A Federated Peer-to-Peer Network Game Architecture

Sean Rooney, Daniel Bauer, Rudy Deydier
IBM Zurich Research Laboratory

Presented by Chi-Hong Chao

Outline

- Introduction
- Solutions
- Overview of the P2P Solution
- Architecture
- Performance Evaluation
- Conclusion

Introduction

- Gaming is a relatively neglected topic for communications research
- It also presents distinct challenges to network designers, particularly in respect to its sensitivity to latency and loss.
- A massive multiplayer online game is being concerned.
- Currently, commercial large games use a central server.

Introduction

- Problems with central server
  - The game provider can't know how popular a game will be.
    - Server farm is too big
      - Waste money, resource
    - Server farm is too small
      - Lose money, make players unhappy
  - Bottleneck of central server
    - CPU, Bandwidth, Storage capacity

Solutions

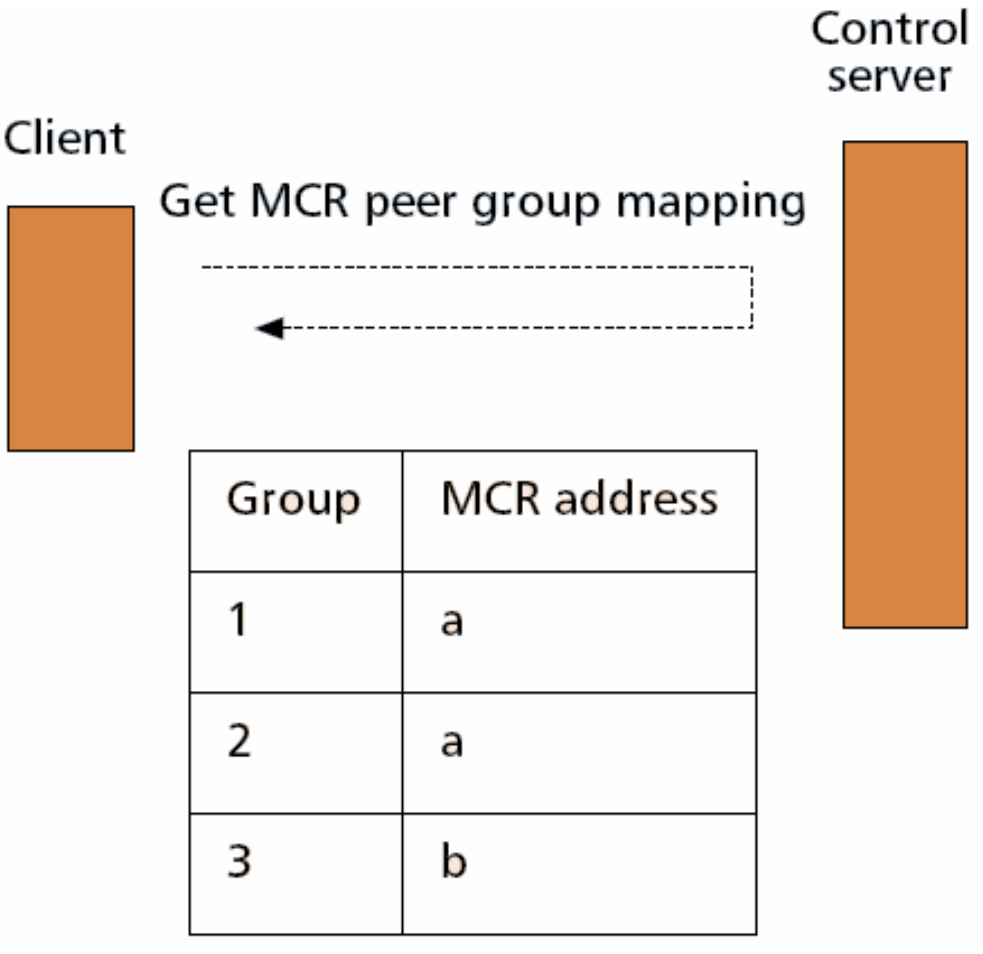
- Renting cycles, storage, and bandwidth from a third party that gains economies of scale by hosting multiple games simultaneously.
  - Problems:

    crosstalk between gameing applications running on the same shared infrastructure can have serious consequences
- A federated Peer-to-Peer Game Architecture.

