QuON

A Quad-Tree Based Overlay Protocol for Distributed Virtual Worlds

Stephan Krause & Helge Backhaus
Universität Karlsruhe - Institut für Telematik

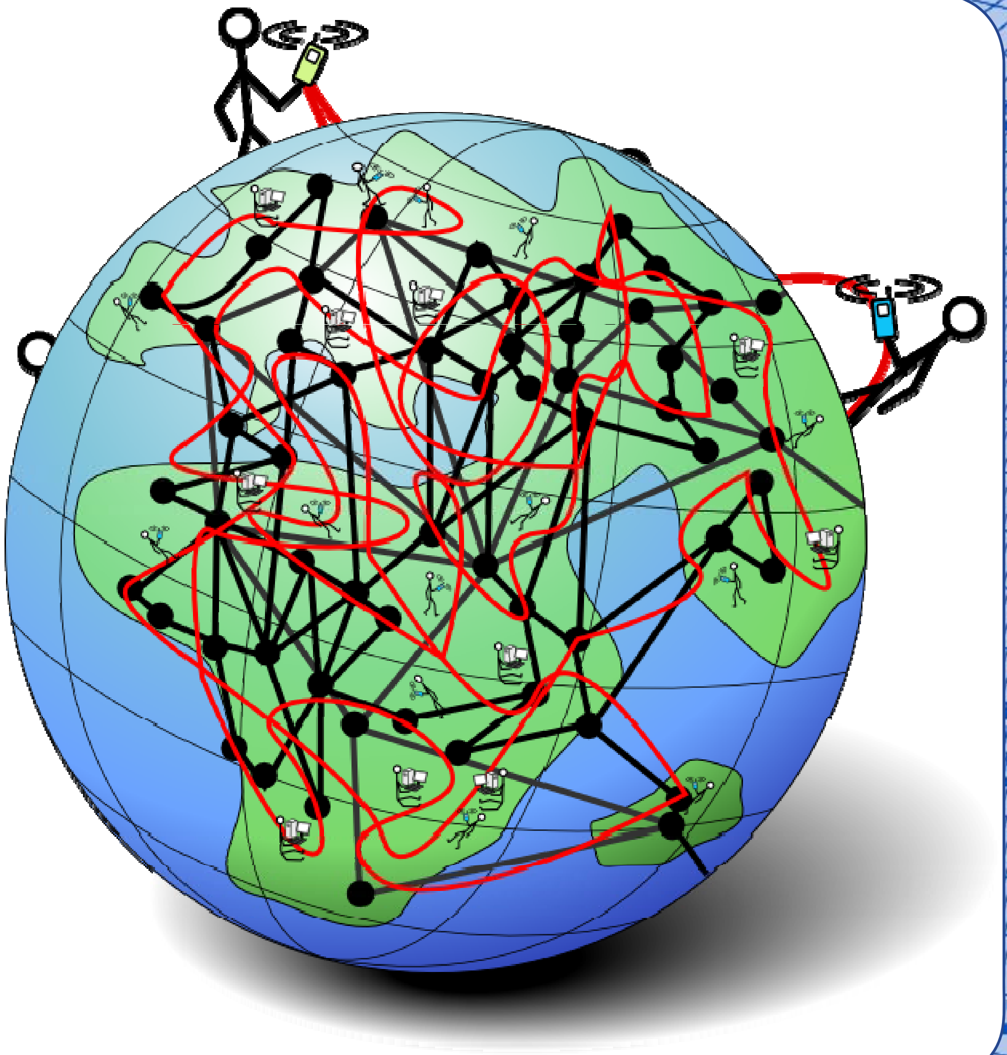
Motivation

Age of Conan
Hyborian Adventures
Warhammer Online
Age of Reckoning
World of Warcraft
Guild Wars
EVE Online
EverQuest II
Pirates of the Burning Sea
Dark Age of Camelot
The Lord of the Rings Online
Mines of Moria

What more could you ask for?

