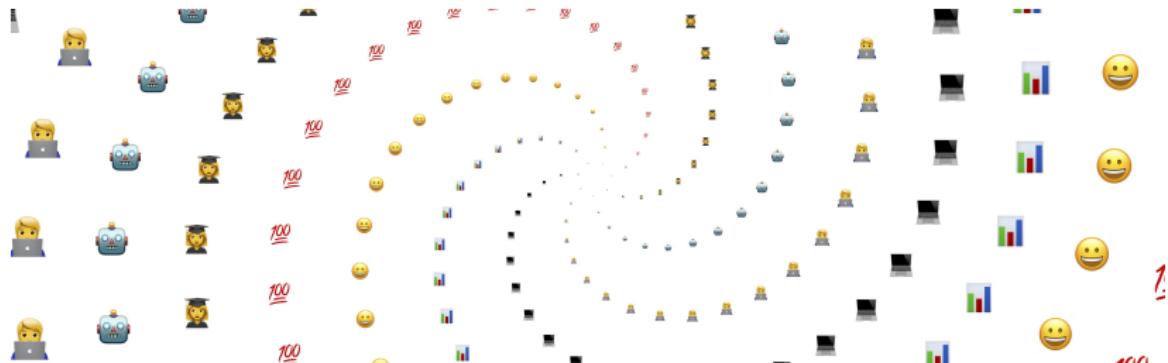


# Internet

Cyril Wendl

Fachschaft Informatik  
Kantonsschule im Lee



Kantonsschule im Lee

# Netzwerk-Spiel (ca. 5')

Sie sind ein Computer im Internet und möchten gerne eine Nachricht an einen anderen Computer senden:

③ Von:	192.168.0.
③ An:	192.168.0.
② Sequenz:	
① Nachricht:	

## Anleitung

- ▶ (Bereits gemacht): Notieren Sie sich eine einzigartige IP-Adresse für Ihren Computer, startend mit 192.168.0.xyz (z.B. 192.168.0.3), wobei xyz  $\in [0, 255]$



# Netzwerk-Spiel (ca. 5')

Sie sind ein Computer im Internet und möchten gerne eine Nachricht an einen anderen Computer senden:

③ Von:	192.168.0.
③ An:	192.168.0.
② Sequenz:	
① Nachricht:	

## Anleitung

- ▶ (Bereits gemacht): Notieren Sie sich eine einzigartige IP-Adresse für Ihren Computer, startend mit 192.168.0.xyz (z.B. 192.168.0.3), wobei xyz  $\in [0, 255]$
- ① (Vorderseite) Schreiben Sie eine Nachricht (max. 10 Zeichen pro Karte)



# Netzwerk-Spiel (ca. 5')

Sie sind ein Computer im Internet und möchten gerne eine Nachricht an einen anderen Computer senden:

③ Von:	192.168.0.
③ An:	192.168.0.
② Sequenz:	
① Nachricht:	

## Anleitung

- ▶ (Bereits gemacht): Notieren Sie sich eine einzigartige IP-Adresse für Ihren Computer, startend mit 192.168.0.xyz (z.B. 192.168.0.3), wobei xyz  $\in [0, 255]$
- ① (Vorderseite) Schreiben Sie eine Nachricht (max. 10 Zeichen pro Karte)
- ② (Vorderseite) Nummerieren Sie die Karten in der Reihenfolge, in der sie gesendet werden sollen (Sequenz)



# Netzwerk-Spiel (ca. 5')

Sie sind ein Computer im Internet und möchten gerne eine Nachricht an einen anderen Computer senden:

③ Von:	192.168.0.
③ An:	192.168.0.
② Sequenz:	
① Nachricht:	

## Anleitung

- ▶ (Bereits gemacht): Notieren Sie sich eine einzigartige IP-Adresse für Ihren Computer, startend mit 192.168.0.xyz (z.B. 192.168.0.3), wobei xyz ∈ [0, 255]
- ① (Vorderseite) Schreiben Sie eine Nachricht (max. 10 Zeichen pro Karte)
- ② (Vorderseite) Nummerieren Sie die Karten in der Reihenfolge, in der sie gesendet werden sollen (Sequenz)
- ③ (Rückseite) Fügen Sie die IP-Adressen der Absender- und Empfänger-Computer hinzu.



# Netzwerk-Spiel (ca. 5')

Sie sind ein Computer im Internet und möchten gerne eine Nachricht an einen anderen Computer senden:

③ Von:	192.168.0.
③ An:	192.168.0.
② Sequenz:	
① Nachricht:	

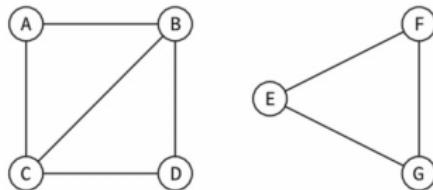
## Anleitung

- ▶ (Bereits gemacht): Notieren Sie sich eine einzigartige IP-Adresse für Ihren Computer, startend mit 192.168.0.xyz (z.B. 192.168.0.3), wobei xyz ∈ [0, 255]
- ① (Vorderseite) Schreiben Sie eine Nachricht (max. 10 Zeichen pro Karte)
- ② (Vorderseite) Nummerieren Sie die Karten in der Reihenfolge, in der sie gesendet werden sollen (Sequenz)
- ③ (Rückseite) Fügen Sie die IP-Adressen der Absender- und Empfänger-Computer hinzu.
- ➡ Übergeben Sie die Karten dem „Router“ (= „Postbote“)



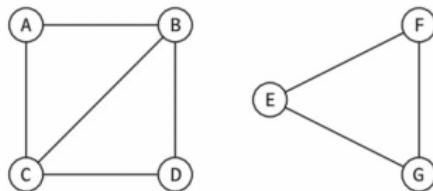
# Was sind Netzwerke?

- ▶ Graphen:



# Was sind Netzwerke?

- ▶ Graphen:

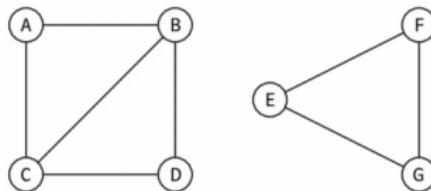


- ▶ Im Alltag:



# Was sind Netzwerke?

- ▶ Graphen:

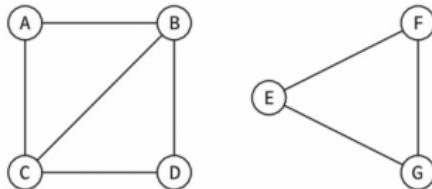


- ▶ Im Alltag:
  - ▶ Kommunikations-Netzwerke: Die Post, DHL, Planzer, ...



# Was sind Netzwerke?

- ▶ Graphen:

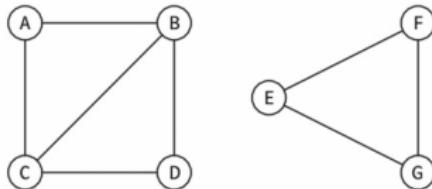


- ▶ Im Alltag:
  - ▶ Kommunikations-Netzwerke: Die Post, DHL, Planzer, ...
  - ▶ Soziale Netzwerke: TikTok, Youtube, Instagram, ...



# Was sind Netzwerke?

- ▶ Graphen:



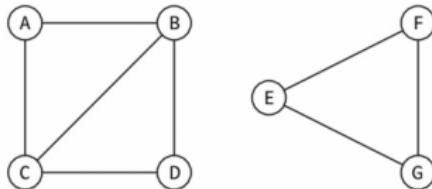
- ▶ Im Alltag:

- ▶ Kommunikations-Netzwerke: Die Post, DHL, Planzer, ...
- ▶ Soziale Netzwerke: TikTok, Youtube, Instagram, ...
- ▶ Transport-Netzwerke: SBB, Flixbus, ...



# Was sind Netzwerke?

- ▶ Graphen:



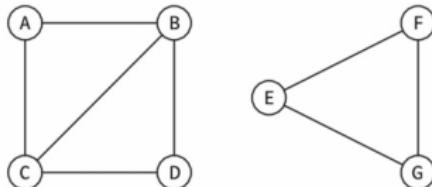
- ▶ Im Alltag:

- ▶ Kommunikations-Netzwerke: Die Post, DHL, Planzer, ...
- ▶ Soziale Netzwerke: TikTok, Youtube, Instagram, ...
- ▶ Transport-Netzwerke: SBB, Flixbus, ...
- ▶ Shopping-Netzwerke: Aliexpress, Uber, ...



# Was sind Netzwerke?

- ▶ Graphen:



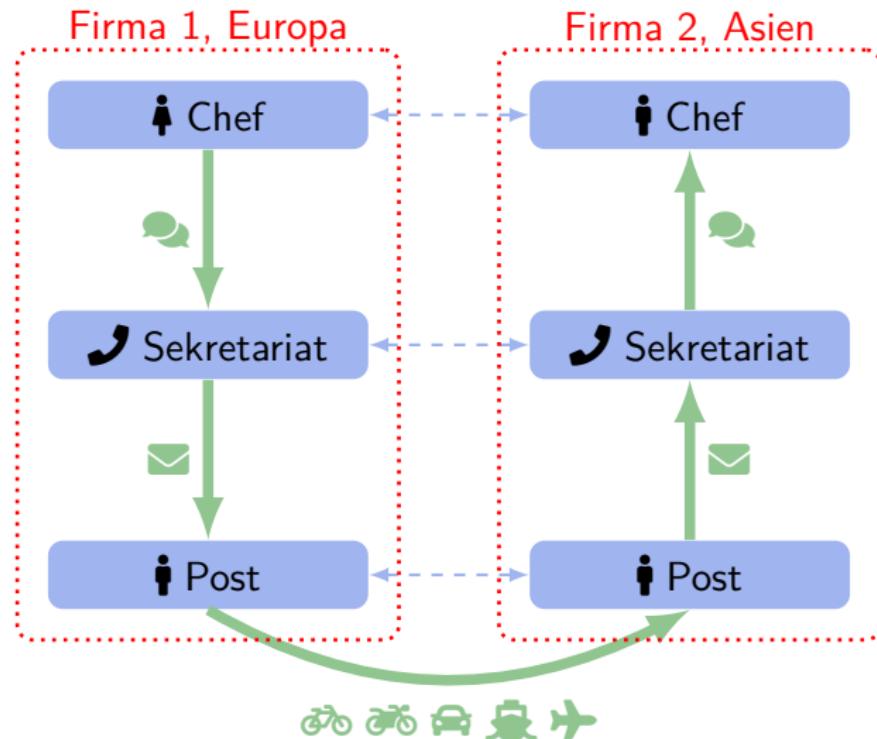
- ▶ Im Alltag:

