

# Topologically-Aware Overlay Construction and Server Selection

( IEEE 2002 )

Sylvia Ratnasamy, Mark Handley, Richard Karp  
et Scott Shenker

Présenté par:  
DEKAR Lyes

# Plan de la présentation

- I. Introduction
- II. Modèle proposé par l'article
- III. Application du modèle aux overlay networks
  - o Structuré
  - o Non structuré
- IV. Application du modèle pour la sélection de serveur
- V. Discussion et Critiques

# Problématique

Une application distribuée, peut être vue comme une couverture qui englobe des nœuds qui constituent un réseau logique au dessus du réseau physique, et c'est ce qui s'appelle Overlay Networks (P2P par exemple).

Le routage dans ces réseaux se fait au niveau applicatif, et peut être en totale contradiction avec le routage du niveau IP

Ignore les caractéristiques du réseau physique sous-adjacent

Engendre des temps de latences considérables, et un trafic inutile

# Un exemple



# Problématique

- o Comment Limiter les temps de latences inutiles ?
- o Comment Optimiser le routage au niveau applicatif ?

Réponse :

Ne peut se faire qu'en étant en phase avec le niveau IP et en prenant en considération la topologie physique.

# Problématique

Comment sélectionner le serveur de contenu de la manière plus la efficace possible ?

Et cela dans le but d'améliorer les temps de réponse.

Réponse :

Il faut choisir le serveur qui soit le plus proche du client ( en terme de temps de latence) . (d'autres facteurs existent bien sûr)

# Problématique

Tout cela demande une connaissance de la topologie !!!!  
(de l'overlay network)

Connaître toute la topologie du réseau,

10 nœuds	oui.....
100 nœuds	peut être....
1000 nœuds	ça va être dur.....
10000 nœuds	hai.....

# Problématique

Tout cela demande une connaissance de la topologie !!!!  
(de l'overlay network)

Connaître toute la topologie du réseau,

10 nœuds	oui.....
100 nœuds	peut être....
1000 nœuds	ça va être dure.....
10000 nœuds	hai.....

## Comment faire ?



# Pré-solution

Notre but ne sera donc pas de connaître toute la topologie du réseau mais seulement une partie.

Une connaissance de la topologie (même limitée) peut améliorer les performances de l'applications distribuée.

L'incorporation de la connaissance de la topologie est une **Optimisation de performance** et non une condition pour le bon fonctionnement de l'application.

La preuve : le P2P actuel marche ....



Et donc une connaissance de la topologie même approximative peut suffire !!!!!

Pour notre article cette connaissance consiste à connaître tout les nœuds du réseau logique (Overlay network) qui sont les plus proches ( en termes de latences ).



# Modèle proposé

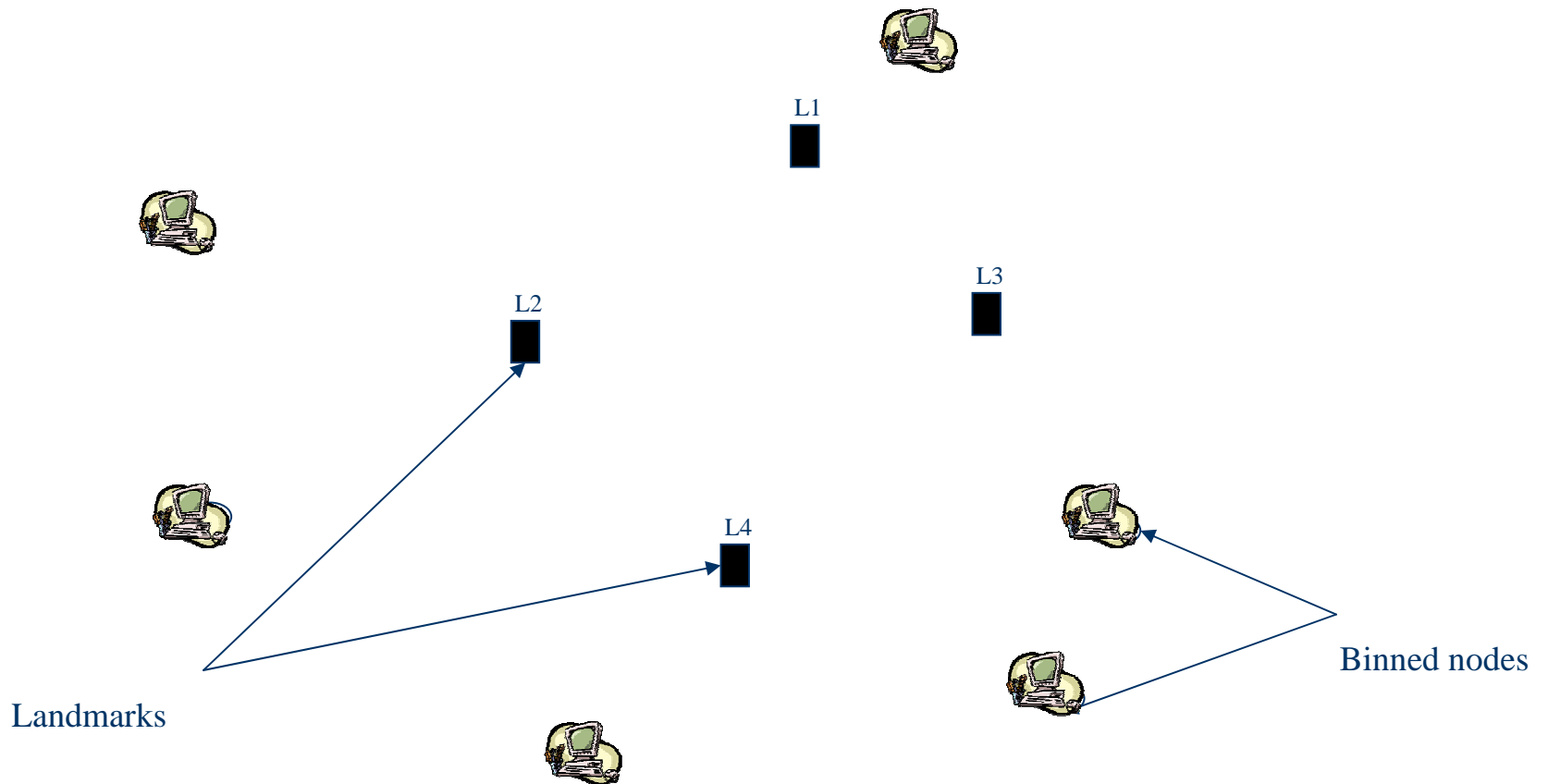
Modèle qui est en fait une « Clusterisation » des noeuds, mais qui a été appelée « Binning » (mise en casier).

Auto partitionnement des nœuds en « Bins », tel que les nœuds qui sont dans le même Bin sont relativement proches les uns des autres en terme de latence.

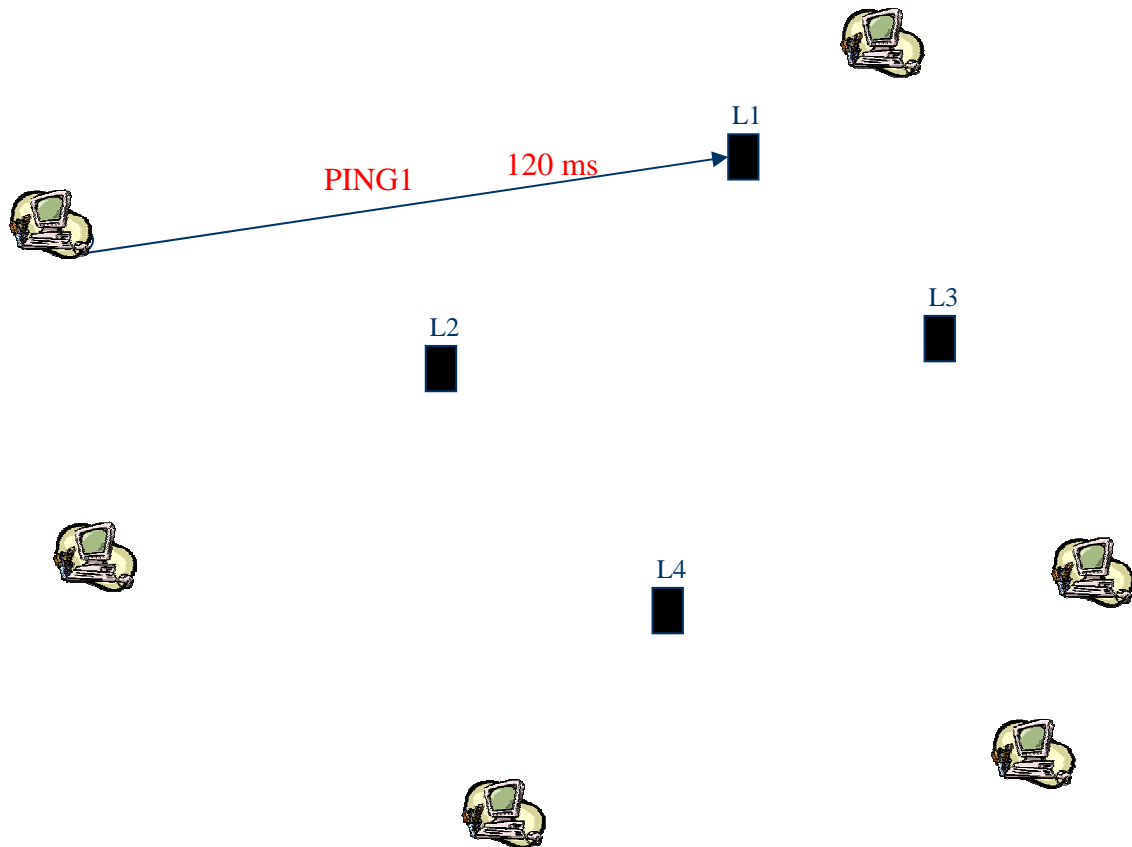
# Principes du modèle proposé

- A. Chaque nœud connaît un ensemble relativement stable de machines, qu'on appellera « LandMarks » (points de repère).
- B. Ces machines sont étalées à travers Internet de manière homogène.
- C. Chaque nœuds du réseau logique calcule son RTT (Round Trip Time) par rapport à chaque landmark.
- D. A partir de ces mesures, chaque nœuds va établir une liste de tout les landmarks par ordre croissant de leurs RTT.
- E. Les nœuds qui ont le même ordre sont supposés être relativement proches les uns des autres et donc font partie du même Bin

