Blue Banana: handling avatar mobility in distributed MMOGs

Sergey Legtchenko, Sébastien Monnet, Gael Thomas

LIP6/UPMC/CNRS/INRIA

Plan

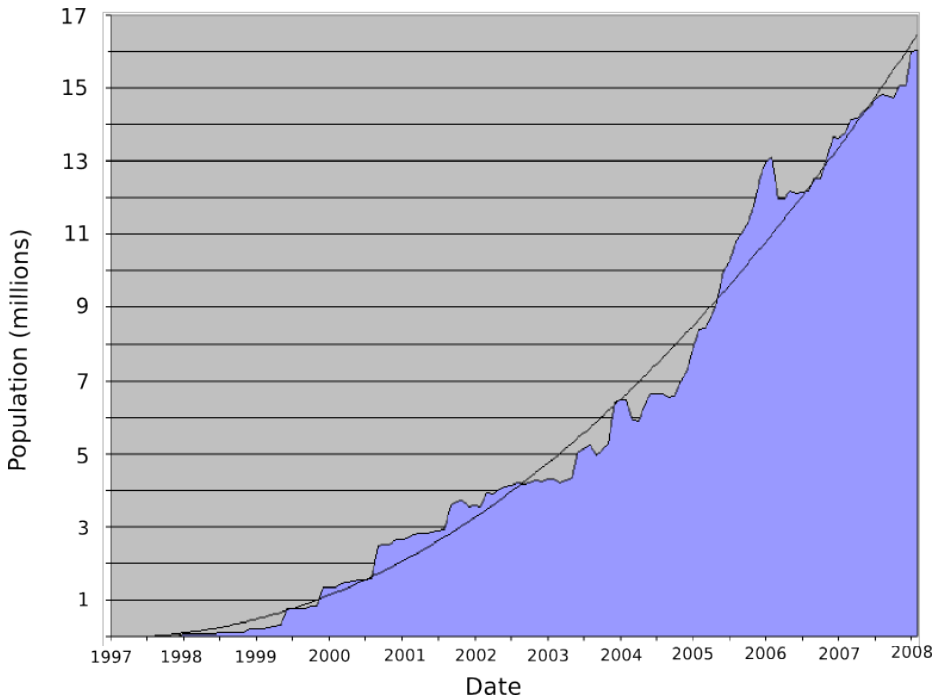
Context : Massively Multiplayer Online Games (MMOGs)

Screenshots of existing MMOGs.

Why decentralize an MMOG application ?

Existing MMOGs :

- Are split into gaming instances to support the load.
- Each instance allows at most about a hundred players.
- Players of different gaming instances are unable to interact.



Cumulated MMOG population.

Decentralizing the MMOG

Properties :

- Possibly millions of users simultaneously interacting.
- The applicative data is spread among the participants.
- An avatar needs to retrieve new data as long as it moves.

Problem :

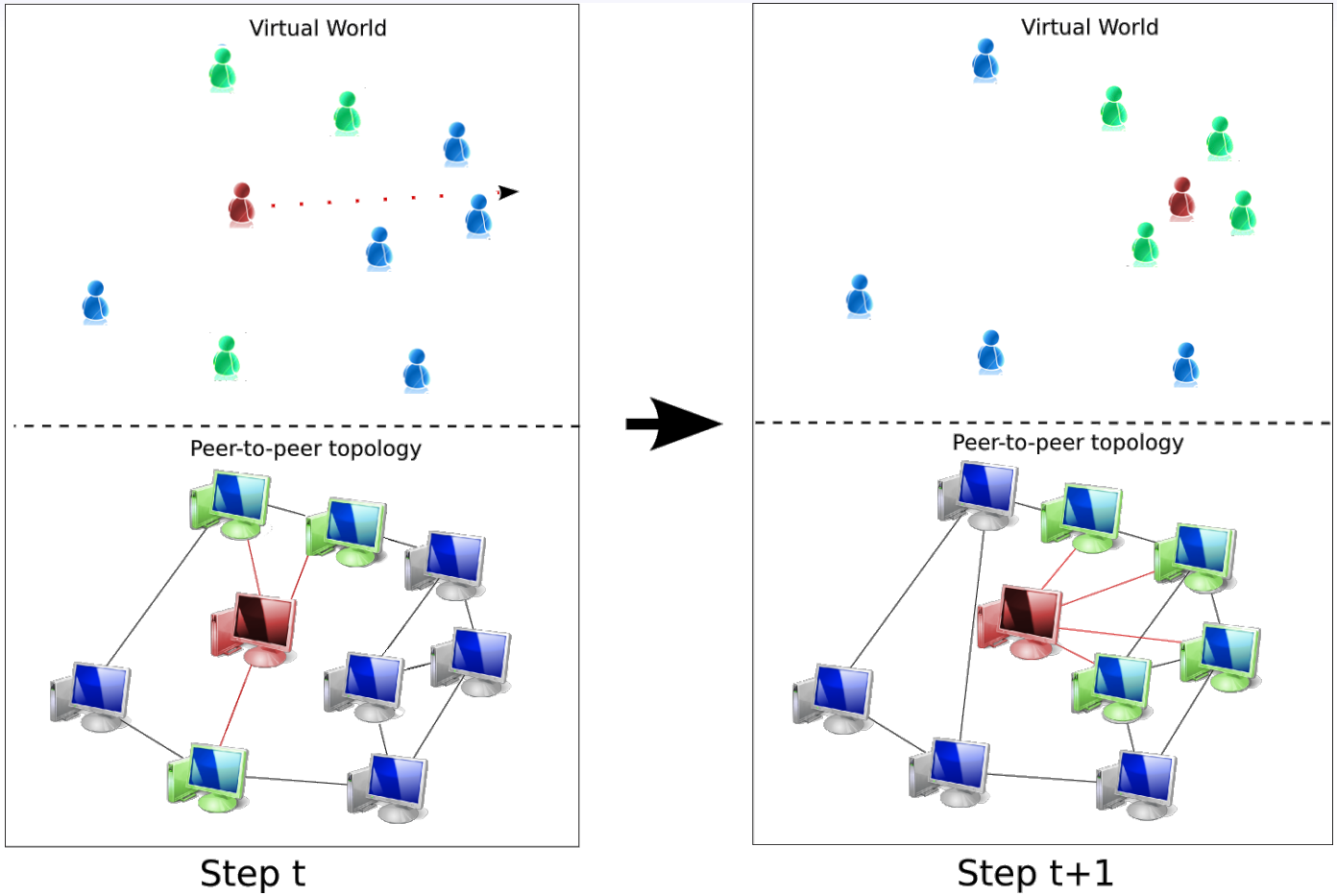
- The application has to be very responsive to avoid failures.

