

I. Baumgart, B. Heep und S. Krause

# OverSim: Ein skalierbares und flexibles Overlay-Framework für Simulation und reale Anwendungen

Gewinner des „KuVS Communication Software-Preis“ 2009



*Ingmar Baumgart* hat an der Universität Karlsruhe (TH) Informatik studiert. Seit 2005 ist er wissenschaftlicher Mitarbeiter am Institut für Telematik der Universität Karlsruhe (TH). Seine Forschungsschwerpunkte sind Overlay-Netze, dezentrale IP-Telefonie sowie Netzwerksicherheit. Im Rahmen des BMBF-Projekts ScaleNet hat er den dezentralen Namensdienst P2PNS entwickelt.



*Bernhard Heep* hat an der Universität Karlsruhe (TH) Informatik studiert. Seit 2005 ist er wissenschaftlicher Mitarbeiter am Institut für Telematik der Universität Karlsruhe (TH). Die Schwerpunkte seiner Forschung liegen im Bereich Topologieadaptation und Latenzminimierung in Overlay- und Peer-to-Peer-Netzen.



*Stephan Krause* erwarb den Grad des Dipl.-Inform. 2005 an der Universität Karlsruhe (TH). Seitdem ist er wissenschaftlicher Mitarbeiter am Institut für Telematik der Universität Karlsruhe (TH). Seine Forschungsschwerpunkte sind Overlay-Netze, Peer-to-Peer-Technologien und Verteilte Virtuelle Welten.

## I EINLEITUNG

Die Evaluierung neuartiger Protokolle ist ein grundlegendes Problem bei der Erforschung von Peer-to-Peer- und Overlay-Netzen. Idealerweise hat man dabei die Möglichkeiten das Protokoll sowohl in simulierten Szenarien mit einer hohen Anzahl von Netzwerkteilnehmern als auch in echten Netzwerkumgebungen, wie beispielsweise dem *PlanetLab*-Testbett, zu erproben. Um diese Anforderungen zu erfüllen und um weitere

Nachteile vorhandener Simulationsframeworks auszugleichen, wurde das Overlay-Simulationsframework *OverSim* [1] im Rahmen des BMBF-Projekts ScaleNet entwickelt.

Im Folgenden werden die besonderen Eigenschaften von *OverSim* kurz dargestellt. Ein Fokus liegt dabei auf der Wiederverwendbarkeit von Simulations-Code in echten Netzwerken. Dazu dienen zwei unterschiedliche P2P-Anwendungen als Demonstrator: *Dezentrales Voice-over-IP* und *virtuelle Welten*.

## II OVERLAY-FRAMEWORK OVERSIM

*OverSim* ist ein auf OMNeT++ basierendes Overlay-Framework für Linux, Windows und Mac OS X. Eine Vielzahl strukturierter und unstrukturierter Overlay-Protokolle wie Chord, Kademlia, Pastry, Bamboo, Koorde, Broose, Gia und Vast sind bereits integriert. Dazu kommen verschiedene darauf aufsetzende Anwendungen wie i3, Scribe, P2PNS und Sim-MUD.

Alle implementierten Protokolle können unverändert sowohl zur Simulation als auch in echten Netzwerken verwendet werden. So ist es beispielsweise möglich, eine auf strukturierten Overlays basierende Anwendung für verteilte Datenablage zunächst simulativ auf Chord und Pastry mit verschiedenen Underlay-Szenarien zu untersuchen um anschließend die selbe Implementierung im PlanetLab zu evaluieren.

*OverSim* hat eine vollständig modulare Architektur und ist dadurch äußerst flexibel bei der Entwicklung neuer Overlay bzw. P2P-Applikationen. Es können alle Komponenten eines P2P-Netzes vollständig abgebildet werden.

*OverSim* nutzt die grafischen Möglichkeiten des OMNeT++-Frameworks um sowohl Overlay- und Underlay-Topologie als auch sämtliche Netzwerkpakete in der GUI detailliert darzustellen (siehe Abb. 1), was u.a. intuitives Debugging ermöglicht. Darüber hinaus verfügt *OverSim* über ein zentrales Modul zur umfassenden Sammlung und Verarbeitung statistischer Daten.

Das Simulationsframework ist umfassend dokumentiert und wurde unter der *GPL* auf der Webseite <http://www.oversim.org/> veröffentlicht. Für das OMNeT++-Framework gilt die *Academic Public License*.

