

# Édition collaborative dans les navigateurs

Brice NÉDELEC

Thèse de doctorat soutenue le

5 Octobre 2016

## Jury

- Président :** Marc GELGON, Professeur, Université de Nantes
- Rapporteurs :** Anne-Marie KERMARREC, Directrice de recherche, INRIA  
Peter VAN ROY, Professeur, Université de Louvain
- Examineurs :** Gérald OSTER, Maître de conférences, Université de Lorraine  
Marc SHAPIRO, Directeur de recherche, INRIA
- Directeur :** Pascal MOLLI, Professeur, Université de Nantes
- Co-directeur :** Achour MOSTÉFAOUI, Professeur, Université de Nantes



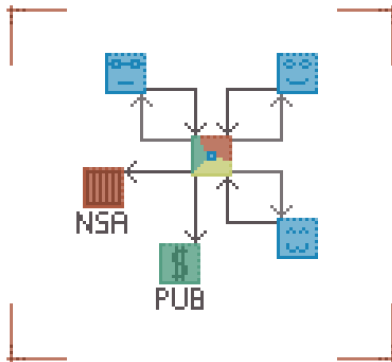
Un éditeur collaboratif permet

- À **plusieurs personnes**
- De **lire et modifier** un document

Grâce au Web,

- N'importe quel outil accédant à l'**internet** (*ordinateur, smartphone, tablette*) permet de **créer et d'éditer** un document aisément.
- Un simple lien permet de le **partager facilement** avec des amis ou des collègues.

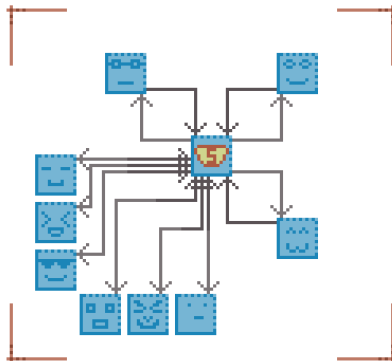
Problèmes de **confidentialité**, **censure**, **intelligence économique**, **legislation**, etc.



*En 2013, les révélations sur PRISM montre que la NSA possède des accès aux données hébergées par Google, Facebook, YouTube, Microsoft, Yahoo!, Skype, AOL et Apple.*

# Introduction > Problèmes de passage à l'échelle

Problèmes de passage à l'échelle, notamment en **nombre de collaborateurs**.



*En 2013, Coursera rassembla 41000 étudiants sur un seul cours. Les limitations de l'outil collaboratif utilisé conduisirent au “désastre”<sup>[1]</sup>.*

[1] [V. Strauss. How online class about online learning failed miserably. 2013.](#)

✓ Temps réel



✗ Sans fournisseur de services



Web ✓



Des milliers d'utilisateurs éditant simultanément ✗

✓ Temps réel



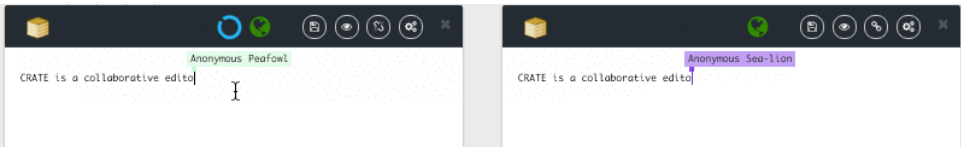
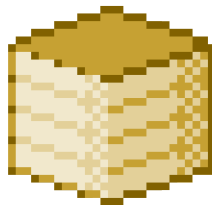
L'édition collaborative temps réel  
est-elle possible sur le Web,  
sans l'intervention d'un tiers  
et sans limites quant aux dimensions du système ?



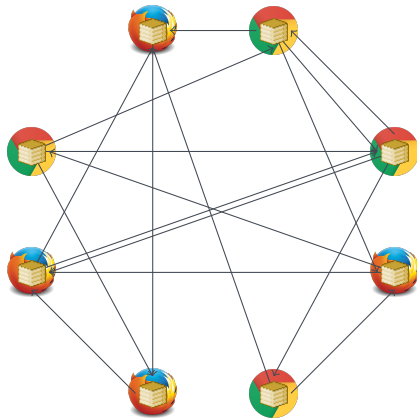
Des milliers d'utilisateurs édi-  
tant simultanément ✗

CRATE est un éditeur collaboratif

- temps réel ✓
- fonctionnant dans les navigateurs Web ✓
- sans fournisseur de services ✓
- passant à l'échelle ✓



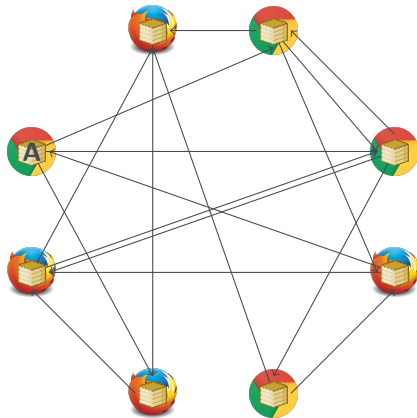
- Chaque éditeur possède une copie locale du document;
- Chaque éditeur est connecté à d'autres éditeurs.
- Chaque caractère tapé est directement inséré dans la copie locale;
- La modification est disséminée à l'ensemble du réseau;
- Les éditeurs recevant la modification l'appliquent.



⇒ **taille de messages** × **nombre de messages**

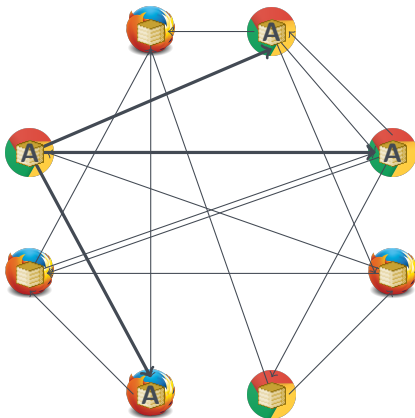


- Chaque éditeur possède une copie locale du document;
- Chaque éditeur est connecté à d'autres éditeurs.
- **Chaque caractère tapé est directement inséré dans la copie locale;**
- La modification est disséminée à l'ensemble du réseau;
- Les éditeurs recevant la modification l'appliquent.



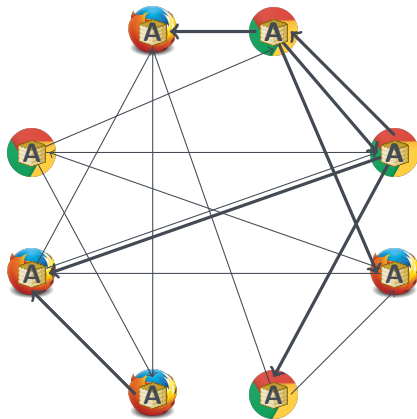
⇒ **taille de messages** × **nombre de messages**

- Chaque éditeur possède une copie locale du document;
- Chaque éditeur est connecté à d'autres éditeurs.
- Chaque caractère tapé est directement inséré dans la copie locale;
- **La modification est disséminée à l'ensemble du réseau;**
- **Les éditeurs recevant la modification l'appliquent.**



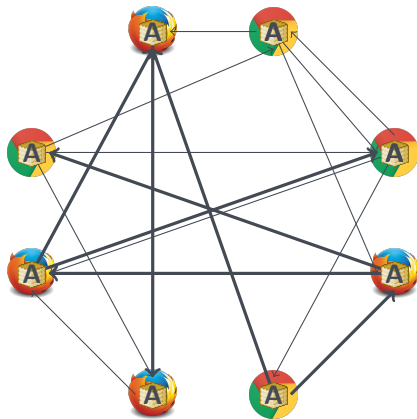
⇒ **taille de messages × nombre de messages**

