

Integration des Overlay-Multicast-Dienstes Ariba/MCPO in die hybride Multicast-Plattform HVMcast

Nora Berg

Nora.Berg@haw-hamburg.de

iNET RG, Department of Computer Science
Hamburg University of Applied Sciences

9. Januar 2013



Hochschule für Angewandte
Wissenschaften Hamburg
Hamburg University of Applied Sciences

- 1 Einführung
- 2 Future Internet Frameworks
- 3 Umsetzung
- 4 Fazit & Ausblick

Motivation

- viele Anwendungen würden von Gruppenkommunikation profitieren
- Multicast zur effizienten Datenverteilung in Gruppen

Problemstellung

- Ende-zu-Ende-Verbindungen teilweise eingeschränkt
- Heterogenität in Netzwerktechnologien und Protokollen
- Multicast auf fast allen Ebenen des Netzwerkstacks
- Programmierer muss während der Entwicklung entscheiden, in welcher Domäne das Programm laufen soll

HVMcast

- allgemeine Schnittstelle für Gruppenkommunikation
- abstrahiert von Multicast-Technologien
- verbindet Multicast-Domänen

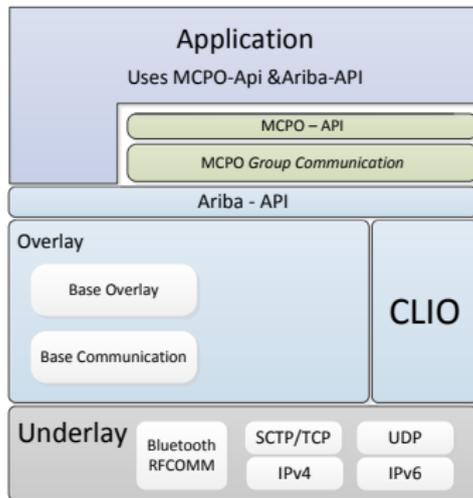
Ariba/MCPO

- Ariba: Ende-zu-Ende-Verbindungen unabhängig von Übertragungstechnologien
- MCPO: Multicast-Overlay über Ariba

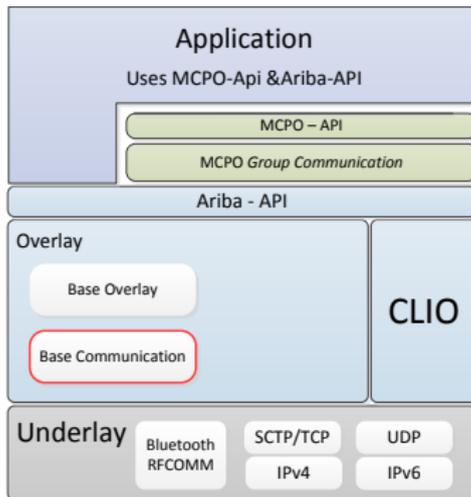
- Erweiterung von H \forall Mcast um Multicast-Technologie Ariba/MCPO
- Erhöhung der Reichweite von H \forall Mcast durch die von Ariba implementierten Netzwerk-Technologien

- Ariba
 - zuverlässige Ende-zu-Ende Datenübertragung
 - Peer-2-Peer Netzwerk (genannt: *SpoVNet*)
 - verbirgt Protokollheterogenität vor der Anwendung
 - überbrückt Middleboxes (Firewalls, NATs)
 - unterstützt Mobilität
- MCPO
 - Multicast-Overlay über Ariba mittels NICE-Protokoll

- Ein Endpunkt pro Netzwerkinterface im Underlay
- Overlay abstrahiert von Netzwerk-Technologien & verwaltet P2P-Netzwerk
- CLIO: stellt Informationen aus unterliegenden Schichten bereit (optional)
- Applikation benutzt 2 APIs



- Verbindungen über die einzelnen Underlay-Protokolle
- realisiert ID/Locator-Split
 - *ID*: Knoten ID (Hashcode über Knotennamen etc...)
 - *Locator*: eine Adresse pro Netzwerkinterface
- Heartbeats, ob Adresse noch aktiv ist
- Datenübertragung per TCP oder SCTP

