

# Mapping of group names and addresses in hybrid multicast

Sebastian Wölke

HAW Hamburg

31.05.2012

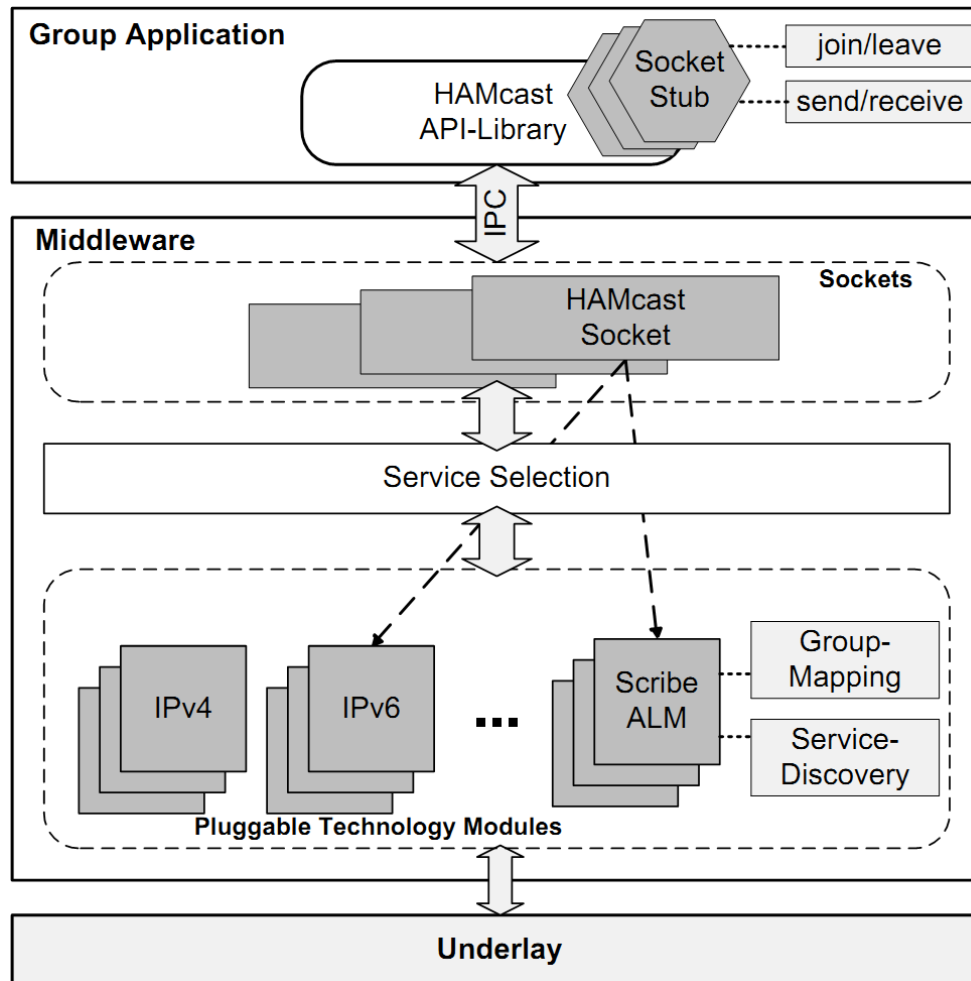
# Übersicht

- **HAMcast**
- URI Mapping
- Mapping-Verfahren
- Das Sourcen Problem

# Ziele von HAMcast

- Integration verschiedenster Multicast Technologien
- Generische Multicast-API
  - Komplexität aus den Anwendungen nehmen

# HAMcast Prototype Implementierung



# Terminologie

## Anwendungs ID (Identifizier)

- URI, Gruppenname (GN)
- Sourcename (SN)

## Routing ID (Locator)

- Gruppenadresse (GA)
- Sourceadresse (SA)

## Multicast Domain

- eine Technologie
- Host, Router, ...

