Fouille de graphes et réseaux sociaux

Raphaël Fournier-S'niehotta

CNAM Paris, fournier@cnam.fr

HTT-FOD
RCP216
2020-2021

le cnam

Plan

Introduction

Expérience de Milgram (1967)

Stanley Milgram (1933-1984), psychologue social américain. Connu notamment pour les expériences de soumission à l'autorité.

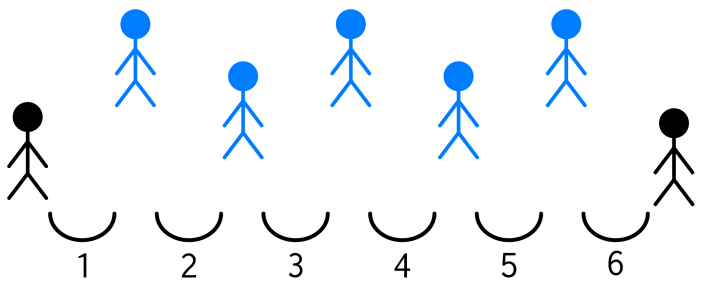
- Objectif de l'expérience : faire transiter une lettre de Omaha, NE à Boston, MA

Règle :

- une personne initie la chaîne
- transition de la main à la main à des personnes que l'on connaît, chacune étant supposée se rapprocher de la destination

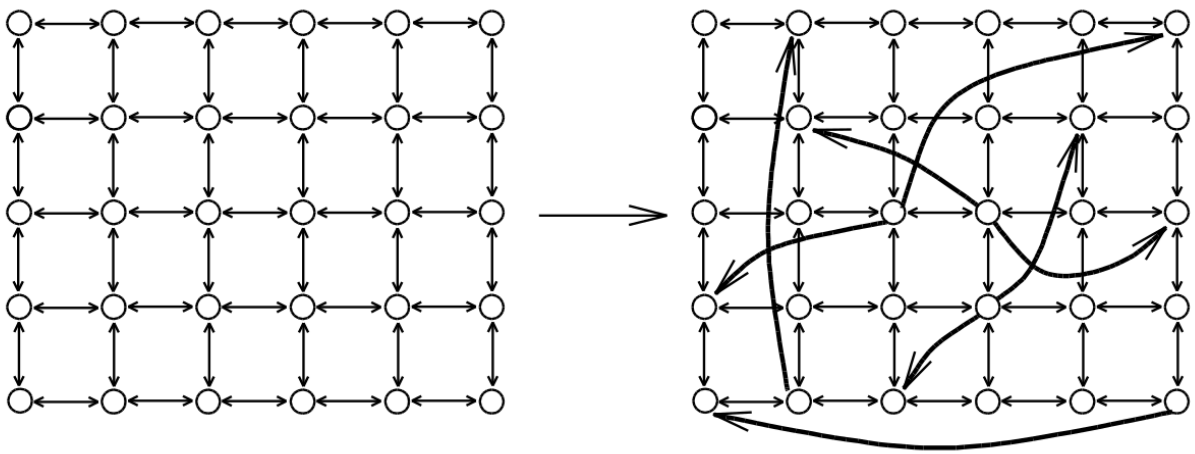
Expérience de Milgram (1967)

- Résultats
  - 64 lettres sur 296 arrivent
  - Chemins avec 5 intermédiaires en moyenne.
- Remarques :
  - Chemin interrompu $\neq$ Il n'existe pas de chemin.
  - Chemin de longueur x $\neq$ Il n'existe pas de chemin de longueur <x
- Conclusions :
  - Il existe des chemins courts.
  - Les intermédiaires arrivent à les trouver sans connaissance globale du réseau.



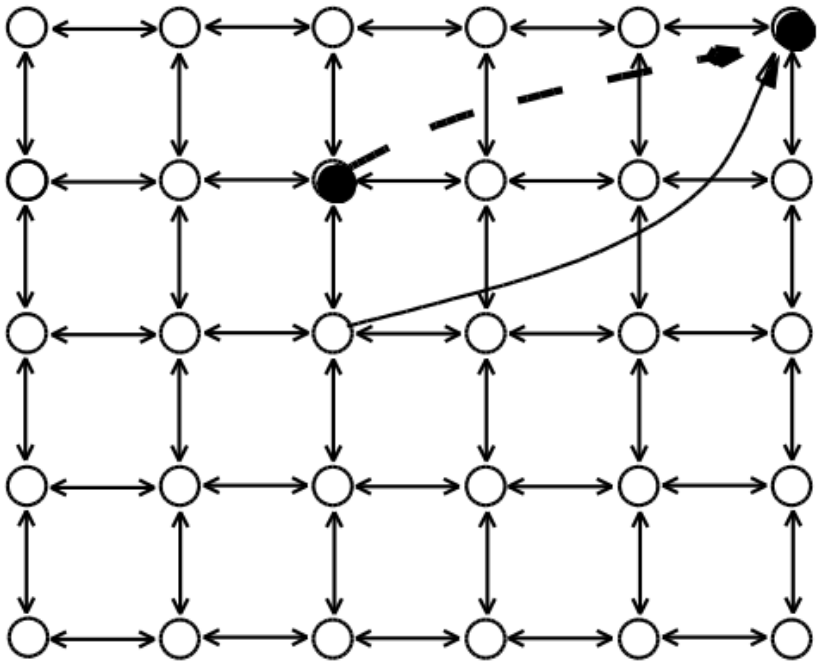
Expérience de Milgram : modélisation

- Objectif : formaliser l'expérience de Milgram
- Travail de D. Watts/S. Strogatz, puis de J. Kleinberg
- Initialement une grille (amis proches).
- On ajoute q voisins quelconques à chaque sommet (amis lointains).



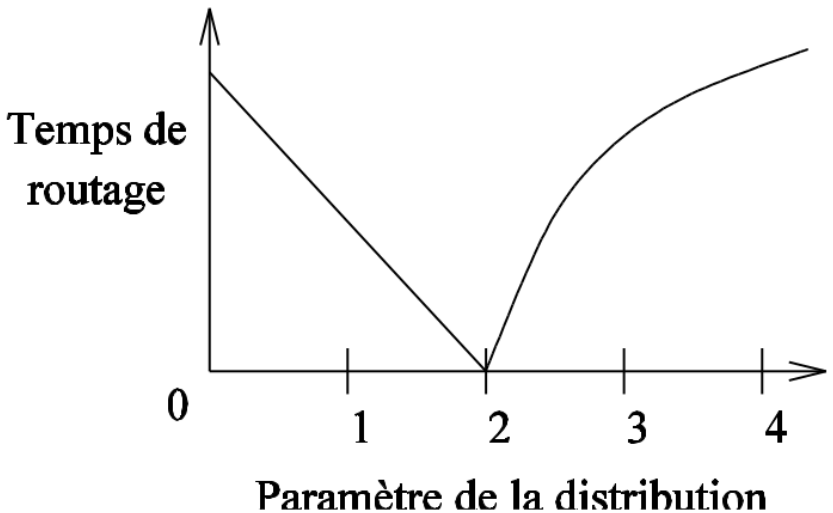
Expérience de Milgram : modélisation

- Un sommet connaît :
  - Sa position, celle de ses voisins, celle de la destination.
  - Il envoie le message à son voisin le plus proche de la destination.



Expérience de Milgram : modélisation

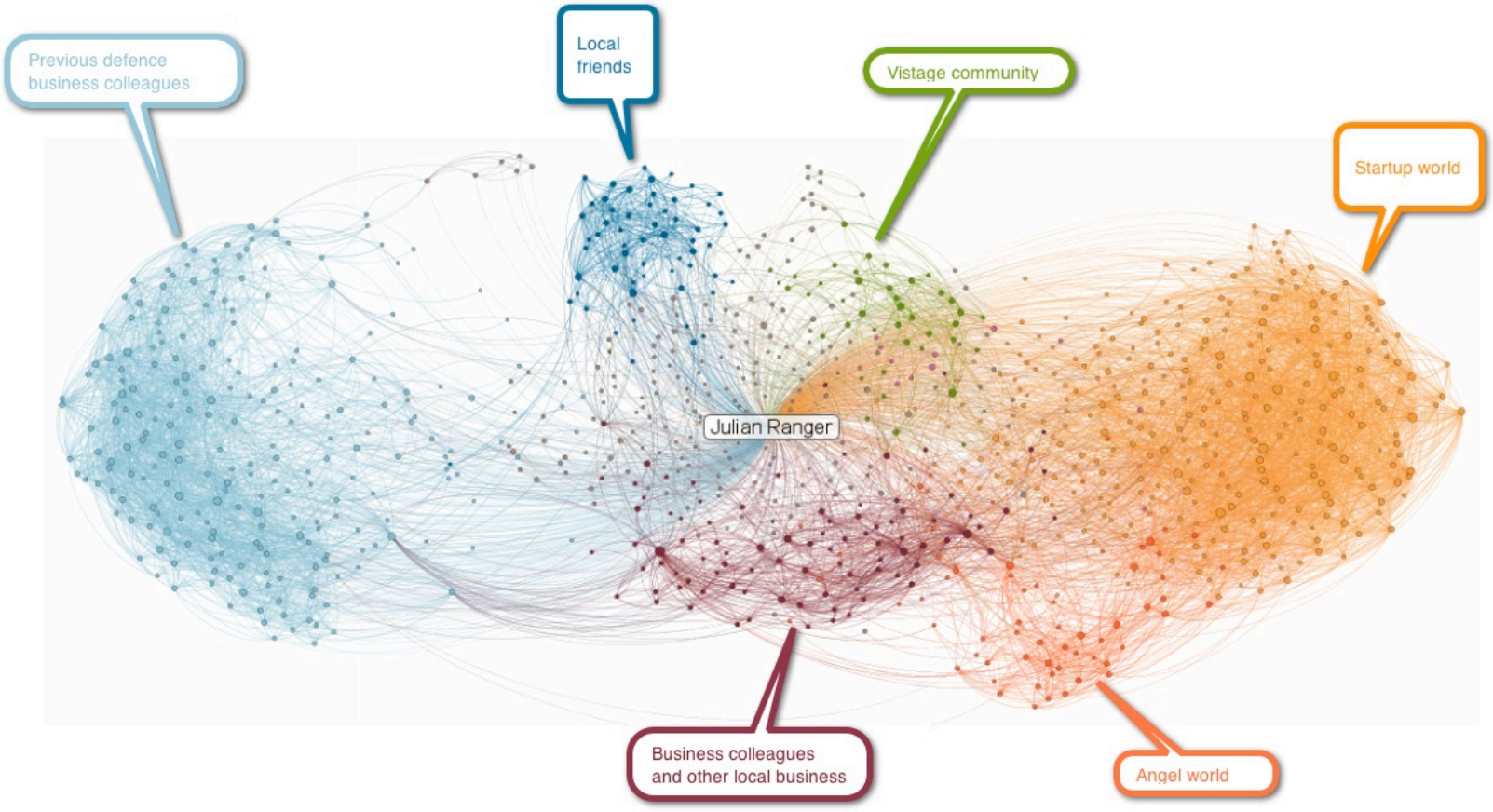
- Un seul lien supplémentaire pour chaque sommet u.
- La destination choisie avec une probabilité dépendant de sa distance à u.
- Dans la majorité des cas, pas de chemins courts
- **Mais :**



Petits mondes

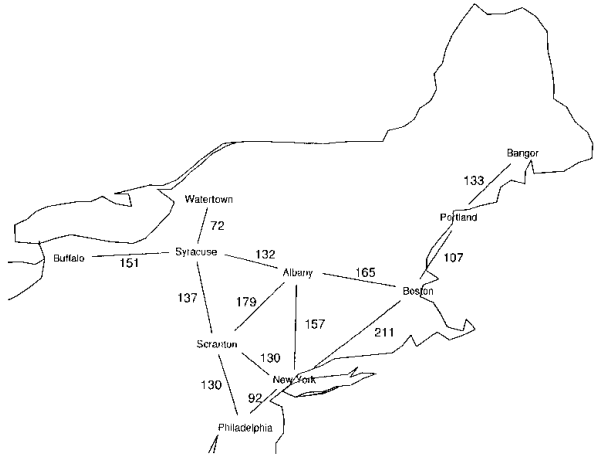
- On parle de “petit monde” si tous les individus d’un réseau peuvent être connectés par un chemin court
- Des chercheurs ont mis en évidence ce phénomène dans des réseaux de domaines très variés (neurones de C. elegans, électricité aux USA)
- modèle
  - structure sous-jacente traduisant les liens entre la plupart des nœuds
  - quelques liens aléatoires expliquent le diamètre faible
  - spécialisation régionale avec transfert efficace d’information entre zones

Réseau personnel : LinkedIn Maps



Réseau routier national

- Noeuds : les villes / communes
- Arêtes : (auto)routes
- Valuation possible : distance, ou temps de parcours
- Orientation possible



Des questions :

- quel est le plus court chemin passant par des villes données ?
- quel chemin traverse le moins de villes pour aller d'un point à un autre ?
- peut-on passer par toutes les villes sans passer deux fois par la même route ? (*voyageur de commerce*)

Individus : Kevin Bacon Game

Kevin Bacon (1958–), acteur américain, qui a joué dans plus de 75 films.