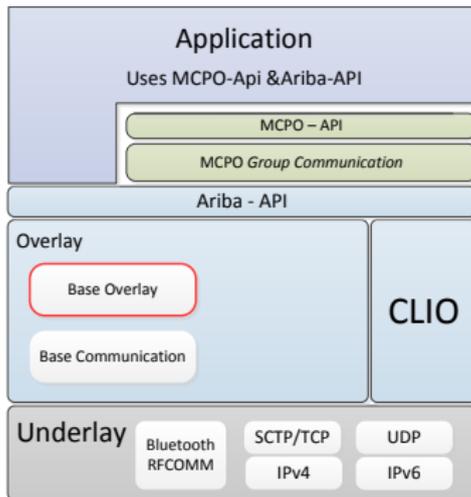


2 Arten Verbindungen

- *Direkte Verbindungen*: 2 Knoten können nativ miteinander kommunizieren
- *Indirekte Verbindungen*: 2 Knoten müssen über einen dritten Knoten (Relay) kommunizieren (TURN)
- *Relay*: Protokollübersetzung und NAT-Traversal

Base Overlay

- Organisation des SpoVNet
- Umwandeln der Knoten ID in den zugehörigen Endpunkt-Deskriptor
 - *Chord*: zum Finden von Knoten, und zur initialen Positionierung des eigenen Knotens im SpoVNet
 - Datenübertragung per TCP bzw. SCTP an die zum Zielknoten gehörige Adressen

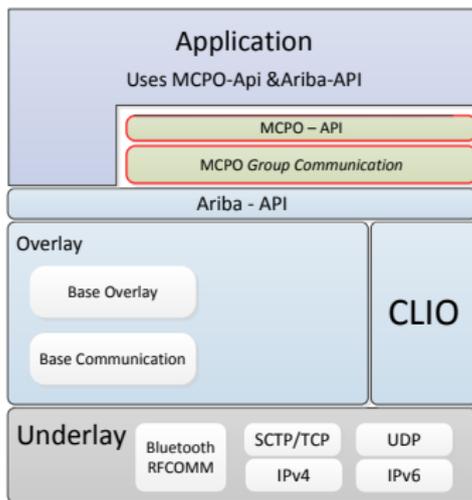


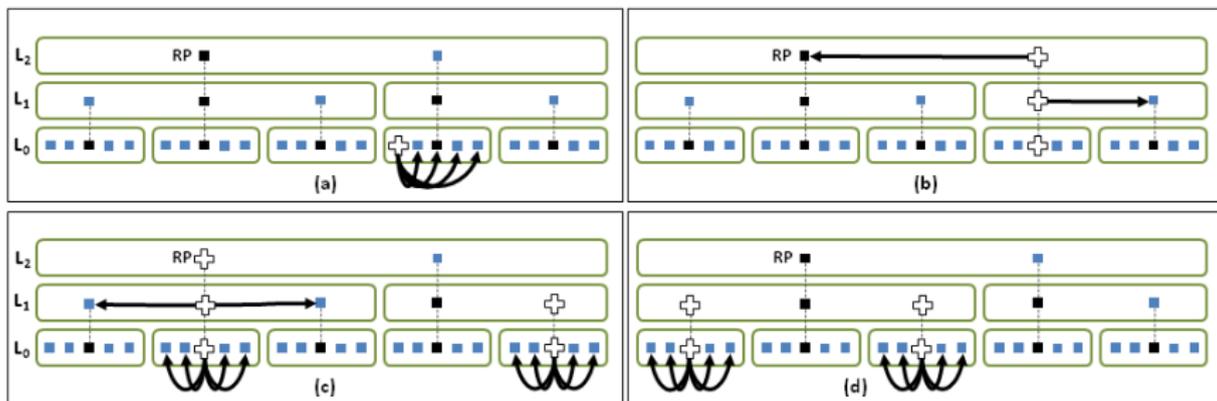
Group Communication

- verwendet das NICE-Multicast-Protokoll
- implementiert Ariba-API

MCPO-API

- stellt Multicast-Funktionen zur Verfügung (*join, leave, sendToGroup*)
- Applikation benötigt zusätzlich Bestandteile der Ariba-API
- Gruppenadressierung mittels Unsigned Integer





■ Common Member ■ Cluster Leader ⊕ Current Forwarding Member

1

- Knoten in Cluster
- Leader-Election pro Cluster
- Cluster-Hierarchie aus Leadern
- Nachrichten-Flooding innerhalb eines Clusters
- Clustering benötigt Verwaltung bei *join* und *leave*