- Chaque éditeur possède une copie locale du document;
- Chaque éditeur est connecté à d'autres éditeurs.
- Chaque caractère tapé est directement inséré dans la copie locale;
- **La modification est disséminée à l'ensemble du réseau;**
- **Les éditeurs recevant la modification l'appliquent.**



⇒ **taille de messages × nombre de messages**

- Chaque éditeur possède une copie locale du document;
- Chaque éditeur est connecté à d'autres éditeurs.
- Chaque caractère tapé est directement inséré dans la copie locale;
- **La modification est disséminée à l'ensemble du réseau;**
- **Les éditeurs recevant la modification l'appliquent.**

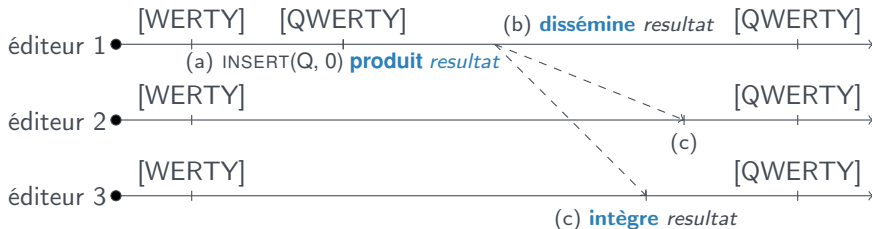


⇒ taille de messages × nombre de messages

- **taille des messages** : LSEQ qui, dans le contexte de l'édition collaborative, borne la taille des messages de manière sous-linéaire par rapport au nombre d'insertions effectuées dans le document.
- **nombre de messages** : SPRAY qui s'adapte automatiquement à la taille du réseau de manière logarithmique et qui supporte le processus complexe d'établissement de connexion disponible dans les navigateurs Web.

1. Introduction
2. Structure de séquences répartie à large échelle
3. Un protocole d'échantillonnage aléatoire adaptatif
4. Conclusion et perspectives

La **réplication optimiste**<sup>[2]</sup> améliore la **disponibilité** d'un document et sa **réactivité** aux changements effectués.

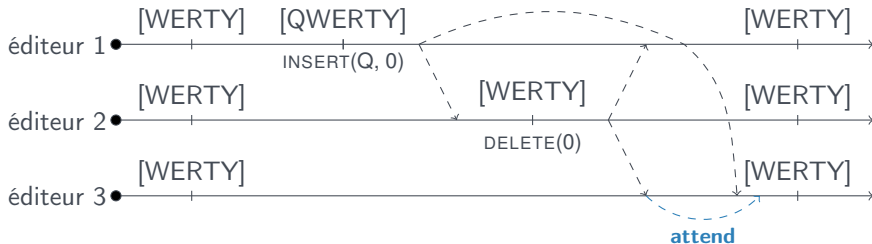


[2] Y. Saito and M. Shapiro. "Optimistic replication". (2005).

In: [ACM Comput. Surv.](#)

L'édition collaborative temps réel nécessite un système préservant les trois propriétés<sup>[3]</sup> :

- **Convergence;**
- **Causalité;**
- **Intention.**



[3] C. Sun et al. "Achieving convergence, causality preservation, and intention preservation in real-time cooperative editing systems". In: [TOCHI \(1998\)](#).



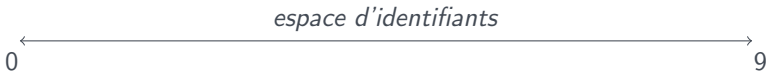
Une séquence se définit par un **ordre dense** sur ses éléments : les éléments sont ordonnés et il est toujours possible d'insérer un élément entre deux autres éléments.

Pour créer cet ordre dense, il est possible d'allouer un **identifiant** unique et immuable à chaque caractère<sup>[4]</sup>. Les identifiants permettent de retrouver l'ordre des éléments de la séquence.

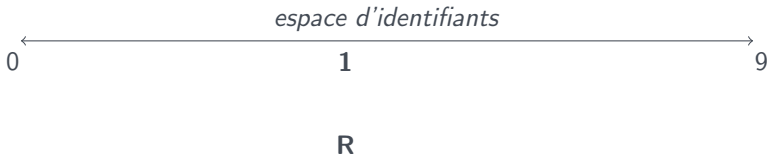
---

[4] S. Weiss et al. "Logoot-Undo: Distributed Collaborative Editing System on P2P Networks". In: [TPDS](#) (2010).

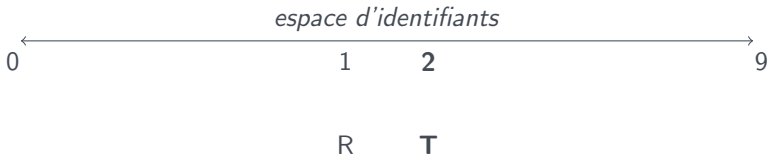
INSERT(R, 0) INSERT(T, 1) INSERT(Y, 2)  
INSERT(E, 0) INSERT(W, 0) INSERT(Q, 0)



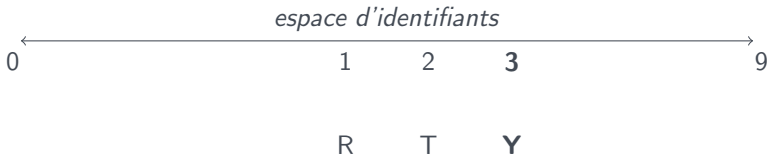
INSERT(R, 0) INSERT(T, 1) INSERT(Y, 2)  
INSERT(E, 0) INSERT(W, 0) INSERT(Q, 0)



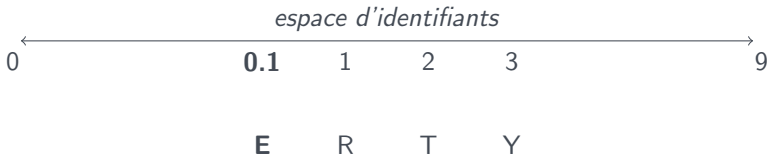
INSERT(R, 0) INSERT(T, 1) INSERT(Y, 2)  
INSERT(E, 0) INSERT(W, 0) INSERT(Q, 0)



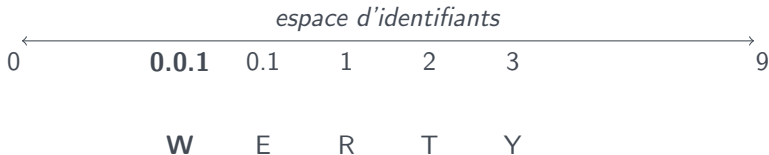
INSERT(R, 0) INSERT(T, 1) INSERT(Y, 2)  
INSERT(E, 0) INSERT(W, 0) INSERT(Q, 0)



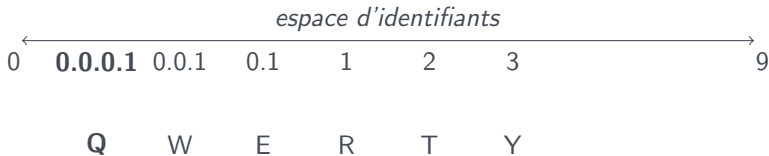
INSERT(R, 0) INSERT(T, 1) INSERT(Y, 2)  
INSERT(E, 0) INSERT(W, 0) INSERT(Q, 0)



INSERT(R, 0) INSERT(T, 1) INSERT(Y, 2)  
INSERT(E, 0) INSERT(W, 0) INSERT(Q, 0)

