Networking Issues and Solutions in Online Games Part III

Claudio E. Palazzi

Credits to Nadjib Achir, Khaled Boussetta, Arnaud Kaiser, Mark Claypool, David Finkel, Alexander Grant, Michael Solano

Analysis of network impact on games

- **We analyze :**
  1. The game network traffic
  2. The network impact on gaming experience

- **Definition of context for our study:**
  - Which type of video game to use?
  - Evaluation methodology?
  - Define a metric to quantify gaming experience?

Test dilemma

- Testing game playability in a standard way is unpractical
  - We cannot ask beta-tester to play 48 hours no-stop
    - They get bored
    - They get tired
    - They get sick
    - They might just not like the game
    - They could take revenge on the previous winner
    - Some of them could be a better player than others
  - Humans (unconsciously) self-adjust to the network conditions
  - I won -> it was playable

The NORT project

- NORT: Network-Oriented Reliable Tester
- A testbed of synthetic players
  - Never tired
  - Experiments can run forever
  - Do not need to like the game
  - They all have exactly the same capabilities
- Playing tons of matches under different network conditions highlights how the gameplay is affected

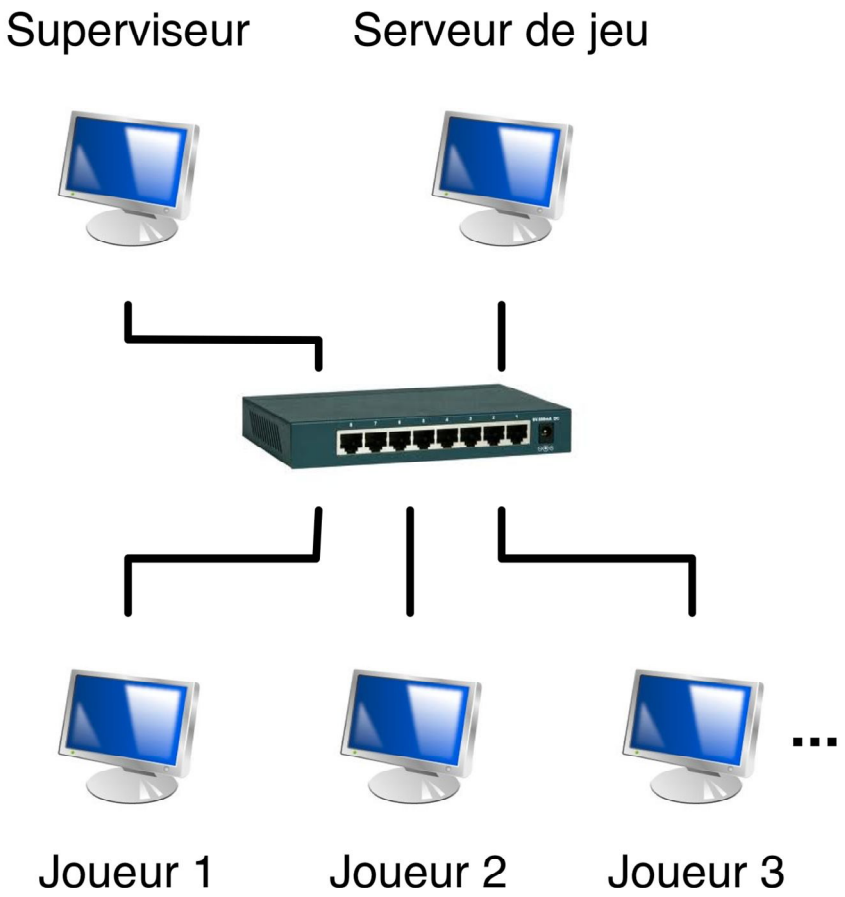
Definition of the context of our study

- **Game type: Quake III Arena**
  - First Person Shooter game (FPS)
  - FPS are the most sensitive to network delays [1][2][3]
  - Released by *id Software* in 1999, in open source since 2005
    - Bot players available online
- **Evaluation methodology: emulation testbed**
  - Repeated experiments (about 15h/experiment)
  - Control of the network quality
- **Quality of gaming metric**
  - Metrics in literature: score
  - FPS specific metrics: inter-frag time
  - [Mobility metrics: game disconnection]

[1] M. Claypool, *The effect of latency on user performance in Real-Time Strategy games*, 2005
[2] L. Pantel et al, *On the impact of delay on real-time multiplayer games*, 2002
[3] G. Armitage, *An experimental estimation of latency sensitivity in multiplayer Quake III*, 2003

Emulation testbed

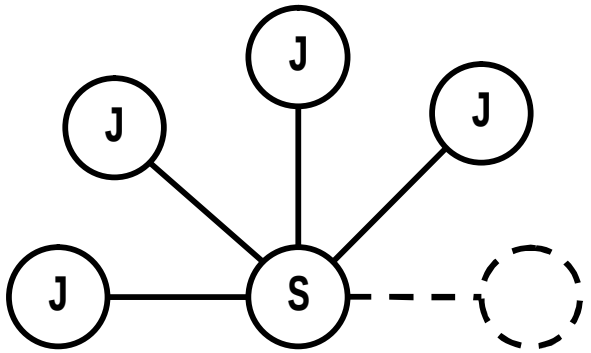
- **Scheduler**
  - Start/stop the testbed
  - [Mobility emulation]
- **Game server**
  - Register gaming actions in a log file
- **Players**
  - Autonomous bots
  - Energy emulation



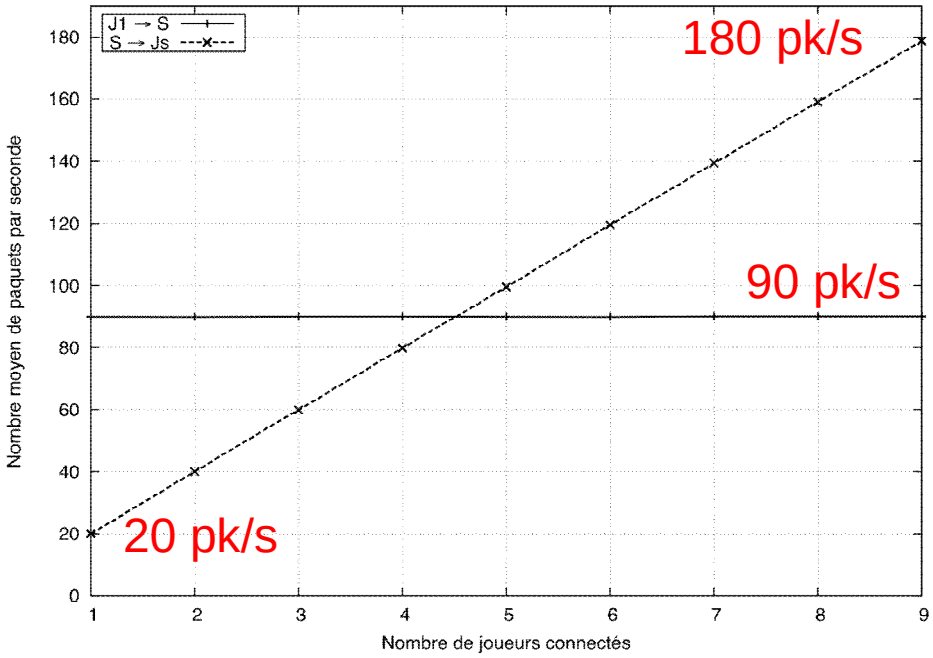
Testbed

Traffic analysis

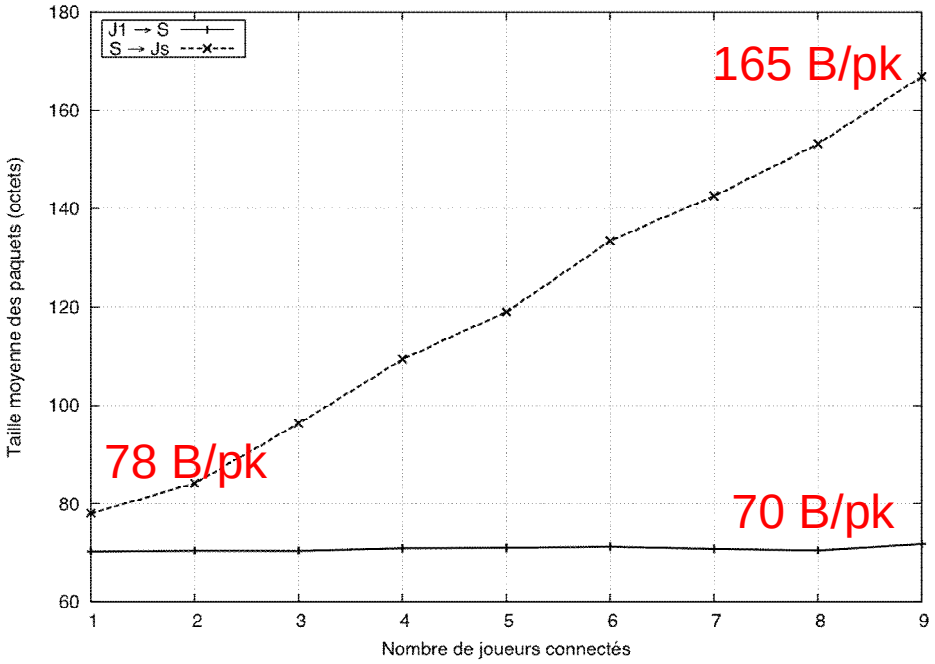
- Mono-hop topology
- S→C packet period: each 50 ms
- C→S packet period: each 11 ms



Average nb of packets vs nb players



Average packets size vs nb players



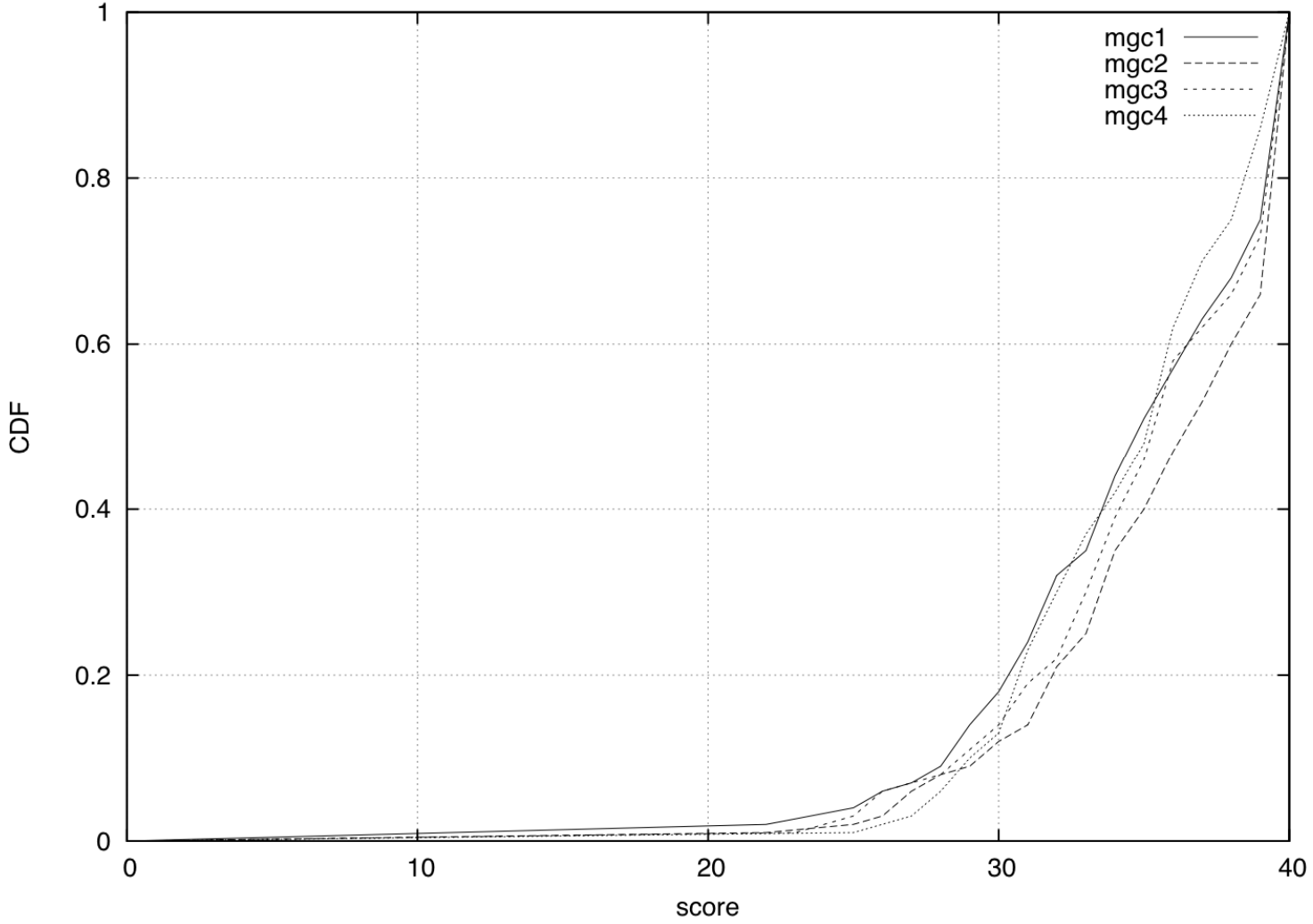
The metrics

- Score/Rank
- Frags Per Minute (FPM)

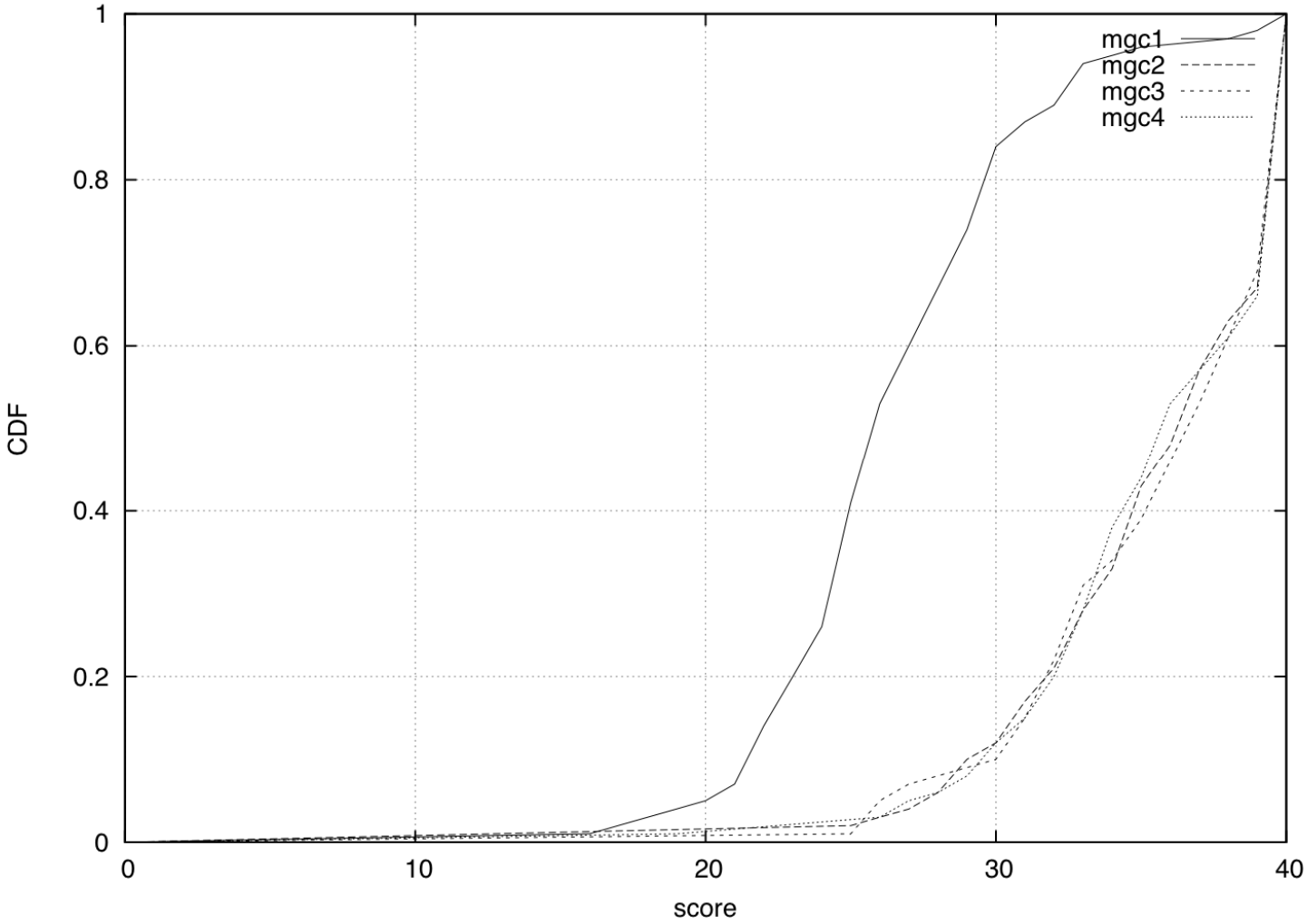
Delay and fairness

- It is a common belief that a FPS is playable when delay is below 150ms
  - It is playable, but is it still fair?

