Structure et dynamique des réseaux

Cours 12 : Diffusion d'information

Clémence Magnien, Lionel Tabourier, Fabien Tarissan

LIP6 – CNRS et Université Pierre et Marie Curie

prenom.nom@lip6.fr

Emploi du temps

Rappel : rendu du TP 5 : lundi 4 janvier minuit

Modifications des cours/TP des prochaines semaines :

- mardi 13 janvier : cours de 13h45 à 15h45 - pas de TP
- mardi 20 janvier : cours de 13h45 à 15h45 puis TP de 16h à 18h
- mercredi 28 janvier : 4h de TP le matin

Plan

Outline

Contexte

Dissémination d'information

Contexte : 1 source vers tous les nœuds du réseau.

- Plusieurs solutions/algorithmes : push vs. pull, innodation, gossip, ...
- Plusieurs critères d'évaluation :
  - passage à l'échelle ?
  - efficacité : peu de redondance ? Tous les nœuds sont atteints ?
  - simplicité : connaissance locale ou globale de la topologie ?
- Impact de la topologie sous-jacente ?

Plusieurs approches

Algorithmes de type :

- *push* : sur réception d'un message, un nœud décide comment le propager
- *pull* : périodiquement, chaque nœud interroge ses voisins
- *push & pull* : succession de phases de type *push* puis *pull*.

Chaque catégorie décline ses propres variantes

Focus sur des algorithmes de type *push*

Le principe général est le suivant :

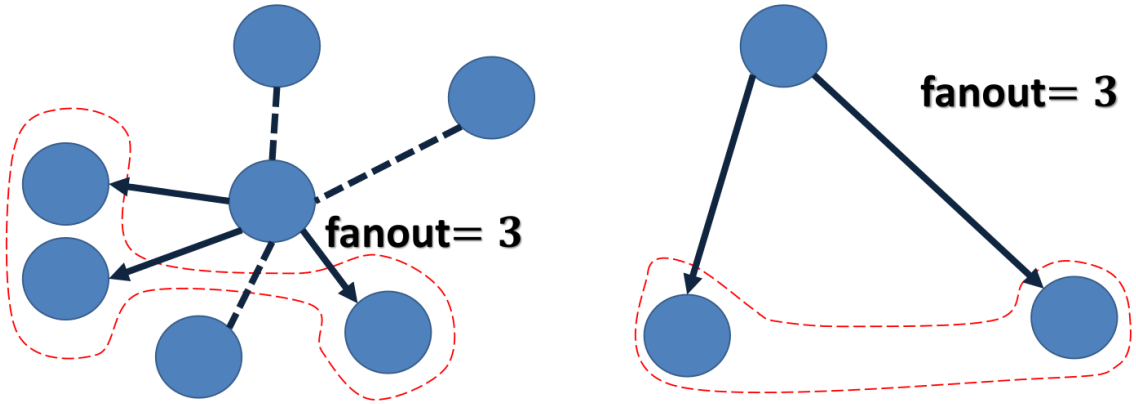
1. Initialement, la source envoie le message *msg* à tous ses voisins
2. Pour tout nœud, s'il n'a pas déjà reçu *msg* :
   1. Ajoute *msg* à sa liste de messages reçus
   2. Applique *GossipX*($< msg >, param$)

Plusieurs algorithmes *GossipX* possibles

Fixed fanout Gossip (*GossipFF*)

GossipFF :

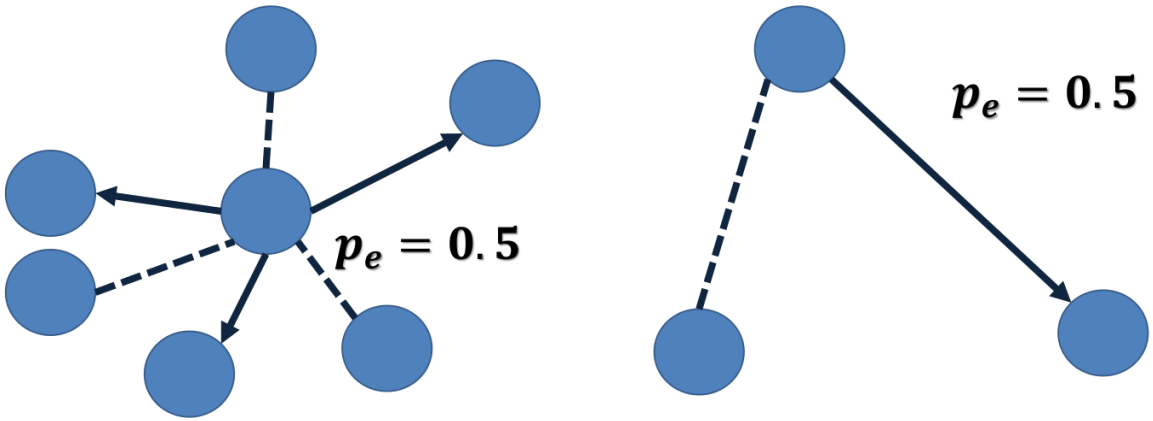
- Paramètre : *fanout*
- Tout nœud diffuse le message à *fanout* voisins.



Probabilistic edge Gossip (*GossipPE*)

GossipPE :

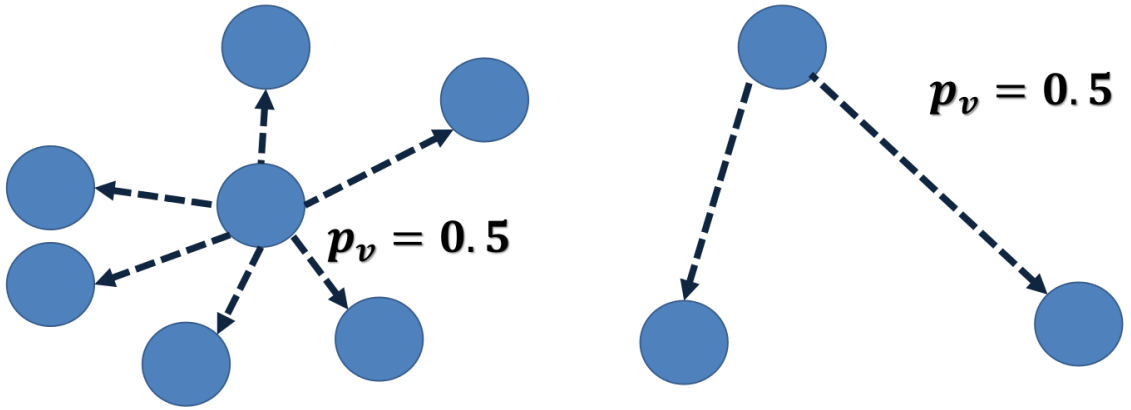
- Paramètre : p_e
- Tout nœud diffuse le message à chacun de ses voisins avec une probabilité de p_e.



Variante : probabilistic edge Gossip (*GossipPB*)

GossipPB :

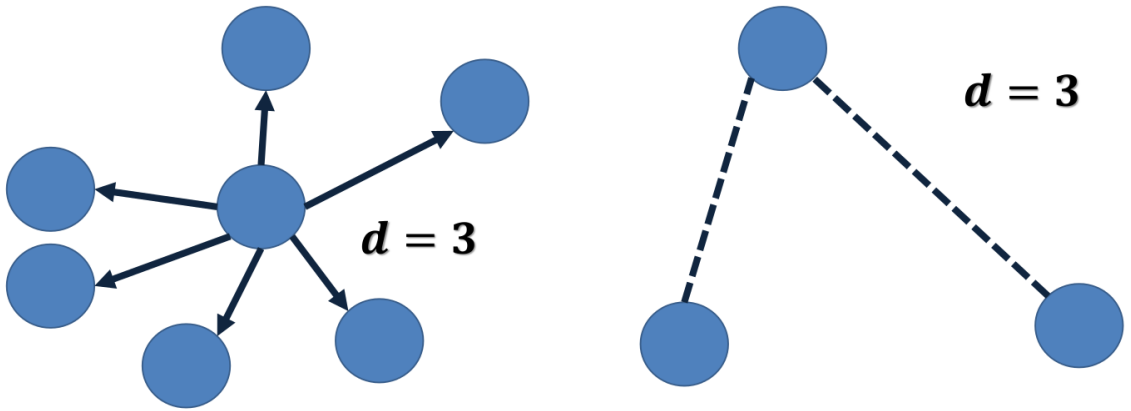
- Paramètre : p_v
- Tout nœud diffuse le message à tous ses voisins avec une probabilité p_v.