Overview of the P2P Solution

- The game is divided into areas of interest (federations).
- Broadcast within a federation.
- Most of the logic specific to a given game is executed at the client.
- Control server are only for administration.
- MCR (MultiCast Reflector)
- Shaker (The transport protocol)

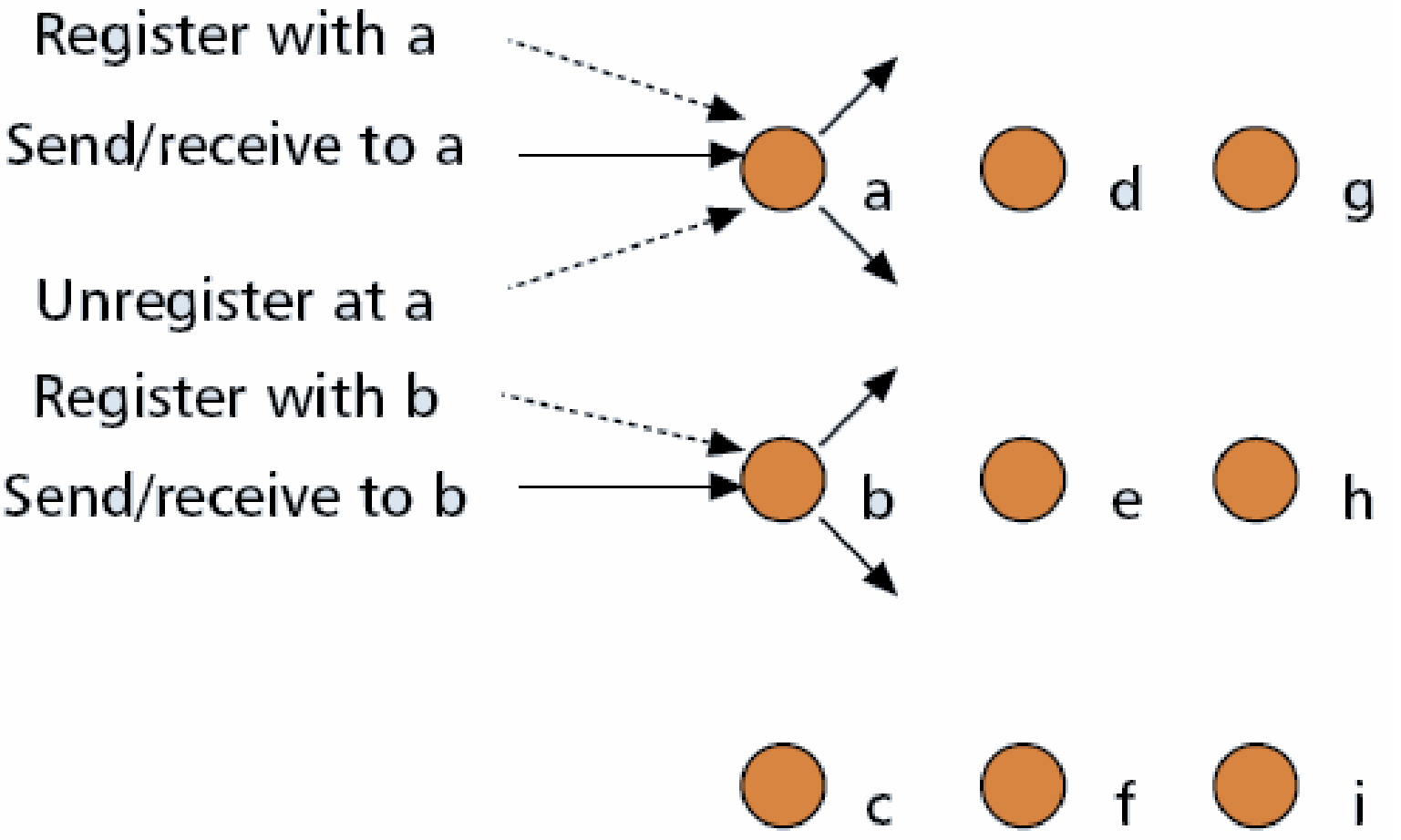
Architecture (Control Layer)