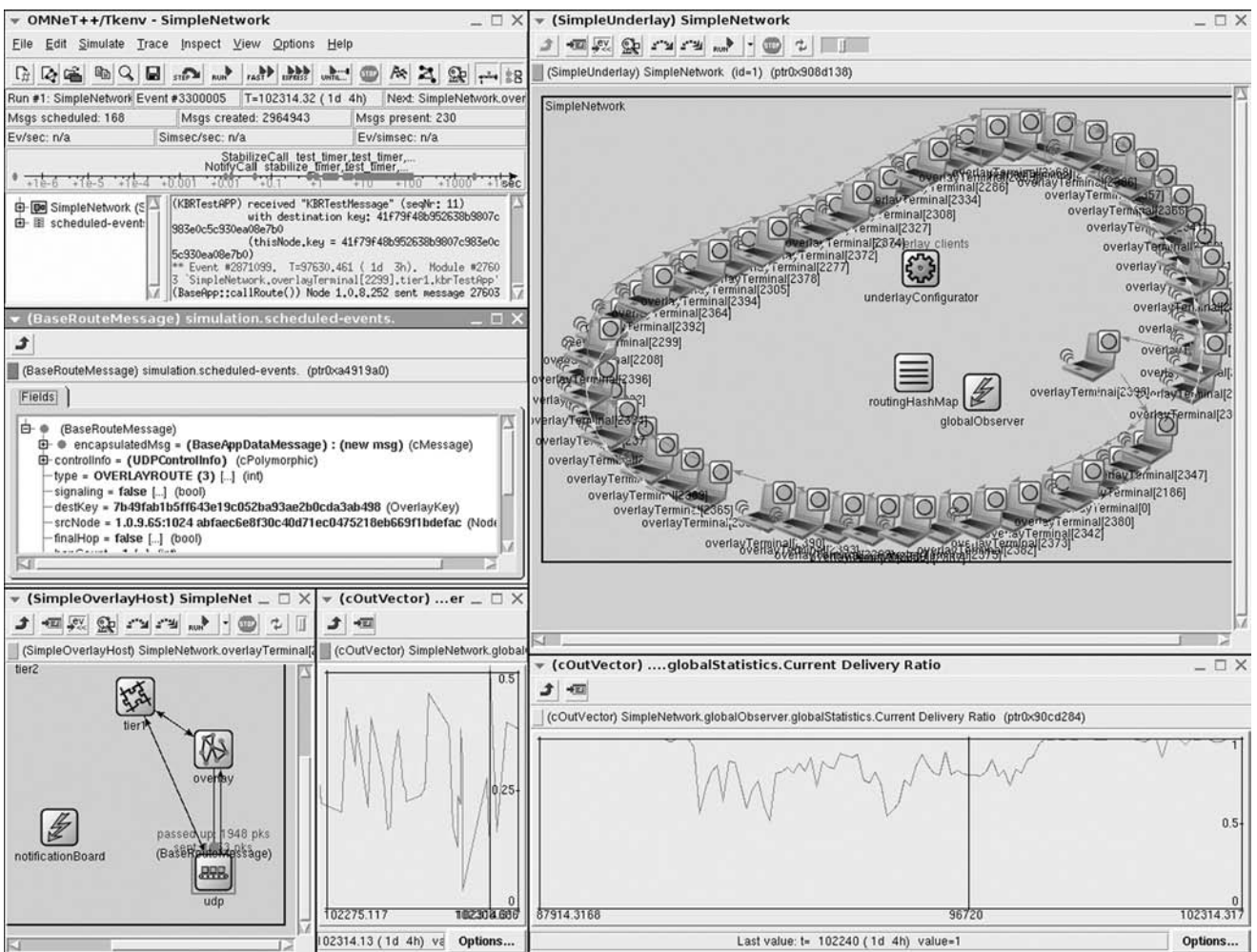


Abb. 1 Visualisierung der Chord-Topologie unter Churn

## A Underlay Abstraktion

OverSim bietet dem Anwender verschiedene Modelle zur Abstraktion des dem Overlay zugrundeliegenden Netzwerks. Dabei kann zwischen verschiedenen Komplexitätsstufen bzw. Underlay-Modulen gewählt werden, je nach dem wie exakt die unteren Netzwerkschichten modelliert werden sollen und wie groß das Overlay-Netz dimensioniert werden soll:

1) *INET-Underlay*: Um komplexe Simulationen heterogener Netzwerke mit vollständig modelliertem IP-Stack durchzuführen, kann ein auf dem INET-Framework aufbauendes Underlay-Modul verwendet werden. Neben verschiedenartigen Zugangsnetzen und -routern kann auch ein vollständiges Backbone simuliert werden. Sowohl Backbone- als auch Access-Router können Teilnehmer des simulierten Overlay-Netztes sein. Mobilität der Overlay-Knoten kann durch Wechsel des Zugangsnetzes modelliert werden.

