

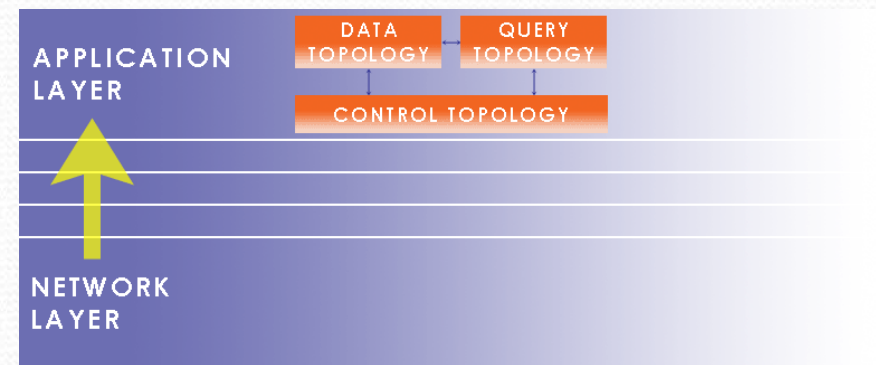
# A Proposed Novel Thematic-Based Approach to the Group Communication

- Raymond Pardede
- Csaba Horváth
- Dr. Gábor Hosszú
- Prof. Dr. Ferenc Kovács

Department of Electron Devices  
Budapest University of Technology and Economics

# Project Overview

- Group Collaboration on the Application Layer
- Multi-layer structure with different layer-topologies:
  - Control Topology
  - Data Transfer Topology
  - Query Topology for searching
- Multi-protocol structure to obtain optimized operation in each topology based on the executed testing



- **Introduction**
- Application Layer Multicast (ALM)
- Thematic-based Multicast Concept (TMC)
- TMC Control Topology
- TMC Data Topology
- TMC Query Topology
- ThemaCAST Project

# Objectives

**Application-Layer  
MULTICAST**

**GROUP COMMUNICATION SYSTEM**

**ROBUST**

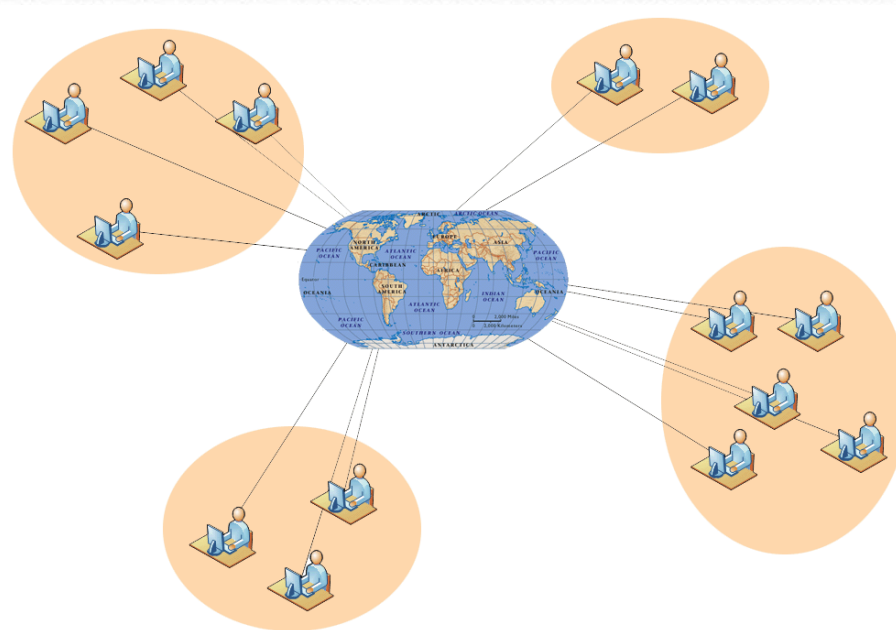
**SCALABLE**

**EFFICIENT**

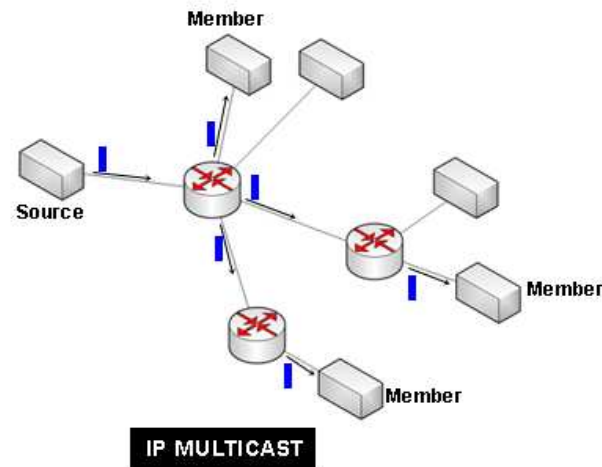
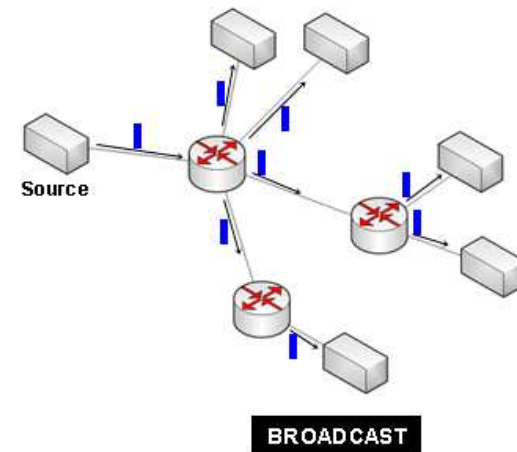
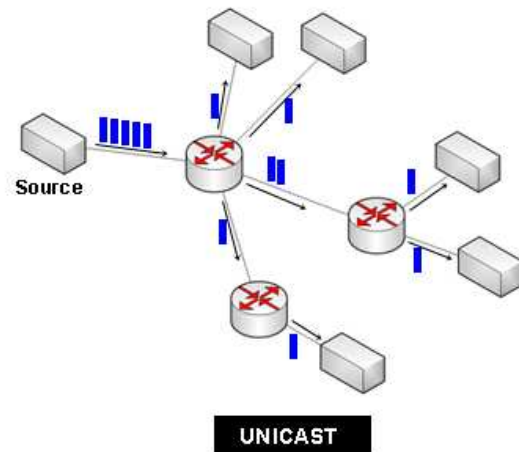
# Project Scopes

Designing a system to solve specific scenario:

- there are many interested **hosts in certain institutes**, and
- those institutes are relatively **far from each other**



