

Technische Grundlagen der Informatik 2

– Teil 6:

Netzwerkschicht (Layer 3)

Philipp Rettberg / Sebastian Harnau

Block 10/18

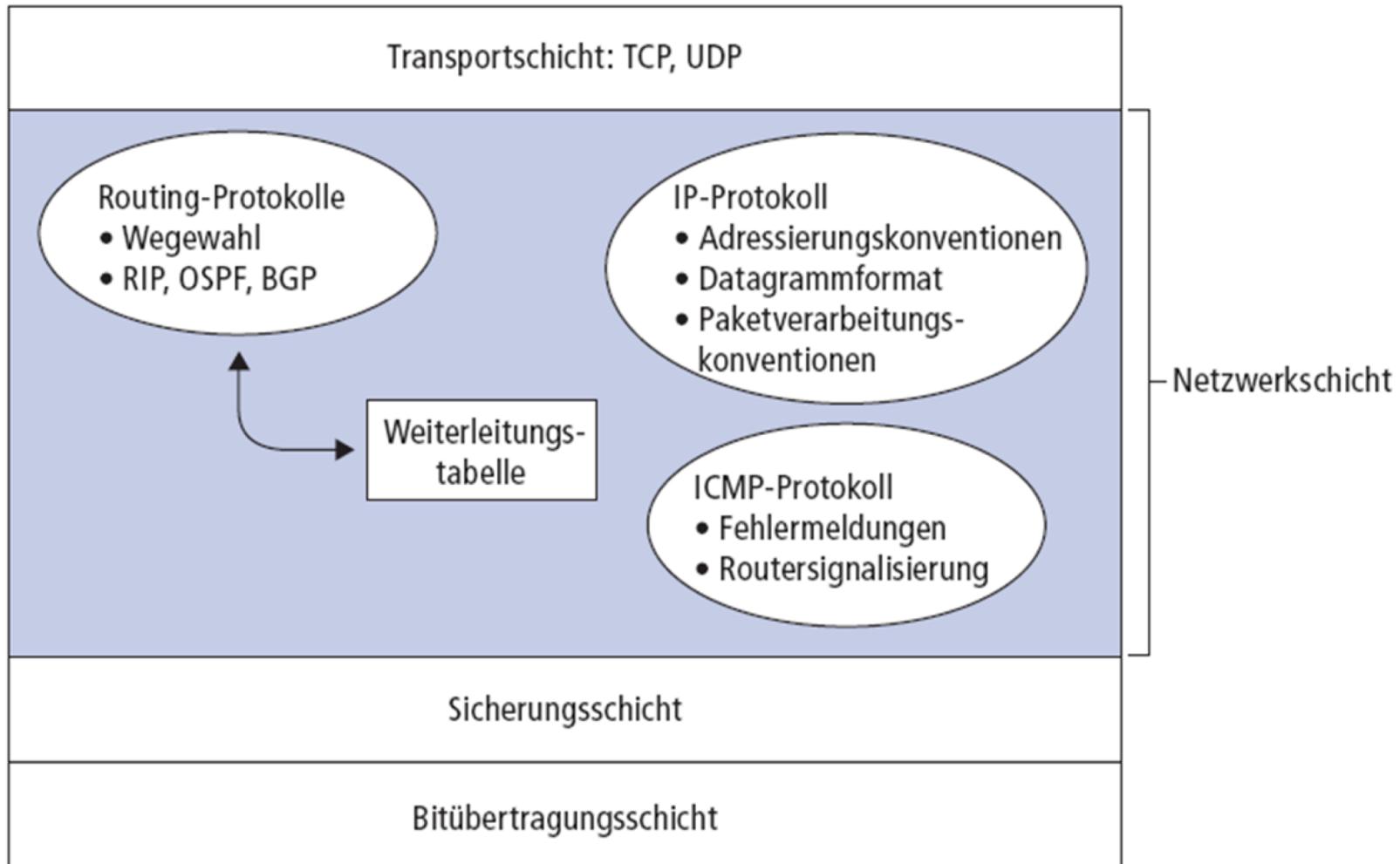
Netzwerkschicht (Layer 3)

IP, Router, Routing...

Aufgabe der Netzwerkschicht

- Wie gelangen Pakete/Datagramme von einem Host zu einem anderen Host durch die Verbünde vieler Netzwerke?

„Protokolle“ der Netzwerkschicht



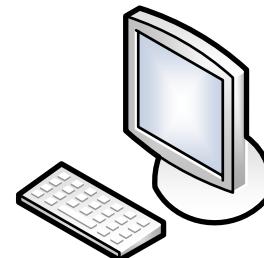
Adressierung von Hosts bzw. Routern

Namen/Alias
im DNS



host.netzid.local

IP-Adresse
eines Interfaces



192.168.178.21

MAC-Adresse
der Netzwerkkarte



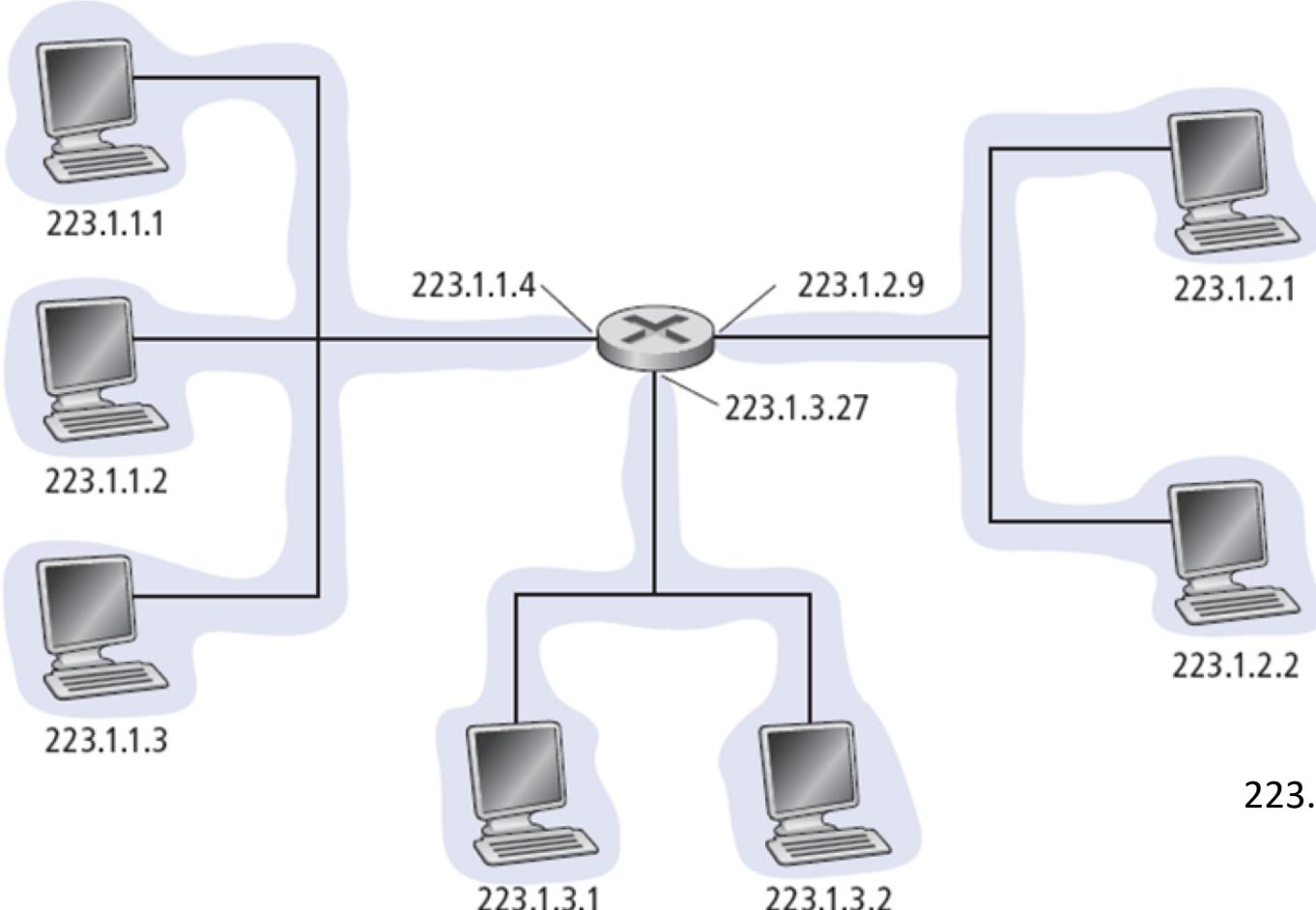
C4-54-44-52-A3-85

Layer 7
(Applikation)

Layer 3
(Netzwerkschicht)

Layer 2
(Sicherungsschicht)

IPv4-Adressierung



IP-Adresse:

- 32-Bit-Kennung für das Interface (Schnittstelle) eines Endsystems oder eines Routers

$223.1.1.1 = \underline{11011111} \underline{00000001} \underline{00000001} \underline{00000001}$

223 1 1 1

IPv4-Adressierung

Interface:

Verbindung zwischen dem System und dem Link

- Wird normalerweise durch eine Netzwerkkarte bereitgestellt
- Router haben typischerweise mehrere Interfaces
- Endsysteme können ebenfalls mehrere Interfaces haben
- Jedes Interface besitzt eine IP-Adresse

