

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

Ashwin Bharambe

Jeffrey Pang

Srinivasan Seshan

Xinyu Zhuang

Carnegie Mellon

John R. Douceur

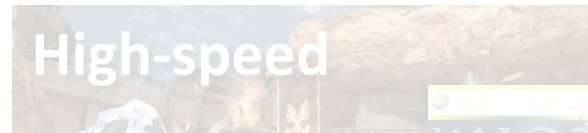
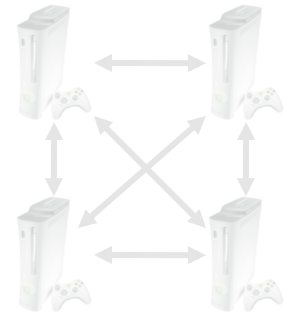
Jacob R. Lorch

Thomas Moscibroda

Microsoft®
Research

High-Speed, Large-Scale, P2P: **Pick 2**

- Many console games are peer hosted to save costs

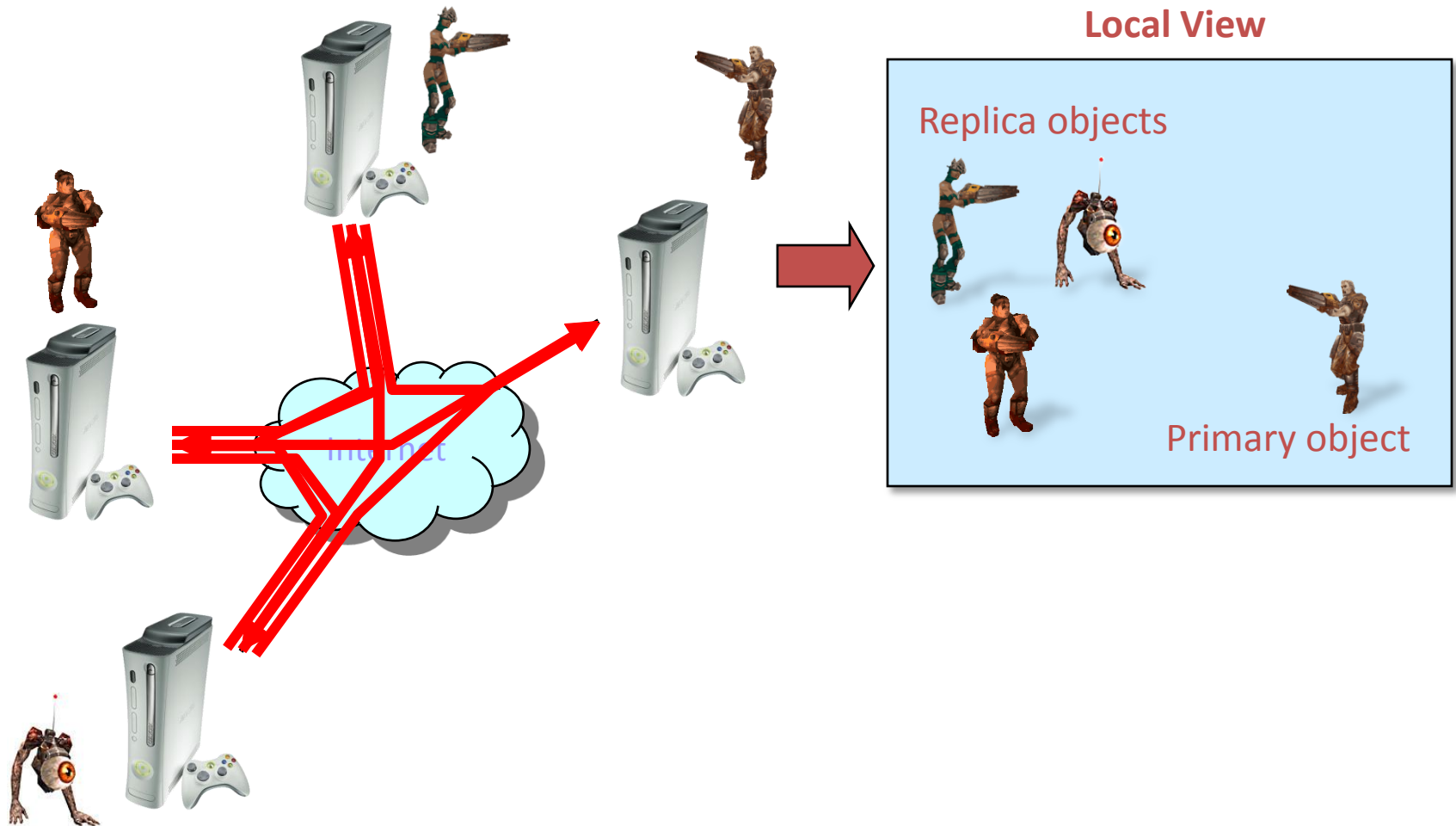


Question: Can we achieve all 3?

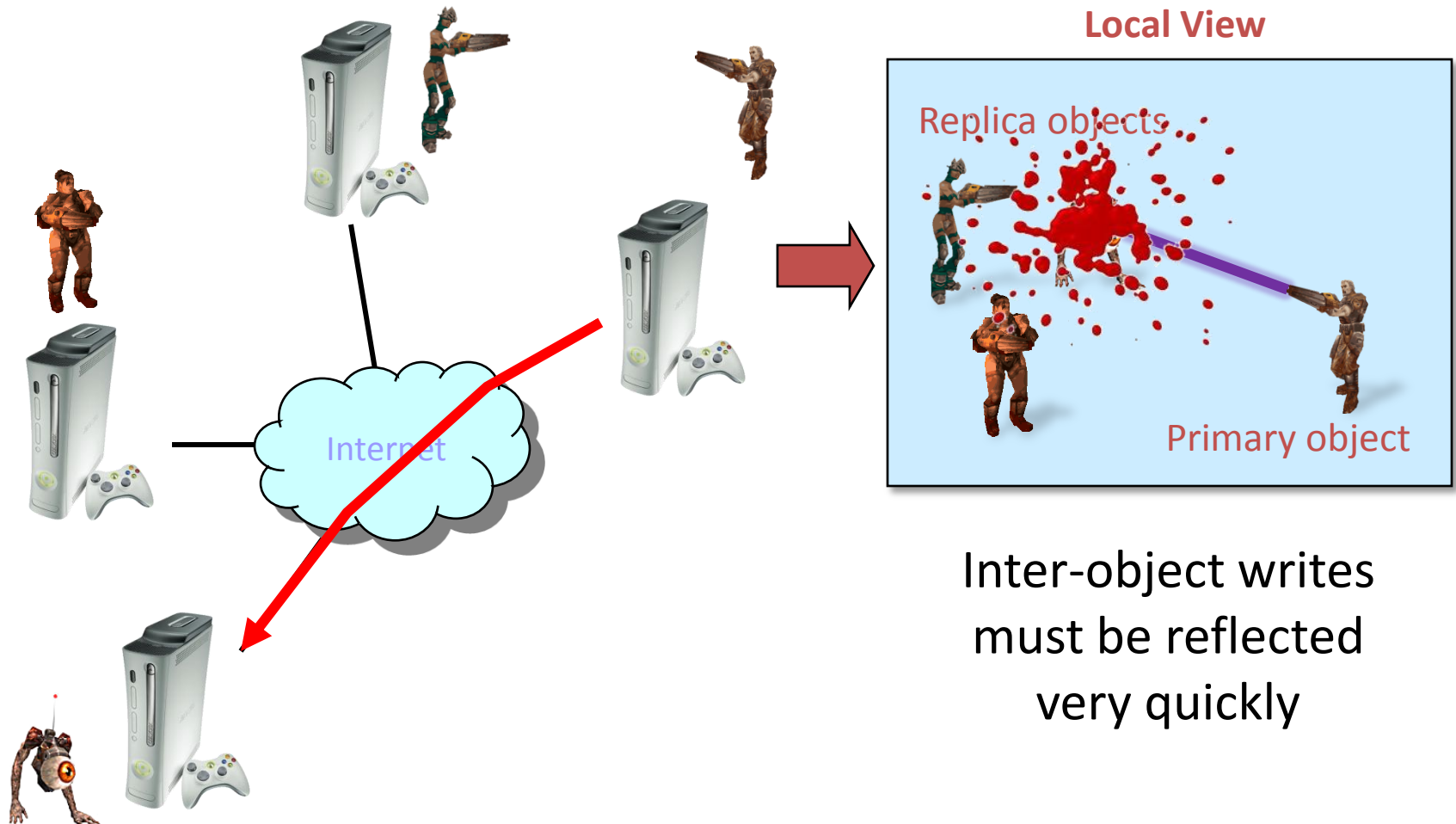
- 1000+ player games need dedicated servers