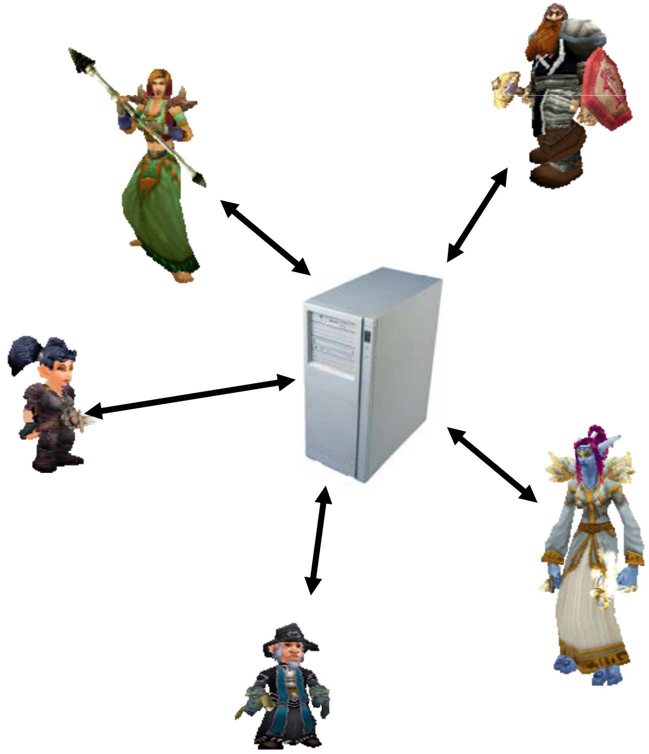
Our goal

- One shared world for all players
- An arbitrary number of players in the game-world



State of the art

- Almost all MMOGs and Virtual Worlds use central server farms
- How to make it scale?
  - Sharding: Multiple realms
    - World of Warcraft ~ 5,000 concurrent players/realm
  - Lots of proxy servers
    - EVE online ~50,000 concurrent players
- Additional drawback:
  - Infrastructure is expensive

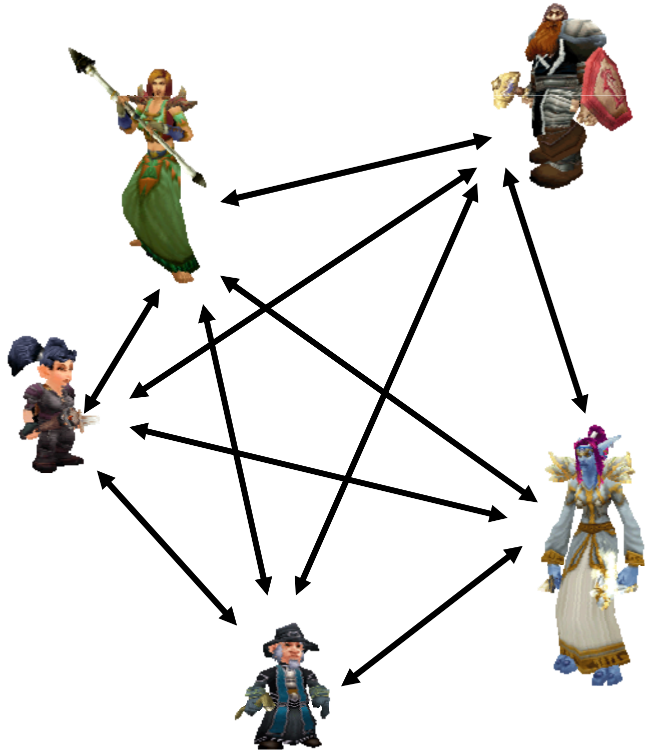


Possible solutions?

