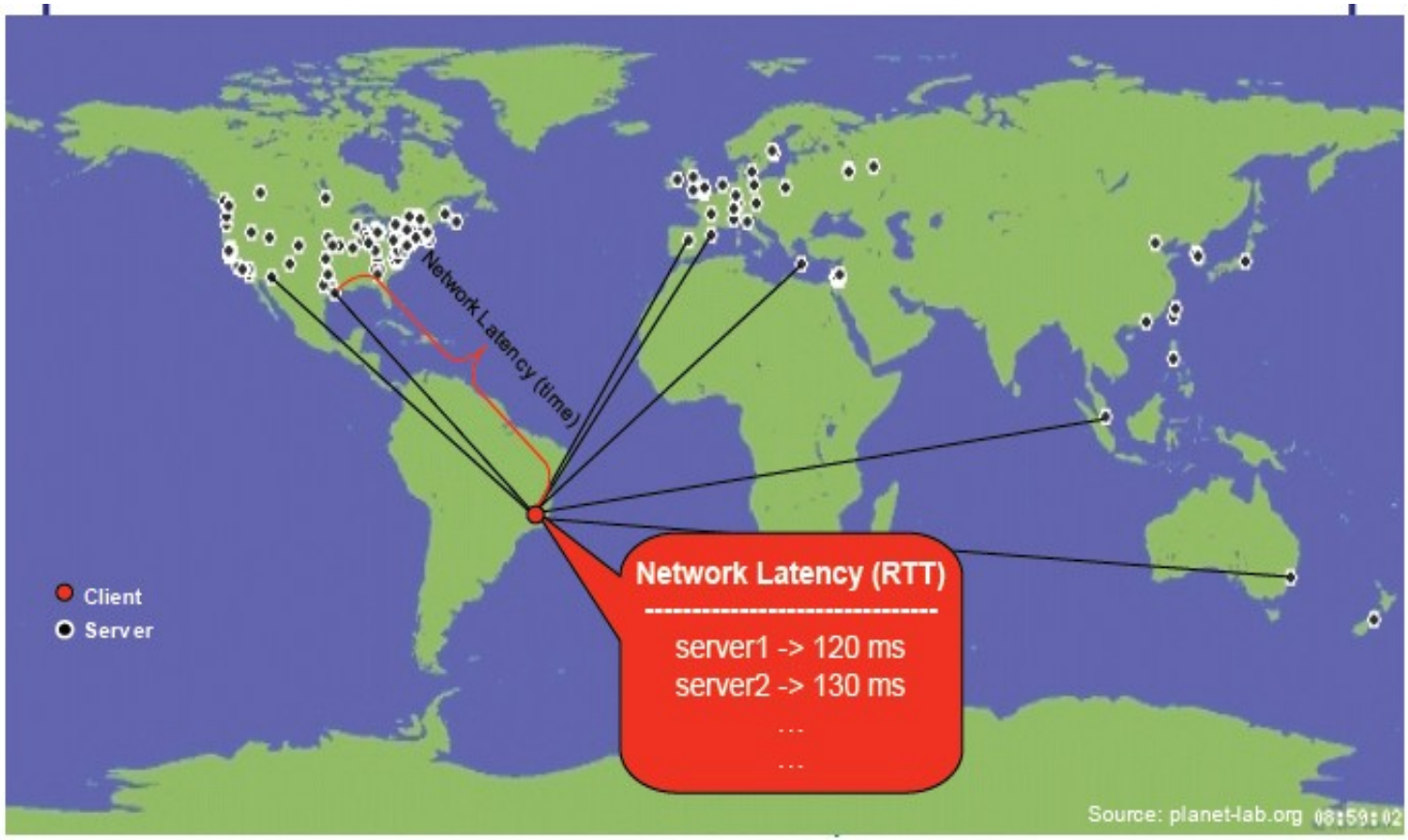
Lezione n.17
19/05/2009
INTERNET COORDINATES
Materiale Didattico
Bulford capitolo 10
Laura Ricci

PROXIMITY AWARE OVERLAYS

- In molte applicazioni distribuite è possibile scegliere dinamicamente un insieme di hosts con cui connettersi per ricevere servizi/risorse
- Problema: individuare in modo efficiente servizi/risorse
- Esempi:
  - client server
    - scelta del server a cui si intende connettersi per ricevere un servizio
    - individuare il server 'migliore' per partecipare ad un gioco multiplayer
  - peer to peer:
    - costruzione di proximity-aware overlays
    - DHT: mapping intelligente dell'overlay logico sulla rete fisica
    - esempio: scelta vicini nell'overlay CAN

APPROCCIO CLASSICO

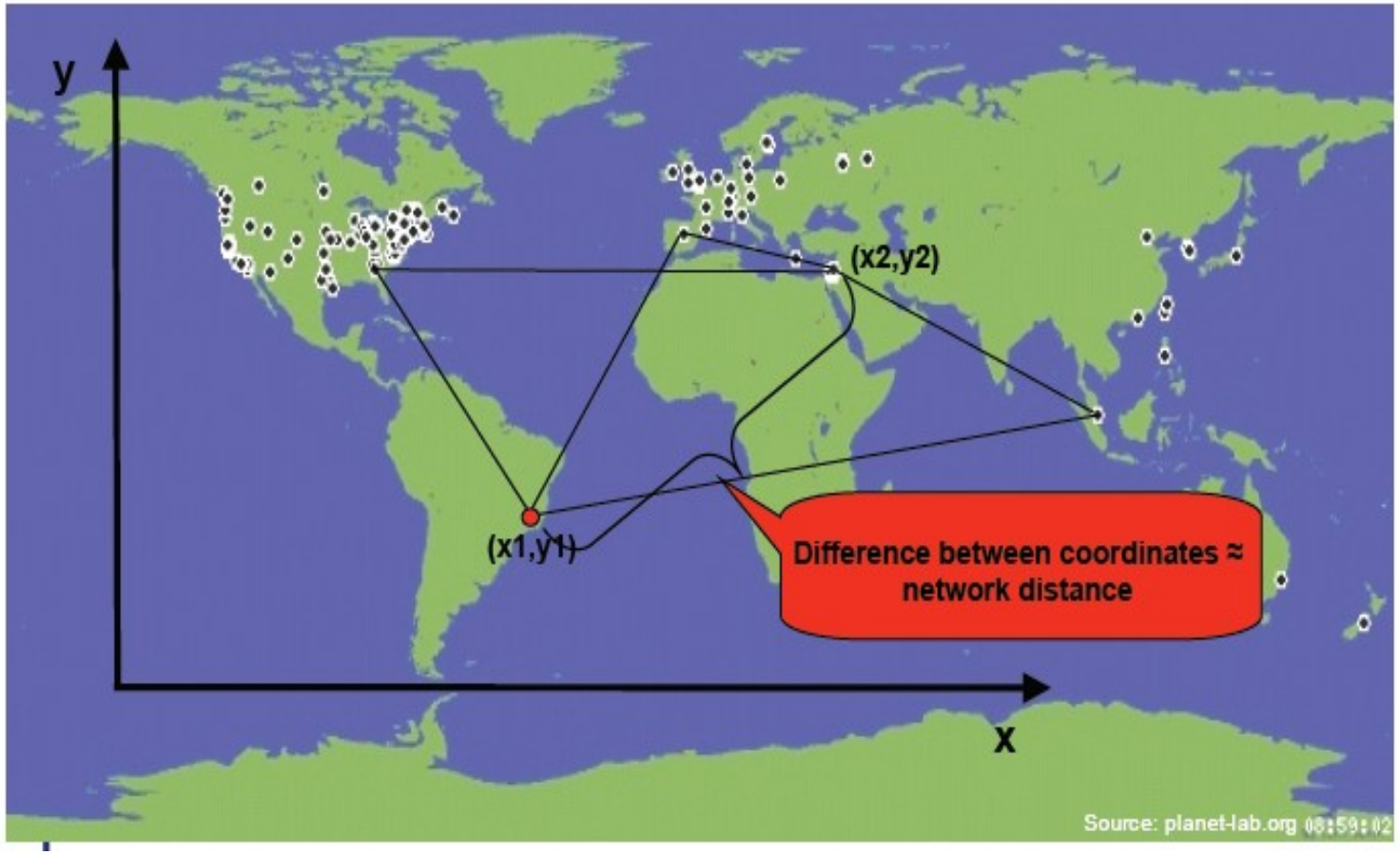


- Quale è il server più vicino ad un client in Brasile?
- Network Distance = Network Round Trip Time (RTT)
- Il client deve effettuare n misure per scegliere il server