## Inter-domain Multicast Gateway (IMG)

- verbindet Multicast Domains

## ASM

- Any Source Multicast

## SSM

- Source Specific Multicast

# URI - Uniform Resource Identifier

scheme :// group @ instantiation : port / sec-credentials

## Beispiele:

ip://239.99.99.99@141.22.27.212

ip://[ff15::123:456:7]



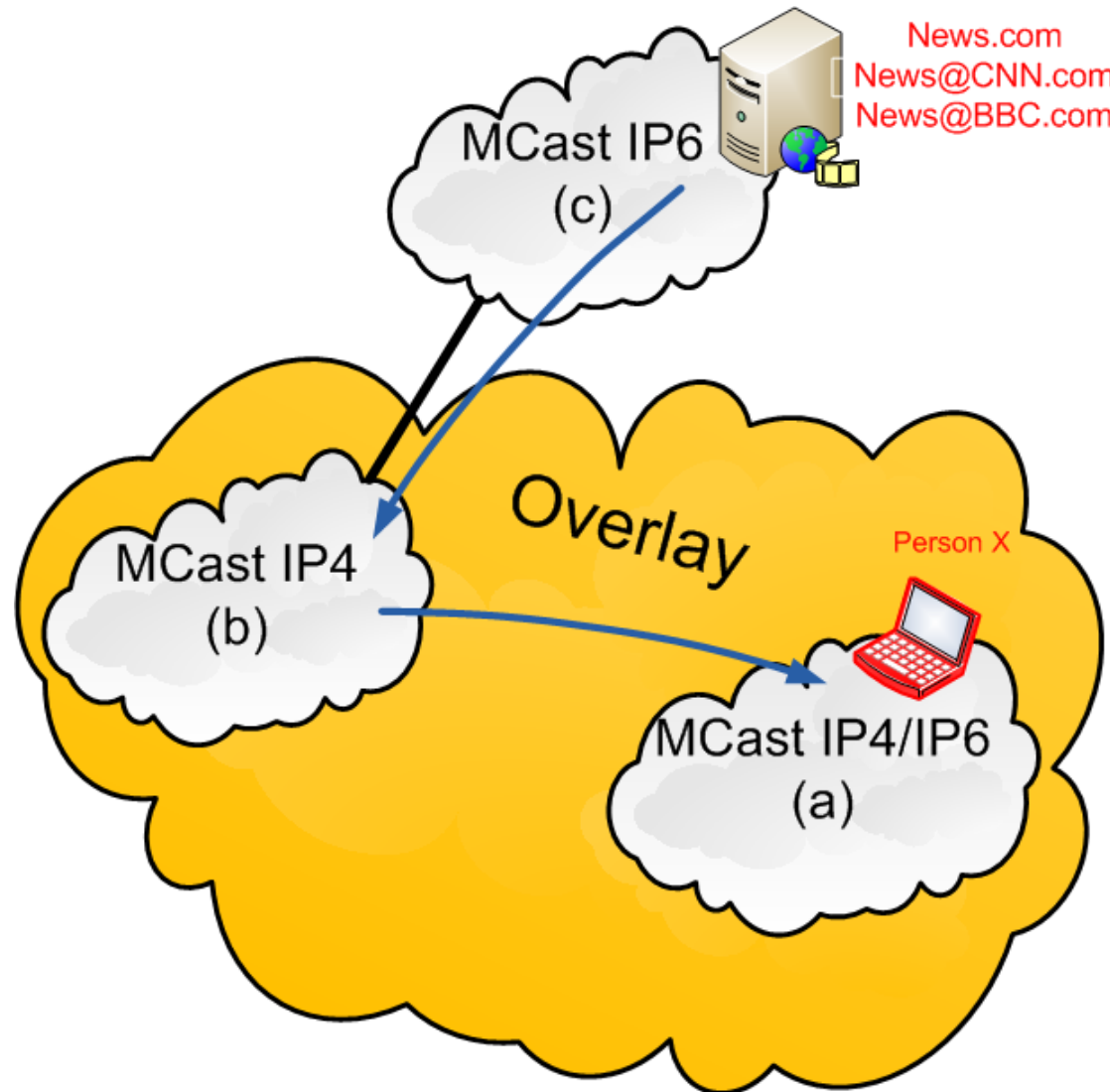
URI-Embedding

opaque://News:1234

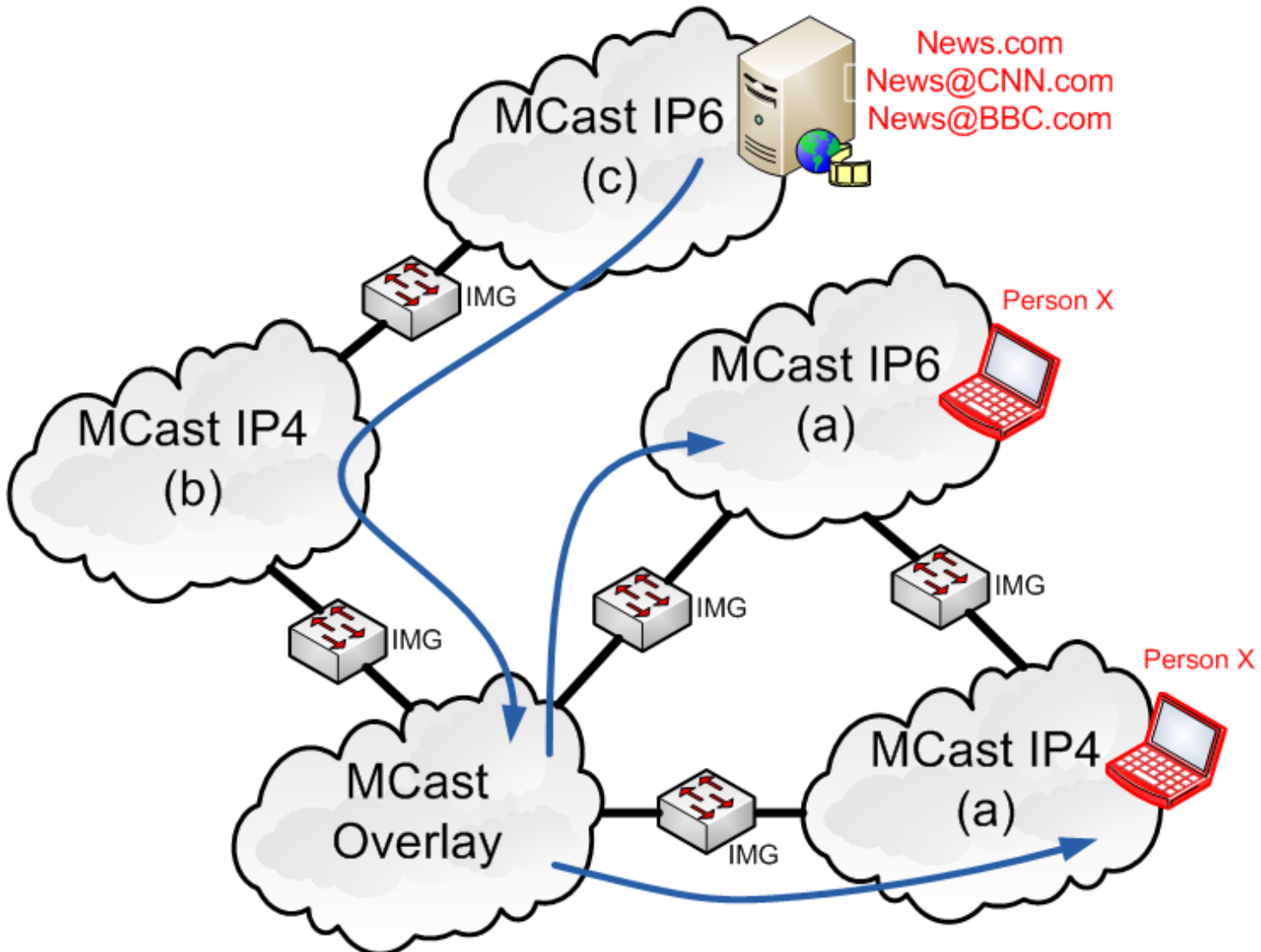
opaque://News@\*:1234

opaque://News@BBC.com:5678

# HAMcast Szenario - logische Sicht



# HAMcast Szenario - tatsächliche Topologie

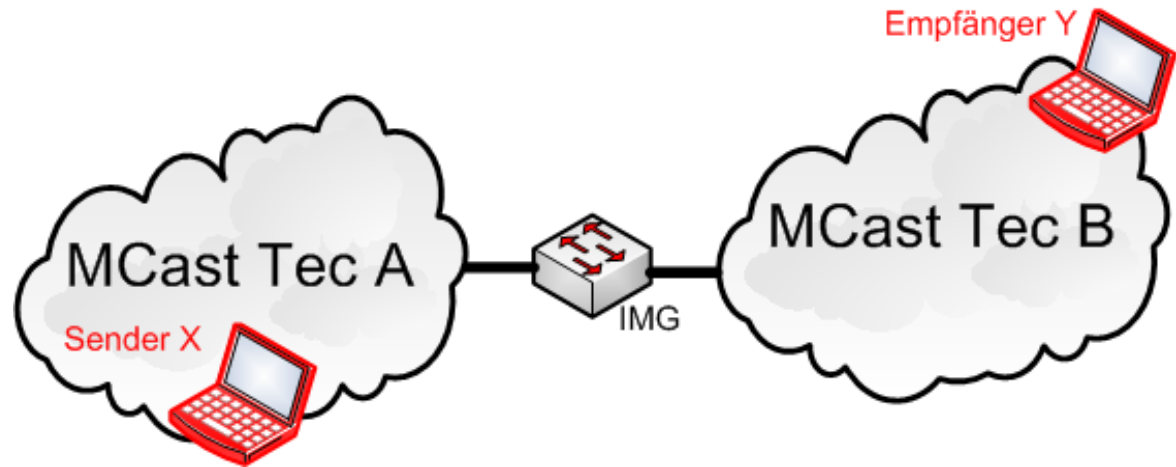




# Übersicht

- HAMcast
- **URI Mapping**
- Mapping-Verfahren
- Das Sourcen Problem

