OVERLAY P2P PER AMBIENTI VIRTUALI DISTRIBUITI

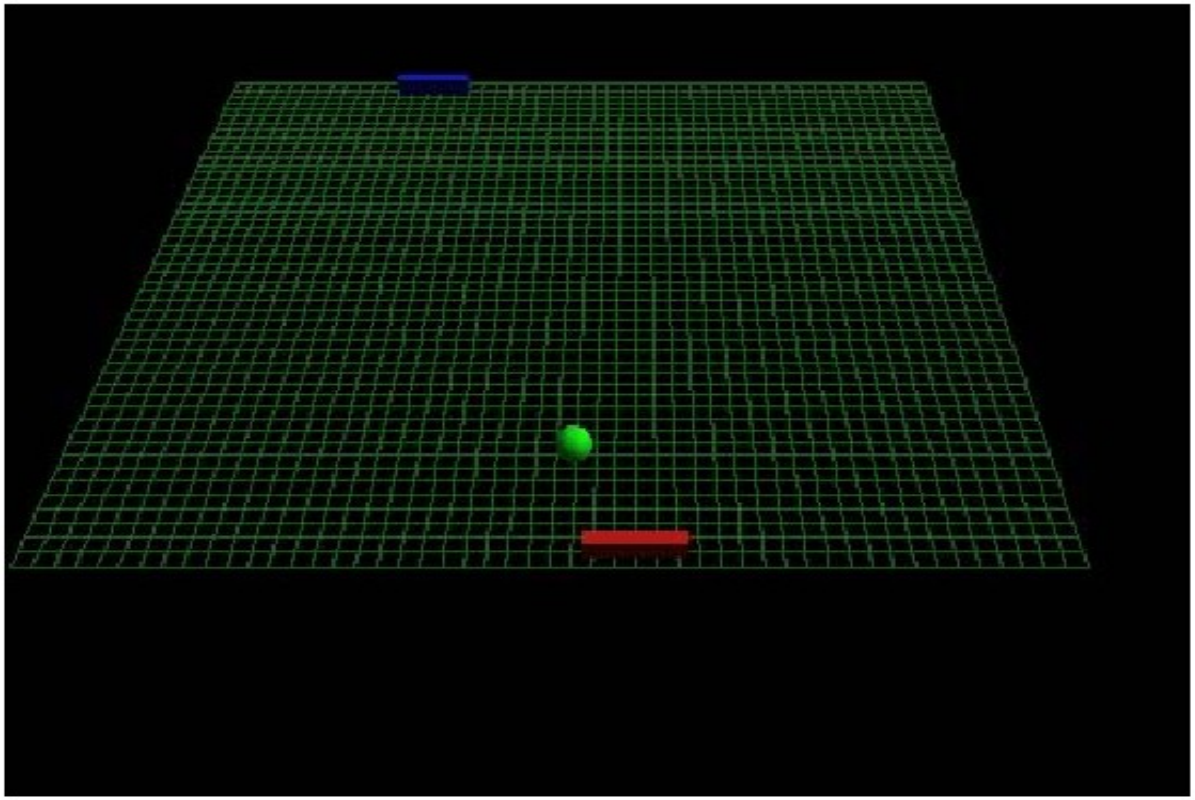
30/3/2009

Laura Ricci

DISTRIBUTED VIRTUAL ENVIRONMENTS

- Distributed Virtual Environments (DVE)
  - Un insieme di Avatars.
    - Interagiscono in un ambiente virtuale.
    - Sono controllati da computers distribuiti geograficamente.
  - Un insieme di oggetti
    - Oggetti passivi.
    - AI controlled (Agenti: stato e codice)
    - Ownership variabile
- Esempi di (D)VE sono:
  - Simulazioni di guerra su larga scala.
  - Massively Multiplayer Online Games (MMOG).
  - Role Play Game (RPG)
  - First Person Shooter (FPS)
  - Simulazioni distribuite di modelli fisici complesse. Ad esempio simulazione di ecosistemi, n-body simulations,etc...

DISTRIBUTED VIRTUAL ENVIRONMENTS



PONG

- 2 giocatori,
- la pallina gestita dai giocatori, ownership variabile

FIRST PERSON SHOOTERS

- La visuale è in prima persona
- Lo scopo primario del gioco è quello di combattere
- Il modello fisico e' molto accurato

FIRST PERSON SHOOTERS

- Comabattimento verso altri giocatori o verso entità controllata dal computer (bots)
- Miglioni di giocatori combattono in un mondo virtuale
- Possibilità di definire squadre, eleggere comandanti cge defisicono la strategia generale della squadra (associazione di chat al gioco)
- Scopo del gioco:accumulare punti uccidendo gli avversari, catturare un trofeo (esempio una bandiera) entrando nella base nemica, impedendo contemporaneamente al nemico di entrare nella propria base
- Ai giocatori può essere permesso di scegliere tra diversi ruoli,caratterizzati da diversi equipaggiamenti, punti di forza,....

ROLE PLAYING GAMES

- La visuale è in terza persona
- Ampio utilizzo di icone e operazioni di trascinamento
- MMORPG Massively Multiplayer Online Games

REAL TIME STRATEGY

Molti avatars guidati da un singolo peer
Basati sulla definizione di tattiche e strategie

CLASSIFICAZIONE EVENTI

- **Modifica dello stato:** evento la cui esecuzione ha un effetto permanente sullo stato.

- **Correlati:** va manteneuta la corretta sequanzializzazione degli eventi sui diversi host

- **Transitori:** eventi la cui esecuzione comporta una momentanea modifica dello stato.

MESSAGGI DI HEARTBEAT

- Con il termine HB si indica un messaggio che fornice informazioni sulla posizione di una entita' nella mappa virtuale in un preciso istante di tempo (Timestamp).

- Gli HB sono inviati con una frequenza generalmente alta, fino a 5 volte al secondo.

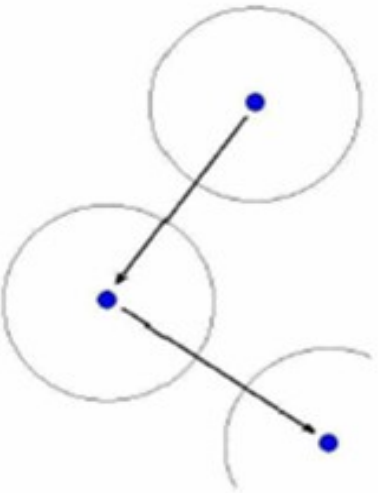
- Ricadono nella categoria di eventi transitori:

  - La ricezione di un HB ***K*** relativo ad un particolare Avatar ***X*** e recante il timestamp con valore ***t***, rende obsolete le informazioni di tutti gli altri HB relativi ad ***X*** recanti un timestamp con valore ***t'*** < ***t*** .

