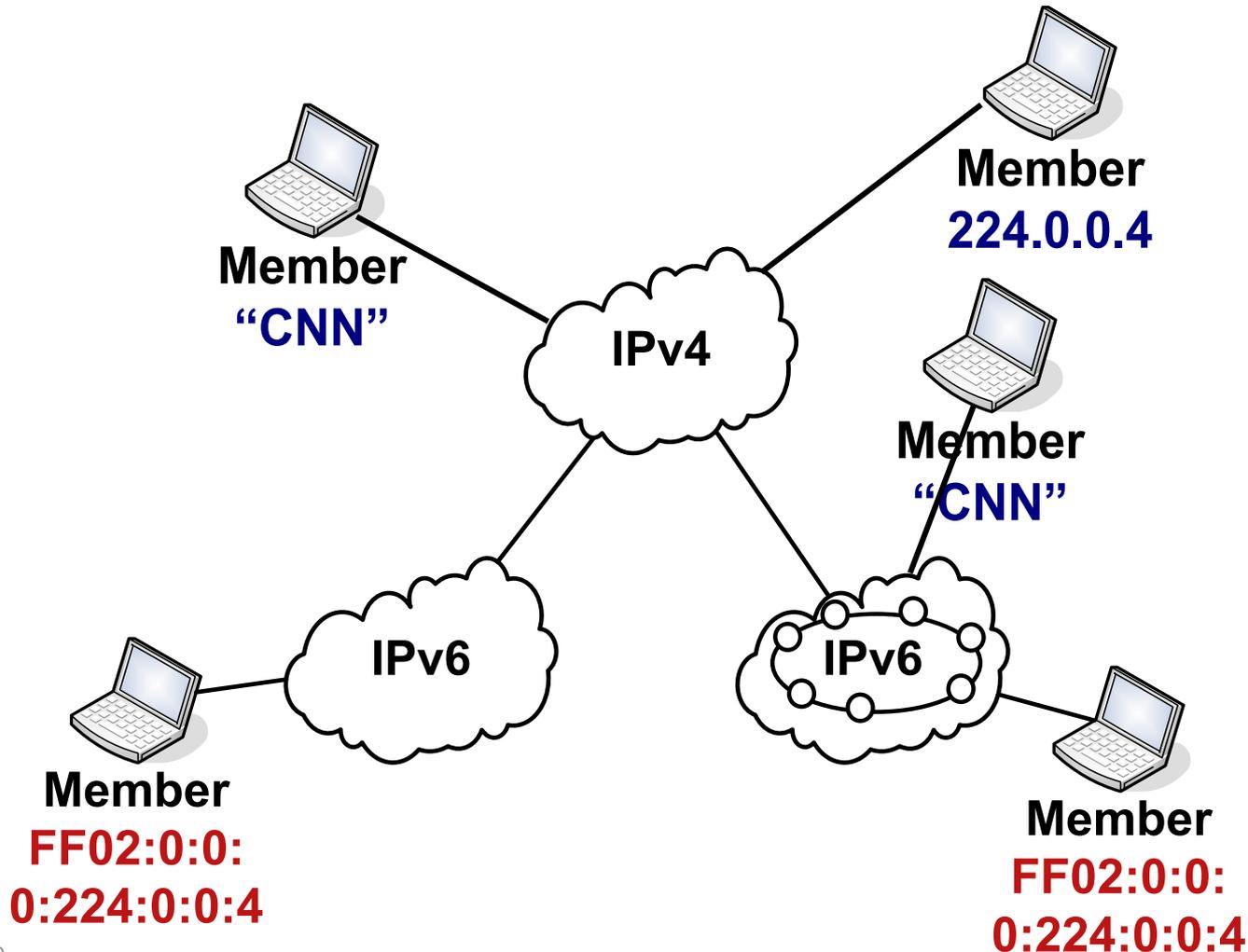


# A Common API for Transparent Hybrid Multicast

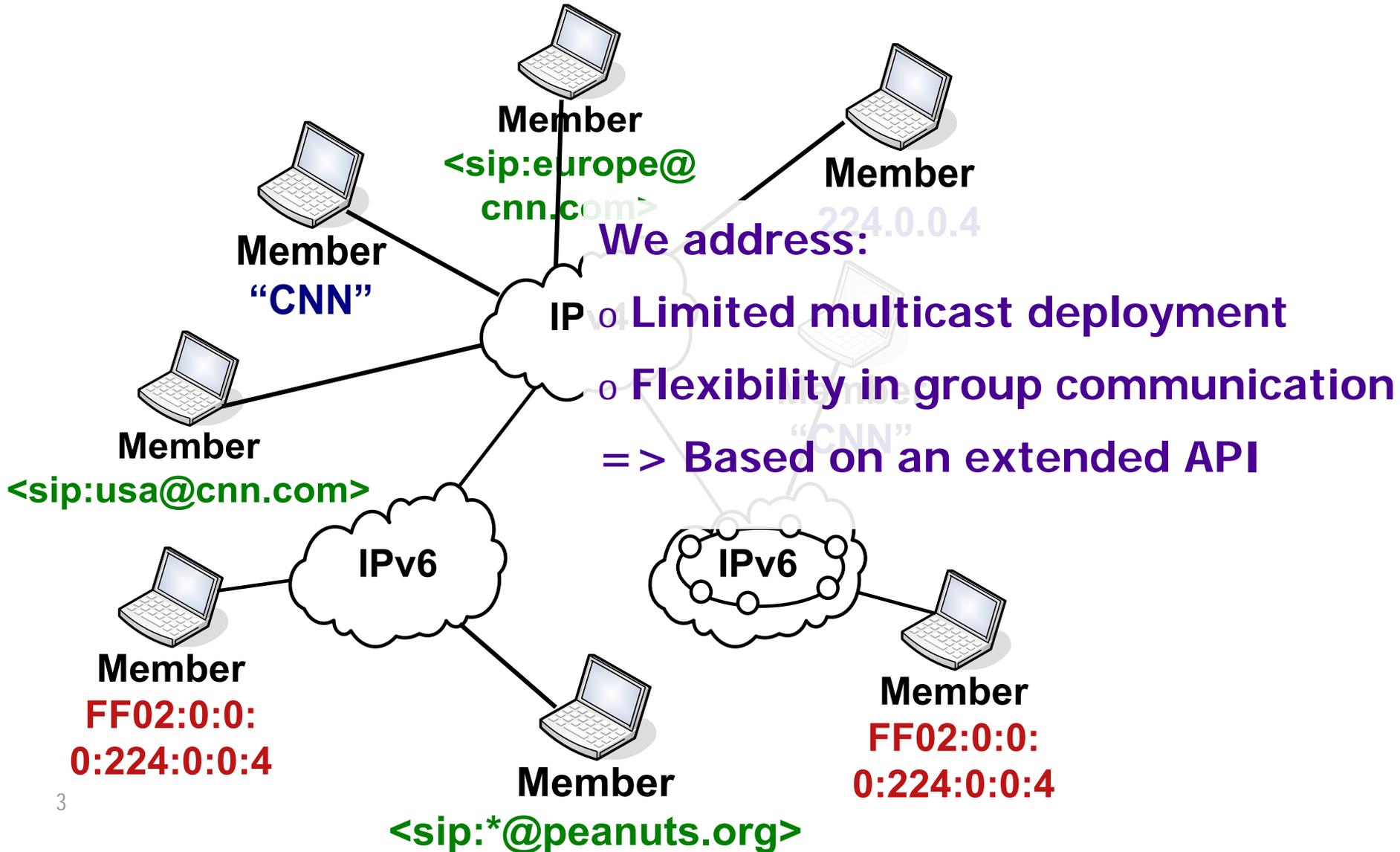
Matthias Wählisch, Thomas C. Schmidt  
Stig Venaas

{waehlich, t.schmidt}@ieee.org,  
stig@cisco.com

# Current State – Example: Join CNN



# Future – Example: Join \*.CNN.COM



# Agenda

1. Problem Statement
2. Requirements
3. Overview of our solution
4. API for Endhosts and Gateways
5. Open Issues
6. Conclusion

# Problem Statement

- o Group communication is implemented on **different layers** and is based on **different technologies**
  - This results in **several forwarding paths** and **varying** group addresses (**namespaces**)
  - ... and makes application programming really complicated