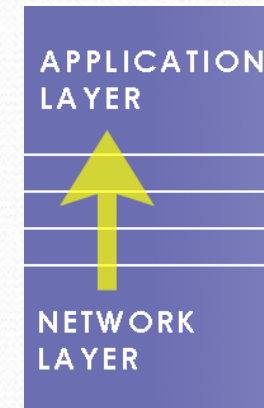
# Overview: Unicast, Broadcast, & IP Multicast



# IP Multicast: Issues & Proposed Solution

- **Issues:** Upgrading infrastructures for deployment purpose
  - Costly
  - Time Consuming

- **Proposed Solution:**
  - Moving the multicast functionality
    - from *network layer* to *application layer*
  - namely *Application-Layer Multicast (ALM)*

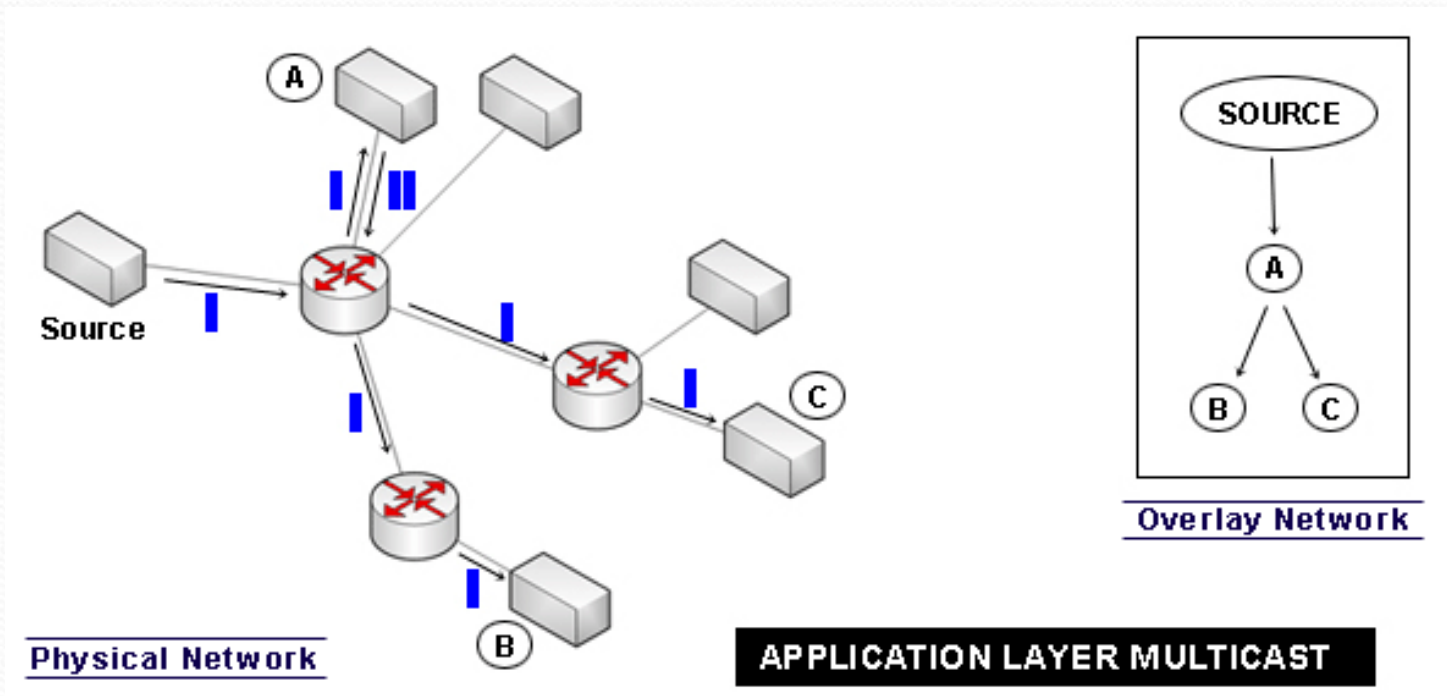


- Introduction
- **Application Layer Multicast (ALM)**
- Thematic-based Multicast Concept (TMC)
- TMC Control Topology
- TMC Data Topology
- TMC Query Topology
- ThemaCAST Project

# Advantages of ALM

- Easy to deploy
- Does not need router support
- Has infinite multicast address
  - It does not need IP multicast (Class D) address

# ALM Example



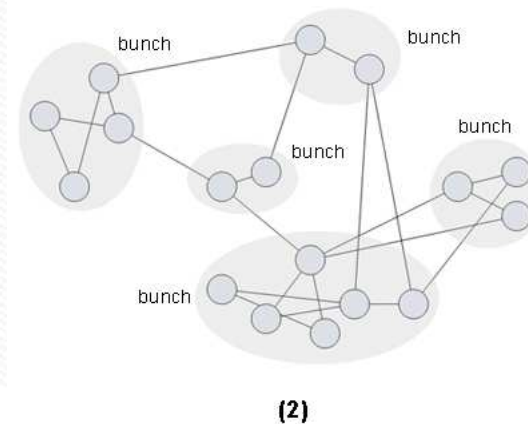
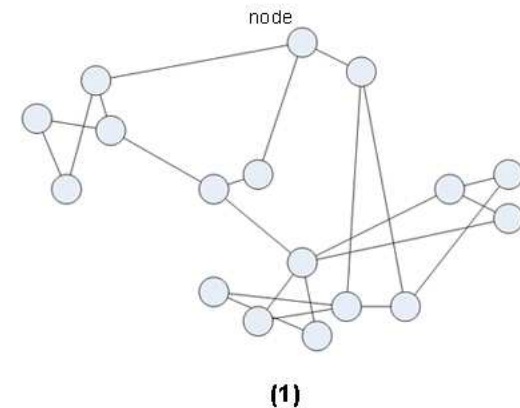
# ALM Design Challenges

- Obtain Network Efficiency
- Achieve Scalable System
- Attain Robust Structure

- Introduction
- Application Layer Multicast (ALM)
- **Thematic-based Multicast Concept (TMC)**
- TMC Control Topology
- TMC Data Topology
- TMC Query
- ThemaCAST Project