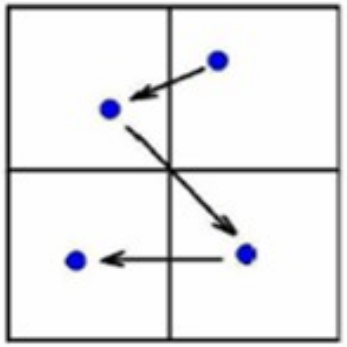
AREE DI INTERESSE

- Area di Interesse di un avatar A= Zona del mondo che contiene le entita' che possono essere 'percepite' da A

- **AOI Mobili:**
  - Area che circonda l'avatar a cui e' associata.
  - Modificata dinamicamente quando l'avatar si sposta.



- **AOI Statiche:**
  - Strettamente legate ad una particolare zona del DVE.
  - Puo' cambiare nel tempo ma non e' legata ad un particolare avatar.

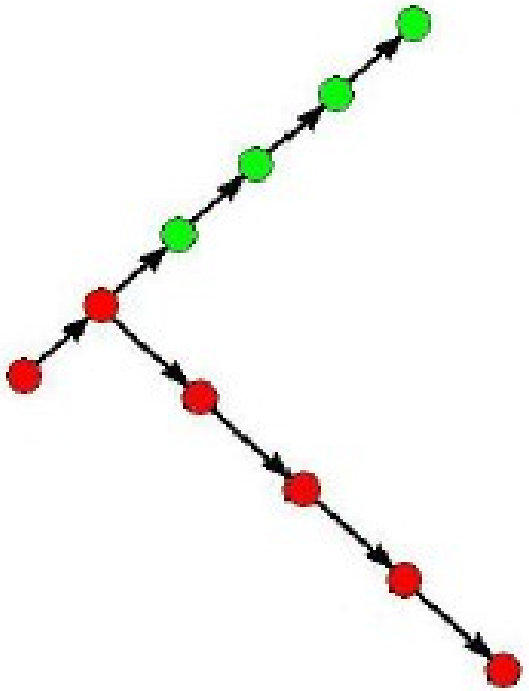


- Usate per migliorare la scalabilita' del DVE.

NOTIFICA DEGLI HEARTBEATS

Notifica statica

- Frequenza stabilita a priori (1/200 ms)
- Per ottimizzare la banda tale frequenza è minore del frame rate.
- Utilizza Dead Reckoning per prevedere la posizione nei frames per i quali non ricevo informazioni.
- Caso pessimo in figura.



DEAD REACKONING

- **Dead Reckoning (DR)**: Nasce in ambiente areonautico per prevedere la posizione di velivoli di cui si e' perso il contatto.

- Sfrutta:

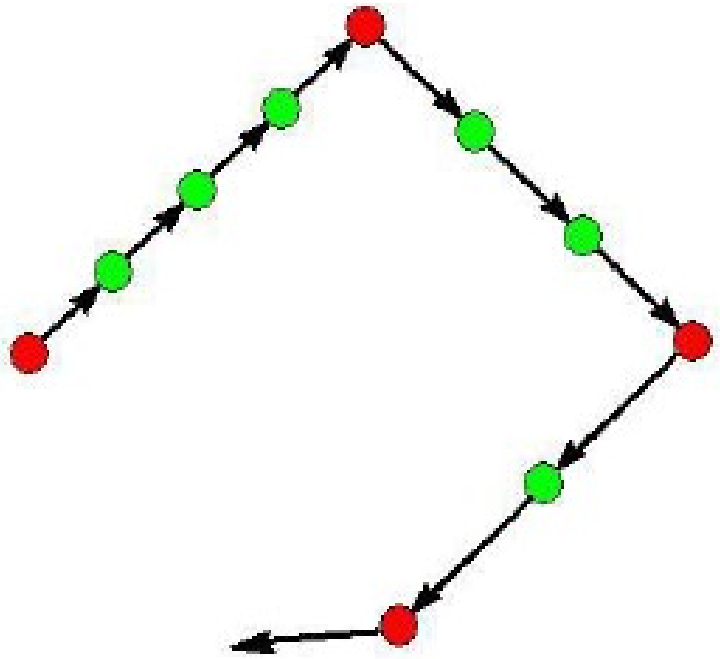
  - Le ultime informazioni ricevute dagli altri avatars
  - Il modello fisico

- Per prevedere, in caso di ritardo nelle notifiche degli HeartBeats, la posizione futura degli Avatars.

NOTIFICA DEGLI HEARTBEATS

Notifica dinamica

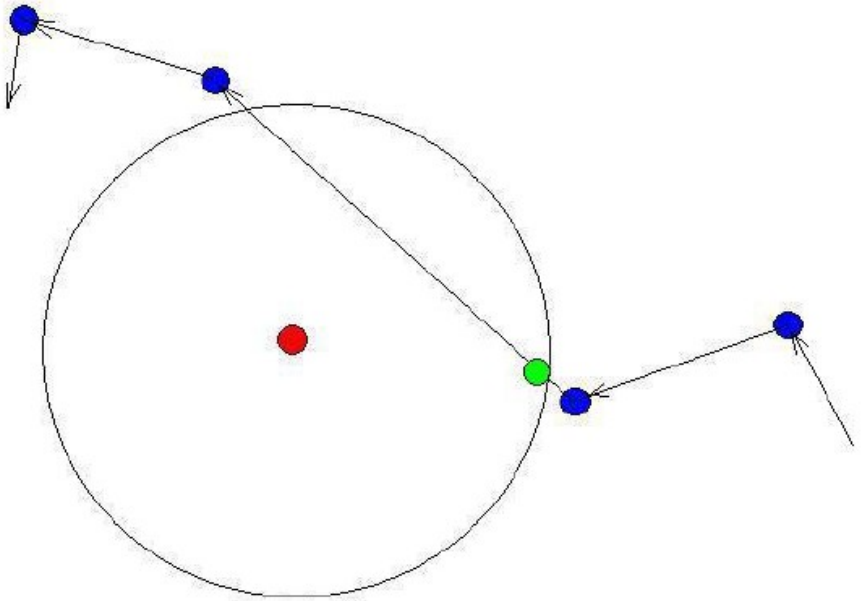
- Utilizzabile se gli spostamenti sono deterministici
- Frequenza variabile
- Utilizza DR localmente per capire quando inviare un Heartbeat di correzione.
- *In generale* si notifica quando nella previsione di una delle grandezze fisiche si commette un errore superiore alla soglia scelta per tale grandezza.