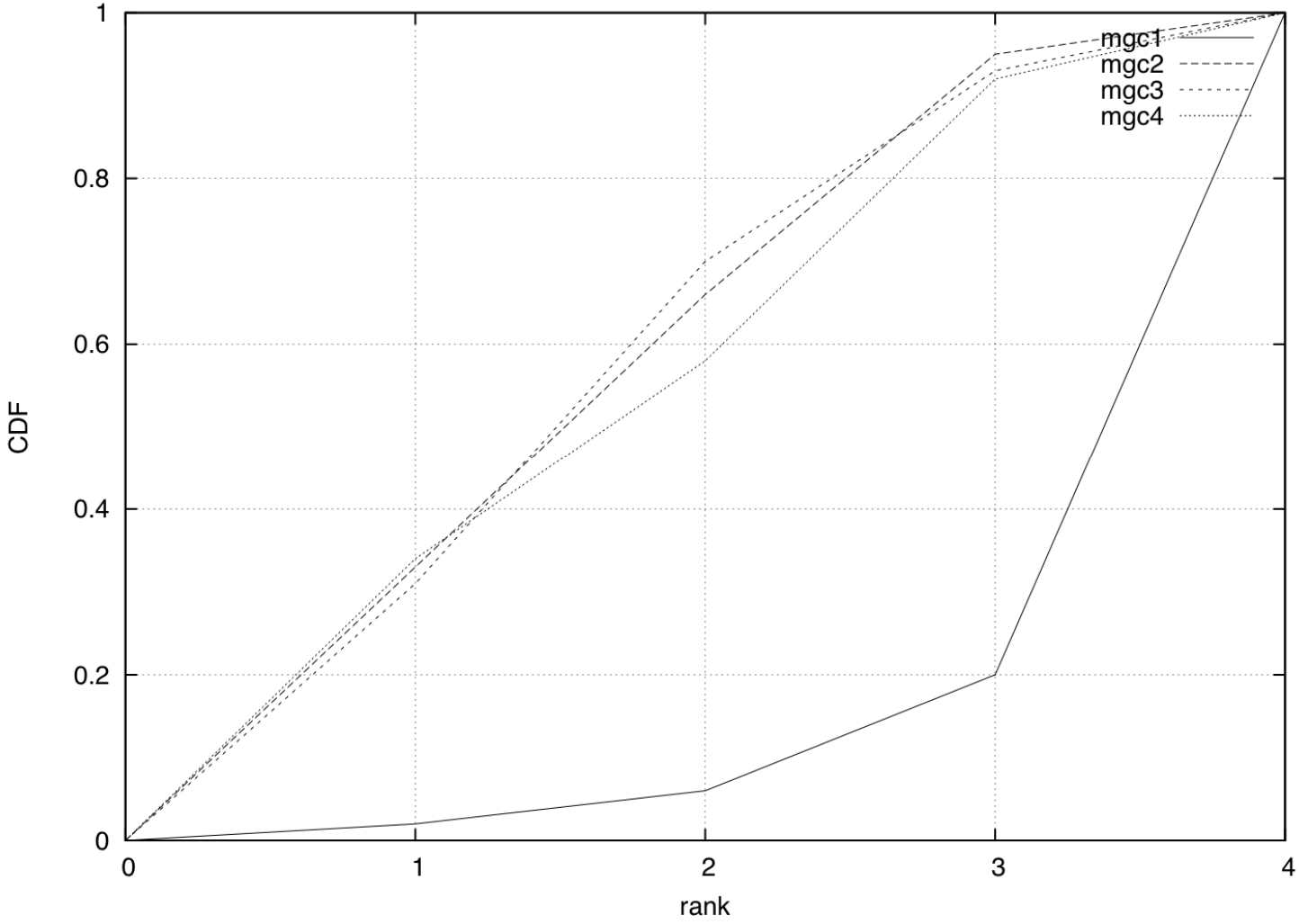
Score: no delay for anyone



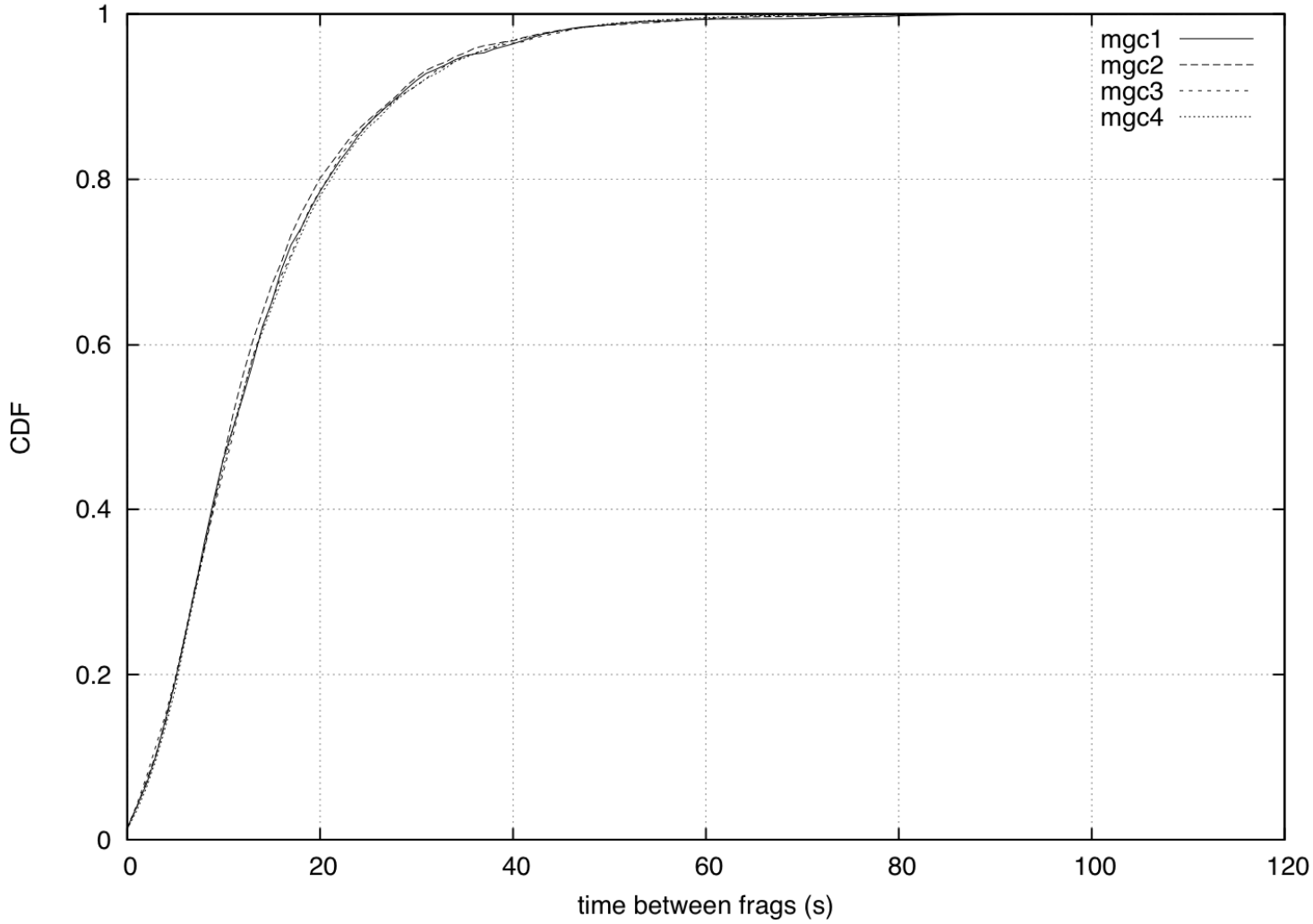
Score: one player with 50ms of delay



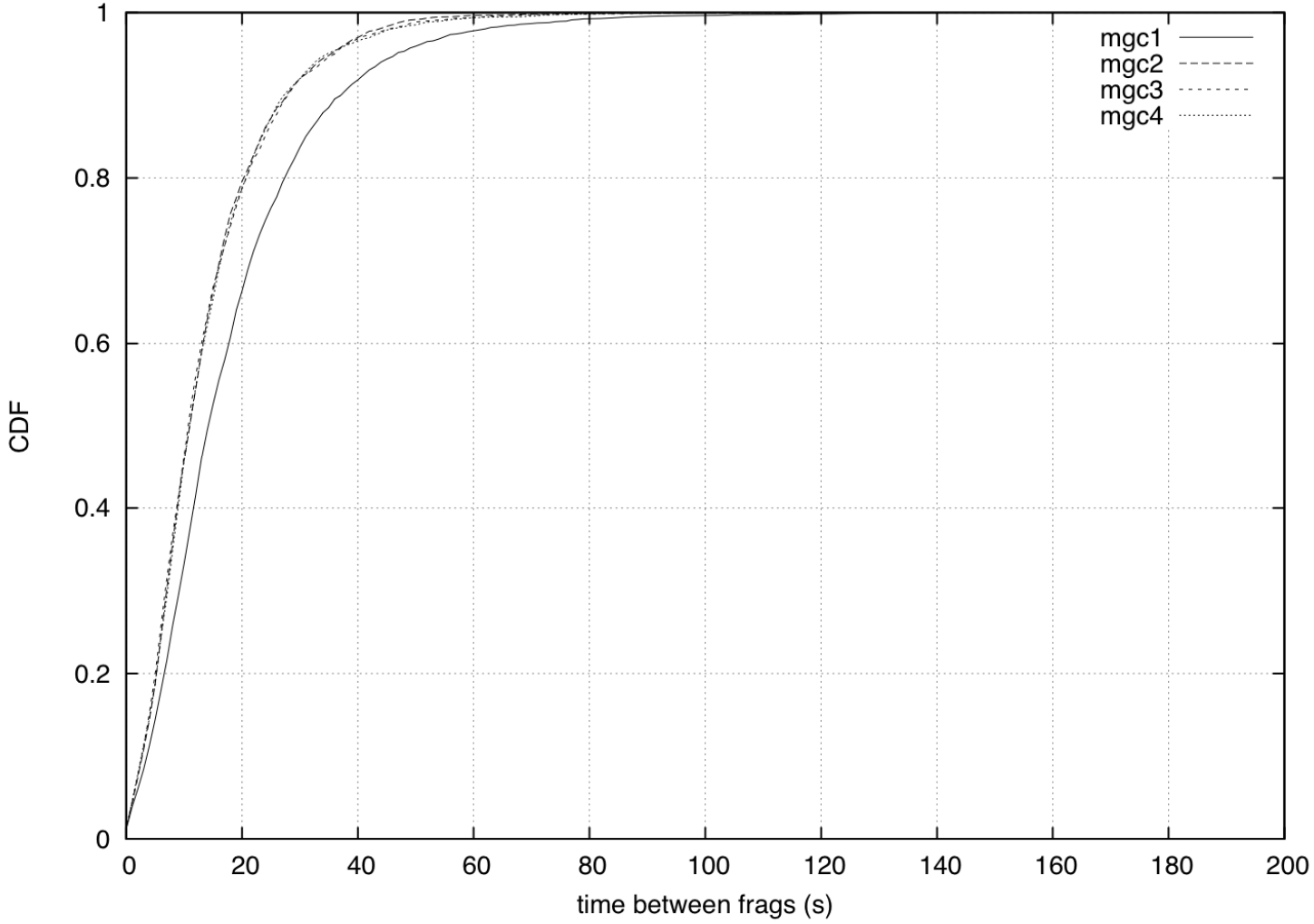
Rank: one player with 50ms of delay



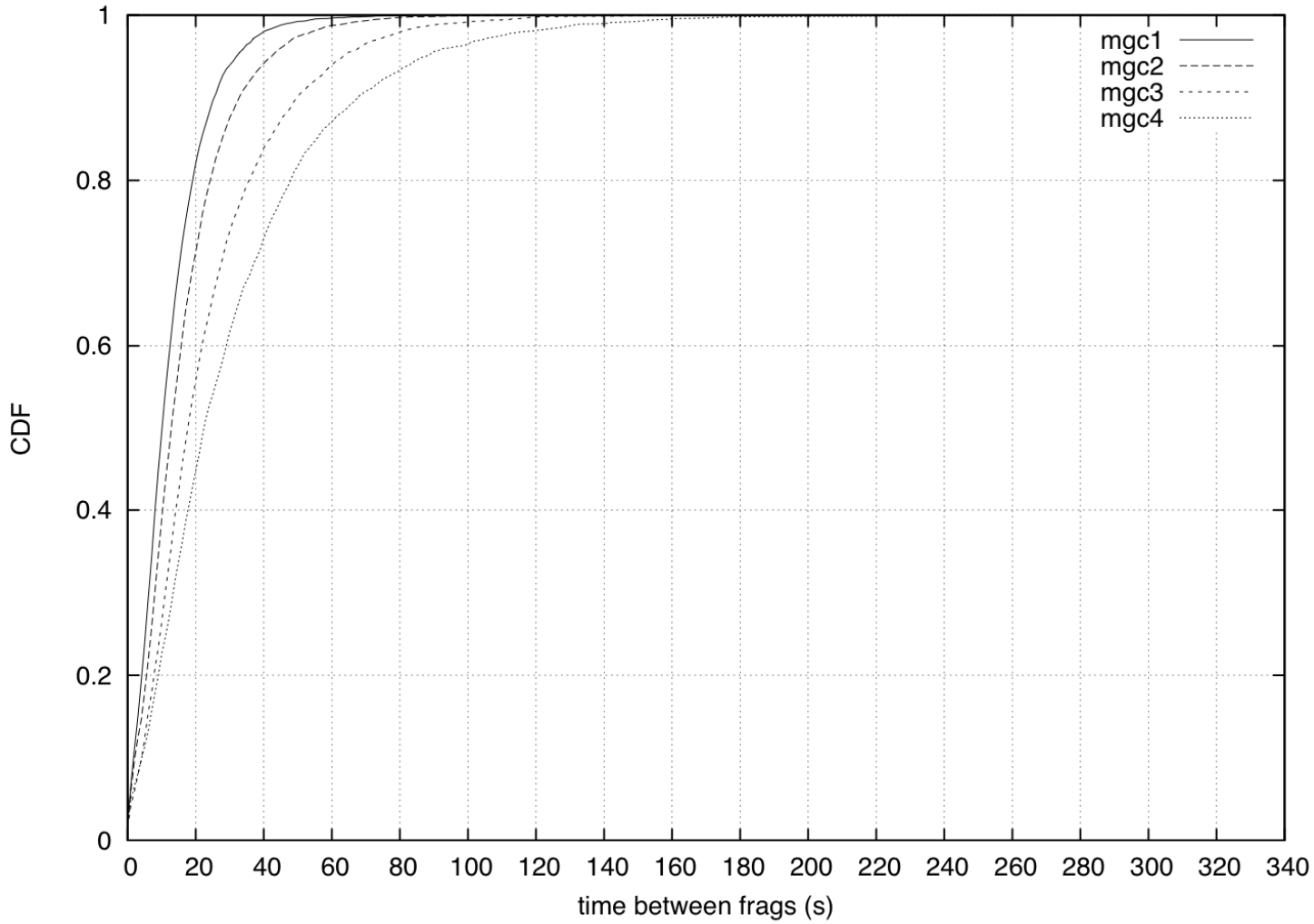
FPM: no delay for anyone



FPM: one player with 50ms of delay

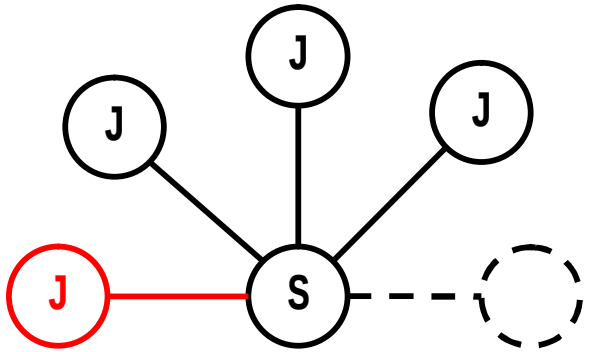


FPM: one player with 50ms of delay

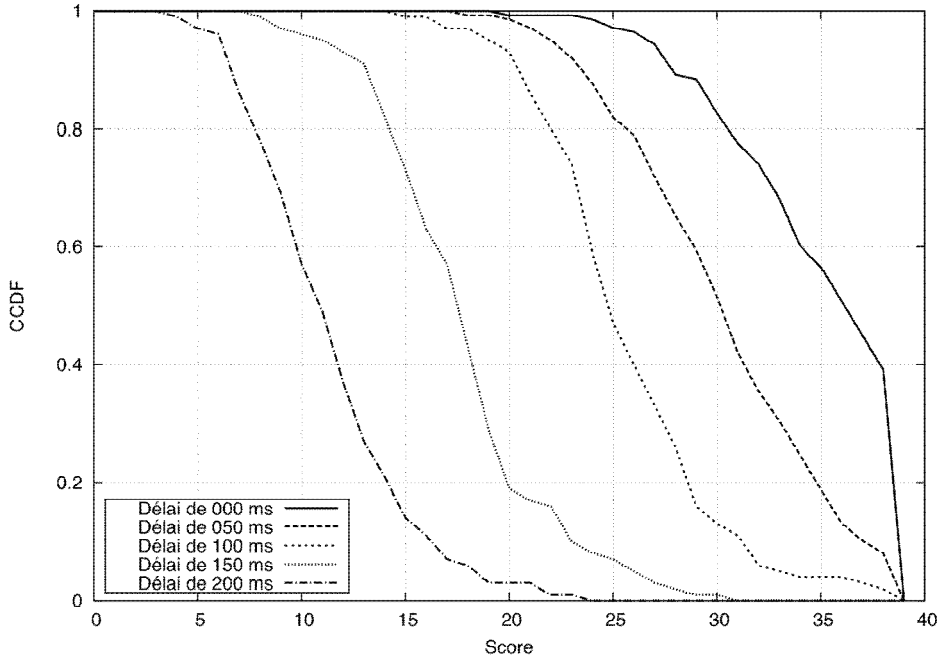


Network impact: delay

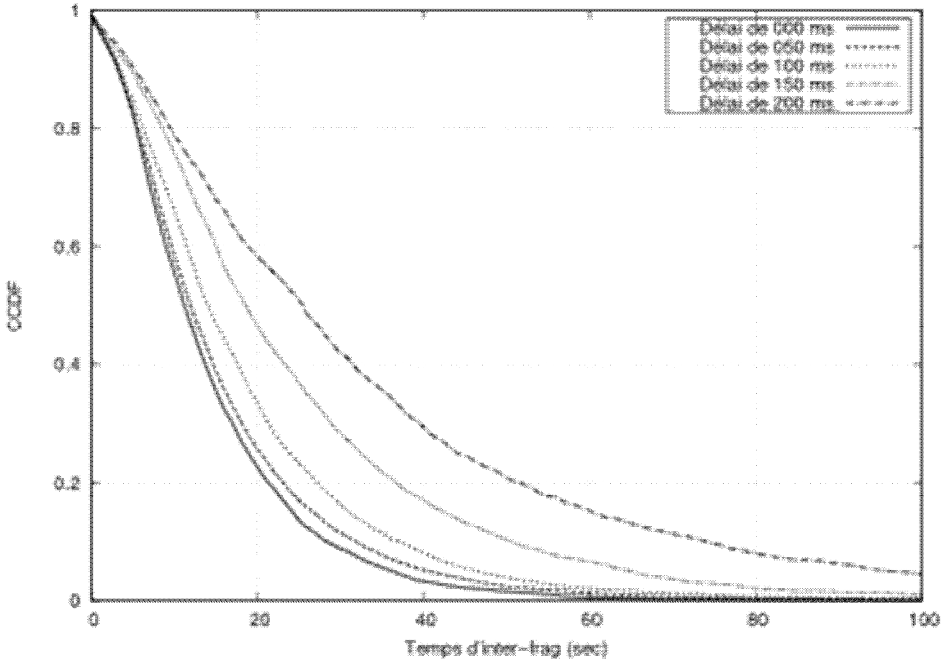
- **Mono-hop topology**



CCDF score

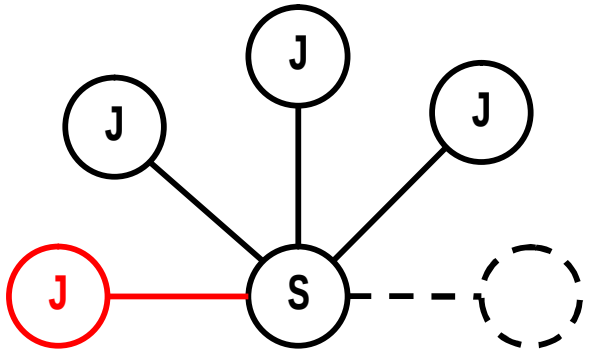


CCDF inter-frag time

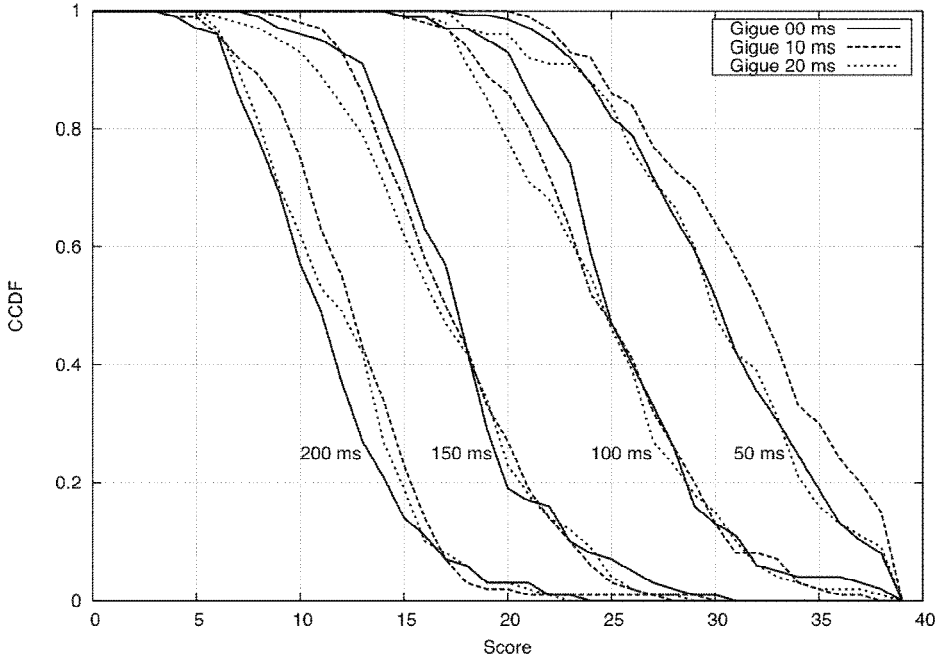


Network impact: jitter

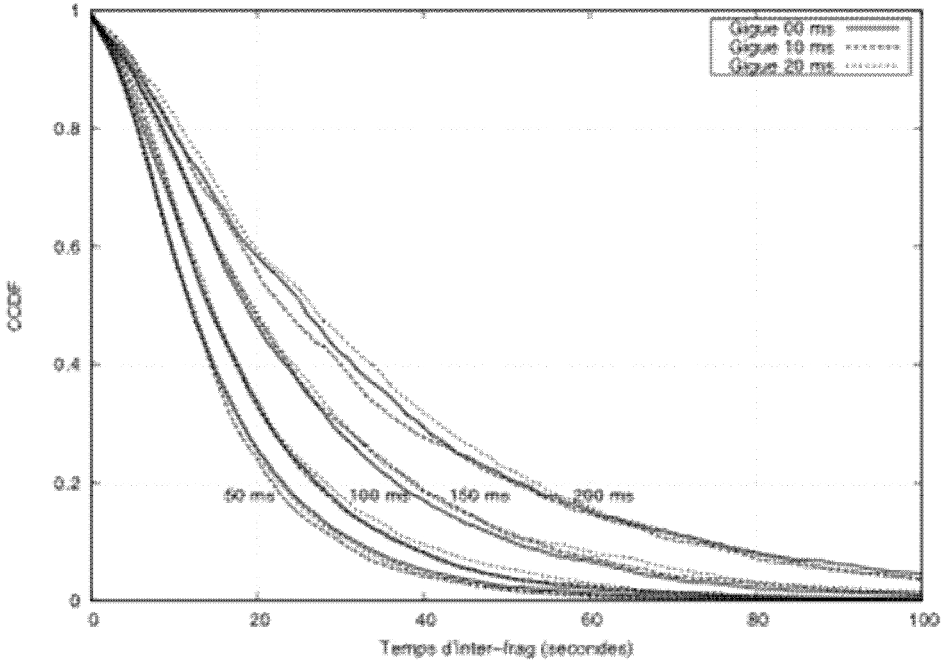
- **Mono-hop topology**



CCDF score

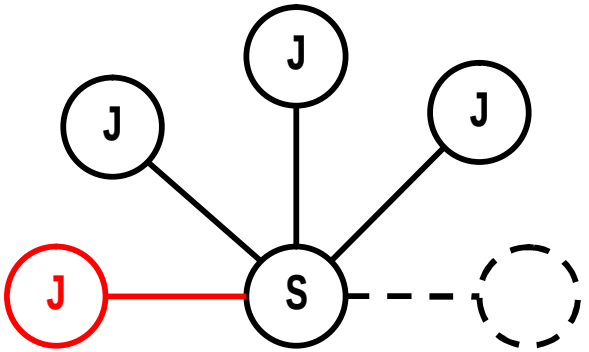


