# Donnybrook: Enabling Large-Scale, High-Speed, Peer-to-Peer Games


Ashwin Bharambe
Jeffrey Pang
Srinivasan Seshan
Xinyu Zhuang

Carnegie Mellon

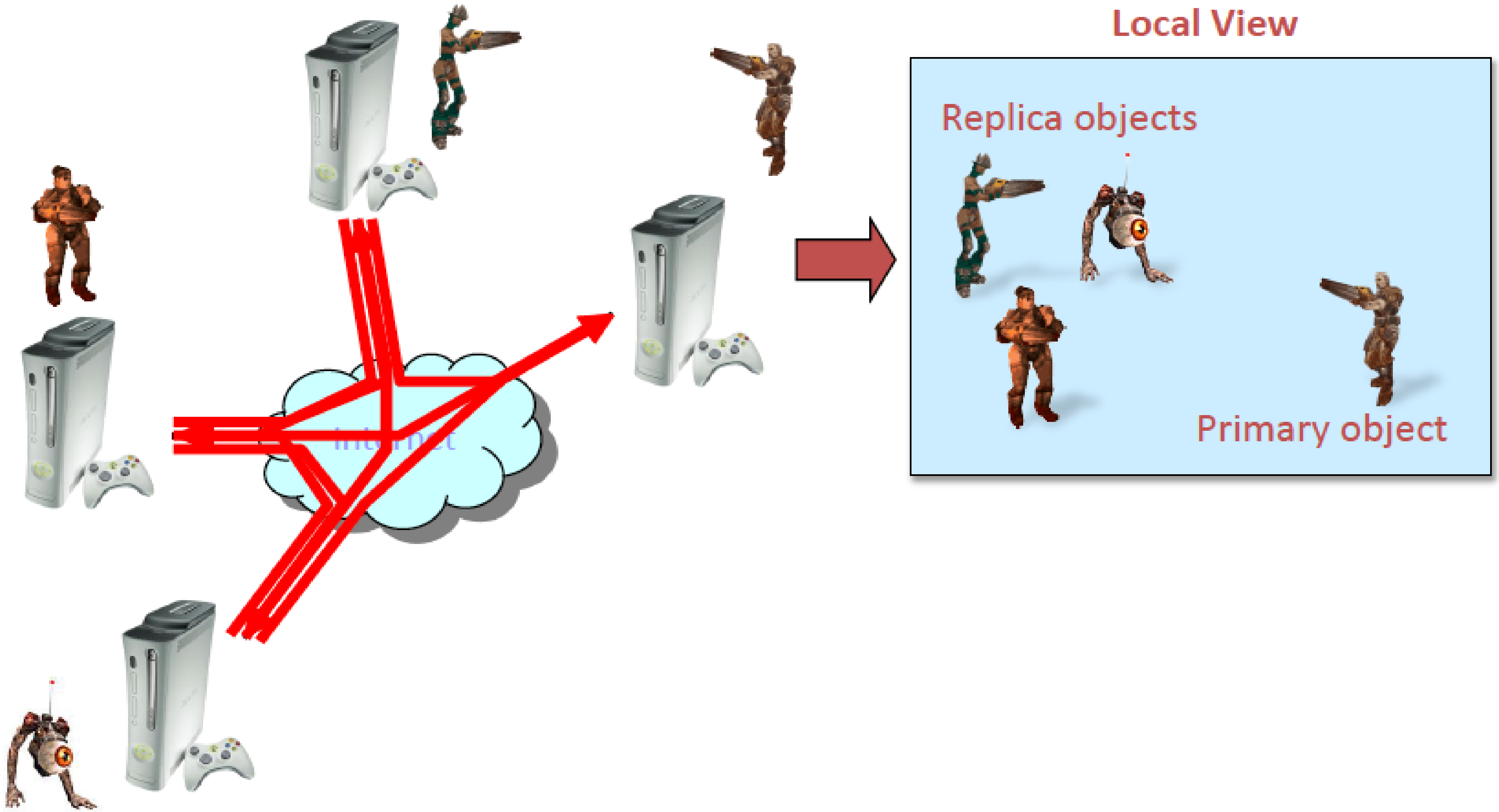
John R. Douceur
Jacob R. Lorch
Thomas Moscibroda