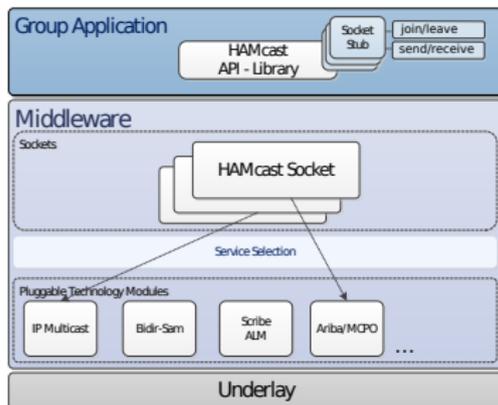
¹<http://ariba-underlay.org/wiki/Documentation/MCPO>

Überblick

- bietet universellen Zugriff auf verschiedene Multicast-Technologien über allgemeine API
- abstraktes Namensschema für Gruppenadressierung
- Interdomain Multicast Gateway (IMG) zur Vermittlung zwischen Multicast-Domänen

Komponenten

- Multicast-API biete abstrakte Multicast-Funktionen (*join, leave...*)
- Middleware verbindet Client mit Technologie-Modul
- Modul implementiert Schnittstelle zu Multicast-Technologie

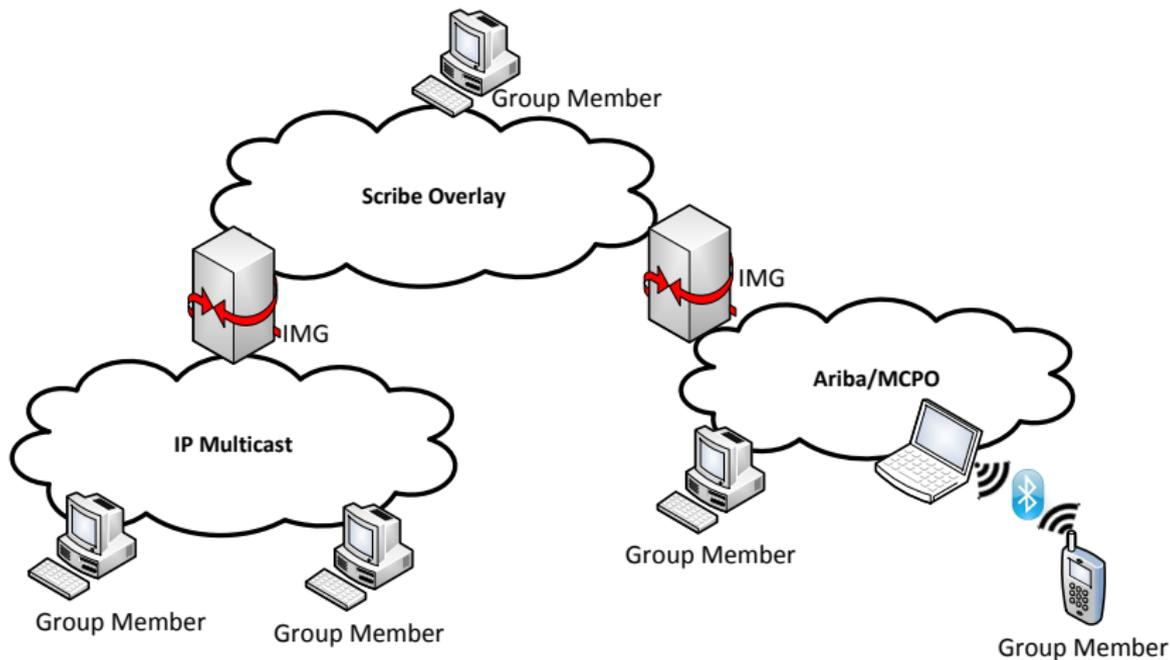


Adressierung

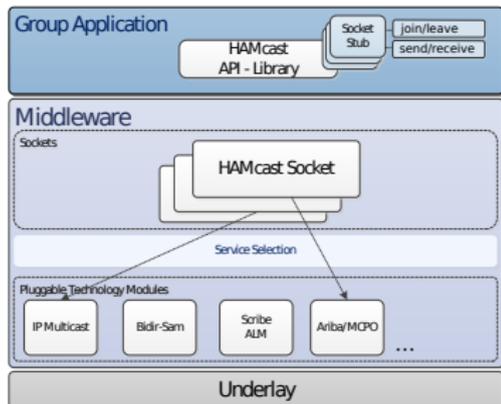
Addressierung mittels Multicast-URI :

