Conception et Planification de Réseaux

Optimisation des Métriques de Routage IP

Olivier BRUN et Jean-Marie GARCIA

Plan

- Introduction
- Modélisation du problème
- Algorithme de résolution
- Résultats
- Conclusion

Introduction

Contexte des réseaux

- **Evolution de l’Internet vers une architecture de communication globale**
  - **Implique la gestion de services multimédia beaucoup plus sophistiqués que le traditionnel service au mieux (best effort) du monde IP**
  - **Les futurs réseaux IP à intégration de services devront être capables de supporter un spectre très large d’exigences de QoS**
    - **bornes sur le délai de bout-en-bout des communications, sur leur bande-passante, sur leur gigue, et/ou sur leur taux de pertes**
    - **Contrat de QoS entre un fournisseur d’accès Internet (ISP) et ses clients : SLA (Service LevelAgreement)**

Contexte des réseaux

- **Les opérateurs doivent adapter en permanence leurs infrastructures de communication**
  - **Nouvelles exigences de QoS, augmentation continue du traffic Internet, impératifs de plus en plus fort de sécurisation.**
  - **Le contexte concurrentiel qui induit des marges bénéficiaires réduites, ne permet plus d'améliorer les performances d'un réseau par un surdimensionnement excessif des équipements.**
    - **Les opérateurs et les ISPs s'intéressent de plus en plus aux techniques d'ingénierie de trafic, moins coûteuse et généralement basées sur des technologies déjà existantes,**
    - **Eviter les phénomènes de congestion du trafic et les dégradations du service qui en résultent.**

Ingéniérie de trafic

- **Concept clé : optimisation du routage**
  - **Utilisation plus efficace des ressources réseaux existantes en adaptant le routage aux trafics transportés.**
  - **Utilisation plus efficace des ressources réseaux existantes en adaptant le routage aux trafics transportés.**
    - **en particulier pour les réseaux ayant une distribution de trafic non uniforme.**
  - **Les spécificités de routage IP rendent très difficile l'ingénierie de trafic dans les réseaux de l'Internet.**

Le Routage IP

- **Calcul des routes dans un réseau IP**
  - **Effectué de manière décentralisée par des protocoles de routage.**
    - **OSPF (Open Shortest Path First) et ISIS (Intermediate System to Intermediate System) sont actuellement les deux protocoles de routage intra-domaine les plus utilisés dans l'Internet.**
  - **Un flot est routé suivant un plus court chemin vers la destination**
    - **partage de charge équitable en un noeud si plusieurs liens sortants sont sur des plus courts chemins vers la destination.**
  - **Le poids des liens, et par conséquent les plus courts chemins utilisés par le routage, peuvent être changés par l'administrateur du réseau.**
    - **Minimisation du nombre de hops (poids à 1) ou du délais de propagation (poids proportionnel à la distance physique).**
    - **Heuristique standard recommandée par Cisco : prendre des poids inversement proportionnel à la capacité des liens de transmission.**
    - **Mauvaise utilisation des ressources, variations de trafic, pannes.**

Optimisation des Métriques IP

- ## Historique
  - **On a longtemps considéré que des protocoles comme OSPF ou ISIS n'étaient pas assez flexibles pour une ingénierie de trafic efficace.**
    - **Une des raisons de l'introduction des technologies de commutation d'étiquettes MPLS (Multi-Protocol Label Switching) qui permettent de fixer le chemin de chaque flot dans le réseau.**
  - **Récemment, la communauté scientifique, sous l'impulsion de Fortz et Thorup en particulier, s'est intéressée de plus en plus au problème d'optimisation des métriques de routage IP**
    - **Problème NP-difficile.**
    - **Une affectation suffisamment intelligente des métriques de routage permet souvent d'obtenir un routage aussi performant que le routage MPLS sans coût financier supplémentaire.**
    - **Performances proches de celles d'un routage optimal par partage de charge dans certains cas.**

Optimisation des Métriques IP

Objectifs

- **Améliorer les performances globales d'un réseau IP**
  - en réduisant la congestion des liens,
  - en minimisant les délais et les taux de pertes de bout-en-bout des flots.
- **Dégager plus de bande-passante résiduelle sur les chemins empruntés par les flots pour :**
  - augmenter la capacité du réseau à tolérer sans dégradation forte du service des variations brusques du trafic sur une courte période de temps,
  - réduire la fréquence des modifications du plan de routage du à des augmentations prévisibles de trafic.
- **Augmenter la robustesse du réseau en prenant en compte la défaillance des équipements dans le processus d'optimisation.**

Modélisation du problème

Modélisation du problème

- **Plusieurs étapes :**
  - **Modélisation du réseau,**
    - **Modélisation de la topologie,**
    - **Définition d'une solution au problème.**
  - **Modélisation du trafic,**
  - **Formulation du coût d'une solution,**

Modélisation du réseau

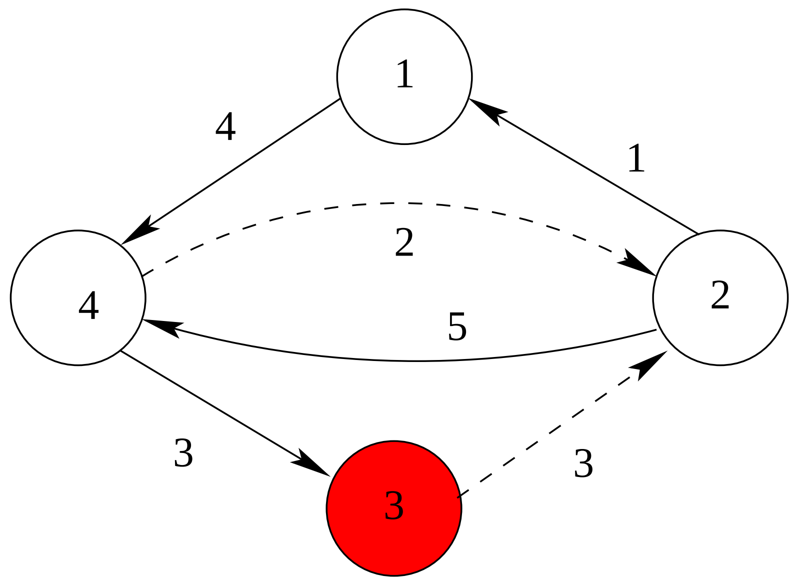
- **Un réseau IP est modélisé sous la forme d'un graphe orienté**
  - les noeuds correspondent aux routeurs et les arcs aux interfaces de communication des routeurs.
  - On note N le nombre de noeuds du graphe et M le nombre d'arcs du graphe.
  - A chaque arc a du graphe est associé un poids w_a qui représente la métrique de l'interface correspondante et qui peut prendre toute valeur dans l'intervalle $\Omega = [1, 2^{16} - 1]$.
  - A chaque arc a du graphe est associé une capacité C_a en paquets/seconde.

Solution du problème

- **Une solution admissible du problème d'optimisation des métriques de routage IP est un vecteur $w = (w_1, \ldots, w_M) \in \Omega^M$.**
  - **Le nombre de solutions du problèmes est donc environ $2^{16\,M}$.**
  - **Pour les réseaux qui nous intéressent, les valeurs de M vont se situer entre quelques dizaines et quelques milliers.**
  - **Etant donné une solution admissible $w = (w_1, \ldots, w_M)$, on peut calculer par un algo. de plus courts chemins (PCC) :**
    - $D_i^x(w)$ **est la distance du noeud i à la destination x,**
    - $\delta_{i,j}^x(w) = 1$ **si l'arc (i, j) est sur un PCC vers la destination x et 0 sinon,**
    - $n_i^x(w) = \sum_j \delta_{i,j}^x(w)$ **représente le nombre d'arcs sortants du noeud i qui sont sur un PCC vers la destination x.**