- Use P2P technologies:
  - Players communicate directly without central servers
- But how?
  - Fully meshed network does not work
  - Multicast does not help
    - O(n²) messages

→ „Interest Management“

- Deliver only “interesting” messages



Possible Solutions

- Simple idea: divide the map!
  - Messages will only be send to players in the current segment
- Chose some responsible node
  - Coordinates message dissemination
- To avoid overloading
  - Split the segment if it is too full
  - Use Application Layer Multicast
- Problems:
  - “Supernode” is single point of failure
    - Backup mechanisms are needed
  - Introduces additional delay

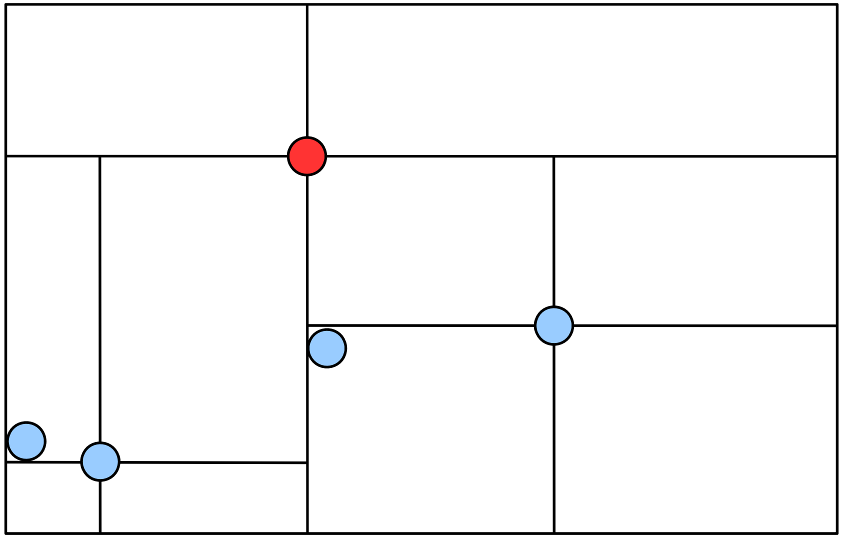
Mutual Notification

- Area of Interest (AoI)
  - Direct connection to all players inside AoI
  - Mutual notification of new neighbors
  - Problem: Direct neighbors not sufficient
    - Additional connections needed
- Advantages:
  - No bottlenecks or single points of failure
  - No arbitrary zones and zone borders
  - Optimal Delay (1 Hop)
- Example (related work): VAST
  - Uses a Voronoi diagram for classifying and discovering neighbors

QuON

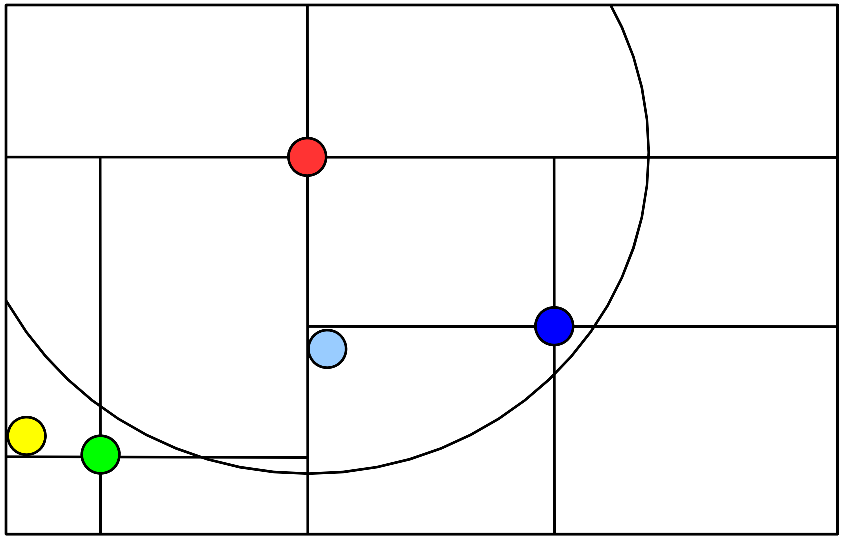
- „Quad-tree based Overlay Network“
- Basic Idea: Mutual Notification
  - Direct Connections to all Neighbors inside AoI
  - Mutual notification of new neighbors
- To guarantee connectedness
  - One „Binding Neighbor“ per quadrant
    - The closest neighbor in this quadrant
    - Is updated on every movement
    - Binding-neighbor information is exchanged periodically
      - This guarantees the closest possible binding neighbor will be found
    - „Soft-state Neighbors“ for symmetric neighbor relationships

Excursus: (Point-Region) Quad-Tree



- Tree structure developed for 2D coordinates
- Each vertex can have up to 4 children
- Advantage: simple
- Easy computation of nearest neighbors
- In QuON: no globally valid quad-tree!