Microsoft Research

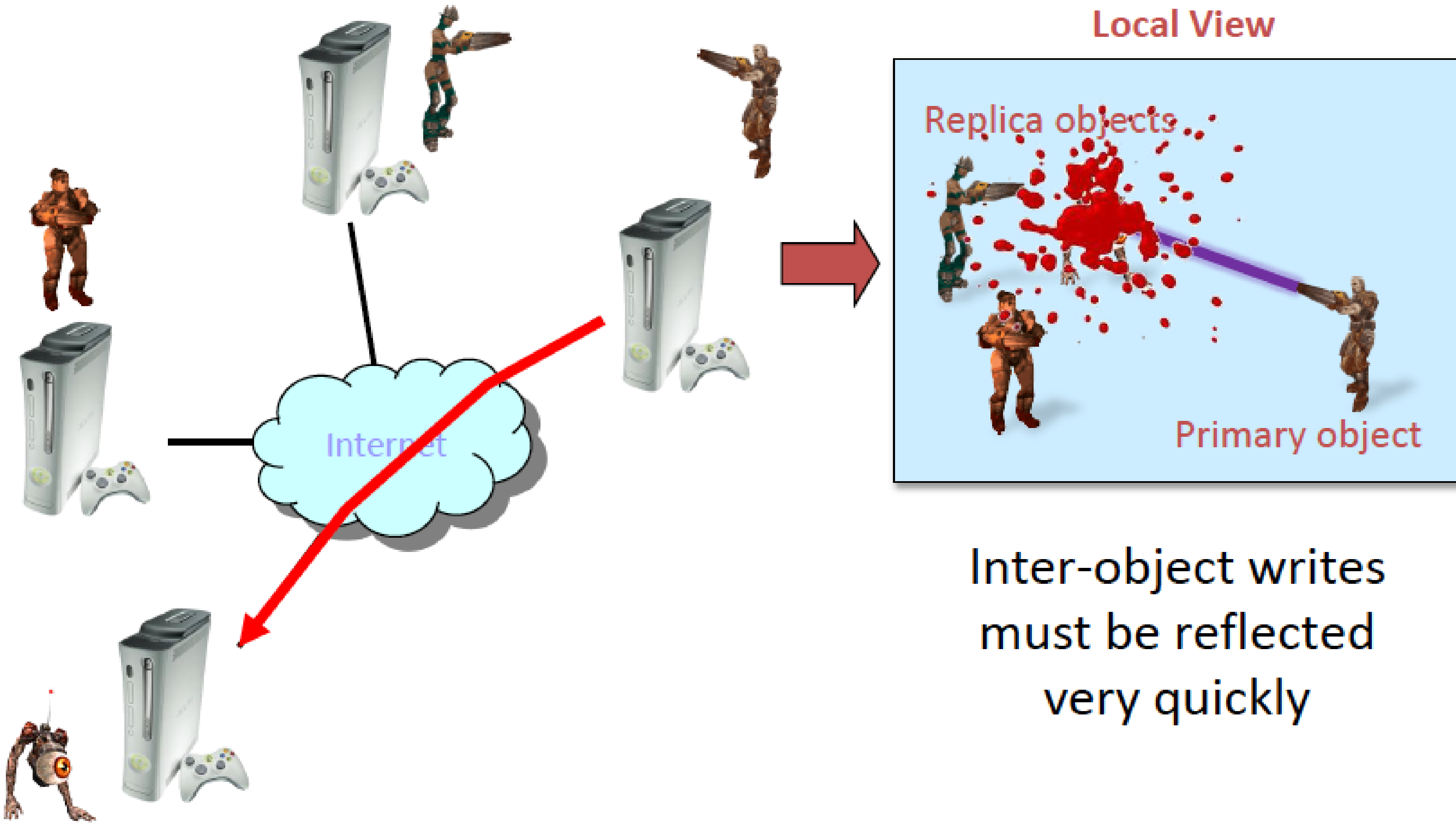

Presenter: Ercan Ucan

*Some of the slides are borrowed with permission from J. Pang*

# Peer-to-Peer(P2P)

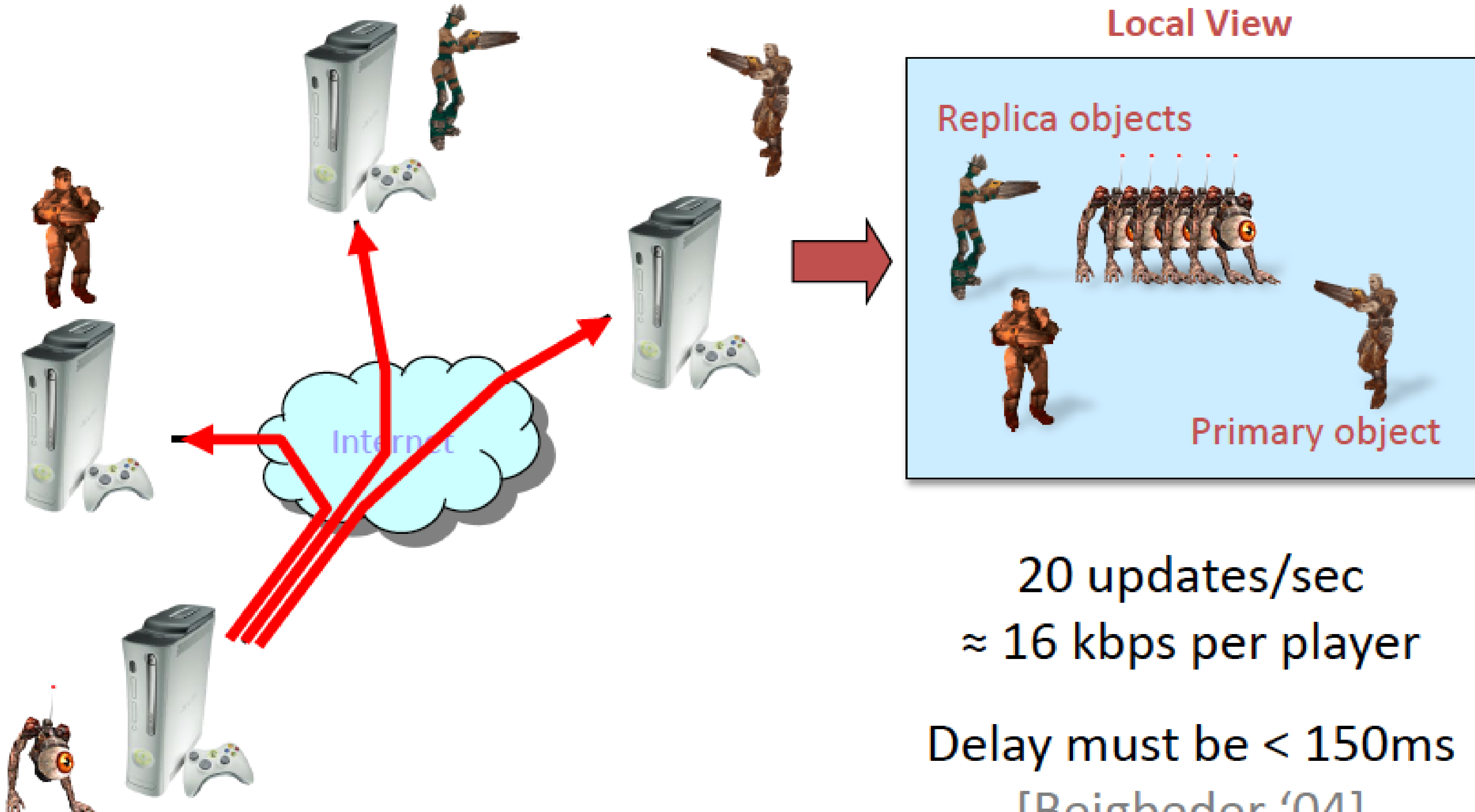


# High-Speed




Inter-object writes must be reflected very quickly

# High-Speed



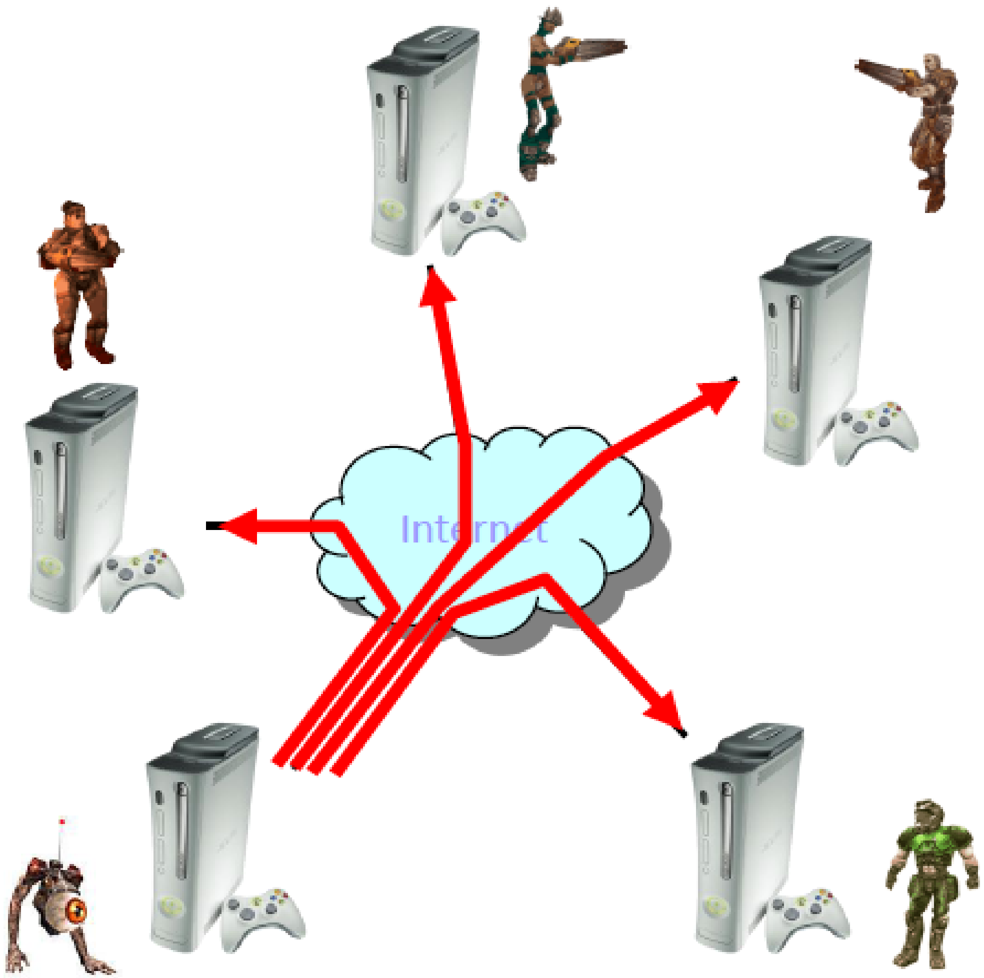

20 updates/sec

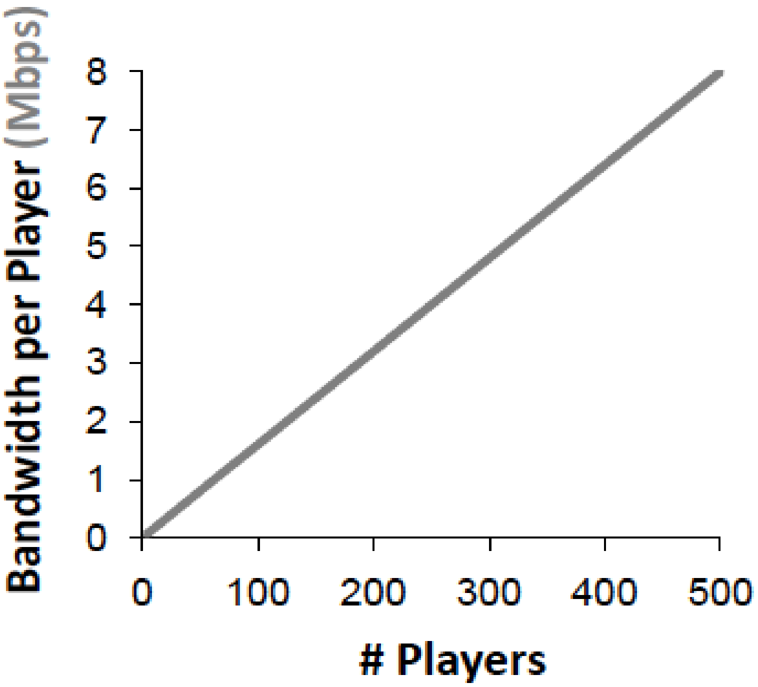
≈ 16 kbps per player

Delay must be < 150ms

[Beigbeder '04]

# Large-Scale


Internet


Bandwidth per Player (Mbps)
0
1
2
3
4
5
6
7
8
0
100
200
300
400
500
# Players

# Challenge:

- Many console games are peer hosted to save costs

- Limits high-speed games to 32 players

- Large scale, high-speed, peer management

Challenge: How to achieve all 3?

- No gaming architecture does it yet!

# 3 problems in peer managed games

- Insufficient capacity

  Key Limitation is upload capacity

- Resource heterogeneity

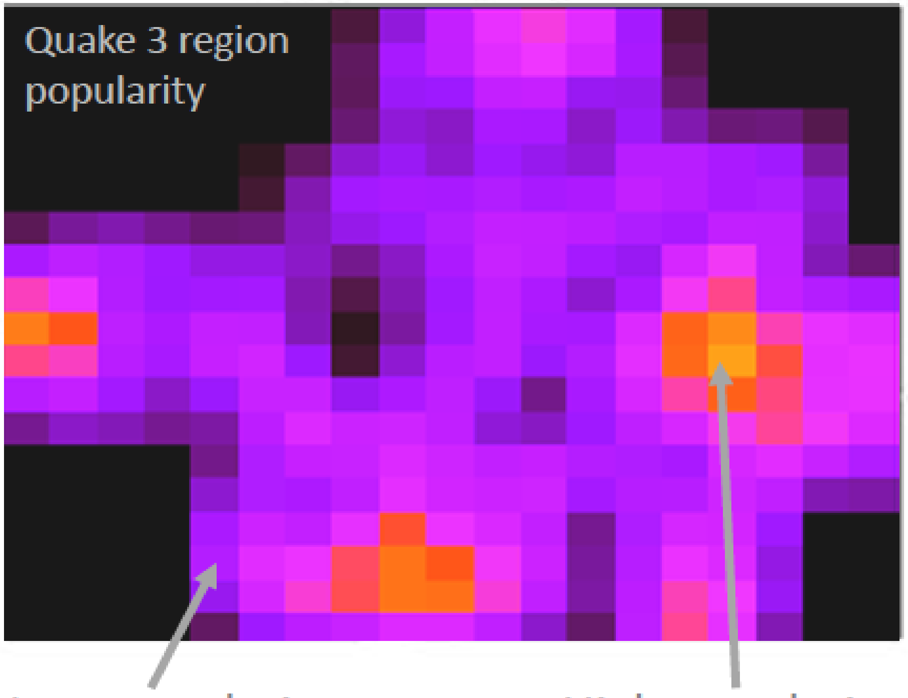
  How to schedule sending messages?