IPv4-Adressierung Netzwerke

223.1.1.1

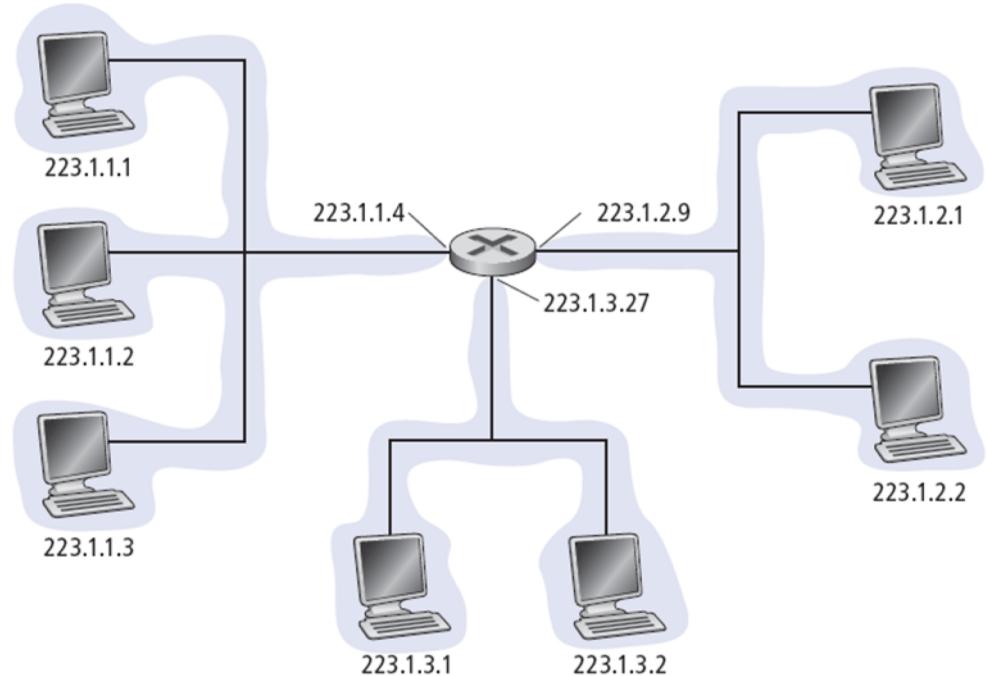
Eine IP-Adresse hat zwei Bestandteile:

netid: die oberen Bits der Adresse, identifiziert ein Netzwerk

223.1.1.1

hostid: die unteren Bits der Adresse, identifiziert ein Interface eines Systems

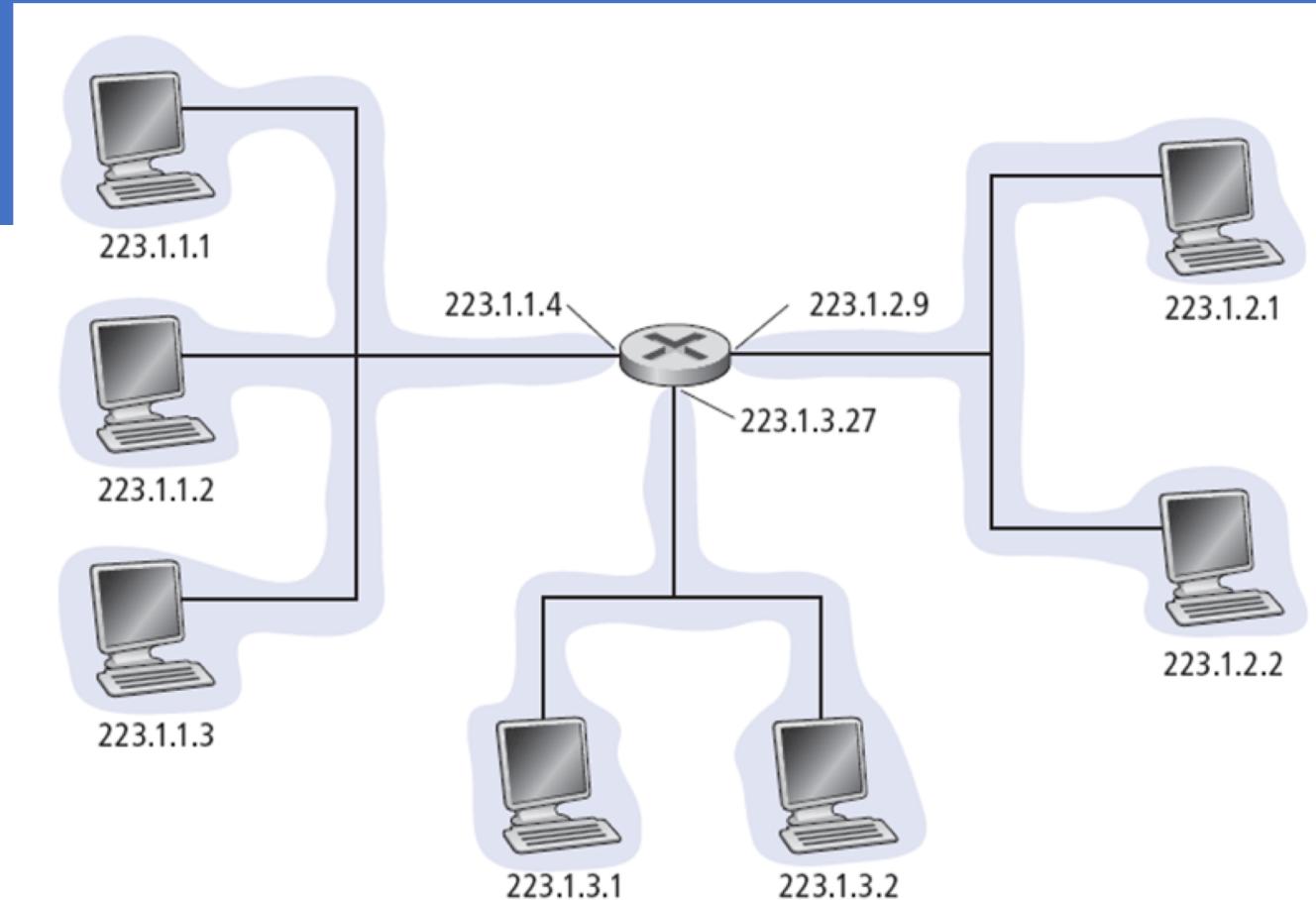
223.1.1.1



Subnetzwerk

Was ist ein (Sub-)Netzwerk?

- Alle Interfaces mit derselben **netid** formen ein Netzwerk
- Alle Interfaces eines Netzwerkes können sich direkt (ohne einen Router zu durchqueren) erreichen

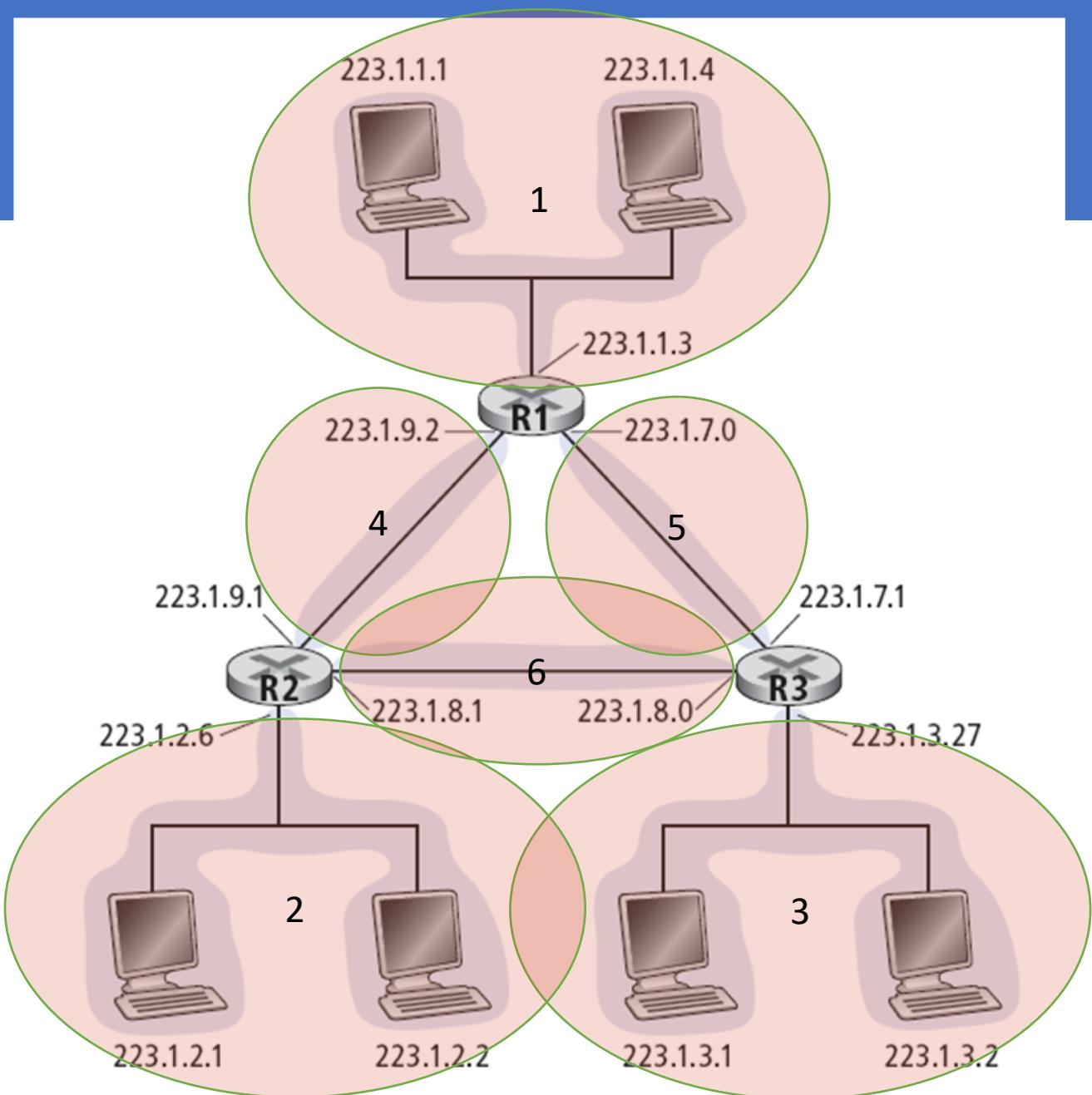


Drei IP-Netzwerke, die mit einem Router verbunden sind.
Die **netid** steht hier in den oberen 24 Bit.

Subnetzmaske: /24 oder 255.255.255.0

Subnetzwerke

Wieviele Subnetze sind hier erkennbar?