CCDF inter-frag time

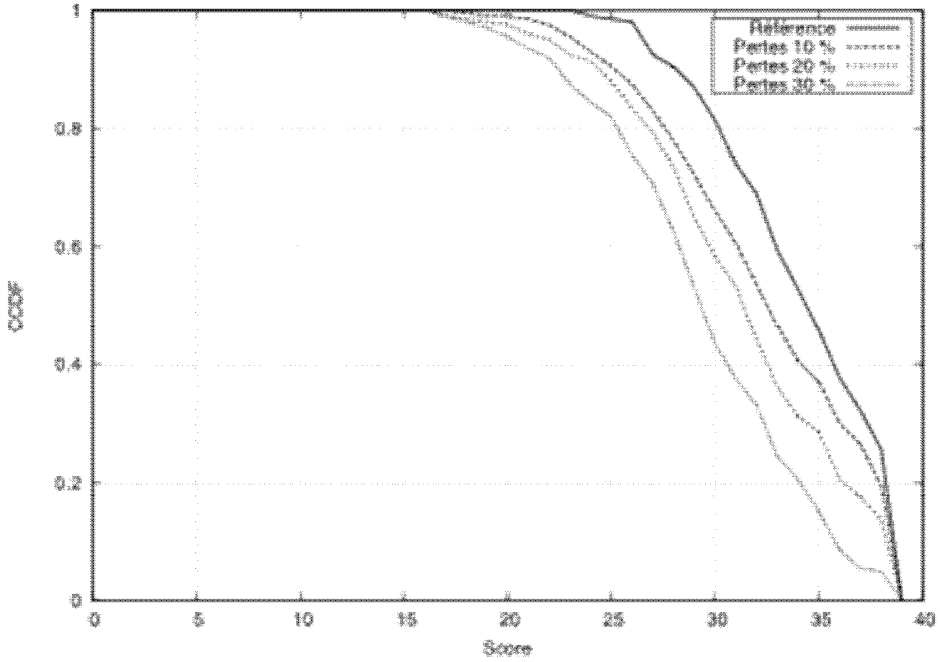


Network impact: losses

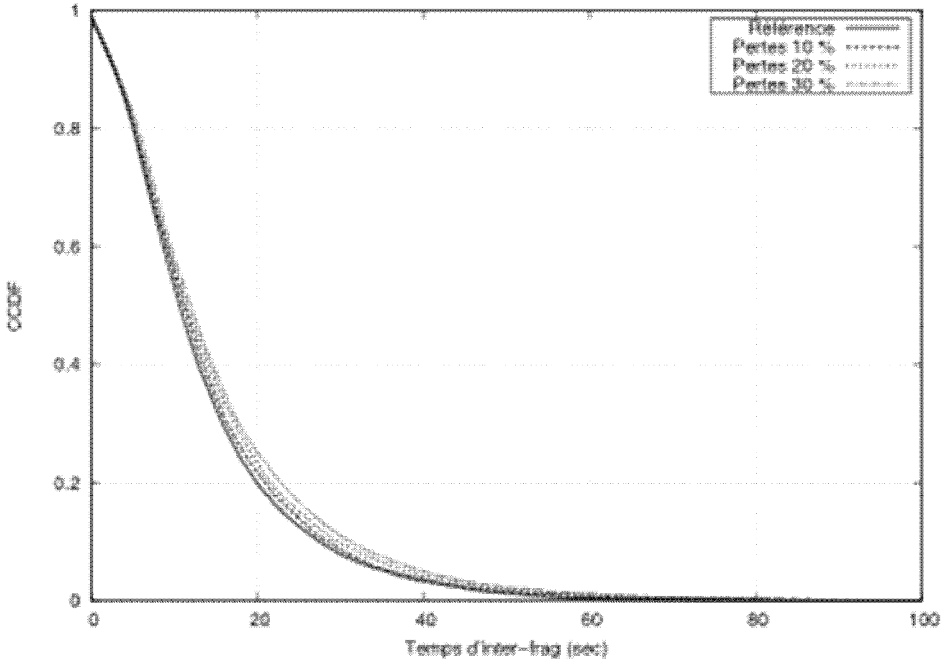
- **Mono-hop topology**



CCDF score



CCDF inter-frag time



Games in Ad Hoc Networks

- Multiplayer games on wired infrastructure
  - Home, office, Internet cafè, lab
  - LAN, Internet
- Radio interface is now integrated to portable terminals
  - WiFi, Bluetooth, 3G, LTE…
- Smartphones, Google glass… -> Play outdoor
- How about games on ad hoc networks?

Issues

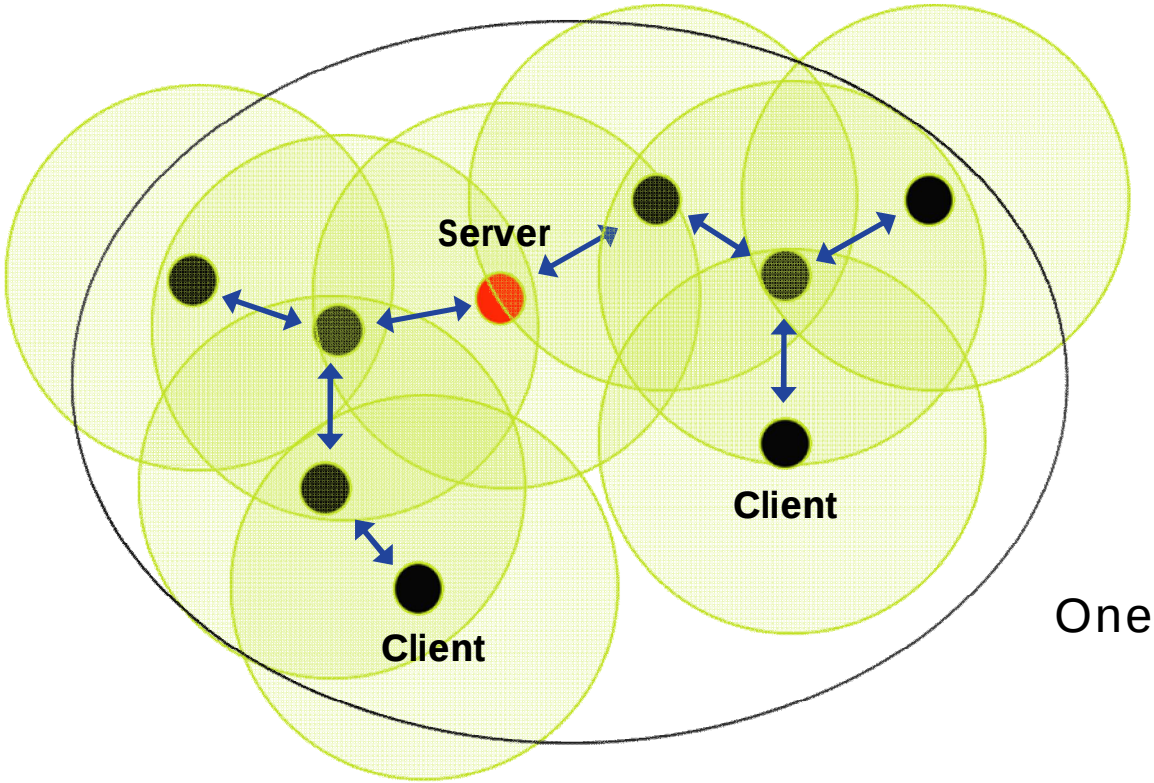
- For multiplayer games, the game play is sensitive to network resource availability
  - Not really in term of bandwidth (few Kbps are sufficient), but in term of:
    - Connectivity
    - packet losses
    - end to end delays
    - Jitter

