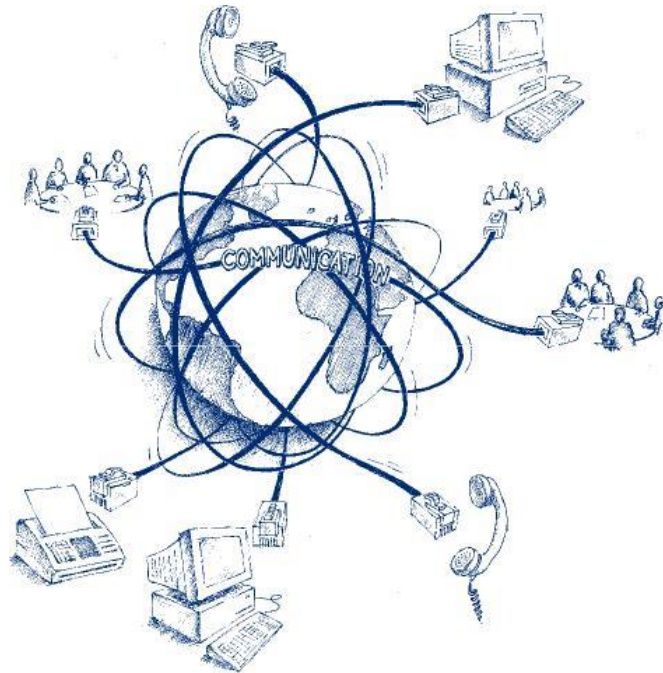


Optimierte Kooperation von HTTP/2 und Multipath TCP



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/2

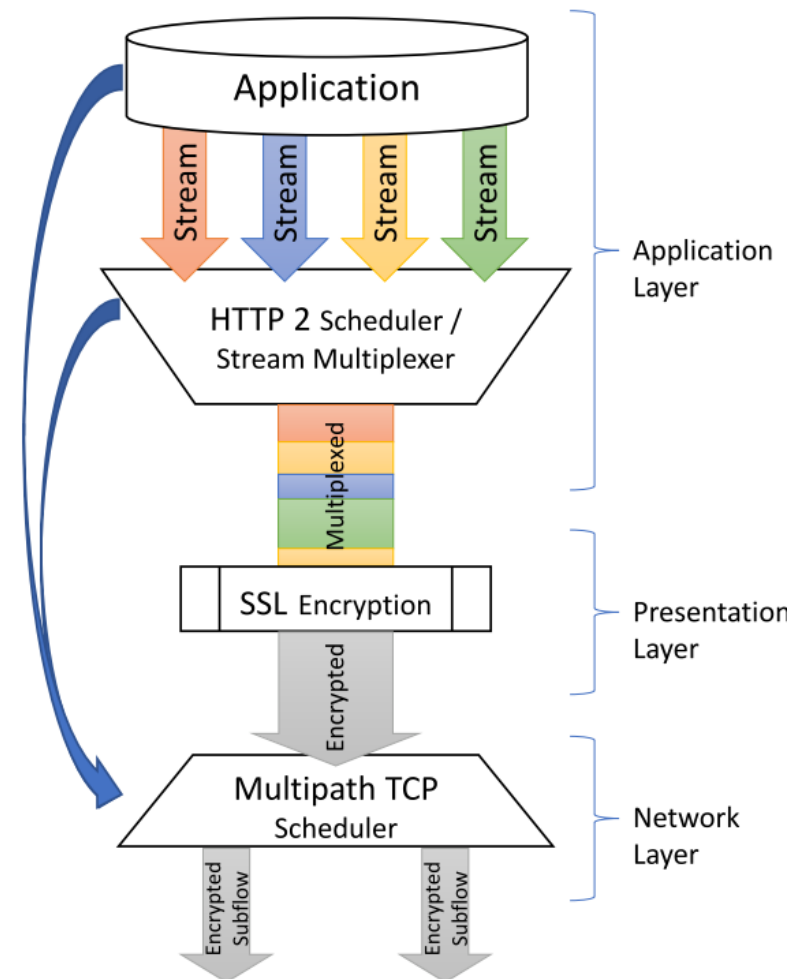
- HyperText Transfer Protocol
- Semantik beibehalten
- effizienteres Binärprotokoll
- für uns wichtig: **nur eine TCP-Verbindung durch Multiplexing**

