PairAPair : algorithmique des protocoles de pair à pair

Laurent Viennot

ACI MD, PaRISTIC 2005

Équipes

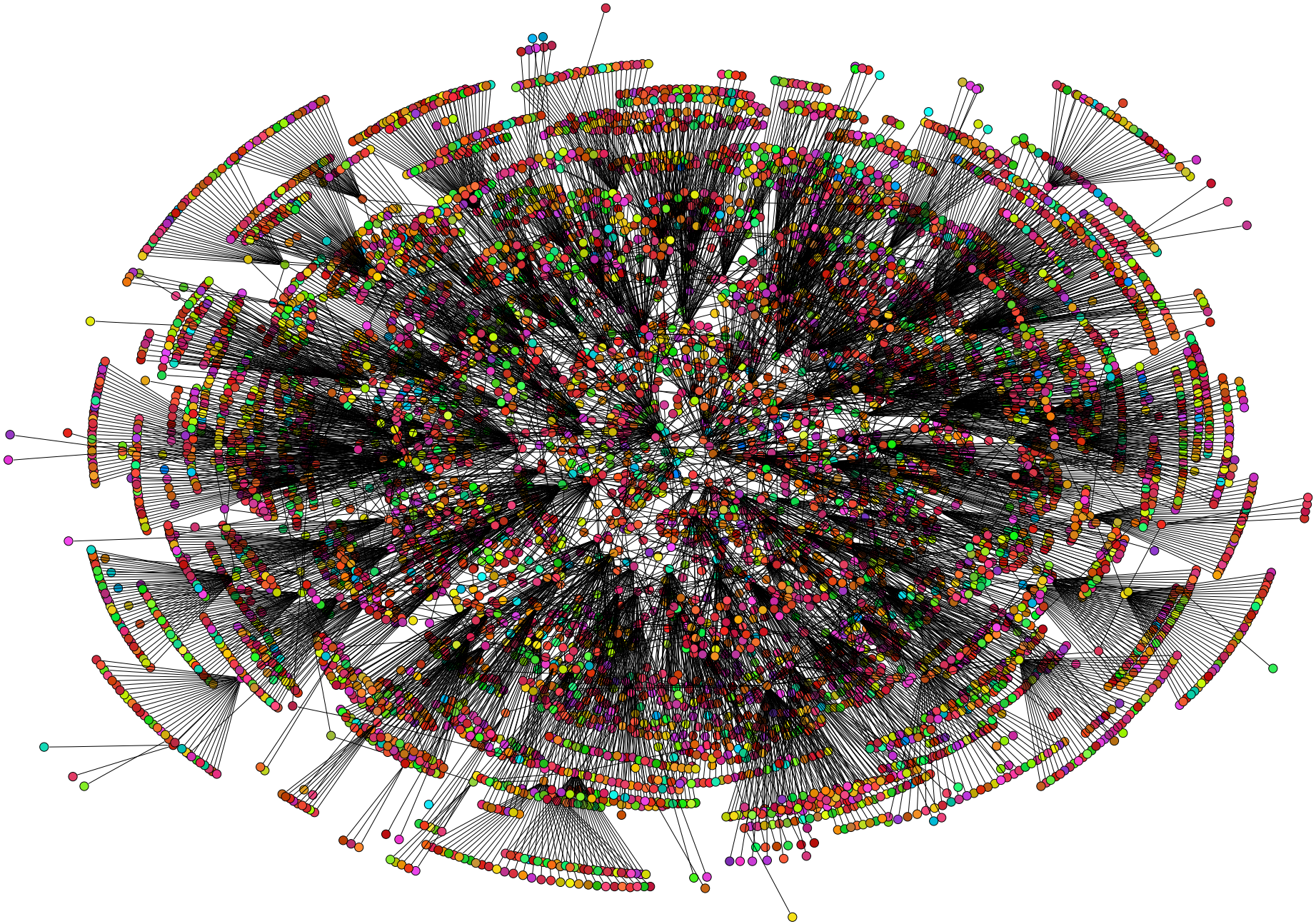
Fondateurs

- Inria-Liafa , Gyroweb (Laurent Viennot)
- LRI, Grafcomm (Pierre Fraigniaud)
- LaBRI, Graphes (Cyril Gavoille)
- Inria, Hipercom (Cédric Adjih)

Auto-sponsorisés réguliers

- Inria-Insa Lyon, Ares (Eric Fleury)
- FT, MAPS/MMC (Joaquin Keller)

Qu’est-ce qu’un réseau de pair à pair ?



Qu'est-ce qu'un réseau de pair à pair ?

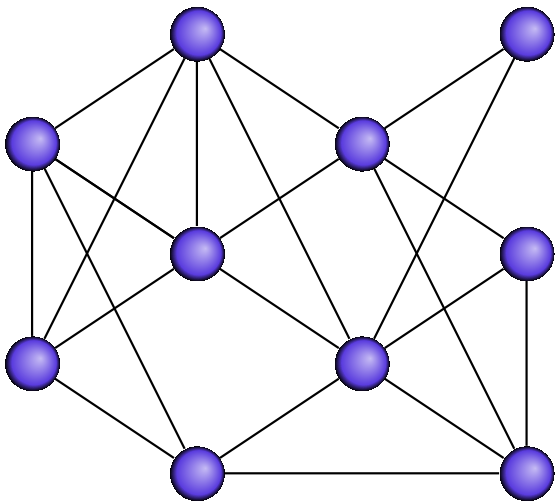
Un réseau logique décentralisé

- Pair à pair = « peer-to-peer ».
- Réseau décentralisé, pas de serveur.
- Connexions client-client : entre pairs.
- Réseau logique entre les clients.

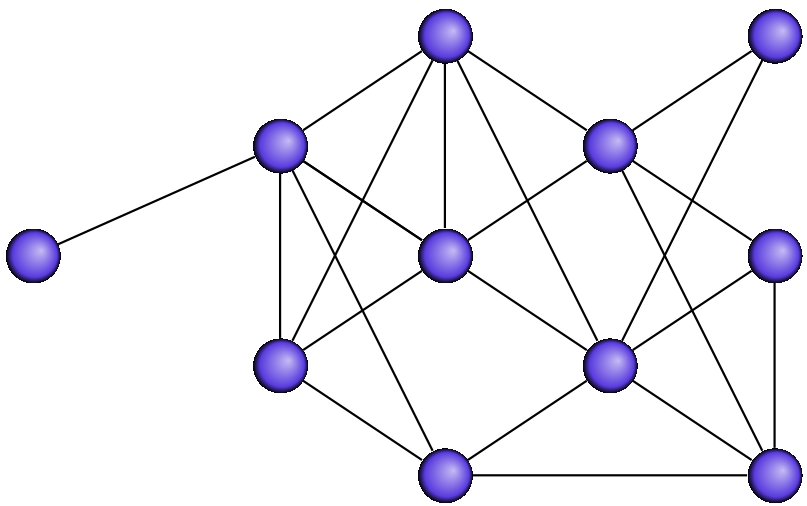
Concrètement

- Chacun fait tourner un programme sur son ordinateur.
- Ce programme ce connecte (via Internet) à d'autres ordinateurs faisant tourner le même programme.
- Une fois connecté, on peut s'échanger des données, des fichiers par exemple.

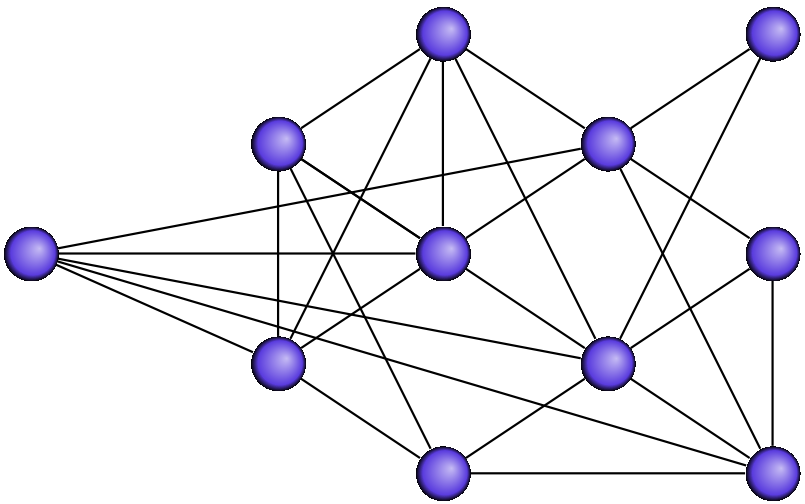
Arrivée d'un pair



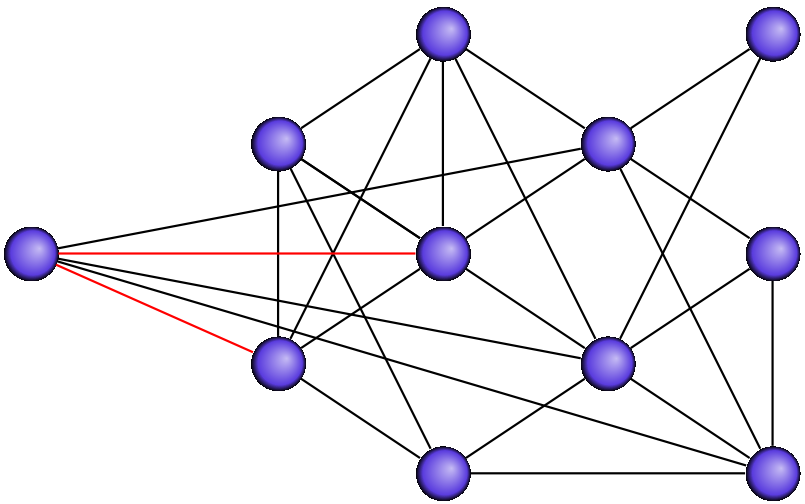
Arrivée d'un pair



Arrivée d'un pair



Arrivée d'un pair



Routage

Problème à résoudre

- Comment trouver celui qui a telle donnée ?
- Il y a beaucoup de pairs.
- À qui me connecter ?
- Il y a sans cesse des nouveaux arrivant et des partants.