Issues in MANET

- MANET (Mobile Ad-hoc Network)
  - Users are mobile
    1. Risk of disconnection
       - Some players (clients) isolated for the server
       - Server isolated from the players
    2. Energy consumption
    3. Possible impact on delays, jitter & packet losses due to misfit ad-hoc routing protocols

Connectivity

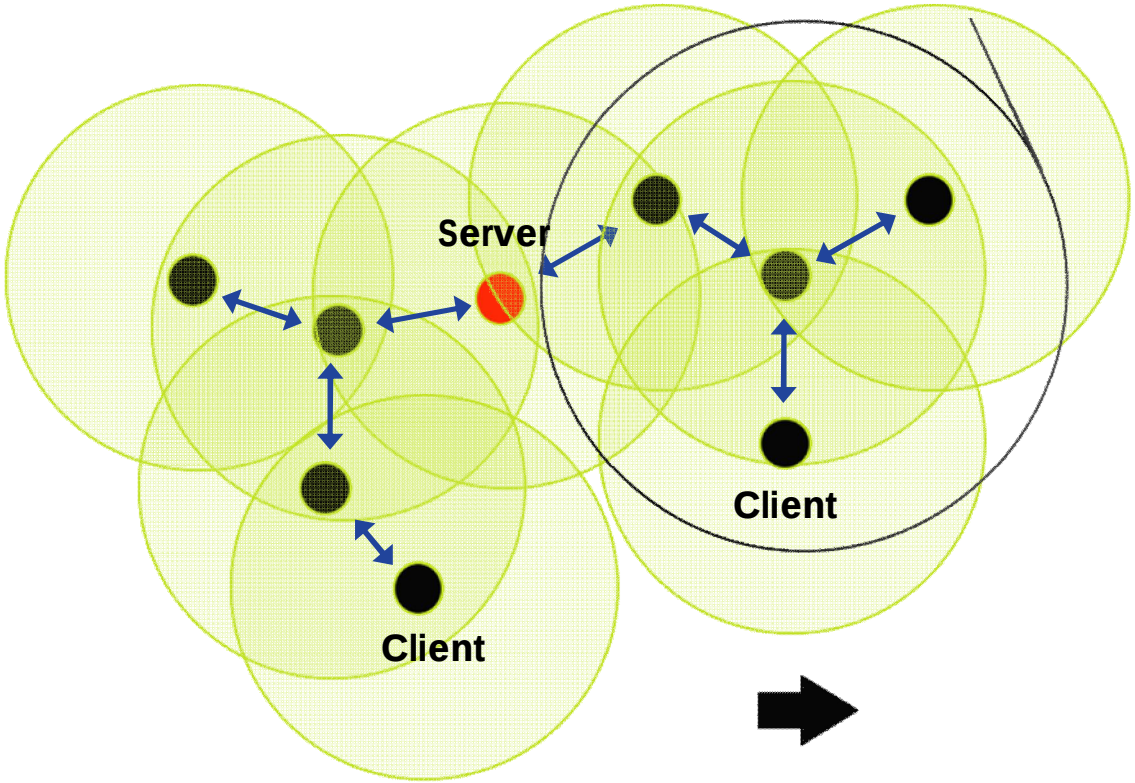
- The classical gaming model today is client-server
  - Client run by the player's PC
  - Server can be located locally or remotely in the internet
- If considering Ad Hoc environment



One Game session

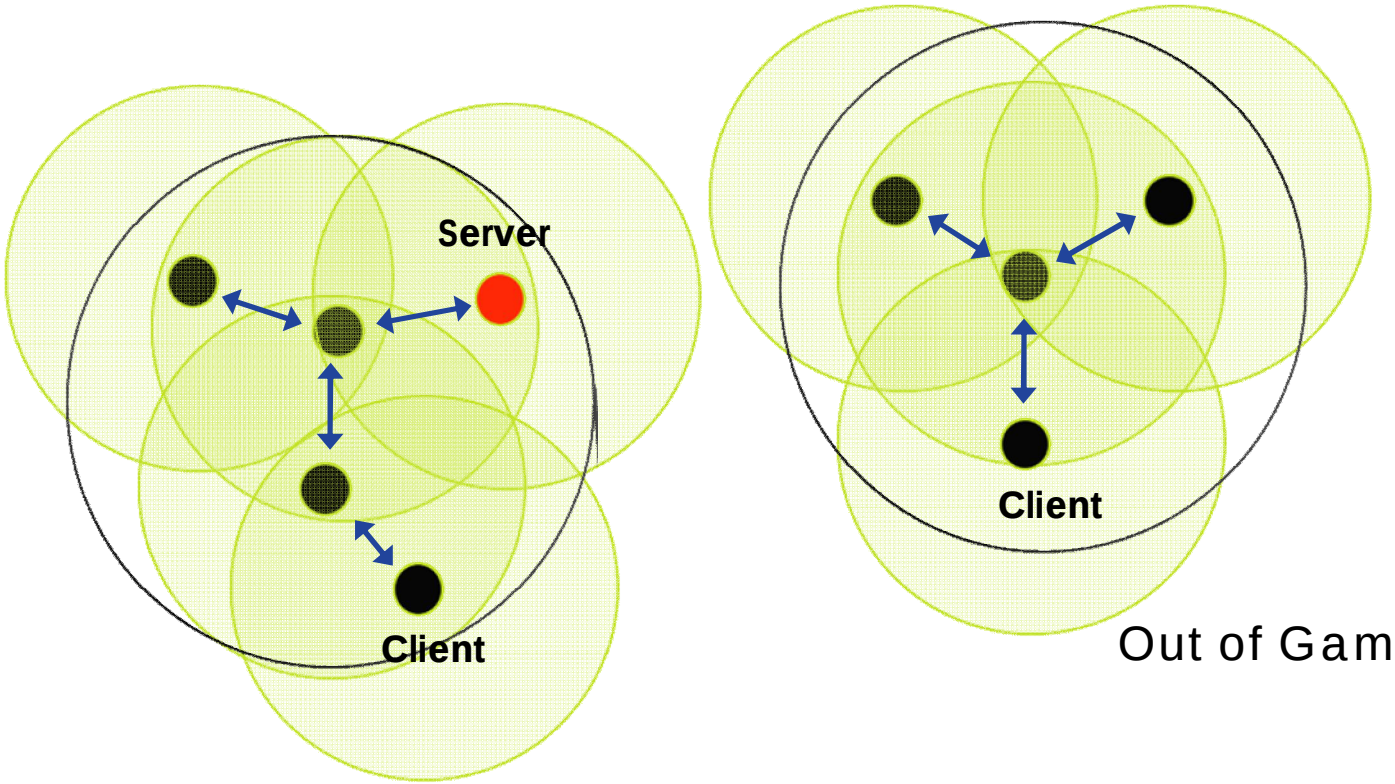
Connectivity

- The classical gaming model today is client-server
  - Client run by the player's PC
  - Server can be located locally or remotely in the internet
- If considering Ad Hoc environment



Connectivity

- The classical gaming model today is client-server
  - Client run by the player's PC
  - Server can be located locally or remotely in the internet
- If considering Ad Hoc environment



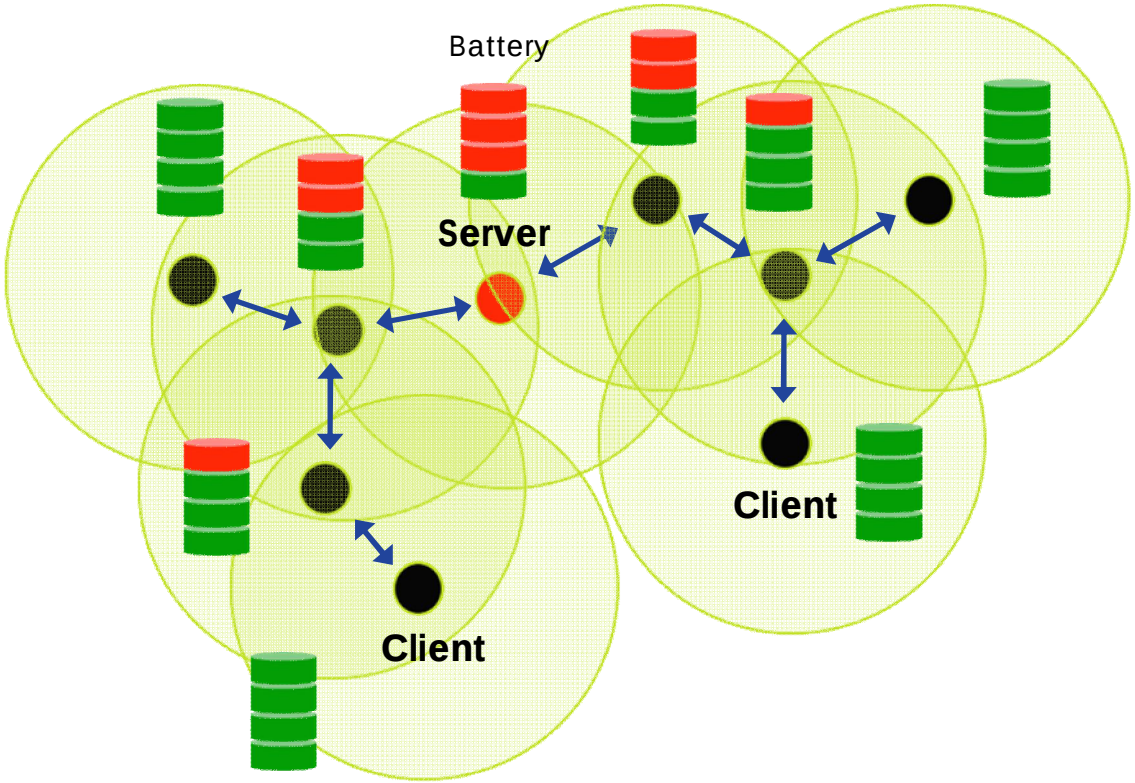
Out of Game session

Energy

- Energy consumption is not an issue for home players
  - Players can play for a very long time
- If considering Ad Hoc environment
  - Energy consumption is a major concern in case of Ad Hoc environment
    - Mobile consoles are **battery powered**,
    - …with **limited capacity**
    - Support:
      - game-related computation,
      - Visualization,
      - Communication
- Battery go down very fast

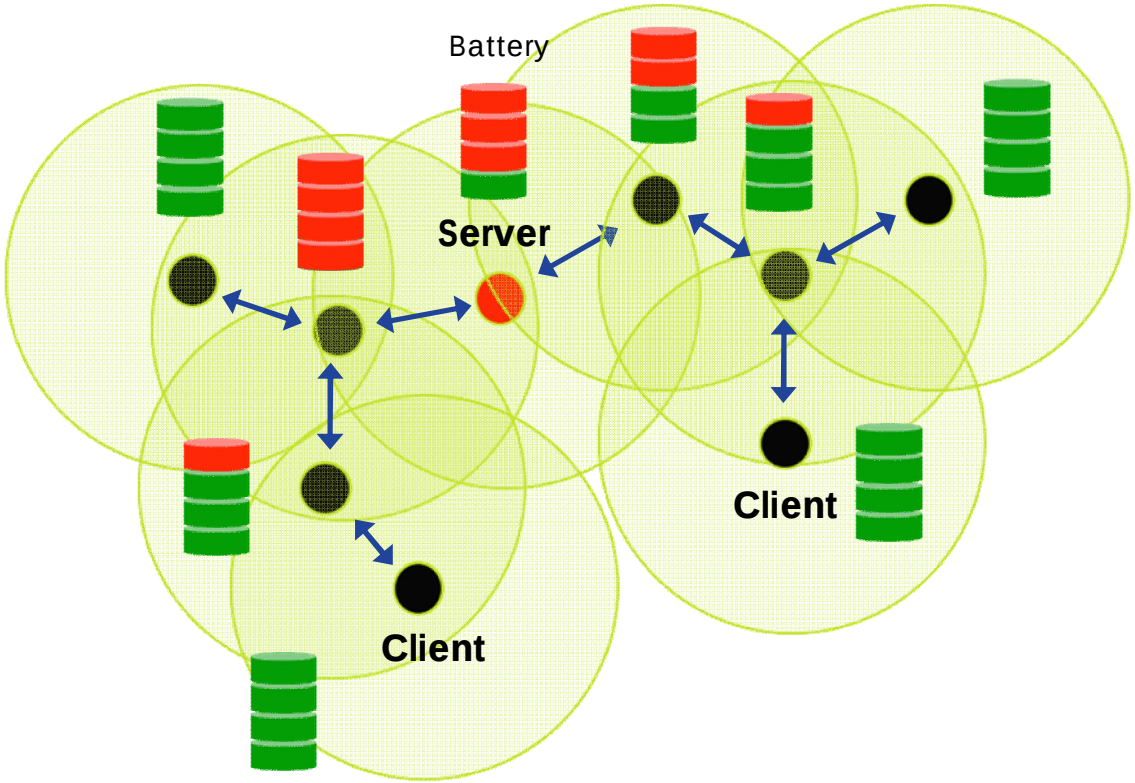
Energy

- Central nodes in a MANET -> faster energy decrease
  - Forward packets from some clients to the server
  - Forward packets from the server to some clients
  - Generate packets to the server



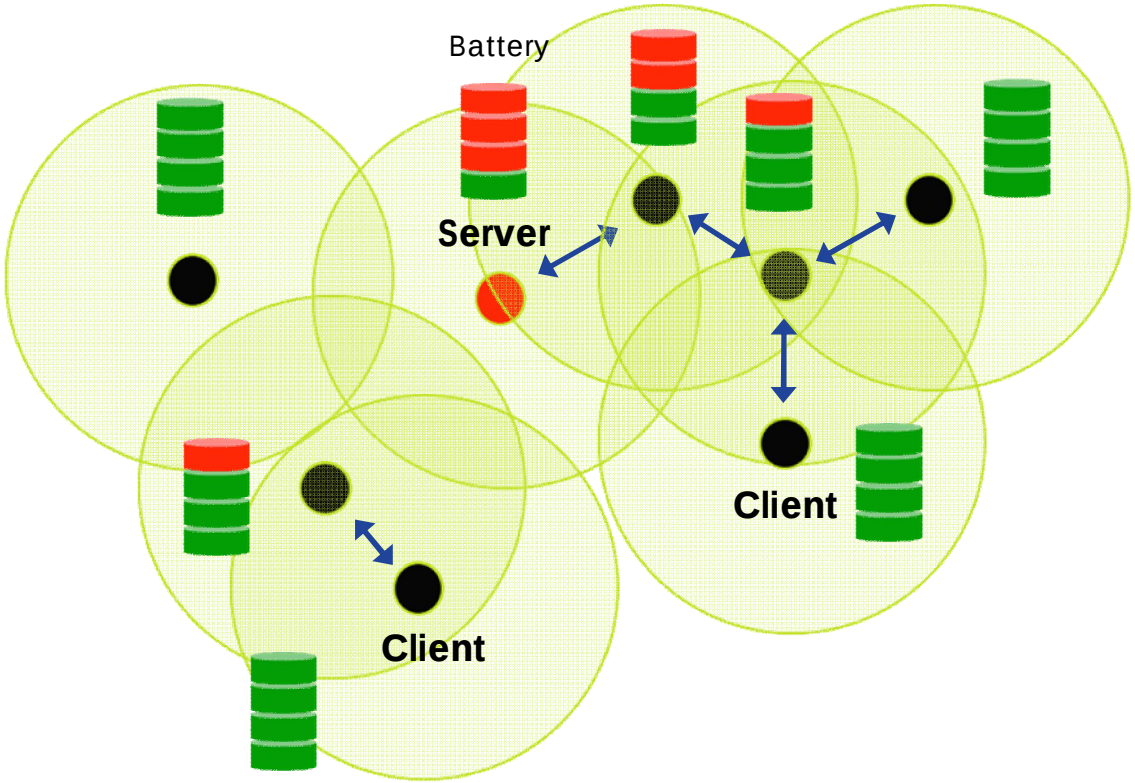
Energy

- Central nodes in a MANET -> faster energy decrease
  - Forward packets from some clients to the server
  - Forward packets from the server to some clients
  - Generate packets to the server