`scheme: // group @ instantiation : port / credentials`

Beispiel

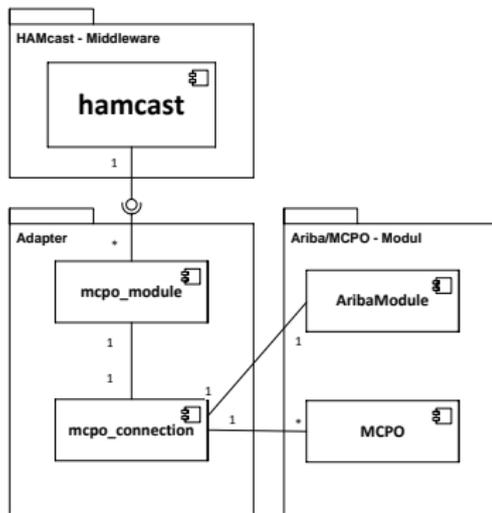


- HAMcast-Modul als Wrapper für Ariba/MCPO
- Schnittstellen beider Architekturen werden implementiert
- Abbildung zwischen Multicast-Funktionen
- Abbildung der HAMcast-URI auf die MCPO-Adressierung



Adapter-Architektur

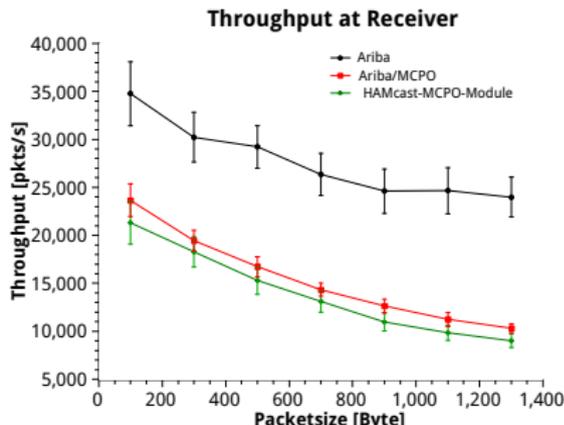
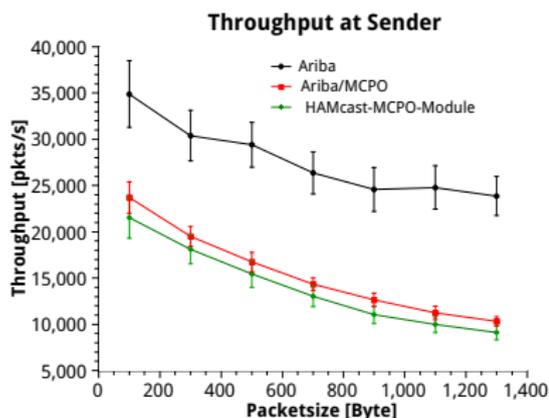
- `mcpo_module`
 - implementiert `hamcast_module` - Schnittstelle
 - initialisierung verwaltete `mcpo_connection` instanzen
 - verwaltet Ariba-Interfaces
- `mcpo_connection`
 - implementiert die Ariba/MCPO - Schnittstelle
 - bildet URI-Host auf MCPO-Objekt ab
 - entspricht einem Netzwerkinterface