- Interest heterogeneity

  Leverage spare upload capacity to help forward updates and keep 100-150 ms latency

# Area-of-Interest(AOI) Filtering

- Only receive updates from players in your AOI
  - Colyseus [Bharambe '06]
  - VON [Hu '06]
  - SimMUD [Knutsson '04]
- Problems:
  - Open-area maps, large battles
  - Region populations naturally follow a power-law [Bharambe '06, Pittman '07]




**Requirement**: ~1000 players in *same* AOI

# Motivation and Goals

- Donnybrook: Interest Sets
  - Reduces mean bandwidth demands
- Donnybrook: Update Dissemination
  - Handles interest and bandwidth heterogeneity

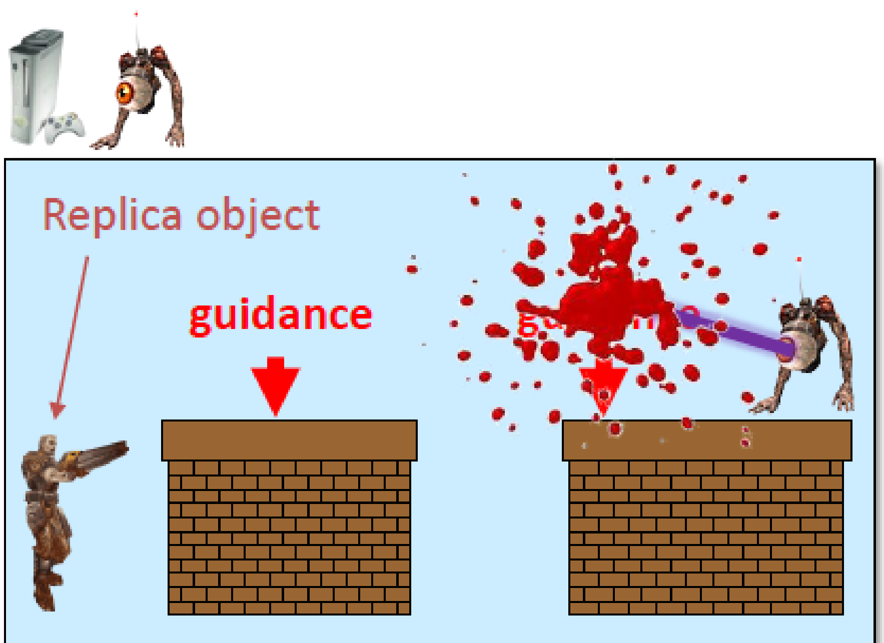
- **Intuition**: A human can only focus on a constant number of objects at once [Cowan '01, Robson '81]
  - Only need a constant number of high accuracy replicas

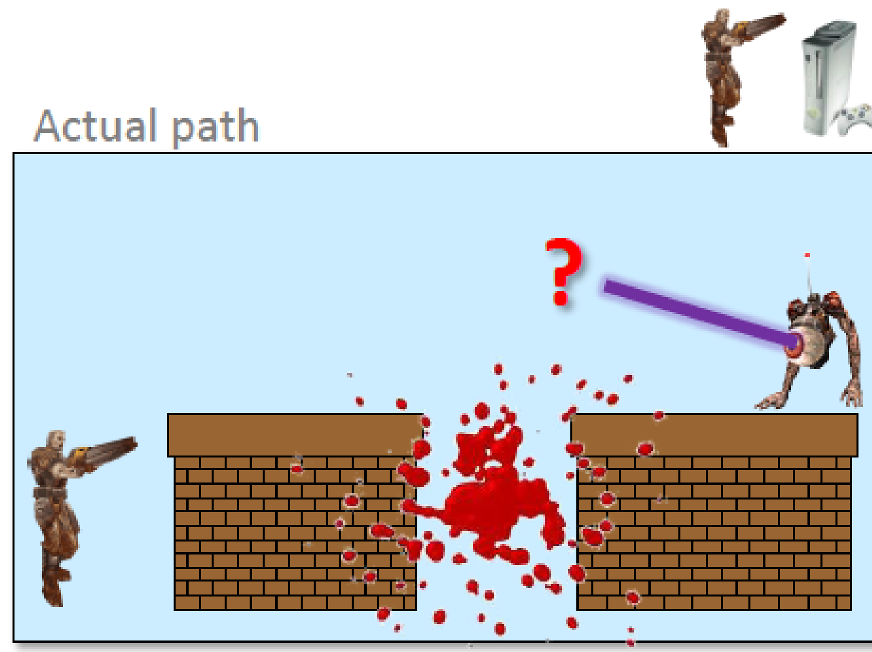

- **Interest Set**: The 5 players that I am most interested in
  - *Subscribe* to these players to receive 20 updates/sec
  - Only get 1 update/sec from everyone else

# Smoothing Infrequent Updates

- Send *guidance* (predictions) instead of state updates
- *Guidable AI* extrapolates transitions between points
  - E.g., game path-finding code

- **Problem**: Predictions are not always accurate
  - Interactions appear inconsistent
  - Jarring if player is paying attention