INTERNET COORDINATES

- Internet Coordinates:
  - assegnare ai nodi della rete coordinate in uno spazio Euclideo
  - la distanza euclidea tra i nodi approssima il delay tra di loro
- Vantaggio di questo approccio:
  - effettuare un insieme limitato di misurazioni
  - assegnare le coordinate ai nodi sulla base di queste misurazioni
  - evitare il calcolo dei RTT tra tutte le coppie di nodi della rete
- Richiede la definizione di un embedding di uno spazio metrico finito in uno spazio Euclideo
- Network Embedding
  - Lipschitz Embedding
  - Numerical Optimization Embedding

INTERNET COORDIANTES



INTERNET COORDINATES: MOTIVAZIONE

- La scelta dei servers/peers si basa spesso sulla valutazione delle latenze che caratterizzano le comunicazioni con gli hosts
- La latenza viene calcolata mediante il RTT (Round Trip Time)
  - misura il tempo impiegato da un pacchetto di dimensione trascurabile per viaggiare da un nodo della rete ad un altro e tornare indietro.
- Calcolo del RTT:
  - on demand: l'applicazione 'interroga' ogni host 'candidato' inviando separatamente un pacchetto (ping)
    - svantaggi: metodo costoso e poco scalabile
  - internet coordinates:
    - embedding dei nodi e dei loro RTT in uno spazio virtuale VS
    - ad ogni host vengono associate le sue coordinate in VS
    - la distanza euclidea tra due nodi nello spazio VS rappresenta una stima del RTT tra i due nodi

INTERNET COORDINATES: VANTAGGI

- Embedding in uno spazio virtuale. Vantaggi:
  - le coordinate di un nodo n possono essere comunicate ad un insieme S di altri nodi della rete. Ogni nodo in S può
    - calcolare la propria distanza da n (network distance) utilizzando le coordinate di n senza inviare a questo nodo messaggi espliciti di ping
    - non sono richieste comunicazioni aggiuntive
    - in un'architettura P2P
      - un host calcola le proprie coordinate al momento della entrata nella rete
      - quando un host 'scopre' altri hosts, riceve le loro coordinate e può calcolare la distanza dagli host mediante le loro coordinate
- In un overlay P2P
  - la scelta dei vicini nell'overlay può essere guidata dalla conoscenza delle loro coordinate (esempio in CAN)
  - le coordinate riassumono una grande quantità di informazioni
    - Es: da k coordinate posso ricavare le distanze tra k^2 hosts

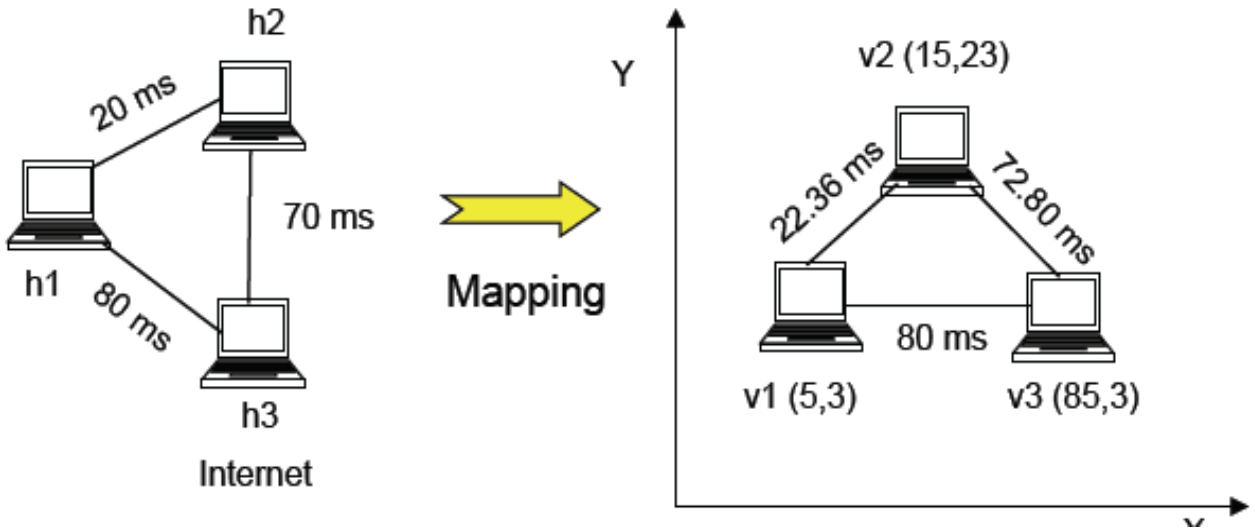
NETWORK EMBEDDING

- Spazio Metrico: è una coppia M=(X,d), in cui X è un insieme finito, e d è una metrica
- Metrica = Funzione che esprime la distanza tra due nodi e gode delle seguenti proprietà
  - Simmetria $d(x,y) = d(y,x)$
  - Posività $d(x,y) \geq 0$, $d(x,y) = 0$ sse $x=y$
  - Diseguaglianza Triangolare $d(x,y) + d(x,z) \geq d(y,z)$
- Un embedding di uno spazio metrico M=(X,d) in uno spazio vettoriale (R^k,δ), (dove δ è la funzione di distanza nello spazio vettoriale) è una funzione

$$\phi : X \rightarrow R^k$$

- (X,d) ed (R^k,δ) sono detti isometrici sse $\delta(\phi(x),\phi(y)) = d(x,y)$
- In generale risulta molto difficile definire una isometria. Infatti, in generale, l'embedding 'non conserva' esattamente le distanze

NETWORK EMBEDDING



- Problema:trovare un mapping α: $H \rightarrow R^k$ tale che $d(h_i,h_j) \cong D(v_i,v_j)$
- H è lo spazio originale (grafo che rappresenta Internet)
- V è lo spazio vettoriale target di dimensione k
- Ad esempio $H=\{h_1,h_2,h_3\}$ $V=\{v_1,v_2,v_3\}$ $d(h_1,h_2) \cong D(v_1,v_2)$

NUMERICAL OPTIMIZATION EMBEDDING

- la funzione che definisce l'embedding viene trattata come la funzione obiettivo di un problema di minimizzazione
- in generale si cerca di minimizzare la differenza tra la distanza definita nello spazio metrico e la distanza definita nello spazio vettoriale
- minimizzazione dello stress dell'embedding, definito come

$$\frac{\sum_{x,y \in X} (\delta(\phi(x), \phi(y)) - d(x,y))^2}{\sum_{x,y \in X} d(x,y)^2}.$$

- risoluzione di un problema di ottimizzazione non lineare in cui le variabili sono le N*K coordinate degli N punti nello spazio vettoriale R^k

LIPSCHITZ EMBEDDING

- Dato lo spazio metrico (X,d), si definisce un insieme D di sottoinsiemi di X, $D=\{L_1,L_2,....L_n\}$
- Si modifica la funzione di distanza d considerando la distanza di un nodo dall'insieme $L \subset X$ come segue

$$d(x,L)=\min_{y \in L} d(x,y)$$

- Il Lipschitz Embedding di X rispetto a D è la funzione ϕ tale che

$$\phi(x) = [d(x,L_1),d(x,L_2),....,d(x,L_n)]$$

- Un Lipschitz Embedding definisce quindi
  - uno spazio in cui ogni asse corrisponde ad un sottoinsieme dei nodi
  - le coordinate attribuite ad x nello spazio vettoriale corrispondono alla distanza minima di x dall'elemento più vicino di L_i
- Se L_i contiene un solo elemento, questo elemento viene detto landmark. In questo caso la componente i-esima del vettore $\phi(x)$ rappresenta semplicemente la distanza dall'i-esimo landmark

LIPSCHITZ EMBEDDING

- Intuizione: in uno spazio metrico, due punti vicini a,b hanno distanza confrontabile rispetto ad un terzo punto x
- Questo è motivato dalla disuguglianza triangolare

$$abs(d(a,x) - d(b,x)) \leq d(a,b)$$

- Se la distanza tra a e b è piccola, allora le distanze tra a, rispettivamente b, ed x sono confrontabili

INTERNET COORDINATES: APPROCCI PROPOSTI

Classificazionedegli approcci presentati in letteratura. I diversi approcci possono essere classificati rispetto a:

- Tipo di embedding: Numerical Optimization, Lipschitz.
- Utilizzo di landmarks vs. algoritmi completamente distribuito
  - Landmark based: viene definito un insieme di punti di riferimento (landmarks). Questi punti di riferimento possono essere utilizzati per definire un Lipschitz Embedding, oppure come base per l'ottimizzazione numerica
    - GNP Global Network Positioning
    - PIC (Practical Network Coordinates)
    - Virtual Landmarks
  - Completamente Distribuiti: modellano la rete ed i nodi mediante un sistema fisico e ne simulano il comportamento
    - Vivaldi
    - Big Bang Simulations .....

