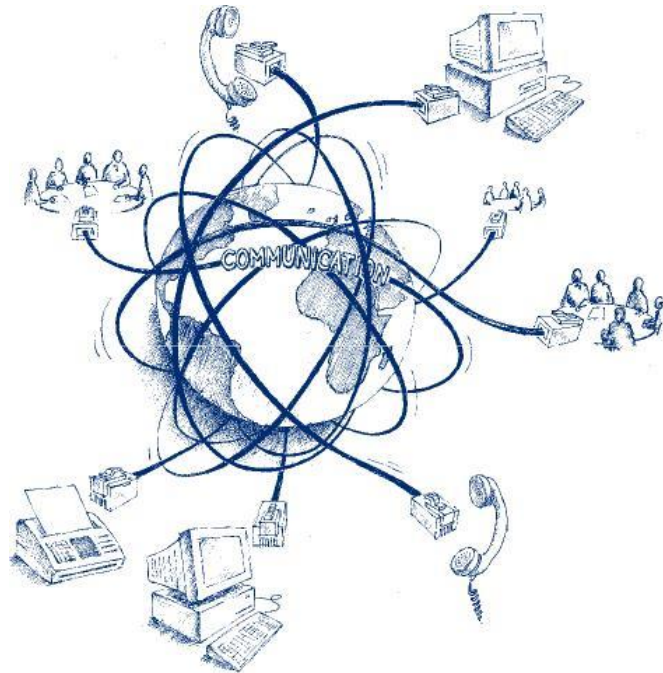


# Optimizing Cooperation of HTTP/2 and Multipath TCP



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT



Betreuer: Alexander Frömmgen  
Vortragender: Maximilian Weller

Prof. Dr.-Ing. Ralf Steinmetz  
KOM - Multimedia Communications Lab

Source: [http://www.sycor-asia.com/opencms/as/products\\_services/complementary\\_services/Telecommunication/](http://www.sycor-asia.com/opencms/as/products_services/complementary_services/Telecommunication/)

PPT-for-all\_\_v.3.4\_office2010\_\_2012.09.10.pptx

8-May-17

© author(s) of these slides including research results from the KOM research network and TU Darmstadt; otherwise it is specified at the respective slide

Template all v.3.4

- Grundlagen
- Ziel der Arbeit
- Herangehensweise
- Implementierung
- Evaluation
- Ergebnis
- Weitere Ideen

## HTTP/2

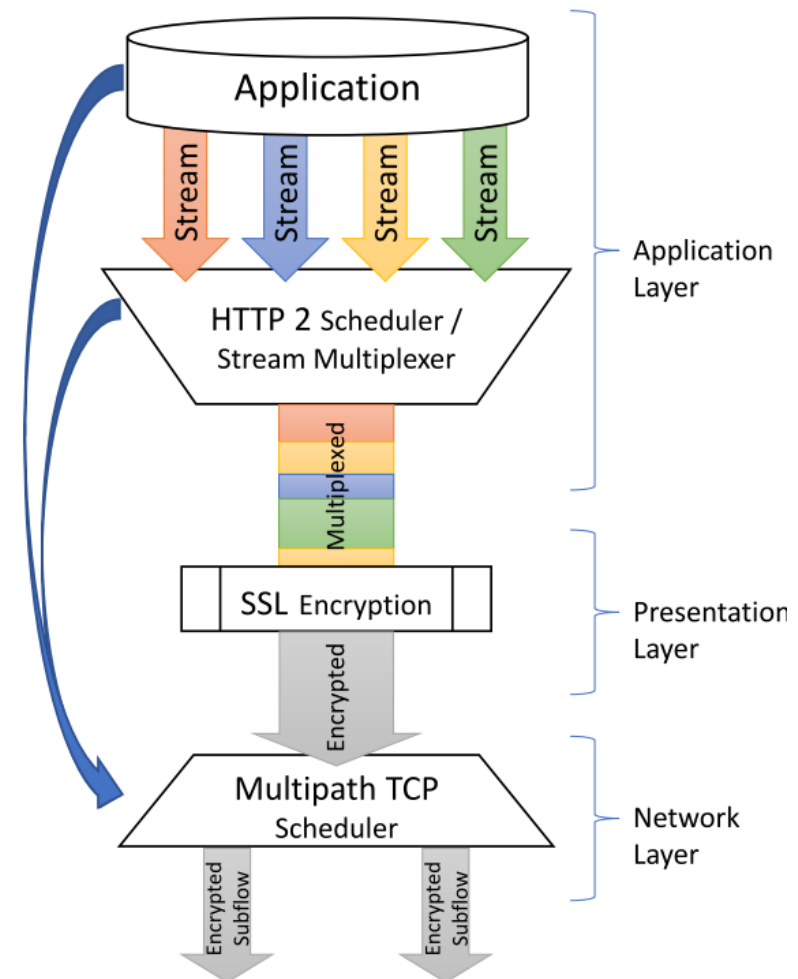
- HyperText Transfer Protocol
- Semantik beibehalten
- effizienteres Binärprotokoll
- für uns wichtig: **Multiplexing**

