SAM API Concepts Revisited

Matthias Wählisch, Thomas C. Schmidt

Xiaoming Fu, Jun Lei

{waehlisch, t.schmidt}@ieee.org

{fu, lei}@informatik.uni-goettingen.de

Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

link·lab

Freie Universität Berlin

GEORG-AUGUST-UNIVERSITÄT
GÖTTINGEN

Motivation and Objectives

o Motivation

    o Transparent application development

        o Operational for both: ALM and native Multicast

    o Reusable code, simulation modules, etc.

o Two perspectives

    - Application: Communication with middleware API

    - ALM stack: "Pluggable" Internet communication modules

o Objectives

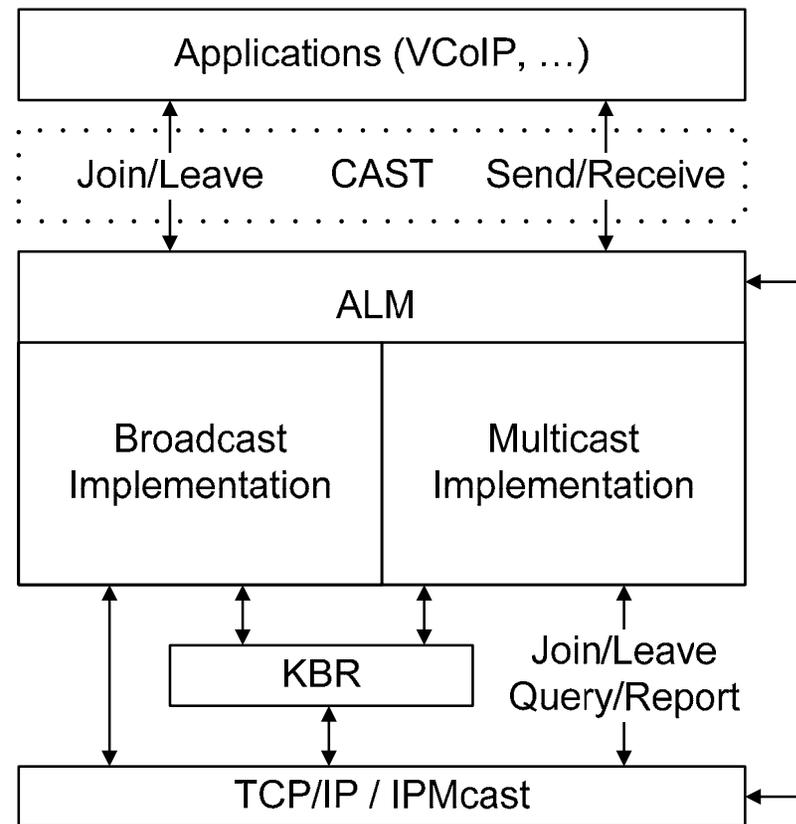   - Common group API for developers

    - Basic packet definition

Dabek et al. Model

o Tier0-2: Several middleware layers between application and network

o Tier3: End user application

o CAST defines shared tree approach and API calls

o Multicast scheme depends on the tier2 application

    - CAST should be understood as wrapper

Dabek et al., 2003

3
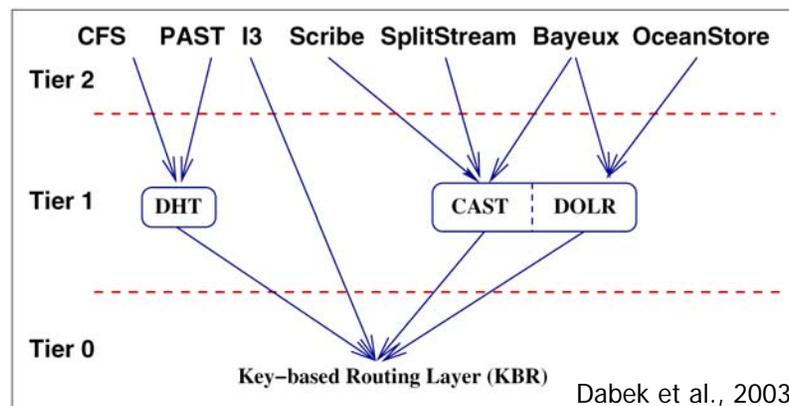
Looking at End User App: Basic Setup

o **Join a Multicast group**

- Which type of address?

o **State maintenance**

- Listener queries/reports

o **ALM-API should provide dedicated broadcast**

- Delivery to all members w/o active subscription

```
┌──────────────────────────────────────────┐
│          Applications (VCoIP, …)          │
└──────────────────────────────────────────┘
  ↕                              ↕
  Join/Leave    CAST    Send/Receive
  ↕                              ↕
┌──────────────────────────────────────────┐
│                   ALM                     │
├────────────────────┬─────────────────────┤
│     Broadcast      │     Multicast        │
│  Implementation    │  Implementation      │
└────────────────────┴─────────────────────┘
     ↕         ↕              ↕
        ┌──────────┐      Join/Leave
        │   KBR    │     Query/Report
        └──────────┘
     ↕         ↕              ↕
┌──────────────────────────────────────────┐
│            TCP/IP / IPMcast               │
└──────────────────────────────────────────┘
```

Issue #1: Namespace (1)

o Namespaces:

- Network addresses: IP (Unicast, Multicast)

- Application addresses: URI (HTTP, SIP, File System, …)

- And probably more … ??

o ALM-Middleware may handle addresses in the following way:

- Direct address mapping to overlay group address

- Semantic interpretation of address

o Delivery upcall may want to recover original address in namespace

- Root peer ID is unknown for application and can change

Issue #1: Namespace (2)

o Namespace support allows, e.g., group aggregation

o Ex.: *@irtf.org for sam@irtf.org, mobopts@irtf.org, etc.

- Compound namespaces

- Equal to broadcast in a dedicated namespace

o Possible approach for SCRIBE:

- Decompose address

- Cascade RP per namespace

- Source sends data to 'level0' address

o Possible approach for CAN: Use CAN dimension

Issue #2: Broadcast

a) Application sends msg. to dedicated broadcast channel

 - Overlay must internally avoid collisions with bcast key

b) Application sends to reserved, but well-known address

 - Regular IPv4/v6 broadcast address

 - All-systems/nodes Multicast group address

o Suggest a default broadcast address

o ALM-Middleware guarantees all-node reachability by, e.g., automatic pre-JOIN

Issue #3: IP Mobility

o IP Mobility results in socket invalidation

o Hide IP change on application layer w/o Mobile IP

- Choose IP independent namespace

- Decouple overlay ID from IP address

o Reinitialize overlay for routing tables and proximity

- API call needed, see P2PP

Issue #4: Clarify the term "Group Member Management" in ALM

o Native Multicast

- Receiver join any source or source specific group

- Avoids delivering of multicast packets without receivers

o Meaning of group member management in ALM?

- Routing maintenance

o Create tree, heartbeat, …

- Should the middleware provide information about group members to application (s. SAMTK)?

o One can use DHT to save information

Conclusion

o SAM API definition should

- Be independent of specific P2P protocol

- Preserve original addresses

- Support different namespaces as predefined by API

- Offer broadcast handling (address selection ...)

o Clarification of terms needed: Using P2PP glossary?