Degree Threshold Gossip (*GossipDT*)

GossipDT :

- Paramètre : d
- Tout nœud diffuse le message à chacun de ses voisins si son degré est supérieur à d.



Évaluation des performances

Plusieurs critères entrent en jeux :

- Complexité en message : redondance des messages

$$M = \frac{\text{Nb Copies}}{\text{Nb noeuds} - 1}$$

- Infection : pourcentage de nœuds qui ont reçus un message

$$R = \frac{\text{Nb nœuds infectés}}{\text{Nb nœuds}}$$

- Fiabilité : pourcentage de messages qui sont reçus par tous les nœuds

$$R = \frac{\text{Nb diffusions complètes}}{\text{Nb diffusions}}$$

- Latence : nombre de sauts maximum

$$L = \max(\min(\textit{source} \longrightarrow \textit{noeuds}))$$

Évaluation des performances

D'autres métriques dérivées possibles :

- End-to-end delay
- Validité
- Intégrité
- Last Delivery Hop (LDH)
- ...

Quelques résultats généraux

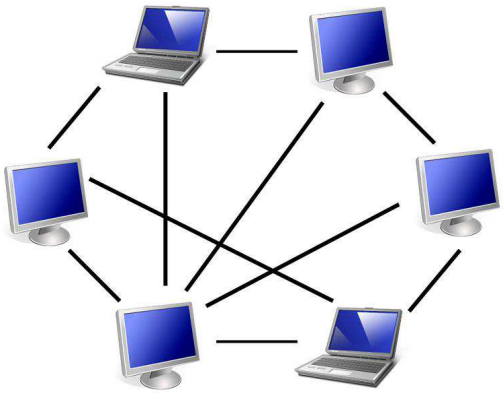
Quelques tendances (Randomize rumor spreading, Karp et.al., Symposium on Fondations of Computer Science, 2000) :

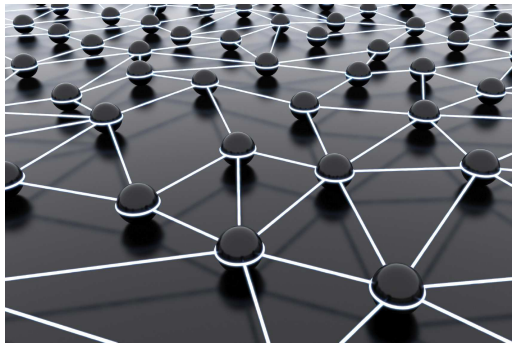
Algo	Complexité	Latence
Push	$ln(N)$	$log_2(N) + ln(N)$
Pull	$ln(ln(N))$	$ln(N) + ln(ln(N))$
Push & Pull	$ln(ln(N))$	$log_3(N) + ln(ln(N))$

- Pull : permet aux sites isolés de facilement récupérer l'information
- Push & Pull : souvent implémentés
- Attention : Pull et Push & Pull impliquent des échanges d'informations en plus de la diffusion

Cas d'applications envisagés

Différents contextes d'applications



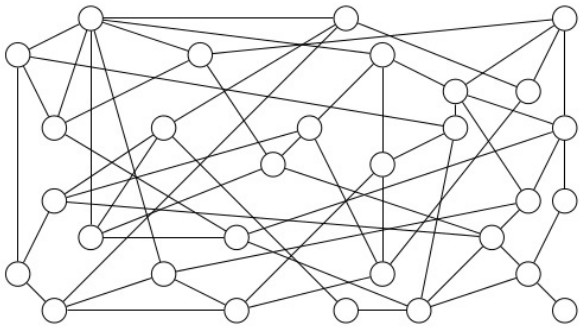


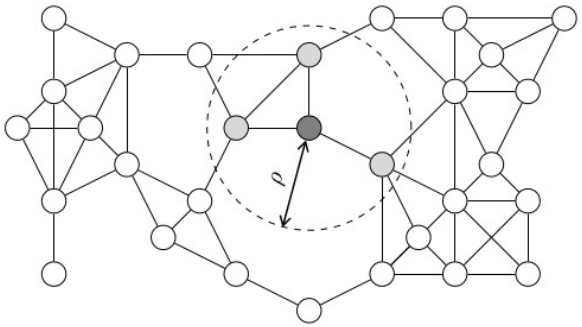
$\Longrightarrow$ Différentes topologies !

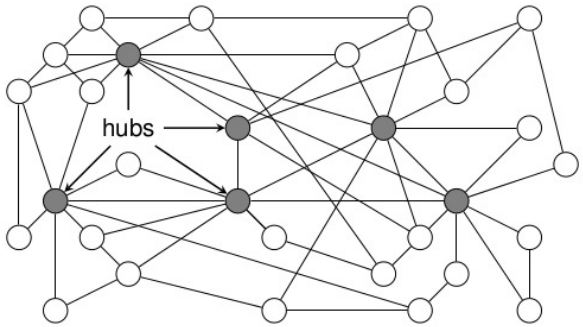
Quels modèles pour la topologie

En fonction des propriétés voulues/supposées

- Graphes Erdös -Rényi
- Random Geometric Graph
- Graphes sans échelles







Comment comparer les algorithmes Gossip

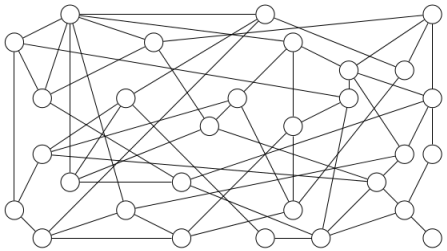
Chaque algorithme a un paramètre différent.
$\Longrightarrow$ Besoin d'unification : *fanout effectiff* F_{eff}

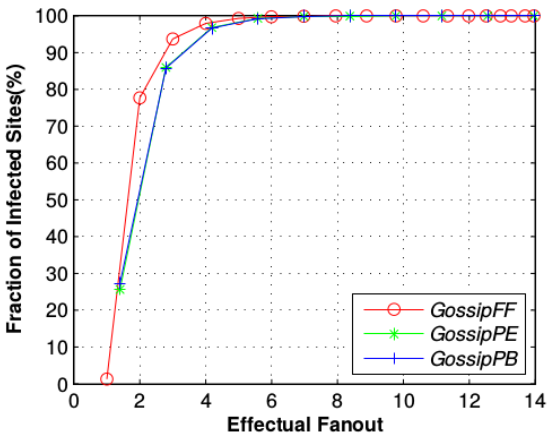
[Fair Comparison of Gossip Algorithms over Large-Scale Random Topologies, Hu et.at., 31th IEEE International Symposium on Reliable Distributed Systems (SRDS'12)]