Solution :

- Anticipate avatar movement, and prefetch peers along the predicted trajectory.

Plan

Mechanisms of MMOG overlays



The p2p topology dynamically adapts to the avatar movement.

Transcient failures

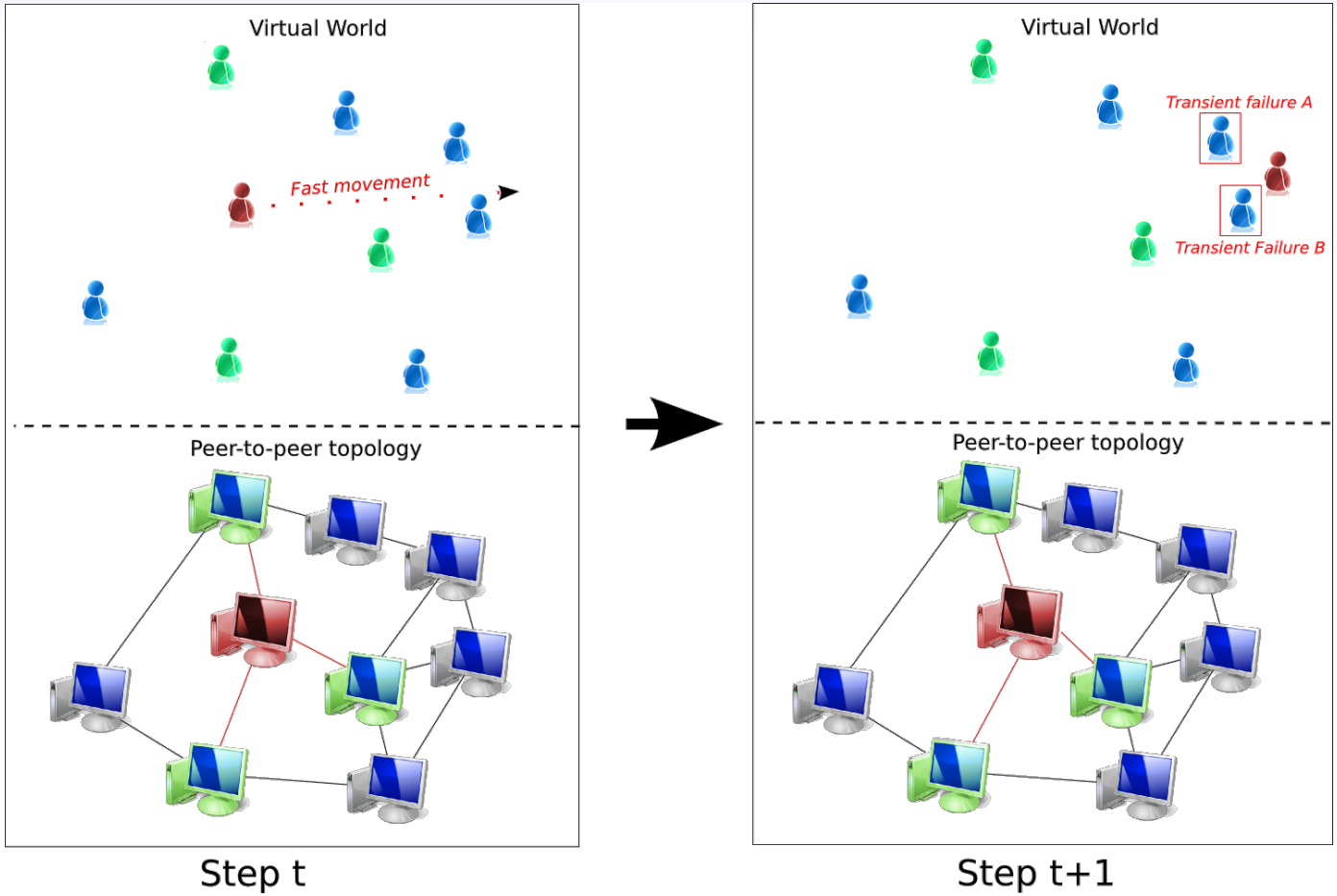
Definition :

There is a transient failure if two players are close in the MMOG but not neighbors in the p2p topology.

