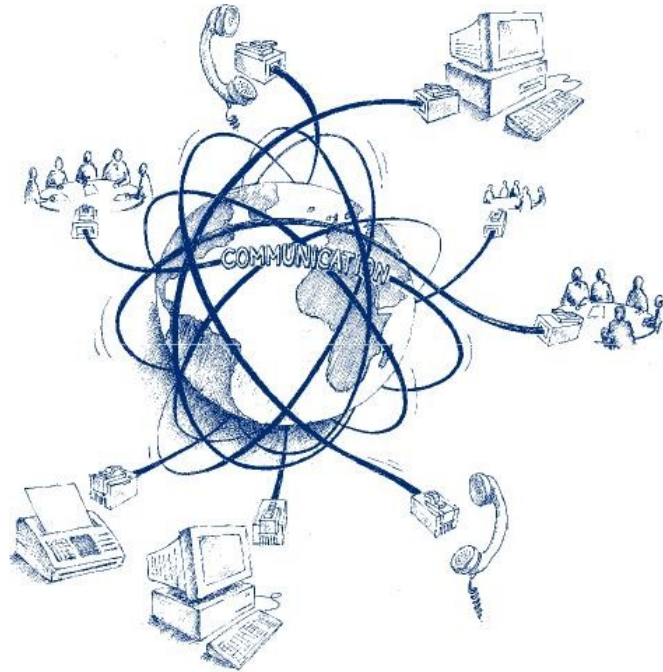


Optimierte Kooperation von HTTP/2 und Multipath TCP



TECHNISCHE
UNIVERSITÄT
DARMSTADT



Betreuer: Alexander Frömmgen

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

Source: http://www.sycor-asia.com/opencms/as/products_services/complementary_services/Telecommunication/



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

HTTP/2

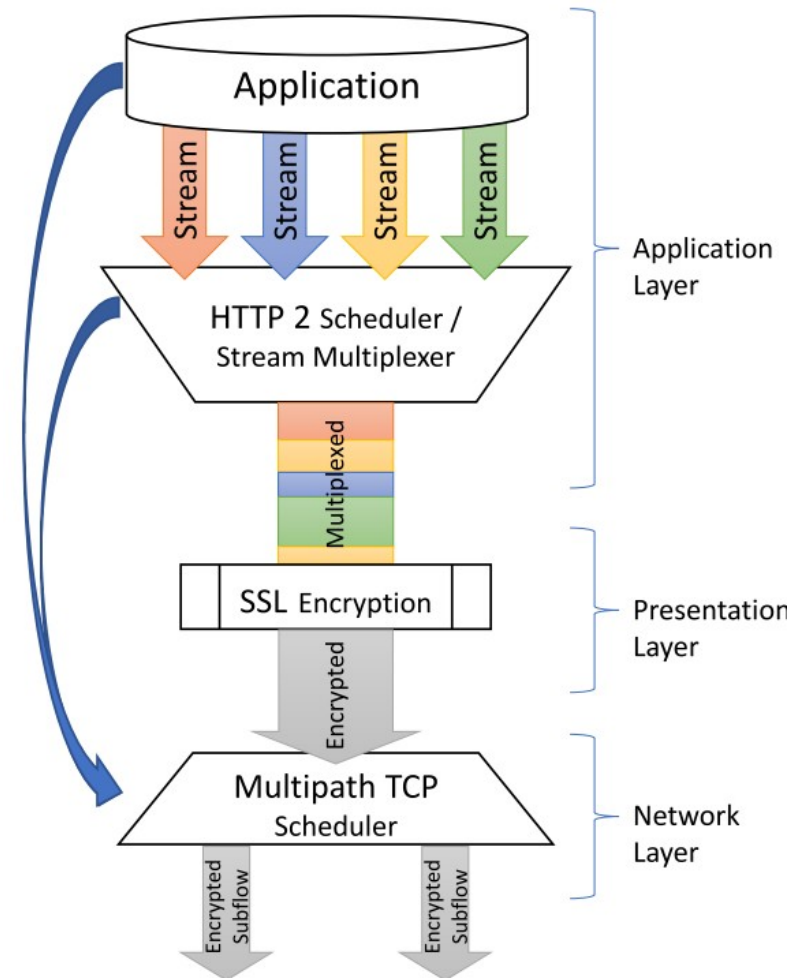
- HyperText Transfer Protocol
- Semantik beibehalten
- effizienteres Binärprotokoll
- für uns wichtig: **nur eine TCP-Verbindung**