## Objectives:

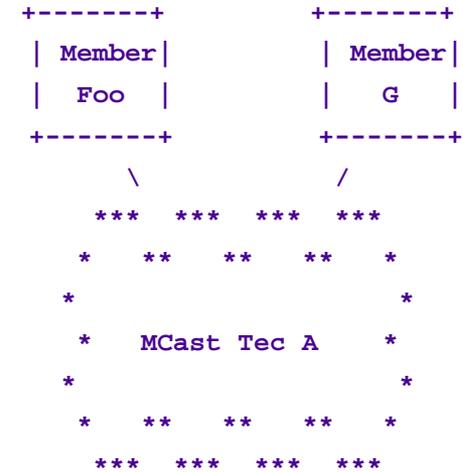
1. Enable any application programmer to implement independently of underlying delivery mechanisms
2. Make applications efficient, but robust w.r.t. deployment aspects

# Requirements

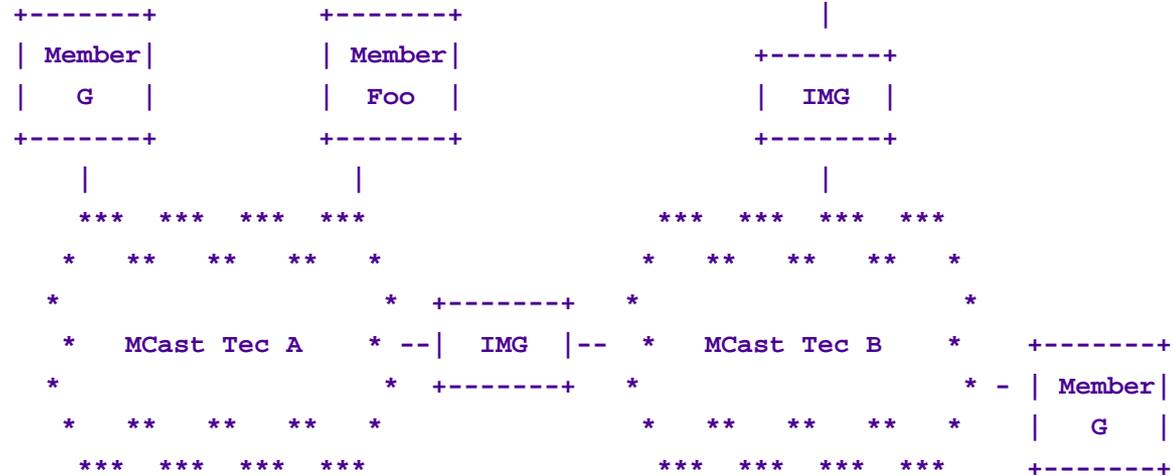
- o Design of a common group communication API
- o Flexible namespace support in group addressing
  - Separate routing and addressing scheme from application design
- o Mapping between different namespaces
- o Allocation of group addresses
- o Gateway function to forward multicast data between different technologies
  - Consistent view on multicast states at a single host

# Reference Scenarios

- o Domains running same technology but remaining isolated

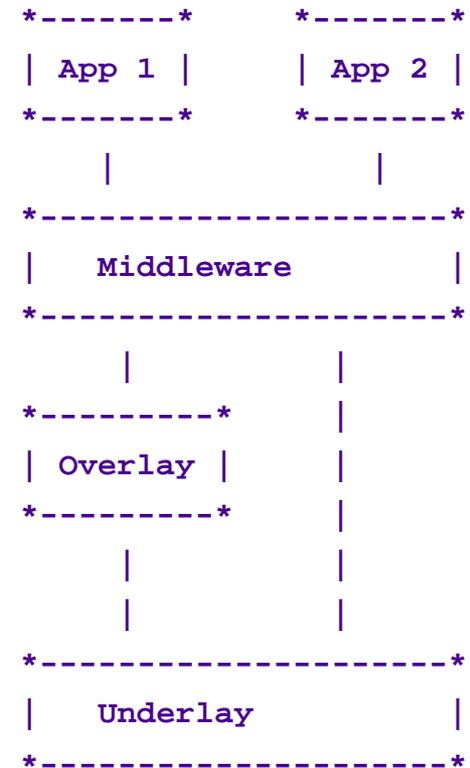


- o Domains running distinct technologies but hosts are members of the same group