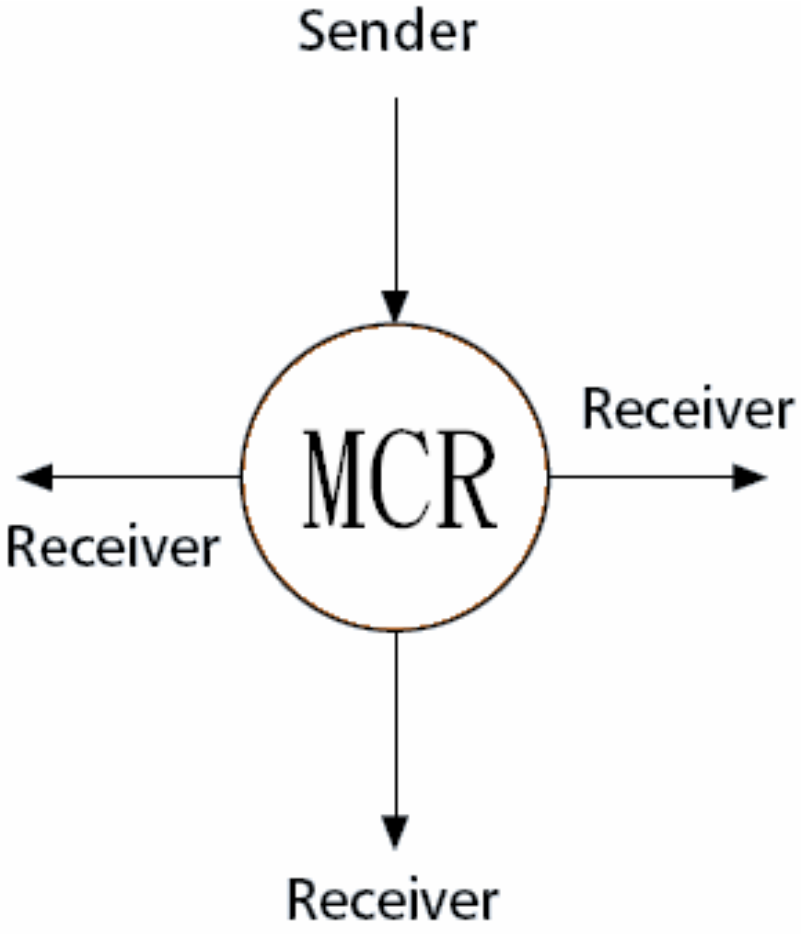
Group	MCR address
1	a
2	a
3	b

- While a peer joining the game, it first connects to the Control server and gets the information of MCR.

Architecture (Control Layer)



Architecture (Control Layer)