2) *Simple-Underlay*: Mit dem Simple-Underlay verfügt OverSim über ein hochskalierbares Modell zur Simulation von bis zu 100.000 Overlay-Knoten. Paketverzögerungen werden wahlweise konstant oder mit Hilfe von synthetischen Koordinaten berechnet. Dazu verfügt OverSim über einen aus Internetmessungen gewonnenen Datensatz von Netzwerkkoordinaten, die zur Berechnung realistischer Delays verwendet werden können.

Zudem können unterschiedliche Zugangsnetzeigenschaften wie Bandbreite, Laufzeitverzögerung und Bitfehlerrate modelliert werden.

3) *SingleHost-Underlay*: Dieses spezielle Underlay-Modul verbindet die Simulation mit einem realen Netzwerk: Eine einzelne OverSim-Instanz emuliert in diesem Fall einen einzelnen Overlay-Knoten, der mit anderen emulierten Knoten über Netzwerke wie z.B. dem Internet kommunizieren kann. Eine für Demonstrationszwecke geeignete Variante emuliert auch mehrere Overlay-Knoten auf einem Host. Diese Knoten sind mit einem speziellen Router-Modul verbunden, das als Gateway zum echten Netzwerk dient.

Alle Underlay-Modelle bieten dem darüberliegenden Overlay eine einheitliche UDP/IP-Schnittstelle, so dass das Underlay-Modul —für das Overlay völlig transparent— beliebig ausgetauscht werden kann.

## B Overlay-Protokolle

Zur Unterstützung bei der Implementierung neuer Overlay-Protokolle und zu deren besserer Vergleichbarkeit bietet OverSim dem Entwickler verschiedene gemeinsame Funktionen und Schnittstellen von verbreiteten Overlay-Protokollen an. Dazu

zählen Bootstrapping, ein generischer Lookup-Mechanismus und eine *Remote Procedure Call* (RPC)-Schnittstelle, die auch von allen anderen Komponenten genutzt werden kann.

Die RPC-Schnittstelle bietet eine einheitliche Verarbeitung von Zeitüberschreitungen und übernimmt gegebenenfalls notwendige Sendewiederholungen. Dabei ist es möglich RPCs sowohl direkt als auch durch das Overlay geroutet, d.h. statt an eine *TransportAddress* (IP, UDP-Port) an einen Schlüssel zu senden.

Sowohl für Signalisierungs- und geroutete Nachrichten als auch für RPCs wird für alle Protokolle ein einheitliches Nachrichtenformat verwendet. Mehrere rekursive und iterative Routing-Varianten werden ebenfalls unterstützt.

Die Overlay-Schicht stellt für darauf aufbauende Anwendungen eine erweiterte *Common API* [2] zur Verfügung, was ein breites Spektrum an Anwendungen möglich macht. Zusätzliche Schnittstellen für unstrukturierte Overlays, virtuelle Welten und *Application Layer Multicast* (ALM)-Protokolle werden ebenfalls unterstützt.

Bei Verwendung der GUI ist OverSim in der Lage die Topologie der Overlay-Struktur zu visualisieren. Zur Unterstützung Abb. 2. Die modulare Architektur von OverSim von Topologieadaptation und Latenzminimierung in Overlay-Netzen stehen den Protokollen Schnittstellen zu physikalischen Abstandsmessungen und implementierte Koordinatensysteme (z.B. Vivaldi) zur Optimierung ihrer Routingtabellen zur Verfügung.