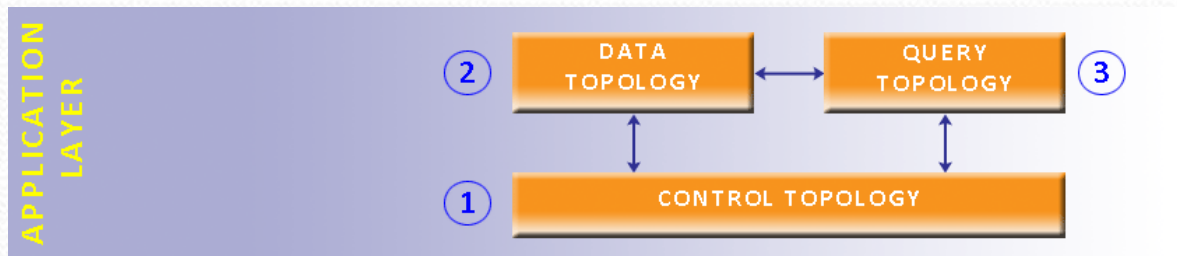
# Thematic-based Multicast Concept (TMC)

- TMC is based on assumption that:
  - Nodes' distributions are not uniform
  - There is some kind of clustering in the nodes' placement
  - The groups of nodes (**bunch**) are relatively far from each other



# TMC Overlays

- Three topologies for obtaining robust system:
  - Control Topology
  - Data Topology
  - Query Topology



# TMC Overlays (2)

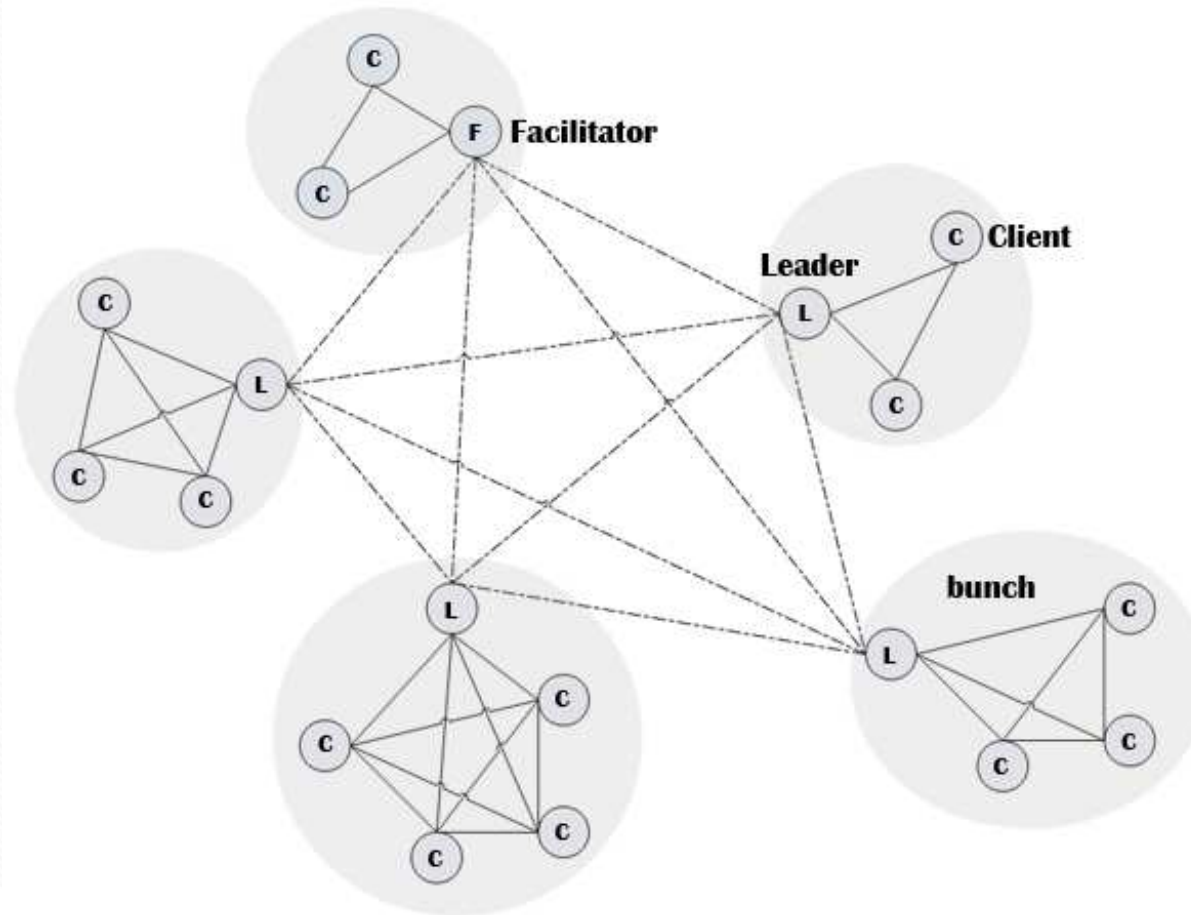
- **Control Topology**
  - Foundation of the overall layers
  - Provides such information to the upper layers:
    - node's neighbors
    - new member
    - link failures, etc.
- **Data Topology**
  - Provides bandwidth optimization in sending data packets
- **Query Topology**
  - Provides services in searching files

- Introduction
- Application Layer Multicast (ALM)
- Thematic-based Multicast Concept (TMC)
- **TMC Control Topology**
- TMC Data Topology
- TMC Query Topology
- ThemaCAST Project

# Basic Structure: Components

- **Client:** Member of the communication group
- **Facilitator**
  - Root of the Communication Group
  - One Facilitator in Each Communication Group
- **Bunch**
  - Group of clients that relatively close each other
  - Full Graph connection among clients
- **Bunch Leader**
  - Client in the bunch selected as the leader