Consequences of the failure :

- Additional delay for gameplay-critical updates.
- Inconsistent or jerky view of the Virtual Environement, making the game *unplayable*.

Topology adaptation and transient failures



Delay in topology adaptation leads to applicative transient failures.

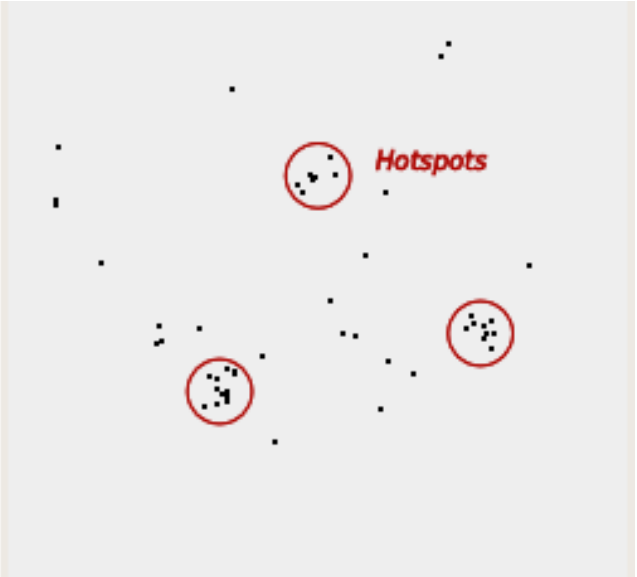
Avatar behavior

Distribution :

- A few high density hotspots
- A lot of desert zones

Mobility :

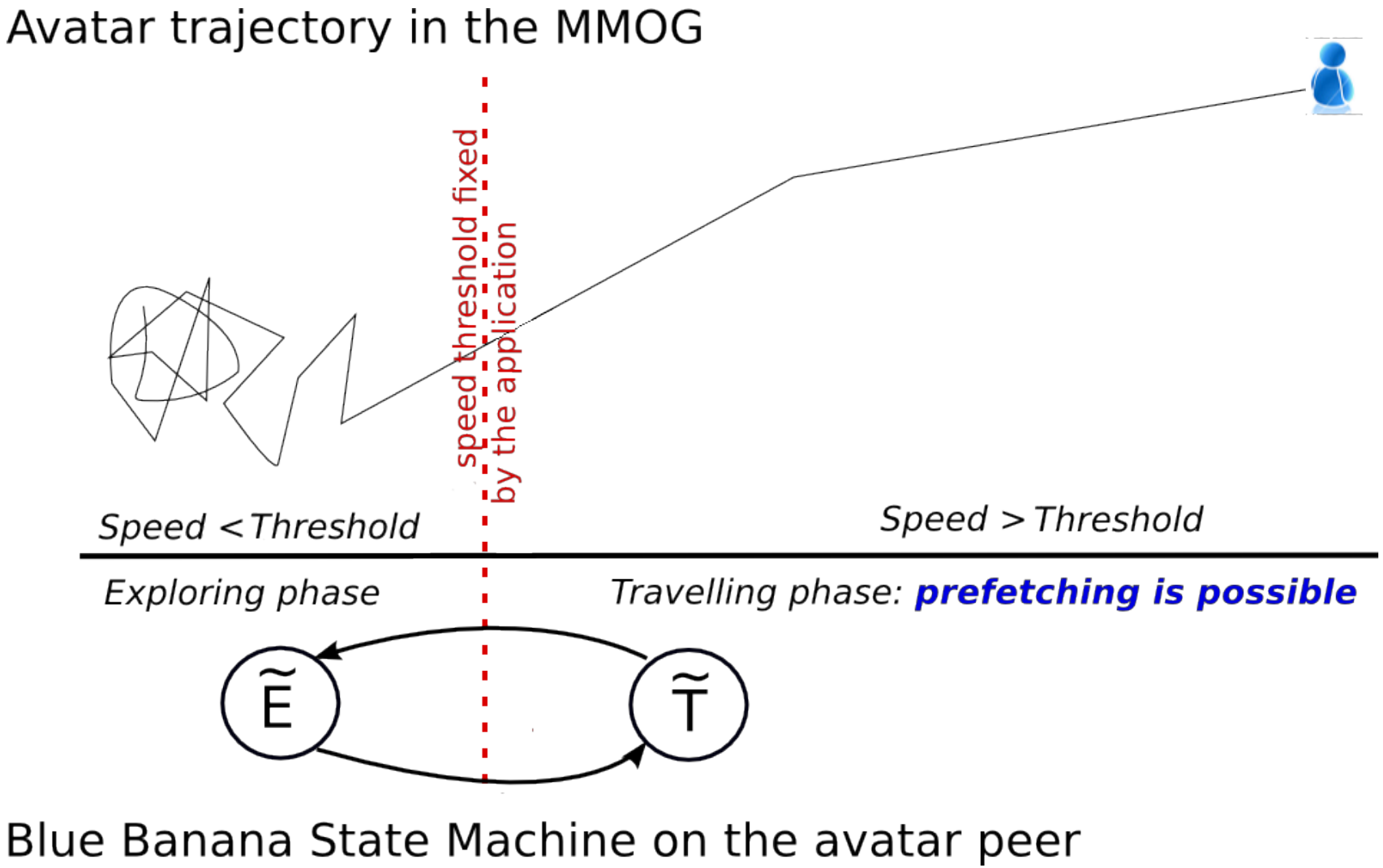
- Slow chaotic movement in the hotspots
- Fast straight movement in desert zones



A sample of Second Life avatar distribution.