Internet

- Comment atteindre celui qui gère telle adresse IP ?
- À qui suis-je connecté ?

Routage

Problème à résoudre

- Comment trouver celui qui a telle donnée ?
- Il y a beaucoup de pairs.
- À qui me connecter ?
- Il y a sans cesse des nouveaux arrivant et des partants.

Internet

- Comment atteindre celui qui gère telle adresse IP ?
- À qui suis-je connecté ?

Routage

Problème à résoudre

- Comment trouver celui qui a telle donnée ?
- Il y a beaucoup de pairs.
- À qui me connecter ?
- Il y a sans cesse des nouveaux arrivant et des partants.

Internet

- Comment atteindre celui qui gère telle adresse IP ?
- À qui suis-je connecté ?

Routage

Problème à résoudre

- Comment trouver celui qui a telle donnée ?
- Il y a beaucoup de pairs.
- À qui me connecter ?
- Il y a sans cesse des nouveaux arrivant et des partants.

Internet

- Comment atteindre celui qui gère telle adresse IP ?
- À qui suis-je connecté ?

Routage

Problème à résoudre

- Comment trouver celui qui a telle donnée ?
- Il y a beaucoup de pairs.
- À qui me connecter ?
- Il y a sans cesse des nouveaux arrivant et des partants.

Internet

- Comment atteindre celui qui gère telle adresse IP ?
- À qui suis-je connecté ?

Routage

Problème à résoudre

- Comment trouver celui qui a telle donnée ?
- Il y a beaucoup de pairs.
- À qui me connecter ?
- Il y a sans cesse des nouveaux arrivant et des partants.

Internet

- Comment atteindre celui qui gère telle adresse IP ?
- À qui suis-je connecté ?

Réseaux non structurés

Routage par inondation

- Diffuser la question à tout le monde.
- Quand un client reçoit une requête, il la propage à ses voisins (inondation).
- Avec n participants posant tous des requêtes, on arrive à $\Omega(n^2)$ messages.

Optimisations

- Hiérarchiser le réseau.
- Utiliser des propriétés du graphe de connexion tel qu'il est pour optimiser les mécanismes du protocole [FGL05].
- Analyse de systèmes existants [lBV04, GlB04].

Réseaux non structurés

Routage par inondation

- Diffuser la question à tout le monde.
- Quand un client reçoit une requête, il la propage à ses voisins (inondation).
- Avec n participants posant tous des requêtes, on arrive à $\Omega(n^2)$ messages.

Optimisations

- Hiérarchiser le réseau.
- Utiliser des propriétés du graphe de connexion tel qu'il est pour optimiser les mécanismes du protocole [FGL05].
- Analyse de systèmes existants [IBV04, GIB04].

Réseaux non structurés

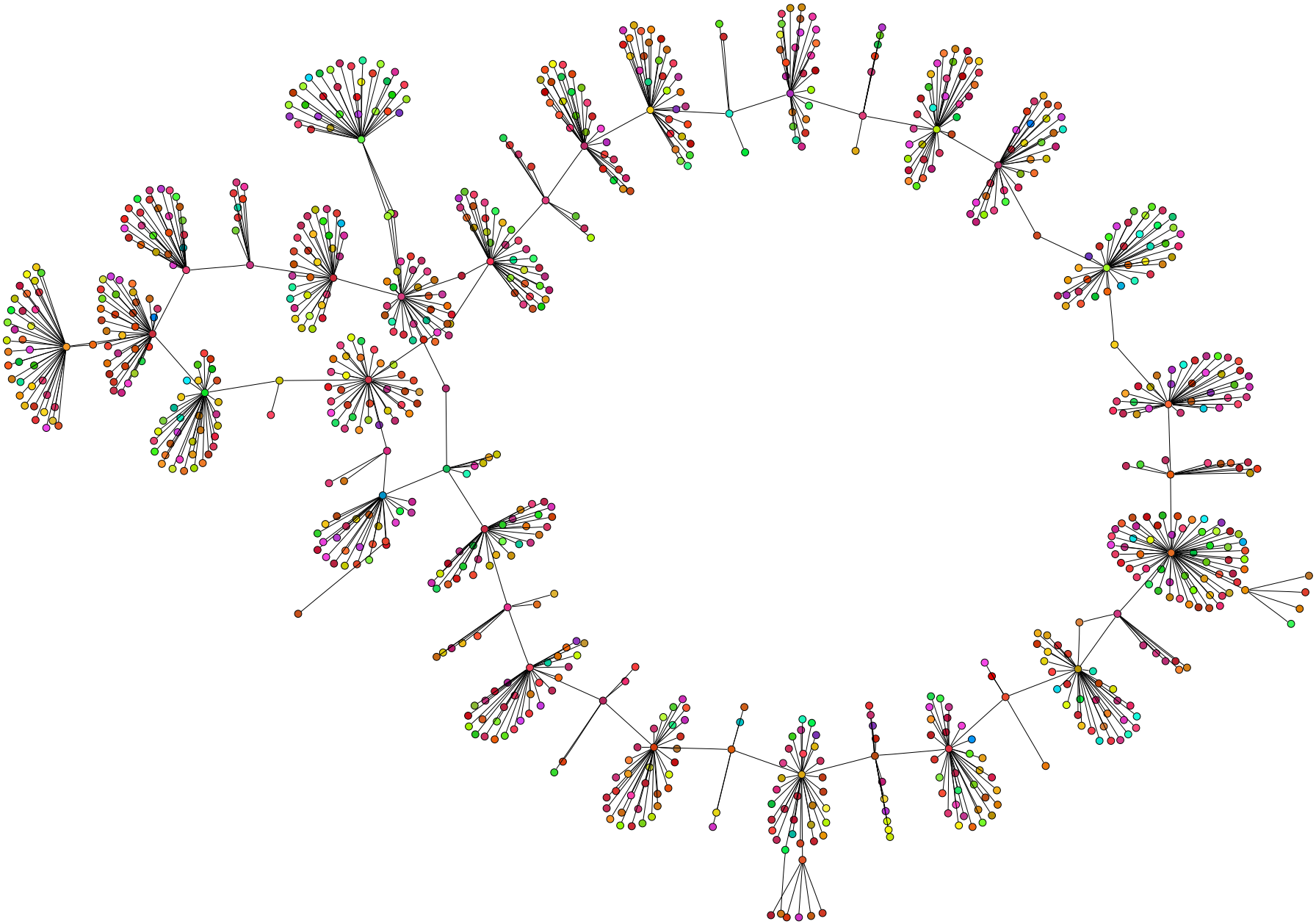
Routage par inondation

- Diffuser la question à tout le monde.
- Quand un client reçoit une requête, il la propage à ses voisins (inondation).
- Avec n participants posant tous des requêtes, on arrive à $\Omega(n^2)$ messages.