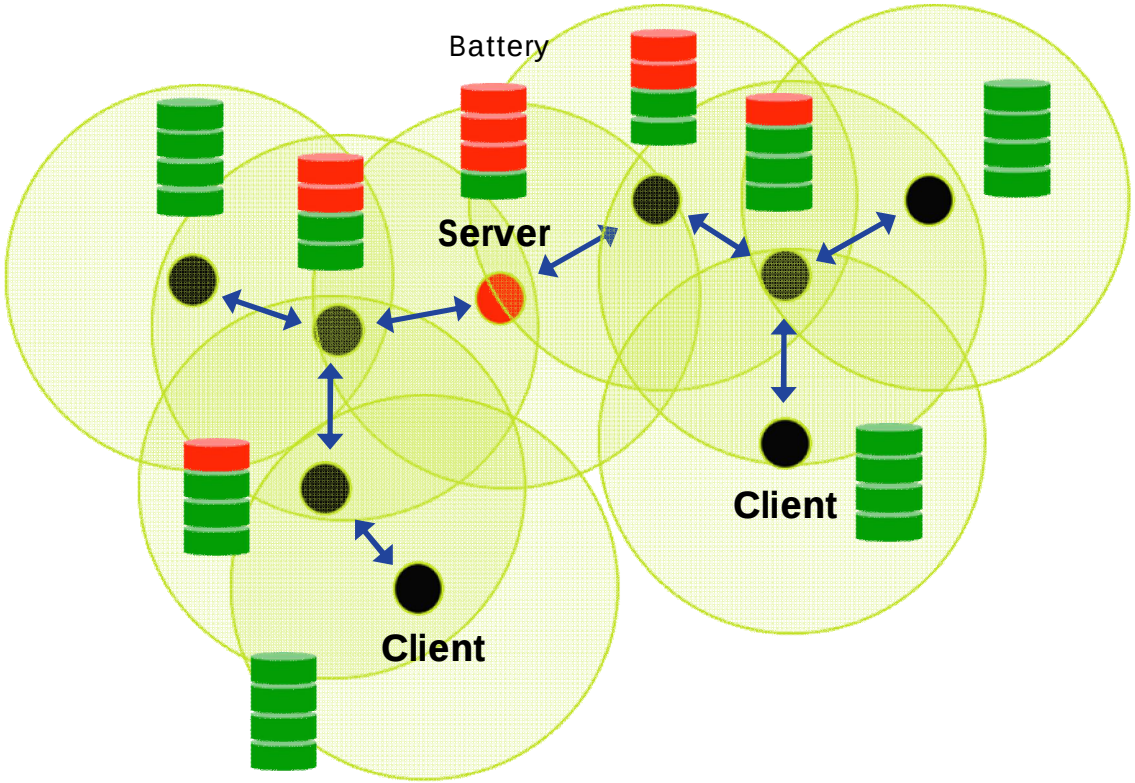
Energy

- Central nodes in a MANET -> faster energy decrease
  - Forward packets from some clients to the server
  - Forward packets from the server to some clients
  - Generate packets to the server



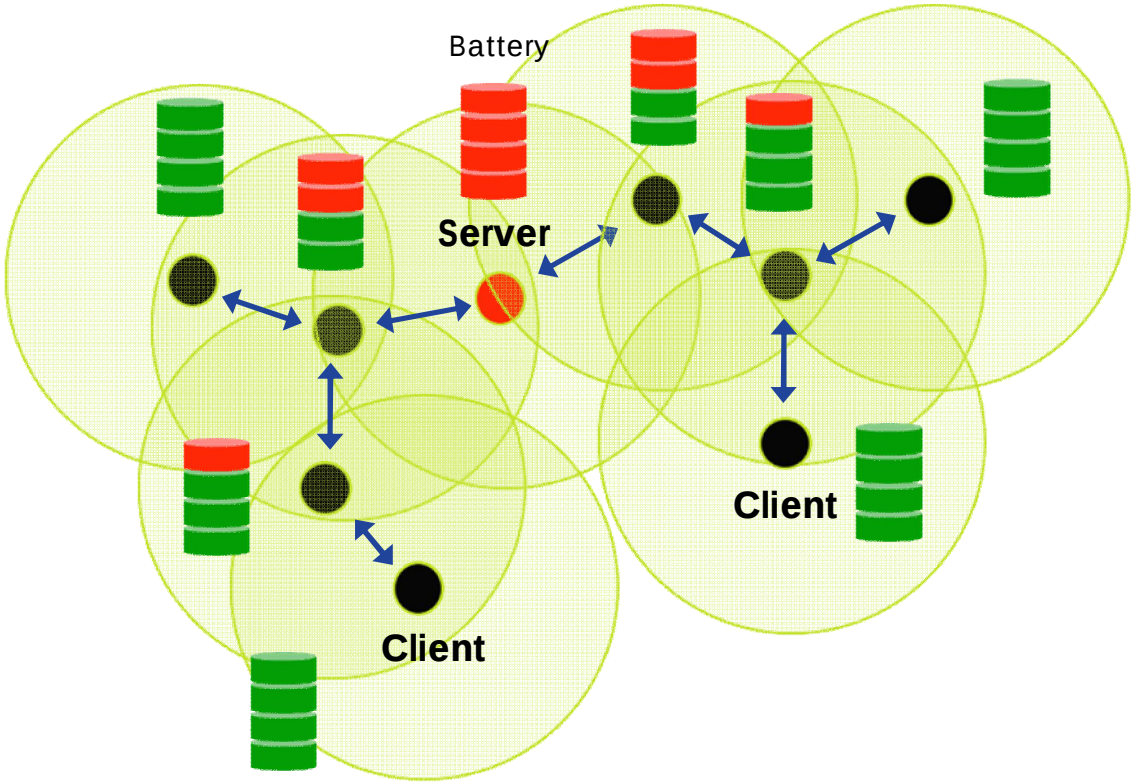
Energy

- Energy consumption of the game server
  - Receive packets of each client
  - Send packets to each client



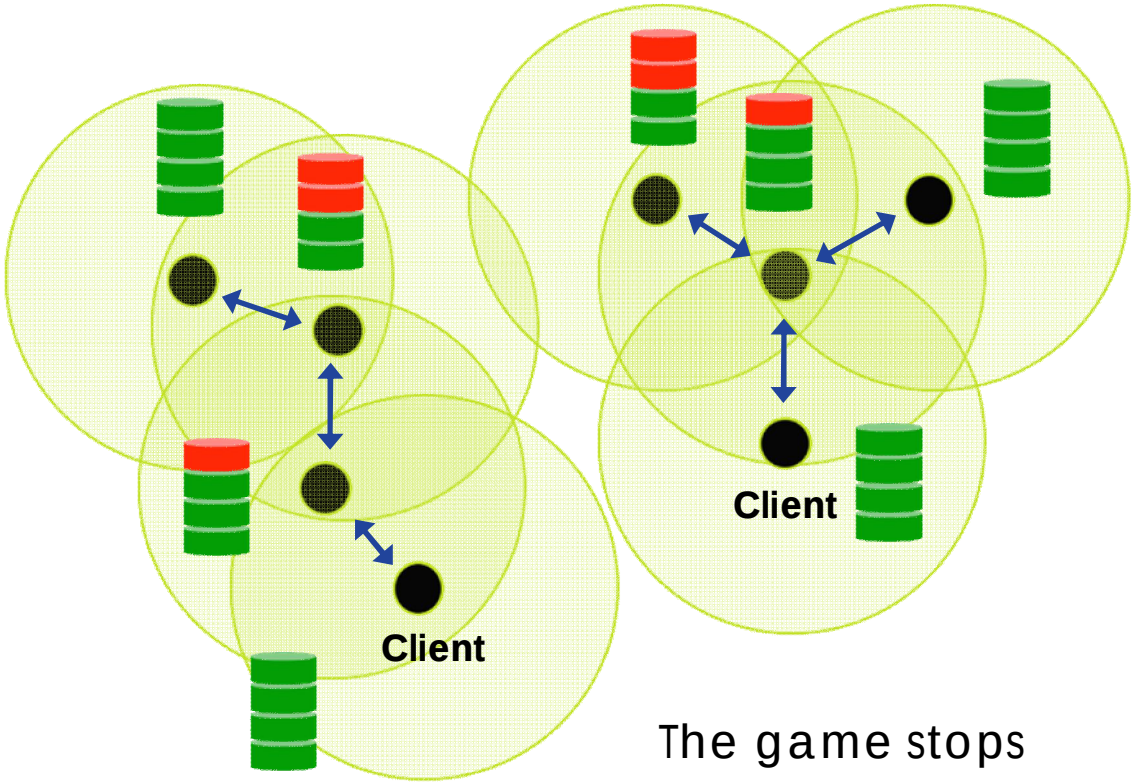
Energy

- Energy consumption of the game server
  - Receive packets of each client
  - Send packets to each client



Energy

- Energy consumption of the game server
  - Receive packets of each client
  - Send packets to each client



The game stops

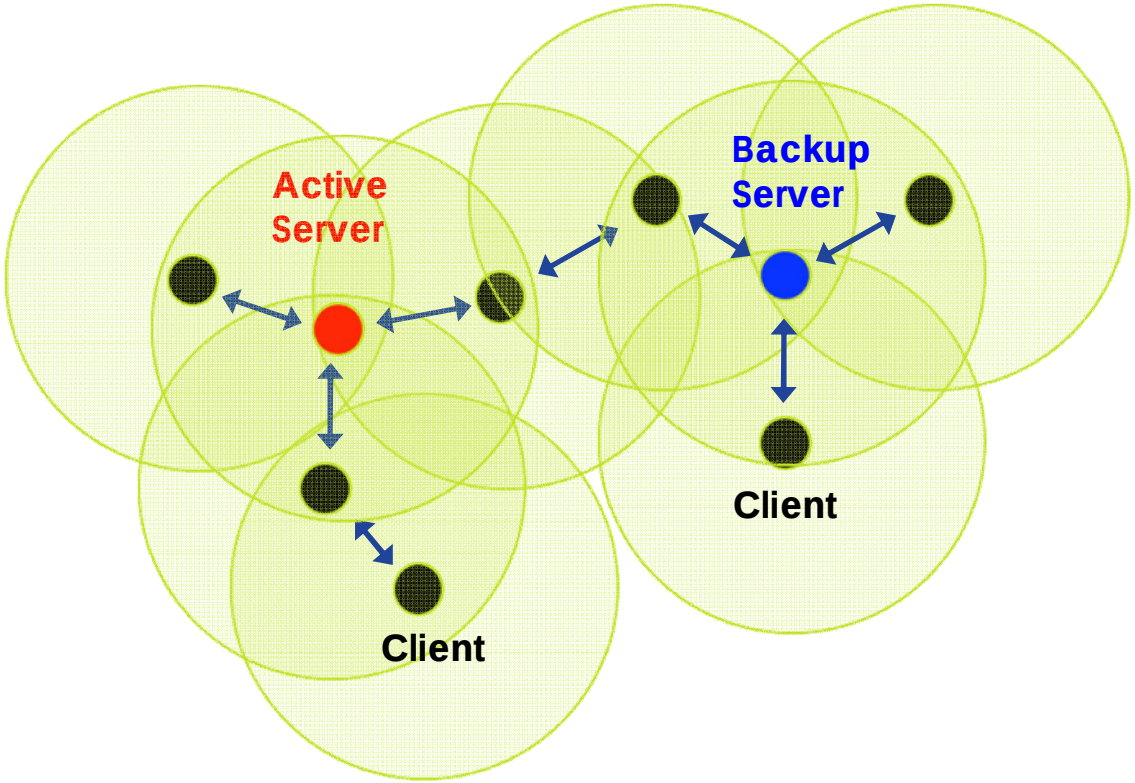
How to adapt multiplayer gaming to ad hoc environment ?

- Convert the Client-Server Architecture to pure Peer-to-Peer architecture
  - Advantages
    - no dedicated clients or server terminals
    - only equal peer nodes that can simultaneously act as a client and a server
    - every peer sends its local player actions to all other peers in a completely distributed manner
  - Drawbacks
    - each peer is responsible of computing the overall game state
      - Need of efficient synchronization process
    - **High energy consumption**
      - **Unfairly distributed among the various nodes (e.g., server)**

Hybrid Architecture

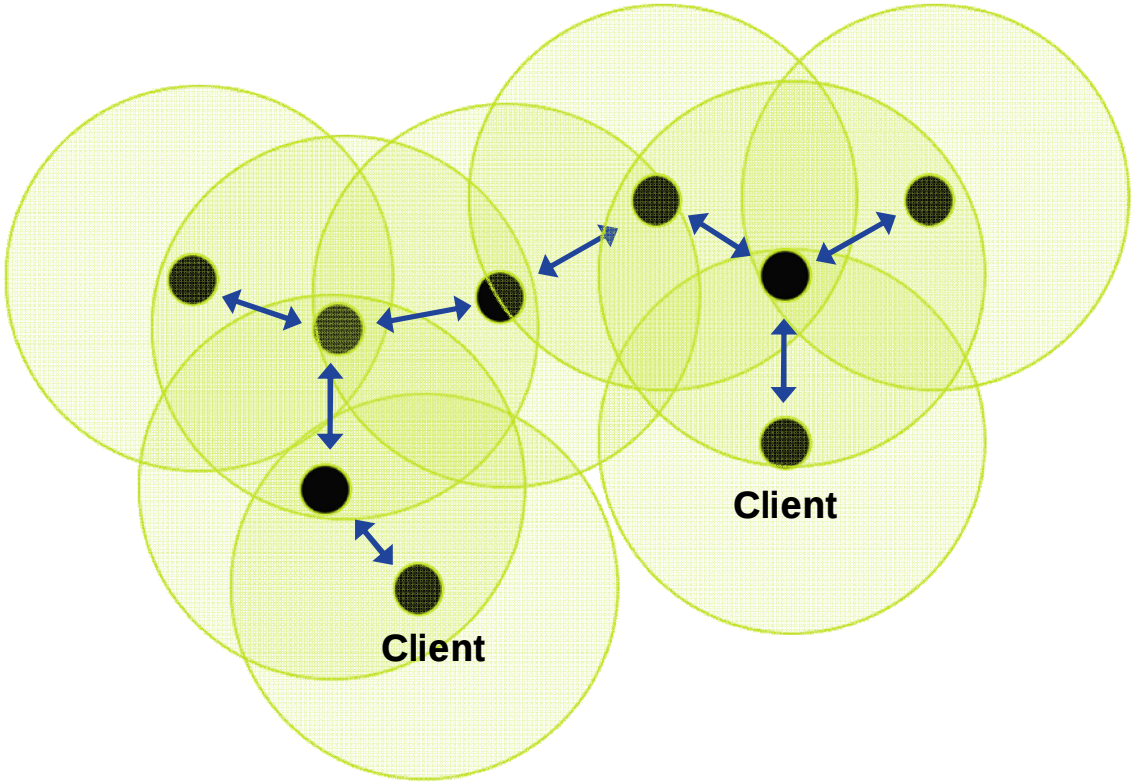
- ***More than one server***
  - At any time, only one server is active, while others are just kept synchronized
- Three category of participant in each game session:
  1. ***Active server***:
     - A node that holds game status
     - Communicates with nodes of its game session
     - Keeps synchronized backup servers
  2. ***Backup server (or Dormient server):***
     - Client that can be active server in the future
  3. ***Client:***
     - Node that only plays