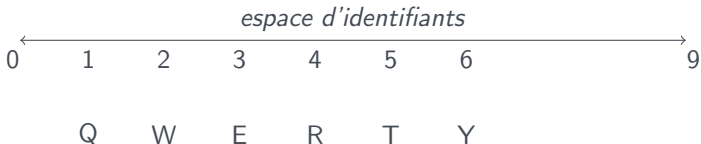


INSERT(R, 0) INSERT(T, 1) INSERT(Y, 2)  
INSERT(E, 0) INSERT(W, 0) INSERT(Q, 0)

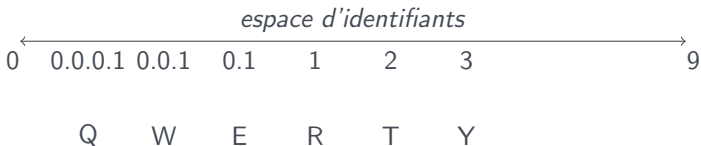


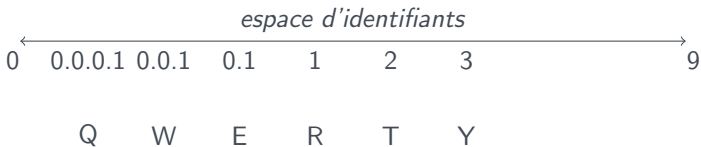


**Avec la connaissance** de la position des caractères dans la séquence finale, il est possible d'obtenir une allocation **optimale** avec des identifiants de taille **logarithmique** par rapport à la taille du document.



**Sans connaissance**, les identifiants peuvent grandir très rapidement. Une stratégie adaptée à l'édition de gauche à droite devient catastrophique lorsque l'ordre d'insertion est opposé.

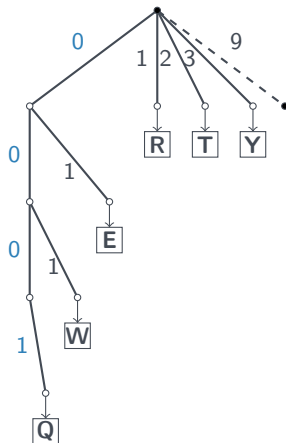




Factoriser les parties communes des identifiants construit un **arbre** représentant la même séquence.

La taille des chemins impacte les **performances** de l'éditeur et le **trafic généré** par l'éditeur.

- Il est crucial d'allouer les plus petits chemins possibles.



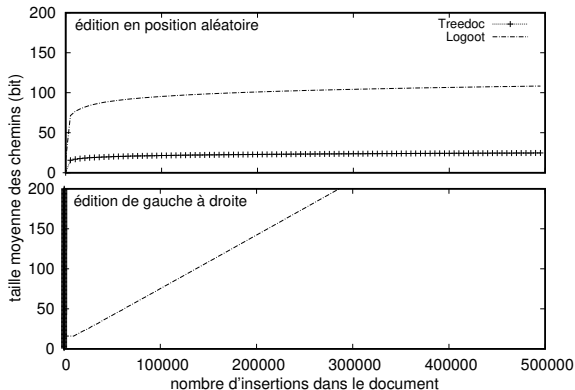
### Une fonction d'allocation sous-linéaire

Soit la séquence d'identifiants  $s(I) = id_1.id_2 \dots id_I$ , et  $s(I+1) = s(I) \cup \text{INSERT}(p, \_, n)$  où  $p, q \in s(I)$  et  $p <_{\mathcal{I}} q$ . Soit  $|s(I)|$  la taille de la représentation binaire de la séquence. La fonction INSERT doit allouer des identifiants tels que :

$$|s(I+1)| - |s(I)| < \mathcal{O}(I) \quad (1)$$

Logoot<sup>[5]</sup>, Treedoc<sup>[6]</sup> :

- Arité maximale de l'arbre reste fixe
- Adapté à l'édition de gauche à droite
- Progression linéaire des identifiants



[5] S. Weiss et al. "Logoot: a scalable optimistic replication algorithm for collaborative editing on P2P networks". In: [ICDCS. 2009](#).

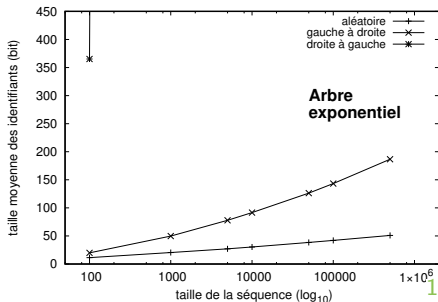
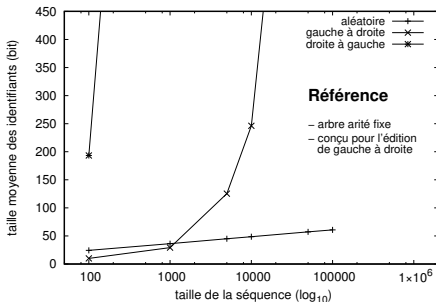
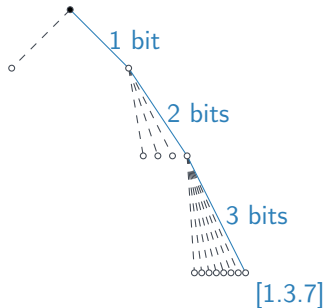
[6] N. Preguiça et al. "A commutative replicated data type for cooperative editing". In: [ICDCS. 2009](#).

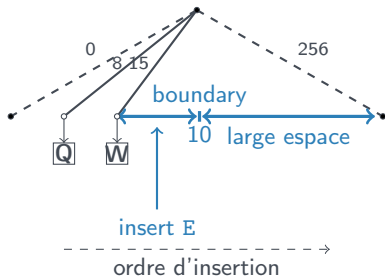
Fonction d'allocation d'identifiants :

- 1 Une structure d'arbre dont l'arité maximale augmente avec la profondeur;
- 2 Deux sous-fonctions d'allocation conçues pour gérer des comportements d'édition opposés;
- 3 Une fonction assignant à chaque profondeur de l'arbre une sous-fonction d'allocation parmi celles disponibles.

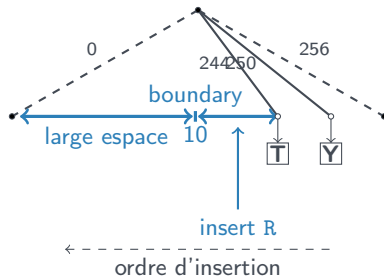
L'union des identifiants forme un arbre dont **l'arité maximale double à chaque niveau**.

- Un chemin de profondeur 3 tel que [1.3.7] nécessite 6 bits pour être représenté en mémoire.
- Un chemin de profondeur  $e$  nécessite  $\sum_{i=1}^e i = \frac{e^2+e}{2}$  bits.

