

Übungen zu Einführung in Rechnernetze

1. Übung

Matthias Flittner, Tim Gerhard
[flittner, tim.gerhard]@kit.edu

Institut für Telematik, Prof. Zitterbart



© Peter Baumung

1. Allerlei zum Internet
2. Das Hypertext Transfer Protocol (HTTP)
3. Datenübertragung und Verzögerung
4. HTTP in der Praxis

a) Maus vs. Realität

- Wichtigste Unterschiede zu: <https://youtu.be/8PNRrOGJqUI>



■ Heimanschluss

- Kein Analog-Modem mit serielllem Anschluss mehr
- Oft Router zur Anbindung mehrerer Geräte ans Internet

■ Internet-Service-Provider

- Kein Zustand (Wartenummer) pro Nutzer
- Agiert nicht als Stellvertreter

■ Internet-Auskunft

- Nicht zentral in Frankfurt sondern DNS-Hierarchie

■ Router im Internet

- Bei Stau nicht Umleitung sondern Paketverlust



Mehr dazu im
Laufe der VL

b) Wie ist das Internet tatsächlich aufgebaut?



- Das Internet ist ein weltweites Verbund von Netzen

Netzrand

■ Am Netzrand

- Endsysteme: Client und Server
- Server sind häufig in Datenzentren

■ Zugangsnetze

- Heimnetz
- Mobiles Zugangsnetz
- Unternehmensnetz

■ Physikalische Medien

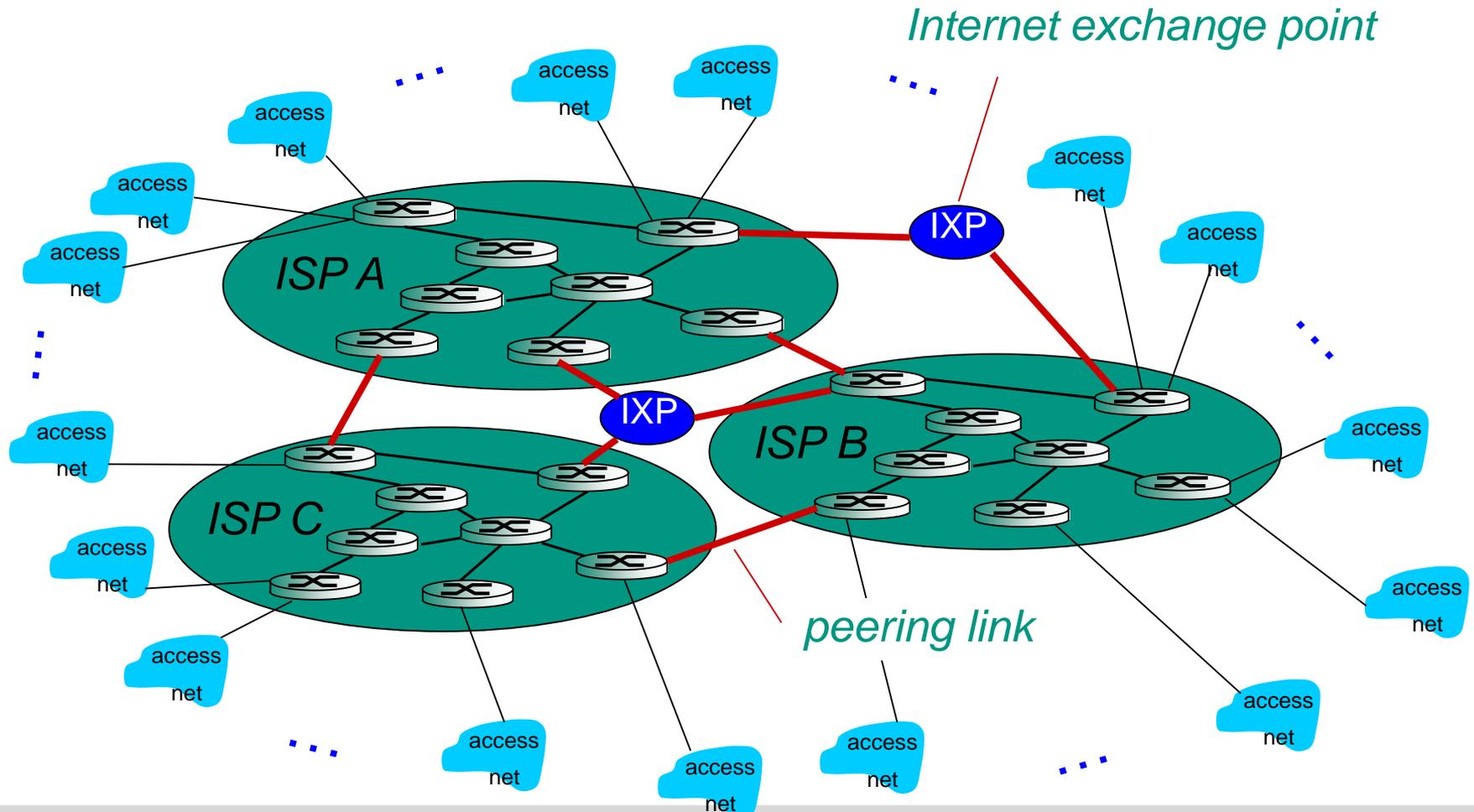
- Drahtgebunden
- Drahtlos

Beispiel Google

- Ca. 50 – 100 Datenzentren weltweit
- Davon ca. 15 große Datenzentren mit jeweils mehr als 100.000 Server