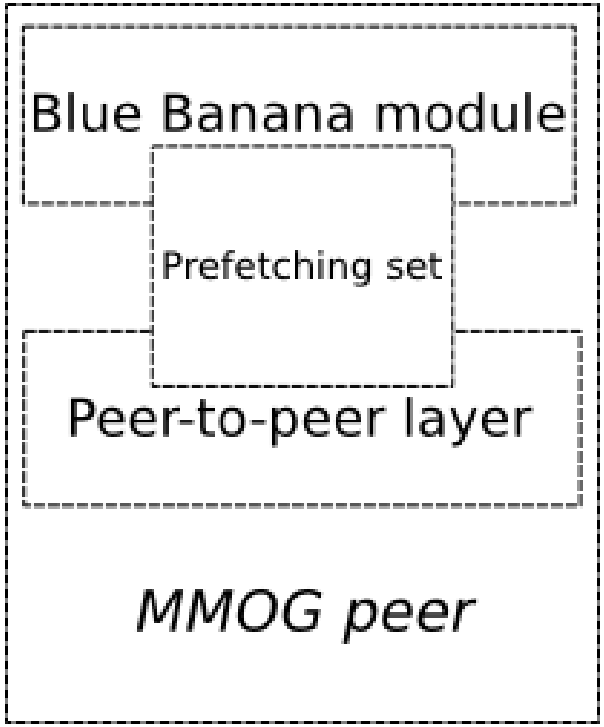
Plan

Movement anticipation



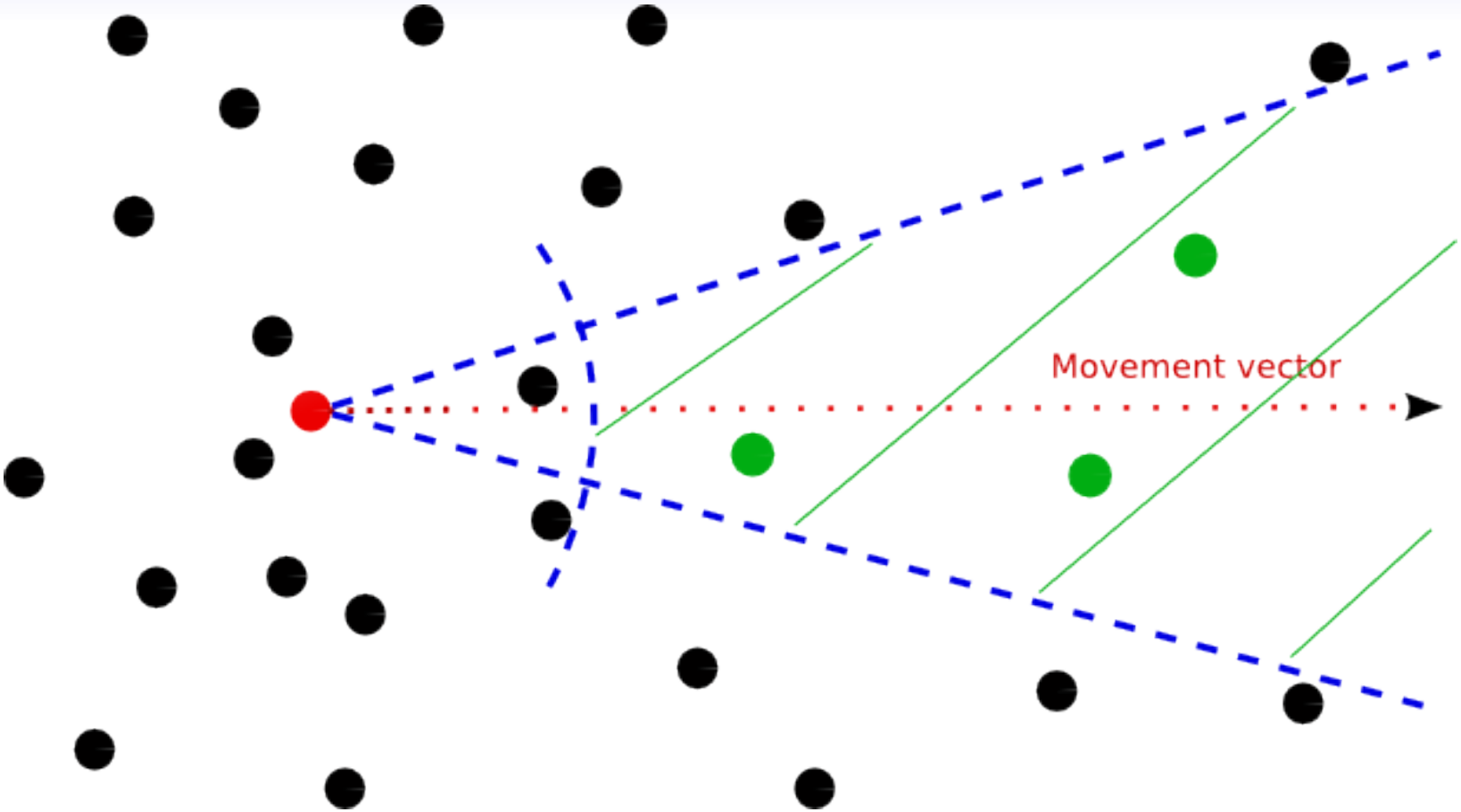
Blue Banana : a prefetching module

- Predicts avatar movement
- Fetches useful peers into the prefetching set
- The p2p layer retrieves prefetched peers from the prefetching set