# URI Mapping und Recovery



URI ==> IP4

URI <== IP4

URI ==> IP6

URI <== IP6

URI ==> Overlay Hash

URI <== Overlay Hash

URI ==> Namespace X

URI <== Namespace X

# Strategien zum Erlernen und Finden von Mapping-Informationen #1

## Explizites Lernen

- Hashfunktionen
- Dynamisches Allokieren
- Erfragen
- Lookup-Table

## Implizites Lernen

- URI in einem Tec spezifischen Paketen enthalten

## URI-Embedding

- `ip://239.99.99.99@141.22.27.212`

# Kardinalität von Adress- und Namensräumen

## IP4-Multicast

- 224.0.0.0 bis 239.255.255.255
- $2^{28}$  mögliche Gruppen

## IP6-Multicast

- ff00::/8
- $2^{120}$  mögliche Gruppen

## Overlay Hash

- z.B.  $2^{160}$  mögliche Gruppen

## URI Namensräume

- beliebig groß

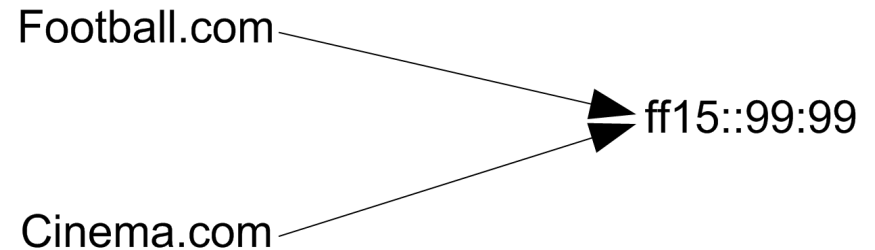
# Kollisionswahrscheinlichkeit - Geburtstagsparadoxon

$$P = 1 - \frac{m \cdot (m - 1) \cdot (m - 2) \cdots (m - (n - 1))}{m^n}$$

Voraussetzung: Kollisionsresistente Hashfunktion

n Gruppen-Namen	P für eine IP4-Domain (m=2 <sup>28</sup> )	P für eine IP6-Domain (m=2 <sup>120</sup> )
10	0,0000001676	3,385*10 <sup>-31</sup>
100	0,0000184400	3,723*10 <sup>-29</sup>
500	0,0004646222	9,385*10 <sup>-28</sup>
1.000	0,0018590546	3,757*10 <sup>-27</sup>
5.000	0,0454899478	9,402*10 <sup>-26</sup>
10.000	0,1699324542	3,761*10 <sup>-25</sup>
100.000	0,9999999918	3,761*10 <sup>-23</sup>

