

# Entwurf und Realisierung eines Adapters für UniLoG zur Lastgenerierung an IP-basierten Schnittstellen

Diplomarbeit

Martin Kulas

Universität Hamburg, Department Informatik  
Arbeitsgruppe Telekommunikation und Rechnernetze (TKRN)

Echtzeit 2010

Boppard am Rhein, 18. November 2010

# Übersicht

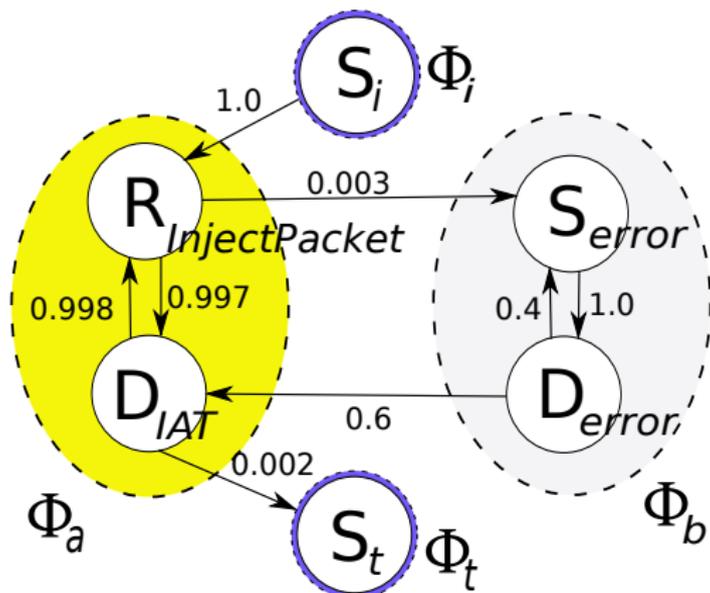
- 1 Motivation
- 2 Lastspezifikation mit Benutzerverhaltensautomaten
- 3 Der Real-Time Load Generator für IPv4-Netze
- 4 Präzision und Leistungsfähigkeit des IP-Adapters
- 5 Fallstudie: Untersuchung der Übertragungsqualität von Audio- und Videoströmen eines Films im WLAN unter verschiedenen künstlichen IP-Hintergrundlasten
- 6 Resümee

# Motivation

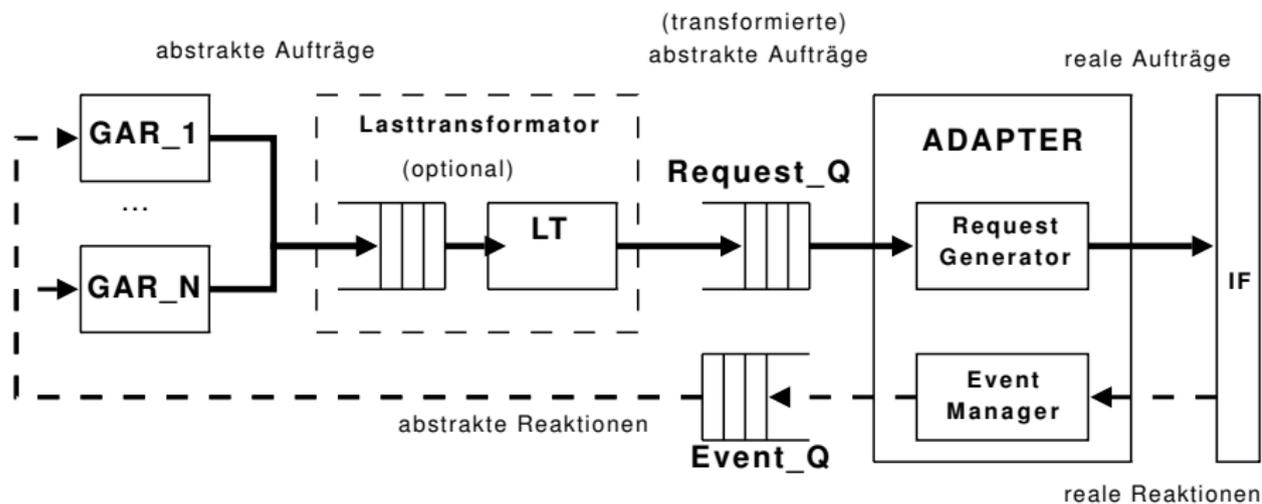
- Rechnernetze auf Basis des Protokolls IP sind allgegenwärtig.
  - Man erwartet, dass die Netze so leistungsstark sind, dass sie die Anforderungen der Anwendungsdienste erfüllen und eine zufriedenstellende Dienstgüte den Benutzern garantieren.
  - Heterogenität der Netze und Komplexität der Dienste erschweren Netzbetreibern und Dienstentwicklern, die Leistungsfähigkeit der Netze und die Benutzbarkeit der Dienste zu prognostizieren.
  - Das Spezialwerkzeug *Lastgenerator* erzeugt gewünschte Lastniveaus durch Nachahmung von Benutzerverhalten.
  - Experimentatoren verlangen eine präzise und leistungsstarke Lastgenerierung mit flexibler Lastspezifikation.
- Nur im Echtzeitbetrieb ist ein Lastgenerator sinnvoll.

