

Pull-Multicast

Vortrag Master-Seminar

Sebastian Zagaria

Hochschule für Angewandte Wissenschaften Hamburg

07.06.2012

Agenda

- 1 **Einleitung**
 - Motivation
 - Push und Pull
 - HAMcast
- 2 **Masterarbeit**
 - Gruppenverwaltung
 - API
- 3 **Ausblick**

Outline

- 1 **Einleitung**
 - Motivation
 - Push und Pull
 - HAMcast
- 2 Masterarbeit
 - Gruppenverwaltung
 - API
- 3 Ausblick

Motivation

Ziele

- Entwicklung eines Pull-Multicast
- Implementierung eines Prototypen für HAMcast
- Publish/Subscribe Funktionalität in einem Pull Netzwerk

Datenverteilung in P2P Netzwerken (1)

Push-Multicast

- Verteilung der Daten über eine Baumstruktur
- Unidirektional (sequenziell)
- Nicht geeignet für das Verteilen von Dateien
- Hoher Managementaufwand beim erstellen vom Overlay und Baum
- Findet hauptsächlich Anwendung in der Forschung



Push-Multicast Beispiel

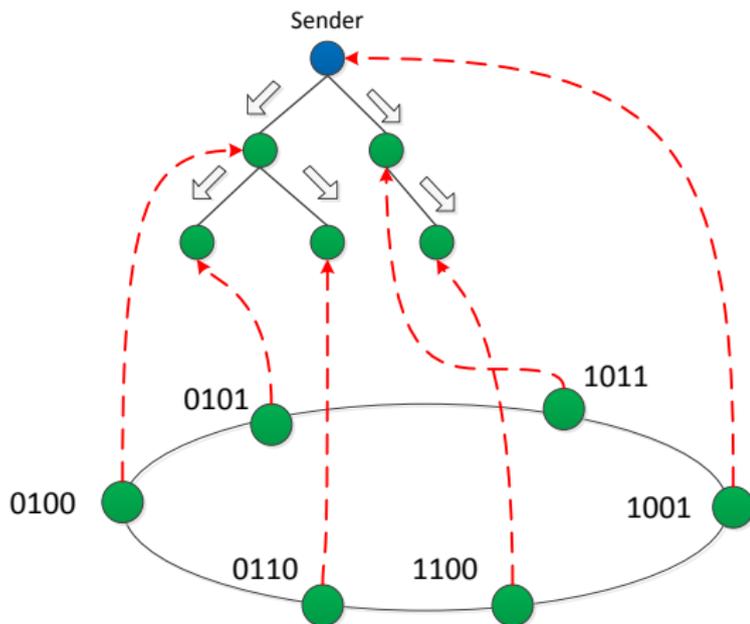


Abbildung: Push-Multicast



Daten Verteilung in P2P Netzwerken (2)

Mesh-Pull

- Verteilung der Daten über ein Mesh-Netzwerk (unstrukturiert)
- Bidirektional (Chunk-Trading)
- Das Downloaden von Dateien kann fortgesetzt werden
- Video-Streaming und Time-Shifting
- Nicht geeignet für Live-Streams



Mesh-Pull Beispiel

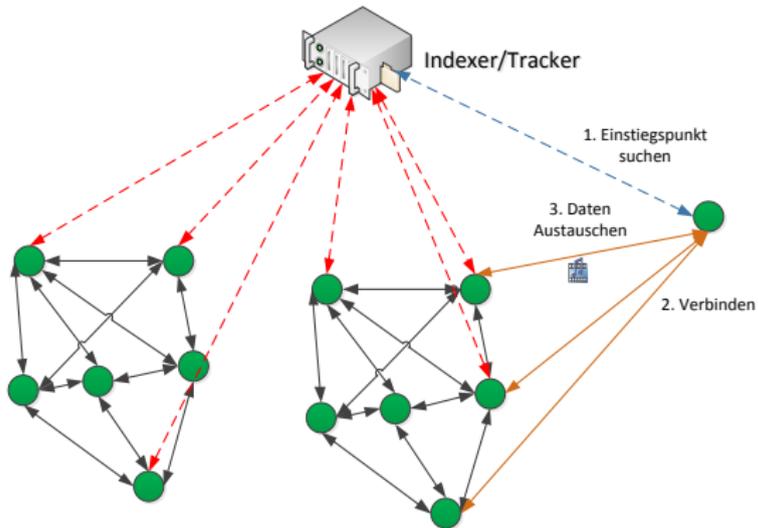


Abbildung: Mesh-Pull

Outline

- 1 **Einleitung**
 - Motivation
 - Push und Pull
 - HAMcast
- 2 **Masterarbeit**
 - Gruppenverwaltung
 - API
- 3 **Ausblick**