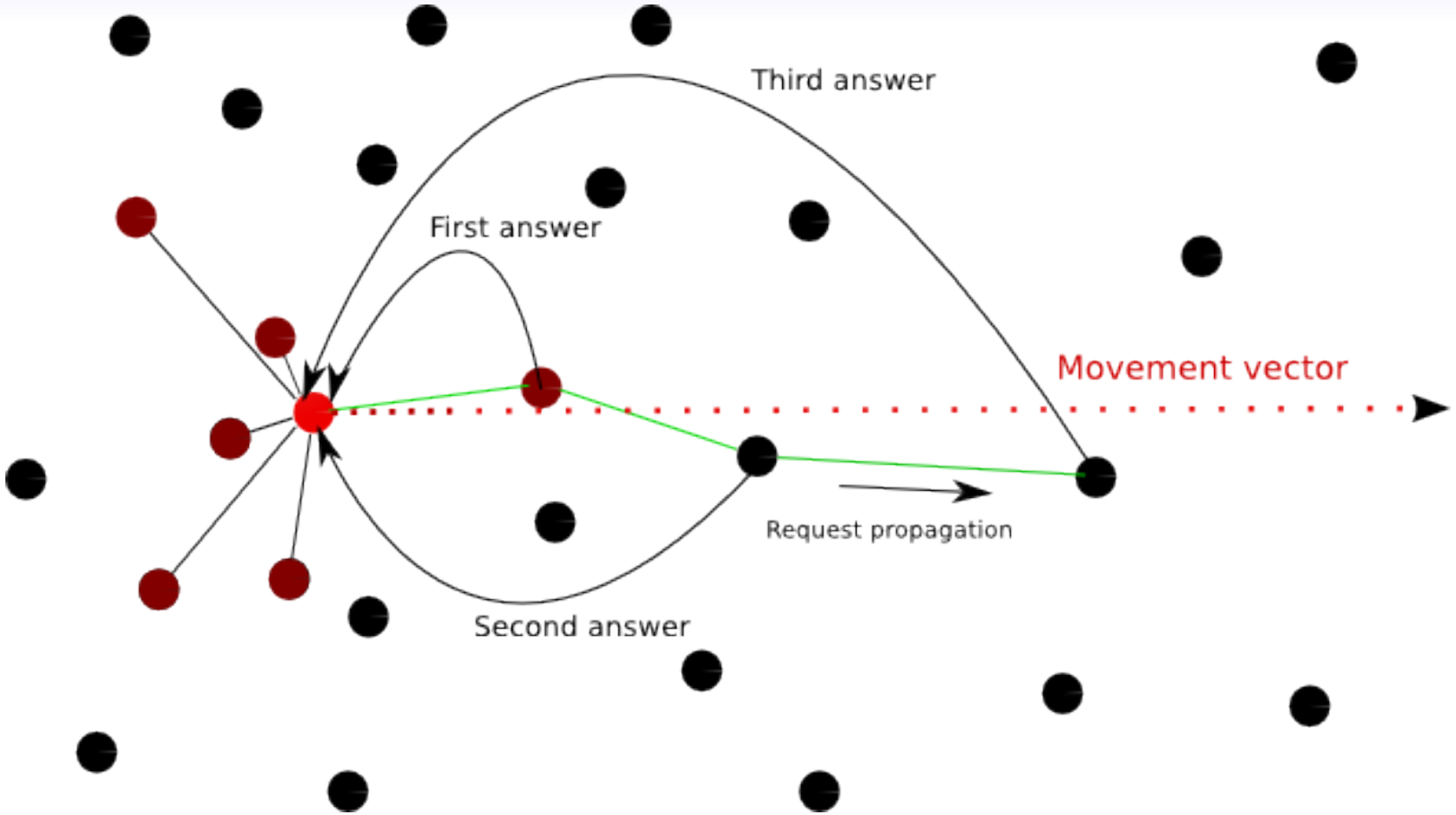
Blue Banana architecture

Choosing peers for prefetching



All peers inside the cone are prefetching candidates.

Prefetching algorithm



The sending of a prefetching request by a moving peer.

