Construction auto-stabilisante de forêts de diffusion optimale

Bertrand Ducourthial
UMR CNRS 6599 HEUDIASYC
Université de Technologie de Compiègne
Bertrand.Ducourthial@utc.fr

Sébastien Tixeuil
UMR CNRS 8623 LRI
Université de Paris-Sud
tixeuil@lri.fr

8 mai 2000

Résumé

En utilisant un modèle générique d'algorithmes tolérant les défaillances, nous obtenons un algorithme réparti adaptatif permettant de calculer on-line la forêt des meilleurs diffusions, sur la base d'un critère d'un taux par lien.

Plan

- Forêt de diffusion
- Propiétés de l'algorithme recherché

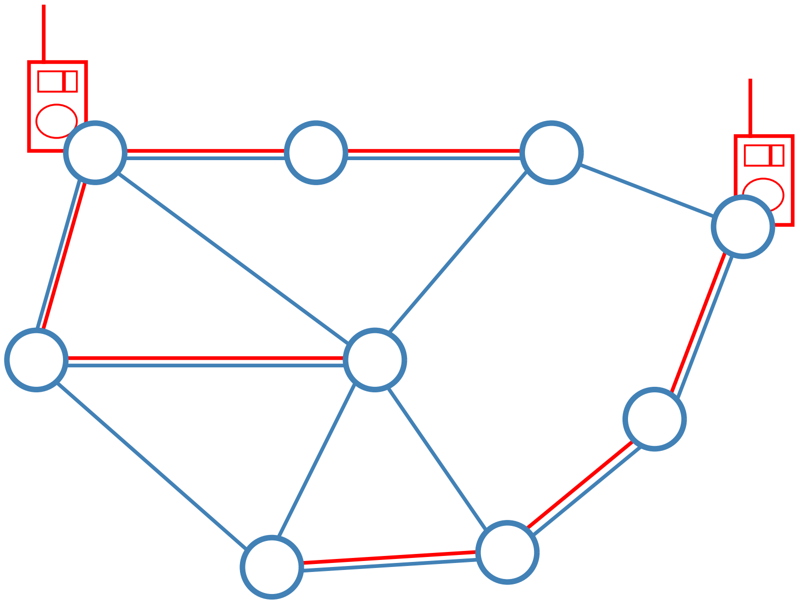
- Auto-stabilisation

- Algorithme paramétré
- Opérateurs

- Application
- Exemple

- Conclusion

Forêt de diffusion



- **Contexte**
  - Émission radio-phonique, serveurs ftp, ...
  - Réseaux filaires ou non
- **Forêt**
  - tous les nœuds sont concernés
    $\rightsquigarrow$ broadcast multi-source (réseau logique)
  - même information détenue par plusieurs sources
- **But**
  - choix de la meilleure source et du meilleur chemin
  - $\rightsquigarrow$ forêt couvrante de diffusion optimale
  - réactivité

Calcul de la forêt

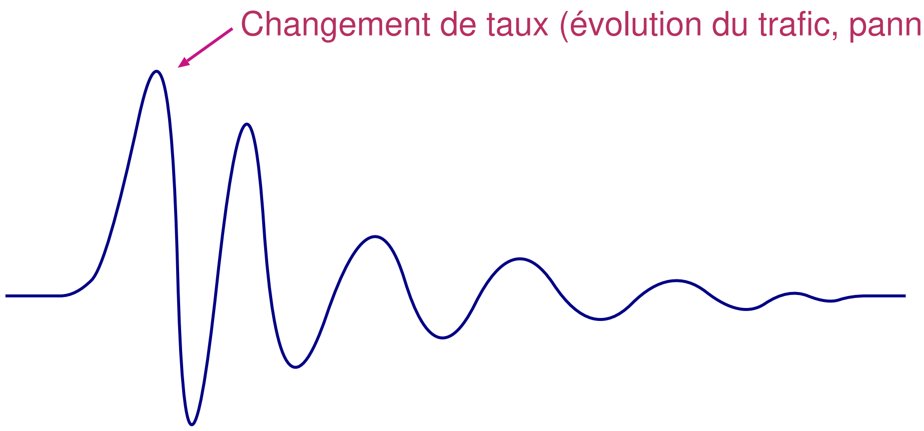
- Critère
  - distance (réelle ou nombre de nœuds)
  - débit
  - taux par lien :
    fiabilité, perte de paquets, *etc.*
- Algorithme
  - taux par lien $\rightsquigarrow$ taux par chemin
    $\rightsquigarrow$ meilleur chemin $\rightsquigarrow$ meilleure forêt
  - algorithme réparti
- Réactivité
  - évolution des taux
  - modification du réseau
  - pannes (liens, nœuds)
  - technique auto-stabilisante