Modélisation des Plus Courts Chemins

- **Exemple de Modélisation des PCC (vers x=3)**



$$D_1^3 = 7,\ D_2^3 = 8,\ D_3^3 = 0,\ D_4^3 = 3$$

$$\delta_{1,4}^3 = 1,\ \delta_{2,1}^3 = 1,\ \delta_{2,4}^3 = 1,\ \delta_{3,2}^3 = 0,\ \delta_{4,2}^3 = 0,\ \delta_{4,3}^3 = 1$$

$$n_1^3 = 1,\ n_2^3 = 2,\ n_3^3 = 0,\ n_4^3 = 1$$

Modélisation du Trafic

- **Le réseau écoule un ensemble de K flots de communication.**
  - **Chaque flot $k = 1, \ldots, K$ est caractérisé par sa source $s(k)$, sa destination $t(k)$ et sa demande en bande-passante d^k.**
  - **Chaque flot est un couple origine-destination unique ($K \leq N, (N-1)$)**
  - **Aggrégation des trafics ayant mêmes origine-destination : même routage,**
  - **La prise en compte de DiffServ et du routage par ToS (Type Of Service) peuvent être vues comme des extensions de l'algorithme présenté**

Modélisation du Trafic

Notations

- λ_i^x : **le trafic direct émis de i vers la destination x**

$$\lambda_i^x = \sum_{\substack{k, \\ s(k)=i,\ t(k)=x}} d^k$$

- $\gamma_i^x(w)$: **le trafic reçu au noeud i, direct et en transit, pour la destination x.**

$$\gamma_i^x(w) = \lambda_i^x + \sum_{j \neq x} \frac{\delta_{j,i}^x(w)}{n_j^x(w)} \gamma_j^x(w)$$

- **La charge du lien (i,j) s'écrit alors :**

$$Y_{i,j}(w) = \sum_{x=1}^{N} \frac{\delta_{i,j}^x(w)}{n_i^x(w)} \gamma_i^x(w)$$

Modélisation du problème

- **Pour un vecteur de métriques $w = (w_1, \dots, w_M)$,**
  - **Par un algorithme de PCC, on peut calculer les variables $\delta^x_{i,j}(w)$ et $n^x_i(w)$ représentant les PCC pour la destination x**
  - **On en déduit le trafic $\gamma^x_i(w)$ au noeud i pour chaque destination x**

$$\gamma^x_i(w) = \lambda^x_i \; + \sum_{j \neq x} \frac{\delta^x_{j,i}(w)}{n^x_j(w)} \gamma^x_j(w)$$

  - **On obtient ainsi le trafic $Y_{i,j}(w)$ sur chaque lien (i, j)**

$$Y_{i,j}(w) = \sum_{x=1}^{N} \frac{\delta^x_{i,j}(w)}{n^x_i(w)} \gamma^x_i(w)$$

  - **Quel est le coût associé à cette charge des liens ?**

Formulation du Coût d'une Solution

Coût additif

$$\Phi(w) = \sum_{l=1}^{M} \Phi_l(w)$$

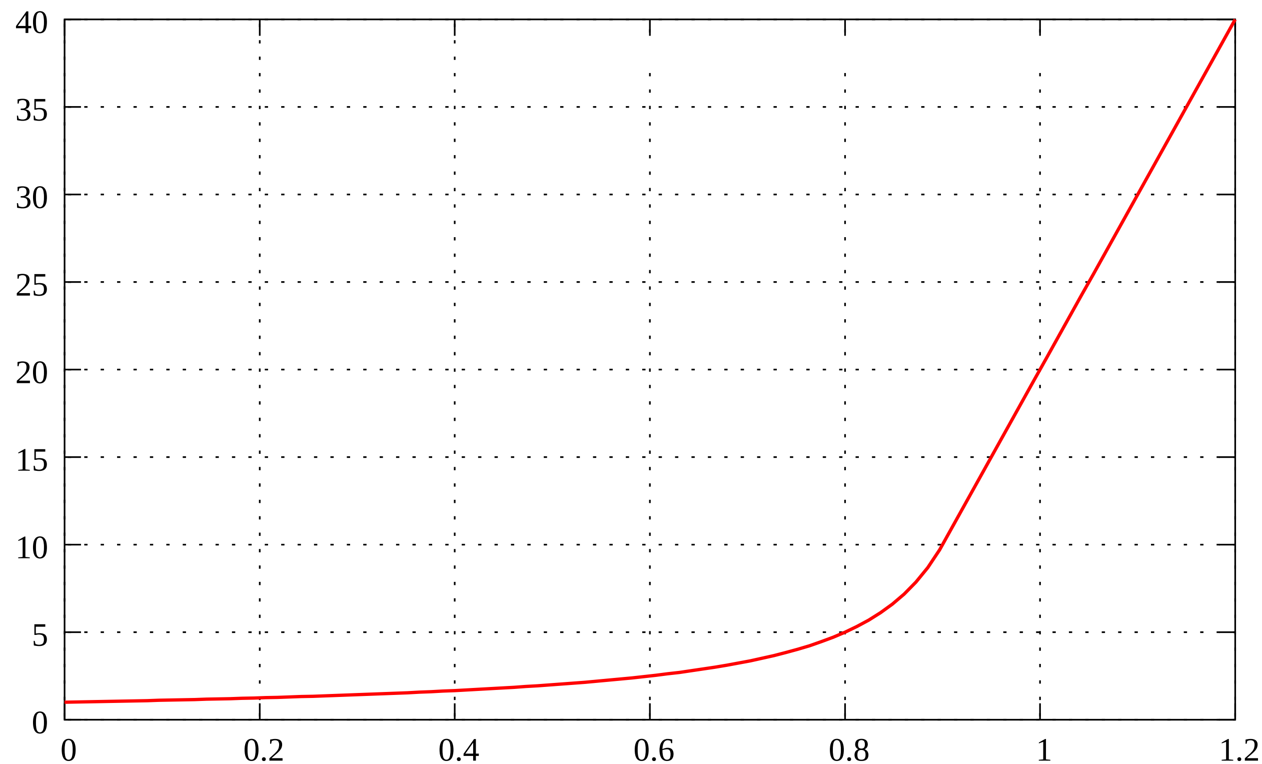
- $\Phi_l(w)$ **représente la contribution du lien l au coût global**
- **On peut prendre par exemple le délais moyen de séjour M/M/1 :**

$$\Phi_{i,j}(Y_{i,j}) = \begin{cases} \frac{Y_{i,j}}{C_{i,j}-Y_{i,j}} & \text{si } Y_{i,j} \leq \overline{Y}_{i,j} \\ \frac{\overline{Y}_{i,j}}{C_{i,j}-\overline{Y}_{i,j}} + K_2 \frac{C_{i,j}}{[C_{i,j}-\overline{Y}_{i,j}]^2} \left[Y_{i,j} - \overline{Y}_{i,j}\right] & \text{si } Y_{i,j} > \overline{Y}_{i,j} \end{cases}$$

  - $\overline{Y}_{i,j} = K_1\, C_{i,j}$ **est un seuil de charge (**$K_1 = 0.9$ **par exemple)**
  - **Le coefficient** $K_2 > 1$ **est utilisé pour associer un coût aux solutions non admissibles au sens des capacités (**$Y_{i,j} \geq C_{i,j}$**).**
  - **La pénalité appliquée dans le cas** $Y_{i,j} > \overline{Y}_{i,j}$ **correspond à la dérivée du nombre moyen de paquets au seuil de charge** $\overline{Y}_{i,j}$ **multipliée par le coefficient** K_2**.**

Coût d’une solution

- **Allure du coût précédent**



$K_1 = 0.9$, $K_2 = 1$ et $C_{i,j} = 1$.

Formulation du problème

- **Le problème s'écrit :**

$$\min_{w \in \Omega^M} \sum_{(i,j)} \Phi_{i,j}\left[Y_{i,j}(w)\right]$$

avec,

$$Y_{i,j}(w) = \sum_{x=1}^{N} \frac{\delta_{i,j}^{x}(w)}{n_i^x(w)} \gamma_i^x(w) \quad \forall i,j = 1 \dots N \tag{1}$$

$$\gamma_i^x(w) = \lambda_i^x + \sum_{j \neq x} \frac{\delta_{j,i}^{x}(w)}{n_j^x(w)} \gamma_j^x(w) \quad \forall i,x = 1 \dots N \tag{2}$$

Les paramètres $\delta_{i,j}^{x}(w)$ et $n_i^x(w)$ étant obtenus directement à partir du vecteur de métriques w par résolution d'un problème de PCC.