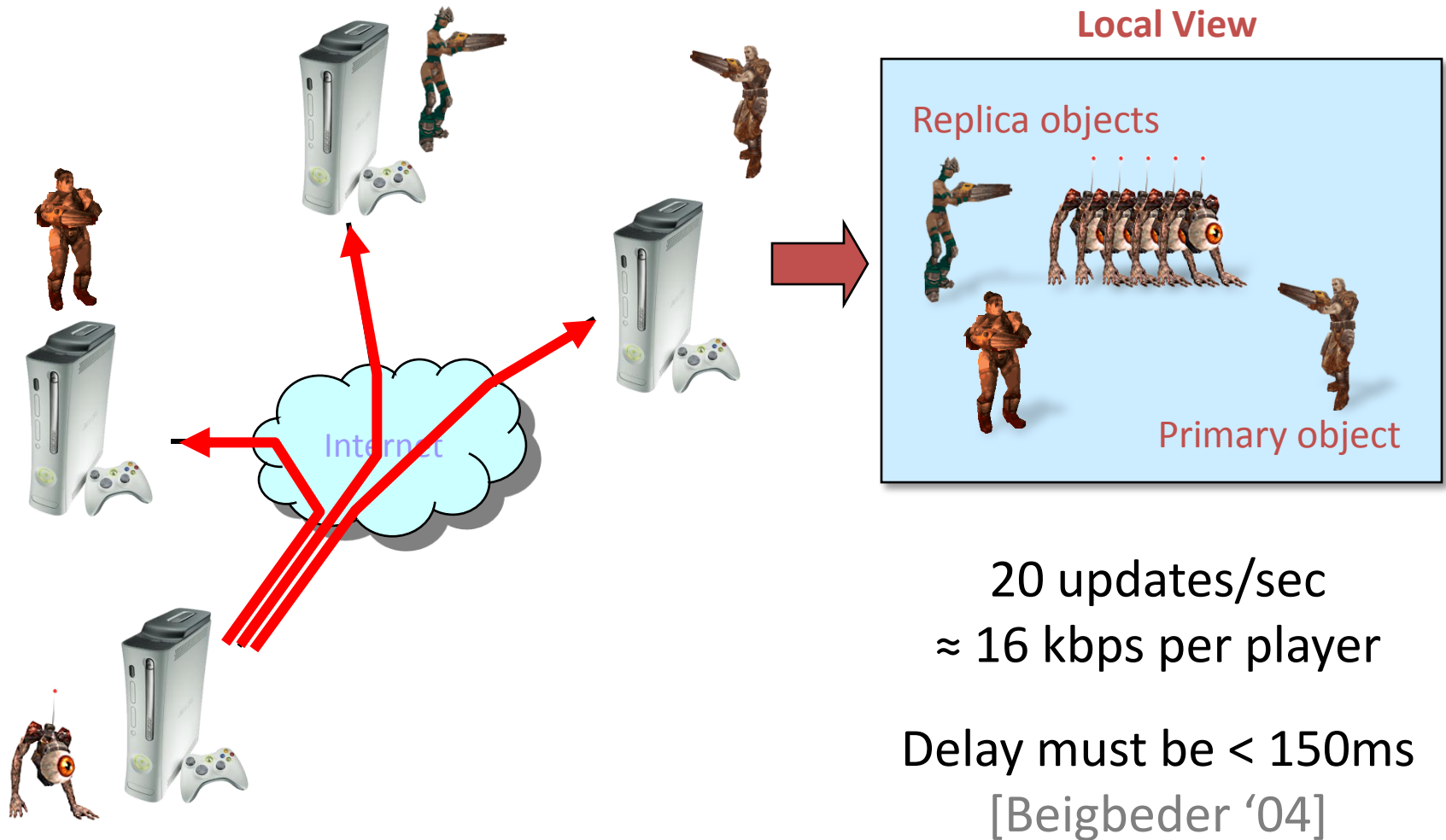
P2P



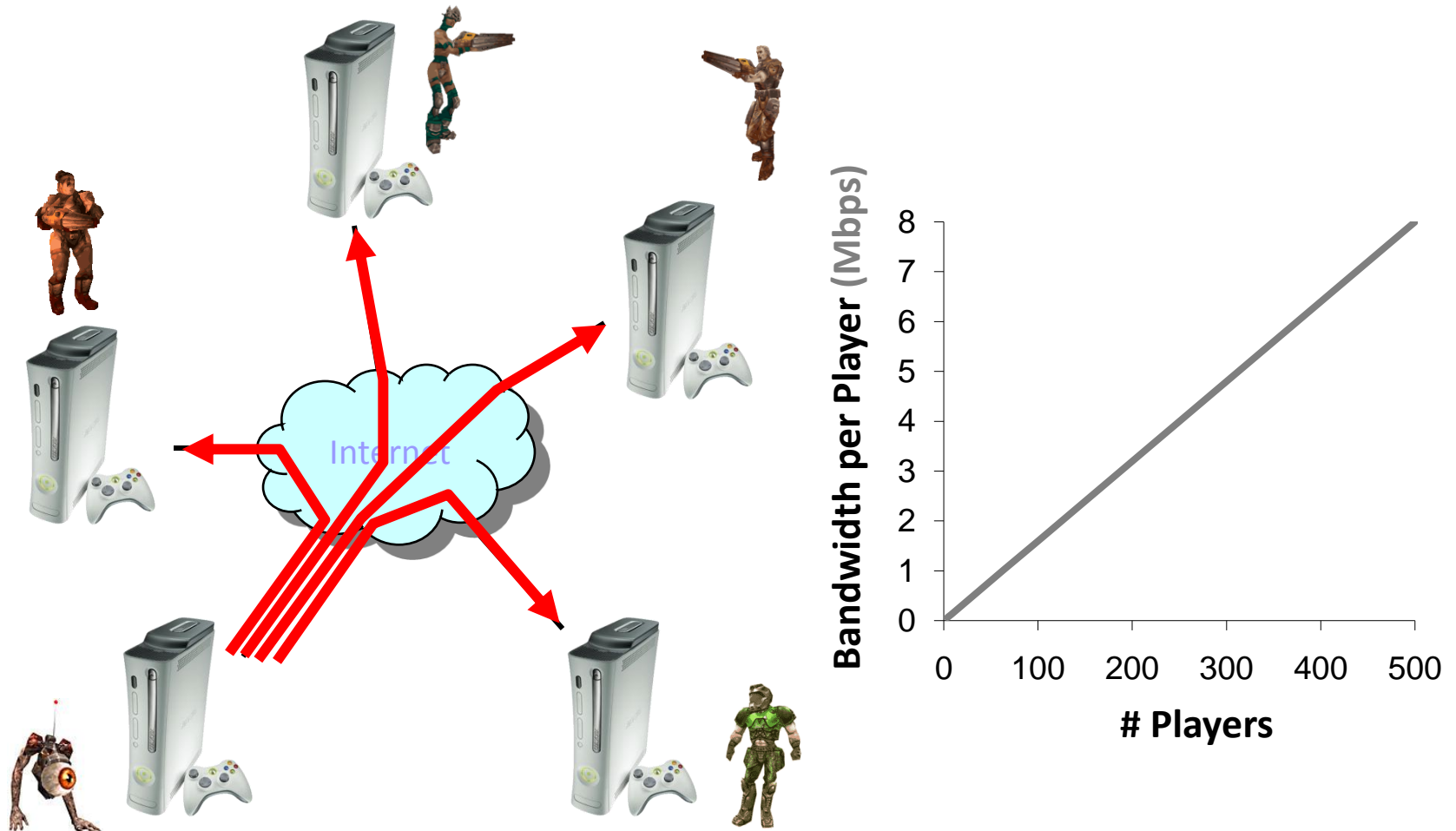
High-Speed



High-Speed

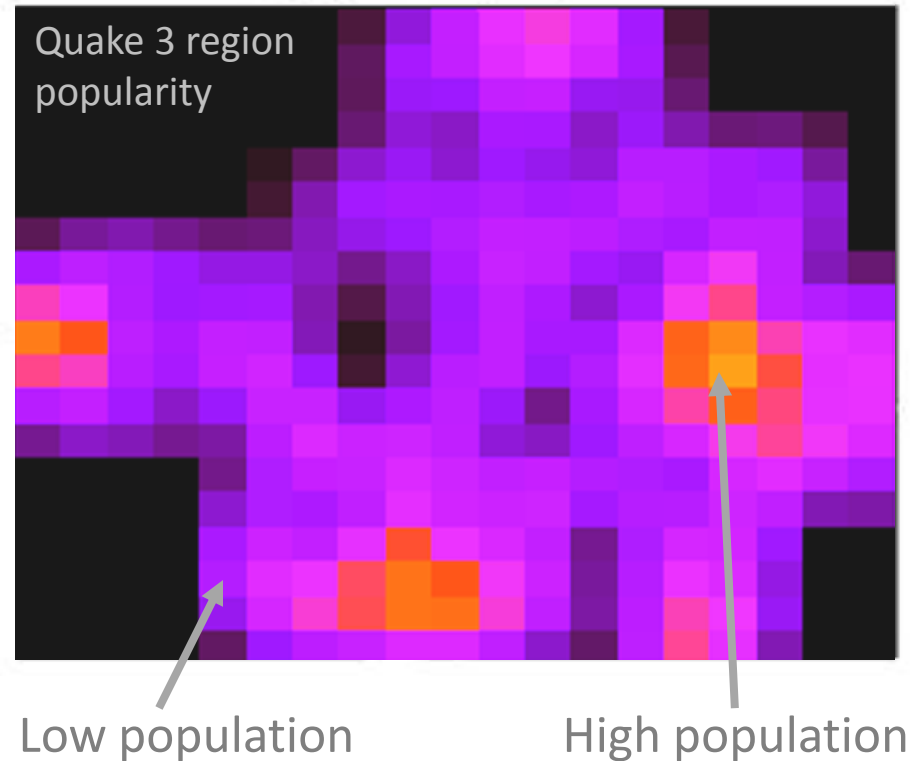


Large-Scale



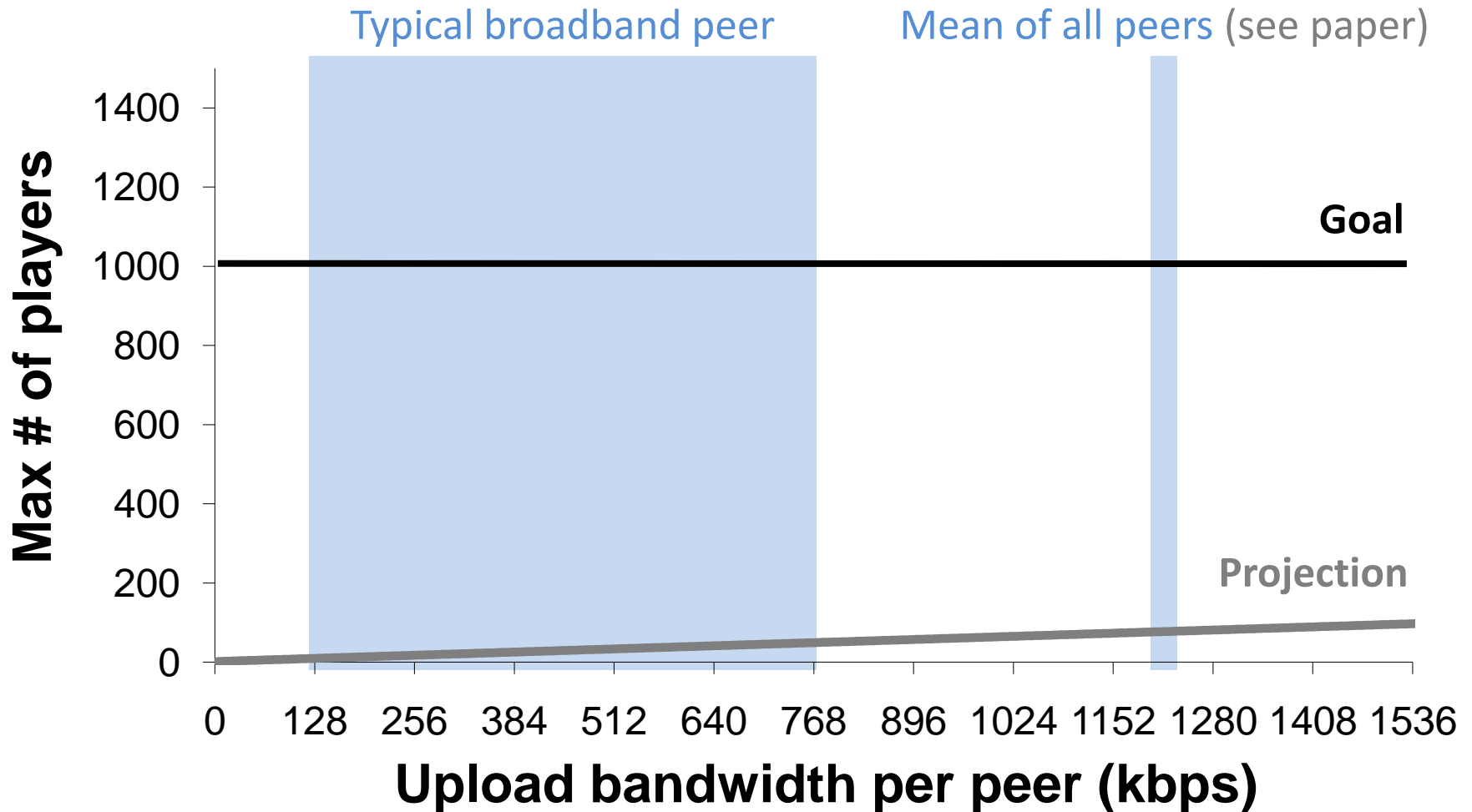
Area-of-Interest (AOI) Filtering

- 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

Projected Scalability



Not Enough Bandwidth



Ideal
20 updates/sec



Cable Modem (128 kbps)
5 updates/sec

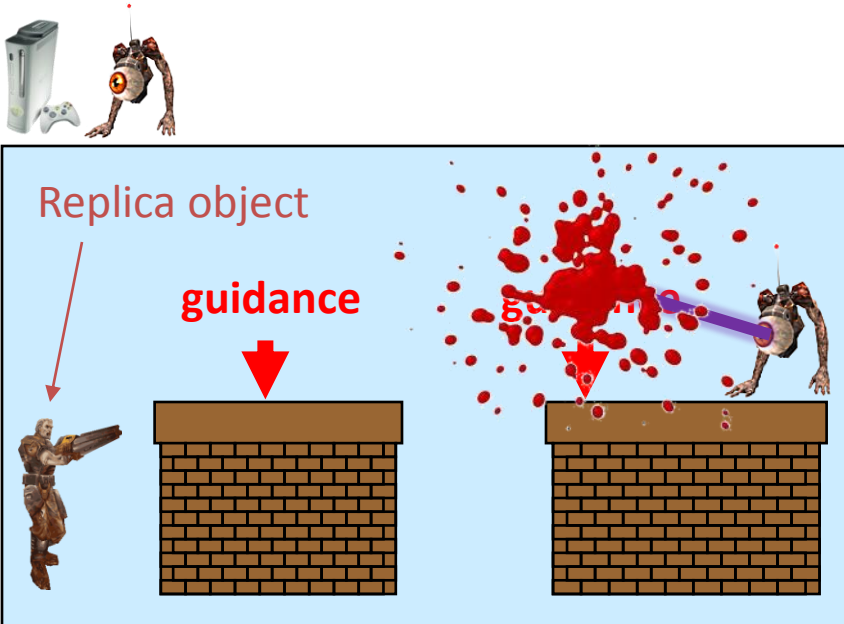
[P2P Quake 3]

Talk Outline

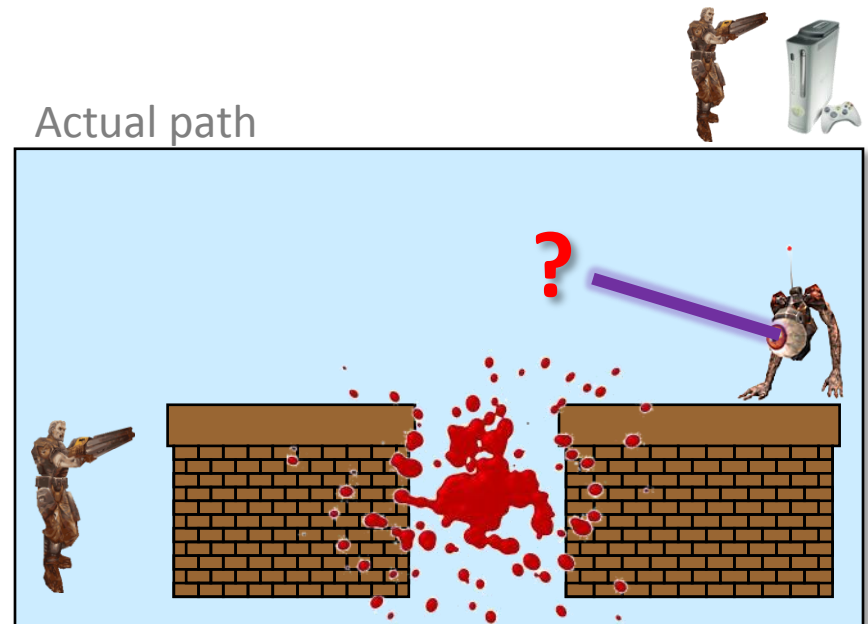
- Motivation and Goals
- Donnybrook: Interest Sets
 - Reduces mean bandwidth demands
- Donnybrook: Update Dissemination
 - Handles interest and bandwidth heterogeneity
- Evaluation

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



Actual path



Donnybrook: Interest Sets

- **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

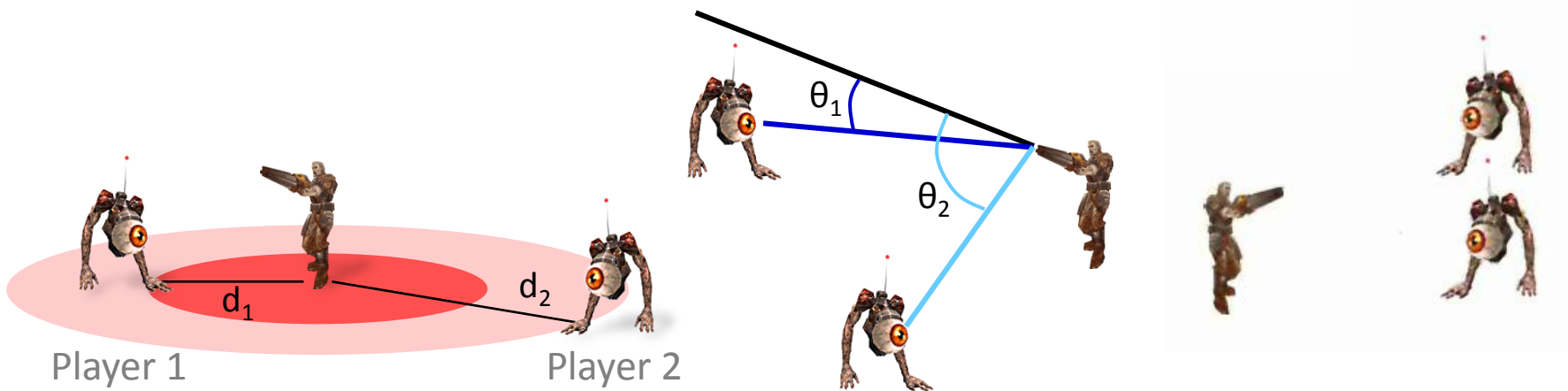


Donnybrook: Interest Sets

- How to estimate human attention?
 - $\text{Attention}(i)$ = how much I am focused on player i