- ▶ Kommunikations-Netzwerke: Die Post, DHL, Planzer, ...
- ▶ Soziale Netzwerke: TikTok, Youtube, Instagram, ...
- ▶ Transport-Netzwerke: SBB, Flixbus, ...
- ▶ Shopping-Netzwerke: Aliexpress, Uber, ...
- ▶ Computer-Netzwerke: **Das Internet**



# Schichtenmodell

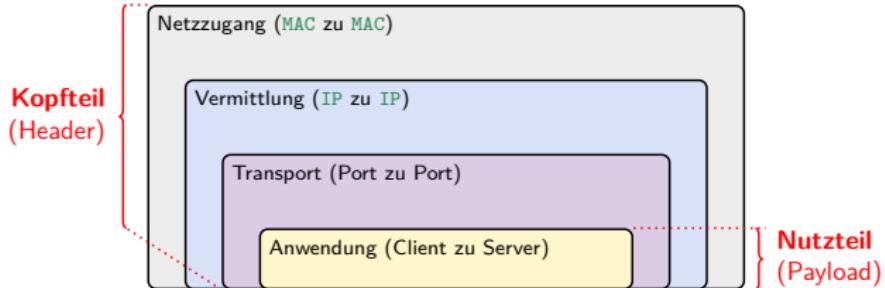
Analogie



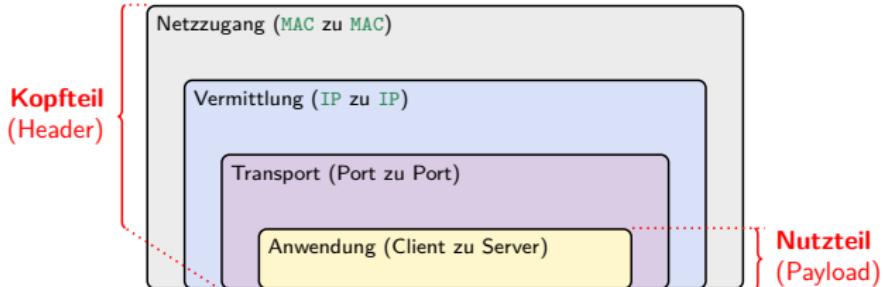
# Schichtenmodell: Übersicht

#	Schicht (OSI)	Schicht (TCP / IP)	Aufgabe	Informationsform	Begriffe / Protokolle
7	Anwendung <i>Application</i>	Anwendung <i>Application</i>	Hilft bei der Identifizierung des Clients und synchronisiert die Kommunikation.	Nachricht	<a href="#">HTTP</a> , <a href="#">HTTPS</a> , <a href="#">DNS</a> , <a href="#">SMTP</a> , <a href="#">FTP</a> ...
6	Darstellung <i>Presentation</i>	Anwendung <i>Application</i>	Daten von der Anwendungsschicht werden extrahiert und für die Übertragung in das erforderliche Format gebracht.	Nachricht	<a href="#">TLS</a> , <a href="#">SSL</a> , <a href="#">ASCII</a> , <a href="#">UTF-8</a> , ...
5	Sitzung <i>Session</i>	Anwendung <i>Application</i>	Stellt Verbindung her, hält diese aufrecht, gewährleistet Authentifizierung und sichert die Sicherheit.	Nachricht (oder verschlüsselte Nachricht)	<a href="#">TLS</a> Handshake
4	Transport <i>Transport</i>	Transport <i>Transport</i>	Nimmt Dienst von der Vermittlungsschicht und stellt ihn der Anwendungsschicht zur Verfügung.	Segment	<a href="#">TCP</a> , <a href="#">UDP</a> , Ports...
3	Vermittlung <i>Network</i>	Internet <i>Internet</i>	Übertragung von Daten von einem Host zu einem anderen, der sich in unterschiedlichen Netzwerken befindet.	Paket	<a href="#">IPv4</a> , <a href="#">IPv6</a> , <a href="#">ICMP</a> , Router, Gateway
2	Sicherung <i>Data Link</i>	Netzzugriff <i>Data Link</i>	Übermittlung von Nachrichten von Knoten zu Knoten.	Frame	<a href="#">ARP</a> , <a href="#">MAC</a> , Switch, Bridge, Ethernet, WiFi, ...
1	Bitübertragung <i>Physical</i>	Netzzugriff <i>Data Link</i>	Herstellung physischer Verbindungen zwischen Geräten.	Bits	Hub, Repeater, Modem, Kabel...

# Schichten & Protokolle



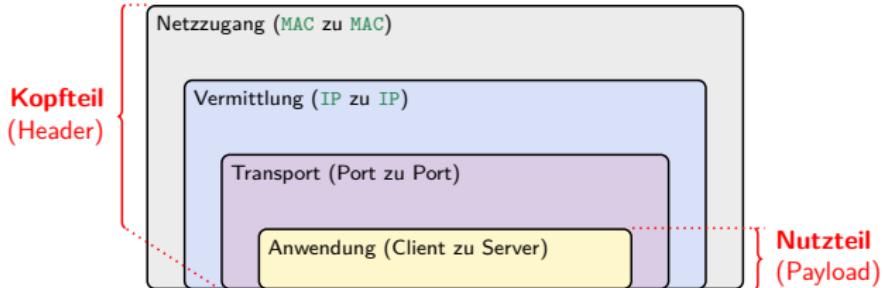
# Schichten & Protokolle



Von:	192.168.0.3	Von:	192.168.0.3
An:	192.168.0.11	An:	192.168.0.11
Sequenz:	1	Sequenz:	2
Nachricht (max. 10 Zeichen):	Hi Bob, wi	Nachricht (max. 10 Zeichen):	e geht's?



# Schichten & Protokolle

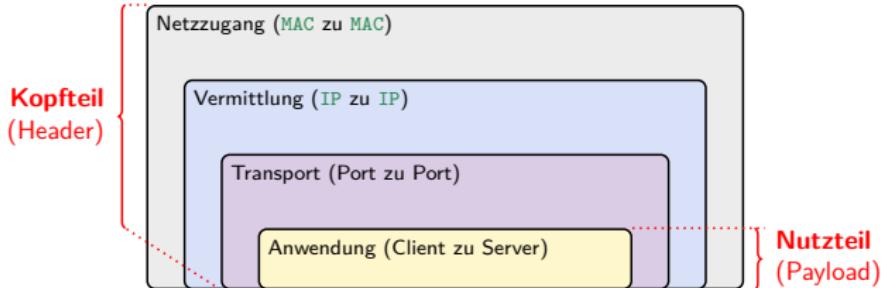


## Schichten

- Z.B. Anwendungs-, Transport-, Vermittlungs-, Netzzugangsschicht



# Schichten & Protokolle

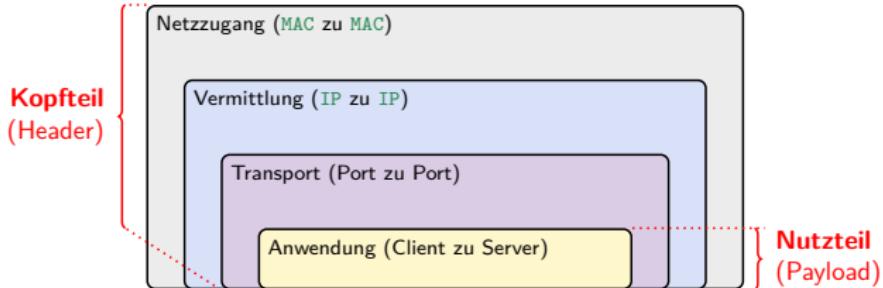


## Schichten

- ▶ Z.B. Anwendungs-, Transport-, Vermittlungs-, Netzzugangsschicht
- ▶ Erlauben es, spezialisierte Aufgaben in einzelne Schichten aufzuteilen: Z.B. Übermittlung von Signalen, Sprache-Regeln etc.



# Schichten & Protokolle

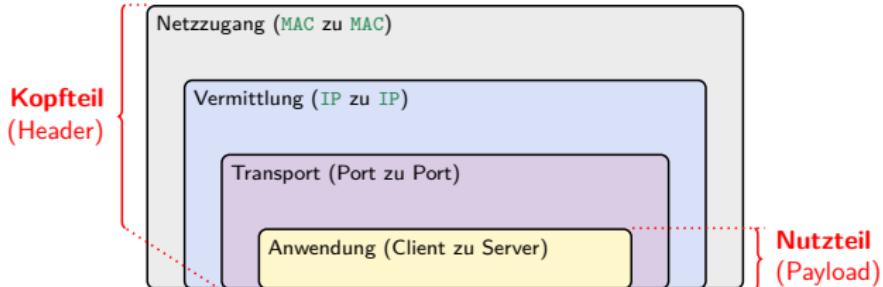


## Schichten

- ▶ Z.B. Anwendungs-, Transport-, Vermittlungs-, Netzzugangsschicht
- ▶ Erlauben es, spezialisierte Aufgaben in einzelne Schichten aufzuteilen: Z.B. Übermittlung von Signalen, Sprache-Regeln etc.



# Schichten & Protokolle



## Schichten

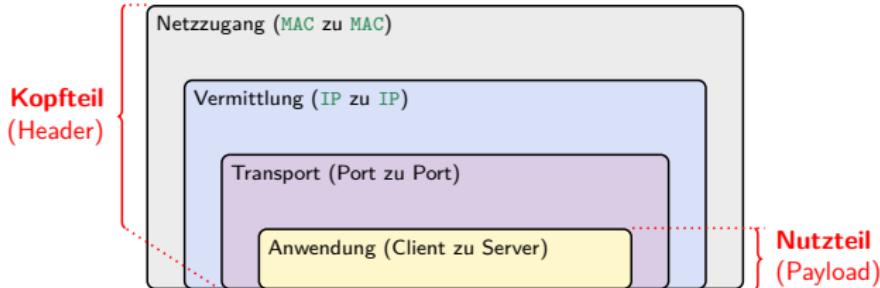
- Z.B. Anwendungs-, Transport-, Vermittlungs-, Netzzugangsschicht
- Erlauben es, spezialisierte Aufgaben in einzelne Schichten aufzuteilen: Z.B. Übermittlung von Signalen, Sprache-Regeln etc.