Algorithme d'Optimisation des métriques IP

Méthodes de Recherche Locale

- **Soit le problème d'optimisation suivant :**

$$\begin{cases} \text{Minimiser :} & F(x) \\ \text{sous la contrainte :} & x \in X \subset \mathbb{R}^n \end{cases}$$

  - **On suppose que F est une fonction quelconque définie de $\mathbb{R}^n$ vers $\mathbb{R}$ et X est un sous-ensemble discret de $\mathbb{R}^n$.**
  - **On définit $V(x) \subset X$, le voisinage d'une solution admissible x**
  - **Partant d'une solution initiale x^0, les techniques de recherche locale vont générer une suite de solutions $x^1, x^2, \ldots$ telles que :**

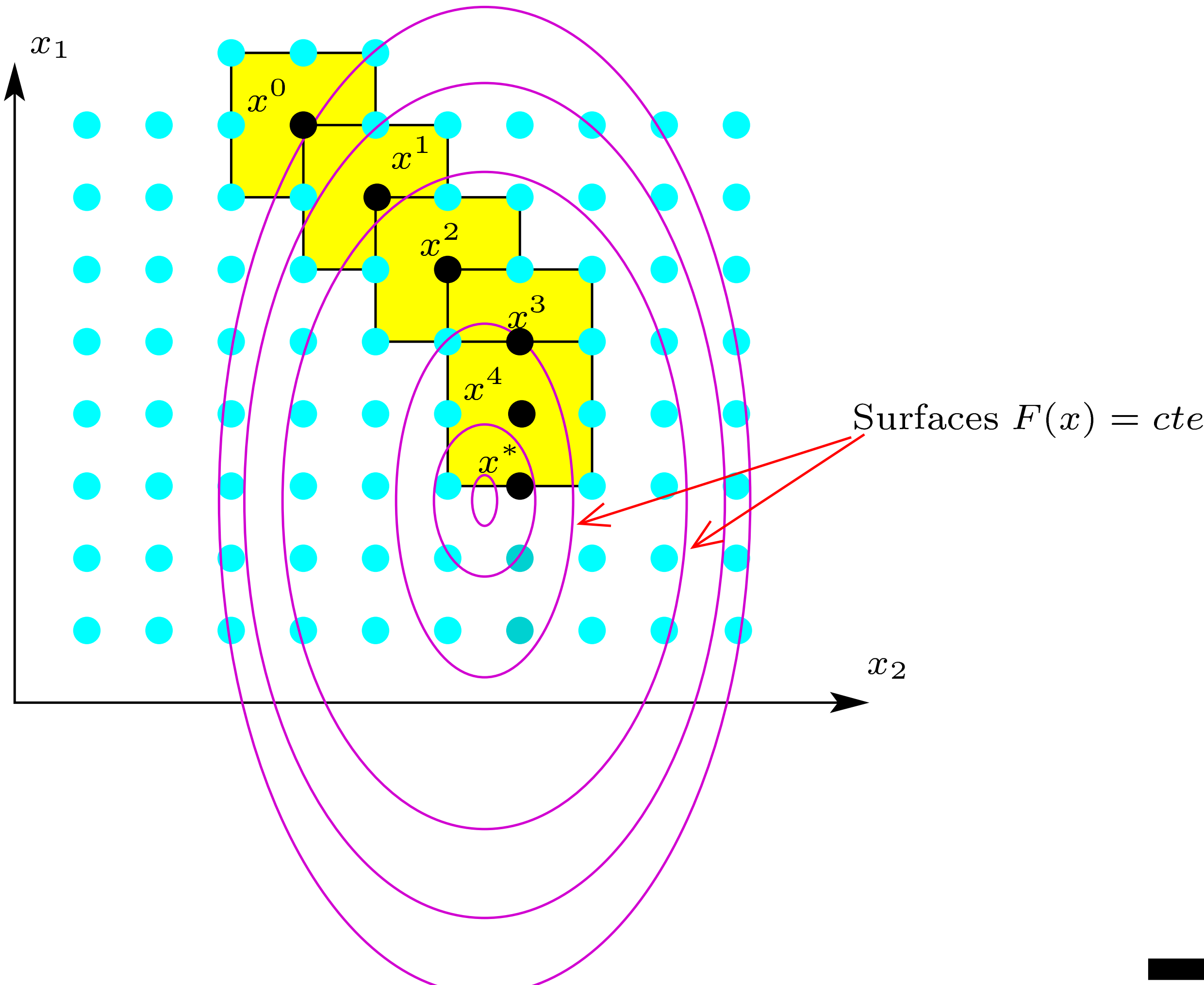
$$\forall i, \quad \begin{cases} x^{i+1} \in V(x^i) \\ F(x^{i+1}) < F(x^i) \end{cases}$$

Méthodes de Recherche Locale

- **Propriétés**
  - Sous des hypothèses de convexité et de connexité par rapport à la structure de voisinage choisie, la recherche converge avec une décroissance monotone du coût $F(x)$ vers la solution optimale x^*.
  - Si le problème n'est pas convexe, en fonction de la solution initiale x^0, la méthode peut converger vers une solution $x^n \neq x^*$ telle que $F(x) \geq F(x^n), \forall x \in V(x^n)$. On peut accepter une solution x^{n+1} telle que $F(x) \geq F(x^n)$ pour sortir de cet optimum local.
  - Les différentes méthodes de recherche locale se distinguent par la technique de génération de x^{i+1} à partir de x^i, et par la technique employée pour sortir des optima locaux.
    - recuit simulé (simulated annealing), recherche tabou (tabu search), etc.

Méthodes de Recherche Locale

Illustration du principe



Algorithme d’Optimisation des Métriques IP

Principe

- **Algorithme de recherche locale.**
  - Le voisinage d’une solution contient M solutions.
  - La solution associée au lien i est obtenue en augmentant la métrique de ce lien de la quantité minimale permettant de dévier du trafic de ce lien.
  - Les métriques ne peuvent qu’augmenter.
- **Calcul du coût de la solution associée au lien i**
  - Déterminer l’augmentation de métrique permettant de dévier du trafic du lien i,
  - Effectuer la modification et re-calculer les PCC,
  - Re-propager les trafics et calculer le coût.
- **Points clés de l’algorithme**
  - Structure de voisinage,
  - Algorithme de PCC dynamique,
  - Propagation dynamique.

Structure de Voisinage

Définition de la structure de voisinage

$$V(w) = \{w^1, w^2, \ldots, w^M\}$$

où :

$$w^i = (w_1, w_2, \ldots, w_i + \Delta_i, \ldots, w_M)$$

avec :

$$\Delta_i = argmin_{\Delta \geq 1} [Y_i(w_1, \ldots, w_i + \Delta, \ldots, w_M) < Y_i(w)]$$

- **Le voisinage d'une solution w contient exactement M solutions.**
- **La solution associée au lien i est obtenue en augmentant la métrique de ce lien de la quantité minimale permettant de réduire la charge de ce lien, c'est à dire de dévier du trafic de ce lien.**
  - **Incorpore les situations de partage de charge.**

Génération du voisinage

Génération du voisin w^i

- **Calcul de la variation minimale de métrique Δ_i du lien i**
- **Notons F_i l'ensemble des flots passant par le lien i. Si $F_i = \emptyset$, $\Delta_i = \infty$ (pas de déviation possible). Sinon :**

(a) On supprime le lien i : $w' = (w_1, \ldots, w_{i-1}, \infty, w_{i+1}, \ldots, w_M)$.

(b) Mise à jour des tables de distances $D_u^v(w')$ par un algorithme de PCC.

(c) On calcule :

$$d_{min} = \min_{f \in F} \left[D_{s(f)}^{t(f)}(w') - D_{s(f)}^{t(f)}(w) \right]$$

, qui correspond à la variation minimale de distance entre la source $s(f)$ et la destination $t(f)$ de chaque flot f passant par le lien i.