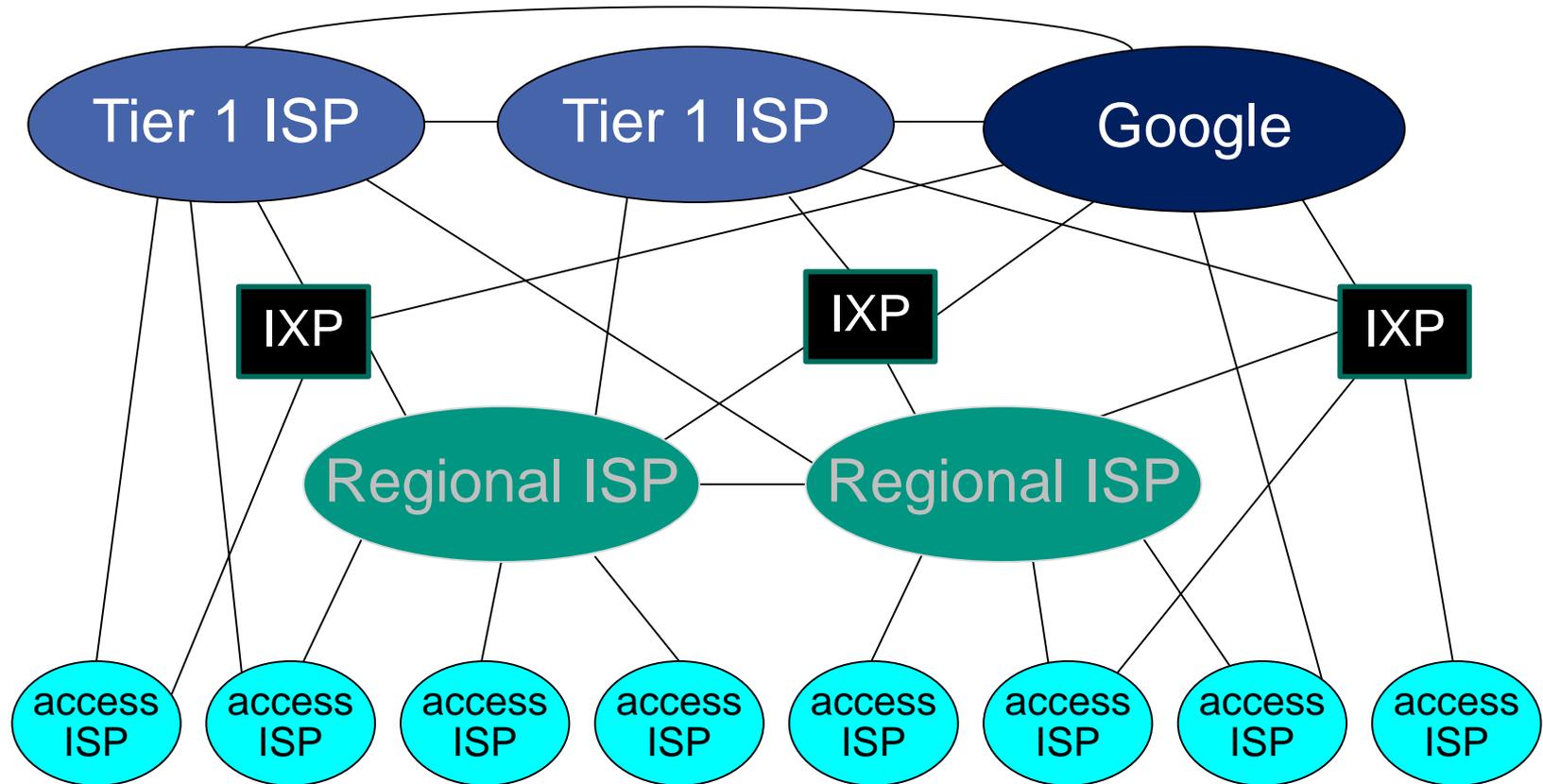
Interne Struktur: Netz von Netzen

- ... die Konkurrenten müssen miteinander vernetzt werden



Interne Struktur: Netz von Netzen

- Im Zentrum: wenige sehr gut verbundene Netze

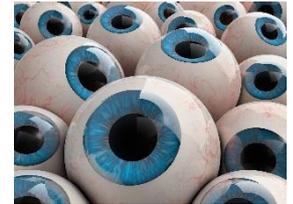


Internet-Service-Provider vs. Content-Provider

■ Verschiedene Arten von „Providern“ im Internet

■ Internet-Service-Provider (ISP)

- Binden „Konsumenten“ an das Internet an
- Oft mehrere tausend Kunden
- z.B. Deutsche Telekom (DTAG), 1&1, Vodafone, Unitymedia
- Aber auch Anbindung von Unternehmen und/oder Hosting



■ Content-Provider

- Bereitstellung / Erzeugung von Inhalten
- Ausführen von Anwendungen (z.B. bei Online-Games, Banking)
- z.B. Google, Netflix, Facebook, Amazon, Yahoo, Web.de, Spiegel, ...