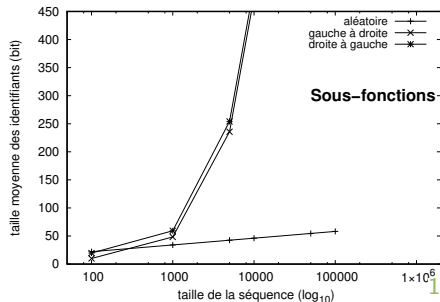
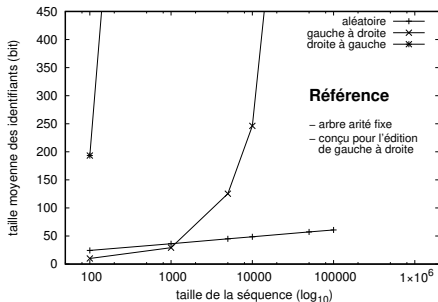




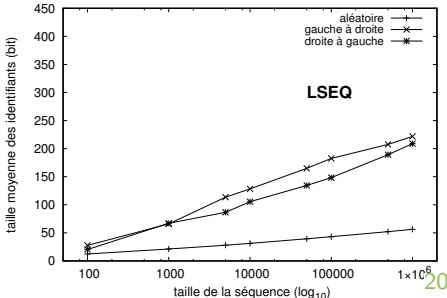
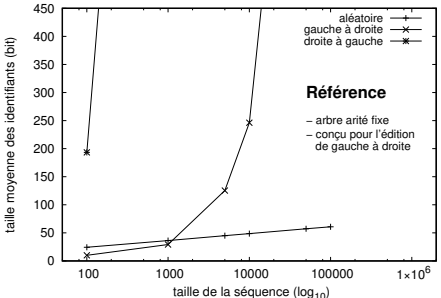
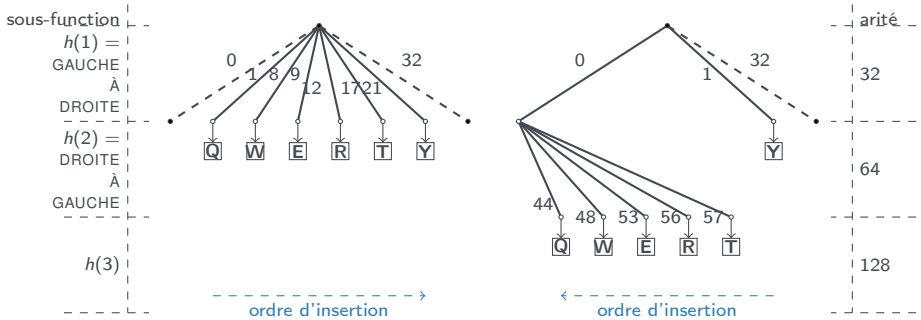
Pour l'édition de **gauche à droite**



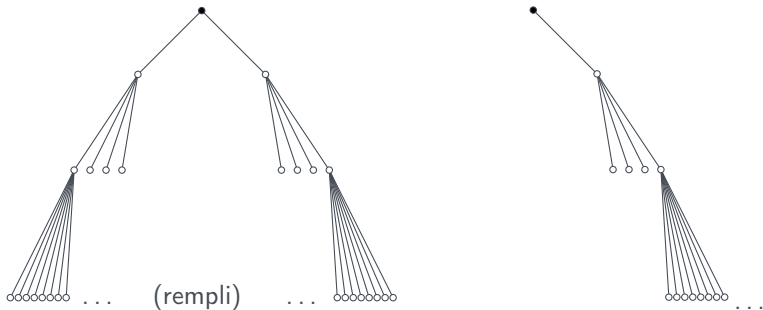
Pour l'édition de **droite à gauche**



# Document Compensation des niveaux sacrifiés







Comportements d'édition étudiés :

- Insertions en position aléatoire;
- Insertions monotone : de gauche à droite ou de droite à gauche;
- L'édition collaborative est une composition de ces comportements.

La structure analysée est un arbre

- mais d'autres structures existent offrant d'autres compromis entre les complexités temporelle et spatiale.

COMPORTEMENT D'ÉDITION	ESPACE DE LSEQ	ESPACE DE LOGOOT / TREEDOC
	IDENTIFIANTS	IDENTIFIANTS
Édition aléatoire	$\mathcal{O}(\log I)$	$\mathcal{O}(\log I)$
Édition monotone	$\mathcal{O}((\log I)^2)$	$\mathcal{O}(I)$

- Indépendante de la structure choisie pour la réplique;
- Important car impactant directement le trafic généré par les éditeurs.

## COMPORTEMENT D'ÉDITION

## TEMPS

## LOCAL

## DISTANT

INS

DEL

INS / DEL

Édition aléatoire

$\mathcal{O}(\sqrt{\log I})$

$\mathcal{O}(1)$

$\mathcal{O}(\log I + \sqrt{\log I})$

Édition monotone

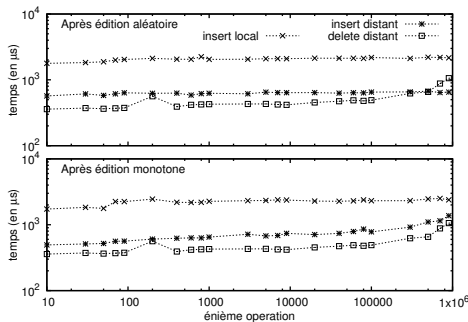
$\mathcal{O}(\log I)$

$\mathcal{O}(1)$

$\mathcal{O}((\log I)^2 + \log I)$

- Répartition entre la partie locale et la partie distante;

→ Avantageux car à 1 opération locale correspond  $N$  exécutions distantes où  $N$  est le nombre de répliques.



- LSEQ est une fonction d'allocation d'identifiants pour les séquences réparties;
- Les identifiants générés ont une complexité sous-linéaire dans le contexte de l'édition collaborative

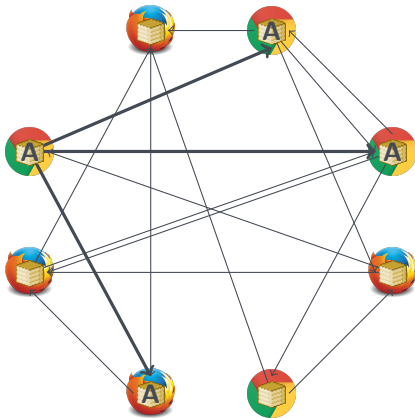
Utiliser LSEQ avec “Split”<sup>[7]</sup> pour adapter la granularité d’allocation aux chaînes de caractères.

- Moins de messages à envoyer sur le réseau;
- Moins de trafic généré par l’éditeur;
- Moins “temps réel”.

---

[7] L. André et al. “Supporting Adaptable Granularity of Changes for Massive-scale Collaborative Editing”. In: [CollaborateCom](#). 2013.

- **Chaque éditeur possède une copie locale du document;**
- Chaque éditeur est connecté à d'autres éditeurs.
- Chaque caractère tapé est directement inséré dans la copie locale;
- La modification est disséminée à l'ensemble du réseau;
- Les éditeurs recevant la modification l'appliquent.



⇒ **taille de messages** × nombre de messages

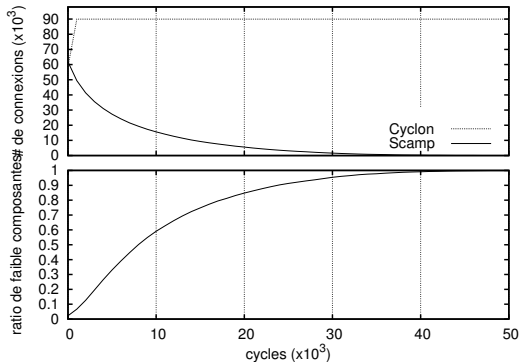
1. Introduction
2. Structure de séquences répartie à large échelle
3. Un protocole d'échantillonnage aléatoire adaptatif
4. Conclusion et perspectives

- ⇒ Le protocole de construction du réseau doit pouvoir supporter un nombre fluctuant d'utilisateurs ET les standards des navigateurs Web.