- Graphe d'acteurs
  - Deux acteurs sont reliés s'ils ont joué dans un même film.
  - Distance entre acteurs ?
    - http://oracleofbacon.org/
    - Distance entre Tom Cruise et Clint Eastwood ? 2 (acteur commun entre Space Cowboys et Eyes Wide Shut)
    - Distance entre Mickey Mouse et Omar Sy ? 4
- graphes produits à partir de http://www.imdb.com/interfaces
- calculs de plus courts chemins

Individus : possesseurs de fichiers P2P

- Propagation d'un fichier d'utilisateurs en utilisateurs
  - ♠ video
- Problèmes et biais de mesure
  - dynamicité du réseau
  - parcours non exhaustif et depuis une source

Autres types de réseaux étudiés

- informatique : pages Web, routeurs, P2P, etc.
- biologie : protéines, neurones cérébraux, etc.
- sciences sociales : amitiés, collaboration, contacts sexuels, etc.
- économie : échanges financiers
- histoire : mariages
- linguistique : synonymie, co-occurrence
- transports : réseau aérien, électrique

Propriétés et problématiques communes

Définitions

Un graphe est défini par un couple G = (V,E) tel que :

- V (pour l'anglais *vertices*) est un ensemble fini de sommets
- E (pour l'anglais *edges*) est un ensemble fini d'arêtes

Un graphe peut être orienté, ou non :

- si oui, les couples $(v_i, v_j) \in E$ sont ordonnés, v_i est le sommet initial, v_j est le sommet terminal.
- on appelle alors le couple (v_i, v_j) un *arc*, représenté graphiquement par $v_i \rightarrow v_j$.
- si non, les couples ne sont pas orientés et (v_i, v_j) est équivalent à (v_j, v_i), et on l'appelle *arête*, représenté par $v_i - v_j$

Terminologie

- l'**ordre** d'un graphe, c'est son nombre de sommets (souvent désigné par n).
- une **boucle** est un arc/une arête reliant un sommet à lui-même
- un graphe dépourvu de boucle est dit **élémentaire**
- un graphe **simple** ne comporte pas de boucle et au plus une arête entre deux sommets
- un graphe **partiel** est le graphe obtenu en supprimant certains arcs ou arêtes
- un **sous-graphe** est le graphe obtenu en supprimant certains sommets et tous les arcs/arêtes incidents aux sommets supprimés.
- un sommet v_i est dit **adjacent** à un autre s'il existe une arête entre eux (on parle de **voisins**).
- le **degré** d'un sommet est le nombre d'arêtes incidentes à ce sommet.
- un graphe est dit **complet** s'il comporte une arête (v_i, v_j) pour toute

Objectifs

Comprendre le comportement des entités
qui interagissent dans le système étudié, et les lois qui les gouvernent

- On cherche donc :
  - quelle est la structure des graphes
  - quelle est l'évolution de cette structure
  - quels sont les phénomènes reposant sur l'existence de ce réseau

Graphes et fouille de données

d'un point de vue Data Mining, un réseau social (graphe), c'est :

- un jeu de données souvent très hétérogène
- multirelationnel et de grande taille
- les noeuds (sommets) sont les objets
- les arêtes sont les relations
- nœuds et arêtes peuvent avoir des attributs, rendant complexe l'analyse

Applications

- Informatique
  - Réseaux : routage, protocoles, sécurité
  - P2P : conception de systèmes, déviances
  - Web : indexation, moteurs de recherche
  - Dessin de graphes
- Sociologie :
  - Diffusion d'innovations, rumeurs
  - Identification de communautés
- Épidémiologie
  - Diffusion de virus, vaccination

Avantages

- Identification du rôle des individus :
  - Leader
  - Suiveur
  - Intermédiaire
- Identification de l'importance d'un groupe en analysant :
  - la taille
  - la cohésion
  - les profils
  - les relations internes et externes
- Repérer les doublons (même réseau)

Méthodologie

- Outils formels
  - Théorie des graphes
  - Analyse statistique
  - Modélisation probabiliste
- Études expérimentales
  - Simulation
  - Utilisation de données réelles
- Étudier des applications
  - Comprendre en profondeur certains réseaux
  - Extraction de concepts généraux

Ce cours

- Problématiques classées dans 4 grandes catégories :
  - Mesure
    - Comment mesurer les réseaux réels ?
  - Modélisation
    - A quoi ressemblent-ils ?
    - Peut-on créer des réseaux artificiels similaires ?
  - Analyse
    - quelles sont leur propriétés ?
  - Algorithmique
    - Comment calculer sur de grands graphes ?
- Détection de communautés (clustering)
- Réputation, prédiction, innovations et leaders

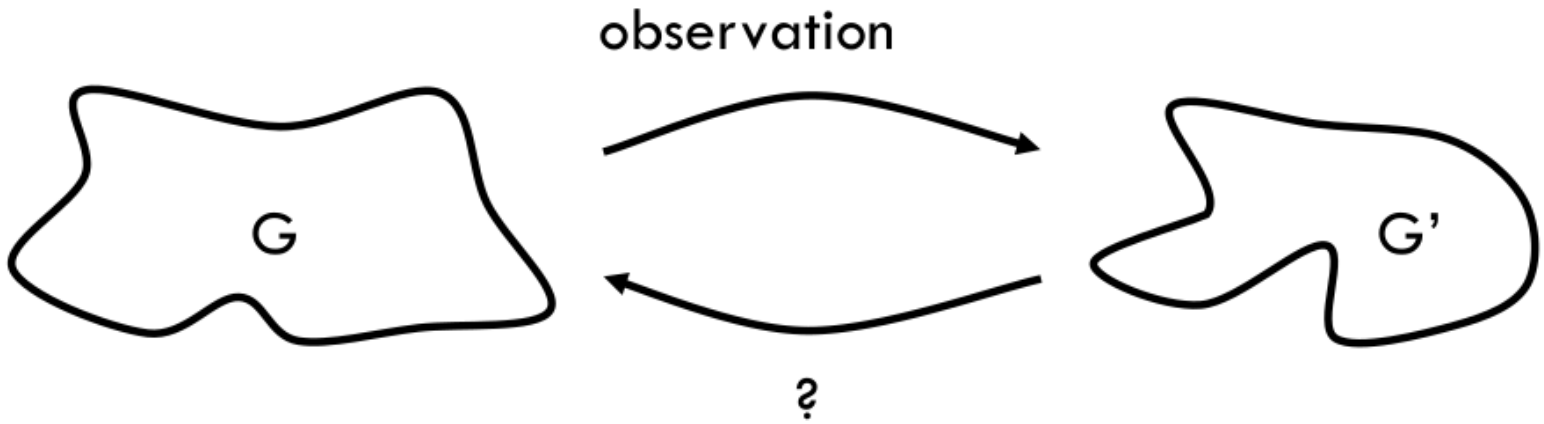
Mesure

Métrologie

- En général : impossibilité d'étudier l'objet réel, seulement une mesure
- Questions :
  - qui a fait la mesure ?
  - quelle proportion a été mesurée ?
  - combien de temps la mesure a-t-elle duré ?
  - quelles étaient les contraintes / biais ?
  - la mesure peut-elle être reproduite ?

Métrologie

- Étude du biais introduit par l'observation
- Que dire de l'objet réel à partir de l'observation ?
- Nouveaux protocoles de mesures, etc.



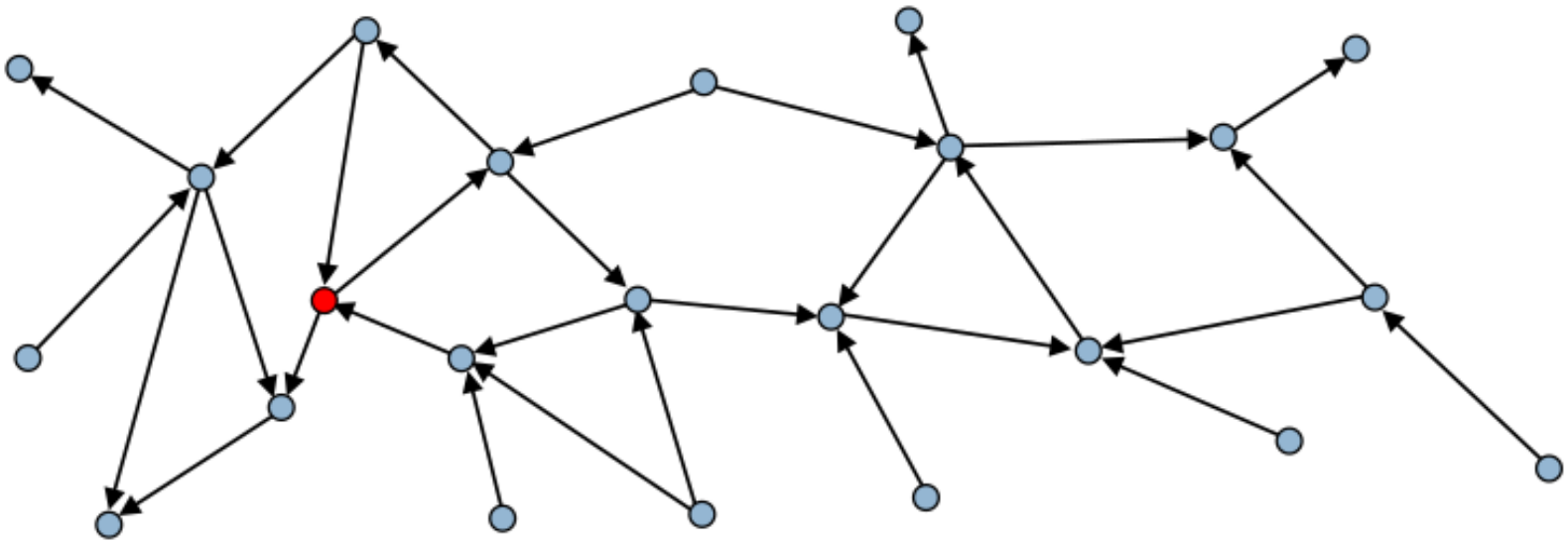
- Évaluer la représentativité des "cartes"

Une approche

- On simule la mesure sur un graphe aléatoire
- Modélisation du processus de mesure :
  - Internet : traceroute = chemins courts
  - Web : crawl = parcours en largeur
- Modélisation du réseau :
  - Graphes aléatoires
  - Respect des degrés, du clustering, etc.