GLOBAL NETWORK POSITIONING

Questo approccio prevede due diverse fasi

Fase n.1

- Selezionare un piccolo sottoinsieme di host che costituiscono dei punti di riferimento (L= Landmarks). Questi hosts vengono utilizzati dagli altri per 'orientarsi' nella rete
  - Landmarks
  - Lighthouses
  - Beacons

- Calcolare il round trip time tra i diversi landmarks
  - vengono utilizzati messaggi ICMP di ping tra i landmarks
  - ogni landmarks misura il proprio RTT rispetto agli altri
  - uno dei Landmark
    - raccoglie le misurazioni effettuate dagli altri landmarks e costruisce una matrice delle distanze (simmetrica). Ogni entrata contiene un valore medio del RTT ottenuto da diverse misurazioni.
    - calcola le coordinate degli altri landmarks e le distribuisce

GLOBAL NETWORK POSITIONING

- Il problema della determinazione delle coordinate dei landmarks all'interno dello spazio virtuale equivale ad un problema di minimizzazione dell'errore

  (multidimensional global minimization problem)

$$\text{Min } (\text{err}(d_{ij}, D_{ij}))$$

  - Risolubile con metodi classici di ottimizzazione (numerical optimization embedding)
  - Infinite soluzioni: poichè è rilevante solo la posizione relativa dei diversi landmarks, poichè le coordinate sono utilizzate per calcolare le distanza tra due nodi, qualsiasi posizione è accettabile.

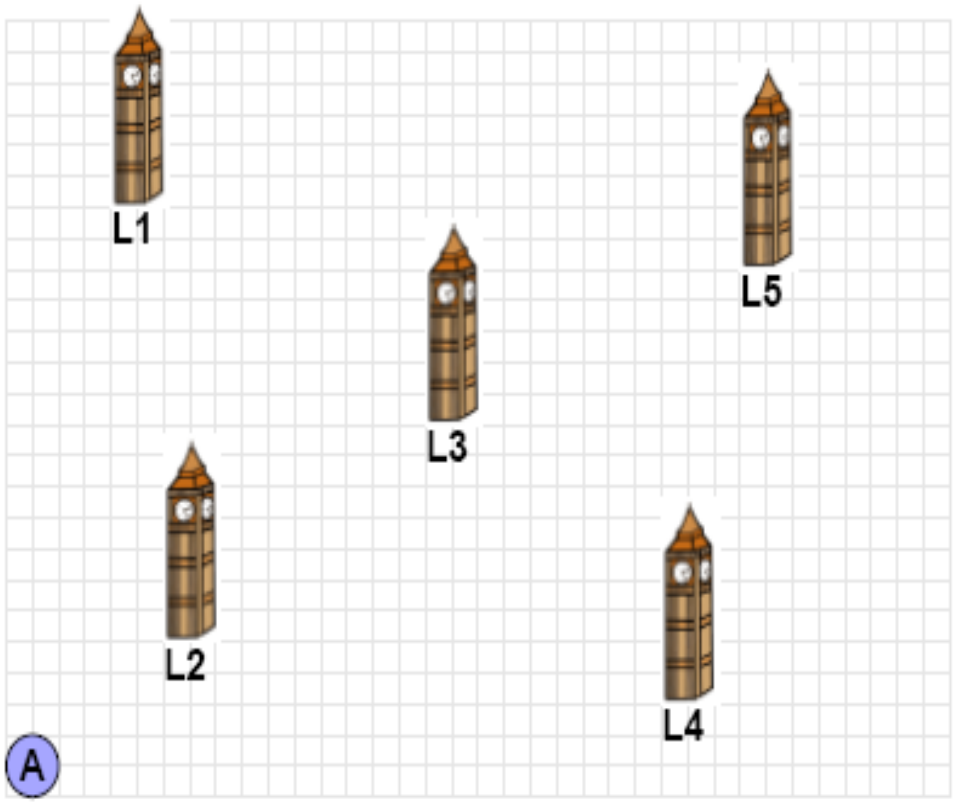
Fase n.2. Inserimento di un nuovo host

- L' host che si unisce alla reta
  - calcola il proprio RTT rispetto ai landmarks
  - calcola le proprie coordinate nello spazio virtuale con la stessa tecnica utilizzata per la risoluzione del problema di ottimizzazione.

GLOBAL NETWORK POSITIONING: FASE 2

Il nodo A invia un ping ad ognuno dei landmarks per determinare la propria posizione

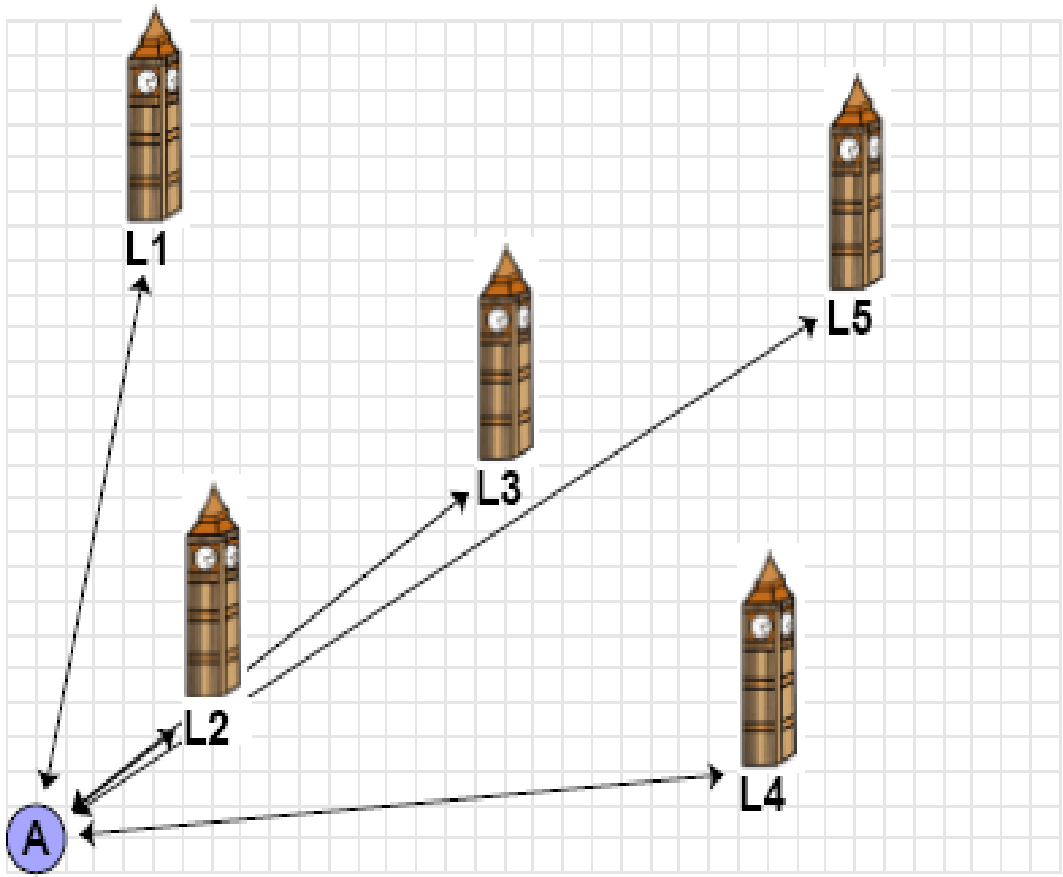
	Coord	Dist
L1	(40,320)	
L2	(60,180)	
L3	(160,250)	
L4	(250,160)	
L5	(280,300)	



GLOBAL NETWORK POSITIONING: FASE 2

il nodo A invia un ping ad ognuno dei landmarks per determinare la propria posizione