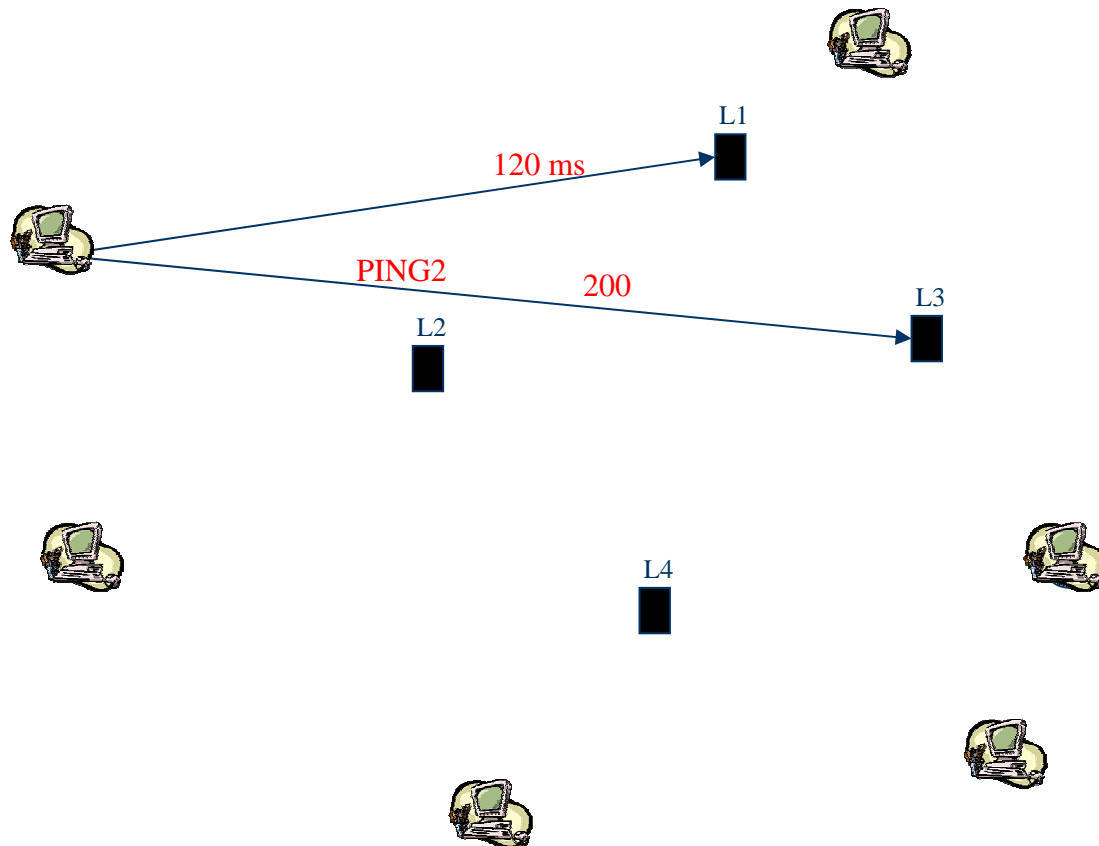
# Modèle proposé



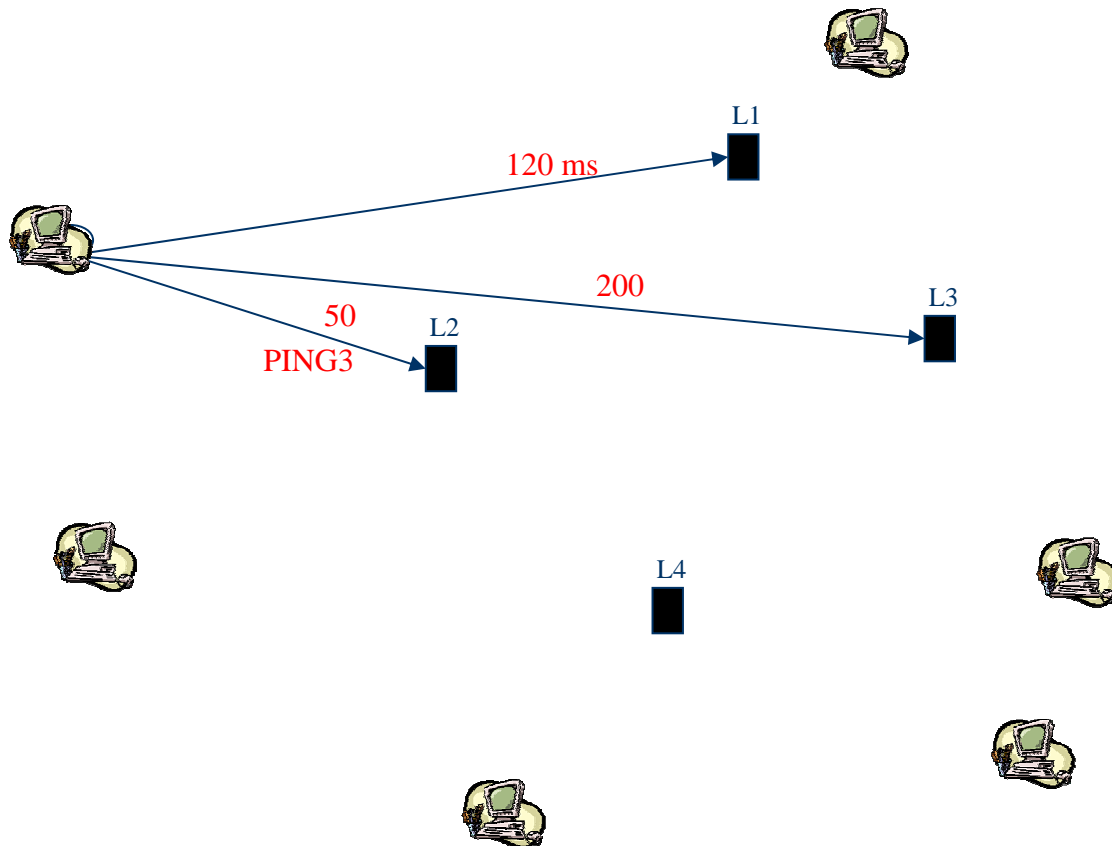
# Modèle proposé



# Modèle proposé

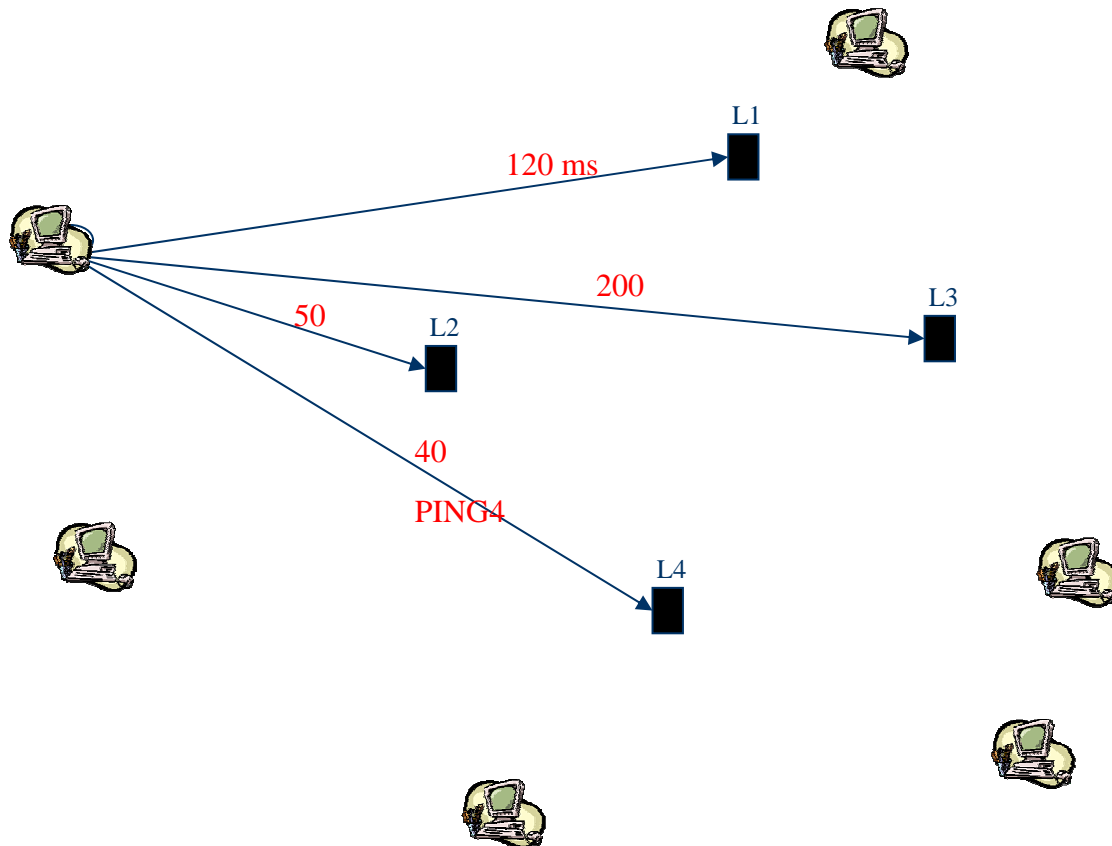


# Modèle proposé

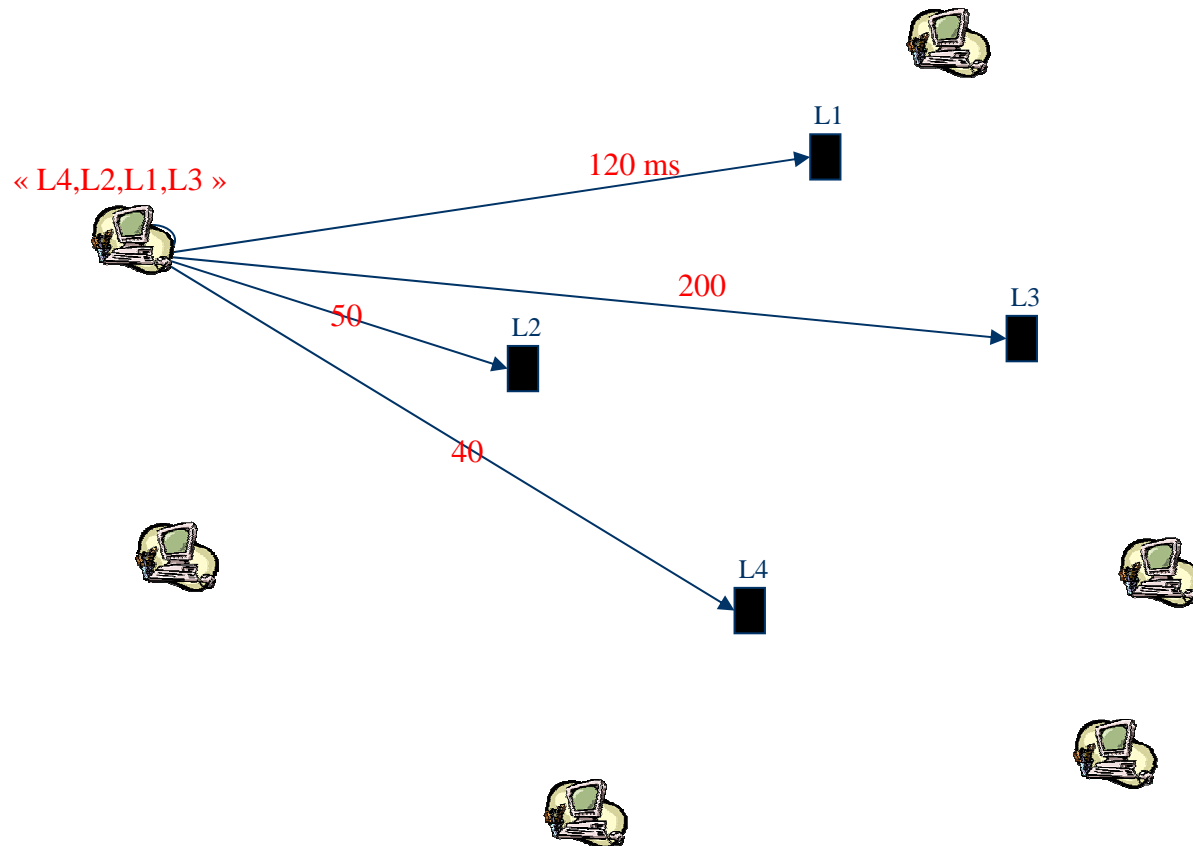




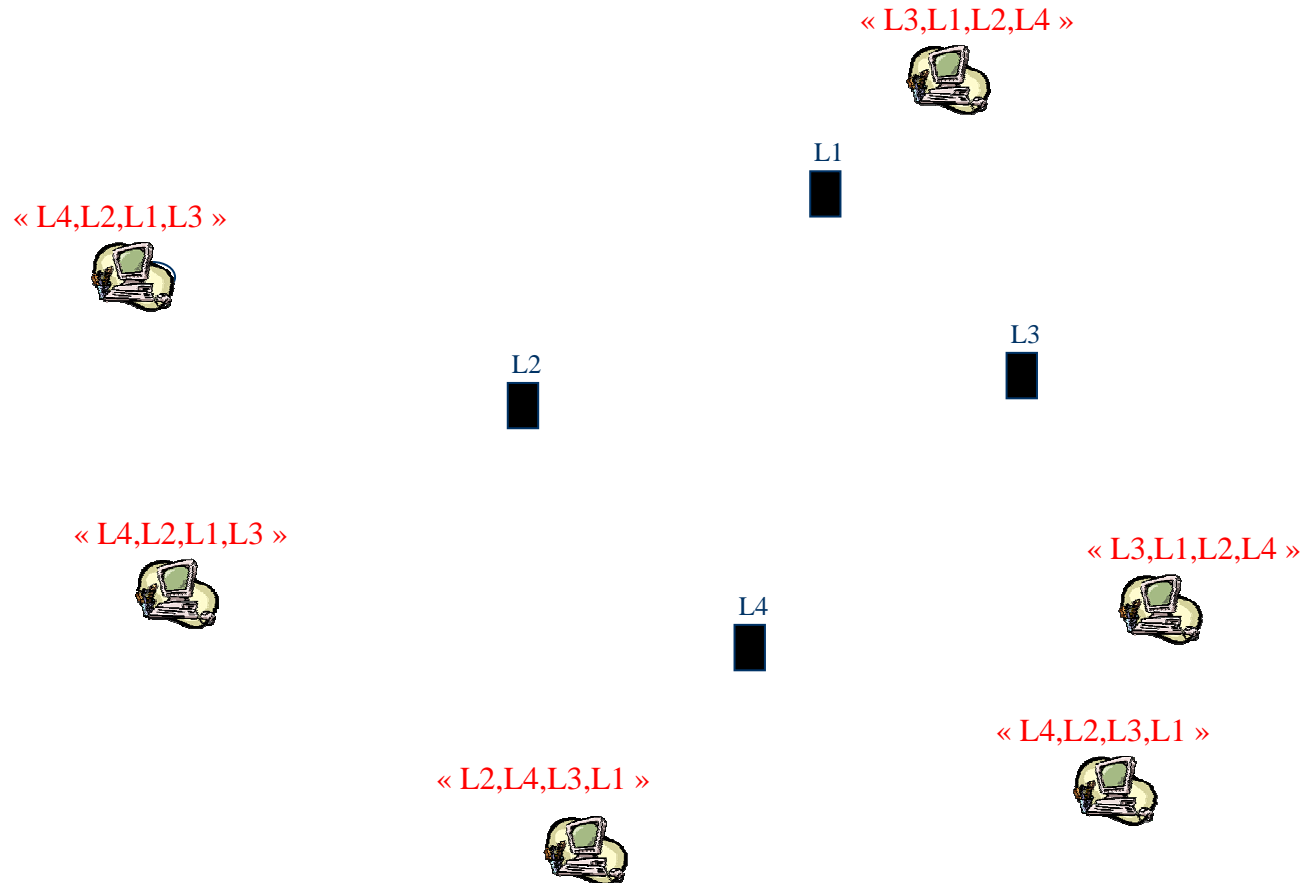
# Modèle proposé



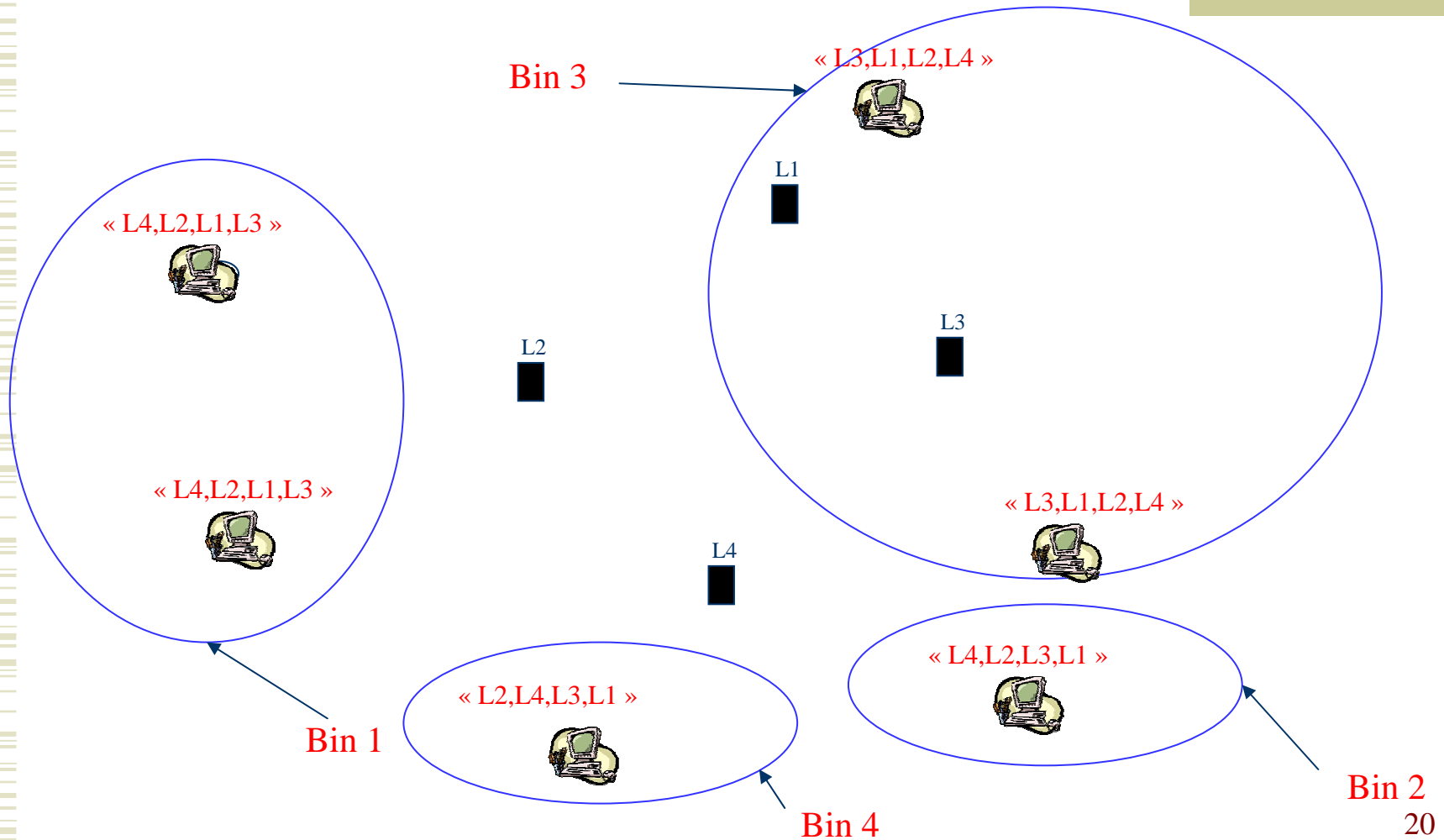
# Modèle proposé



# Modèle proposé



# Modèle proposé



# Remarques sur le modèle

- A. D'après les tests, 8 à 12 landmarks sont largement suffisants pour un bon « binning ». ie : un partitionnement efficace