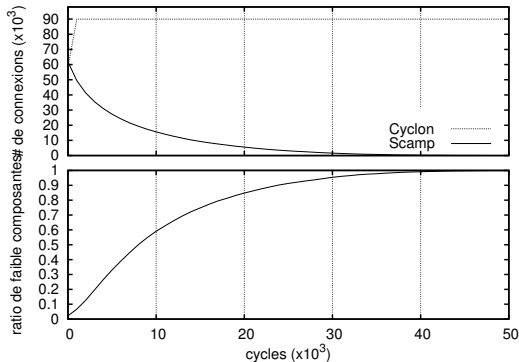
PROTOCOLE	ADAPTATIF	WEB
CYCLON <sup>[8]</sup>	✗	✓
SCAMP <sup>[9]</sup>	✓	✗
SPRAY	✓	✓



[8] S. Voulgaris et al. "CYCLON: Inexpensive Membership Management for Unstructured P2P Overlays". In: [Journal of Network and Systems Management](#) (2005).

[9] A. J. Ganesh et al. "Scamp: Peer-to-Peer Lightweight Membership Service for Large-Scale Group Communication". In: [Networked Group Communication](#). 2001.

PROTOCOLE	ADAPTATIF	WEB
CYCLON <sup>[8]</sup>	✗	✓
SCAMP <sup>[9]</sup>	✓	✗
SPRAY	✓	✓



[8] S. Voulgaris et al. "CYCLON: Inexpensive Membership Management for Unstructured P2P Overlays". In: [Journal of Network and Systems Management](#) (2005).

[9] A. J. Ganesh et al. "Scamp: Peer-to-Peer Lightweight Membership Service for Large-Scale Group Communication". In: [Networked Group Communication](#). 2001.

## Un protocole d'échantillonnage adaptatif

Soit  $t$  une unité de temps arbitraire, soit  $\mathcal{N}^t$  l'ensemble des membres non-byzantins du réseau à un instant  $t$  et soit  $P_i^t$  la vue partielle du nœud  $n_i \in \mathcal{N}^t$ . Un protocole d'échantillonnage aléatoire de paires efficace doit assurer les propriétés suivantes :

- 1 Taille des vues partielles :  $\forall n_i \in \mathcal{N}^t, |P_i^t| \approx \mathcal{O}(\ln |\mathcal{N}^t|)$
- 2 Établissement de connexion :  $\mathcal{O}(1)$

- Les vues partielles reflètent les besoins du réseau;
- Limiter le nombre de sauts pour établir une connexion.

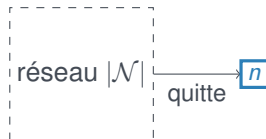
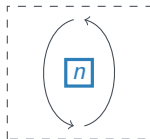
## SPRAY:

- Protocole d'échantillonnage aléatoire de paires;
- Invariant : nombre de connexions  $\approx |\mathcal{N}| \log |\mathcal{N}|$ , sans connaissance globale, i.e.,  $|\mathcal{N}|$  n'est connu de personne.
- Établissement de connexions de proche en proche.

Le réseau est **dynamique** : à n'importe quel moment, un pair peut rejoindre ou quitter le réseau.



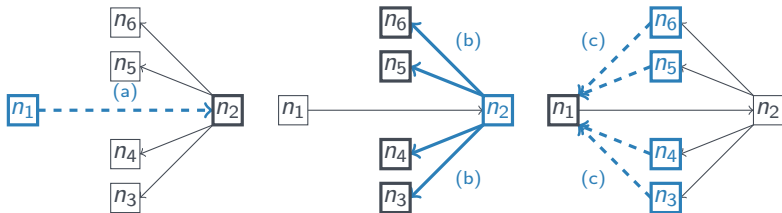
Ajoute  $1 + \log |\mathcal{N}|$   
connexions



Retire  $1 + \log |\mathcal{N}|$   
connexions

L'entrée d'un nouveau pair s'accompagne de  $1 + \log |\mathcal{N}|$  connexions.

- **1** : Le pair entrant se connecte à son contact;
- **$\log |\mathcal{N}|$**  : On suppose que le protocole répond au problème et chaque pair possède une taille de vue partielle  $\approx \log |\mathcal{N}|$ .

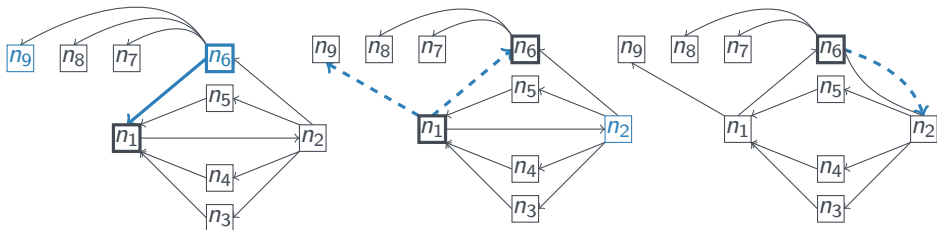


⇒ Les vues partielles sont déséquilibrées

Le nombre de connexions reste **constant**.

Les deux pairs impliqués dans l'échange périodique donne la **moitié** de leur vue partielle. Les pairs sont choisis **aléatoirement**.

→ La taille des vues converge vers la **moyenne** des deux.

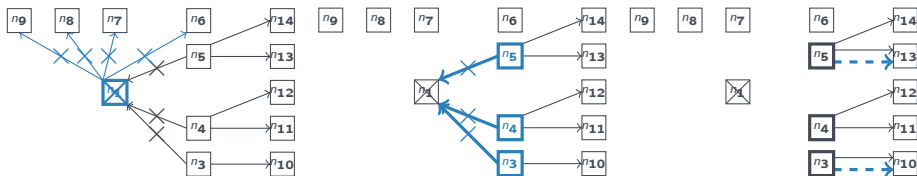


- ⇒ Équilibre rapidement la taille des vues partielles;
- ⇒ Disperse les groupes fortement connectés.

Le départ d'un pair doit engendrer la suppression de  $1 + \log |\mathcal{N}|$  connexions.

Problème : la vue sortante + la vue entrante  $\approx 2 \log |\mathcal{N}|$

- $\log |\mathcal{N}|$  : La vue partielle du pair quittant le réseau;
- 1 : Un parmi les  $\approx \log |\mathcal{N}|$  pairs détectant le départ ne recrée pas de connexion.



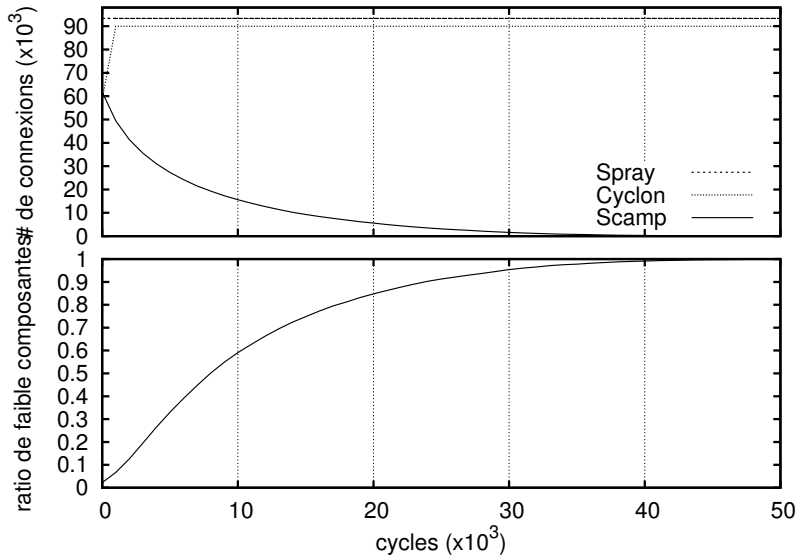
⇒ Présence de doublons assure la cohérence du nombre de connexions.