## Protokolle

- Definiert Ablauf einer Kommunikation, Aufbau der Datenpakete



# Schichten & Protokolle



## Schichten

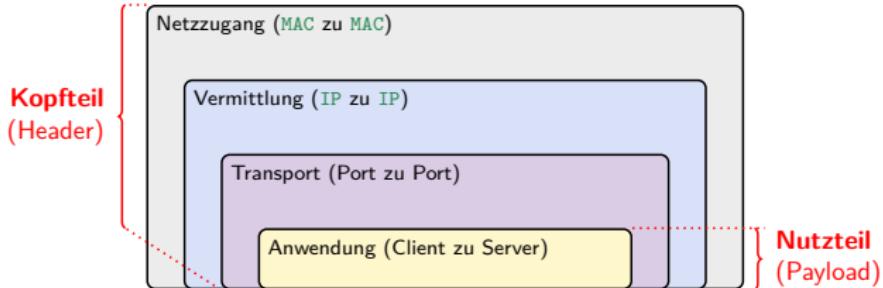
- Z.B. Anwendungs-, Transport-, Vermittlungs-, Netzzugangsschicht
- Erlauben es, spezialisierte Aufgaben in einzelne Schichten aufzuteilen: Z.B. Übermittlung von Signalen, Sprache-Regeln etc.

## Protokolle

- Definiert Ablauf einer Kommunikation, Aufbau der Datenpakete
- Gibt es auf jeder Ebene des Schichtenmodells



# Schichten & Protokolle



## Schichten

- Z.B. Anwendungs-, Transport-, Vermittlungs-, Netzzugangsschicht
- Erlauben es, spezialisierte Aufgaben in einzelne Schichten aufzuteilen: Z.B. Übermittlung von Signalen, Sprache-Regeln etc.

## Protokolle

- Definiert Ablauf einer Kommunikation, Aufbau der Datenpakete
- Gibt es auf jeder Ebene des Schichtenmodells
- Analogie: Begrüßungsformen, *small talk*, etc.



# Schichtenmodell

## Vorteile

- ▶ **Modularität:** Aufbau auf bestehenden Schichten



# Schichtenmodell

## Vorteile

- ▶ **Modularität:** Aufbau auf bestehenden Schichten
- ▶ **Austauschbarkeit / Flexibilität**



# Schichtenmodell

## Vorteile

- ▶ **Modularität:** Aufbau auf bestehenden Schichten
- ▶ **Austauschbarkeit / Flexibilität**
  - ▶ Übertragung über Kabel, WLAN etc. kann ausgetauscht werden, ohne dass darüberliegende Schichten tangiert werden



# Schichtenmodell

## Vorteile

- ▶ **Modularität:** Aufbau auf bestehenden Schichten
- ▶ **Austauschbarkeit / Flexibilität**
  - ▶ Übertragung über Kabel, WLAN etc. kann ausgetauscht werden, ohne dass darüberliegende Schichten tangiert werden
  - ▶ Analogie: Anstelle von Lastwagen kann Motorrad verwendet werden



# Netzwerk-Spiel

## Diskussion

- ▶ Was sind die Aufgaben der einzelnen Schichten?



# Netzwerk-Spiel

## Diskussion

- ▶ Was sind die Aufgaben der einzelnen Schichten?
  - ▶ Anwendungsschicht



# Netzwerk-Spiel

## Diskussion

- ▶ Was sind die Aufgaben der einzelnen Schichten?
  - ▶ Anwendungsschicht
  - ▶ Transportschicht



# Netzwerk-Spiel

## Diskussion

- ▶ Was sind die Aufgaben der einzelnen Schichten?
  - ▶ Anwendungsschicht
  - ▶ Transportschicht
  - ▶ Internetschicht



# Netzwerk-Spiel

## Diskussion

- ▶ Was sind die Aufgaben der einzelnen Schichten?
  - ▶ Anwendungsschicht
  - ▶ Transportschicht
  - ▶ Internetschicht
  - ▶ Netzzugangsschicht



# Netzwerk-Spiel

## Diskussion

- ▶ Was sind die Aufgaben der einzelnen Schichten?
  - ▶ Anwendungsschicht
  - ▶ Transportschicht
  - ▶ Internetschicht
  - ▶ Netzzugangsschicht
- ▶ Sind Pakete immer angekommen?



# Netzwerk-Spiel

## Diskussion

- ▶ Was sind die Aufgaben der einzelnen Schichten?
  - ▶ Anwendungsschicht
  - ▶ Transportschicht
  - ▶ Internetschicht
  - ▶ Netzzugangsschicht
- ▶ Sind Pakete immer angekommen?
- ▶ Wo waren die Flaschenhälse bei der Übermittlung der Nachrichten?



# Netzwerk-Spiel

## Diskussion

- ▶ Was sind die Aufgaben der einzelnen Schichten?
  - ▶ Anwendungsschicht
  - ▶ Transportschicht
  - ▶ Internetschicht
  - ▶ Netzzugangsschicht
- ▶ Sind Pakete immer angekommen?
- ▶ Wo waren die Flaschenhälse bei der Übermittlung der Nachrichten?
- ▶ Was ist der Unterschied zwischen einem Paket und einer Nachricht?



# Schichtenmodell

Video

Youtube-Video (13')



- ▶ Was ist eine URL?
  - ▶ Was macht „Mr. IP“?
  - ▶ Was macht der Router?
  - ▶ Was macht die Firewall?
  - ▶ Was macht der Proxy Server?
  - ▶ Was macht der „Ping of Death“? 💀
- 3-4er-Gruppen, ca. 5'



Kantonsschule im Lee

# Netzwerk-Spiel: Internetschicht

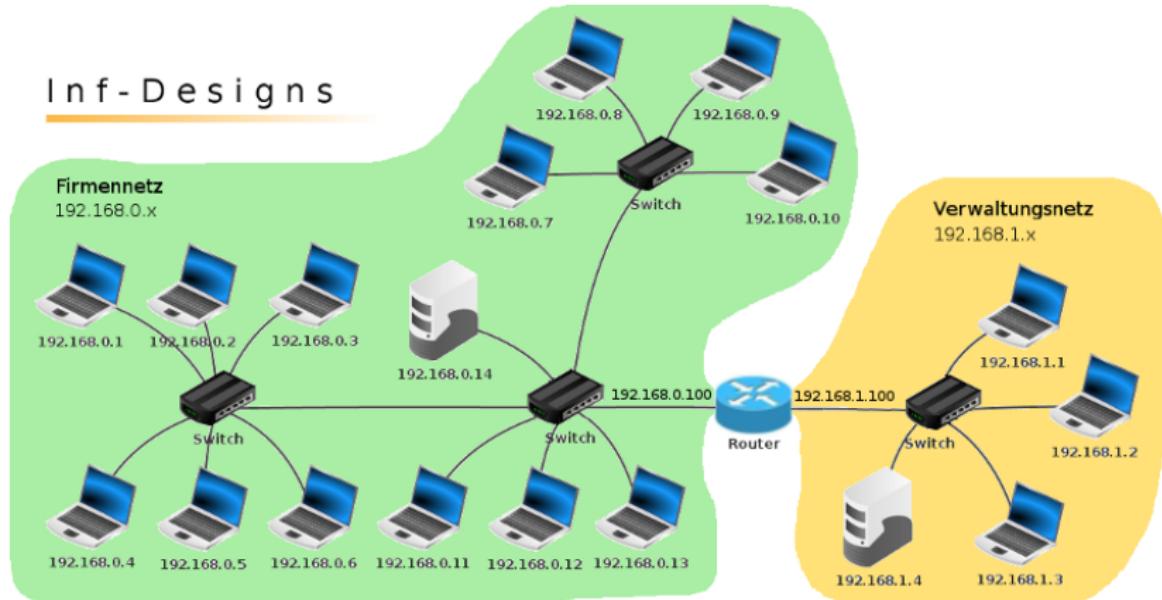
Endgeräte fügen die IP-Adressen hinzu:

Von:	192.168.0.3	Von:	192.168.0.3
An:	192.168.0.11	An:	192.168.0.11
Sequenz:	1	Sequenz:	2
Nachricht (max. 10 Zeichen):	Hi Bob, wi	Nachricht (max. 10 Zeichen):	e geht's?



# Netzwerk-Spiel: Netzzugangsschicht

## Inf-Designs



# Netzwerk-Spiel: Netzzugangsschicht

Von:	192.168.0.3	Von:	192.168.0.3
An:	192.168.0.11	An:	192.168.0.11
Sequenz:	1	Sequenz:	2
Nachricht (max. 10 Zeichen):	Hi Bob, wi	Nachricht (max. 10 Zeichen):	e geht's?



# Netzwerk-Spiel: Netzzugangsschicht

MAC-Adresse

Von:	48:2C:6A:1E:59:3D	Von:	48:2C:6A:1E:59:3D
An:	60:4E:46:F5:08:09	An:	60:4E:46:F5:08:09
Sequenz:	1	Sequenz:	2
Nachricht:	Hi Bob, wi		
Nachricht:	e geht's?		

## MAC-Adresse

- Analogie: Übergabe Brief von einer Hand in die andere



# Netzwerk-Spiel: Netzzugangsschicht

MAC-Adresse

Von:	48:2C:6A:1E:59:3D	Von:	48:2C:6A:1E:59:3D
An:	60:4E:46:F5:08:09	An:	60:4E:46:F5:08:09
Sequenz:	1	Sequenz:	2
Nachricht:	Hi Bob, wi	Nachricht:	e geht's?

## MAC-Adresse

- ▶ Analogie: Übergabe Brief von einer Hand in die andere
- ▶ „Fingerabdruck“ eines Computers im Internet



# Netzwerk-Spiel: Netzzugangsschicht

MAC-Adresse

Von:	48:2C:6A:1E:59:3D	Von:	48:2C:6A:1E:59:3D
An:	60:4E:46:F5:08:09	An:	60:4E:46:F5:08:09
Sequenz:	1	Sequenz:	2
Nachricht:	Hi Bob, wi		
Nachricht:	e geht's?		