Neighbor classification



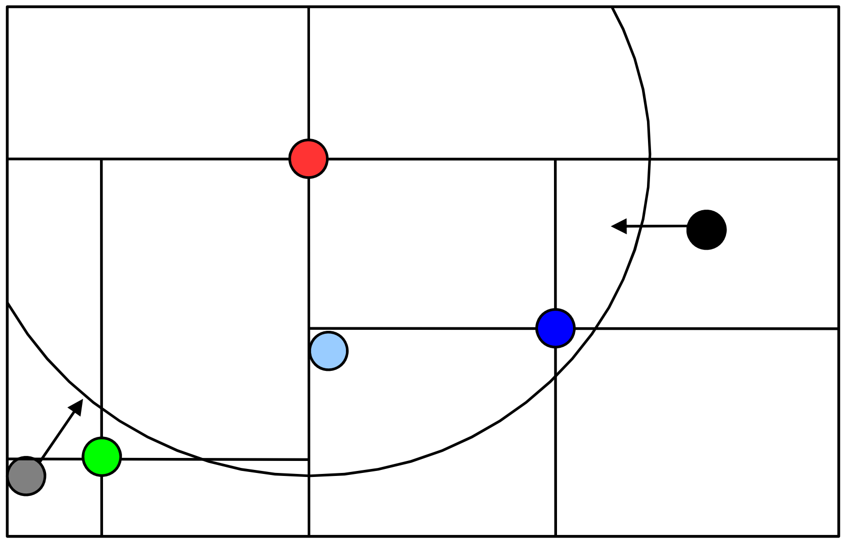
🔵 is direct neighbor of 🔴

🟢 🩵 are binding neighbor of 🔴

🔴 is binding neighbor of 🔵 🟡

🟡 is soft-state neighbor of 🔴

Finding new neighbors



- ⚫ is neighbor of 🔵, who informs 🔴
- ⚪ is binding neighbor of 🟢
- ⚪ becomes new binding neighbor of 🔴 before entering 🔴‘s AoI

Joining/Leaving the Overlay

- Join
  - Connect to arbitrary bootstrap node
  - Greedy forward until initial position is reached
    - Can be improved by position cache
- Leave
  - Graceful: Send leave notification to all neighbors
    - Contains a list with all neighbors
  - Ungraceful: Failure will be detected by lack of update messages
    - Direct neighbors simply discard failed player
    - Binding neighbors replace failed player by next candidate
    - In rare cases → Backup needed

Backup Mechanisms

- Failures
  - Keep last binding neighbor outside of AoI as backup neighbor
  - If binding neighbors fails
    - If still nodes inside AoI, one of them becomes new binding neighbor
    - Else if one soft-state neighbor is binding neighbor candidate, he will become new binding neighbor
    - Else connect to backup binding neighbor and get recursively forwarded to correct binding neighbor
- Latency
  - “AoI Buffer” when classifying neighbors
  - Size depends on average latency and maximum move speed