Multipath TCP

- Transmission Control Protocol
- mehrere Pfade zu einer MPTCP-Verbindung „zusammenschalten“
- → schneller, zuverlässiger
- Scheduler ordnet und verteilt Segmente

Ziel

- Bessere Entscheidungen / Abwägungen im Scheduler durch Kooperation

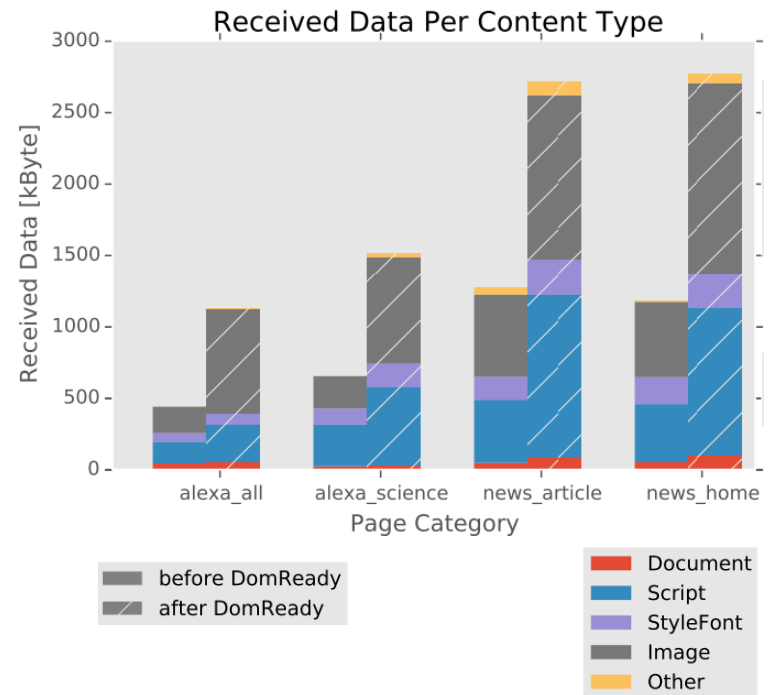


Metriken für Webseiten-Performance

- Load
- DOMContentLoaded
- First Meaningful Paint

Analyse des Aufbaus von Webseiten

- Document und Style
 - Voraussetzung für DOM
- Bilder und Skripte
 - größte Datenmenge, Anzahl Requests
 - Bilder: nach DOM-ready
 - Skripte: unterschiedlich
- „eilige“ Ressourcen:
 - Document, Style, Skripte im Seitenkopf