IPv4-Adressierung Adressklassen

- Früher wurden IP-Adressen in Adressklassen aufgeteilt
- Die Klasse bestimmte das Verhältnis der Längen netid/hostid
- Dies nennt man „classfull“ addressing oder auch klassenbasierte Adressierung

Klasse

A	0	netid		hostid		hostid	1.0.0.0 bis 127.255.255.255
B	10	netid			hostid		128.0.0.0 bis 191.255.255.255
C	110		netid			hostid	192.0.0.0 bis 223.255.255.255
D	1110		Multicast-Adresse				224.0.0.0 bis 239.255.255.255
← 32 Bits →							

IPv4 Adressierung von Subnetzen

Klasse-A- und –B-Adressen haben Platz für mehr Endsysteme, als man in einem Netzwerk sinnvoll unterbringen kann

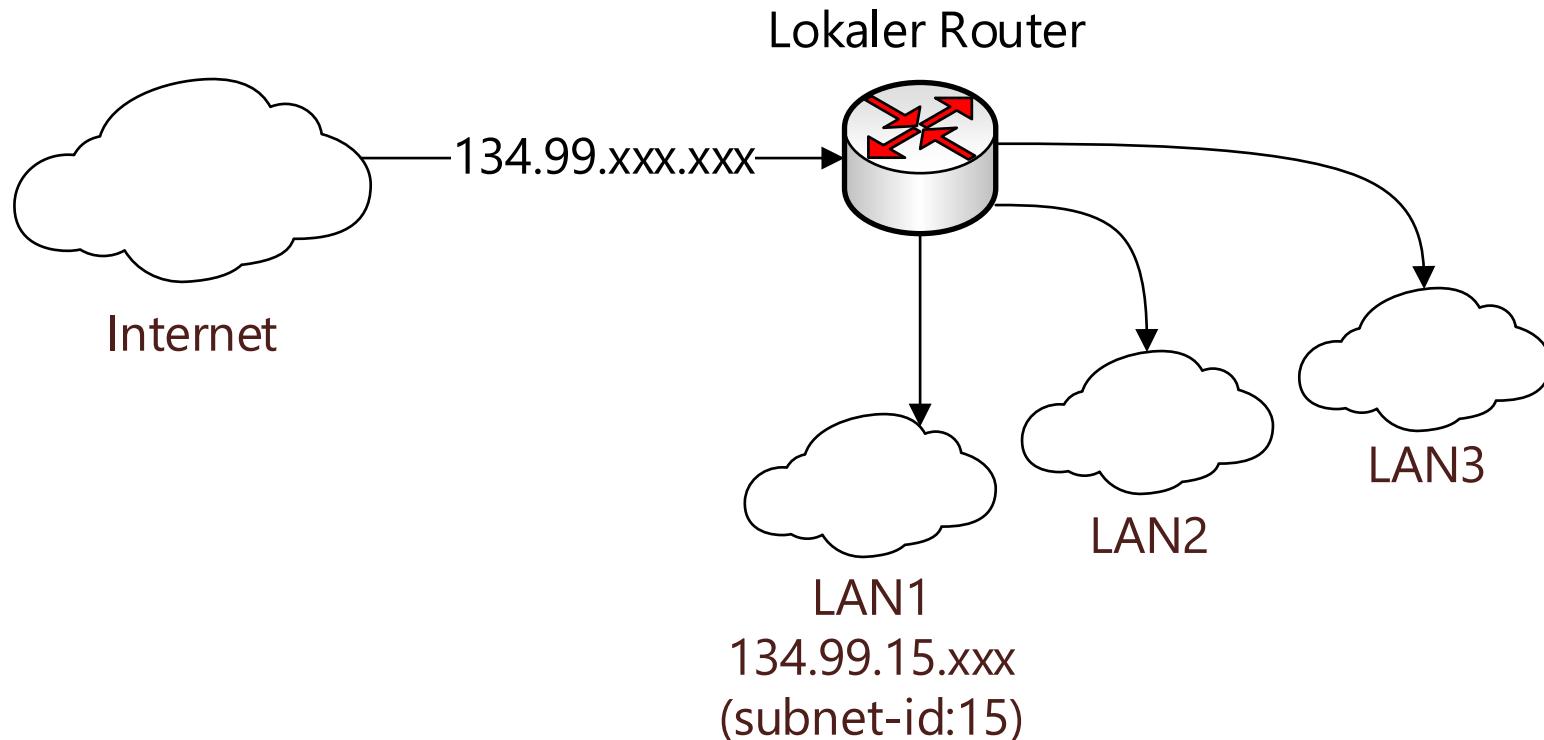
Daher teilt man die hostid weiter auf, z.B. so:



Die Unterteilung (subnetid, hostid) ist eine lokale Entscheidung und wird von der Organisation vorgenommen, der die netid zugeordnet wurde

IPv4 Adressierung von Subnetzen

Die subnetid ist außerhalb des Netzwerkes, für das sie verwendet wird, nicht sichtbar:



IPv4 Adressierung von Subnetzen

Subnetzmaske (subnet mask)

- Wird für jede IP-Adresse eines Systems im System gespeichert
- Sie identifiziert, welcher Teil der Adresse zur subnetid und welcher zur hostid gehört

	16 Bit	8 Bit	8 Bit
Beispiel (Class B)	1111111111111111	11111111	00000000

subnet mask: 255.255.255.0 oder auch /24

IPv4 Adressierung von Subnetzen

Die eigene IP-Adresse in Verbindung mit der Subnetzmaske erlaubt Rückschlüsse darüber, wo sich eine andere IP-Adresse befindet:

- im selben Subnetz (also direkt erreichbar)
- im selben Netzwerk, aber in einem anderen Subnetz
- in einem anderen Netzwerk

Beispiel Subnetzmasken

Gegeben:

- Eigene IP-Adresse: 134.155.48.10
- Subnetzmaske: 255.255.255.0
- Adresse A: 134.155.48.96, Adresse B: 134.155.55.96



Überprüfen der beiden Adressen:

- 134.155.48.10 & 255.255.255.0 =
 - 134.155.48.0
- 134.155.48.96 & 255.255.255.0 =
 - 134.155.48.0 → **identisch, gleiches Subnetz**
- 134.155.55.96 & 255.255.255.0 =
 - 134.155.55.0 → **verschieden, anderes Subnetz**

192.168.11.0 - 192.168.11.255

192.168.11.0 -> Networkname
" . " . " . 255 -> Broadcast Address

↓
254 nutzbare Adressen (253 mit Router)

IPv4 Subnetzmasken mit variabler Länge

Problem: Gegeben sei ein Klasse-C-Netzwerk, welches in zwei Subnetze mit 50 Endsystemen und ein Subnetz mit 100 Endsystemen unterteilt werden soll.

Das funktioniert nicht mit einer einzelnen Subnetzmaske!

- 255.255.255.128: zwei Netze mit je 128 hostids
- 255.255.255.192: vier Netze mit je 64 hostids

Lösung: **Subnetzmasken variabler Länge** *Hinweis: Richtigkeit der Lösung auf nächster Seite ::*

- Unterteile den Adressraum zunächst mit der kürzeren Subnetzmaske (1 Bit im Beispiel)
- Unterteile eine Hälfte davon weiter mit der längeren Subnetzmaske (2 Bit im Beispiel)
- Resultat: Subnetze verschiedener Größe

Teilung in 3 Netze mit 409, 50, 50 IPs:

192.168.11.0 /25

$\begin{array}{r} 0\ 000\ 0000 \\ 0\ 111\ 1111 \end{array}$ \Rightarrow Broadcast: 192.168.11.727 \rightarrow 725 IPs

192.168.11.728 /25

$\begin{array}{r} 1\ 000\ 0000 \\ 1\ 111\ 1111 \end{array}$ \Rightarrow Broadcast: 192.168.11.255 \rightarrow 725 IPs

\hookrightarrow 7 SubNetze nochmal teilen

192.168.11.728 /26

\Rightarrow Broadcast: 192.168.11.791 \rightarrow 62 IPs

192.168.11.792 /26

\Rightarrow Broadcast: 192.168.11.255 \rightarrow 62 IPs

Graphisch:



Nur in der Mitte durchschneiden und
NUR hinten durchschneiden, wegen Subnetz-Kosten
mit 1-sei von vorne nach hinten

✓

○

IPv4 Subnetzmasken mit variabler Länge (Beispiel)

Beispiel: Klasse-C-Netzwerk: 193.43.55.x

- Subnetzmaske für das Subnetz mit der ID 0 (100 Endsysteme):
 - 255.255.255.128
 - Adressen in diesem Subnetz: 193.43.55.0-127
- Subnetzmaske für die Subnetze mit den IDs 2 und 3 (je 50 Endsysteme):
 - 255.255.255.192
 - Adressen im Subnetz 2: 193.43.55.128-191
 - Adressen im Subnetz 3: 193.43.55.192-255

Diskussion Klassenbasierte Adressierung

Verteilung der Adressen:

- Durch zentrale Organisationen (z.B. IANA)
- Netzweise (also z.B. Klasse-B-Netz für ein Unternehmen)
- Relativ chaotisch:
 - Zuteilung der numerisch nächsten netid an den nächsten Nachfrager

Diskussion Klassenbasierte Adressierung

Probleme:

- Verschwendungen von IP-Adressen:
 - Das Unternehmen könnte 2^{16} , also mehr als 65.000 Adressen vergeben
 - Nur ein Bruchteil davon wird genutzt
- Routing-Tabellen werden schnell sehr groß:
 - Ein Eintrag für jede netid
 - Keine Möglichkeit, Einträge zusammenzufassen
- Routing-Tabellen müssen mit hoher Frequenz aufgefrischt werden
 - Immer, wenn ein Netzwerk hinzukommt, wegfällt oder sich verändert, muss dies im ganzen Internet bekanntgegeben werden

IPv4 Klassenlose Adressierung

Idee:

- Die Aufteilung nach netid/hostid wird **immer** explizit per Subnetzmaske durchgeführt
- Keine explizite Unterscheidung zwischen netid und subnetid

Schreibweise:

- a.b.c.d/x, wobei x die Länge der netid (hier auch Präfix oder englisch prefix) bestimmt
- Alternative Schreibweise zur Subnetzmaske
- Beispiel:
 - 192.48.96.0/23
 - 11000000.00110000.01100000.00000000 (**netidhostid**)

IPv4 Klassenlose Adressierung

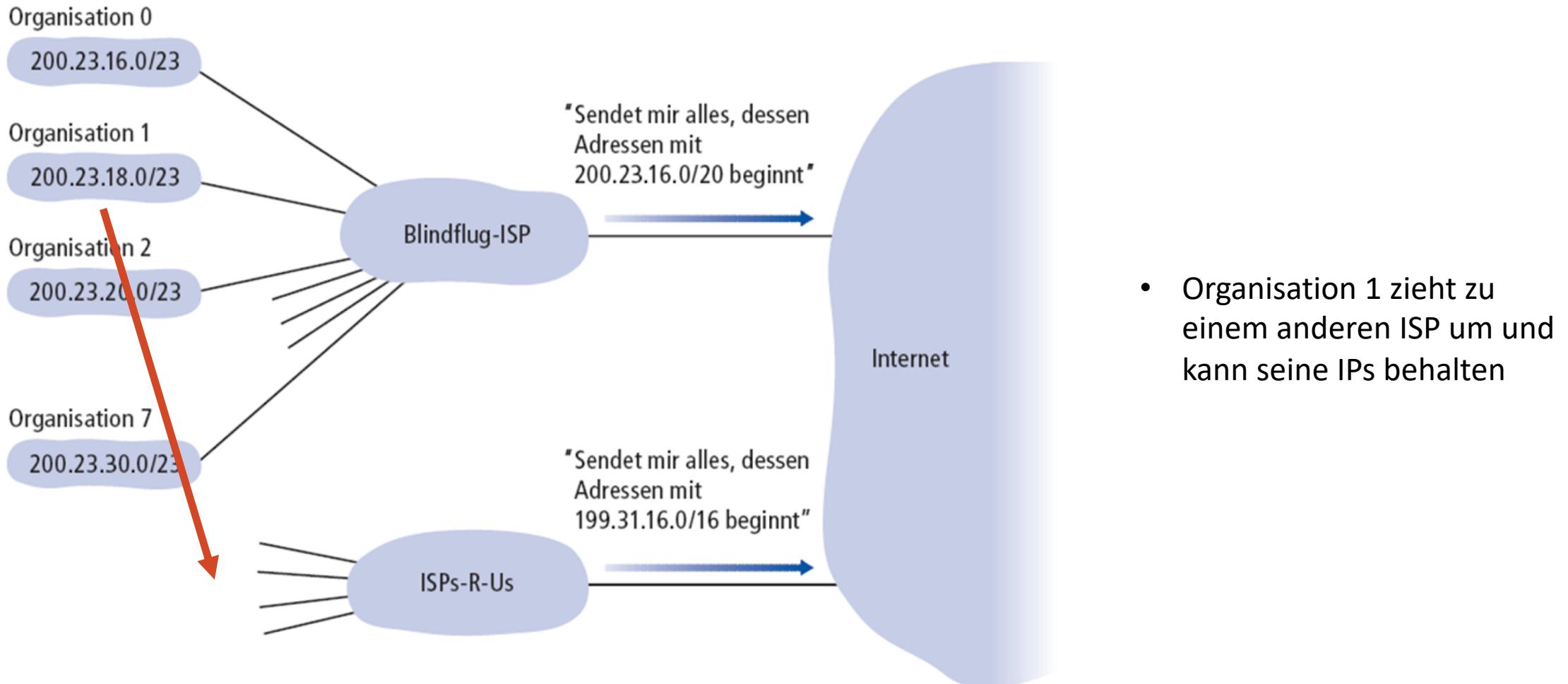
Bei der klassenlosen Adressierung werden zusammenhängende Adressbereiche von der Internet Assigned Numbers Authority (IANA) an die Regional Internet Registries (RIR) vergeben:

- APNIC (Asia Pacific Network Information Centre) - Asien/Pazifik
- ARIN (American Registry for Internet Numbers) - Nordamerika und Afrika südlich der Sahara
- LACNIC (Regional Latin-American and Caribbean IP Address Registry) – Lateinamerika und einige karibische Inseln
- RIPE NCC (Réseaux IP Européens) - Europa, Mittlerer Osten, Zentralasien und afrikanische Länder nördlich des Äquators

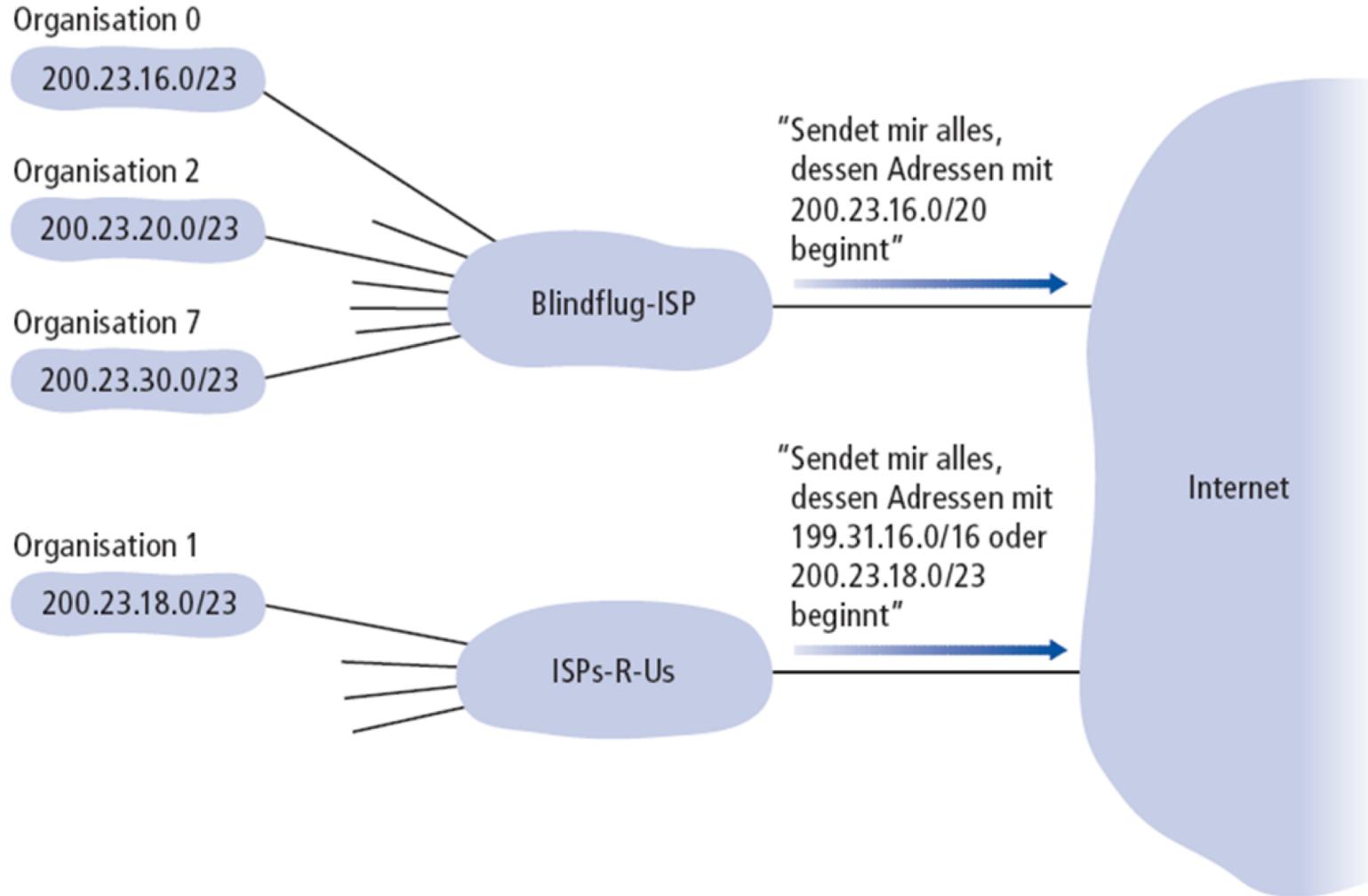
IPv4 Klassenlose Adressierung

- Die RIRs vergeben zusammenhängende Adressbereiche an ISPs
- ISPs vergeben zusammenhängende Adressbereiche an ihre Kunden
- Dadurch: Aggregation in Routing-Tabellen möglich
- Classless Interdomain Routing (CIDR) verwendet dies

Beispiel Klassenlose Adressierung (Longest-Prefix-Matching)



Beispiel Klassenlose Adressierung (Longest-Prefix-Matching)



- Router verwenden die sog. **Longest-Prefix-Matching-Regel** (längstes übereinstimmendes Präfix), so dass der Umzug von Organisation 1 technisch einfach zu realisieren ist.

Besondere IPv4-Adressen

Address Block	Present Use	Reference
0.0.0.0/8	„This“ Network	[RFC1700, page 4]
10.0.0.0/8	Private-Use Networks	[RFC1918]
14.0.0.0/8	Public-Data Networks	[RFC1700, page 181]
24.0.0.0/8	Cable Television Networks	---
39.0.0.0/8	Reserved but subject to allocation	[RFC1797]
127.0.0.0/8	Loopback	[RFC1700, page 5]
128.0.0.0/8	Reserved but subject to allocation	---
169.254.0.0/16	Link Local	---
172.16.0.0/12	Private-Use Networks	[RFC1918]

Besondere IPv4-Adressen

Address Block	Present Use	Reference
191.255.0.0/16	Reserved but subject to allocation	---
192.0.0.0/24	Reserved but subject to allocation	---
192.0.2.0/24	Test Net	---
192.88.99.0/24	6to4 Relay Anycast	[RFC3068]
192.168.0.0/16	Private-Use Networks	[RFC1918]
198.18.0.0/15	Network Interconnect Device Benchmark Testing	[RFC2544]
223.255.255.0/24	Reserved but subject to allocation	---
224.0.0.0/4	Multicast	[RFC3171]
240.0.0.0/4	Reserved for Future Use	[RFC1700, page 4]

Besondere IPv4-Adressen (praktisch relevant)

Address Block	Present Use	Reference
0.0.0.0/8	„This“ Network	[RFC1700, page 4]
10.0.0.0/8	Private-Use Networks	[RFC1918]
127.0.0.0/8	Loopback	[RFC1700, page 5]
169.254.0.0/16	Link Local	---
172.16.0.0/12	Private-Use Networks	[RFC1918]
192.168.0.0/16	Private-Use Networks	[RFC1918]
224.0.0.0/4	Multicast	[RFC3171]

Warum gibt es die, in RFC1918 definierten, „Private-Use“-Networks?

↳ Heimnetzwerke

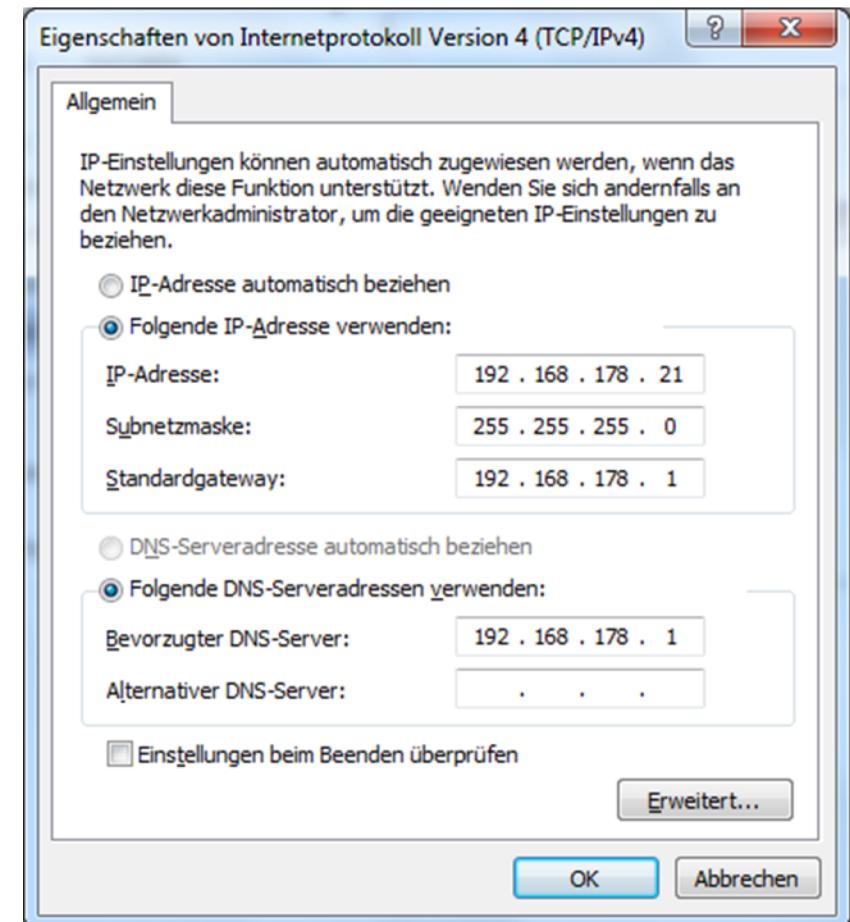
IP-Adressvergabe

Wie erhält ein Host seine IP-Adresse?