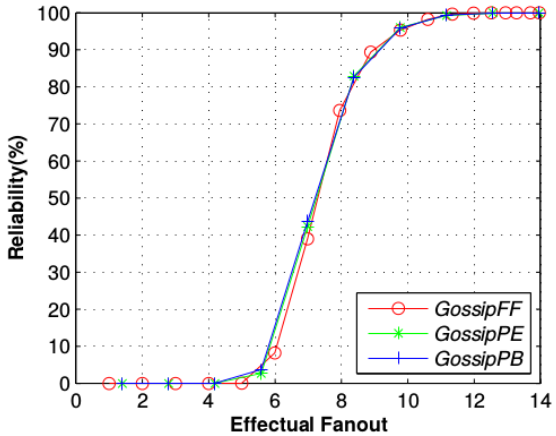
- *GossipPE* : $F_{eff} = p_e * \tilde{d}$
- *GossipPB* : $F_{eff} = p_b * \tilde{d}$
- *GossipFF* : $F_{eff} = \sum_{k=1}^{fanout-1} P(k) * k + \sum_{k=fanout}^{N-1} P(k) * fanout$

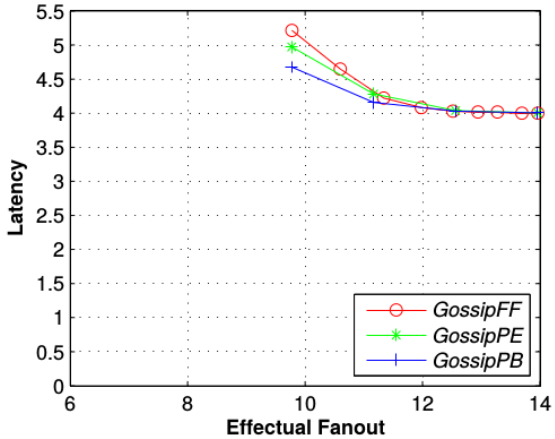
où $\tilde{d}$ est le degré moyen.