# Lastspezifikation mit Benutzerverhaltensautomaten

- UniLoG (Unified Load Generator) ist ein Lastgeneratorkonzept mit Trennung von Lastspezifikation und Lastgenerierung.



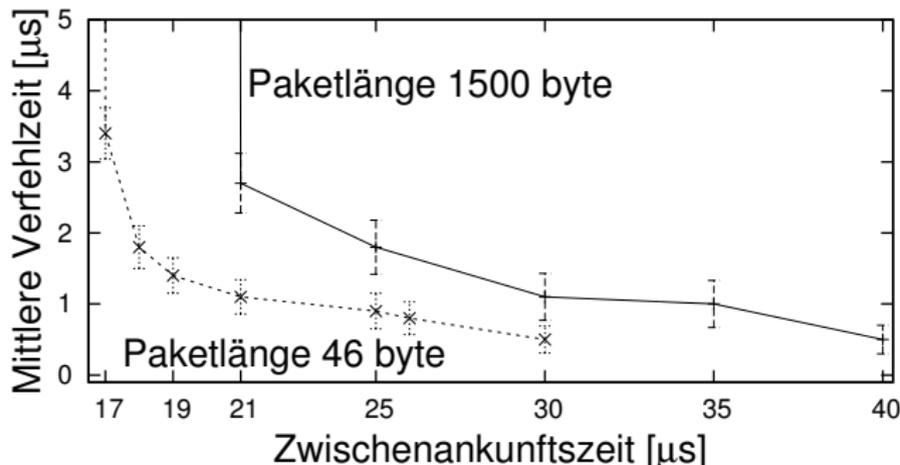
# Lastgenerierung mit Real-Time Load Generator



# Der Real-Time Load Generator für IPv4-Netze

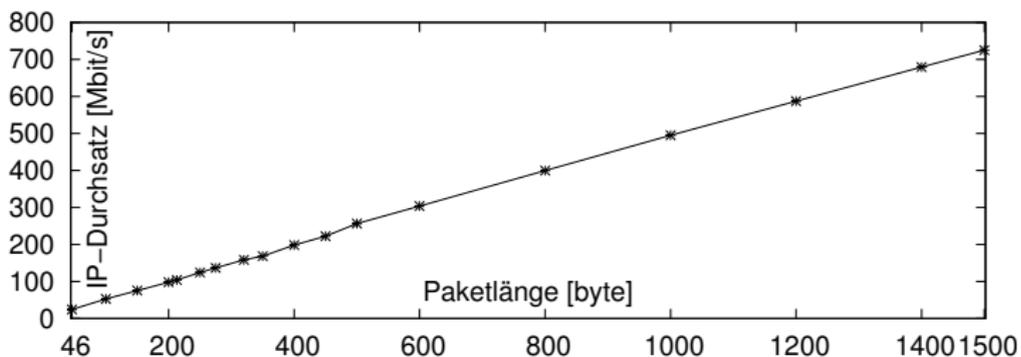
- Aufgaben des IP-Adapters mit Echtzeitanforderung:
  - 1 Konvertierung abstrakter Aufträge in reale Aufträge
  - 2 Übergabe realer Aufträge an der IP-Schnittstelle zu vorgegebenen Zeitpunkten
  - 3 Erfassung realer Reaktionen der IP-Schicht
  - 4 Umwandlung realer Reaktionen in abstrakte Reaktionen
- Abstrakter Auftrag ist `InjectPacket`, u. a. mit den Attributen Zieladresse und Paketlänge.
- Implementierung in der Programmiersprache C++ unter Verwendung der Bibliothek `libnet` zur Paketinjektion.