## MAC-Adresse

- ▶ Analogie: Übergabe Brief von einer Hand in die andere
- ▶ „Fingerabdruck“ eines Computers im Internet
- ▶ 48 bit, die normalerweise in hexadezimaler Weise angegeben werden



# Netzwerk-Spiel: Netzzugangsschicht

MAC-Adresse

Von:	48:2C:6A:1E:59:3D	Von:	48:2C:6A:1E:59:3D
An:	60:4E:46:F5:08:09	An:	60:4E:46:F5:08:09
Sequenz:	1	Sequenz:	2
Nachricht: Hi Bob, wi		Nachricht: e geht's?	

## MAC-Adresse

- ▶ Analogie: Übergabe Brief von einer Hand in die andere
- ▶ „Fingerabdruck“ eines Computers im Internet
- ▶ 48 bit, die normalerweise in hexadezimaler Weise angegeben werden
- ▶ Z.B. 48:2C:6A:1E:59:3D



# Netzwerk-Spiel: Netzzugangsschicht

MAC-Adresse

Von:	48:2C:6A:1E:59:3D	Von:	48:2C:6A:1E:59:3D
An:	60:4E:46:F5:08:09	An:	60:4E:46:F5:08:09
Sequenz:	1	Sequenz:	2
Nachricht: Hi Bob, wi		Nachricht: e geht's?	

## MAC-Adresse

- ▶ Analogie: Übergabe Brief von einer Hand in die andere
- ▶ „Fingerabdruck“ eines Computers im Internet
- ▶ 48 bit, die normalerweise in hexadezimaler Weise angegeben werden
- ▶ Z.B. 48:2C:6A:1E:59:3D
- ▶ Jede Netzwerkkarte hat eine MAC-Adresse



Wie gelangt man überhaupt von einem Computer zum anderen?

- ▶ Media Access Control (MAC)-Adresse: In jedem Computer (in Netzwerkkarte) eingebaute Adresse



Wie gelangt man überhaupt von einem Computer zum anderen?

- ▶ **MAC**-Adresse: In jedem Computer (in Netzwerkkarte) eingebaute Adresse
- ▶ So etwas wie der „Fingerabdruck“ eines Computers



Wie gelangt man überhaupt von einem Computer zum anderen?

- ▶ **MAC**-Adresse: In jedem Computer (in Netzwerkkarte) eingebaute Adresse
- ▶ So etwas wie der „Fingerabdruck“ eines Computers
- ▶ Analogie: Übergabe Brief von einer Hand in die andere



Wie gelangt man überhaupt von einem Computer zum anderen?

- ▶ **MAC**-Adresse: In jedem Computer (in Netzwerkkarte) eingebaute Adresse
- ▶ So etwas wie der „Fingerabdruck“ eines Computers
- ▶ Analogie: Übergabe Brief von einer Hand in die andere
- ▶ 48 bit, die normalerweise in hexadezimaler Weise angegeben werden



Wie gelangt man überhaupt von einem Computer zum anderen?

- ▶ MAC-Adresse: In jedem Computer (in Netzwerkkarte) eingebaute Adresse
- ▶ So etwas wie der „Fingerabdruck“ eines Computers
- ▶ Analogie: Übergabe Brief von einer Hand in die andere
- ▶ 48 bit, die normalerweise in hexadezimaler Weise angegeben werden
- ▶ Z.B. 48:2C:6A:1E:59:3D

# Address Resolution Protocol (ARP)

## Verbindungsschicht

- ▶ Daten werden zwar mit Ziel-Internet Protocol (IP)-Adresse versehen, können jedoch nur von MAC zu MAC übergeben werden!



# Address Resolution Protocol (ARP)

## Verbindungsschicht

- ▶ Daten werden zwar mit Ziel-IP-Adresse versehen, können jedoch nur von MAC zu MAC übergeben werden!
- ▶ Muss verwendet werden, um herauszufinden, welche MAC-Adresse zu welcher IP-Adresse gehört.



# Address Resolution Protocol (ARP)

## Verbindungsschicht

- ▶ Daten werden zwar mit Ziel-IP-Adresse versehen, können jedoch nur von MAC zu MAC übergeben werden!
- ▶ Muss verwendet werden, um herauszufinden, welche MAC-Adresse zu welcher IP-Adresse gehört.
- ▶ Beispiel



# Address Resolution Protocol (ARP)

## Verbindungsschicht

- ▶ Daten werden zwar mit Ziel-IP-Adresse versehen, können jedoch nur von MAC zu MAC übergeben werden!
- ▶ Muss verwendet werden, um herauszufinden, welche MAC-Adresse zu welcher IP-Adresse gehört.
- ▶ Beispiel
  - ▶ Computer **192.168.0.33** will Nachricht senden an **192.168.0.34**, im selben Subnetz!



# Address Resolution Protocol (ARP)

## Verbindungsschicht

- ▶ Daten werden zwar mit Ziel-IP-Adresse versehen, können jedoch nur von MAC zu MAC übergeben werden!
- ▶ Muss verwendet werden, um herauszufinden, welche MAC-Adresse zu welcher IP-Adresse gehört.
- ▶ Beispiel
  - ▶ Computer 192.168.0.33 will Nachricht senden an 192.168.0.34, im selben Subnetz!
  - ▶ 192.168.0.33 sendet daher eine Nachricht an alle Computer in 192.168.0.x : „Wer hat die IP 192.168.0.34“?



# Address Resolution Protocol (ARP)

## Verbindungsschicht

- ▶ Daten werden zwar mit Ziel-IP-Adresse versehen, können jedoch nur von MAC zu MAC übergeben werden!
- ▶ Muss verwendet werden, um herauszufinden, welche MAC-Adresse zu welcher IP-Adresse gehört.
- ▶ Beispiel
  - ▶ Computer 192.168.0.33 will Nachricht senden an 192.168.0.34, im selben Subnetz!
  - ▶ 192.168.0.33 sendet daher eine Nachricht an alle Computer in 192.168.0.x : „Wer hat die IP 192.168.0.34“?
  - ▶ Computer 192.168.0.34 antwortet: „Ich! Meine MAC ist 60:4E:46:F5:08:09.“



# Address Resolution Protocol (ARP)

## Verbindungsschicht

- ▶ Daten werden zwar mit Ziel-IP-Adresse versehen, können jedoch nur von MAC zu MAC übergeben werden!
- ▶ Muss verwendet werden, um herauszufinden, welche MAC-Adresse zu welcher IP-Adresse gehört.
- ▶ Beispiel
  - ▶ Computer 192.168.0.33 will Nachricht senden an 192.168.0.34, im selben Subnetz!
  - ▶ 192.168.0.33 sendet daher eine Nachricht an alle Computer in 192.168.0.x : „Wer hat die IP 192.168.0.34“?
  - ▶ Computer 192.168.0.34 antwortet: „Ich! Meine MAC ist 60:4E:46:F5:08:09“
  - ▶ Nun kann der Computer seine Nachricht an den Empfänger übergeben.



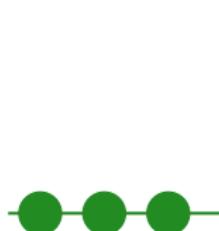
# Auftrag ( $\sim 20'$ )

- ▶  1.2-1.4
- ▶  Challenge: 1.5

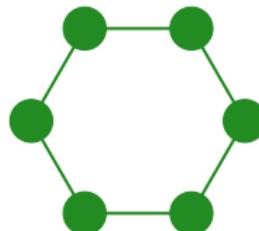


# Topologie, grössere Netzwerke

Wie bauen wir am besten ein grösseres Netzwerk auf?



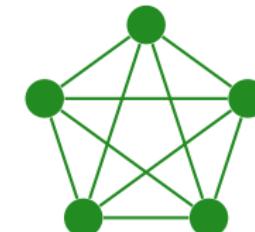
(a) Bus



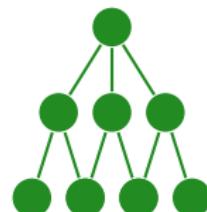
(b) Ring



(c) Stern



(d) Vollvermascht



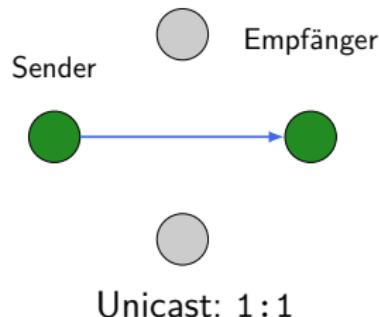
(e) Baum

Abbildung: Beispiele von Netzwerk-Topologien



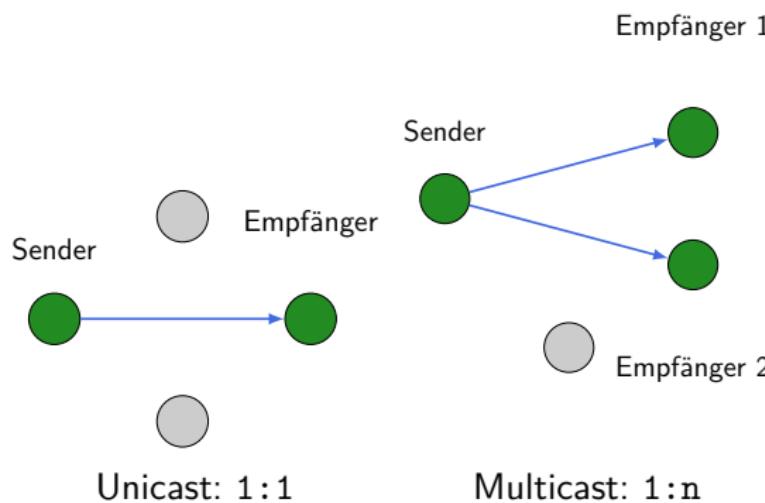
# Unicast, Broadcast, Multicast

- ▶ **Unicast:** Nachricht wird an einen einzigen Empfänger gesendet
- ▶ **Broadcast:** Nachricht wird an alle Empfänger im selben Subnetz gesendet
- ▶ **Multicast:** Nachricht wird an eine ausgewählte Gruppe von Empfängern gesendet