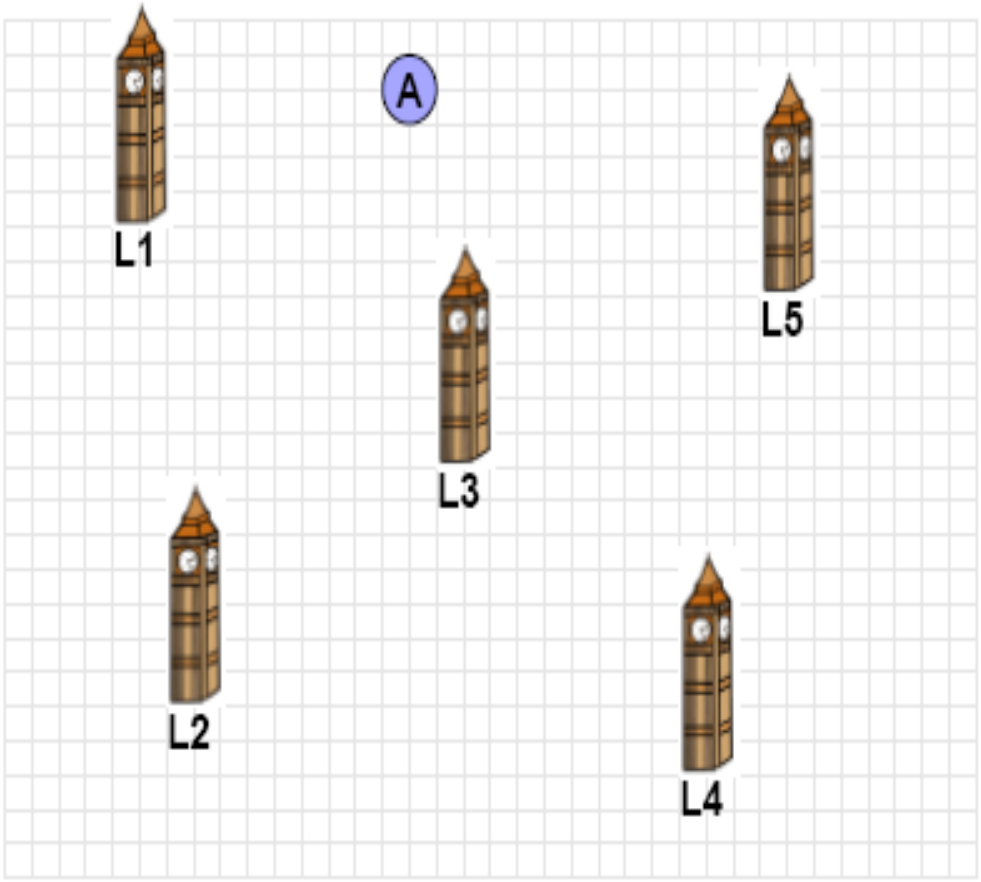
	Coord	Dist
L1	(40,320)	117 ms
L2	(60,180)	201 ms
L3	(160,250)	110 ms
L4	(250,160)	223 ms
L5	(280,300)	143 ms



GLOBAL NETWORK POSITIONING: FASE 2

Il nodo A invia un ping ad ognuno dei landmarks per detrminare la propria posizione

	Coord	Dist
L1	(40,320)	117 ms
L2	(60,180)	201 ms
L3	(160,250)	110 ms
L4	(250,160)	223 ms
L5	(280,300)	143 ms

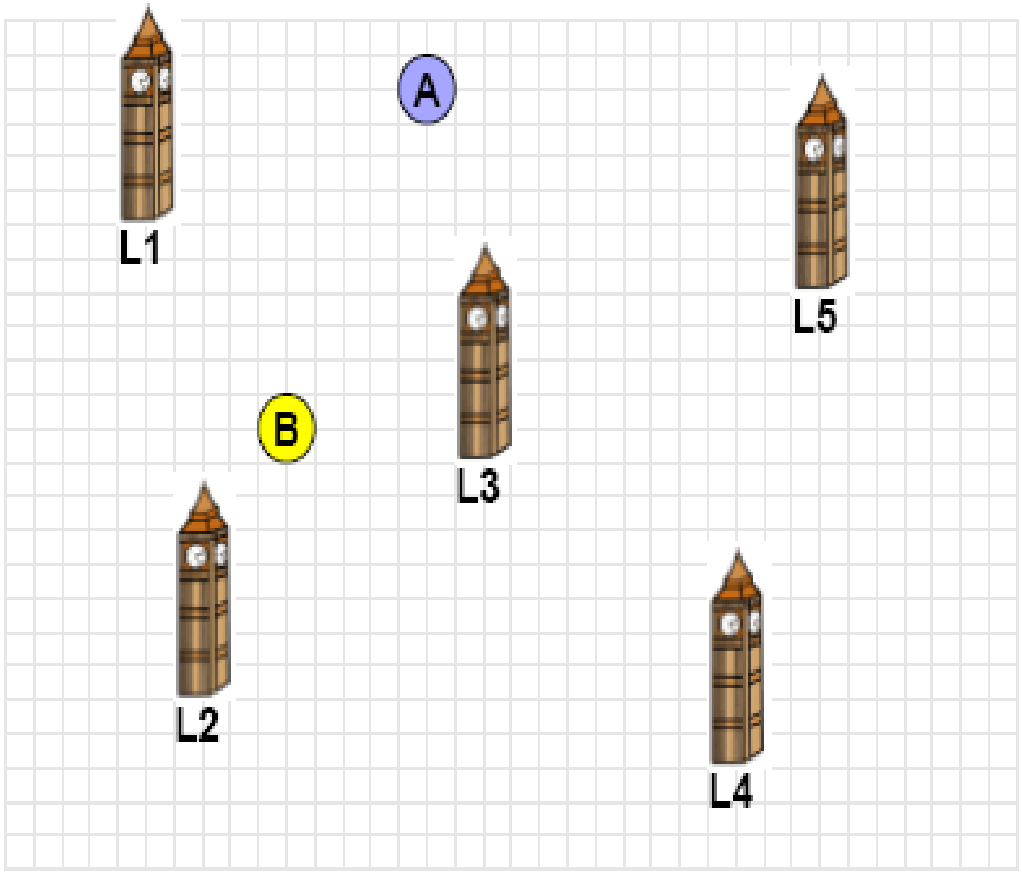


GLOBAL NETWORK POSITIONING: FASE 2

il nodo A invia un ping ad ognuno dei landmarks per determinare la propria posizione

il nodo B segue la stessa procedura: invia un ping ad ognuno dei landmarks per determinare la prorpia posizione

il RTT tra A e B puo' essere determinato calcolando la distanza tra A e B (mediante le loro coordinate) senza effettuare una misurazione esplicita



VIRTUAL LANDMARKS

- Basato sulla definizione di landmarks per il calcolo di Lipschitz embeddings
- Definizione statica di un alto numero di landmarks
- Ogni nodo si attribuisce le coordinate nello spazio vettoriale sulla base della sua distanza dai landmarks (Lipschitz Embedding)
- Poiché si utilizza un alto numero di landmarks, si utilizzano ulteriore tecniche per ridurre la dimensione dello spazio risultante dall'embedding
- Riduzione della dimensione dello spazio generato dall'embedding: utilizzo di tecniche basate su Principal Component Analysis (PCA) e Singular Value Decomposition (SVD)

INTERNET COORDINATES: APPROCCI DISTRIBUITI

Approccio completamente distribuito

- ogni nodo può calcolare le proprie coordinate in modo distribuito, mediante l'analisi dei messaggi ricevuti dagli altri nodi
- non sono richiesti messaggi espliciti per il calcolo del RTT (come nella soluzione on demand)
- Esempio:
  - Vivaldi
  - Big Bang Simulation