Hybrid Architecture



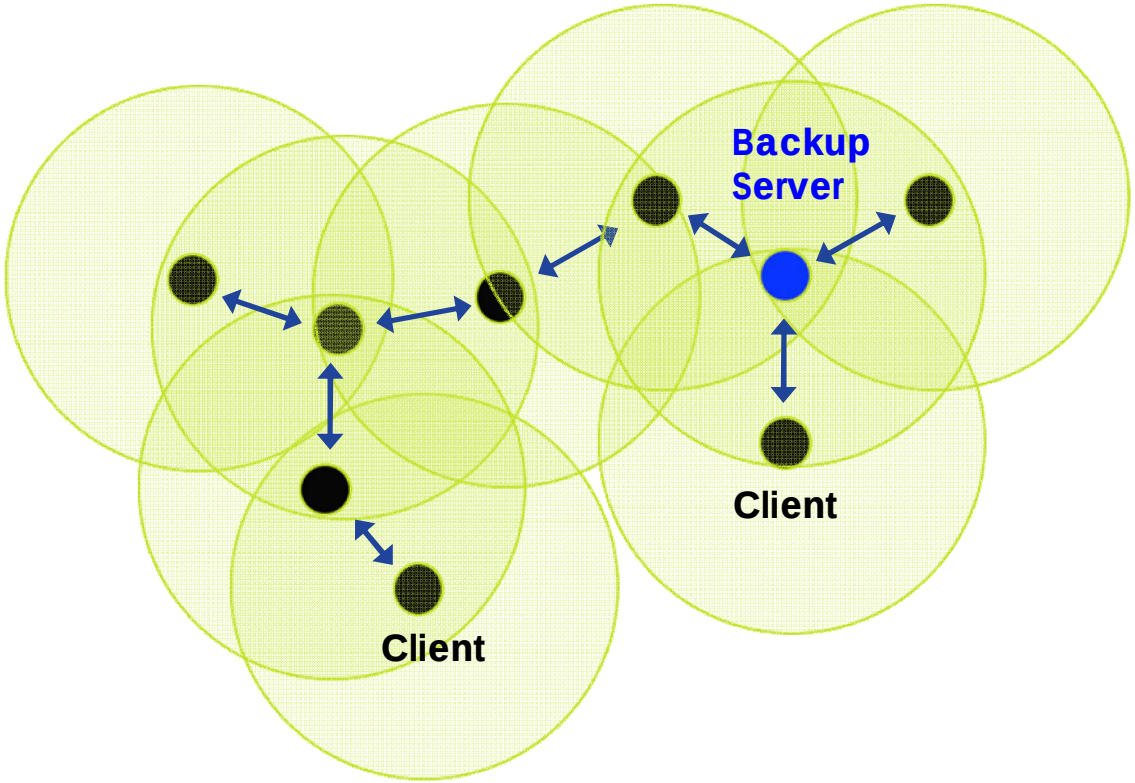
Server and backup servers election

- Active server election is done in round robin way
- Backup servers election is done using a reachability matrix
  - Maximizing the number of reached nodes



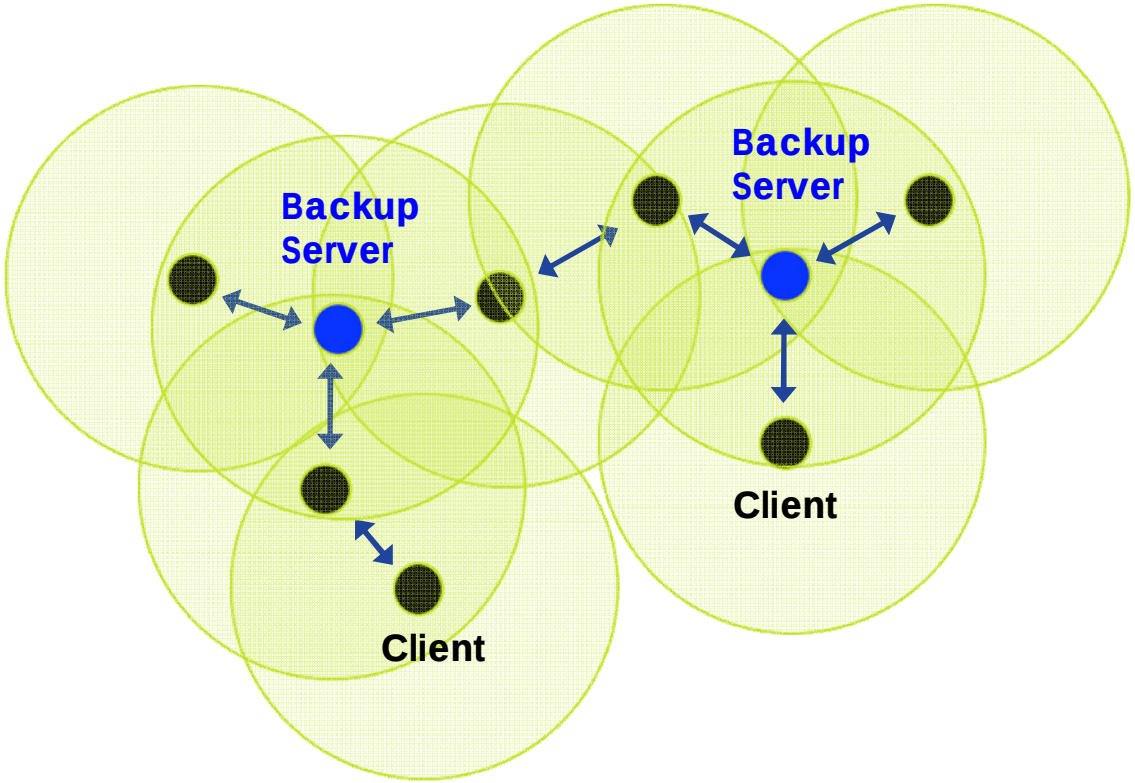
Server and backup servers election

- Active server election is done in round robin way
- Backup servers election is done using a reachability matrix
  - Maximizing the number of reached nodes



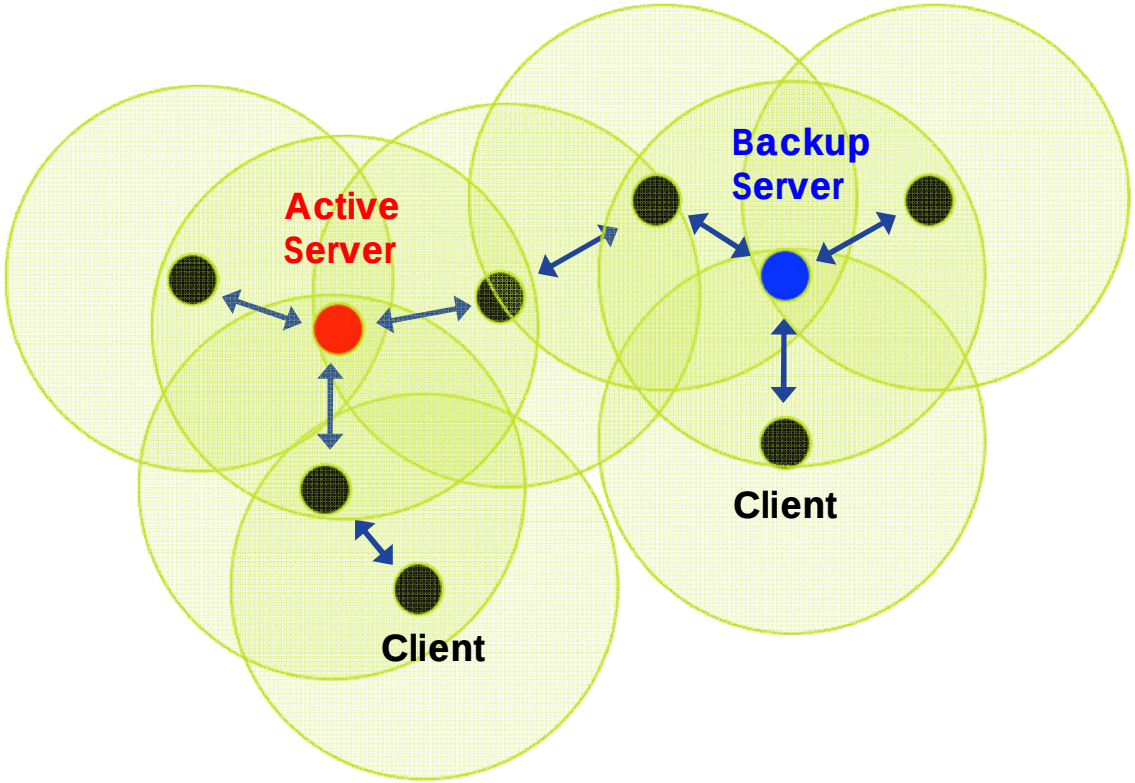
Server and backup servers election

- Active server election is done in round robin way
- Backup servers election is done using a reachability matrix
  - Maximizing the number of reached nodes



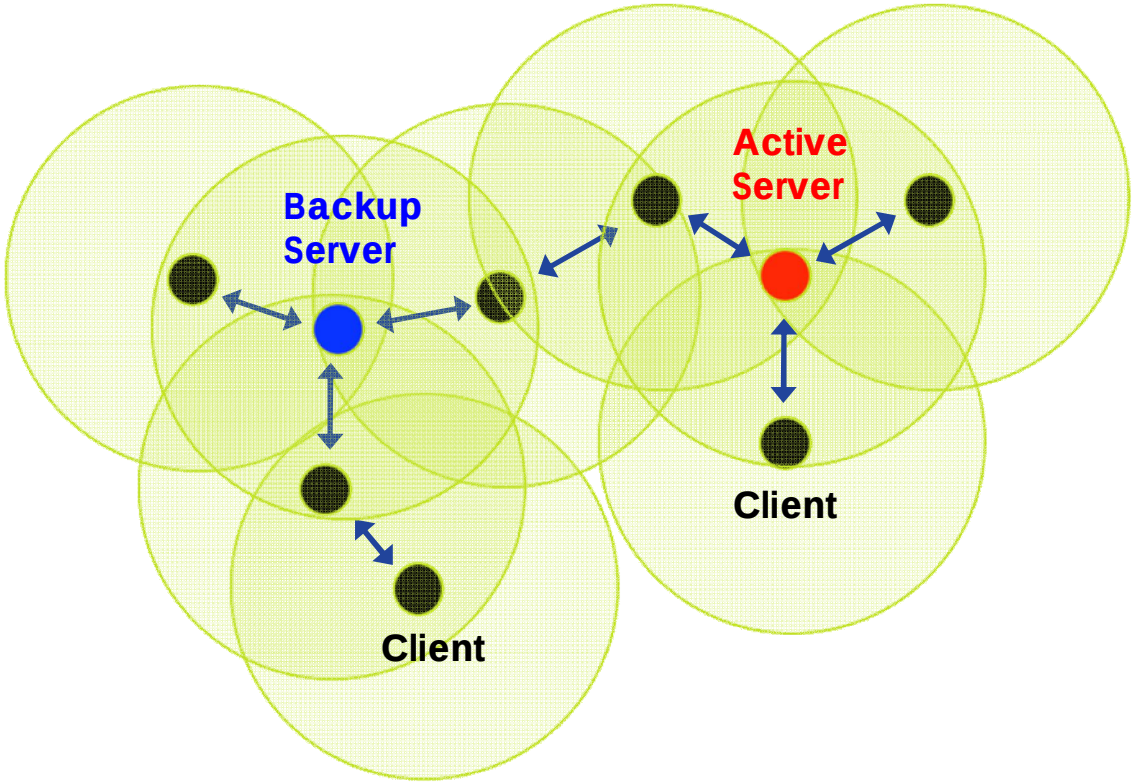
Server and backup servers election

- Active server election is done in round robin way
- Backup servers election is done using a reachability matrix
  - Maximizing the number of reached nodes



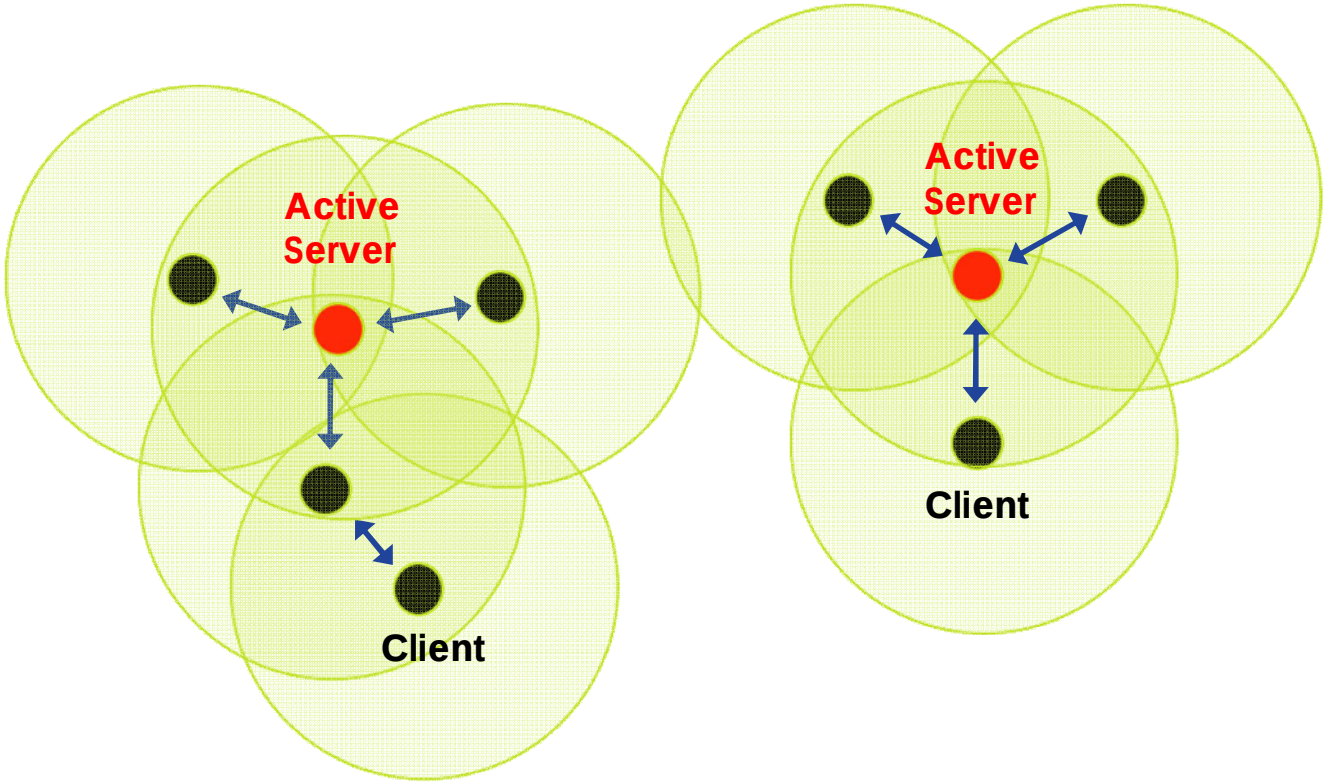
Server and backup servers election

- Active server election is done in round robin way
- Backup servers election is done using a reachability matrix
  - Maximizing the number of reached nodes



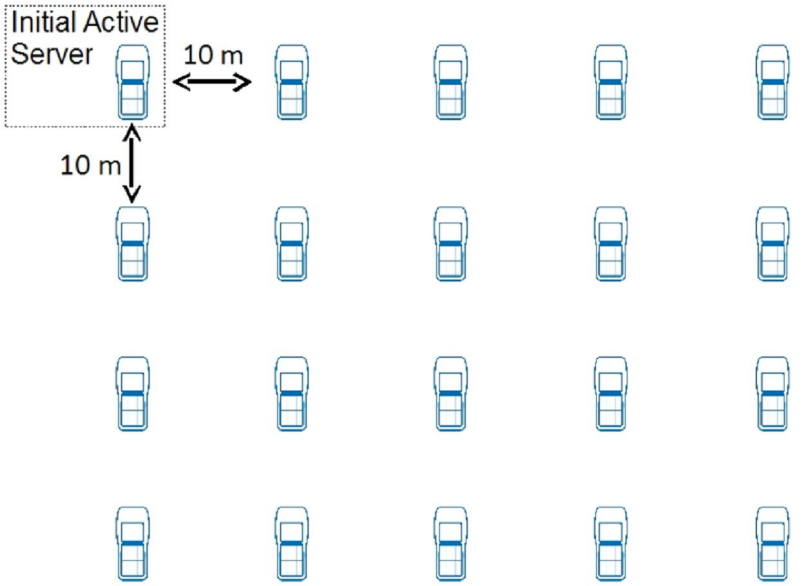
Partition management

- In case of node movement
  - Periodically the backup servers check if they can communicate with the active server
    - If not, one of them becomes the active server