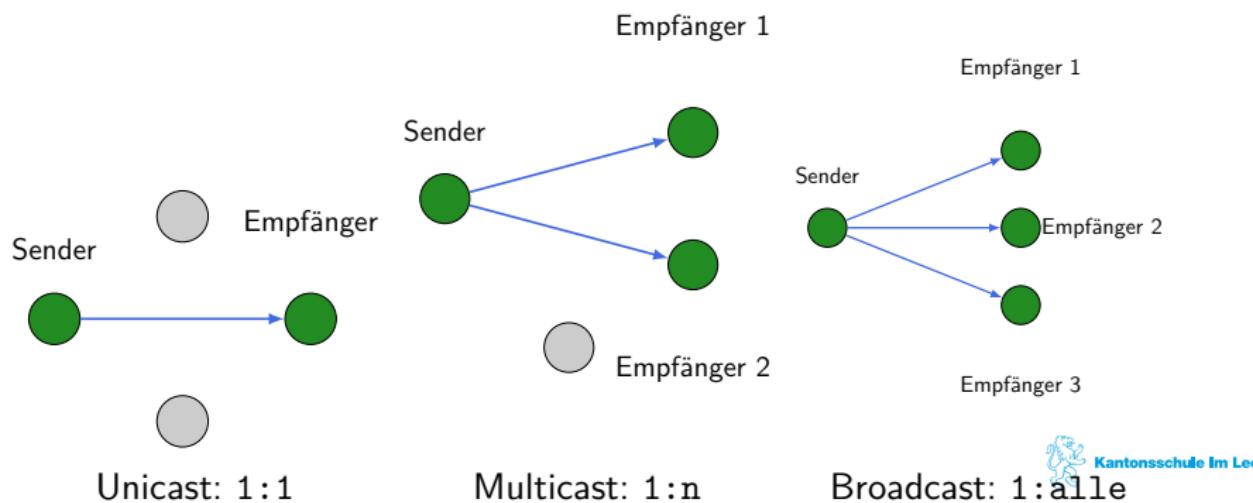
# Unicast, Broadcast, Multicast

- ▶ **Unicast:** Nachricht wird an einen einzigen Empfänger gesendet
- ▶ **Broadcast:** Nachricht wird an alle Empfänger im selben Subnetz gesendet
- ▶ **Multicast:** Nachricht wird an eine ausgewählte Gruppe von Empfängern gesendet



# Unicast, Broadcast, Multicast

- ▶ **Unicast:** Nachricht wird an einen einzigen Empfänger gesendet
- ▶ **Broadcast:** Nachricht wird an alle Empfänger im selben Subnetz gesendet
- ▶ **Multicast:** Nachricht wird an eine ausgewählte Gruppe von Empfängern gesendet



# Auftrag ( $\sim 10'$ )



1.8-1.11



Kantonsschule im Lee

# IP-Adresse

## Vermittlungsschicht

- ▶ IP-Adresse = Netzwerk-Adresse eines Computers



# IP-Adresse

## Vermittlungsschicht

- ▶ IP-Adresse = Netzwerk-Adresse eines Computers
- ▶ Analogie: Wohnadresse einer Person (Strasse & Stadt)



# IP-Adresse

## Vermittlungsschicht

- ▶ IP-Adresse = Netzwerk-Adresse eines Computers
- ▶ Analogie: Wohnadresse einer Person (Strasse & Stadt)
- ▶ Häufig im IPv4-Format: 4 bytes, also 4 Zahlen von 0-255, z.B.:  
**11000000.10101000.00000001.00000101**



- ▶ IP-Adresse = Netzwerk-Adresse eines Computers
- ▶ Analogie: Wohnadresse einer Person (Strasse & Stadt)
- ▶ Häufig im IPv4-Format: 4 bytes, also 4 Zahlen von 0-255, z.B.:  
`11000000.10101000.00000001.00000101`
- ▶ Wird meist dezimal angegeben, z.B. `192.168.1.5`

- ▶ IP-Adresse = Netzwerk-Adresse eines Computers
- ▶ Analogie: Wohnadresse einer Person (Strasse & Stadt)
- ▶ Häufig im IPv4-Format: 4 bytes, also 4 Zahlen von 0-255, z.B.:  
`11000000.10101000.00000001.00000101`
- ▶ Wird meist dezimal angegeben, z.B. `192.168.1.5`
- ▶ Wie viele IPv4-Adressen gibt es? Reicht das?

- ▶ IP-Adresse = Netzwerk-Adresse eines Computers
- ▶ Analogie: Wohnadresse einer Person (Strasse & Stadt)
- ▶ Häufig im IPv4-Format: 4 bytes, also 4 Zahlen von 0-255, z.B.:  
`11000000.10101000.00000001.00000101`
- ▶ Wird meist dezimal angegeben, z.B. `192.168.1.5`
- ▶ Wie viele IPv4-Adressen gibt es? Reicht das?
  - ▶  $4 \cdot 8 \text{ bits} = 32 \text{ bits}$

$2^{32} \sim 4 \text{ Milliarden Adressen} \rightarrow \text{zu wenig!}$



- ▶ IP-Adresse = Netzwerk-Adresse eines Computers
- ▶ Analogie: Wohnadresse einer Person (Strasse & Stadt)
- ▶ Häufig im IPv4-Format: 4 bytes, also 4 Zahlen von 0-255, z.B.:  
`11000000.10101000.00000001.00000101`
- ▶ Wird meist dezimal angegeben, z.B. `192.168.1.5`
- ▶ Wie viele IPv4-Adressen gibt es? Reicht das?
  - ▶  $4 \cdot 8$  bits = 32 bits

$2^{32} \sim 4$  Milliarden Adressen → zu wenig!

- ▶ IPv6: 128 bits, fast unendlich viele Adressen

$2^{128} \sim 3.4 \cdot 10^{38}$  Adressen



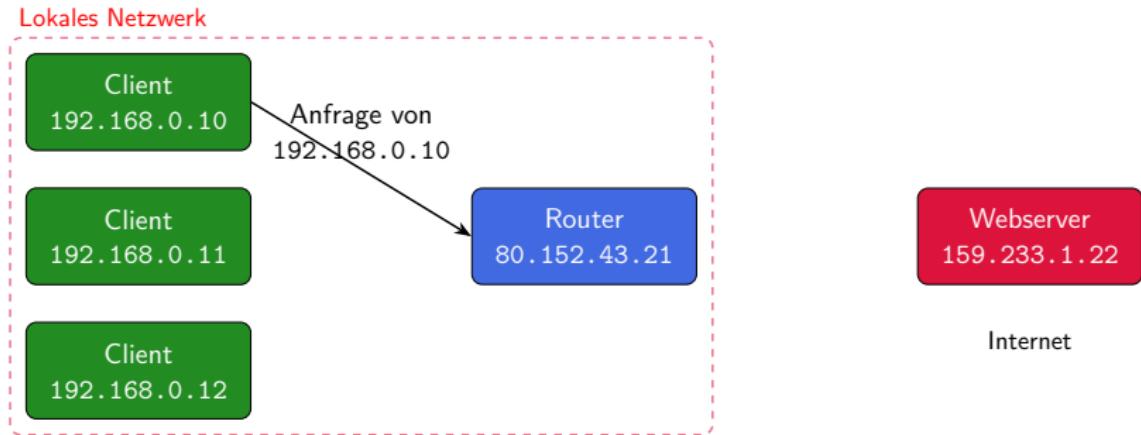
## Lokale vs. globale IP-Adresse

- ▶ Aufgrund von IPv4-Mangel: Unterscheidung zwischen lokalen und globalen IP-Adressen
- ▶ **Lokale IP:** nur im lokalen Netzwerk gültig (z.B. 192.168.x.x, 10.x.x.x)
- ▶ **Globale IP:** weltweit eindeutig (z.B. 8.8.8.8)



# Lokale vs. globale IP-Adresse

## Network Address Translation (NAT)



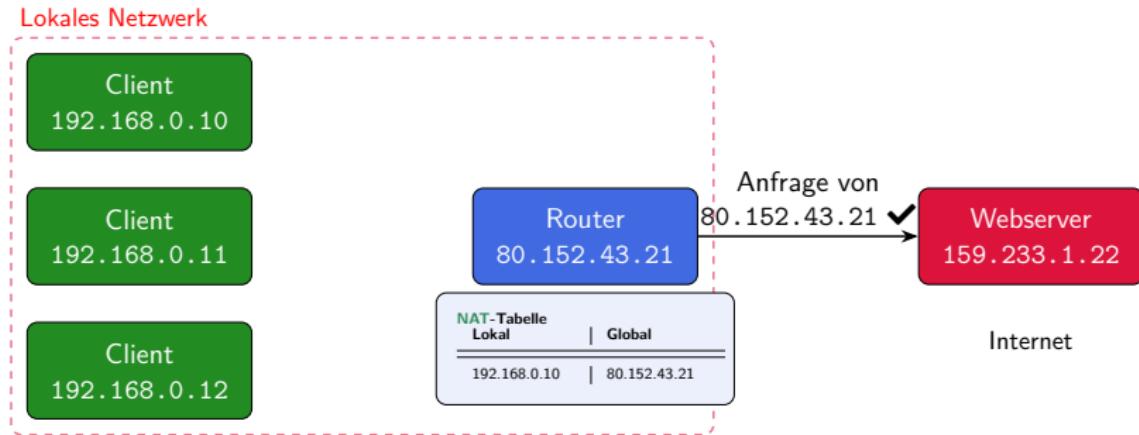
### Ausgehende Anfrage

- ▶ Client sendet Anfrage mit lokaler IP-Adresse
- ▶ Router ersetzt lokale IP durch seine globale IP



# Lokale vs. globale IP-Adresse

## Network Address Translation (NAT)



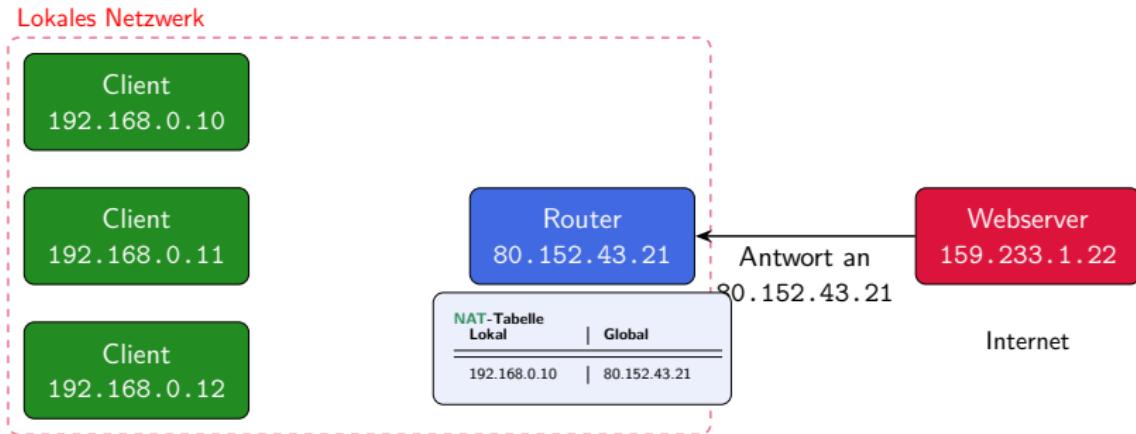
## Ausgehende Anfrage

- ▶ Client sendet Anfrage mit lokaler IP-Adresse
- ▶ Router ersetzt lokale IP durch seine globale IP