$\text{Attention}(i) =$

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









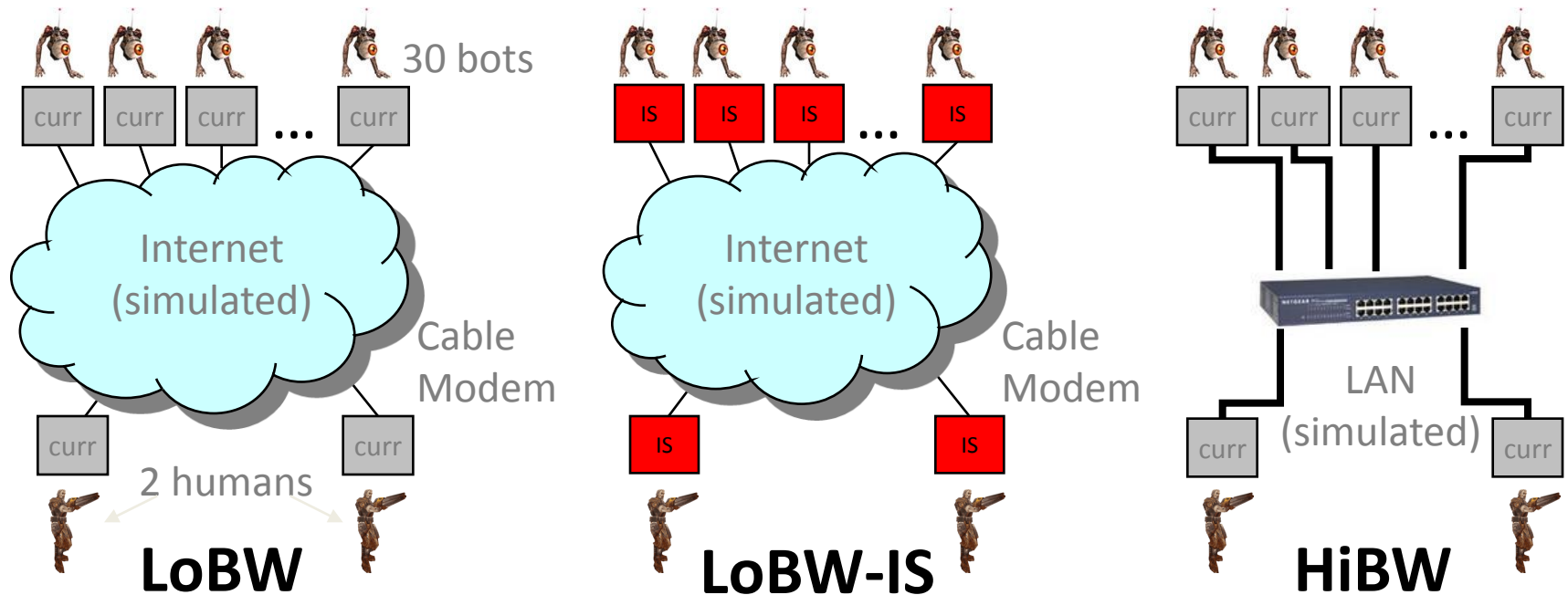


Interest Set Evaluation

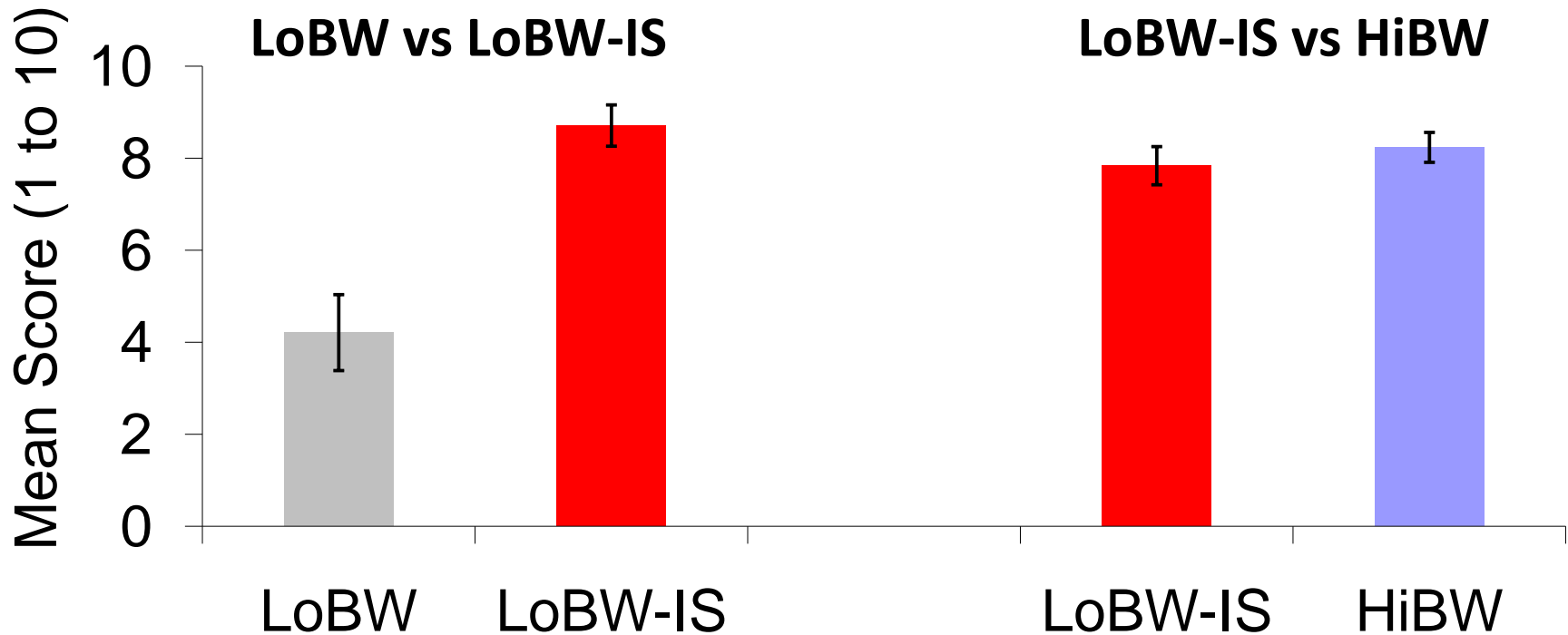
Question: Do Interest Sets improve fun in LoBW games?

Question: Do they make LoBW games as fun as HiBW?

User study: each pair of players compares 2 of 3 versions:

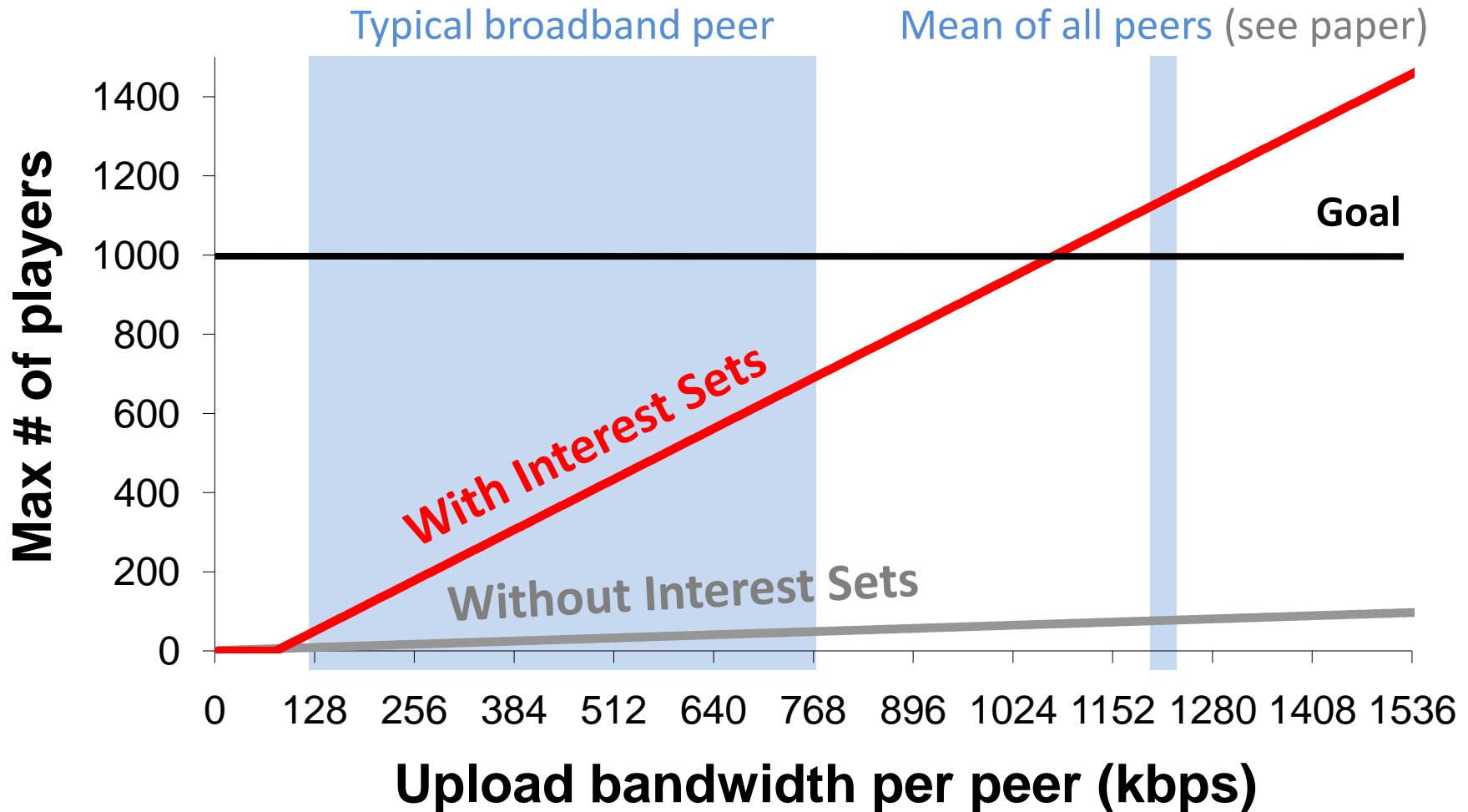


User Study Results



Survey: How fun was each version?

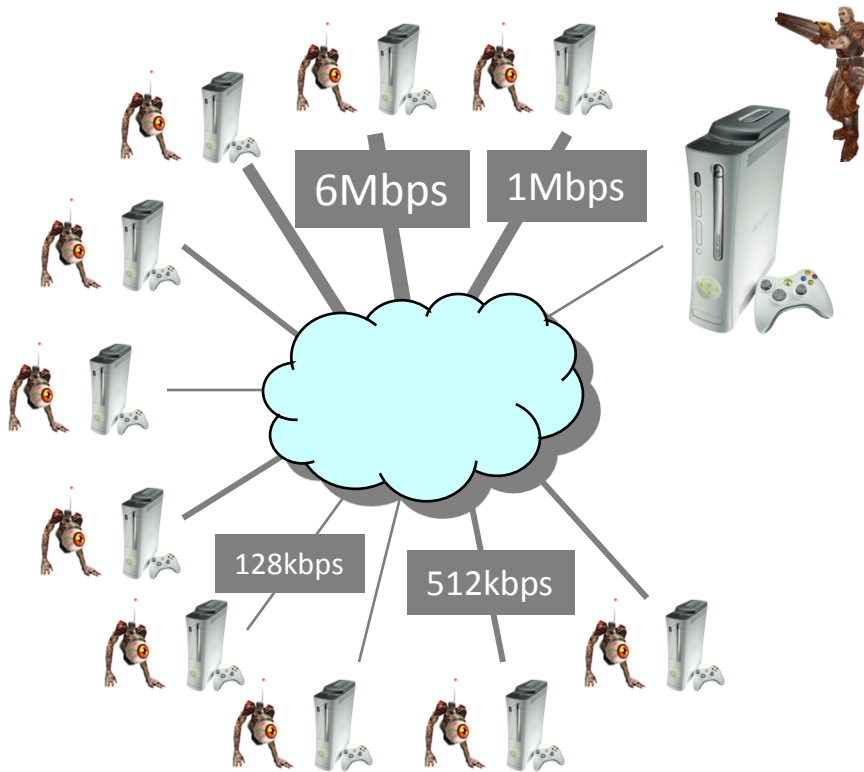
Projected Scalability



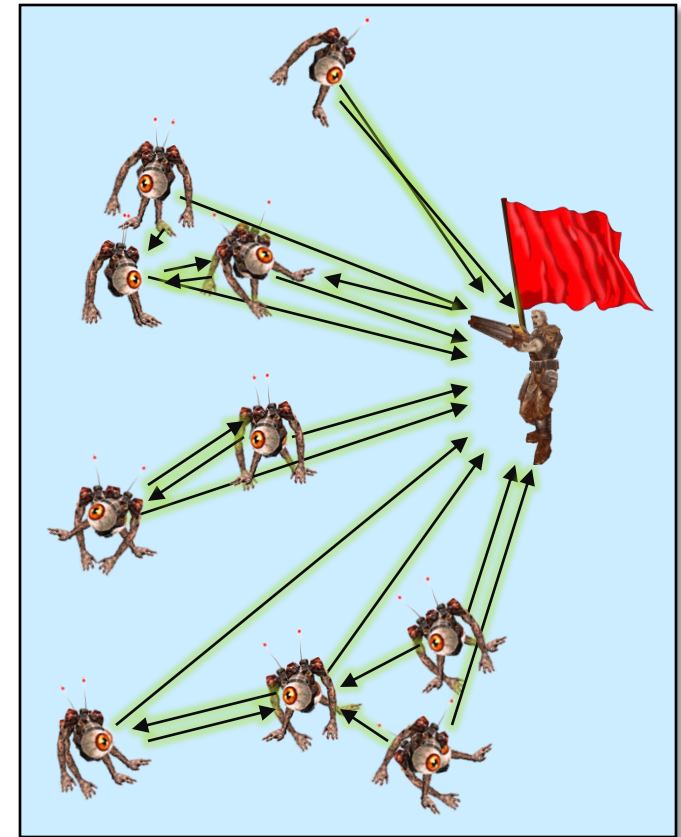
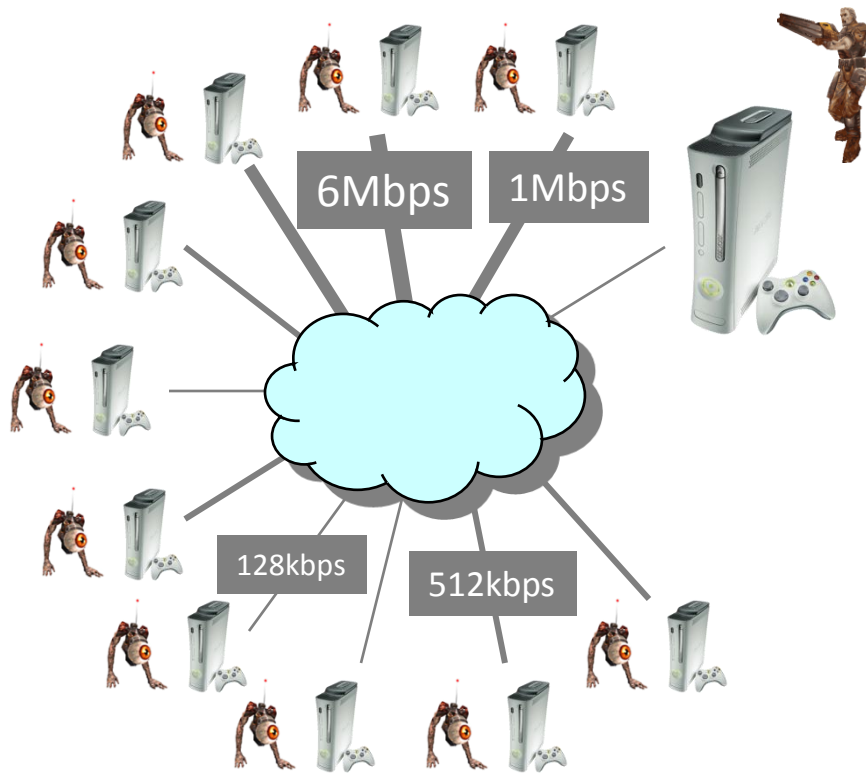
Talk Outline

