicklung beteiligt (DOD/ARPA, IEEE, IETF)
 - ⇒ Bei der Datenübertragung zwischen zwei Rechnern im Internet sind eine Vielzahl an Programmen, Protokollen und Geräten beteiligt.
 - ⇒ Gesamtüberblick ist nur durch Einführung eines Modells möglich
- Problem für Anfänger: das Modell ist zwar sehr mächtig, aber auch sehr abstrakt und zu Beginn (mir) unverständlich
 - ⇒ Annäherung an das Modell mit einem **Beispiel** von der praktischen Seite her.

Ein Modell des Internets

Behutsame Hinführung an die berühmt-berüchtigten OSI-Schichten

- Das Internet ist historisch über Jahrzehnte gewachsen
- Verschiedene Institutionen waren an der Entwicklung beteiligt (DOD/ARPA, IEEE, IETF)
 - ⇒ Bei der Datenübertragung zwischen zwei Rechnern im Internet sind eine Vielzahl an Programmen, Protokollen und Geräten beteiligt.
 - ⇒ Gesamtüberblick ist nur durch Einführung eines Modells möglich
- Problem für Anfänger: das Modell ist zwar sehr mächtig, aber auch sehr abstrakt und zu Beginn (mir) unverständlich
 - ⇒ Annäherung an das Modell mit einem **Beispiel** von der praktischen Seite her.

Versand von Küchenbauteilen

Lieferung von Einbauküchen an ein Möbelhaus

- Hersteller verpackt Einzelteile einer Küche in Kartons. Die Kartons werden bezeichnet um sie wieder der richtigen Küche zuordnen zu können.
- Je ein Karton wird auf eine Euro-Palette gestellt → **einheitliches Format** im Speditionswesen. Die Paletten werden mit Adressaufklebern des Absenders und Empfängers versehen.
- Die Paletten werden mit unterschiedlichen **Verkehrsmitteln** transportiert. Es kann vorkommen, dass die Paletten einer Küche auf mehrere Fahrzeuge verteilt befördert werden.
- Das Möbelhaus erhält mehrere Paletten mit Kartons mehrerer Küchen. Durch die Kennzeichnung der Kartons werden diese richtig an den Endkunden ausgeliefert.

Versand von Küchenbauteilen

Lieferung von Einbauküchen an ein Möbelhaus

- Hersteller verpackt Einzelteile einer Küche in Kartons. Die Kartons werden bezeichnet um sie wieder der richtigen Küche zuordnen zu können.
- Je ein Karton wird auf eine Euro-Palette gestellt → **einheitliches Format** im Speditionswesen. Die Paletten werden mit Adressaufklebern des Absenders und Empfängers versehen.
- Die Paletten werden mit unterschiedlichen **Verkehrsmitteln** transportiert. Es kann vorkommen, dass die Paletten einer Küche auf mehrere Fahrzeuge verteilt befördert werden.
- Das Möbelhaus erhält mehrere Paletten mit Kartons mehrerer Küchen. Durch die Kennzeichnung der Kartons werden diese richtig an den Endkunden ausgeliefert.

Versand von Küchenbauteilen

Lieferung von Einbauküchen an ein Möbelhaus

- Hersteller verpackt Einzelteile einer Küche in Kartons. Die Kartons werden bezeichnet um sie wieder der richtigen Küche zuordnen zu können.
- Je ein Karton wird auf eine Euro-Palette gestellt → **einheitliches Format** im Speditionswesen. Die Paletten werden mit Adressaufklebern des Absenders und Empfängers versehen.
- Die Paletten werden mit unterschiedlichen **Verkehrsmitteln** transportiert. Es kann vorkommen, dass die Paletten einer Küche auf mehrere Fahrzeuge verteilt befördert werden.
- Das Möbelhaus erhält mehrere Paletten mit Kartons mehrerer Küchen. Durch die Kennzeichnung der Kartons werden diese richtig an den Endkunden ausgeliefert.

Versand von Küchenbauteilen

Lieferung von Einbauküchen an ein Möbelhaus

- Hersteller verpackt Einzelteile einer Küche in Kartons. Die Kartons werden bezeichnet um sie wieder der richtigen Küche zuordnen zu können.
- Je ein Karton wird auf eine Euro-Palette gestellt → **einheitliches Format** im Speditionswesen. Die Paletten werden mit Adressaufklebern des Absenders und Empfängers versehen.
- Die Paletten werden mit unterschiedlichen **Verkehrsmitteln** transportiert. Es kann vorkommen, dass die Paletten einer Küche auf mehrere Fahrzeuge verteilt befördert werden.
- Das Möbelhaus erhält mehrere Paletten mit Kartons mehrerer Küchen. Durch die Kennzeichnung der Kartons werden diese richtig an den Endkunden ausgeliefert.

Versand von Küchenbauteilen

Lieferung von Einbauküchen an ein Möbelhaus

- Hersteller verpackt Einzelteile einer Küche in Kartons. Die Kartons werden bezeichnet um sie wieder der richtigen Küche zuordnen zu können.
- Je ein Karton wird auf eine Euro-Palette gestellt → **einheitliches Format** im Speditionswesen. Die Paletten werden mit Adressaufklebern des Absenders und Empfängers versehen.
- Die Paletten werden mit unterschiedlichen **Verkehrsmitteln** transportiert. Es kann vorkommen, dass die Paletten einer Küche auf mehrere Fahrzeuge verteilt befördert werden.
- Das Möbelhaus erhält mehrere Paletten mit Kartons mehrerer Küchen. Durch die Kennzeichnung der Kartons werden diese richtig an den Endkunden ausgeliefert.

Versand von Küchenbauteilen

Lieferung von Einbauküchen an ein Möbelhaus

- Hersteller verpackt Einzelteile einer Küche in Kartons. Die Kartons werden bezeichnet um sie wieder der richtigen Küche zuordnen zu können.
- Je ein Karton wird auf eine Euro-Palette gestellt → **einheitliches Format** im Speditionswesen. Die Paletten werden mit Adressaufklebern des Absenders und Empfängers versehen.
- Die Paletten werden mit unterschiedlichen **Verkehrsmitteln** transportiert. Es kann vorkommen, dass die Paletten einer Küche auf mehrere Fahrzeuge verteilt befördert werden.
- Das Möbelhaus erhält mehrere Paletten mit Kartons mehrerer Küchen. Durch die Kennzeichnung der Kartons werden diese richtig an den Endkunden ausgeliefert.

Versand von Küchenbauteilen

Lieferung von Einbauküchen an ein Möbelhaus

- Hersteller verpackt Einzelteile einer Küche in Kartons. Die Kartons werden bezeichnet um sie wieder der richtigen Küche zuordnen zu können.
- Je ein Karton wird auf eine Euro-Palette gestellt → **einheitliches Format** im Speditionswesen. Die Paletten werden mit Adressaufklebern des Absenders und Empfängers versehen.
- Die Paletten werden mit unterschiedlichen **Verkehrsmitteln** transportiert. Es kann vorkommen, dass die Paletten einer Küche auf mehrere Fahrzeuge verteilt befördert werden.
- Das Möbelhaus erhält mehrere Paletten mit Kartons mehrerer Küchen. Durch die Kennzeichnung der Kartons werden diese richtig an den Endkunden ausgeliefert.

Versand von Küchenbauteilen

Lieferung von Einbauküchen an ein Möbelhaus

- Hersteller verpackt Einzelteile einer Küche in Kartons. Die Kartons werden bezeichnet um sie wieder der richtigen Küche zuordnen zu können.
- Je ein Karton wird auf eine Euro-Palette gestellt → **einheitliches Format** im Speditionswesen. Die Paletten werden mit Adressaufklebern des Absenders und Empfängers versehen.
- Die Paletten werden mit unterschiedlichen **Verkehrsmitteln** transportiert. Es kann vorkommen, dass die Paletten einer Küche auf mehrere Fahrzeuge verteilt befördert werden.
- Das Möbelhaus erhält mehrere Paletten mit Kartons mehrerer Küchen. Durch die Kennzeichnung der Kartons werden diese richtig an den Endkunden ausgeliefert.

Versand von Küchenbauteilen

Lieferung von Einbauküchen an ein Möbelhaus

- Hersteller verpackt Einzelteile einer Küche in Kartons. Die Kartons werden bezeichnet um sie wieder der richtigen Küche zuordnen zu können.
- Je ein Karton wird auf eine Euro-Palette gestellt → **einheitliches Format** im Speditionswesen. Die Paletten werden mit Adressaufklebern des Absenders und Empfängers versehen.
- Die Paletten werden mit unterschiedlichen **Verkehrsmitteln** transportiert. Es kann vorkommen, dass die Paletten einer Küche auf mehrere Fahrzeuge verteilt befördert werden.
- Das Möbelhaus erhält mehrere Paletten mit Kartons mehrerer Küchen. Durch die Kennzeichnung der Kartons werden diese richtig an den Endkunden ausgeliefert.

Versand von Küchenbauteilen

Lieferung von Einbauküchen an ein Möbelhaus

- Hersteller verpackt Einzelteile einer Küche in Kartons. Die Kartons werden bezeichnet um sie wieder der richtigen Küche zuordnen zu können.
- Je ein Karton wird auf eine Euro-Palette gestellt → **einheitliches Format** im Speditionswesen. Die Paletten werden mit Adressaufklebern des Absenders und Empfängers versehen.
- Die Paletten werden mit unterschiedlichen **Verkehrsmitteln** transportiert. Es kann vorkommen, dass die Paletten einer Küche auf mehrere Fahrzeuge verteilt befördert werden.
- Das Möbelhaus erhält mehrere Paletten mit Kartons mehrerer Küchen. Durch die Kennzeichnung der Kartons werden diese richtig an den Endkunden ausgeliefert.

Versand von Küchenbauteilen

Lieferung von Einbauküchen an ein Möbelhaus

- Hersteller verpackt Einzelteile einer Küche in Kartons. Die Kartons werden bezeichnet um sie wieder der richtigen Küche zuordnen zu können.
- Je ein Karton wird auf eine Euro-Palette gestellt → **einheitliches Format** im Speditionswesen. Die Paletten werden mit Adressaufklebern des Absenders und Empfängers versehen.
- Die Paletten werden mit unterschiedlichen **Verkehrsmitteln** transportiert. Es kann vorkommen, dass die Paletten einer Küche auf mehrere Fahrzeuge verteilt befördert werden.
- Das Möbelhaus erhält mehrere Paletten mit Kartons mehrerer Küchen. Durch die Kennzeichnung der Kartons werden diese richtig an den Endkunden ausgeliefert.

Paketversand einer Datei

Das Palettenbeispiel wird auf das Internet angewendet:
Übertragung einer Datei im Internet

- Datei wird in maximal 1500 Byte grosse Teile zerlegt
- Jedes Teilstück wird mit einem Header versehen, um im Ziel die Datei wieder zusammensetzen zu können.
→ **Segment**
- Der Header enthält u.a. die Nummer des Endkunden:
→ **Portadresse**
- Segmente palettieren: jedes Segment erhält nochmals einen Header mit der Internet-Zieladresse und weiteren Informationen für die Transportbürokratie. → **IP-Paket**
- Verladen der Paletten: Die IP-Pakete werden mit einem Header und einer Prüfsumme versehen.
Transportmittel → **Frame**
Transportweg → **Bitübertragung**

Paketversand einer Datei

Das Palettenbeispiel wird auf das Internet angewendet: Übertragung einer Datei im Internet

- Datei wird in maximal 1500 Byte grosse Teile zerlegt
- Jedes Teilstück wird mit einem Header versehen, um im Ziel die Datei wieder zusammensetzen zu können.
→ **Segment**
- Der Header enthält u.a. die Nummer des Endkunden:
→ **Portadresse**
- Segmente palettieren: jedes Segment erhält nochmals einen Header mit der Internet-Zieladresse und weiteren Informationen für die Transportbürokratie. → **IP-Paket**
- Verladen der Paletten: Die IP-Pakete werden mit einem Header und einer Prüfsumme versehen.
Transportmittel → **Frame**
Transportweg → **Bitübertragung**

Paketversand einer Datei

Das Palettenbeispiel wird auf das Internet angewendet:
Übertragung einer Datei im Internet

- Datei wird in maximal 1500 Byte grosse Teile zerlegt
- Jedes Teilstück wird mit einem Header versehen, um im Ziel die Datei wieder zusammensetzen zu können.
→ **Segment**
- Der Header enthält u.a. die Nummer des Endkunden:
→ **Portadresse**
- Segmente palettieren: jedes Segment erhält nochmals einen Header mit der Internet-Zieladresse und weiteren Informationen für die Transportbürokratie. → **IP-Paket**
- Verladen der Paletten: Die IP-Pakete werden mit einem Header und einer Prüfsumme versehen.
Transportmittel → **Frame**
Transportweg → **Bitübertragung**

Paketversand einer Datei

Das Palettenbeispiel wird auf das Internet angewendet:
Übertragung einer Datei im Internet

- Datei wird in maximal 1500 Byte grosse Teile zerlegt
- Jedes Teilstück wird mit einem Header versehen, um im Ziel die Datei wieder zusammensetzen zu können.
→ **Segment**
- Der Header enthält u.a. die Nummer des Endkunden:
→ **Portadresse**
- Segmente palettieren: jedes Segment erhält nochmals einen Header mit der Internet-Zieladresse und weiteren Informationen für die Transportbürokratie. → **IP-Paket**
- Verladen der Paletten: Die IP-Pakete werden mit einem Header und einer Prüfsumme versehen.
Transportmittel → **Frame**
Transportweg → **Bitübertragung**

Paketversand einer Datei

Das Palettenbeispiel wird auf das Internet angewendet:
Übertragung einer Datei im Internet

- Datei wird in maximal 1500 Byte grosse Teile zerlegt
- Jedes Teilstück wird mit einem Header versehen, um im Ziel die Datei wieder zusammensetzen zu können.
→ **Segment**
- Der Header enthält u.a. die Nummer des Endkunden:
→ **Portadresse**
- Segmente palettieren: jedes Segment erhält nochmals einen Header mit der Internet-Zieladresse und weiteren Informationen für die Transportbürokratie. → **IP-Paket**
- Verladen der Paletten: Die IP-Pakete werden mit einem Header und einer Prüfsumme versehen.
Transportmittel → **Frame**
Transportweg → **Bitübertragung**

Paketversand einer Datei

Das Palettenbeispiel wird auf das Internet angewendet:
Übertragung einer Datei im Internet

- Datei wird in maximal 1500 Byte grosse Teile zerlegt
- Jedes Teilstück wird mit einem Header versehen, um im Ziel die Datei wieder zusammensetzen zu können.
→ **Segment**
- Der Header enthält u.a. die Nummer des Endkunden:
→ **Portadresse**
- Segmente palettieren: jedes Segment erhält nochmals einen Header mit der Internet-Zieladresse und weiteren Informationen für die Transportbürokratie. → **IP-Paket**
- Verladen der Paletten: Die IP-Pakete werden mit einem Header und einer Prüfsumme versehen.
Transportmittel → **Frame**
Transportweg → **Bitübertragung**

Paketversand einer Datei

Das Palettenbeispiel wird auf das Internet angewendet:
Übertragung einer Datei im Internet

- Datei wird in maximal 1500 Byte grosse Teile zerlegt
- Jedes Teilstück wird mit einem Header versehen, um im Ziel die Datei wieder zusammensetzen zu können.
→ **Segment**
- Der Header enthält u.a. die Nummer des Endkunden:
→ **Portadresse**
- Segmente palettieren: jedes Segment erhält nochmals einen Header mit der Internet-Zieladresse und weiteren Informationen für die Transportbürokratie. → **IP-Paket**
- Verladen der Paletten: Die IP-Pakete werden mit einem Header und einer Prüfsumme versehen.
Transportmittel → **Frame**
Transportweg → **Bitübertragung**

Paketversand einer Datei

Das Palettenbeispiel wird auf das Internet angewendet:
Übertragung einer Datei im Internet

- Datei wird in maximal 1500 Byte grosse Teile zerlegt
- Jedes Teilstück wird mit einem Header versehen, um im Ziel die Datei wieder zusammensetzen zu können.
→ **Segment**
- Der Header enthält u.a. die Nummer des Endkunden:
→ **Portadresse**
- Segmente palettieren: jedes Segment erhält nochmals einen Header mit der Internet-Zieladresse und weiteren Informationen für die Transportbürokratie. → **IP-Paket**
- Verladen der Paletten: Die IP-Pakete werden mit einem Header und einer Prüfsumme versehen.
Transportmittel → **Frame**
Transportweg → **Bitübertragung**

Paketversand einer Datei

Das Palettenbeispiel wird auf das Internet angewendet:
Übertragung einer Datei im Internet

- Datei wird in maximal 1500 Byte grosse Teile zerlegt
- Jedes Teilstück wird mit einem Header versehen, um im Ziel die Datei wieder zusammensetzen zu können.
→ **Segment**
- Der Header enthält u.a. die Nummer des Endkunden:
→ **Portadresse**
- Segmente palettieren: jedes Segment erhält nochmals einen Header mit der Internet-Zieladresse und weiteren Informationen für die Transportbürokratie. → **IP-Paket**
- Verladen der Paletten: Die IP-Pakete werden mit einem Header und einer Prüfsumme versehen.
Transportmittel → **Frame**
Transportweg → **Bitübertragung**

Paketversand einer Datei

Das Palettenbeispiel wird auf das Internet angewendet:
Übertragung einer Datei im Internet

- Datei wird in maximal 1500 Byte grosse Teile zerlegt
- Jedes Teilstück wird mit einem Header versehen, um im Ziel die Datei wieder zusammensetzen zu können.
→ **Segment**
- Der Header enthält u.a. die Nummer des Endkunden:
→ **Portadresse**
- Segmente palettieren: jedes Segment erhält nochmals einen Header mit der Internet-Zieladresse und weiteren Informationen für die Transportbürokratie. → **IP-Paket**
- Verladen der Paletten: Die IP-Pakete werden mit einem Header und einer Prüfsumme versehen.
Transportmittel → **Frame**
Transportweg → **Bitübertragung**

Paketversand einer Datei

Das Palettenbeispiel wird auf das Internet angewendet:
Übertragung einer Datei im Internet

- Datei wird in maximal 1500 Byte grosse Teile zerlegt
- Jedes Teilstück wird mit einem Header versehen, um im Ziel die Datei wieder zusammensetzen zu können.
→ **Segment**
- Der Header enthält u.a. die Nummer des Endkunden:
→ **Portadresse**
- Segmente palettieren: jedes Segment erhält nochmals einen Header mit der Internet-Zieladresse und weiteren Informationen für die Transportbürokratie. → **IP-Paket**
- Verladen der Paletten: Die IP-Pakete werden mit einem Header und einer Prüfsumme versehen.
Transportmittel → **Frame**
Transportweg → **Bitübertragung**

Paketversand einer Datei

Das Palettenbeispiel wird auf das Internet angewendet:
Übertragung einer Datei im Internet

- Datei wird in maximal 1500 Byte grosse Teile zerlegt
- Jedes Teilstück wird mit einem Header versehen, um im Ziel die Datei wieder zusammensetzen zu können.
→ **Segment**
- Der Header enthält u.a. die Nummer des Endkunden:
→ **Portadresse**
- Segmente palettieren: jedes Segment erhält nochmals einen Header mit der Internet-Zieladresse und weiteren Informationen für die Transportbürokratie. → **IP-Paket**
- Verladen der Paletten: Die IP-Pakete werden mit einem Header und einer Prüfsumme versehen.

Transportmittel → **Frame**

Transportweg → **Bitübertragung**

Paketversand einer Datei

Das Palettenbeispiel wird auf das Internet angewendet:
Übertragung einer Datei im Internet

- Datei wird in maximal 1500 Byte grosse Teile zerlegt
- Jedes Teilstück wird mit einem Header versehen, um im Ziel die Datei wieder zusammensetzen zu können.
→ **Segment**
- Der Header enthält u.a. die Nummer des Endkunden:
→ **Portadresse**
- Segmente palettieren: jedes Segment erhält nochmals einen Header mit der Internet-Zieladresse und weiteren Informationen für die Transportbürokratie. → **IP-Paket**
- Verladen der Paletten: Die IP-Pakete werden mit einem Header und einer Prüfsumme versehen.
Transportmittel → **Frame**
Transportweg → **Bitübertragung**

Paketversand einer Datei

Das Palettenbeispiel wird auf das Internet angewendet:
Übertragung einer Datei im Internet

- Datei wird in maximal 1500 Byte grosse Teile zerlegt
- Jedes Teilstück wird mit einem Header versehen, um im Ziel die Datei wieder zusammensetzen zu können.
→ **Segment**
- Der Header enthält u.a. die Nummer des Endkunden:
→ **Portadresse**
- Segmente palettieren: jedes Segment erhält nochmals einen Header mit der Internet-Zieladresse und weiteren Informationen für die Transportbürokratie. → **IP-Paket**
- Verladen der Paletten: Die IP-Pakete werden mit einem Header und einer Prüfsumme versehen.
Transportmittel → **Frame**
Transportweg → **Bitübertragung**

Erster Kontakt mit den Schichten

Endkunde	http, smtp, dns, ...	Schicht 7
nummeriertes Bauteil	Segment mit Portadresse	Schicht 4
Palette mit Adressen	IP-Paket mit Header (1984)	Schicht 3
Umladestation	Router	Schicht 3
LKW, Bahnwaggon, Schiff	Ethernet-, TokenRing-, FDDI-Frame	Schicht 2
StVO, StVZO, ...	Bitübertragungsschicht	Schicht 1

Erster Kontakt mit den Schichten

Endkunde	http, smtp, dns, ...	Schicht 7
nummeriertes Bauteil	Segment mit Portadresse	Schicht 4
Palette mit Adressen	IP-Paket mit Header (1984)	Schicht 3
Umladestation	Router	Schicht 3
LKW, Bahnwaggon, Schiff	Ethernet-, TokenRing-, FDDI-Frame	Schicht 2
StVO, StVZO, ...	Bitübertragungsschicht	Schicht 1

Erster Kontakt mit den Schichten

Endkunde	http, smtp, dns, ...	Schicht 7
nummeriertes Bauteil	Segment mit Portadresse	Schicht 4
Palette mit Adressen	IP-Paket mit Header (1984)	Schicht 3
Umladestation	Router	Schicht 3
LKW, Bahnwaggon, Schiff	Ethernet-, TokenRing-, FDDI-Frame	Schicht 2
StVO, StVZO, ...	Bitübertragungsschicht	Schicht 1

Erster Kontakt mit den Schichten

Endkunde	http, smtp, dns, ...	Schicht 7
nummeriertes Bauteil	Segment mit Portadresse	Schicht 4
Palette mit Adressen	IP-Paket mit Header (1984)	Schicht 3
Umladestation	Router	Schicht 3
LKW, Bahnwaggon, Schiff	Ethernet-, TokenRing-, FDDI-Frame	Schicht 2
StVO, StVZO, ...	Bitübertragungsschicht	Schicht 1

Erster Kontakt mit den Schichten

Endkunde	http, smtp, dns, ...	Schicht 7
nummeriertes Bauteil	Segment mit Portadresse	Schicht 4
Palette mit Adressen	IP-Paket mit Header (1984)	Schicht 3
Umladestation	Router	Schicht 3
LKW, Bahnwaggon, Schiff	Ethernet-, TokenRing-, FDDI-Frame	Schicht 2
StVO, StVZO, ...	Bitübertragungsschicht	Schicht 1

Erster Kontakt mit den Schichten

Endkunde	http, smtp, dns, ...	Schicht 7
nummeriertes Bauteil	Segment mit Portadresse	Schicht 4
Palette mit Adressen	IP-Paket mit Header (1984)	Schicht 3
Umladestation	Router	Schicht 3
LKW, Bahnwaggon, Schiff	Ethernet-, TokenRing-, FDDI-Frame	Schicht 2
StVO, StVZO, ...	Bitübertragungsschicht	Schicht 1

Erster Kontakt mit den Schichten

Endkunde	http, smtp, dns, ...	Schicht 7
nummeriertes Bauteil	Segment mit Portadresse	Schicht 4
Palette mit Adressen	IP-Paket mit Header (1984)	Schicht 3
Umladestation	Router	Schicht 3
LKW, Bahnwaggon, Schiff	Ethernet-, TokenRing-, FDDI-Frame	Schicht 2
StVO, StVZO, ...	Bitübertragungsschicht	Schicht 1

Erster Kontakt mit den Schichten

Endkunde	http, smtp, dns, ...	Schicht 7
nummeriertes Bauteil	Segment mit Portadresse	Schicht 4
Palette mit Adressen	IP-Paket mit Header (1984)	Schicht 3
Umladestation	Router	Schicht 3
LKW, Bahnwaggon, Schiff	Ethernet-, TokenRing-, FDDI-Frame	Schicht 2
StVO, StVZO, ...	Bitübertragungsschicht	Schicht 1

Erster Kontakt mit den Schichten

Endkunde	http, smtp, dns, ...	Schicht 7
nummeriertes Bauteil	Segment mit Portadresse	Schicht 4
Palette mit Adressen	IP-Paket mit Header (1984)	Schicht 3
Umladestation	Router	Schicht 3
LKW, Bahnwaggon, Schiff	Ethernet-, TokenRing-, FDDI-Frame	Schicht 2
StVO, StVZO, ...	Bitübertragungsschicht	Schicht 1

Erster Kontakt mit den Schichten

Endkunde	http, smtp, dns, ...	Schicht 7
nummeriertes Bauteil	Segment mit Portadresse	Schicht 4
Palette mit Adressen	IP-Paket mit Header (1984)	Schicht 3
Umladestation	Router	Schicht 3
LKW, Bahnwaggon, Schiff	Ethernet-, TokenRing-, FDDI- Frame	Schicht 2
StVO, StVZO, ...	Bitübertragungsschicht	Schicht 1

Erster Kontakt mit den Schichten

Endkunde	http, smtp, dns, ...	Schicht 7
nummeriertes Bauteil	Segment mit Portadresse	Schicht 4
Palette mit Adressen	IP-Paket mit Header (1984)	Schicht 3
Umladestation	Router	Schicht 3
LKW, Bahnwaggon, Schiff	Ethernet-, TokenRing-, FDDI- Frame	Schicht 2
StVO, StVZO, ...	Bitübertragungsschicht	Schicht 1

Erster Kontakt mit den Schichten

Endkunde	http, smtp, dns, ...	Schicht 7
nummeriertes Bauteil	Segment mit Portadresse	Schicht 4
Palette mit Adressen	IP-Paket mit Header (1984)	Schicht 3
Umladestation	Router	Schicht 3
LKW, Bahnwaggon, Schiff	Ethernet-, TokenRing-, FDDI- Frame	Schicht 2
StVO, StVZO, ...	Bitübertragungsschicht	Schicht 1