# Lokale vs. globale IP-Adresse

## Network Address Translation (NAT)



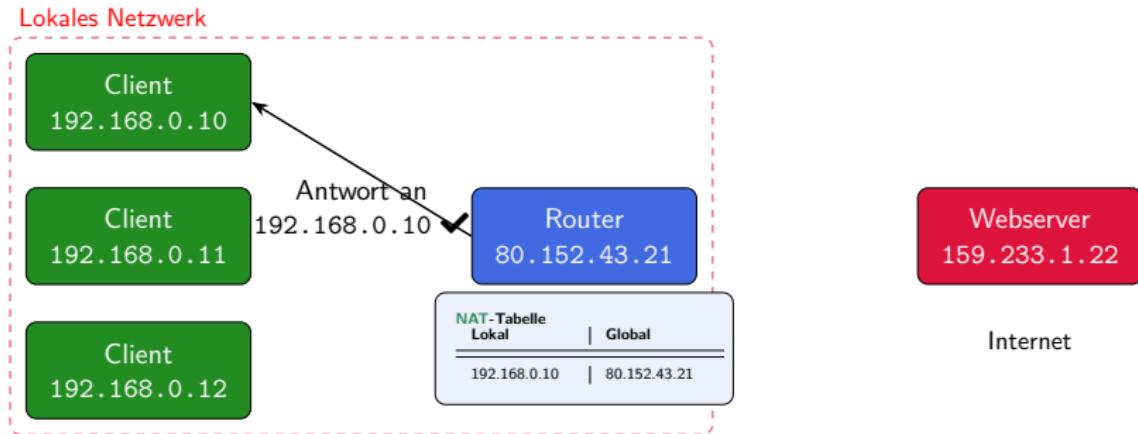
## Eingehende Antwort

- ▶ Server antwortet an die globale IP des Routers
- ▶ Router schlägt in NAT-Tabelle nach und leitet an lokale IP weiter



# Lokale vs. globale IP-Adresse

## Network Address Translation (NAT)



## Eingehende Antwort

- ▶ Server antwortet an die globale IP des Routers
- ▶ Router schlägt in NAT-Tabelle nach und leitet an lokale IP weiter



# Eigene IP-Adresse herausfinden

- ▶ MacOS, Terminal (via Spotlight-Suche):  
    > ipconfig getifaddr en0
- ▶ Windows, Programm „cmd“ (via  + ):  
    > ipconfig





# ping

## Vermittlungsschicht

- ▶ Im Terminal / cmd: `> ping [ip address]`, z.B.:  
`> ping 159.233.1.22`



Kantonsschule im Lee



# ping

## Vermittlungsschicht

- ▶ Im Terminal / cmd: `> ping [ip address]`, z.B.:  
`> ping 159.233.1.22`
- ▶ Minimale Nachricht an andere IP-Adresse schicken



Kantonsschule im Lee



# ping

## Vermittlungsschicht

- ▶ Im Terminal / cmd: > ping [ip address], z.B.:  
    > ping 159.233.1.22
- ▶ Minimale Nachricht an andere IP-Adresse schicken
- ▶ Ziel: schauen ob eine IP existiert, bzw. erreichbar ist, wird mittels ICMP realisiert



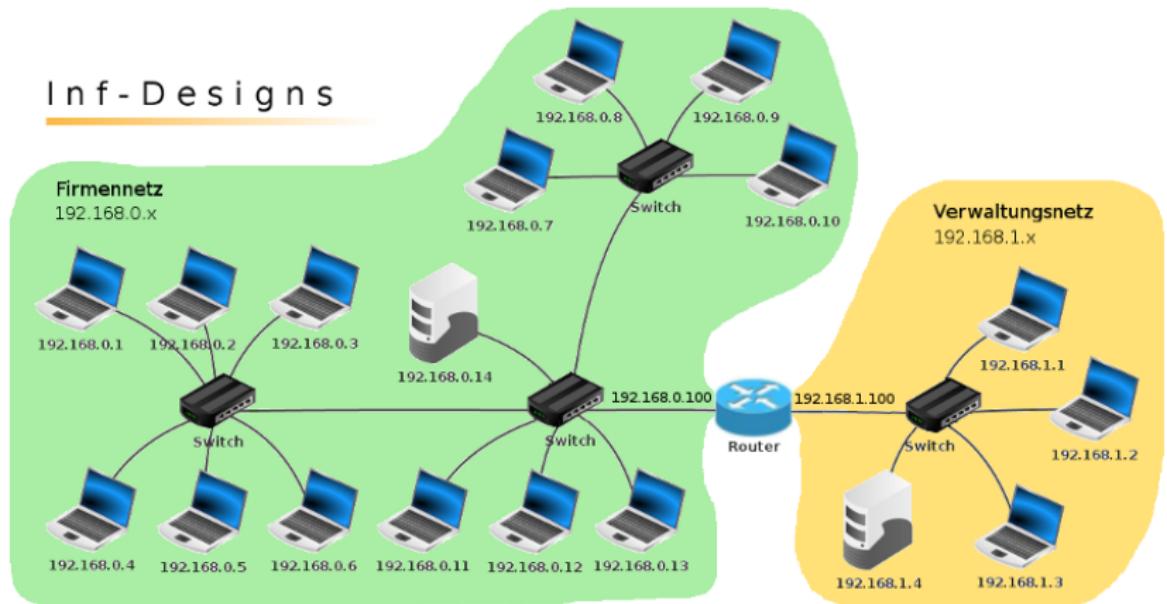
Kantonsschule im Lee

## Auftrag

- ▶  1.12 : Eigene IPv4-Adresse herausfinden
- ▶  Challenge: 1.13 :  ping im echten Netzwerk testen
- ▶  1.16 - 1.18 : Notation IPv4 und IPv6
- ▶  Challenge: 1.14, 1.15 : Python, ARP-Scan



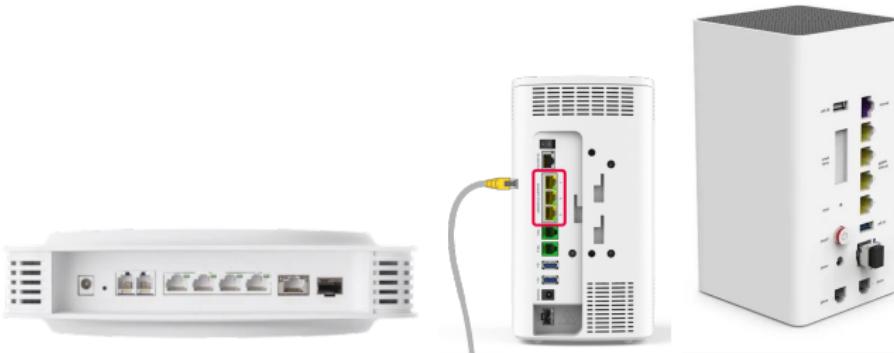
### Inf-Designs



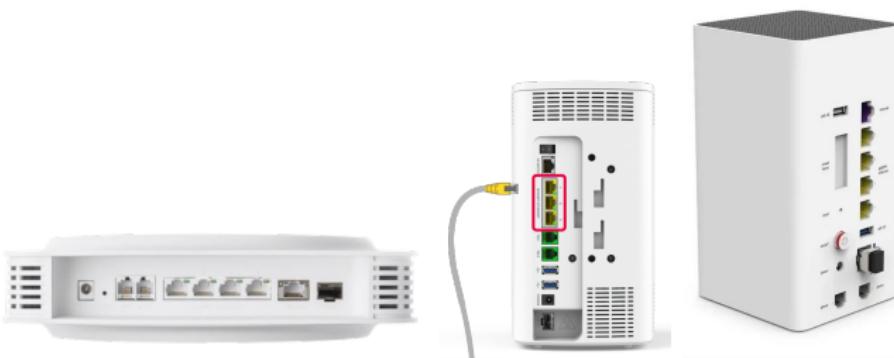
### Switch oder Router?!?



## Switch oder Router?!?



## Switch oder Router?!?



Beides (meistens) im gleichen Gerät!



## Subnetzwerke

- ▶ Verschiedene Geräte gehören zum gleichen Subnetz, falls Sie



## Subnetzwerke

- ▶ Verschiedene Geräte gehören zum gleichen Subnetz, falls Sie
  - ▶ Physisch miteinander verbunden sind (WLAN, Kabel, etc.)



## Subnetzwerke

- ▶ Verschiedene Geräte gehören zum gleichen Subnetz, falls Sie
  - ▶ Physisch miteinander verbunden sind (WLAN, Kabel, etc.)
  - ▶ Die gleiche Subnetzmaske teilen (später mehr dazu)



## Netzmaske

- ▶ Sieht ähnlich aus wie eine IP-Adresse: xxx.xxx.xxx.xxx



## Netzmaske

- ▶ Sieht ähnlich aus wie eine IP-Adresse: xxx.xxx.xxx.xxx
- ▶ Fasst mehrere IPs zu einer Gruppe (= Subnetz) zusammen



## Netzmaske

- ▶ Sieht ähnlich aus wie eine IP-Adresse: `xxx.xxx.xxx.xxx`
- ▶ Fasst mehrere IPs zu einer Gruppe (= Subnetz) zusammen
- ▶ Gibt an, wie viele Stellen einer IP-Adresse gleich sein müssen, damit mehrere Computer zum gleichen (Sub-)Netzwerk gehören.