# Basic Structure: Sample Figure

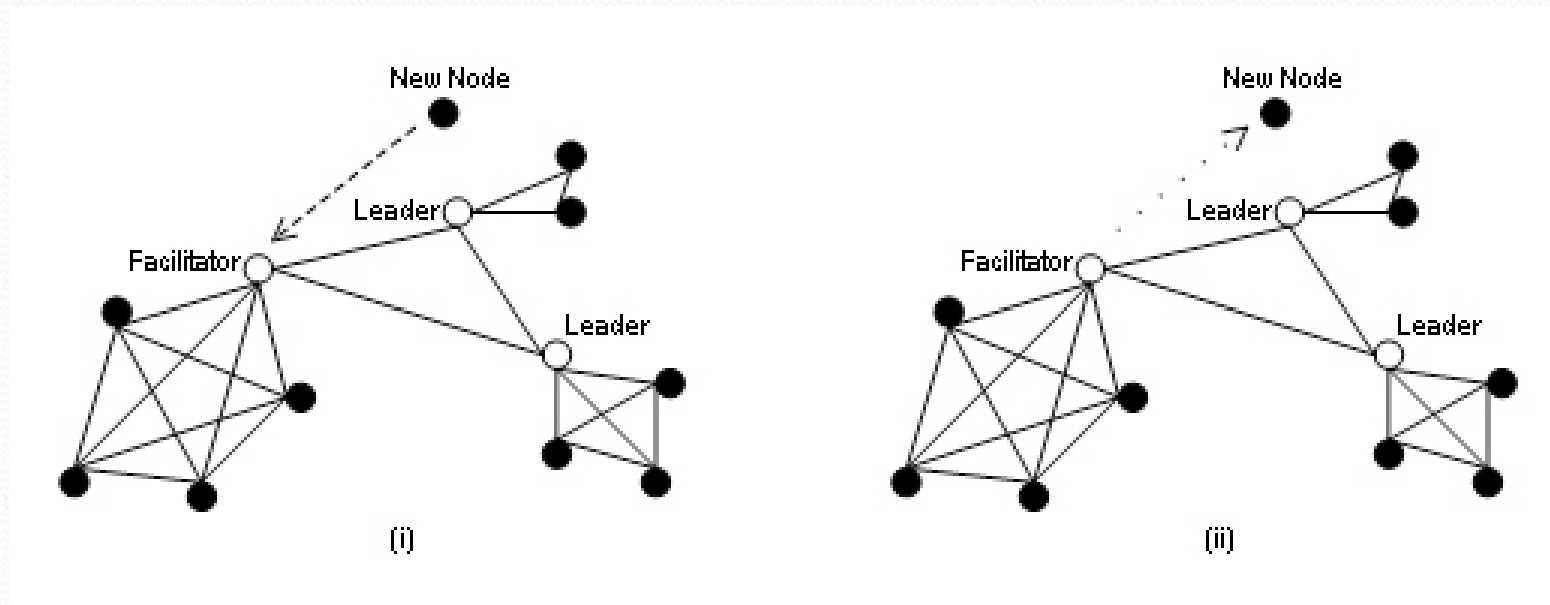


F: Group Facilitator  
L: Bunch Leader  
C: Client

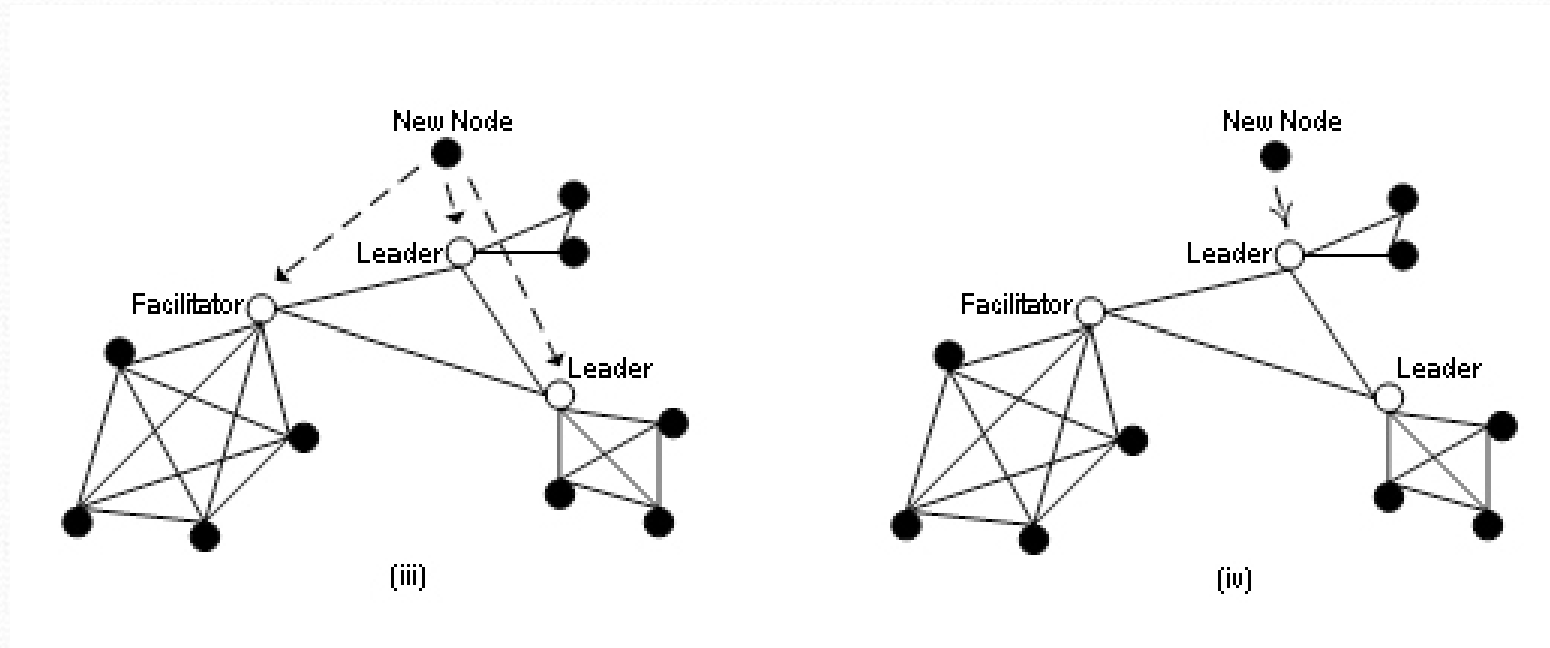
# Initiating TMC Group

- Initiated by one client (node)
- Consequently, the originator client:
  - is assigned as group Facilitator
  - is selected as bunch Leader
- In this stage, a new client registers to the originator client to join the TMC Group

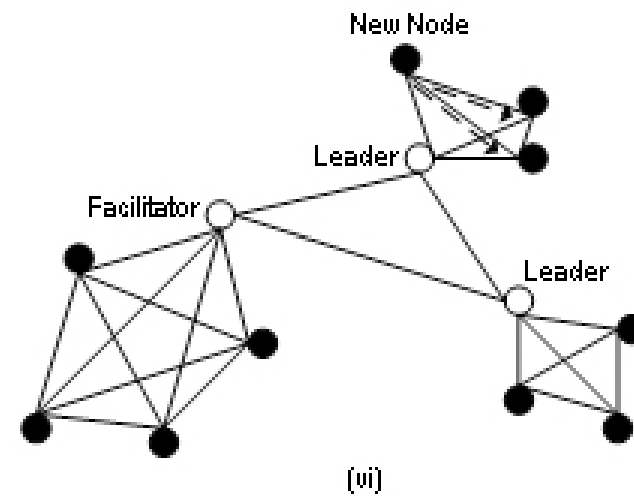
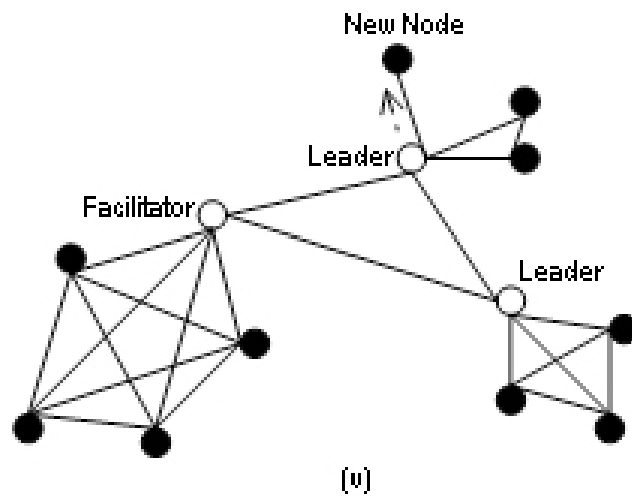
# Joining the TMC Group (1)