VIVALDI: INTRODUZIONE

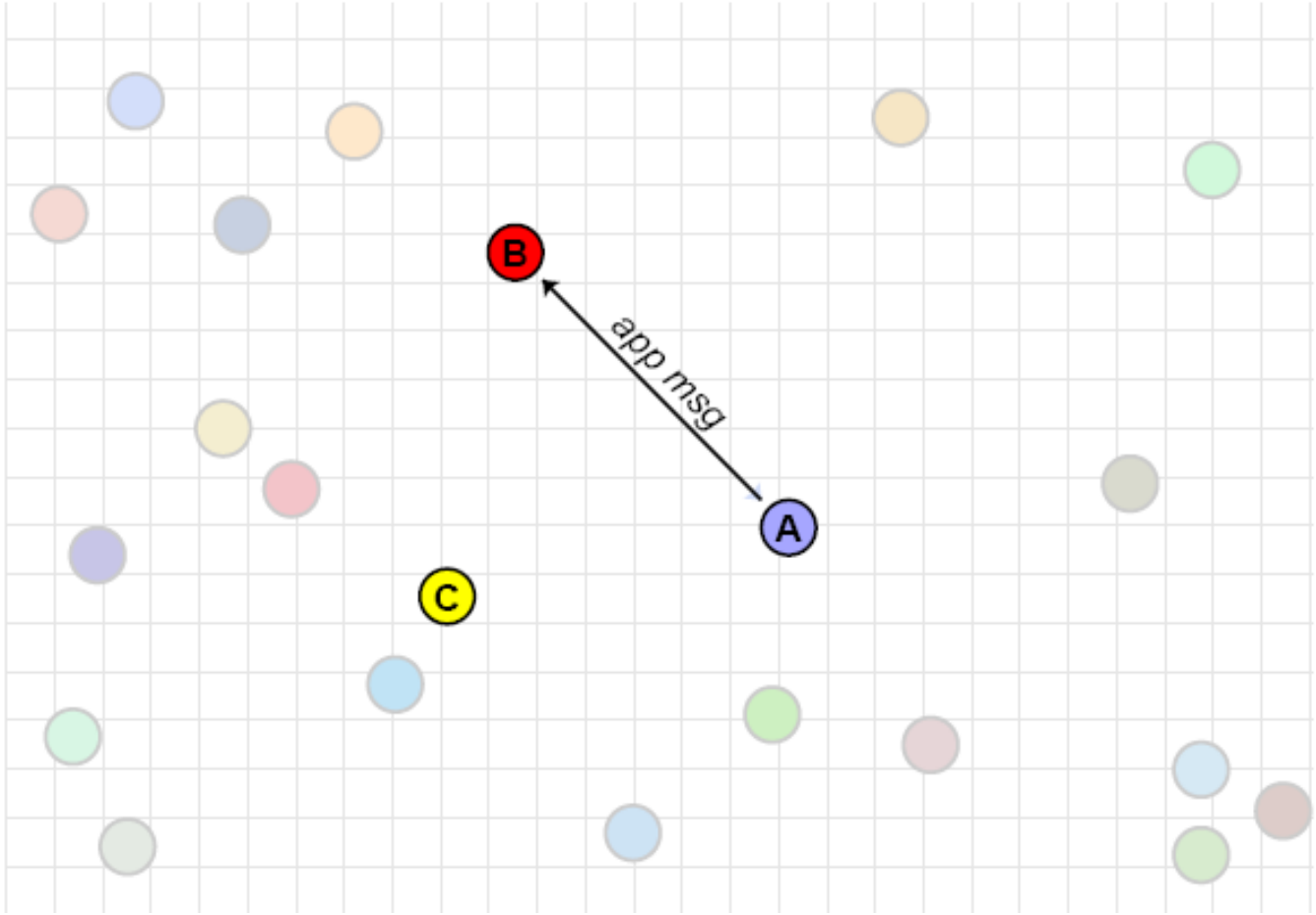
- Vivaldi: Algoritmo completamente distribuito per il calcolo delle coordinate
  - non richiede la definizione di landmarks
- Sfrutta le interazione tra gli host per trasmettere informazioni utili per il calcolo delle coordinate
  - coordinate del mittente
  - RTT del messaggio ricevuto (esempio con Remote Procedure Call)
- Le informazioni ricevute ad ogni interazione consentono di raffinare le proprie coordinate
- Il calcolo delle coordinate utilizza approssimazioni successive
- Definite condizioni necessarie per la convergenza dell'algoritmo

VIVALDI: DISTRIBUTED COORDINATES

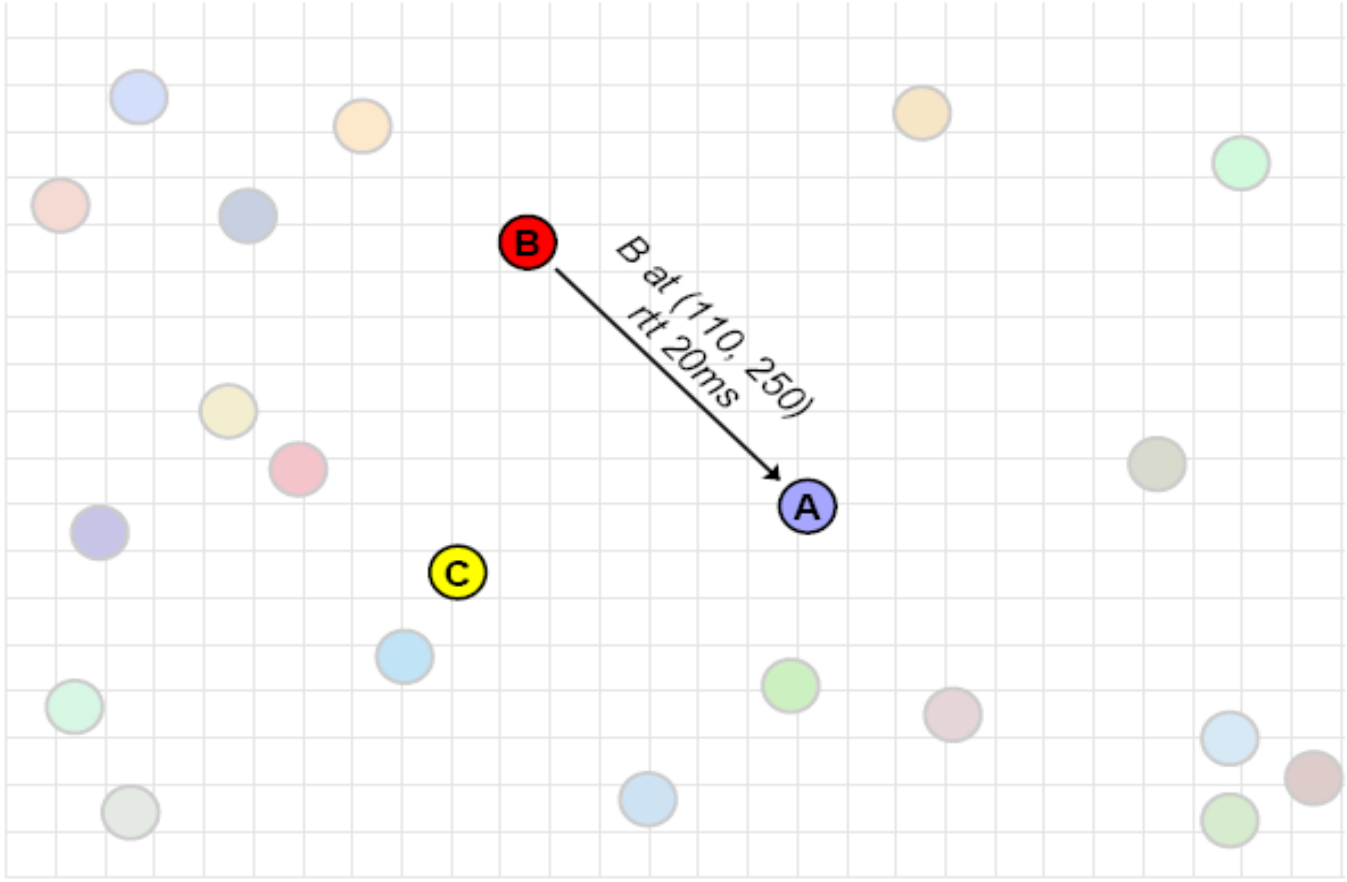
Modello della rete di hosts:

- ogni coppia di nodi connessi da una molla
- il valore del RTT tra i nodi N_i e N_j corrisponde alla lunghezza della molla 'a riposo' (non sottoposta a sollecitazioni)
- la distanza tra i nodi N_i e N_j
  - fornisce, in ogni istante, un'approssimazione del valore del RTT
  - l'approssimazione corrisponde ad una deformazione della molla esistente tra N_i ed N_j
  - la lunghezza della molla deformata (sottoposta a sollecitazione) è proporzionale all'approssimazione
- Rete= sistema di molle sottoposte a sollecitazione
- Errore commesso nell'approssimazione è proporzionale all'energia potenziale del sistema di molle
- Minimizzazione dell'errore: spostamento della posizione di ogni nodo (host) sottoposto alle forze delle molle agenti su di esso (Legge di Hook)