## Multipath TCP

- Transmission Control Protocol
- mehrere Pfade zu einer TCP-Verbindung „zusammenschalten“
- → schneller, zuverlässiger

## Ziel

- Bessere Entscheidungen durch Kooperation

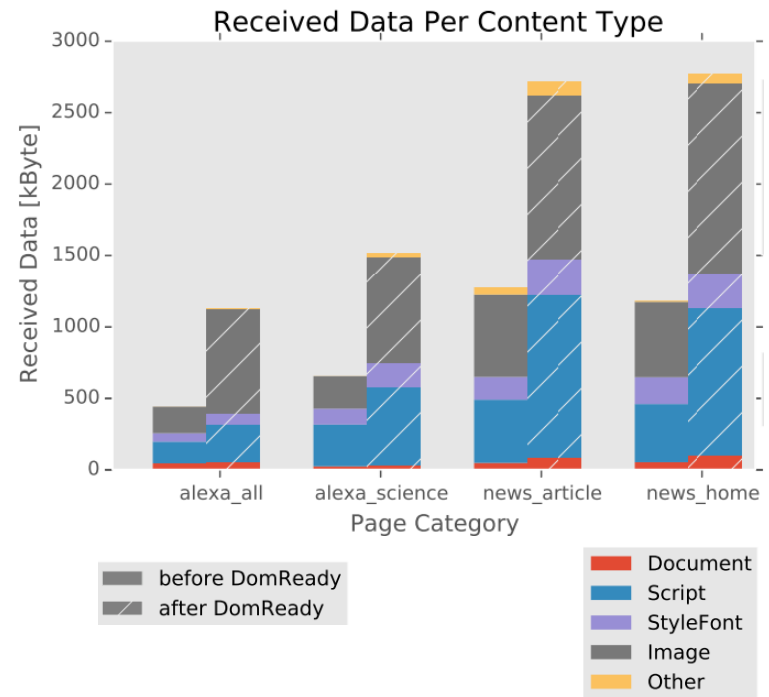


## Metriken für Webseiten-Performance

- DOMContentLoaded
- First Meaningful Paint
- Load

## Analyse des Aufbaus von Webseiten

- Bilder und Skripte
  - größte Datenmenge, Anzahl Requests
  - teilw. nach DOM
- Document und Style
  - vor DOM



# Herangehensweise

---

## Kooperation

- Anwendung gibt beschreibende Hinweise
- Multipath TCP Scheduler bezieht diese in Entscheidung ein
- alternativ: Anwendung wählt zu verwendenden Algorithmus

## Optimierungen

- Bei Verbindungen mit unterschiedlicher Latenz:
  - Nur eilige Ressourcen über schnellere, aber teurere Verbindung senden
- Bei Paketverlust / schwankender Latenz:
  - Latenzkritische Anfragen redundant über mehrere Verbindungen senden
- Wenn Übertragung fast fertig:
  - Letzte Segmente erneut senden