$\rightsquigarrow$ *algorithme réparti auto-stabilisant pour la forêt des meilleurs taux*

Auto-stabilisation : définition

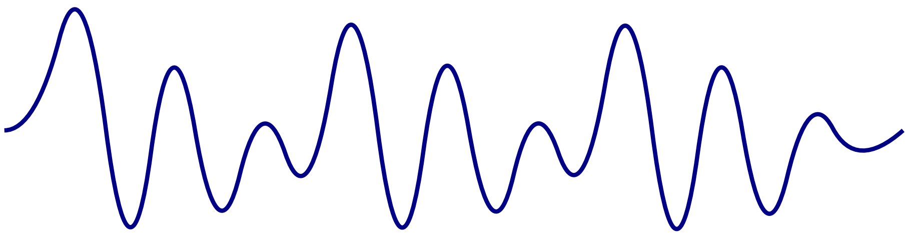
- Fautes temporaires
  - faute $\leadsto$ modification des données
  - registres, RAM susceptibles d'être corrompues
  - ROM et code non altérables

  $\leadsto$ modèle général pour supporter les modifications des taux (évolution du trafic, du réseau, panne...)
- Rétablissement après panne
  - une fois que les fautes cessent, le système retrouve un comportement correct en un temps fini [Dijkstra 74, Schneider 93]
  - initialisation quelconque



Auto-stabilisation : limitation

- Temps de convergence
  - valeur erronée sur un nœud
    $\rightsquigarrow$ tous ses descendants sont affectés
  - si absence de circuits, $O(D)$
  - topologie quelconque $O(D + |\mathbb{S}|)$
- Inconvénient
  - forêt mouvante



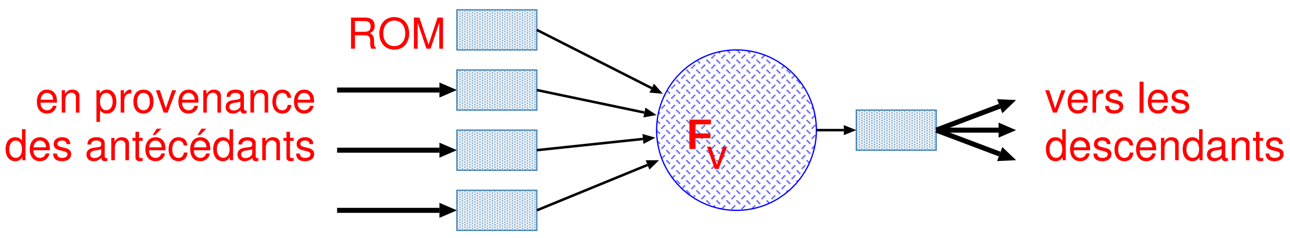
- Parade : espacer les changements de taux
  - modification du réseau, pannes $\rightsquigarrow$ ok
  - évolution du trafic
    $\rightsquigarrow$ moyennes mobiles, seuil d'alerte