Erster Kontakt mit den Schichten

Endkunde	http, smtp, dns, ...	Schicht 7
nummeriertes Bauteil	Segment mit Portadresse	Schicht 4
Palette mit Adressen	IP-Paket mit Header (1984)	Schicht 3
Umladestation	Router	Schicht 3
LKW, Bahnwaggon, Schiff	Ethernet-, TokenRing-, FDDI- Frame	Schicht 2
StVO, StVZO, ...	Bitübertragungsschicht	Schicht 1

Erster Kontakt mit den Schichten

Endkunde	http, smtp, dns, ...	Schicht 7
nummeriertes Bauteil	Segment mit Portadresse	Schicht 4
Palette mit Adressen	IP-Paket mit Header (1984)	Schicht 3
Umladestation	Router	Schicht 3
LKW, Bahnwaggon, Schiff	Ethernet-, TokenRing-, FDDI- Frame	Schicht 2
StVO, StVZO, ...	Bitübertragungsschicht	Schicht 1

Erster Kontakt mit den Schichten

Endkunde	http, smtp, dns, ...	Schicht 7
nummeriertes Bauteil	Segment mit Portadresse	Schicht 4
Palette mit Adressen	IP-Paket mit Header (1984)	Schicht 3
Umladestation	Router	Schicht 3
LKW, Bahnwaggon, Schiff	Ethernet-, TokenRing-, FDDI- Frame	Schicht 2
StVO, StVZO, ...	Bitübertragungsschicht	Schicht 1

Erster Kontakt mit den Schichten

Endkunde	http, smtp, dns, ...	Schicht 7
nummeriertes Bauteil	Segment mit Portadresse	Schicht 4
Palette mit Adressen	IP-Paket mit Header (1984)	Schicht 3
Umladestation	Router	Schicht 3
LKW, Bahnwaggon, Schiff	Ethernet-, TokenRing-, FDDI-Frame	Schicht 2
StVO, StVZO, ...	Bitübertragungsschicht	Schicht 1

Erster Kontakt mit den Schichten

Endkunde	http, smtp, dns, ...	Schicht 7
nummeriertes Bauteil	Segment mit Portadresse	Schicht 4
Palette mit Adressen	IP-Paket mit Header (1984)	Schicht 3
Umladestation	Router	Schicht 3
LKW, Bahnwaggon, Schiff	Ethernet-, TokenRing-, FDDI-Frame	Schicht 2
StVO, StVZO, ...	Bitübertragungsschicht	Schicht 1

Erster Kontakt mit den Schichten

Endkunde	http, smtp, dns, ...	Schicht 7
nummeriertes Bauteil	Segment mit Portadresse	Schicht 4
Palette mit Adressen	IP-Paket mit Header (1984)	Schicht 3
Umladestation	Router	Schicht 3
LKW, Bahnwaggon, Schiff	Ethernet-, TokenRing-, FDDI-Frame	Schicht 2
StVO, StVZO, ...	Bitübertragungsschicht	Schicht 1

Erster Kontakt mit den Schichten

Endkunde	http, smtp, dns, ...	Schicht 7
nummeriertes Bauteil	Segment mit Portadresse	Schicht 4
Palette mit Adressen	IP-Paket mit Header (1984)	Schicht 3
Umladestation	Router	Schicht 3
LKW, Bahnwaggon, Schiff	Ethernet-, TokenRing-, FDDI- Frame	Schicht 2
StVO, StVZO, ...	Bitübertragungsschicht	Schicht 1

Erster Kontakt mit den Schichten

Endkunde	http, smtp, dns, ...	Schicht 7
nummeriertes Bauteil	Segment mit Portadresse	Schicht 4
Palette mit Adressen	IP-Paket mit Header (1984)	Schicht 3
Umladestation	Router	Schicht 3
LKW, Bahnwaggon, Schiff	Ethernet-, TokenRing-, FDDI-Frame	Schicht 2
StVO, StVZO, ...	Bitübertragungsschicht	Schicht 1

Eine wichtige Regel:

Wichtige Regel, für später merken:

Wie die Paletten auch, werden die IP-Pakete beim **Umladen** nicht geändert

Eine wichtige Regel:

Wichtige Regel, für später merken:

Wie die Paletten auch, werden die IP-Pakete beim **Umladen** nicht geändert

Die Paletten sind überall

- Original-Idee von IP: ein hardwareunabhängiges, paketorientiertes Übertragungsprotokoll zwischen den bestehenden LANs.
- paketorientiert: eine komplette Küche wird nicht in einem riesigen Lastzug, sondern verteilt auf kleinere Fahrzeuge transportiert
- Da überall palettenkompatible Geräte entwickelt werden, werden die Paletten nun nach und nach auch **auf den Firmengeländen (LAN)** und sogar **innerhalb der Firmengebäude (Host=PC)** der Küchenhersteller verwendet.

Die Paletten sind überall

- Original-Idee von IP: ein hardwareunabhängiges, paketorientiertes Übertragungsprotokoll zwischen den bestehenden LANs.
- paketorientiert: eine komplette Küche wird nicht in einem riesigen Lastzug, sondern verteilt auf kleinere Fahrzeuge transportiert
- Da überall palettenkompatible Geräte entwickelt werden, werden die Paletten nun nach und nach auch **auf den Firmengeländen (LAN)** und sogar **innerhalb der Firmengebäude (Host=PC)** der Küchenhersteller verwendet.

Die Paletten sind überall

- Original-Idee von IP: ein hardwareunabhängiges, paketorientiertes Übertragungsprotokoll zwischen den bestehenden LANs.
- paketorientiert: eine komplette Küche wird nicht in einem riesigen Lastzug, sondern verteilt auf kleinere Fahrzeuge transportiert
- Da überall palettenkompatible Geräte entwickelt werden, werden die Paletten nun nach und nach auch **auf den Firmengeländen (LAN)** und sogar **innerhalb der Firmengebäude (Host=PC)** der Küchenhersteller verwendet.

Die Paletten sind überall

- Original-Idee von IP: ein hardwareunabhängiges, paketorientiertes Übertragungsprotokoll zwischen den bestehenden LANs.
- paketorientiert: eine komplette Küche wird nicht in einem riesigen Lastzug, sondern verteilt auf kleinere Fahrzeuge transportiert
- Da überall palettenkompatible Geräte entwickelt werden, werden die Paletten nun nach und nach auch **auf den Firmengeländen (LAN)** und sogar **innerhalb der Firmengebäude (Host=PC)** der Küchenhersteller verwendet.

Die Paletten sind überall

- Original-Idee von IP: ein hardwareunabhängiges, paketorientiertes Übertragungsprotokoll zwischen den bestehenden LANs.
- paketorientiert: eine komplette Küche wird nicht in einem riesigen Lastzug, sondern verteilt auf kleinere Fahrzeuge transportiert
- Da überall palettenkompatible Geräte entwickelt werden, werden die Paletten nun nach und nach auch **auf den Firmengeländen (LAN)** und sogar **innerhalb der Firmengebäude (Host=PC)** der Küchenhersteller verwendet.

Die Paletten sind überall

- Original-Idee von IP: ein hardwareunabhängiges, paketorientiertes Übertragungsprotokoll zwischen den bestehenden LANs.
- paketorientiert: eine komplette Küche wird nicht in einem riesigen Lastzug, sondern verteilt auf kleinere Fahrzeuge transportiert
- Da überall palettenkompatible Geräte entwickelt werden, werden die Paletten nun nach und nach auch **auf den Firmengeländen (LAN)** und sogar **innerhalb der Firmengebäude (Host=PC)** der Küchenhersteller verwendet.

Die Paletten sind überall

IP ist das Standardprotokoll für Netzwerkanwendungen im Internet, im lokalen Netzwerk und sogar innerhalb eines Rechners (→ localhost / 127.0.0.1 / ::1)

Beispiel für ein Netzwerk, bei dem das noch anders war:
NetBIOS / NetBEUI : die verpackten Küchenteile werden einfach so in den LKW geladen **ohne** Palette. Nachteil: nicht routbar

Die Paletten sind überall

IP ist das Standardprotokoll für Netzwerkanwendungen im Internet, im lokalen Netzwerk und sogar innerhalb eines Rechners (→ localhost / 127.0.0.1 / ::1)

Beispiel für ein Netzwerk, bei dem das noch anders war:
NetBIOS / NetBEUI : die verpackten Küchenteile werden einfach so in den LKW geladen **ohne** Palette. Nachteil: nicht routbar

Die Paletten sind überall

IP ist das Standardprotokoll für Netzwerkanwendungen im Internet, im lokalen Netzwerk und sogar innerhalb eines Rechners (→ localhost / 127.0.0.1 / ::1)

Beispiel für ein Netzwerk, bei dem das noch anders war:
NetBIOS / NetBEUI : die verpackten Küchenteile werden einfach so in den LKW geladen **ohne** Palette. Nachteil: nicht routbar

Inhalt

Vom LAN zum Internet

Rahmen, Pakete, virtuelle Verbindungen

Versuch: Daten mit nc übertragen und Datenverkehr mitschneiden

IP-Adressen und Subnetze

Address Resolution Protocol

Routing

Die beiden Schichtenmodelle

Unser Palettenmodell stösst bei der Erforschung weiterer Details an Grenzen → es muss ein besseres Modell her:

1. Das Modell des DoD: einfach, anschaulich, **bildet das real existierende Internet ab**. Nachteil: im hardwarenahen Bereich zu grob
2. Das Modell der ISO: Open System Interconnection: **OSI**: detailliert, keine Implementierung, für Netzwerker **unumgänglich**

Die beiden Schichtenmodelle

Unser Palettenmodell stösst bei der Erforschung weiterer Details an Grenzen → es muss ein besseres Modell her:

1. Das Modell des DoD: einfach, anschaulich, **bildet das real existierende Internet ab**. Nachteil: im hardwarenahen Bereich zu grob
2. Das Modell der ISO: Open System Interconnection: **OSI**: detailliert, keine Implementierung, für Netzwerker **unumgänglich**

Die beiden Schichtenmodelle

Unser Palettenmodell stösst bei der Erforschung weiterer Details an Grenzen → es muss ein besseres Modell her:

1. Das Modell des DoD: einfach, anschaulich, **bildet das real existierende Internet ab**. Nachteil: im hardwarenahen Bereich zu grob
2. Das Modell der ISO: Open System Interconnection: **OSI**: detailliert, keine Implementierung, für Netzwerker **unumgänglich**

Die beiden Schichtenmodelle

Unser Palettenmodell stösst bei der Erforschung weiterer Details an Grenzen → es muss ein besseres Modell her:

1. Das Modell des DoD: einfach, anschaulich, **bildet das real existierende Internet ab**. Nachteil: im hardwarenahen Bereich zu grob
2. Das Modell der ISO: Open System Interconnection: **OSI**: detailliert, keine Implementierung, für Netzwerker **unumgänglich**

Die beiden Schichtenmodelle

Unser Palettenmodell stösst bei der Erforschung weiterer Details an Grenzen → es muss ein besseres Modell her:

1. Das Modell des DoD: einfach, anschaulich, **bildet das real existierende Internet ab**. Nachteil: im hardwarenahen Bereich zu grob
2. Das Modell der ISO: Open System Interconnection: **OSI**: detailliert, keine Implementierung, für Netzwerker **unumgänglich**

Die beiden Schichtenmodelle

Unser Palettenmodell stösst bei der Erforschung weiterer Details an Grenzen → es muss ein besseres Modell her:

1. Das Modell des DoD: einfach, anschaulich, **bildet das real existierende Internet ab**. Nachteil: im hardwarenahen Bereich zu grob
2. Das Modell der ISO: Open System Interconnection: **OSI**: detailliert, keine Implementierung, für Netzwerker **unumgänglich**

Die beiden Schichtenmodelle

Unser Palettenmodell stösst bei der Erforschung weiterer Details an Grenzen → es muss ein besseres Modell her:

1. Das Modell des DoD: einfach, anschaulich, **bildet das real existierende Internet ab**. Nachteil: im hardwarenahen Bereich zu grob
2. Das Modell der ISO: Open System Interconnection: **OSI**: detailliert, keine Implementierung, für Netzwerker **unumgänglich**

Die beiden Schichtenmodelle

Unser Palettenmodell stösst bei der Erforschung weiterer Details an Grenzen → es muss ein besseres Modell her:

1. Das Modell des DoD: einfach, anschaulich, **bildet das real existierende Internet ab**. Nachteil: im hardwarenahen Bereich zu grob
2. Das Modell der ISO: Open System Interconnection: **OSI**: detailliert, keine Implementierung, für Netzwerker **unumgänglich**

Die beiden Schichtenmodelle

Unser Palettenmodell stösst bei der Erforschung weiterer Details an Grenzen → es muss ein besseres Modell her:

1. Das Modell des DoD: einfach, anschaulich, **bildet das real existierende Internet ab**. Nachteil: im hardwarenahen Bereich zu grob
2. Das Modell der ISO: Open System Interconnection: **OSI**: detailliert, keine Implementierung, für Netzwerker **unumgänglich**

Die beiden Schichtenmodelle

Unser Palettenmodell stösst bei der Erforschung weiterer Details an Grenzen → es muss ein besseres Modell her:

1. Das Modell des DoD: einfach, anschaulich, **bildet das real existierende Internet ab**. Nachteil: im hardwarenahen Bereich zu grob
2. Das Modell der ISO: Open System Interconnection: **OSI**: detailliert, keine Implementierung, für Netzwerker unumgänglich

Die beiden Schichtenmodelle

Unser Palettenmodell stösst bei der Erforschung weiterer Details an Grenzen → es muss ein besseres Modell her:

1. Das Modell des DoD: einfach, anschaulich, **bildet das real existierende Internet ab**. Nachteil: im hardwarenahen Bereich zu grob
2. Das Modell der ISO: Open System Interconnection: **OSI**: detailliert, keine Implementierung, für Netzwerker **unumgänglich**

Die beiden Schichtenmodelle

DOD

OSI

Die beiden Schichtenmodelle

DOD

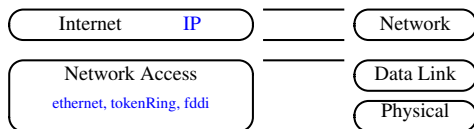
OSI



Die beiden Schichtenmodelle

DOD

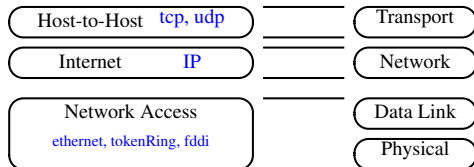
OSI



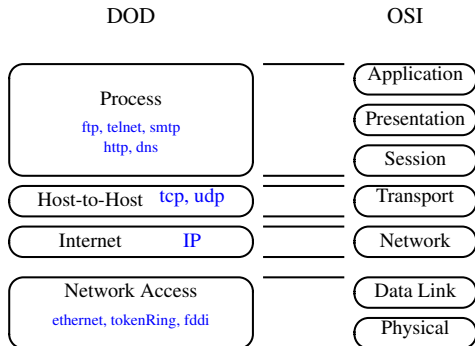
Die beiden Schichtenmodelle

DOD

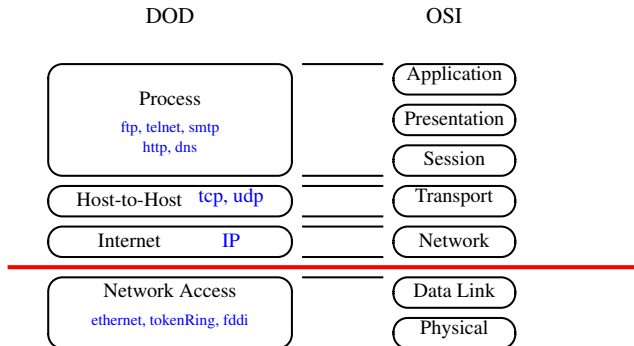
OSI



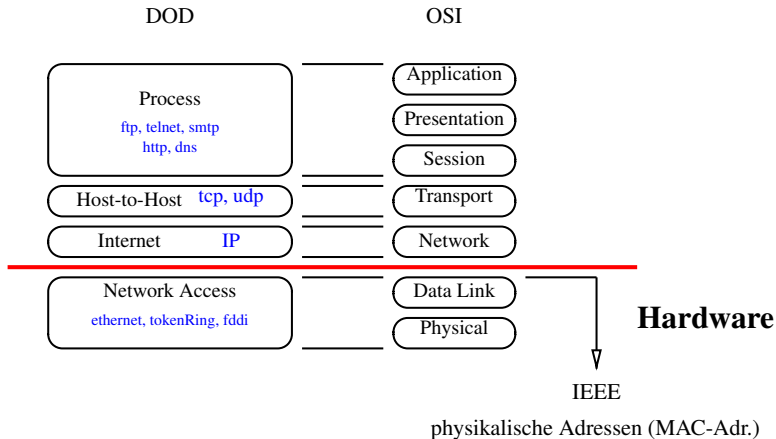
Die beiden Schichtenmodelle



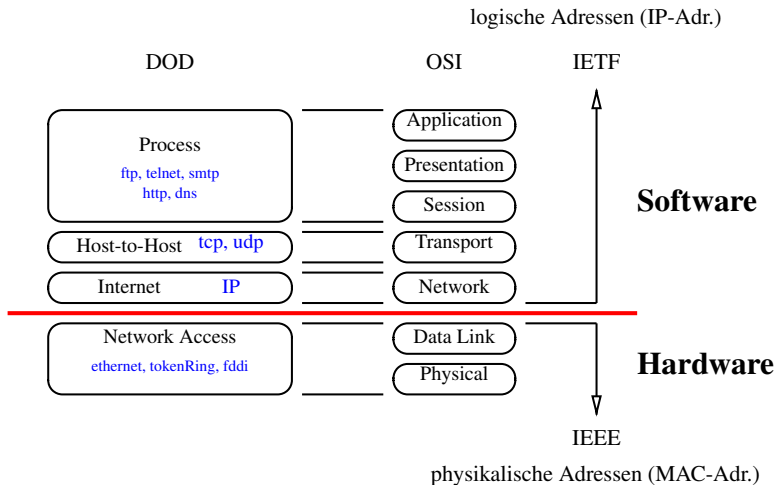
Die beiden Schichtenmodelle



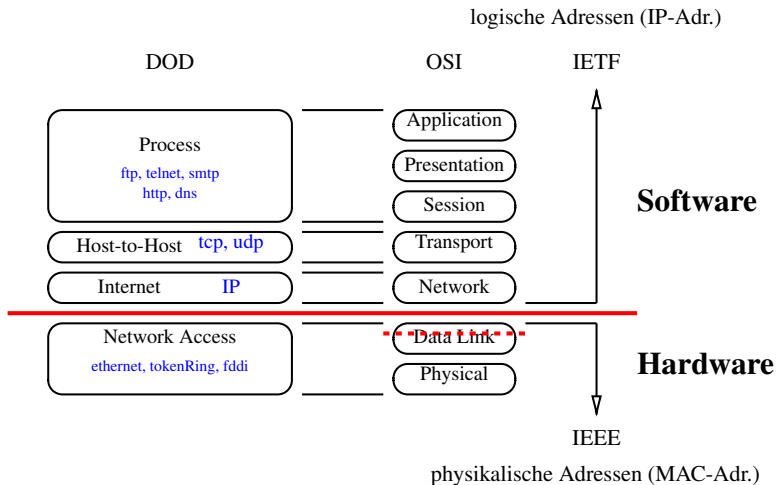
Die beiden Schichtenmodelle



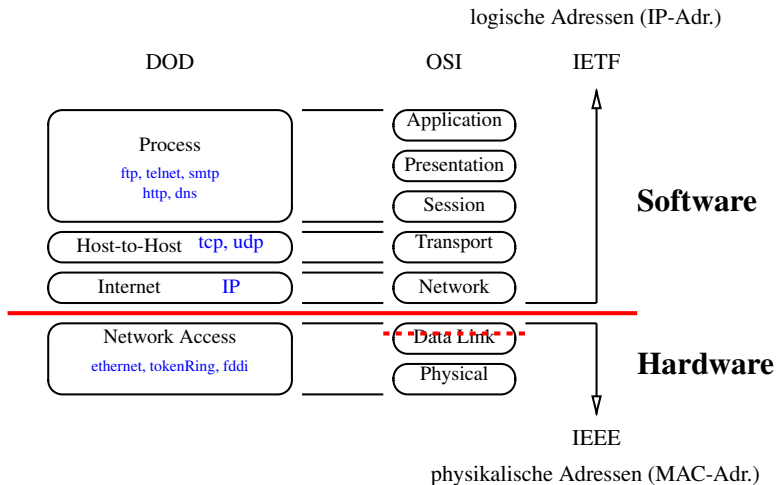
Die beiden Schichtenmodelle



Die beiden Schichtenmodelle



Die beiden Schichtenmodelle



Virtuelle Verbindungen

- Problem: IP arbeitet paketorientiert. Die Endanwendung möchte aber einen kontinuierlichen Strom von einzelnen Bytes.
- Lösung: zwischen IP und die Anwendung kommt eine neue Schicht, die die Paketorientierung **komplett versteckt** und eine Datenschnittstelle zur Verfügung stellt, die sich **wie eine Datei verhält** (UNIX: alles ist eine Datei)

Virtuelle Verbindungen

- Problem: IP arbeitet paketerorientiert. Die Endanwendung möchte aber einen kontinuierlichen Strom von einzelnen Bytes.
- Lösung: zwischen IP und die Anwendung kommt eine neue Schicht, die die Paketerorientierung **komplett versteckt** und eine Datenschnittstelle zur Verfügung stellt, die sich **wie eine Datei verhält** (UNIX: alles ist eine Datei)

Virtuelle Verbindungen

- Problem: IP arbeitet paketorientiert. Die Endanwendung möchte aber einen kontinuierlichen Strom von einzelnen Bytes.
- Lösung: zwischen IP und die Anwendung kommt eine neue Schicht, die die Paketorientierung **komplett versteckt** und eine Datenschnittstelle zur Verfügung stellt, die sich **wie eine Datei verhält** (UNIX: alles ist eine Datei)

Virtuelle Verbindungen

- Problem: IP arbeitet paketorientiert. Die Endanwendung möchte aber einen kontinuierlichen Strom von einzelnen Bytes.
- Lösung: zwischen IP und die Anwendung kommt eine neue Schicht, die die Paketorientierung **komplett versteckt** und eine Datenschnittstelle zur Verfügung stellt, die sich **wie eine Datei verhält** (UNIX: alles ist eine Datei)

Virtuelle Verbindungen

- Problem: IP arbeitet paketorientiert. Die Endanwendung möchte aber einen kontinuierlichen Strom von einzelnen Bytes.
- Lösung: zwischen IP und die Anwendung kommt eine neue Schicht, die die Paketorientierung **komplett versteckt** und eine Datenschnittstelle zur Verfügung stellt, die sich **wie eine Datei verhält** (UNIX: alles ist eine Datei)

Virtuelle Verbindungen

- Problem: IP arbeitet paketorientiert. Die Endanwendung möchte aber einen kontinuierlichen Strom von einzelnen Bytes.
- Lösung: zwischen IP und die Anwendung kommt eine neue Schicht, die die Paketorientierung **komplett versteckt** und eine Datenschnittstelle zur Verfügung stellt, die sich **wie eine Datei verhält** (UNIX: alles ist eine Datei)

Virtuelle Verbindungen mit TCP

- **Transmission Control Protocol - TCP**
- Aus Sicht des Programmentwicklers: TCP stellt **virtuelle** Verbindung her (Bytes durchnummeriert).
Zurück zum Palettenbeispiel:
reale Verbindung Paletten-Förderband
virtuelle Verbindung Just-in-Time-Lieferung →
aufwändiges Transportmanagement
notwendig
- Endpunkte der Verbindung: Sockets (→ Rohrpost)
 - mehrere Sockets pro Host
 - Socket über **Portnummer** identifiziert
 - bestimmte Portnummern werden Anwendungen zugeordnet (eingehende Verbindungen)

Virtuelle Verbindungen mit TCP

- Transmission Control Protocol - TCP
- Aus Sicht des Programmierers: TCP stellt **virtuelle** Verbindung her (Bytes durchnummeriert).
Zurück zum Palettenbeispiel:
reale Verbindung Paletten-Förderband
virtuelle Verbindung Just-in-Time-Lieferung →
aufwändiges Transportmanagement
notwendig
- Endpunkte der Verbindung: Sockets (→ Rohrpost)
 - mehrere Sockets pro Host
 - Socket über **Portnummer** identifiziert
 - bestimmte Portnummern werden Anwendungen zugeordnet (eingehende Verbindungen)

Virtuelle Verbindungen mit TCP

- Transmission Control Protocol - TCP
- Aus Sicht des Programmentwicklers: TCP stellt **virtuelle** Verbindung her (Bytes durchnummeriert).

Zurück zum Palettenbeispiel:

reale Verbindung Paletten-Förderband

virtuelle Verbindung Just-in-Time-Lieferung →

aufwändiges Transportmanagement
notwendig

- Endpunkte der Verbindung: Sockets (→ Rohrpost)
 - mehrere Sockets pro Host
 - Socket über **Portnummer** identifiziert
 - bestimmte Portnummern werden Anwendungen zugeordnet (eingehende Verbindungen)

Virtuelle Verbindungen mit TCP

- Transmission Control Protocol - TCP
- Aus Sicht des Programmentwicklers: TCP stellt **virtuelle** Verbindung her (Bytes durchnummeriert).

Zurück zum Palettenbeispiel:

reale Verbindung Paletten-Förderband

virtuelle Verbindung Just-in-Time-Lieferung →

aufwändiges Transportmanagement
notwendig

- Endpunkte der Verbindung: Sockets (→ Rohrpost)
 - mehrere Sockets pro Host
 - Socket über **Portnummer** identifiziert
 - bestimmte Portnummern werden Anwendungen zugeordnet (eingehende Verbindungen)

Virtuelle Verbindungen mit TCP

- Transmission Control Protocol - TCP
- Aus Sicht des Programmentwicklers: TCP stellt **virtuelle** Verbindung her (Bytes durchnummeriert).
Zurück zum Palettenbeispiel:

reale Verbindung Paletten-Förderband

virtuelle Verbindung Just-in-Time-Lieferung →
aufwändiges Transportmanagement
notwendig

- Endpunkte der Verbindung: Sockets (→ Rohrpost)
 - mehrere Sockets pro Host
 - Socket über **Portnummer** identifiziert
 - bestimmte Portnummern werden Anwendungen zugeordnet (eingehende Verbindungen)

Virtuelle Verbindungen mit TCP

- Transmission Control Protocol - TCP
- Aus Sicht des Programmentwicklers: TCP stellt **virtuelle** Verbindung her (Bytes durchnummeriert).

Zurück zum Palettenbeispiel:

reale Verbindung Paletten-Förderband

virtuelle Verbindung Just-in-Time-Lieferung →
aufwändiges Transportmanagement
notwendig

- Endpunkte der Verbindung: Sockets (→ Rohrpost)
 - mehrere Sockets pro Host
 - Socket über **Portnummer** identifiziert
 - bestimmte Portnummern werden Anwendungen zugeordnet (eingehende Verbindungen)

Virtuelle Verbindungen mit TCP

- Transmission Control Protocol - TCP
- Aus Sicht des Programmierers: TCP stellt **virtuelle** Verbindung her (Bytes durchnummeriert).