■ Aber: ISPs und Content-Provider benötigen Konnektivität untereinander

- Bezahlung dafür? ISPs? Content-Provider? Beide?
- Gleichbehandlung von großen und kleinen Anbietern?
- Aktuelle Debatte dazu: **Netzneutralität**

c) Standardisierung / Entwicklung des Internets

■ Internet-Standards

- Online frei zugänglich: <http://rfc-editor.org/rfcsearch.html>
- z.B. HTTP (RFC 7230, RFC 7540) oder SMTP (RFC 5321)



■ Internet Engineering Task Force (IETF)



- Loser Zusammenschluss von „Experten“ (jeder kann mitmachen)
- Arbeiten in verschiedenen *Working Groups (WG)*
- Zusätzliche verschiedene Organe: *NomCom, IESG, IAB, IASA, IAOC*
- Konsens per Diskussion per RFCs, Treffen, Mailinglisten, „Humming“



■ Weitere Organisationen

■ Internet Society (ISOC)

- Non-profit organisation
- Standards, public policy, access, and education
- Dachorganisation der IETF

■ Internet Corporation for Assigned Names and Numbers (ICANN)

- Organisiert die Verwaltung von Namen und Adressen im Internet
- Vorher Internet Assigned Numbers Authority (IANA)

Things Attack the Internet!

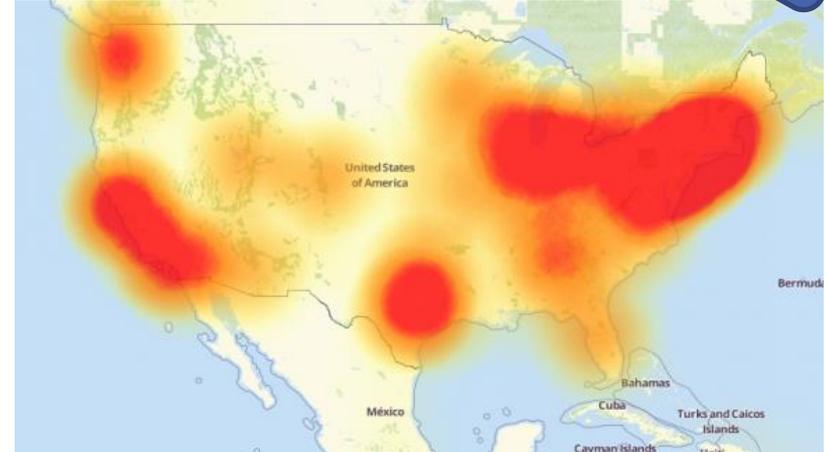
- Ab 18.09.2016:
DDoS-Angriff auf den Hoster OVH
 - Größter beobachteter Angriff in der Internetgeschichte, ca. 1,1 Terabit/s
 - >180.000 verschiedene Geräte
- 
[\[https://www.heise.de/newsticker/meldung/Rekord-DDoS-Attacke-mit-1-1-Terabit-pro-Sekunde-gesichtet-3336494.html\]](https://www.heise.de/newsticker/meldung/Rekord-DDoS-Attacke-mit-1-1-Terabit-pro-Sekunde-gesichtet-3336494.html)
- 20.09.2016:
DDoS-Angriff auf KrebsOnSecurity.com
 - ca. 620 Gigabit/s
- 
[\[http://krebsonsecurity.com/2016/09/krebsonsecurity-hit-with-record-ddos\]](http://krebsonsecurity.com/2016/09/krebsonsecurity-hit-with-record-ddos)
- 21.10.2016:
DDoS-Angriff auf DynDNS
 - 
[\[krebsonsecurity.com/2016/10/hacked-cameras-dvrs-powered-todays-massive-internet-outage\]](http://krebsonsecurity.com/2016/10/hacked-cameras-dvrs-powered-todays-massive-internet-outage)
- Fortsetzung folgt ...

Sep 18 10:49:12 tcp_ack 20Mpps 232Gbps
Sep 18 10:58:32 tcp_ack 15Mpps 173Gbps
Sep 18 11:17:02 tcp_ack 19Mpps 224Gbps
Sep 18 11:44:17 tcp_ack 19Mpps 227Gbps
Sep 18 19:05:47 tcp_ack 66Mpps 735Gbps
Sep 18 20:49:27 tcp_ack 81Mpps 360Gbps
Sep 18 22:43:32 tcp_ack 11Mpps 136Gbps
Sep 18 22:44:17 tcp_ack 38Mpps 442Gbps
Sep 19 10:13:57 tcp_ack 10Mpps 117Gbps
Sep 19 11:53:57 tcp_ack 13Mpps 159Gbps
Sep 19 11:54:42 tcp_ack 52Mpps 607Gbps
Sep 19 22:51:57 tcp_ack 10Mpps 115Gbps
Sep 20 01:40:02 tcp_ack 22Mpps 191Gbps
Sep 20 01:40:47 tcp_ack 93Mpps 799Gbps
Sep 20 01:50:07 tcp_ack 14Mpps 124Gbps
Sep 20 01:50:32 tcp_ack 72Mpps 615Gbps
Sep 20 03:12:12 tcp_ack 49Mpps 419Gbps
Sep 20 11:57:07 tcp_ack 15Mpps 178Gbps
Sep 20 11:58:02 tcp_ack 60Mpps 698Gbps
Sep 20 12:31:12 tcp_ack 17Mpps 201Gbps
Sep 20 12:32:22 tcp_ack 50Mpps 587Gbps
Sep 20 12:47:02 tcp_ack 18Mpps 210Gbps
Sep 20 12:48:17 tcp_ack 49Mpps 572Gbps
Sep 21 05:09:42 tcp_ack 32Mpps 144Gbps
Sep 21 20:21:37 tcp_ack 22Mpps 122Gbps
Sep 22 00:50:57 tcp_ack 16Mpps 191Gbps