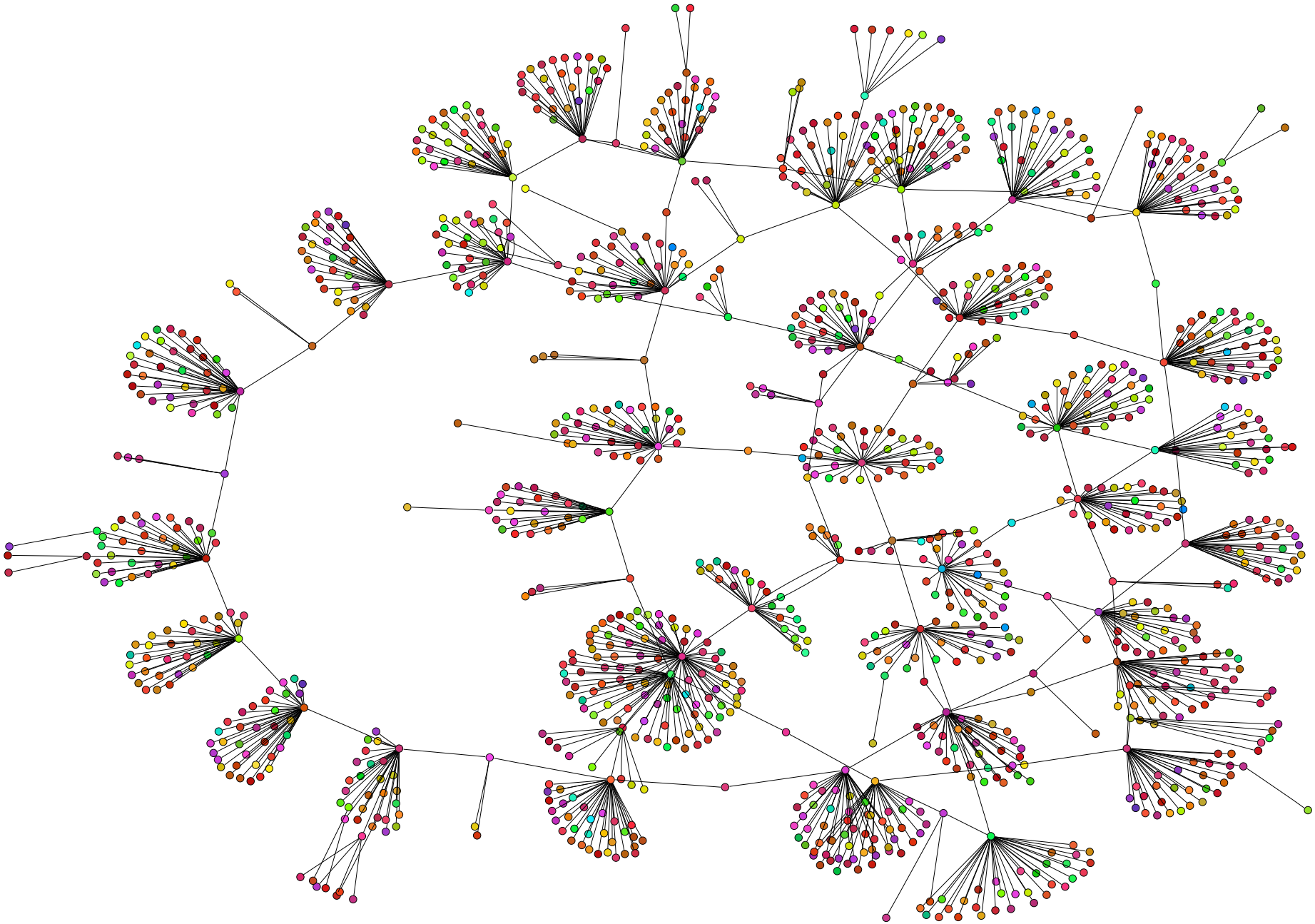
Optimisations

- Hiérarchiser le réseau.
- Utiliser des propriétés du graphe de connexion tel qu'il est pour optimiser les mécanismes du protocole [FGL05].
- Analyse de systèmes existants [lBV04, GlB04].