- B. En cas de panne d'un Landmark, chaque nœud va le supprimer de sa liste et donc on aura une restructuration totale du réseau.

# Remarques sur le modèle

- A. D'après les tests, 8 à 12 landmarks sont largement suffisants pour un bon « binning ». ie : un partitionnement efficace
- B. En cas de panne d'un Landmark, chaque nœud va le supprimer de sa liste et donc on aura une restructuration totale du réseau.

**Important :** Le modèle est « **Scalable** » car chaque nœud découvre son Bin de manière autonome, et cela ne nécessite ni communication ni coordination entre les nœuds.



C'est bien, c'est beau mais cela va servir a quoi ?

# Application du modèle

C'est bien, c'est beau mais cela va servir a quoi ?

La meilleur façon de le découvrir est de le voir en application

Deux applications :

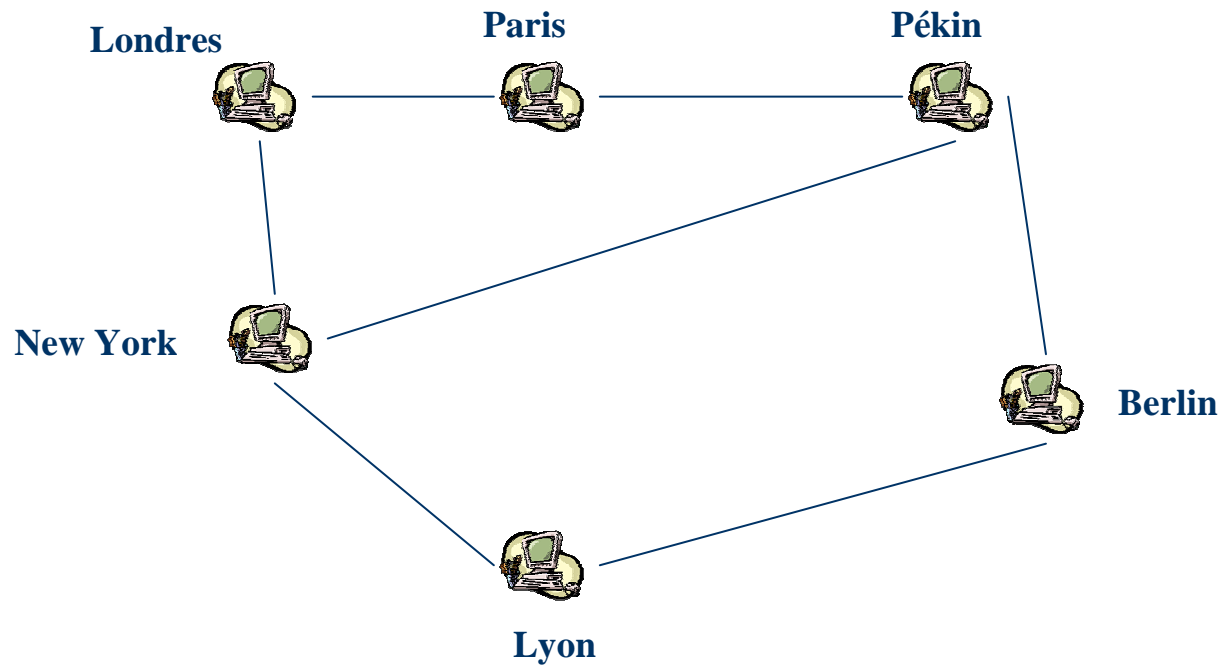
- Les overlays networks
- La sélection de serveur



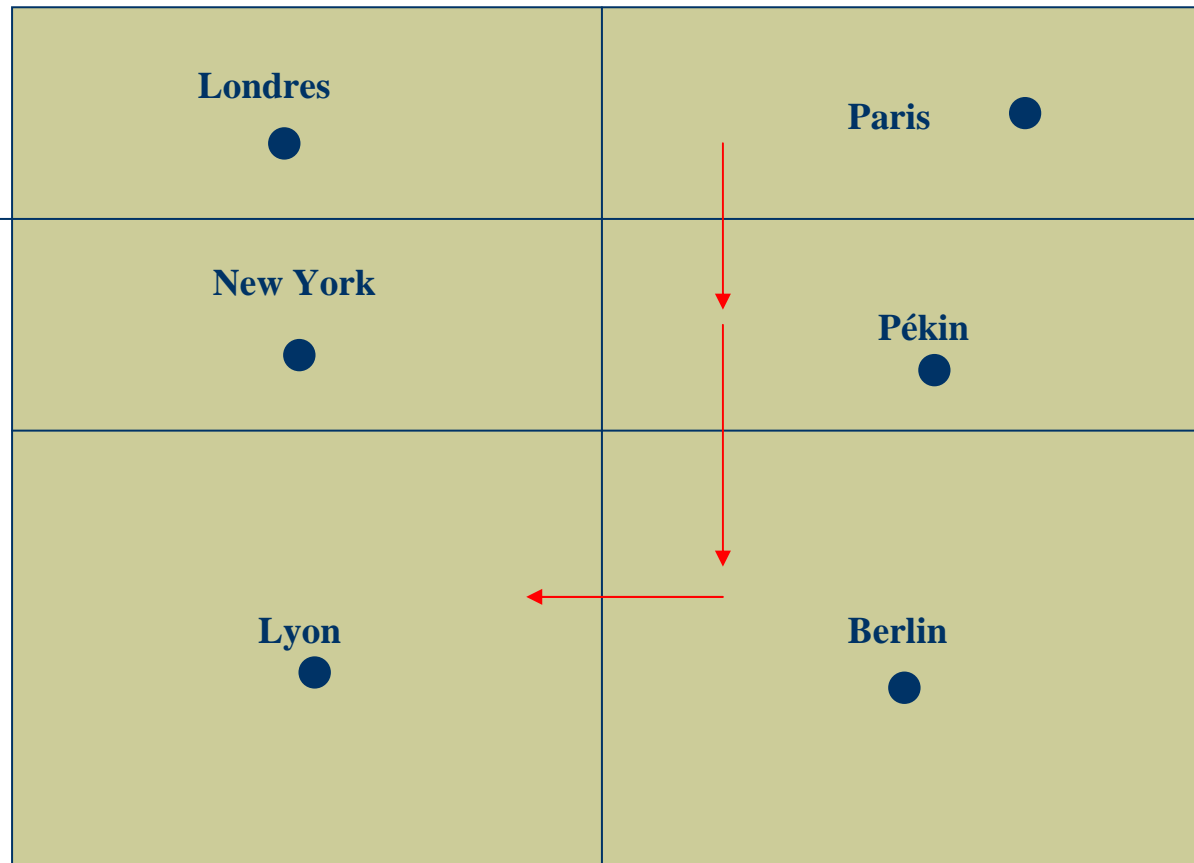
# Les overlays Networks (ex P2P )