Grafik: Eingehende Datenraten bei OVH im Angriffszeitraum

DDoS-Angriff 21.10.2016

- Dienste wie Twitter, Paypal, Netflix und Spotify für Stunden nicht nur schlecht erreichbar
- Ursache:
Gestörte DNS-Namensauflösung
 - Betroffene Dienste nutzten DNS-Dienstleistungen von DynDNS
- Hintergrund:
Distributed-Denial-of-Service-Angriff auf Infrastruktur von DynDNS
 - Von IP-Kameras und Vidorecordern ausgehend
 - Botnetz-Familie „Mirai“ aus schlecht gesicherten/gewarteten Geräten
 - Etwa: Telnet/SSH-Zugang mit Standardpasswörtern



Grafik: Räumliche Verteilung der betroffenen Dienste

Das Darknet

- Linktipp: <https://youtu.be/dmPgXYa24xg>



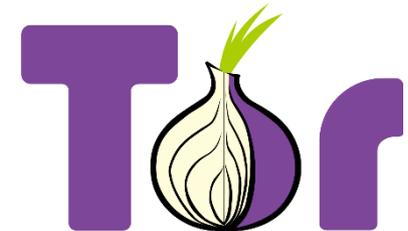
- „Darknet“ ist ein schwammiger Begriff

- „Overlay“ über das normale Internet
- Nutzung spezieller Software
- Zugangsadressen müssen vorher bekannt sein (sind geheim)
- Marktplatz: Waffen, Rauschgift, Falschgeld, Kreditkarten, Hacker Tools, ...
- Bezahlung: Bitcoin damit sich Transaktionen nicht nachvollziehen lassen



- Ein Beispiel für Darknet

- Onion-Routing mit Tor
- Gewährleistet Sender- und Empfängeranonymität
- Unterstützt “hidden services” (.onion-Adressen)

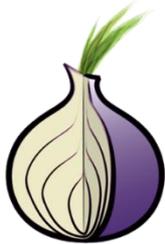


- Aber!

- Darknet wird nicht nur für illegale Zwecke genutzt
- z.B. auch zum Schutz der Privatsphäre oder für [Instituts-Feedback-Tool](#)

Feedback zur Vorlesung

- Hintergrund: normale Evaluation (1x/Semester) ist zu langsam...



Anonymes Feedback zu Vorlesung und Institut

- Wie?

- Tor-Browser herunterladen: <https://torproject.org/projects/torbrowser.html>

- Auch für Android verfügbar: Orfox



- Feedback abgeben (via Tor) unter: <http://tmfeedozvp6v65nf.onion/>

- Nicht-anonymes Feedback empfangen wir natürlich auch weiterhin sehr gern per E-Mail oder live nach der Vorlesung



Offene Fragen: e) + f)

- Vorteile des Internets gegenüber analoger Welt
 - Schneller, effizienter, Skalierung
 - Benutzbarkeit
 - Global
 - ...

- Vorteile der analogen Welt gegenüber dem Internet
 - Emotionaler, direkter menschlicher Kontakt
 - Mehrere Kanäle gleichzeitig (Geruch, Haptik, ...)
 - Höhere Auflösung 😊
 - ...

- Bei welchen Tätigkeiten / Situationen wird das Internet auch in Zukunft keine Rolle spielen?
 - Mündliche Prüfungen?
 - Autonomes Fahren?
 - Handwerkliche Tätigkeit?
 - ...

Einführung in Rechnernetze – 1. Übungsblatt

1. Allerlei zum Internet
2. Das Hypertext Transfer Protocol (HTTP)
3. Datenübertragung und Verzögerung
4. HTTP in der Praxis

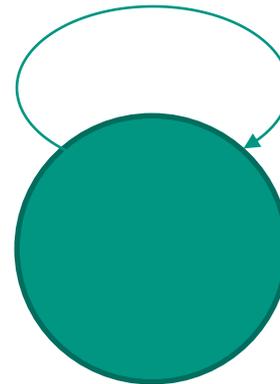
Hypertext - Definition

- Struktur von Dokumenten in Rechnernetzen (z.B. dem World Wide Web), die Verweise auf andere Dokumente beinhalten (Hyperlink) und die Möglichkeit bieten, durch Anklicken dieses Verweises das korrespondierende Dokument direkt aufzurufen.

Quelle: Gabler Wirtschaftslexikon [<http://wirtschaftslexikon.gabler.de/Definition/hypertext.html>]

HTTP: Zustandslosigkeit

- Request-Response-Pattern
- Keine Zustandsänderung zwischen mehreren Anfragen
 - Eventuell vorhandene Zustände (Login, Facebook-Posts, ...) sind Zustände der Anwendungen, die HTTP nutzen



HTTP-Request-Header (Auswahl)

- Host: www.kit.edu
 - Host kennt sich doch eigentlich
 - Aber: erlaubt mehrere Domänen für einen einzigen Server

- User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/57.0.2987.133 Safari/537.36
 - Kennung der Anwendung
 - Hier: Chrome 57
 - Erlaubt dem Server, Features zu deaktivieren, die der Browser nicht unterstützt
 - Warum so kompliziert?

- Accept-Language: de-DE,de;q=0.8,en-US;q=0.6,en;q=0.4
 - Browser weiß, welche Sprachen sein Nutzer versteht
 - Gibt Interface-Sprachen mit Präferenz an
 - Server wählt am besten passende Sprache

HTTP-Response-Header (Auswahl)

- Server: Apache/2.4.10 (Debian)
 - Verwendeter Server
 - Warum?
 - Gefahren?

- Set-Cookie: PHPSESSID=g0p26838900uog9nc6017g1jf1; path=/; HttpOnly
 - Server kann Informationen ablegen, die der Browser in der nächsten Anfrage sendet
 - Session-Information, Einstellungen, ...