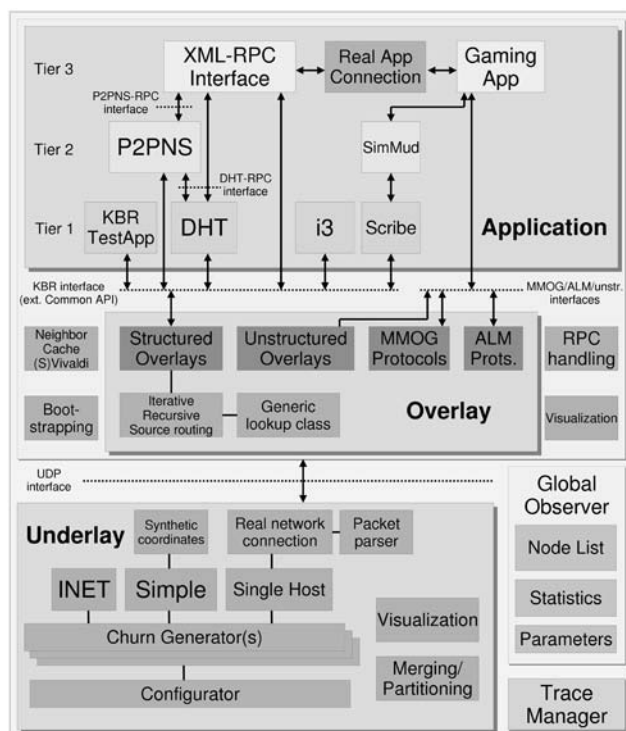


Abb. 2 Die modulare Architektur von OverSim

### C P2P-Applikationen und Nutzermodelle

Für Overlay-Applikationen in OverSim ist sowohl ein schichten- als auch ein komponentenbasierter Aufbau möglich (siehe Abb. 2). Neben einheitlichen Schnittstellen zum Underlay (UDP/IP und RPC) und der *Common API* zur KBR-Schicht ist

es über eine XML-RPC-Schnittstelle möglich externe —also von OverSim unabhängige —Anwendungen mit dem simulierten oder realen Overlay-Netz zu verbinden.

Der *Trace Manager* kann verwendet werden, um Applikationsmodulen ein zeitlich festgelegtes Verhalten vorzuschreiben, was die Modellierung komplexer Nutzermodelle ermöglicht.

### D Modellierung von Churn

OverSim bietet verschiedene Modelle zum Erzeugen von Knotenfluktuation (sog. Churn). Diese basieren entweder auf unterschiedlichen Wahrscheinlichkeitsverteilungen von Lebens- und Totzeit einzelner Knoten oder auf periodisch auftretenden Churn-Ereignissen.

Durch die Verwendung mehrerer *Churn-Generatoren* ist es möglich unterschiedlichen Gruppen von Overlay-Knoten verschiedene Churn- und Ausfallraten zuzuteilen.

## III DEMONSTRATOR

Im Rahmen einer Demonstration des OverSim-Frameworks können die folgenden Anwendungen gezeigt werden:

### A Simulation und Visualisierung von Overlay-Netzen

Um die Fähigkeiten von OverSim als Simulator zu präsentieren kann das Verhalten von Overlay-Protokollen unter Churn visualisiert werden.

Die sich durch Beitritt und Ausfall einzelner Knoten permanent verändernde Overlay-Topologie wird in der GUI dargestellt. Änderungen und Reparaturen des Overlays können so unmittelbar visuell nachvollzogen werden.

Auf allen Overlay-Knoten läuft eine Testapplikation, die periodisch Nachrichten an zufällig gewählte *NodeIDs* verschickt. Die Rate der erfolgreich ausgelieferten Nachrichten wird in Echtzeit in einem separaten Fenster grafisch dargestellt. Der Effekt einer Veränderung der Churn-Rate ist so deutlich zu beobachten.

### B Dezentrales Voice-over-IP (P2PSIP)

Eine zweite Anwendung auf Basis des OverSim-Frameworks stellt die Realisierung eines dezentralen Voice-over-IP-Systems dar (P2PSIP). Dezentrale VoIP-Netze können beispielsweise im Katastrophenfall eingesetzt werden um ohne Rückgriff auf zentrale Infrastrukturkomponenten eine schnell zu errichtende Kommunikationsplattform aufzubauen.

Unser P2PSIP-Ansatz verwendet den dezentralen Namensdienst P2PNS [3] um SIP-Namen zu registrieren oder aufzulösen. Der Verbindungsaufbau basiert dabei bis auf die dezentrale Namensauflösung auf dem standardisierten SIP-Protokoll. Dadurch können auf einfache Weise unmodifizierte SIP-Telefone an das P2PSIP-Netzwerk angeschlossen werden.

Der P2PSIP-Demonstrator besteht aus mehreren Nokia N800 Internet-Tablets mit 802.11g Schnittstellen, einem Linux-Laptop, einem unmodifizierten SIP-Telefon sowie einem Wählscheibentelefon.