Simulation parameters

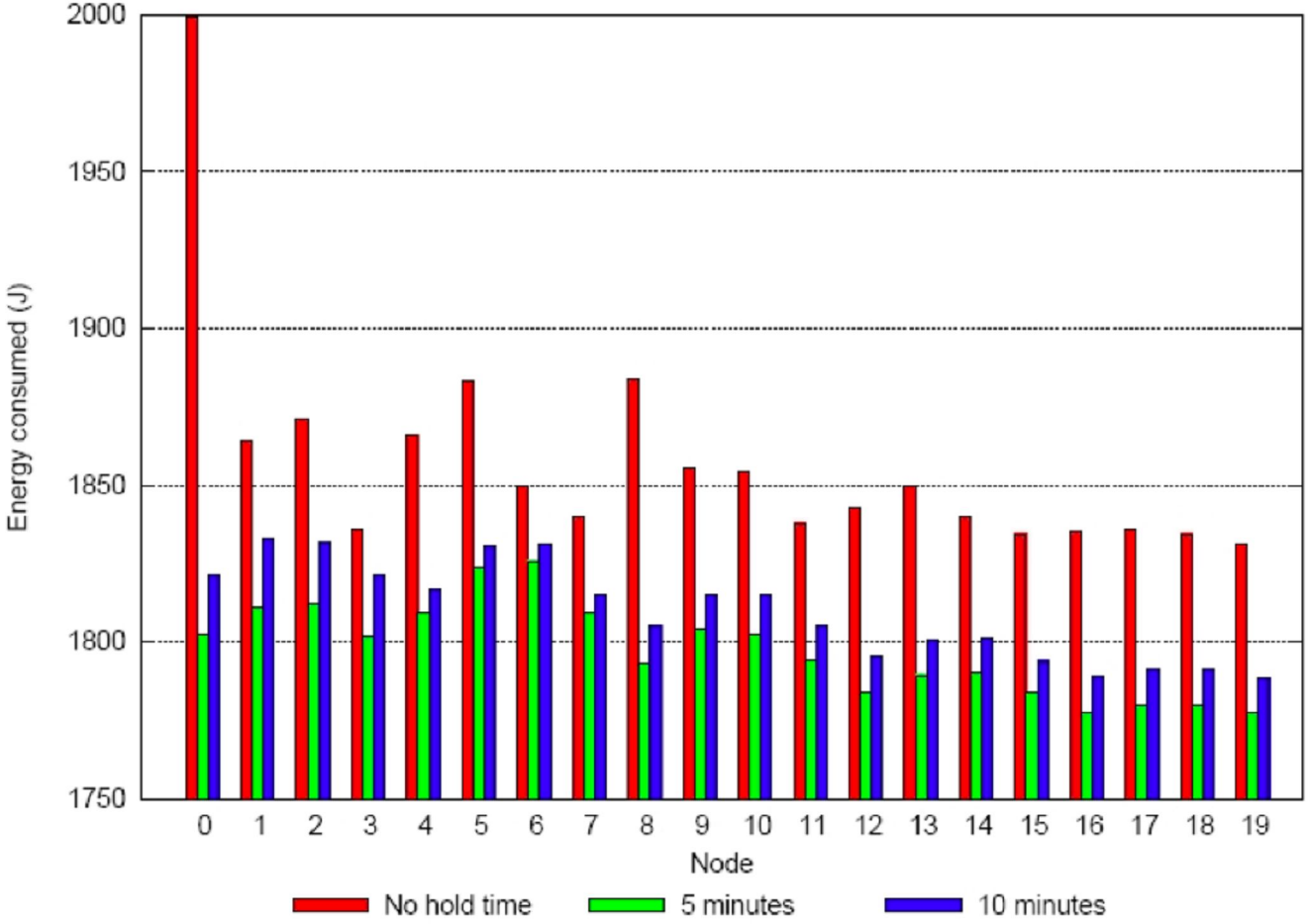
- Area of 100 m x 50 m
- 20 nodes
- 8 possible servers
- Movement models: Grid & RPGM
  - RPGM: Reference Point Group Mobility
- MAC protocol: 802.11g
- Routing protocols: AODV, DSDV
- Transmission range: 20 m
- Server-generated flow: 200 bytes every 50 ms
- Client-generated flow: 40 bytes every 300ms
- Server hold time:
  - no hold time (C/S architecture)
  - 5 minutes, 10 minutes
- Simulation duration: 2 hours



Grid topology

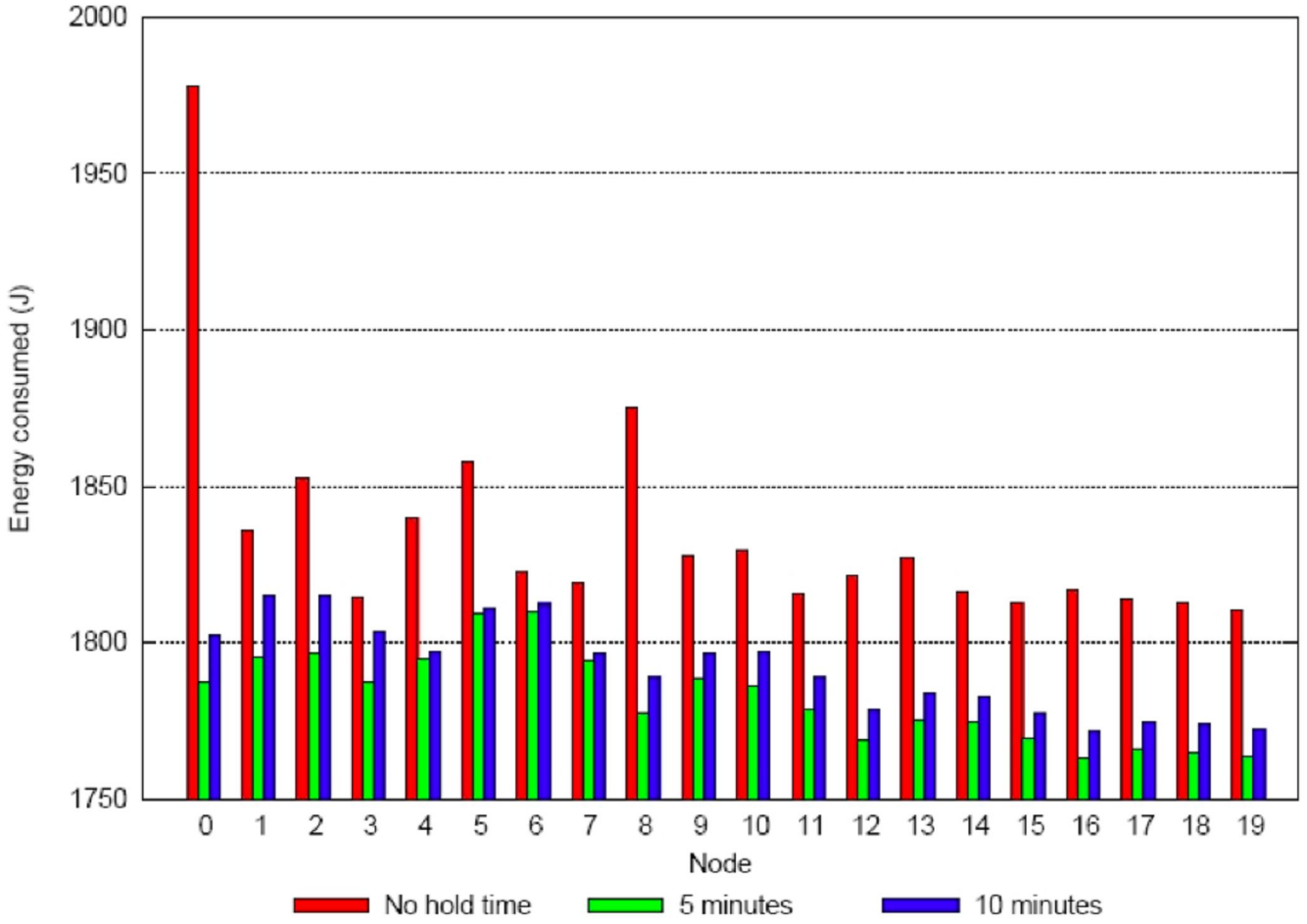
Preliminary Results

- Energy consumption with AODV and Grid model

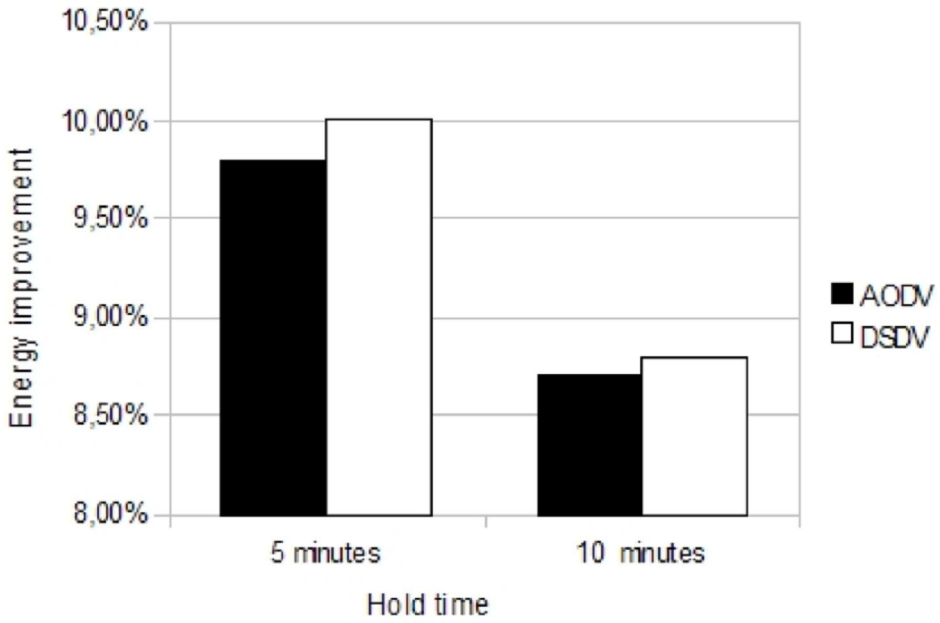


Preliminary Results

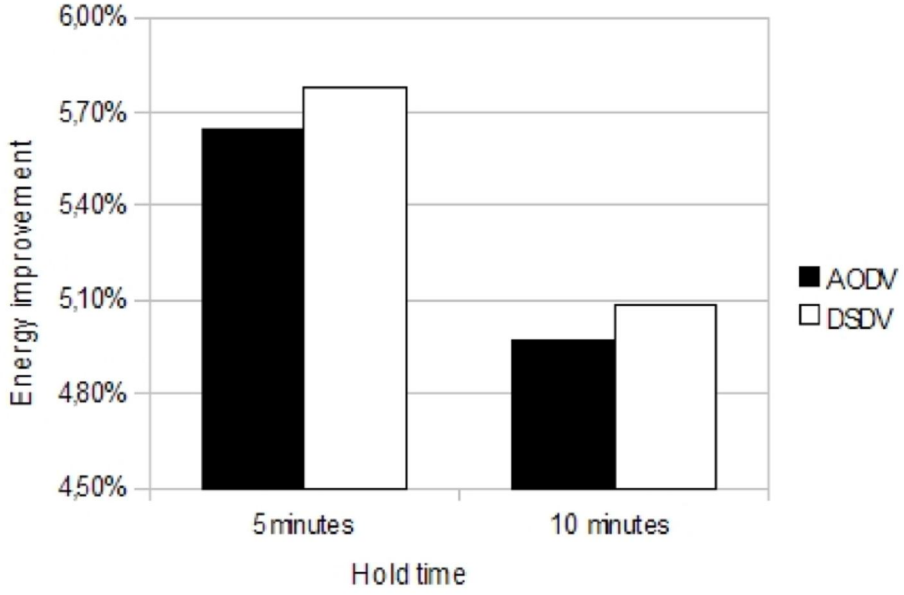
- Energy consumption with DSDV and Grid model.



Preliminary Results



Energy improvement of backup server solution with respect to a traditional scheme; Grid model



Energy improvement of backup server solution with respect to a traditional scheme; RPGM model

What is OnLive and why is it important?

- Gaming in the cloud
- Thin client, no special hw requirements
  - PC, Mac, OnLive mini-console
- Game video streamed to client
- Importance:
  - Allows playing AAA games on simple devices
  - Provide access to legacy games on next-gen consoles without hardware compatibility

Goal of the study

- How does the magic of OnLive work?
- Study network traffic *turbulence* of games on OnLive
  - Packet size
  - Inter-packet time
  - Overall bitrate up and down
- Controlled variation of network parameters
- Different genres of games

Motivation

- Network operators to planning for capacity
- Building traffic models for simulators
- Traffic classification to identify thin-client game flows, up and downstream
  - Can allow for treatments that help performance
- Selected games of similar hardware requirements

Unreal Tournament III (2007)

First-person shooter

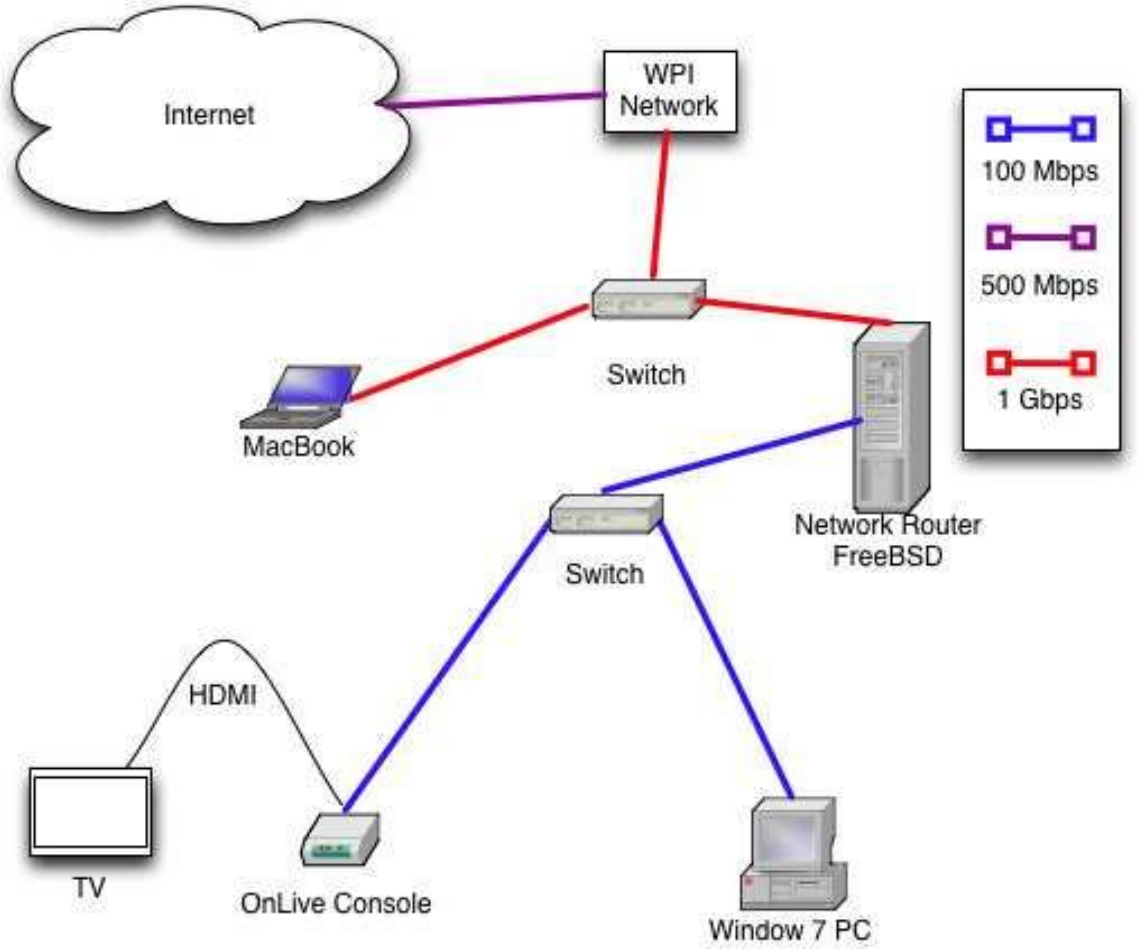
Batman: Arkham Asylum (2009)

Third-person action-adventure

Grand Ages: Rome

Real-time strategy (2009)

