

# Pull-Multicast for the HAMcast API

## Vortrag Anwendung 2

Sebastian Zagaria

Hochschule für Angewandte Wissenschaften Hamburg

01.12.2011

# Agenda

- 1 **Einleitung**
  - Gruppen Kommunikation
  - Daten Verteilung im Overlay
  - Motivation
- 2 **Related Work**
  - HAMcast [1] [2]
  - Design and Implementation of a Generic Library for P2P Streaming [3]
  - BiToS: Enhancing BitTorrent for Supporting Streaming Applications [4]
- 3 **Ausblick**

# Outline

- 1 **Einleitung**
  - Gruppen Kommunikation
  - Daten Verteilung im Overlay
  - Motivation
- 2 **Related Work**
  - HAMcast [1] [2]
  - Design and Implementation of a Generic Library for P2P Streaming [3]
  - BiToS: Enhancing BitTorrent for Supporting Streaming Applications [4]
- 3 **Ausblick**

# Problemstellung

## ● Problem

- Viele Benutzer haben Interesse an einer Datei
- Daten Verteilung über Unicast ist sehr ineffizient
- Sender muss Daten für jeden Benutzer replizieren und versenden
- Sender und Empfänger müssen sich kennen

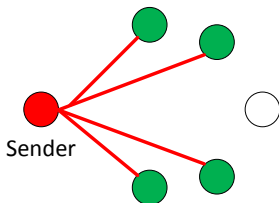
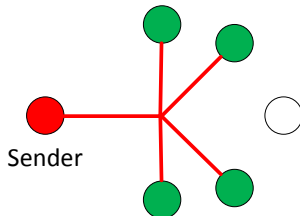


Abbildung: Nicht effektive Gruppenkommunikation

# Publish/Subscribe

## • Lösung

- Publish/Subscribe (Gruppenkommunikation)
- Publisher(Sender) Verteilt Daten an eine Gruppe von Subscriber (Empfänger)
- Publisher und Subscriber müssen sich nicht kennen
- Daten werden im Netzwerk an geeigneten Stellen repliziert



# Outline

- 1 **Einleitung**
  - Gruppen Kommunikation
  - **Daten Verteilung im Overlay**
  - Motivation
- 2 **Related Work**
  - HAMcast [1] [2]
  - Design and Implementation of a Generic Library for P2P Streaming [3]
  - BiToS: Enhancing BitTorrent for Supporting Streaming Applications [4]
- 3 **Ausblick**

# Verteilung der Daten

- Push-Multicast
  - Verteilung der Daten über eine Baum Struktur
  - Unidirektional (Sequenziell)
  - Nicht geeignet für das verteilen von Dateien
- Mesh-Pull
  - Verteilung der Daten über ein Mesh-Netzwerk (Unstrukturiert)
  - Bidirektional (Chunk-Trading)
  - Das Downloaden von Dateien kann fortgesetzt werden
  - Time-Shifting bei Video-Streams

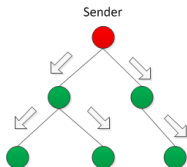


Abbildung: Multicast-Push

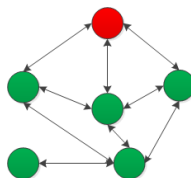


Abbildung: Mesh-Pull



# Outline

## 1 Einleitung

- Gruppen Kommunikation
- Daten Verteilung im Overlay
- **Motivation**

## 2 Related Work

- HAMcast [1] [2]
- Design and Implementation of a Generic Library for P2P Streaming [3]
- BiToS: Enhancing BitTorrent for Supporting Streaming Applications [4]

## 3 Ausblick



# Motivation

## • Ziele

- Entwicklung und Implementierung eines Pull-Multicast für HAMcast
- Die Betrachtung von Chunk-Trading Overlays aus einer neuen Perspektive
- Komplexität aus der Anwendung nehmen

## • Herausforderungen

- Gruppenverwaltung
- Kozeptionierung für eine Generelle Gruppenkommunikations-API