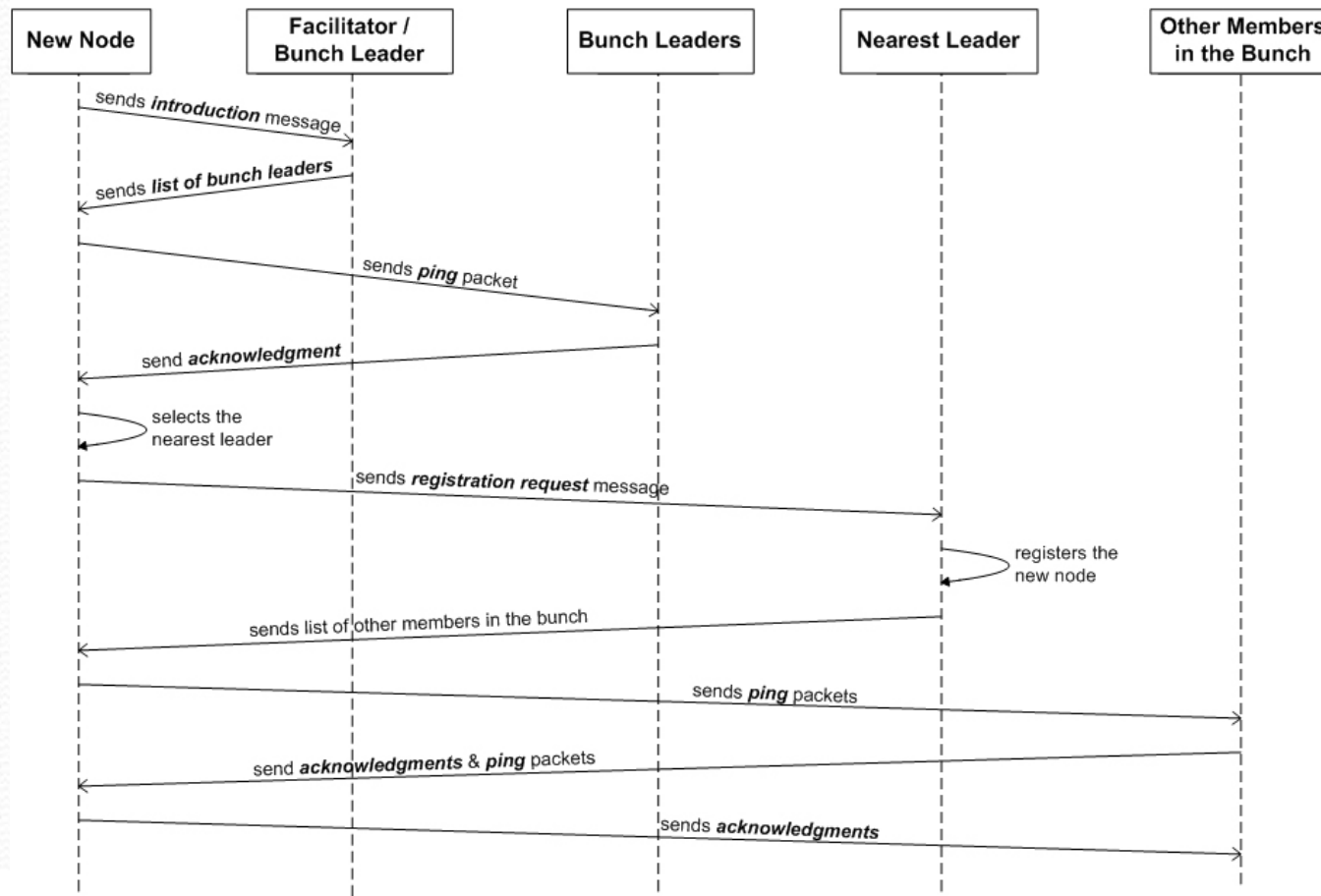
# Joining the TMC Group (2)



# Joining the TMC Group (3)



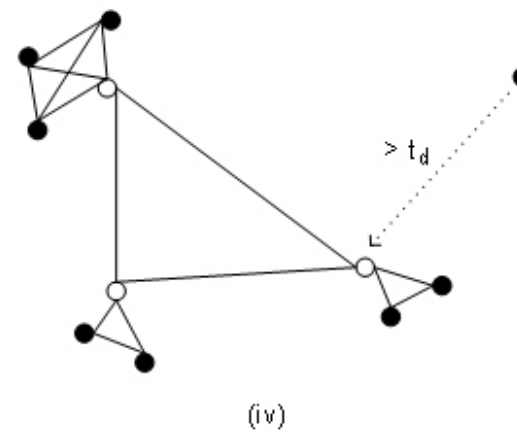
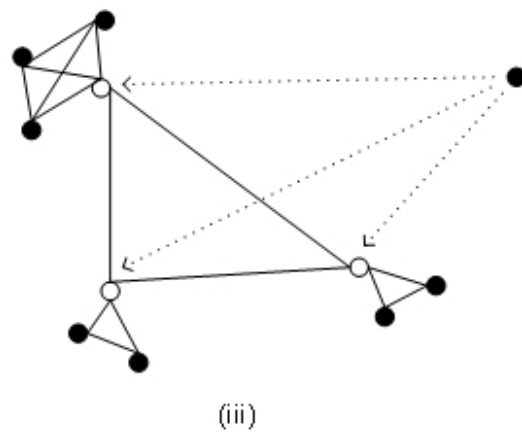
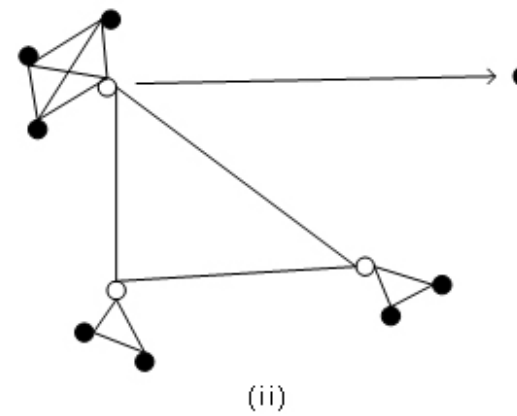
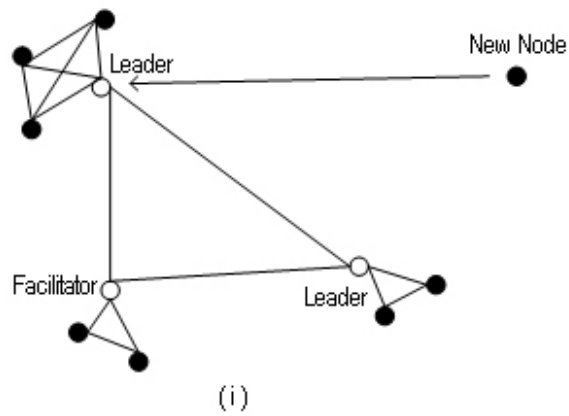
# Joining the TMC Group (4)