Zurück zum Palettenbeispiel:

reale Verbindung Paletten-Förderband

virtuelle Verbindung Just-in-Time-Lieferung →
aufwändiges Transportmanagement
notwendig

- Endpunkte der Verbindung: Sockets (→ Rohrpost)
 - mehrere Sockets pro Host
 - Socket über **Portnummer** identifiziert
 - bestimmte Portnummern werden Anwendungen zugeordnet (eingehende Verbindungen)

Virtuelle Verbindungen mit TCP

- Transmission Control Protocol - TCP
- Aus Sicht des Programmentwicklers: TCP stellt **virtuelle** Verbindung her (Bytes durchnummeriert).

Zurück zum Palettenbeispiel:

reale Verbindung Paletten-Förderband

virtuelle Verbindung Just-in-Time-Lieferung →
aufwändiges Transportmanagement
notwendig

- Endpunkte der Verbindung: Sockets (→ Rohrpost)
 - mehrere Sockets pro Host
 - Socket über **Portnummer** identifiziert
 - bestimmte Portnummern werden Anwendungen zugeordnet (eingehende Verbindungen)

Virtuelle Verbindungen mit TCP

- Transmission Control Protocol - TCP
- Aus Sicht des Programmentwicklers: TCP stellt **virtuelle** Verbindung her (Bytes durchnummeriert).
Zurück zum Palettenbeispiel:
reale Verbindung Paletten-Förderband
virtuelle Verbindung Just-in-Time-Lieferung →
aufwändiges Transportmanagement
notwendig
- Endpunkte der Verbindung: Sockets (→ Rohrpost)
 - mehrere Sockets pro Host
 - Socket über **Portnummer** identifiziert
 - bestimmte Portnummern werden Anwendungen zugeordnet (eingehende Verbindungen)

Virtuelle Verbindungen mit TCP

- Transmission Control Protocol - TCP
- Aus Sicht des Programmentwicklers: TCP stellt **virtuelle** Verbindung her (Bytes durchnummeriert).
Zurück zum Palettenbeispiel:
reale Verbindung Paletten-Förderband
virtuelle Verbindung Just-in-Time-Lieferung →
aufwändiges Transportmanagement
notwendig
- Endpunkte der Verbindung: Sockets (→ Rohrpost)
 - mehrere Sockets pro Host
 - Socket über **Portnummer** identifiziert
 - bestimmte Portnummern werden Anwendungen zugeordnet (eingehende Verbindungen)

Virtuelle Verbindungen mit TCP

- Transmission Control Protocol - TCP
- Aus Sicht des Programmentwicklers: TCP stellt **virtuelle** Verbindung her (Bytes durchnummeriert).

Zurück zum Palettenbeispiel:

reale Verbindung Paletten-Förderband

virtuelle Verbindung Just-in-Time-Lieferung →
aufwändiges Transportmanagement
notwendig

- Endpunkte der Verbindung: Sockets (→ Rohrpost)
 - mehrere Sockets pro Host
 - Socket über **Portnummer** identifiziert
 - bestimmte Portnummern werden Anwendungen zugeordnet (eingehende Verbindungen)

Virtuelle Verbindungen mit TCP

- Transmission Control Protocol - TCP
- Aus Sicht des Programmentwicklers: TCP stellt **virtuelle** Verbindung her (Bytes durchnummeriert).
Zurück zum Palettenbeispiel:
reale Verbindung Paletten-Förderband
virtuelle Verbindung Just-in-Time-Lieferung →
aufwändiges Transportmanagement
notwendig
- Endpunkte der Verbindung: Sockets (→ Rohrpost)
 - mehrere Sockets pro Host
 - Socket über **Portnummer** identifiziert
 - bestimmte Portnummern werden Anwendungen zugeordnet (eingehende Verbindungen)

Praktischer Versuch mit nc

nc: Net Cat

- nc = Schweizer Taschenmesser des Netzwerklers
- nc verbindet Standard-Eingabe (Tastatur) mit Standard-Ausgabe (Terminalfenster) eines entfernten Rechners
- Server:

```
nc -l 5555
```

5555 oder andere Portnummer > 1024

- Client:

```
nc r023-tafel.wara.de 5555
```

statt `r023-tafel.wara.de` Rechnername eines Hosts,
auf dem `nc -l 5555` läuft nehmen

Praktischer Versuch mit nc

nc: Net Cat

- nc = Schweizer Taschenmesser des Netzwerklers
- nc verbindet Standard-Eingabe (Tastatur) mit Standard-Ausgabe (Terminalfenster) eines entfernten Rechners
- Server:

```
nc -l 5555
```

5555 oder andere Portnummer > 1024

- Client:

```
nc r023-tafel.wara.de 5555
```

statt `r023-tafel.wara.de` Rechnername eines Hosts,
auf dem `nc -l 5555` läuft nehmen

Praktischer Versuch mit nc

nc: Net Cat

- nc = Schweizer Taschenmesser des Netzwerkers
- nc verbindet Standard-Eingabe (Tastatur) mit Standard-Ausgabe (Terminalfenster) eines entfernten Rechners
- Server:

```
nc -l 5555
```

5555 oder andere Portnummer > 1024

- Client:

```
nc r023-tafel.wara.de 5555
```

statt `r023-tafel.wara.de` Rechnername eines Hosts, auf dem `nc -l 5555` läuft nehmen

Praktischer Versuch mit nc

nc: Net Cat

- nc = Schweizer Taschenmesser des Netzwerklers
- nc verbindet Standard-Eingabe (Tastatur) mit Standard-Ausgabe (Terminalfenster) eines entfernten Rechners
- Server:

```
nc -l 5555
```

5555 oder andere Portnummer > 1024

- Client:

```
nc r023-tafel.wara.de 5555
```

statt `r023-tafel.wara.de` Rechnername eines Hosts,
auf dem `nc -l 5555` läuft nehmen

Praktischer Versuch mit nc

nc: Net Cat

- nc = Schweizer Taschenmesser des Netzwerklers
- nc verbindet Standard-Eingabe (Tastatur) mit Standard-Ausgabe (Terminalfenster) eines entfernten Rechners
- Server:

```
nc -l 5555
```

5555 oder andere Portnummer > 1024

- Client:

```
nc r023-tafel.wara.de 5555
```

statt `r023-tafel.wara.de` Rechnername eines Hosts,
auf dem `nc -l 5555` läuft nehmen

Praktischer Versuch mit nc

nc: Net Cat

- nc = Schweizer Taschenmesser des Netzwerkers
- nc verbindet Standard-Eingabe (Tastatur) mit Standard-Ausgabe (Terminalfenster) eines entfernten Rechners
- Server:

```
nc -l 5555
```

5555 oder andere Portnummer > 1024

- Client:

```
nc r023-tafel.wara.de 5555
```

statt `r023-tafel.wara.de` Rechnername eines Hosts,
auf dem `nc -l 5555` läuft nehmen

Praktischer Versuch mit nc

nc: Net Cat

- nc = Schweizer Taschenmesser des Netzwerklers
- nc verbindet Standard-Eingabe (Tastatur) mit Standard-Ausgabe (Terminalfenster) eines entfernten Rechners
- Server:

```
nc -l 5555
```

5555 oder andere Portnummer > 1024

- Client:

```
nc r023-tafel.wara.de 5555
```

statt `r023-tafel.wara.de` Rechnername eines Hosts,
auf dem `nc -l 5555` läuft nehmen

Praktischer Versuch mit nc

nc: Net Cat

- nc = Schweizer Taschenmesser des Netzwerklers
- nc verbindet Standard-Eingabe (Tastatur) mit Standard-Ausgabe (Terminalfenster) eines entfernten Rechners
- Server:

```
nc -l 5555
```

5555 oder andere Portnummer > 1024

- Client:

```
nc r023-tafel.wara.de 5555
```

statt `r023-tafel.wara.de` Rechnername eines Hosts,
auf dem `nc -l 5555` läuft nehmen

Praktischer Versuch mit nc

nc: Net Cat

- nc = Schweizer Taschenmesser des Netzwerkers
- nc verbindet Standard-Eingabe (Tastatur) mit Standard-Ausgabe (Terminalfenster) eines entfernten Rechners
- Server:

```
nc -l 5555
```

5555 oder andere Portnummer > 1024

- Client:

```
nc r023-tafel.wara.de 5555
```

statt `r023-tafel.wara.de` Rechnername eines Hosts,
auf dem `nc -l 5555` läuft nehmen

Praktischer Versuch mit nc

nc: Net Cat

- nc = Schweizer Taschenmesser des Netzwerklers
- nc verbindet Standard-Eingabe (Tastatur) mit Standard-Ausgabe (Terminalfenster) eines entfernten Rechners
- Server:

```
nc -l 5555
```

5555 oder andere Portnummer > 1024

- Client:

```
nc r023-tafel.wara.de 5555
```

statt `r023-tafel.wara.de` Rechnername eines Hosts,
auf dem `nc -l 5555` läuft nehmen

Datentelegramme mit UDP

- Schwesterprotokoll: UDP
 - UDP arbeitet auch mit Portnummern
 - kein Bytestrom
 - ⇒ bis 64kiB grosse **Datagramme**
 - ⇒ wegschicken und vergessen
- deutlich schneller als TCP
- keine Sicherung der Übertragung

Datentelegramme mit UDP

- Schwesterprotokoll: UDP
 - UDP arbeitet auch mit Portnummern
 - kein Bytestrom
 - ⇒ bis 64kiB grosse **Datagramme**
 - ⇒ wegschicken und vergessen
- deutlich schneller als TCP
- keine Sicherung der Übertragung

Datentelegramme mit UDP

- Schwesterprotokoll: UDP
 - UDP arbeitet auch mit Portnummern
 - kein Bytestrom
 - ⇒ bis 64kiB grosse **Datagramme**
 - ⇒ wegschicken und vergessen
- deutlich schneller als TCP
- keine Sicherung der Übertragung

Datentelegramme mit UDP

- Schwesterprotokoll: UDP
 - UDP arbeitet auch mit Portnummern
 - kein Bytestrom
 - ⇒ bis 64kiB grosse **Datagramme**
 - ⇒ wegschicken und vergessen
- deutlich schneller als TCP
- keine Sicherung der Übertragung

Datentelegramme mit UDP

- Schwesterprotokoll: UDP
 - UDP arbeitet auch mit Portnummern
 - kein Bytestrom
 - ⇒ bis 64kiB grosse **Datagramme**
 - ⇒ wegschicken und vergessen
- deutlich schneller als TCP
- keine Sicherung der Übertragung

Datentelegramme mit UDP

- Schwesterprotokoll: UDP
 - UDP arbeitet auch mit Portnummern
 - kein Bytestrom
 - ⇒ bis 64kiB grosse **Datagramme**
 - ⇒ wegschicken und vergessen
- deutlich schneller als TCP
- keine Sicherung der Übertragung

Datentelegramme mit UDP

- Schwesterprotokoll: UDP
 - UDP arbeitet auch mit Portnummern
 - kein Bytestrom
 - ⇒ bis 64kiB grosse **Datagramme**
 - ⇒ wegschicken und vergessen
- deutlich schneller als TCP
- keine Sicherung der Übertragung

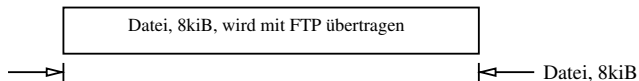
Datentelegramme mit UDP

- Schwesterprotokoll: UDP
 - UDP arbeitet auch mit Portnummern
 - kein Bytestrom
 - ⇒ bis 64kiB grosse **Datagramme**
 - ⇒ wegschicken und vergessen
- deutlich schneller als TCP
- keine Sicherung der Übertragung

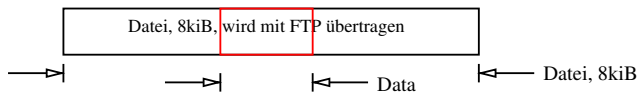
Kapselung: mehrfache Verpackung der Daten

Datei, 8kiB, wird mit FTP übertragen

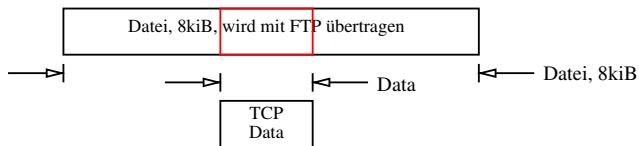
Kapselung: mehrfache Verpackung der Daten



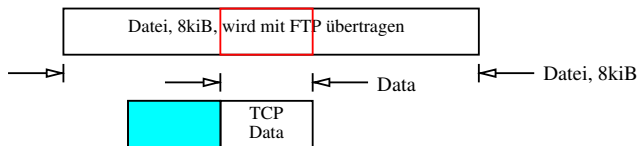
Kapselung: mehrfache Verpackung der Daten



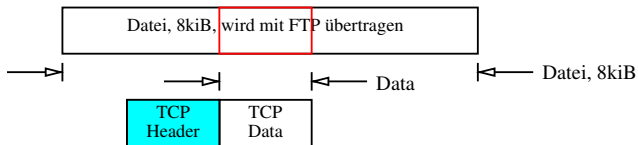
Kapselung: mehrfache Verpackung der Daten



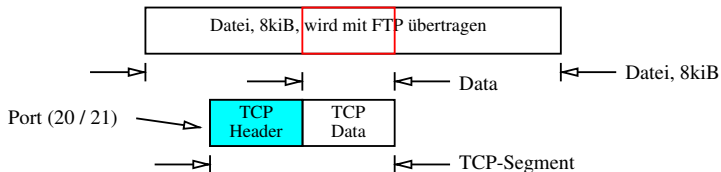
Kapselung: mehrfache Verpackung der Daten



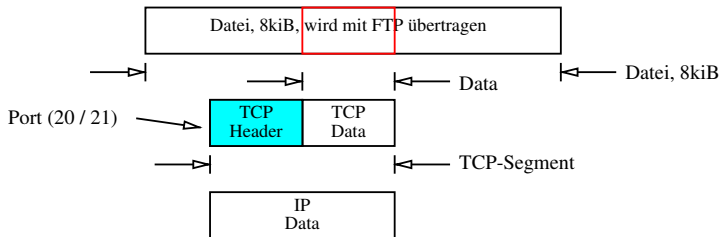
Kapselung: mehrfache Verpackung der Daten



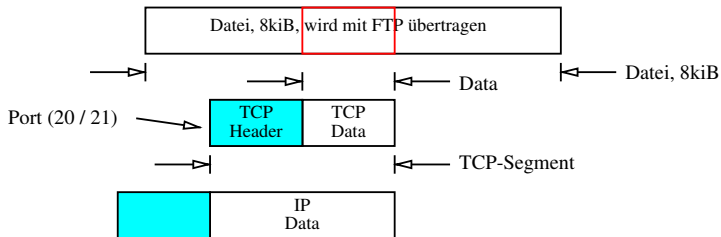
Kapselung: mehrfache Verpackung der Daten



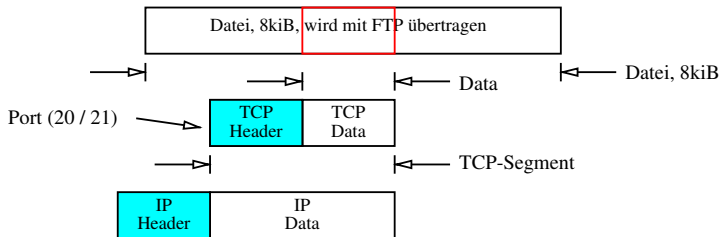
Kapselung: mehrfache Verpackung der Daten



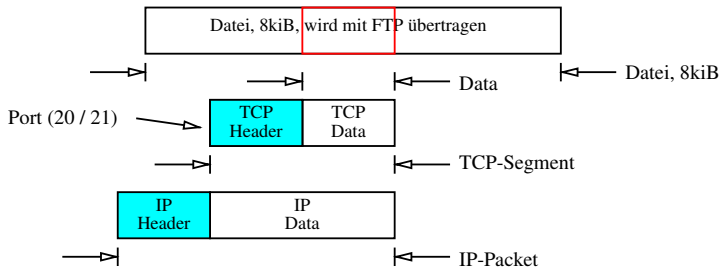
Kapselung: mehrfache Verpackung der Daten



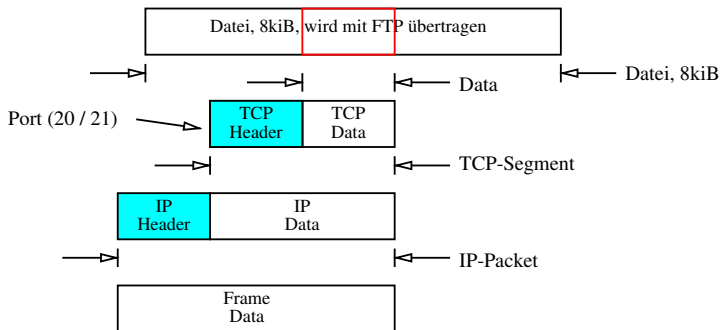
Kapselung: mehrfache Verpackung der Daten



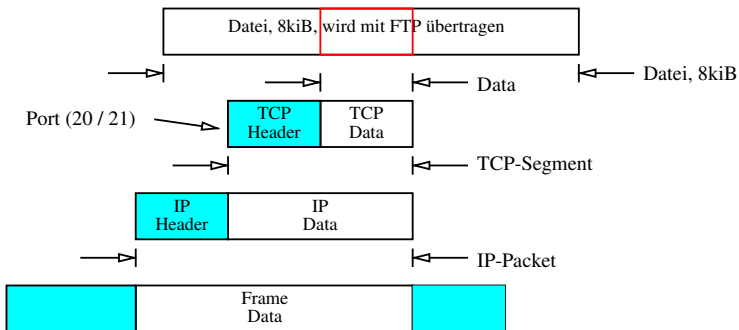
Kapselung: mehrfache Verpackung der Daten



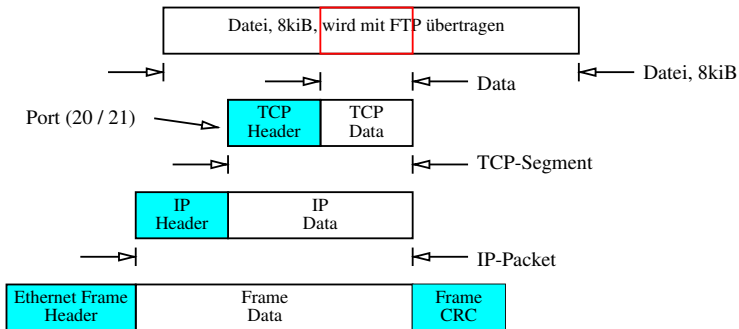
Kapselung: mehrfache Verpackung der Daten



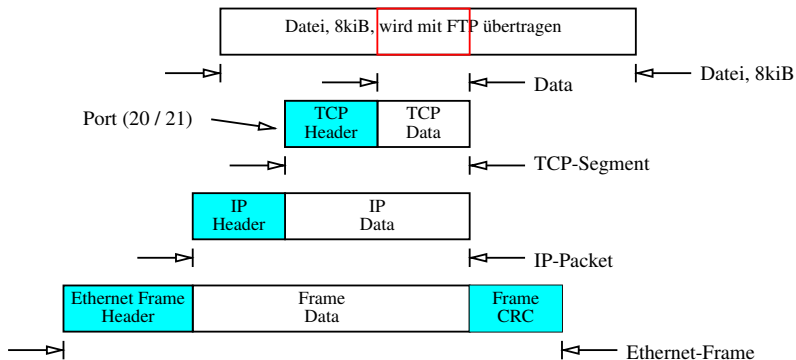
Kapselung: mehrfache Verpackung der Daten



Kapselung: mehrfache Verpackung der Daten



Kapselung: mehrfache Verpackung der Daten



Inhalt

Vom LAN zum Internet

Rahmen, Pakete, virtuelle Verbindungen

Versuch: Daten mit nc übertragen und Datenverkehr mitschneiden

IP-Adressen und Subnetze

Address Resolution Protocol

Routing

Analyse des Datenverkehrs

Im Folgenden soll der Datenverkehr einer Client-Server-Verbindung mit nc analysiert werden.

Dazu benötigt man:

- Netzwerk-Mitschneideprogramm: tcpdump oder wireshark
- Kenntnis des Hexadezimalsystems: wird vorausgesetzt
- Details des TCP- und IP-Headers und des Ethernet-Frames: siehe nächste Folien

Analyse des Datenverkehrs

Im Folgenden soll der Datenverkehr einer Client-Server-Verbindung mit nc analysiert werden.

Dazu benötigt man:

- Netzwerk-Mitschneideprogramm: tcpdump oder wireshark
- Kenntnis des Hexadezimalsystems: wird vorausgesetzt
- Details des TCP- und IP-Headers und des Ethernet-Frames: siehe nächste Folien

Analyse des Datenverkehrs

Im Folgenden soll der Datenverkehr einer Client-Server-Verbindung mit nc analysiert werden.

Dazu benötigt man:

- Netzwerk-Mitschneideprogramm: tcpdump oder wireshark
- Kenntnis des Hexadezimalsystems: wird vorausgesetzt
- Details des TCP- und IP-Headers und des Ethernet-Frames: siehe nächste Folien

Analyse des Datenverkehrs

Im Folgenden soll der Datenverkehr einer Client-Server-Verbindung mit nc analysiert werden.

Dazu benötigt man:

- Netzwerk-Mitschneideprogramm: tcpdump oder wireshark
- Kenntnis des Hexadezimalsystems: wird vorausgesetzt
- Details des TCP- und IP-Headers und des Ethernet-Frames: siehe nächste Folien

Analyse des Datenverkehrs

Im Folgenden soll der Datenverkehr einer Client-Server-Verbindung mit nc analysiert werden.

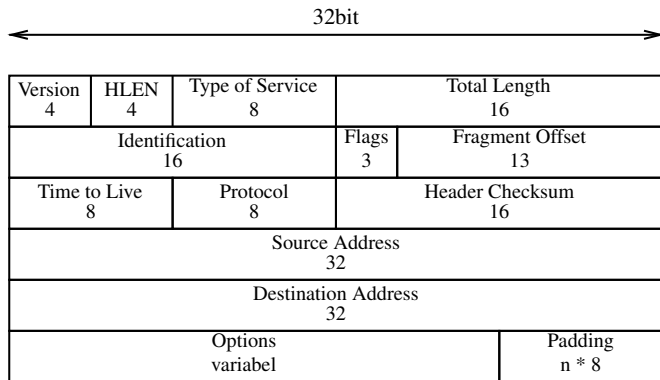
Dazu benötigt man:

- Netzwerk-Mitschneideprogramm: tcpdump oder wireshark
- Kenntnis des Hexadezimalsystems: wird vorausgesetzt
- Details des TCP- und IP-Headers und des Ethernet-Frames: siehe nächste Folien

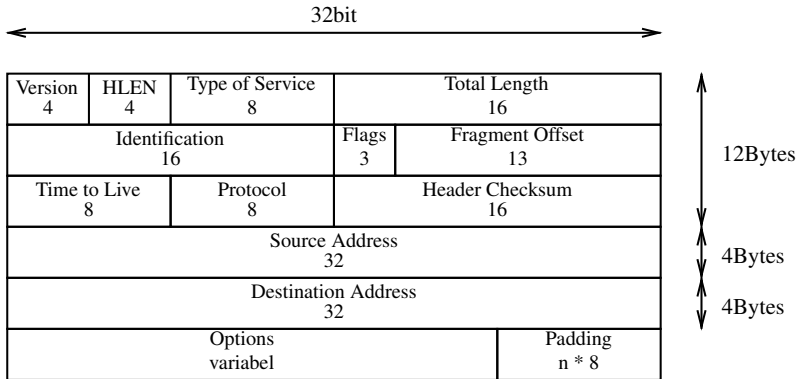
Der IP-Header im Detail

Version 4	HLEN 4	Type of Service 8	Total Length 16	
Identification 16			Flags 3	Fragment Offset 13
Time to Live 8		Protocol 8	Header Checksum 16	
Source Address 32				
Destination Address 32				
Options variabel				Padding n * 8

Der IP-Header im Detail



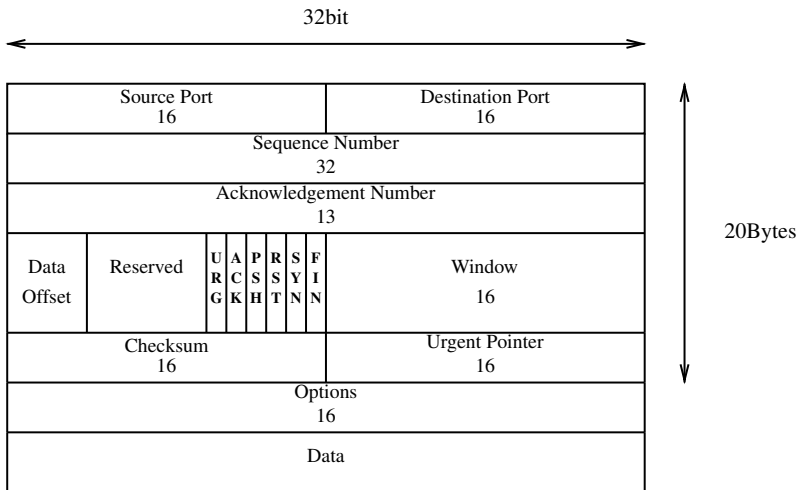
Der IP-Header im Detail



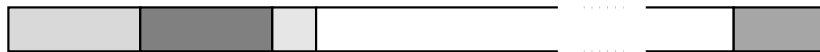
Der TCP-Header im Detail

Source Port 16				Destination Port 16				
Sequence Number 32								
Acknowledgement Number 13								
Data Offset	Reserved	U R G	A C K	P S H	R S T	S Y N	F I N	Window 16
Checksum 16				Urgent Pointer 16				
Options 16								
Data								