Plan

Need for synthetic traces

Traces from existing MMOGs are :

- Small-scaled (due to centralized architecture)
- Hard to obtain

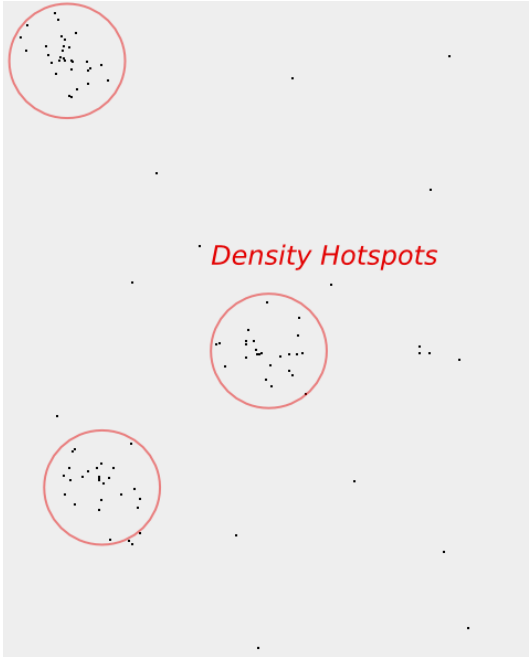
It is essential to build a mobility and distribution model derived from real traces

- Synthetic traces inherit the salient characteristics of real avatar behavior
- Are scalable at will

Trace generator characteristics

Derived from traces gathered from one of the most popular Virtual Environments : **Second Life**.

- **High density hotspots.**
- **Desert zones** uniformly populated with a small proportion of the overall population.
- **Movement :** fast and straight in the desert zones, slow and chaotic inside the hotspots.



Screenshot of a generated distribution

Trace evaluation

Metrics used to compare synthetic traces with Second Life traces :

- Average avatar walk length
- Population density (avatars per cell)
- Clustering coefficient (proportion of the neighbors of an avatar that are neighbors with each other)

For the three metrics, our traces are similar to Second Life traces (more precisions in the publication).

Blue Banana evaluation

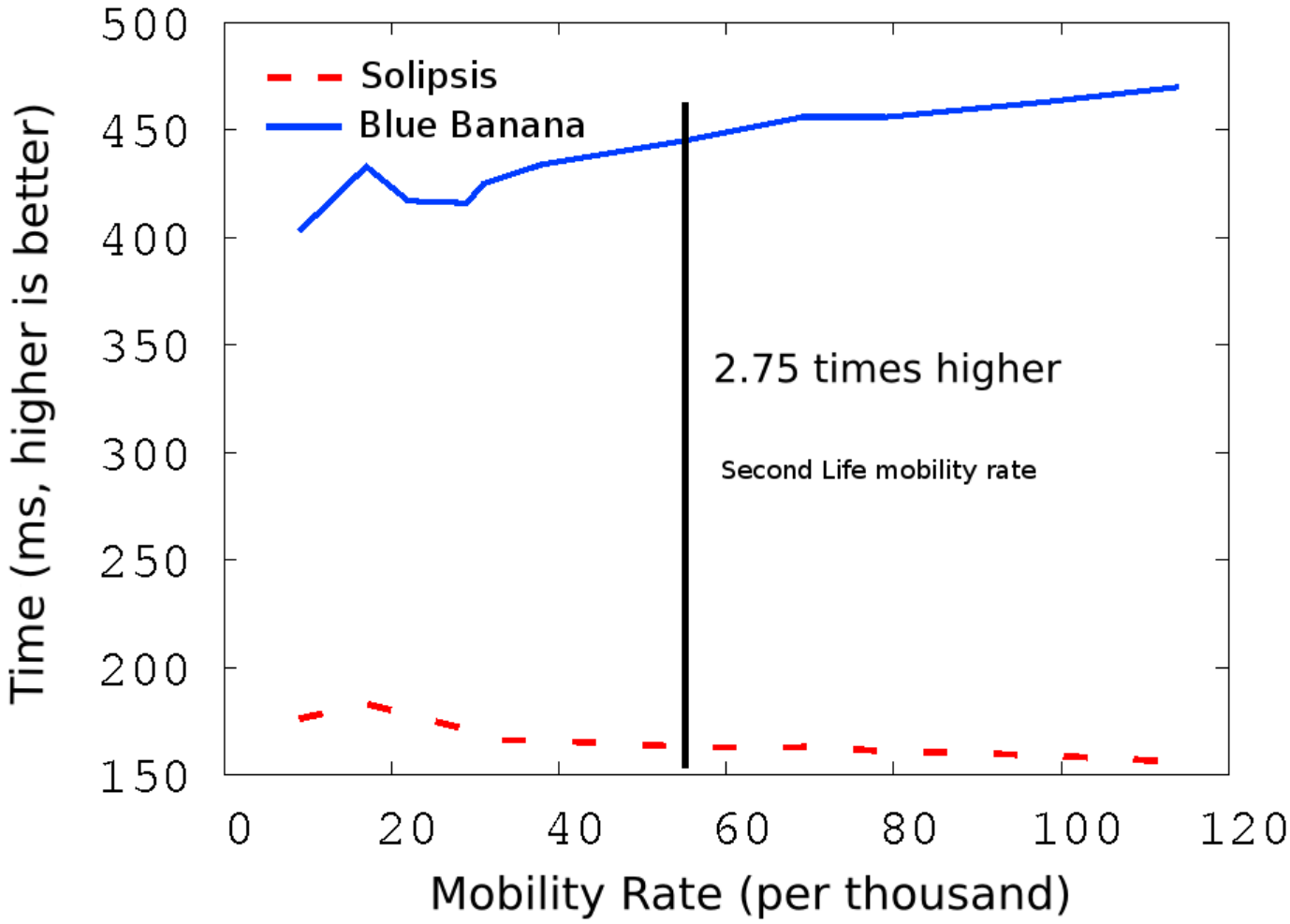
- Use of the discrete event simulator Peersim.
- Injection of the generated traces during the simulation to imitate avatar activity.
- We evaluated the perfomance of our system by varying the rate of fast moving avatars (*the mobility rate*) between 0.5% and 12%, 6% being the rate observed in the Second Life traces.
- Implementation of Blue Banana on top of **Solipsis**, an existing MMOG overlay.