# Chunk-Trading Overlay [5]

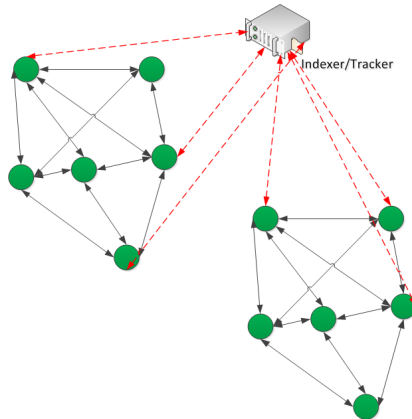


Abbildung: Chunk Trading Overlay

# Outline

- 1 Einleitung
  - Gruppen Kommunikation
  - Daten Verteilung im Overlay
  - Motivation
- 2 Related Work
  - **HAMcast [1] [2]**
  - Design and Implementation of a Generic Library for P2P Streaming [3]
  - BiToS: Enhancing BitTorrent for Supporting Streaming Applications [4]
- 3 Ausblick



# HAMcast

## • Ziele

- Komplexität aus der Anwendung zu Nehmen
- Generelle Multicast-API
- Unterstützung und Verbindung verschiedenster Multicast-Technologien

# Aufbau von HAMcast

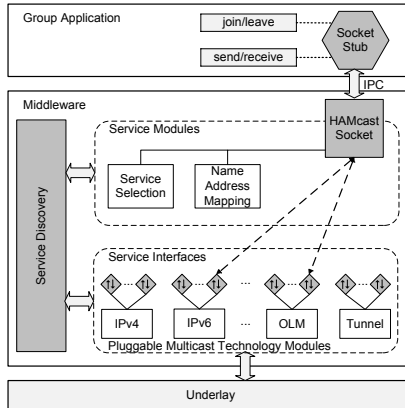


Abbildung: Aufbau der HAMcast Middleware



# Hybrider Multicast mit HAMcast

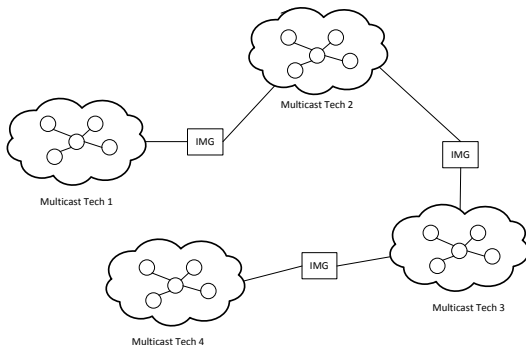


Abbildung: Aufbau der HAMcast Middleware

# HAMcast URI

- *scheme* *://* *group* *@* *instantiation* *:* *port* */* *sec-credentials*
  - „scheme“ bezeichnet den Namensraum, z.B. IP, Scribe
  - „group“ gibt die Gruppen ID an
  - „instantiation“ identifiziert die Entität, die eine Instanz der Gruppe generiert
  - „sec-credentials“ ist optional
- **Beispiel:**
  - Beispiel URI für das Chunk-Trading  
*scheduling://Content@BsNode/1..15*

# Outline

- 1 Einleitung
  - Gruppen Kommunikation
  - Daten Verteilung im Overlay
  - Motivation
- 2 **Related Work**
  - HAMcast [1] [2]
  - **Design and Implementation of a Generic Library for P2P Streaming [3]**
  - BiToS: Enhancing BitTorrent for Supporting Streaming Applications [4]
- 3 **Ausblick**



# GRAPES

- GRAPES (Generic Resource Aware P2P Environment for Streaming)
- Generische API die es ermöglichen soll P2P-Streaming Anwendungen zu Entwickeln
- Autoren
  - Luca Abeni u.w.
  - Universität Trento
- Ermöglicht es ein Overlay aus vier verschiedenen Modulen zu Erstellen
  - **Peer Sampling**: Wird verwendet um einem Overlay beizutreten
  - **Signalling and Chunk Trading** : Ermöglicht Kommunikation zwischen den Peers und das Austauschen von Chunks
  - **Chunk Buffer** : Zum Speichern und verwalten der Chunks
  - **Scheduling** : Implementierung eines Chunk-Scheduling Algorithmus

# Peer Sampling

- Funktionen des Peer Sampling Modules
  - **ParseData():** Message Paser für Sampling-Messages
  - **AddNeighbor():** Zum Setzen eines oder mehrerer BootStrap-Nodes
  - **RemoveNeighbor():** Entfernt einen Peer aus der lokalen Liste
  - **GetNeighborhood():** Gibt eine zufällige Liste von Peers zurück
  - **Grow/ShrinkNeighborhood():** Vergrößert oder verringert die Liste der Peers
  - **Get/ChageMetadata():** Zum Setzen oder finden von Meta-Informationen der Peers



