

# Emulation von Rechnernetzen und Netzkopplungen

Christian Scherpe

Arbeitsgruppe Telekommunikation und Rechnernetze  
Universität Hamburg, Fachbereich Informatik  
Vogt-Kölln-Straße 30  
D-22527 Hamburg  
scherpe@informatik.uni-hamburg.de

**Abstract:** In diesem Paper werden Architektur und prototypische Implementierung eines Netzemulators (NetEmu) vorgestellt. Dabei wurde beim Design insbesondere Wert auf hohe Flexibilität und vielfältige Einsatzmöglichkeiten gelegt. Dazu gehören verschiedene Auswerteverfahren zur Ermittlung von Verzögerungen und Verlusten einzelner Dateneinheiten, die Berücksichtigung von Hintergrundlast und insbesondere auch die Möglichkeit, komplexe Szenarien wie gekoppelte Rechnernetze in Echtzeit zu behandeln, was beispielsweise durch Parallelisierung ermöglicht werden kann. Die praktische Einsetzbarkeit von NetEmu wurde im Rahmen von Fallstudien gezeigt.

## 1 Einleitung

In den letzten Jahren war ein starkes Wachstum im Bereich der Echtzeitkommunikation zu beobachten. Dies ist teilweise durch Anwendungen wie Videokonferenzen, Videotelephonie getrieben, andererseits sind die Netze selber immer leistungsfähiger geworden. Dies macht es bei der Diskussion von Dienstgüte (QoS) erforderlich, nicht nur die Leistungsfähigkeit einzelner Konfigurationen zu beobachten, sondern auch die Auswirkung auf Anwendungen. Die Leistungsfähigkeit alleine lässt sich gut durch klassische Netzsimulatoren beurteilen, für die Bewertung der Auswirkung kommt noch hinzu, dass der Simulator reale Last unter Echtzeitbedingungen verarbeiten muss. Zum einen ergeben sich damit quantitative Resultate, andererseits kann auch die Dienstgüte, mit der eine verteilte Anwendung durch den Einfluss des Kommunikationsnetzes arbeitet, qualitativ und damit subjektiv beurteilt werden. Diese Möglichkeit macht den Einsatz von Netzemulatoren notwendig und eröffnet ein eigenes Anwendungsspektrum.

## 2 Architektur des Netzemulators NetEmu

Bei der Ermittlung einer Architektur ist es insbesondere wichtig, zunächst die Anforderungen hinsichtlich Funktionalität und Leistungsfähigkeit zu definieren. Hauptanforderungen sind Einhaltung der Echtzeitbedingungen sowie hinreichende Präzision und Realitätsnähe des Emulators. Darüber hinaus gibt es noch eine Reihe weiterer Anforderungen wie die Möglich-

keit, verschiedene Auswerteverfahren (evtl. in Kombination) zu nutzen, Teilnetze auf einzelne Komponenten auslagern zu können und die Einspeisung zusätzlicher Hintergrundlast.

Diesen Anforderungen (detaillierter in [BüS00]) wird die Architektur aus Abb. 1 gerecht. Das Front-End nimmt externe Dateneinheiten einer verteilten Anwendung A sowie der Hintergrundlasten  $B_i$  entgegen, speichert zwischen, ist für Auslieferung bzw. Verluste verantwortlich und koordiniert die Prognosekomponenten, die ihrerseits Verzögerungen und Verluste ermitteln. Im Falle der Emulation gekoppelter Netze sind die Prognosekomponenten evtl. für jedes der beteiligten Transitnetze sukzessive aufzurufen. Hierfür stehen verschiedene Auswerteverfahren wie diskrete Simulation, analytische Modellierung oder (belastungsabhängig) die Verwendung von Traces zur Verfügung. Das „Mapping“ der Komponenten auf die Rechner eines LANs erfolgt dann so, dass die Echtzeitfähigkeit optimiert wird.

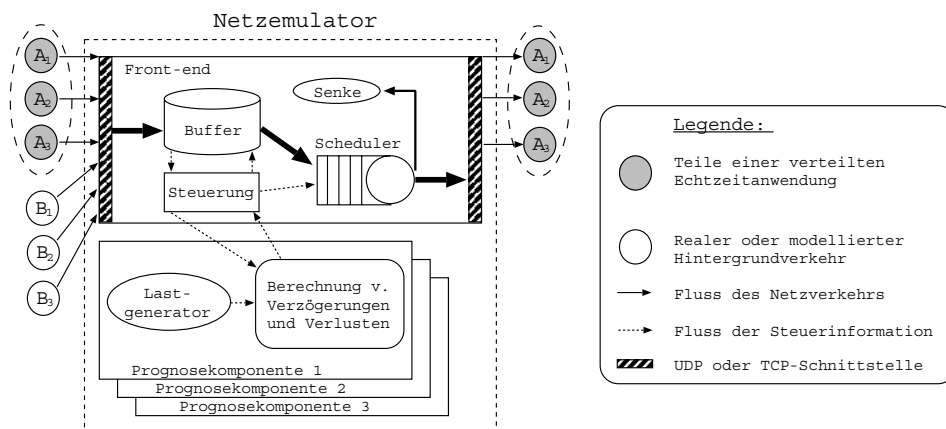


Abbildung 1: Gesamtarchitektur von NetEmu

### 3 Zusammenfassende Bewertung

Die vorgestellte Architektur wurde in Form eines Prototypen implementiert und praktisch eingesetzt. In [SWS03] ist eine umfangreiche Fallstudie präsentiert, in der der Einsatz anhand einer verteilten Videoanwendung demonstriert wurde. Darüber hinaus wird in [BüS00] die Leistungsfähigkeit von NetEmu pilotartig untersucht.

### Literaturverzeichnis

- [BüS00] Bühring F., Scherpe C., Modellierungsverfahren zur lastabhängigen Verhaltensprognose von Kommunikationsnetzen in Echtzeit, Proc. 14. Symp. Simulationstechnik, ASIM 2000, Hamburg, 2000.
- [SWS03] Scherpe C., Wolfinger B.E., Salzmann I., Model Based Network Emulation to Study the Behavior and Quality of Real-Time Applications, 7th IEEE Internat. Symp. on Distributed Simulation and Real Time Applications (DS-RT 2003), Okt. 2003, Delft.