- A. Structuré : le réseau des hots et les objets sont structurés de telle manière qu'on peut garantir d'atteindre un objet en un certain nombre de hops.  
Exemple : CAN, Chord, Pastry, Tapestry...
  
- B. Non structuré : ni connaissance ni structure, pas de garantie.  
Exemple : Gnutelle, FreeNet...

# Un réseau structuré classique

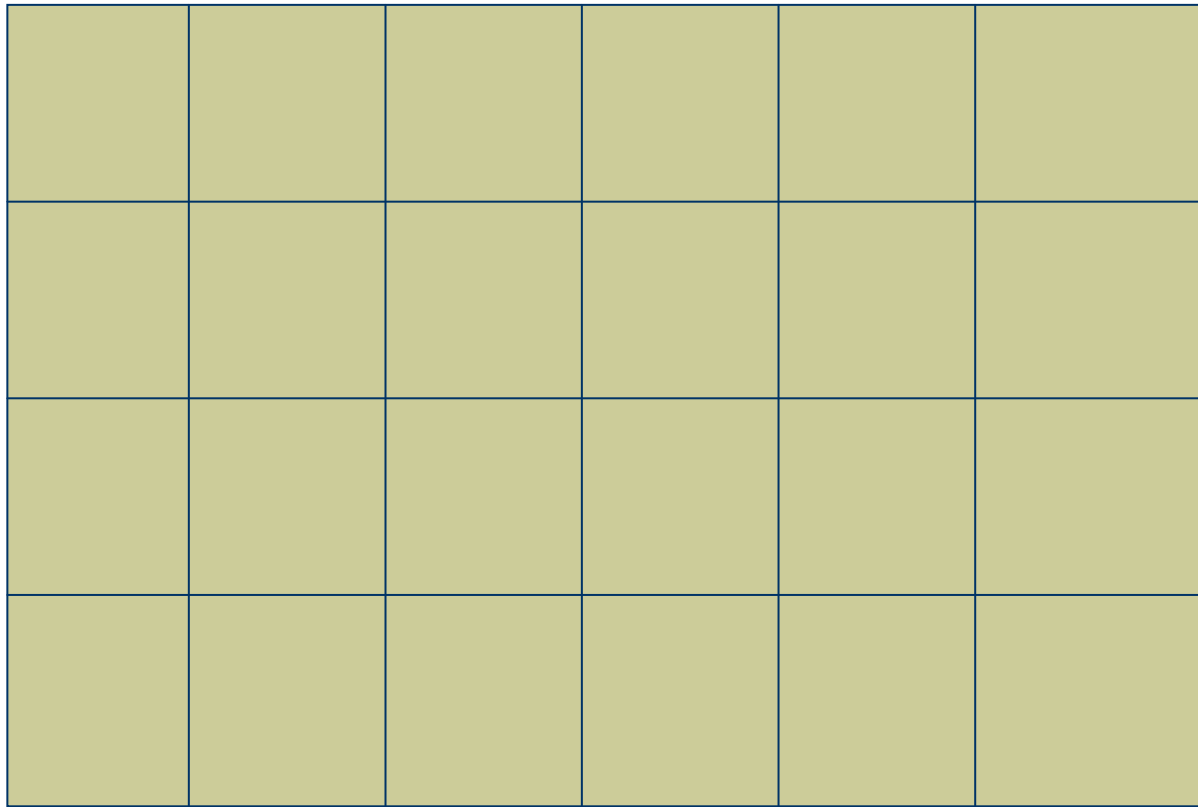


# CAN classique



# Application de Notre model dans CAN

4 LandMarks → Partitionner l'espace Can en  $4!$  zones, soit 24



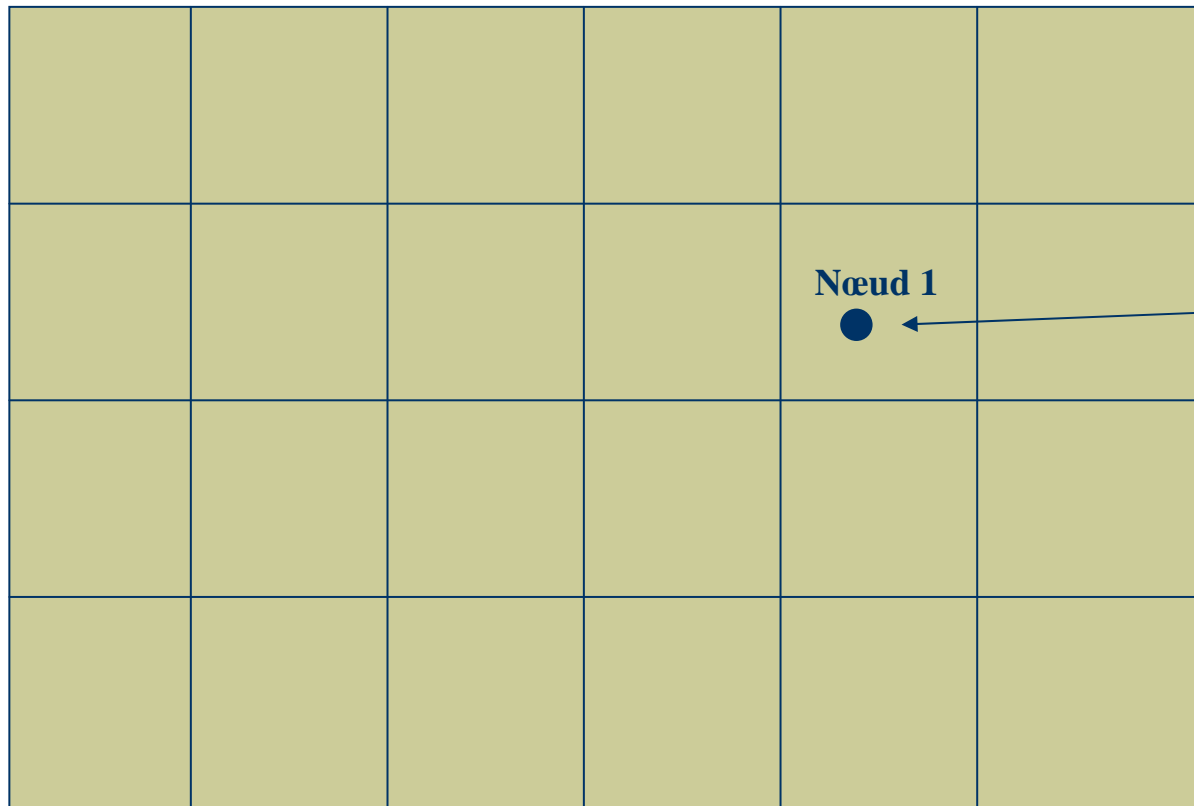
# Application de Notre model dans CAN

Chaque zone correspond à un Bin particulier.