NOTIFICA DEGLI HEARTBEATS

- Aree Fisse: Quando l'Avatar entra in una nuova area si spedisce un HeartBeat a tutti gli Hosts dell'area.

- Aree Mobili: l'individuazione dinamica dello spostamento di un avatar in un'area risulta complessa

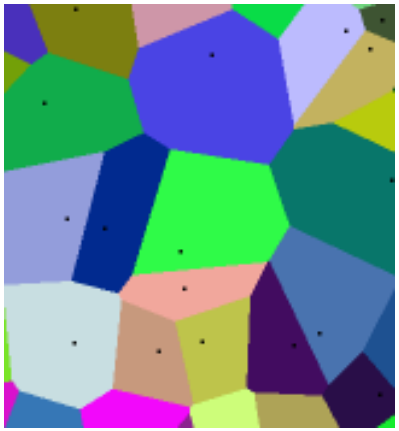


OVERLAY PER DVE

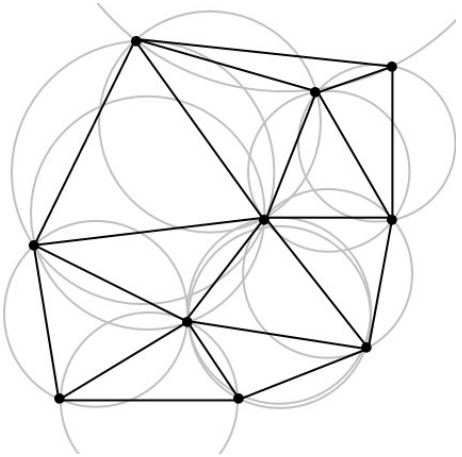
- L'overlay dovrebbe contenere
  - Almeno un insieme di collegamenti tra peer fisicamente vicini nell'ambiente virtuale
  - Eventualmente un collegamento tra un peer e tutti gli altri peer collocati nella propria area di interesse
- Overlay altamente dinamica
- I collegamenti cambiano a causa dei movimenti dei peer
- Uso di protocolli 'leggeri' a livello trasporto
- Necessari meccanismi per l'acquisizione dinamica della conoscenza di nuovi poeer
- I meccanismi devono garantire che non si verifichino disconnessioni all'inteno del mondo virtuale

DIAGRAMMI DI VORONOI

- Dato un insieme di punti S nel piano, il diagramma di Voronoi per S e' la partizione del piano che associa ogni regione V(p) con ogni punto p di S in modo che ogni punto del piano in V(p) risulta **piu' vicino** a p rispetto che a ogni altro punto in S.

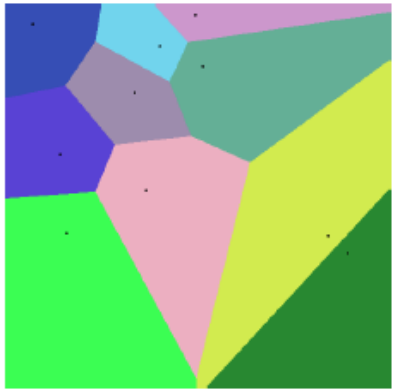


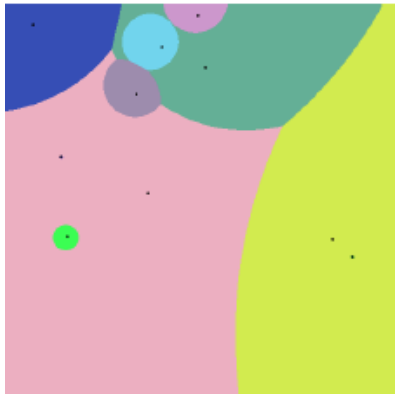
- **Proprieta'**
  - Il grafo duale per un diagramma di Voronoi, corrisponde alla triangolazione di Delaunay per lo stesso insieme di punti S.
  - Le coppie di punti a due a due piu' vicini, corrispondono a 2 celle adiacenti nel diagramma di Voronoi.