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

Problem: Bandwidth Heterogeneity

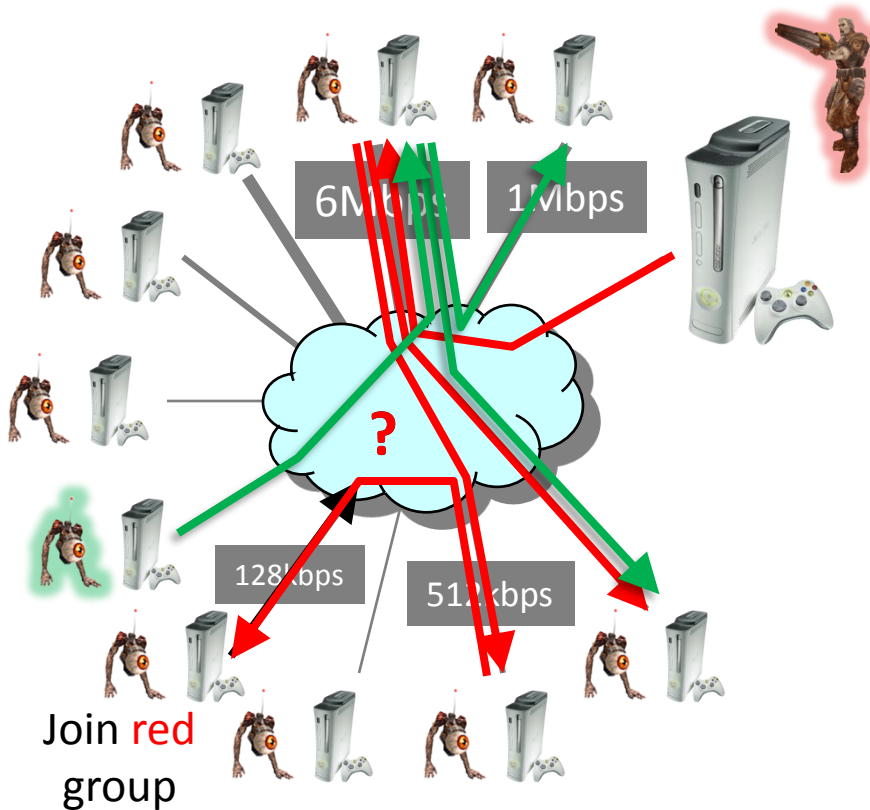


Problem: Interest Heterogeneity



→ Attention

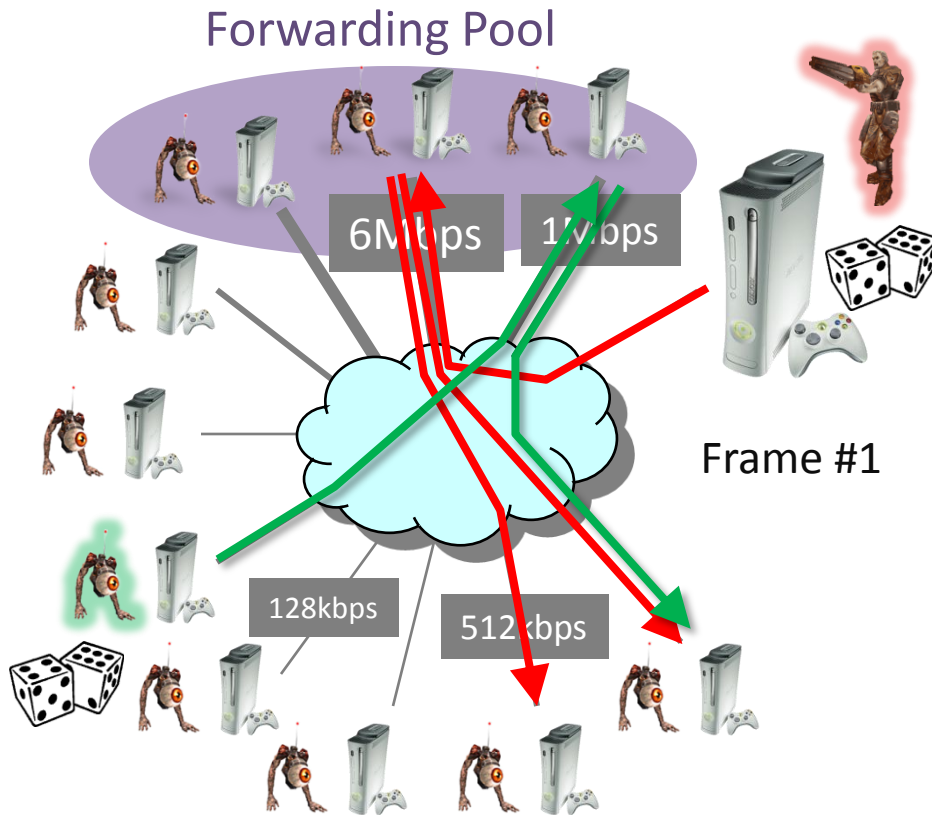
Why not Overlay Multicast?



- Main requirements:
 1. Strict delay bound (150ms)
 2. Frequent membership changes (68% turnover/sec)
 3. Bandwidth heterogeneity
 4. 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

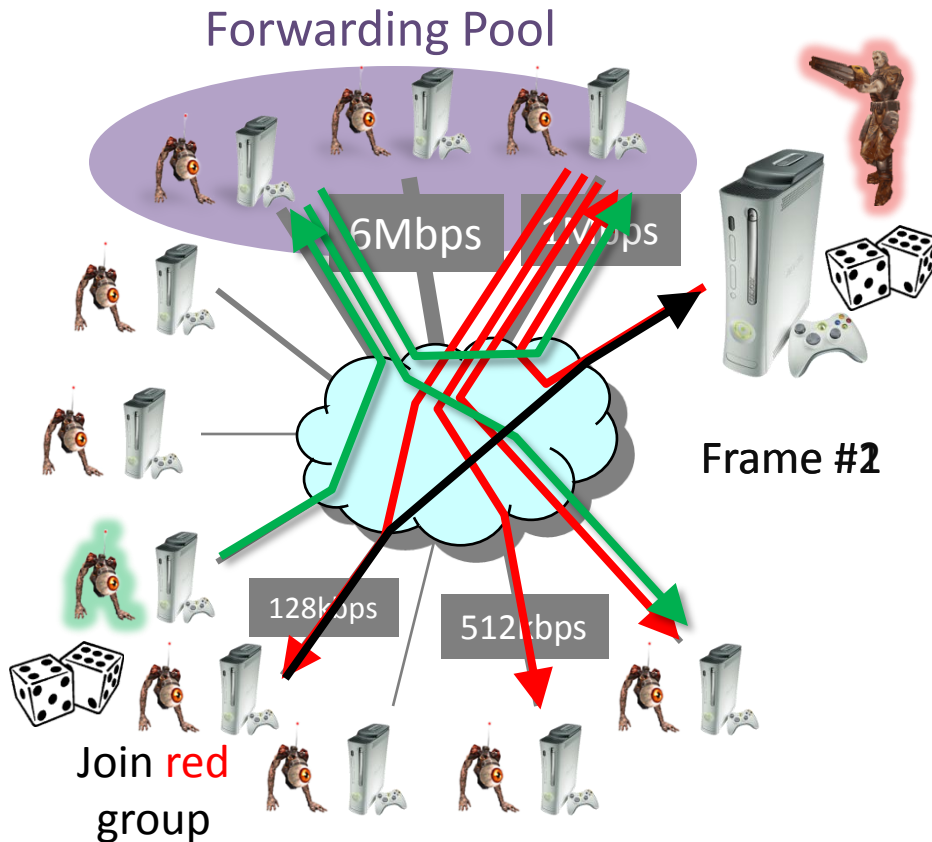
Donnybrook: Update Dissemination



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

Randomized *source*-initiated tree construction

Donnybrook: Update Dissemination



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

Randomized *source*-initiated tree construction

Donnybrook: Update Dissemination

- Main requirements:
 - ✓ 1. Strict delay bound: **constant tree depth**
 - ✓ 2. Freq. membership changes: **uncoordinated tree construction**
 - ✓ 3. Bandwidth heterogeneity: **high bandwidth forwarding pool**
 - ✓ 4. 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 $\frac{1}{2}$ forwarder capacity)
⇒ drops happen rarely and only cause loss for 1 frame
 - 5-10% loss is OK [Beigbeder '04]

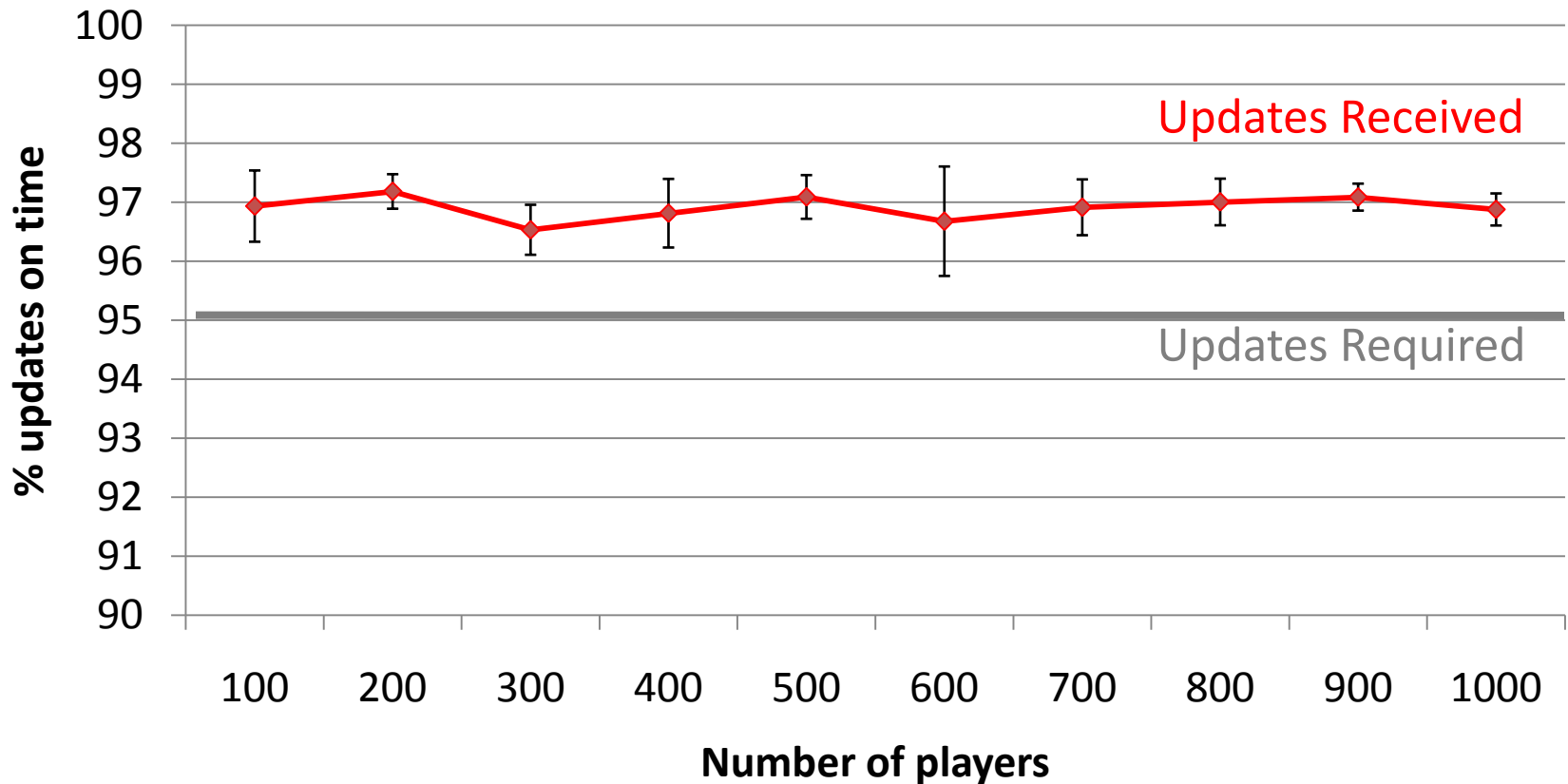
Update Dissemination Evaluation

Question: Does this approach deliver enough updates on time to preserve fun game play?
(i.e. 90-95% of updates in 150ms [Beigbeder '04])

Evaluation setup (see paper for details)

Implementation	Quake 3 with interest sets and update dissemination
Workload	Synthetic 100-1000 player games using “bots” <ul style="list-style-type: none">• based on real 32 player CTF games [Bharambe '06]
Network	Packet-level network simulator <ul style="list-style-type: none">• bandwidth model: P2P hosts [Piatek '07]• latency model: Halo 3 players [Lee '08]• loss model: two-state Gilbert model [Zhang '01]

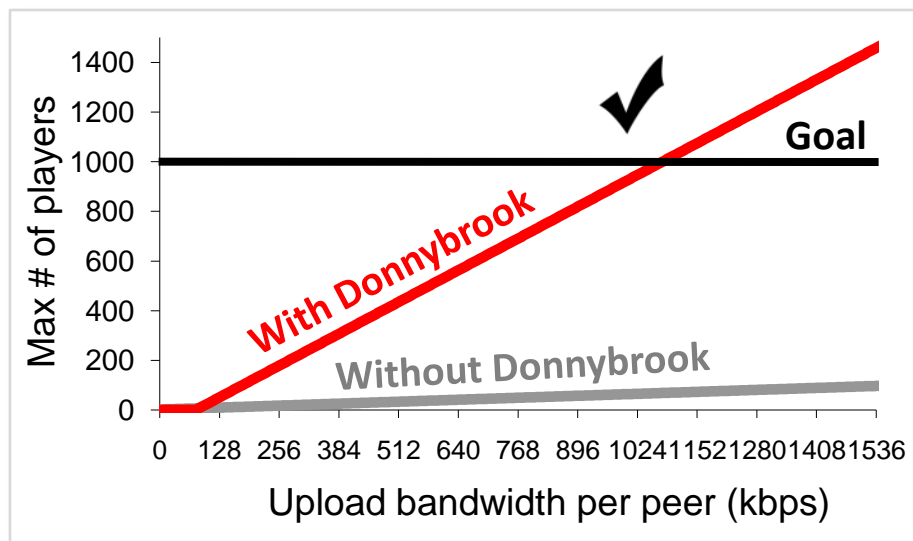
Evaluation Results



Enough updates are delivered on time at all scales

Donnybrook Summary

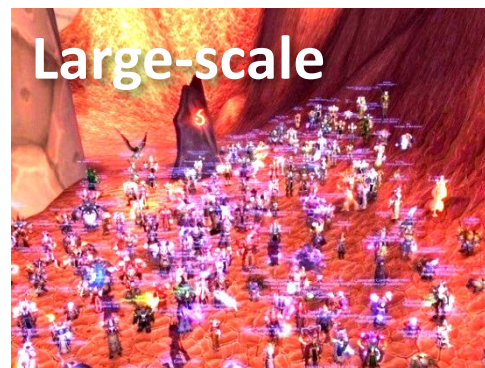
- **Key techniques:**
 - Interest sets:
reduce bandwidth demands
 - Update dissemination:
handles heterogeneity
- **Ongoing work:**
 - 1000 player deployment



+



+



Questions?