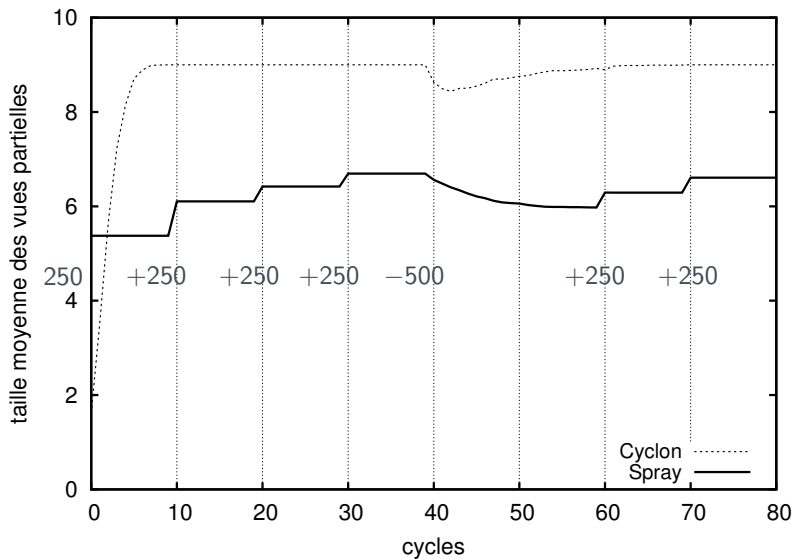
Simulations PEERSIM.

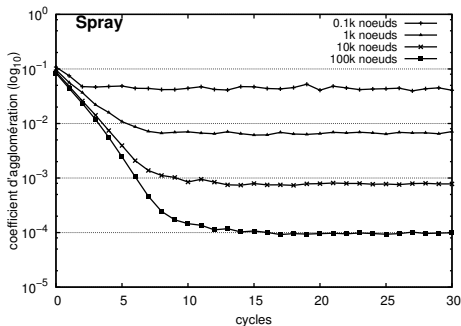
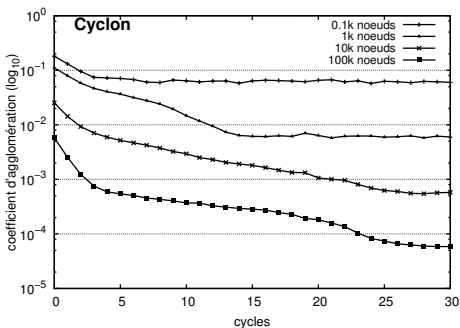
Résultats attendus :

- Convergence rapide vers un réseau possédant des propriétés similaires à celles des graphes aléatoires;
- Les doublons sont peu nombreux et n'impactent pas significativement les propriétés du réseau;
- Les protocoles basés sur SPRAY bénéficient de son adaptativité.

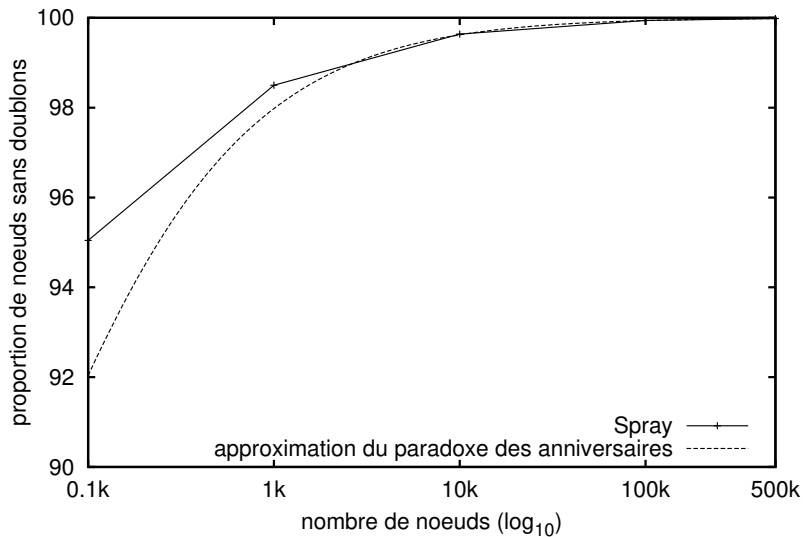


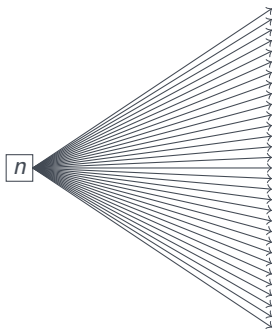






- CYCLON configuré avec
  - ▶  $\ln(1000) \approx 7$  voisins et
  - ▶ mélange de 3 voisins.





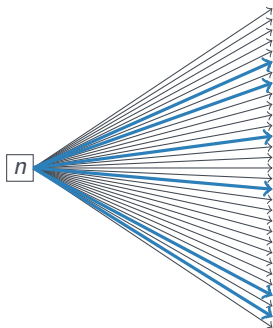
`diffuse(msg1)` `diffuse(msg2)` `diffuse(msg3)`

#### ■ CYCLON

- 1 vue partielle de 30 voisins, fanout de  $\ln(100) + 1 \approx 6$ ;
- 2 vue partielle de 30 voisins, fanout de  $\ln(100) + 3 \approx 8$ .

#### ■ SPRAY

- 1 vue partielle de  $6 \cdot \ln |\mathcal{N}|$  voisins, fanout de  $\ln(|\mathcal{N}|) + 1$ ;
- 2 vue partielle de  $6 \cdot \ln |\mathcal{N}|$  voisins, fanout de  $\ln(|\mathcal{N}|) + 3$ ;



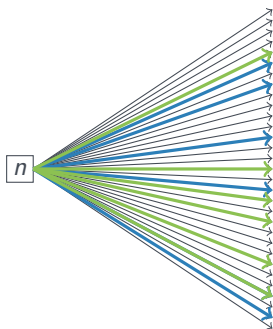
$\text{diffuse}(\text{msg}_1)$   $\text{diffuse}(\text{msg}_2)$   $\text{diffuse}(\text{msg}_3)$

## ■ CYCLON

- 1 vue partielle de 30 voisins, fanout de  $\ln(100) + 1 \approx 6$ ;
- 2 vue partielle de 30 voisins, fanout de  $\ln(100) + 3 \approx 8$ .

## ■ SPRAY

- 1 vue partielle de  $6 \cdot \ln |\mathcal{N}|$  voisins, fanout de  $\ln(|\mathcal{N}|) + 1$ ;
- 2 vue partielle de  $6 \cdot \ln |\mathcal{N}|$  voisins, fanout de  $\ln(|\mathcal{N}|) + 3$ ;



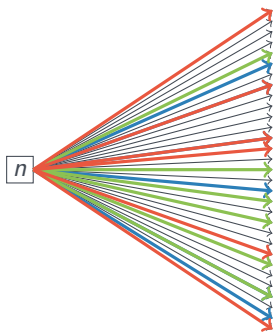
*diffuse(msg<sub>1</sub>)* *diffuse(msg<sub>2</sub>)* *diffuse(msg<sub>3</sub>)*

#### ■ CYCLON

- 1 vue partielle de 30 voisins, fanout de  $\ln(100) + 1 \approx 6$ ;
- 2 vue partielle de 30 voisins, fanout de  $\ln(100) + 3 \approx 8$ .