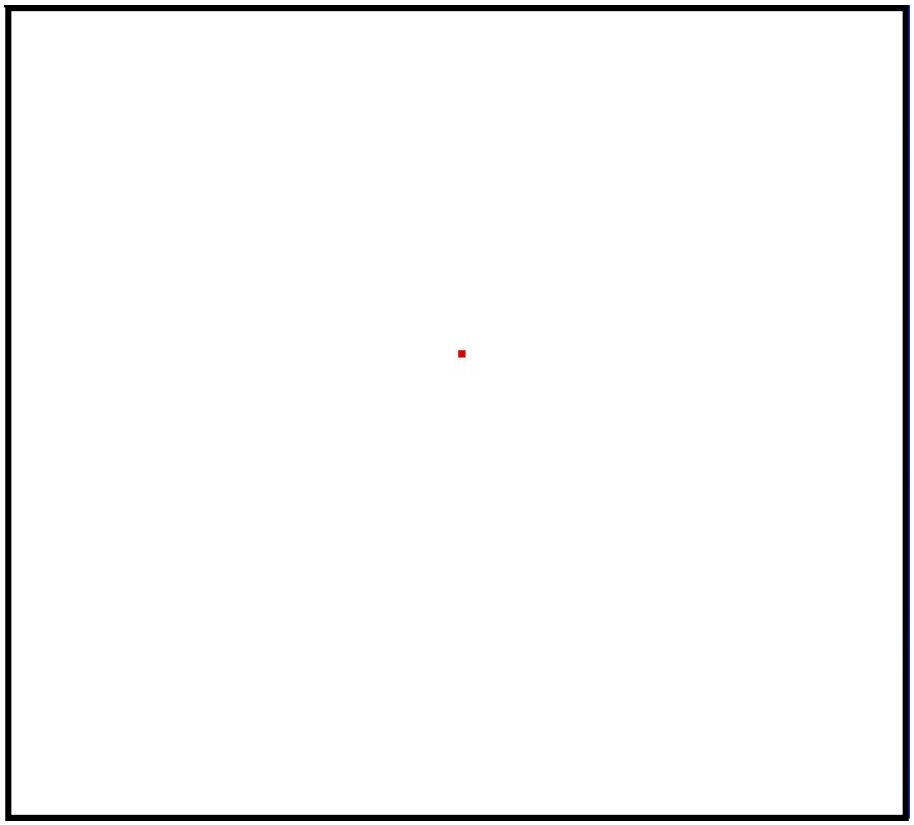
DIAGRAMMI DI VORONOI PESATI

- Diagrammi Pesati di Voronoi: Associano un peso ad ogni nodo.
  - Additivi
  - Moltiplicativi
- Idea: Determinare le aree di Voronoi in base al peso dei nodi.
- Bilanciamento del carico:
  - Peers con molte risorse (CPUs, bandwith,..):
    - Sono descritti da pesi maggiori
    - Gestiscono un maggior numero di oggetti