(d) On définit Δ_i de la façon suivante :

$$\Delta_i = \begin{cases} 1 & \text{si } d_{min} = 0 \\ d_{min} & \text{si } 0 < d_{min} < \infty \\ \infty & \text{si } d_{min} = \infty \end{cases}$$

Génération du voisinage

Génération du voisin w^i

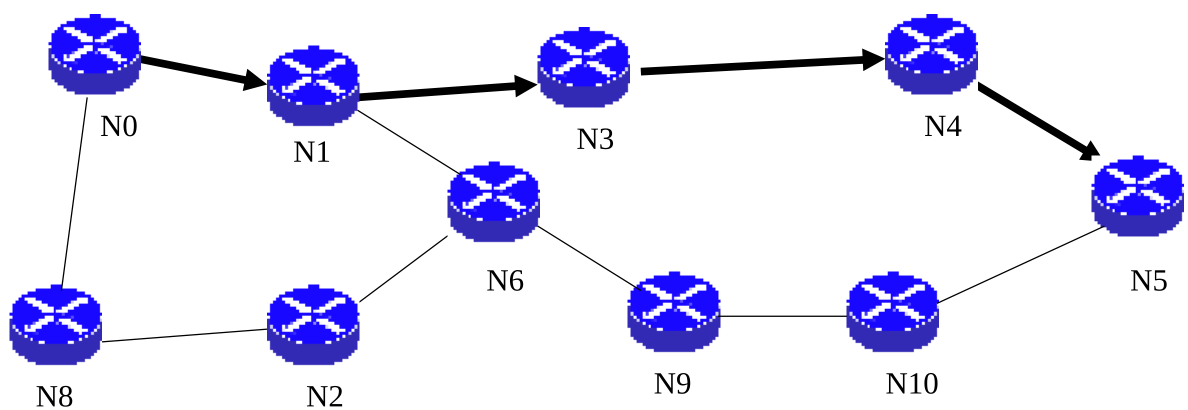
- **L'ensemble de flots F_i est obtenu lors de la propagation.**
- **Les distances $D_u^v(w')$ sont obtenues par un algorithme de PCC dynamique. On recopie les distances $D_u^v(w)$ pour pouvoir revenir rapidement à la solution courante avant de générer w^{i+1}.**
- **Dans la solution $w^i = (w_1, w_2, \ldots, w_i + \Delta_i, \ldots, w_M)$, il existera au moins un flot de F_i qui sera dévié tout ou partie du lien i :**
  - Si $d_{min} = 0$, cela signifie qu'il n'y avait pas unicité du plus court chemin pour au moins un des flots de F_i (partage de charge). En posant $\Delta_i = 1$, ce flot sera intégralement dévié du lien i,
  - Si $0 < d_{min} < \infty$, la solution w^i introduit du partage de charge pour au moins un des flots de F_i,
  - Si $d_{min} = \infty$, il n'existe pas d'autre chemin ne passant pas par le lien i pour les flots de F_i. On supprime donc ce lien de ceux dont la métrique peut être modifiée.

Génération du voisinage

Exemple

- **Toutes les métriques sont à 1. Ce réseau doit écouler deux demandes : l'une de N_1 vers N_5 et l'autre de N_0 vers N_5. Les PCC sont marquées en gras.**

$$D_{N_1}^{N_5}(w) = 3 \quad \text{et} \quad D_{N_0}^{N_5}(w) = 4$$



- **On veut déterminer $\Delta_{N3 \rightarrow N4}$.**

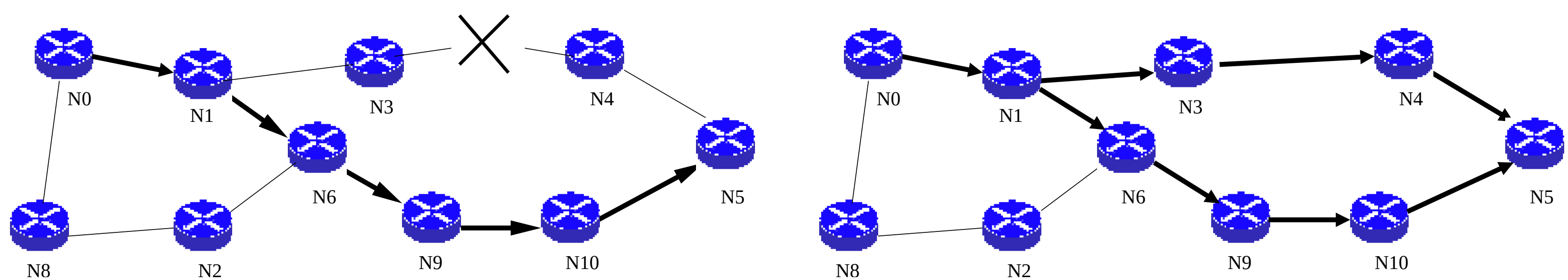
Génération du voisinage

Exemple

- **On supprime le lien N3-N4 ($w'_{N3 \rightarrow N4} = \infty$) :**

$$D_{N_1}^{N_5}(w') = 4 \quad \text{et} \quad D_{N_0}^{N_5}(w') = 5$$

$$\Delta_{N3 \rightarrow N4} = d_{min} = \min\{\, 4 - 3 \,,\, 5 - 4 \,\} = 1$$



Algorithme d'optimisation des métriques

Initialisation

Lecture de la solution initiale $w = (w_1, \ldots, w_M)$

Calcul des plus courts chemins : $D_u^v(w),\ \delta_{u,v}^x(w),\ n_u^v(w)$

Propagation des flots sur les plus courts chemins : $\gamma_u^v(w),\ Y_{u,v}(w)$

Calcul du coût $\Phi(w)$ de la solution initiale.

$w^* = w$ {Init. solution de coût minimum}

Algorithme d'optimisation des métriques

Boucle Principale

while Convergence() = false **do**
  $\Phi_{min} = \infty$
  for $i = 1 \dots M$ **do**
    Calcul de Δ_i : $w^i = (w_1, w_2, \dots, w_i + \Delta_i, \dots, w_M)$
    Calcul des plus courts chemins : $D_u^v(w^i),\ \delta_{u,v}^x(w^i),\ n_u^v(w^i)$
    Propagation des flots : $\gamma_u^v(w^i),\ Y_{u,v}(w^i)$
    Calcul du coût $\Phi(w^i)$
    if $\Phi(w^i) \leq \Phi_{min}$ **then**
      $w_{next} = w^i$ et $\Phi_{min} = \Phi(w^i)$
    end if
  end for
  $w = w_{next}$
  if $\Phi(w) < \Phi(w^*)$ **then**
    $w^* = w$ {Copie de la solution de coût minimum}
  end if
end while

Algorithme d'optimisation des métriques

- **Les étapes clés (pour chaque lien i)**
  - Calcul de la variation de métrique Δ_i : calcul de PCC
  - Calcul des PCC associés à la solution w^i
  - Calcul du coût : propagation des trafics
- **Il faut donc optimiser**
  - Le calcul des PCC suite à une augmentation de métrique,
  - La propagation des trafics suite à une augmentation de métrique,

Algorithme de PCC dynamique

- ## Algorithme de Ramalingam & Reps
  - La modification de la métrique d'un seul lien n'impacte souvent qu'un petit nombre de PCC.
  - Des algorithmes, dits de plus courts chemins dynamiques, permettent de traiter ce type de problèmes plus efficacement ques les algorithmes de Dijkstra ou de Bellman-Ford.
  - On présente une amélioration de l'algorithme de Ramalingam et Reps pour le cas d'une augmentation de métrique.

