



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Mapping between group names and addresses in hybrid multicast

Sebastian Wölke

Ausarbeitung AW2

Inhaltsverzeichnis

1	Einleitung	1
2	Master Konzept	3
3	Verwandte Arbeiten	4
3.1	Universal Multicast	4
3.2	Delivery Centric Middleware	6
3.3	Zero Configuration Networking	8
4	Zusammenfassung und Ausblick	10

1 Einleitung

Anwendungen und Dienste wie Videokonferenzen, Internet Radio, IPTV und soziale Netze basieren auf Gruppenkommunikationen. Es werden dieselben Daten zu mehreren Empfängern gesendet. Eine effiziente und skalierbare Gruppenkommunikation ist Multicast, welches darauf ausgelegt ist, redundanten Datenverkehr zu vermeiden. Da allerdings die meisten Internet Service Provider (ISP) Multicast nicht aktiviert haben, existieren häufig nur lokale Multicast-Inseln mit den unterschiedlichsten Multicast-Technologien, wie zum Beispiel IPv4 Multicast, IPv6 Multicast und verschiedene Multicast-Technologien auf dem Overlay. Diese Technologien lassen sich wiederum zwischen verschiedene Varianten unterscheiden, wie zum Beispiel Any Source Multicast (ASM) und Source Specific Multicast (SSM). Die hohe Varianz erschwert den Einsatz von Multicast, weshalb es kaum genutzt wird und so für ISPs auch keine Anreiz gibt Multicast zu aktivieren. Eine Lösung für diesen Teufelskreis bietet das Projekt HVMcast [MCSW10] (Hybrid Adaptive Mobile Multicast) an. HVMcast ist eine Multi-Service Architektur, mit dem Ziel Multicast Inseln miteinander zu verbinden und die Technologien und Varianten zu integrieren.

HVMcast implementiert eine *Common Multicast API* [WSV13], um den Zugriff auf einen Technologie übergreifenden Multicast-Service zu ermöglichen. Diese API ist variantenübergreifend und erlaubt einen transparenten Zugriff im Under- und Overlay. Mithilfe einer Vermittlungsschicht der Middleware [MCSW12], wird selbstständig die effizienteste Multicast-Technologie im Netzwerk gewählt und verwendet. Die Middleware zieht so die Komplexität aus den Anwendungen und vereinfacht das Entwickeln neuer.

Um Multicast-Gruppen, trotz unterschiedlicher Technologien global identifizieren zu können, ist eine Abstraktion von Identifizierung und Lokalisierung erforderlich. Aufgrund dieses *Identifier-Locator-Splits* wird bei HVMcast zwischen einem technologieunabhängigen Gruppennamen (ID) und einer technologieabhängigen Gruppenadressen (Locator) differenziert. Dies führt dazu, dass in jeder Domäne eine Abbildung zwischen Gruppenname und Gruppenadresse benötigt wird. Wobei das Mapping einem Konzept des späten Bindens folgt, wodurch erst zur Laufzeit entschieden wird, welche Multicast-Technologie verwendet wird.

Motivation

Meine Aufgabe ist es, im HVMcast-Projekt, die technologietransparente Gruppenkommunikation über Gruppennamen umzusetzen. Zurzeit fehlt ein dynamisches Mapping zwischen Gruppennamen und Gruppenadresse, weshalb das Verwenden der einzelnen Multicast-Technologien und das Übersetzen zwischen den Technologien statisch vorkonfiguriert werden muss. Diese statischen Vorkonfigurationen sollen durch dynamische ersetzt werden, wodurch die Implementierung transparent und einsatztauglich wird.

Ziel dieser Ausarbeitung ist es Verwandte Arbeiten zu analysieren und nach Möglichkeit Erkenntnisse aus diesen zum Thema *dynamisches Mapping* zwischen technologieunabhängigen Gruppennamen und technologieabhängigen Gruppenadressen zu gewinnen.

Gliederung der Ausarbeitung

Die vorliegende Arbeit gliedert sich wie folgt: Im zweiten Kapitel wird mein Master-Konzept vorgestellt. Es wird in diesem Zusammenhang meine Aufgabenstellung analysiert und in Teilprobleme unterteilt, für die in Kapitel 3 in Verwandten Arbeiten nach Lösungen gesucht werden. Abschließend wird die Ausarbeitung kurz zusammengefasst und ein Ausblick für das weitere Vorgehen aufgezeigt.