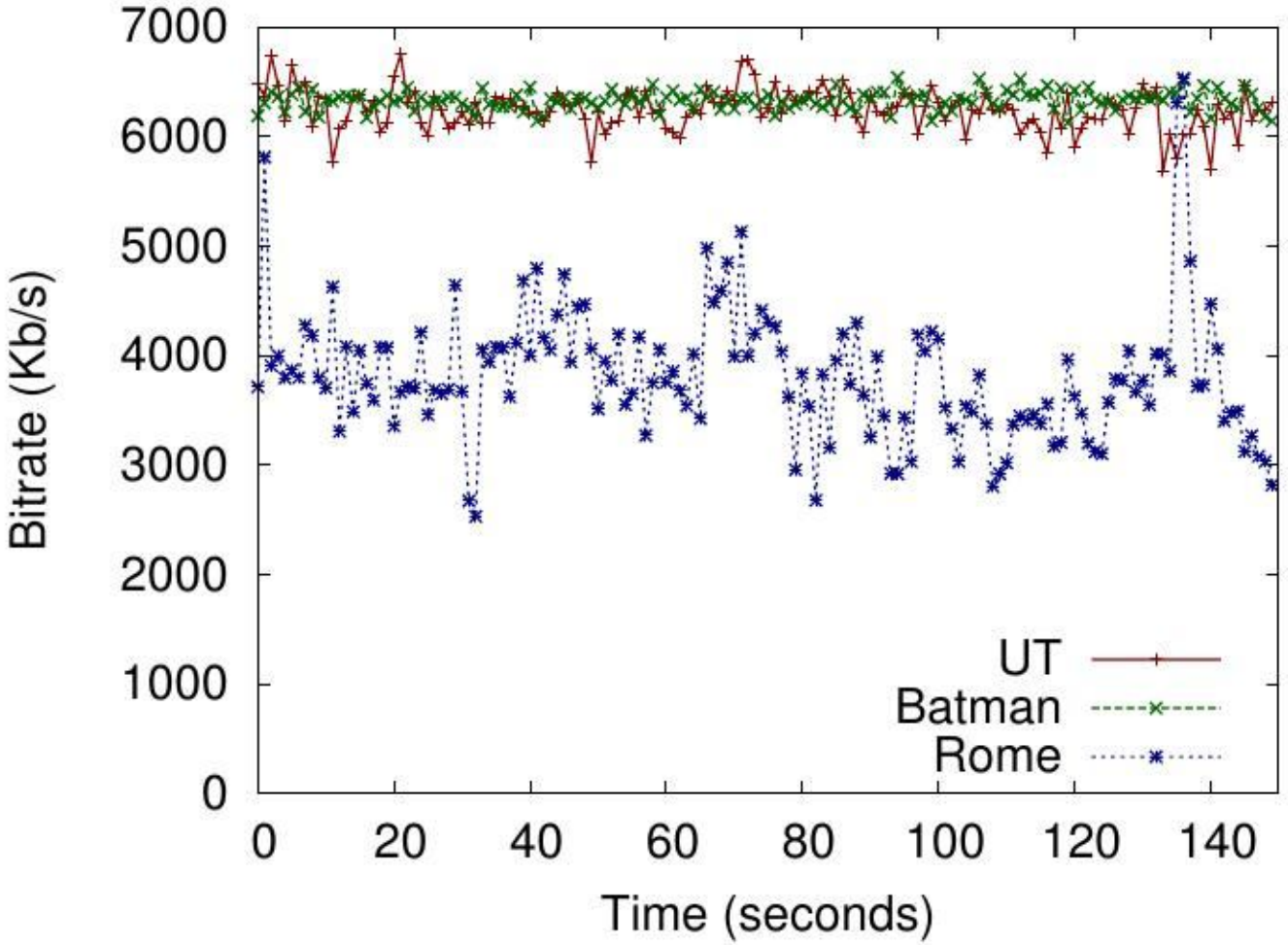
Experimental set-up



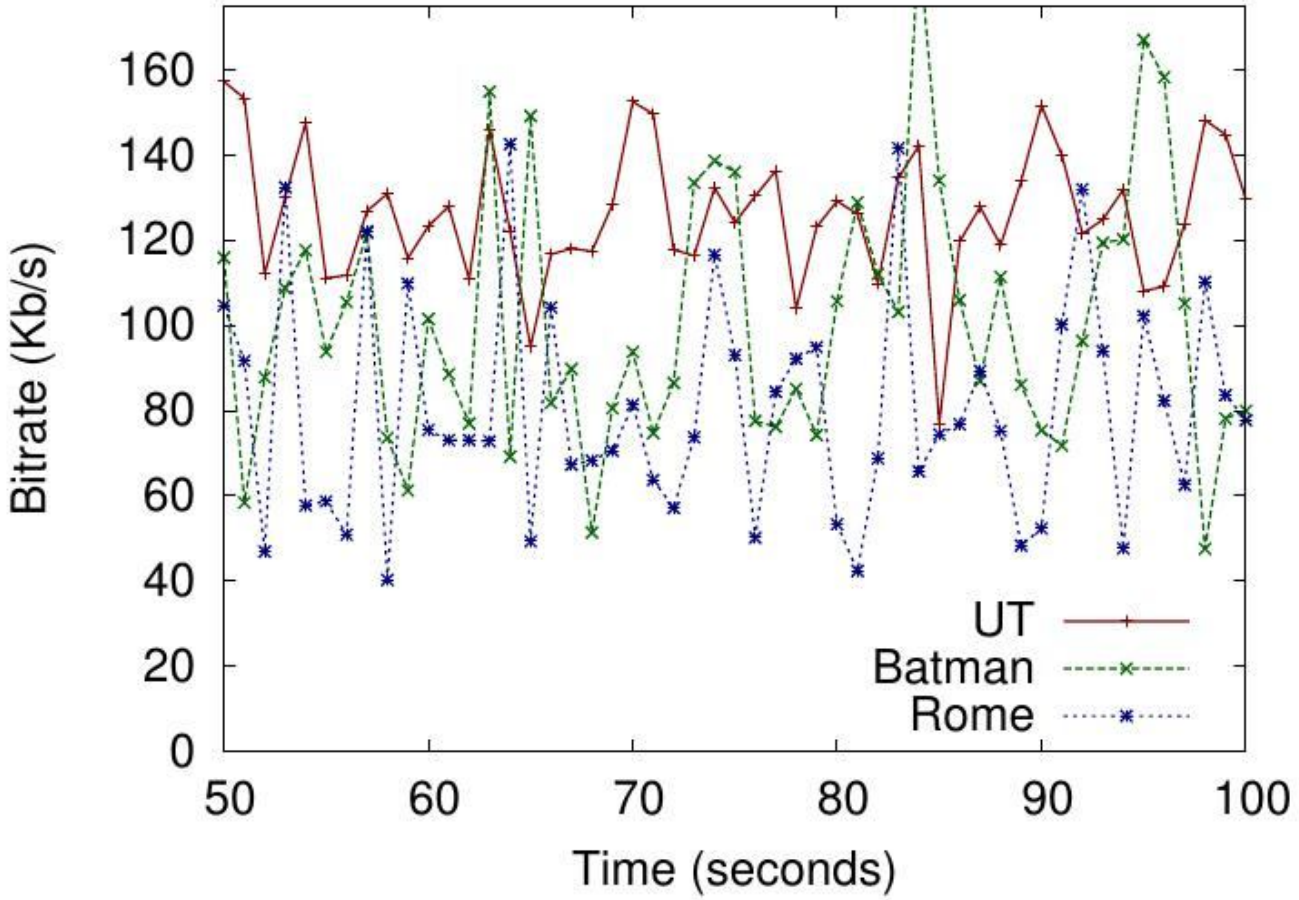
Design of experiments

- All traffic measured UDP
- Varied capacity, loss and latency
- Parameters:
  - Capacity (down:up) 5:1 Mb/s, 10:2 Mb/s, and unrestricted
  - Latency (round-trip) 0, 40, and 70 ms
  - Loss (downstream) 0%, 1%, and 1.5%
  - Iterations: 2.5 minute game runs, 3 iterations for each experiment, following longer pilot studies.

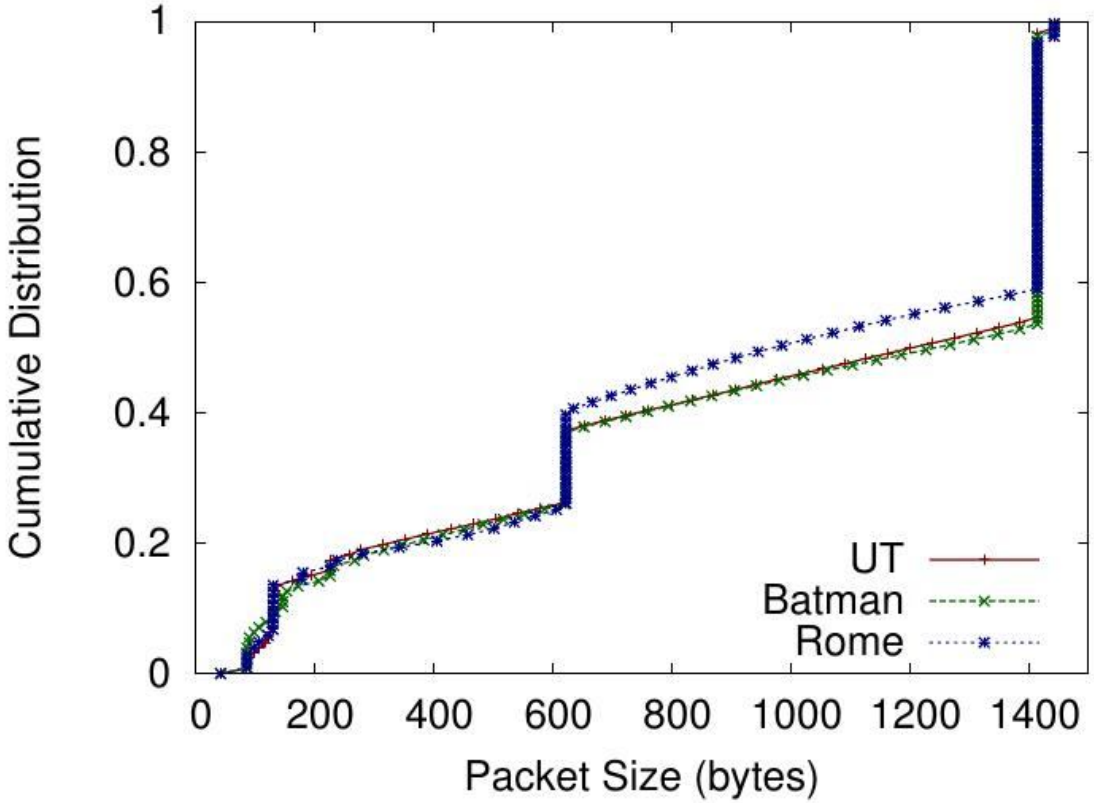
Downstream bitrate
unrestricted



Upstream bitrate
unrestricted



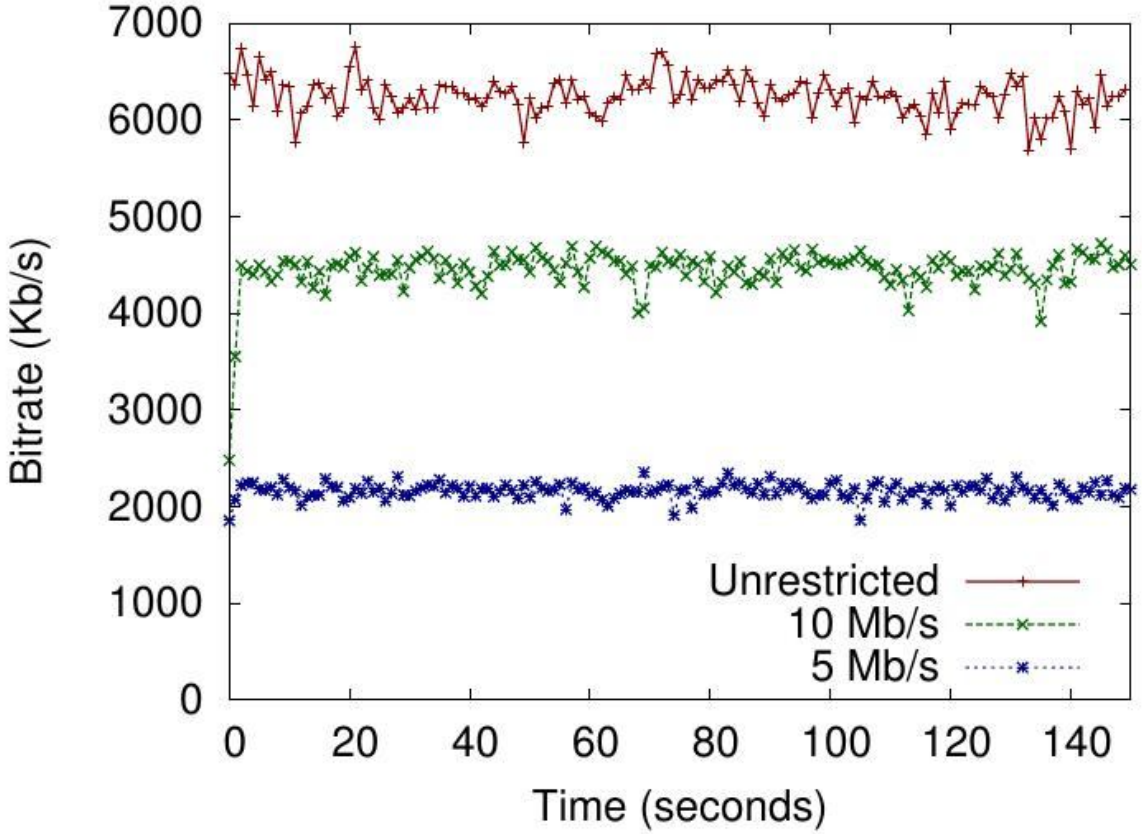
Downstream packet size
unrestricted



- About 30% of pkts are 1414B (smaller than MTU)
- About 10% of pkts are 622B
- About 60% of pckts are distributed fairly uniformly in [100, 1400]B

Downstream bitrate (UT only)

Capacity restriction



- Bitrate responds to restriction in capacity. In each of the restricted cases, the bitrate is about half of the restricted capacity
- Tests with increase in latency did not affect bitrate

Measured frame rates on PC (UT only)

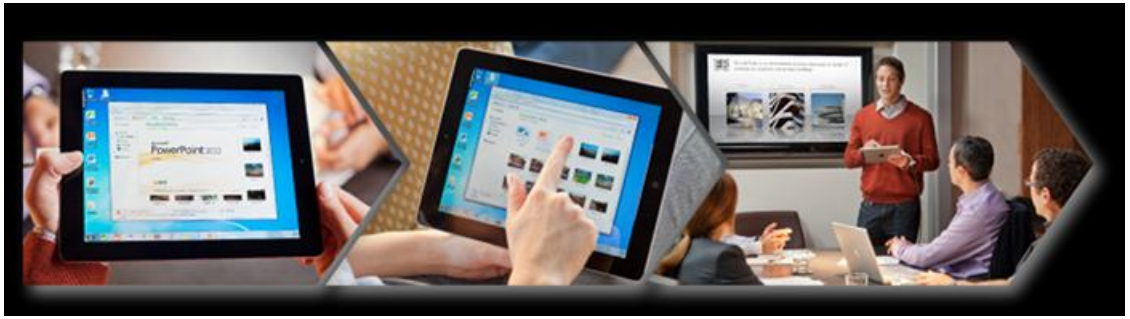
Experiment	Frame Rate
Unrestricted	60 fps
40 ms latency	60 fps
70 ms latency	60 fps
1% loss	30 fps
1.5% loss	30 fps
10 Mb/s	30 fps
5 Mb/s	25 fps

- Other games have similar values
- Frame rate is adjusted in presence of loss and capacity variations, not in presence of latency changes

Conclusions

- OnLive games have high downstream bitrates, moderate upstream bitrates
- The characteristics of game traffic are similar for all genres tested
- Bitrates do not adapt to loss or latency, do adapt to capacity limits.
- Frame rates adapt to both capacity limits and loss, but not to latency.

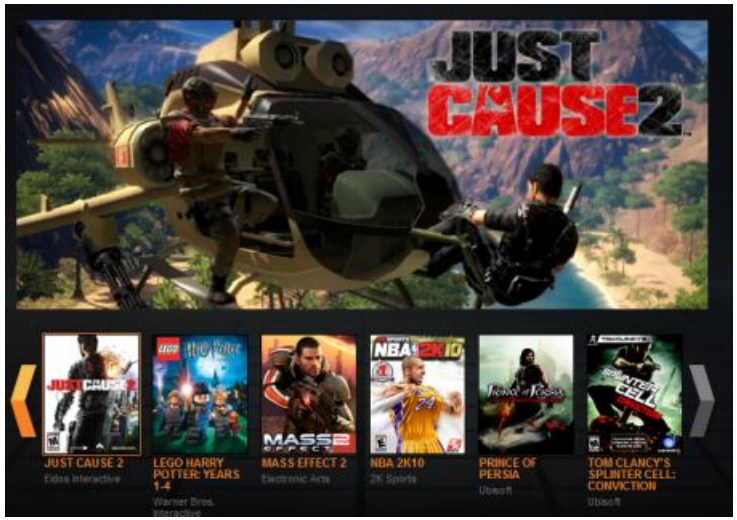
Project idea: traffic analysis



- OnLive Desktop

- OnLive for Tablets (iOS and Android)

Project idea



- study of additional OnLive games

Comparison of OnLive, GaiKai, other thin client systems

onLive

GAIKAI™