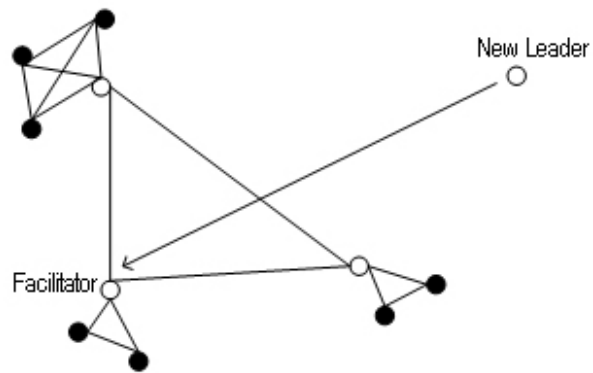
# New Bunch Creation (1)

- Bunch (Technical Detail Definition):
  - consists of clients located not further than  $t_d$  time from the bunch leader
- When joining the TMC group:
  - If the closest bunch leader is not further than  $t_d$  time, **join the bunch**
  - If the closest bunch leader is further than  $t_d$  time, **create a new bunch**

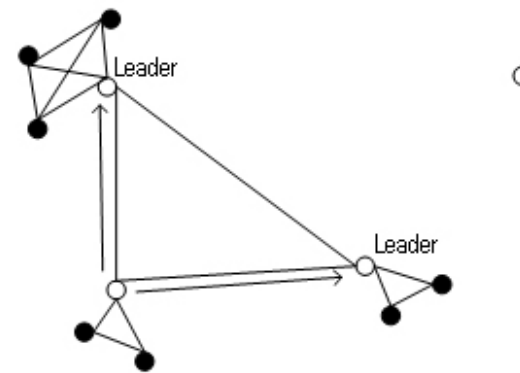
# New Bunch Creation (2)



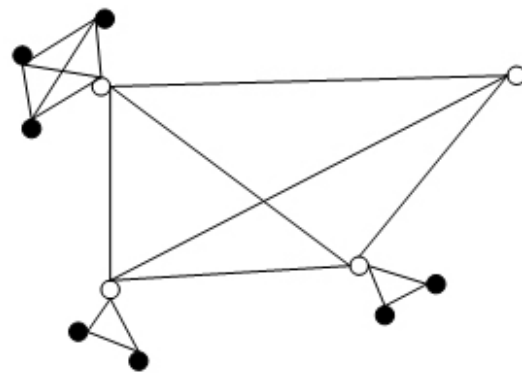
# New Bunch Creation (3)



(v)



(vi)



(vii)

# Bunch Leader

Bunch Leader tasks:

- Serving the registration process
  - Joining the TMC Group
  - Joining the bunch
- Monitoring the state of all leaders in the group

Ideal bunch leader = the highest performance client in the bunch

# Client Performance Point

- Performance Value ( $P_v$ ):

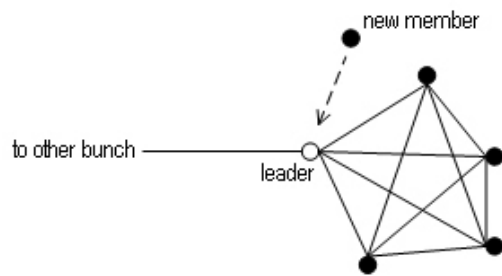
$$P_v = S \cdot N$$

- $S$  = Hardware Specification
- $N$  = Normalizing Coefficient

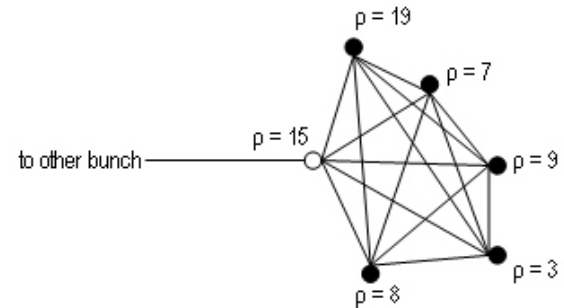
- Performance Point ( $\rho$ ):

$$\rho = \sum_{i=1}^n P_{vi}$$

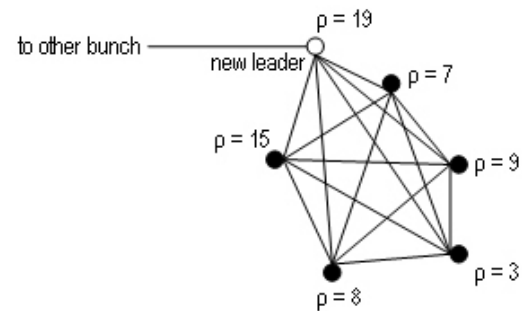
# Leader Selection Algorithm: Occasionally Executed



(a)



(b)