2 Master Konzept

Das Gruppenmapping lässt sich in drei Einzelprobleme unterteilen. So können die Einzellösungen unabhängig voneinander getestet werden und bei Bedarf durch bessere ausgetauscht werden. Außerdem entsteht dadurch ein Baukastensystem, wodurch eine Anpassung an die jeweilige Multicast-Technologie ermöglicht wird.

Auf der untersten Ebene werden für alle Domänen-Teilnehmer konsistente Mapping-Informationen zur Verfügung gestellt (Intra Domain Mapping). Dabei wird angenommen, dass alle Teilnehmer gleichberechtigt sind. Diese Mapping-Informationen können zum Beispiel durch einen zentralen Mappingdienst bereitgestellt werden oder die Teilnehmer könnten untereinander Vereinbarungen treffen. Weiterhin kann ein Mapping auch ohne zusätzlichen Kommunikationsaufwand, zum Beispiel durch das Abbilden des Gruppennamens auf eine technologiespezifische Adresse durch eine Hashfunktion, realisiert werden. Diese Konzepte ermöglichen Clients einer Domäne, Gruppendaten über Gruppennamen auszutauschen.

Auf der zweiten Ebene werden sowohl Sender als auch Empfänger auf Namensebene in der Domäne bekannt gemacht. Denn der IMG (Inter Domain Gateway), mit dem Ziel die Gruppendaten über die Domänengrenzen hinweg zu leiten, könnte zum Beispiel eine Position im Netzwerk haben, an der er nicht alle Multicast Pakete sieht und erst die Gruppen abonnieren muss um die Daten zu empfangen. Deswegen muss ihm mit Hilfe von Join und Source Register Nachrichten explizit mitgeteilt werden, welche Sender und Empfänger in der Domäne existieren.

Auf der obersten Ebene werden die Gruppendaten nach bestimmten Regeln über die Domänengrenzen hinweg geroutet (Inter Domain Routing). Dabei können mehrere Annahmen getroffen oder Freiheiten zugelassen werden. So könnte in einem einfachen Fall angenommen werden, dass die IMGs baumförmig und gerichtet strukturiert sind. In einem komplexeren Szenario könnten aber auch, wie in einem Meshnetzwerk, die Topologie unstrukturiert sein, wodurch der Routingaufwand deutlich steigt.

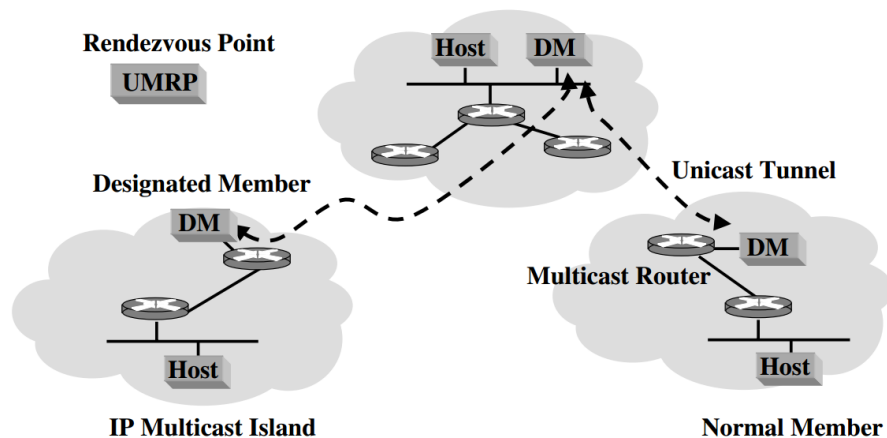
3 Verwandte Arbeiten

Ich möchte die verwandten Arbeiten UM (Universal IP Multicast Delivery), Juno (Delivery Centric Middleware) und Zeroconf (Zero Configuration Networking) vorstellen. UM [ZWJ⁺06] hat ähnliche Ambitionen wie HVMcast und möchte Multicast global verfügbar machen. UM stellt ein Multicast-Overlay vor und beschreibt wie Multicast-Inseln dynamisch mit diesem verbunden werden können. Juno ist eine Delivery-Centric Middleware [TMK⁺12] und hat dasselbe Grundkonzept wie HVMcast. Es beinhalten beide die Konzepte eine rekonfigurierbare Middleware mit Technologie Modulen, eine generische API, welche nach dem Publish/Subscribe Paradigma funktioniert und es wird der ID-Locator Split genutzt. Zeroconf [Che09] habe ich ausgewählt, weil mit deren Konzept sich das erste Teilproblem meines Master Konzeptes lösen lässt.