- Cache-Control, Pragma, Date, Expires
 - Information für Caches

- Connection: Keep-Alive
 - Reaktion auf Request-Header
 - Hier: Nutzung von persistent-HTTP zugestimmt

```
HTTP/1.1 200 OK
Date: Fri, 31 Mar 2017 07:53:06 GMT
Server: Apache/2.4.10 (Debian)
Keep-Alive: timeout=15, max=96
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8
Content-Length: 230
```

■ Vorteile:

- Besser lesbar für Entwickler
- Erweiterbar
- Optionale Header müssen nicht gesendet werden

■ Nachteile:

- Aufwändiger zu interpretieren
- Größerer Kommunikations-Overhead
- Server muss mit fehlender Information rechnen

Einführung in Rechnernetze – 1. Übungsblatt

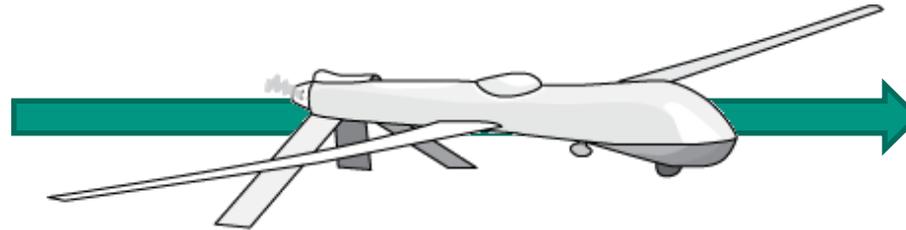
1. Allerlei zum Internet
2. Das Hypertext Transfer Protocol (HTTP)
3. Datenübertragung und Verzögerung
4. HTTP in der Praxis

a) Datentransport mit Drohnen

- Sie sollen eine große Datenmenge vom Campus Süd zum Campus Nord übertragen. Ein Kommilitone schlägt vor, Lieferdrohnen zu verwenden. Die Flugstrecke beträgt 10km.
- Eine Lieferdrohne erreicht eine Geschwindigkeit von 90km/h und transportiert eine Festplatte mit einer Kapazität von 4TByte. Es befindet sich immer nur genau eine Drohne in der Luft. Am Campus Nord gelandete Drohnen verbleiben dort und müssen nicht zurückfliegen.
- Vernachlässigen Sie im Folgenden die Zeit welche die Drohne zum Starten, Beschleunigen, Abbremsen und Landen benötigt. Gehen Sie davon aus, dass die Daten bereits passend auf Festplatten gespeichert sind und so keine weiteren Verzögerungen anfallen.

- Wie viel Zeit benötigt eine Lieferdrohne für den Weg vom Campus Süd zum Campus Nord?
- Welcher Datenrate entspricht dies, angenommen die Speicherkapazität der transportierten Festplatte wird vollständig ausgenutzt?

a) Datentransport mit Drohnen



- Geschwindigkeit: 90km/h
- Flugstrecke: 10km
- Datenmenge pro Drohne: 4TByte

- Gesucht:
 - Ausbreitungsverzögerung? t_a
 - Datenrate? r
 - „Packzeit“? t_s

Ausbreitungsverzögerung t_a

- Zeitspanne zwischen Absenden eines Signals und dessen Eintreffen am anderen Ende des Mediums
 - Abhängig von
 - Ausbreitungsgeschwindigkeit v
 - in üblichen Medien (Kabel, Glasfaser) etwa 2/3 Lichtgeschwindigkeit
 - Länge des Mediums d

$$t_a = \frac{d}{v}$$

Einbezug der Ausbreitungsgeschwindigkeit



Signal
senden



Medium

Signal
empfangen



Entfernung in Meter



Ausbreitungsverzögerung in Sekunden

Datenraten

- Geschwindigkeit, mit der Daten im Netz übertragen werden
 - Typischerweise gemessen in

Einheit		Bezeichnung
bit/s		„bit pro Sekunde“ / „bit per second“
kbit/s	1000 bit/s	„Kilo bit pro Sekunde“
Mbit/s	10^6 bit/s	„Mega bit pro Sekunde“
Gbit/s	10^9 bit/s	„Giga bit pro Sekunde“
Tbit/s	10^{12} bit/s	„Tera bit pro Sekunde“
Pbit/s	10^{15} bit/s	„Peta bit pro Sekunde“

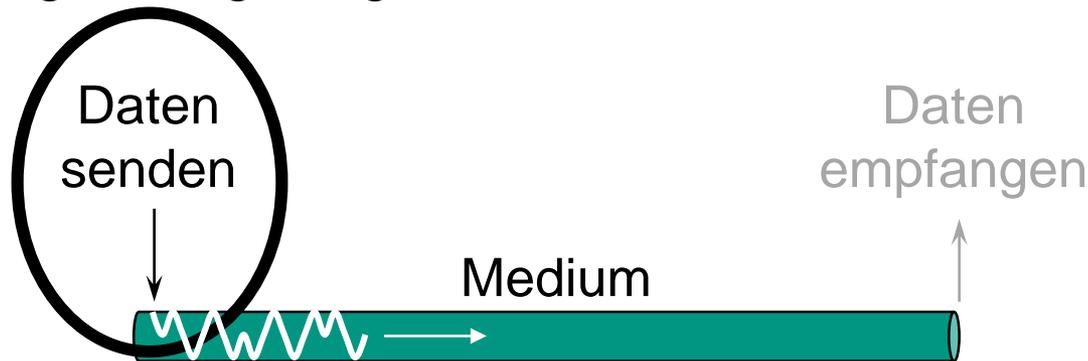
- Achtung: Stets Multiplikator (1000 vs. 1024) beachten
 - 1 kbit/s = 1000 bit/s