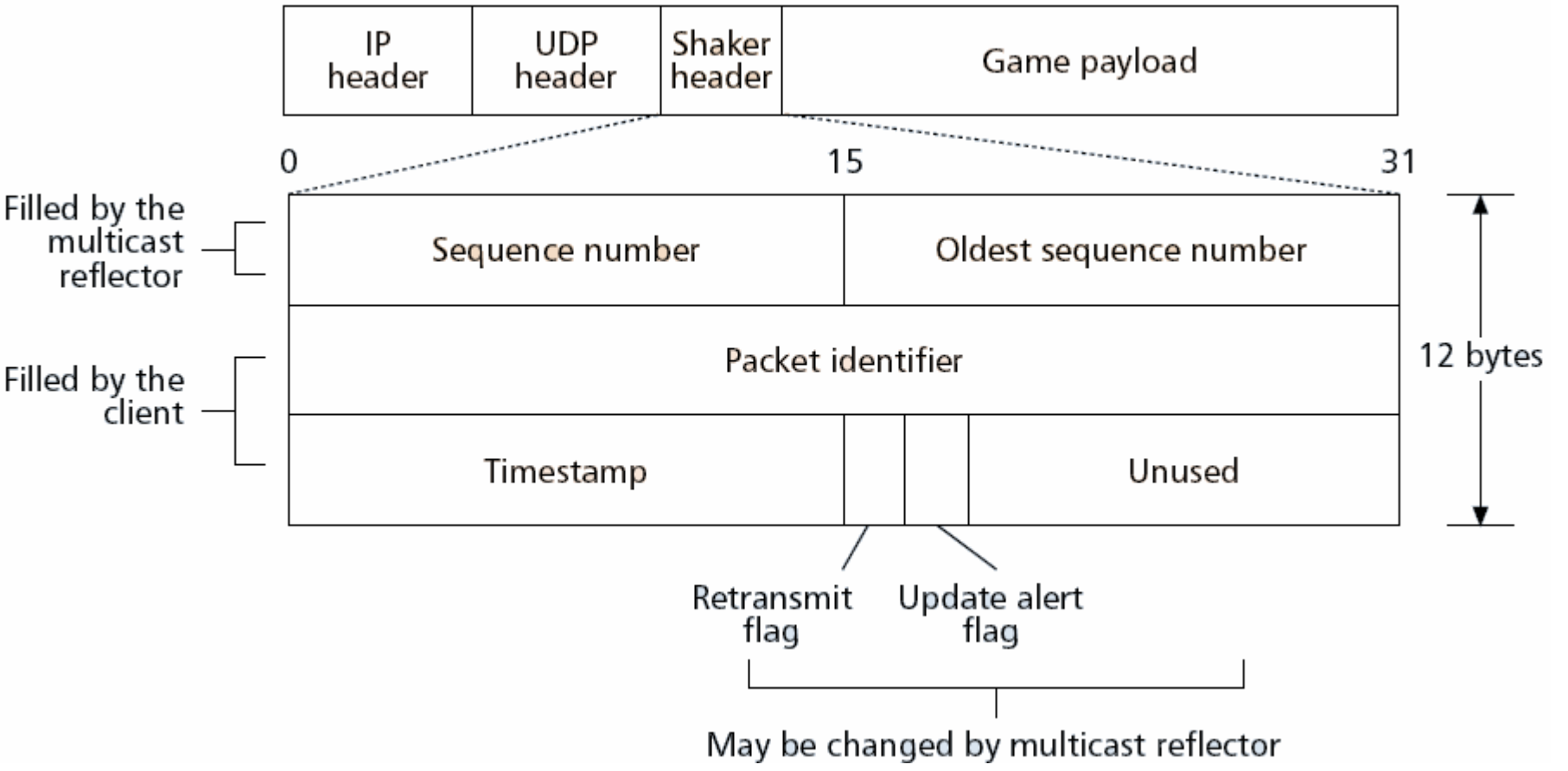


- Broadcast via MCR.
- The MCR is unaware of game logic.

Architecture (Control Layer)

- If belonging were binary, a player would either belong to a group or not, and consequently receive all or no information about that group.
- This would not be desirable as it would mean that if a player required some information in a group, they would receive all of it and then have to do the filtering themselves.
- The affinity values act like a filter in a publish/subscribe mechanism.

Architecture (Shaker Transport Protocol)

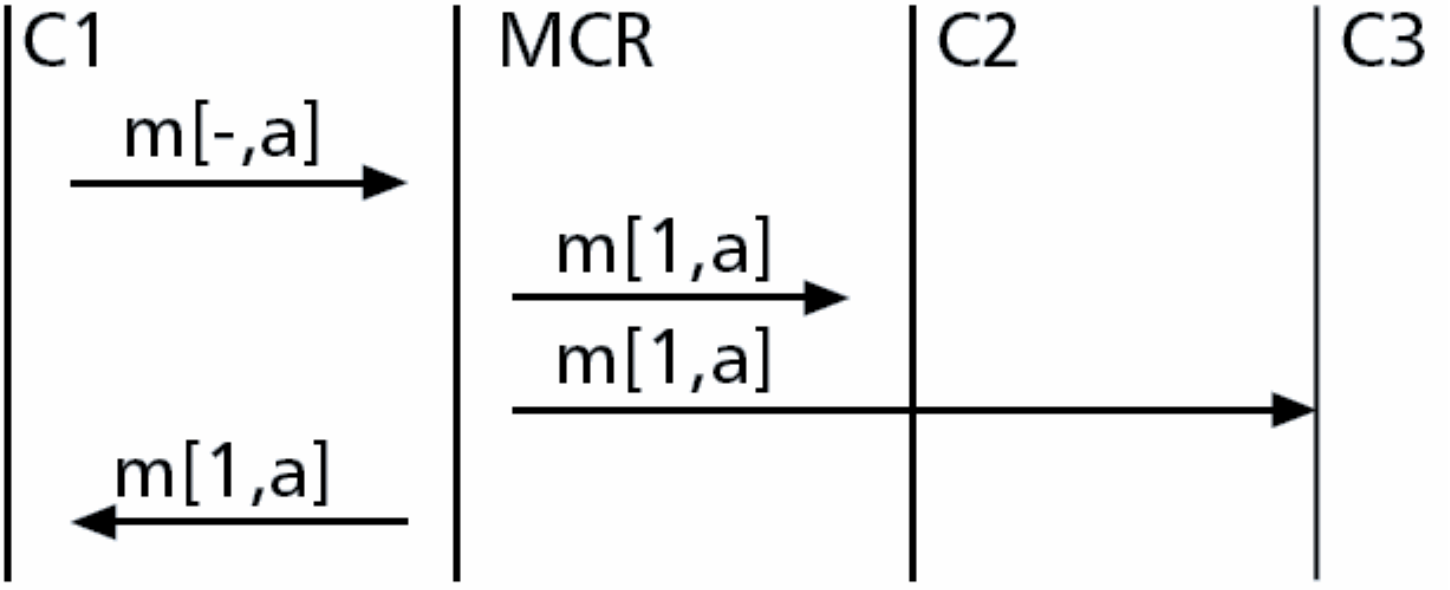


Architecture (Shaker Transport Protocol)