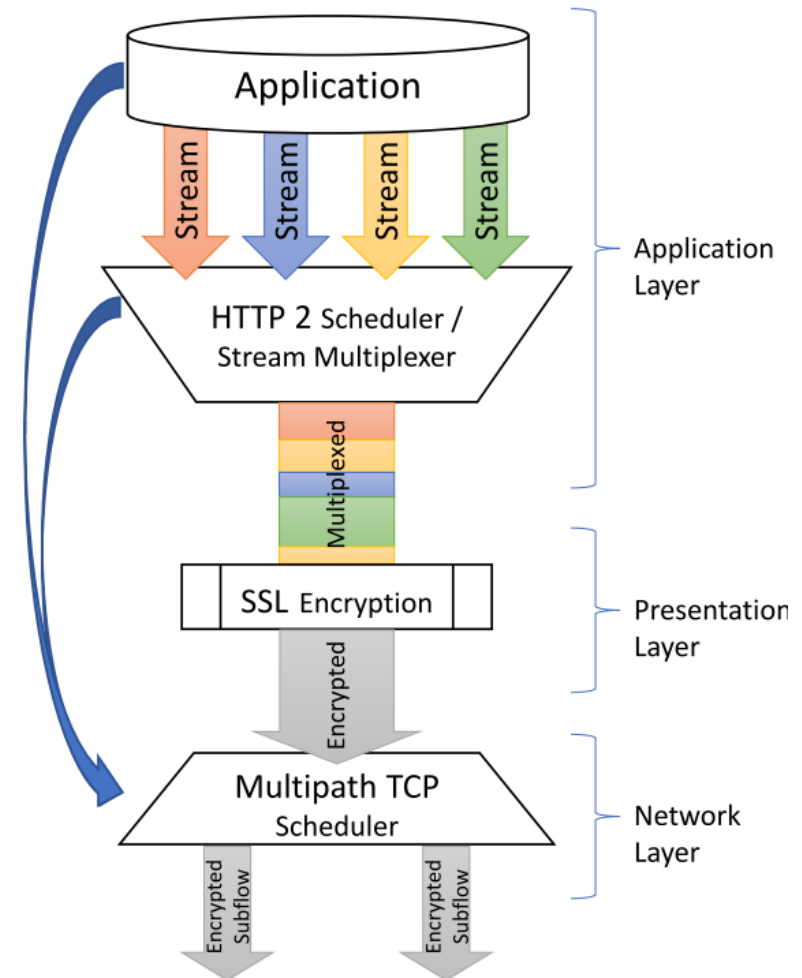


Optimierungen

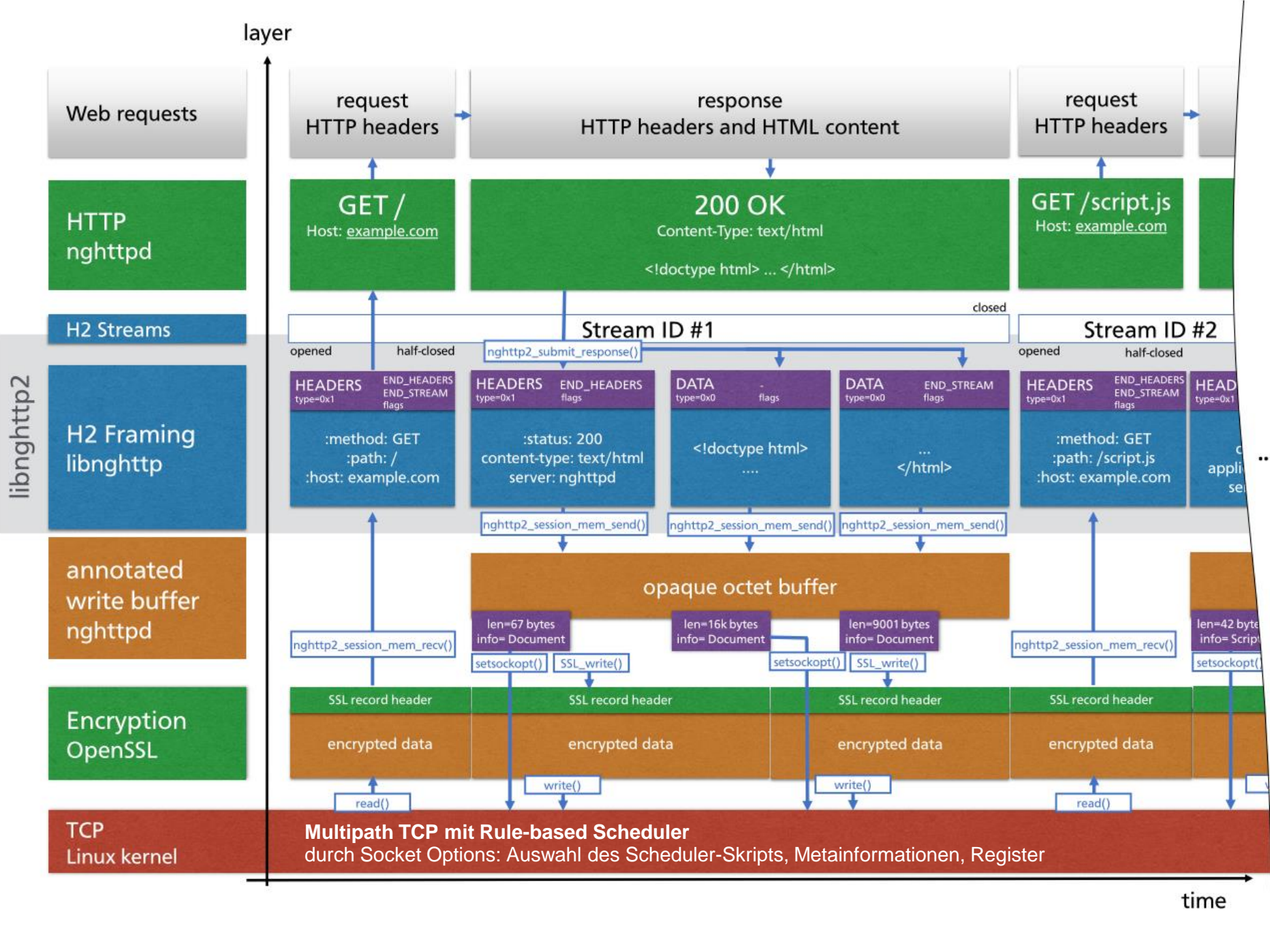
- Bei Verbindungen mit unterschiedlicher Latenz:
 - Nur eilige Ressourcen über schnellere, aber teurere Verbindung senden
- Bei Paketverlust / schwankender Latenz:
 - Latenzkritische Anfragen redundant über mehrere Verbindungen senden
- Wenn Übertragung fast fertig:
 - Letzte Segmente erneut senden