#### ■ SPRAY

- 1 vue partielle de  $6 \cdot \ln |\mathcal{N}|$  voisins, fanout de  $\ln(|\mathcal{N}|) + 1$ ;
- 2 vue partielle de  $6 \cdot \ln |\mathcal{N}|$  voisins, fanout de  $\ln(|\mathcal{N}|) + 3$ ;



*diffuse(msg<sub>1</sub>)* *diffuse(msg<sub>2</sub>)* *diffuse(msg<sub>3</sub>)*

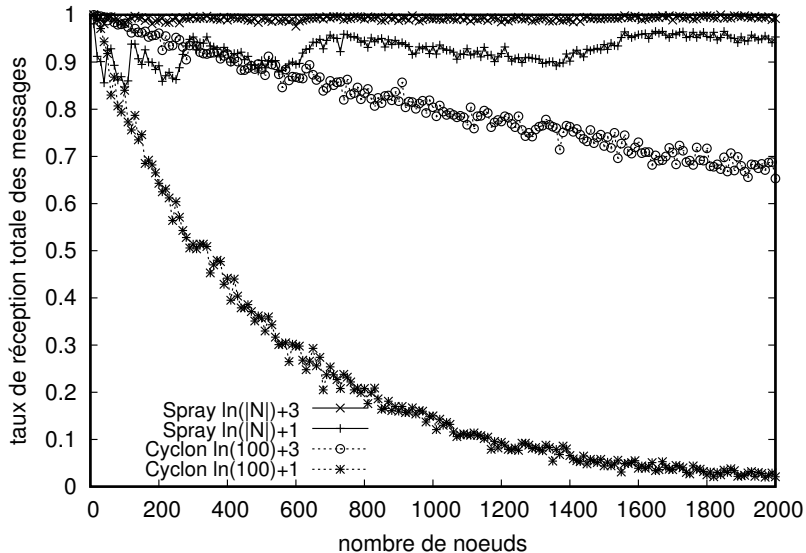
#### ■ CYCLON

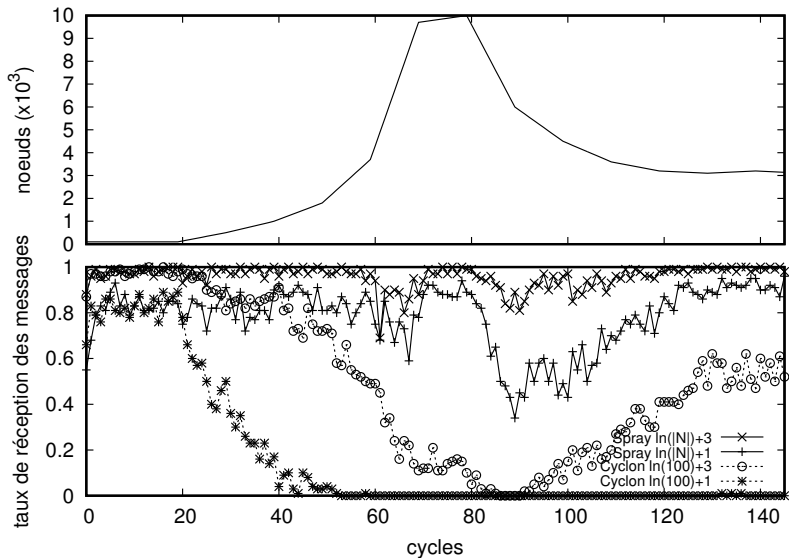
- 1 vue partielle de 30 voisins, fanout de  $\ln(100) + 1 \approx 6$ ;
- 2 vue partielle de 30 voisins, fanout de  $\ln(100) + 3 \approx 8$ .

#### ■ SPRAY

- 1 vue partielle de  $6 \cdot \ln |\mathcal{N}|$  voisins, fanout de  $\ln(|\mathcal{N}|) + 1$ ;
- 2 vue partielle de  $6 \cdot \ln |\mathcal{N}|$  voisins, fanout de  $\ln(|\mathcal{N}|) + 3$ ;







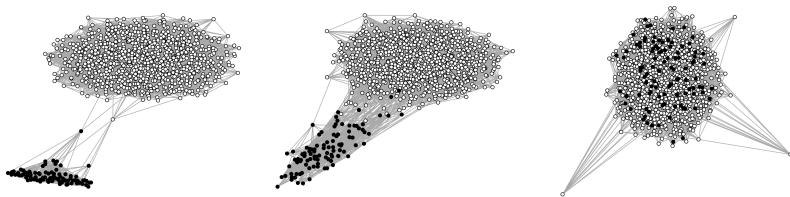
- SPRAY est un protocole d'échantillonnage aléatoire de pair adaptatif.
- SPRAY est adapté au processus complexe d'établissement de connexion de WebRTC.
- Les protocoles construits au dessus de SPRAY bénéficient de cette adaptativité.

## Les réseaux fusionnant préservent leurs propriétés

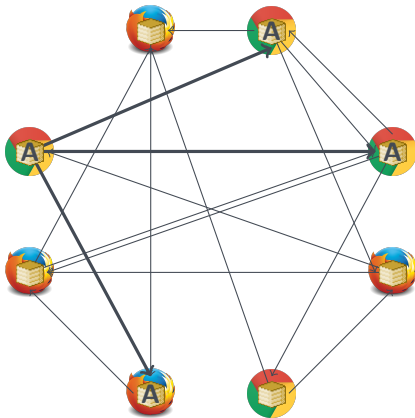
Soit  $\mathcal{N}_1, \mathcal{N}_2, \dots, \mathcal{N}_k$  des réseaux de taille arbitraire. On a :

$$\sum_{i=1}^k |\mathcal{N}_i| \ln(|\mathcal{N}_i|) < \left( \sum_{i=1}^k |\mathcal{N}_i| \right) \ln \left( \sum_{i=1}^k |\mathcal{N}_i| \right) \quad (2)$$

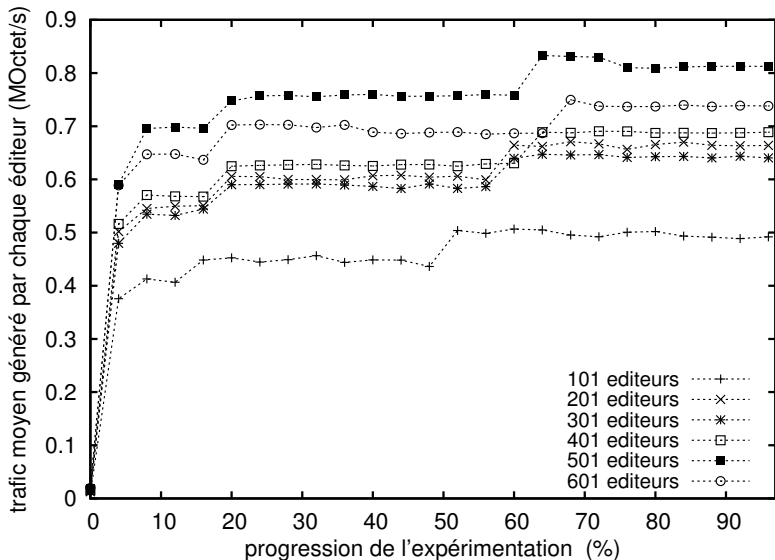
Comment adapter le nombre d'arcs effectifs (à gauche) pour qu'il atteigne le nombre d'arcs requis (à droite)?



- Chaque éditeur possède une copie locale du document;
- Chaque éditeur est connecté à d'autres éditeurs.
- Chaque caractère tapé est directement inséré dans la copie locale;
- La modification est disséminée à l'ensemble du réseau;
- Les éditeurs recevant la modification l'appliquent.