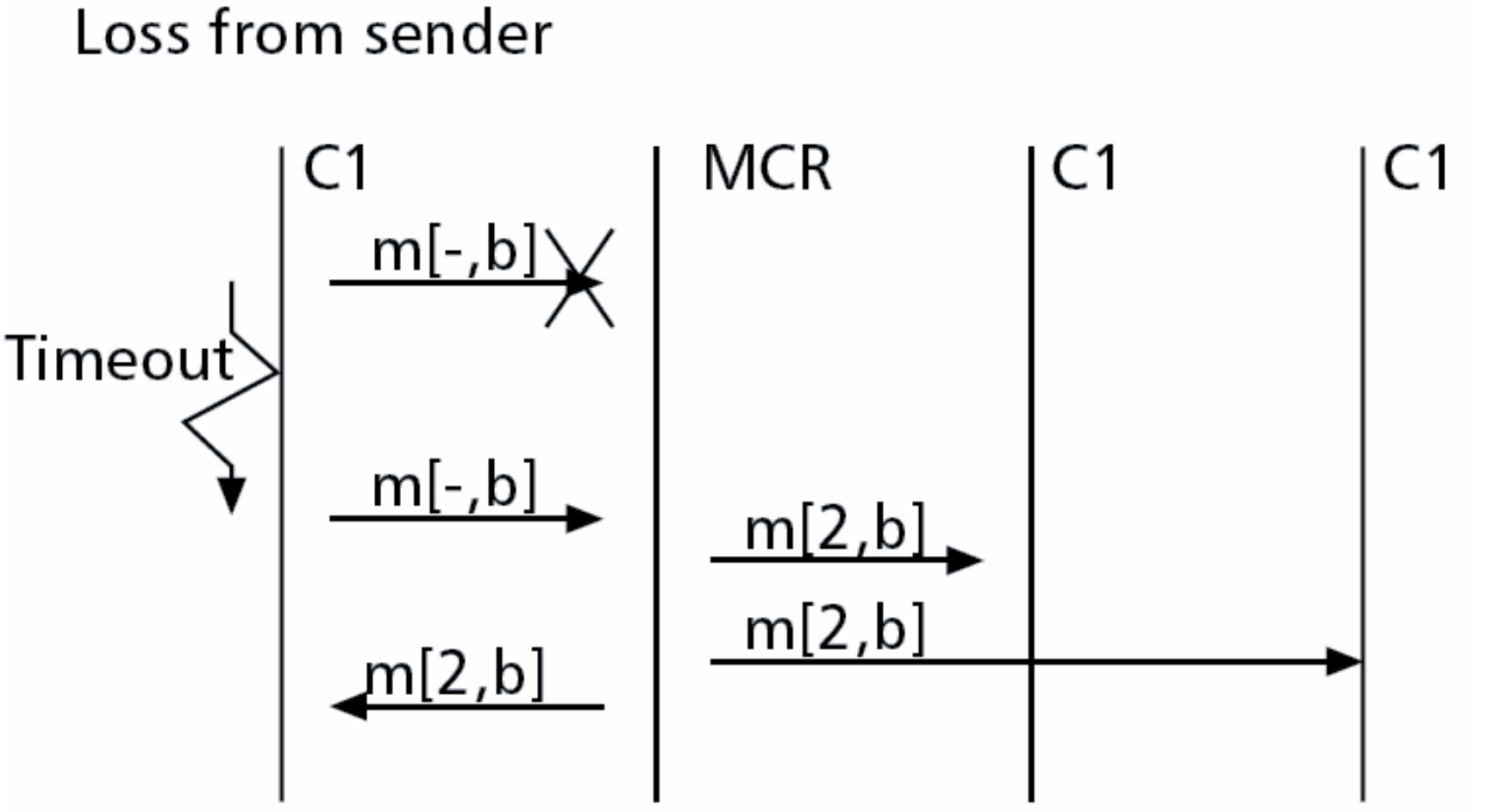
- When a Shaker packet arrives at the MCR the forwarding mechanism adds a packet Sequence Number, and the packet is stored in a buffer at the MCR.
- The MCR can only keep a finite number of already transmitted packets in memory; the Oldest Sequence Number is the threshold that the oldest packet can be retransmitted.
- The 32-bit Packet Identifier is divided into two parts. The top 16 bits identify the sender, while the bottom 16 bits are a monotonically increasing series.