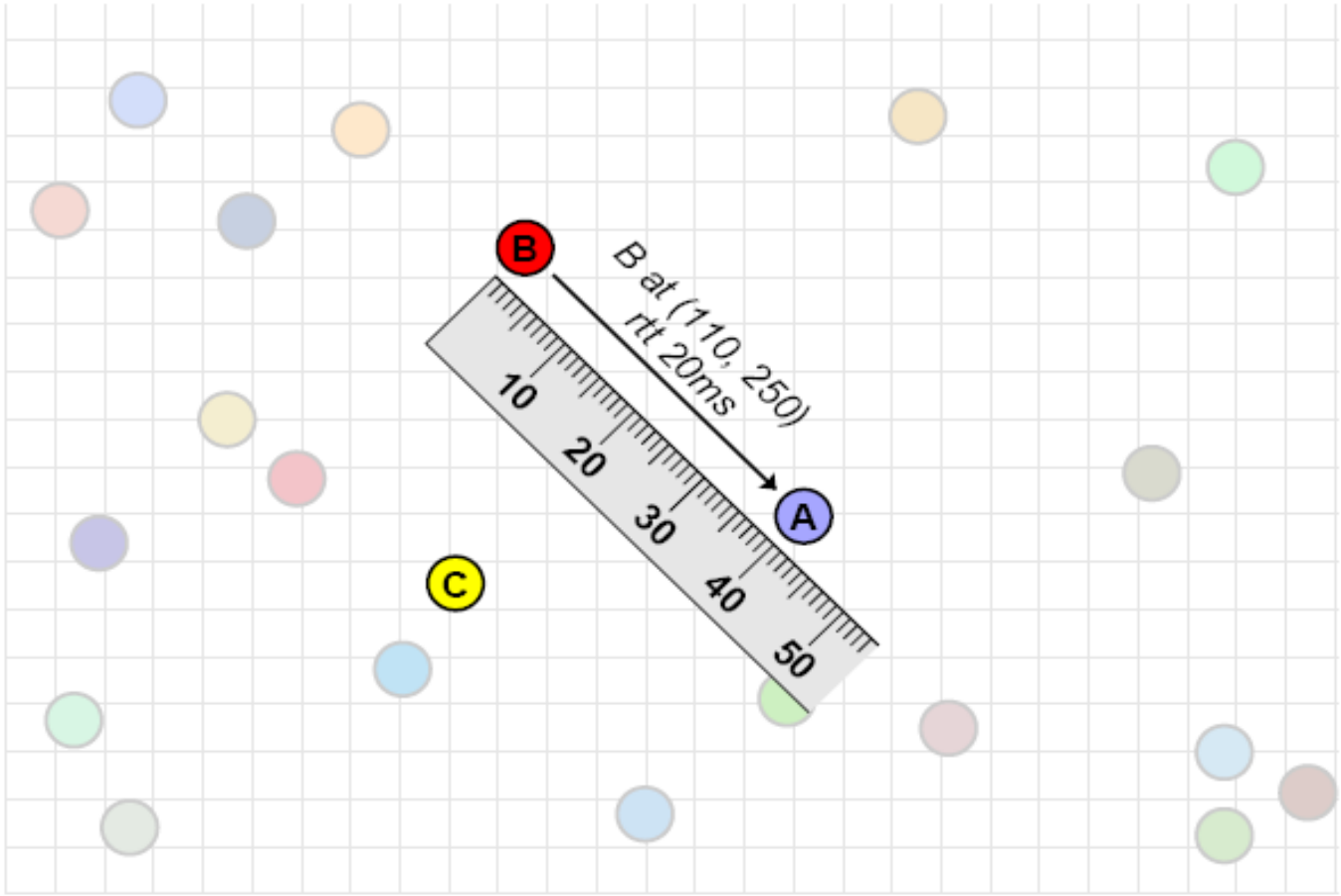
VIVALDI: UN SEMPLICE ESEMPIO



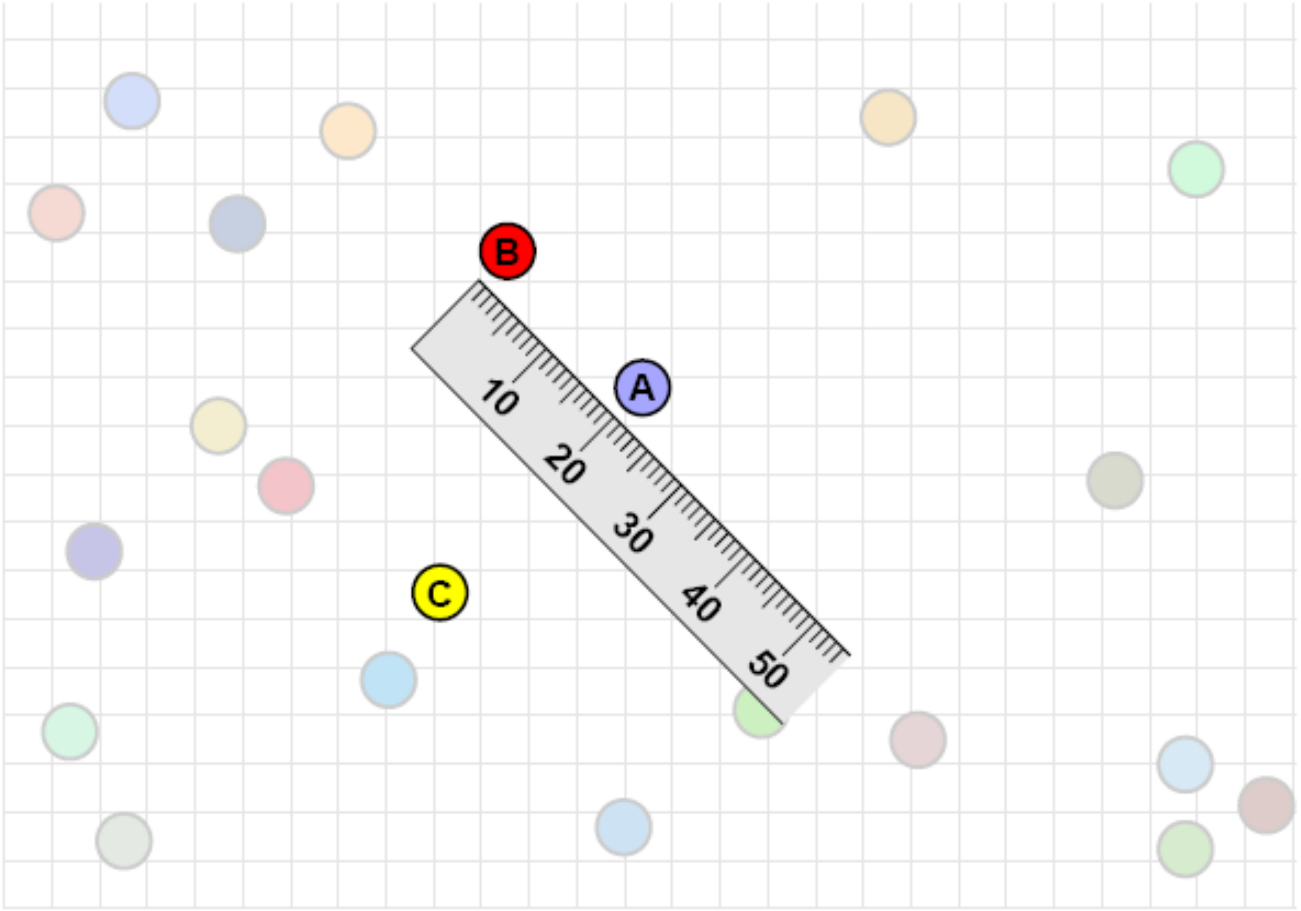
VIVALDI: UN SEMPLICE ESEMPIO



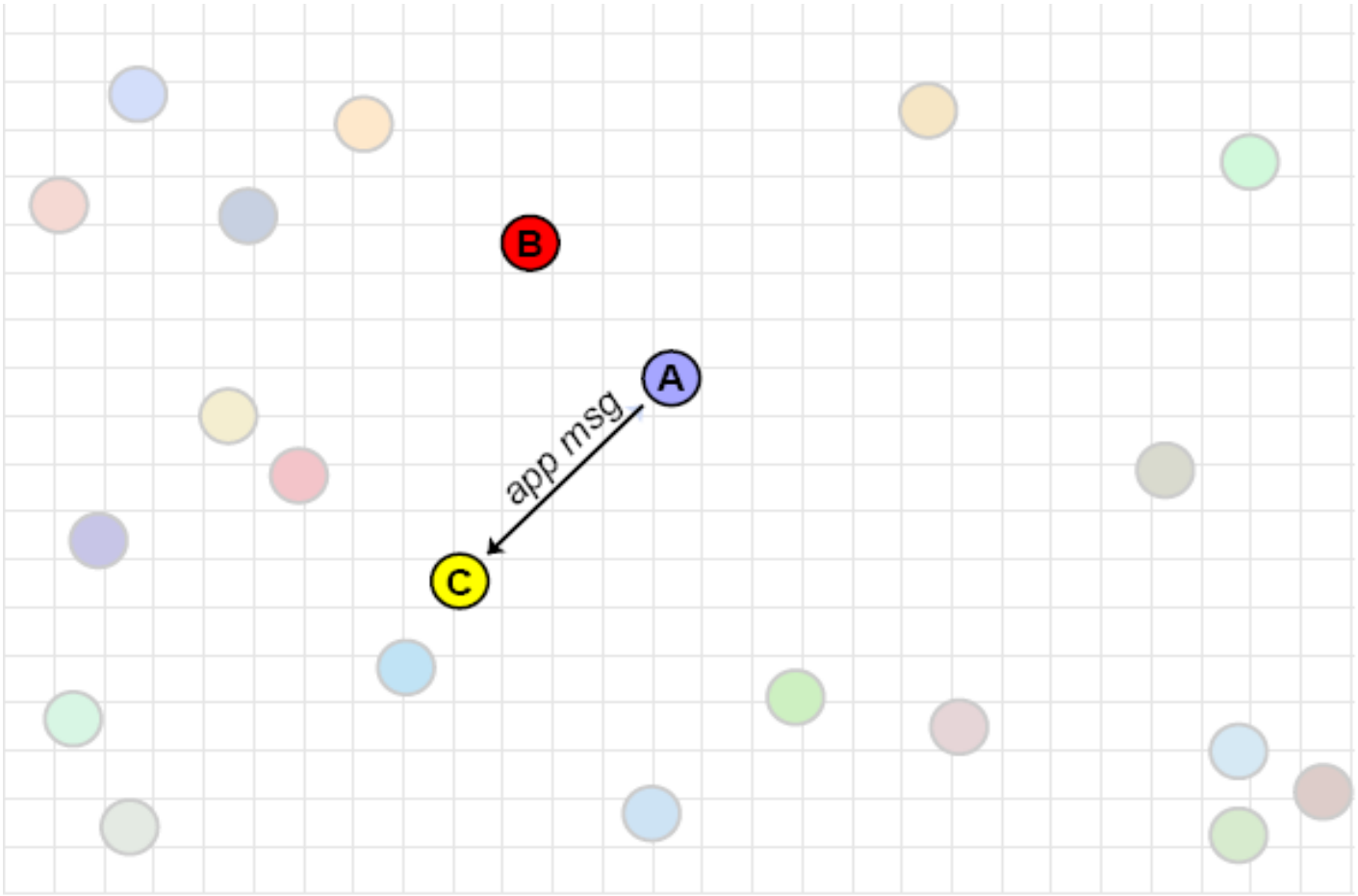
VIVALDI: UN SEMPLICE ESEMPIO



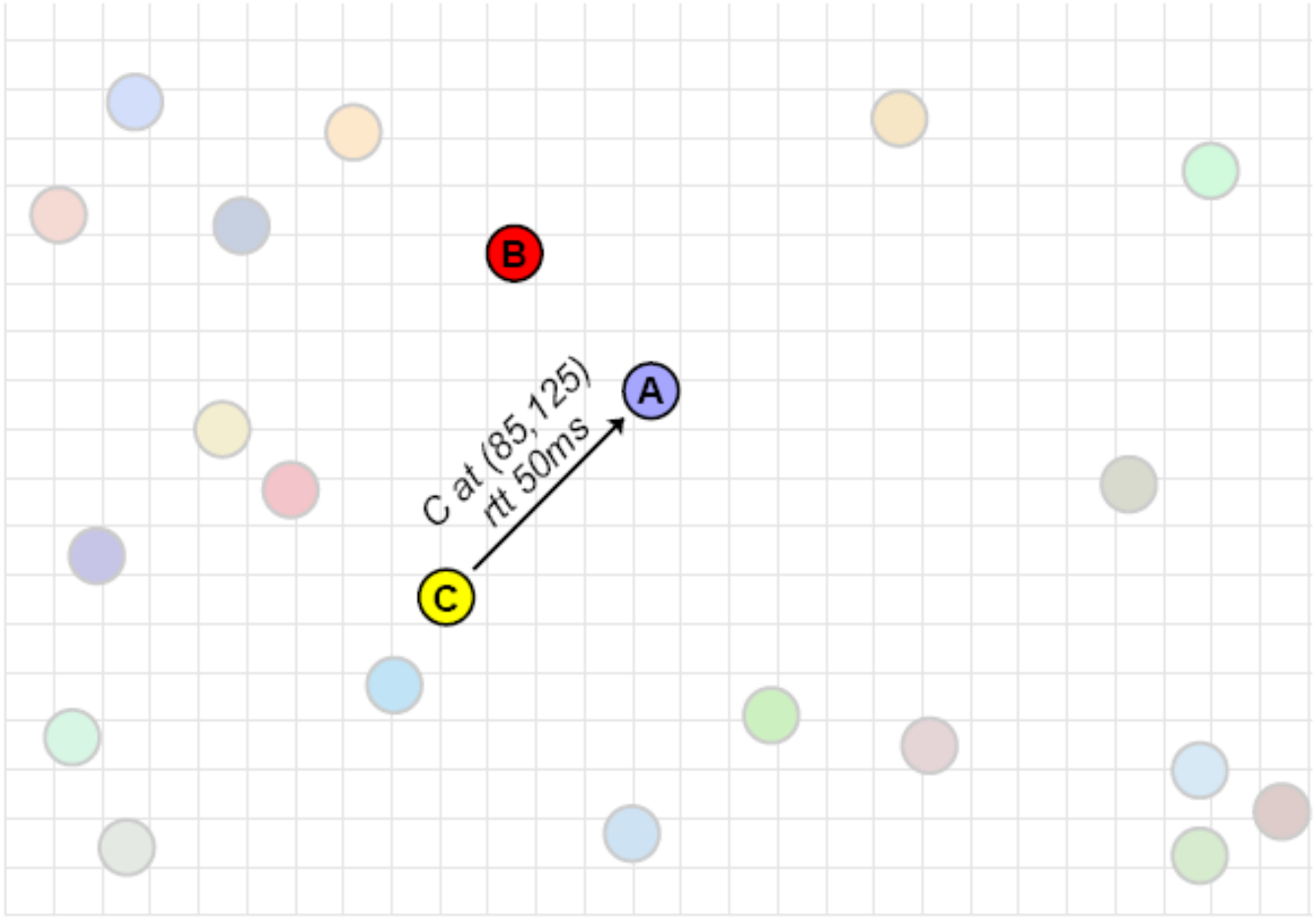
VIVALDI: UN SEMPLICE ESEMPIO



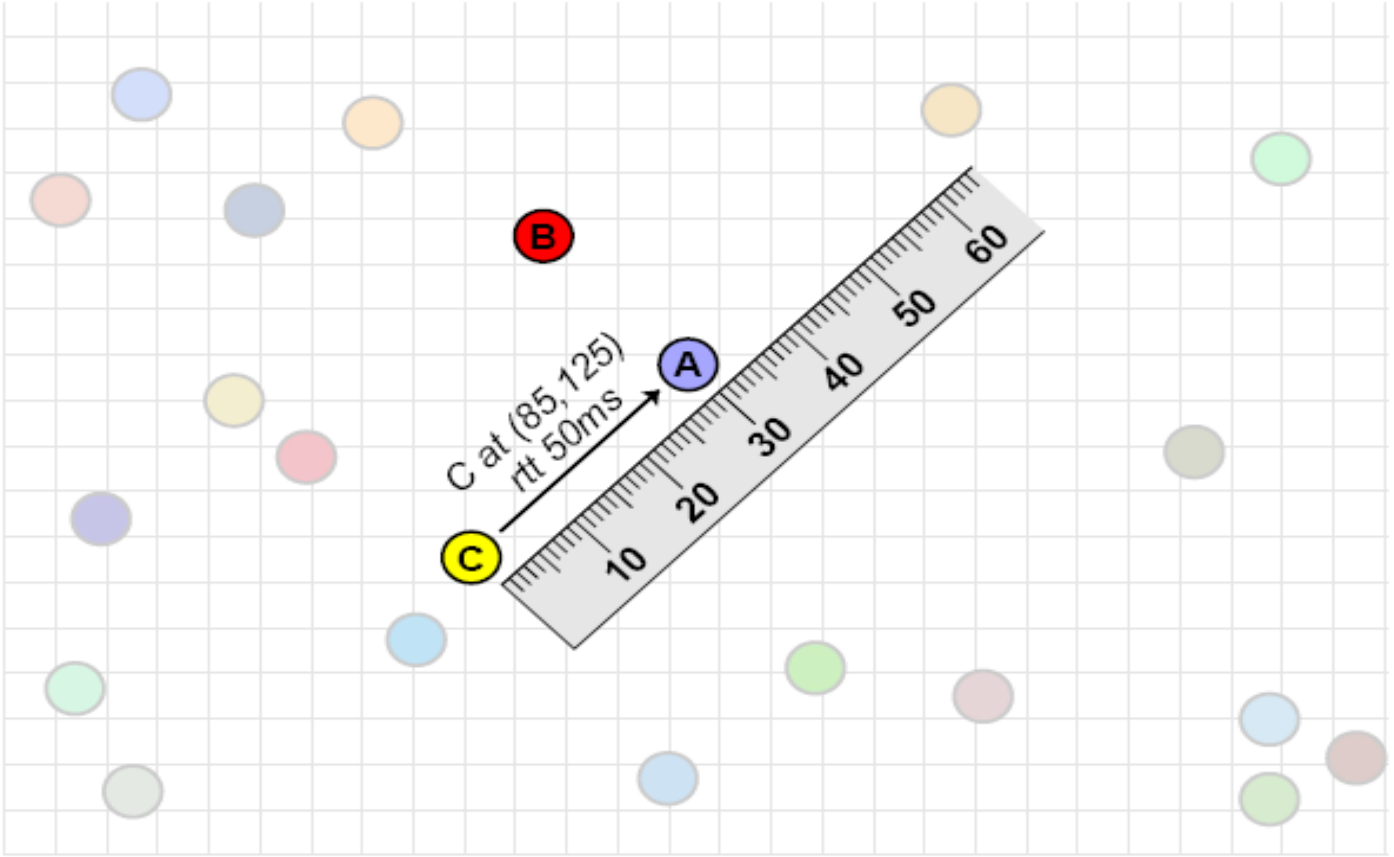
VIVALDI: UN SEMPLICE ESEMPIO



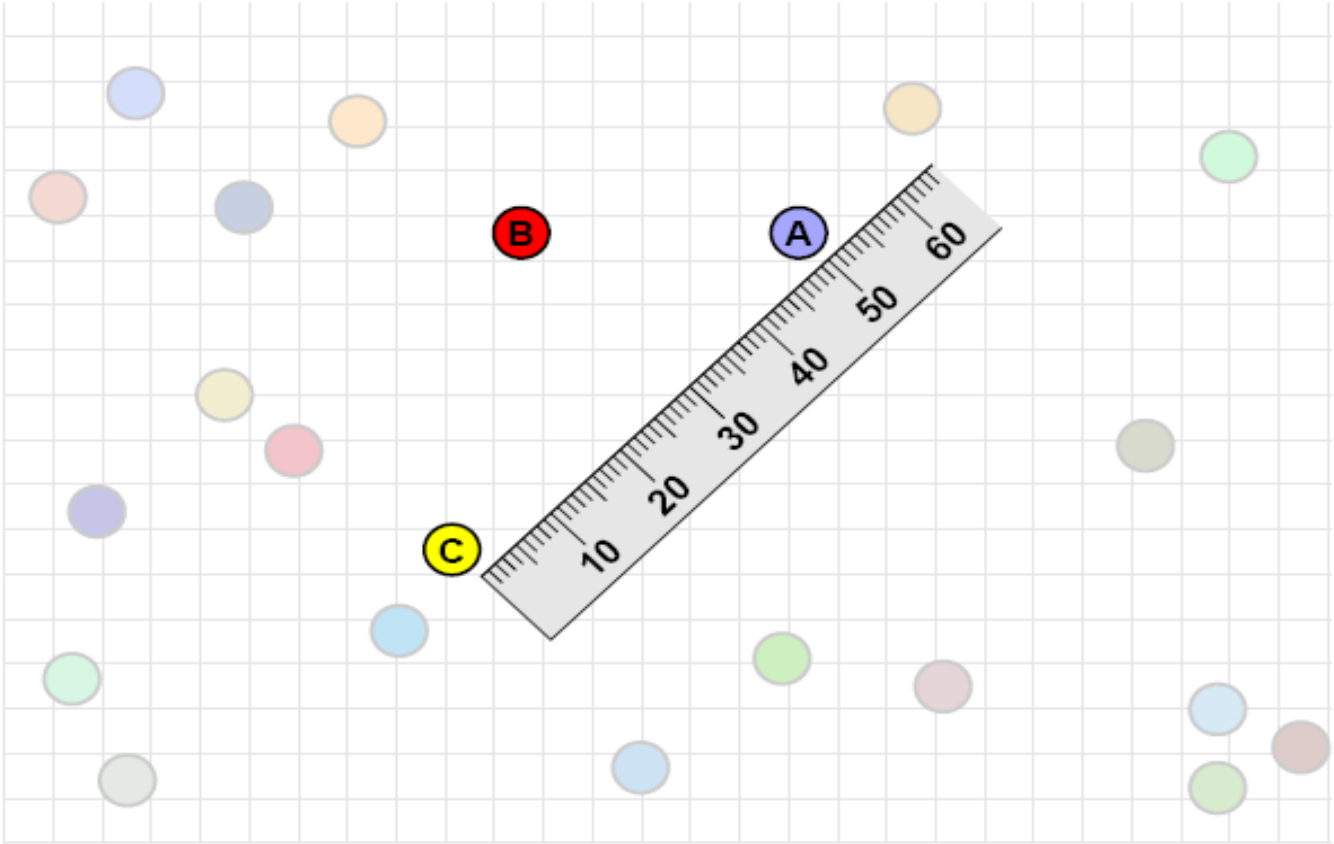
VIVALDI: UN SEMPLICE ESEMPIO



VIVALDI: UN SEMPLICE ESEMPIO



VIVALDI: UN SEMPLICE ESEMPIO



VIVALDI: L'ALGORITMO CENTRALIZZATO

- la rete viene modellata come un insieme di nodi connessi da molle
- i nodi vengono 'mossi' nello spazio virtuale a seconda delle forze esercitate dalle molle su di essi
- errore di predizione nella distanza tra due nodi = differenza tra il RTT tra i nodi e la distanza tra i nodi nello spazio virtuale
- L'algoritmo itera su tutti i nodi. Ogni nodo viene spostato nello spazio virtuale a seconda delle forze esercitate su di esso
- Terminazione: La somma degli errori di predizione è inferiore ad una certa soglia
- Errore di predizione

$$E = \sum_i \sum_j \left(L_{ij} - \|x_i - x_j\|\right)^2$$

  - Lij = RTT tra il nodo i ed il nodo j
  - || xi -xj|| = distanza nello spazio astratto tra i nodi i e j

VIVALDI: L'ALGORITMO CENTRALIZZATO

- Basato sulla legge di Hook (legge che descrive la deformazione di una molla sottoposta ad una sollecitazione)