Algorithme de Ramalingam & Reps

Contexte

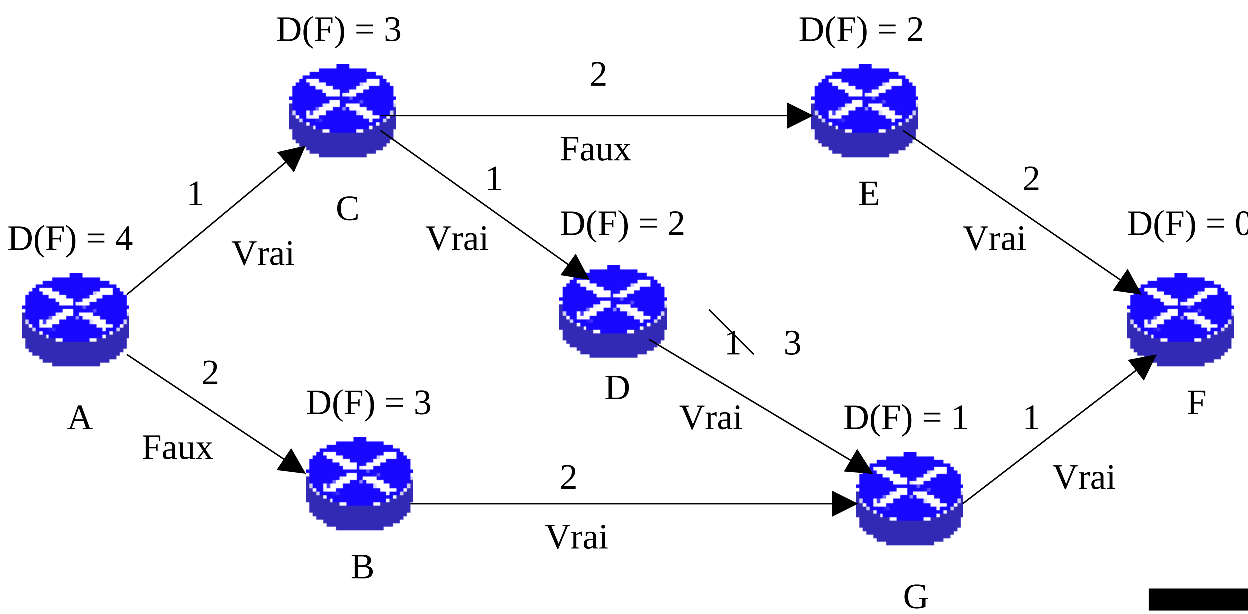
- On suppose que le calcul complet de toutes les tables de distance a été fait.
- On se place ici dans le cas où la métrique d'un lien (s, t) a augmentée d'une valeur Δ.
- On note w le vecteur de métriques initial et w' celui obtenu apres modification. Au début de l'algorithme, on a pour chaque noeud x et chaque lien (u,v):

$$D_u^v(w') = D_u^v(w),\ \delta_{u,v}^x(w') = \delta_{u,v}^x(w),\ n_u^v(w') = n_u^v(w)$$

Algorithme de Ramalingam & Reps

Exemple

- On suppose que le lien modifié est $(s,t) = (D,G)$ et que sa métrique passe de 1 à 3 ($\Delta = 2$).
- La destination pour laquelle on veut re-calculer les PCC est le noeud F. Sur la figure, on indique :
  - Pour chaque lien (u,v), sa métrique w_{uv} et son utilisation vers F, $\delta^F_{u,v}(w)$,
  - Pour chaque noeud u, sa distance à F : $D^F_u(w)$.



Algorithme de Ramalingam & Reps

1ère étape : propagation amont du changement.

- Elle permet de remonter le long des chemins utilisés pour joindre F et d'identifier les noeuds et les liens impactés par la modification $w_{s,t} \to w_{s,t} + \Delta$.

- A1. On vérifie que ce changement fera évoluer des distances vers F. Pour cela, il faut que : $\delta^F_{s,t}(w) = 1$ et $n^F_s(w) = 1$. Sinon, on fait $\delta^F_{s,t}(w') = 0$ et l'algorithme est terminé.

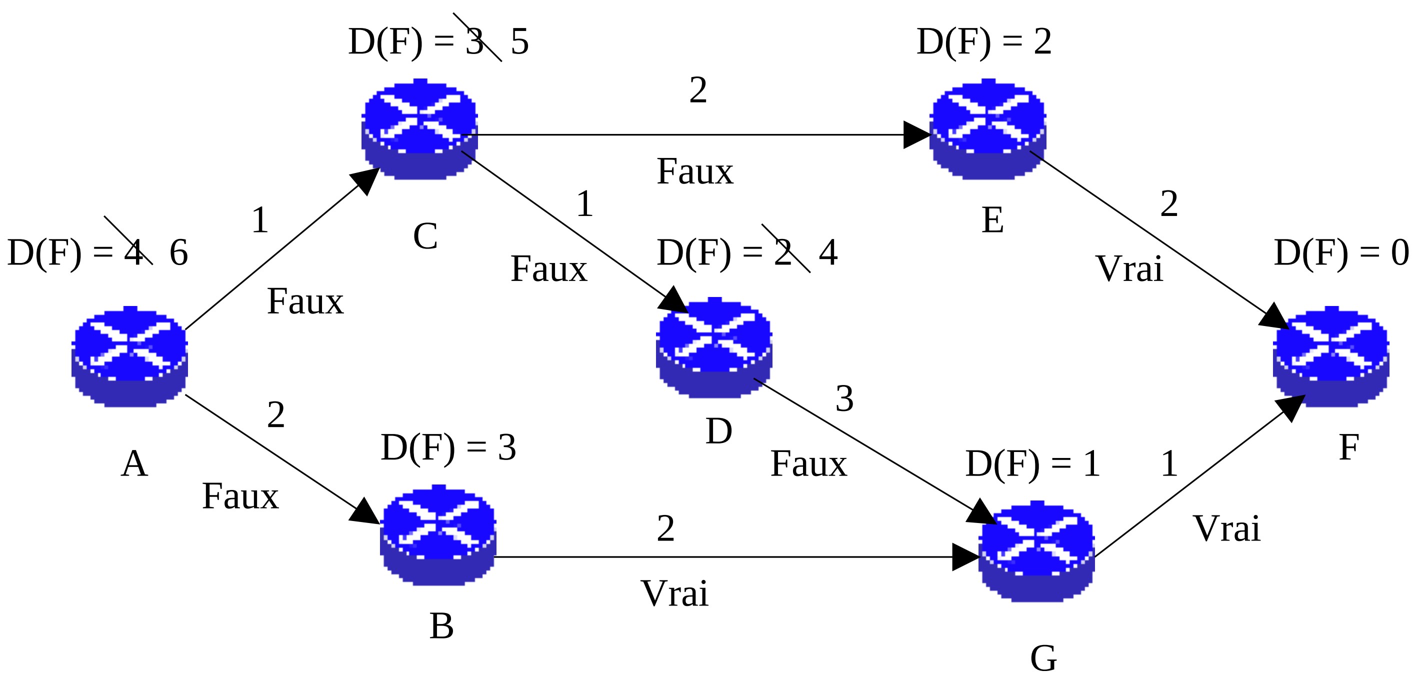
- A2. On initialise une liste Q avec le nœud source de l'interface modifiée : $Q = \{s\}$.

- A3. Pour chaque nœud $v \in Q$ et pour chaque lien (u, v) utilisé, i.e. $\delta^F_{u,v}(w) = 1$, on fait $D^F_u(w') = D^F_u(w) + \Delta$. Si $n^F_u(w) = 1$ alors, u est ajouté dans Q et on fixe $\delta^F_{u,v}(w') = 0$ (lien non utilisé).

Algorithme de Ramalingam & Reps

Exemple : illustration de l'étape 1

- **Evolution des distances vers F car $\delta^F_{D,G}(w) = 1$ et $n^F_D(w) = 1$.**
- **Au départ, $Q = \{D\}$. En remontant les PCC utilisés : $Q = \{D, C, A\}$**



Algorithme de Ramalingam & Reps

2nd étape : mise à jour des distances

- **Si $\Delta = 1$, vu que $n_s^F(w) = 1$ (cf A1), on a $\delta_{s,t}^F(w') = 1$ même après changement et les distances $D_u^F(w')$, $u \in Q$, calculées à l'étape 1 sont exactes.**
- **Sinon, il est possible que le lien (s,t) ne soit plus utilisé, auquel cas ces distances peuvent avoir été sur-évaluées. Si $\Delta > 1$, il faut les ajuster.**

- B1. On traite le routeur source s. On va regarder s'il existe un chemin au départ de s tel que sa distance soit plus petite que $D_s^F(w) + \Delta$. On calcule :

$$d = \min_{(s,v)} \left[{w'}_{s,v} \; + \; D_v^F(w) \right]$$

Si $d < D_s^F(w')$, on pose $\Delta_1 = d - D_s^F(w')$. Δ_1 représente l'erreur sur l'augmentation de distance qui a été propagée à tous les noeuds de Q. On met à jour la distance de s à F : $D_s^F(w') = d$.