# Auswirkungen von Kollisionen #1



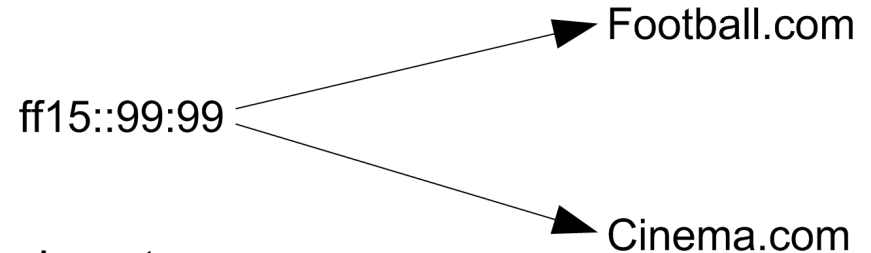
## ASM

- Hosts muss die Streams auf Anwendungsebene trennen.

## SSM

- Streams können anhand der Source-Adresse unterschieden werden
- Trennen der Streams auf Netzwerkebene.

# Auswirkungen von Kollisionen #2



Voraussetzung: Kollision wird vom IMG erkannt.

## ASM

- IMG muss die Gruppenadresse an beide Gruppennamen weiterleiten.
- Empfänger muss Gruppendaten auf Anwendungsebene filtern.

## SSM

- Streams können an Hand der Source-Adresse unterschieden werden
- Trennen der Streams auf Netzwerkebene.

# Übersicht

- HAMcast
- URI Mapping
- **Mapping-Verfahren**
- Das Sourcen Problem



# Strategien zum Erlernen und finden von Mapping-Informationen #1

## Explizites Lernen

- Hashfunktionen
- Dynamisches Allokieren
- Erfragen
- Lookup-Table

## Implizites Lernen

- URI in Tec spezifischen Paketen enthalten

## URI-Embedding

- `ip://239.99.99.99@141.22.27.212`

# Strategien zum Erlernen und Finden von Mapping-Informationen #2

	Explizites Lernen	Implizites Lernen	URI-Embedding
<b>URI =&gt; GA (IMG)</b>	Hashfunktion	--	falls technologie-spezifische URI
	Dynamisch Allokieren		
	Lookup-Table		
<b>GA =&gt; URI (IMG)</b>	<del>Hashfunktion</del>	URI ist in Daten Paketen enthalten	falls nativer Host
	Lookup-Table		
<b>URI =&gt; GA (Host)</b>	Hashfunktion	hört das Tuppel (URI, GA) im Netzwerk	falls technologie-spezifische URI
	Erfragen		
<b>GA =&gt; URI (Host)</b>	--	--	--

# Mapping-Verfahren Stateless #1

	Explizites Lernen	Implizites Lernen	URI-Embedding
<b>URI =&gt; GA (IMG)</b>	Hashfunktion	--	falls technologie- spezifische URI
	Dynamisch Allokieren		
	Lookup-Table		
<b>GA =&gt; URI (IMG)</b>	Hashfunktion	URI ist in Daten Paketen enthalten	falls nativer Host
	Lookup-Table		
<b>URI =&gt; GA (Host)</b>	Hashfunktion	hört das Tuppel (URI, GA) im Netzwerk	falls technologie- spezifische URI
	Erfragen		
<b>GA =&gt; URI (Host)</b>	--	--	--

# Mapping-Verfahren Stateless #2

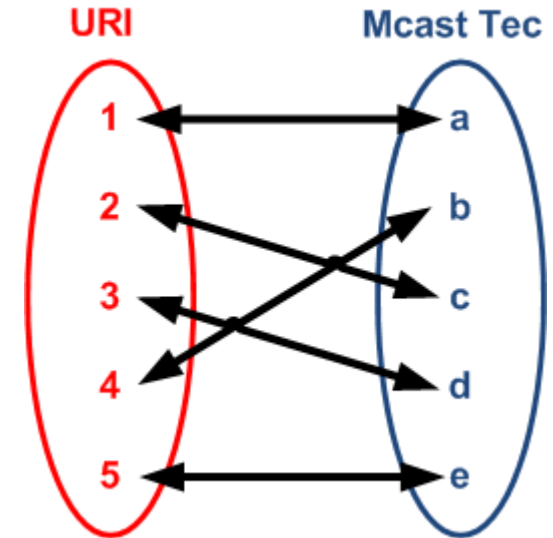
Perfekt wäre ein zustandsloses Mapping, welches ein Reverse-Mapping erlaubt.