Sendezeit t_s

- Zeit zwischen Beginn und Abschluss der Sendung
 - Abhängig von
 - Datenmenge X
 - Datenrate r des Mediums (Leitung/Verbindung)
 - Anzahl der pro Zeiteinheit übertragenen Daten (i.d.R. bit/s)

$$t_s = \frac{X}{r}$$

- Achtung: Nach Abschluss der Sendung sind die Daten noch nicht beim Empfänger!
 - ⇒ Ausbreitungsverzögerung



a) Datentransport mit Drohnen

■ Ausbreitungsverzögerung: $t_a = \frac{d}{v}$

■ $t_a := \frac{10\text{km}}{90\frac{\text{km}}{\text{h}}} = \frac{1}{9}h = 400s$

■ Datenrate: $r := \frac{\text{Datenmenge}}{\text{Zeit}}$

■ $r := \frac{4\text{TByte}}{400s} = 10\frac{\text{GByte}}{s} = 80\frac{\text{Gbit}}{s}$

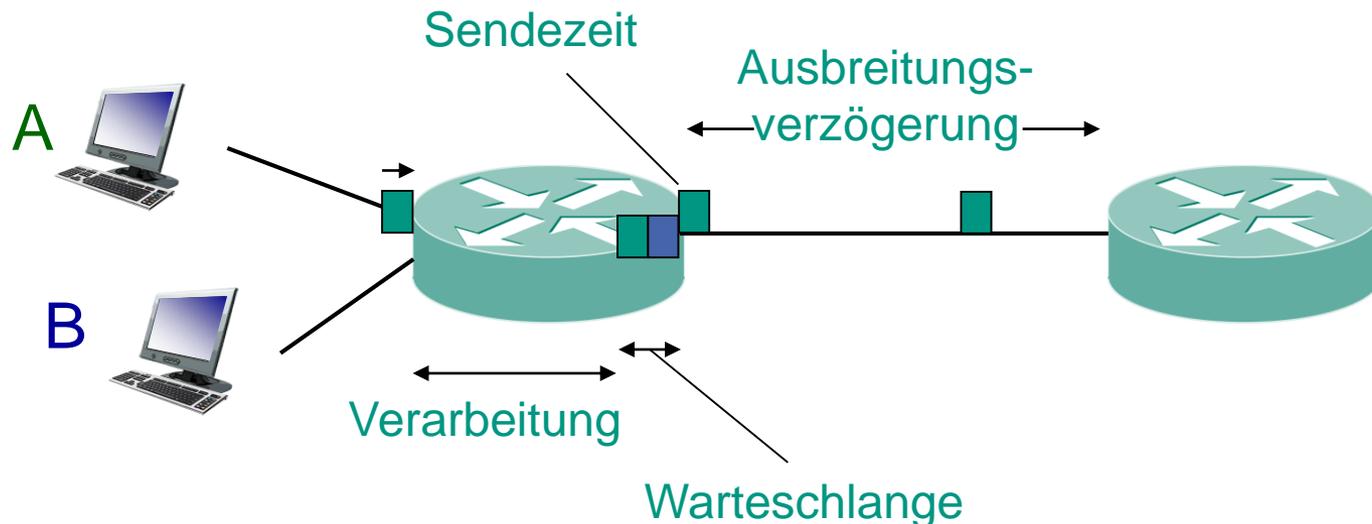
■ Packzeit (Sendezeit): $t_s := \frac{\text{Datenmenge}}{\text{Datenrate}}$

■ $t_s := \frac{4\text{TByte}}{80\frac{\text{Gbit}}{s}} = \frac{4.000\text{Gbit}}{10\frac{\text{Gbit}}{s}} = 400s$



Verzögerungen in einem Router im Überblick

- Übertragungsdauer
 - Sendezeit + Ausbreitungsverzögerung
- Weitere Verzögerungen in einem Router
 - Pufferung der Daten in Warteschlange (Queue)
 - Warten bis Ausgangslink für Übertragung verfügbar
 - Abhängig von Last für diesen Link
 - Verarbeitung
 - Überprüfung hinsichtlich Bitfehler, Bestimmung des Ausgangslinks ...
 - i.d.R. < Millisekunden



b) Verzögerung: non-persistent vs. persistent HTTP

- Die Website `www.myawesomewebsite.com` besteht aus einem HTML-Dokument, welches 4 Bilder, 2 Skripte und 1 Stylesheet von der gleichen Domain, und 2 Skripte von dem Werbenetzwerk `net.ad` einbindet. Die `net.ad`-Skripte binden zusätzlich noch 4 Bilder von ihren eigenen Servern in die Website ein. Die Übertragung einer kleinen Nachricht hin und zurück dauert zur `www.myawesomewebsite.com` $RTT_1 = 200\text{ms}$, zu `net.ad` $RTT_2 = 50\text{ms}$, in jede Richtung gleich. Die Bilddateien sind je $X = 100\text{KByte}$ groß, und die Übertragungsrate betrage $r = 5\text{MByte/s}$. Die Sendezeiten anderer Dateitypen sei vernachlässigbar.
- Wie lange dauert der Aufbau der Seite, wenn Sie non-persistent HTTP einsetzen? Wie viel Zeit können Sie durch den Einsatz von persistent HTTP sparen?