3.1 Universal Multicast

Das Ziel von *Universal IP Multicast Delivery* (UM) ist es eine globale und skalierende Gruppenkommunikation zur Verfügung zu stellen. Mit IP-Multicast können zwar sehr effektiv Gruppendaten verteilt werden, ist aber von dem Deployment der Router abhängig. Dem gegenüber stehen weniger performante Multicast-Overlays, welche die Multicast-Funktionalität auf End-Hosts auslagern, um die Abhängigkeit durch die Router zu umgehen. UM kombiniert ein Overlay-Konzept mit IP-Multicast, um Multicast Inseln zu verbinden und innerhalb dieser effizient Gruppendaten zu verteilen. Universal IP Multicast Delivery ist ein Framework, das aus drei Komponenten besteht. Das *Host multicast tree protocol* (HMTP) ist ein Multicast Overlay, welches die Multicast Inseln verbindet und über das *Host group management protocol* (HGMP) werden IP-Multicast Inseln an das Overlay angeschlossen. Zusätzlich stellt UM ein Daemon zur Verfügung, um Clients direkt an das Overlay anzubinden.

Abbildung 3.1 zeigt einen Architektur-Überblick mit drei Multicast-Inseln, welche mit Hilfe von *Designated Members* (DM) über Unicast-Tunnel miteinander verbunden sind. In jeder Insel gibt es einen oder mehrere DMs welche dynamisch pro Multicast Gruppe bestimmt werden und Gruppendaten in andere Multicast-Inseln weiterleitet. Der *Rendezvous Point* (UMRP) dient dabei

Abbildung 3.1: UM Architektur-Überblick, [ZWJ⁺06]

als Bootstrap Mechanismus und hilft DMs dem Overlay beizutreten.

Zur globalen und eindeutigen Identifizierung von Gruppen werden bei UM *Gruppen Identifier* (GID) verwendet. Ein GID setzt sich aus der IPv4-Adresse des zuständigen UMRP und einer beliebigen Gruppen-ID in der Form $GID := \langle \text{UMRP IP Adresse} \rangle + "/" + \langle \text{Gruppen ID} \rangle$ zusammen (z.B. $GID = 131.179.96.162/1234$). Für mehr Benutzerfreundlichkeit bietet UM die Möglichkeit Gruppennamen anstatt von GIDs zu verwenden. Der Gruppename lässt sich über ein zweistufiges Mapping auf eine GID abbilden. Zum Beispiel wird der Gruppename *forest.cs.ucla.edu/mytalk* zunächst von einem DNS-Dienst zur IP-Adresse 131.179.96.162 des UMRPs aufgelöst und anschließend wird *mytalk* vom UMRP selbst aufgelöst.

HMTTP ist ein End-Host Multicast Protokoll, das eine Baumstruktur zwischen den DMs aufbaut und sich mit Hilfe von Round-Trip-Time Messungen an der Topologie der unteren Schichten orientiert.

HGMP bindet, unabhängig von den jeweiligen Multicast-Protokollen, Multicast-Inseln an ein Multicast-Overlay. HGMP stellt dafür Insel-begrenzt eine *Well-Known Group* zur Verfügung, über welche die derzeitigen verwendeten Gruppen periodisch angekündigt werden. Alle Multicast-Clients horchen auf die Well-Known Group und können die angekündigten Gruppen beitreten. Wenn ein Client eine neue Gruppe erstellen möchte, ernennt er sich zum DM und definiert dynamisch die Multicast-Gruppen *DATA_GROUP* und *ASSERTION_GROUP*. Über die Gruppe *DATA_GROUP* werden die Gruppendaten in der Multicast-Insel verteilt und vom DM in das

Overlay geleitet. Zusätzlich sendet der DM periodisch ASSERTION Nachrichten als *Heart Beat* an die ASSERTION_GROUP. Fällt der DM aus oder sendet er eine *QUIT*-Nachricht wird über ein Auswahl-Verfahren ein neuer DM gewählt.