Pro:

- Mapping-Aufwand  $O(1)$
- Benötigt keinen Lookup Speicher

Contra:

- Bild und Urbildmenge müssen gleich groß sein



nicht auf HAMcast anwendbar

# Mapping-Verfahren Stateful #1

	Explizites Lernen	Implizites Lernen	URI-Embedding
<b>URI =&gt; GA (IMG)</b>	Hashfunktion	--	falls technologie- spezifische URII
	Dynamisch Allokieren		
	Lookup-Table		
<b>GA =&gt; URI (IMG)</b>	<del>Hashfunktion</del>	URI ist in Daten Paketen enthalten	falls nativer Host
	Lookup-Table		
<b>URI =&gt; GA (Host)</b>	Hashfunktion	hört das Tuppel (URI, GA) im Netzwerk	falls technologie- spezifische URII
	Erfragen		
<b>GA =&gt; URI (Host)</b>	--	--	--

# Mapping-Verfahren Stateful #2

Bei dem Stateful Verfahren wird keine Heuristik verwendet. Gruppenadressen werden dynamisch von IMGs vergeben und von Hosts erfragt.

Pro:

- generisch, funktioniert bei alle Technologien
- keine Kollisionen

Contra:

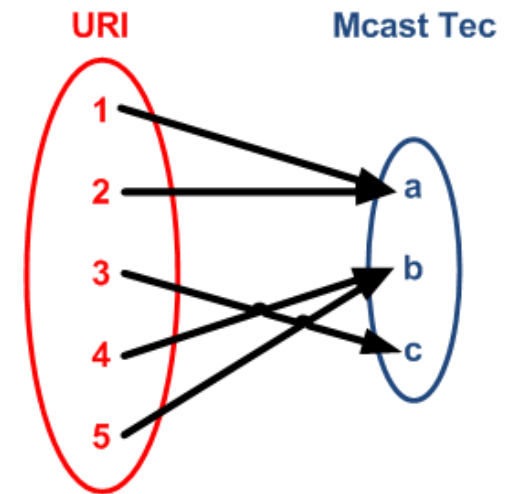
- benötigt Lookup Speicher
- hard / soft states
- Mapping Aufwand zwischen  $O(1)$  und  $O(n)$
- IMG fragen kostet Zeit

# Mapping-Verfahren Half Stateless #1

	Explizites Lernen	Implizites Lernen	URI-Embedding
<b>URI =&gt; GA (IMG)</b>	Hashfunktion	--	falls technologie-spezifische URII
	Dynamisch Allokieren		
	Lookup-Table		
<b>GA =&gt; URI (IMG)</b>	<del>Hashfunktion</del>	URI ist in Daten Paketen enthalten	falls nativer Host
	Lookup-Table		
<b>URI =&gt; GA (Host)</b>	Hashfunktion	hört das Tuppel (URI, GA) im Netzwerk	falls technologie-spezifische URII
	Erfragen		
<b>GA =&gt; URI (Host)</b>	--	--	--

# Mapping-Verfahren Half Stateless #2

Bei dem Half Stateless Verfahren wird mit einer Hashfunktion aus der URI eine Gruppenadresse erstellt. Reconvert wird die URI über ein Lookup Table.