Cable Modem



Cable Modem with Donnybrook

<http://www.epicbattle.us>

==== Clarification Slides =====

Mitigating Cheating

- Existing defenses can prevent software cheats
 - Deploy on consoles (relatively closed platforms)
 - Use trusted hardware (e.g., Xbox 360 TPM)
 - Encrypt all packets between nodes
- Donnybrook is uniquely vulnerable to traffic analysis
 - Examine update packets you send to determine receivers
 - ⇒ Allows you to see who is paying attention to you
 - Drop update/guidance packets that you receive
 - ⇒ Causes all replicas on your node to act using “dumb” AI
- Ongoing work on traffic analysis defense
 - Choose forwarders to conceal packet source/destinations
 - Punish player if expected message rates are not maintained

Game Execution Model

- **Game State:**

- Collection of distinct *objects* (players, missiles, items, etc.)

- **Game Execution:**

- Each object has a **Think** function:

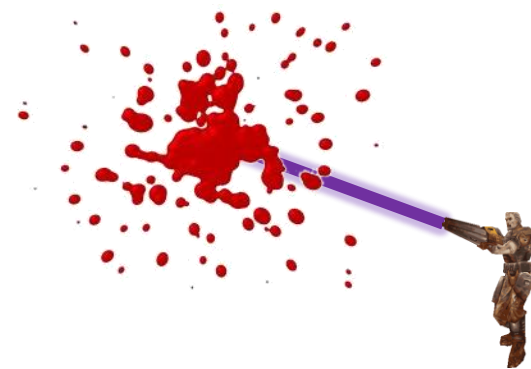
```
Think() { ReadPlayerInput(); DoActions(); ... }
```

- Execute each object once per *frame*:

```
Each 50ms do {  
    foreach object do {  
        object->Think();  
    }  
}
```

Pairwise Rapid Agreement

- **Interaction:** when player A modifies player B (i.e. A performs a write on B)
- **Goal:** modification is consistent and applied quickly
- **Insight:** # interactions scales slowly
 - Occur at human time scales \Rightarrow infrequent
 - Involve only 2 players \Rightarrow unicast
- **Solution:** prioritize all inter-object writes
 - Player A sends mod to Player B
 - Player B applies mod, sends result to A
- PRAs required in Quake III:
 - Damage, Death, Item Pickup, Door Opening



[Pang, IPTPS '07]

Guidance

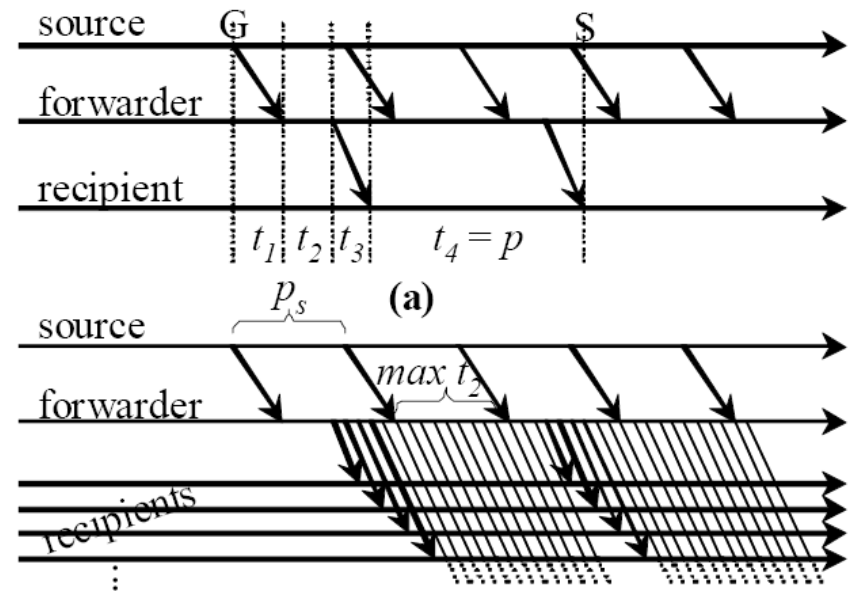
- **Motivation:** state updates get stale fast
 - **Example:** players can travel the diameter of a Quake 3 map in seconds
 - **Goal:** send prediction of state at time of next expected guidance
 - **Example:** predict where a player will be at the next guidance
- **Predicted Properties:**
 - **Predict position:** simulate where physics brings player in next second
 - **Predict viewing angle:** use view angles to estimate player's target aim
 - **Predict Events:** use #-shots-fired to estimate when a player is “shooty”

Guidable AI

- **Problem:** Guidable AI peers receive very infrequent guidance
- **Solution:** Smooth state changes with AI
 - **Position:** use existing path finding code to make replica move smoothly
 - **Angle:** have AI turn smoothly toward predicted targets
- **Convergence**
 - **Motivation:** Players in focus should be represented more accurately, but should not “warp” to actual position
 - **Solution:** Converge to actual state when receiving frequent updates
 - Focus on player B
 - ⇒ In player B’s Focus Set, get frequent updates
 - ⇒ $\text{Error}(\text{replica}, \text{actual})$ decreases with each update
 - When $\text{Error}() < \epsilon$, use player B’s update snapshots instead of AI
 - $\text{Error}(a,b) = \text{distance}(a.\text{position}, b.\text{position})$

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)

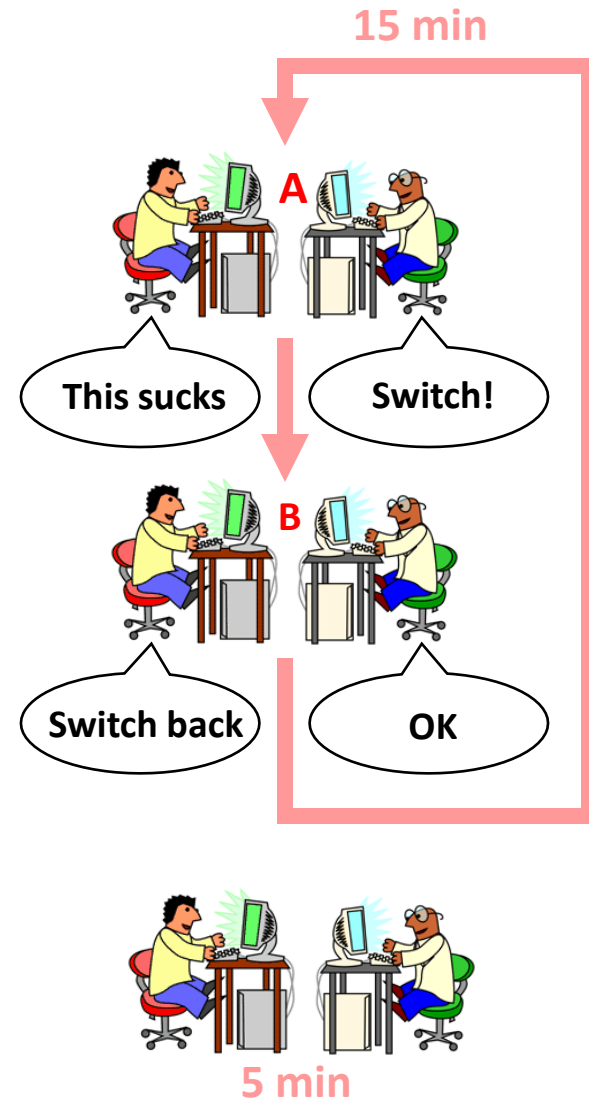


===== User Study Slides =====

User Study Setup

User Study Procedure

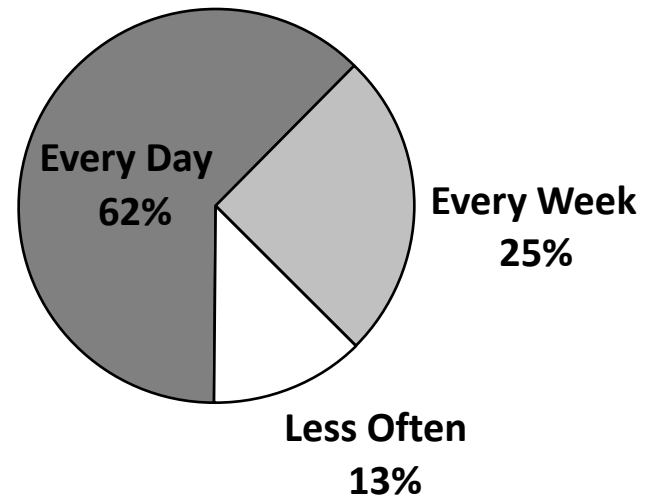
- Before experiment, practice on HiBW
- Tell players two Quake III “servers” exist: A and B
- Start playing on A, can vote to switch to B
- When both players vote, game continues on B
- Can vote to switch back and forth
- Analog to how players choose game servers (if good, stay, otherwise leave and try another)
- Play new game on least-used version so they can compare



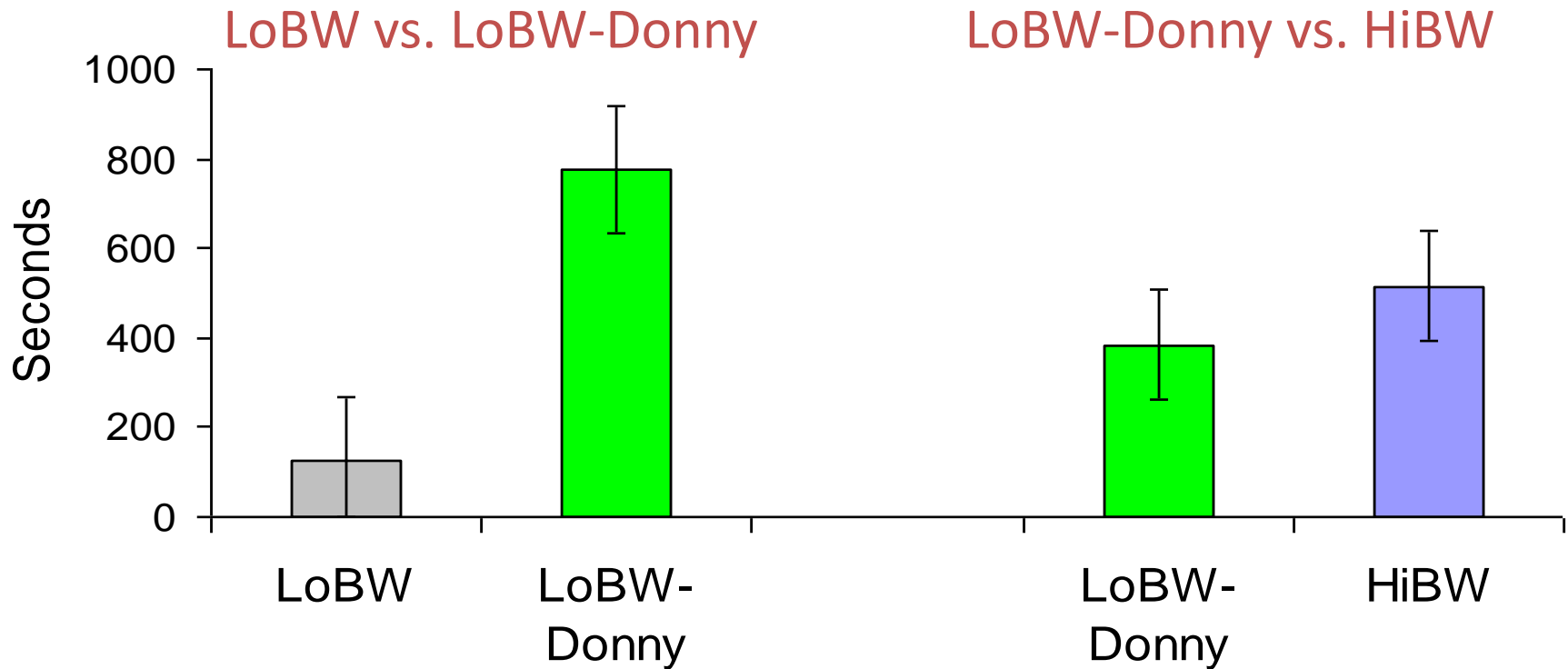
User Study Stats

- **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?