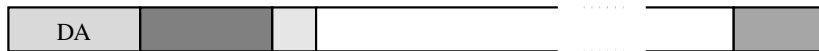
Der TCP-Header im Detail



Ein Ethernet-Frame im Detail

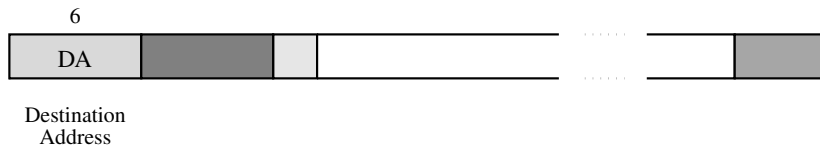


Ein Ethernet-Frame im Detail

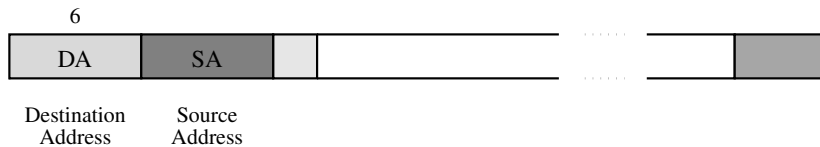


Destination
Address

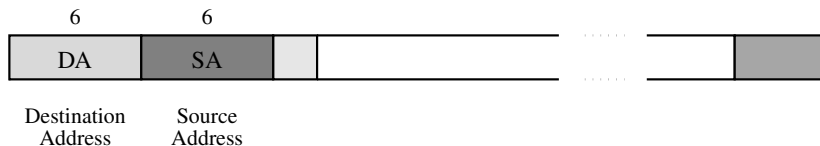
Ein Ethernet-Frame im Detail



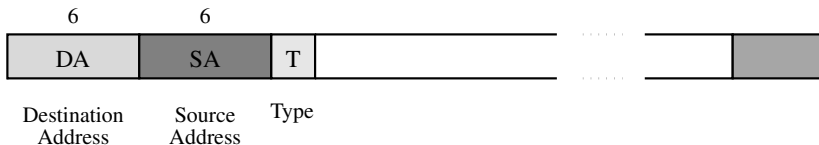
Ein Ethernet-Frame im Detail



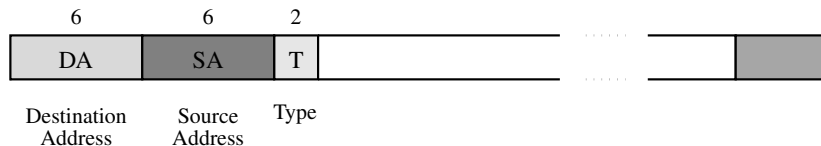
Ein Ethernet-Frame im Detail



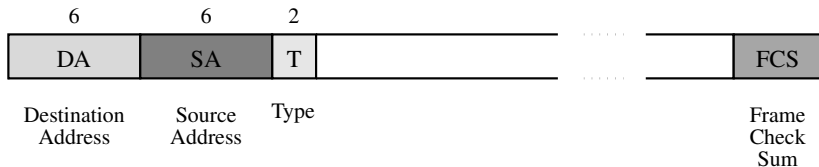
Ein Ethernet-Frame im Detail



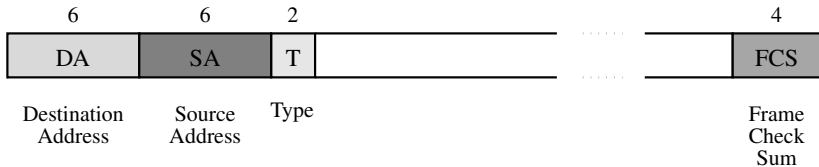
Ein Ethernet-Frame im Detail



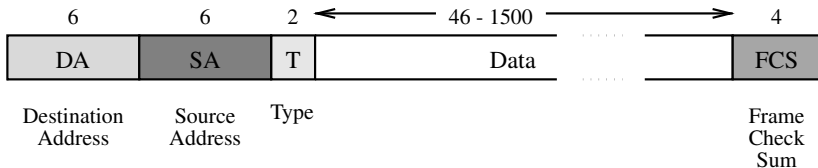
Ein Ethernet-Frame im Detail



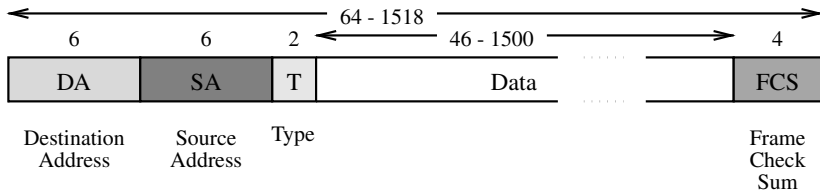
Ein Ethernet-Frame im Detail



Ein Ethernet-Frame im Detail



Ein Ethernet-Frame im Detail



Ein Ethernet-Frame mit eingebettetem IP-Paket

Ein Ethernet-**Frame**

Ein Ethernet-Frame mit eingebettetem IP-Paket

Ein Ethernet-**Frame** mit eingebettetem IP-**Paket**

Ein Ethernet-Frame mit eingebettetem IP-Paket

Ein Ethernet-**Frame** mit eingebettetem IP-**Paket** mit eingebettetem TCP-**Segment**

Ein Ethernet-Frame mit eingebettetem IP-Paket

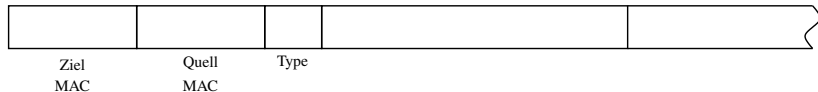
Ein Ethernet-**Frame** mit eingebettetem IP-**Paket** mit eingebettetem TCP-**Segment**

Die eigentlichen Daten stehen weiter rechts und sind nicht mehr auf dem Bild:

Ein Ethernet-Frame mit eingebettetem IP-Paket

Ein Ethernet-**Frame** mit eingebettetem IP-**Paket** mit eingebettetem TCP-**Segment**

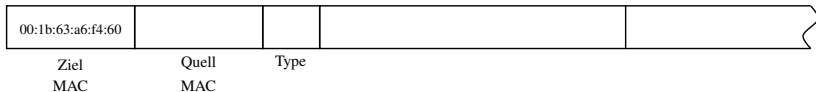
Die eigentlichen Daten stehen weiter rechts und sind nicht mehr auf dem Bild:



Ein Ethernet-Frame mit eingebettetem IP-Paket

Ein Ethernet-**Frame** mit eingebettetem IP-**Paket** mit eingebettetem TCP-**Segment**

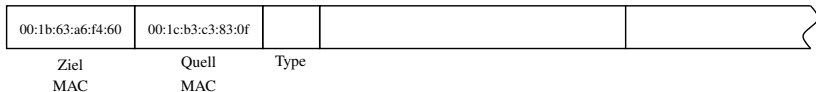
Die eigentlichen Daten stehen weiter rechts und sind nicht mehr auf dem Bild:



Ein Ethernet-Frame mit eingebettetem IP-Paket

Ein Ethernet-**Frame** mit eingebettetem IP-**Paket** mit eingebettetem TCP-**Segment**

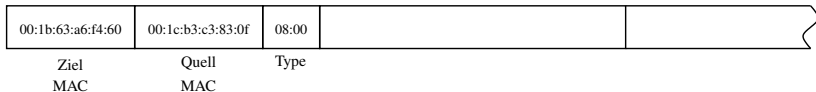
Die eigentlichen Daten stehen weiter rechts und sind nicht mehr auf dem Bild:



Ein Ethernet-Frame mit eingebettetem IP-Paket

Ein Ethernet-**Frame** mit eingebettetem IP-**Paket** mit eingebettetem TCP-**Segment**

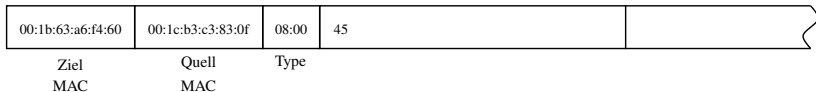
Die eigentlichen Daten stehen weiter rechts und sind nicht mehr auf dem Bild:



Ein Ethernet-Frame mit eingebettetem IP-Paket

Ein Ethernet-**Frame** mit eingebettetem IP-**Paket** mit eingebettetem TCP-**Segment**

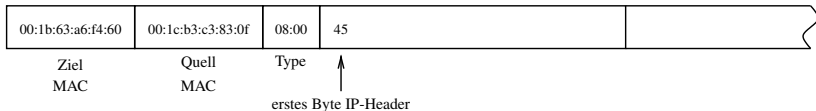
Die eigentlichen Daten stehen weiter rechts und sind nicht mehr auf dem Bild:



Ein Ethernet-Frame mit eingebettetem IP-Paket

Ein Ethernet-**Frame** mit eingebettetem IP-**Paket** mit eingebettetem TCP-**Segment**

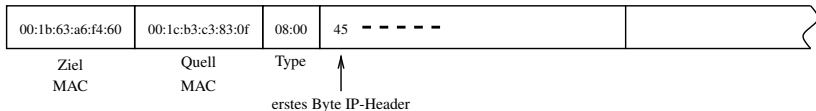
Die eigentlichen Daten stehen weiter rechts und sind nicht mehr auf dem Bild:



Ein Ethernet-Frame mit eingebettetem IP-Paket

Ein Ethernet-**Frame** mit eingebettetem IP-**Paket** mit eingebettetem TCP-**Segment**

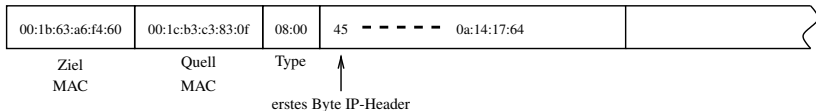
Die eigentlichen Daten stehen weiter rechts und sind nicht mehr auf dem Bild:



Ein Ethernet-Frame mit eingebettetem IP-Paket

Ein Ethernet-**Frame** mit eingebettetem IP-**Paket** mit eingebettetem TCP-**Segment**

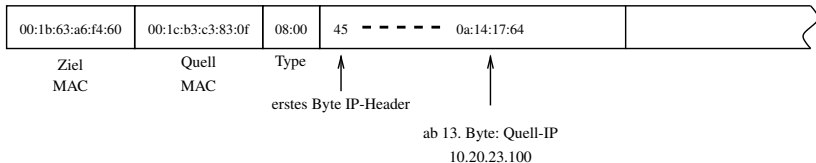
Die eigentlichen Daten stehen weiter rechts und sind nicht mehr auf dem Bild:



Ein Ethernet-Frame mit eingebettetem IP-Paket

Ein Ethernet-**Frame** mit eingebettetem IP-**Paket** mit eingebettetem TCP-**Segment**

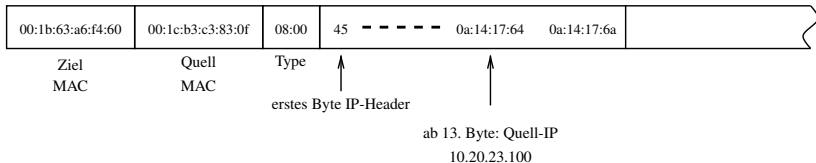
Die eigentlichen Daten stehen weiter rechts und sind nicht mehr auf dem Bild:



Ein Ethernet-Frame mit eingebettetem IP-Paket

Ein Ethernet-**Frame** mit eingebettetem IP-**Paket** mit eingebettetem TCP-**Segment**

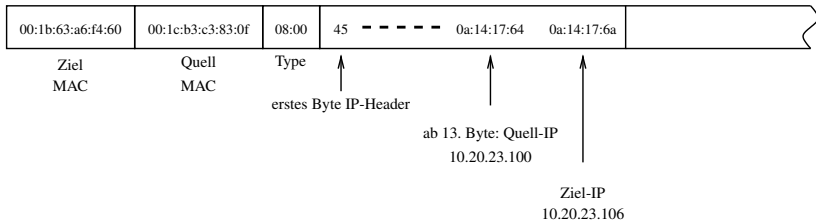
Die eigentlichen Daten stehen weiter rechts und sind nicht mehr auf dem Bild:



Ein Ethernet-Frame mit eingebettetem IP-Paket

Ein Ethernet-**Frame** mit eingebettetem IP-**Paket** mit eingebettetem TCP-**Segment**

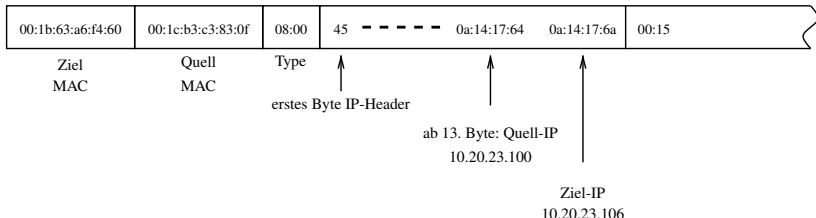
Die eigentlichen Daten stehen weiter rechts und sind nicht mehr auf dem Bild:



Ein Ethernet-Frame mit eingebettetem IP-Paket

Ein Ethernet-**Frame** mit eingebettetem IP-**Paket** mit eingebettetem TCP-**Segment**

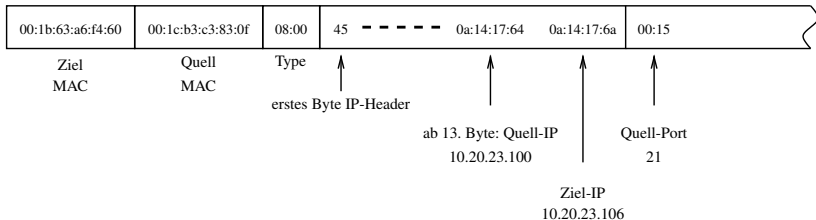
Die eigentlichen Daten stehen weiter rechts und sind nicht mehr auf dem Bild:



Ein Ethernet-Frame mit eingebettetem IP-Paket

Ein Ethernet-**Frame** mit eingebettetem IP-**Paket** mit eingebettetem TCP-**Segment**

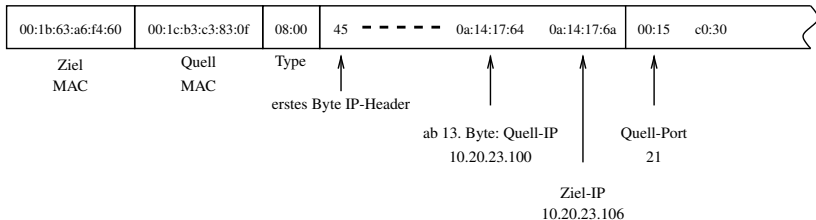
Die eigentlichen Daten stehen weiter rechts und sind nicht mehr auf dem Bild:



Ein Ethernet-Frame mit eingebettetem IP-Paket

Ein Ethernet-**Frame** mit eingebettetem IP-**Paket** mit eingebettetem TCP-**Segment**

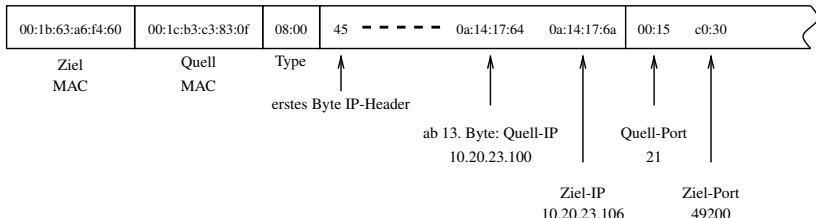
Die eigentlichen Daten stehen weiter rechts und sind nicht mehr auf dem Bild:



Ein Ethernet-Frame mit eingebettetem IP-Paket

Ein Ethernet-**Frame** mit eingebettetem IP-**Paket** mit eingebettetem TCP-**Segment**

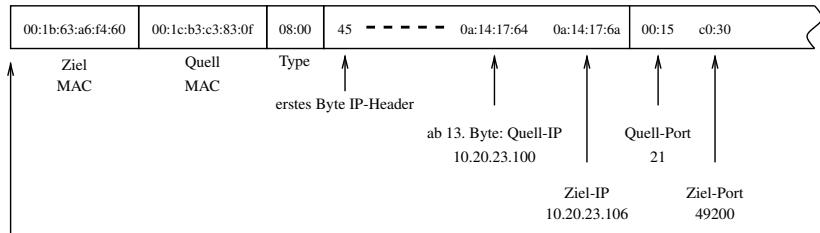
Die eigentlichen Daten stehen weiter rechts und sind nicht mehr auf dem Bild:



Ein Ethernet-Frame mit eingebettetem IP-Paket

Ein Ethernet-**Frame** mit eingebettetem IP-**Paket** mit eingebettetem TCP-**Segment**

Die eigentlichen Daten stehen weiter rechts und sind nicht mehr auf dem Bild:



Ab hier wird aufgezeichnet

Aufzeichnung von Netzwerkverkehr

Aufgabe:

Aufzeichnung von Netzwerkverkehr

Aufgabe:

Mit dem Linux-Kommando `tcpdump` sollen Netzwerkpakete aufgezeichnet und anschliessend analysiert werden.

Aufzeichnung von Netzwerkverkehr

Aufgabe:

Mit dem Linux-Kommando `tcpdump` sollen Netzwerkpakete aufgezeichnet und anschliessend analysiert werden.

Hier die wichtigsten Optionen von `tcpdump`:

Aufzeichnung von Netzwerkverkehr

Aufgabe:

Mit dem Linux-Kommando `tcpdump` sollen Netzwerkpakete aufgezeichnet und anschliessend analysiert werden.

Hier die wichtigsten Optionen von `tcpdump`:

- n alle Ports und IP-Adressen werden als Zahlen und nicht mit ihrem Namen ausgegeben

Aufzeichnung von Netzwerkverkehr

Aufgabe:

Mit dem Linux-Kommando `tcpdump` sollen Netzwerkpakete aufgezeichnet und anschliessend analysiert werden.

Hier die wichtigsten Optionen von `tcpdump`:

- n alle Ports und IP-Adressen werden als Zahlen und nicht mit ihrem Namen ausgegeben
- t der Zeitstempel bei der Aufzeichnung wird unterdrückt

Aufzeichnung von Netzwerkverkehr

Aufgabe:

Mit dem Linux-Kommando `tcpdump` sollen Netzwerkpakete aufgezeichnet und anschliessend analysiert werden.

Hier die wichtigsten Optionen von `tcpdump`:

- n alle Ports und IP-Adressen werden als Zahlen und nicht mit ihrem Namen ausgegeben
- t der Zeitstempel bei der Aufzeichnung wird unterdrückt
- i nach -i wird das Netzwerkinterface angegeben (z.B. `eth0`)

Aufzeichnung von Netzwerkverkehr

Aufgabe:

Mit dem Linux-Kommando `tcpdump` sollen Netzwerkpakete aufgezeichnet und anschliessend analysiert werden.

Hier die wichtigsten Optionen von `tcpdump`:

- n alle Ports und IP-Adressen werden als Zahlen und nicht mit ihrem Namen ausgegeben
- t der Zeitstempel bei der Aufzeichnung wird unterdrückt
- i nach -i wird das Netzwerkinterface angegeben (z.B. `eth0`)
- XX es wird der Header **und** der Paketinhalt Hexadezimal und in ASCII ausgegeben

Aufzeichnung von Netzwerkverkehr

Aufgabe:

Mit dem Linux-Kommando `tcpdump` sollen Netzwerkpakete aufgezeichnet und anschliessend analysiert werden.

Hier die wichtigsten Optionen von `tcpdump`:

- n alle Ports und IP-Adressen werden als Zahlen und nicht mit ihrem Namen ausgegeben
- t der Zeitstempel bei der Aufzeichnung wird unterdrückt
- i nach -i wird das Netzwerkinterface angegeben (z.B. `eth0`)
- XX es wird der Header **und** der Paketinhalt Hexadezimal und in ASCII ausgegeben
- s 200 es werden 200Bytes pro Rahmen aufgezeichnet

Aufzeichnung von Netzwerkverkehr

Suchmuster am Ende des Befehls kann man ein Suchmuster angeben. Z.B.

```
ip host 10.1.25.103
```

→ Nur Pakete mit der Adresse 10.1.25.103 werden aufgezeichnet.

In den aufgezeichneten Paketen soll Folgendes gekennzeichnet werden:

- alle PDUs (Frame, Packet, Segment)
- MAC-Ziel, -Quelladresse
- Protocoll-Type (0800 / 0806), IP-Version, HLEN
- IP-Ziel, -Quelladresse

Aufzeichnung von Netzwerkverkehr

Suchmuster am Ende des Befehls kann man ein Suchmuster angeben. Z.B.

```
ip host 10.1.25.103
```

→ Nur Pakete mit der Adresse 10.1.25.103 werden aufgezeichnet.

In den aufgezeichneten Paketen soll Folgendes gekennzeichnet werden:

- alle PDUs (Frame, Packet, Segment)
- MAC-Ziel, -Quelladresse
- Protocoll-Type (0800 / 0806), IP-Version, HLEN
- IP-Ziel, -Quelladresse

Aufzeichnung von Netzwerkverkehr

Suchmuster am Ende des Befehls kann man ein Suchmuster angeben. Z.B.

```
ip host 10.1.25.103
```

→ Nur Pakete mit der Adresse 10.1.25.103 werden aufgezeichnet.

In den aufgezeichneten Paketen soll Folgendes gekennzeichnet werden:

- alle PDUs (Frame, Packet, Segment)
- MAC-Ziel, -Quelladresse
- Protocoll-Type (0800 / 0806), IP-Version, HLEN
- IP-Ziel, -Quelladresse

Aufzeichnung von Netzwerkverkehr

Suchmuster am Ende des Befehls kann man ein Suchmuster angeben. Z.B.

```
ip host 10.1.25.103
```

→ Nur Pakete mit der Adresse 10.1.25.103 werden aufgezeichnet.

In den aufgezeichneten Paketen soll Folgendes gekennzeichnet werden:

- alle PDUs (Frame, Packet, Segment)
- MAC-Ziel, -Quelladresse
- Protocoll-Type (0800 / 0806), IP-Version, HLEN
- IP-Ziel, -Quelladresse

Aufzeichnung von Netzwerkverkehr

Suchmuster am Ende des Befehls kann man ein Suchmuster angeben. Z.B.

```
ip host 10.1.25.103
```

→ Nur Pakete mit der Adresse 10.1.25.103 werden aufgezeichnet.

In den aufgezeichneten Paketen soll Folgendes gekennzeichnet werden:

- alle PDUs (Frame, Packet, Segment)
- MAC-Ziel, -Quelladresse
- Protocoll-Type (0800 / 0806), IP-Version, HLEN
- IP-Ziel, -Quelladresse

Aufzeichnung von Netzwerkverkehr

Suchmuster am Ende des Befehls kann man ein Suchmuster angeben. Z.B.

```
ip host 10.1.25.103
```

→ Nur Pakete mit der Adresse 10.1.25.103 werden aufgezeichnet.

In den aufgezeichneten Paketen soll Folgendes gekennzeichnet werden:

- alle PDUs (Frame, Packet, Segment)
- MAC-Ziel, -Quelladresse
- Protocoll-Type (0800 / 0806), IP-Version, HLEN
- IP-Ziel, -Quelladresse

Aufzeichnung von Netzwerkverkehr

Suchmuster am Ende des Befehls kann man ein Suchmuster angeben. Z.B.

```
ip host 10.1.25.103
```

→ Nur Pakete mit der Adresse 10.1.25.103 werden aufgezeichnet.

In den aufgezeichneten Paketen soll Folgendes gekennzeichnet werden:

- alle PDUs (Frame, Packet, Segment)
- MAC-Ziel, -Quelladresse
- Protocoll-Type (0800 / 0806), IP-Version, HLEN
- IP-Ziel, -Quelladresse

Aufzeichnung von Netzwerkverkehr

Suchmuster am Ende des Befehls kann man ein Suchmuster angeben. Z.B.

```
ip host 10.1.25.103
```

→ Nur Pakete mit der Adresse 10.1.25.103 werden aufgezeichnet.