Multipath TCP

- Transmission Control Protocol
- mehrere Pfade zu einer MPTCP-Verbindung „zusammenschalten“
- □ schneller, zuverlässiger

Ziel

- Bessere Scheduler-Entscheidungen durch Kooperation

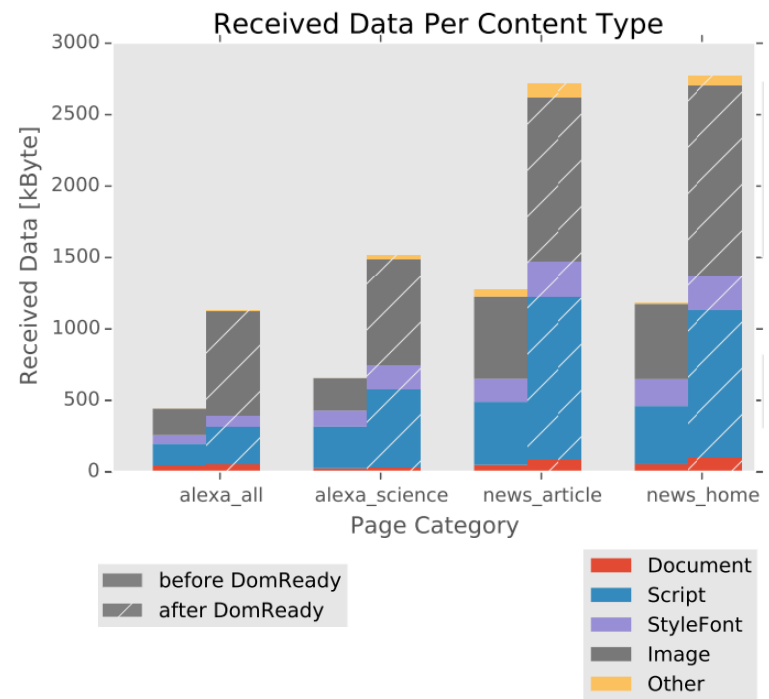


Metriken für Webseiten-Performance

- DOMContentLoaded
- First Meaningful Paint
- Load

Analyse des Aufbaus von Webseiten

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



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



HTTP/2 Scheduler /
Stream Multiplexer

Multipath TCP
Scheduler



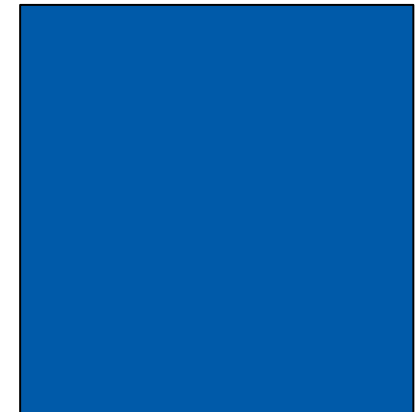
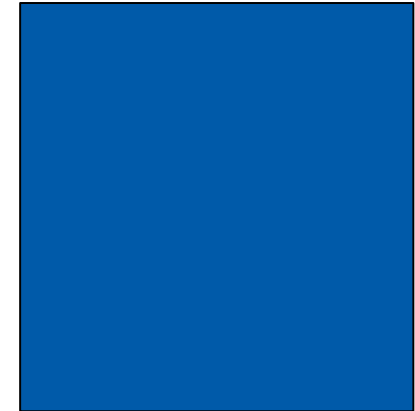
Implementierung