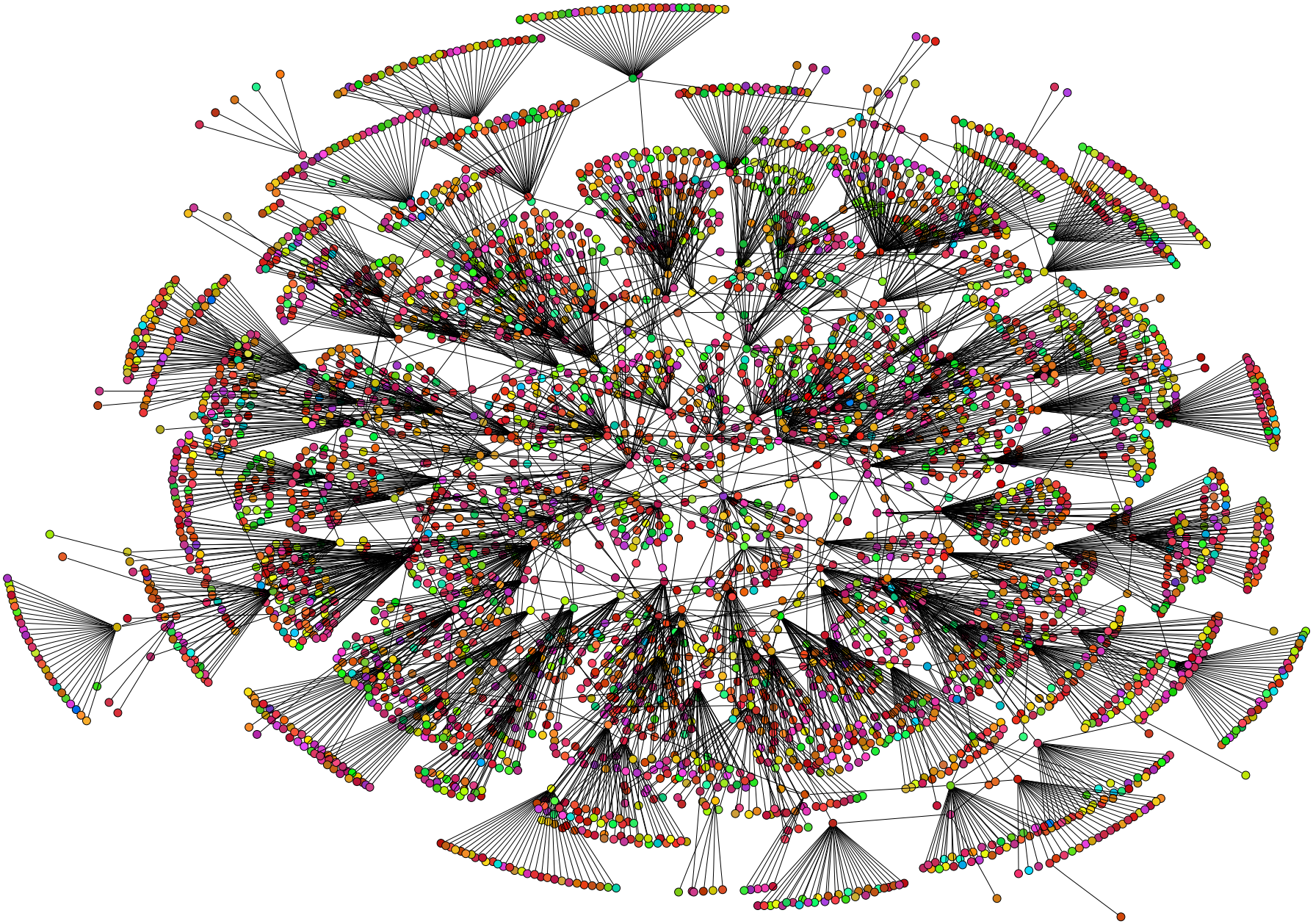
Gnutella, 50 nœuds visités



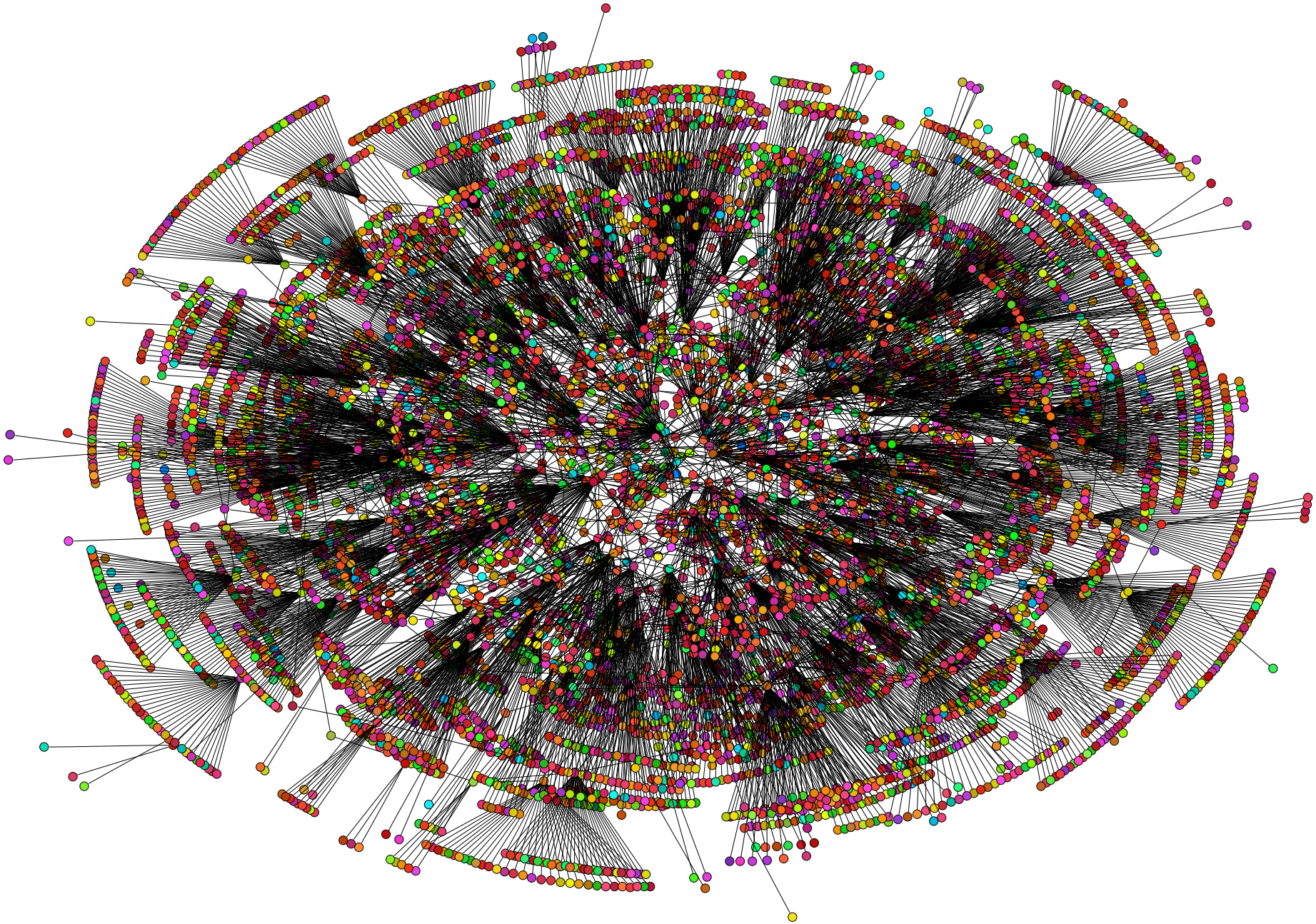
Gnutella, 100 nœuds visités



Gnutella, 250 nœuds visités



Gnutella, 400 nœuds visités



Réseaux structurés

Choisir ses voisins

- Donner des identifiants aux données (par exemple, Sha1 du nom de fichier).
- Choisir ses voisins en fonctions des identifiants des données qu'ils possèdent (skiplist par exemple).
- Donner des identifiants aux nœuds et choisir ses voisins en fonction de leurs identifiants (tables de hachages distribuées (DHT), par exemple).

Qui indexe telle donnée ?

- Chacun indexe ses propres données (skiplist).
- La donnée d'identifiant k est indexé par le nœud d'identifiant u le plus proche de k (DHT).

DHT base

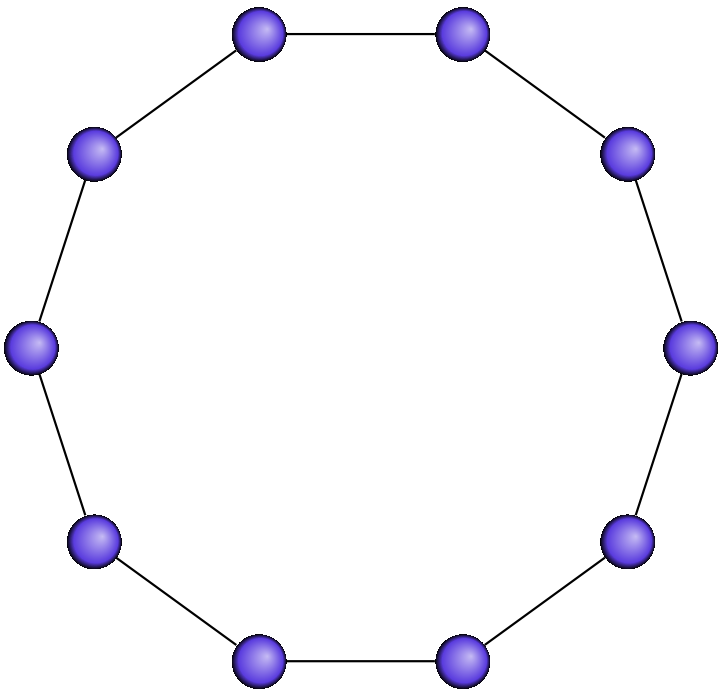
Hachage

- Espace logique K.
- Fonction de hachage h : donnée $A \rightarrow h(A) = k \in K$.
- Stocker A sur le nœud u tel que dist(k,u) minimale.

Routage par la clé

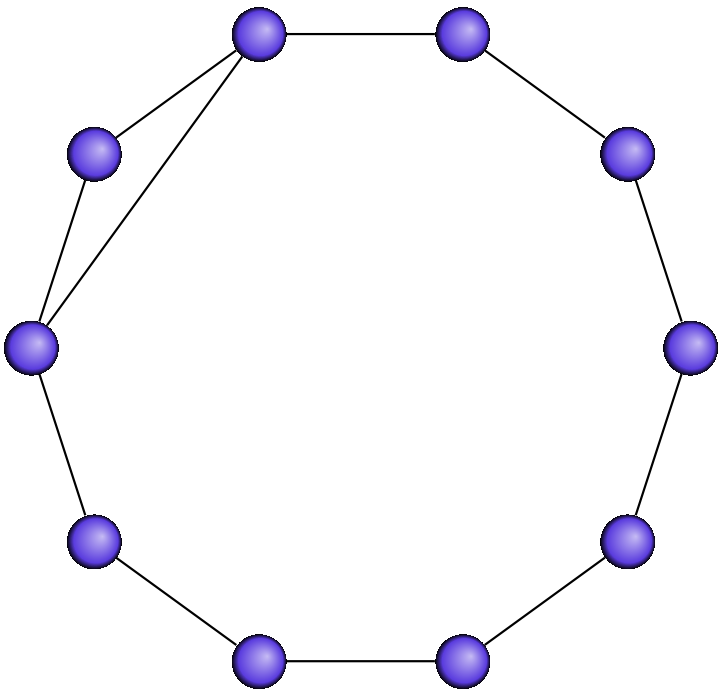
- u connaît les nœuds v tels que dist(u,v) petite.
- Si u n'est pas le plus proche de k, il connaît au moins un nœud strictement plus proche.
- De proche en proche, on peut trouver le nœud le plus proche de k.

DHT Tore de dimension *d*

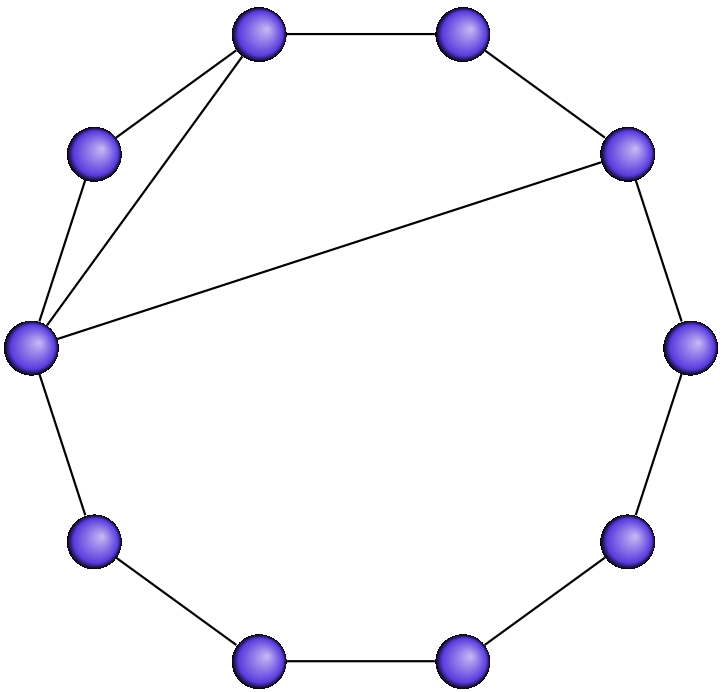


Exemple : CAN.