Algorithme de Ramalingam & Reps

2nd étape : mise à jour des distances ($\Delta > 1$)

- **B2.** **Pour tout noeud $u \in Q$, on corrige les distances en faisant :**
  $D_u^F(w') = D_u^F(w') - \Delta_1$**. On calcule alors :**

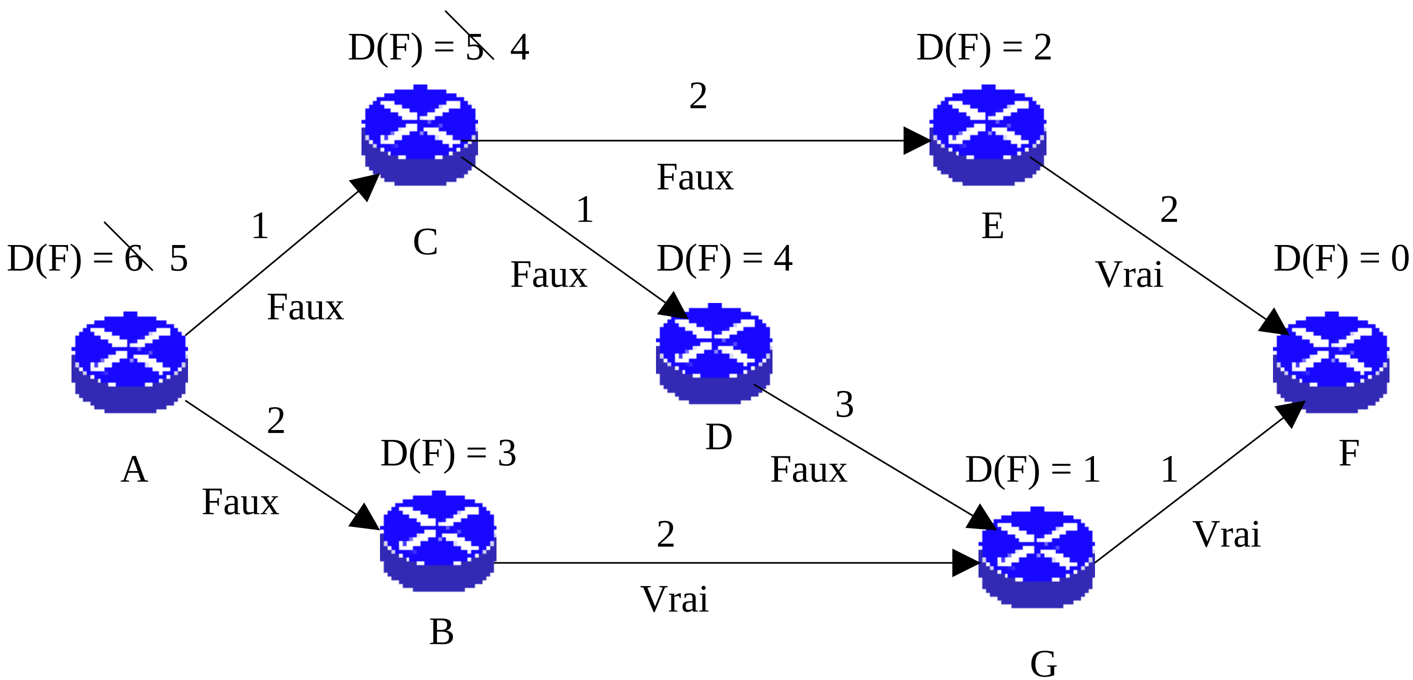
$$d = \min_{(u,v)} \left[{w'}_{u,v} \ + \ D_v^F(w) \right]$$

Si $d < D_u^F(w')$, on corrige la distance de u à F en faisant $D_u^F(w') = d$, puis on insère u dans une map H à l'associant à la distance d.

- **B3.** **Cette dernière étape permet de réajuster les distances faussées du fait de l'ordre dans lequel les routeurs ont été traités à l'étape B2. Si le traitement précédent etait fait dans l'ordre des distances à la destination, cette sous-partie serait inutile.**
  Tant que $H \neq \emptyset$, on retire u de H où $u = argmin_{u \in H}(D_u^F(w'))$.
  Pour toutes les interfaces (v, u) entrantes dans u, on compare : $d_1 = D_v^F(w')$ à $d_2 = D_u^F(w') + {w'}_{v,u}$. Si v vérifie $d_1 > d_2$ alors, on fixe $D_v^F(w') = d_2$ et on ajoute le couple $(v, D_v^F(w'))$ dans H.

Algorithme de Ramalingam & Reps

- **Exemple : illustration de l'étape 2**
  - **On obtient $\Delta_1 = 0$ à l'étape B1.**
  - **On insère $(C, 4)$ puis $(A, 5)$ à l'étape B2.**



Algorithme de Ramalingam & Reps

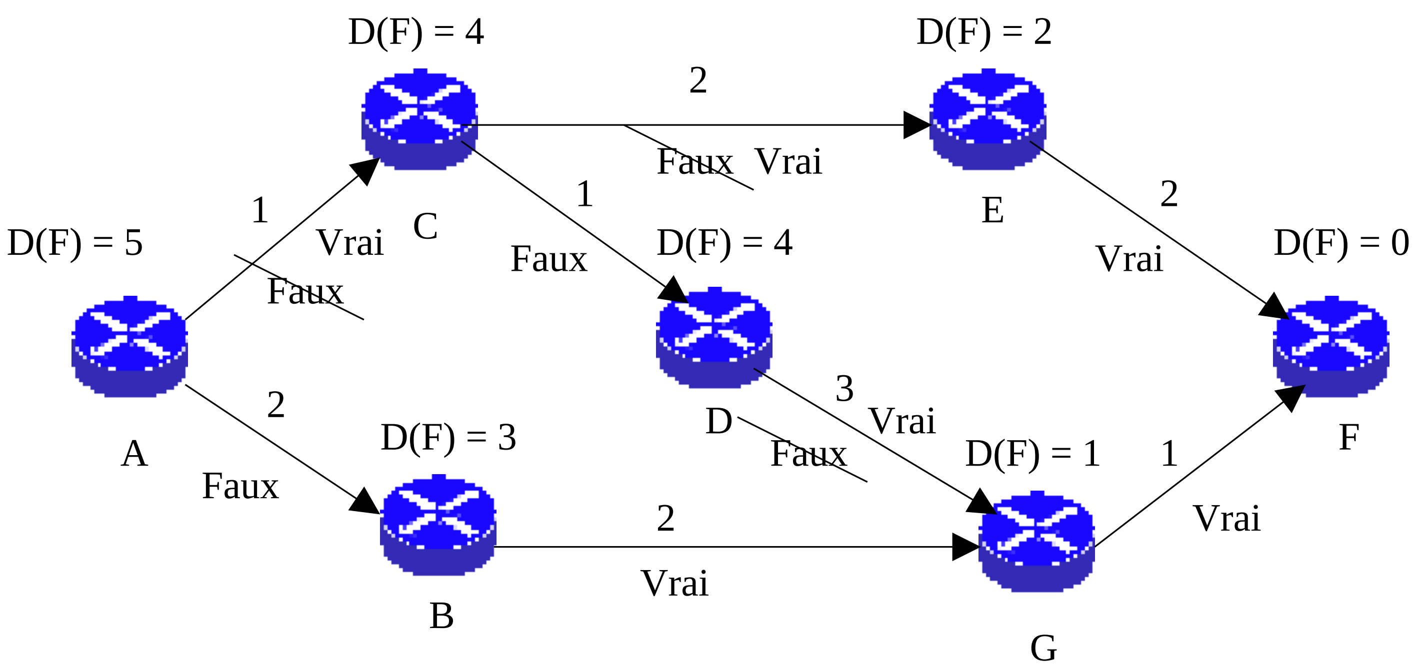
- **3ème étape : calcul des PCC à l'aide des nouvelles distances.**
  - Elle permet de corriger les flags d'utilisation des liens $\delta^F_{u,v}(w')$, ainsi que le nombre de PCC vers F pour chaque noeud u, $n^F_u(w')$.
  - Pour chaque noeud $u \in Q$, on va tester les distances sur les differents liens sortants. Pour chaque lien (u, v) on compare : $D^F_u(w')$ à $D^F_v(w') + w'_{u,v}$. Si l'egalité est verifiée, on fait:

$$\delta^F_{u,v}(w') = 1 \quad \text{et} \quad n^F_u(w') = n^F_u(w') + 1$$

Algorithme de Ramalingam & Reps

Exemple : illustration de l'étape 3

- **On peut remarquer que nous n'avons modifié que quelques noeuds : D, C et A.**
- **L'algorithme de Dijkstra demanderait un coût calculatoire bien plus élevé pour obtenir le même résultat.**



La propagation des trafics

Introduction

- La propagation des trafics permet de calculer la charge des liens, qui est necessaire pour pouvoir calculer le coût.
- On considère ici la propagation vers un noeud x. Il faudra itérer sur toutes les destinations x pour calculer la charge $Y_{i,j}(w)$ des liens.
- La relation de base (conservation des flots) est :

$$Y_{i,j}^{x}(w) = \frac{\lambda_i^x + \sum\limits_{k \neq x} Y_{k,i}^{x}(w)}{n_i^x}$$

- Propagation statique et propagation dynamique :
  - Le cas statique propage tous les trafics.
  - Le cas dynamique est utilisé suite à un changement de métrique. Il ne propage que des variations de trafic pour les noeuds impactés par la modification de métrique.