In den aufgezeichneten Paketen soll Folgendes gekennzeichnet werden:

- alle PDUs (Frame, Packet, Segment)
- MAC-Ziel, -Quelladresse
- Protocoll-Type (0800 / 0806), IP-Version, HLEN
- IP-Ziel, -Quelladresse

Aufzeichnung von Netzwerkverkehr

Vorgehensweise:

- Rechner starten, Ubuntu-Symbol anklicken
- Als lfb-Benutzer anmelden, Terminal starten (Zubehör)
- im Terminal das Kommando su ausführen → **Passwort wird bekannt gegeben**
- tcpdump starten

Beispiel für Aufruf von tcpdump:

```
tcpdump -ntXXi eth0 -s 200 ip host r023-lehrer
```

Aufzeichnung von Netzwerkverkehr

Vorgehensweise:

- Rechner starten, Ubuntu-Symbol anklicken
- Als lfb-Benutzer anmelden, Terminal starten (Zubehör)
- im Terminal das Kommando su ausführen → **Passwort wird bekannt gegeben**
- tcpdump starten

Beispiel für Aufruf von tcpdump:

```
tcpdump -ntXXi eth0 -s 200 ip host r023-lehrer
```

Aufzeichnung von Netzwerkverkehr

Vorgehensweise:

- Rechner starten, Ubuntu-Symbol anklicken
- Als lfb-Benutzer anmelden, Terminal starten (Zubehör)
- im Terminal das Kommando `su` ausführen → **Passwort wird bekannt gegeben**
- `tcpdump` starten

Beispiel für Aufruf von `tcpdump`:

```
tcpdump -ntXXi eth0 -s 200 ip host r023-lehrer
```

Aufzeichnung von Netzwerkverkehr

Vorgehensweise:

- Rechner starten, Ubuntu-Symbol anklicken
- Als lfb-Benutzer anmelden, Terminal starten (Zubehör)
- im Terminal das Kommando su ausführen → **Passwort wird bekannt gegeben**
- tcpdump starten

Beispiel für Aufruf von tcpdump:

```
tcpdump -ntXXi eth0 -s 200 ip host r023-lehrer
```

Aufzeichnung von Netzwerkverkehr

Vorgehensweise:

- Rechner starten, Ubuntu-Symbol anklicken
- Als lfb-Benutzer anmelden, Terminal starten (Zubehör)
- im Terminal das Kommando su ausführen → **Passwort wird bekannt gegeben**
- tcpdump starten

Beispiel für Aufruf von tcpdump:

```
tcpdump -ntXXi eth0 -s 200 ip host r023-lehrer
```

Aufzeichnung von Netzwerkverkehr

Vorgehensweise:

- Rechner starten, Ubuntu-Symbol anklicken
- Als lfb-Benutzer anmelden, Terminal starten (Zubehör)
- im Terminal das Kommando su ausführen → **Passwort wird bekannt gegeben**
- tcpdump starten

Beispiel für Aufruf von tcpdump:

```
tcpdump -ntXXi eth0 -s 200 ip host r023-lehrer
```


Aufzeichnung von Netzwerkverkehr

Vorgehensweise:

- Rechner starten, Ubuntu-Symbol anklicken
- Als lfb-Benutzer anmelden, Terminal starten (Zubehör)
- im Terminal das Kommando su ausführen → **Passwort wird bekannt gegeben**
- tcpdump starten

Beispiel für Aufruf von tcpdump:

```
tcpdump -ntXXi eth0 -s 200 ip host r023-lehrer
```

Inhalt

Vom LAN zum Internet

Rahmen, Pakete, virtuelle Verbindungen

Versuch: Daten mit nc übertragen und Datenverkehr mitschneiden

IP-Adressen und Subnetze

Address Resolution Protocol

Routing

32bit IPv4-Adressen

- IPv4-Adresen sind 32 bit lang \Rightarrow es gibt $2^{32} = 4\,294\,967\,296$ Adressen
- IPv4-Adressen werden in der **Dotted Decimal Notation** geschrieben:

129.143.14.155

- Die 32bit der Adresse werden in 4 Oktette geteilt jedes Oktett wird ins Dezimalsystem umgewandelt \rightarrow mathematisch unsinnige Schreibweise.
- Beispiel:

10000001	.	10001111	.	00001110	.	10011011
129	.	143	.	14	.	155

32bit IPv4-Adressen

- IPv4-Adressen sind 32 bit lang \Rightarrow es gibt $2^{32} = 4\,294\,967\,296$ Adressen
- IPv4-Adressen werden in der **Dotted Decimal Notation** geschrieben:

129.143.14.155

- Die 32bit der Adresse werden in 4 Oktette geteilt jedes Oktett wird ins Dezimalsystem umgewandelt \rightarrow mathematisch unsinnige Schreibweise.
- Beispiel:

10000001	.	10001111	.	00001110	.	10011011
129	.	143	.	14	.	155

32bit IPv4-Adressen

- IPv4-Adresen sind 32 bit lang \Rightarrow es gibt $2^{32} = 4\,294\,967\,296$ Adressen
- IPv4-Adressen werden in der **Dotted Decimal Notation** geschrieben:

129.143.14.155

- Die 32bit der Adresse werden in 4 Oktette geteilt jedes Oktett wird ins Dezimalsystem umgewandelt \rightarrow mathematisch unsinnige Schreibweise.
- Beispiel:

10000001	.	10001111	.	00001110	.	10011011
129	.	143	.	14	.	155

32bit IPv4-Adressen

- IPv4-Adresen sind 32 bit lang \Rightarrow es gibt $2^{32} = 4\,294\,967\,296$ Adressen
- IPv4-Adressen werden in der **Dotted Decimal Notation** geschrieben:

129.143.14.155

- Die 32bit der Adresse werden in 4 Oktette geteilt jedes Oktett wird ins Dezimalsystem umgewandelt \rightarrow mathematisch unsinnige Schreibweise.
- Beispiel:

10000001	.	10001111	.	00001110	.	10011011
129	.	143	.	14	.	155

32bit IPv4-Adressen

- IPv4-Adressen sind 32 bit lang \Rightarrow es gibt $2^{32} = 4\,294\,967\,296$ Adressen
- IPv4-Adressen werden in der **Dotted Decimal Notation** geschrieben:

129.143.14.155

- Die 32bit der Adresse werden in 4 Oktette geteilt jedes Oktett wird ins Dezimalsystem umgewandelt \rightarrow mathematisch unsinnige Schreibweise.
- Beispiel:

10000001 . 10001111 . 00001110 . 10011011
129 . 143 . 14 . 155

32bit IPv4-Adressen

- IPv4-Adressen sind 32 bit lang \Rightarrow es gibt $2^{32} = 4\,294\,967\,296$ Adressen
- IPv4-Adressen werden in der **Dotted Decimal Notation** geschrieben:

129.143.14.155

- Die 32bit der Adresse werden in 4 Oktette geteilt jedes Oktett wird ins Dezimalsystem umgewandelt \rightarrow mathematisch unsinnige Schreibweise.
- Beispiel:

10000001 . 10001111 . 00001110 . 10011011
129 . 143 . 14 . 155

32bit IPv4-Adressen

- IPv4-Adresen sind 32 bit lang \Rightarrow es gibt $2^{32} = 4\,294\,967\,296$ Adressen
- IPv4-Adressen werden in der **Dotted Decimal Notation** geschrieben:

129.143.14.155

- Die 32bit der Adresse werden in 4 Oktette geteilt jedes Oktett wird ins Dezimalsystem umgewandelt \rightarrow mathematisch unsinnige Schreibweise.
- Beispiel:

10000001 . 10001111 . 00001110 . 10011011
129 . 143 . 14 . 155

32bit IPv4-Adressen

- IPv4-Adressen sind 32 bit lang \Rightarrow es gibt $2^{32} = 4\,294\,967\,296$ Adressen
- IPv4-Adressen werden in der **Dotted Decimal Notation** geschrieben:

129.143.14.155

- Die 32bit der Adresse werden in 4 Oktette geteilt jedes Oktett wird ins Dezimalsystem umgewandelt \rightarrow mathematisch unsinnige Schreibweise.
- Beispiel:

10000001 . 10001111 . 00001110 . 10011011
129 . 143 . 14 . 155

32bit IPv4-Adressen

- IPv4-Adresen sind 32 bit lang \Rightarrow es gibt $2^{32} = 4\,294\,967\,296$ Adressen
- IPv4-Adressen werden in der **Dotted Decimal Notation** geschrieben:

129.143.14.155

- Die 32bit der Adresse werden in 4 Oktette geteilt jedes Oktett wird ins Dezimalsystem umgewandelt \rightarrow mathematisch unsinnige Schreibweise.
- Beispiel:

10000001 . 10001111 . 00001110 . 10011011
129 . 143 . 14 . 155

32bit IPv4-Adressen

- IPv4-Adressen sind 32 bit lang \Rightarrow es gibt $2^{32} = 4\,294\,967\,296$ Adressen
- IPv4-Adressen werden in der **Dotted Decimal Notation** geschrieben:

129.143.14.155

- Die 32bit der Adresse werden in 4 Oktette geteilt jedes Oktett wird ins Dezimalsystem umgewandelt \rightarrow mathematisch unsinnige Schreibweise.
- Beispiel:

10000001 . 10001111 . 00001110 . 10011011
129 . 143 . 14 . 155

32bit IPv4-Adressen

- IPv4-Adressen sind 32 bit lang \Rightarrow es gibt $2^{32} = 4\,294\,967\,296$ Adressen
- IPv4-Adressen werden in der **Dotted Decimal Notation** geschrieben:

129.143.14.155

- Die 32bit der Adresse werden in 4 Oktette geteilt jedes Oktett wird ins Dezimalsystem umgewandelt \rightarrow mathematisch unsinnige Schreibweise.
- Beispiel:

10000001 . 10001111 . 00001110 . 10011011
129 . 143 . 14 . 155

32bit IPv4-Adressen

- IPv4-Adressen sind 32 bit lang \Rightarrow es gibt $2^{32} = 4\,294\,967\,296$ Adressen
- IPv4-Adressen werden in der **Dotted Decimal Notation** geschrieben:

129.143.14.155

- Die 32bit der Adresse werden in 4 Oktette geteilt jedes Oktett wird ins Dezimalsystem umgewandelt \rightarrow mathematisch unsinnige Schreibweise.
- Beispiel:

10000001 . 10001111 . 00001110 . 10011011
129 . 143 . 14 . 155

32bit IPv4-Adressen

- IPv4-Adresen sind 32 bit lang \Rightarrow es gibt $2^{32} = 4\,294\,967\,296$ Adressen
- IPv4-Adressen werden in der **Dotted Decimal Notation** geschrieben:

129.143.14.155

- Die 32bit der Adresse werden in 4 Oktette geteilt jedes Oktett wird ins Dezimalsystem umgewandelt \rightarrow mathematisch unsinnige Schreibweise.
- Beispiel:

10000001 . 10001111 . 00001110 . 10011011
129 . 143 . 14 . 155

32bit IPv4-Adressen

- IPv4-Adressen sind 32 bit lang \Rightarrow es gibt $2^{32} = 4\,294\,967\,296$ Adressen
- IPv4-Adressen werden in der **Dotted Decimal Notation** geschrieben:

129.143.14.155

- Die 32bit der Adresse werden in 4 Oktette geteilt jedes Oktett wird ins Dezimalsystem umgewandelt \rightarrow mathematisch unsinnige Schreibweise.
- Beispiel:

10000001 . 10001111 . 00001110 . 10011011
129 . 143 . 14 . 155

32bit IPv4-Adressen

- IPv4-Adressen sind 32 bit lang \Rightarrow es gibt $2^{32} = 4\,294\,967\,296$ Adressen
- IPv4-Adressen werden in der **Dotted Decimal Notation** geschrieben:

129.143.14.155

- Die 32bit der Adresse werden in 4 Oktette geteilt jedes Oktett wird ins Dezimalsystem umgewandelt \rightarrow mathematisch unsinnige Schreibweise.
- Beispiel:

10000001 . 10001111 . 00001110 . 10011011
129 . 143 . 14 . 155

32bit IPv4-Adressen

- IPv4-Adressen sind 32 bit lang \Rightarrow es gibt $2^{32} = 4\,294\,967\,296$ Adressen
- IPv4-Adressen werden in der **Dotted Decimal Notation** geschrieben:

129.143.14.155

- Die 32bit der Adresse werden in 4 Oktette geteilt jedes Oktett wird ins Dezimalsystem umgewandelt \rightarrow mathematisch unsinnige Schreibweise.
- Beispiel:

10000001 . 10001111 . 00001110 . 10011011
129 . 143 . 14 . 155

32bit IPv4-Adressen

- IPv4-Adresen sind 32 bit lang \Rightarrow es gibt $2^{32} = 4\,294\,967\,296$ Adressen
- IPv4-Adressen werden in der **Dotted Decimal Notation** geschrieben:

129.143.14.155

- Die 32bit der Adresse werden in 4 Oktette geteilt jedes Oktett wird ins Dezimalsystem umgewandelt \rightarrow mathematisch unsinnige Schreibweise.
- Beispiel:

10000001 . 10001111 . 00001110 . 10011011
129 . 143 . 14 . 155

32bit IPv4-Adressen

- IPv4-Adressen sind 32 bit lang \Rightarrow es gibt $2^{32} = 4\,294\,967\,296$ Adressen
- IPv4-Adressen werden in der **Dotted Decimal Notation** geschrieben:

129.143.14.155

- Die 32bit der Adresse werden in 4 Oktette geteilt jedes Oktett wird ins Dezimalsystem umgewandelt \rightarrow mathematisch unsinnige Schreibweise.
- Beispiel:

10000001 . 10001111 . 00001110 . 10011011
129 . 143 . 14 . 155

Bildung von Subnetzen

- 4 Milliarden Adressen ergeben ein zu grosses Netz: das Netz wird in kleinere **Subnetze** aufgeteilt.
- Die Trennung in Subnetze erfolgt ausschliesslich auf **logischer Ebene**:

31

0

1 0 0 0 0 0 0 1 1 0 0 0 1 1 1 1 0 0 0 0 1 1 1 0 1 0 0 1 1 0 1 1

Alle IP-Adressen, die im Netz-ID-Anteil übereinstimmen, gehören zum selben Subnetz.

Bildung von Subnetzen

- 4 Milliarden Adressen ergeben ein zu grosses Netz: das Netz wird in kleinere **Subnetze** aufgeteilt.
- Die Trennung in Subnetze erfolgt ausschliesslich auf **logischer Ebene**:

31

0

1 0 0 0 0 0 0 1 1 0 0 0 1 1 1 1 0 0 0 0 1 1 1 0 1 0 0 1 1 0 1 1

Alle IP-Adressen, die im Netz-ID-Anteil übereinstimmen, gehören zum selben Subnetz.

Bildung von Subnetzen

- 4 Milliarden Adressen ergeben ein zu grosses Netz: das Netz wird in kleinere **Subnetze** aufgeteilt.
- Die Trennung in Subnetze erfolgt ausschliesslich auf **logischer Ebene**:

31

0

1 0 0 0 0 0 0 1 1 0 0 0 1 1 1 1 0 0 0 0 1 1 1 0 1 0 0 1 1 0 1 1

Alle IP-Adressen, die im Netz-ID-Anteil übereinstimmen, gehören zum selben Subnetz.

Bildung von Subnetzen

- 4 Milliarden Adressen ergeben ein zu grosses Netz: das Netz wird in kleinere **Subnetze** aufgeteilt.
- Die Trennung in Subnetze erfolgt ausschliesslich auf **logischer Ebene**:

31

0

1 0 0 0 0 0 0 1 1 0 0 0 1 1 1 1 0 0 0 0 1 1 1 0 1 0 0 1 1 0 1 1

Alle IP-Adressen, die im Netz-ID-Anteil übereinstimmen, gehören zum selben Subnetz.

Bildung von Subnetzen

- 4 Milliarden Adressen ergeben ein zu grosses Netz: das Netz wird in kleinere **Subnetze** aufgeteilt.
- Die Trennung in Subnetze erfolgt ausschliesslich auf **logischer Ebene**:

31

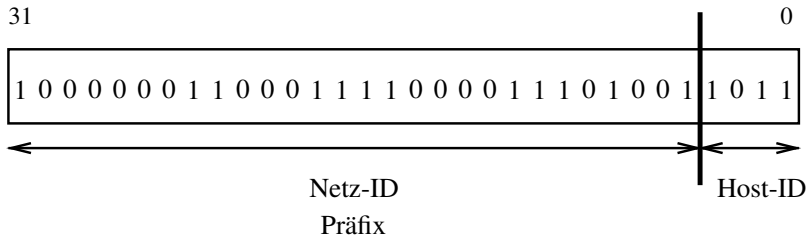
0

1 0 0 0 0 0 0 1 1 0 0 0 1 1 1 1 0 0 0 0 1 1 1 0 1 0 0 1 1 0 1 1

Alle IP-Adressen, die im Netz-ID-Anteil übereinstimmen, gehören zum selben Subnetz.

Bildung von Subnetzen

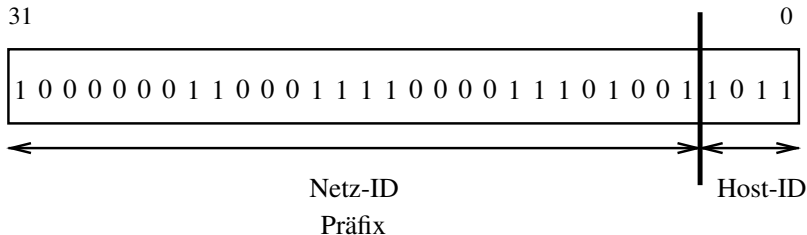
- 4 Milliarden Adressen ergeben ein zu grosses Netz: das Netz wird in kleinere **Subnetze** aufgeteilt.
- Die Trennung in Subnetze erfolgt ausschliesslich auf **logischer Ebene**:



Alle IP-Adressen, die im Netz-ID-Anteil übereinstimmen, gehören zum selben Subnetz.

Bildung von Subnetzen

- 4 Milliarden Adressen ergeben ein zu grosses Netz: das Netz wird in kleinere **Subnetze** aufgeteilt.
- Die Trennung in Subnetze erfolgt ausschliesslich auf **logischer Ebene**:



Alle IP-Adressen, die im Netz-ID-Anteil übereinstimmen, gehören zum selben Subnetz.

Subnetzmaske und Präfixlänge

- Es muss festgelegt werden, wo die Trennung zwischen Netz-ID und Host-ID verläuft.
- Zwei Möglichkeiten:
 1. Angabe einer Subnetzmaske (s.u.)
 2. Angabe der **Präfixlänge**:

10.16.0.0/12 \Rightarrow Netz-ID ist 12bit lang

Diese Schreibweise wird CIDR- oder VLSM Schreibweise genannt:

CIDR Classless Internet Domain Routing

VLSM Variable Length Subnet Mask

Subnetzmaske und Präfixlänge

- Es muss festgelegt werden, wo die Trennung zwischen Netz-ID und Host-ID verläuft.
- Zwei Möglichkeiten:
 1. Angabe einer Subnetzmaske (s.u.)
 2. Angabe der **Präfixlänge**:

10.16.0.0/12 \Rightarrow Netz-ID ist 12bit lang

Diese Schreibweise wird CIDR- oder VLSM Schreibweise genannt:

CIDR Classless Internet Domain Routing
VLSM Variable Length Subnet Mask

Subnetzmaske und Präfixlänge

- Es muss festgelegt werden, wo die Trennung zwischen Netz-ID und Host-ID verläuft.
- Zwei Möglichkeiten:
 1. Angabe einer Subnetzmaske (s.u.)
 2. Angabe der **Präfixlänge**:

10.16.0.0/12 \Rightarrow Netz-ID ist 12bit lang

Diese Schreibweise wird CIDR- oder VLSM Schreibweise genannt:

CIDR Classless Internet Domain Routing
VLSM Variable Length Subnet Mask

Subnetzmaske und Präfixlänge

- Es muss festgelegt werden, wo die Trennung zwischen Netz-ID und Host-ID verläuft.
- Zwei Möglichkeiten:
 1. Angabe einer Subnetzmaske (s.u.)
 2. Angabe der **Präfixlänge**:

10.16.0.0/12 \Rightarrow Netz-ID ist 12bit lang

Diese Schreibweise wird CIDR- oder VLSM Schreibweise genannt:

CIDR Classless Internet Domain Routing
VLSM Variable Length Subnet Mask

Subnetzmaske und Präfixlänge

- Es muss festgelegt werden, wo die Trennung zwischen Netz-ID und Host-ID verläuft.
- Zwei Möglichkeiten:
 1. Angabe einer Subnetzmaske (s.u.)
 2. Angabe der **Präfixlänge**:

10.16.0.0/12 \Rightarrow Netz-ID ist 12bit lang

Diese Schreibweise wird CIDR- oder VLSM Schreibweise genannt:

CIDR Classless Internet Domain Routing
VLSM Variable Length Subnet Mask

Subnetzmaske und Präfixlänge

- Es muss festgelegt werden, wo die Trennung zwischen Netz-ID und Host-ID verläuft.
- Zwei Möglichkeiten:
 1. Angabe einer Subnetzmaske (s.u.)
 2. Angabe der **Präfixlänge**:

10.16.0.0/12 \Rightarrow Netz-ID ist 12bit lang

Diese Schreibweise wird CIDR- oder VLSM Schreibweise genannt:

CIDR Classless Internet Domain Routing
VLSM Variable Length Subnet Mask

Subnetzmaske und Präfixlänge

- Es muss festgelegt werden, wo die Trennung zwischen Netz-ID und Host-ID verläuft.
- Zwei Möglichkeiten:
 1. Angabe einer Subnetzmaske (s.u.)
 2. Angabe der **Präfixlänge**:

10.16.0.0/12 \Rightarrow Netz-ID ist 12bit lang

Diese Schreibweise wird CIDR- oder VLSM Schreibweise genannt:

CIDR Classless Internet Domain Routing
VLSM Variable Length Subnet Mask

Subnetzmaske und Präfixlänge

- Es muss festgelegt werden, wo die Trennung zwischen Netz-ID und Host-ID verläuft.
- Zwei Möglichkeiten:
 1. Angabe einer Subnetzmaske (s.u.)
 2. Angabe der **Präfixlänge**:

10.16.0.0/12 \Rightarrow Netz-ID ist 12bit lang

Diese Schreibweise wird CIDR- oder VLSM Schreibweise genannt:

CIDR Classless Internet Domain Routing
VLSM Variable Length Subnet Mask

Subnetzmaske und Präfixlänge

- Es muss festgelegt werden, wo die Trennung zwischen Netz-ID und Host-ID verläuft.
- Zwei Möglichkeiten:
 1. Angabe einer Subnetzmaske (s.u.)
 2. Angabe der **Präfixlänge**:

10.16.0.0/12 \Rightarrow Netz-ID ist 12bit lang

Diese Schreibweise wird CIDR- oder VLSM Schreibweise genannt:

CIDR Classless Internet Domain Routing

VLSM Variable Length Subnet Mask

Subnetzmaske und Präfixlänge

- Es muss festgelegt werden, wo die Trennung zwischen Netz-ID und Host-ID verläuft.
- Zwei Möglichkeiten:
 1. Angabe einer Subnetzmaske (s.u.)
 2. Angabe der **Präfixlänge**:

10.16.0.0/12 \Rightarrow Netz-ID ist 12bit lang

Diese Schreibweise wird CIDR- oder VLSM Schreibweise genannt:

CIDR Classless Internet Domain Routing
VLSM Variable Length Subnet Mask

Details der Subnetzmaske

Die Subnetzmaske ist eine spezielle IP-Adresse, deren Netz-ID nur aus 1 und deren Host-ID nur aus 0 besteht:

Beispiel:

```
1111 1111 . 1111 1111 . 1111 1111 . 1111 0000
 255     .   255     .   255     .   240
```

Details der Subnetzmaske

Die Subnetzmaske ist eine spezielle IP-Adresse, deren Netz-ID nur aus 1 und deren Host-ID nur aus 0 besteht:

Beispiel:

```
1111 1111 . 1111 1111 . 1111 1111 . 1111 0000
 255     .   255     .   255     .   240
```

Details der Subnetzmaske

Die Subnetzmaske ist eine spezielle IP-Adresse, deren Netz-ID nur aus 1 und deren Host-ID nur aus 0 besteht:

Beispiel:

```
1111 1111 . 1111 1111 . 1111 1111 . 1111 0000  
255 . 255 . 255 . 240
```

Details der Subnetzmaske

Die Subnetzmaske ist eine spezielle IP-Adresse, deren Netz-ID nur aus 1 und deren Host-ID nur aus 0 besteht:

Beispiel:

```
1111 1111 . 1111 1111 . 1111 1111 . 1111 0000
 255     .   255     .   255     .   240
```

Details der Subnetzmaske

Die Subnetzmaske ist eine spezielle IP-Adresse, deren Netz-ID nur aus 1 und deren Host-ID nur aus 0 besteht:

Beispiel:

1111 1111 . 1111 1111 . 1111 1111 . 1111 0000
255 . 255 . 255 . 240

Details der Subnetzmaske

Die Subnetzmaske ist eine spezielle IP-Adresse, deren Netz-ID nur aus 1 und deren Host-ID nur aus 0 besteht:

Beispiel:

```
1111 1111 . 1111 1111 . 1111 1111 . 1111 0000
 255     .   255     .   255     .   240
```

Details der Subnetzmaske

Die Subnetzmaske ist eine spezielle IP-Adresse, deren Netz-ID nur aus 1 und deren Host-ID nur aus 0 besteht:

Beispiel:

```
1111 1111 . 1111 1111 . 1111 1111 . 1111 0000  
255     .   255     .   255     .   240
```

Details der Subnetzmaske

Die Subnetzmaske ist eine spezielle IP-Adresse, deren Netz-ID nur aus 1 und deren Host-ID nur aus 0 besteht:

Beispiel:

```
1111 1111 . 1111 1111 . 1111 1111 . 1111 0000
 255     .   255     .   255     .   240
```


Details der Subnetzmaske

Die Subnetzmaske ist eine spezielle IP-Adresse, deren Netz-ID nur aus 1 und deren Host-ID nur aus 0 besteht:

Beispiel:

```
1111 1111 . 1111 1111 . 1111 1111 . 1111 0000
 255     .   255     .   255     .   240
```

Details der Subnetzmaske

Die Subnetzmaske ist eine spezielle IP-Adresse, deren Netz-ID nur aus 1 und deren Host-ID nur aus 0 besteht:

Beispiel:

```
1111 1111 . 1111 1111 . 1111 1111 . 1111 0000
 255     .   255     .   255     .   240
```

Details der Subnetzmaske

Die Subnetzmaske ist eine spezielle IP-Adresse, deren Netz-ID nur aus 1 und deren Host-ID nur aus 0 besteht:

Beispiel:

```
1111 1111 . 1111 1111 . 1111 1111 . 1111 0000
 255     .   255     .   255     .   240
```

Details der Subnetzmaske

Die Subnetzmaske ist eine spezielle IP-Adresse, deren Netz-ID nur aus 1 und deren Host-ID nur aus 0 besteht:

Beispiel:

```
1111 1111 . 1111 1111 . 1111 1111 . 1111 0000  
255 . 255 . 255 . 240
```

Netz- und Broadcastadresse

- Bei der **Netzadresse** eines Subnetzes sind alle Host-ID-Bits = 0 \Rightarrow die Netzadresse ist die niedrigste Adresse in einem Subnetz
- Die Netzadresse wird für das Routing benötigt.
- Bei der **Broadcastadresse** eines Subnetzes, sind alle Host-ID-Bits = 1 \Rightarrow die Netzadresse ist die höchste Adresse in einem Subnetz
- Die IP-Broadcastadresse wird z.B. verwendet, um Drucker- und Dateifreigaben in einem Subnetz zu finden. Auch das Routing Protokoll RIPv1 arbeitet mit Broadcasts auf Layer3.

Netz- und Broadcastadresse

- Bei der **Netzadresse** eines Subnetzes sind alle Host-ID-Bits = 0 \Rightarrow die Netzadresse ist die niedrigste Adresse in einem Subnetz
- Die Netzadresse wird für das Routing benötigt.
- Bei der **Broadcastadresse** eines Subnetzes, sind alle Host-ID-Bits = 1 \Rightarrow die Netzadresse ist die höchste Adresse in einem Subnetz
- Die IP-Broadcastadresse wird z.B. verwendet, um Drucker- und Dateifreigaben in einem Subnetz zu finden. Auch das Routing Protokoll RIPv1 arbeitet mit Broadcasts auf Layer3.

Netz- und Broadcastadresse

- Bei der **Netzadresse** eines Subnetzes sind alle Host-ID-Bits = 0 \Rightarrow die Netzadresse ist die niedrigste Adresse in einem Subnetz
- Die Netzadresse wird für das Routing benötigt.
- Bei der **Broadcastadresse** eines Subnetzes, sind alle Host-ID-Bits = 1 \Rightarrow die Netzadresse ist die höchste Adresse in einem Subnetz
- Die IP-Broadcastadresse wird z.B. verwendet, um Drucker- und Dateifreigaben in einem Subnetz zu finden. Auch das Routing Protokoll RIPv1 arbeitet mit Broadcasts auf Layer3.

Netz- und Broadcastadresse

- Bei der **Netzadresse** eines Subnetzes sind alle Host-ID-Bits = 0 \Rightarrow die Netzadresse ist die niedrigste Adresse in einem Subnetz
- Die Netzadresse wird für das Routing benötigt.
- Bei der **Broadcastadresse** eines Subnetzes, sind alle Host-ID-Bits = 1 \Rightarrow die Netzadresse ist die höchste Adresse in einem Subnetz
- Die IP-Broadcastadresse wird z.B. verwendet, um Drucker- und Dateifreigaben in einem Subnetz zu finden. Auch das Routing Protokoll RIPv1 arbeitet mit Broadcasts auf Layer3.

Netz- und Broadcastadresse

- Bei der **Netzadresse** eines Subnetzes sind alle Host-ID-Bits = 0 \Rightarrow die Netzadresse ist die niedrigste Adresse in einem Subnetz
- Die Netzadresse wird für das Routing benötigt.
- Bei der **Broadcastadresse** eines Subnetzes, sind alle Host-ID-Bits = 1 \Rightarrow die Netzadresse ist die höchste Adresse in einem Subnetz
- Die IP-Broadcastadresse wird z.B. verwendet, um Drucker- und Dateifreigaben in einem Subnetz zu finden. Auch das Routing Protokoll RIPv1 arbeitet mit Broadcasts auf Layer3.