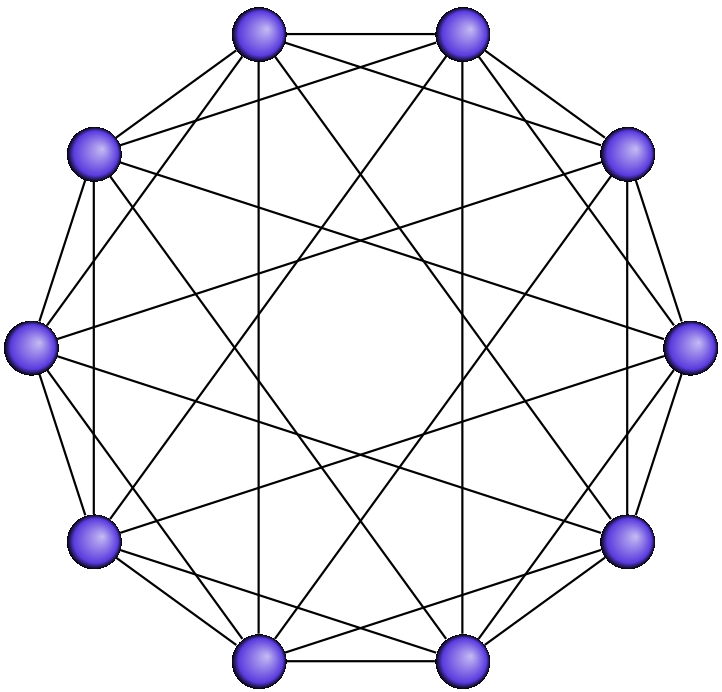
DHT Tore de dimension d augmenté



DHT Tore de dimension *d* augmenté

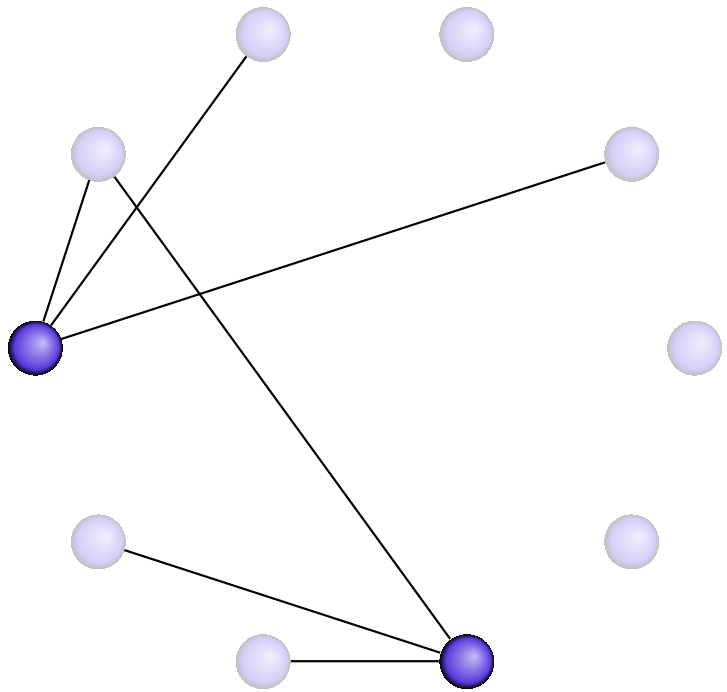


DHT Tore de dimension d augmenté



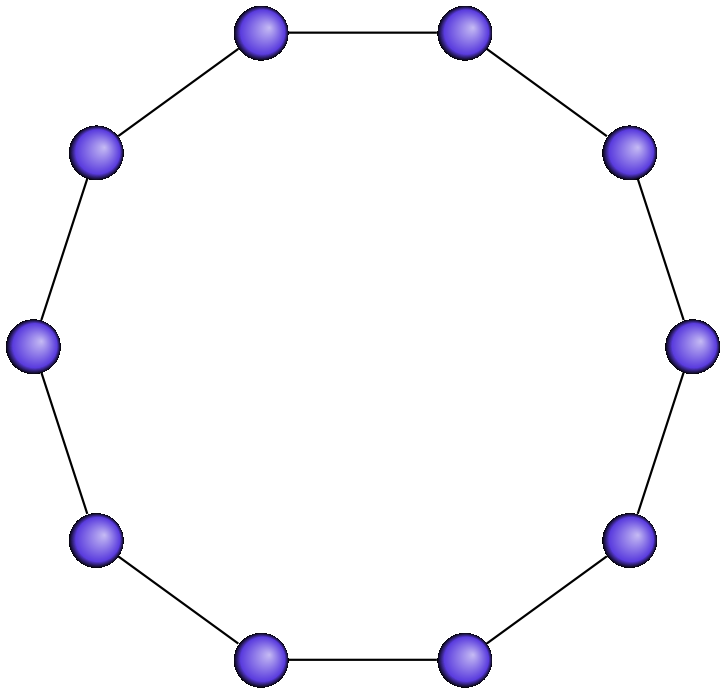
Idée de Chord ($d = 2$).

DHT Tore de dimension d augmenté



Skiplist ($d = 2$).

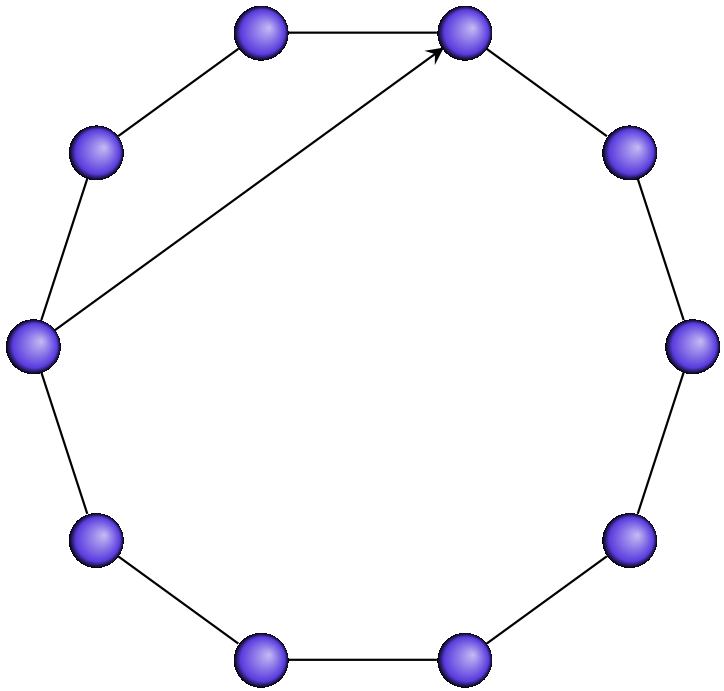
Petit monde à la Kleinberg



Tore augmenté aléatoirement

- Un tore connu de tous.
- Plus un lien aléatoire pour chaque nœud u vers v avec probabilité proportionnelle à $1/|B(\text{dist}(u,v))|$.
- Un nœud ne connaît pas les liens aléatoires des autres.

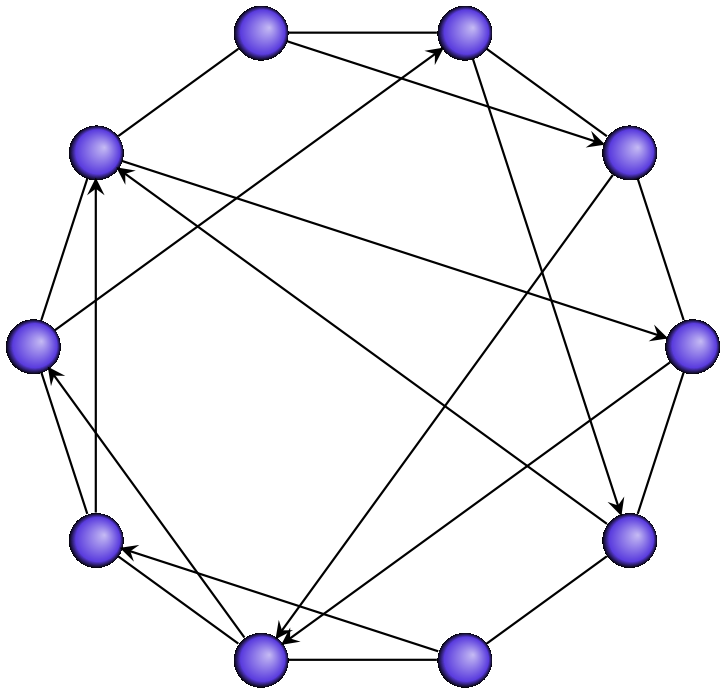
Petit monde à la Kleinberg



Tore augmenté aléatoirement

- Un tore connu de tous.
- Plus un lien aléatoire pour chaque nœud u vers v avec probabilité proportionnelle à $1/|B(\text{dist}(u,v))|$.
- Un nœud ne connaît pas les liens aléatoires des autres.

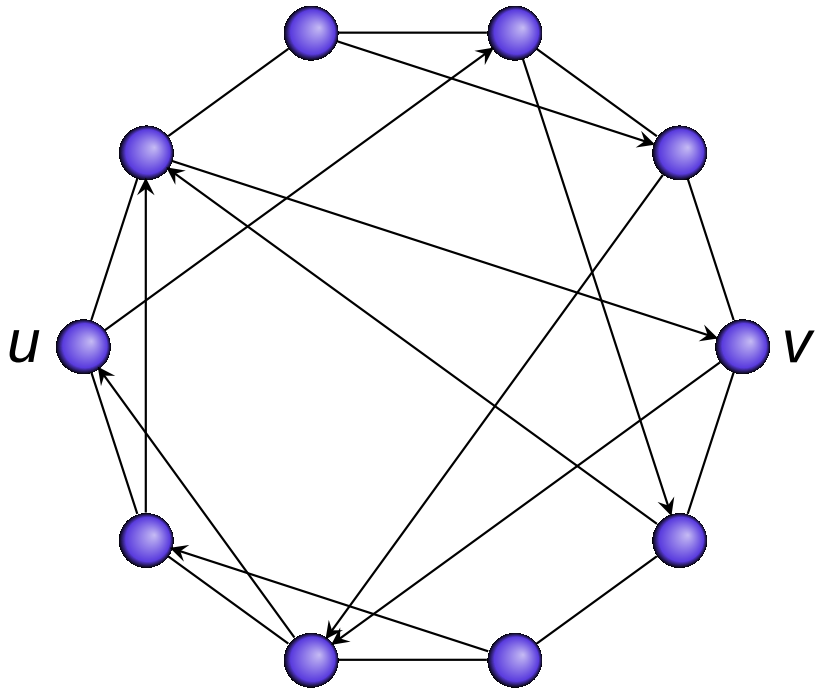
Petit monde à la Kleinberg



Tore augmenté aléatoirement

- Un tore connu de tous.
- Plus un lien aléatoire pour chaque nœud u vers v avec probabilité proportionnelle à $1/|B(\text{dist}(u,v))|$.
- Un nœud ne connaît pas les liens aléatoires des autres.