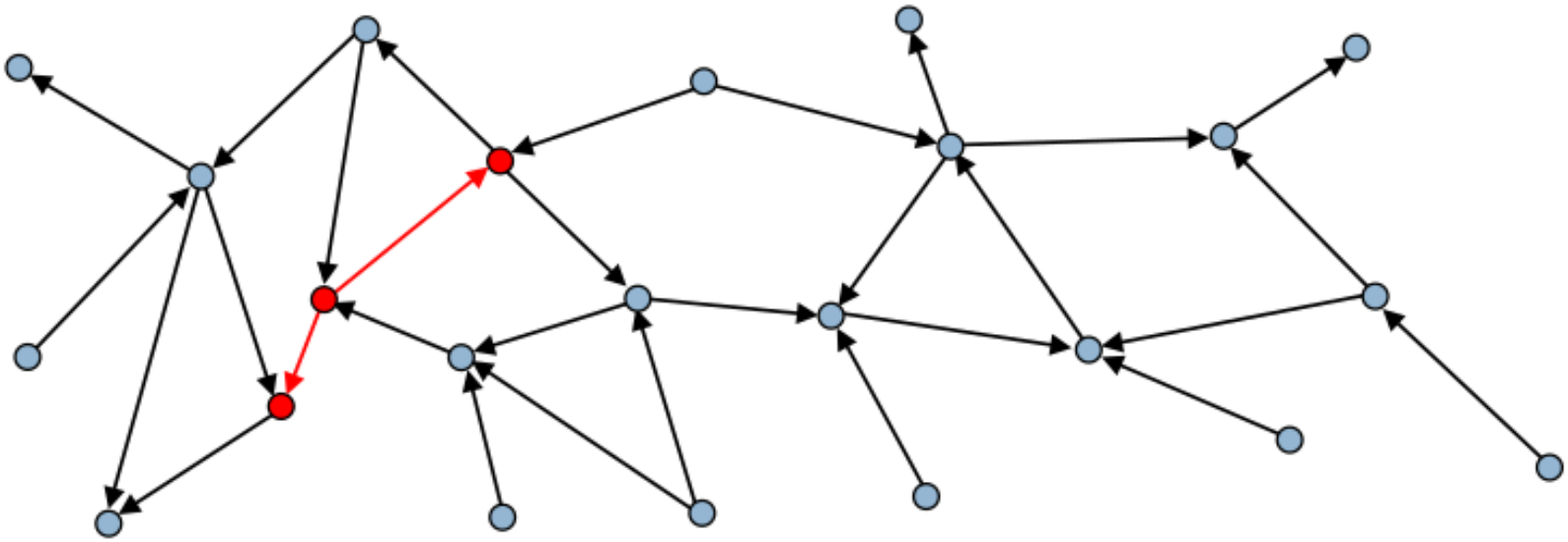
Mesure de l'Internet

- Processus de mesure par parcours en largeur depuis plusieurs sources
- Réseau : orienté, non connexe, dynamique



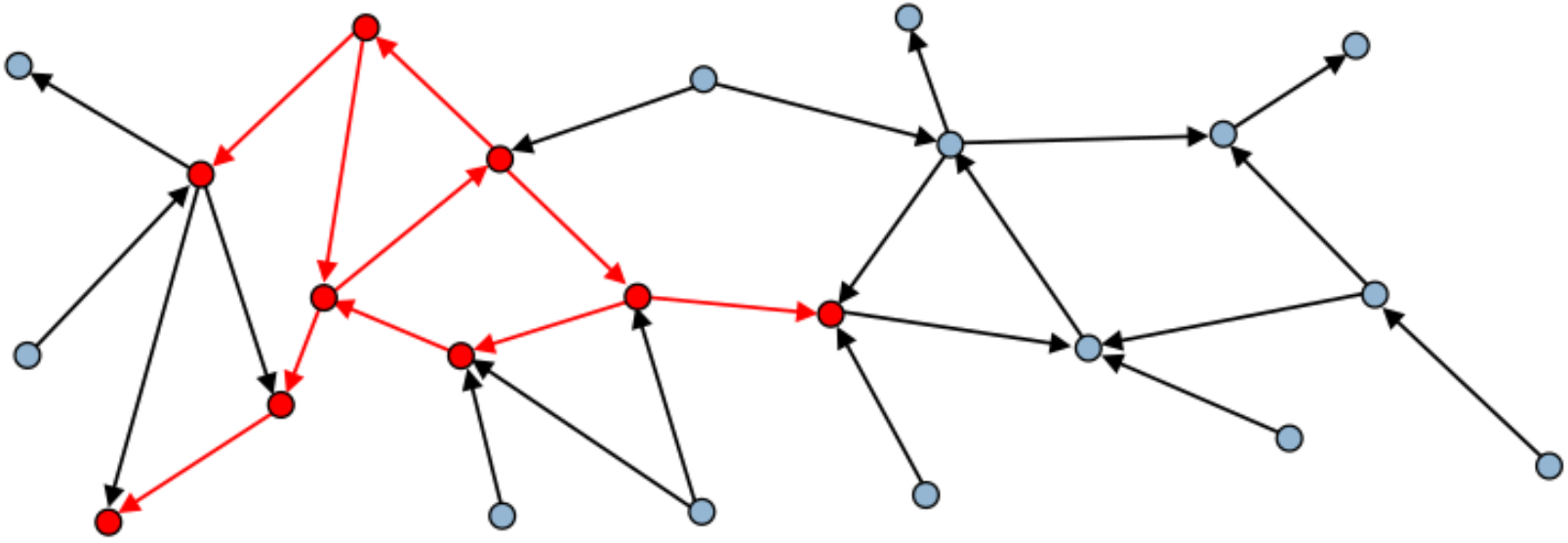
Mesure de l'Internet

- Processus de mesure par parcours en largeur depuis plusieurs sources
- Réseau : orienté, non connexe, dynamique



Mesure de l'Internet

- Processus de mesure par parcours en largeur depuis plusieurs sources
- Réseau : orienté, non connexe, dynamique

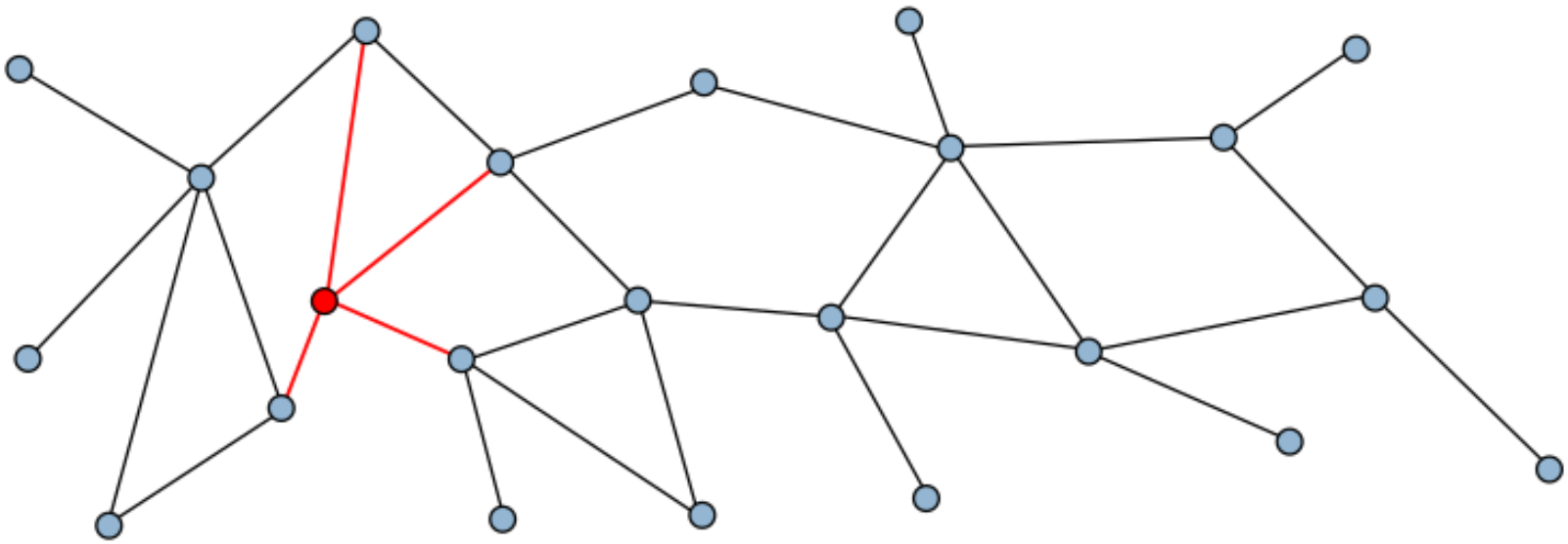


Mesure de réseaux sociaux

Processus de mesure :

- Réseau égocentrés
- Listes de diffusion, communautés

Réseau : orienté, non connexe, dynamique

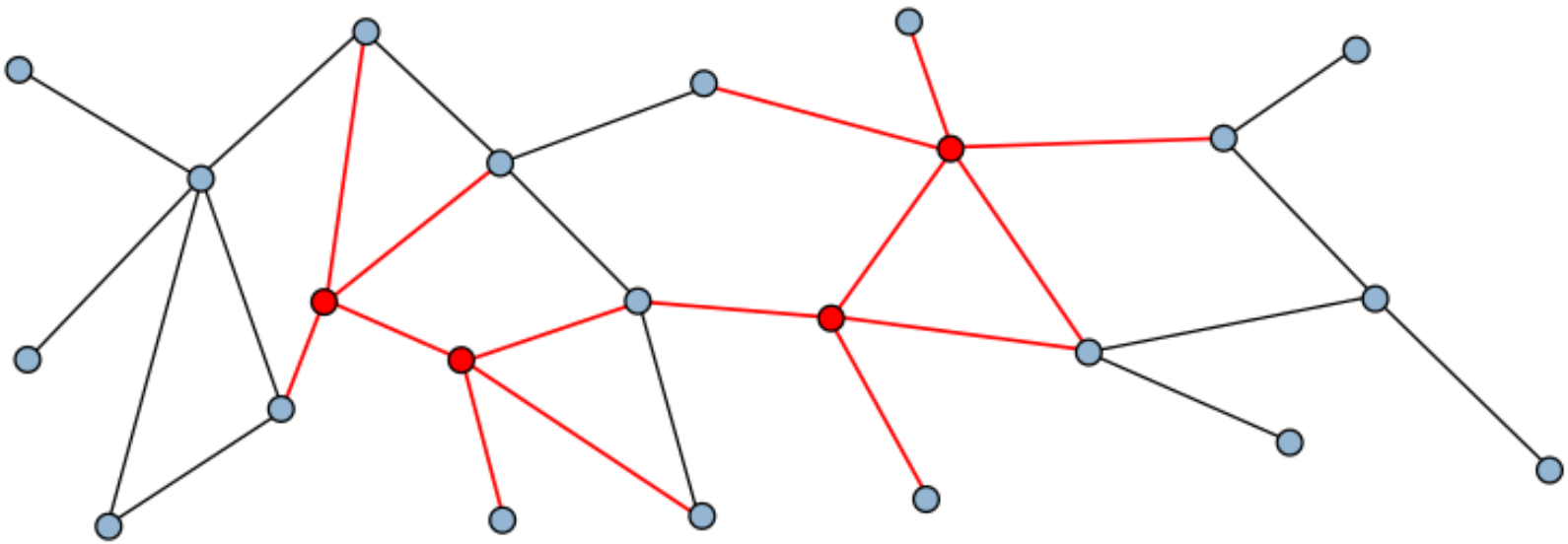


Mesure de réseaux sociaux

Processus de mesure :