Architecture (Shaker Transport Protocol)

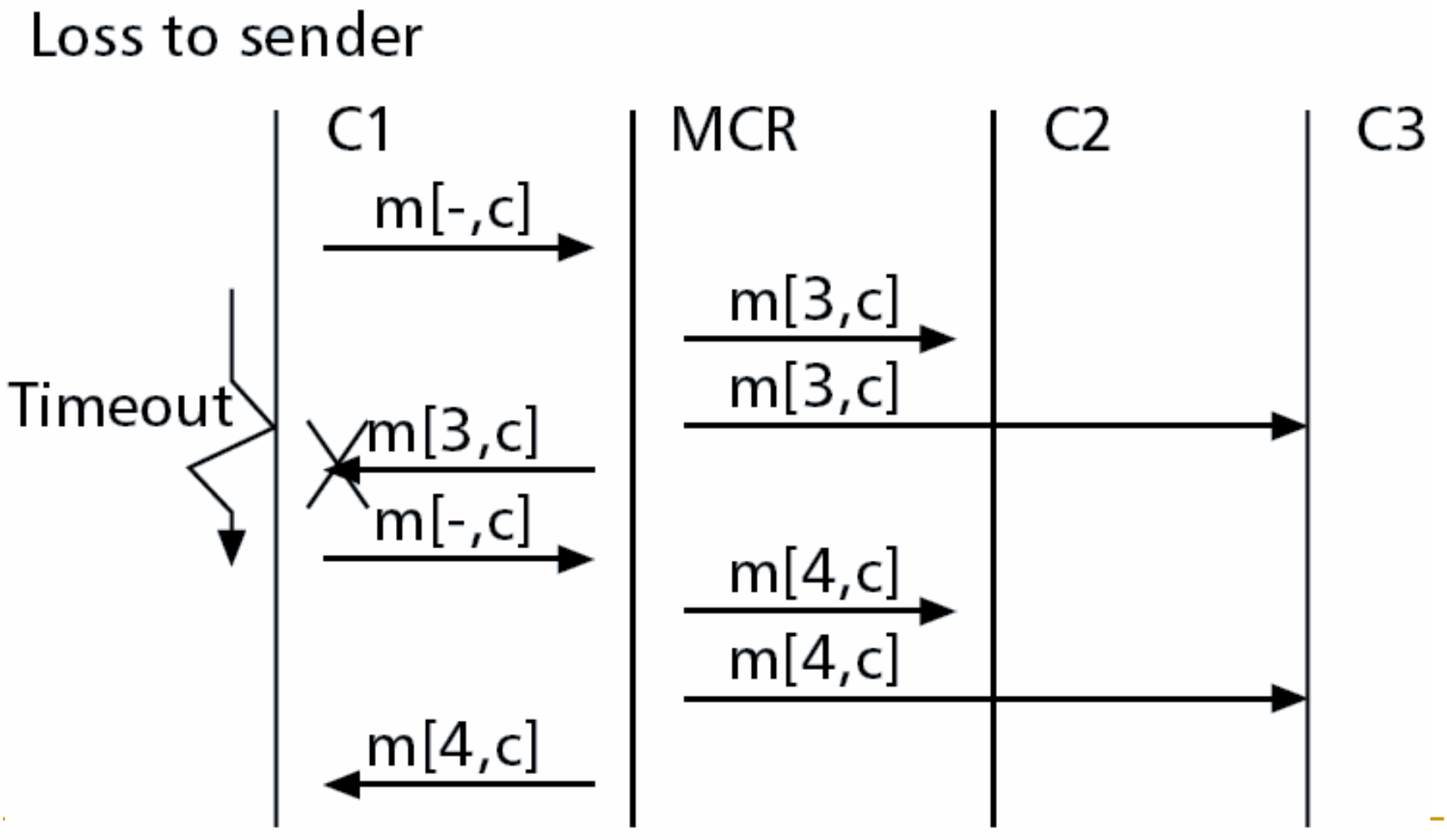
No loss



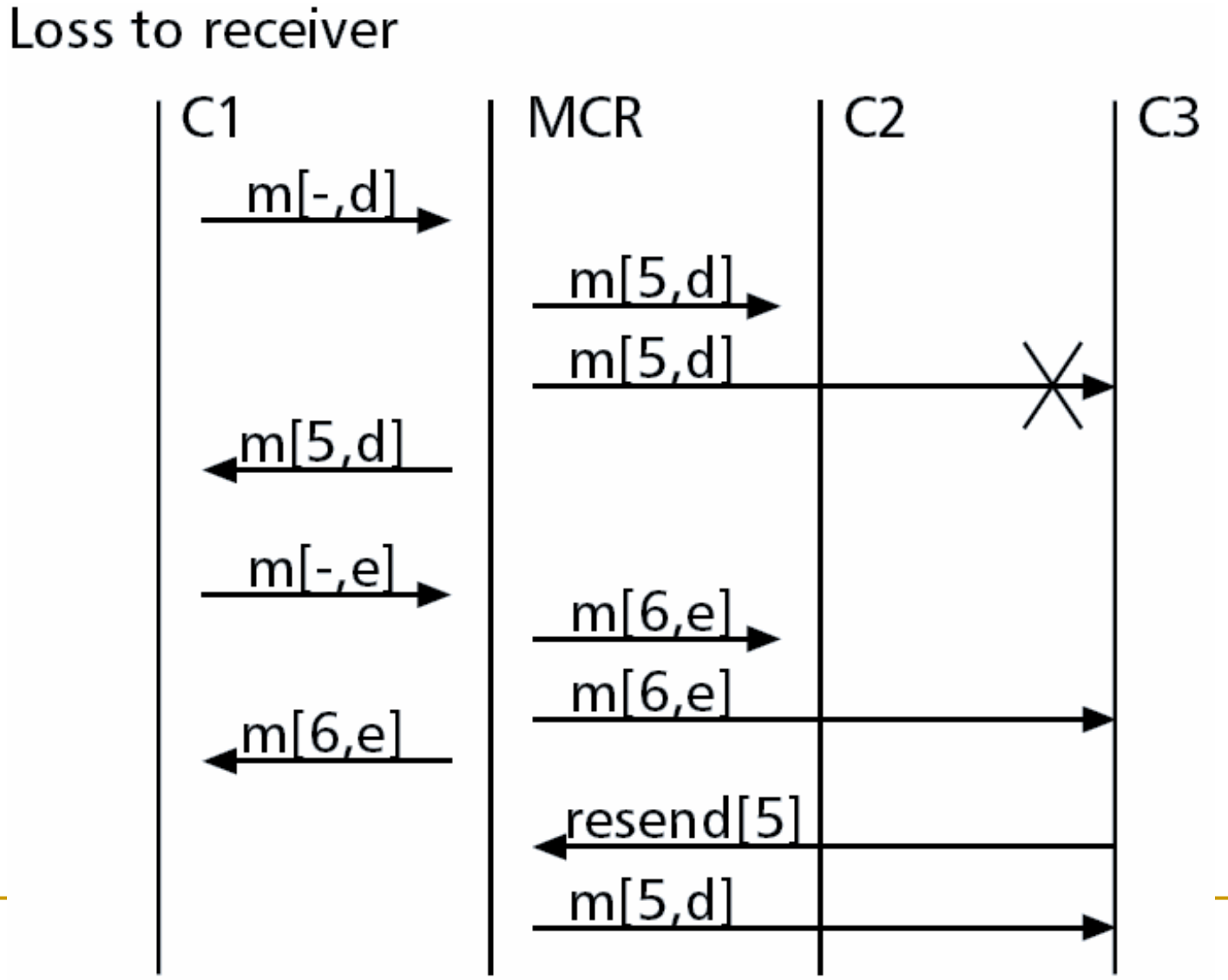