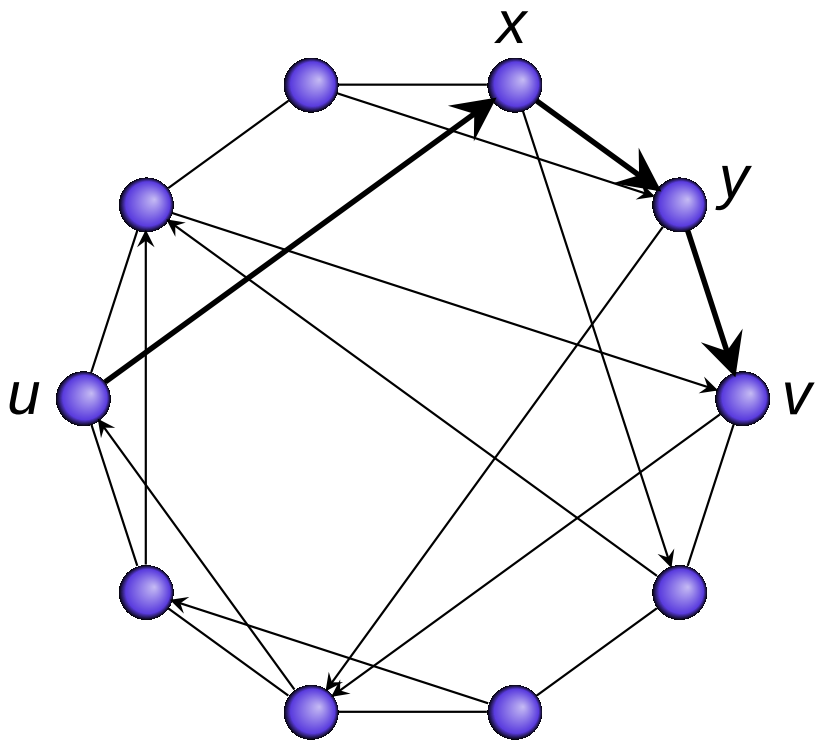
Petit monde à la Kleinberg



Tore augmenté aléatoirement

- Un tore connu de tous.
- Plus un lien aléatoire pour chaque nœud u vers v avec probabilité proportionnelle à $1/|B(\text{dist}(u,v))|$.
- Un nœud ne connaît pas les liens aléatoires des autres.

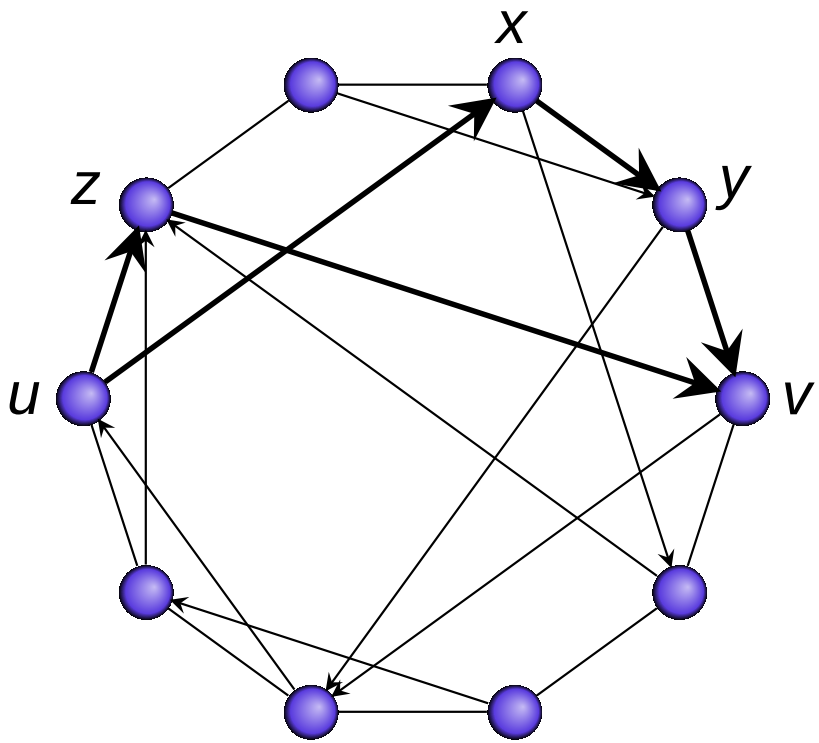
Petit monde à la Kleinberg



Tore augmenté aléatoirement

- Un tore connu de tous.
- Plus un lien aléatoire pour chaque nœud u vers v avec probabilité proportionnelle à $1/|B(\text{dist}(u,v))|$.
- Un nœud ne connaît pas les liens aléatoires des autres.

Petit monde à la Kleinberg



Tore augmenté aléatoirement

- Un tore connu de tous.
- Plus un lien aléatoire pour chaque nœud u vers v avec probabilité proportionnelle à $1/|B(\text{dist}(u,v))|$.
- Un nœud ne connaît pas les liens aléatoires des autres.

Petit monde à la Kleinberg

Routage glouton

- Faire passer au voisin le plus proche de la destination dans le graphe connu.
- Routes en $O(\log^2 n)$.

Extensions

- [FGP04] Un nœud connaît les lien aléatoires des nœuds dans sa boule de taille $\log n$. Routes en $O(\log^{1+1/d} n)$.
- [DHLS05] Peut-on rajouter des liens aléatoire à un graphe quelconque de sorte à obtenir des routes courtes ? Oui si la taille des boules ne croît pas trop vite avec le rayon.
- Il existe un résultat similaire si la métrique du graphe est s-doubling (Slivkins).

Autres DHTs structurées

Hypercube

- Chaque nœud $u = u_1u_2u_3\cdots$ connaît des noeuds de préfixes :
  $\overline{u_1}$, $u_1\overline{u_2}$, $u_1u_2\overline{u_3}$, ...
- Routage préfixe : pour atteindre $k_1k_2k_3\cdots$, passer par des nœuds de préfixes :
  k_1 , k_1k_2 , $k_1k_2k_3$, ...
- Plus proche = long préfixe commun = xor petit.
- Tables de voisinage en $O(\log n)$ et routes en $O(\log n)$.
- Exemples : Pastry, Kademlia.

Autres DHTs structurées

Graphe de Bruijn

- Chaque nœud $u = u_1u_2u_3\cdots$ connaît des noeuds de préfixes :
  $0u_1u_2u_3\cdots$, $1u_1u_2u_3\cdots$.
- Routage par décallage : pour atteindre $k_1k_2k_3\cdots$, passer par des nœuds de préfixes :
  $k_\ell u_1u_2u_3\cdots$, $k_{\ell-1}k_\ell u_1u_2\cdots$, $k_{\ell-2}k_{\ell-1}k_\ell u_1\cdots$, ... ,
  $k_2\cdots k_\ell u_1$, $k_1\cdots k_\ell$. $(\ell = \log n)$
- Tables de voisinage en $O(1)$ et routes en $O(\log n)$.

Résultats

- DHT de Bruijn [FG03], avec redondance de liens [GV04].
- Architecture de DHT pour l'indexation de mots clés [GV06].
- Graphe de Bruijn de clusters pour la distribution d'un flux de données [GV05].

Autres DHTs structurées

Graphe de Bruijn

- Chaque nœud $u = u_1u_2u_3\cdots$ connaît des noeuds de préfixes :
  $0u_1u_2u_3\cdots$, $1u_1u_2u_3\cdots$.
- Routage par décallage : pour atteindre $k_1k_2k_3\cdots$, passer par des nœuds de préfixes :
  $k_\ell u_1u_2u_3\cdots$, $k_{\ell-1}k_\ell u_1u_2\cdots$, $k_{\ell-2}k_{\ell-1}k_\ell u_1\cdots$, ... ,
  $k_2\cdots k_\ell u_1$, $k_1\cdots k_\ell$. $(\ell = \log n)$
- Tables de voisinage en $O(1)$ et routes en $O(\log n)$.

Résultats

- DHT de Bruijn [FG03], avec redondance de liens [GV04].
- Architecture de DHT pour l'indexation de mots clés [GV06].
- Graphe de Bruijn de clusters pour la distribution d'un flux de données [GV05].

Réseau logique versus réseau physique

Réseau logique rapide

- Spanner peu dense :
  sur les $O(n^2)$ liens de la clique des n pairs, on ne veut en garder que $o(n)$ par nœud.
- Routes courtes (par rapport au réseau physique) :
  la route de u à v ne doit pas être plus longue (somme des RTTs) que f fois le lien direct de u à v (RTT).