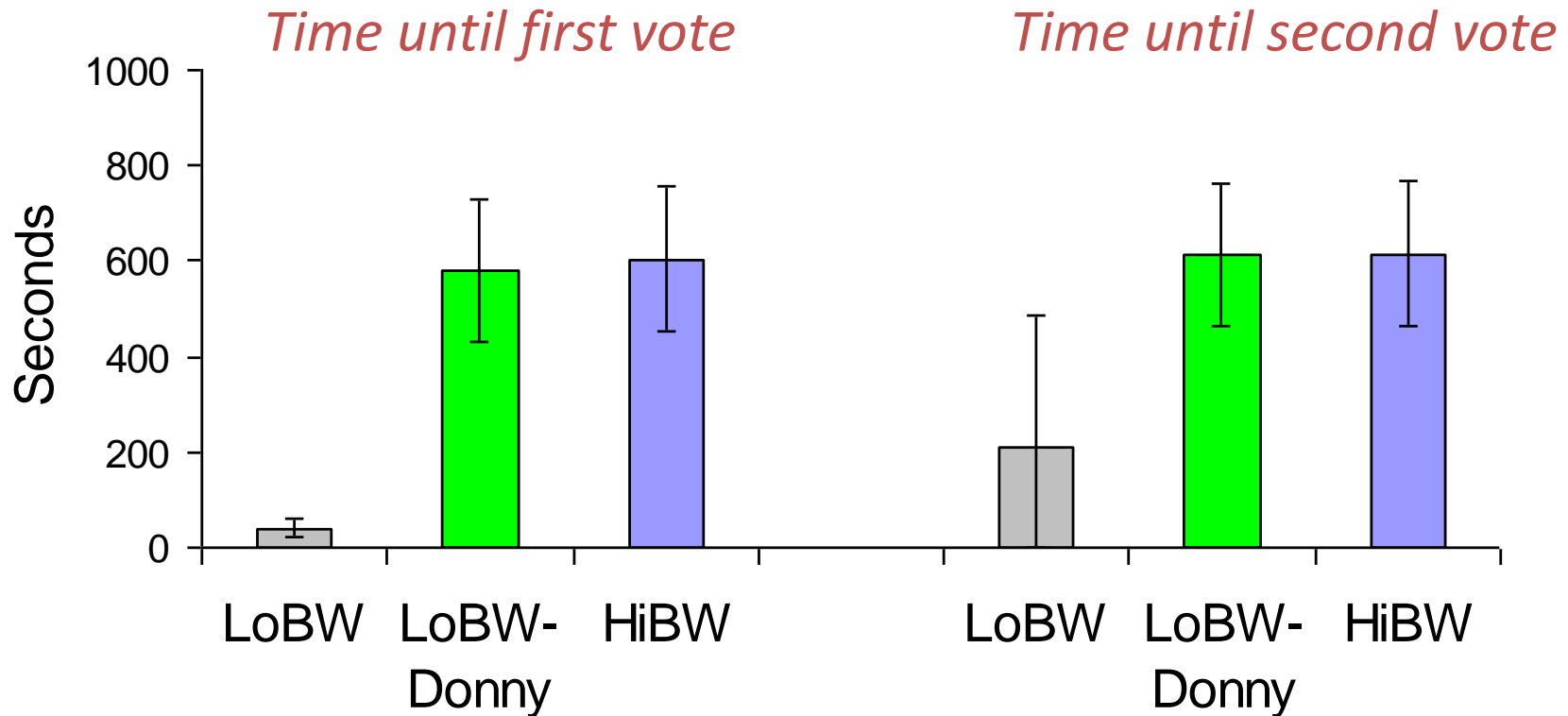


User Study: Total Stay Time



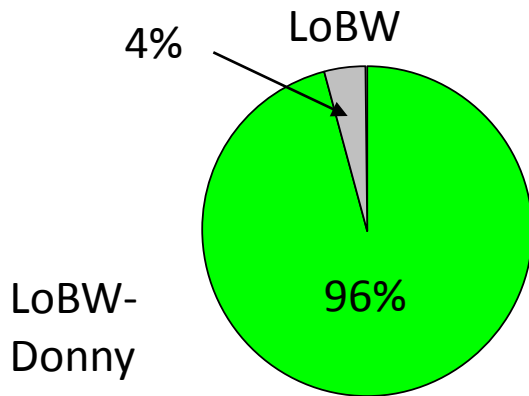
How long does a pair play on each version?

User Study: Departure Time

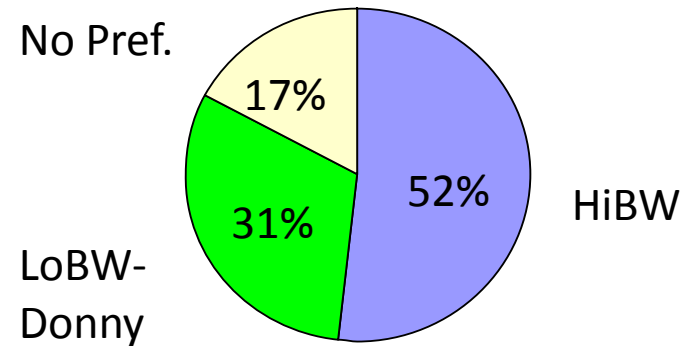


How long before a player wants to switch?

User Study: Preference



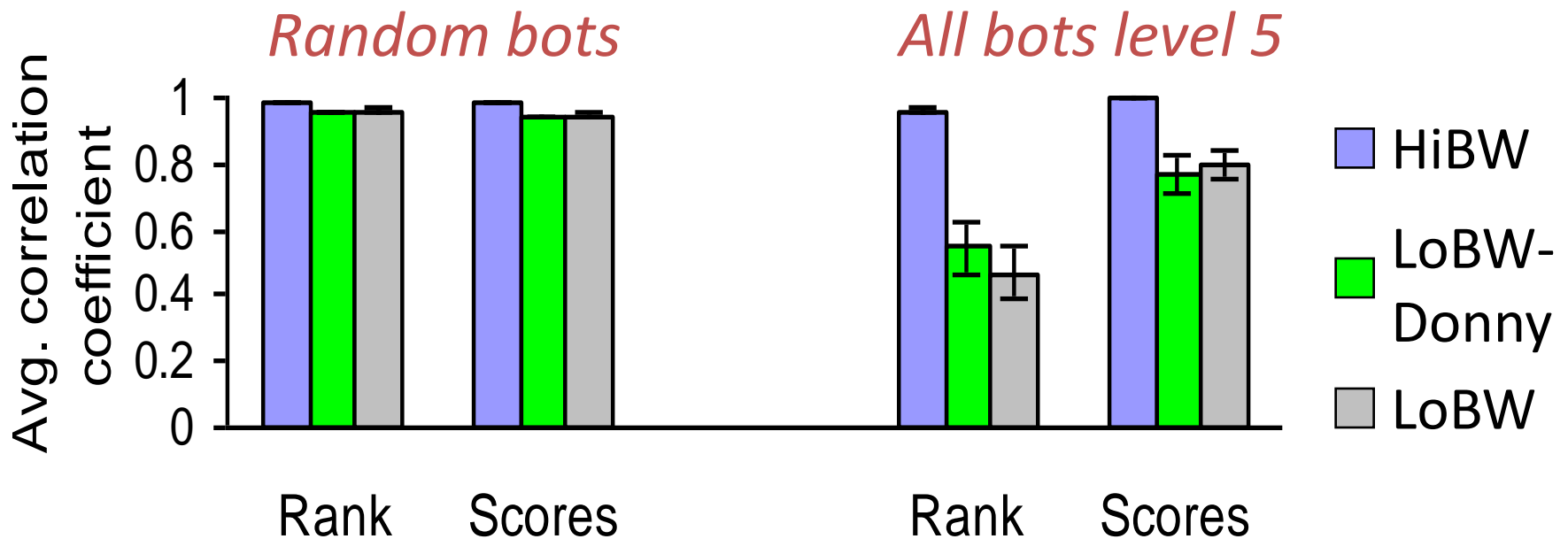
LoBW-Donny vs. LoBW



LoBW-Donny vs. HiBW

Survey: Was A or B more Fun?

Interest Sets: Fairness

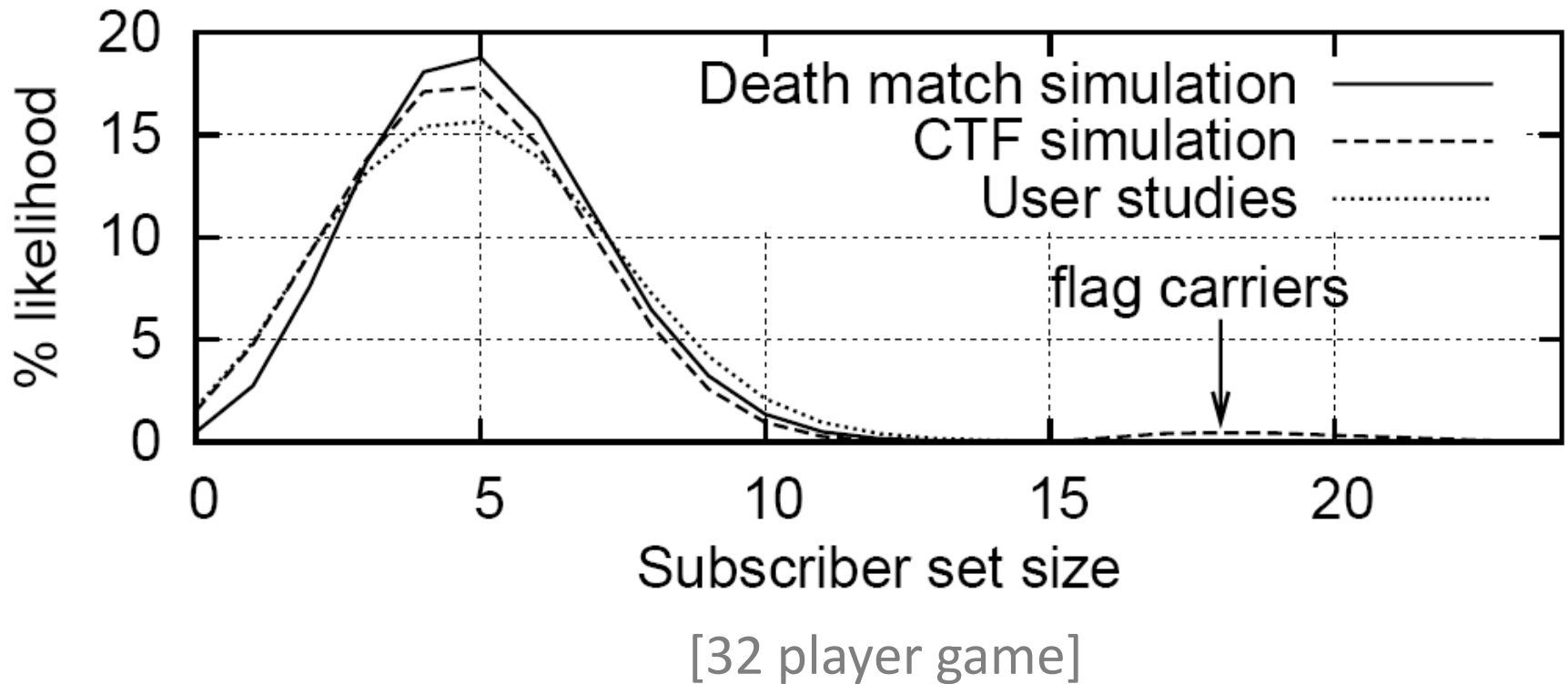


[Experiment with 16 bots at different skill levels]