Reasons to choose Solipsis

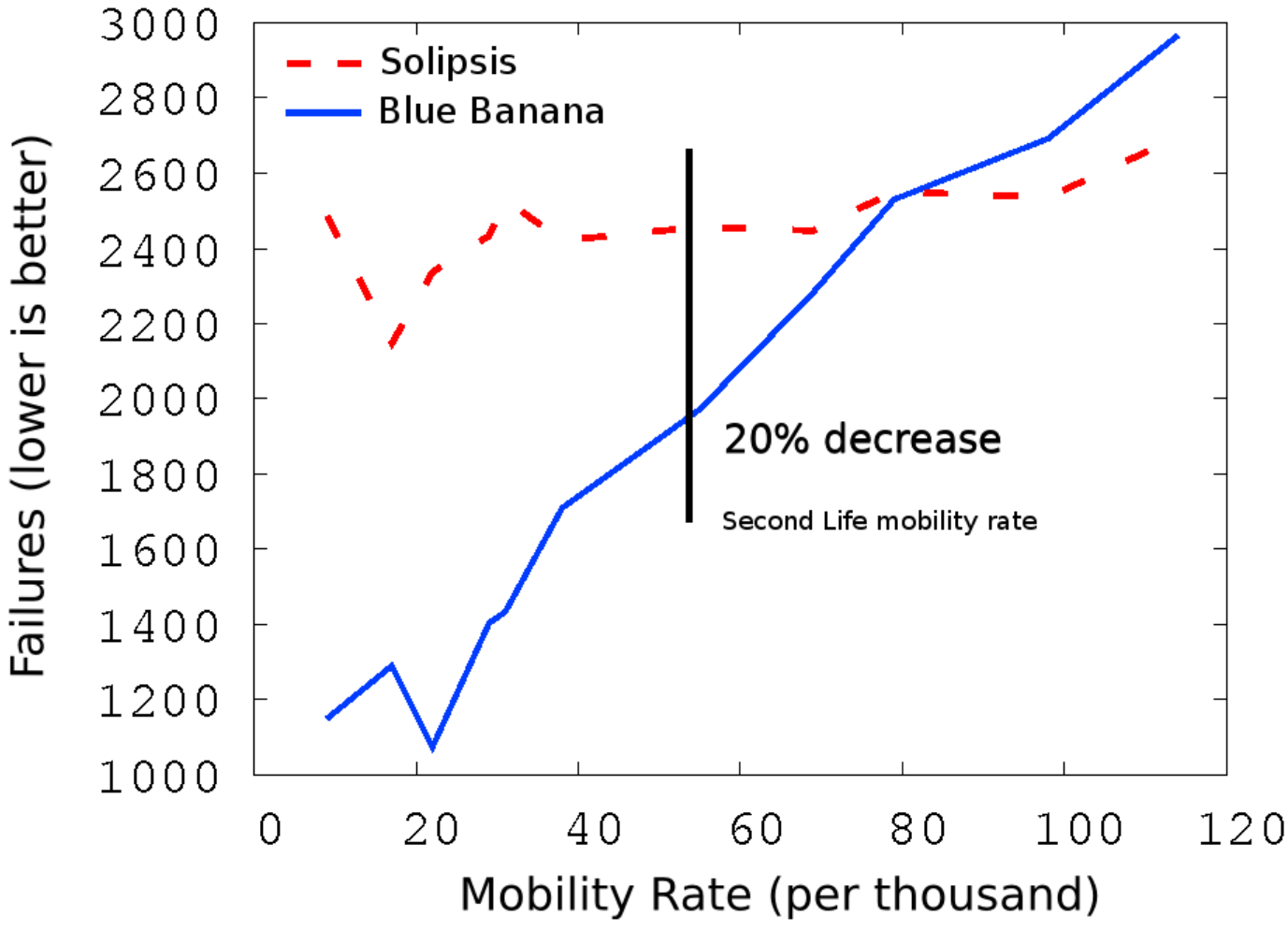
One of the first overlays dynamically modifying its topology to peer virtual movement.

- Convenient for simulation : lightweight and simple.
- Designed for MMOGs : efficient mechanisms to handle peer virtual movement.