HAMcast

Ziele

- Komplexität aus der Anwendung zu Nehmen
- Implementierung der Common-API (Generelle Multicast-API)
- Unterstützung verschiedenster Multicast-Technologien

Aufbau von HAMcast

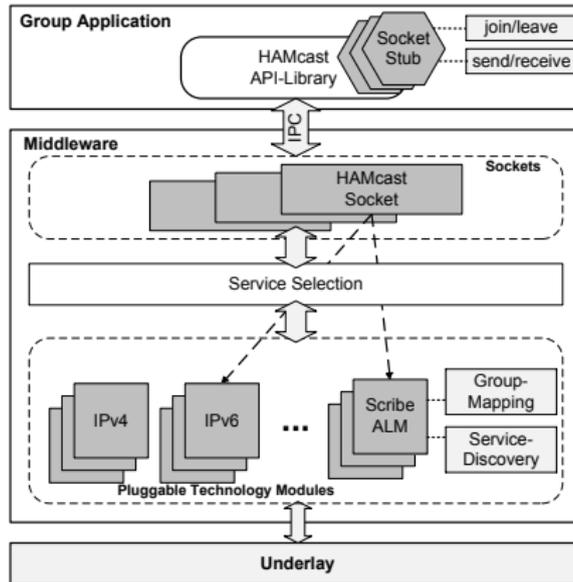


Abbildung: Aufbau von HAMcast



Outline

- 1 Einleitung
 - Motivation
 - Push und Pull
 - HAMcast
- 2 Masterarbeit
 - Gruppenverwaltung
 - API
- 3 Ausblick

Gruppenverwaltung

Fragestellungen

- Was muss gespeichert werden ?
 - Gruppenadresse, Peer und Datei Information
- Wie wird es gespeichert ?
 - Server-Based oder Dezentralisiert

Gruppenverwaltung

Fragestellungen

- Was muss gespeichert werden ?
 - Gruppenadresse, Peer und Datei Information
- Wie wird es gespeichert ?
 - Server-Based oder Dezentralisiert

Gruppenverwaltung

Fragestellungen

- Was muss gespeichert werden ?
 - Gruppenadresse, Peer und Datei Information
- Wie wird es gespeichert ?
 - Server-Based oder Dezentralisiert

Gruppenverwaltung

Fragestellungen

- Was muss gespeichert werden ?
 - Gruppenadresse, Peer und Datei Information
- Wie wird es gespeichert ?
 - Server-Based oder Dezentralisiert