IP-Adressvergabe

Durch **manuelle** Konfiguration:

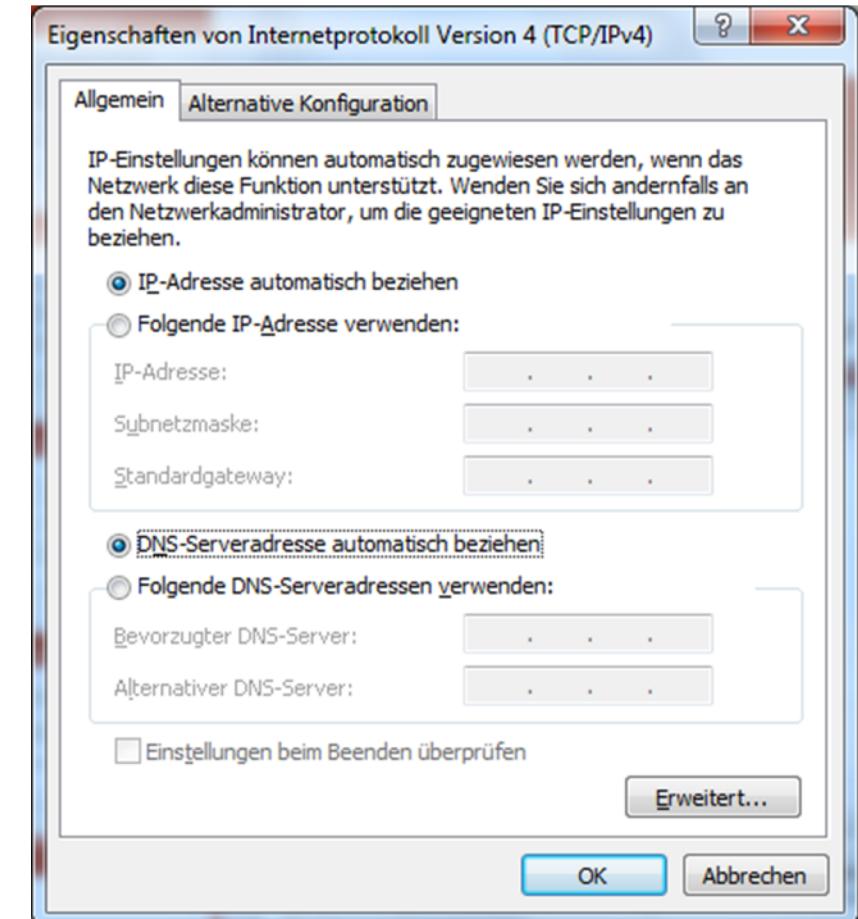
- IP-Adresse
- Subnetzmaske
- Weitere Parameter



IP-Adressvergabe

DHCP: Dynamic Host Configuration Protocol

- dynamisches Beziehen der Adresse von einem Server
- “Plug-and-Play”



DHCP – Dynamic Host Configuration Protocol

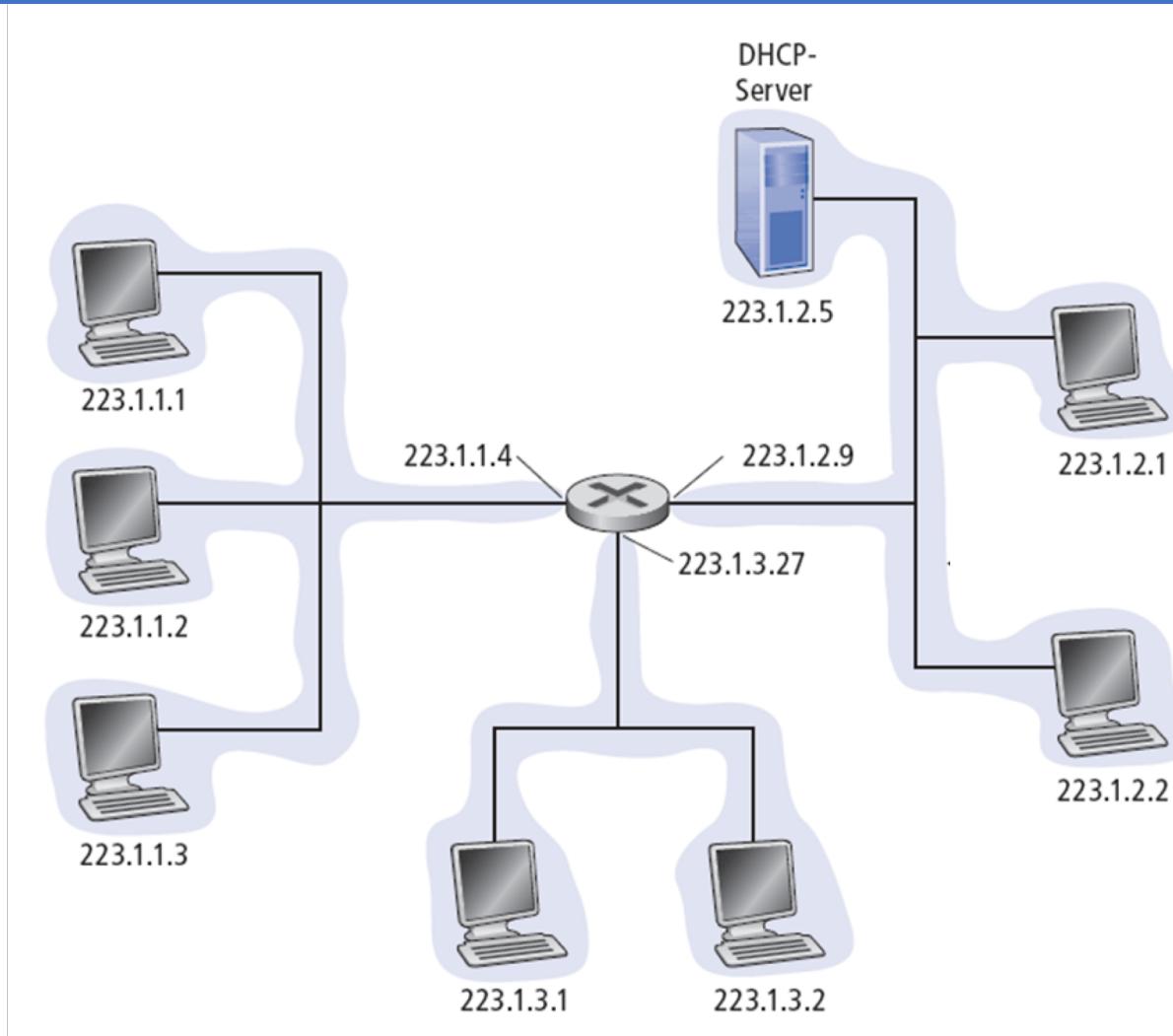
Ziele von DHCP

- Automatische Vergabe von Adressen und Parametern
- Keine Konfiguration der Endsysteme notwendig
- Unterstützung von mobilen Benutzern

Prinzipieller Ablauf

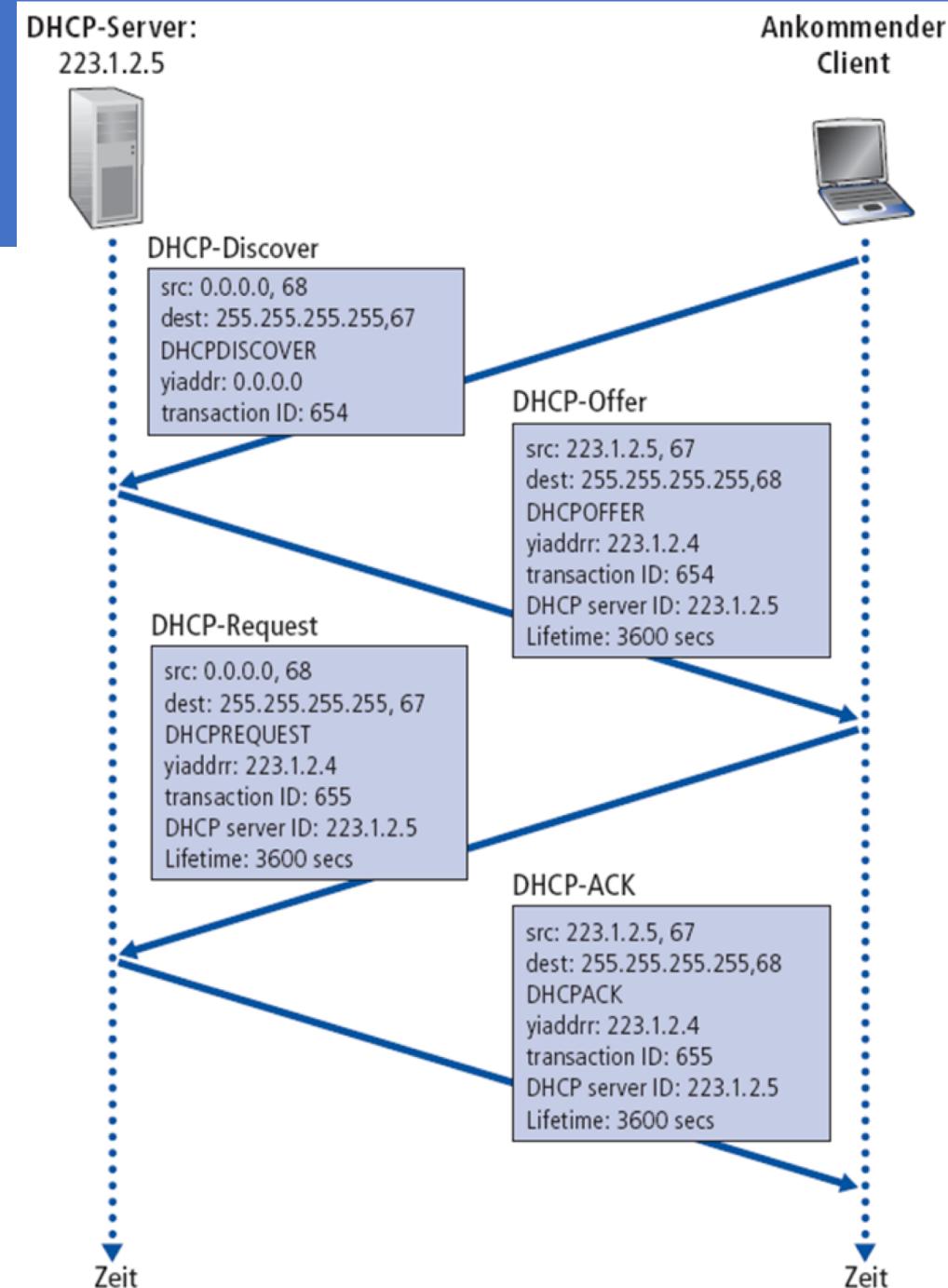
- Endsystem schickt eine DHCP-Discover-Nachricht per IP-Broadcast (Adresse 255.255.255.255)
- DHCP-Server antwortet mit einer DHCP-Offer-Nachricht
- Endsystem beantragt eine IP-Adresse: DHCP-Request-Nachricht
- DHCP-Server vergibt Adresse: DHCP-Ack-Nachricht

DHCP-Szenario



DHCP-Szenario

- DHCP verwendet UDP
- DHCP-Nachrichten werden an die MAC-Broadcast-Adresse geschickt
- Es gibt ein Feld, in dem eine eindeutige Kennung des Clients verpackt ist. Dies ist meist die MAC-Adresse.