HTTP-Dauer

■ Zu ladende Dokumente:

- <http://www.myawesomesite.com/>
 - 1 HTML-Dokument
 - 1 Stylesheet
 - 2 Skripte
 - 4 Bilder} 8 Objekte

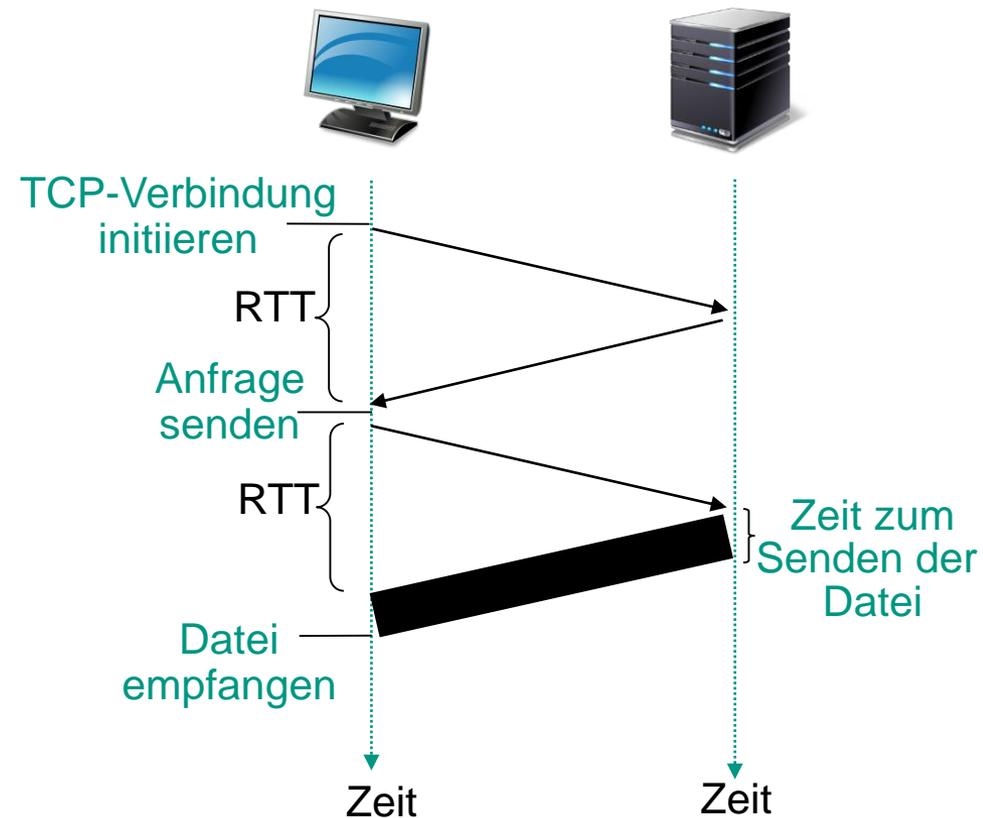
- <http://net.ad/>
 - 2 Skripte
 - 4 Bilder} 6 Objekte

■ RTTs:

- $RTT_1 = 200\text{ms}$
- $RTT_2 = 50\text{ms}$

Non-persistent HTTP: Antwortzeit

- Round-Trip-Time (RTT)
 - Zeit, die ein Paket vom Sender zum Empfänger und zurück benötigt
- HTTP-Antwortzeit
 - Eine RTT für Verbindungsaufbau
 - Eine RTT für HTTP-Anfrage und erste Bytes der Antwort
 - Zeit t_s für das Senden der Datei
- Antwortzeit: $2 \cdot \text{RTT} + t_s$



HTTP-Dauer

■ Zu ladende Dokumente:

- <http://www.myawesomesite.com/>
 - 1 HTML-Dokument
 - 1 Stylesheet
 - 2 Skripte
 - 4 Bilder
 - <http://net.ad/>
 - 2 Skripte
 - 4 Bilder
- } 8 Objekte
- } 6 Objekte

■ RTTs:

- $RTT_1 = 200\text{ms}$
- $RTT_2 = 50\text{ms}$

■ Sendezeit für ein Bild

- $t_s = X/r = 20\text{ms}$

HTTP-Dauer – non-persistent-HTTP

8 Objekte
von myawesomewebsite.com

$$t = 8 \cdot 2 \cdot \text{RTT}_1 + 6 \cdot 2 \cdot \text{RTT}_2 + 8 \cdot t_s = 3960\text{ms}$$

je 2 RTT pro Objekt

net.ad

8 Objekte

$$t = (8 + 1) \cdot RTT_1 + (6 + 1) \cdot RTT_2 + 8 \cdot t_s = 2230\text{ms}$$

plus 1 RTT Verbindungsaufbau

Zeitersparnis: 3960ms - 2230ms = 1730ms

Einführung in Rechnernetze – 1. Übungsblatt

1. Allerlei zum Internet
2. Das Hypertext Transfer Protocol (HTTP)
3. Datenübertragung und Verzögerung
4. HTTP in der Praxis

```
HEAD / HTTP/1.1  
Host: www.kit.edu
```

- Bedeutung
 - Root-Dokument (“/”) – <http://www.kit.edu/>
 - Nur HTTP-Header, nicht das eigentliche Dokument

- Antwort des Servers
 - Server antwortet mit Header

- Gesamtes Dokument anfordern:
 - GET statt HEAD in erster Zeile

(b) – <http://i72tmdjango.tm.kit.edu/first.txt>

```
GET /first.txt HTTP/1.1  
Host: i72tmdjango.tm.kit.edu
```