La propagation statique

Algorithme

- Parcours des noeuds des plus éloignés aux plus proches de x
- Pour chaque noeud i, on calcule le trafic $\gamma_i^x(w)$ reçu pour la destination x : somme du trafic direct et des trafics $Y_{k,i}^x(w)$ reçu sur chaque lien entrant (k, i).
- Ce trafic est réparti sur les liens sortants (partage de charge).

$Q = \{1, \ldots, N\} - \{x\}$
while $Q \neq \emptyset$ **do**
  $i = argmax_{u \in Q} D_u^x(w)$
  $\gamma_i^x = \lambda_i^x$
  $\forall k$ tel que $\delta_{k,i}^x(w) = 1 : \gamma_i^x(w) = \gamma_i^x(w) + Y_{k,i}^x(w)$
  $\forall j$ tel que $\delta_{i,j}^x(w) = 1 : Y_{i,j}^x(w) = \gamma_i^x(w) / n_i^x$
  $Q = Q - \{i\}$
end while

La propagation dynamique

Algorithme

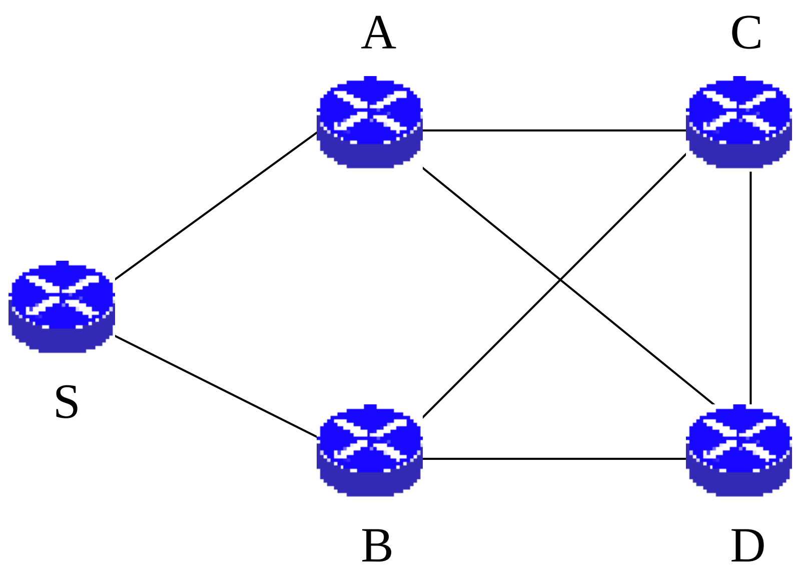
- **Soit M la liste des noeuds affectés par le changement de métrique.**
  - **Un noeud est affecté s'il est source ou destination d'un lien dont l'utilisation a été modifiée suite au changement de métrique.**

while $M \neq \emptyset$ **do**
  $i = argmax_{u \in M} D_u^x(w)$ {i est le noeud le plus éloigné de x}
  $M = M - \{i\}$
  $Y = \lambda_i^x(w)$ {Ajout du trafic direct en i vers x}
  $\forall k$ tel que $\delta_{k,i}^x(w) = 1 : Y = Y + Y_{k,i}^x(w)$ {Trafic des liens entrants}
  $Y = Y/n_i^x$ {Partage de charge}
  for j tel que $\delta_{i,j}^x = 1$ **do**
    if $Y_{i,j}^x \neq Y$ **then**
      $Y_{i,j}^x = Y$ {Propagation sur les liens sortants}
      $M = M \cup \{j\}$ {Le trafic vers j a changé}
    end if
  end for
end while

Résultats

Topologie à 5 nœuds et avec une demande

- **Données :**



16 Kbps : A/D, D/A, B/D,et D/B.

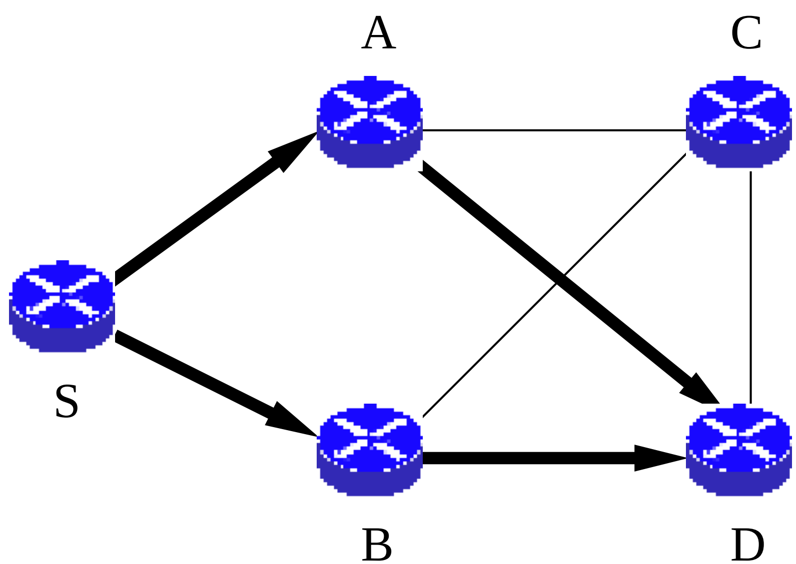
64 Kbps : C/D et D/C.