Berechnung der Netzadresse

Berechnung der Netzadresse:

Die Netzadresse ergibt sich aus der bitweisen
&-Verknüpfung einer IP-Adresse mit der
zugehörigen Netzmaske:

129	.	143	.	14	.	155
1000 0001	.	1000 1111	.	0000 1110	.	1001 1011
1111 1111	.	1111 1111	.	1111 1111	.	1111 0000
1000 0001	.	1000 1111	.	0000 1110	.	1001 0000
129	.	143	.	14	.	144

Berechnung der Netzadresse

Berechnung der Netzadresse:

Die Netzadresse ergibt sich aus der bitweisen
&-Verknüpfung einer IP-Adresse mit der
zugehörigen Netzmaske:

129	.	143	.	14	.	155
1000 0001	.	1000 1111	.	0000 1110	.	1001 1011
1111 1111	.	1111 1111	.	1111 1111	.	1111 0000
1000 0001	.	1000 1111	.	0000 1110	.	1001 0000
129	.	143	.	14	.	144

Berechnung der Netzadresse

Berechnung der Netzadresse:

Die Netzadresse ergibt sich aus der bitweisen
&-Verknüpfung einer IP-Adresse mit der
zugehörigen Netzmaske:

129	.	143	.	14	.	155
1000 0001	.	1000 1111	.	0000 1110	.	1001 1011
1111 1111	.	1111 1111	.	1111 1111	.	1111 0000
1000 0001	.	1000 1111	.	0000 1110	.	1001 0000
129	.	143	.	14	.	144

Berechnung der Netzadresse

Berechnung der Netzadresse:

Die Netzadresse ergibt sich aus der bitweisen
&-Verknüpfung einer IP-Adresse mit der
zugehörigen Netzmaske:

129	.	143	.	14	.	155
1000 0001	.	1000 1111	.	0000 1110	.	1001 1011
1111 1111	.	1111 1111	.	1111 1111	.	1111 0000
<hr/>						
1000 0001	.	1000 1111	.	0000 1110	.	1001 0000
129	.	143	.	14	.	144

Berechnung der Netzadresse

Berechnung der Netzadresse:

Die Netzadresse ergibt sich aus der bitweisen
&-Verknüpfung einer IP-Adresse mit der
zugehörigen Netzmaske:

129	.	143	.	14	.	155
1000 0001	.	1000 1111	.	0000 1110	.	1001 1011
1111 1111	.	1111 1111	.	1111 1111	.	1111 0000
1000 0001	.	1000 1111	.	0000 1110	.	1001 0000
129	.	143	.	14	.	144

Berechnung der Netzadresse

Berechnung der Netzadresse:

Die Netzadresse ergibt sich aus der bitweisen
&-Verknüpfung einer IP-Adresse mit der
zugehörigen Netzmaske:

129	.	143	.	14	.	155
1000 0001	.	1000 1111	.	0000 1110	.	1001 1011
1111 1111	.	1111 1111	.	1111 1111	.	1111 0000
1000 0001	.	1000 1111	.	0000 1110	.	1001 0000
129	.	143	.	14	.	144

Berechnung der Netzadresse

Berechnung der Netzadresse:

Die Netzadresse ergibt sich aus der bitweisen
&-Verknüpfung einer IP-Adresse mit der
zugehörigen Netzmaske:

129	.	143	.	14	.	155
1000 0001	.	1000 1111	.	0000 1110	.	1001 1011
1111 1111	.	1111 1111	.	1111 1111	.	1111 0000
1000 0001	.	1000 1111	.	0000 1110	.	1001 0000
129	.	143	.	14	.	144

Berechnung der Netzadresse

Berechnung der Netzadresse:

Die Netzadresse ergibt sich aus der bitweisen
&-Verknüpfung einer IP-Adresse mit der
zugehörigen Netzmaske:

129	.	143	.	14	.	155
1000 0001	.	1000 1111	.	0000 1110	.	1001 1011
1111 1111	.	1111 1111	.	1111 1111	.	1111 0000
1000 0001	.	1000 1111	.	0000 1110	.	1001 0000
129	.	143	.	14	.	144

Berechnung der Netzadresse

Berechnung der Netzadresse:

Die Netzadresse ergibt sich aus der bitweisen
&-Verknüpfung einer IP-Adresse mit der
zugehörigen Netzmaske:

129	.	143	.	14	.	155
1000 0001	.	1000 1111	.	0000 1110	.	1001 1011
1111 1111	.	1111 1111	.	1111 1111	.	1111 0000
1000 0001	.	1000 1111	.	0000 1110	.	1001 0000
129	.	143	.	14	.	144

Berechnung der Netzadresse

Berechnung der Netzadresse:

Die Netzadresse ergibt sich aus der bitweisen
&-Verknüpfung einer IP-Adresse mit der
zugehörigen Netzmaske:

129	.	143	.	14	.	155
1000 0001	.	1000 1111	.	0000 1110	.	1001 1011
1111 1111	.	1111 1111	.	1111 1111	.	1111 0000
1000 0001	.	1000 1111	.	0000 1110	.	1001 0000
129	.	143	.	14	.	144

Berechnung der Netzadresse

Berechnung der Netzadresse:

Die Netzadresse ergibt sich aus der bitweisen
&-Verknüpfung einer IP-Adresse mit der
zugehörigen Netzmaske:

129	.	143	.	14	.	155
1000 0001	.	1000 1111	.	0000 1110	.	1001 1011
1111 1111	.	1111 1111	.	1111 1111	.	1111 0000
1000 0001	.	1000 1111	.	0000 1110	.	1001 0000
129	.	143	.	14	.	144

Berechnung der Netzadresse

Berechnung der Netzadresse:

Die Netzadresse ergibt sich aus der bitweisen
&-Verknüpfung einer IP-Adresse mit der
zugehörigen Netzmaske:

129	.	143	.	14	.	155
1000 0001	.	1000 1111	.	0000 1110	.	1001 1011
1111 1111	.	1111 1111	.	1111 1111	.	1111 0000
1000 0001	.	1000 1111	.	0000 1110	.	1001 0000
129	.	143	.	14	.	144

Berechnung der Netzadresse

Berechnung der Netzadresse:

Die Netzadresse ergibt sich aus der bitweisen
&-Verknüpfung einer IP-Adresse mit der
zugehörigen Netzmaske:

129	.	143	.	14	.	155
1000 0001	.	1000 1111	.	0000 1110	.	1001 1011
1111 1111	.	1111 1111	.	1111 1111	.	1111 0000
1000 0001	.	1000 1111	.	0000 1110	.	1001 0000
129	.	143	.	14	.	144

Berechnung der Netzadresse

Berechnung der Netzadresse:

Die Netzadresse ergibt sich aus der bitweisen
&-Verknüpfung einer IP-Adresse mit der
zugehörigen Netzmaske:

129	.	143	.	14	.	155
1000 0001	.	1000 1111	.	0000 1110	.	1001 1011
1111 1111	.	1111 1111	.	1111 1111	.	1111 0000
1000 0001	.	1000 1111	.	0000 1110	.	1001 0000
129	.	143	.	14	.	144

Berechnung der Netzadresse

Berechnung der Netzadresse:

Die Netzadresse ergibt sich aus der bitweisen
&-Verknüpfung einer IP-Adresse mit der
zugehörigen Netzmaske:

129	.	143	.	14	.	155
1000 0001	.	1000 1111	.	0000 1110	.	1001 1011
1111 1111	.	1111 1111	.	1111 1111	.	1111 0000
1000 0001	.	1000 1111	.	0000 1110	.	1001 0000
129	.	143	.	14	.	144

Berechnung der Netzadresse

Berechnung der Netzadresse:

Die Netzadresse ergibt sich aus der bitweisen
&-Verknüpfung einer IP-Adresse mit der
zugehörigen Netzmaske:

129	.	143	.	14	.	155
1000 0001	.	1000 1111	.	0000 1110	.	1001 1011
1111 1111	.	1111 1111	.	1111 1111	.	1111 0000
1000 0001	.	1000 1111	.	0000 1110	.	1001 0000
129	.	143	.	14	.	144

Berechnung der Netzadresse

Berechnung der Netzadresse:

Die Netzadresse ergibt sich aus der bitweisen
&-Verknüpfung einer IP-Adresse mit der
zugehörigen Netzmaske:

129	.	143	.	14	.	155
1000 0001	.	1000 1111	.	0000 1110	.	1001 1011
1111 1111	.	1111 1111	.	1111 1111	.	1111 0000
1000 0001	.	1000 1111	.	0000 1110	.	1001 0000
129	.	143	.	14	.	144

Berechnung der Netzadresse

Berechnung der Netzadresse:

Die Netzadresse ergibt sich aus der bitweisen
&-Verknüpfung einer IP-Adresse mit der
zugehörigen Netzmaske:

129	.	143	.	14	.	155
1000 0001	.	1000 1111	.	0000 1110	.	1001 1011
1111 1111	.	1111 1111	.	1111 1111	.	1111 0000
1000 0001	.	1000 1111	.	0000 1110	.	1001 0000
129	.	143	.	14	.	144

Berechnung der Netzadresse

Berechnung der Netzadresse:

Die Netzadresse ergibt sich aus der bitweisen
&-Verknüpfung einer IP-Adresse mit der
zugehörigen Netzmaske:

129	.	143	.	14	.	155
1000 0001	.	1000 1111	.	0000 1110	.	1001 1011
1111 1111	.	1111 1111	.	1111 1111	.	1111 0000
1000 0001	.	1000 1111	.	0000 1110	.	1001 0000
129	.	143	.	14	.	144

Berechnung der Netzadresse

Berechnung der Netzadresse:

Die Netzadresse ergibt sich aus der bitweisen
&-Verknüpfung einer IP-Adresse mit der
zugehörigen Netzmaske:

129	.	143	.	14	.	155
1000 0001	.	1000 1111	.	0000 1110	.	1001 1011
1111 1111	.	1111 1111	.	1111 1111	.	1111 0000
1000 0001	.	1000 1111	.	0000 1110	.	1001 0000
129	.	143	.	14	.	144

Berechnung der Netzadresse

Berechnung der Netzadresse:

Die Netzadresse ergibt sich aus der bitweisen
&-Verknüpfung einer IP-Adresse mit der
zugehörigen Netzmaske:

129	.	143	.	14	.	155
1000 0001	.	1000 1111	.	0000 1110	.	1001 1011
1111 1111	.	1111 1111	.	1111 1111	.	1111 0000
1000 0001	.	1000 1111	.	0000 1110	.	1001 0000
129	.	143	.	14	.	144

Berechnung der Netzadresse

Berechnung der Netzadresse:

Die Netzadresse ergibt sich aus der bitweisen
&-Verknüpfung einer IP-Adresse mit der
zugehörigen Netzmaske:

129	.	143	.	14	.	155
1000 0001	.	1000 1111	.	0000 1110	.	1001 1011
1111 1111	.	1111 1111	.	1111 1111	.	1111 0000
1000 0001	.	1000 1111	.	0000 1110	.	1001 0000
129	.	143	.	14	.	144

Berechnung der Netzadresse

Berechnung der Netzadresse:

Die Netzadresse ergibt sich aus der bitweisen
&-Verknüpfung einer IP-Adresse mit der
zugehörigen Netzmaske:

129	.	143	.	14	.	155
1000 0001	.	1000 1111	.	0000 1110	.	1001 1011
1111 1111	.	1111 1111	.	1111 1111	.	1111 0000
1000 0001	.	1000 1111	.	0000 1110	.	1001 0000
129	.	143	.	14	.	144

Netzadressen

Selbstverständlich muss man auch bei einer Netzadresse angeben, wieviele Bits der Netz-ID-Anteil hat:

10.16.0.0/12 0000 1010 0001 0000 0000 0000 0000 0000 alle Host-ID-Bits sind 0 \Rightarrow Adresse ist eine Netzadresse

10.16.0.0/24 hier sind die letzten 8bit Host-ID, damit ist die Adresse ebenfalls eine Netzadresse

172.16.6.112/28 ebenfalls eine Netzadresse

172.16.6.112/27 keine Netzadresse

Netzadressen

Selbstverständlich muss man auch bei einer Netzadresse angeben, wieviele Bits der Netz-ID-Anteil hat:

10.16.0.0/12 0000 1010 0001 0000 0000 0000 0000 0000 alle Host-ID-Bits sind 0 \Rightarrow Adresse ist eine Netzadresse

10.16.0.0/24 hier sind die letzten 8bit Host-ID, damit ist die Adresse ebenfalls eine Netzadresse

172.16.6.112/28 ebenfalls eine Netzadresse

172.16.6.112/27 keine Netzadresse

Netzadressen

Selbstverständlich muss man auch bei einer Netzadresse angeben, wieviele Bits der Netz-ID-Anteil hat:

10.16.0.0/12 0000 1010 0001 0000 0000 0000 0000 0000 alle
Host-ID-Bits sind 0 \Rightarrow Adresse ist eine
Netzadresse

10.16.0.0/24 hier sind die letzten 8bit Host-ID, damit ist die
Adresse ebenfalls eine Netzadresse

172.16.6.112/28 ebenfalls eine Netzadresse

172.16.6.112/27 keine Netzadresse

Netzadressen

Selbstverständlich muss man auch bei einer Netzadresse angeben, wieviele Bits der Netz-ID-Anteil hat:

10.16.0.0/12 0000 1010 0001 0000 0000 0000 0000 0000 alle Host-ID-Bits sind 0 \Rightarrow Adresse ist eine Netzadresse

10.16.0.0/24 hier sind die letzten 8bit Host-ID, damit ist die Adresse ebenfalls eine Netzadresse

172.16.6.112/28 ebenfalls eine Netzadresse

172.16.6.112/27 keine Netzadresse

Netzadressen

Selbstverständlich muss man auch bei einer Netzadresse angeben, wieviele Bits der Netz-ID-Anteil hat:

10.16.0.0/12 0000 1010 0001 0000 0000 0000 0000 0000 alle Host-ID-Bits sind 0 \Rightarrow Adresse ist eine Netzadresse

10.16.0.0/24 hier sind die letzten 8bit Host-ID, damit ist die Adresse ebenfalls eine Netzadresse

172.16.6.112/28 ebenfalls eine Netzadresse

172.16.6.112/27 keine Netzadresse

Netzadressen

Selbstverständlich muss man auch bei einer Netzadresse angeben, wieviele Bits der Netz-ID-Anteil hat:

10.16.0.0/12 0000 1010 0001 0000 0000 0000 0000 0000 alle Host-ID-Bits sind 0 \Rightarrow Adresse ist eine Netzadresse

10.16.0.0/24 hier sind die letzten 8bit Host-ID, damit ist die Adresse ebenfalls eine Netzadresse

172.16.6.112/28 ebenfalls eine Netzadresse

172.16.6.112/27 keine Netzadresse

Netzadressen

Selbstverständlich muss man auch bei einer Netzadresse angeben, wieviele Bits der Netz-ID-Anteil hat:

10.16.0.0/12 0000 1010 0001 0000 0000 0000 0000 0000 alle Host-ID-Bits sind 0 \Rightarrow Adresse ist eine Netzadresse

10.16.0.0/24 hier sind die letzten 8bit Host-ID, damit ist die Adresse ebenfalls eine Netzadresse

172.16.6.112/28 ebenfalls eine Netzadresse

172.16.6.112/27 keine Netzadresse

Regeln zu Netz- und Broadcastadressen

Ein paar wichtige Regeln:

- Die Grösse (Anzahl Adressen) eines Netzes ist immer einer Potenz von 2.
- Eine Netzadresse muss immer durch die Grösse ihres Netzes teilbar sein.
- Netze mit gleicher Maske / Präfixlänge sind gleich gross
- Die Netzadresse ist die niedrigste Adresse in einem Subnetz
- Die Broadcastadresse ist die höchste Adresse in einem Subnetz

Regeln zu Netz- und Broadcastadressen

Ein paar wichtige Regeln:

- Die Grösse (Anzahl Adressen) eines Netzes ist immer einer Potenz von 2.
- Eine Netzadresse muss immer durch die Grösse ihres Netzes teilbar sein.
- Netze mit gleicher Maske / Präfixlänge sind gleich gross
- Die Netzadresse ist die niedrigste Adresse in einem Subnetz
- Die Broadcastadresse ist die höchste Adresse in einem Subnetz

Regeln zu Netz- und Broadcastadressen

Ein paar wichtige Regeln:

- Die Grösse (Anzahl Adressen) eines Netzes ist immer einer Potenz von 2.
- Eine Netzadresse muss immer durch die Grösse ihres Netzes teilbar sein.
- Netze mit gleicher Maske / Präfixlänge sind gleich gross
- Die Netzadresse ist die niedrigste Adresse in einem Subnetz
- Die Broadcastadresse ist die höchste Adresse in einem Subnetz

Regeln zu Netz- und Broadcastadressen

Ein paar wichtige Regeln:

- Die Grösse (Anzahl Adressen) eines Netzes ist immer einer Potenz von 2.
- Eine Netzadresse muss immer durch die Grösse ihres Netzes teilbar sein.
- Netze mit gleicher Maske / Präfixlänge sind gleich gross
- Die Netzadresse ist die niedrigste Adresse in einem Subnetz
- Die Broadcastadresse ist die höchste Adresse in einem Subnetz

Regeln zu Netz- und Broadcastadressen

Ein paar wichtige Regeln:

- Die Grösse (Anzahl Adressen) eines Netzes ist immer einer Potenz von 2.
- Eine Netzadresse muss immer durch die Grösse ihres Netzes teilbar sein.
- Netze mit gleicher Maske / Präfixlänge sind gleich gross
- Die Netzadresse ist die niedrigste Adresse in einem Subnetz
- Die Broadcastadresse ist die höchste Adresse in einem Subnetz

Regeln zu Netz- und Broadcastadressen

Ein paar wichtige Regeln:

- Die Grösse (Anzahl Adressen) eines Netzes ist immer einer Potenz von 2.
- Eine Netzadresse muss immer durch die Grösse ihres Netzes teilbar sein.
- Netze mit gleicher Maske / Präfixlänge sind gleich gross
- Die Netzadresse ist die niedrigste Adresse in einem Subnetz
- Die Broadcastadresse ist die höchste Adresse in einem Subnetz

Inhalt

Vom LAN zum Internet

Rahmen, Pakete, virtuelle Verbindungen

Versuch: Daten mit nc übertragen und Datenverkehr mitschneiden

IP-Adressen und Subnetze

Address Resolution Protocol

Routing

Verbindung zwischen Schicht2 und Schicht3

Problem: IP-Adresse ist **hardwareunabhängig** ⇒
IP-Adresse kann **nicht** an eine bestimmte
MAC-Adresse gebunden sein.
Beispiel: Austausch der Netzwerkkarte:
MAC-Adresse ändert sich, IP-Adresse bleibt

Lösung: Address Resolution Protocol, ARP

Funktion: HostA möchte IP-Paket an HostB senden, kennt
aber nur dessen IP-Adresse.
HostA sendet Layer2-Broadcast: "Wer hat die IP
10.20.23.100?"
10.20.23.100 antwortet mit einem Paket, in dem
die gesuchte MAC-Adresse enthalten ist.

Verbindung zwischen Schicht2 und Schicht3

Problem: IP-Adresse ist **hardwareunabhängig** ⇒

IP-Adresse kann **nicht** an eine bestimmte
MAC-Adresse gebunden sein.

Beispiel: Austausch der Netzwerkkarte:

MAC-Adresse ändert sich, IP-Adresse bleibt

Lösung: Address Resolution Protocol, ARP

Funktion: HostA möchte IP-Paket an HostB senden, kennt
aber nur dessen IP-Adresse.

HostA sendet Layer2-Broadcast: "Wer hat die IP
10.20.23.100?"

10.20.23.100 antwortet mit einem Paket, in dem
die gesuchte MAC-Adresse enthalten ist.

Verbindung zwischen Schicht2 und Schicht3

Problem: IP-Adresse ist **hardwareunabhängig** ⇒
IP-Adresse kann **nicht** an eine bestimmte
MAC-Adresse gebunden sein.

Beispiel: Austausch der Netzwerkkarte:
MAC-Adresse ändert sich, IP-Adresse bleibt

Lösung: Address Resolution Protocol, ARP

Funktion: HostA möchte IP-Paket an HostB senden, kennt
aber nur dessen IP-Adresse.

HostA sendet Layer2-Broadcast: "Wer hat die IP
10.20.23.100?"

10.20.23.100 antwortet mit einem Paket, in dem
die gesuchte MAC-Adresse enthalten ist.

Verbindung zwischen Schicht2 und Schicht3

Problem: IP-Adresse ist **hardwareunabhängig** ⇒
IP-Adresse kann **nicht** an eine bestimmte
MAC-Adresse gebunden sein.

Beispiel: Austausch der Netzwerkkarte:

MAC-Adresse ändert sich, IP-Adresse bleibt

Lösung: Address Resolution Protocol, ARP

Funktion: HostA möchte IP-Paket an HostB senden, kennt
aber nur dessen IP-Adresse.

HostA sendet Layer2-Broadcast: "Wer hat die IP
10.20.23.100?"

10.20.23.100 antwortet mit einem Paket, in dem
die gesuchte MAC-Adresse enthalten ist.

Verbindung zwischen Schicht2 und Schicht3

Problem: IP-Adresse ist **hardwareunabhängig** ⇒
IP-Adresse kann **nicht** an eine bestimmte
MAC-Adresse gebunden sein.
Beispiel: Austausch der Netzwerkkarte:
MAC-Adresse ändert sich, IP-Adresse bleibt

Lösung: Address Resolution Protocol, ARP

Funktion: HostA möchte IP-Paket an HostB senden, kennt
aber nur dessen IP-Adresse.
HostA sendet Layer2-Broadcast: "Wer hat die IP
10.20.23.100?"
10.20.23.100 antwortet mit einem Paket, in dem
die gesuchte MAC-Adresse enthalten ist.

Verbindung zwischen Schicht2 und Schicht3

Problem: IP-Adresse ist **hardwareunabhängig** ⇒
IP-Adresse kann **nicht** an eine bestimmte
MAC-Adresse gebunden sein.
Beispiel: Austausch der Netzwerkkarte:
MAC-Adresse ändert sich, IP-Adresse bleibt

Lösung: Address Resolution Protocol, ARP

Funktion: HostA möchte IP-Paket an HostB senden, kennt
aber nur dessen IP-Adresse.
HostA sendet Layer2-Broadcast: "Wer hat die IP
10.20.23.100?"
10.20.23.100 antwortet mit einem Paket, in dem
die gesuchte MAC-Adresse enthalten ist.

Verbindung zwischen Schicht2 und Schicht3

Problem: IP-Adresse ist **hardwareunabhängig** ⇒
IP-Adresse kann **nicht** an eine bestimmte
MAC-Adresse gebunden sein.
Beispiel: Austausch der Netzwerkkarte:
MAC-Adresse ändert sich, IP-Adresse bleibt

Lösung: Address Resolution Protocol, ARP

Funktion: HostA möchte IP-Paket an HostB senden, kennt
aber nur dessen IP-Adresse.
HostA sendet Layer2-Broadcast: "Wer hat die IP
10.20.23.100?"
10.20.23.100 antwortet mit einem Paket, in dem
die gesuchte MAC-Adresse enthalten ist.

Verbindung zwischen Schicht2 und Schicht3

Problem: IP-Adresse ist **hardwareunabhängig** ⇒
IP-Adresse kann **nicht** an eine bestimmte
MAC-Adresse gebunden sein.
Beispiel: Austausch der Netzwerkkarte:
MAC-Adresse ändert sich, IP-Adresse bleibt

Lösung: Address Resolution Protocol, ARP

Funktion: HostA möchte IP-Paket an HostB senden, kennt
aber nur dessen IP-Adresse.

HostA sendet Layer2-Broadcast: "Wer hat die IP
10.20.23.100?"

10.20.23.100 antwortet mit einem Paket, in dem
die gesuchte MAC-Adresse enthalten ist.

Verbindung zwischen Schicht2 und Schicht3

Problem: IP-Adresse ist **hardwareunabhängig** ⇒
IP-Adresse kann **nicht** an eine bestimmte
MAC-Adresse gebunden sein.
Beispiel: Austausch der Netzwerkkarte:
MAC-Adresse ändert sich, IP-Adresse bleibt

Lösung: Address Resolution Protocol, ARP

Funktion: HostA möchte IP-Paket an HostB senden, kennt
aber nur dessen IP-Adresse.
HostA sendet Layer2-Broadcast: "Wer hat die IP
10.20.23.100?"
10.20.23.100 antwortet mit einem Paket, in dem
die gesuchte MAC-Adresse enthalten ist.

Verbindung zwischen Schicht2 und Schicht3

Problem: IP-Adresse ist **hardwareunabhängig** ⇒
IP-Adresse kann **nicht** an eine bestimmte
MAC-Adresse gebunden sein.
Beispiel: Austausch der Netzwerkkarte:
MAC-Adresse ändert sich, IP-Adresse bleibt

Lösung: Address Resolution Protocol, ARP

Funktion: HostA möchte IP-Paket an HostB senden, kennt
aber nur dessen IP-Adresse.
HostA sendet Layer2-Broadcast: "Wer hat die IP
10.20.23.100?"
10.20.23.100 antwortet mit einem Paket, in dem
die gesuchte MAC-Adresse enthalten ist.

Inhalt

Vom LAN zum Internet

Rahmen, Pakete, virtuelle Verbindungen

Versuch: Daten mit nc übertragen und Datenverkehr mitschneiden

IP-Adressen und Subnetze

Address Resolution Protocol

Routing

Router und Routen

Router Router sind **Computer** (mit CPU, RAM, ROM, OS), die IP-Pakete anhand der Layer3-Adresse (IP-Adresse) zwischen verschiedenen IP-Netzen weiterleiten → Packet-Forwarding.

Weltweit erster Router für ARPANET: Honeywell 316 Minicomputer.

Beginn des ARPANET: 30. August 1969

Packet-Forwarding Der Router entscheidet, in welches Netz er ankommende Daten weiterleiten soll anhand der Layer 3 IP-Zieladresse der ankommenden IP-Pakete.

Routen 3 Möglichkeiten:

- Route = Zielnetz + IP-Adr. Next Hop (GW)
- Route = Zielnetz + Exit Interface
- Route = eigenes Netz

Router und Routen

Router Router sind **Computer** (mit CPU, RAM, ROM, OS), die IP-Pakete anhand der Layer3-Adresse (IP-Adresse) zwischen verschiedenen IP-Netzen weiterleiten → Packet-Forwarding.
Weltweit erster Router für ARPANET: Honeywell 316 Minicomputer.
Beginn des ARPANET: 30. August 1969

Packet-Forwarding Der Router entscheidet, in welches Netz er ankommende Daten weiterleiten soll anhand der Layer 3 IP-Zieladresse der ankommenden IP-Pakete.

Routen 3 Möglichkeiten:

- Route = Zielnetz + IP-Adr. Next Hop (GW)
- Route = Zielnetz + Exit Interface
- Route = eigenes Netz

Router und Routen

Router Router sind **Computer** (mit CPU, RAM, ROM, OS), die IP-Pakete anhand der Layer3-Adresse (IP-Adresse) zwischen verschiedenen IP-Netzen weiterleiten → Packet-Forwarding.