Kooperation

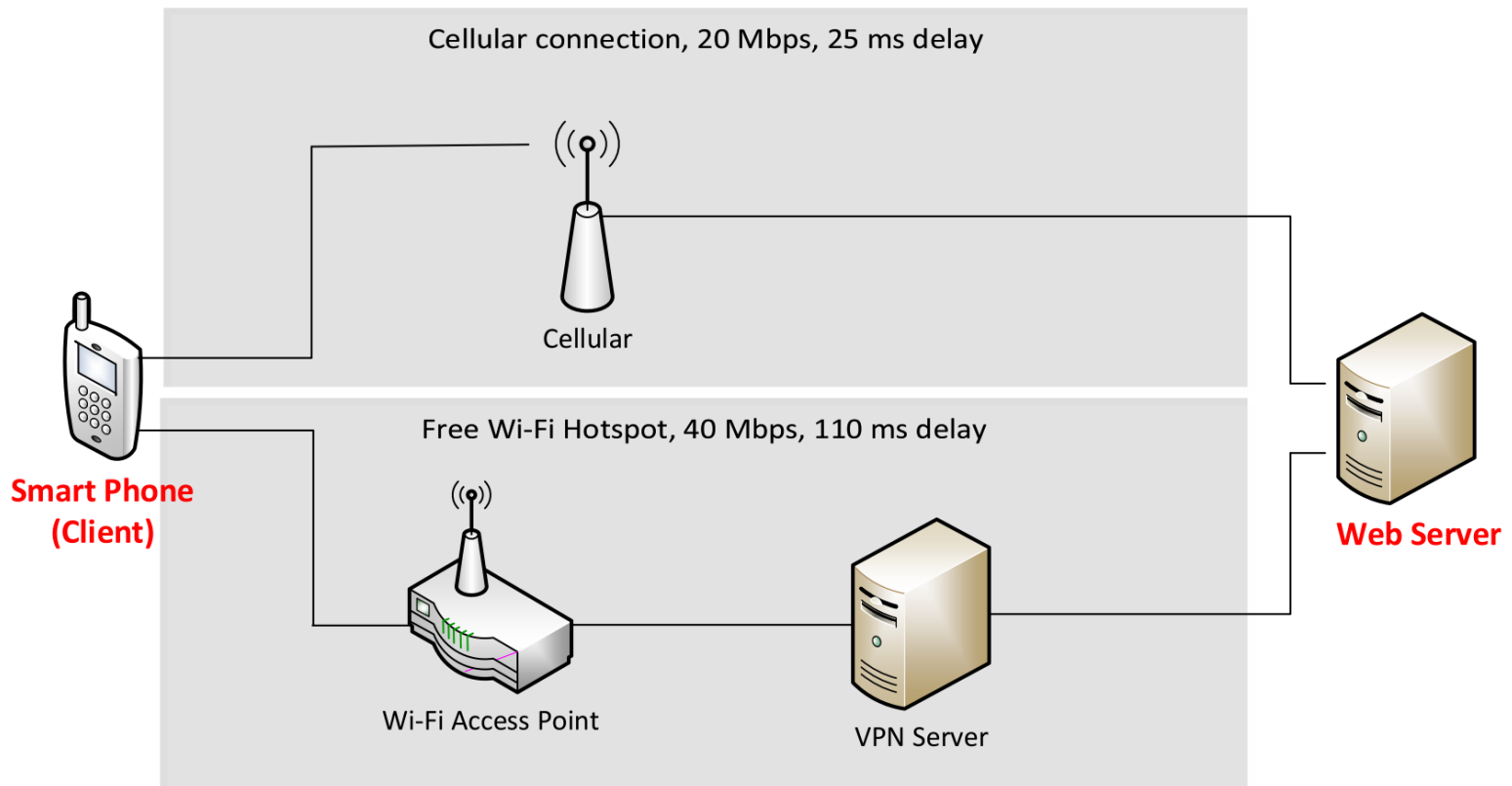
- Anwendung gibt beschreibende Hinweise
- Multipath TCP Scheduler bezieht diese in Entscheidung ein