Relevanz

HVMcast baut auf die Idee von Universal Multicast auf und erweitert und optimiert Teile des Konzeptes. Trotzdem bietet UM eine protokollunabhängige Idee für das Abbilden der HVMcast-URI (Gruppenname) auf eine Gruppenadresse. Zwar ist der UM-Gruppenname weniger komplex als die HVMcast-URIs und benötigt ein aufwendiges dreistufiges Mapping von UM-Gruppenname über den UM Group Identifier zur IP-Multicast Adresse. Aber das Mapping von UM Group Identifier auf IP-Multicast Adresse wird periodisch über ein Well-Known Group angekündigt und dieses Konzept kann auf das Mapping in den HVMcast-Domänen adaptiert werden.

3.2 Delivery Centric Middleware

Juno ist eine Adaptive Delivery Centric Middleware mit dem Ziel einen generischen Zugriff auf Delivery-Services, wie zum Beispiel HTTP, FTP, BitTorrent und Gnutella zu ermöglichen. Zusätzlich können Anforderungen, wie Übertragungs-Geschwindigkeit, Sicherheit und Übertragungskosten, an die Zustellung des Contents gestellt werden. Juno besteht aus zwei Komponenten, einer Delivery-Centric API und einer dynamisch rekonfigurierbaren Middleware.

Der Content wird Technologie übergreifend, nach dem Prinzip des ID/Locater Splits, durch eine *ContentID* identifiziert. Dies stellt eine Vereinfachung sowohl für Anwender als auch Entwickler da, denn ein Anwender hat Interesse am Content und wie schnell oder mit welchen Kosten er zugestellt wird. Wo der Content abgerufen wird und welche der möglichen Quellen genutzt werden, wird zur Entlastung von der Juno Middleware selbstständig anhand der Zustellungsanforderungen entschieden. Dem Anwendungsentwickler ermöglicht dieses Konzept Zustellungs-transparente Software zu designen und zu warten. Dies verkürzt die Entwicklungszeit und vereinfacht das Erweitern durch neue Zustellungsdienste. Die ContentID kann ein einfacher Hashwert, aber auch ein komplexer Magnet Link [GOJ02] sein. Ein Magnet Link ist ein Link auf einen Content ohne einen Hinweis darauf zu geben, wo der Content gespeichert ist. Es handelt sich um einen Uniform Resource Identifier (URI), der den Content mit einer Menge von Hashes unterschiedlicher Hashverfahren kennzeichnet.

Die Zustellungsanforderungen werden über eine Menge des dreier Tupels (*attribute*, *comparator*, *value*) definiert. *Attribute* legt die Anforderung fest, zum Beispiel die Datenrate. *Comparator* ist ein Vergleichsoperator und *value* legt den Grenzwert fest. Ein Zustellungsanforderung kann zum Beispiel aus diesem dreier Tupel bestehen: (Datenrate, >, 50mbit/s) und (Sicherheitslevel, ==, max).

Die API stellt zwei Interfaces zur Verfügung, das Provider und das Consumer Interface. Aufgeteilt nach dem Publish/Subscribe Paradigma. Der Consumer kann Content mit definierten Zustellungsanforderungen abonnieren, dieses wieder rückgängig machen und auch die Zustellungsanforderungen dynamisch anpassen. Der Provider kann Daten zur Verfügung stellen, er gibt dafür einen InputStream an und eine Menge von Zustellungsanforderungen von denen er ausgeht, sie einhalten zu können. Weiterhin kann der Provider den Content wieder löschen.

Abbildung 3.2 zeigt den Aufbau der Juno Middleware. Auf der obersten Ebene greift die Anwendung über das Consumer- oder das Provider-Interface auf die Middleware zu. Das Discovery Framework ist für das Mapping zwischen ID und Locator zuständig und durchsucht bei einer Content-Anfragen der Anwendung alle ihm zur Verfügung stehenden Zustellungsdienste, welche als Plug-in nachgeladen werden können. In einem günstigen Fall liefert das Discovery Framework eine Menge an Quellen aus verschiedenen Technologien zurück und übergibt sie an das Delivery Framework. Das Delivery Framework hat anschließend die Aufgabe, die beste Quelle oder die besten Quellen für die Anwendung anhand der Zustellungsanforderungen auszusuchen. Daraufhin wird der Content je nach Bedarf zum Beispiel in eine Datei geschrieben oder als Stream zurück gegeben.