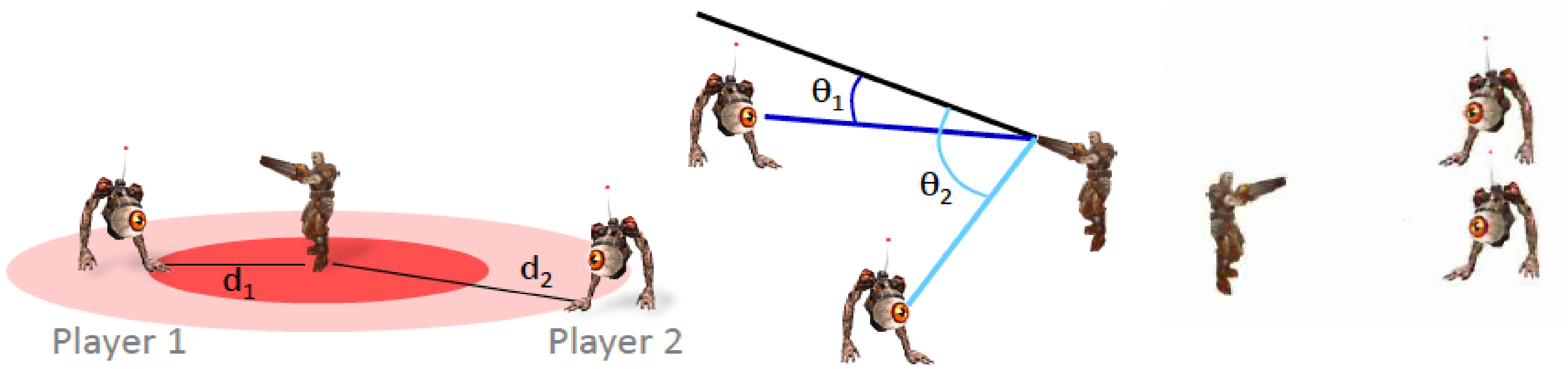


# Interest Sets

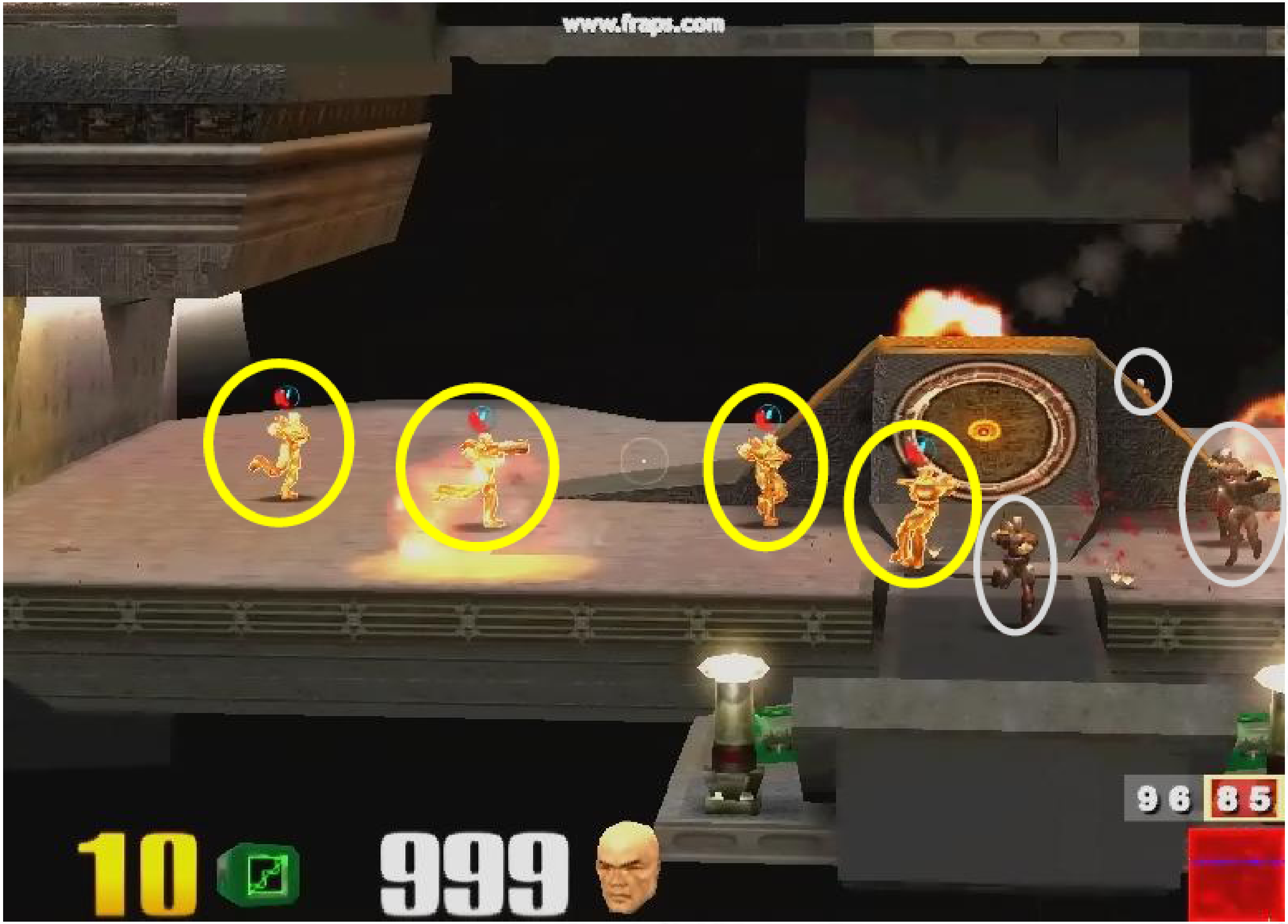
- How to estimate human attention?
  - Attention(**i**) = how much I am focused on player **i**

$$\text{Attention}(\mathbf{i}) = f_{proximity}(d_i) + f_{aim}(\theta_i) + f_{interaction\text{-}recency}(t_i)$$




www.fraps.com
Not in Interest Set
= Interest Set
11
999
90
79

# Interest Sets: Weights


www.fraps.com
96
85
10
999

# Dissemination (Main requirements)

- Strict delay bound (150ms)
- Frequent membership changes (68% turnover/sec)
- Bandwidth heterogeneity
- Many overlapping groups
  - Previous overlay multicast:
- Unstructured [Narada, NICE]: Hard to meet 2 and 4
- Structured [Splitstream]: Hard to meet 1 and 3

**Problem**: *subscriber*-initiated tree construction needs lots of coordination overhead or is inflexible

# Randomized *source*-initiated tree construction

- Well connected peers join *forwarding pool*
  - Based on relative bandwidth and latency thresholds
- These nodes advertise their forwarding capacity
  - Piggy-backed on low freq. updates
- Sources randomly pick enough forwarders to satisfy needs each frame
  - Avoids need for coordination
  - Fixed tree depth to bound delay

# Dissemination

- Main requirements:

- Strict delay bound: constant tree depth
- Freq. membership changes: uncoordinated tree construction
- Bandwidth heterogeneity: high bandwidth forwarding pool
- Many overlapping groups: shared forwarding resources

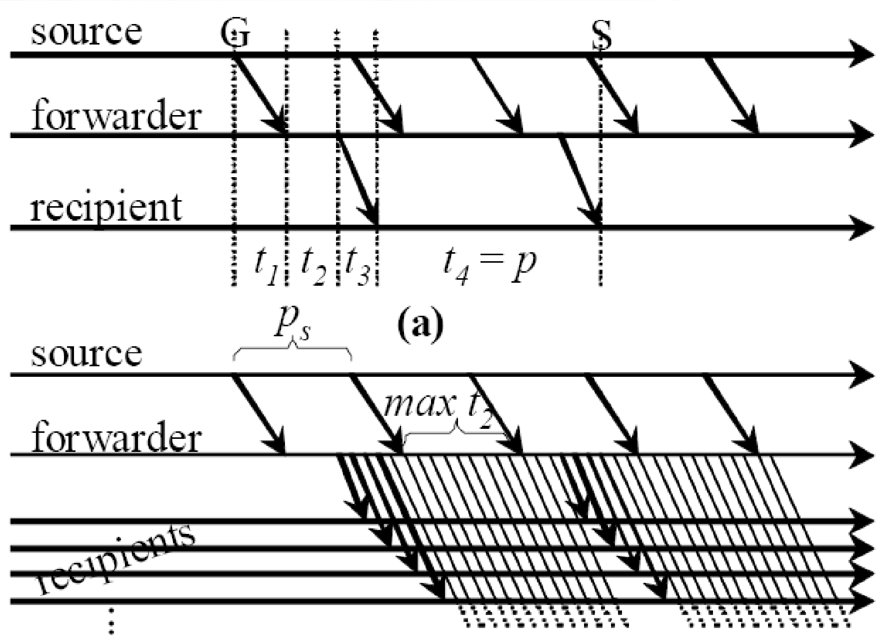
Trade-off: If too many sources pick the same forwarder then the forwarder must drop some updates--Leave some headroom (advertise only ½ forwarder capacity)

drops happen rarely and only cause loss for 1 frame(5-10% loss is OK [Beigbeder '04])

# Guidance Forwarding

- Every player needs guidance from every other once a sec
- Non-forwarding pool players contribute spare bandwidth to forwarding guidance
- Nodes coordinate to match sources to forwarders (configuration changes rarely)
- Sources send fresh guidance to a forwarder once a frame
- Forwarders stagger guidance to avoid queuing delay

➸ Ensures all recipients get guidance at most 1 frame old (plus transmission delay)



# User Study

- **LoBW-IS vs. LoBW**: 12 trials
- **LoBW-IS vs. HiBW**: 32 trials
- 88 total participants

How often did you play FPS games in the past?

| Every Day | Every Week | Less Often |
| --- | --- | --- |
| 62 % | 25 % | 13 % |

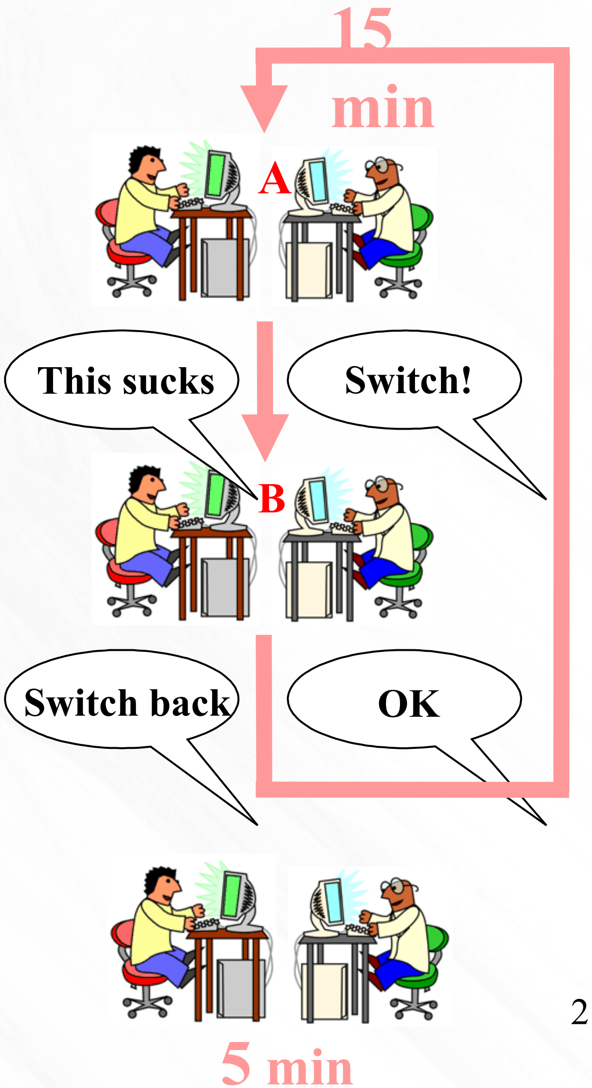
# User Study Procedure

- Before experiment, practice on HiBW
- Tell players two Quake III "servers" exist: A and B
- Start playing on A, can vote to switch to B