Pro:

- Mapping von Gruppen-Namen nach Gruppenadresse schneller ( $O(1)$ )

Contra:

- Kollisionen möglich



# Unbekannte Source- und Gruppenadressen #1

	Explizites Lernen	Implizites Lernen	URI-Embedding
<b>URI =&gt; GA (IMG)</b>	Hashfunktion	--	falls technologie-spezifische URII
	Dynamisch Allokieren		
	Lookup-Table		
<b>GA =&gt; URI (IMG)</b>	<del>Hashfunktion</del>	URI ist in Daten Paketen enthalten	falls nativer Host
	Lookup-Table		
<b>URI =&gt; GA (Host)</b>	Hashfunktion	hört das Tuppel (URI, GA) im Netzwerk	falls technologie-spezifische URII
	Erfragen		
<b>GA =&gt; URI (Host)</b>	--	--	--

# Unbekannte Source- und Gruppenadressen #2

Ist die Source- oder die Gruppenadresse nicht bekannt lässt sie sich in eine technologiespezifische URI einbetten, zum Beispiel:

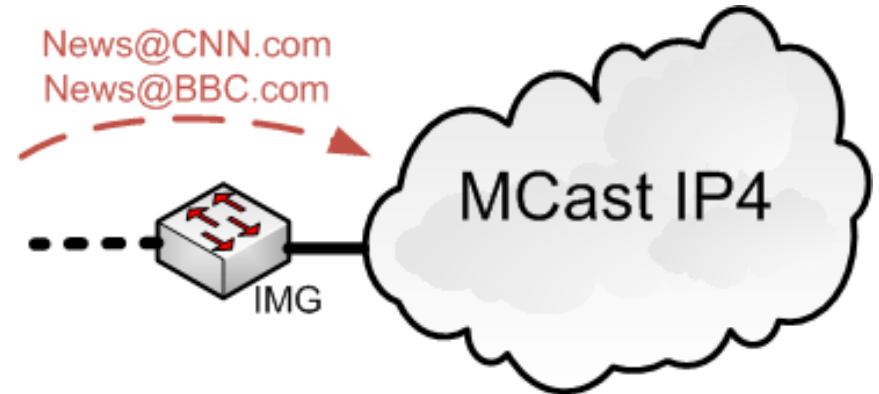
```
ip://group@source
```

Andere Technologien handhaben die URI wie jede andere auch.

# Übersicht

- HAMcast
- URI Mapping
- Mapping-Verfahren
- **Das Sourcen Problem**

# Das Problem mit den Quellen



Werden z.B. die Gruppen **News@CNN.com** und **News@BBC.com** in eine IP4-Domäne geroutet, ändert sich die Source zum IMG.



Die Gruppendaten lassen sich nicht mehr unterscheiden

# Zusammenfassung und Ausblick

- HAMcast
- URI Mapping
- Mapping-Verfahren
- Das Sourcen Problem

A glowing white speech bubble with a thick white outline and a soft white glow. Inside the bubble, the text "noch Fragen?" is written in a bold, white, sans-serif font. The word "noch" is on the top line, and "Fragen?" is on the bottom line. The speech bubble has a tail pointing towards the bottom-left corner.

**noch  
Fragen?**

# Quellen

**[DC90]** DEERING Stephen; CHERITON David; Multicast Routing in Datagram Internetworks and Extended LANs. ACM Trans. Comput. Syst., 8(2):85–110, 1990.

**[INET 2012]** INET group. HAMcast - Hybrid Adaptive Mobile Multicast. Website, 2012. Available online at <http://www.realmv6.org/hamcast.html>; visited on May 25th 2012.

**[Wählisch u. a. 2010]** WÄHLISCH, Matthias ; SCHMIDT, Thomas C. ; VENAAS, Stig: A Common API for Transparent Hybrid Multicast. URL <http://tools.ietf.org/html/draft-waehlich-sam-common-api>, July 2010 (04). – IRTF Internet Draft – work in progress

**[Cotton u. a. 2010]** COTTON M.; VEGODA L.; MEYER D. IANA Guidelines for IPv4 Multicast Address Assignments, IETF, RFC 5771, March 2010

**[Arkko u. a. 2010]** ARKKO J.; BRADNER S.; IANA Allocation Guidelines for the IPv6 Routing Header, IETF, RFC 5871, May 2010

**[Wählisch u. a. 2012]** WÄHLISCH, Matthias ; SCHMIDT, Thomas C.; Why We Shouldn't Forget Multicast in Name-oriented Publish/Subscribe. Open Archive: arXiv.org, 2012. Technical Report