Résultats sur graphes aléatoires





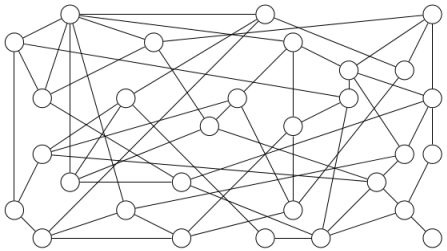


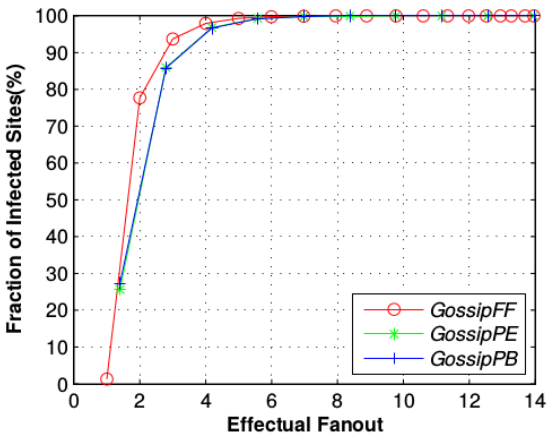


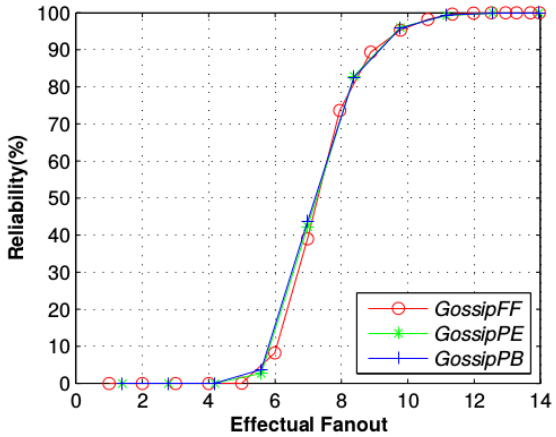
Résultat

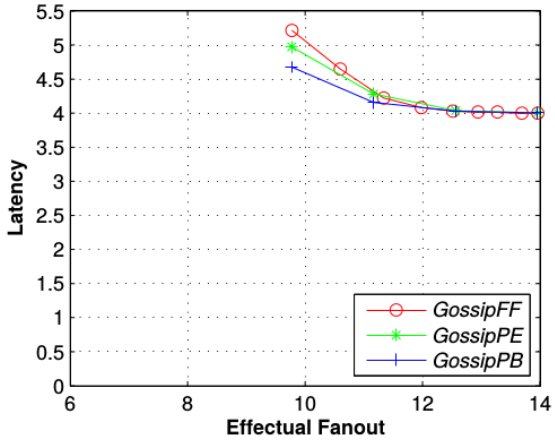
Tous équivalents

Résultats sur graphes aléatoires





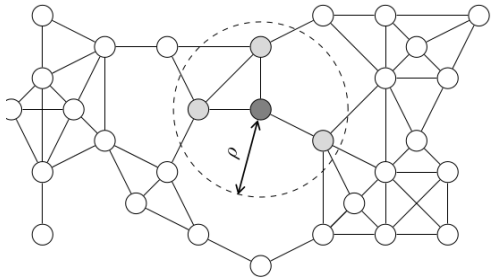


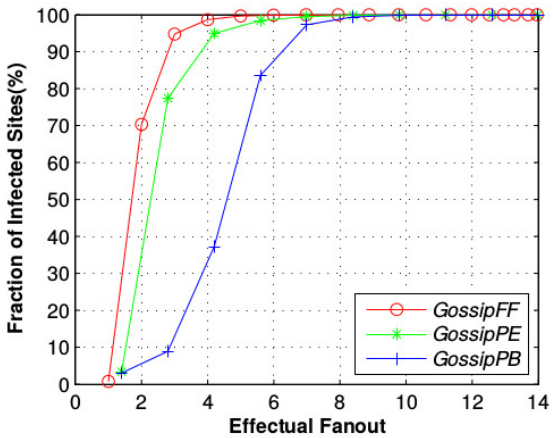


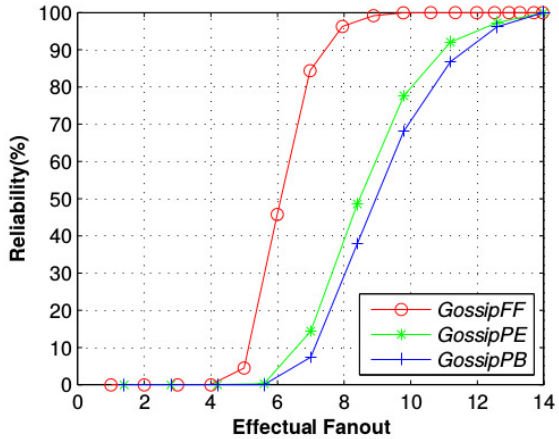
Résultat

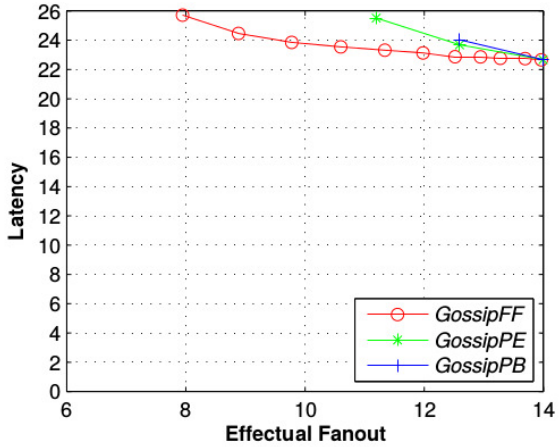
Tous équivalents

Résultats sur graphes géométriques





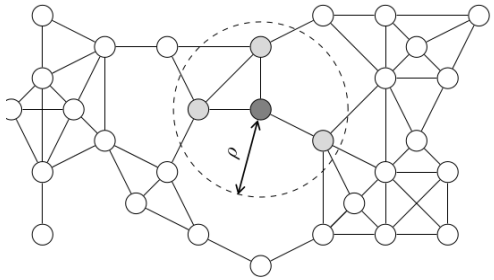


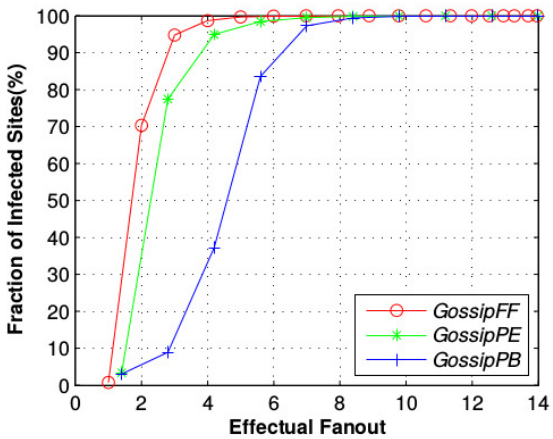


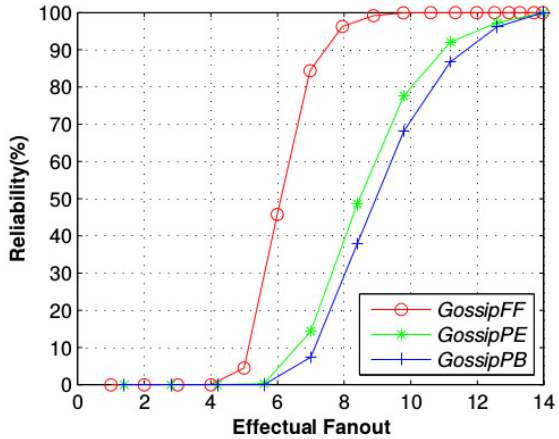
Résultat

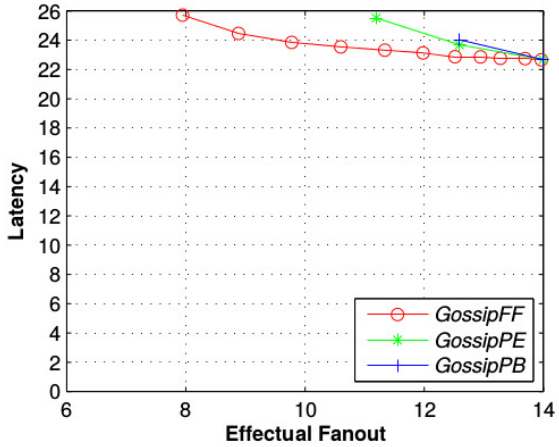
$F_{eff}FF \leq F_{eff}PE \leq F_{eff}PB$

Résultats sur graphes géométriques





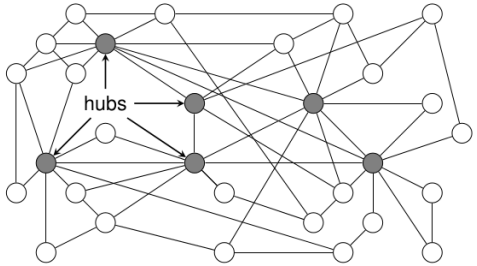


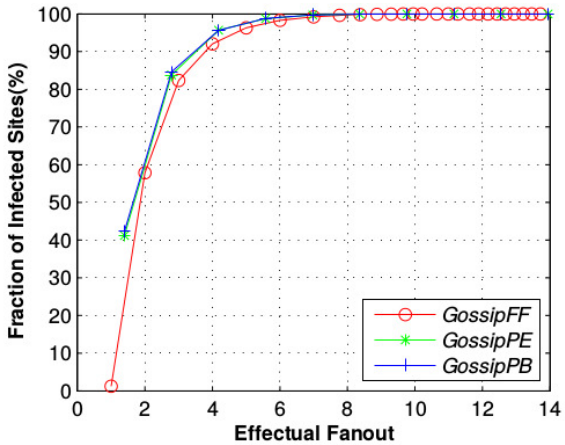


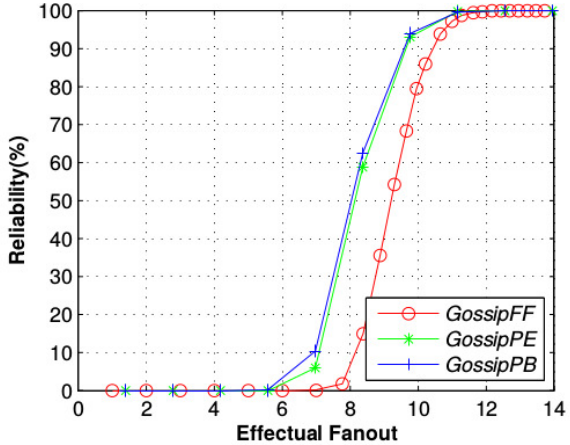
Résultat

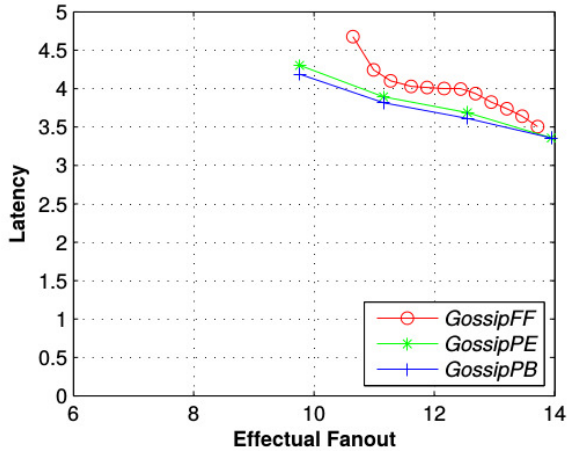
$F_{eff}FF \leq F_{eff}PE \leq F_{eff}PB$