# Präzision



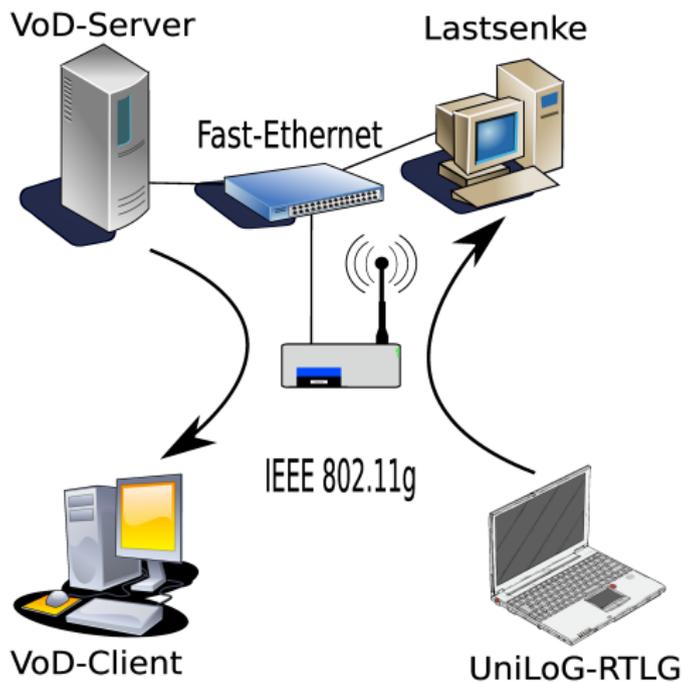
- Experimentierumgebung: Rechner mit Intel Core 2 Duo E6400 (2,13 GHz), dem Betriebssystem FreeBSD 7.2 und dem Netzadapter Broadcom BCM5754 (angeschlossen an ein 1000Base-T-Netz).

# Leistungsfähigkeit



- Die Experimentierumgebung blieb unverändert zur Präzisionsmessung.

# Fallstudie: Video-on-Demand über ein belastetes WLAN



# Subjektive Dienstgüteverschlechterung

- Das neunte ausgegebene Bild unter künstlichen VBR-Lasten unterschiedlicher Höhe mit 1500 byte langen IP-Paketen:



8 Mbit/s



16 Mbit/s

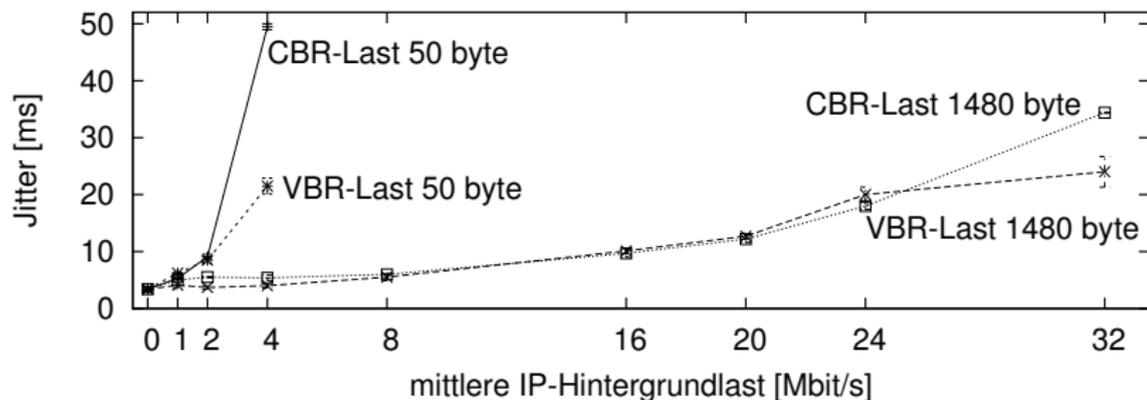


24 Mbit/s



32 Mbit/s

# Objektive Qualitätsverschlechterung am Beispiel Jitter



- Steigende Hintergrundlasten verschlechtern wie erwartet die Qualität der Filmwiedergabe.
- Hintergrundlasten mit kleinen Paketen treiben das WLAN bereits bei niedrigen Lastniveaus (4 Mbit/s) in den Hochlastbereich.

# Resümee

- Der IP-Lastgenerator erlaubt flexible Lastmodellierung und erfüllt Echtzeitanforderungen an die Lastgenerierung.
- Fallstudie hat den praktischen Nutzen des Lastgenerators gezeigt.
- Der IP-Lastgenerator ermöglicht detaillierte Leistungs- und Dienstgüteuntersuchungen in Rechnernetzen.
- Beispielsweise ist man mit IP-Lastgeneratoren in der Lage, verteilte Dienstverweigerungsangriffe (Distributed Denial of Service) durchzuführen und ihre Auswirkungen zu studieren.