Weltweit erster Router für ARPANET: Honeywell 316 Minicomputer.

Beginn des ARPANET: 30. August 1969

Packet-Forwarding Der Router entscheidet, in welches Netz er ankommende Daten weiterleiten soll anhand der Layer 3 IP-Zieladresse der ankommenden IP-Pakete.

Routen 3 Möglichkeiten:

- Route = Zielnetz + IP-Adr. Next Hop (GW)
- Route = Zielnetz + Exit Interface
- Route = eigenes Netz

Router und Routen

Router Router sind **Computer** (mit CPU, RAM, ROM, OS), die IP-Pakete anhand der Layer3-Adresse (IP-Adresse) zwischen verschiedenen IP-Netzen weiterleiten → Packet-Forwarding.

Weltweit erster Router für ARPANET: Honeywell 316 Minicomputer.

Beginn des ARPANET: 30. August 1969

Packet-Forwarding Der Router entscheidet, in welches Netz er ankommende Daten weiterleiten soll anhand der Layer 3 IP-Zieladresse der ankommenden IP-Pakete.

Routen 3 Möglichkeiten:

- Route = Zielnetz + IP-Adr. Next Hop (GW)
- Route = Zielnetz + Exit Interface
- Route = eigenes Netz

Router und Routen

Router Router sind **Computer** (mit CPU, RAM, ROM, OS), die IP-Pakete anhand der Layer3-Adresse (IP-Adresse) zwischen verschiedenen IP-Netzen weiterleiten → Packet-Forwarding.

Weltweit erster Router für ARPANET: Honeywell 316 Minicomputer.

Beginn des ARPANET: 30. August 1969

Packet-Forwarding Der Router entscheidet, in welches Netz er ankommende Daten weiterleiten soll anhand der Layer 3 IP-Zieladresse der ankommenden IP-Pakete.

Routen 3 Möglichkeiten:

- Route = Zielnetz + IP-Adr. Next Hop (GW)
- Route = Zielnetz + Exit Interface
- Route = eigenes Netz

Router und Routen

Router Router sind **Computer** (mit CPU, RAM, ROM, OS), die IP-Pakete anhand der Layer3-Adresse (IP-Adresse) zwischen verschiedenen IP-Netzen weiterleiten → Packet-Forwarding.
Weltweit erster Router für ARPANET: Honeywell 316 Minicomputer.

Beginn des ARPANET: 30. August 1969

Packet-Forwarding Der Router entscheidet, in welches Netz er ankommende Daten weiterleiten soll anhand der Layer 3 IP-Zieladresse der ankommenden IP-Pakete.

Routen 3 Möglichkeiten:

- Route = Zielnetz + IP-Adr. Next Hop (GW)
- Route = Zielnetz + Exit Interface
- Route = eigenes Netz

Router und Routen

Router Router sind **Computer** (mit CPU, RAM, ROM, OS), die IP-Pakete anhand der Layer3-Adresse (IP-Adresse) zwischen verschiedenen IP-Netzen weiterleiten → Packet-Forwarding.
Weltweit erster Router für ARPANET: Honeywell 316 Minicomputer.
Beginn des ARPANET: 30. August 1969

Packet-Forwarding Der Router entscheidet, in welches Netz er ankommende Daten weiterleiten soll anhand der Layer 3 IP-Zieladresse der ankommenden IP-Pakete.

Routen 3 Möglichkeiten:

- Route = Zielnetz + IP-Adr. Next Hop (GW)
- Route = Zielnetz + Exit Interface
- Route = eigenes Netz

Router und Routen

Router Router sind **Computer** (mit CPU, RAM, ROM, OS), die IP-Pakete anhand der Layer3-Adresse (IP-Adresse) zwischen verschiedenen IP-Netzen weiterleiten → Packet-Forwarding.
Weltweit erster Router für ARPANET: Honeywell 316 Minicomputer.
Beginn des ARPANET: 30. August 1969

Packet-Forwarding Der Router entscheidet, in welches Netz er ankommende Daten weiterleiten soll anhand der Layer 3 IP-Zieladresse der ankommenden IP-Pakete.

Routen 3 Möglichkeiten:

- Route = Zielnetz + IP-Adr. Next Hop (GW)
- Route = Zielnetz + Exit Interface
- Route = eigenes Netz

Router und Routen

Router Router sind **Computer** (mit CPU, RAM, ROM, OS), die IP-Pakete anhand der Layer3-Adresse (IP-Adresse) zwischen verschiedenen IP-Netzen weiterleiten → Packet-Forwarding.
Weltweit erster Router für ARPANET: Honeywell 316 Minicomputer.
Beginn des ARPANET: 30. August 1969

Packet-Forwarding Der Router entscheidet, in welches Netz er ankommende Daten weiterleiten soll anhand der Layer 3 IP-Zieladresse der ankommenden IP-Pakete.

Routen 3 Möglichkeiten:

- Route = Zielnetz + IP-Adr. Next Hop (GW)
- Route = Zielnetz + Exit Interface
- Route = eigenes Netz

Router und Routen

Router Router sind **Computer** (mit CPU, RAM, ROM, OS), die IP-Pakete anhand der Layer3-Adresse (IP-Adresse) zwischen verschiedenen IP-Netzen weiterleiten → Packet-Forwarding.

Weltweit erster Router für ARPANET: Honeywell 316 Minicomputer.

Beginn des ARPANET: 30. August 1969

Packet-Forwarding Der Router entscheidet, in welches Netz er ankommende Daten weiterleiten soll anhand der Layer 3 IP-Zieladresse der ankommenden IP-Pakete.

Routen 3 Möglichkeiten:

- Route = Zielnetz + IP-Adr. Next Hop (GW)
- Route = Zielnetz + Exit Interface
- Route = eigenes Netz

Router und Routen

Router Router sind **Computer** (mit CPU, RAM, ROM, OS), die IP-Pakete anhand der Layer3-Adresse (IP-Adresse) zwischen verschiedenen IP-Netzen weiterleiten → Packet-Forwarding.

Weltweit erster Router für ARPANET: Honeywell 316 Minicomputer.

Beginn des ARPANET: 30. August 1969

Packet-Forwarding Der Router entscheidet, in welches Netz er ankommende Daten weiterleiten soll anhand der Layer 3 IP-Zieladresse der ankommenden IP-Pakete.

Routen 3 Möglichkeiten:

- Route = Zielnetz + IP-Adr. Next Hop (GW)
- Route = Zielnetz + Exit Interface
- Route = eigenes Netz

Router und Routen

Router Router sind **Computer** (mit CPU, RAM, ROM, OS), die IP-Pakete anhand der Layer3-Adresse (IP-Adresse) zwischen verschiedenen IP-Netzen weiterleiten → Packet-Forwarding.
Weltweit erster Router für ARPANET: Honeywell 316 Minicomputer.
Beginn des ARPANET: 30. August 1969

Packet-Forwarding Der Router entscheidet, in welches Netz er ankommende Daten weiterleiten soll anhand der Layer 3 IP-Zieladresse der ankommenden IP-Pakete.

Routen 3 Möglichkeiten:

- Route = Zielnetz + IP-Adr. Next Hop (GW)
- Route = Zielnetz + Exit Interface
- Route = eigenes Netz

Router und Routen

Router Router sind **Computer** (mit CPU, RAM, ROM, OS), die IP-Pakete anhand der Layer3-Adresse (IP-Adresse) zwischen verschiedenen IP-Netzen weiterleiten → Packet-Forwarding.
Weltweit erster Router für ARPANET: Honeywell 316 Minicomputer.
Beginn des ARPANET: 30. August 1969

Packet-Forwarding Der Router entscheidet, in welches Netz er ankommende Daten weiterleiten soll anhand der Layer 3 IP-Zieladresse der ankommenden IP-Pakete.

Routen 3 Möglichkeiten:

- Route = Zielnetz + IP-Adr. Next Hop (GW)
- Route = Zielnetz + Exit Interface
- Route = eigenes Netz

Router und Routen

Router Router sind **Computer** (mit CPU, RAM, ROM, OS), die IP-Pakete anhand der Layer3-Adresse (IP-Adresse) zwischen verschiedenen IP-Netzen weiterleiten → Packet-Forwarding.
Weltweit erster Router für ARPANET: Honeywell 316 Minicomputer.
Beginn des ARPANET: 30. August 1969

Packet-Forwarding Der Router entscheidet, in welches Netz er ankommende Daten weiterleiten soll anhand der Layer 3 IP-Zieladresse der ankommenden IP-Pakete.

Routen 3 Möglichkeiten:

- Route = Zielnetz + IP-Adr. Next Hop (GW)
- Route = Zielnetz + Exit Interface
- Route = eigenes Netz

Router und Routen

Router Router sind **Computer** (mit CPU, RAM, ROM, OS), die IP-Pakete anhand der Layer3-Adresse (IP-Adresse) zwischen verschiedenen IP-Netzen weiterleiten → Packet-Forwarding.
Weltweit erster Router für ARPANET: Honeywell 316 Minicomputer.
Beginn des ARPANET: 30. August 1969

Packet-Forwarding Der Router entscheidet, in welches Netz er ankommende Daten weiterleiten soll anhand der Layer 3 IP-Zieladresse der ankommenden IP-Pakete.

Routen 3 Möglichkeiten:

- Route = Zielnetz + IP-Adr. Next Hop (GW)
- Route = Zielnetz + Exit Interface
- Route = eigenes Netz

Routingentscheidung, Routingtabelle und Standardgateway

Routingentscheidung Router vergleichen die Zielnetze in der Routingtabelle mit den Zielnetzen der ankommenden Pakete.

Die Präfixlänge gibt an wieviele Bit von vorne gezählt übereinstimmen müssen

Bei mehreren Übereinstimmungen: **best match** (grösste Anzahl passender Bits)

Routingtabelle Liste mit Routen

Default Gateway Das Default Gateway ist ein Eintrag in der Routingtabelle. Besonderheit: das Zielnetz hat die spezielle IP **0.0.0.0/0**

0.0.0.0/0 passt zu **jeder** IP-Adresse: 0 von 0 Bit müssen übereinstimmen

Routingentscheidung, Routingtabelle und Standardgateway

Routingentscheidung Router vergleichen die Zielnetze in der Routingtabelle mit den Zielnetzen der ankommenden Pakete.

Die Präfixlänge gibt an wieviele Bit von vorne gezählt übereinstimmen müssen

Bei mehreren Übereinstimmungen: **best match** (grösste Anzahl passender Bits)

Routingtabelle Liste mit Routen

Default Gateway Das Default Gateway ist ein Eintrag in der Routingtabelle. Besonderheit: das Zielnetz hat die spezielle IP **0.0.0.0/0**

0.0.0.0/0 passt zu **jeder** IP-Adresse: 0 von 0 Bit müssen übereinstimmen

Routingentscheidung, Routingtabelle und Standardgateway

Routingentscheidung Router vergleichen die Zielnetze in der Routingtabelle mit den Zielnetzen der ankommenden Pakete.

Die Präfixlänge gibt an wieviele Bit von vorne gezählt übereinstimmen müssen

Bei mehreren Übereinstimmungen: **best match** (grösste Anzahl passender Bits)

Routingtabelle Liste mit Routen

Default Gateway Das Default Gateway ist ein Eintrag in der Routingtabelle. Besonderheit: das Zielnetz hat die spezielle IP **0.0.0.0/0**

0.0.0.0/0 passt zu **jeder** IP-Adresse: 0 von 0 Bit müssen übereinstimmen

Routingentscheidung, Routingtabelle und Standardgateway

Routingentscheidung Router vergleichen die Zielnetze in der Routingtabelle mit den Zielnetzen der ankommenden Pakete.

Die Präfixlänge gibt an wieviele Bit von vorne gezählt übereinstimmen müssen

Bei mehreren Übereinstimmungen: **best match** (grösste Anzahl passender Bits)

Routingtabelle Liste mit Routen

Default Gateway Das Default Gateway ist ein Eintrag in der Routingtabelle. Besonderheit: das Zielnetz hat die spezielle IP **0.0.0.0/0**

0.0.0.0/0 passt zu **jeder** IP-Adresse: 0 von 0 Bit müssen übereinstimmen

Routingentscheidung, Routingtabelle und Standardgateway

Routingentscheidung Router vergleichen die Zielnetze in der Routingtabelle mit den Zielnetzen der ankommenden Pakete.

Die Präfixlänge gibt an wieviele Bit von vorne gezählt übereinstimmen müssen

Bei mehreren Übereinstimmungen: **best match** (grösste Anzahl passender Bits)

Routingtabelle Liste mit Routen

Default Gateway Das Default Gateway ist ein Eintrag in der Routingtabelle. Besonderheit: das Zielnetz hat die spezielle IP **0.0.0.0/0**

0.0.0.0/0 passt zu **jeder** IP-Adresse: 0 von 0 Bit müssen übereinstimmen

Routingentscheidung, Routingtabelle und Standardgateway

Routingentscheidung Router vergleichen die Zielnetze in der Routingtabelle mit den Zielnetzen der ankommenden Pakete.

Die Präfixlänge gibt an wieviele Bit von vorne gezählt übereinstimmen müssen

Bei mehreren Übereinstimmungen: **best match** (grösste Anzahl passender Bits)

Routingtabelle Liste mit Routen

Default Gateway Das Default Gateway ist ein Eintrag in der Routingtabelle. Besonderheit: das Zielnetz hat die spezielle IP **0.0.0.0/0**

0.0.0.0/0 passt zu **jeder** IP-Adresse: 0 von 0 Bit müssen übereinstimmen

Routingentscheidung, Routingtabelle und Standardgateway

Routingentscheidung Router vergleichen die Zielnetze in der Routingtabelle mit den Zielnetzen der ankommenden Pakete.

Die Präfixlänge gibt an wieviele Bit von vorne gezählt übereinstimmen müssen

Bei mehreren Übereinstimmungen: **best match** (grösste Anzahl passender Bits)

Routingtabelle Liste mit Routen

Default Gateway Das Default Gateway ist ein Eintrag in der Routingtabelle. Besonderheit: das Zielnetz hat die spezielle IP **0.0.0.0/0**

0.0.0.0/0 passt zu **jeder** IP-Adresse: 0 von 0 Bit müssen übereinstimmen

Routingentscheidung, Routingtabelle und Standardgateway

Routingentscheidung Router vergleichen die Zielnetze in der Routingtabelle mit den Zielnetzen der ankommenden Pakete.

Die Präfixlänge gibt an wieviele Bit von vorne gezählt übereinstimmen müssen

Bei mehreren Übereinstimmungen: **best match** (grösste Anzahl passender Bits)

Routingtabelle Liste mit Routen

Default Gateway Das Default Gateway ist ein Eintrag in der Routingtabelle. Besonderheit: das Zielnetz hat die spezielle IP **0.0.0.0/0**

0.0.0.0/0 passt zu **jeder** IP-Adresse: 0 von 0 Bit müssen übereinstimmen

Routingentscheidung, Routingtabelle und Standardgateway

Routingentscheidung Router vergleichen die Zielnetze in der Routingtabelle mit den Zielnetzen der ankommenden Pakete.

Die Präfixlänge gibt an wieviele Bit von vorne gezählt übereinstimmen müssen

Bei mehreren Übereinstimmungen: **best match** (grösste Anzahl passender Bits)

Routingtabelle Liste mit Routen

Default Gateway Das Default Gateway ist ein Eintrag in der Routingtabelle. Besonderheit: das Zielnetz hat die spezielle IP **0.0.0.0/0**
0.0.0.0/0 passt zu **jeder** IP-Adresse: 0 von 0 Bit müssen übereinstimmen

Routingentscheidung, Routingtabelle und Standardgateway

Routingentscheidung Router vergleichen die Zielnetze in der Routingtabelle mit den Zielnetzen der ankommenden Pakete.

Die Präfixlänge gibt an wieviele Bit von vorne gezählt übereinstimmen müssen

Bei mehreren Übereinstimmungen: **best match** (grösste Anzahl passender Bits)

Routingtabelle Liste mit Routen

Default Gateway Das Default Gateway ist ein Eintrag in der Routingtabelle. Besonderheit: das Zielnetz hat die spezielle IP **0.0.0.0/0**

0.0.0.0/0 passt zu **jeder** IP-Adresse: 0 von 0 Bit müssen übereinstimmen

Routingentscheidung, Routingtabelle und Standardgateway

Routingentscheidung Router vergleichen die Zielnetze in der Routingtabelle mit den Zielnetzen der ankommenden Pakete.

Die Präfixlänge gibt an wieviele Bit von vorne gezählt übereinstimmen müssen

Bei mehreren Übereinstimmungen: **best match** (grösste Anzahl passender Bits)

Routingtabelle Liste mit Routen

Default Gateway Das Default Gateway ist ein Eintrag in der Routingtabelle. Besonderheit: das Zielnetz hat die spezielle IP **0.0.0.0/0**

0.0.0.0/0 passt zu **jeder** IP-Adresse: 0 von 0 Bit müssen übereinstimmen

Routingentscheidung, Routingtabelle und Standardgateway

Routingentscheidung Router vergleichen die Zielnetze in der Routingtabelle mit den Zielnetzen der ankommenden Pakete.

Die Präfixlänge gibt an wieviele Bit von vorne gezählt übereinstimmen müssen

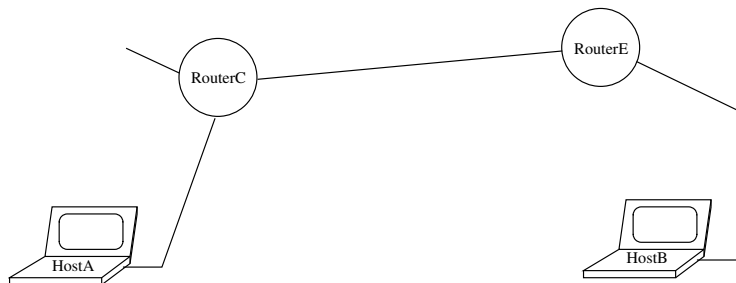
Bei mehreren Übereinstimmungen: **best match** (grösste Anzahl passender Bits)

Routingtabelle Liste mit Routen

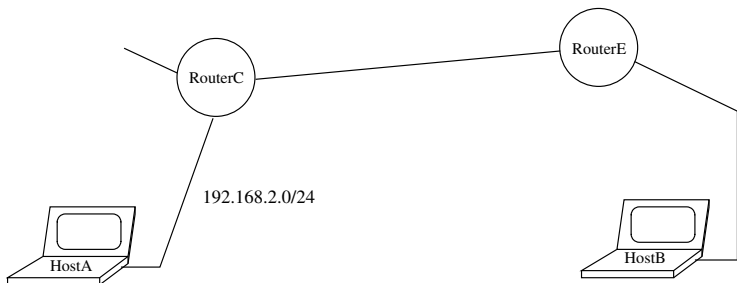
Default Gateway Das Default Gateway ist ein Eintrag in der Routingtabelle. Besonderheit: das Zielnetz hat die spezielle IP **0.0.0.0/0**