Donnybrook preserves coarse skill-level differences

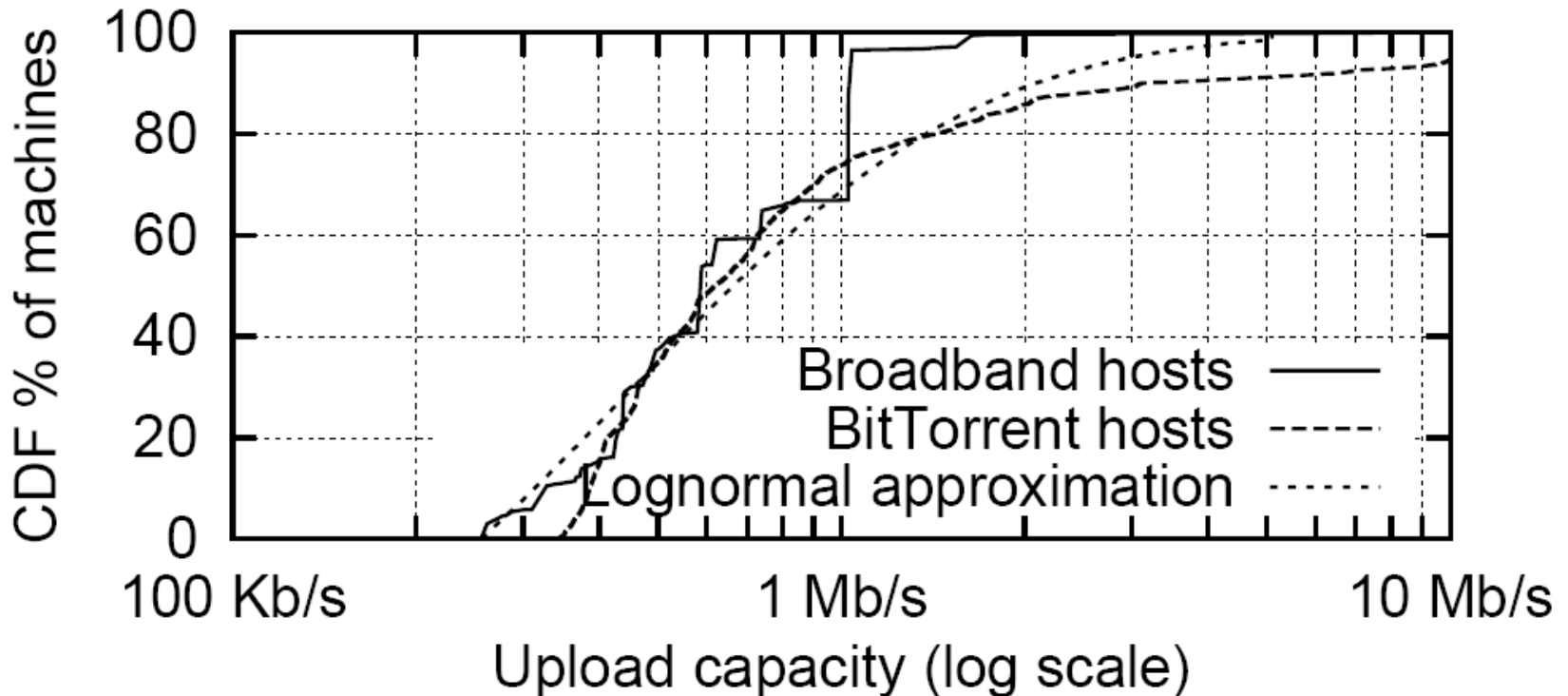
===== Game Stats Slides =====

Subscriber Set Size



Some players have lots of subscribers

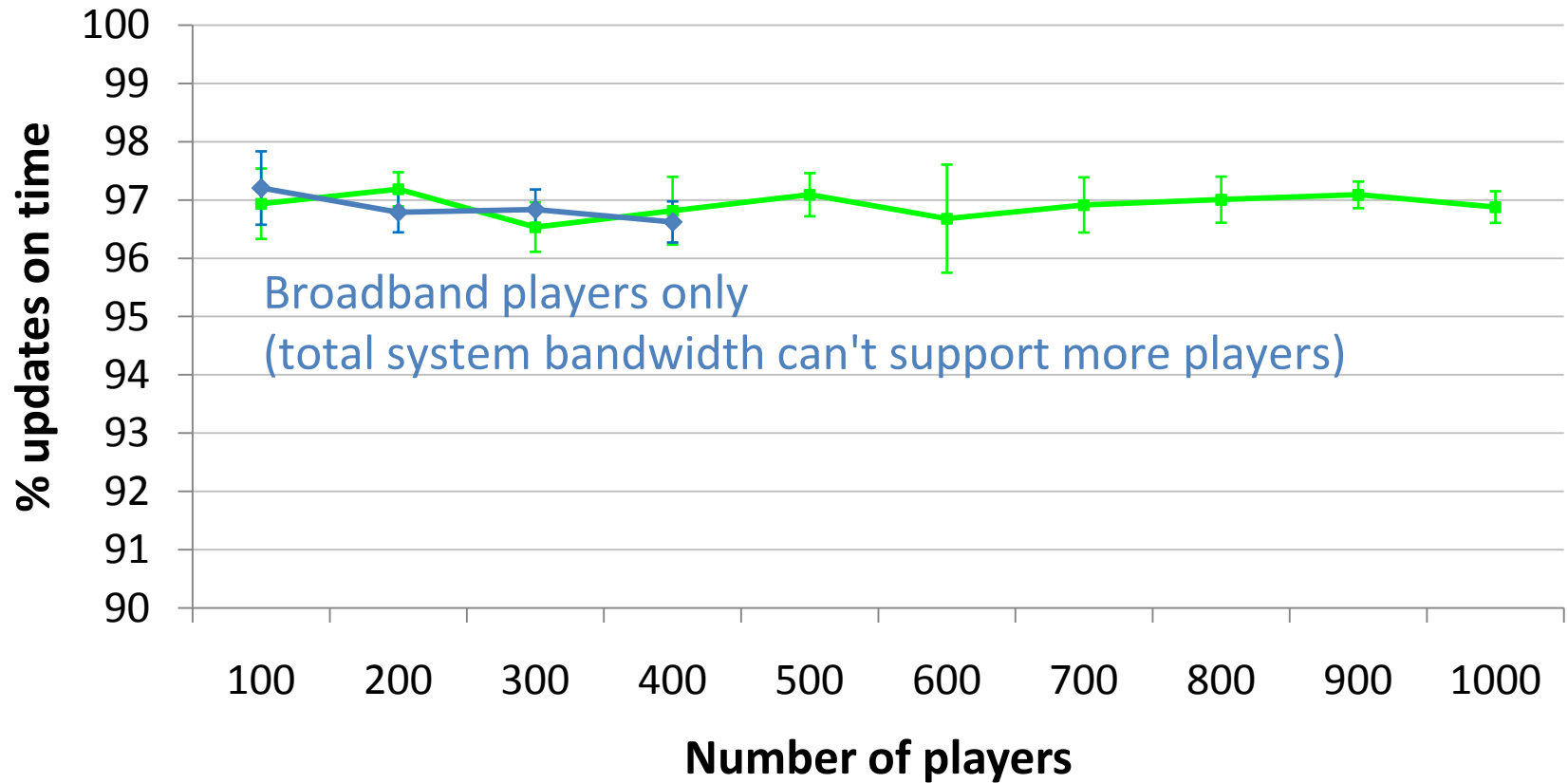
Bandwidth Distributions



Most peers have < 768 kbps, some have much more

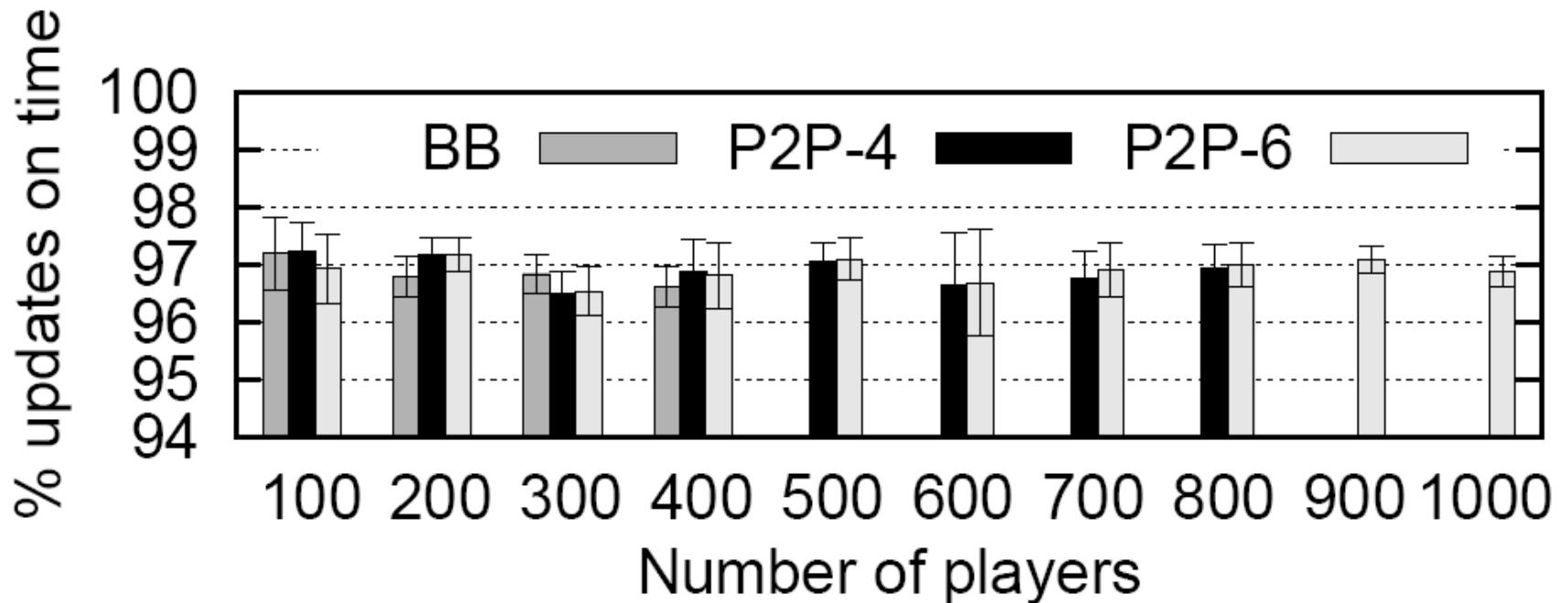
===== Evaluation Slides =====

Evaluation: Broadband Only



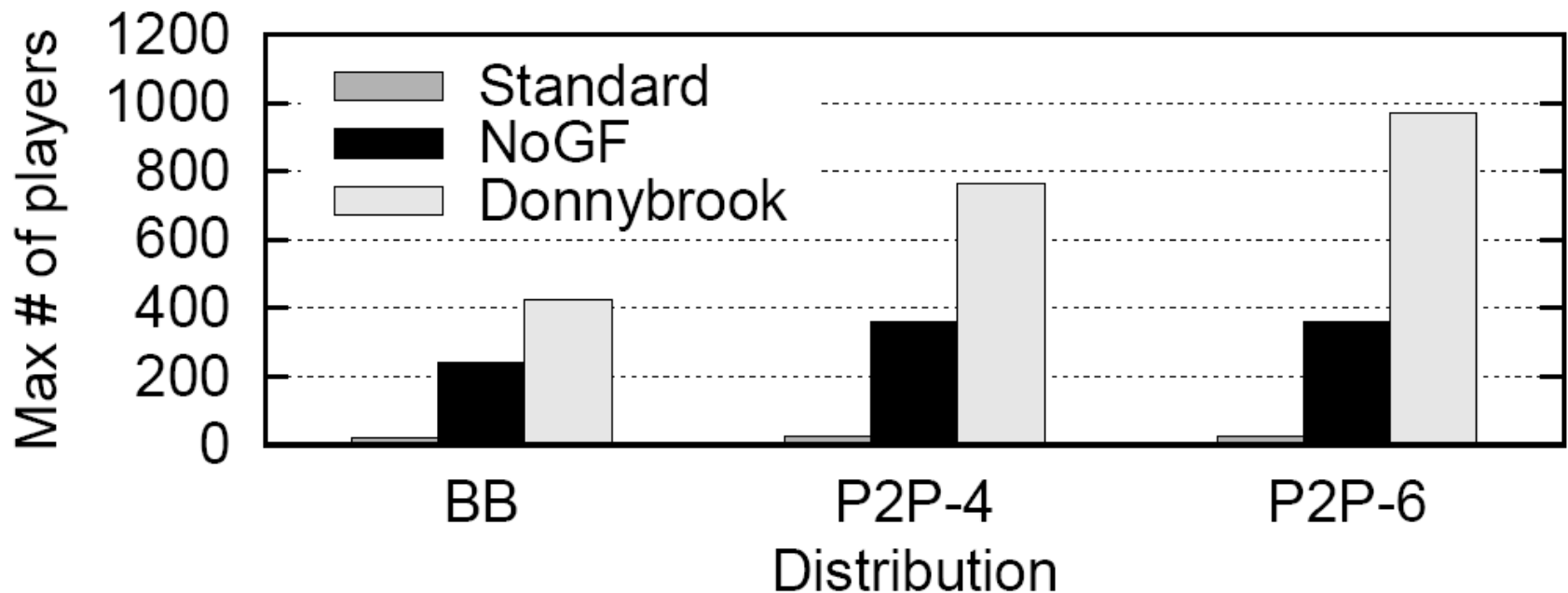
Enough updates are delivered at all supported scales

Evaluation: Other BW Distributions



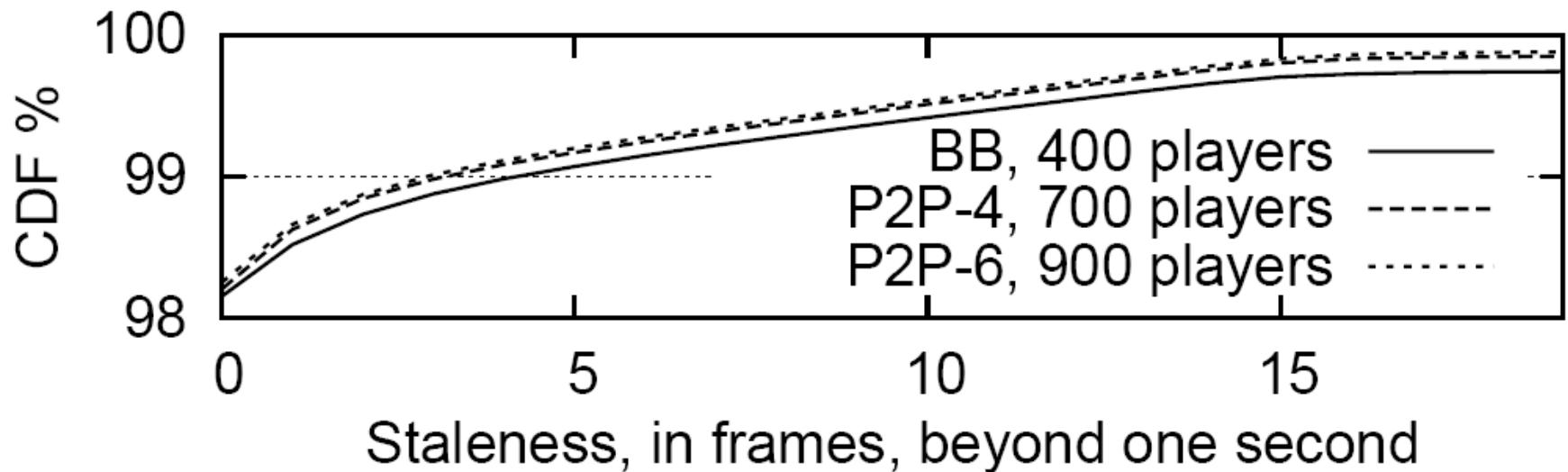
Enough updates are delivered at all supported scales

Evaluation: Scale



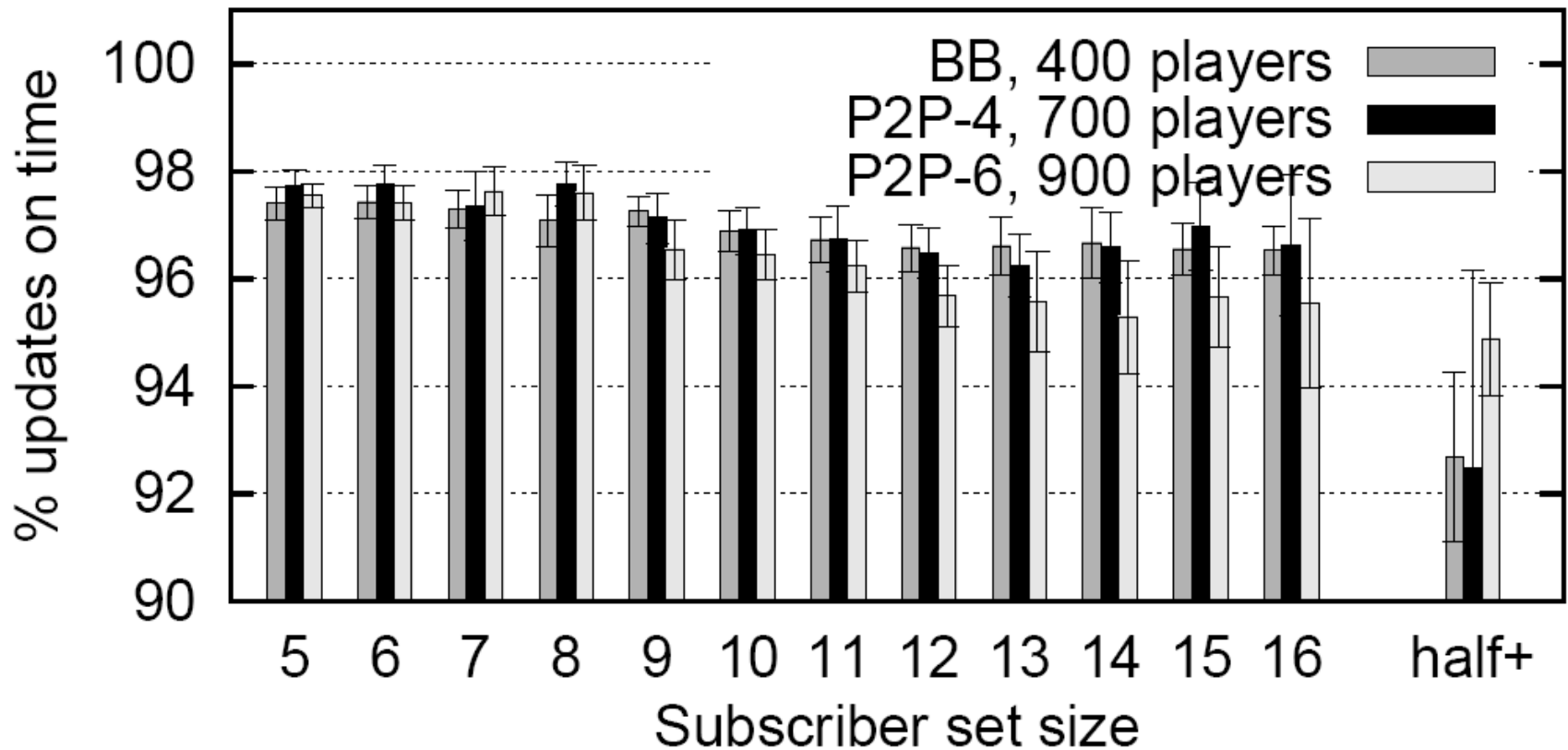
Donnybrook enables 100s of players in many BW models