Auf den Nokia N800 und dem Laptop laufen jeweils ein SIP-Client, ein SIP-Proxy auf Basis von *OpenSER* sowie eine OverSim-Instanz im *SingleHost-Modus*. OverSim enthält eine Implementierung des dezentralen Namensdienstes P2PNS und stellt dessen Funktionalität über eine XMLRPC-Schnittstelle dem SIP-Proxy zur Verfügung. Die einzelnen Overlay-Instanzen kommunizieren über die 802.11-Schnittstellen und bilden so eine Kademia-Overlaytopologie. Um die Overlaytopologie praxisnäher zu gestalten, werden durch eine weitere OverSim-Instanz zusätzliche Overlay-Knoten emuliert. Diese OverSim-Instanz bietet zudem eine GUI, um den emulierten Netzwerkverkehr und die Overlay-Topologie zu visualisieren.

Zusätzlich zu dem lokalen Demonstrator wurde die P2PSIP-Implementierung auf Basis von OverSim auch im weltweiten *PlanetLab*-Forschungsnetzwerk auf rund 450 Knoten installiert. Über einen Management- und Monitorknoten wird der Zustand des Netzes kontinuierlich überwacht und ausgefallene P2PNS-Instanzen durch neue Knoten ersetzt. Über einen Webserver im Monitorknoten lässt sich der Netzzustand in Google Maps und Google Earth visualisieren (siehe <http://www.p2p.name.org/>). Zusätzlich können über den Webserver die einzelnen OverSim-Instanzen über das XMLRPC-Interface ferngesteuert werden. Auf diese Weise können beispielsweise die Overlay-Routingtabellen der Knoten abfragen oder Daten in der DHT abgelegt werden. Schließlich können sich interessierte Nutzer mit einem SIP-Client am P2PSIP-Netz anmelden um dort echte Anrufe durchzuführen.

### C Virtuelle Welten

Eine weitere Anwendung des OverSim-Frameworks ist die Entwicklung und Evaluierung P2P-basierter virtueller Welten und MMOGs. Konventionelle virtuelle Welten verwenden eine Client/Server-Architektur. Diese zeigt sich jedoch bei steigenden Nutzerzahlen als nur bedingt skalierbar. Eine mögliche Alternative ist die Verwendung von P2P-Technologien, um die Skalierbarkeit zu erhöhen.

Zur Zeit sind vier verschiedene P2P-Protokolle für virtuelle Welten in OverSim implementiert: Das ALM-basierte Protokoll Sim-

MUD, das Supernode-basierte Protokoll PubSubMMOG sowie die Mutual-Notification-Protokolle Vast und QuON. Statistische Evaluierungen dieser Protokolle sind dank des modularen Aufbaus von OverSim einfach zu erstellen.

Bei virtuellen Welten sind jedoch nicht alleine statistische Kennzahlen von Bedeutung: Die Bewertung des Spielgefühls durch Nutzer ist als Maßzahl für die Akzeptanz eines Protokolls unverzichtbar. Hierfür werden auf mehreren Rechnern je eine Instanz von OverSim mit einer einfachen Spieleapplikation gestartet. Diese können von Spielern gesteuert werden, verschiedene Interaktionen zwischen den Spielern sind möglich. Um die Anzahl an Teilnehmern der virtuellen Welt zu erhöhen, werden zudem weitere Teilnehmer auf einem Rechner emuliert und mit den anderen Spielern verbunden.

Diese Anwendungen zeigen, dass sich OverSim sowohl in echten Netzwerken als auch auf ressourcenbeschränkten Geräten wie den Internet-Tablets eignet, um schnell und einfach neuartige Overlay-Anwendungen zu entwickeln.

### DANKSAGUNG

Diese Arbeit wurde vom Bundesministerium für Bildung und Forschung im Rahmen des *ScaleNet*-Projekts 01BU567 gefördert.

### REFERENCES

- [1] Baumgart, I.; Heep, B.; and Krause, S.: OverSim: A flexible overlay network simulation framework. In: Proceedings of 10th IEEE Global Internet Symposium (GI '07) in conjunction with IEEE INFOCOM 2007, Anchorage, AK, USA, May 2007, pp. 79-84.
- [2] Dabek, F.; Zhao, B.; Druschel, P.; Kubiatowicz, J.; and Stoica, I.: Towards a common api for structured peer-to-peer overlays. In: Proceedings of the 2nd International Workshop on Peer-to-Peer Systems (IPTPS '03), vol. Volume 2735/2003, 2003, pp. 33-44.
- [3] Baumgart, I.: P2PNS: A Secure Distributed Name Service for P2PSIP. In: Proceedings of the Sixth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom 2008), Hong Kong, China, Mar. 2008.