Résultats sur graphes sans échelles





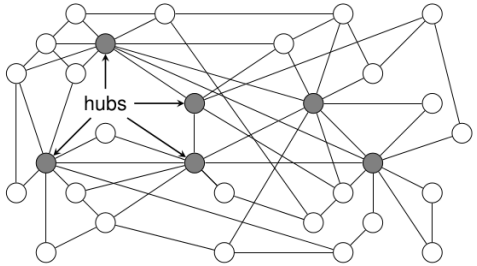


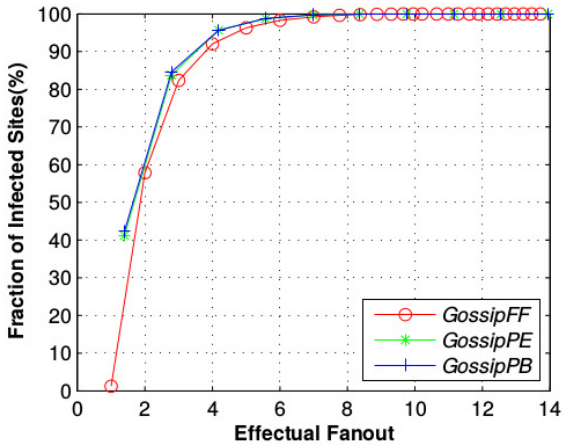


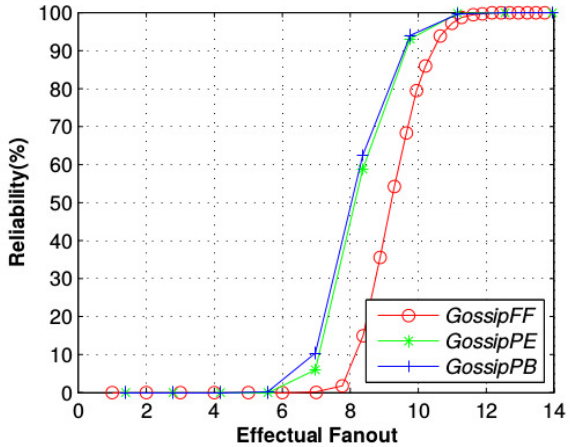
Résultat

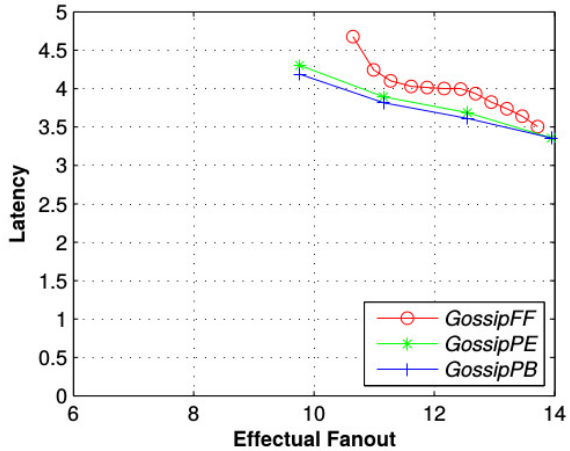
$F_{eff}PE \leq F_{eff}PB \leq F_{eff}FF$

Résultats sur graphes sans échelles





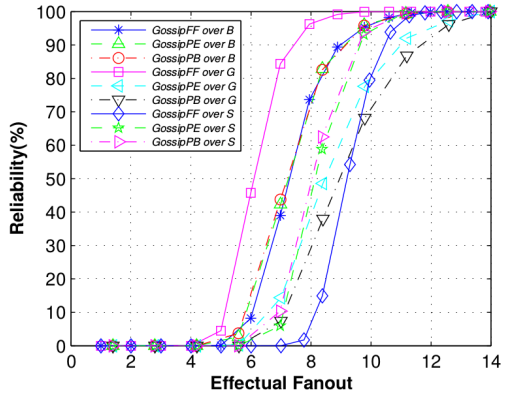




Résultat

$F_{eff}PE \leq F_{eff}PB \leq F_{eff}FF$

Vue générale



Autre comparaison : gain vis-à-vis de l'innondation

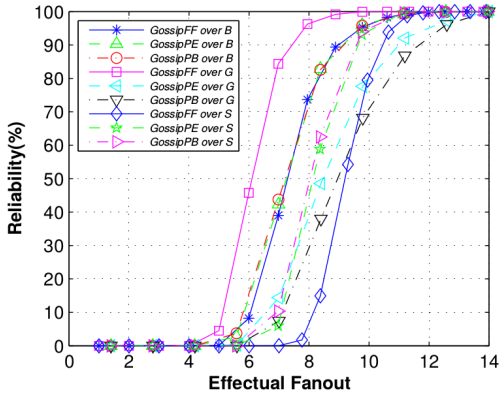
- Pour atteindre une fiabilité de 80% :

Algo	ER	Géo	BA
GossipFF	40%	52%	23%
GossipPB	40%	27%	34%
GossipPE	40%	31%	34%

- Pour atteindre une fiabilité de 99% :

Algo	ER	Géo	BA
GossipFF	14%	40%	14%
GossipPB	14%	0%	21%
GossipPE	14%	0%	21%

Vue générale



Autre comparaison : gain vis-à-vis de l'innondation

- Pour atteindre une fiabilité de 80% :

Algo	ER	Géo	BA
GossipFF	40%	52%	23%
GossipPB	40%	27%	34%
GossipPE	40%	31%	34%

- Pour atteindre une fiabilité de 99% :

Algo	ER	Géo	BA
GossipFF	14%	40%	14%
GossipPB	14%	0%	21%
GossipPE	14%	0%	21%

Outline

Pourquoi le P2P

Étudier les phénomènes de diffusions réels

Motivation :

- De plus en plus de plateforme de diffusion à grande échelle
- Émergence de comportement collectifs à partir d'interactions locales
- Couche *technologique* vs. couche *sociale* (micro/macro)

Quelques problématiques :

- Comment acquérir de l'information ? **(mesure)**
- Quelles sont les propriétés pertinentes ? **(analyse)**
- Quels sont les bons modèles ? **(modélisation)**
- Comment calculer efficacement ? **(algorithmique)**

Un autre point de vue

Phénomènes de diffusion dans les réseaux

Une **trace de diffusion** est composée de :

1. un graphe sous-jacent (statique)
2. une information temporelle sur qui diffuse l'information à qui

Examples:

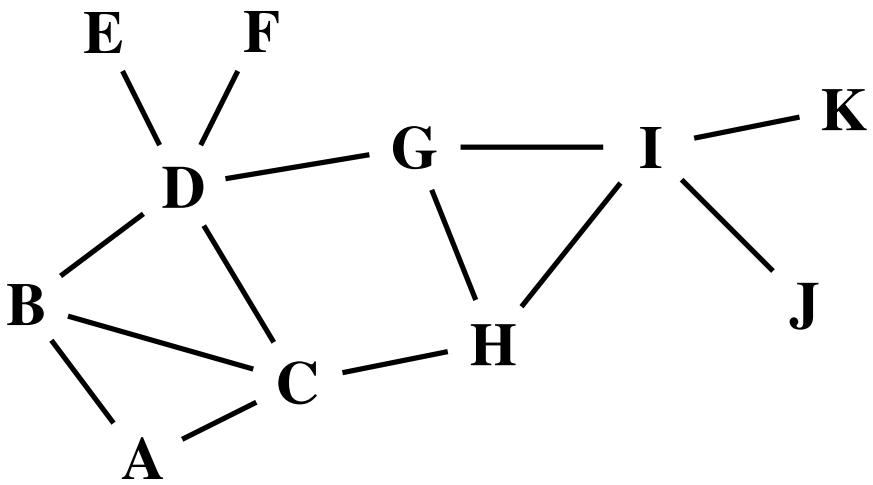
- virus dans les réseaux de contacts
- rumeurs dans les réseaux sociaux
- fichiers dans les réseaux P2P
- ...

Un autre point de vue

Phénomènes de diffusion dans les réseaux

Une **trace de diffusion** est composée de :

1. un graphe sous-jacent (statique)
2. une information temporelle sur qui diffuse l'information à qui

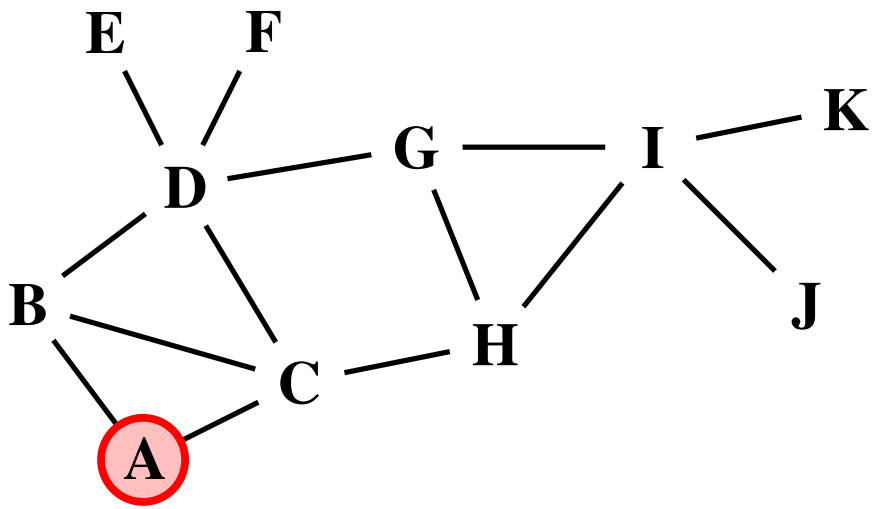


Un autre point de vue

Phénomènes de diffusion dans les réseaux

Une **trace de diffusion** est composée de :