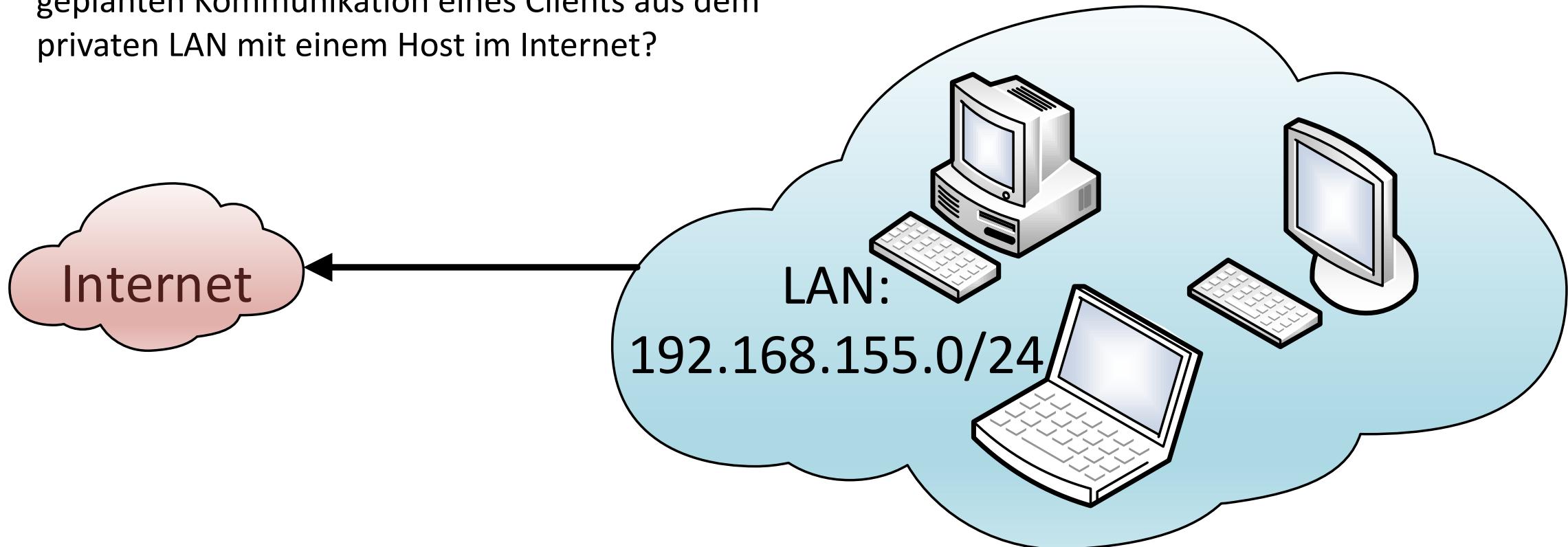


DHCP – Recherche

- Was passiert, wenn ein DHCP-Server seine Adressen komplett vergeben hat und ein weiterer Client kommt ins Netz?
- Wie werden Adressen wieder freigegeben, die vom DHCP-Server bezogen wurden, aber nun nicht mehr genutzt werden bzw. werden sollen?
- Wie wird sichergestellt, dass keine Adressen doppelt vergeben werden?
- Welche Konfigurationen lassen sich über DHCP an die Clients verteilen (Stichwort: DHCP-Optionen)?

Netzwerkzugriffe aus „privaten Netzwerken“

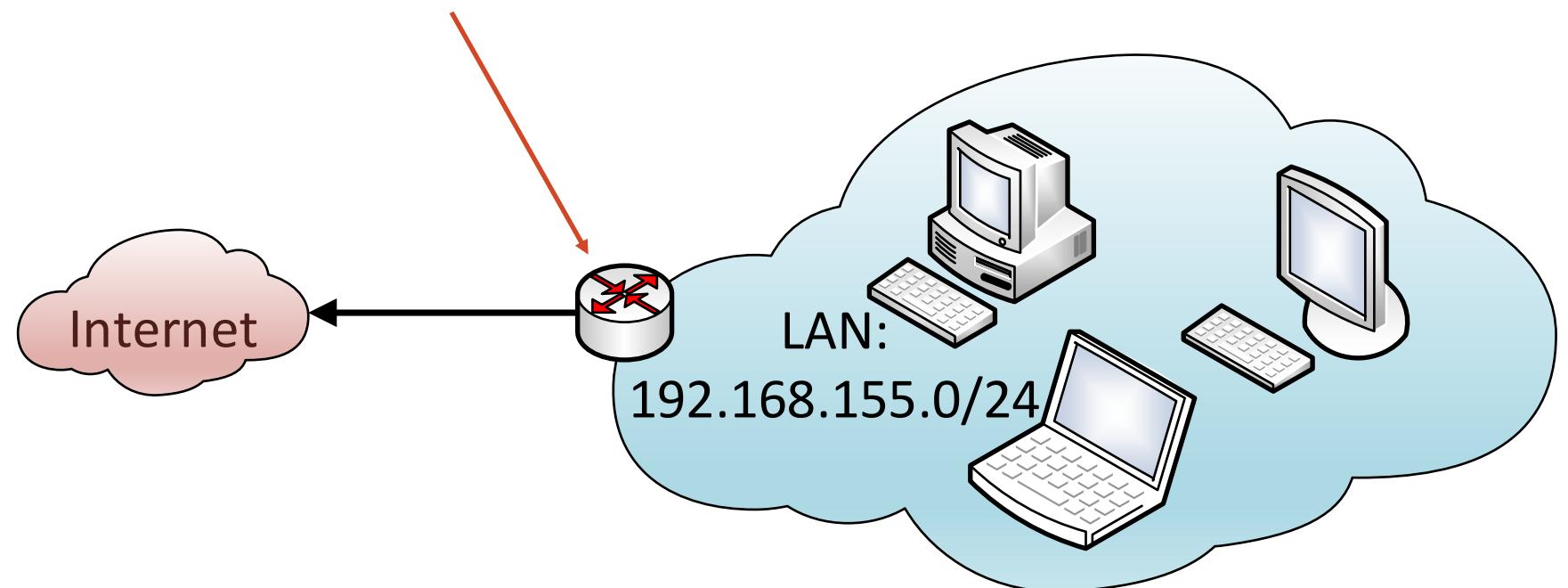
Welche Stolpersteine gibt es bei dieser geplanten Kommunikation eines Clients aus dem privaten LAN mit einem Host im Internet?



Netzwerkzugriffe aus „privaten Netzwerken“

Grundlegend stellt der ISP für den Zugriff auf das Internet mindestens eine nutzbare öffentliche IP-Adresse zur Verfügung.

Diese würde an einem Interface des Routers konfiguriert werden.



Network Address Translation (NAT)

Motivation:

- Häufig hat man nur eine IP-Adresse, aber mehrere Endsysteme
- Diese ist meist nur temporär (per DHCP) zugewiesen
- Man möchte bei einem Provider-Wechsel nicht die IP-Adressen der Endsysteme verändern
- IP-Adressen im eigenen Netzwerk sollen aus Sicherheitsgründen nicht vom Internet aus sichtbar sein
- Interne IP-Adressen sollen veränderbar sein, ohne dass der Rest des Internets darüber informiert werden muss

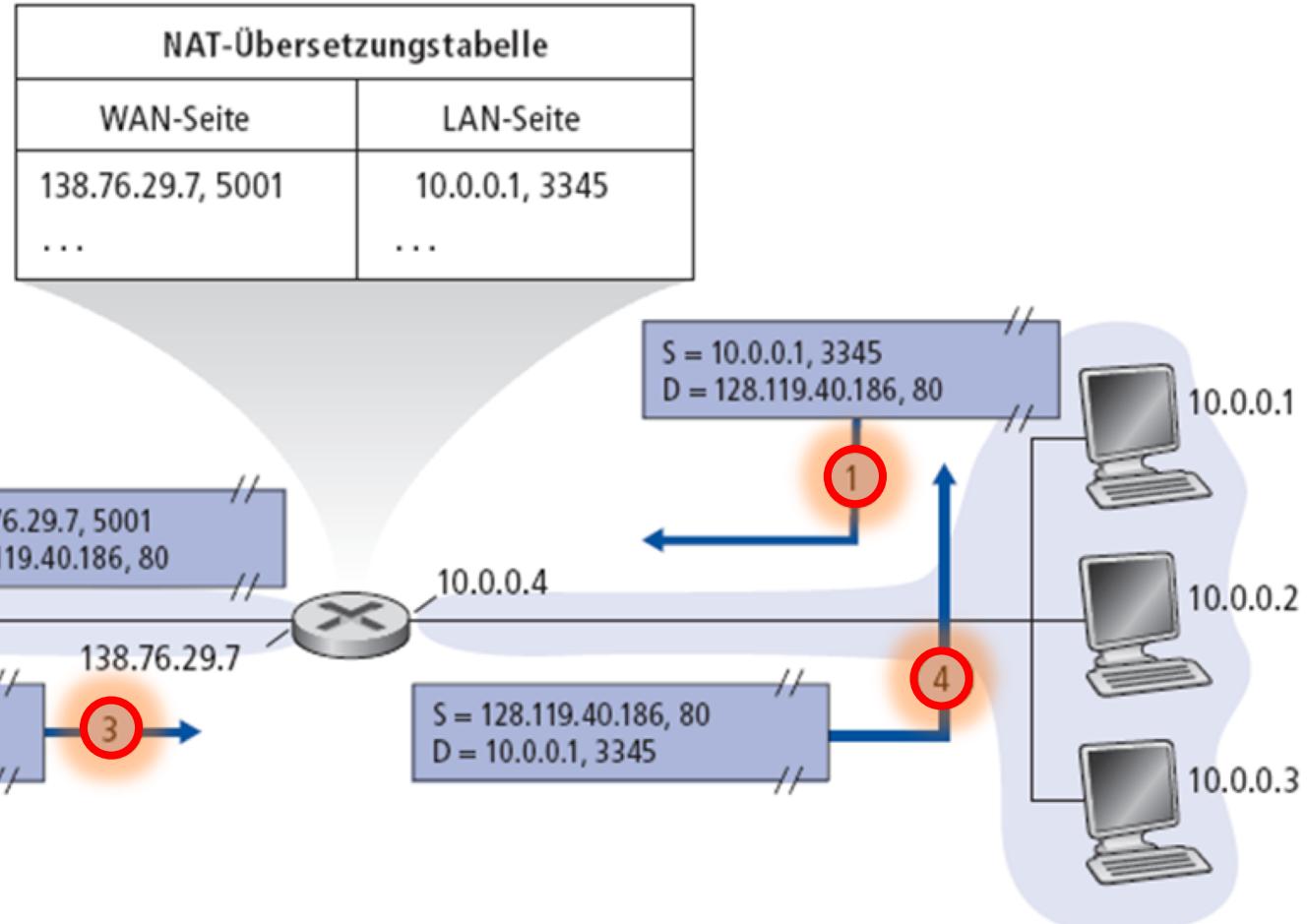
Network Address Translation (NAT)