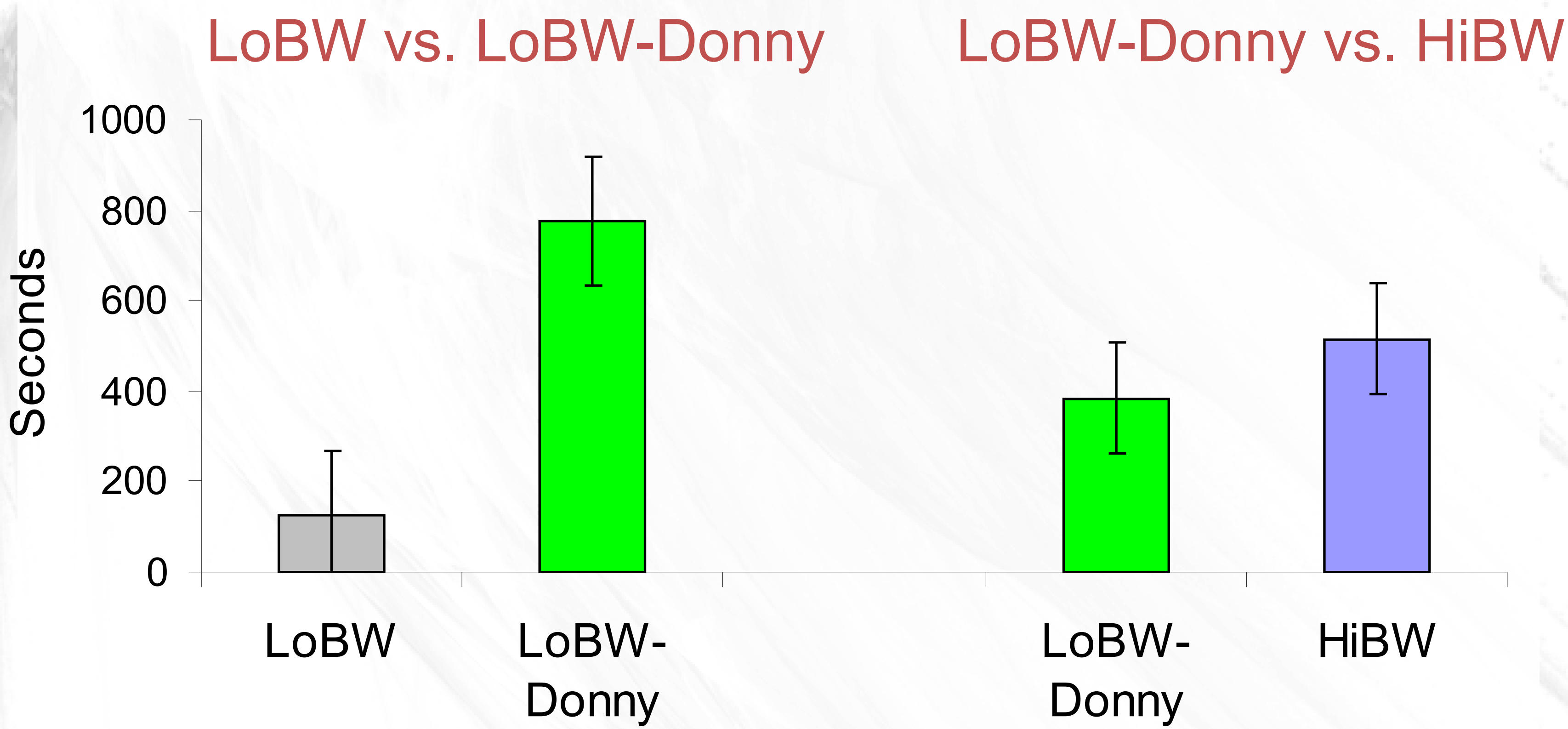
- When both players vote, game continues on B

- Can vote to switch back and forth
- Analog to how players choose game servers
(if good, stay, otherwise leave and try another)

- Play new game on least-used version so they can compare

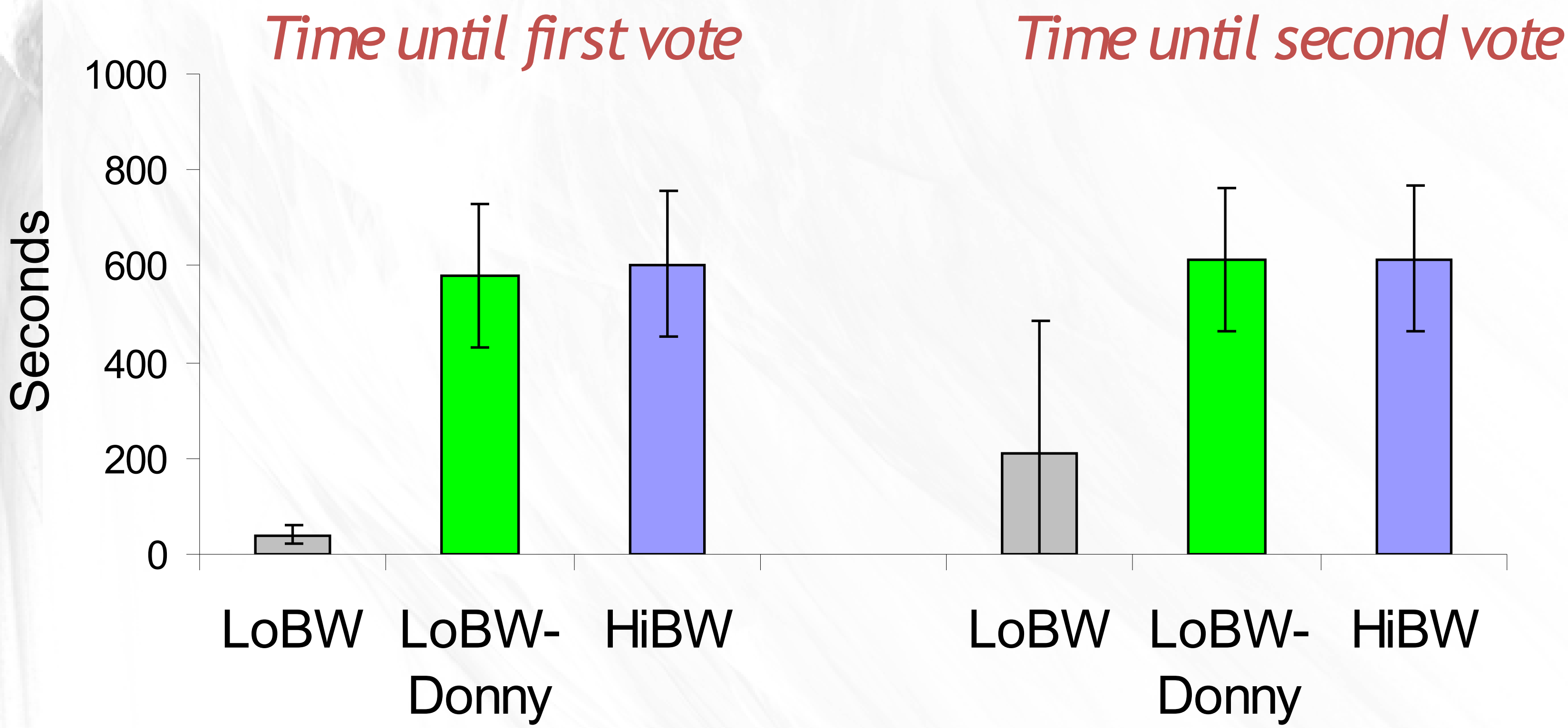


# User Study



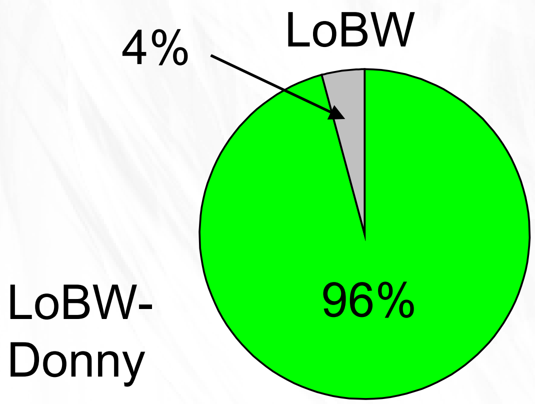

How long does a pair play on each version?

# User Study

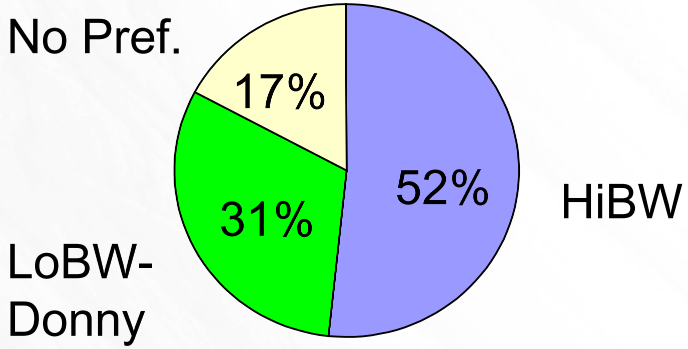



How long before a player wants to switch?