- Réseau égocentrés
- Listes de diffusion, communautés

Réseau : orienté, non connexe, dynamique

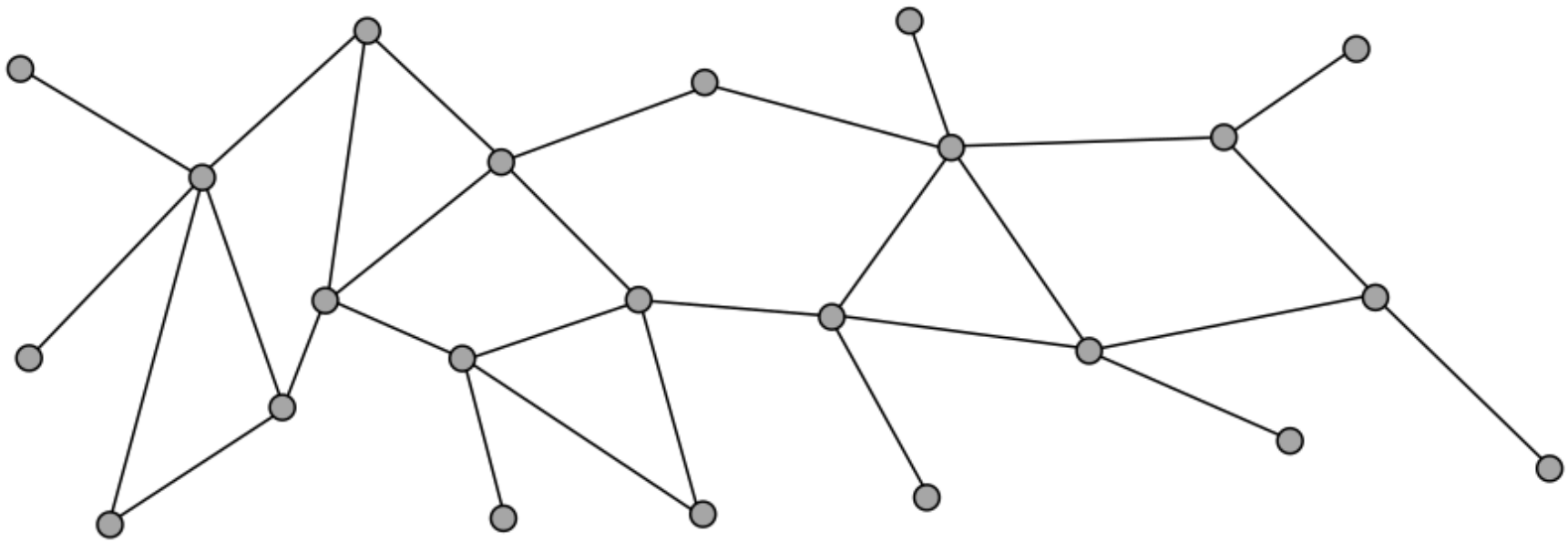


Questions

- Influence sur le résultat de :
  - Nombre de sources et destinations
  - Propriétés du réseau
  - Localisation des sources et destinations
- Modélisation :
  - Traceroute = plus courts chemins (un ou tous)
  - Graphe = graphe aléatoire (modèle à choisir)

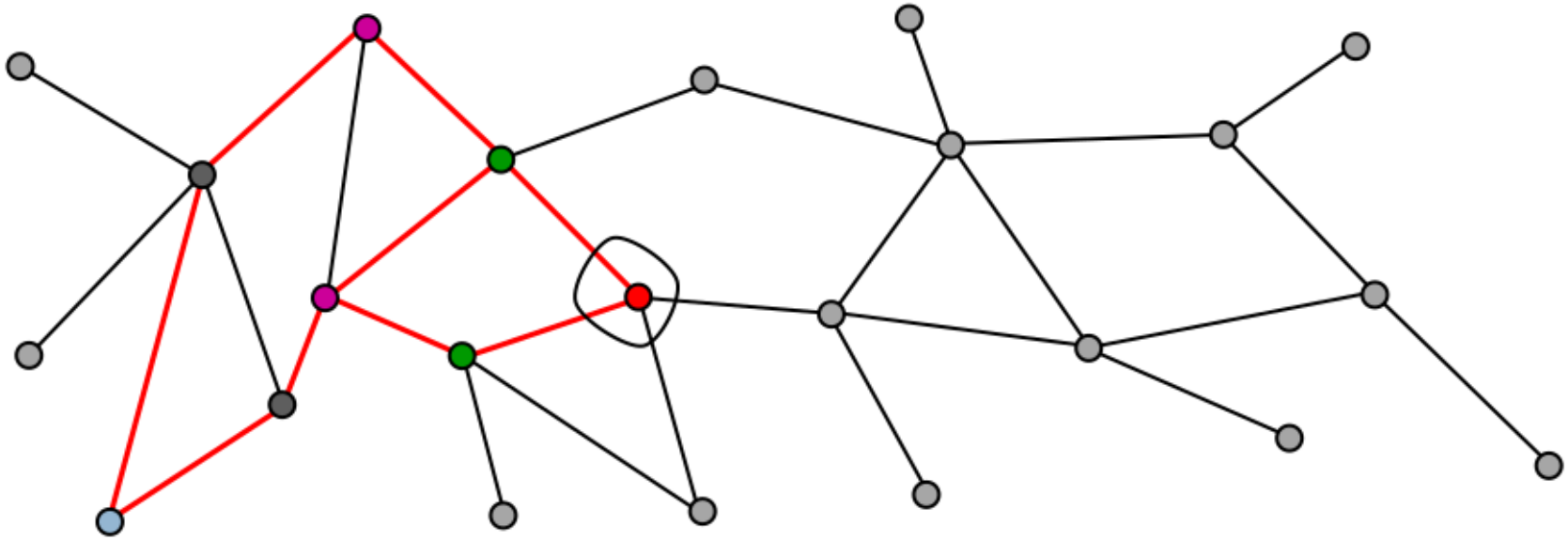
Que voit-on ?

- D'une source vers tout le monde
  - liens rouges découverts (sur plus courts chemins)
  - on répète pour les autres destinations
  - liens noirs invisibles



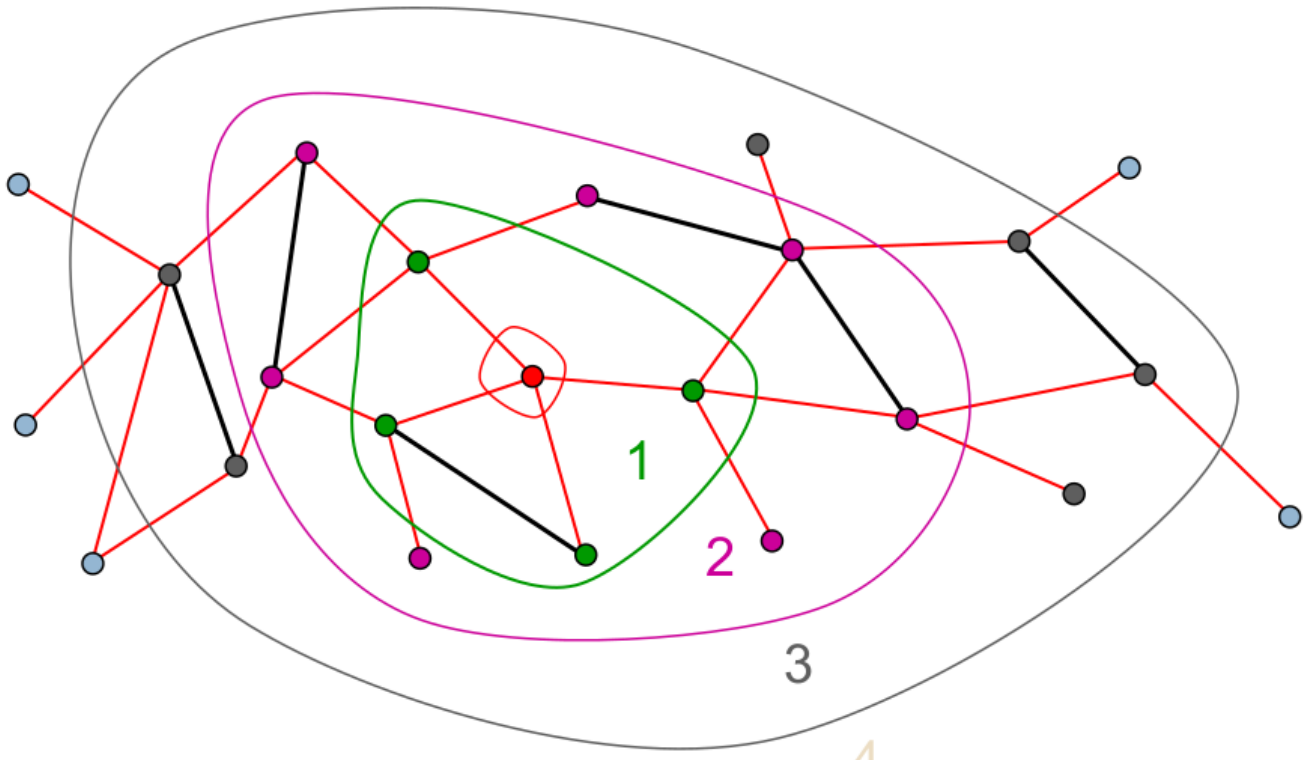
Que voit-on ?

- D'une source vers tout le monde
  - liens rouges découverts (sur plus courts chemins)
  - on répète pour les autres destinations
  - liens noirs invisibles



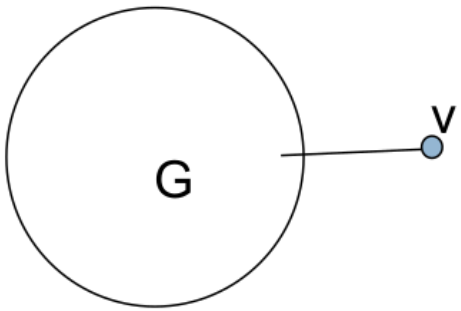
Que voit-on ?