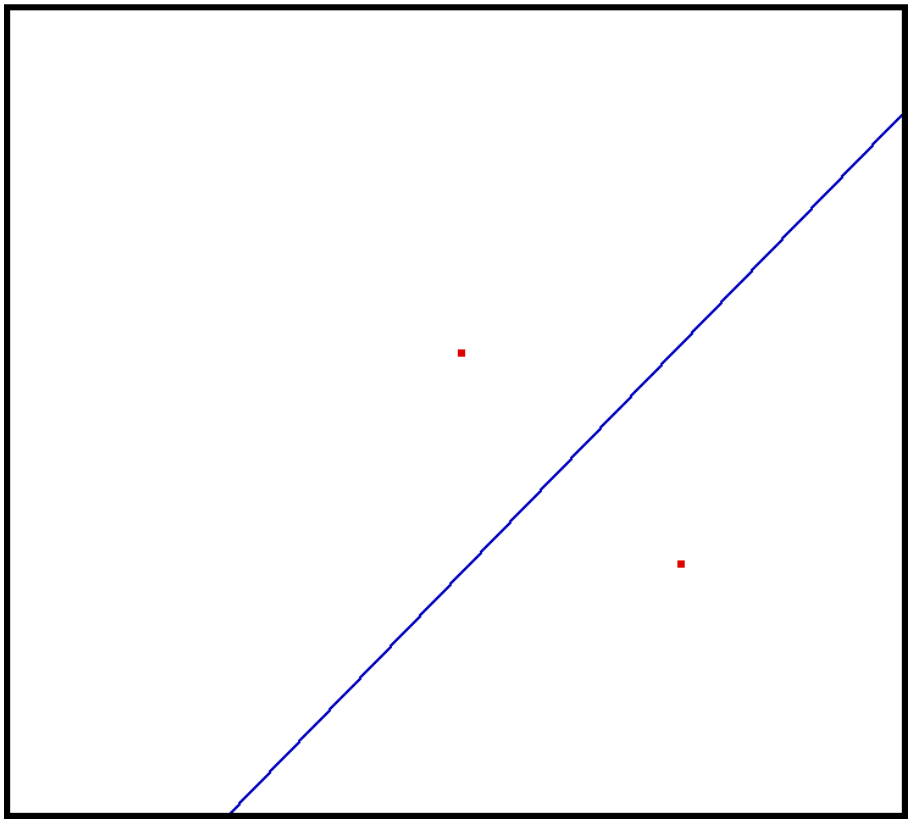




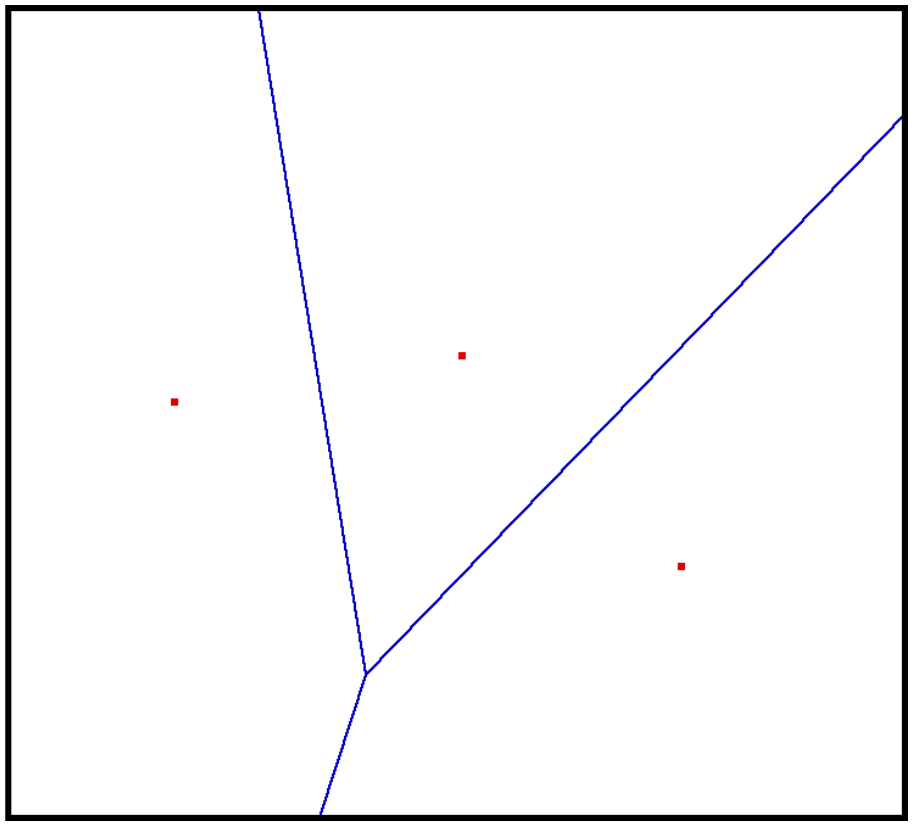
DIAGRAMMI DI VORONOI



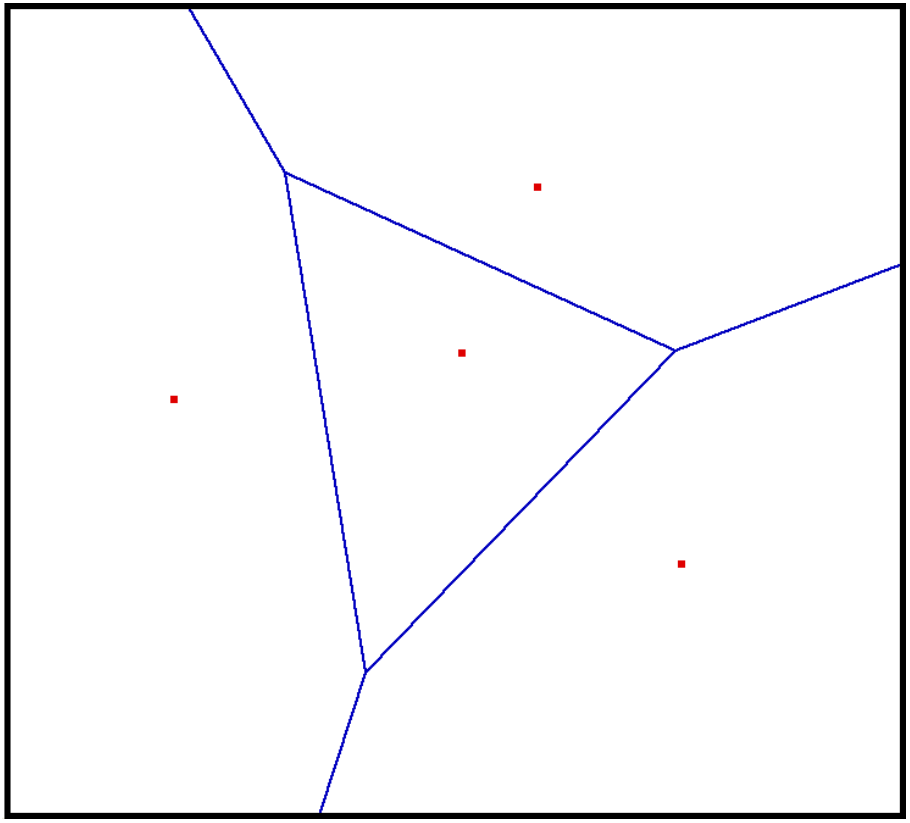
DIAGRAMMI DI VORONOI



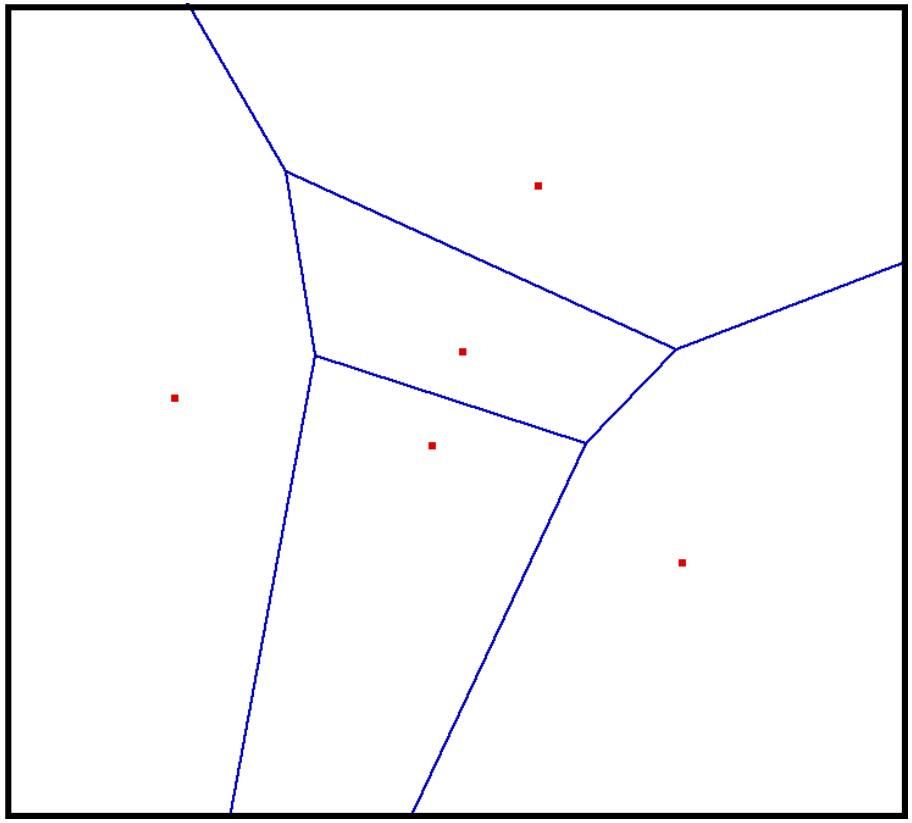
DIAGRAMMI DI VORONOI



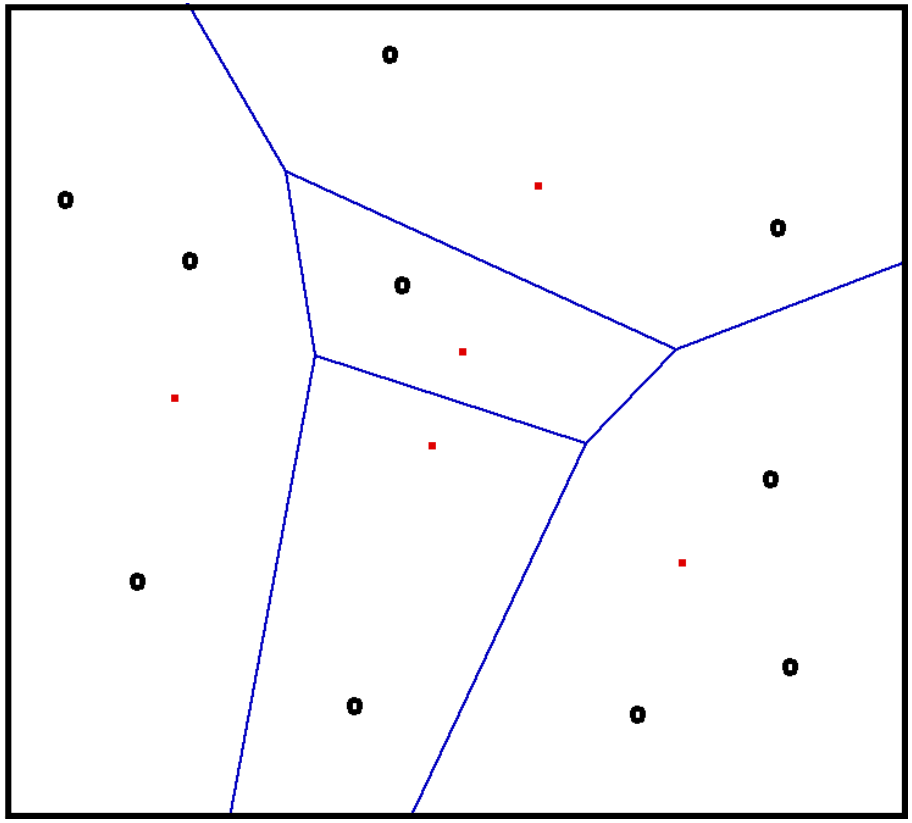
DIAGRAMMI DI VORONOI



DIAGRAMMI DI VORONOI

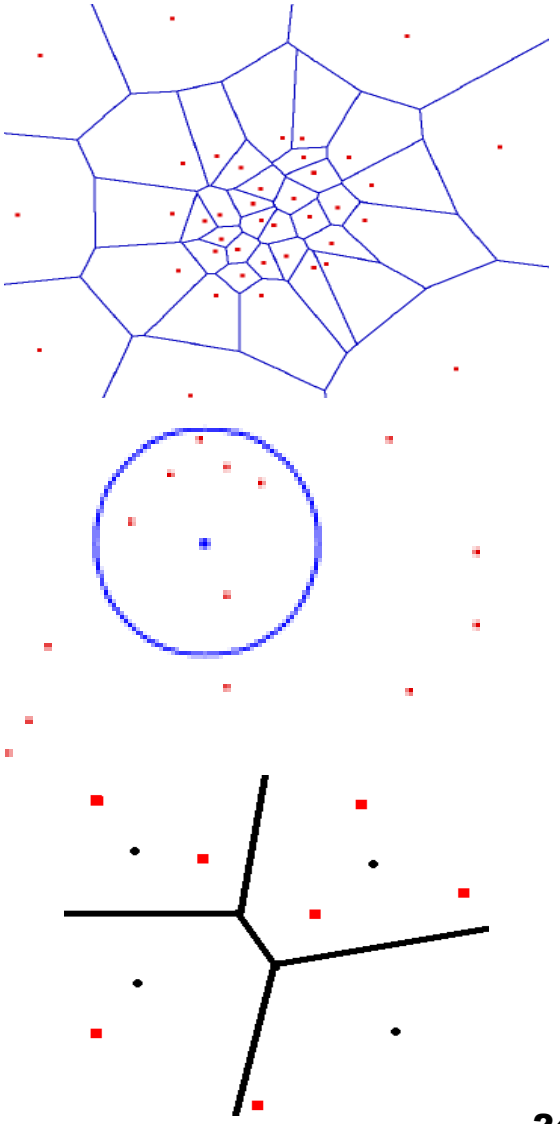


DIAGRAMMI DI VORONOI



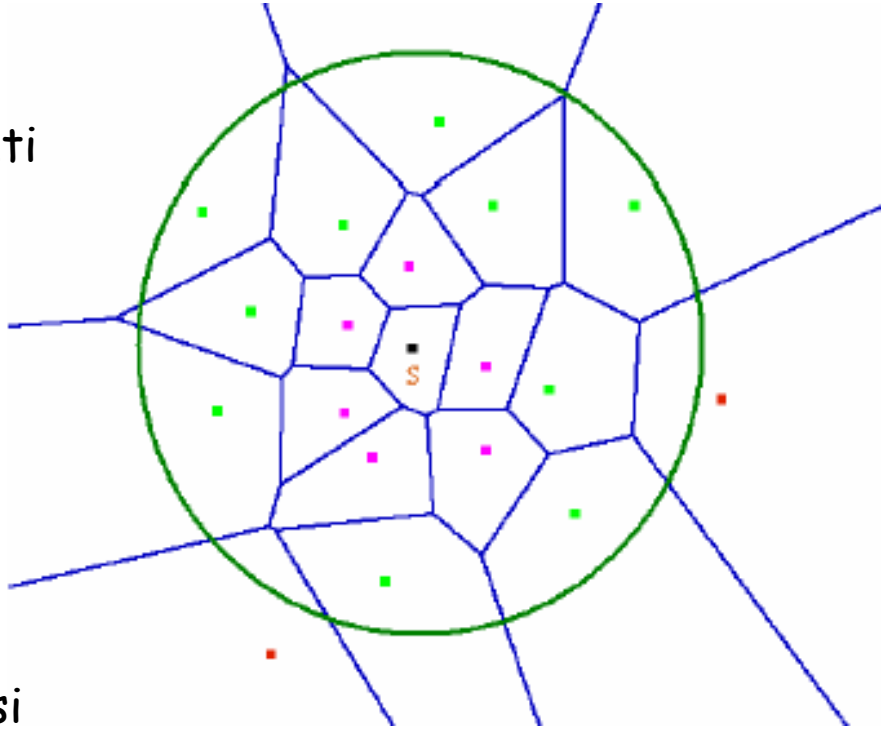
OVELAYS BASATE SU DIAGRAMMI DI VORONOI

- Un'architettura P2P basata sui diagrammi di Voronoi.
- Scopo:
  - Incrementare la scalabilita' dei DVE.
  - Sfruttare AOI per preservare la localita' delle rinformazioni.