⇒ **taille de messages** × **nombre de messages**



1. Introduction
2. Structure de séquences répartie à large échelle
3. Un protocole d'échantillonnage aléatoire adaptatif
4. Conclusion et perspectives

**L'édition collaborative temps réel est-elle possible sur le Web sans l'intervention d'un tiers et sans limites quant aux dimensions du système ? ✓**

Contributions :

- 1 Fonction d'allocation d'identifiants polylogarithmique par rapport au nombre d'insertions dans la séquence;
- 2 Protocole d'échantillonnage adaptatif fournissant des vues logarithmiques par rapport à la taille du réseau supportant le processus de connexion des navigateurs Web.



Localement, CRATE utilise toujours un vecteur de  $N$  entiers où  $N$  est le nombre de personnes ayant jamais participé à l'édition.

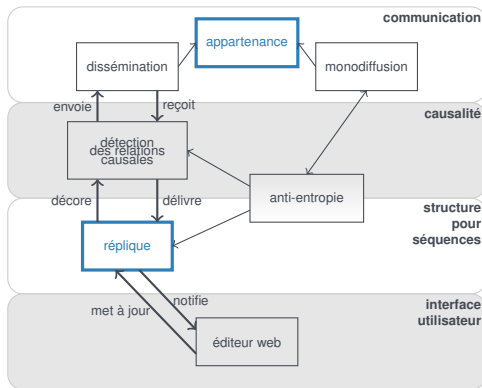
Remplacer ce vecteur de taille  $N$  par un vecteur de taille  $K$ ,  $K \ll N$  dépendant du taux de concurrence.

Intuition : sans concurrence,  $K = 1$  serait suffisant pour caractériser toutes les relations causales.

Cependant, une procédure est toujours nécessaire dans les cas ambigus.

CRATE est un éditeur collaboratif décentralisé fonctionnant dans les navigateurs.

Quelles autres applications décentralisées est-il possible de développer ?



Anonymous Turkey

Merci !

## Definition (Spécification séquentielle d'une séquence)

Soit une série d'opérations  $H$  produisant la séquence

$s(H) = \{p_1, p_2 \dots p_k\}$  avec  $p_{1..k} \in \mathcal{P}$  où  $\mathcal{P}$  est un ensemble muni d'un ordre dense  $(\mathcal{P}, <_{\mathcal{P}})$  tel que :

$\forall p \in \mathcal{P}, p_{\perp} <_{\mathcal{P}} p <_{\mathcal{P}} p_{\perp}$  et  $p_{\perp} <_{\mathcal{P}} p_1 <_{\mathcal{P}} p_2 <_{\mathcal{P}} \dots <_{\mathcal{P}} p_k <_{\mathcal{P}} p_{\perp}$ .

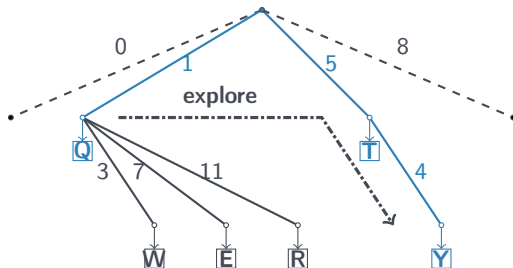
L'insertion d'un élément  $e$  en position  $i$  dans la séquence  $s(H)$  est définie de la façon suivante :

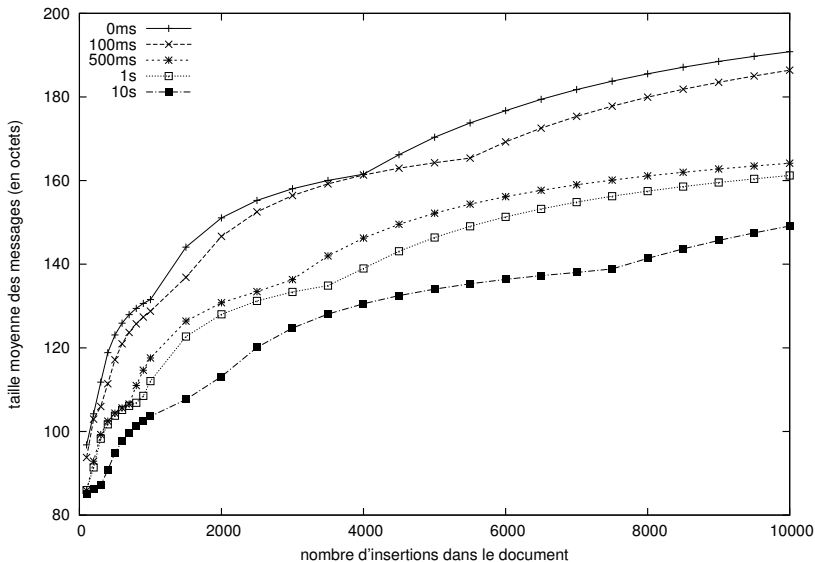
$$s(H \cup \text{INSERT}(i, e)) \rightarrow s(H) \cup \begin{cases} \{p, p_{\perp} <_{\mathcal{P}} p <_{\mathcal{P}} p_{\perp}\} & i = 0 \wedge |s(H)| = 0 \\ \{p, p_{\perp} <_{\mathcal{P}} p <_{\mathcal{P}} p_1\} & i = 0 \wedge |s(H)| > 0 \\ \{p, p_k <_{\mathcal{P}} p <_{\mathcal{P}} p_{\perp}\} & i = k \\ \{p, p_i <_{\mathcal{P}} p <_{\mathcal{P}} p_{i+1}\} & \text{sinon} \end{cases} \quad (3)$$

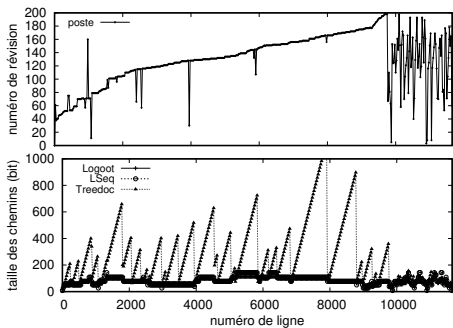
La suppression de l'élément en position  $i$  dans la séquence  $s(H)$  est définie de la façon suivante :

$$s(H \cup \text{DELETE}(i)) \rightarrow s(H) \setminus \{p_i\} \quad (4)$$

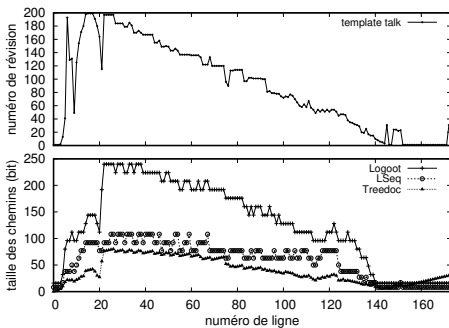
COMPORTEMENT D'ÉDITION	TEMPS LOOKUP
Édition aléatoire	$\mathcal{O}(2\sqrt{\log l})$
Édition monotone / pire cas	$\mathcal{O}(l)$



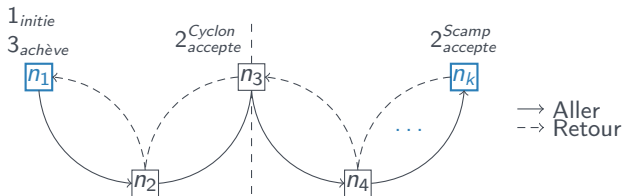




Document édité en fin



Document édité au début



$$P_E^{Scamp} = 1 - (1 - P_f)^{k+1} \quad (5)$$

$$\begin{aligned} P_{E, webrtc}^{Scamp} &= 1 - ((1 - P_f)^{2(k+1)} (1 - P_f)^{2k} \dots (1 - P_f)^2) \\ &= 1 - (1 - P_f)^{k^2 + 3k + 2} \end{aligned} \quad (6)$$