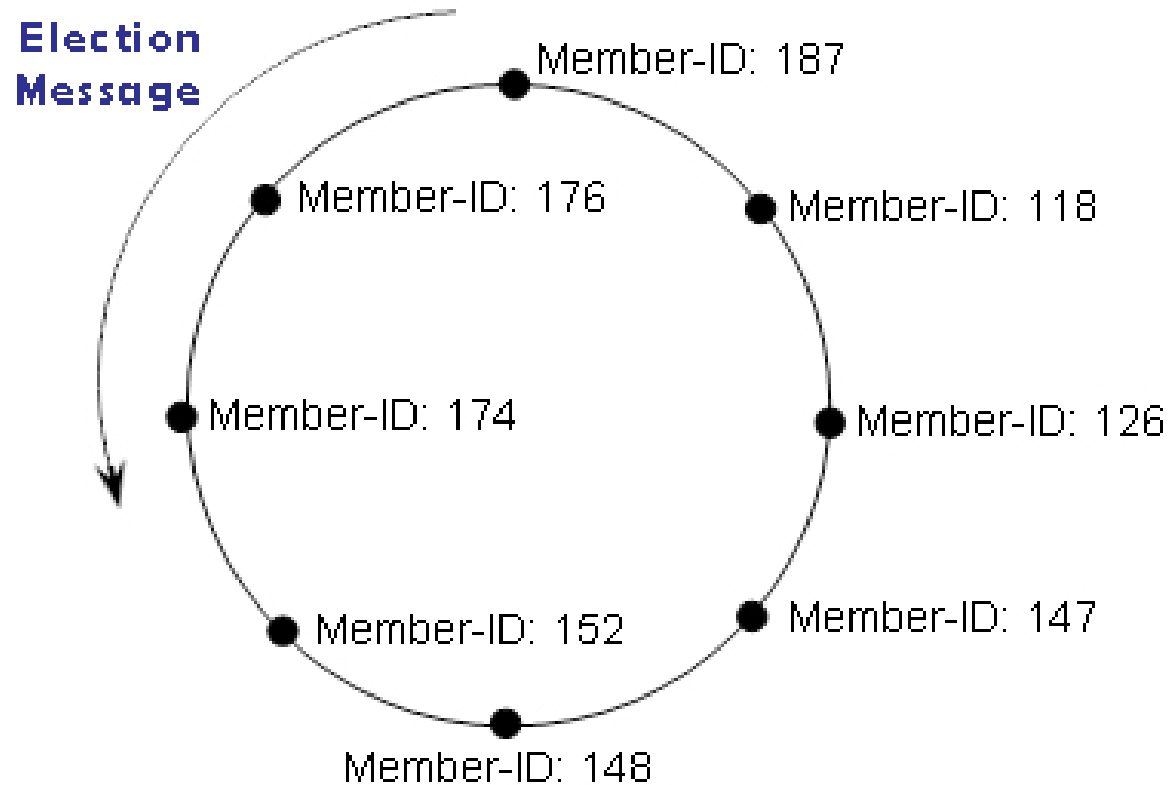


(c)

# Leader Selection Algorithm: Periodically Executed (1)

- Each client in the bunch has unique *Member-ID*
  - Using Hash Function
  - Converted from IP Address
- All bunch clients are plotted into a ring shape
- Each client sends *election message* to the left neighbour
  - Contains *Member-ID* & *Performance Point* ( $\rho$ )
- When node receives election message:
  - Compare the performance point
- Node receives its own message: declare as new leader

# Leader Selection Algorithm: Periodically Executed (2)



# Emergency Maintenance Procedure

- Executed when one bunch leader leaves the group
- Bunch leader sends *live message* periodically to clients
- After  $p$  time not receiving live message:
  - Client sends urgent message to any leader (random)
  - Leader registers the client as new member (temporarily)
- The client contacts “old neighbors”
  - Performing *Leader Selection Algorithm*
  - Leave the temporary bunch
  - Create a new bunch with “old neighbors”

# Emergency Maintenance Procedure

- Executed when one bunch leader leaves the group
- Bunch leader sends *live message* periodically to clients
- After  $p$  time not receiving live message:
  - Client sends urgent message to any leader (random)
  - Leader registers the client as new member (temporarily)
- The client contacts “old neighbors”
  - Performing *Leader Selection Algorithm*
  - Leave the temporary bunch
  - Create a new bunch with “old neighbors”

- Introduction
- Application Layer Multicast (ALM)
- Thematic-based Multicast Concept (TMC)
- TMC Control Topology
- **TMC Data Topology**
- TMC Query Topology
- ThemaCAST Project

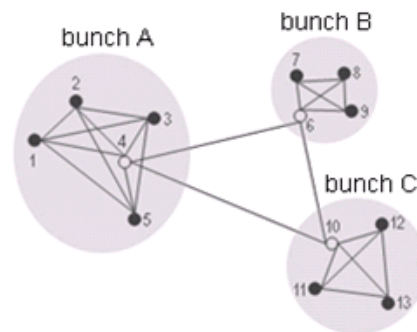
# Basic Structures & Protocols (1)

- Small Population Phase
  - Intra Bunch
    - Member inside the bunch is less than  $m$
    - Data is simply broadcasted inside the bunch
    - Hierarchical Flooding
  - Inter Bunch
    - Number of bunch in the group is less than  $b$
    - Data is simply broadcasted to the bunch leaders in the group

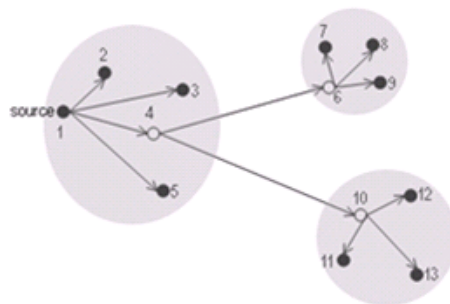
# Basic Structures & Protocols (2)

- Large Population Phase
  - Intra Bunch
    - Member inside the bunch is larger than  $m$
    - Using *Route Root Leader* (R2L) algorithm
  - Inter Bunch
    - Number of bunch in the group is larger than  $b$
    - Using *Dijkstra Algorithm*

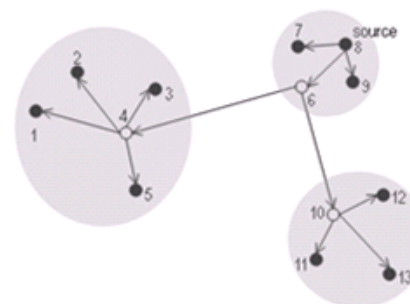
# Small Population



**CONTROL TOPOLOGY**



Example 1



Example 2

**DATA TOPOLOGY**

# Large Population – Intra Bunch

- Using *Route Root Leader (R2L)* Algorithm
- Bunch Leader is the root of the R2L tree
- In two connected clients:
  - *Parent Node*: closer to the root
  - *Child Node*: farther from the root
- Forming the R2L tree
  - The goal: to find the parent
  - each client contacts the closet neighbor
  - Compares the distance to the root

# Large Population – Inter Bunch

- Using *Dijkstra Algorithm*
- Similarly with the *Open Shortest Path First (OSPF)*
- Bunch Leaders use Control Topology information