Architecture (Shaker Transport Protocol)



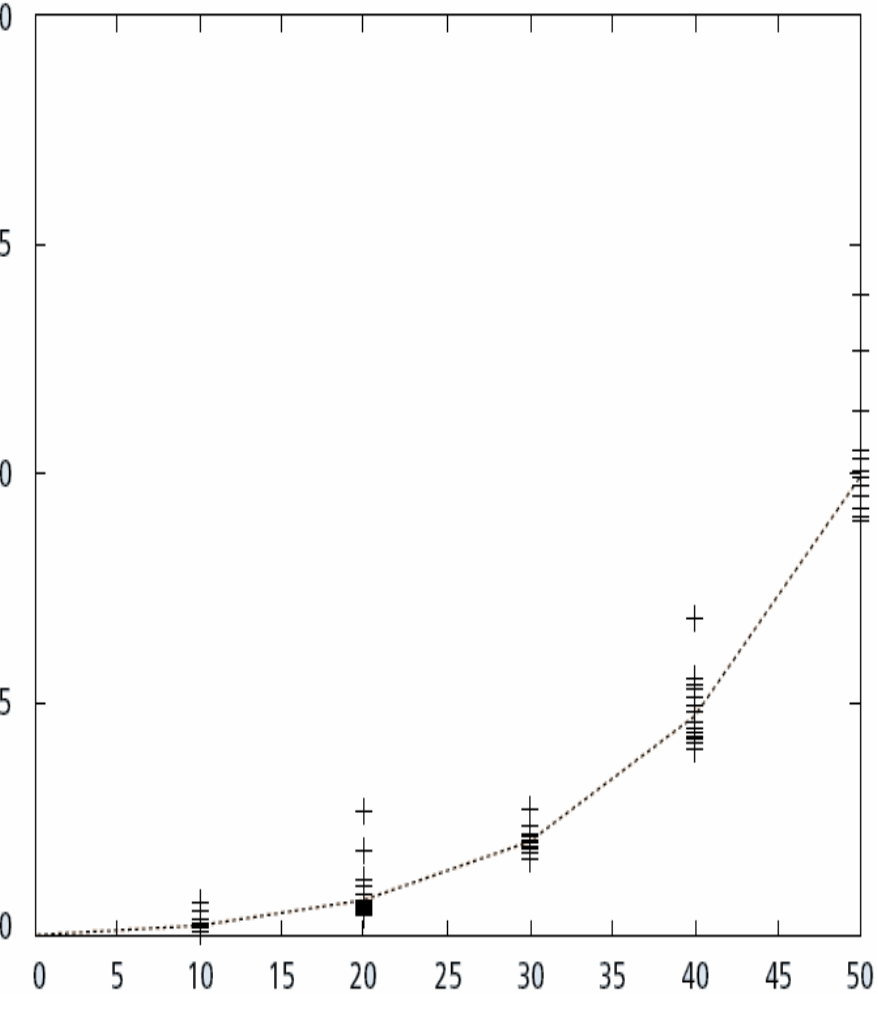
Architecture (Shaker Transport Protocol)