  - Definire una strategia distribuita per la gestione degli oggetti passivi.



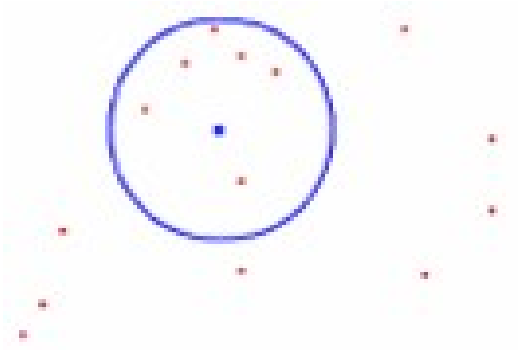
OVERLAY BASATE SU DIAGRAMMI DI VORONOI

- Ogni nodo **n** costruisce e mantiene un diagramma di voronoi basato sulle coordinate spaziali dei vicini appartenenti all'AOI(**n**) .(Nodi viola)
- Ogni nodo crea connessioni dirette con tutti i suoi vicini nel grafo di Voronoi ed eventualmente mantiene connessioni dirette anche con gli altri nodi appartenenti alla sua AOI. (Nodi verdi)
- Le connessioni sono mantenute basandosi sullo spazio virtuale e non sulla prossimita' fisica dei Peers.