- D'une source vers tout le monde
  - liens rouges découverts (sur plus courts chemins)
  - on répète pour les autres destinations
  - liens noirs invisibles

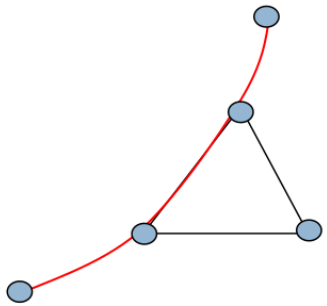


Zones dures à mesurer

- Sommet de degré 1 : uniquement visible si source ou destination

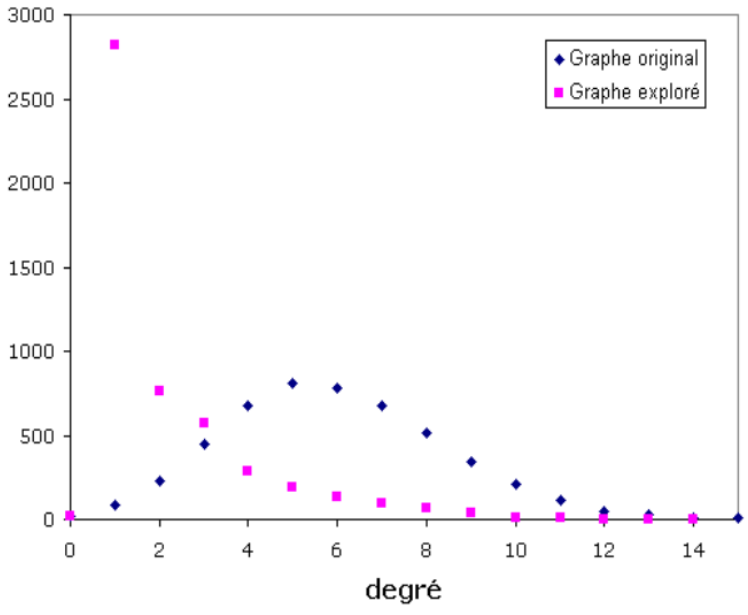


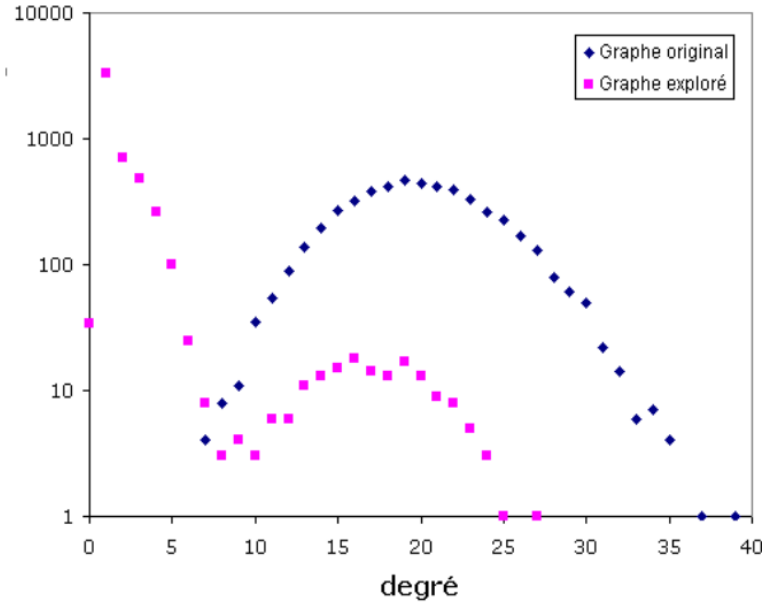
- graphe complet : visiter tous les liens



Distribution de degrés

- différences entre original et mesuré
  - beaucoup de sommets de faible degré
  - peu de sommets de fort degré
- mauvaise estimation de la propriété réelle





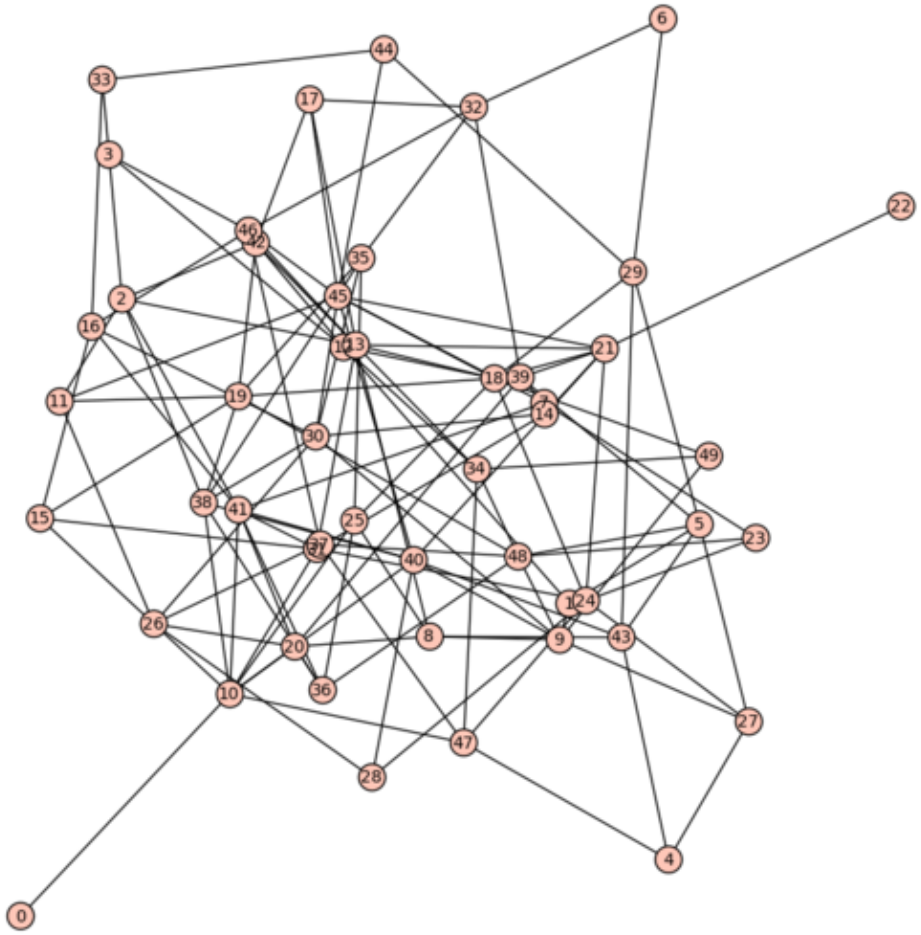
Modélisation

Générer des graphes réalistes

- Est-ce que les propriétés observées sur les graphes réels sont “normales”
  - On peut comparer avec un graphe aléatoire ayant certaines propriétés
- Simulation de phénomènes (attaques, diffusion, etc.)
- Évaluation de protocoles
- Compréhension
- Prévision

Tout aléatoire

- Créer n sommets/nœuds
- Ajouter au hasard m liens ($m <= n^2$)



Propriété attendue

- Graphe aléatoire, $n = m = 4950$
- Graphe réel : clique de 100 sommets, autres nœuds de degré 0
- Probable ?
  - proba degré 0 : $p = (1 - \frac{2}{n})^n \sim 0.14$
  - on attend donc : $n \times p \sim 683$ sommets de degré 0
  - graphe réel peu probable

Propriétés observées

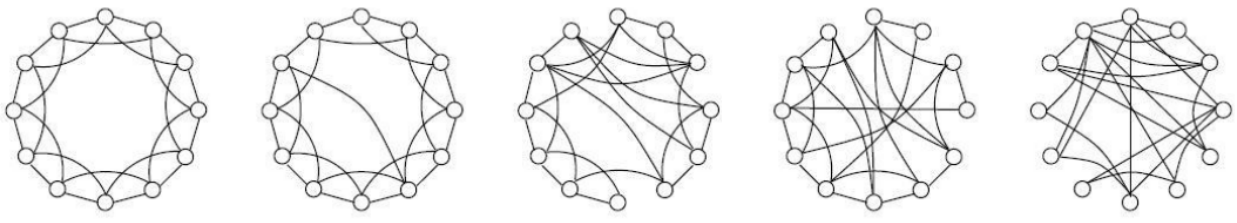
- densité fixée
- Connexité : composante géante de taille $O(n)$
- Distance moyenne, diamètre $\sim$ log(n)
- Distribution des degrés homogène
- Clustering proche de 0
- Pas de structure communautaire

Basé sur la distribution de degrés

- Attachement préférentiel
  - ajout de sommets un à un
  - ajout de lien à des sommets déjà connectés
- Modèle configurationnel (*configuration model*)
  - on prend n sommets
  - on fixe le degré de chaque sommet
  - on ajoute des liens au hasard en respectant les degrés
- ne génèrent pas de clustering

Basé sur le Coefficient de clustering

- Mélanger un graphe très rigide :
  - Donne du clustering et une distance moyenne courte
  - Ne donne pas de degrés hétérogènes !

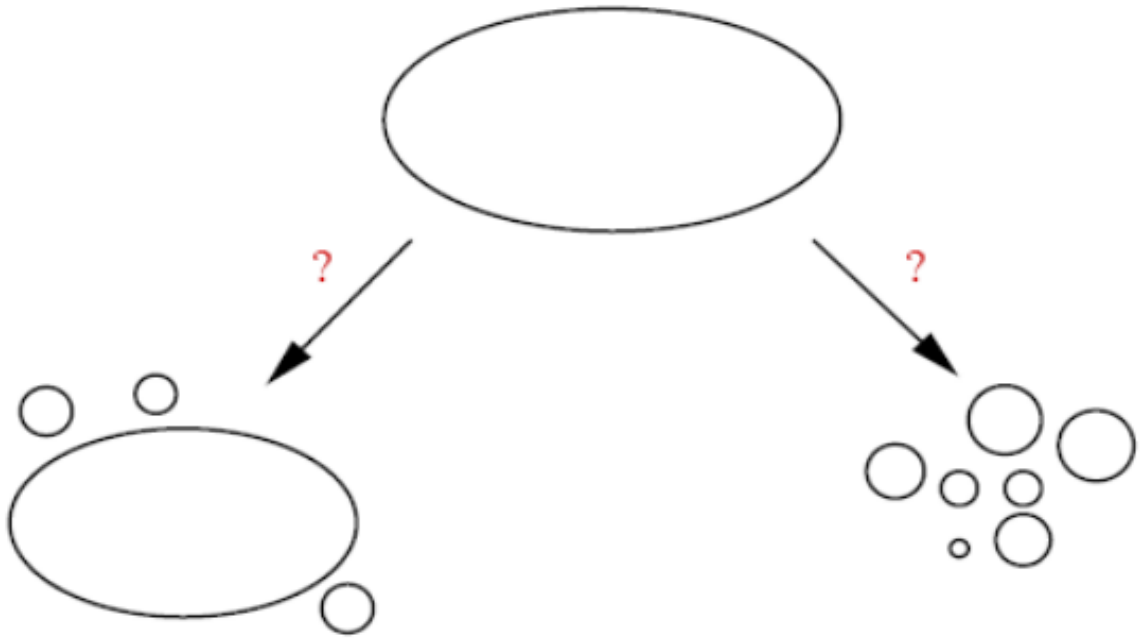


Application : robustesse

- Étude des phénomènes visant des sommets :
  - Internet : pannes ou attaques sur routeurs.
  - Réseaux sociaux : maladies, rumeurs, ...
  - Échanges d'e-mails : virus informatiques.
- Deux types d'atteintes
  - Pannes : aléatoires.
  - Attaques : ciblées.
- But : Comprendre ces phénomènes pour pouvoir :
  - Prédire.
  - Construire des stratégies d'attaque/défense.

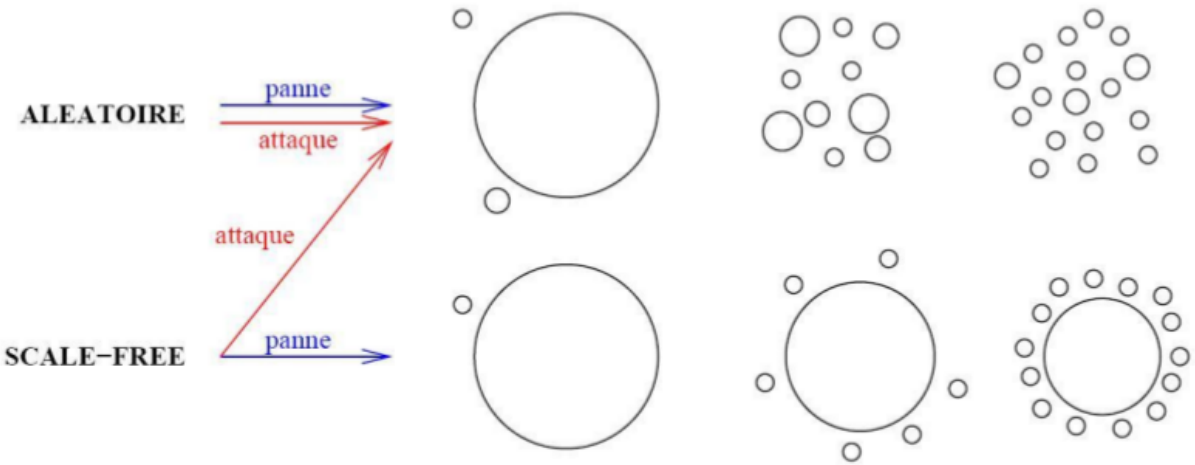
Impact d'une panne/attaque

- Critères :
  - Basés sur la distance.
  - Tailles des composantes connexes.
  - etc.