État de l'art

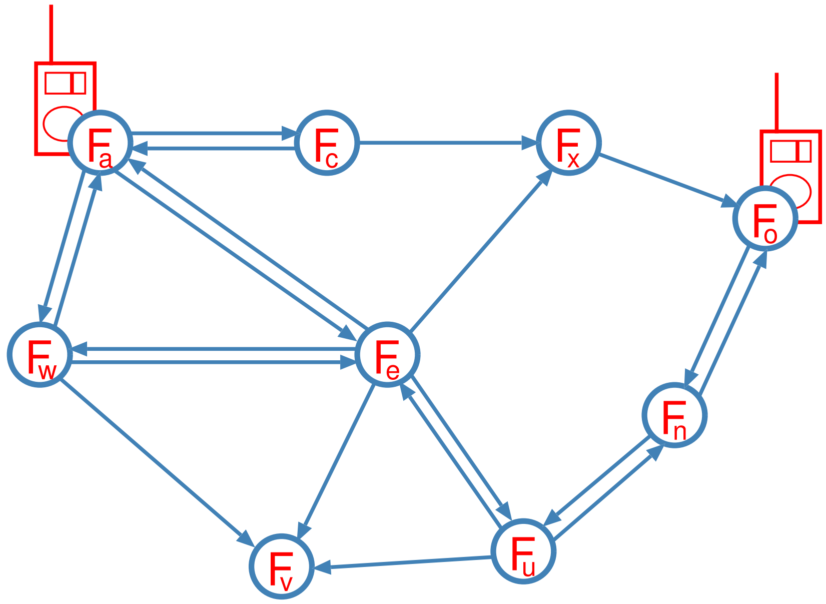
- Auto-stabilization
  - graphes non orientés [Dolev Israeli Moran 93, Afek et all 90]
  - graphes orientés fortement connexe [Alstein et all 94]
    $\leadsto$ topologie virtuelle connue ($e.g.$ arbre) [Alima et all 98]
- Tâches silencieuses
  - convergence vers une configuration et non une exécution
  - existence d'une configuration terminale [Dolev Gouda Schneider 96]
- Tâches silencieuses auto-stabilisantes
  - graphes orientés, non fortement connexe
  - mais sans circuit $\leadsto$ DAG [Bourgon et all 95]
- Notre travail
  - tâches silencieuses auto-stabilisantes sur tout graphe orienté
  - s'il existe un opérateur spécifique pour instancié l'algorithme

Algorithme paramétré

Algorithme



sortie $\leftarrow$ $\mathbf{F}$ (ROM,entrées)

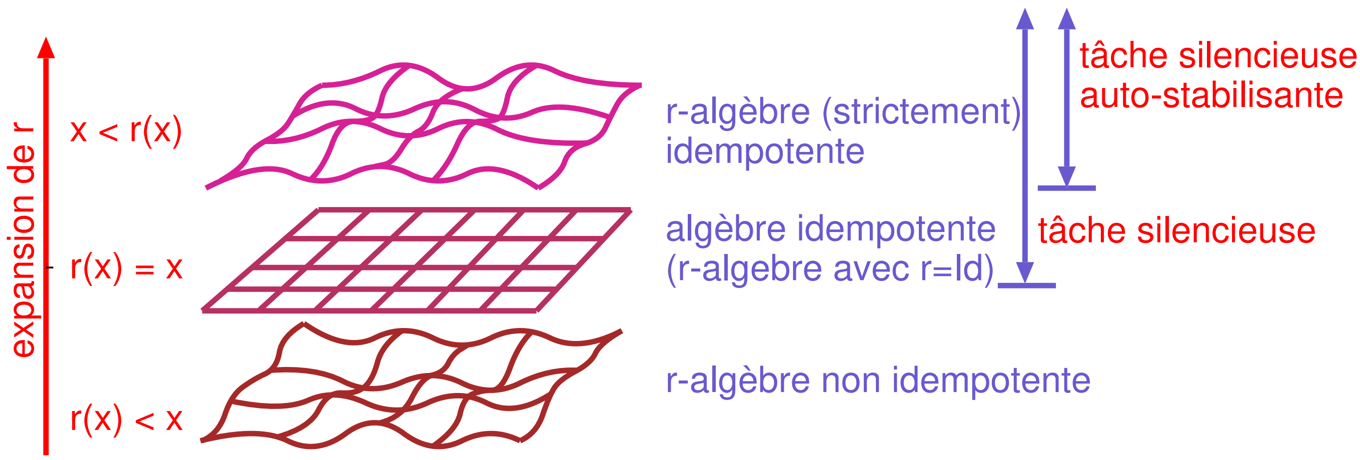


Paramètres (*i.e.* opérateurs F)?