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

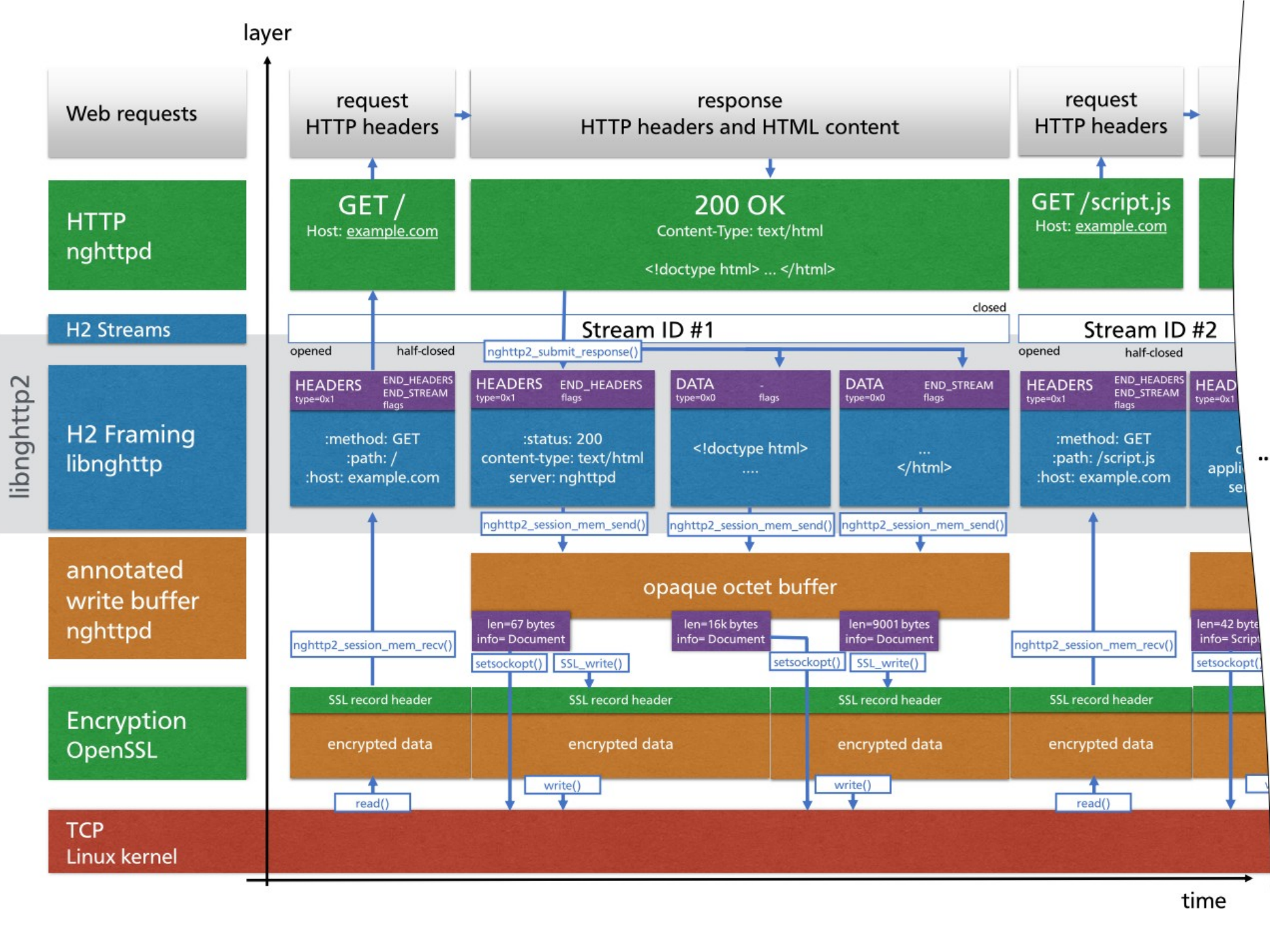


Anpassungen an Nghttpd

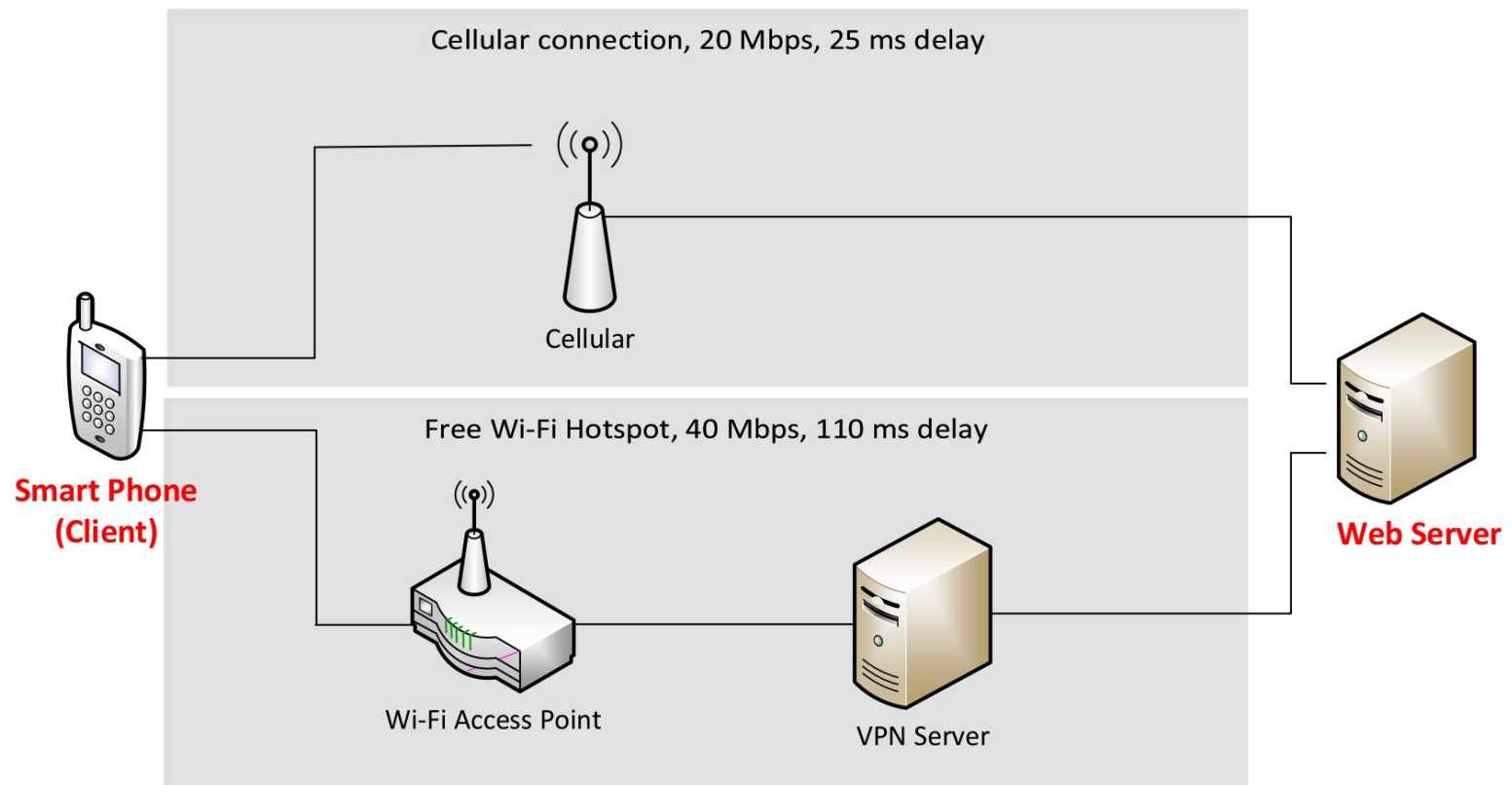
- RBS-Skript auswählen
- Content Type der Streams ermitteln
- Metainformation über Buffer beibehalten, an Scheduler weiterreichen
- Füllstand des Buffers über Register weitergeben