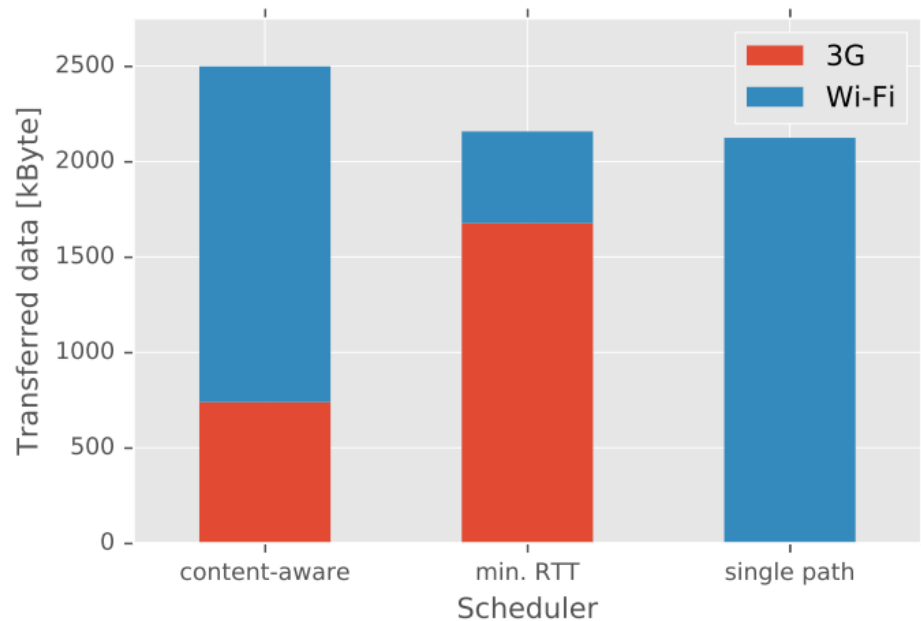
IMPLEMENTIERUNG



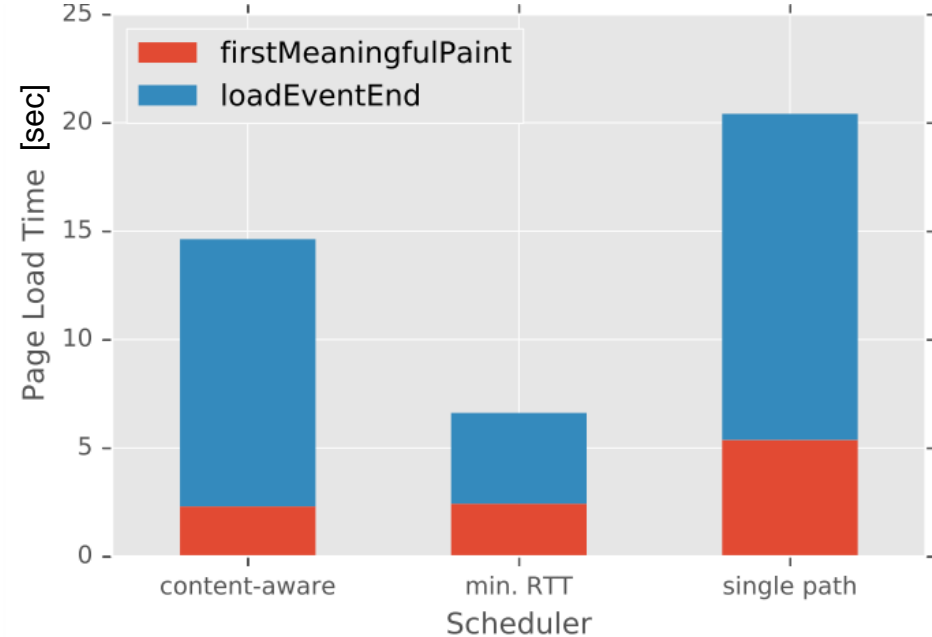
Evaluation



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 Priorisierung

Thank you for your attention! Questions?