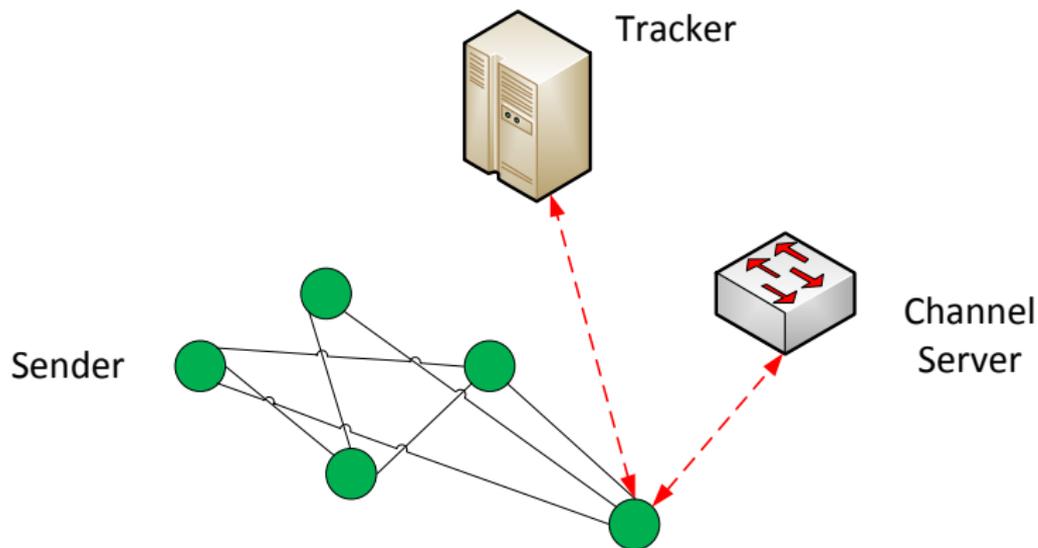


Abbildung: Server-Based Gruppenkommunikation

Server-Based

• Vorteil

- Geringer Managementaufwand für Peers
- Daten verfügbar solange es Peers gibt, die diese austauschen

• Nachteile

- Single-Point-of-Failure
- Hohe Kosten für den Betrieb
- Skalierungs-Probleme bei einer hohen Anzahl an Nutzern

Dezentralisiert

• Vorteil

- Single-Point-of-Failure
- Geringere Kosten
- Skaliert gut

• Nachteil

- Höherer Managementaufwand für die Peers
- keine Administration möglich (Sicherheitskopien)



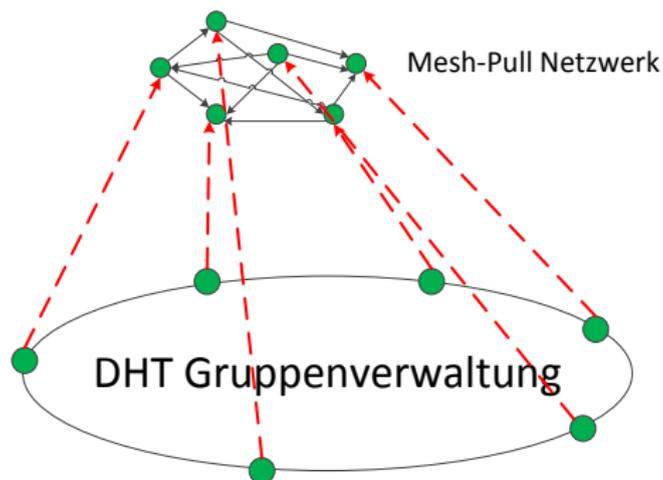


Abbildung: Pull-Multicast mit DHT basierter Gruppenverwaltung (1)

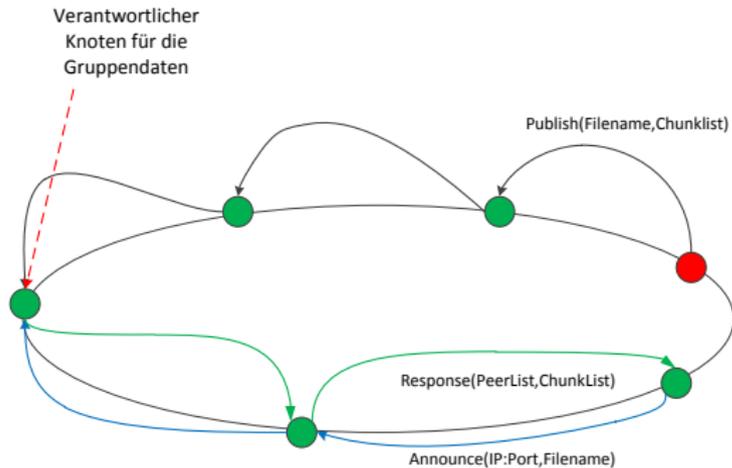


Abbildung: Pull-Multicast mit DHT basierten Gruppenverwaltung (2)

Outline

- 1 Einleitung
 - Motivation
 - Push und Pull
 - HAMcast
- 2 Masterarbeit
 - Gruppenverwaltung
 - API
- 3 Ausblick

API-Calls

Die vier wichtigsten HAMcast-Calls

- join()
- leave()
- send()
- receive()

- zusätzliche Service-Calls für z.B Monitoring
- keine Anpassung der API möglich
- Codierung von Modul spezifischen Argumenten über die URI
 - *scheme* *://* *group* *@* *instantiation* *:* *port* */* *sec-credentials*