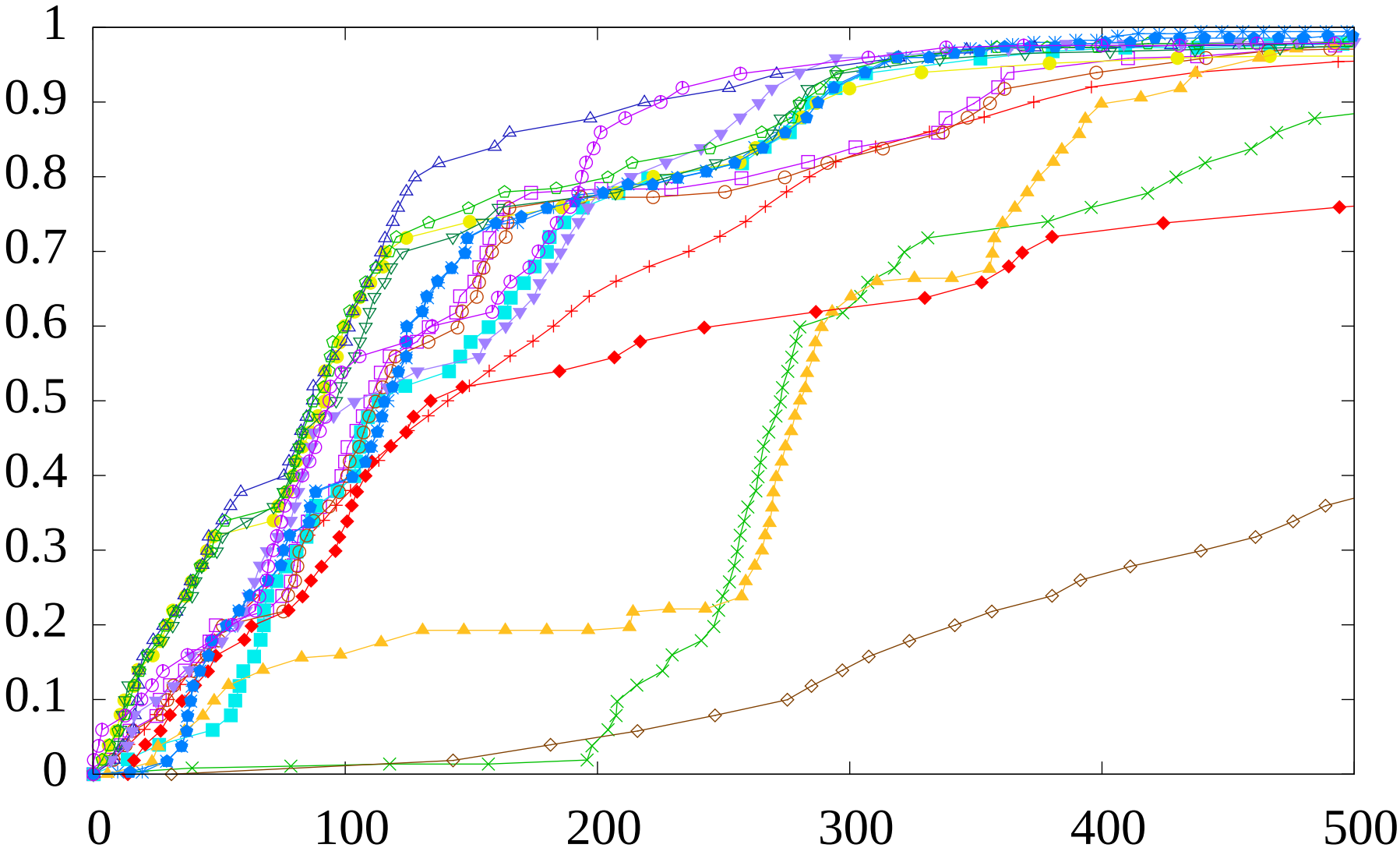
Résultats théoriques

- Avec $\widetilde{O}(n^{1/k})$ bits d'information par nœud, et des en-têtes de $O(\log^2 n)$ bits, on peut obtenir des tables de routage avec un facteur d'élongation $f \leq 4k-5$ ($k \geq 2$).
- $f < 3$ impossible avec $o(n)$ bits par nœud.
- Rafinements, notamment sans renommer les nœuds [AGM04,AGMNT04,DG04].

Métrique s-doubling

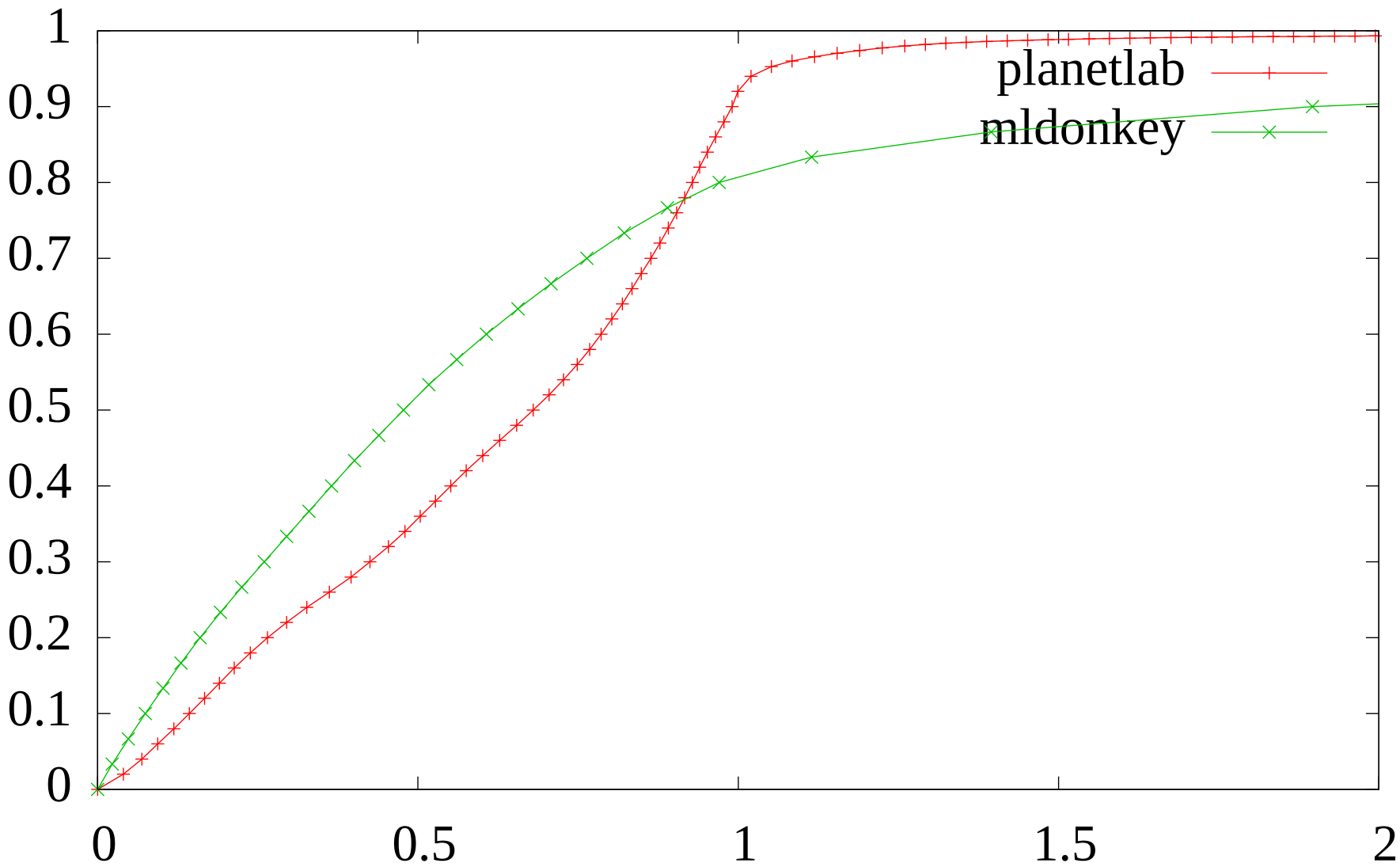
- Toute boule de rayon r peut-être couverte par 2^s boules de rayons $r/2$.
- Un espace euclidien de dimension d a une métrique $O(d)$-doubling.
- Si la métrique du graphe est s-doubling, alors $O(\varepsilon^{-O(s)} \log \Delta \log n)$ bits par nœud permettent de router avec facteur d'élongation $1+\varepsilon$ au plus (en-têtes de $O(s \log \frac{1}{\varepsilon} \log \Delta)$, Δ est l'« aspect ratio »).
- $O(\varepsilon^{-O(s)} \log^3 n)$ bits par nœuds, en-têtes de $2^{O(s)} \log^3 n$ [AGGM05].
- Espace métrique (graphe = clique) : $O(\varepsilon^{-O(s)} \log^2 n)$ bits par nœuds, en-têtes de $\log n$ [AGGM05].

RTTs dans Internet, *s*-doubling ?