Résultats

- Suppression :
  - Panne = aléatoire
  - Attaque = ciblée (plus fort degré d'abord)
- Question : qui vacciner pour limiter une épidémie ?



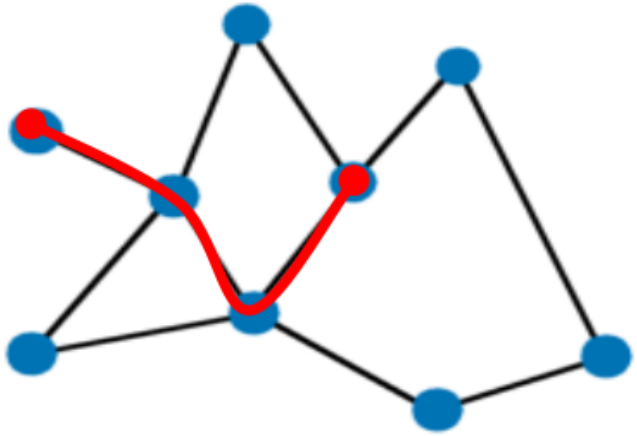
Analyse

Analyse ?

- Objectifs de l'analyse (statistique) :
  - Description (statistique)
  - Obtenir de l'information pertinente
  - Interprétation des résultats obtenus
- Comment ?
  - Propriétés connues
  - Définition de propriétés (statistiques) pertinentes
  - Corrélations entre ces propriétés
  - Comparaison avec des graphes aléatoires
  - Observation de la croissance des graphes, etc.

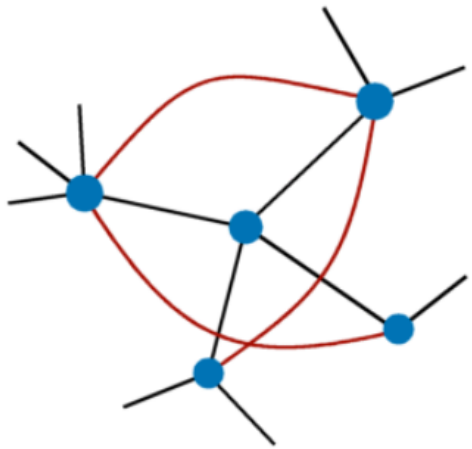
Propriétés classiques

- Longueur des chemins : distance moyenne



Propriétés classiques

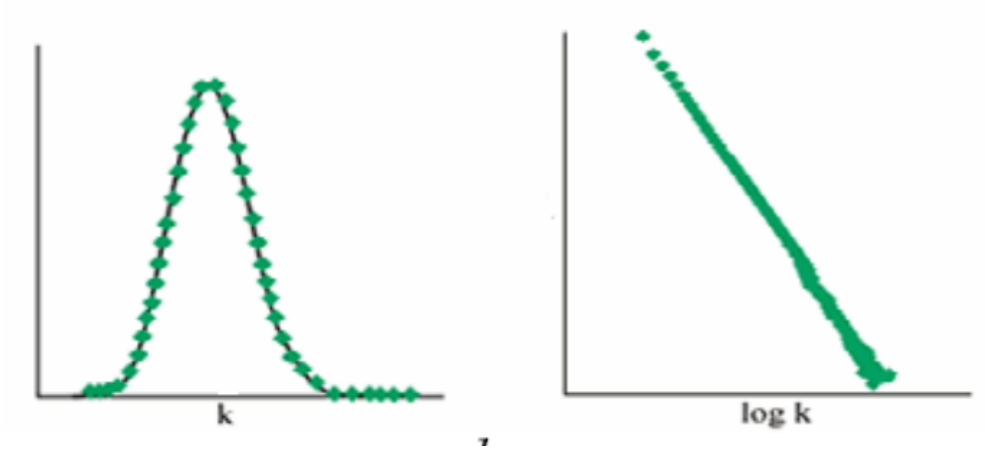
- Clustering
  - densité de liens autour d'un nœud
  - comparé à la densité globale



$$c(i) = \frac{2*|(x,y) \in E, x,y \in N(i)|}{k_i(k_i-1)} \text{ (ou 0 si } d(i) < 2))$$

Propriétés classiques

- Distribution de degrés
  - Taille ou salaire des individus



$$P_d \sim e^{-\lambda} \cdot \frac{\lambda^d}{d!}$$

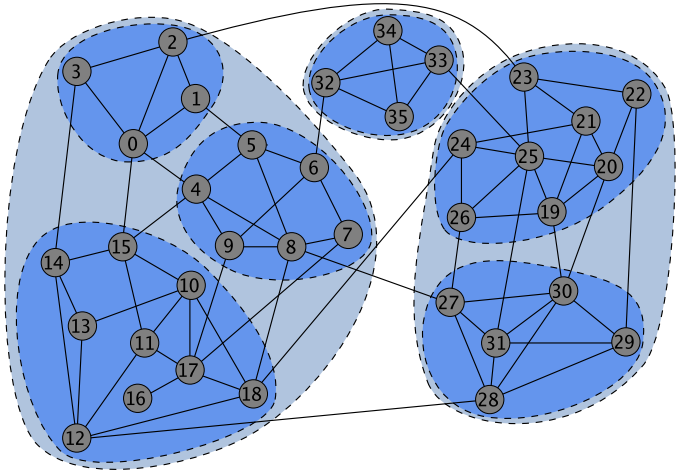
$$P_d \sim d^{-\alpha}$$

Propriétés classiques

- Composantes connexes
  - Ensemble maximal de sommets tel qu'il existe un chemin entre toute paire de sommets de l'ensemble
  - Graphe connexe = une seule composante connexe

Propriétés classiques

- Communautés
  - ensemble de nœuds très densément liés
  - peu de connexion en dehors de l'ensemble



Propriétés classiques

- Autres propriétés
  - Centralité
    - degré
    - intermédiarité : Nombre de plus courts chemins passant par un sommet. $C(v) = \sum_{s \neq t \neq v \in V} \frac{s_{st}(v)}{s_{st}}$
    - proximité : Inverse de la distance à tous les autres sommets $C(x) = \frac{1}{\sum_y d(y,x)}$
  - Taille des cliques

Propriétés des réseaux réels

- faible densité
- fort clustering
- faible distance moyenne
- distribution de degré fortement hétérogène
- composante géante
- présence de communautés

propriétés différentes de celles des graphes aléatoires

Exemple d'analyse : réseau de contacts

- Nombreux équipements avec capacités sans-fil :
  - Ordinateurs, téléphones, PDA, GPS, cartes Navigo...
  - Réseaux sans-fils de plus en plus omniprésents
- Contacts physiques ou virtuels permanents :
  - Rencontres physiques, appels téléphoniques, envoi de mails...
- Objectifs :
  - Tirer parti des contacts naturels des individus
  - Transmission de l'information de proche en proche
  - Réseau dynamique, non connexe : problèmes de routage ...

Proximité physique ou radio

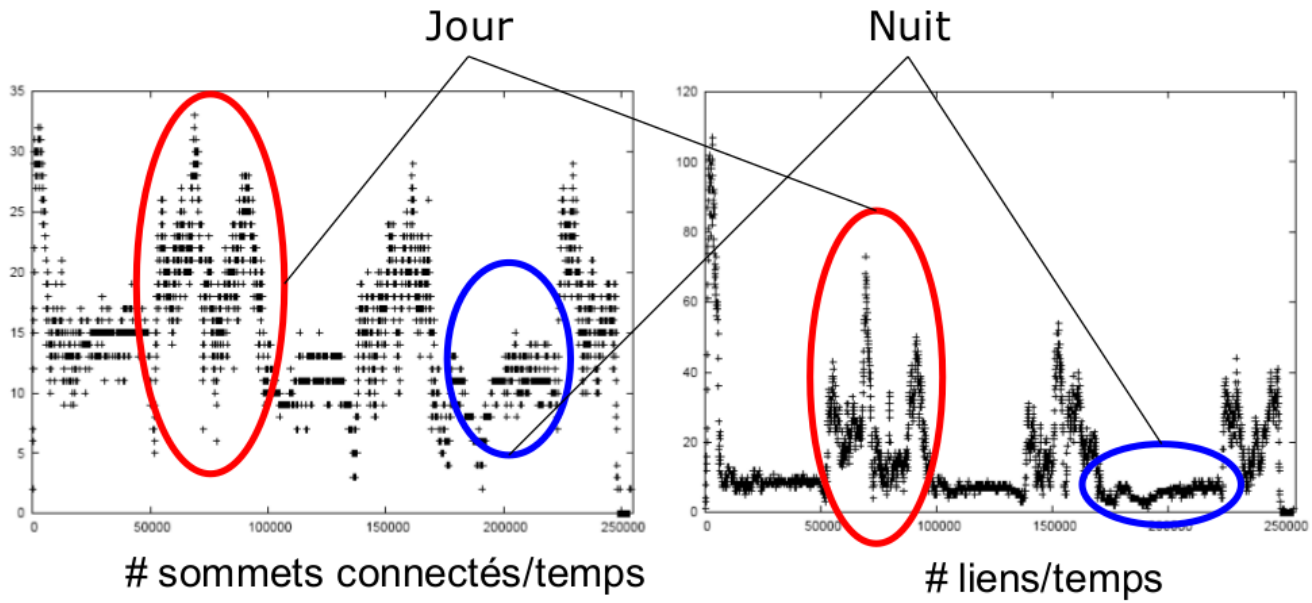
- Quels contacts entre individus ?
  - Physique
  - proximité géographique
  - déplacements
- mesure de la mobilité
  - suivi de déplacements
  - géolocalisation : couteux, dur à mettre en œuvre
  - équipement de chaque individu
- application informatique/télécom : déploiement de réseau dans des environnements “hostiles” (zones militaire, forêts)