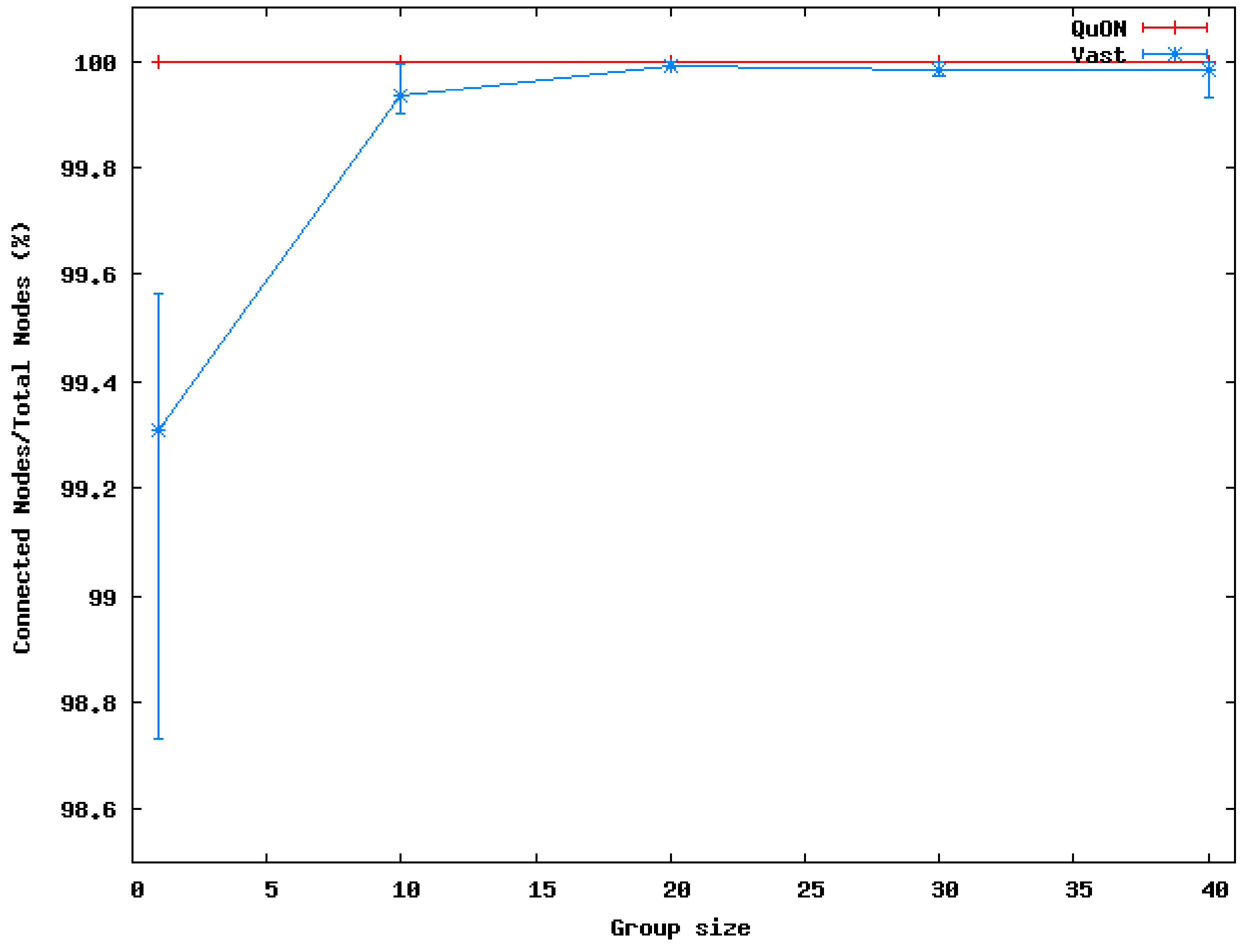
Simulation Settings

- OverSim as simulation framework
- “Simple Underlay” using latencies from CAIDA’s skitter project
- 500 nodes
- 2 simulated hours
- Heavy tailed session times (100 min average)
- Playground size 1,000m*1,000m
- Movement speed 5m/s, 6 updates/sec
- Group based random waypoint
  - Group sizes from 1 to 40