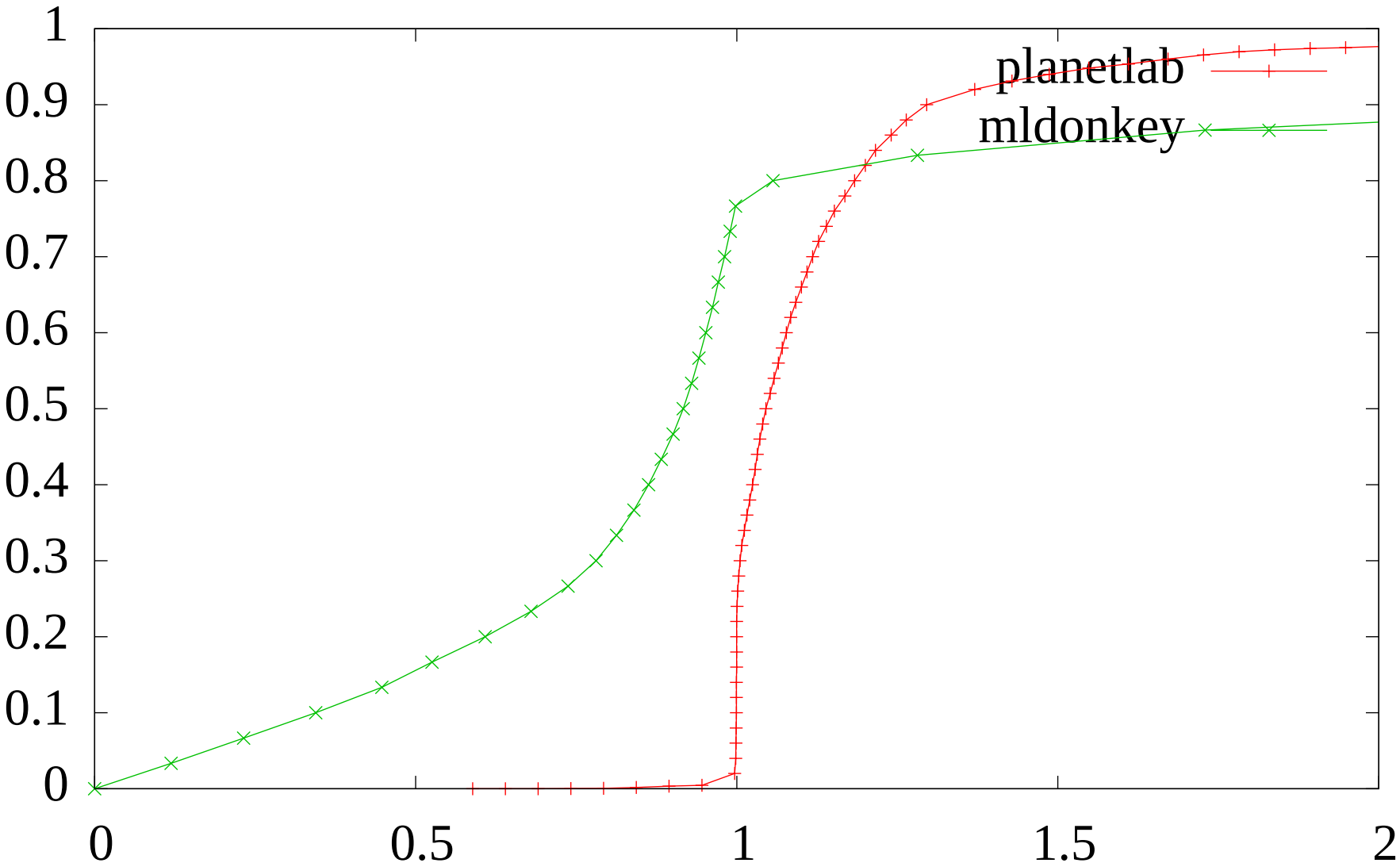
Distribution cumulée des *ab* pour quelques *a* pris au hasard dans Planetlab.

RTTs dans Internet, métrique ?



Distribution cumulée des $\frac{ac}{ab+bc}$ pour tout triangle a, b, c.

RTTs dans Internet



Distribution cumulée des $\frac{ac}{\min_b(ab+bc)}$ pour toute paire a, c.

Contexte du pair à pair

But

- Réseau logique plus rapide que le réseau physique ?

Contraintes

- Le graphe est une clique avec des poids sur les arêtes.
- Pas vraiment de limite de mémoire par nœud (100 000 adresses = 600 Ko).
- p pings par nœud par heure (1 ping = 1 mesure de RTT).

Contexte du pair à pair

But

- Réseau logique plus rapide que le réseau physique ?

Contraintes

- Le graphe est une clique avec des poids sur les arêtes.
- Pas vraiment de limite de mémoire par nœud (100 000 adresses = 600 Ko).
- p pings par nœud par heure (1 ping = 1 mesure de RTT).

Contexte du pair à pair

But

- Réseau logique plus rapide que le réseau physique ?

Contraintes

- Le graphe est une clique avec des poids sur les arêtes.
- Pas vraiment de limite de mémoire par nœud (100 000 adresses = 600 Ko).
- p pings par nœud par heure (1 ping = 1 mesure de RTT).

Réseau ad hoc et réseau de pair à pair

Similitudes

- Décentralisé sans infrastructure.
- Dynamique des connexions plutôt que de l'arrivée et des départs des nœuds.

Résultats

- Multicast par réseau logique [JR05].
- Réseau ad hoc hiérarchique au moyen d'une table de Hachage distribuée [FM05].
- Coopération : encouragement au relayage [FG05].

Éthique : encouragement à la coopération

Dilemme du téléchargeur peer-to-peer

Recevoir ses données avec le moins d'effort possible.
Matrice de satisfaction de A et B :

$A \setminus B$	B Donne	B Attend
A Donne	$0.9 \backslash 0.9$	$-0.1 \backslash 1$
A Attend	$1 \backslash -0.1$	$0 \backslash 0$

Étude du protocole BitTorrent

- Autres stratégies que le « prêté pour un rendu » [dM05].
- Modèle multi-joueur [F05].
- Utilisation du même principe pour encourager les nœuds d'un réseau ad hoc à relayer [FG05].

Éthique : encouragement à la coopération

Dilemme du téléchargeur peer-to-peer

Recevoir ses données avec le moins d'effort possible.

Matrice de satisfaction de A :

A	B Donne
A Donne	0.9
A Attend	1

Étude du protocole BitTorrent

- Autres stratégies que le « prêté pour un rendu » [dM05].
- Modèle multi-joueur [F05].
- Utilisation du même principe pour encourager les nœuds d'un réseau ad hoc à relayer [FG05].

Éthique : encouragement à la coopération

Dilemme du téléchargeur peer-to-peer

Recevoir ses données avec le moins d'effort possible.

Matrice de satisfaction de A :

A	B Attend
A Donne	-0.1
A Attend	0

Étude du protocole BitTorrent

- Autres stratégies que le « prêté pour un rendu » [dM05].
- Modèle multi-joueur [F05].
- Utilisation du même principe pour encourager les nœuds d'un réseau ad hoc à relayer [FG05].

Éthique : encouragement à la coopération

Dilemme du téléchargeur peer-to-peer

Recevoir ses données avec le moins d'effort possible.

Matrice de satisfaction de B :

B	B Donne	B Attend
A Donne	0.9	1
A Attend	-0.1	0

Étude du protocole BitTorrent

- Autres stratégies que le « prêté pour un rendu » [dM05].
- Modèle multi-joueur [F05].
- Utilisation du même principe pour encourager les nœuds d'un réseau ad hoc à relayer [FG05].

Éthique : encouragement à la coopération

Dilemme du téléchargeur peer-to-peer

Recevoir ses données avec le moins d'effort possible.
Matrice de satisfaction de A et B :

$A \setminus B$	B Donne	B Attend
A Donne	$0.9 \backslash 0.9$	$-0.1 \backslash 1$
A Attend	$1 \backslash -0.1$	$0 \backslash 0$

Heuristique dans le jeu itéré : donner à ceux qui ont donné au coup d'avant (« tit for tat » ou « prêté pour un rendu »).

Étude du protocole BitTorrent

- Autres stratégies que le « prêté pour un rendu » [dM05].
- Modèle multi-joueur [F05].
- Utilisation du même principe pour encourager les nœuds d'un réseau ad hoc à relayer [FG05].

Éthique : encouragement à la coopération

Dilemme du téléchargeur peer-to-peer

Recevoir ses données avec le moins d'effort possible.
Matrice de satisfaction de A et B :

$A \setminus B$	B Donne	B Attend
A Donne	$0.9 \backslash 0.9$	$-0.1 \backslash 1$
A Attend	$1 \backslash -0.1$	$0 \backslash 0$

Heuristique dans le jeu itéré : donner à ceux qui ont donné au coup d'avant (« tit for tat » ou « prêté pour un rendu »).

Étude du protocole BitTorrent

- Autres stratégies que le « prêté pour un rendu » [dM05].
- Modèle multi-joueur [F05].
- Utilisation du même principe pour encourager les nœuds d'un réseau ad hoc à relayer [FG05].