- u(xi -xj) indica la direzione delle forza esercitata su I
- si suppone che la valutazione delle forze esercitate su ogni nodo sia ripetuta ad intervalli di tempo regolari
- t = lunghezza dell'intervallo di tempo
- maggiore è il valore di t, maggiore è lo spostamento di un nodo in ogni intervallo di tempo
- La valutazione di un valore ottimale di t rappresenta uno dei punti critici dell'algoritmo

VIVALDI : L'ALGORITMO CENTRALIZZATO

```
// Input: latency matrix and initial coordinates
// Output: more accurate coordinates in x
compute_coordinates(L, x)
  while (error (L, x) > tolerance)
      foreach i
          F = 0
          foreach j
              // Compute error/force of this spring. (1)
              e = L_ij − ||x_i − x_j||
              // Add the force vector of this spring to the total force. (2)
              F = F + e × u(x_i − x_j)
          // Move a small step in the direction of the force. (3)
          x_i = x_i + t × F
```

VIVALDI: L'ALGORITMO DISTRIBUITO

$$
\begin{array}{ll}
\multicolumn{2}{l}{\textsc{Vivaldi}(\vec{x_j}, w_j, l_{ij})} \\
1 & w_s = \frac{w_i}{w_i + w_j} \\
2 & \epsilon = \frac{|\|\vec{x_i} - \vec{x_j}\| - l_{ij}|}{l_{ij}} \\
3 & \alpha = c_e \times w_s \\
4 & w_i = (\alpha \times \epsilon) + ((1 - \alpha) \times w_i) \\
5 & \delta = c_c \times w_s \\
6 & \vec{x_i} = \vec{x_i} + \delta \times (\|\vec{x_i} - \vec{x_j}\| - l_{ij}) \times u(\vec{x_i} - \vec{x_j})