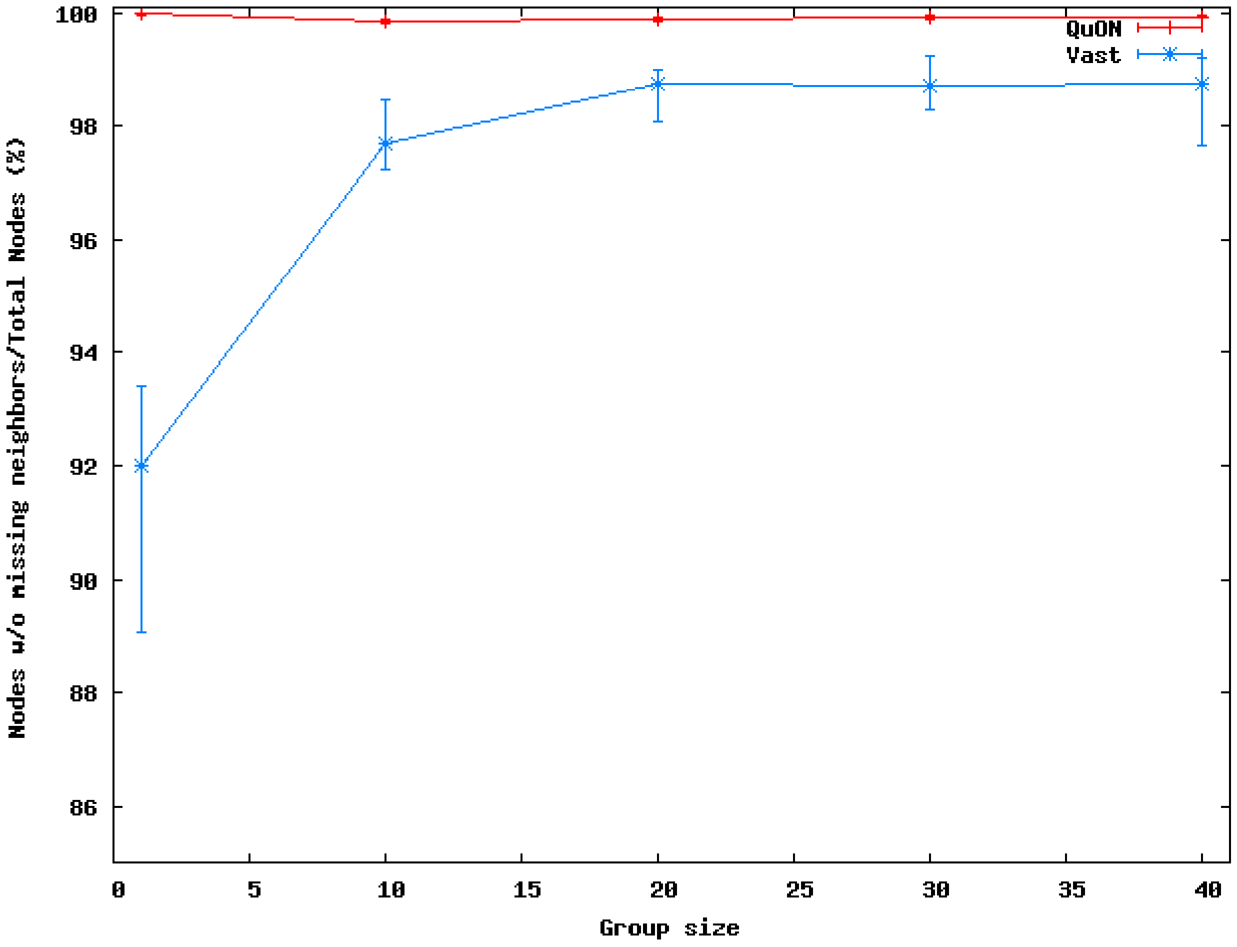
Metrics

- Connectedness
  - Percentage of players that have at least one neighbor
  - Percentage < 100% → at least one player lost all connections to the overlay
- Neighbor awareness
  - Percentage of players with no missing neighbors
  - Percentage < 100% → at least one player is missing at least one of his neighbors
- Bandwidth
  - Average and maximum bandwidth
- Latency
  - Time until movement update reaches neigbors
  - One Hop → ~ 90ms in all settings