Un nouveau nœud doit d'abord déterminer à quel Bin il appartient en faisant les mesures nécessaires

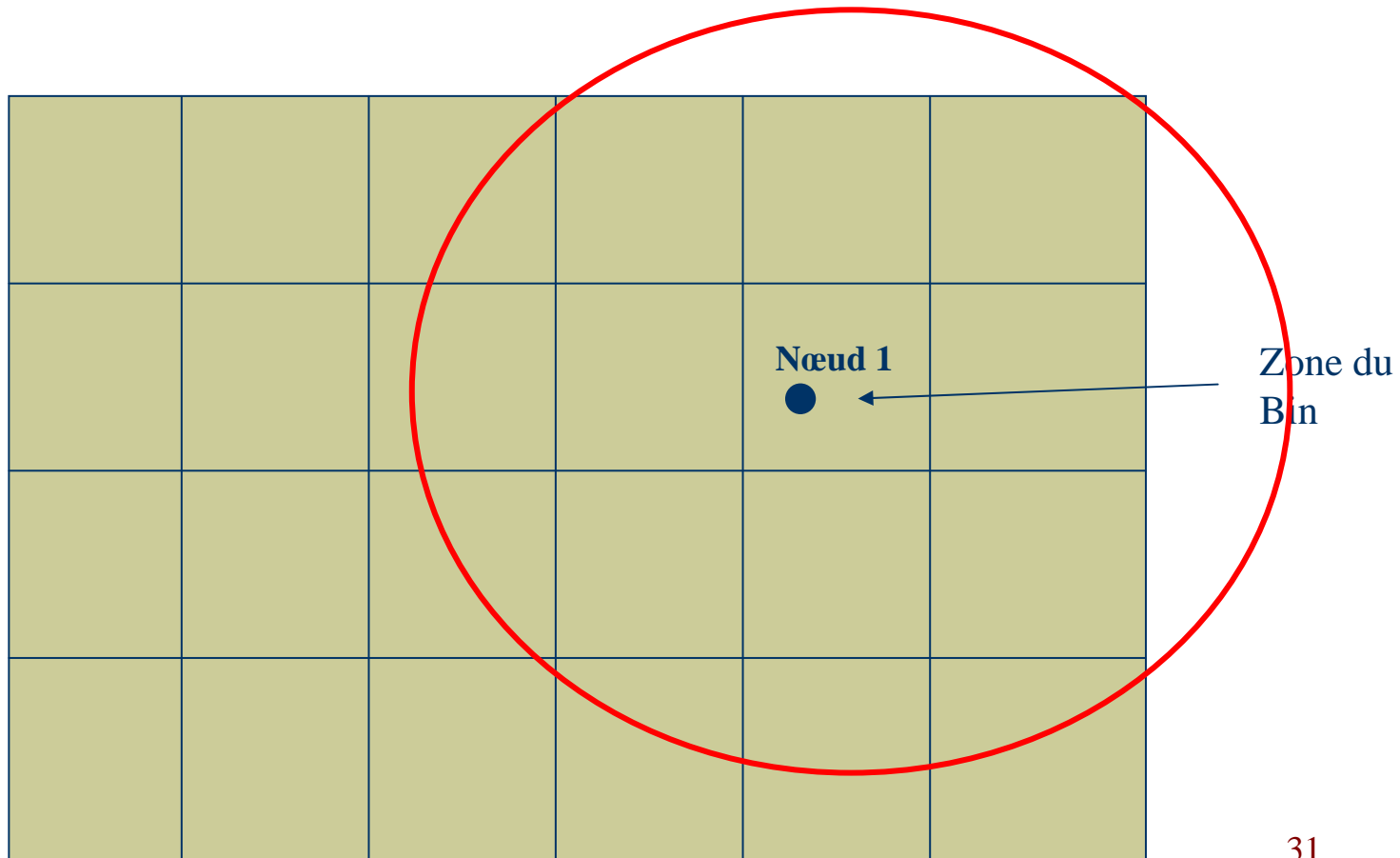
Le nœud sera ensuite déposé aléatoirement dans la zone correspondante à son Bin.