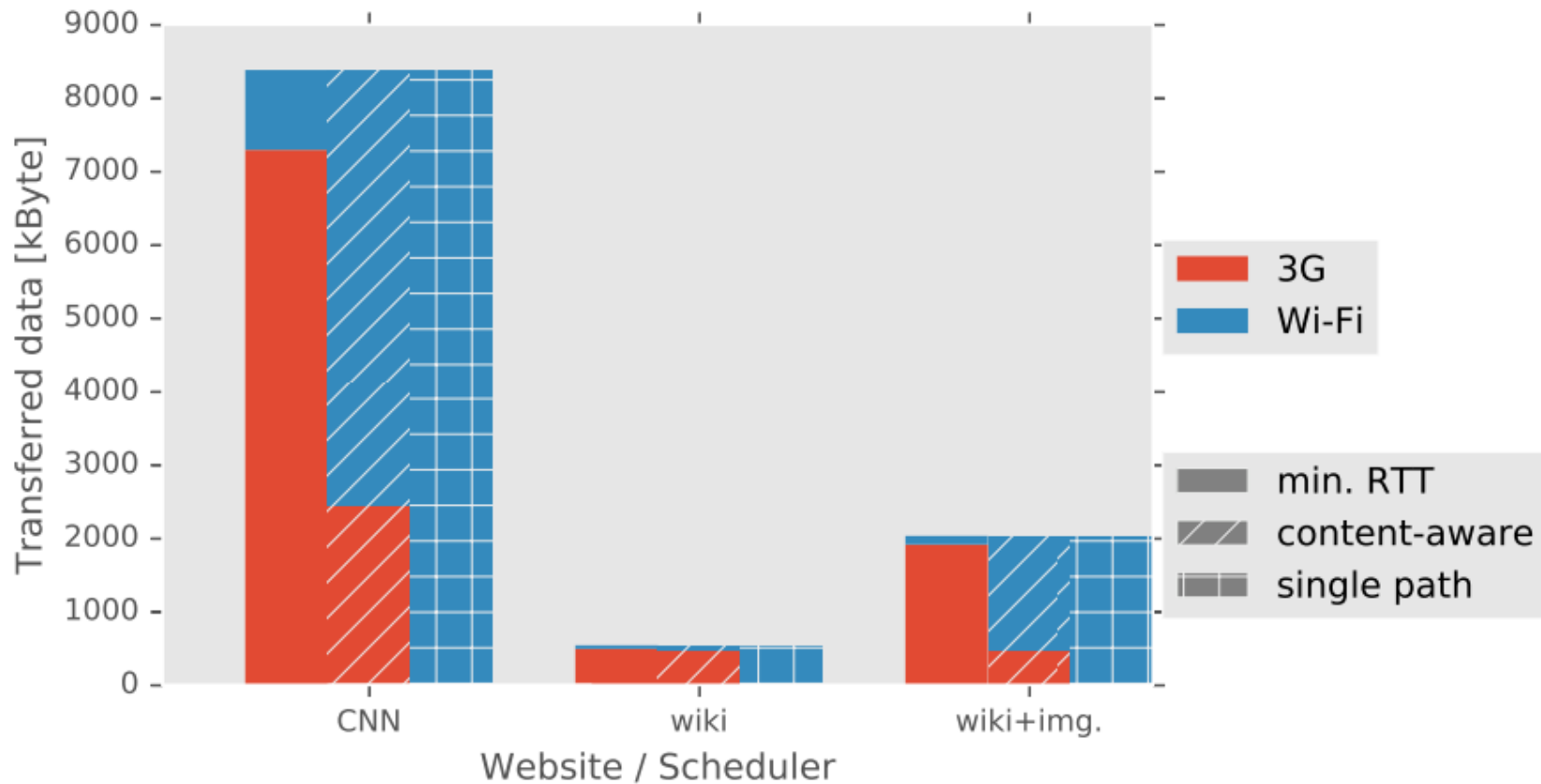


Figure 5.3.: Received bytes per interface for Evaluation A

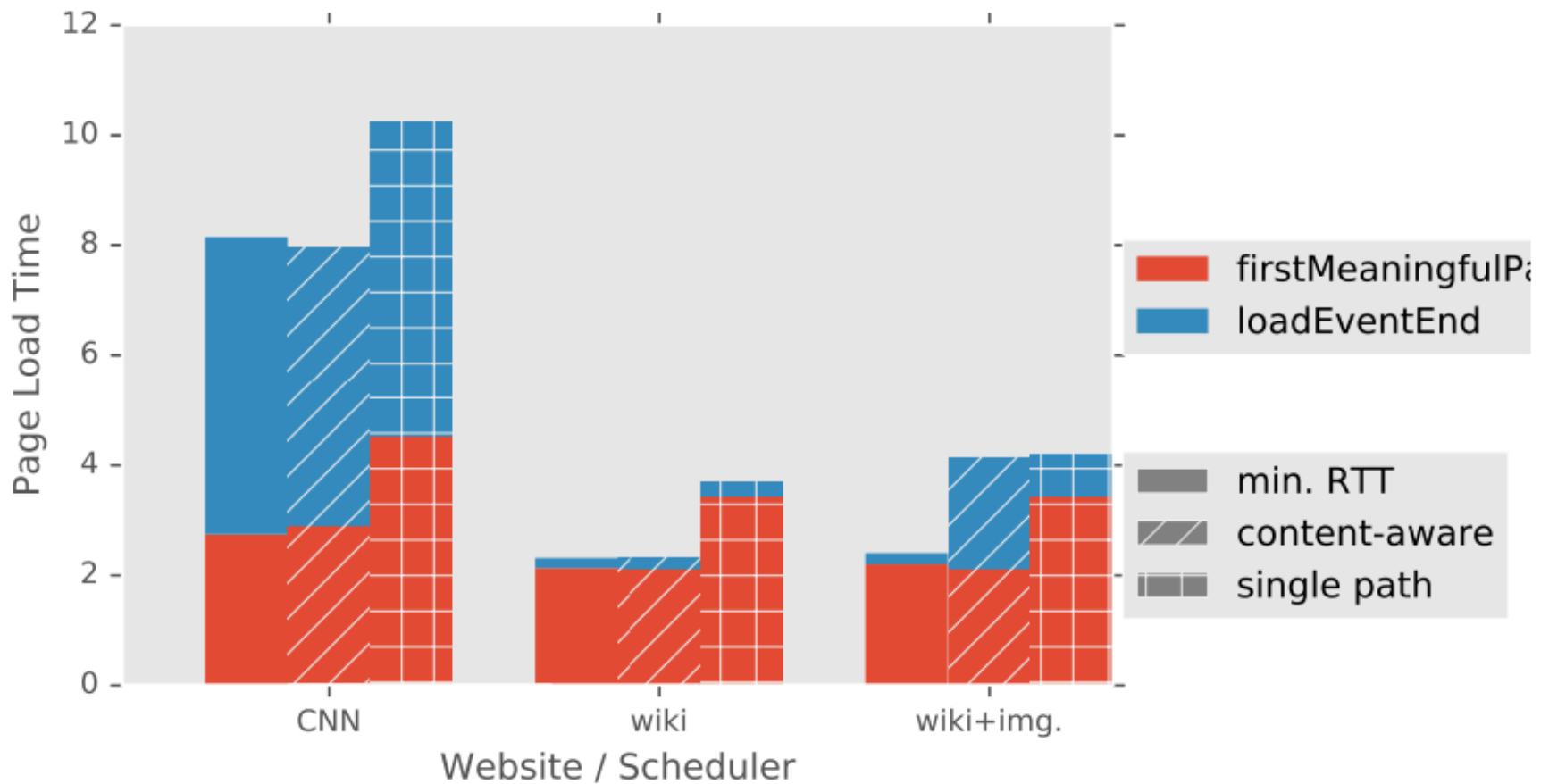


Figure 5.4.: Mean page load times for Evaluation A

Implementierung

Webserver als Grundlage

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

Multipath TCP

Anpassungen an Nghttpd

Implementierung

Multipath TCP

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

Webserver als Grundlage

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

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