URI-Mapping

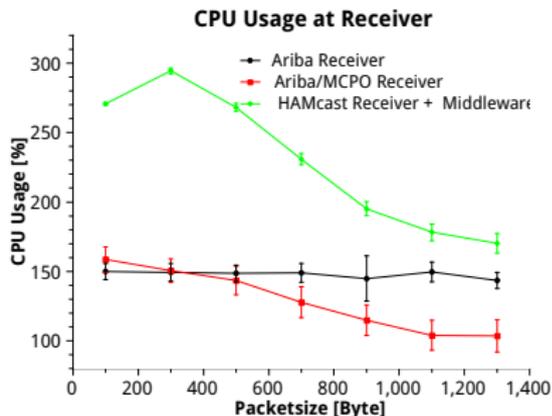
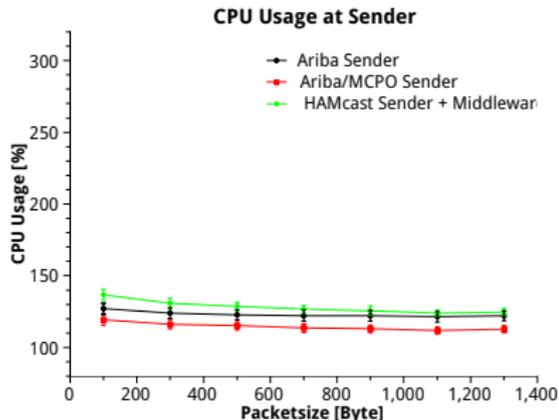
- mehrere Ariba-SpoVNETs über `Middleware.ini`
- `HAMcast`-URI-Host auf ein `MCPO`-Objekt
- `HAMcast`-URI-Port auf eine `MCPO`-Port (Group)

- 2 PCs direkt über Gigabit - Netzwerk verbunden
- Verschiedene Paketgrößen zwischen 100-1300 Byte
- gemessene Werte:
 - Durchsatz
 - CPU-Belastung
 - Paketverlust
- im Vergleich zwischen:
 - Ariba
 - Ariba/MCPO
 - Middleware + MCPO-Modul



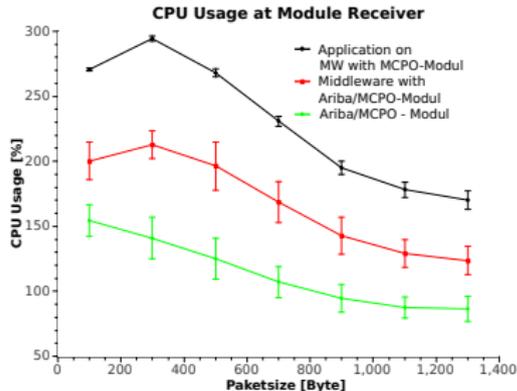
- Ariba-Unicast schneller, da ohne komplexes NICE-Overlay
- Einfluss des Adapters gering
- Durchsatz deutlich geringer als bei Scribe oder BIDIR-SAM
- Flaschenhals ist weder Adapter noch Netzwerk
- Minimaler Paketverlust durch MCPO

- Ariba
 - zuverlässig, weil TCP-Unicast
 - daher kein Paketverlust
- MCPO
 - konzeptionell *kein* zuverlässiger Multicast
 - geringer Paketverlust (bis 0.005% max)
- Ariba/MCPO - Adapter
 - nur bei Paketen ≤ 500 Byte
 - geringer Paketverlust (bis 0.05% max)
- Messung: 100.000 Pakete versendet (Dauer: ~ 50 sek)
- alle 20 Sekunden im Durchschnitt verschwindet 1 Paket
- Pro Messung \varnothing 2.85 Pakete



- bei Sender konstante CPU-Belastung
- Auslastung bei Receiver abhängig von Paketanzahl

- CPU-Belastung steigt mit Anzahl der empfangenen Pakete
- Middleware: Zusatzbelastung durch Kopieroperation
- Applikation: Zusatzbelastung durch Libhamcast mit IPC



- H \forall Mcast wurde um die Funktionalitat Ariba/MCPOs erweitert
 - MCPO als sehr stabiles Overlay
 - durch Ariba selbst bei mobilen Benutzern oder Middleboxes
 - zusatzliche Bluetooth Funktionalitat
- (fast) kein Paketverlust
- geringer Paketdurchsatz (ein zehntel von z.B. Scribe-Modul)
- mithilfe der H \forall Mcast-API ist Ariba/MCPO einfach und vielseitig benutzbar
 - sogar von Java aus mittels H \forall Mcast-Java-API

- Ist es *sinnvoll* ein nicht-zuverlässiges Layer über ein zuverlässiges zu implementieren?
- Wäre es konzeptionell nicht besser MCPO zu einem *zuverlässigen Multicast-Dienst* zu machen?
- Inwiefern *beeinflussen* sich die Frameworks (HAMcast und Ariba/MCPO) gegenseitig?
- Welche *Auswirkungen* hat das?



Thank you for your attention.
Questions?

iNET: <http://inet.cpt.haw-hamburg.de>