## Netzmaske

- ▶ Sieht ähnlich aus wie eine IP-Adresse: **xxx.xxx.xxx.xxx**
- ▶ Fasst mehrere IPs zu einer Gruppe (= Subnetz) zusammen
- ▶ Gibt an, wie viele Stellen einer IP-Adresse gleich sein müssen, damit mehrer Computer zum gleichen (Sub-)Netzwerk gehören.
- ▶ **Beispiel:**

	Dezimal	Binär
IP 1	159.233.1.22	<b>10011111.11101001.00000001.00010110</b>
IP 2	159.233.1.1	<b>10011111.11101001.00000001.00000001</b>
Subnetzmaske	<b>255.255.255.0</b>	<b>11111111.11111111.11111111.00000000</b>

Bedeutet, dass alle IP-Adressen die mit **159.233.1.[...]** anfangen, zum selben Netzwerk gehören



## Netzmaske

- ▶ Sieht ähnlich aus wie eine IP-Adresse: **xxx.xxx.xxx.xxx**
- ▶ Fasst mehrere IPs zu einer Gruppe (= Subnetz) zusammen
- ▶ Gibt an, wie viele Stellen einer IP-Adresse gleich sein müssen, damit mehrer Computer zum gleichen (Sub-)Netzwerk gehören.
- ▶ **Beispiel:**

	Dezimal	Binär
IP 1	159.233.1.22	<b>10011111.11101001.00000001.00010110</b>
IP 2	159.233.1.1	<b>10011111.11101001.00000001.00000001</b>
Subnetzmaske	<b>255.255.255.0</b>	<b>11111111.11111111.11111111.00000000</b>

Bedeutet, dass alle IP-Adressen die mit **159.233.1.[...]** anfangen, zum selben Netzwerk gehören

- ▶ Wird manchmal „informell“ auch so angegeben:

**159.233.1.x**



# Subnetze & Netzmasken

## Übung

Eigene IP	Netzmaske	Ziel-IP	Gleches Subnetz?
213.45.19.89	255.255.255.0	213.45.17.89	
213.45.19.89	255.255.0.0	213.45.17.89	
88.100.11.17	255.255.255.0	88.100.11.254	
88.100.11.17	0.0.0.0	213.45.19.89	
10.0.0.0	255.255.255.252	10.0.0.1	
1.2.3.0	255.255.255.252	1.2.3.5	

- ▶ → IPs dürfen sich nur an denjenigen Stellen unterscheiden, wo in der Maske (binär!) Nullen stehen.



- ▶ Umrechner dezimal → binär:



# Subnetze & Netzmasken

## Aufträge (Skript)

### ► Subnetzmasken

-  Netzmasken: 1.21-1.24
-  Filius: 1.25
-  Challenge: 1.26
-  Filius: 1.27



# Subnetze & Netzmasken

## Weitere Übungen (Lösung)

Netz- maske Sub- netz	255.255.255.0		255.255.0.0		255.255.255.240	
	IP 3.4.5.6	IP 3.3.3.3	IP 3.4.5.6	IP 3.3.3.3	IP 3.4.5.6	IP 3.3.3.3
3.3.3.4	Nein	Ja	Nein	Ja	Nein	Ja
4.4.4.3	Nein	Nein	Nein	Nein	Nein	Nein
3.4.5.99	Ja	Nein	Ja	Nein	Nein	Nein
3.3.4.4	Nein	Nein	Nein	Ja	Nein	Nein
3.4.7.7	Nein	Ja	Ja	Nein	Nein	Nein
3.3.3.17	Nein	Ja	Nein	Ja	Nein	Nein



# Netzwerk-Spiel: **Transportschicht**

Endgeräte trennen die Nachrichten in Segmente auf und fügen Sequenznummern hinzu:

Sequenz:	1	Sequenz:	2
Nachricht (max. 10 Zeichen):	Hi Bob, wi	Nachricht (max. 10 Zeichen):	e geht's?





# TCP (*Transmission Control Protocol*) Header

Transportschicht

TCP = Übertragungssteuerungsprotokoll

Transmission Control Protocol (TCP) Header	
Source Port	Destination Port
Sequence # (SEQ)	
Acknowledgement # (ACK)	
Weitere TCP-Header-Felder	
<b>Nutzdaten</b>	
... Anwendungsdaten ...	



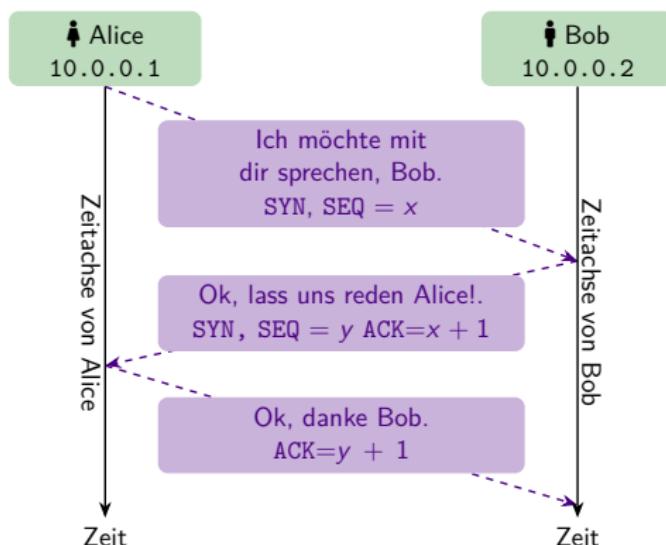
Portnummer	Anwendung	Verwendung
20	FTP	Dateitransfer
21	FTP	Befehlssteuerung
22	SSH	Sichere Shell-Zugriffe
23	Telnet	Unverschlüsselte Fernsteuerung
25	SMTP	E-Mail-Versand
53	DNS	Namensauflösung
80	HTTP	Webseiten-Abruf
110	POP3	E-Mail-Abholung
143	IMAP	E-Mail-Management
443	HTTPS	Verschlüsselter Webseiten-Abruf
993	IMAP	E-Mail-Management über SSL
995	POP3	E-Mail-Abholung über SSL

Gängige TCP-Ports, Anwendungen und deren Verwendung



# ! TCP: 3-way handshake

## Transportschicht



1. Computer A (*client*, 10.0.0.2) erstellt eine Verbindungsanfrage zum Computer B (*server*, 10.0.0.3), mittels einem Paket, das nur das SYN-Flaggen gesetzt hat.
2. Der Server antwortet sowohl mit einem gesetzten SYN und einem ACK-Flaggen
3. Im letzten Schritt antwortet der *client* nochmals mit einem einzelnen ACK-Flaggen  
→ TCP-Verbindung ist nun erstellt!



# TCP: 3-way handshake

## Auftrag

- ▶ Skript:  1.30-1.33 (TCP, Servers / Clients)
- ▶  Challenge: 1.34 (Email-Server)



# Netzwerk-Spiel: **Anwendungsschicht**

Endgeräte verfassen Nachrichten:

Nachricht  
(max. 10 Zeichen):

Hi Bob, wi

Nachricht  
(max. 10 Zeichen):

e geht's?



# Webserver

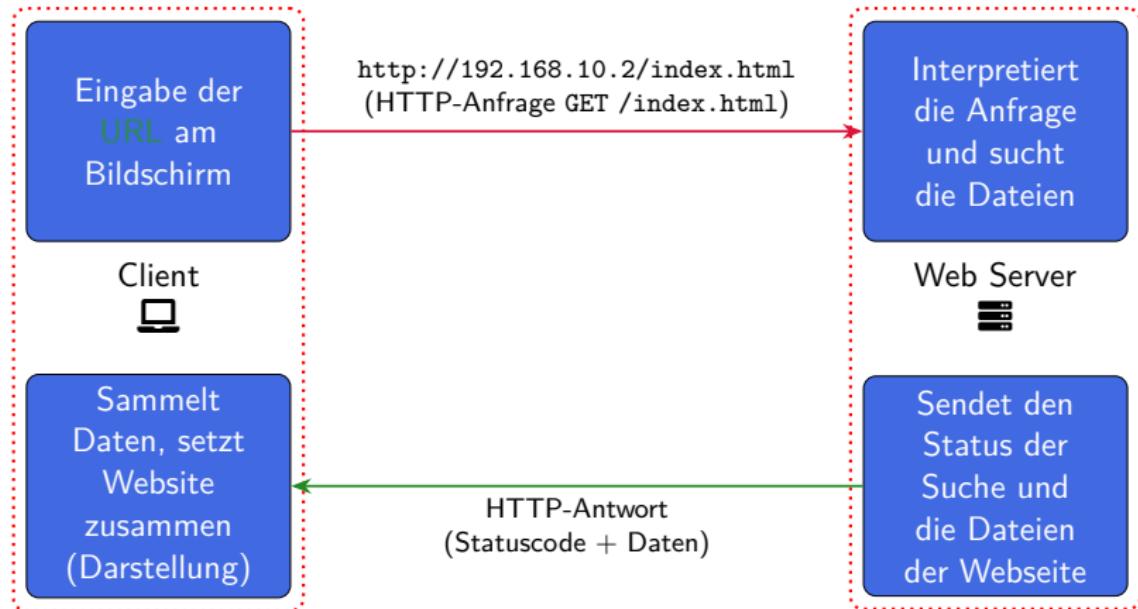


Abbildung: Schematischer Ablauf eines Webseiten-Aufrufs mit dem Hypertext Transfer Protocol (HTTP)-Protokoll





# Client vs Server vs Host

## Anwendungsschicht

### Definitionen

- ▶ **Client**





# Client vs Server vs Host

## Anwendungsschicht

### Definitionen

- ▶ **Client**

- ▶ „Kunde“ (von en. *client*)





# Client vs Server vs Host

## Anwendungsschicht

### Definitionen

#### ► **Client**

- „Kunde“ (von en. *client*)
- Computer(-Programm), das etwas will





# Client vs Server vs Host

## Anwendungsschicht

### Definitionen

#### ► Client

- „Kunde“ (von en. *client*)
- Computer(-Programm), das etwas will

#### ► Server





# Client vs Server vs Host

## Anwendungsschicht

### Definitionen

#### ► Client

- „Kunde“ (von en. *client*)
- Computer(-Programm), das etwas will

#### ► Server

- „Diener“ (von en. *to serve* = bedienen, ≠ „surf“)