1. un graphe sous-jacent (statique)
2. une information temporelle sur qui diffuse l'information à qui

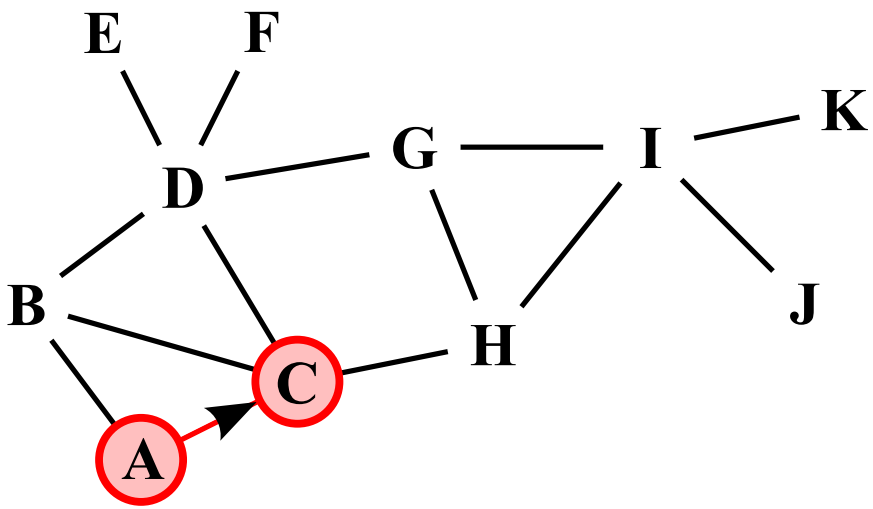


Un autre point de vue

Phénomènes de diffusion dans les réseaux

Une **trace de diffusion** est composée de :

1. un graphe sous-jacent (statique)
2. une information temporelle sur qui diffuse l'information à qui

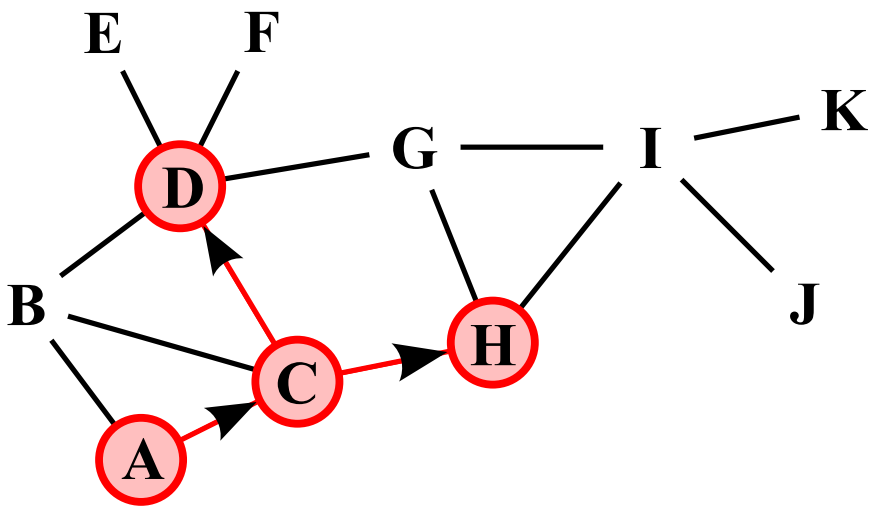


Un autre point de vue

Phénomènes de diffusion dans les réseaux

Une **trace de diffusion** est composée de :

1. un graphe sous-jacent (statique)
2. une information temporelle sur qui diffuse l'information à qui

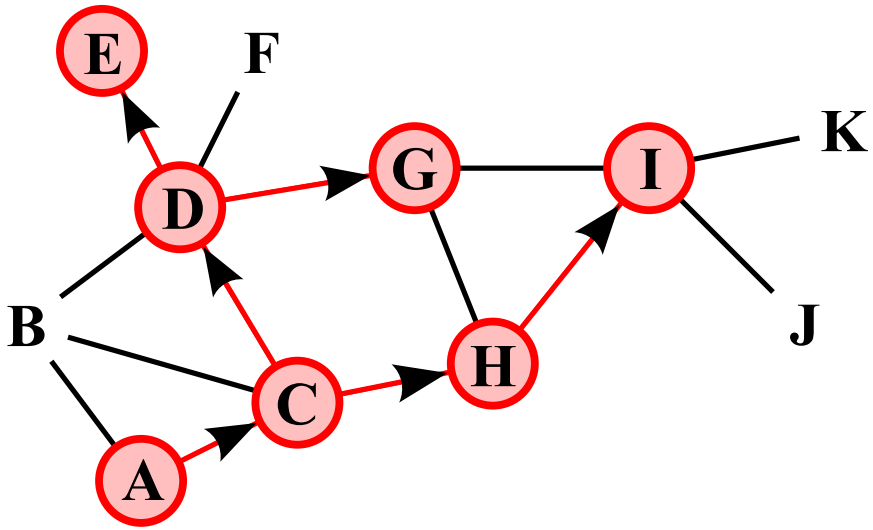


Un autre point de vue

Phénomènes de diffusion dans les réseaux

Une **trace de diffusion** est composée de :

1. un graphe sous-jacent (statique)
2. une information temporelle sur qui diffuse l'information à qui

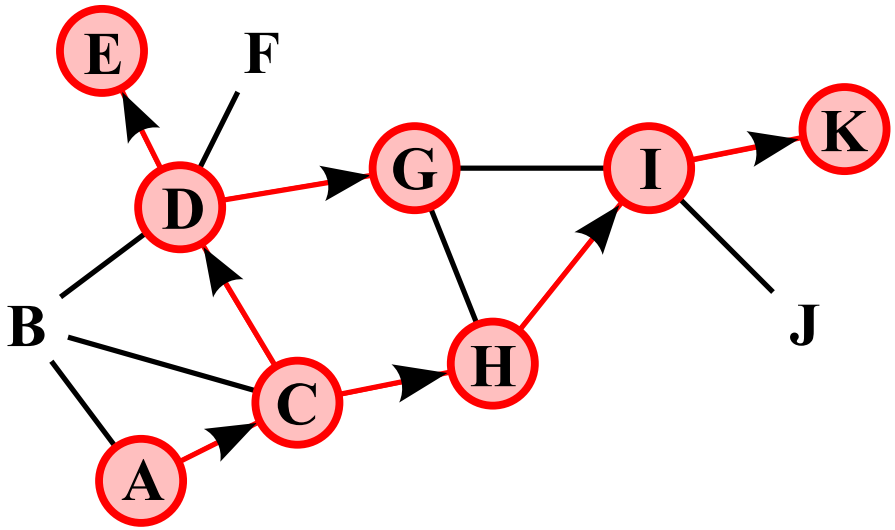


Un autre point de vue

Phénomènes de diffusion dans les réseaux

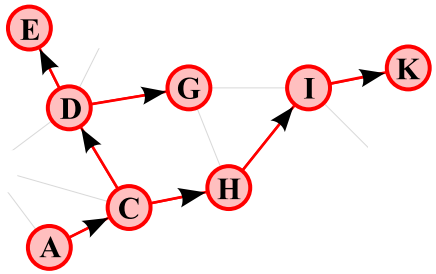
Une **trace de diffusion** est composée de :

1. un graphe sous-jacent (statique)
2. une information temporelle sur qui diffuse l'information à qui

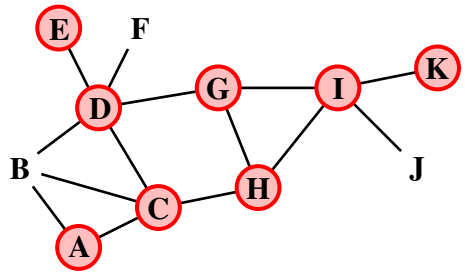


Problématiques de mesures

Comment obtenir une trace de diffusion complète



Cascade de diffusion
réseau sous-jacent ?