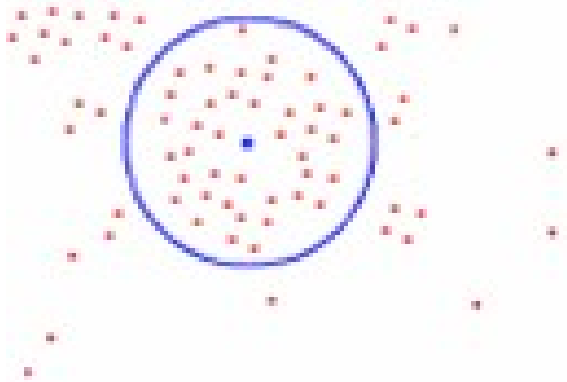


OVERLAY BASATE SU DIAGRAMMI DI VORONOI

- Per mantenere una visione locale, l'overlay deve essere aggiornata fino a qualche volta al secondo.
  - Ogni volta che un peer riceve un heartbeataggiornamento dell'overlay



- L'uso di AOI incrementa la scalabilita', ma nei DVE possono manifestarsi fenomeni di crowding.
  - Battaglie.
  - Citta' virtuali.



- In questi casi un alto numero di avatars interagisce in un ristretto spazio virtuale e la scalabilita' e' a rischio anche con le AOI.

OVERLAY BASATE SU DIAGRAMMI DI VORONOI

- I DVE pongono vincoli real time molto forti.
  - La consistenza dello stato deve essere preservata il prima possibile.
  - Usualmente una soglia temporale deve essere rispettata dalla notifica degli eventi.
- Il calcolo della visione locale della overlay deve essere veloce.
  - Ogni peer:
    - Riceve tutti gli eventi dalla propria area di interesse.
      - Heartbeats (Spostamennti delle Entita')
      - Eventi (Azioni delle Entita').
    - Invia un HeartBeat agli altri peers con un'alta frequenza (fino a 5 volte al sec)

OVERLAY BASATE SU VORONOI

- La definizione di una Overlay basata su Voronoi richiede una gestione efficiente del il grafo di Voronoi locale che rappresenta la visione locale dell'overlay su ogni peer.

- La libreria deve fornire il supporto per:
  - Inserimento di un nodo.
  - Trovare i vicini di un nodo.
  - Trovare il nodo la cui area contiene un particolare punto.

- Libreria VAST: http://vast.sourceforge.net/VON/

GESTIONE OGGETTI PASSIVI

- L'overlay topology deve essere Voronoi-Based
  - Ogni punto della mappa virtuale deve appartenere all'area di Voronoi di un Peer.

- Coordinatore di un oggetto passivo *O* = Il peer la cui area di Voronoi include **O**. (Nota: Un grafo pesato di Voronoi puo' essere utilizzato per bilanciare il numero di oggetti asseganti ad ogni Peer)

- Se un Peer modifica l'oggetto **O**, invia la richiesta di update al coordinatore di **O**, il quale agira' come un server temporaneo per **O**.

- La coordinazione e' sempre delegata.

- Lo stato di un oggetto e' replicato almeno su tutti i Peers che sono entro la sua visibility area.

- Un grafo pesato di Voronoi puo' essere utilizzato per bilanciare il numero di oggetti asseganti ad ogni Peer)

GESTIONE OGGETTI PASSIVI

- Se un coordinatore **OW** di un oggetto **O** è attivo allora quando **O** entra nell'area di Voronoi di un vicino, **OW** delega la coordinazione di **O** a quel vicino.

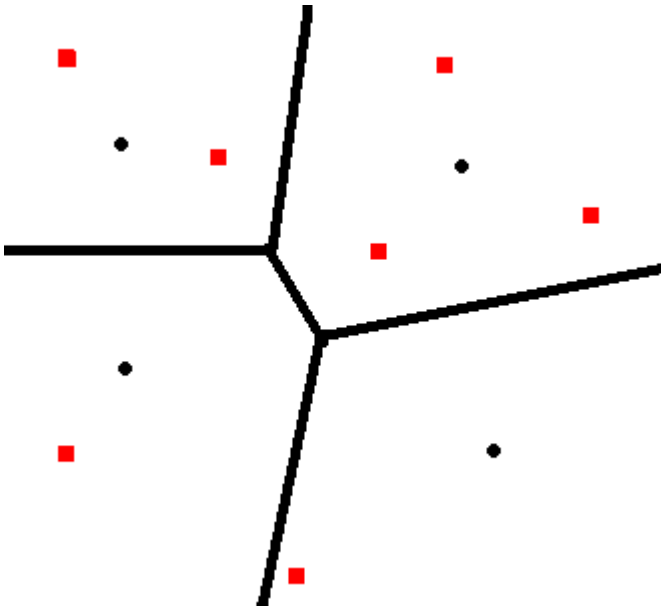
- Se il coordinatore **OW** di un oggetto **O** cade senza delegare la coordinazione allora il peer piu' vicino ad **O** non riceve piu' HeartBeat da **OW**, e dopo un certo intervallo di tempo si autoelegge a coordinatore per **O**.(Unica eccezione)

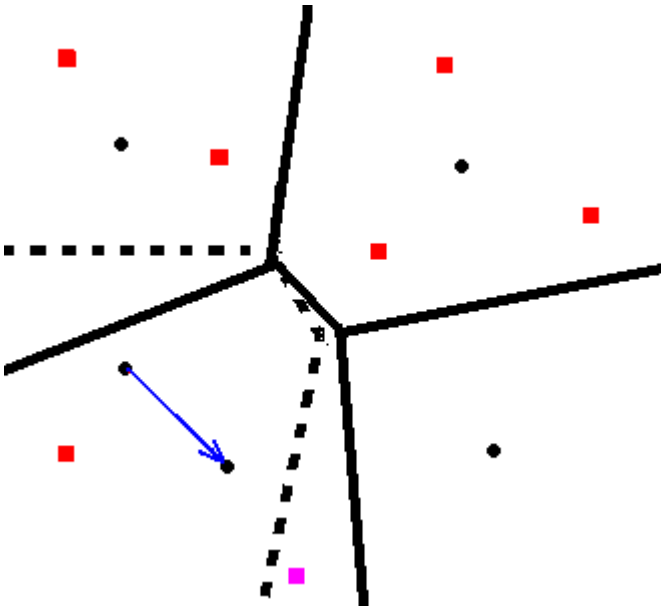
GESTIONE OGGETTI PASSIVI

- Quando i Peers si spostano modificano la topologia delle aree.