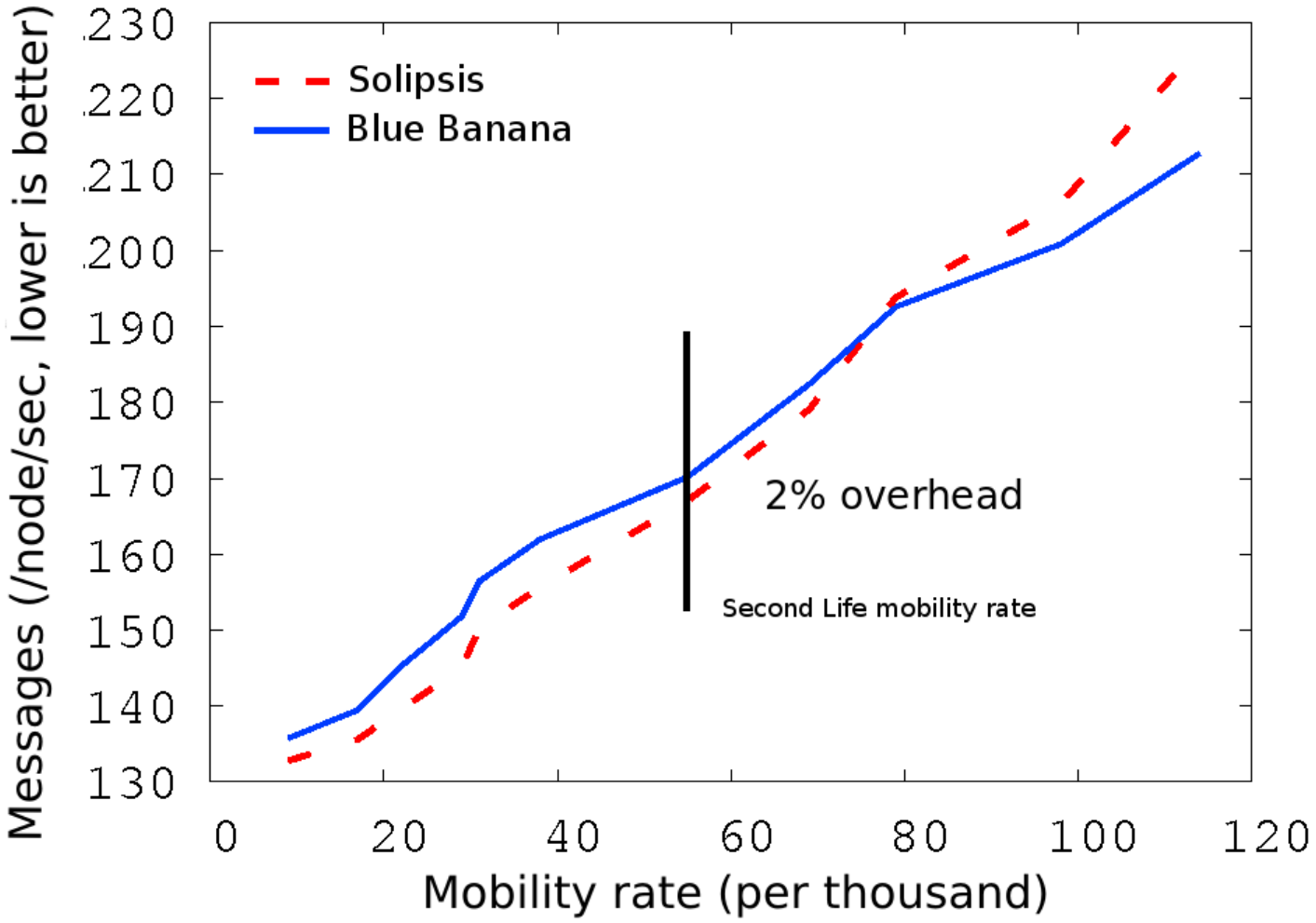
Connection duration with peers ahead of movement



Transient failures : two players are close in the MMOG but not neighbors in the p2p topology



Network cost



Conclusion

Prefetching module :

- Moving node knows the peers ahead of the movement 2.75 times longer.
- Decreases transient failures by 20%.
- Has almost no overhead (only 2%).

Realistic MMOG-avatar behavior trace generator.

Future Work : Enhancement of the behavior prediction with more sophisticated techniques.

Thank you for your attention !

1. Context
2. Avatar movement in MMOG overlays
3. Design of Blue Banana
4. Evaluation

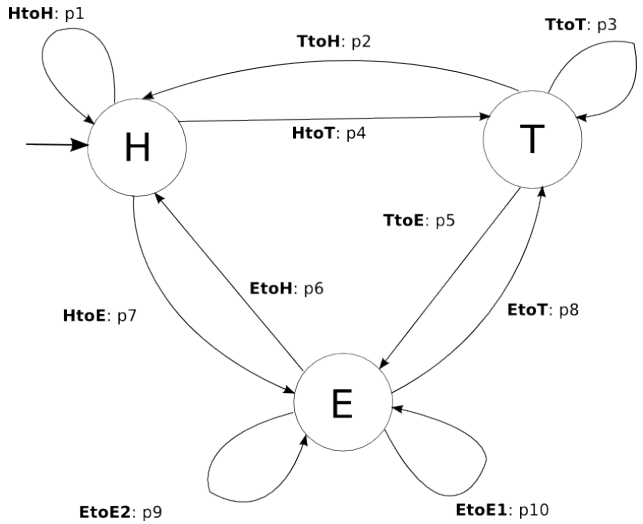
Questions.

Appendix 1 : The trace generator mobility automaton

Possible behaviors :

- *(H)alted :* No movement.
- *(E)xploring :* Slow, chaotic movement.
- *(T)ravelling :* Fast, straight movement.

Given the transition probabilities, each step every avatar decides if its behavior should change.



Synthetic Avatar Behavior State Machine

Appendix 2 : Trace evaluation results