Messungen automatisieren

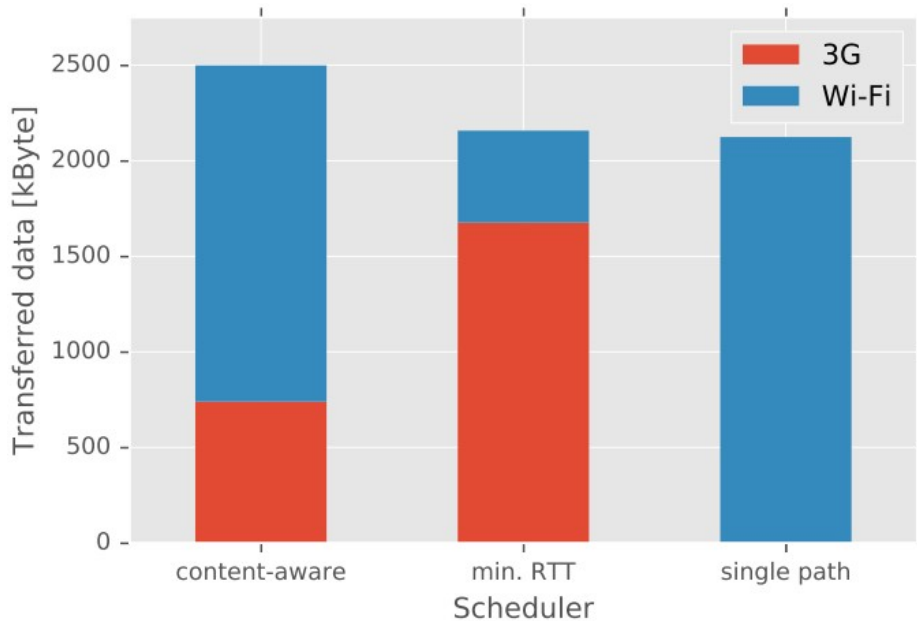
- Google Chrome Entwicklertools
- Ladezeiten
- Ressourcen: Anzahl, Größe, Typ, Origin



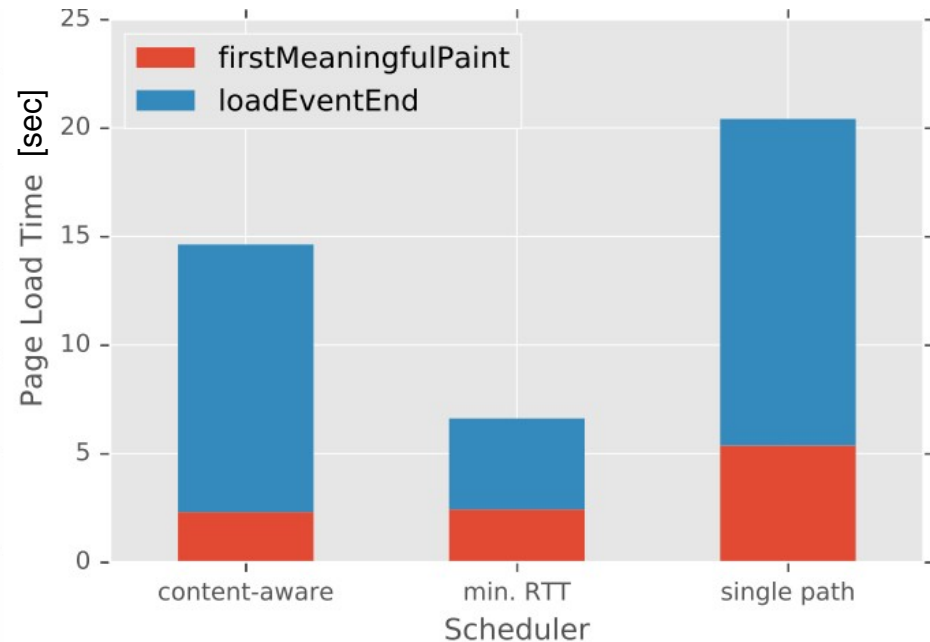
Evaluation



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
- Verbesserung der automatischen Content-Type-/Prioritäts-Erkennung

Thank you for your attention! Questions?



TECHNISCHE
UNIVERSITÄT
DARMSTADT