# Application de Notre model dans CAN

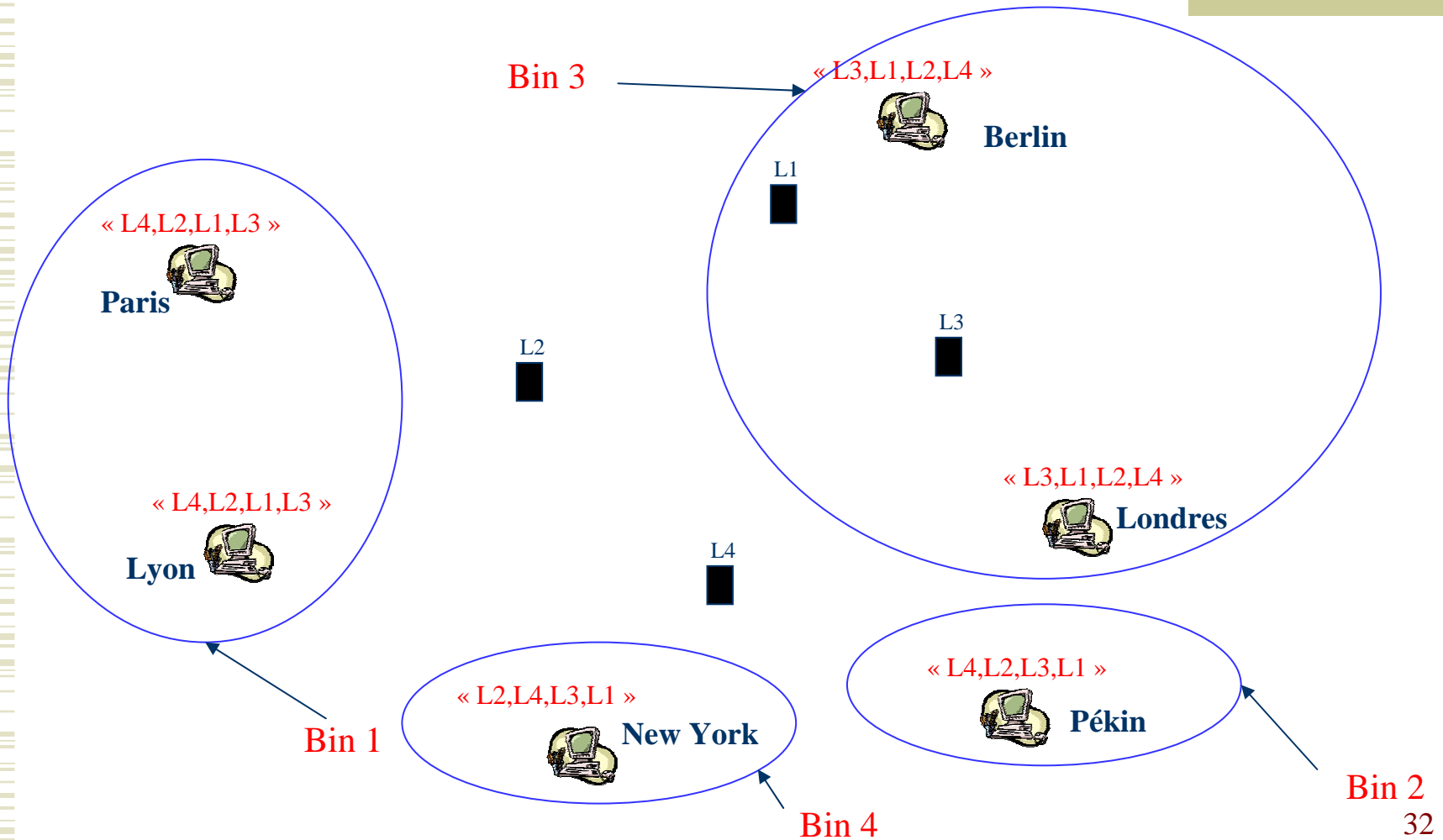


Zone du  
Bin

# Application de Notre model dans CAN

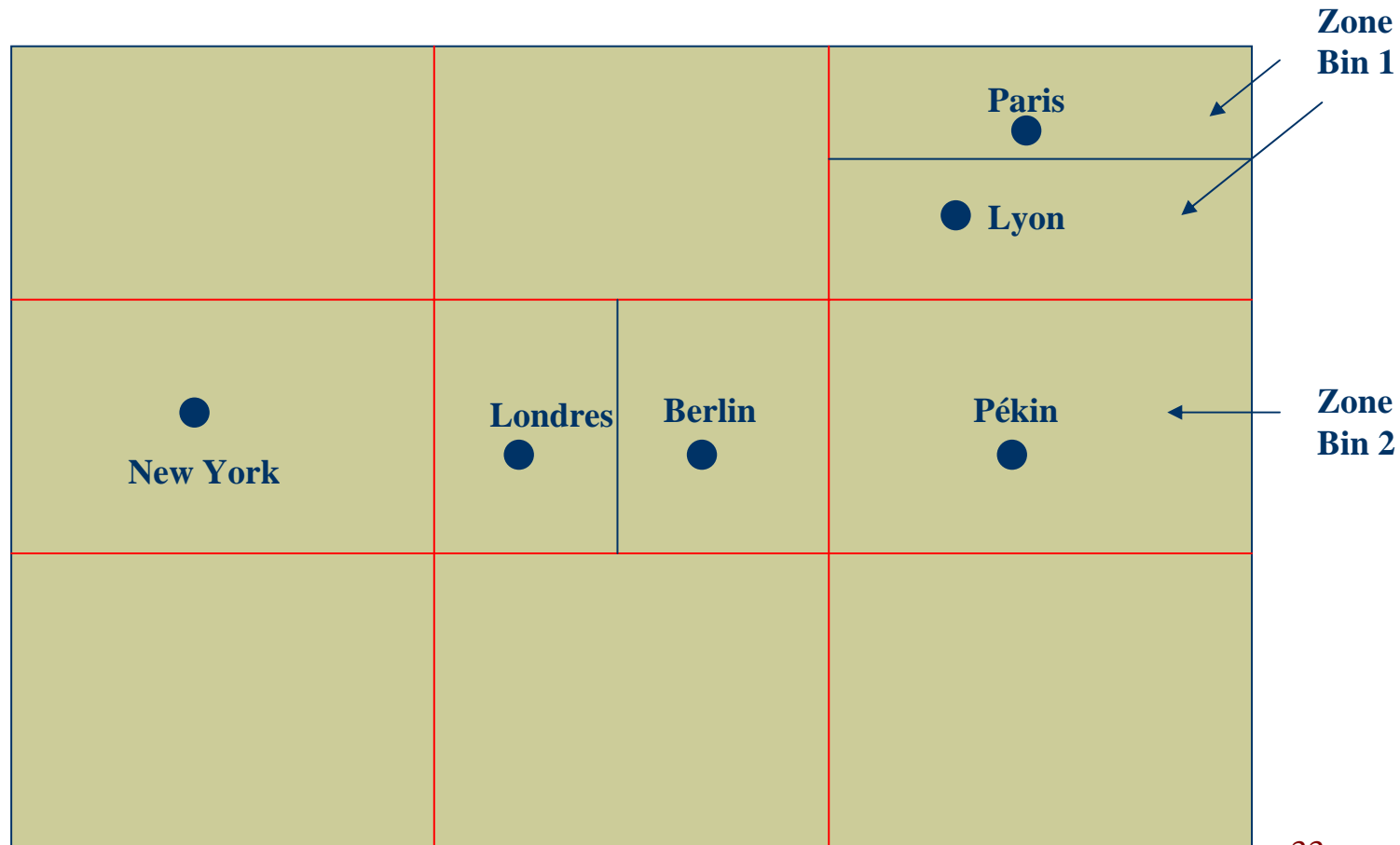


# Modèle proposé

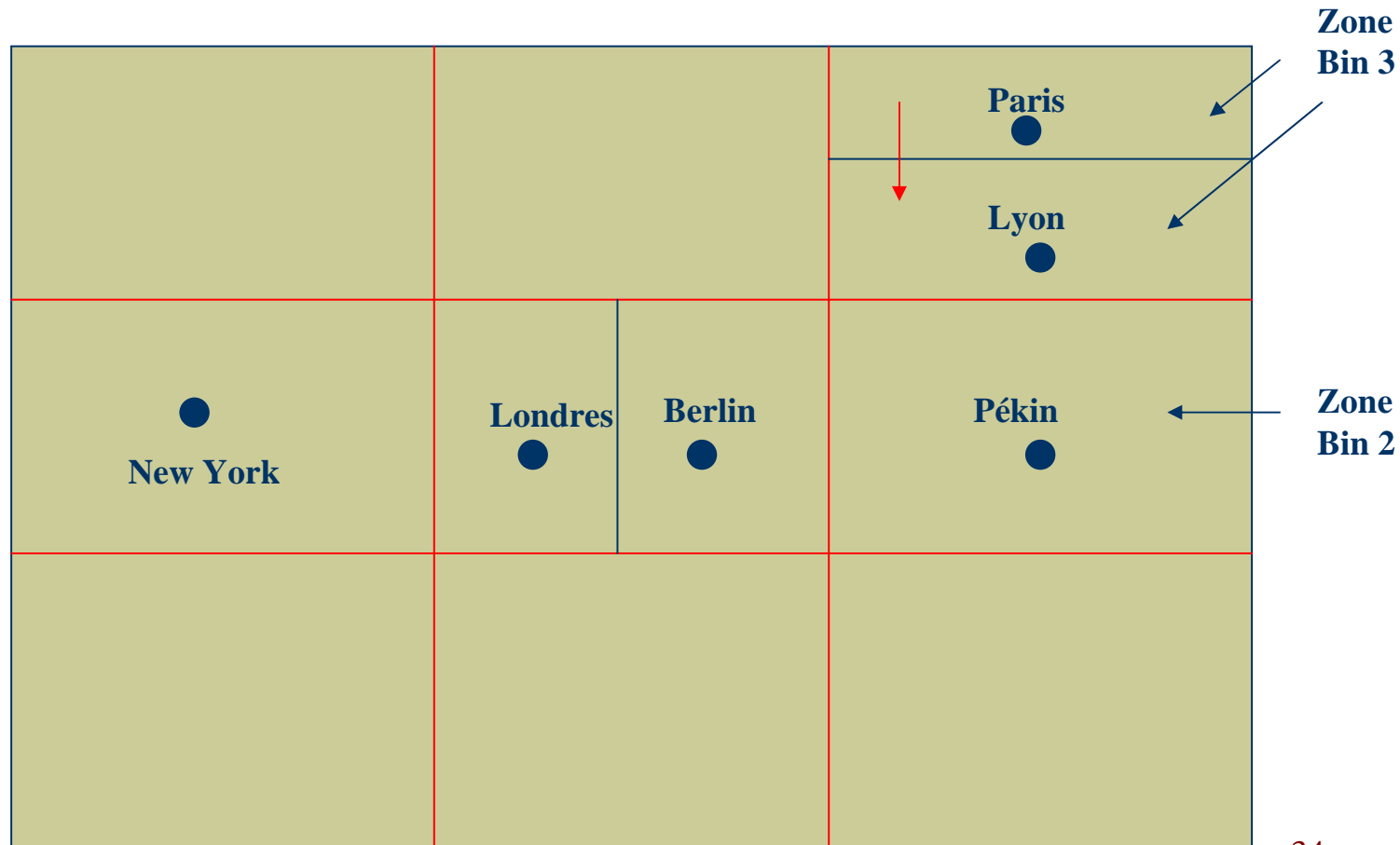




# Application de Notre model dans CAN



# Application de Notre model dans CAN



# Application de Notre model dans CAN

- Résultat :
- un espace CAN qui reflète ou qui prend en compte la topologie physique, et les temps de latences entre les nœuds
  - routage applicatif optimal avec temps de latence minimal

# Application de Notre model pour les reseaux non structurés

Modèle de base : on a un ensemble de  $N$  nœuds et chaque nœud choisi  $K$  nœud de cet ensemble

Stratégie optimale : Short-Long

Chaque nœud choisi les  $K/2$  nœuds les plus proches et les  $K/2$  autres nœuds au hasard.

Problème : pas scalable, car chaque nœud doit connaître tout les autres nœuds pour choisir les  $K/2$  plus proches

# Application de Notre model pour les reseaux non structurés

## Solution :

- En appliquant notre modèle de « Binning », un noeud peut choisir les  $K/2$  nœuds les plus proches de lui parmi ceux de son Bin, ce qu'il lui évitera de connaître tout le réseau.
- Si le nombre  $n$ 'est pas suffisant dans son Bin, le noeud va choisir des nœuds du Bin qui est supposé le plus proche de lui.
- Dans l'article, les auteurs ont supposé que le Bin le plus proche d'un autre Bin est celui qui a le plus de positions en communs avec lui dans la liste des LandMarks



# Application de Notre modèle pour la sélection de serveur



1. S'il existe au moins un serveur dans le même Bin que le client, alors celui-ci sera choisi parmi ceux là.
2. Dans le cas contraire, il faut prendre un serveur dans un Bin qui présente le plus de similarité avec le Bin du client.

# Critiques et Discussions

- a) Pour son application dans CAN, le modèle du Binning n'est pas dynamique car si un Landmark tombe en panne il faut reconstruire tout l'espace CAN.
- b) La latence peut ne pas être un bon facteur de décision pour prédire le débit TCP, surtout pour les nœuds à large bande.
- c) Ce modèle ne résoud pas complètement le problème de scalabilité.