# Implementierung

---

## Webserver als Grundlage

- *Nghttpd*
  - Referenzimplementierung für *libnghttp2*
    - verwendet in Apache HTTP Server
  - Volle HTTP/2-Unterstützung
  - Gut anpassbar

## Multipath TCP

- Implementierung im Linux-Kernel
- RBS: durch Skripte steuerbarer Scheduler
- Kooperation über Socket Options:
  - Auswahl des Scheduler-Skripts
  - Metainformation zu Datenstrom
  - Register

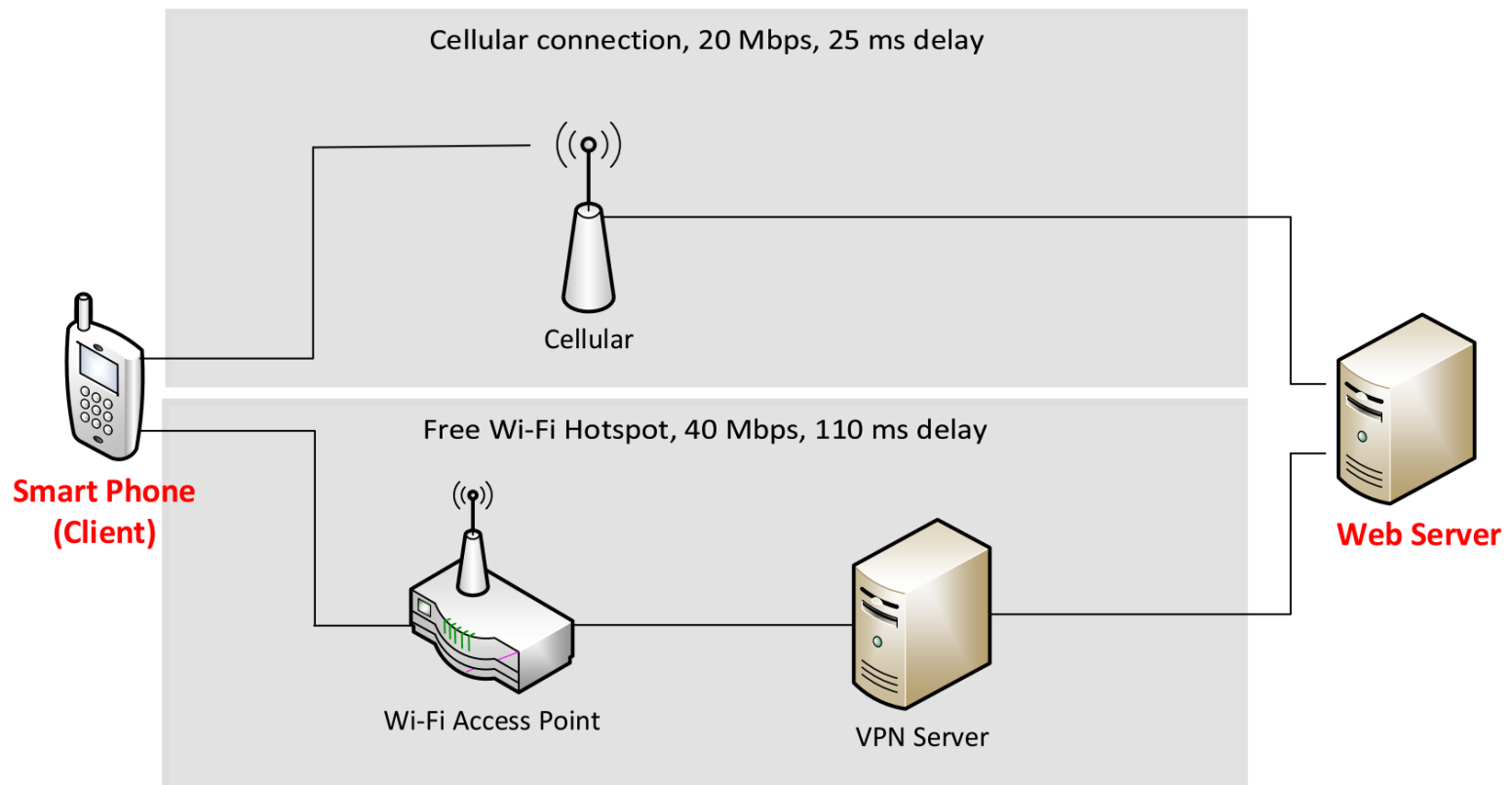
## Anpassungen an Nghttpd

- RBS-Skript auswählen
- Content Type der Streams ermitteln
- Metainformation über Buffer beibehalten, an Scheduler weiterreichen
- Füllstand des Buffers über Register weitergeben

## Messungen automatisieren

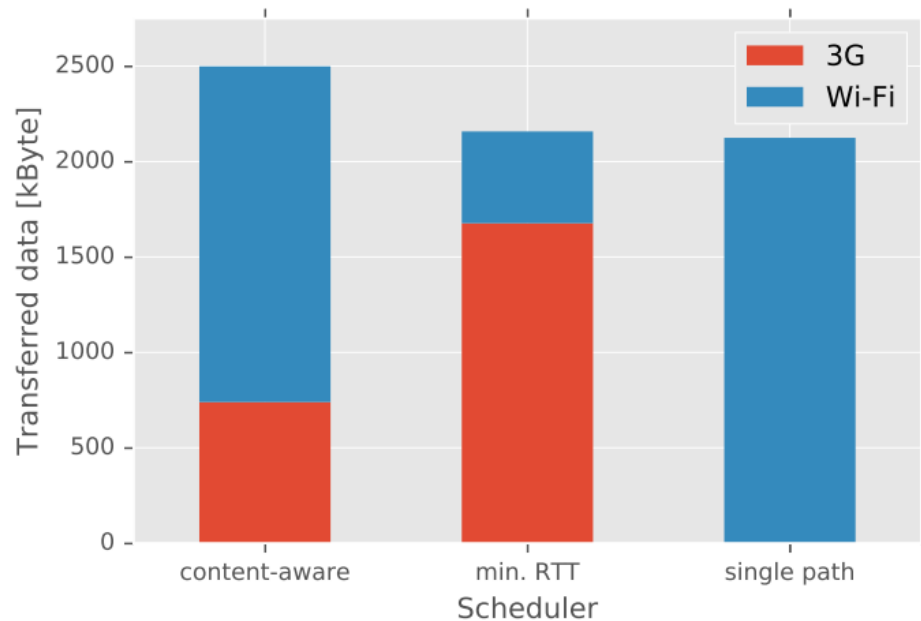
- Google Chrome Entwicklertools
- Ladezeiten
- Ressourcen: Anzahl, Größe, Typ, Origin