# User Study




*LoBW-Donny vs. LoBW*




*LoBW-Donny vs. HiBW*

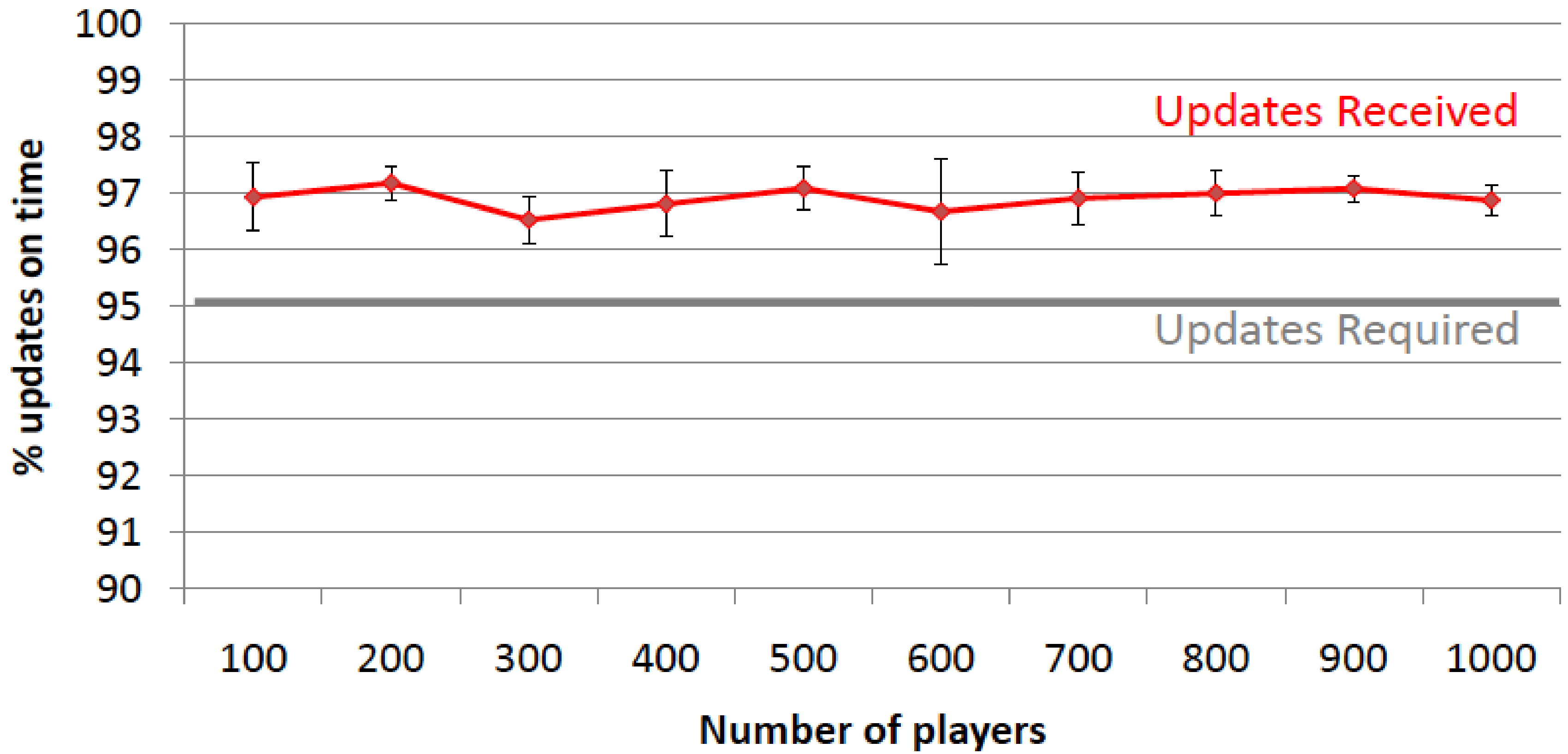
Survey: Was A or B more Fun?

# User Study: Limitations

- Only 2 human players
  - Tried to keep human subjects focus on each other
  - How well interest sets work on human players

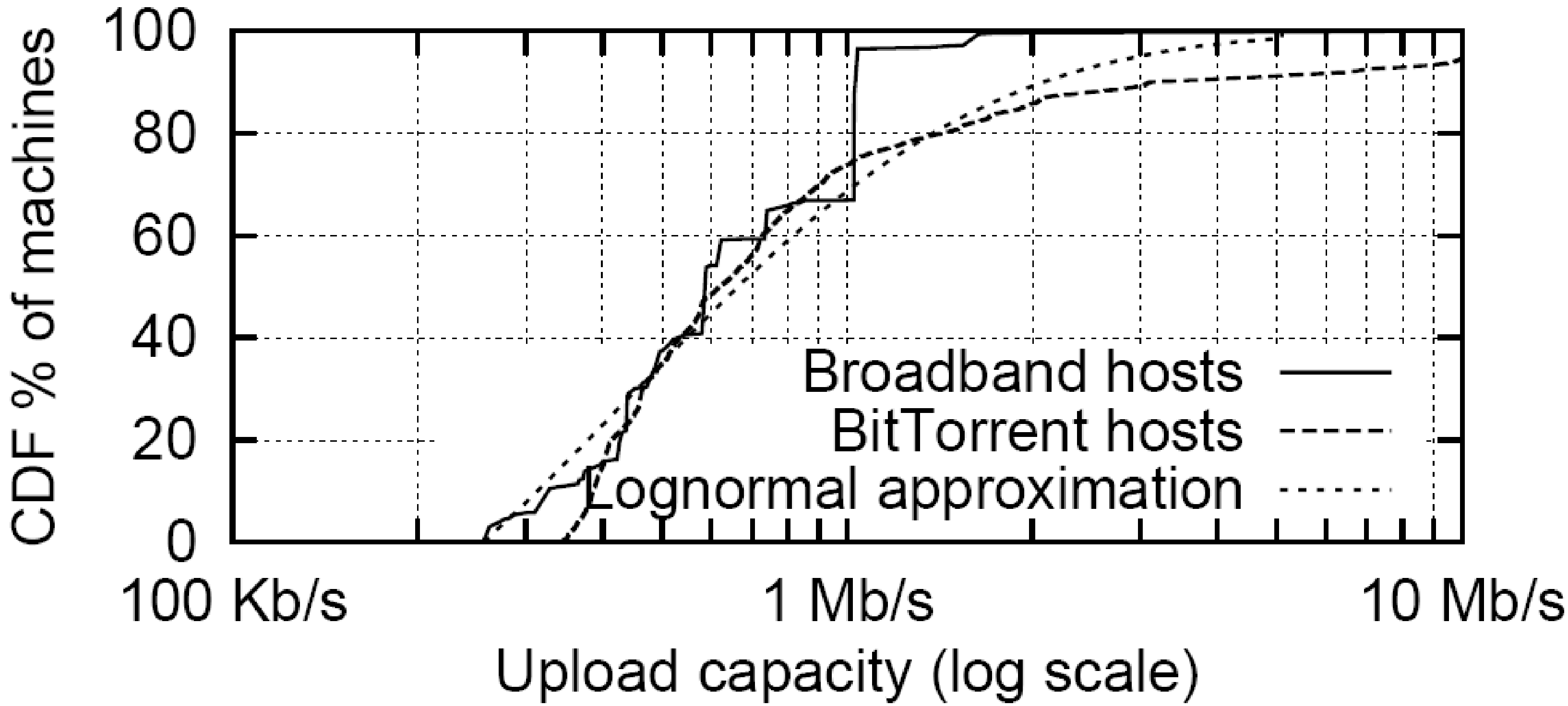
- Only 32 players in total
  - Human cognition does not change
  - Can estimate and tune the interest set size better.