# Signalling and Chunk Trading

- Funktionen des Signalling and Chunk-Trading Module
  - **buffermap message():** Für den Austausch von Chunk Informationen
  - **chunk offer():** Kann genutzt werden um einen Peer Chunks anzubieten
  - **chunk accept():** Als Antwort auf ein chunk offer()
  - **chunk request():** Um Chunks von anderen Peers anzufragen
  - **chunk deliver():** Um einen oder mehrere Chunks zu versenden
  - custom Chunk-Trading über eine low-level API



# Chunk Buffer

- Funktionen des Chunk Buffer Module
  - Zwei verschiedene Management-Policies
    - time sensitive
    - size sensitive
    - Custom-Policies
  - `add chunk()`, `get chunk/s`, `clear()`, `cb distroy()`, `cb init()`

# Bewertung von GRAPES

- Gut geeignet für die Forschung
- Testen und analysieren von Mesh-Overlay Algorithmen
- Nicht gut genug Abstrahiert
  - Für die Entwicklung von P2P - Anwendungen
  - Technologie abhängig

# Outline

- 1 Einleitung
  - Gruppen Kommunikation
  - Daten Verteilung im Overlay
  - Motivation
- 2 **Related Work**
  - HAMcast [1] [2]
  - Design and Implementation of a Generic Library for P2P Streaming [3]
  - **BiToS: Enhancing BitTorrent for Supporting Streaming Applications [4]**
- 3 **Ausblick**

# Autoren und Ziel

- **Autoren**

- Aggelos Valvianos
- Marios Iliofotou
- University of California Riverside

- **Ziel**

- Erweiterung des BitTorrent-Protokolls für Video Streaming

# Komponenten von BiToS

- Drei Hauptkomponenten
  - Received Pieces
    - Enthält alle heruntergeladenen Chunks
    - *Chunk-States* : Downloaded, Not Downloaded, Missed
  - High Priority Set
    - Beinhaltet Chunks die noch nicht heruntergeladen wurden
    - Chunks haben eine höhere Download Priorität
    - *Chunk-States* : Not-Requested oder Currently-Downloading
  - Remaining Pieces Set
    - Chunks die noch nicht heruntergeladen wurden
    - *Chunk-States* : Not-Requested oder Currently-Downloading





# Chunk-Scheduling

- Auswahl der Chunks
  - Auswahl zwischen High Priority- und Remaining Pieces Set
  - Wahrscheinlichkeit 1-P
  - Auswahl über Rarest First
  - deadline Funktion
    - Entfernen von Chunks durch Zeitüberschreitung

# Bewertung von BiToS

- Einfach zu implementierender Algorithmus
- Ermöglicht Video-Streaming mit BitTorrent
- Leichte anpassen des Zeitverhaltens durch die Veränderung von  $P$

- **Fazit**

- Mesh-Overlays und Scheduling-Algorithmen zur Verteilung der Daten sind bereits ausführlich erforscht wurden
- Schwerpunkt der Entwicklung liegt in der Gruppenverwaltung und der API

- **Projekt 2**

- Implementierung eines Push-Multicast als Referenz Technologie (BIDIR-SAM)

-  S. Meiling, D. Charousset, T. C. Schmidt, and M. Wählisch, “System-assisted Service Evolution for a Future Internet – The HAMcast Approach to Pervasive Multicast,” in *Proc. of IEEE GLOBECOM 2010, Workshop MCS 2010*. Piscataway, NJ, USA: IEEE Press, December 2010.
-  M. Wählisch, T. C. Schmidt, and S. Venaas, “A Common API for Transparent Hybrid Multicast,” individual, IRTF Internet Draft – work in progress 04, July 2010. [Online]. Available: <http://tools.ietf.org/html/draft-waehlich-sam-common-api>
-  L. Abeni, C. Kiraly, A. Russo, M. Biazzini, and R. Lo Cigno, *Design and implementation of a generic library for P2P streaming*, ser. AVSTP2P '10. New York, NY, USA: ACM, 2010. [Online]. Available: <http://doi.acm.org/10.1145/1877891.1877902>
-  A. Vlavianos, M. Iliofotou, and M. Faloutsos, “Bitos: Enhancing bittorrent for supporting streaming applications,” in *INFOCOM*

2006. *25th IEEE International Conference on Computer Communications. Proceedings*, april 2006, pp. 1 –6.



X. Hei, Y. Liu, and K. Ross, “Iptv over p2p streaming networks: the mesh-pull approach,” *Communications Magazine, IEEE*, vol. 46, no. 2, pp. 86 –92, february 2008.