Join/Subscribe

Aufgabe

- Das Interesse an Daten bekunden

Aufbau der URI

- `mesh://datei/group@BsNode:port`
 - **mesh://** Bezeichner für das Module
 - **datei/group** Name der gewünschten Datei und Gruppen adresse
 - **BsNode** Adresse für den Bootstrap Node (Optional)



Send/Publish

Aufgabe

- Zum Senden von Daten
- Call muss nur einmal ausgeführt werden

Aufbau der URI

- mesh://datei@BsNode:port
 - **datei** Name und Pfad der Datei
 - **BsNode** Adresse des Bootstrap Node



receive

Aufgabe

- Daten werden auf der Festplatte gespeichert
- Daten werden als Chunks zurückgegeben
 - Wenn keine Pfadangabe durch die Uri erfolgt ist
 - Wichtig für Video-Streaming

Zusammenfassung und Fazit

- Kozeptionierung der Gruppenverwaltung und Pull-Algorithmus
- Implementierung des Pull-Multicast Module
- Performance Verbesserung der DHT

Mögliche Verbesserungen Gruppenverwaltung

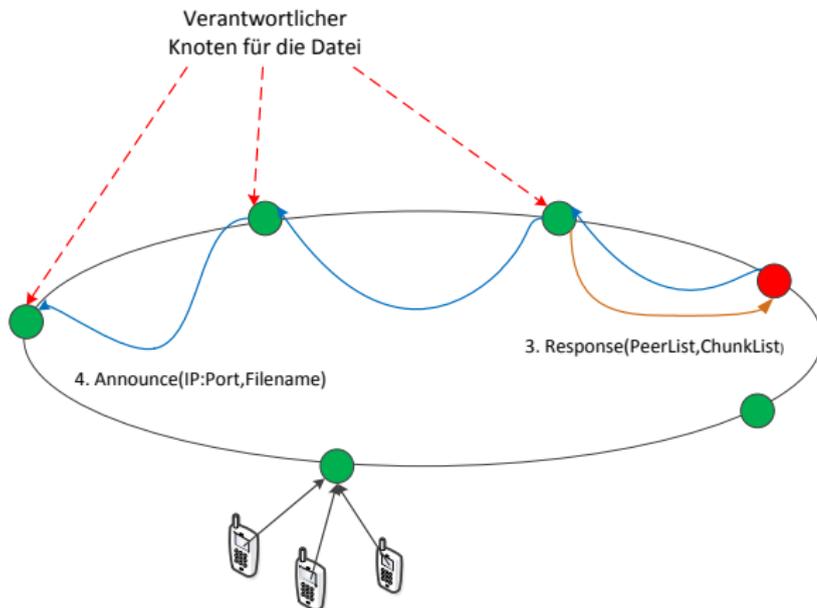


Abbildung: Verbesserungen des DHT Tracker

-  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>
-  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.
-  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.
-  J.-L. Kuo, C.-H. Shih, and Y.-C. Chen, "A comprehensive study of delivery strategies with chunk scheduling for mesh p2p live streaming," in *Proceedings of the 5th International Conference on Ubiquitous Information Management and Communication*, ser.

ICUIMC '11. New York, NY, USA: ACM, 2011, pp. 107:1–107:6. [Online]. Available: <http://doi.acm.org/10.1145/1968613.1968739>



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, Dec. 2010, pp. 913–917.



M. Wählisch, T. C. Schmidt, and S. Venaas, “A Common API for Transparent Hybrid Multicast,” IRTF, IRTF Internet Draft – work in progress 04, January 2012.



B. Cohen, “Incentives build robustness in bittorrent,” 2003.



P. Maymounkov and D. Mazières, “Kademlia: A peer-to-peer information system based on the xor metric,” in *Peer-to-Peer Systems*, ser. Lecture Notes in Computer Science, P. Druschel, F. Kaashoek, and A. Rowstron, Eds. Springer Berlin / Heidelberg, 2002, vol. 2429,

pp. 53–65, 10.1007/3-540-45748-85. [Online]. Available :http://dx.doi.org/10.1007/3-540-45748-8_5

R. Jimenez and B. Knutsson, “CTracker: a Distributed BitTorrent Tracker Based on Chimera,” in *Proc. eChallenges 2008*, vol. 2, Oct. 2008, pp. 941–947.