Relevanz

Bei dem Juno-Konzept sind viele parallelen zum HVMcast-Konzept zu finden. Juno nutzt wie HVMcast das Publish/Subscribe Paradigma, benötigt den ID/Locator Split und dem damit verbundenen Mapping zwischen diesen, stellt eine generische API zur Verfügung und kapselt die komplexen Anteile in einer dynamisch rekonfigurierbaren Middleware. Allerdings ist das ID/Locator Mapping von Juno nicht in das HVMcast-Projekt übertragbar, da die jeweils abstrahierten Technologien, zum Beispiel bei Juno *http* und bei HVMcast *IPv4-Multicast*, zu verschiedenartig sind.

benötigt, wenn zwei Netzwerke, die durch IPv4LL konfiguriert wurden, zusammengeschlossen werden. Dann kann es trotz passiver Konfliktvermeidung dazu führen, dass mehrere PCs dieselbe Adresse haben. Empfängt ein PC ein Paket mit seiner Adresse als Absender, dann wird diese noch von einem anderen verwendet. Jetzt hat dieser PC einmal die Chance seine Adresse zu verteidigen, indem er ein ARP-Paket verschickt. Entdeckt er den Konflikt noch mal muss, um eine Endlosschleife zu vermeiden, eine neue Adresse gewählt werden.

MDNS ist ein auf Multicast basierender Namensdienst. MDNS kann in ad-hoc-Netzwerken den Komfort erhöhen aber auch als lokale Erweiterung für den globalen DNS angesehen werden, da er kostenfrei und ohne Namensbeschränkung verwendet werden kann. Die MDNS Nachrichten-Pakete sind fast identisch zu den klassischen DNS-Nachrichten, außer das Anfragen an eine Multicast-Gruppe geschickt werden anstatt zu einem Server. Jeder Teilnehmer kann sich beliebige DNS-Einträge definieren. Befinden sich zum Beispiel zwei Drucker in einem Netzwerk, können sie sich beide den DNS Eintrag *Drucker* setzen. Fragt anschließend ein PC nach einer Namensauflösung für einen Drucker, antworten beide mit der jeweiligen Adressauflösung. Wobei die DNS-based Service Discovery die Namen standardisiert und strukturiert. MDNS bietet außerdem einige Optimierungen, um die Netzwerklast möglichst gering zu halten.

Relevanz

Zeroconf ermöglicht eine Autokonfiguration von Netzwerken mit dessen Konzepten das erste Teilproblem meines Master Konzeptes (Kapitel 2) gelöst werden kann. Das IPv4LL Konzept ermöglicht eine Vergabe und kontinuierlicher Kontrolle von eindeutigen IP-Adressen, welches auch dazu genutzt werden kann, um Mapping-Informationen konsistent zu halten. Weiterhin bietet MDNS ein Konzept, mit dem ein Mapping zwischen Name und Adresse von einer Gruppe erfragt wird, welches zum Beispiel vom IMG oder von anderen Teilnehmern genutzt werden kann. Allerdings skaliert Zeroconf nicht, IPv4LL ist nur auf kleine Netzwerke anwendbar, da es mit Broadcast-Nachrichten arbeitet und es zusätzlich durch den kleinen Adressraum (169.254.0.0/16) schnell zu Adresskonflikten kommt, die mit weitem Broadcast-Nachrichten aufgelöst werden müssen.

4 Zusammenfassung und Ausblick

Zu Beginn wurde mein Master Konzept vorgestellt, welches das Mapping zwischen Gruppennamen und Gruppenadresse in drei Teilprobleme unterteilt, um die Komplexität der Gesamtaufgabe zu verkleinern. Im ersten Teilproblem, dem Intra Domain Mapping, werden allen Teilnehmern konsistente Mapping Informationen zur Verfügung gestellt. Anschließend werden sowohl Sender als auch Empfänger auf Namens Ebene in der Domäne bekannt gemacht, um auf Basis dieser Informationen domänenübergreifend Routingentscheidungen zu treffen (Inter Domain Routing).