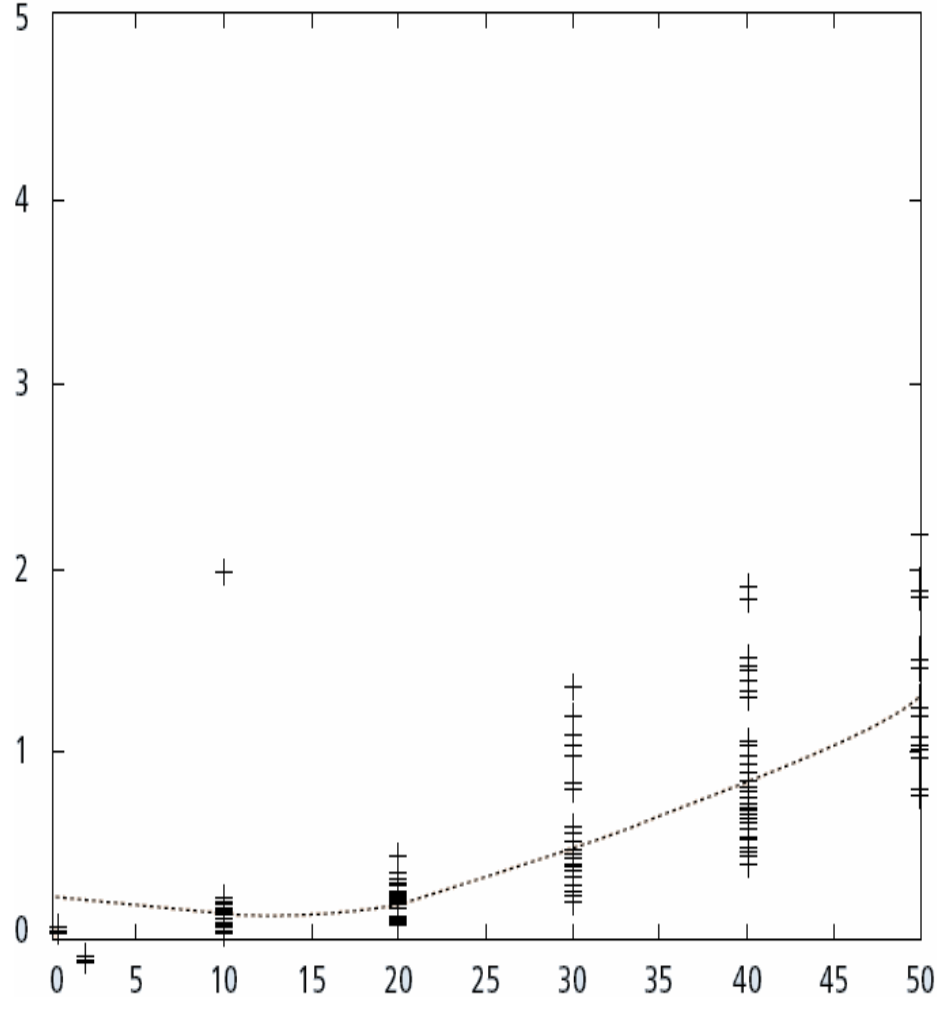


Architecture (Shaker Transport Protocol)



Performance Evaluation





Ratio of late packets

Conclusion & Discussion

- The paper proposed a very practical and effective solution for massive Multiplayer Online Games and the Shaker protocol improve the performance.
- But All the tests were done within a LAN instead of a WAN.
- Why and how this system is scalable compared with a central server system are not fully discussed or tested.