Idee:

- Vergebe lokale (weltweit nicht eindeutige) Adressen („privater Bereich“) an die Systeme im eigenen Netzwerk
- Router zur Anbindung an das Internet übersetzt diese Adressen in eine gültige, weltweit eindeutige IP-Adresse
- Dazu wird die Adressierung auf der Transportschicht gebraucht (missbraucht): **Ports**

NAT

- Alle Datagramme, die das lokale Netz verlassen, haben die gleiche NAT-IP-Adresse als Absender: 138.76.29.7
- Unterscheidung durch Portnummern



NAT

Implementierung: Ein NAT-Router muss Folgendes tun:

- Ausgehende Datagramme: ersetze (Sender-IP-Adresse, Portnummer) im Absenderfeld für jedes ins Internet geleitete Datagramm durch (NAT-IP-Adresse, neue Portnummer)
 - Kommunikationspartner wird die Antworten an (NAT-IP-Adresse, neue Portnummer) schicken
- Speichere in einer NAT-Tabelle die Abbildung zwischen (Sender-IP-Adresse, Portnummer) und (NAT-IP-Adresse, neue Portnummer)
- Ankommende Datagramme: ersetze (NAT-IP-Adresse, neue Portnummer) im Empfängerfeld durch (Sender-IP-Adresse, Portnummer) aus der NAT-Tabelle

NAT

- Wieviele gleichzeitige Verbindungen lassen sich via NAT nutzen?
- Wodurch wird die Zahl der Verbindungen begrenzt?

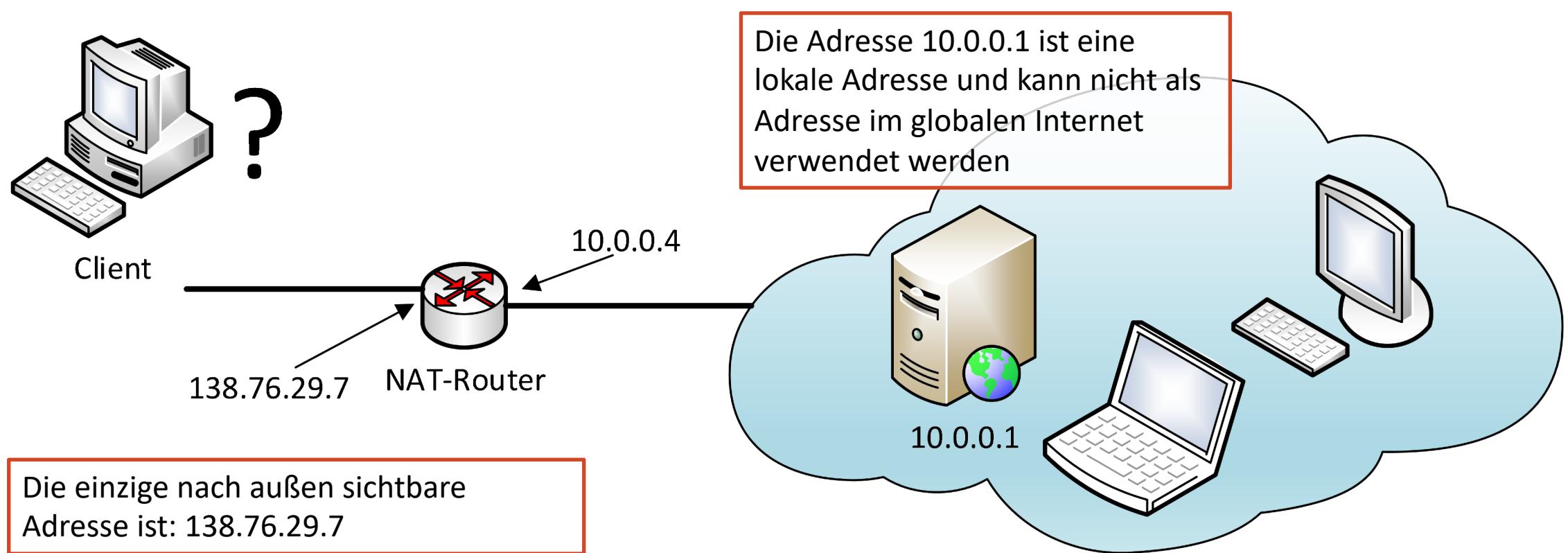
NAT

NAT ist nicht unumstritten:

- Router sollten nur Informationen der Schicht 3 verwenden
- Verletzung des sogenannten Ende-zu-Ende-Prinzips (end-to-end principle):
 - Transparente Kommunikation von Endsystem zu Endsystem, im Inneren des Netzes wird nicht an den Daten „herumgepfuscht“
 - Bei NAT: Der Anwendungsentwickler muss die Präsenz von NAT-Routern berücksichtigen.
Beispiele:
 - Verwenden der IP-Adresse als weltweit eindeutige Nummer
 - Verwenden von UDP
- NAT dient hauptsächlich der Bekämpfung der Adressknappheit im Internet. Dies sollte besser über IPv6 erfolgen

NAT Traversal (Durchqueren von NAT)

Der Client möchte den Server mit der Adresse 10.0.0.1 kontaktieren.

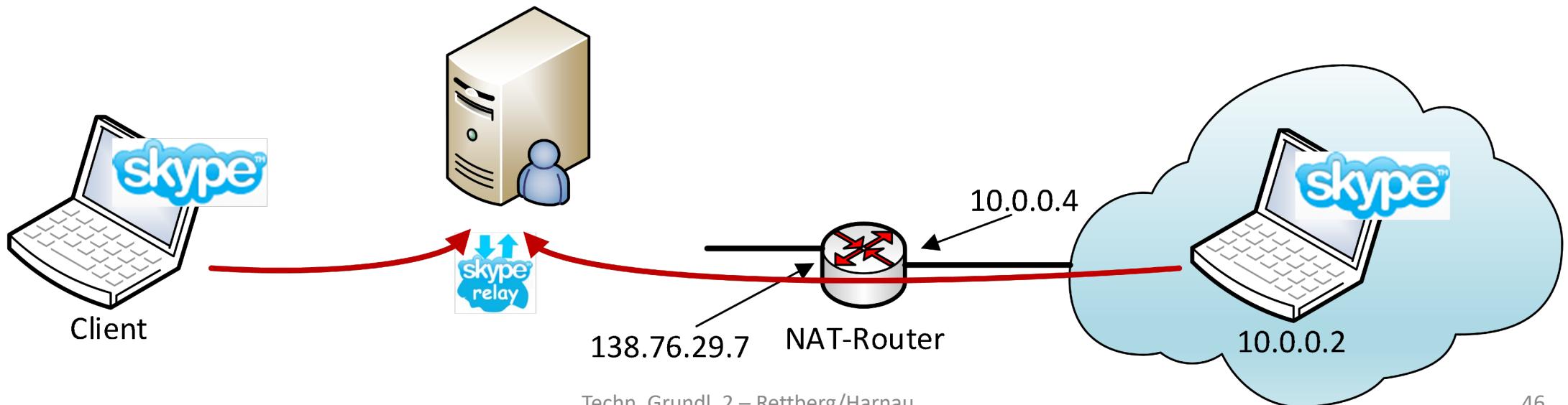


NAT Traversal

- **Lösung 1:** Statische Konfiguration von NAT, so dass eingehende Anfragen geeignet weitergeleitet werden
 - Beispiel: (123.76.29.7, Port 2500) wird immer an 10.0.0.1, Port 25000 weitergeleitet
- **Lösung 2:** Universal Plug and Play (UPnP) Internet Gateway Device (IGD) Protocol.
Dies ermöglicht dem Host hinter dem NAT Folgendes:
 - Herausfinden der öffentlichen IP-Adresse des NAT-Routers (138.76.29.7)
 - Kennenlernen existierender Abbildungen in der NAT-Tabelle
 - Einträge in die NAT-Tabelle einfügen oder aus ihr löschen→ Das heißt automatische Konfiguration von statischen NAT-Einträgen

NAT Traversal

- **Lösung 3:** Relaying (z.B. von Skype verwendet)
 - Server hinter einem NAT-Router baut eine Verbindung zu einem Relay auf (welches nicht hinter einem NAT-Router liegt)
 - Client baut eine Verbindung zum Relay auf
 - Relay leitet die Pakete vom Client zum Server und umgekehrt weiter