Nœuds infectés par une diffusion
liens de transmission ?

Approche

Approche répendue : *les diffusions sont similaires à des épidémies*

modèle SIR

- état des nœuds: *susceptible* $\rightarrow$ *infecté* $\rightarrow$ *retiré*
- les nœuds infectés propagent l'information à leur voisins avec une probabilité de p.

Objectif

Évaluer la pertinence du modèle SIR pour décrire :

- **phénomènes de diffusions**
- **réels**
- s'appuyant sur la **topologie issue du réseau des utilisateurs**

Méthodologie

- Calculer les propriétés pertinentes des diffusions observées
- Calibrer le modèle SIR à partir des données réelles en vue de simulations
- Comparer les propriétés des traces réelles vs. traces générées à partir du modèle SIR

Jeu de données

Notre jeu de données: log des requêtes sur un serveur eDonkey

format: (*timestamp, provider id, client id, file id*)

48h avec 5 M de pairs, 2 M de fichiers et 212 M de requêtes
(8h avec 2 M paires, 800k fichiers et 23 M requêtes)

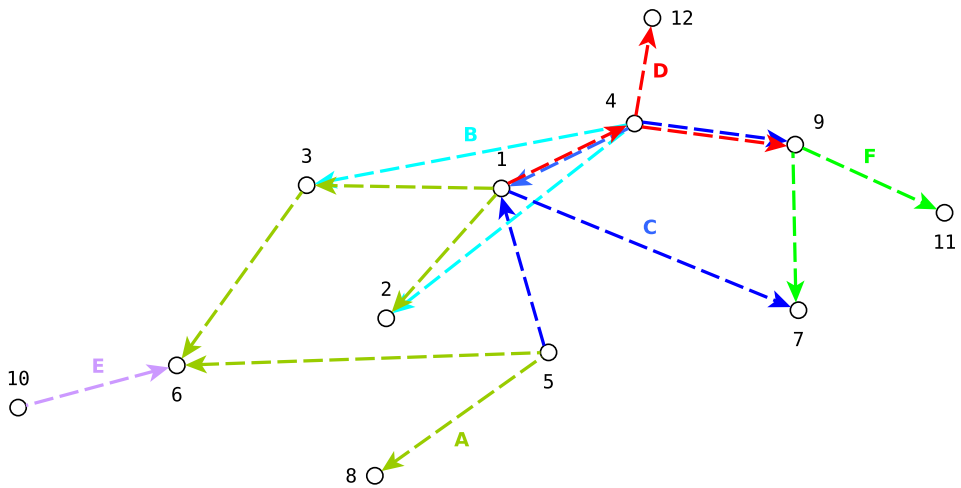
Time	Provider	Client	File
1	1	2	A
2	1	3	A
3	4	2	B
4	4	1	C
4	5	1	C
5	1	4	D
6	5	6	A
6	3	6	A
7	1	7	C

Time	Provider	Client	File
8	5	8	A
9	4	9	D
10	4	3	B
11	10	5	E
12	9	11	F
13	4	9	C
14	4	12	D
15	9	7	F

Cascades de diffusions

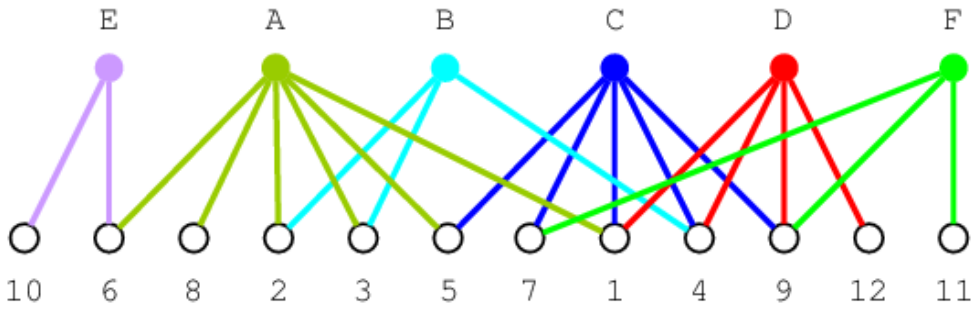
Time	Provider	Client	File
1	1	2	A
2	1	3	A
6	5	6	A
6	3	6	A
8	5	8	A
3	4	2	B
10	4	3	B
4	4	1	C
4	5	1	C

Time	Provider	Client	File
7	1	7	C
13	4	9	C
5	1	4	D
9	4	9	D
14	4	12	D
11	10	5	E
12	9	11	F
15	9	7	F

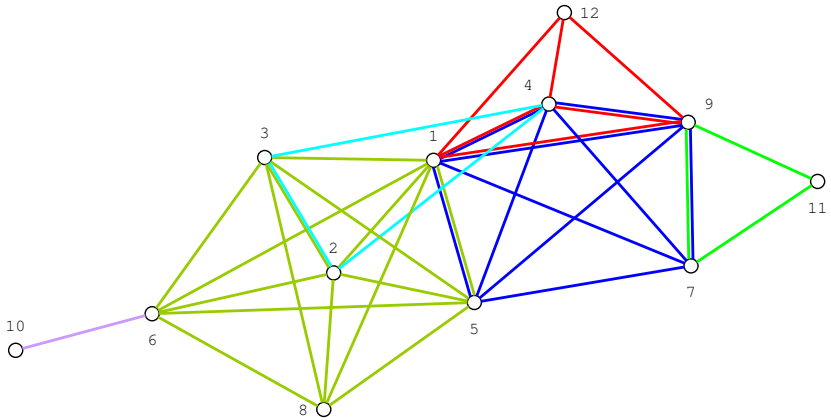


Graphe d'interêt

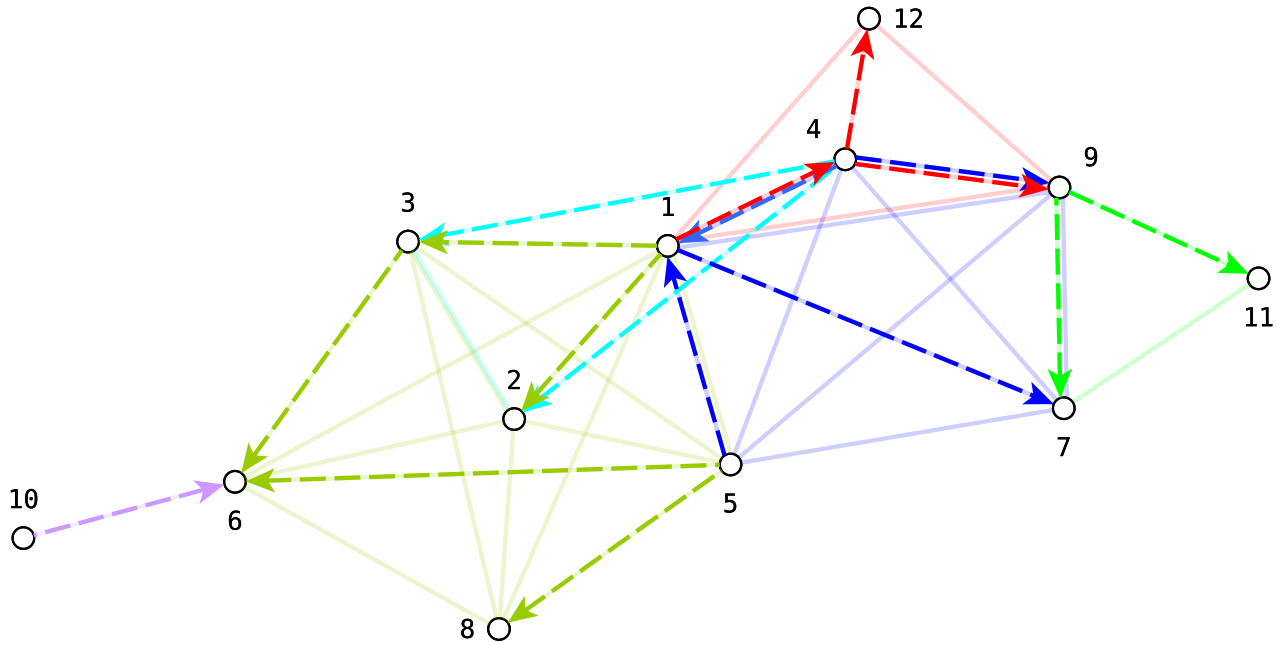
1. Graphe Bipartie pour décrire les *intérêts* entre les pairs