- Gli oggetti passivi cambieranno dinamicamente l'area di appartenenza.

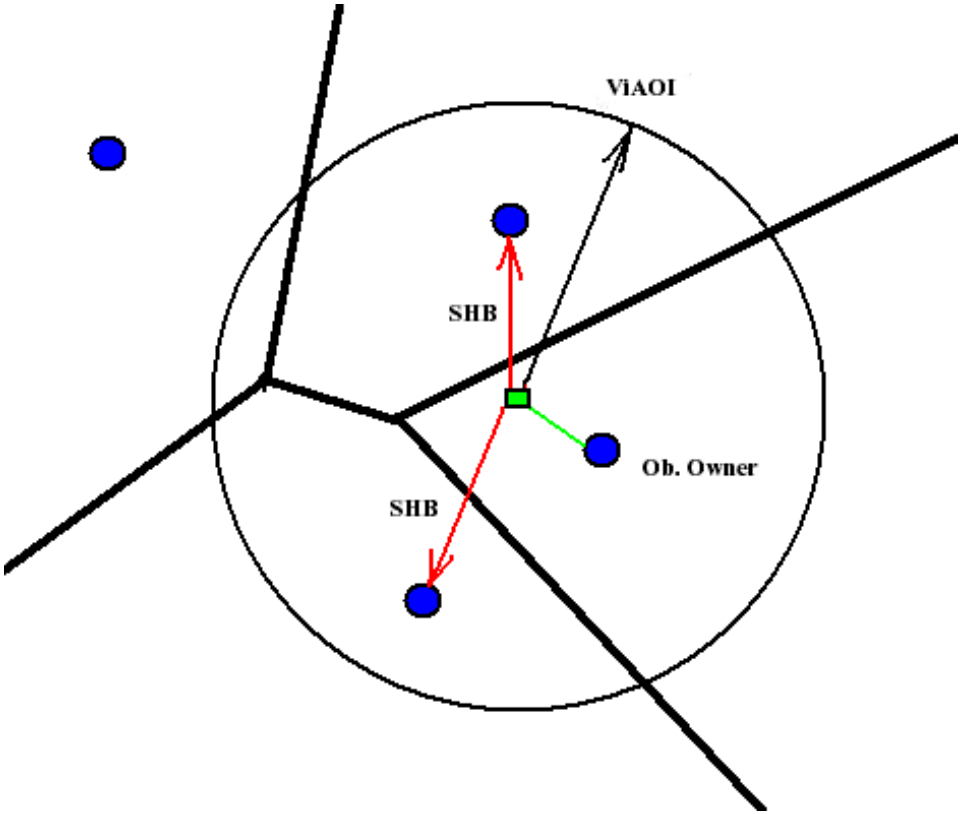
- Possibilità di cambio di ownership multipla in caso di crowding: necessità di utilizzare tecniche opportune per evitare il'rimbalzo' continuo di un oggetto tra peer vicini



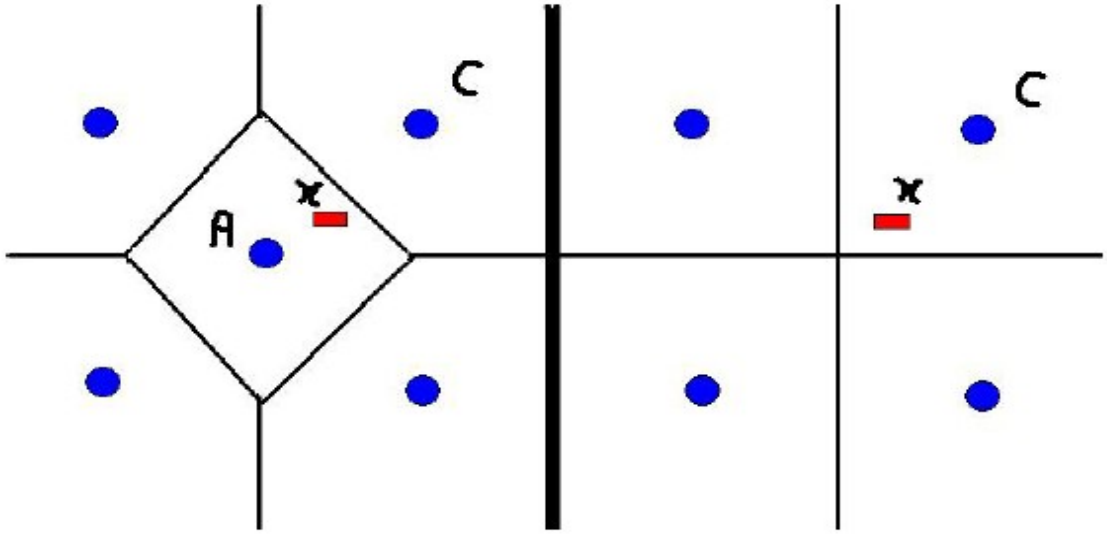


GESTIONE OGGETTI PASSIVI

- Statc Hertbeat = HeartBeat inviato per notificare le modifiche degli oggetti passivi
- SHB per un oggetto ***O***:
  - Sono propagati verso ogni Peers nella area di Visibilità dell'oggetto.
  - Sono inviati dal Peer che gestisce ***O***.
- GLi SHB sono inviati solo quando un oggetto e' creato o modificato.



GESTIONE OGGETTI PASSIVI: RIDONDANZA



- Sfruttando le proprietà dei grafo di Voronoi è possibile gestire il crash dei peer aumentando il grado di ridondanza.

- Un peer P calcolando il grafo di Voronoi eliminando la propria posizione
- P capisce così quali peer ingloberebbero I propri oggetti passivi, nel caso di un suo crash
- P spedisce copie ridondanti dell'oggetto a questi peer, anche se essi non si trovano entro l'area di visibilita' dell'oggetto stesso.