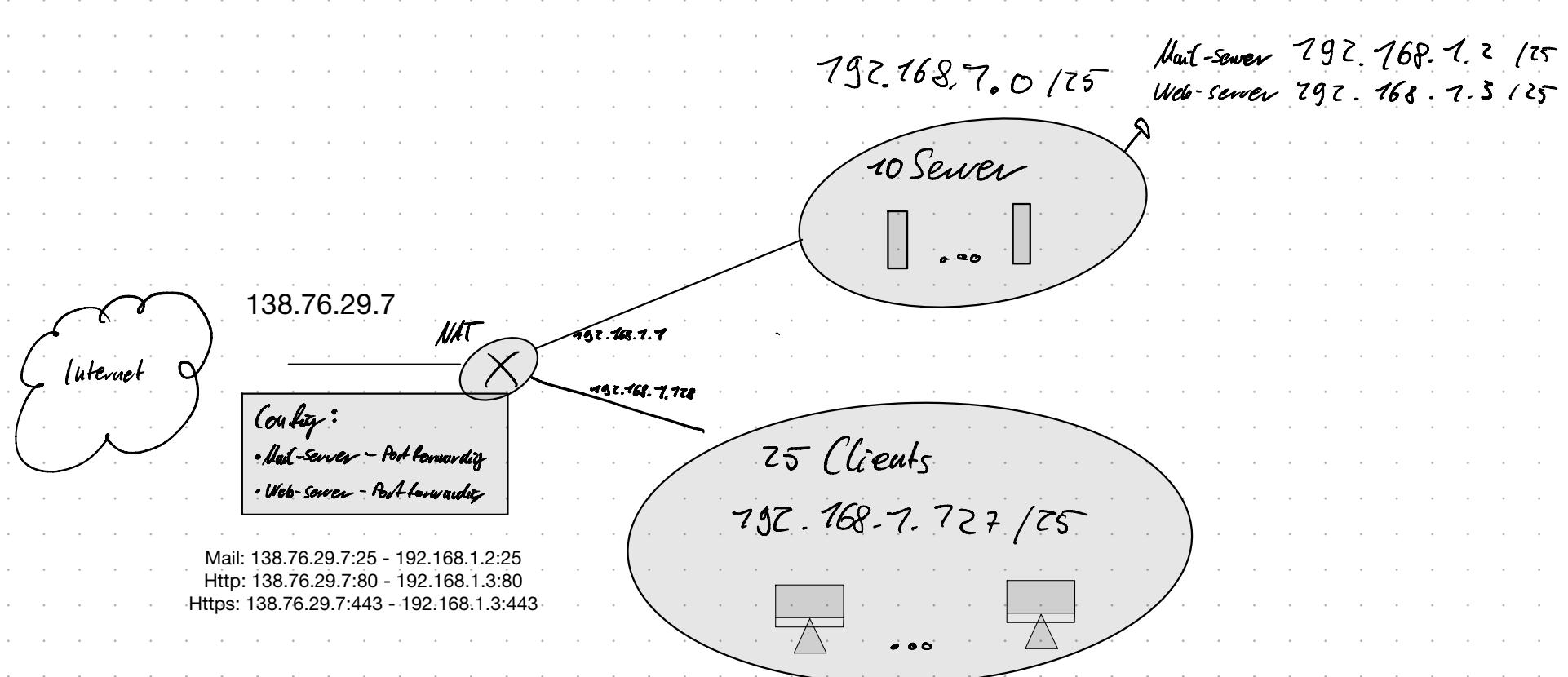


Gruppenarbeit: „Netzwerkdesign“

- Ein Unternehmen besitzt 25 Arbeitsplätze und 10 Server.
- Die Clients sollen nicht im gleichen Netz wie die Server liegen.
- Zwei Server sollen vom Internet aus erreicht werden können: Webserver und Mailserver.
- Sowohl die Server als auch die Arbeitsplätze sollen Zugang zum Internet bekommen.
- Vom ISP hat das Unternehmen eine nutzbare öffentliche IP (138.76.29.7) zugewiesen bekommen.

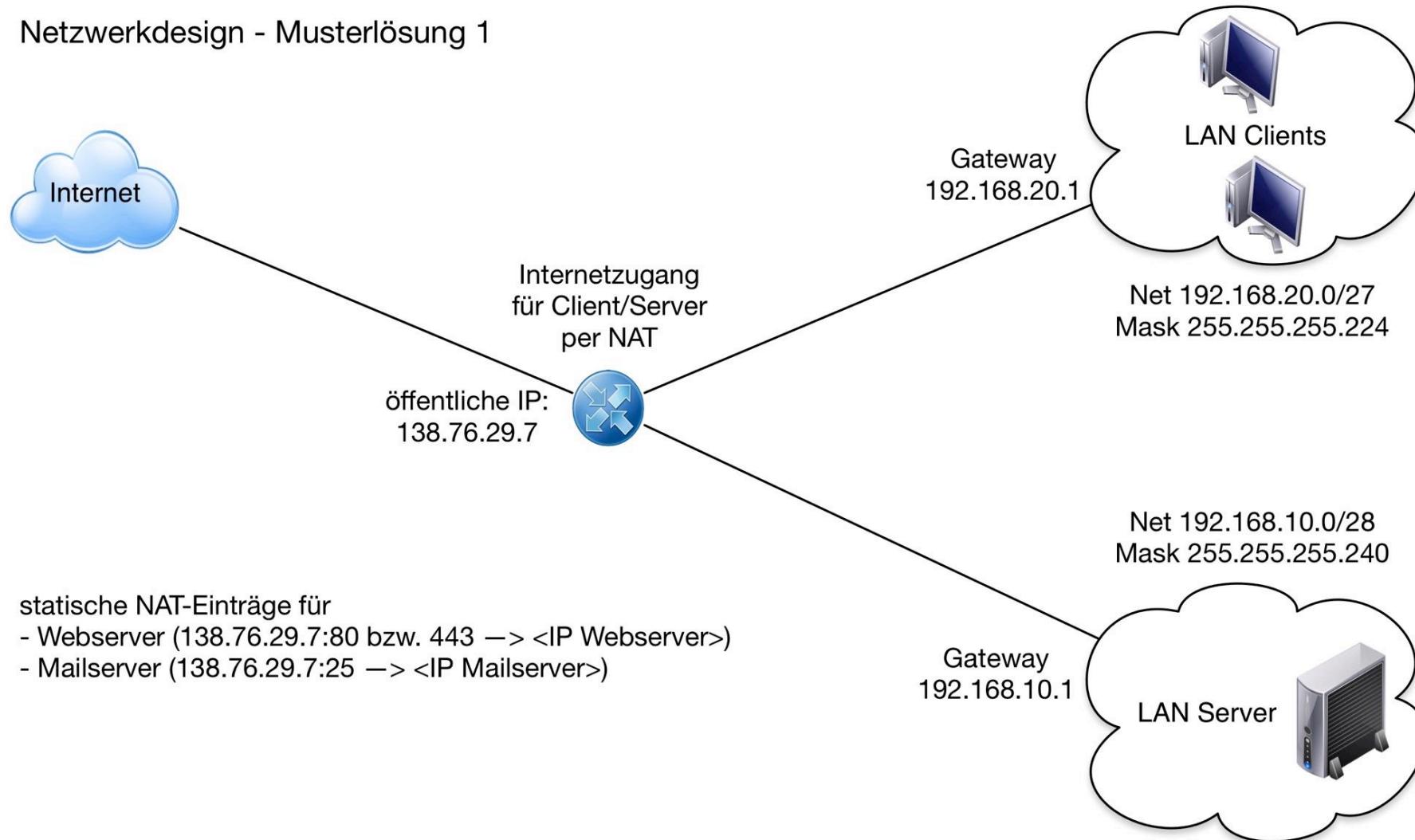
Erarbeiten Sie mithilfe des bisher erworbenen Wissens ein Netzwerkdesign, welches den obigen Anforderungen genügt:

- Schlagen Sie Netze inkl. möglicher Adressen vor.
- Plazieren Sie die Arbeitsplätze, Server und ggf. Router oder weitere Komponenten...



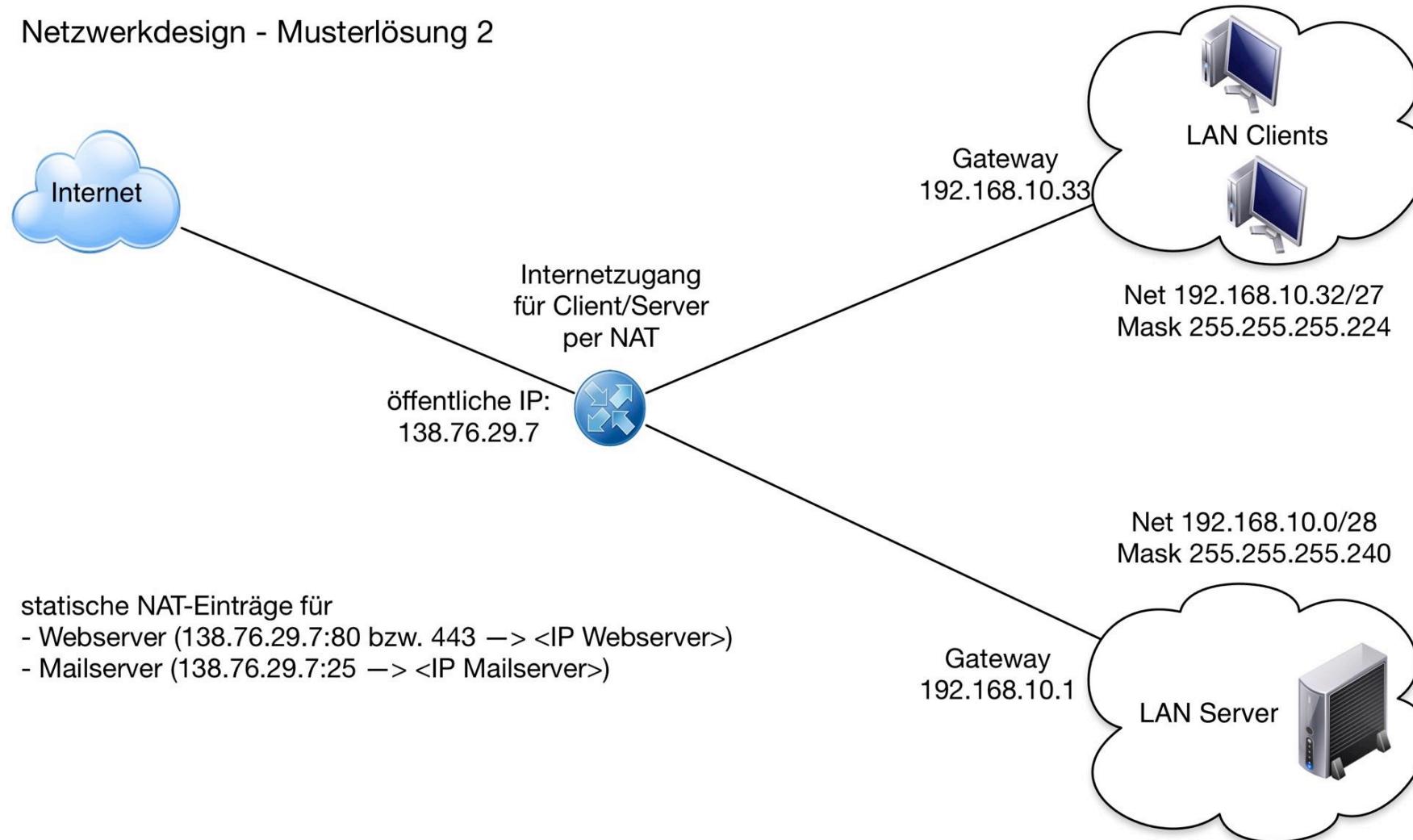
Netzdesign (2 Klasse C Netzwerke)

Netzwerkdesign - Musterlösung 1



Netzdesign (1 Klasse C Netzwerk)

Netzwerkdesign - Musterlösung 2



Netzdesign (1 Klasse C Netzwerk, DMZ)

Netzwerkdesign - Musterlösung 3