2. Projection du graphe bipartie : graphe d'interêt



Graphe d'interêt et diffusion

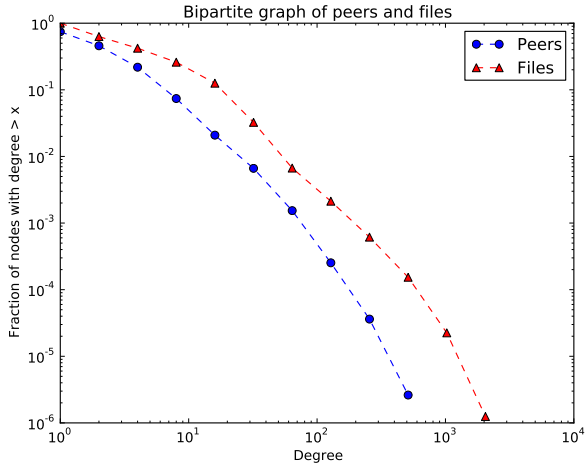


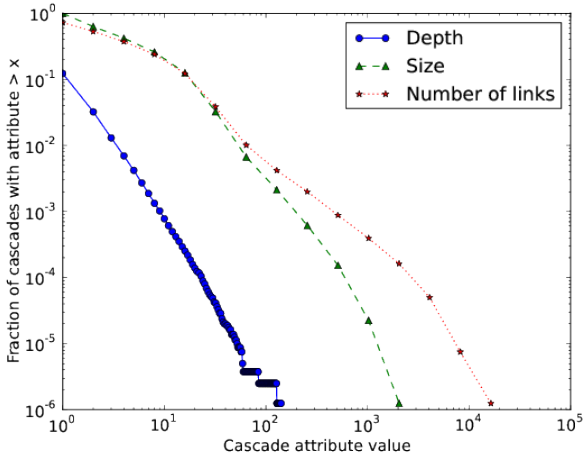
Propriété

La diffusion a lieu sur le graphe d'interêt

Résultats d'analyse

- 99.63% de clients, 4.33% of fournisseurs: *free riders!*
- distribution des degrés dinstinctes pour tous nœuds vs. fournisseurs
- une seule composante génate





Propriétés clefs pour décrire les diffusions

- taille (nombre de nœuds)
- profondeur (longueur du plus long chemin)
- nombre de liens (densité ?)

Protocole

But

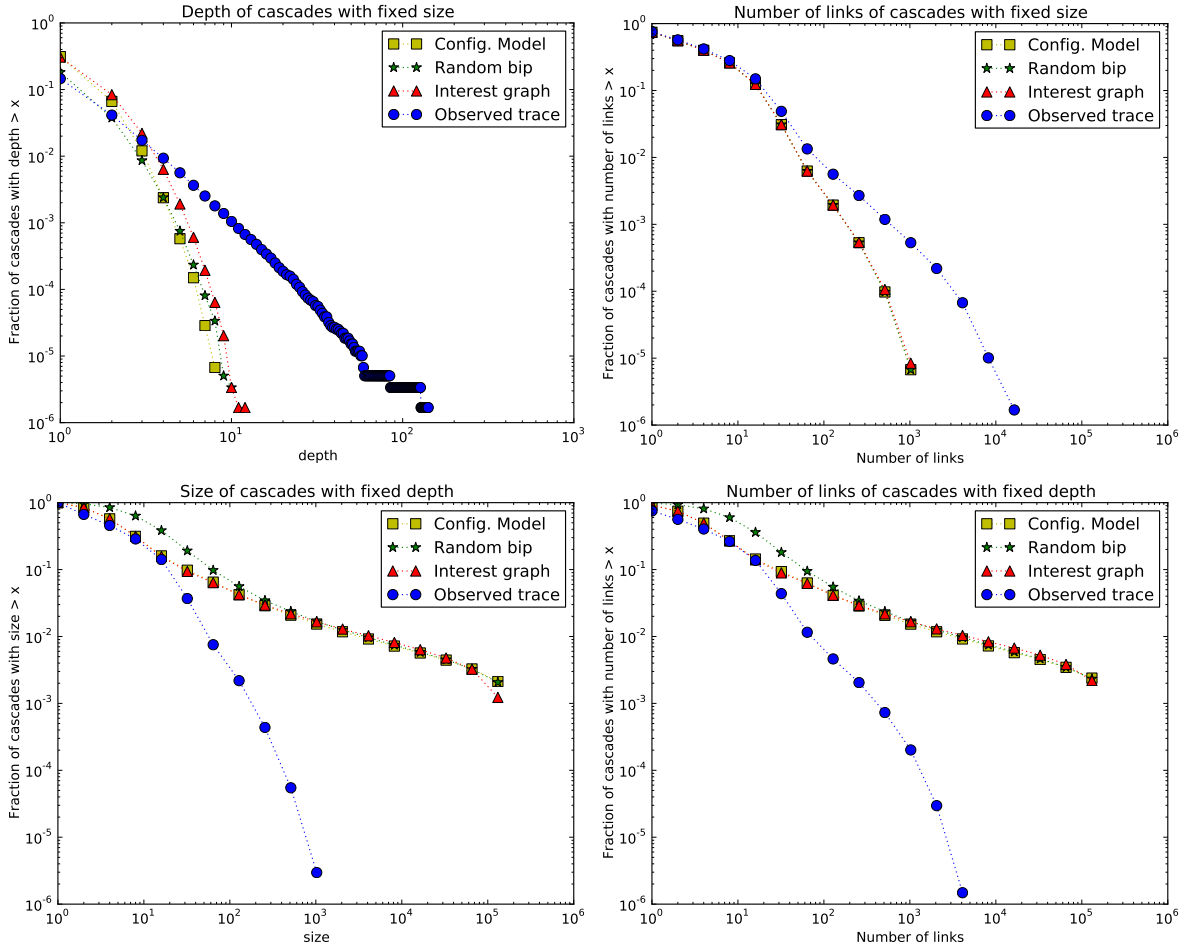
- Évaluer la pertinence du modèle SIR classique
- Étudier l'impact de la topologie sous-jacente sur les phénomènes de diffusion

Réels vs cascades simulées :

- À taille fixée
  ↪ comparaison de la profondeur et nombre de liens
- À profondeur fixée
  ↪ comparaison de la taille et du nombre de liens

Topologies sous-jacentes :

- Graphe d'interêt
- Graphe bipartie aléatoire
- Configuration model
- Erdos-Renyi



Résultat

Diffusions réelles sont plus denses et allongées que les diffusions simulées

Variantes possibles

On peut adapter le modèle pour prendre en compte :

- l'hétérogénéité de popularité des fichiers
- l'hétérogénéité du comportement des utilisateurs
- l'affinité d'intérêts
- dynamique et temporalité