Anschließend wurden die verwandten Arbeiten UM, Juno und Zeroconf vorgestellt. Das Framework UM (Universal IP Multicast Delivery) wurde ausgewählt da es ähnliche Ambitionen wie HVMcast hat und Multicast global verfügbar machen möchte. Zur globalen Identifizierung werden GIDs verwendet, die dynamisch auf Multicast-Adresse abgebildet werden. Bekannt gemacht wird das Mapping periodisch über eine Well-Known-Group. Dieses Mapping-Konzept kann auf das Mappingproblem im HVMcast-Projekt adaptiert werden und löst die ersten beiden Teilprobleme des Master Konzeptes.

Juno ist ein Adaptive Delivery Centric Middleware, mit dem Ziel einen generischen Zugriff auf Delivery-Services zu ermöglichen. Juno nutzt wie HVMcast eine rekonfigurierbare Middleware, eine generische API und IDs um den Content Technologie übergreifend identifizieren zu können. Allerdings ist Mapping zwischen ID und Locator nicht auf das HVMcast-Projekt übertragbar, da die abstrahierten Technologien zu verschiedenartig sind.

Zeroconf ist eine Sammlung von Protokollen zur automatischen Konfiguration von ad-hoc Netzwerken. Zwei der Protokolle lassen auf das Mappingproblem im HVMcast-Projekt adaptieren. IPv4LL koordiniert eine eindeutige Vergabe von IP-Adressen unter den Netzwerkteilnehmer, welches dazu genutzt werden kann, um Mapping-Informationen konsistent zu halten. MDNS ist ein verteilter Namensdienst auf Multicast-Basis, mit dessen Konzept Mapping-Informationen erfragt werden können.

Sowohl UM als auch Zeroconf bieten Ansätze für ein Mapping-Konzept, die im nächsten Schritt geprüft und miteinander verglichen werden. Genügt eine der Konzepte den Skalierungsanforderungen, wird es praktisch umgesetzt und getestet. Anschließend wird darauf aufbauend das Inter Domain Routing untersucht.

Literaturverzeichnis

- [CAG05] S. Cheshire, B. Aboba, and E. Guttman. Dynamic Configuration of IPv4 Link-Local Addresses. RFC 3927, IETF, May 2005.
- [Che09] Stuart Cheshire. Zero configuration networking, <http://www.zeroconf.org/>, June 2009.
- [CK11a] Stuart Cheshire and Marc Krochmal. DNS-Based Service Discovery. Internet-Draft – work in progress 11, IETF, December 2011.
- [CK11b] Stuart Cheshire and Marc Krochmal. Multicast DNS. Internet-Draft – work in progress 15, IETF, December 2011.
- [GOJ02] GOJOMO@bitzi.com. Magnet-uri project, <http://magnet-uri.sourceforge.net/>, June 2002.
- [MCSW10] Sebastian Meiling, Dominik Charousset, Thomas C. Schmidt, and Matthias Wählisch. System-assisted Service Evolution for a Future Internet – The HAMcast Approach to Pervasive Multicast. In *Proc. of IEEE GLOBECOM 2010, Workshop MCS 2010*, pages 913–917, Piscataway, NJ, USA, Dec. 2010. IEEE Press.
- [MCSW12] Sebastian Meiling, Dominik Charousset, Thomas C. Schmidt, and Matthias Wählisch. HAMcast – Evaluierung einer systemzentrierten Middleware-Komponente für einen universellen Multicast-Dienst im Future Internet. *Praxis der Informationsverarbeitung und Kommunikation (PIK)*, 35(2):83–89, Mai 2012.
- [TMK⁺12] Gareth Tyson, Andreas Mauthe, Sebastian Kaune, Paul Grace, and Thomas Plagemann. Juno: An adaptive delivery-centric middleware. In *CCNC*, pages 587–591, 2012.
- [WSV13] Matthias Wählisch, Thomas C. Schmidt, and Stig Venaas. A Common API for Transparent Hybrid Multicast. IRTF Internet Draft – work in progress 07, IRTF, January 2013.

- [ZWJ⁺06] Beichuan Zhang, Wenjie Wang, Sugih Jamin, Daniel Massey, and Lixia Zhang.
Universal ip multicast delivery, 2006.