# Overview

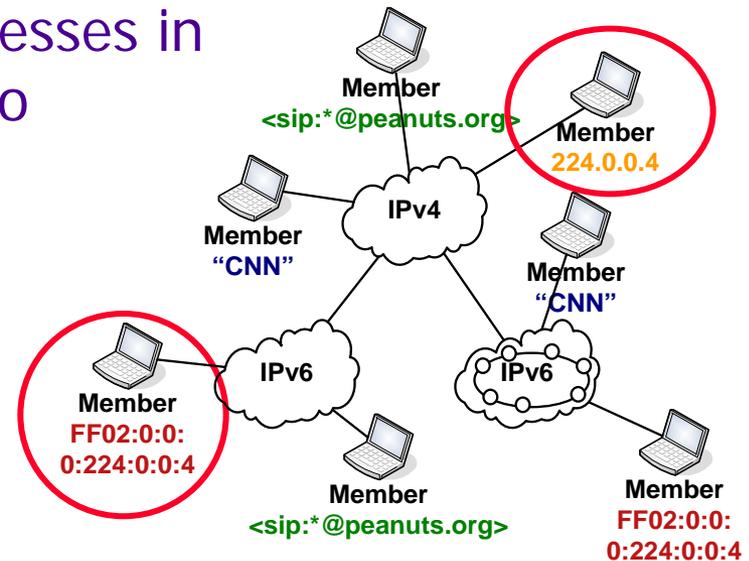
- o Extended multicast functions implemented by a middleware
- o Middleware
  - Provides extended API
  - Bridges data between technol.
- o **General procedure**
  1. App. subscribes/leaves/sends to a **logical group ID**
  2. Middleware maps logical ID to **technical group ID**
  3. Technical ID is allocated or revised if already in use



# Namespace Issue (or Challenge ...)

Scenario: Two (or more) different addresses in different namespaces may belong to

- (1) the same multicast channel (same technical ID)
  - (2) different multicast channels (different technical IDs)
- o Requires some mapping rule
    - Semantic meaning of logical ID?
  - o Can be solved based on an invertible mapping
    - Does not hold in general (cardinality of namespaces)
    - Example: Mapping IPv6 to IPv4
- => We need a common ID per group



# Assumptions

## o Assumptions:

- All group members subscribe to the same **logical** group ID from the same namespace
- There is a 'global' mapping service to map logical to technical group ID(s)
- Domain composition and node attachment to specific technology remain unchanged during multicast session

## o Problem: Traditional applications

- Inter-domain multicast gateway bridges data

# Send/Receive Calls – Required for Endhosts and Gateways

- Mode: Defines multicast technique
- `init(in Namespace n)`
  - Pre-initializes the namespace for a group
- `join(in Address a, in Mode m)`
  - Subscribes to a group
- `leave(in Address a, in Mode m)`
- `send(in Address a, in Mode m)`

# Service Calls – Required for Gateways

- `groupSet(out Address[] g, in Mode m)`
  - Returns all registered multicast groups
- `neighborSet(out Address[] a, in Mode m)`
  - Returns the set of multicast neighbors
- `designatedHost(out Bool b, in Address a)`
  - Checks if the host is designated router
- `updateListener(out Address g, in Mode m)`
  - Upcall informs about change of listener states
- `updateSender(out Address g, in Mode m)`
  - Upcall informs about change of source states

# Open Issues

- o Mapping service for logical to technical ID (e.g., DHT)
- o Encoding of routing addresses and technologies at the mapping service
- o ASM service via SSM delivery
- o Any scenarios not covered by the API?

# (Partial) Conclusion

- o API enables technology-agnostic programming of group-oriented applications
- o API can be used to implement hybrid multicast gateway
  - Draft describes interaction with IP-layer multicast routing protocols (PIM-SM etc.)

# ASM Service via SSM Delivery: Background

- o ASM: Any Source Multicast
  - Multicast address ( $*$ ,G)
- o SSM: Source Specific Multicast
  - Multicast address (S,G)
  - Simplifies routing

**Question:** How do we realize an ASM service over SSM in our architecture?

# Approach

- o (S,G) control tree with S addresses the IMG
- o The IMG manages source subscriptions
  - Receivers joins at least (S,G)
  - IMG learns somehow sources
  - IMG uses (S,G) as control tree
  - Receivers joins all  $S_i$  via SSM
- o How do we learn sources?
  - Hybrid multicast gateway must know sources of its domain
  - Inter-IMG communication to unify source information