# Evaluation

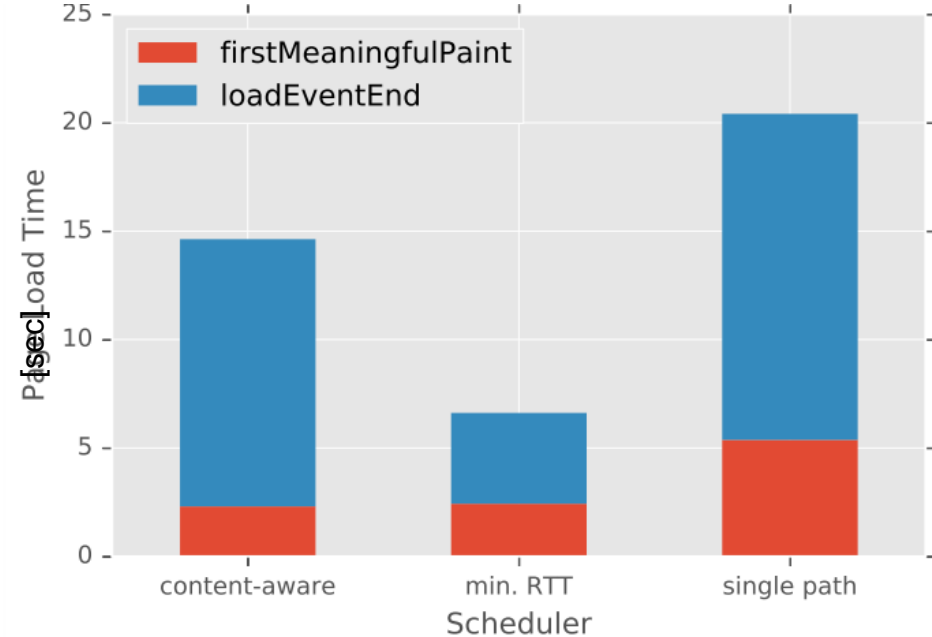




# Evaluation



(a) Received bytes per interface for Evaluation B



(b) Mean page load times for Evaluation B

- Durch Multipath TCP **halbierte Ladezeit**
- Mit Scheduling-Hinweisen **nur 34% der Daten auf „teurem“ Pfad**

## Ergebnisse

- Optimierung durch Hinweise aus der Anwendungsschicht möglich
- Bessere Kompromisse zwischen Geschwindigkeit und Ressourcennutzung
  
- Abhängig von Netzwerksituation
  - z.B. Verbindungen mit unterschiedlicher Latenz

## Weitere Ideen

- Scheduler auf Clientseite einbeziehen
- Untersuchung mit (simuliertem) Nutzerverhalten auf dynamischen Webseiten
- Zur praktischen Nutzung: Portierung auf z.B. Apache
- Scheduler anhand Netzwerksituation wählen

---

# Thank you for your attention! Questions?

---



# Headline

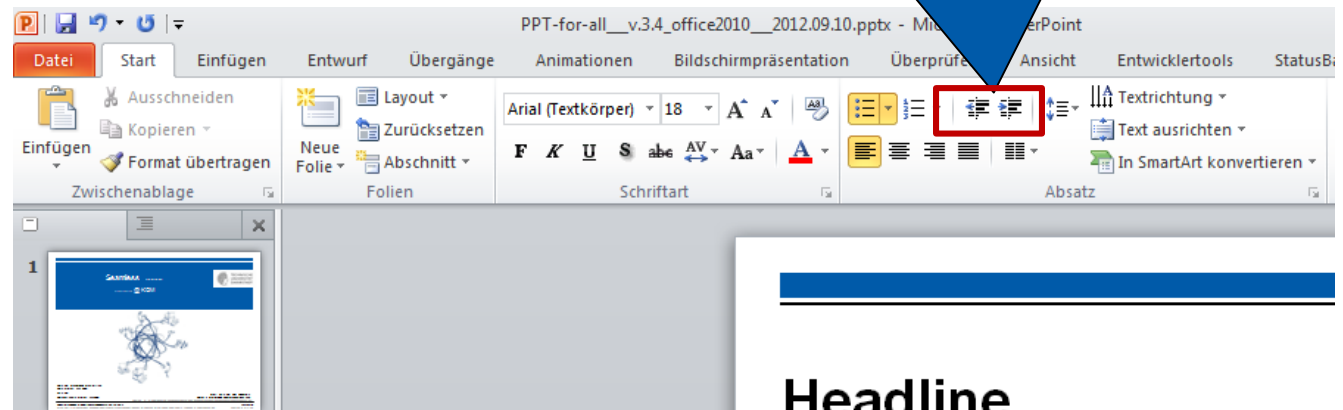
## Level 0

- Level 1
- Level 1
  - Level 2
  - Level 2

## Another Level 0

- Level 1
- Level 1

Please use these buttons  
for creating bullets!



Headline