# Evaluation: Updates on time



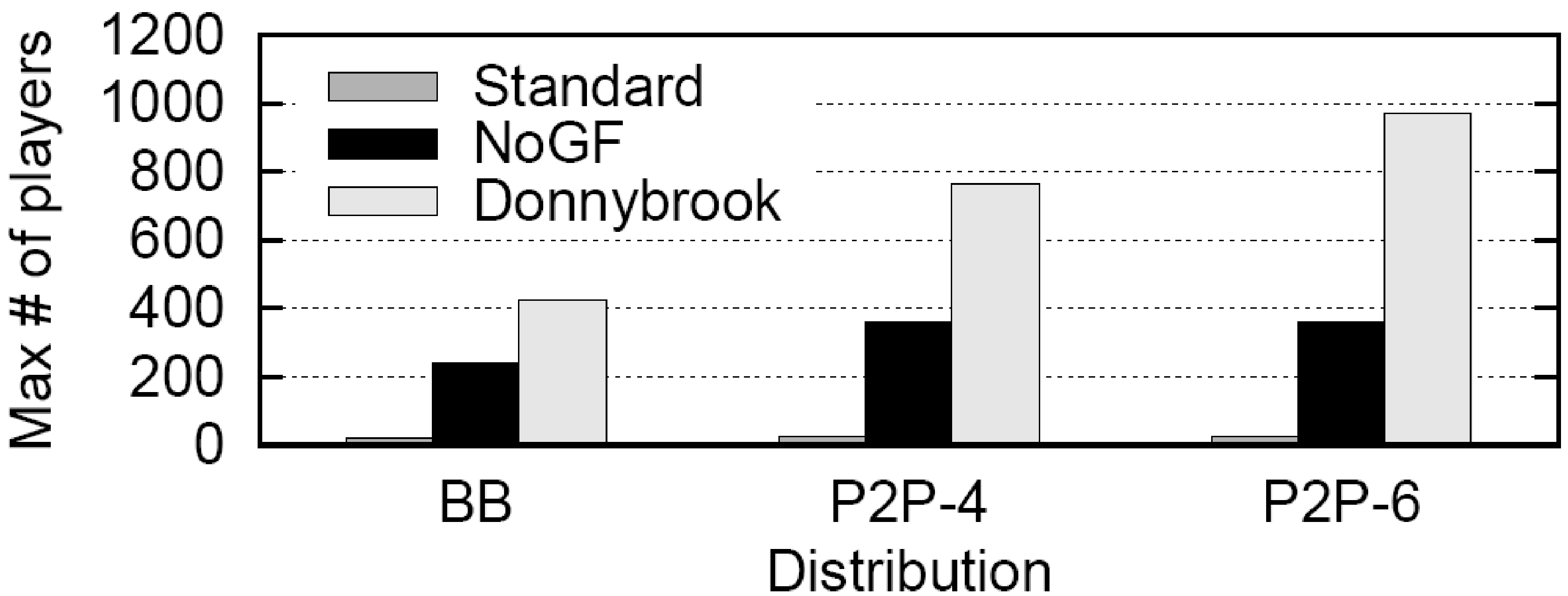

Enough updates are delivered on time at all scales

# Evaluation: BW Models



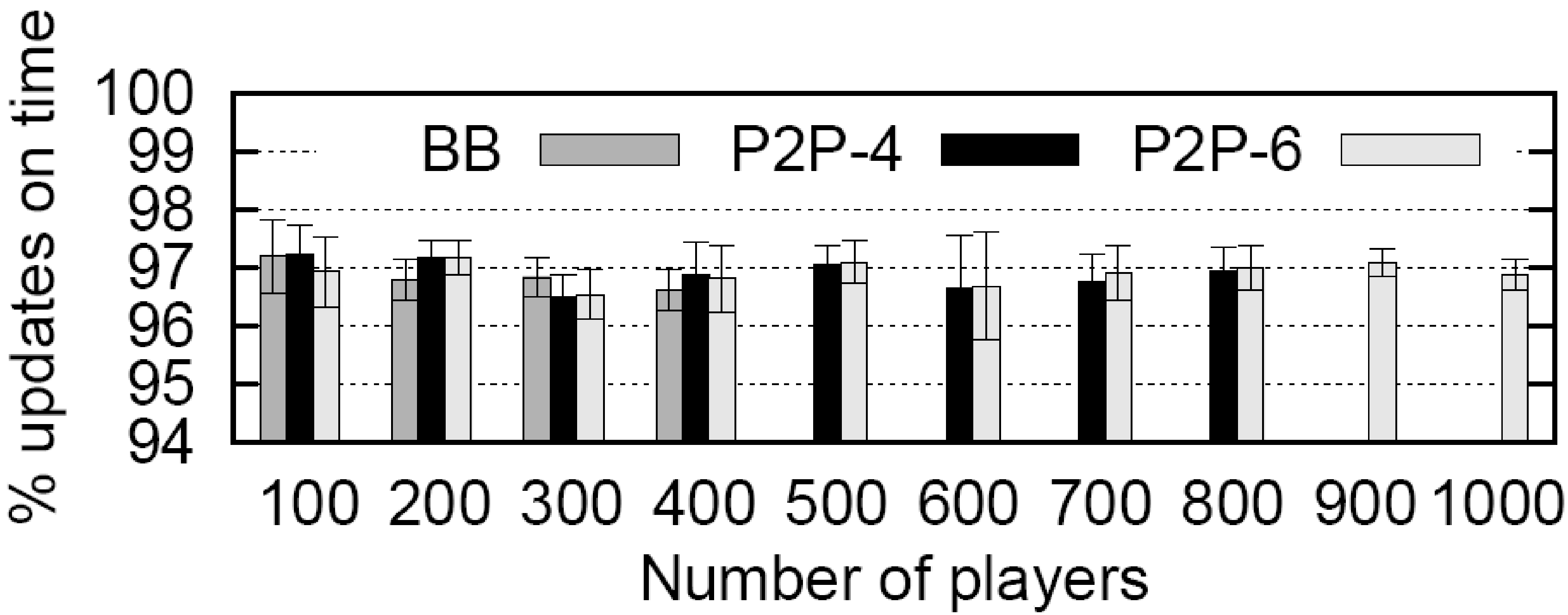

Most peers have < 768 kbps, some have much more

# Evaluation: Scale



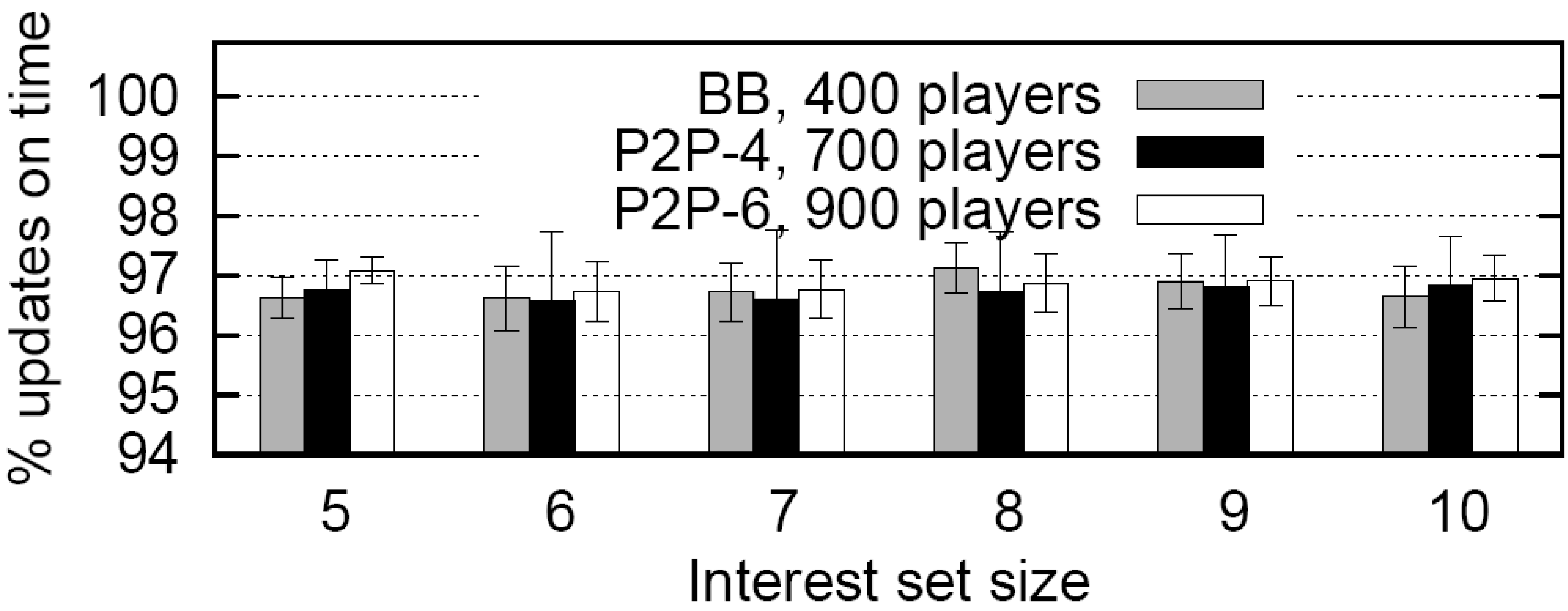

Donnybrook enables 100s of players in many BW models

# Evaluation: Updates on time



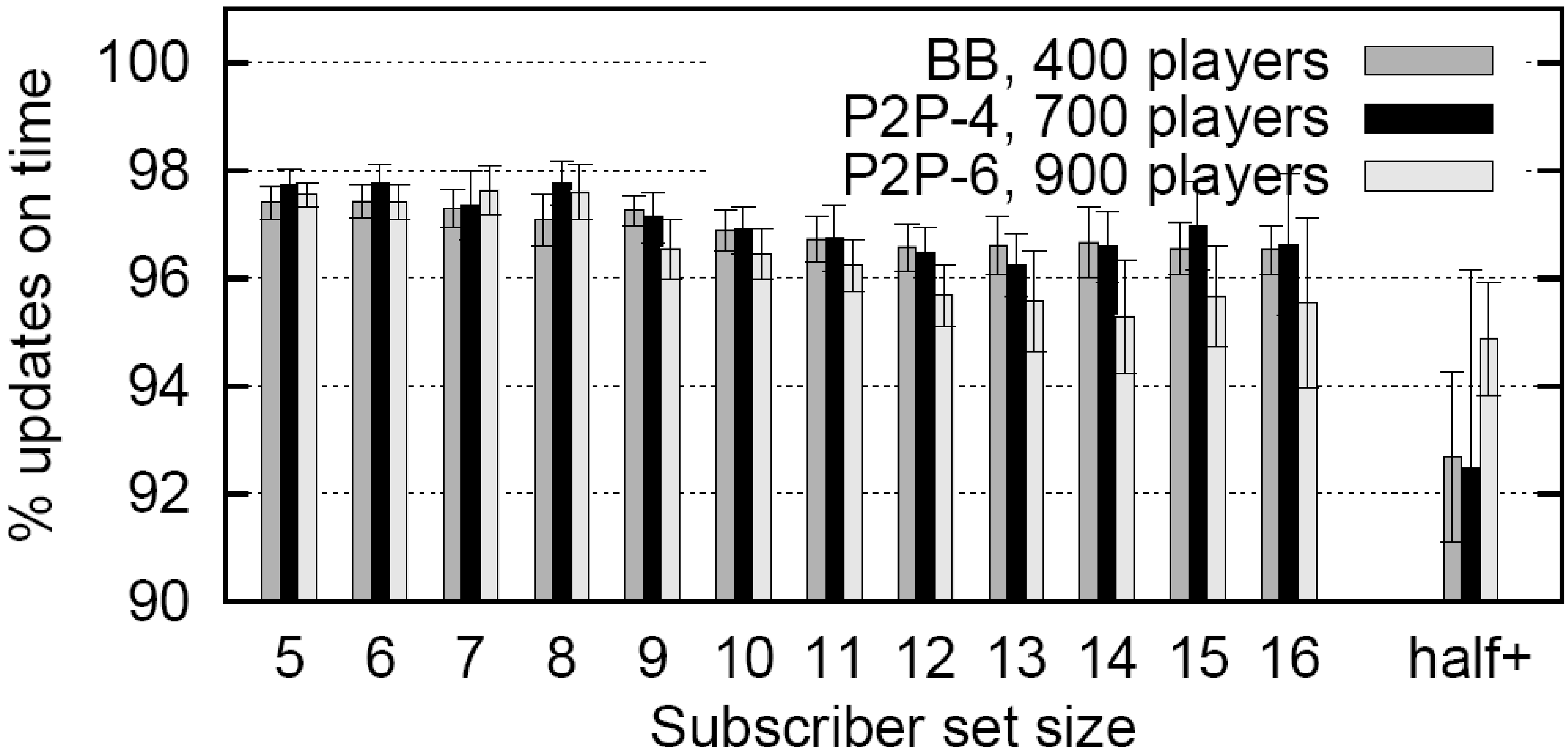

Enough updates are delivered at all supported scales

# Evaluation: Interest Set Size



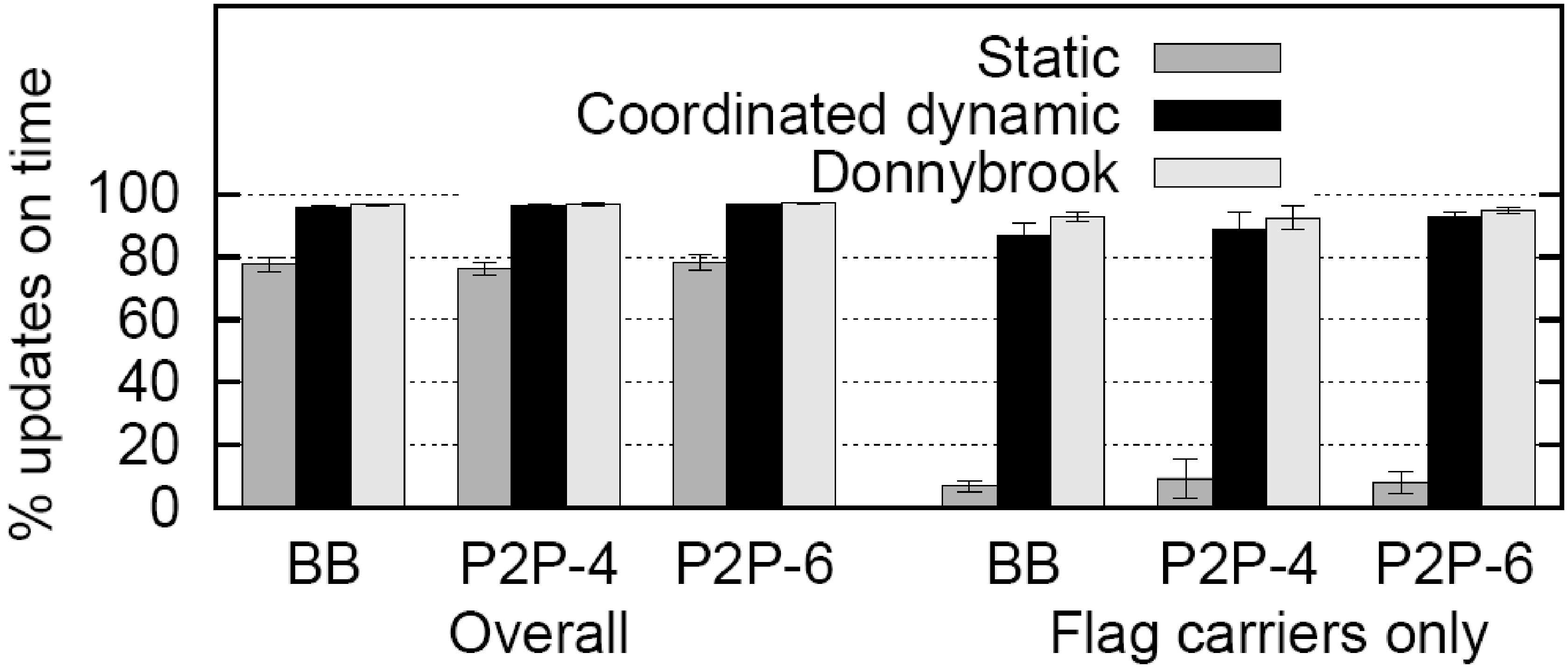

Performance is not sensitive to interest set size