- Introduction
- Application Layer Multicast (ALM)
- Thematic-based Multicast Concept (TMC)
- TMC Control Topology
- TMC Data Topology
- **TMC Query Topology**
- ThemaCAST Project

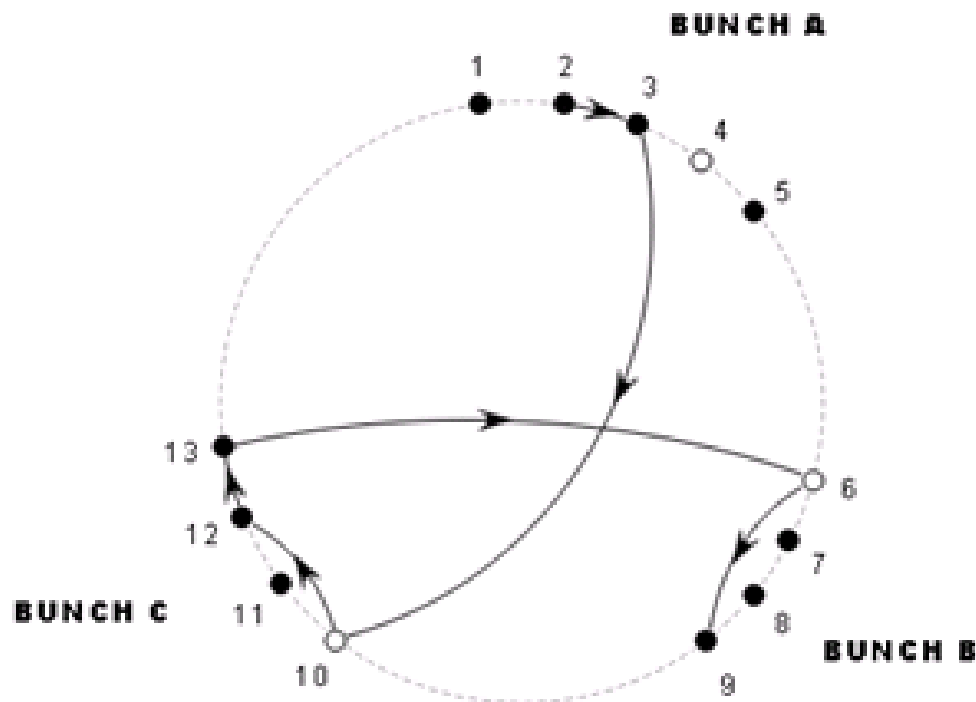
# Basic Concept

- This layer is used for searching purpose
  - File searching (P2P File Sharing)
  - Topic searching (Discussion Forum or Video Conference)
- Using modified *Random-Walk* algorithm
  - Implementing *Small-World Network* concept

# Modified Random-Walk (1)

- Sending Query Packet
  - Set the time to live (*TTL*) field by integer value  $T > 0$
  - Initially sent to the neighbor
- Forwarding Query Packet
  - Drop the Query Packet if  $T = 0$
  - Forward the Query Packet if  $T > 0$ 
    - Forward to neighbor in the bunch with probability:  $p$
    - Forward to other bunch with probability:  $1 - p$

# Modified Random-Walk (2)

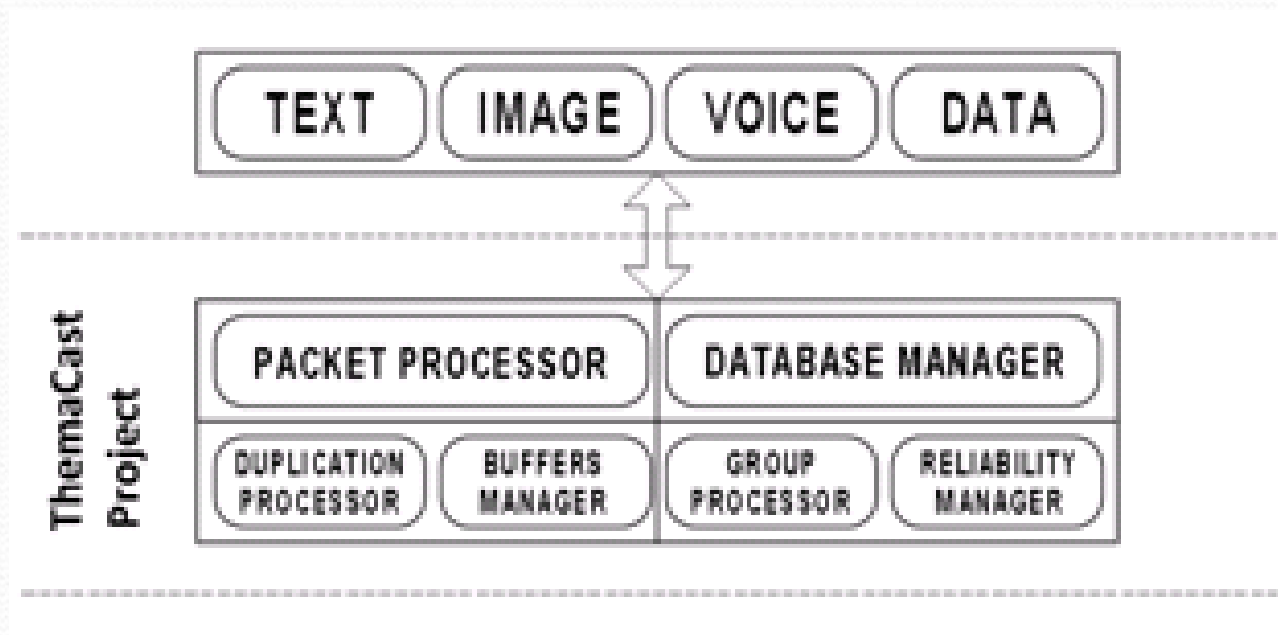


- Introduction
- Application Layer Multicast (ALM)
- Thematic-based Multicast Concept (TMC)
- TMC Control Topology
- TMC Data Topology
- TMC Query Topology
- **ThemaCAST Project**

# ThemaCAST Project

- Project Goal: create reusable class component
- Any multimedia application developed by other parties can use the ThemaCAST services
- ThemaCAST component structure
  - Packet Processor
    - Duplication Processor
    - Buffers Manager
  - Database Manager
    - Group Processor
    - Reliability Manager

# ThemaCAST Architecture



# Conclusion

- TMC presents a novel *Application-Layer Multicast (ALM)* system
- TMC intends to develop a scalable, robust, and efficient group communication system
- The developed method has partly been implemented in the software ThemaCAST
- The main tasks of the future development is completing the implementation issues

# End of Presentation