Étude de cas

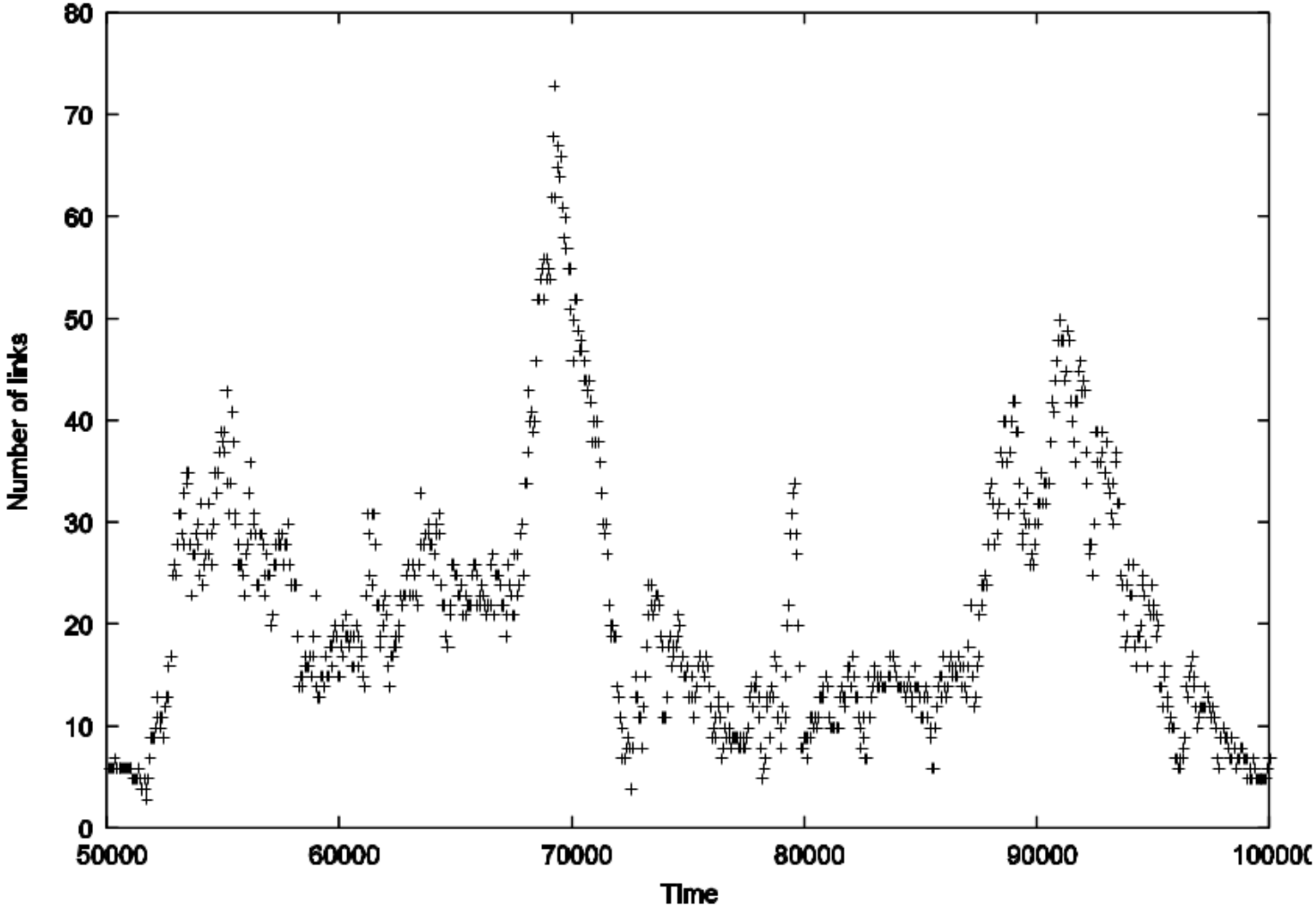
- Conférence INFOCOM 2005, dans un hôtel à Miami (USA)
- 3 jours
- 54 capteurs Bluetooth initialement (perte, pannes, 41 après)
- Fonctions :
  - recherche de contact (5s)
  - attente (110s env)
  - pas de géolocalisation
- données
  - ensemble de liens à chaque instant
  - liens non symétriques
  - `http://plausible.lip6.fr`

Étude de cas

- Effets sociologiques :
  - jour/nuit, repas, pauses, etc.
  - beaucoup de petites variations
  - 50% de sommets isolés
  - max 34 sommets connectés

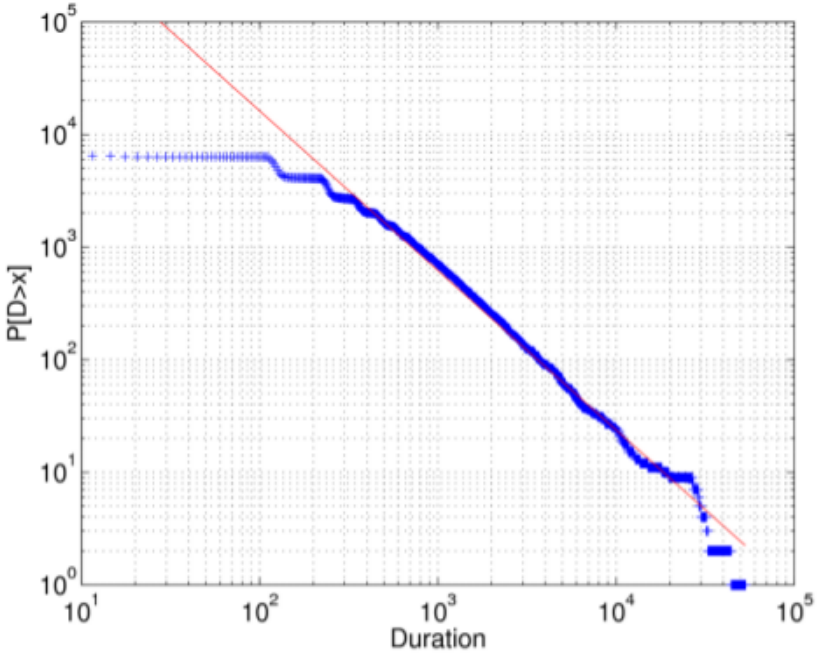


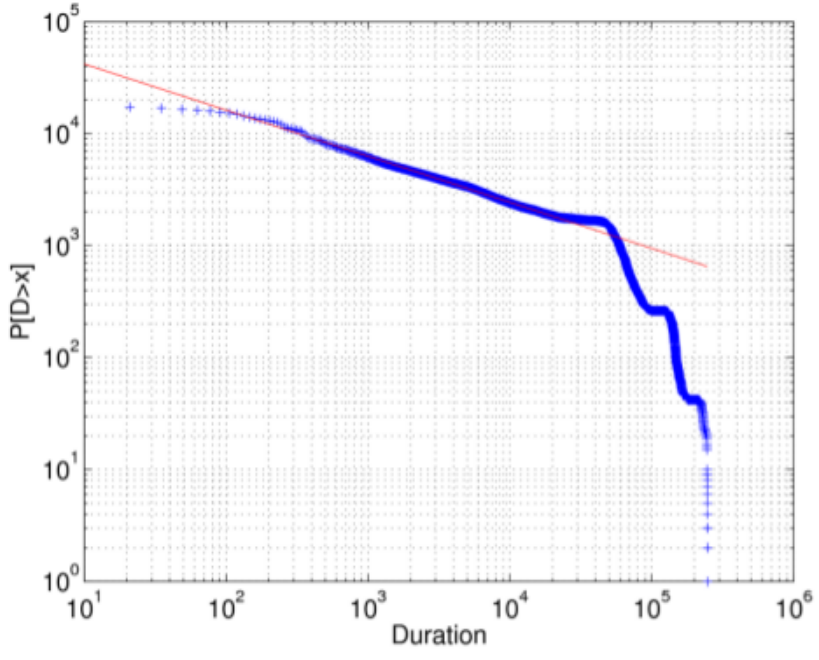
Étude de cas


Number of links
Time
0
10
20
30
40
50
60
70
80
50000
60000
70000
80000
90000
100000

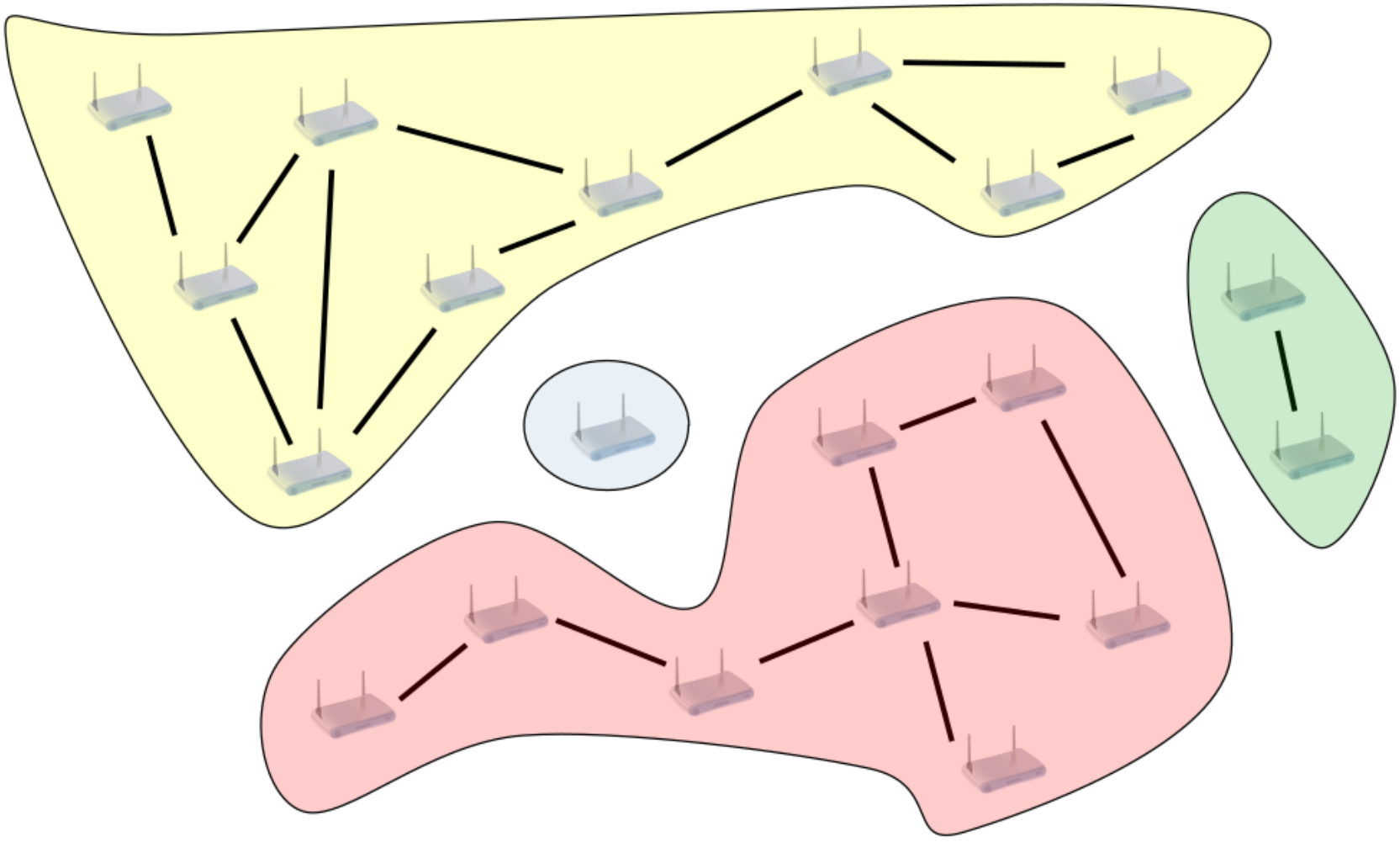
Durée de contacts

- distribution en loi de puissance
- certains liens sont fréquents, d'autres pas
- liens non fréquents pour atteindre des zones spécifiques