(c) – Query Strings

- Querystrings
 - “?” an Ende des Pfades
 - mehrere Schlüssel-Wert-Paare
 - Durch “&” getrennt

userid=5&password=completelyInsecureStuff

HTTP-Anfrage:

```
GET /very_insecure_login?userid=5&password=completelyInsecureStuff HTTP/1.1  
Host: i72tmdjango.tm.kit.edu
```

(d) – Passwörter in URL

- URLs sind öffentlich zugängliche Informationen
 - Passwort auf Bildschirm sichtbar
 - Passwort im Browserverlauf
 - Darf von Caches gespeichert werden



(e) – HTTP-POST

- Zusätzlich zu spezifizieren
 - Content-Type: Wie wird der Content interpretiert?
 - Um ähnlich wie Query Strings zu arbeiten: application/x-www-form-urlencoded
 - Content-Length: Länge des Body
 - (muss bereits hier bekannt sein!)

POST /insecure_login HTTP/1.1

Host: i72tmdjango.tm.kit.edu

Content-Type: application/x-www-form-urlencoded

Content-Length: 42

userid=5&password=completelyInsecureStuff

(f) – Chat

■ Lesen von Nachrichten:

```
GET /chat HTTP/1.1  
Host: i72tmdjango.tm.kit.edu
```

■ Senden einer Nachricht:

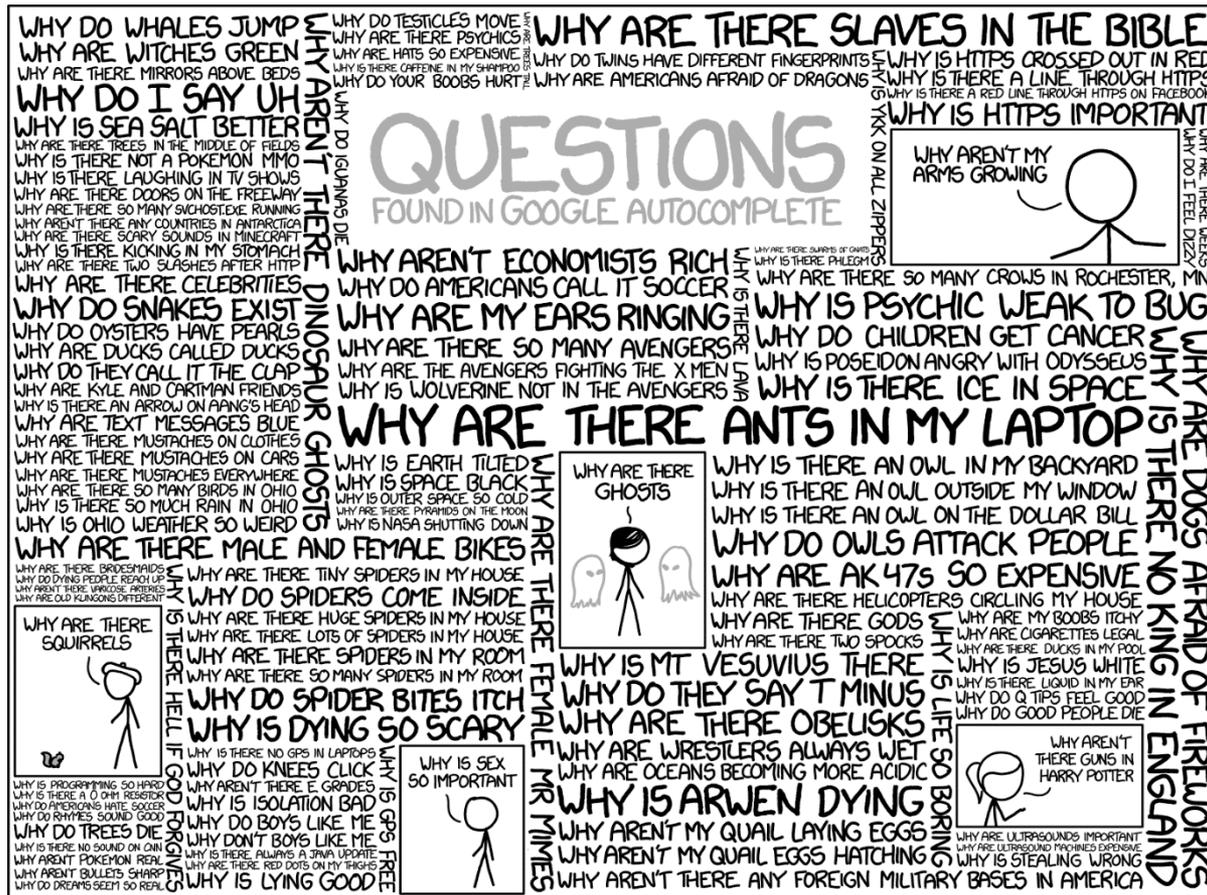
```
POST /chat HTTP/1.1  
Host: i72tmdjango.tm.kit.edu  
Content-Type: application/x-www-form-urlencoded  
Content-Length: 26
```

```
user=someone&message=Hallo
```

(f) – Erkennen von Browsern

- Browser sendet einen User Agent im Header
- Server reagiert auf Vorhandensein eines UA

```
GET / HTTP/1.1
Host: www.kit.edu
Connection: keep-alive
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/57.0.2987.133 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
DNT: 1
Accept-Encoding: gzip, deflate, sdch
Accept-Language: de-DE,de;q=0.8,en-US;q=0.6,en;q=0.4
```



■ Nächste Rechnernetze-Übung am 17.05.2017