Evaluation: Guidance Staleness



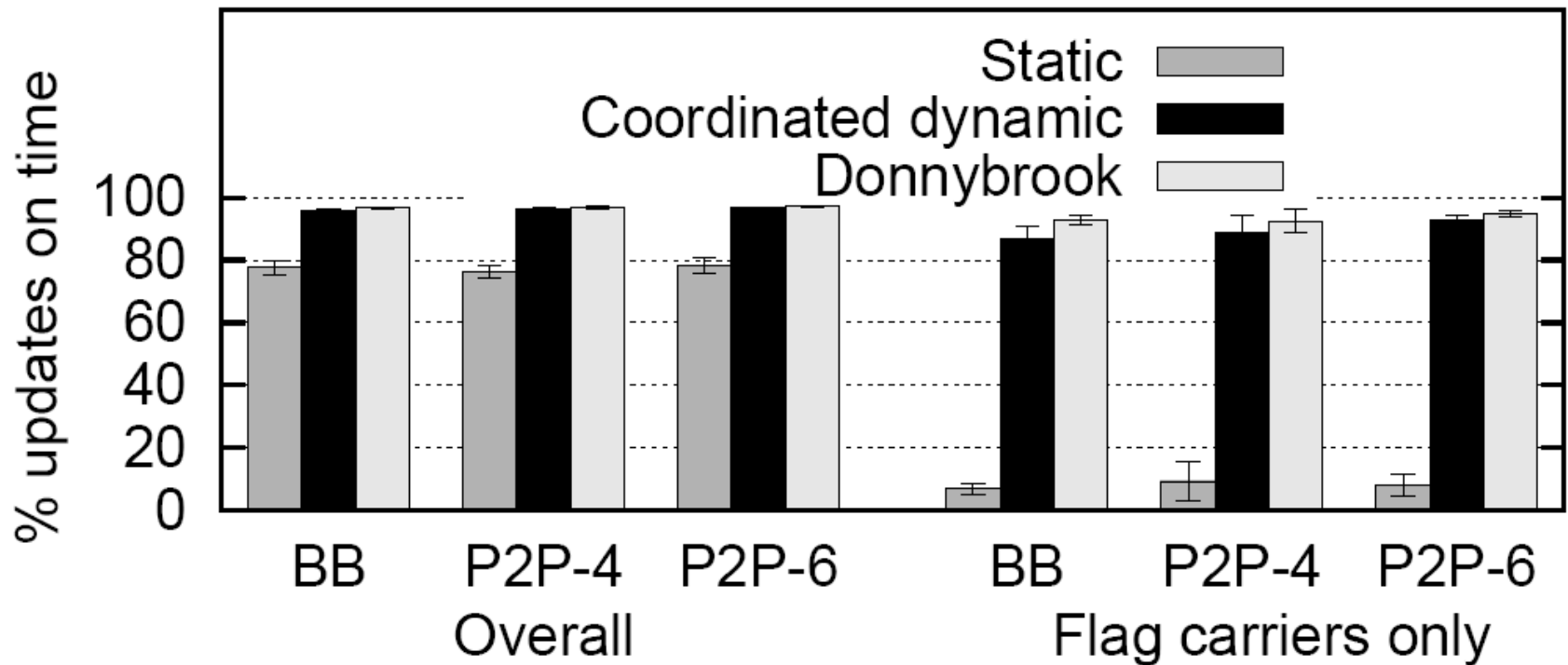
Guidance is almost never stale

Evaluation: Subscriber Set Size



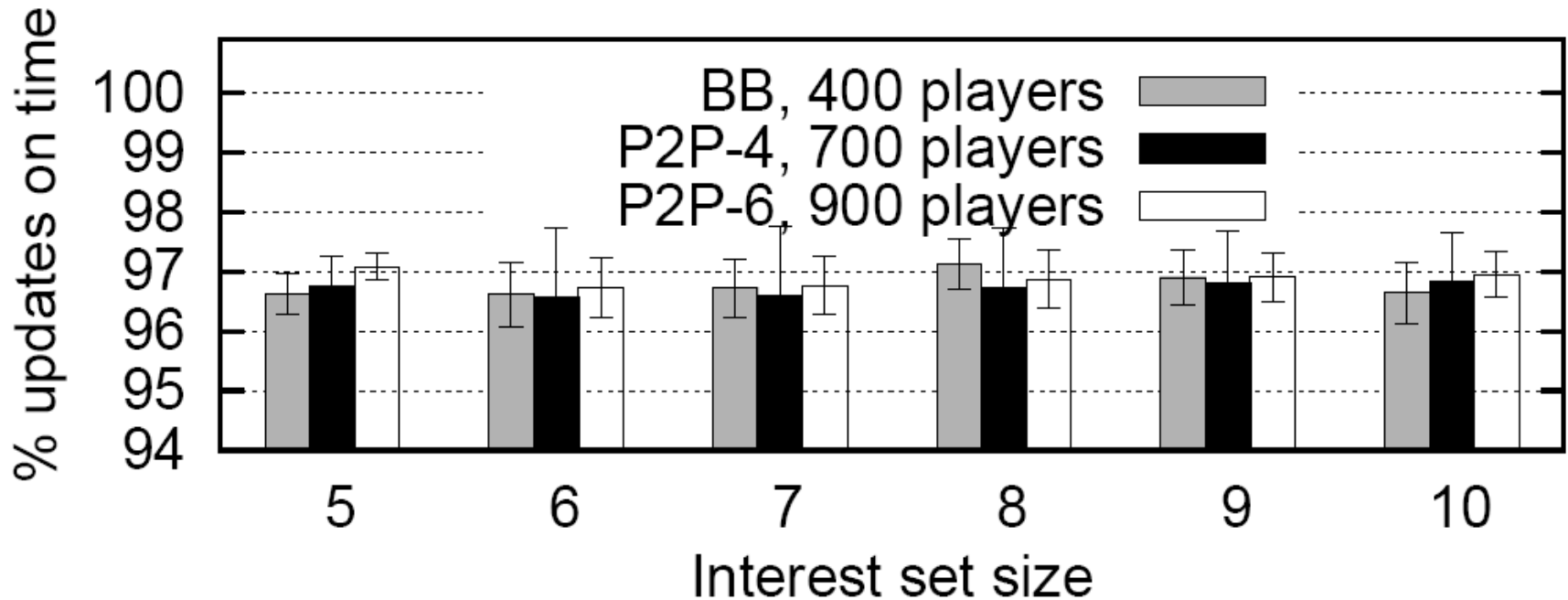
Players with lots of subscribers still get enough updates

Evaluation: Other Approaches



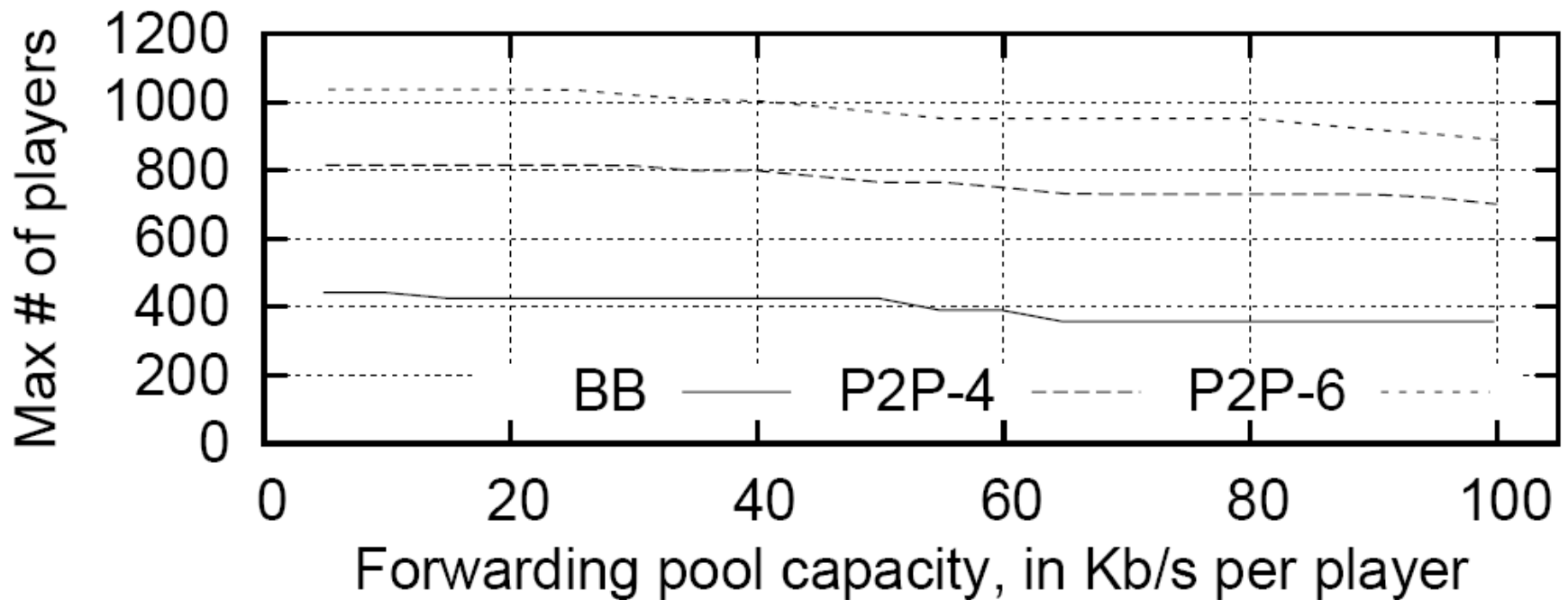
Donnybrook performs better than other approaches

Evaluation: Interest Set Size



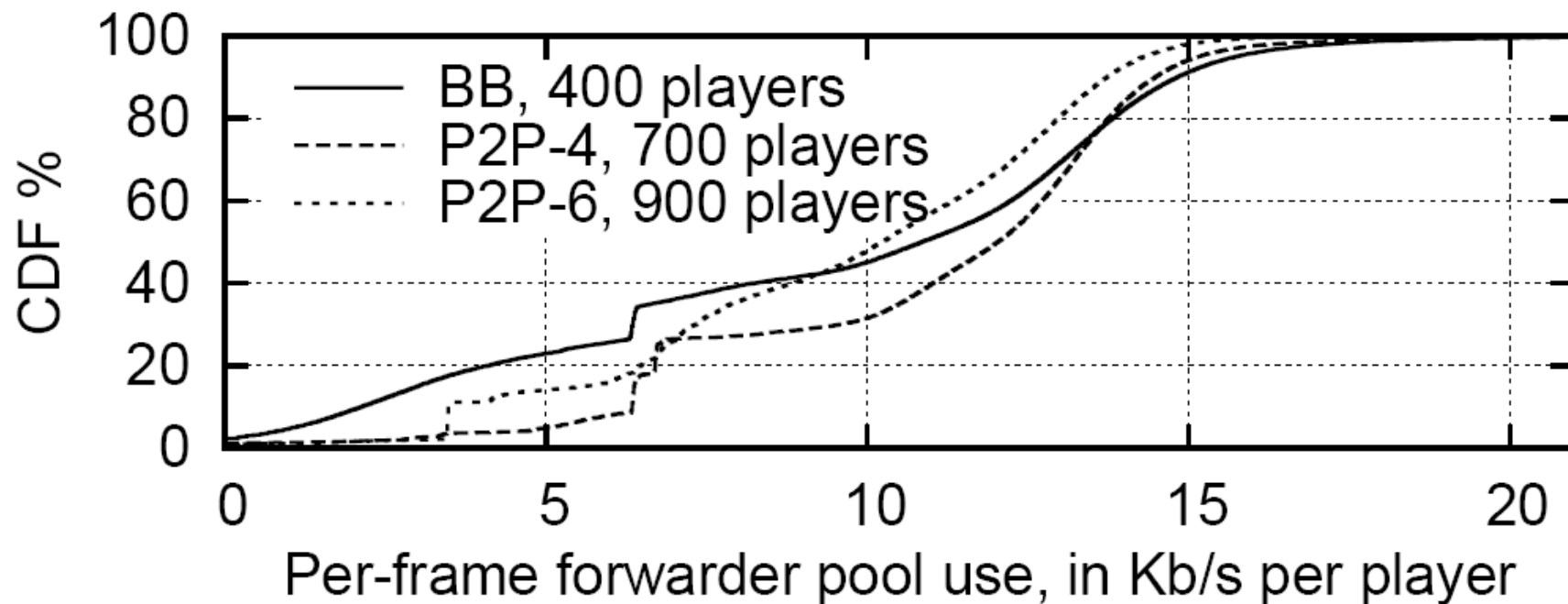
Performance is not sensitive to interest set size

Evaluation: Forwarding Pool Capacity



Capacity set aside does not significantly affect scale

Evaluation: Forwarding Pool Demands



Most forwarding pool requests are small