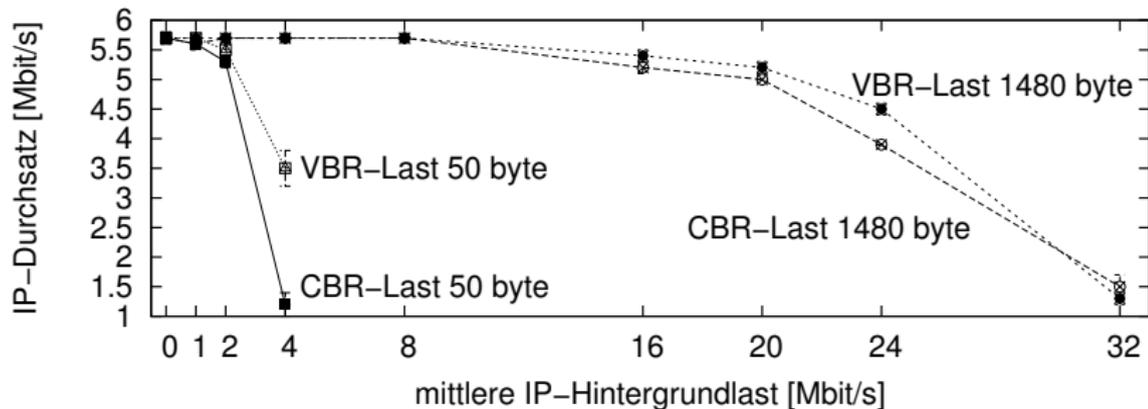
# Schluss

Vielen Dank für Ihre Aufmerksamkeit!

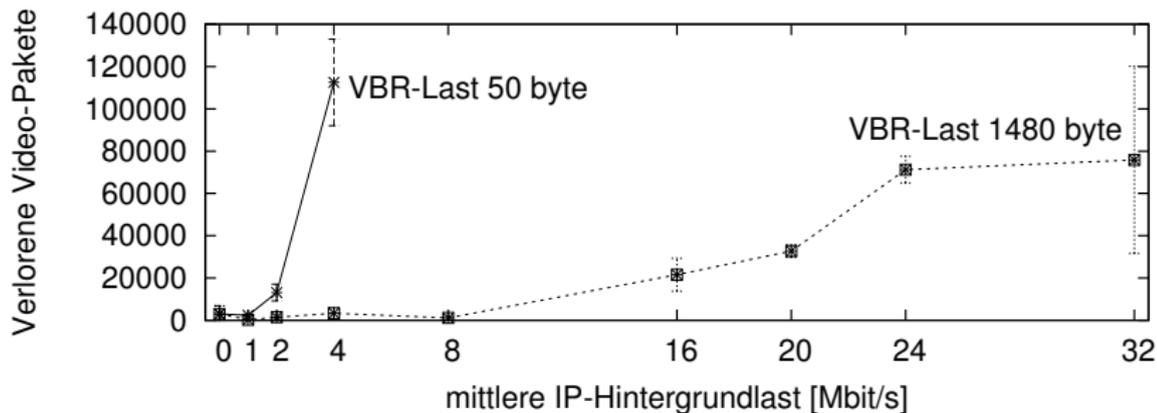
# Anhang

Zusätzliche Folien

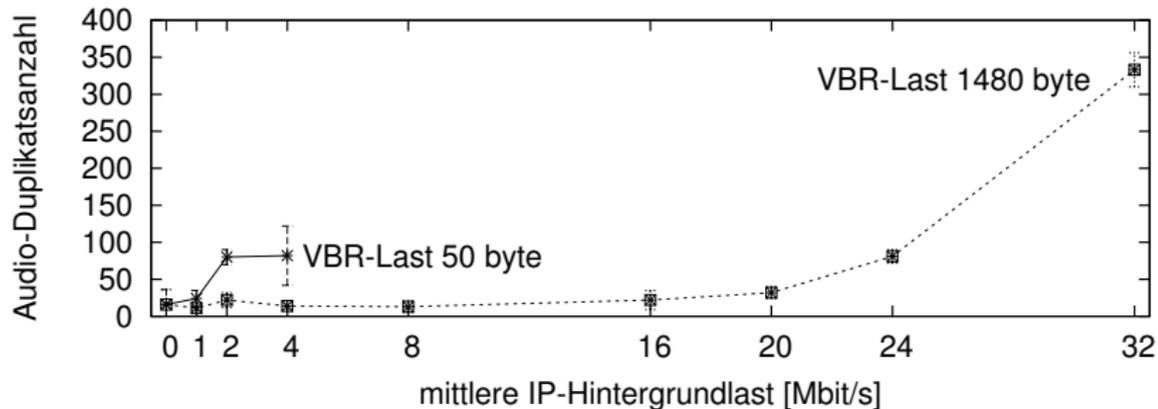
# Qualitätsmaß IP-Durchsatz beim VoD-Client



# Qualitätsmaß Paketverlust im Videostrom



# Qualitätsmaß Paketduplikate im Audiostrom



# Qualitätsmaß Sequenzfehler im Videostream