# Client vs Server vs Host

## Anwendungsschicht

### Definitionen

#### ► Client

- ▶ „Kunde“ (von en. *client*)
- ▶ Computer(-Programm), das etwas will

#### ► Server

- ▶ „Diener“ (von en. *to serve* = bedienen,  $\neq$  „surf“)
- ▶ Computer(-Programm), das etwas hat, das andere Computer(-Programme) brauchen





# Client vs Server vs Host

## Anwendungsschicht

### Definitionen

#### ► Client

- ▶ „Kunde“ (von en. *client*)
- ▶ Computer(-Programm), das etwas will

#### ► Server

- ▶ „Diener“ (von en. *to serve* = bedienen, ≠ „surf“)
- ▶ Computer(-Programm), das etwas hat, das andere Computer(-Programme) brauchen
- ▶ Kann Verbindungsanfragen von anderen Computer(-Programmen) akzeptieren / ablehnen





# Client vs Server vs Host

## Anwendungsschicht

### Definitionen

#### ► Client

- „Kunde“ (von en. *client*)
- Computer(-Programm), das etwas will

#### ► Server

- „Diener“ (von en. *to serve* = bedienen,  $\neq$  „surf“)
- Computer(-Programm), das etwas hat, das andere Computer(-Programme) brauchen
- Kann Verbindungsanfragen von anderen Computer(-Programmen) akzeptieren / ablehnen

#### ► Host





# Client vs Server vs Host

## Anwendungsschicht

### Definitionen

#### ► Client

- ▶ „Kunde“ (von en. *client*)
- ▶ Computer(-Programm), das etwas will

#### ► Server

- ▶ „Diener“ (von en. *to serve* = bedienen, ≠ „surf“)
- ▶ Computer(-Programm), das etwas hat, das andere Computer(-Programme) brauchen
- ▶ Kann Verbindungsanfragen von anderen Computer(-Programmen) akzeptieren / ablehnen

#### ► Host

- ▶ = Computer in einem Netzwerk, allgemein für „server“ oder „client“





# Client vs Server vs Host

## Anwendungsschicht

### Definitionen

#### ► Client

- ▶ „Kunde“ (von en. *client*)
- ▶ Computer(-Programm), das etwas will

#### ► Server

- ▶ „Diener“ (von en. *to serve* = bedienen, ≠ „surf“)
- ▶ Computer(-Programm), das etwas hat, das andere Computer(-Programme) brauchen
- ▶ Kann Verbindungsanfragen von anderen Computer(-Programmen) akzeptieren / ablehnen

#### ► Host

- ▶ = Computer in einem Netzwerk, allgemein für „server“ oder „client“
- ▶ *Host name* = Name eines Computers





# Browser

## Anwendungsschicht

- ▶ Browser = Programm, kann eine Datei (meist .html)...
  - ▶ ...über das Internet *anfordern*
  - ▶ ...nach dem Empfang *darstellen*
- ▶ Das Senden / Empfangen von Dateien übernimmt das Betriebssystem (z.B. Windows, MacOS)
- ▶ Beispiele:
  - ▶  Chrome
  - ▶  Firefox
  - ▶  Safari
  - ▶  Edge
  - ▶ etc.





# URL (Uniform Resource Locator)

Präsentationsschicht: Webseiten

URL = Web-Adresse

http:// www. beispiel. ch /dokumente/ reglemente.html

Protokoll    Server    Domain    TLD    Ordner    Dateiname

- ▶ Erster Teil bezeichnet einen *Server*





# URL (Uniform Resource Locator)

Präsentationsschicht: Webseiten

URL = Web-Adresse

http:// www. beispiel. ch /dokumente/ reglemente.html

Protokoll    Server    Domain    TLD    Ordner    Dateiname

- ▶ Erster Teil bezeichnet einen *Server*
  - ▶ z.B. 58.55.110.9 → www.beispiel.ch





# URL (Uniform Resource Locator)

Präsentationsschicht: Webseiten

URL = Web-Adresse

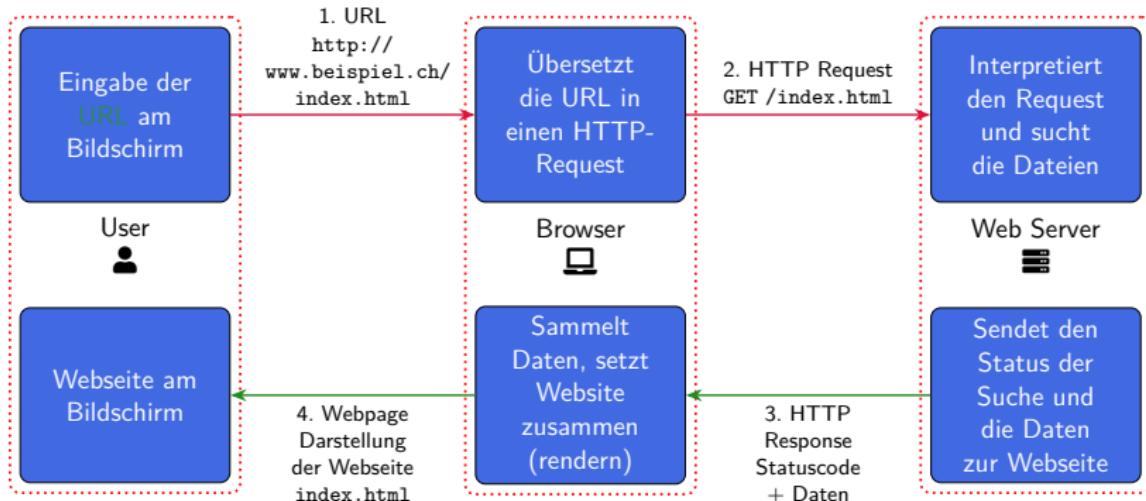
http:// www. beispiel. ch /dokumente/ reglemente.html

Protokoll    Server    Domain    TLD    Ordner    Dateiname

- ▶ Erster Teil bezeichnet einen *Server*
  - ▶ z.B. 58.55.110.9 → www.beispiel.ch
- ▶ Alles nach der TLD ist ein normaler Ordnerpfad auf dem Computer



# HTTP(S)-Protokoll



Beispiel: HTML (*Hypertext Markup Language*)

- ▶ Demo (*Inspect Element* oder *View Page Source*)



Beispiel: HTML (*Hypertext Markup Language*)

- ▶ Demo (*Inspect Element* oder *View Page Source*)
- ▶ Speichert die Struktur einer Webseite



Beispiel: HTML (*Hypertext Markup Language*)

- ▶ Demo (*Inspect Element* oder *View Page Source*)
- ▶ Speichert die Struktur einer Webseite
- ▶ Kann weitere Dateien enthalten, beispielsweise:



Beispiel: HTML (*Hypertext Markup Language*)

- ▶ Demo (*Inspect Element* oder *View Page Source*)
- ▶ Speichert die Struktur einer Webseite
- ▶ Kann weitere Dateien enthalten, beispielsweise:
  - ▶ CSS (Cascading Style Sheet), für die Darstellung (Farbe, Grösse etc.)



Beispiel: HTML (*Hypertext Markup Language*)

- ▶ Demo (*Inspect Element* oder *View Page Source*)
- ▶ Speichert die Struktur einer Webseite
- ▶ Kann weitere Dateien enthalten, beispielsweise:
  - ▶ CSS (Cascading Style Sheet), für die Darstellung (Farbe, Grösse etc.)
  - ▶ JavaScript, jQuery etc. für die Interaktivität (Ein- und Ausblenden von Objekten etc.)



## Beispiel: HTML (*Hypertext Markup Language*)

- ▶ Demo (*Inspect Element* oder *View Page Source*)
- ▶ Speichert die Struktur einer Webseite
- ▶ Kann weitere Dateien enthalten, beispielsweise:
  - ▶ CSS (Cascading Style Sheet), für die Darstellung (Farbe, Grösse etc.)
  - ▶ JavaScript, jQuery etc. für die Interaktivität (Ein- und Ausblenden von Objekten etc.)
- ▶ Wie man eine Webseite baut ist hier nicht das Hauptthema, wird aber in der Interessenwoche vermittelt



# Webserver

## Aufträge (Skript)

- ▶  Filius (Web-Server): 1.35
- ▶  Challenge: Filius (“Echter” Web-Server): 1.36
- ▶  Challenge: 1.37 (HTML)



# Domain Name System (DNS)

- ▶ Wenn Sie auf eine Webseite wie <https://sbb.ch> gehen, woher weiss der Client, von welchem Server er die Informationen anfragen soll?
- ▶ Domain Name System (DNS): URL  $\rightleftarrows$  IP-Adresse!



# DNS

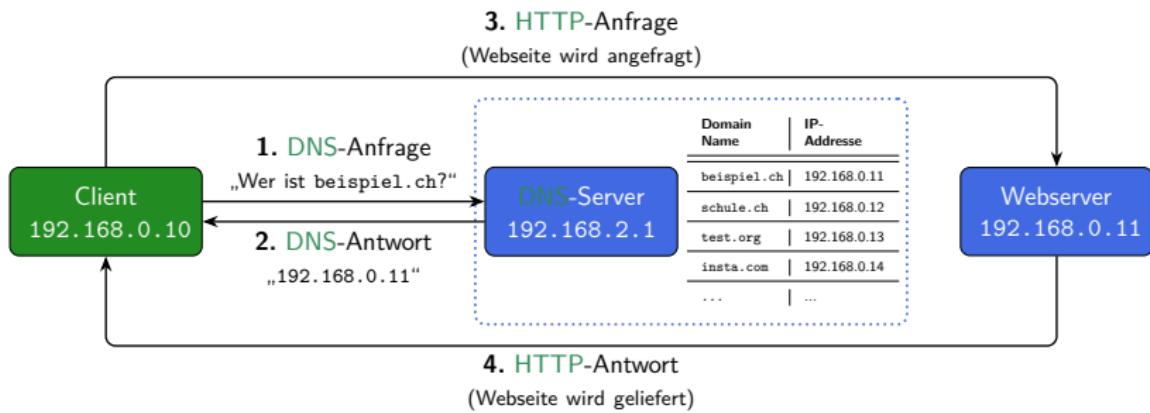


Abbildung: Schematischer Ablauf einer DNS-Anfrage



# DNS

## Aufträge (Skript)

- ▶  Aufgaben 1.34-1.36

### Weitere Aufgaben

- ▶  1.38-1.40
- ▶  Challenge: 1.41-1.44