\end{array}
$$

Il nodo i riceve un messaggio dal nodo j. I parametri sono i seguenti

- $\vec{x_j}$ coordinate del nodo j
- l_{ij} misura del RTT tra il nodo i ed il nodo j
- w_j stima del nodo j sull'accuratezza delle proprie coordinate

VIVALDI: L'ALGORITMO DISTRIBUITO

La convergenza dell'algoritmo è influenzata dal valore di δ

$$w_s = \frac{w_i}{w_i + w_j}$$

$$\delta = c_c \times w_s$$

$$\vec{x}_i = \vec{x}_i + \delta \times (\|\vec{x}_i - \vec{x}_j\| - l_{ij}) \times u(\vec{x}_i - \vec{x}_j)$$

- ad ogni invocazione della procedura Vivaldi sul nodo i, provocata dalla ricezione di un messaggio dal nodo j, viene calcolata la differenza d tra la distanza di i dal nodo j (calcolata rispetto alle sua coordinate attuali) e il RTT rilevato
- lo spostamento verso la posizione ideale di i corrisponde ad una frazione di d determinata dal valore di δ
- il valore di δ dipende dalla stima dell'accuratezza delle proprie coordinate posseduta, rispettivamente, dal nodo locale e da quello remoto

VIVALDI:EMBEDDING ERRORS

- L'embedding considerato utilizza uno spazio metrico, dove vale la disuguaglianza triangolare ($AB+BC \geq AC$)
- Tuttavia, i meccanismi di routing di Internet non garantiscono che la disuguaglianza triangolare valga nel caso dei RTT.
- Nel caso in cui la disuguaglianza non valga (es: 59 < 62) viene genarto un enbedding error, come mostrato in figura