- tâche silencieuse (*i.e* convergence vers une configuration)
- auto-stabilisante (*i.e.* convergence en dépis des pannes)
- conforme aux spécifications (*i.e.* forêt meilleurs taux)

r-opérateurs

- Infimum (s-opérateurs)
  - infimum $\rightsquigarrow$ tâches silencieuses [Tel91]
    ($e.g.$ min(x,y))
  - infimum $\Rightarrow$ non auto-stabilisant [OPODIS98]
- r-opérateurs
  - généralisation des infimums [SIROCCO98]
    ($e.g.$ minc$(x,y) =$ min$(x,y+1)$)
  - r-opérateurs strict. idempotent $\rightsquigarrow$ auto-stabilisation
    atomicité read-write si ordre total [OPODIS98]
    ordre partiel si démon complètement réparti [TCS]



Conditions sur l'opérateur

$$\min(x_0, x_1 + d_1, \ldots, x_n + d_n)$$

- Conditions
  1. $f \ : \mathbb{S}^n \mapsto \mathbb{S}$
     $f(x_0, \ldots, x_n) = x_0 \oplus r_1(x_1) \oplus \cdots \oplus r_n(x_n)$
  2. $\mathbb{S}$ est fini
  3. $\oplus$: infimum $\rightsquigarrow$ ordre total $\preceq_\oplus$
  4. $r_i(x \oplus y) = r_i(x) \oplus r_i(y)$ (homomorphismes)
  5. $x \prec_\oplus r_i(x)$
- Résultat du nœud v

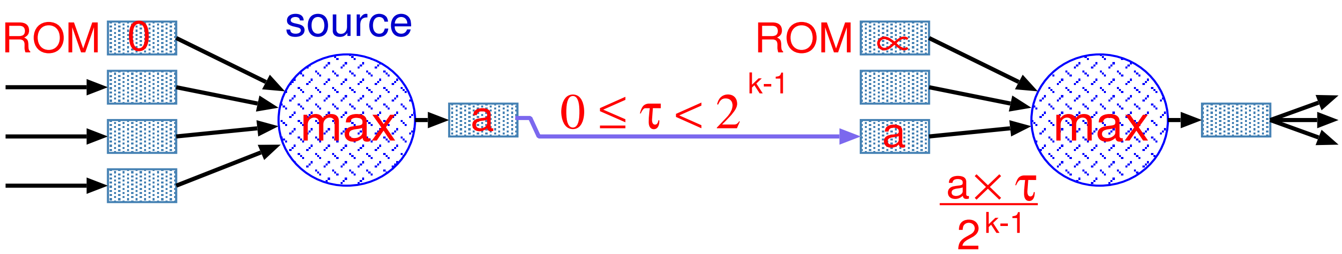
$$\bigoplus \{r_{P_{u \to v}}(\mathrm{ROM}_u),\ u \text{ antécédant de } v \text{ dans le réseau}\}$$

- Utilisation
  - initialisation des ROM : $\perp\mathbb{S}$ (sources) ou $\top\mathbb{S}$
  - détermination des meilleurs antécédants de chaque nœud
  - choix d'un meilleur antécédant $\rightsquigarrow$ forêt