# Evaluation: Subscriber Set Size



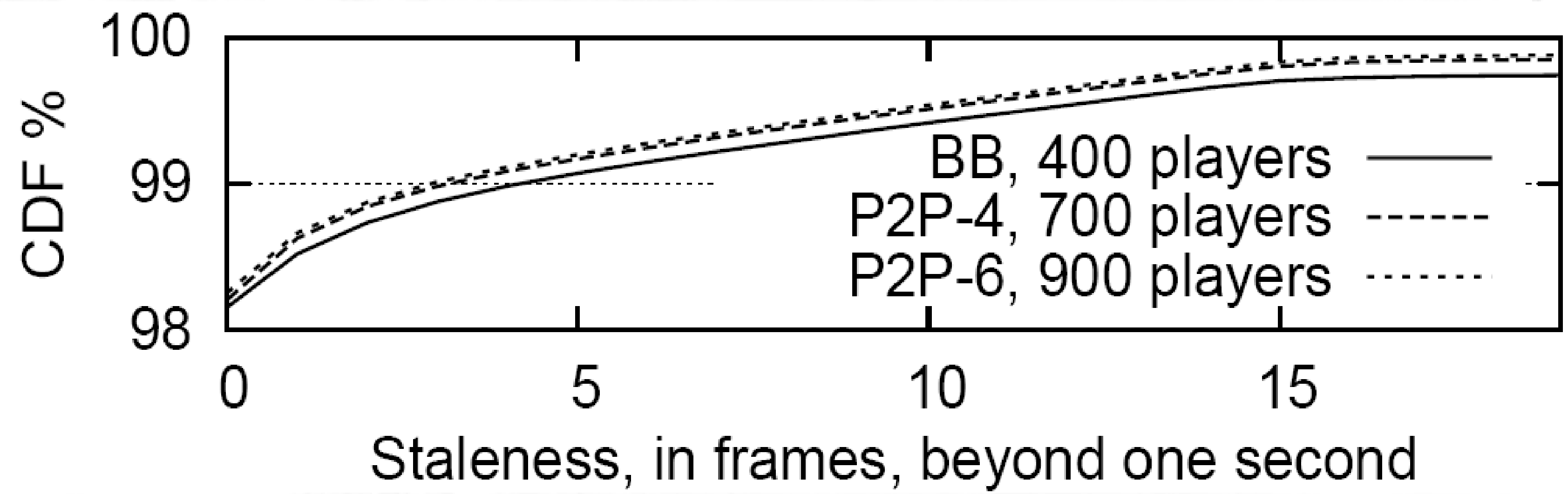

Players with lots of subscribers still deliver enough updates

# Evaluation: Other Approaches




Donnybrook performs better than other approaches

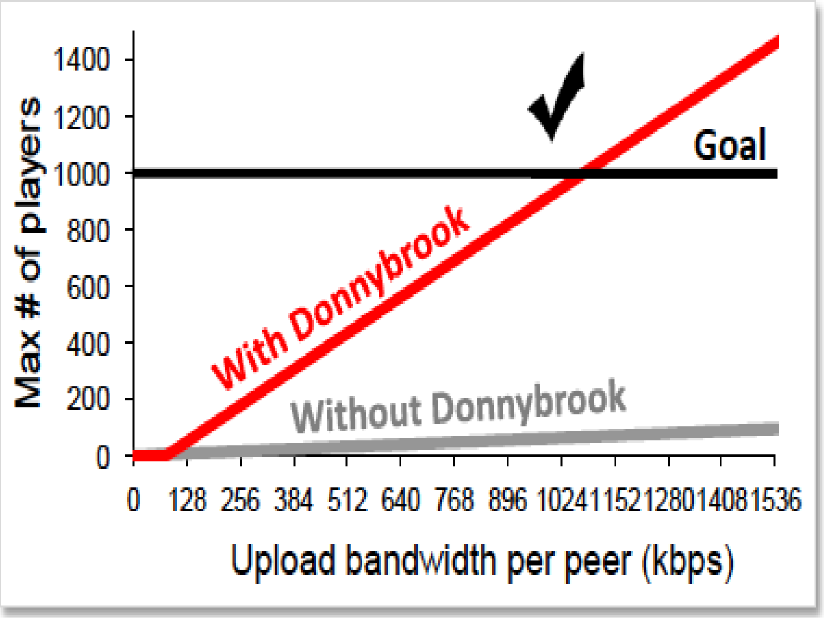
# Evaluation: Guidance staleness

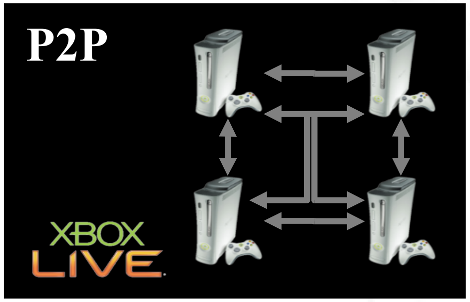



Guidance is almost never stale

# Donnybrook: Summary

- **Key techniques:**
  - Interest Sets:
    - Reduce BW demands
  - Update dissemination:
    - Handles heterogeneity
- **Ongoing Work:**
  - 1000 Player deployment




+


+