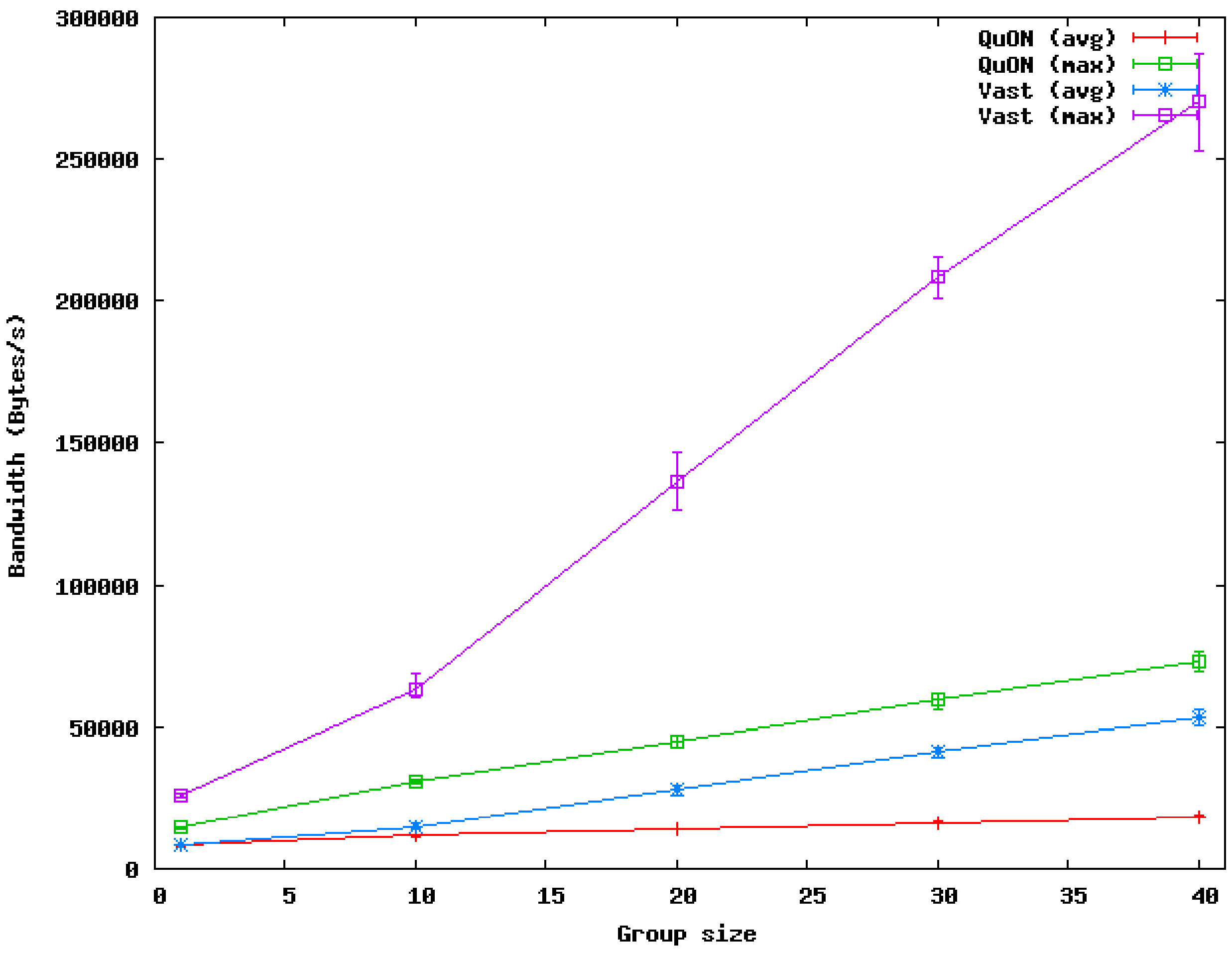
Results/Connectedness



Results/Neighbor awareness



Results/Bandwidth



Summary

- QuON is a new mutual notification protocol for MMOGs and Virtual Worlds
- No server or additional infrastructure needed
- Binding neighbors ensure connectedness and neighbor awareness
  - The are selected with the help of quad-trees
- Simulation results show practical performance
  - Perfect connectedness
  - Very good neighbor awareness
  - Does not exceed reasonable bandwidth requirements