Application

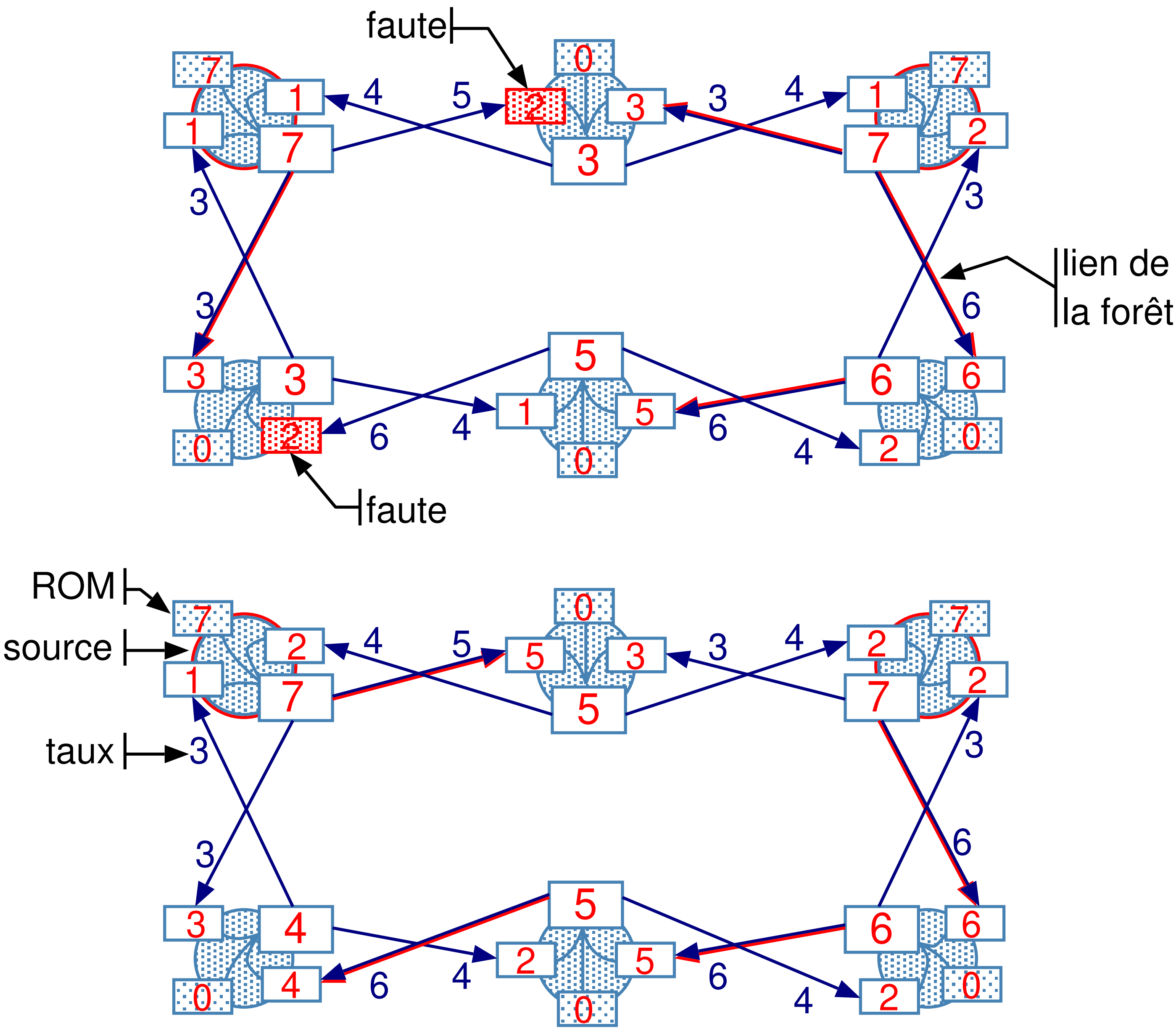
- Taux
  - variables de k bits $\rightsquigarrow \mathbb{S} = \{0, \ldots, 2^k - 1\}$
  - $0 \leq \tau < 2^k - 1$

    ligne hors-service — ligne parfaite
- Opérateur
  - taux d'un chemin : produit $\rightsquigarrow r(x) = x \times \tau / (2^k - 1)$
  - meilleur chemin : maximum $\rightsquigarrow \oplus \equiv \max$



- Conditions vérifiées
  - homomorphisme $\max(x,y) \times \tau = \max(x \times \tau, y \times \tau)$
  - stricte idempotence $x \prec_{\max} r(x)$ *i.e.* $x > r(x)$

Exemple



Conclusion

- Problème
  - diffusion multi-source
  - recherche d'une meilleure forêt de diffusion
  - critère des taux
  - réactivité
- Solution
  - algorithme réparti
  - auto-stabilisation
- Cadre formel utilisé
  - algorithme simple, paramétré par un opérateur
  - conditions sur l'opérateur pour l'auto-stabilisation
  - r-opérateurs strictement idempotent
- Application
  - trouver un opérateur satisfaisant les conditions et les spécifications
  - $\max(x_0, x_1 \times \tau_1, \ldots, x_n \times \tau_n)$