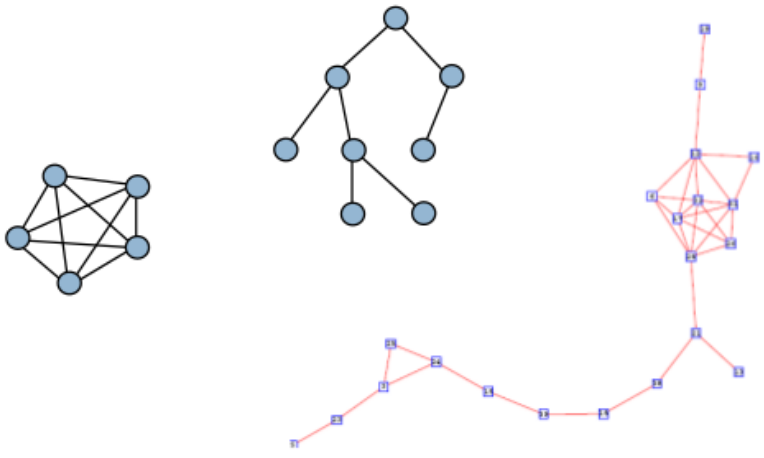


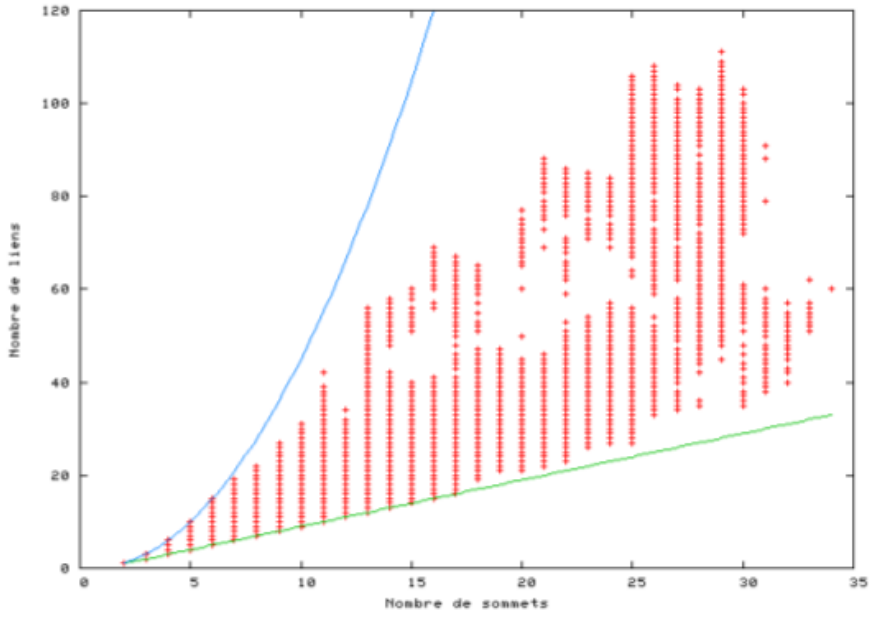
Composantes connexes



Composantes connexes

- Petites composantes : densité variable.
- Grosse composantes : faible densité ($max(nb\_liens) \sim 4.5 \times nb\_sommets$)

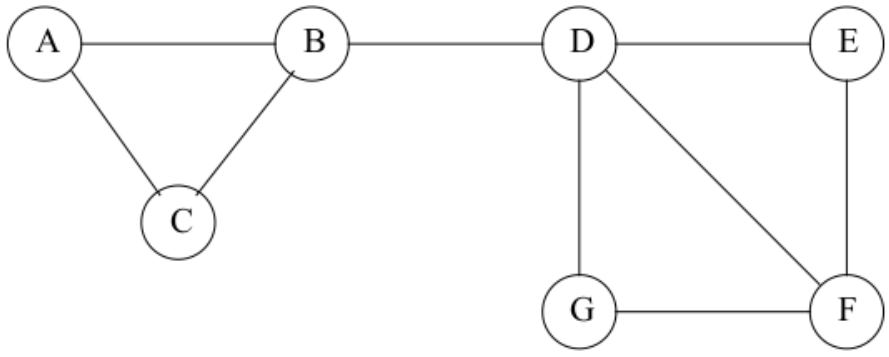




Algorithmique

Matrices de description des graphes

- Matrice d'adjacence : $m_{ij} = 1$ si l'arête (v_i, v_j) existe, 0 sinon.



$$\begin{array}{c} \\ A \\ B \end{array} \begin{array}{c} \begin{array}{ccccccc} A & B & & & & & \end{array} \\ \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

- Matrice des degrés : $m_{ij} = d(v_j)$ pour $i = j$, 0 sinon.

$$\begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix}$$

- Matrice de Laplace (ou "laplacienne") : L = D - A (cf théorie spectrale)

Représentation des graphes

Il y a deux représentations classiques possibles pour les graphes :

- liste d'adjacence (listes chaînées ou tableau de longueurs variables)
  - efficace pour énumérer les successeurs d'un nœud, beaucoup moins les prédecesseurs
- matrice d'adjacence
  - attention au coût pour des graphes "creux"
  - représentations optimisées dans les langages/frameworks modernes

Besoin d'algorithmes spécifiques

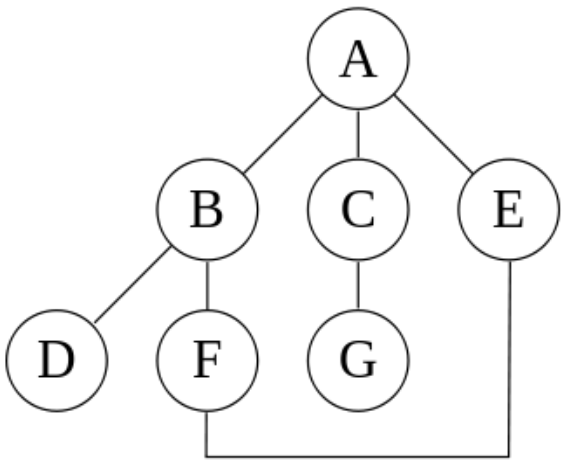
- Gros problème = taille :
  - Internet = Millions de sommets (routeurs)
  - Facebook = plus de 800 millions d'utilisateurs actifs
  - Web = Google connaît plus de 1 000 milliards d'URL distinctes
- il est non trivial de
  - stocker le graphe en mémoire
  - faire des calculs sur le graphe

Exemples

- Compter les triangles d'un graphe (clustering) :
  - naïvement O(n*n*n)
  - $O(m * n^{(1/a)})$ si distribution des degrés en loi de puissance d'exposant a
- Diamètre :
  - complexité théorique : O(nm)
  - approximation en O(m)
- Problèmes NP-complets
- Beaucoup de problèmes spécifiques aux graphes réels (détection de communautés). Approximation (non prouvée) linéaire.

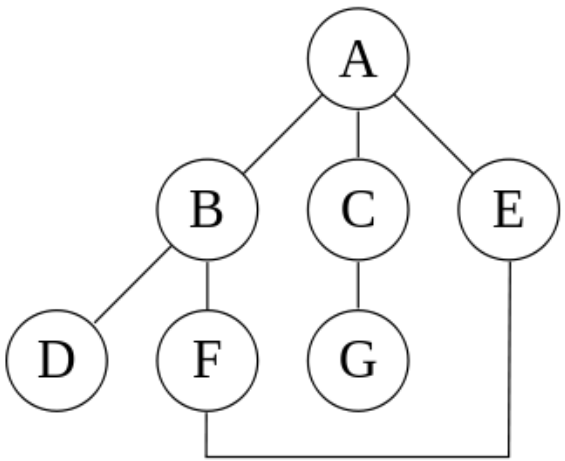
Parcours en profondeur

- En anglais, DFS, pour Depth First Search
- progresse à partir d'un sommet S en s'appelant récursivement pour chaque sommet voisin de S.
- pour chaque sommet, on prend le premier sommet voisin, on explore tous ses voisins (non marqués) avant de revenir au "père"
- Ordre de visite : A, B, D, F, E, C, G
- s'implémente avec une pile (LIFO)



Parcours en largeur

- En anglais, BFS, pour Breadth First Search
- pour chaque sommet, on repère tous ses voisins, on stocke ceux qui ne sont pas marqués dans une file (queue)
- Ordre de visite : A, B, C, E, D, F, G
- s'implémente avec une file (FIFO)
- on obtient les plus courts chemins à la racine



Outils

Frameworks et bibliothèques

- Pregel : Google, 2010. Passage de messages entre nœuds. Diverses implémentations sur Hadoop.
- GraphLab : projet de CMU, 2009. GraphX dans Spark
- Facebook : “Unicorn” (non public) ou Giraph ?
- Titan (HBase + Faunus) : Aurelius, 2013.

- Neo4j : base de données “graphe” open source.
- FlockDB (Twitter), AllegroGraph, GraphDB, ...

Logiciels

- Gephi
- Linkurious
- Tulip
- Guess

Démarche sur un problème réel

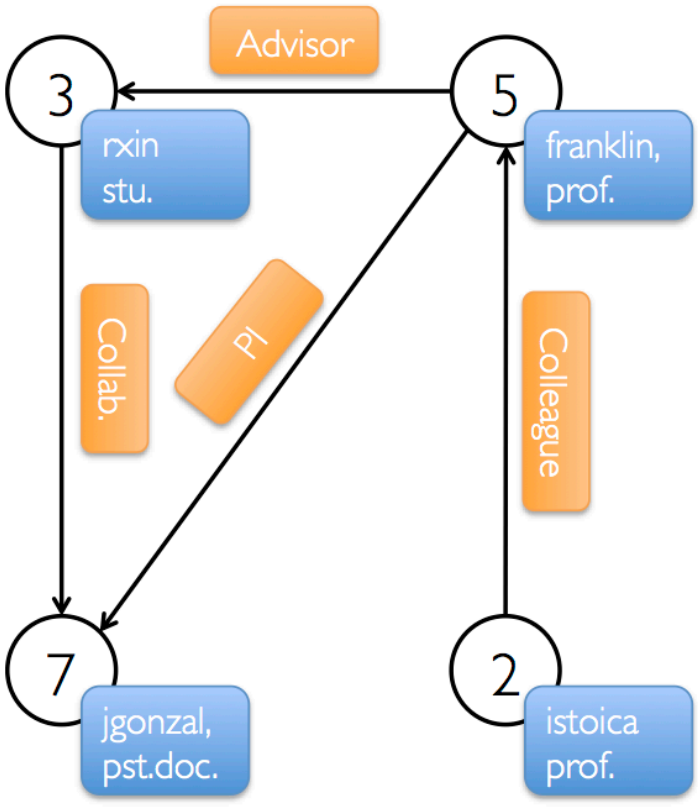
- définir les sommets (page web, compte FB/TW, etc.)
- définir les arêtes (appels 5x/mois, 2x/mois ?)
- identifier les hubs (traiter séparément, enlever)
- identifier les mesures pertinentes (PageRank, degree, triangle, LCC)
- construire la source de données
- écrire le code de traitement
- analyser
- recommencer !

GraphX

- librairie de Spark pour gérer les calculs sur les graphes
- en particulier, le parallèlisme
- introduit une abstraction Graph (au-dessus de RDD) :
  - un multigraphe orienté, avec des propriétés attachées à chaque sommet et chaque arête
  - facilite les cas où il y a plusieurs arêtes entre des noeuds
- https://spark.apache.org/docs/latest/graphx-programming-guide.html

GraphX

Property Graph



Vertex Table

Id	Property (V)
3	(rxin, student)
7	(jgonzal, postdoc)
5	(franklin, professor)
2	(istoica, professor)

Edge Table

SrcId	DstId	Property (E)
3	7	Collaborator
5	3	Advisor
2	5	Colleague
5	7	PI

GraphX

```scala
val sc: SparkContext
// Create an RDD for the vertices
val users: RDD[(VertexId, (String, String))] =
  sc.parallelize(Array((3L, ("rxin", "student")), (7L, ("jgonzal", "postdoc")),
                       (5L, ("franklin", "prof")), (2L, ("istoica", "prof"))))
// Create an RDD for edges
val relationships: RDD[Edge[String]] =
  sc.parallelize(Array(Edge(3L, 7L, "collab"),    Edge(5L, 3L, "advisor"),
                       Edge(2L, 5L, "colleague"), Edge(5L, 7L, "pi")))
// Define a default user in case there are relationship with missing user
val defaultUser = ("John Doe", "Missing")
// Build the initial Graph
val graph = Graph(users, relationships, defaultUser)
```

GraphX : opérateurs

```scala
val graph: Graph[(String, String), String]
// Use the implicit GraphOps.inDegrees operator
val inDegrees: VertexRDD[Int] = graph.inDegrees
```

D'autres opérateurs :

- numEdges/numVertices
- collectNeighbors
- subgraph
- connectedComponents
- triangleCount

Conclusion

Références

- Ce cours repose sur les travaux de :
  - l'équipe ComplexNetworks du LIP6 (UPMC), http://www.complexnetworks.fr (membres passés et présents)
  - en particulier les cours de Jean-Loup Guillaume (PR, U. de La Rochelle) et de Clémence Magnien
  - le livre *Mining Massive datasets* (http://www.mmds.org), de Jure Leskovec, Anand Rajaraman, Jeff Ullman