# Critiques et Discussions

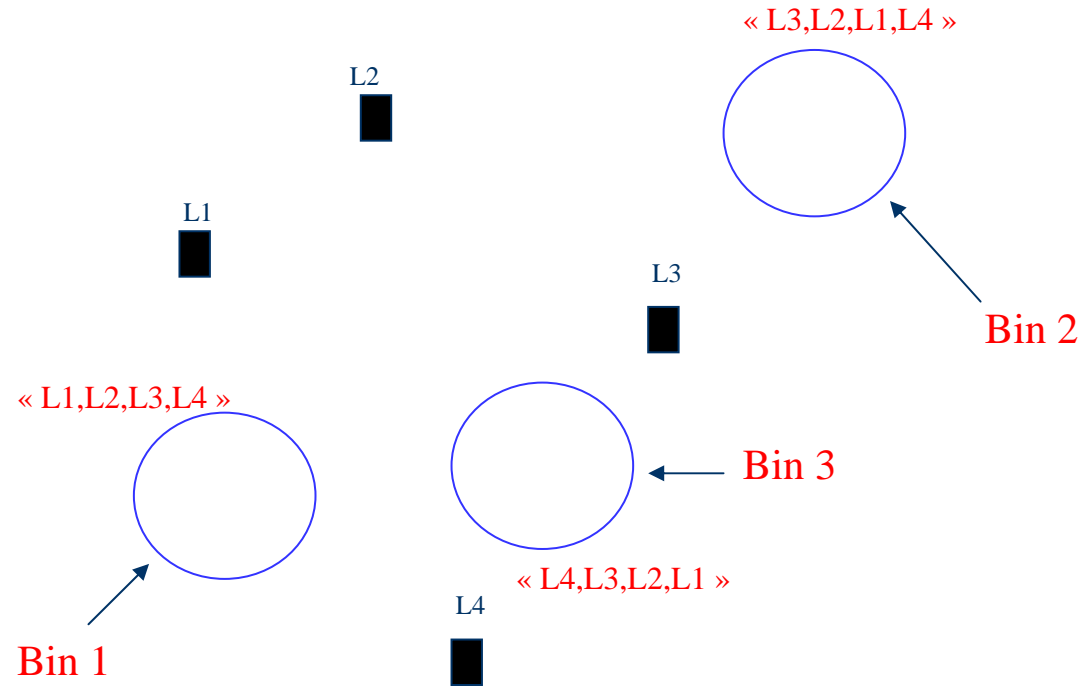


Il faut bien faire attention au choix des LandMarks et à leurs positions

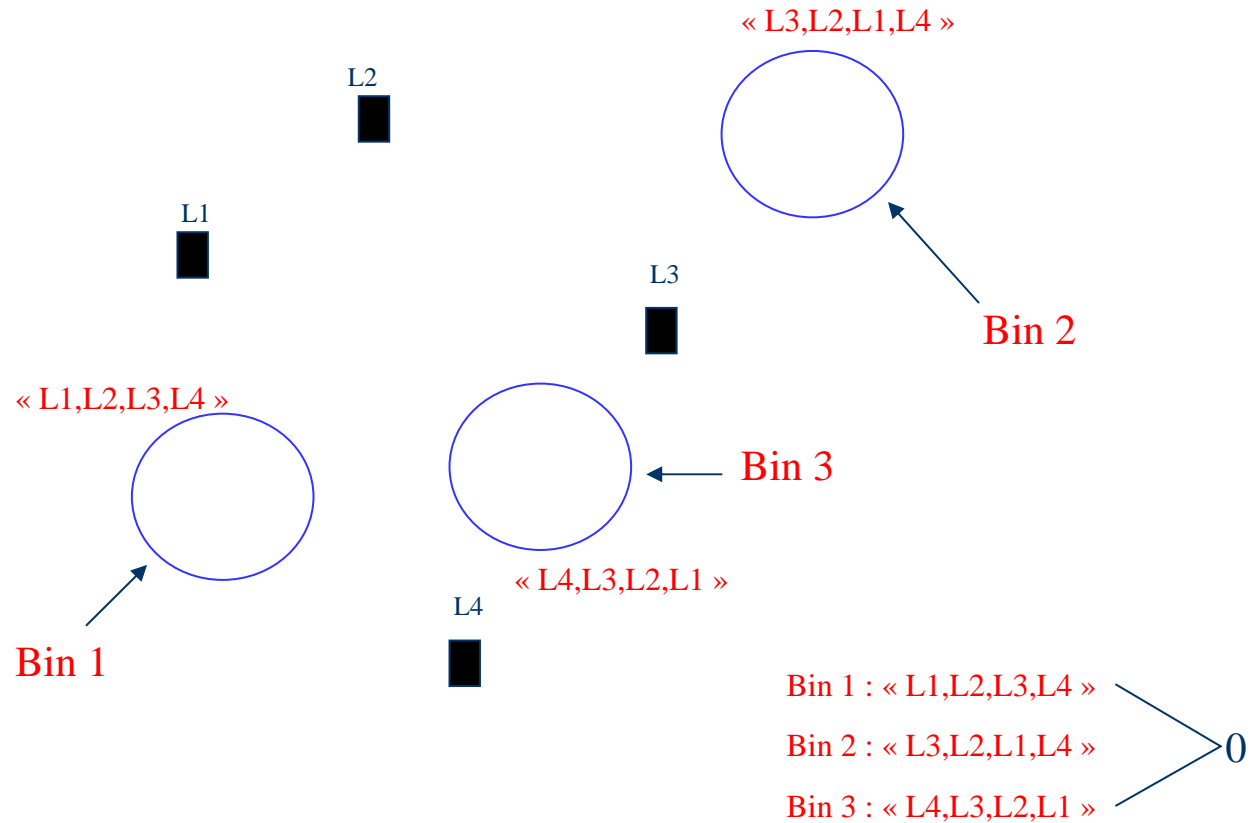
Car tout le modèle et les mesures reposent sur les positions des LandMarks, et un mauvais choix peut donner lieu à une mauvaise interprétation des mesures



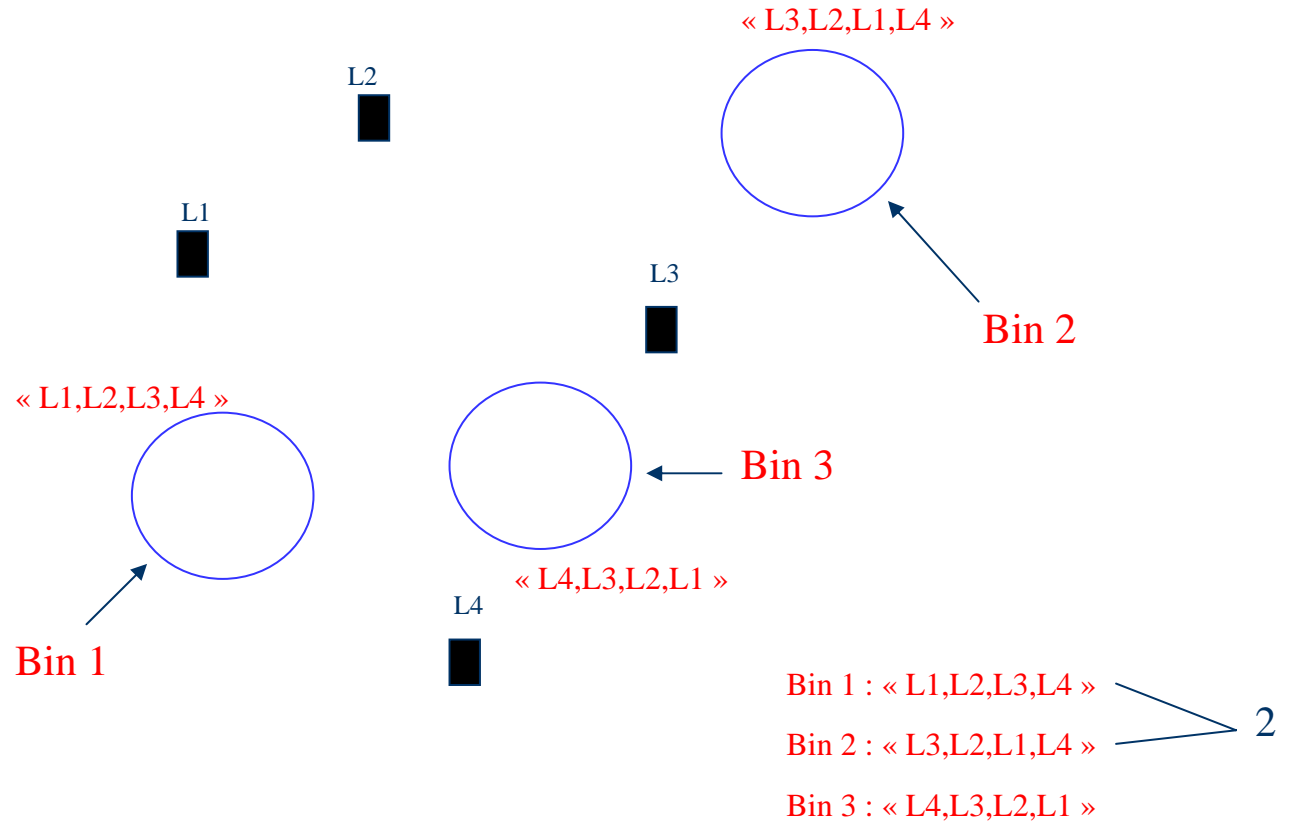
# Mauvais choix



# Mauvais choix



# Mauvais choix





# Optimisation (a mon avis)

En divisant l'espace CAN entre les différents Bins de telle façon que les Bins qui sont adjacents dans l'espace CAN le soient dans la réalité, ne peut qu'optimiser encore plus le routage



# Conclusion

Objectif principal : permettre aux applications le passage à l'échelle

Le principe adopté par les auteurs est que :

La simplicité, la scalabilité et la facilité d'utilisation sont plus importants que la précision.

# References

- **A Scalable Content-Adressable Network**  
S. Ratnasamy, P Francis, M Handley, R Karp and S Shenker
- **Some Finding on the Networks Performance of BroadBand Hosts**  
Karthik Lakshminarayanan Berkley, Venkata N.Padmanabhan Microsoft
- **Structuring Topologically-Aware Overlay Networks using Domain Names**  
Demetrios Zeinalipour-Yazti and Vana Kalogeraki University of California
- **Efficient Topology-Aware Overlay Network**  
Marcel Waldvogel, Roberto Rinaldi IBM Research
- **Turning Heterogeneity into an Advanatage in Overlay Routing**  
Zhichen Xu, Mallik Mahalingam, Magnus Karlsson HP Laboratory
- **Approximate Server Selection Algorithms in Content Distribution Networks**  
Spiridon Bakiras University of Hong Kong
- **Clustering Hosts in P2P and Global Computing Platforms**  
Abhishek Agrawal and Henri Casanova University of California



*Merci pour votre attention*