0.0.0.0/0 passt zu **jeder** IP-Adresse: 0 von 0 Bit müssen übereinstimmen

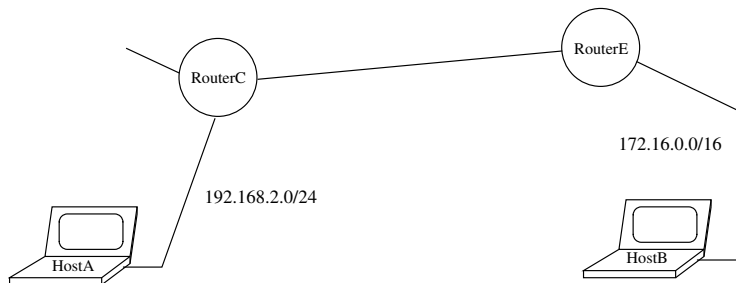
Der Weg eines Pakets durch das Netz



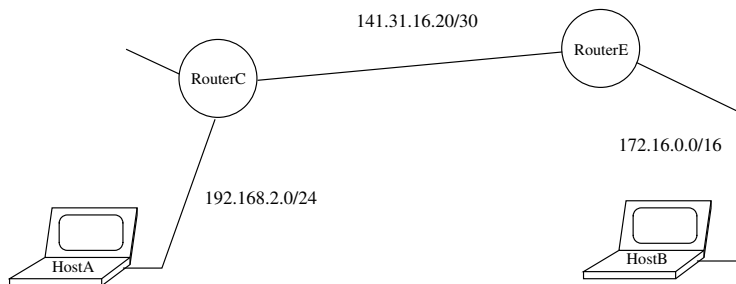
Der Weg eines Pakets durch das Netz



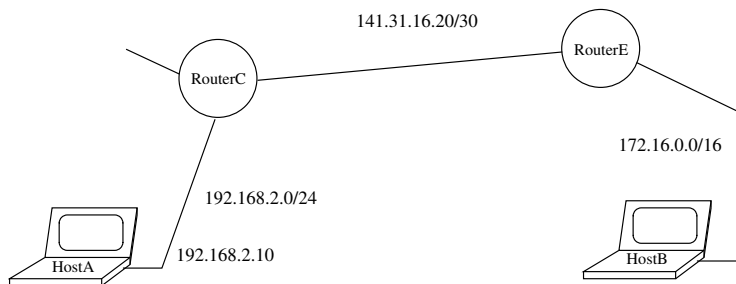
Der Weg eines Pakets durch das Netz



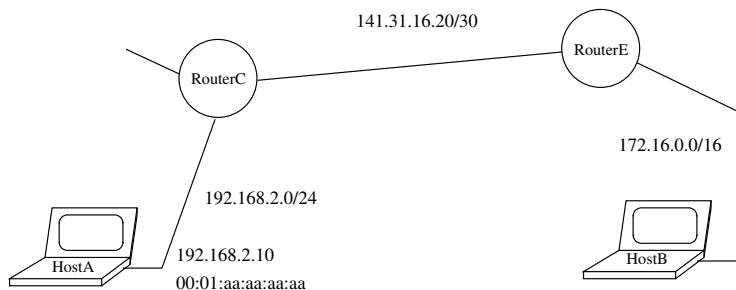
Der Weg eines Pakets durch das Netz



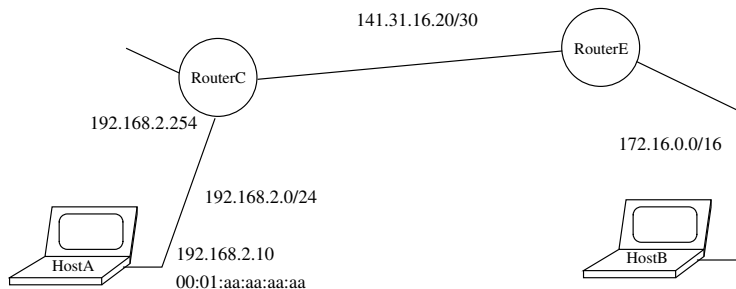
Der Weg eines Pakets durch das Netz



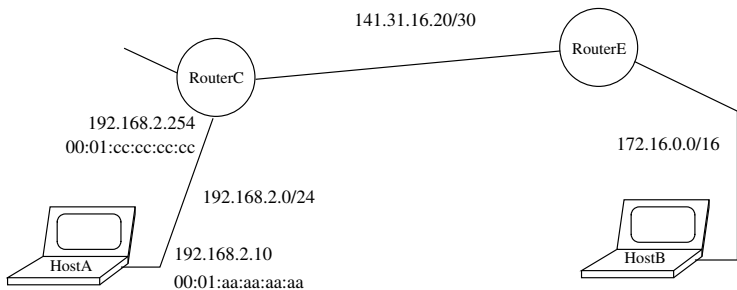
Der Weg eines Pakets durch das Netz



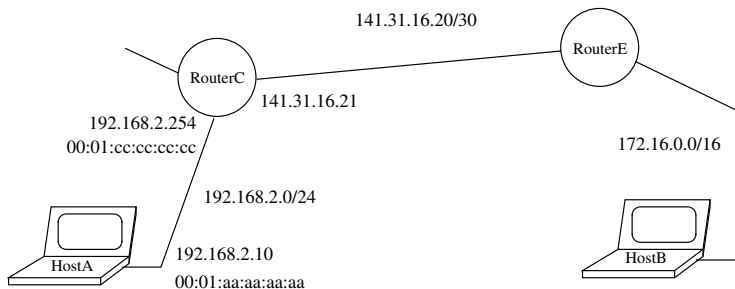
Der Weg eines Pakets durch das Netz



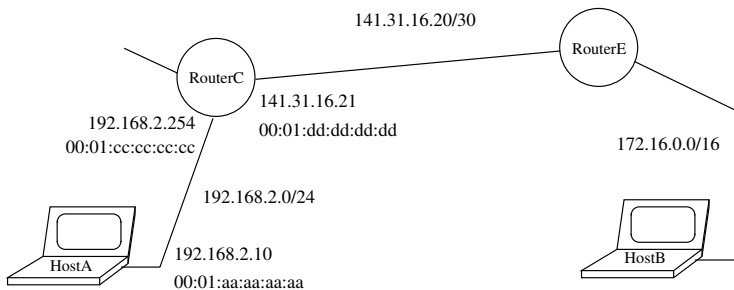
Der Weg eines Pakets durch das Netz



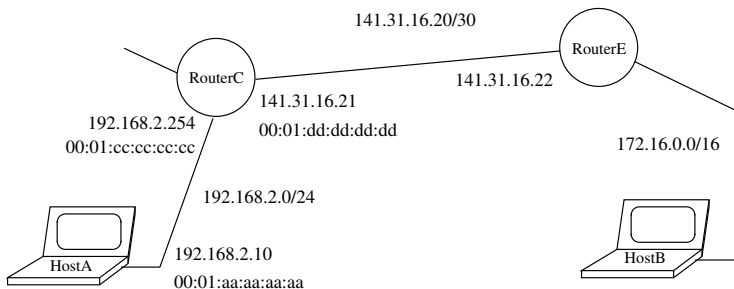
Der Weg eines Pakets durch das Netz



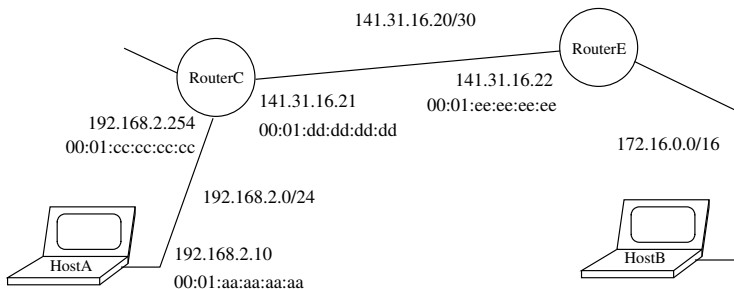
Der Weg eines Pakets durch das Netz



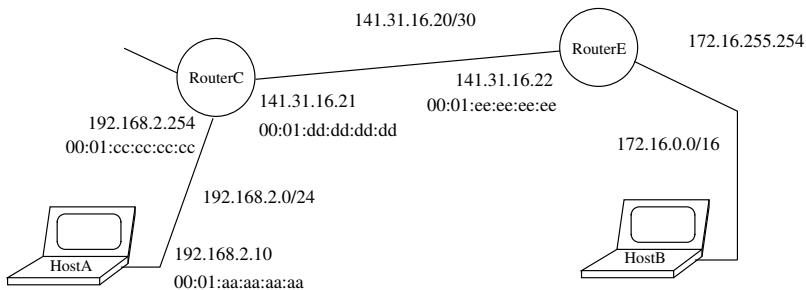
Der Weg eines Pakets durch das Netz



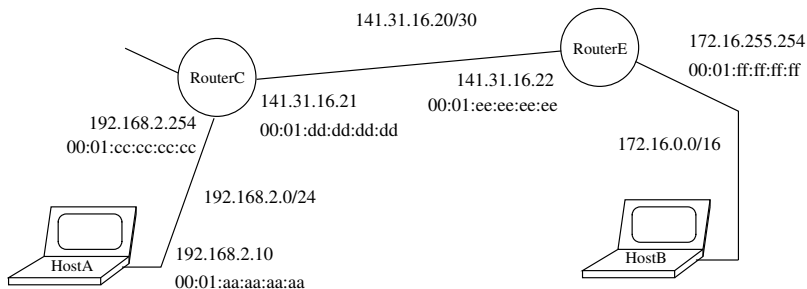
Der Weg eines Pakets durch das Netz



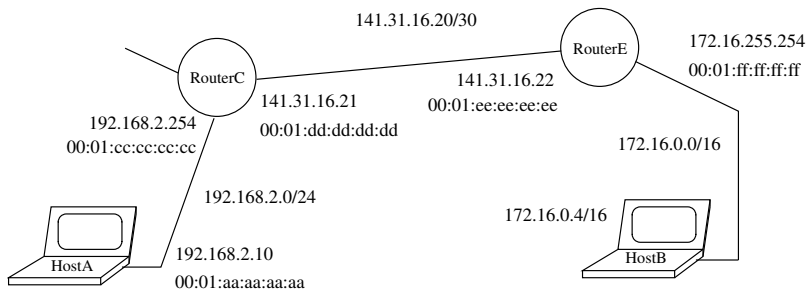
Der Weg eines Pakets durch das Netz



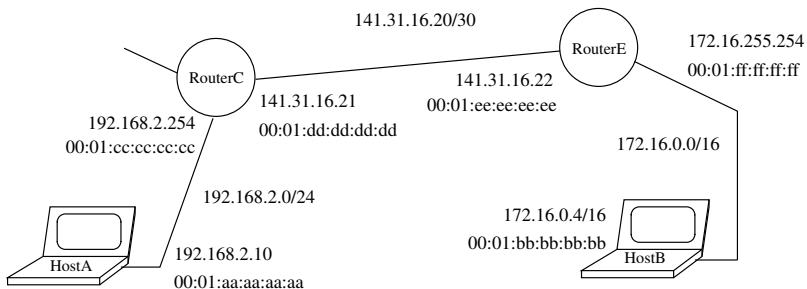
Der Weg eines Pakets durch das Netz



Der Weg eines Pakets durch das Netz

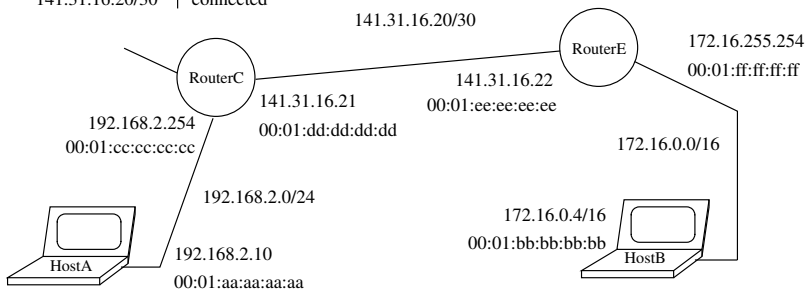


Der Weg eines Pakets durch das Netz

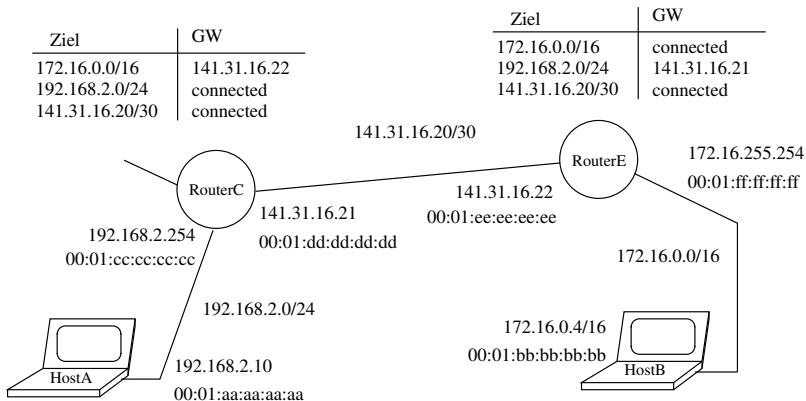


Der Weg eines Pakets durch das Netz

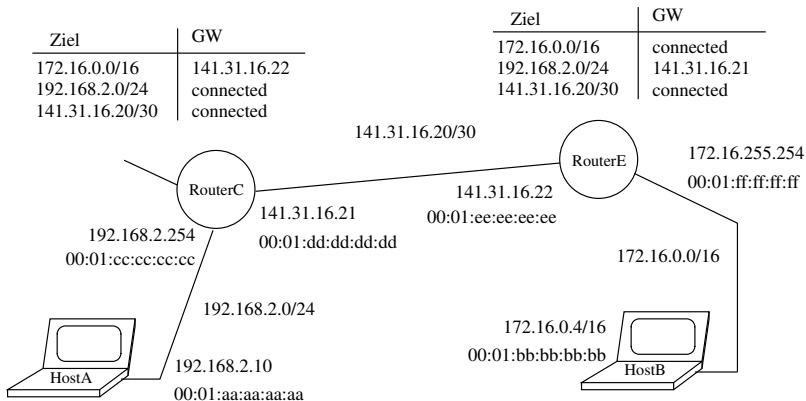
Ziel	GW
172.16.0.0/16	141.31.16.22
192.168.2.0/24	connected
141.31.16.20/30	connected



Der Weg eines Pakets durch das Netz



Der Weg eines Pakets durch das Netz

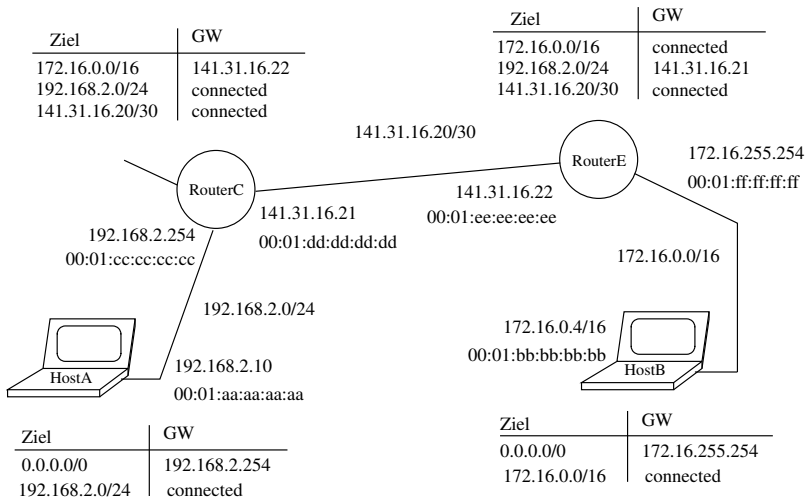


Ziel	GW
172.16.0.0/16	141.31.16.22
192.168.2.0/24	connected
141.31.16.20/30	connected

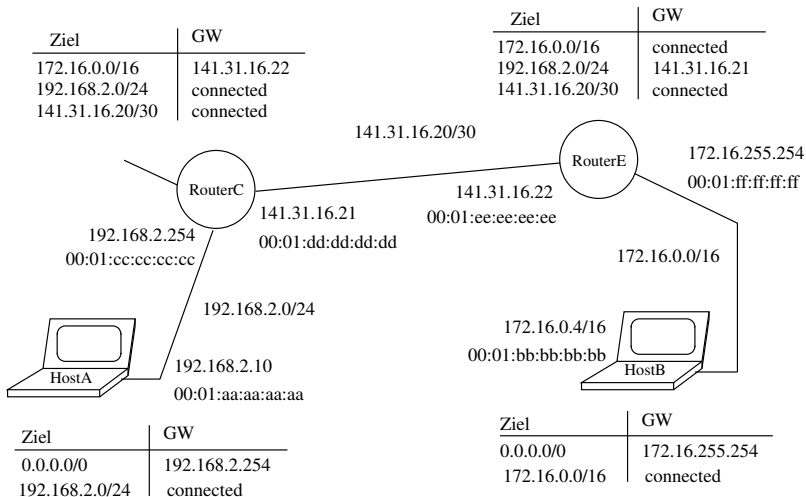
Ziel	GW
172.16.0.0/16	connected
192.168.2.0/24	141.31.16.21
141.31.16.20/30	connected

Ziel	GW
0.0.0.0/0	192.168.2.254
192.168.2.0/24	connected

Der Weg eines Pakets durch das Netz



Der Weg eines Pakets durch das Netz



Der Weg eines Pakets durch das Netz

- HostA sendet Paket an HostB
- ist das Ziel im eigenen Netz? Nein!
- gibt es ein Standardgateway in meiner Routingtabelle?
Ja, IP=192.168.2.254
- Kenne ich die MAC-Adresse von 192.168.2.254? Nein.
Frage an alle (Broadcast): wer hat 192.168.2.254?
- 192.168.2.254 sendet Antwortpaket (ARP-Reply) gesuchte
MAC ist enthalten
- Frame von HostA:

Ziel-MAC	00:01:cc:cc:...
Quell-MAC	00:01::aa:aa:...
Quell-IP	192.168.2.10
Ziel-ip	172.16.0.4

Der Weg eines Pakets durch das Netz

- **HostA sendet Paket an HostB**
- ist das Ziel im eigenen Netz? Nein!
- gibt es ein Standardgateway in meiner Routingtabelle?
Ja, IP=192.168.2.254
- Kenne ich die MAC-Adresse von 192.168.2.254? Nein.
Frage an alle (Broadcast): wer hat 192.168.2.254?
- 192.168.2.254 sendet Antwortpaket (ARP-Reply) gesuchte
MAC ist enthalten
- Frame von HostA:

Ziel-MAC	00:01:cc:cc:...
Quell-MAC	00:01::aa:aa:...
Quell-IP	192.168.2.10
Ziel-ip	172.16.0.4

Der Weg eines Pakets durch das Netz

- HostA sendet Paket an HostB
- ist das Ziel im eigenen Netz? Nein!
- gibt es ein Standardgateway in meiner Routingtabelle?
Ja, IP=192.168.2.254
- Kenne ich die MAC-Adresse von 192.168.2.254? Nein.
Frage an alle (Broadcast): wer hat 192.168.2.254?
- 192.168.2.254 sendet Antwortpaket (ARP-Reply) gesuchte
MAC ist enthalten
- Frame von HostA:

Ziel-MAC	00:01:cc:cc:...
Quell-MAC	00:01::aa:aa:...
Quell-IP	192.168.2.10
Ziel-ip	172.16.0.4

Der Weg eines Pakets durch das Netz

- HostA sendet Paket an HostB
- ist das Ziel im eigenen Netz? Nein!
- gibt es ein Standardgateway in meiner Routingtabelle?
Ja, IP=192.168.2.254
- Kenne ich die MAC-Adresse von 192.168.2.254? Nein.
Frage an alle (Broadcast): wer hat 192.168.2.254?
- 192.168.2.254 sendet Antwortpaket (ARP-Reply) gesuchte MAC ist enthalten
- Frame von HostA:

Ziel-MAC	00:01:cc:cc:...
Quell-MAC	00:01::aa:aa:...
Quell-IP	192.168.2.10
Ziel-ip	172.16.0.4

Der Weg eines Pakets durch das Netz

- HostA sendet Paket an HostB
- ist das Ziel im eigenen Netz? Nein!
- gibt es ein Standardgateway in meiner Routingtabelle?
Ja, IP=192.168.2.254
- Kenne ich die MAC-Adresse von 192.168.2.254? Nein.
Frage an alle (Broadcast): wer hat 192.168.2.254?
- 192.168.2.254 sendet Antwortpaket (ARP-Reply) gesuchte MAC ist enthalten
- Frame von HostA:

Ziel-MAC	00:01:cc:cc:...
Quell-MAC	00:01::aa:aa:...
Quell-IP	192.168.2.10
Ziel-ip	172.16.0.4

Der Weg eines Pakets durch das Netz

- HostA sendet Paket an HostB
- ist das Ziel im eigenen Netz? Nein!
- gibt es ein Standardgateway in meiner Routingtabelle?
Ja, IP=192.168.2.254
- Kenne ich die MAC-Adresse von 192.168.2.254? Nein.
Frage an alle (Broadcast): wer hat 192.168.2.254?
- 192.168.2.254 sendet Antwortpaket (ARP-Reply) gesuchte
MAC ist enthalten
- Frame von HostA:

Ziel-MAC	00:01:cc:cc:...
Quell-MAC	00:01::aa:aa:...
Quell-IP	192.168.2.10
Ziel-ip	172.16.0.4

Der Weg eines Pakets durch das Netz

- HostA sendet Paket an HostB
- ist das Ziel im eigenen Netz? Nein!
- gibt es ein Standardgateway in meiner Routingtabelle?
Ja, IP=192.168.2.254
- Kenne ich die MAC-Adresse von 192.168.2.254? Nein.
Frage an alle (Broadcast): wer hat 192.168.2.254?
- 192.168.2.254 sendet Antwortpaket (ARP-Reply) gesuchte
MAC ist enthalten
- Frame von HostA:

Ziel-MAC	00:01:cc:cc:...
Quell-MAC	00:01::aa:aa:...
Quell-IP	192.168.2.10
Ziel-ip	172.16.0.4

Der Weg eines Pakets durch das Netz

- HostA sendet Paket an HostB
- ist das Ziel im eigenen Netz? Nein!
- gibt es ein Standardgateway in meiner Routingtabelle?
Ja, IP=192.168.2.254
- Kenne ich die MAC-Adresse von 192.168.2.254? Nein.
Frage an alle (Broadcast): wer hat 192.168.2.254?
- 192.168.2.254 sendet Antwortpaket (ARP-Reply) gesuchte MAC ist enthalten
- Frame von HostA:

Ziel-MAC	00:01:cc:cc:...
Quell-MAC	00:01::aa:aa:...
Quell-IP	192.168.2.10
Ziel-ip	172.16.0.4

Der Weg eines Pakets durch das Netz

- HostA sendet Paket an HostB
- ist das Ziel im eigenen Netz? Nein!
- gibt es ein Standardgateway in meiner Routingtabelle?
Ja, IP=192.168.2.254
- Kenne ich die MAC-Adresse von 192.168.2.254? Nein.
Frage an alle (Broadcast): wer hat 192.168.2.254?
- 192.168.2.254 sendet Antwortpaket (ARP-Reply) gesuchte
MAC ist enthalten
- Frame von HostA:

Ziel-MAC	00:01:cc:cc:...
Quell-MAC	00:01::aa:aa:...
Quell-IP	192.168.2.10
Ziel-ip	172.16.0.4

Der Weg eines Pakets durch das Netz

- HostA sendet Paket an HostB
- ist das Ziel im eigenen Netz? Nein!
- gibt es ein Standardgateway in meiner Routingtabelle?
Ja, IP=192.168.2.254
- Kenne ich die MAC-Adresse von 192.168.2.254? Nein.
Frage an alle (Broadcast): wer hat 192.168.2.254?
- 192.168.2.254 sendet Antwortpaket (ARP-Reply) **gesuchte MAC ist enthalten**
- Frame von HostA:

Ziel-MAC	00:01:cc:cc:...
Quell-MAC	00:01::aa:aa:...
Quell-IP	192.168.2.10
Ziel-ip	172.16.0.4

Der Weg eines Pakets durch das Netz

- HostA sendet Paket an HostB
- ist das Ziel im eigenen Netz? Nein!
- gibt es ein Standardgateway in meiner Routingtabelle?
Ja, IP=192.168.2.254
- Kenne ich die MAC-Adresse von 192.168.2.254? Nein.
Frage an alle (Broadcast): wer hat 192.168.2.254?
- 192.168.2.254 sendet Antwortpaket (ARP-Reply) gesuchte
MAC ist enthalten
- Frame von HostA:

Ziel-MAC	00:01:cc:cc:...
Quell-MAC	00:01::aa:aa:...
Quell-IP	192.168.2.10
Ziel-ip	172.16.0.4

Der Weg eines Pakets durch das Netz

- HostA sendet Paket an HostB
- ist das Ziel im eigenen Netz? Nein!
- gibt es ein Standardgateway in meiner Routingtabelle?
Ja, IP=192.168.2.254
- Kenne ich die MAC-Adresse von 192.168.2.254? Nein.
Frage an alle (Broadcast): wer hat 192.168.2.254?
- 192.168.2.254 sendet Antwortpaket (ARP-Reply) gesuchte
MAC ist enthalten
- Frame von HostA:

Ziel-MAC	00:01:cc:cc:...
Quell-MAC	00:01::aa:aa:...
Quell-IP	192.168.2.10
Ziel-ip	172.16.0.4

Der Weg eines Pakets durch das Netz

- HostA sendet Paket an HostB
- ist das Ziel im eigenen Netz? Nein!
- gibt es ein Standardgateway in meiner Routingtabelle?
Ja, IP=192.168.2.254
- Kenne ich die MAC-Adresse von 192.168.2.254? Nein.
Frage an alle (Broadcast): wer hat 192.168.2.254?
- 192.168.2.254 sendet Antwortpaket (ARP-Reply) gesuchte
MAC ist enthalten
- Frame von HostA:

Ziel-MAC	00:01:cc:cc:...
Quell-MAC	00:01::aa:aa:...
Quell-IP	192.168.2.10
Ziel-ip	172.16.0.4

Der Weg eines Pakets durch das Netz

- RouterC nimmt Frame in Empfang und lädt Palette aus → Der Frame wird entfernt und nur das IP-Paket bleibt erhalten
- Habe ich für das Netz der Ziel-IP eine Route in meiner Routingtabelle?
- Wenn nein, Paket verwerfen
- Wenn ja, mit ARP die MAC des Gateways erfragen oder aus Cache lesen
- Frame von RouterC:

Ziel-MAC	00:01:ee:ee:...
Quell-MAC	00:01:dd:dd:...
Quell-IP	192.168.2.10
Ziel-ip	172.16.0.4

Der Weg eines Pakets durch das Netz

- **RouterC nimmt Frame in Empfang und lädt Palette aus** →
Der Frame wird entfernt und nur das IP-Paket bleibt erhalten
- Habe ich für das Netz der Ziel-IP eine Route in meiner Routingtabelle?
- Wenn nein, Paket verwerfen
- Wenn ja, mit ARP die MAC des Gateways erfragen oder aus Cache lesen
- Frame von RouterC:

Ziel-MAC	00:01:ee:ee:...
Quell-MAC	00:01:dd:dd:...
Quell-IP	192.168.2.10
Ziel-ip	172.16.0.4

Der Weg eines Pakets durch das Netz

- RouterC nimmt Frame in Empfang und lädt Palette aus → Der Frame wird entfernt und nur das IP-Paket bleibt erhalten
- Habe ich für das Netz der Ziel-IP eine Route in meiner Routingtabelle?
- Wenn nein, Paket verwerfen
- Wenn ja, mit ARP die MAC des Gateways erfragen oder aus Cache lesen
- Frame von RouterC:

Ziel-MAC	00:01:ee:ee:...
Quell-MAC	00:01:dd:dd:...
Quell-IP	192.168.2.10
Ziel-ip	172.16.0.4

Der Weg eines Pakets durch das Netz

- RouterC nimmt Frame in Empfang und lädt Palette aus → Der Frame wird entfernt und nur das IP-Paket bleibt erhalten
- Habe ich für das Netz der Ziel-IP eine Route in meiner Routingtabelle?
- Wenn nein, Paket verwerfen
- Wenn ja, mit ARP die MAC des Gateways erfragen oder aus Cache lesen
- Frame von RouterC:

Ziel-MAC	00:01:ee:ee:...
Quell-MAC	00:01:dd:dd:...
Quell-IP	192.168.2.10
Ziel-ip	172.16.0.4

Der Weg eines Pakets durch das Netz

- RouterC nimmt Frame in Empfang und lädt Palette aus → Der Frame wird entfernt und nur das IP-Paket bleibt erhalten
- Habe ich für das Netz der Ziel-IP eine Route in meiner Routingtabelle?
- Wenn nein, Paket verwerfen
- Wenn ja, mit ARP die MAC des Gateways erfragen oder aus Cache lesen
- Frame von RouterC:

Ziel-MAC	00:01:ee:ee:...
Quell-MAC	00:01:dd:dd:...
Quell-IP	192.168.2.10
Ziel-ip	172.16.0.4

Der Weg eines Pakets durch das Netz

- RouterC nimmt Frame in Empfang und lädt Palette aus →
Der Frame wird entfernt und nur das IP-Paket bleibt erhalten
- Habe ich für das Netz der Ziel-IP eine Route in meiner Routingtabelle?
- Wenn nein, Paket verwerfen
- Wenn ja, mit ARP die MAC des Gateways erfragen oder aus Cache lesen
- Frame von RouterC:

Ziel-MAC	00:01:ee:ee:...
Quell-MAC	00:01:dd:dd:...
Quell-IP	192.168.2.10
Ziel-ip	172.16.0.4

Der Weg eines Pakets durch das Netz

- RouterC nimmt Frame in Empfang und lädt Palette aus →
Der Frame wird entfernt und nur das IP-Paket bleibt erhalten
- Habe ich für das Netz der Ziel-IP eine Route in meiner Routingtabelle?
- Wenn nein, Paket verwerfen
- Wenn ja, mit ARP die MAC des Gateways erfragen oder aus Cache lesen
- Frame von RouterC:

Ziel-MAC	00:01:ee:ee:...
Quell-MAC	00:01:dd:dd:...
Quell-IP	192.168.2.10
Ziel-ip	172.16.0.4

Der Weg eines Pakets durch das Netz

- RouterC nimmt Frame in Empfang und lädt Palette aus → Der Frame wird entfernt und nur das IP-Paket bleibt erhalten
- Habe ich für das Netz der Ziel-IP eine Route in meiner Routingtabelle?
- Wenn nein, Paket verwerfen
- Wenn ja, mit ARP die MAC des Gateways erfragen oder aus Cache lesen
- Frame von RouterC:

Ziel-MAC	00:01:ee:ee:...
Quell-MAC	00:01:dd:dd:...
Quell-IP	192.168.2.10
Ziel-ip	172.16.0.4

Der Weg eines Pakets durch das Netz

- RouterE nimmt Frame in Empfang und behält nur das Layer3-Paket (IP-Paket).
- Habe ich für das Netz der Ziel-IP eine Route in meiner Routingtabelle?
- Wenn nein, Paket verwerfen
- Wenn ja, mit ARP zugehörige MAC erfragen oder aus Cache lesen
- Frame von RouterE:

Ziel-MAC	00:01:bb:bb:...
Quell-MAC	00:01:ff:ff:...
Quell-IP	192.168.2.10
Ziel-ip	172.16.0.4

Der Weg eines Pakets durch das Netz

- RouterE nimmt Frame in Empfang und behält nur das Layer3-Paket (IP-Paket).
- Habe ich für das Netz der Ziel-IP eine Route in meiner Routingtabelle?
- Wenn nein, Paket verwerfen
- Wenn ja, mit ARP zugehörige MAC erfragen oder aus Cache lesen
- Frame von RouterE:

Ziel-MAC	00:01:bb:bb:...
Quell-MAC	00:01:ff:ff:...
Quell-IP	192.168.2.10
Ziel-ip	172.16.0.4

Der Weg eines Pakets durch das Netz

- RouterE nimmt Frame in Empfang und behält nur das Layer3-Paket (IP-Paket).
- Habe ich für das Netz der Ziel-IP eine Route in meiner Routingtabelle?
- Wenn nein, Paket verwerfen
- Wenn ja, mit ARP zugehörige MAC erfragen oder aus Cache lesen
- Frame von RouterE:

Ziel-MAC	00:01:bb:bb:...
Quell-MAC	00:01:ff:ff:...
Quell-IP	192.168.2.10
Ziel-ip	172.16.0.4

Der Weg eines Pakets durch das Netz

- RouterE nimmt Frame in Empfang und behält nur das Layer3-Paket (IP-Paket).
- Habe ich für das Netz der Ziel-IP eine Route in meiner Routingtabelle?
- Wenn nein, Paket verwerfen
- Wenn ja, mit ARP zugehörige MAC erfragen oder aus Cache lesen
- Frame von RouterE:

Ziel-MAC	00:01:bb:bb:...
Quell-MAC	00:01:ff:ff:...
Quell-IP	192.168.2.10
Ziel-ip	172.16.0.4

Der Weg eines Pakets durch das Netz

- RouterE nimmt Frame in Empfang und behält nur das Layer3-Paket (IP-Paket).
- Habe ich für das Netz der Ziel-IP eine Route in meiner Routingtabelle?
- Wenn nein, Paket verwerfen
- Wenn ja, mit ARP zugehörige MAC erfragen oder aus Cache lesen
- Frame von RouterE:

Ziel-MAC	00:01:bb:bb:...
Quell-MAC	00:01:ff:ff:...
Quell-IP	192.168.2.10
Ziel-ip	172.16.0.4

Der Weg eines Pakets durch das Netz

- RouterE nimmt Frame in Empfang und behält nur das Layer3-Paket (IP-Paket).
- Habe ich für das Netz der Ziel-IP eine Route in meiner Routingtabelle?
- Wenn nein, Paket verwerfen
- Wenn ja, mit ARP zugehörige MAC erfragen oder aus Cache lesen
- Frame von RouterE:

Ziel-MAC	00:01:bb:bb:...
Quell-MAC	00:01:ff:ff:...
Quell-IP	192.168.2.10
Ziel-ip	172.16.0.4

Der Weg eines Pakets durch das Netz

- RouterE nimmt Frame in Empfang und behält nur das Layer3-Paket (IP-Paket).
- Habe ich für das Netz der Ziel-IP eine Route in meiner Routingtabelle?
- Wenn nein, Paket verwerfen
- Wenn ja, mit ARP zugehörige MAC erfragen oder aus Cache lesen
- Frame von RouterE:

Ziel-MAC	00:01:bb:bb:...
Quell-MAC	00:01:ff:ff:...
Quell-IP	192.168.2.10
Ziel-ip	172.16.0.4

Schluss

Vielen Dank für's Zuhören und Mitmachen!