1920 Kbps : les autres.

- **Une demande : 50 Kbps entre S et D.**
- **Métriques à 1.**

Topologie à 5 nœuds et avec une demande

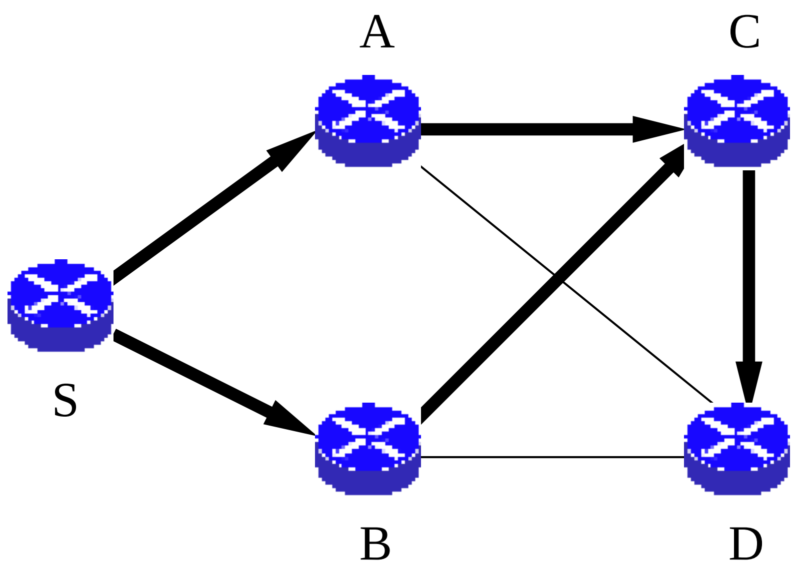
- **Routage initial**



- **25 kbps sur les interfaces A/D et B/D de capacité 16 kbps.**

Topologie à 5 nœuds et avec une demande

Après optimisation

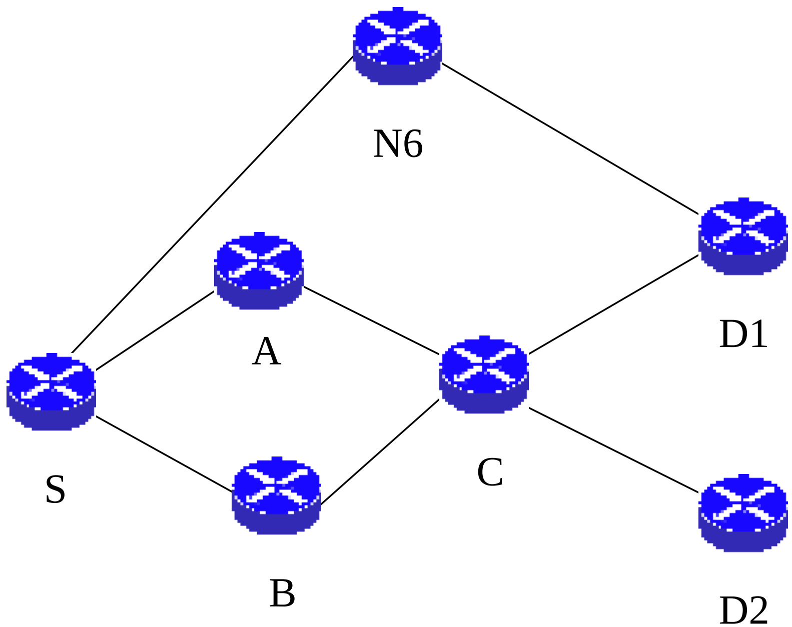


+ 1 sur l'interface A/D
+ 1 sur l'interface B/D.
+ 1 sur l'interface A/D.
+ 1 sur l'interface B/D.

- **La solution initiale est un minimum local.**

Topologie à 7 nœuds et 2 demandes

- **Données :**



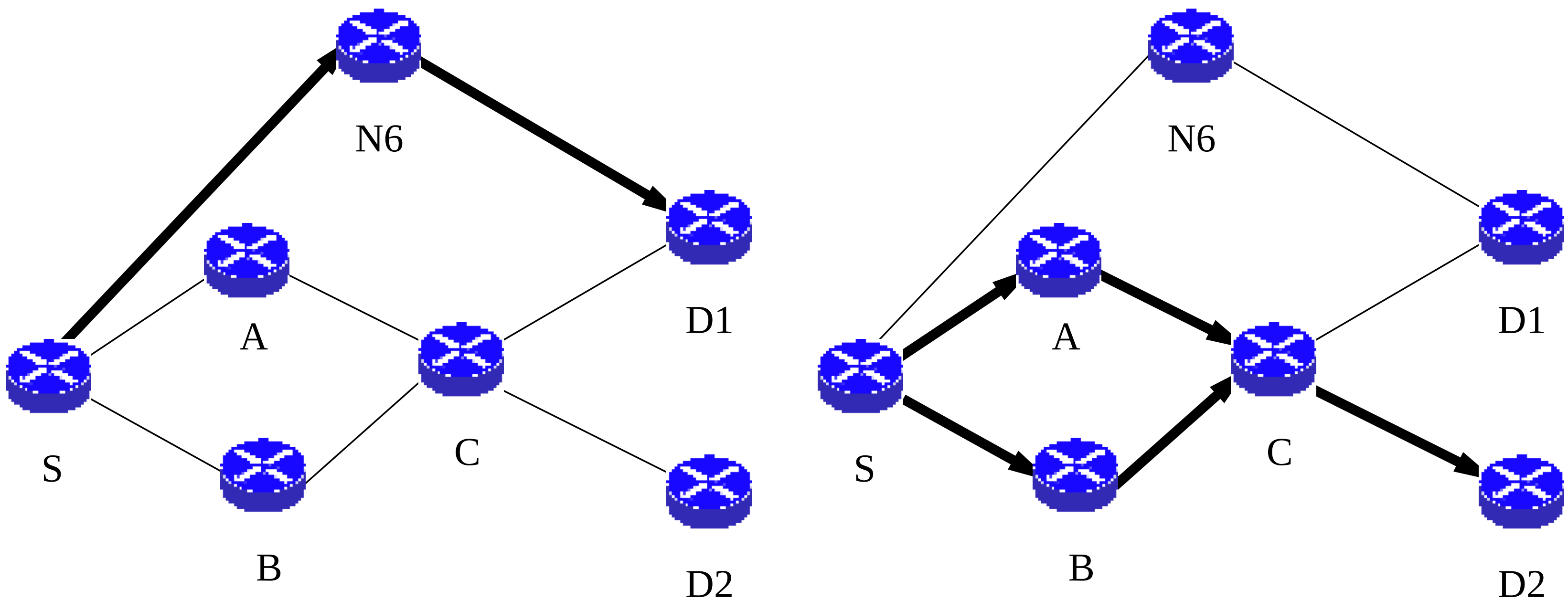
16 Kbps : S/N6.

64 Kbps : les autres.

- **2 demandes : S vers D1 de 20 Kbps et S vers D2 de 10 Kbps.**
- **Métriques à 1.**

Topologie à 7 nœuds et 2 demandes

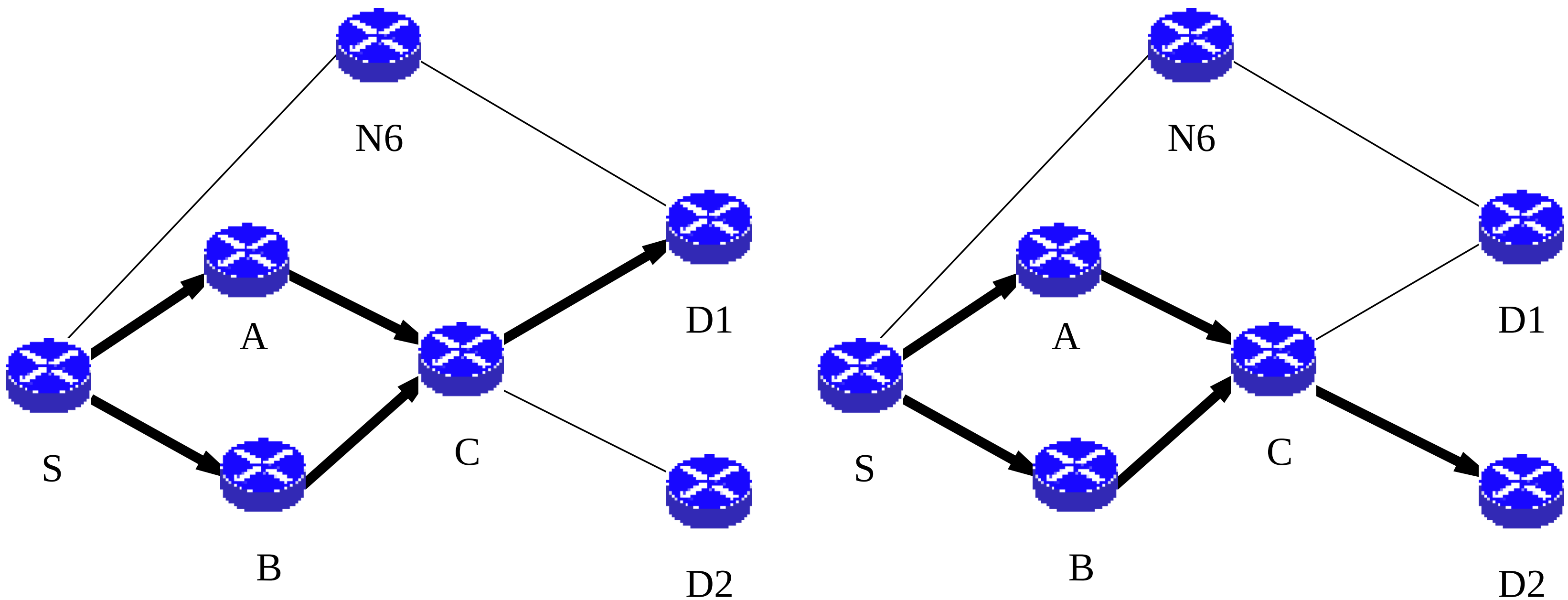
- **Routage initial**



  - **Saturation de l'interface S/N6.**

Topologie à 7 nœuds et 2 demandes

Après optimisation



- **Modifications :**
  - Changement 1 :
    + 1 sur l'interface S/N6
  - Changement 2 :
    + 1 sur l'interface S/N6

Conclusion

Conclusion

Optimisation des métriques de routage IP

- **Problème :**
  - Augmentation du trafic et besoins de QoS
  - Contexte concurrentiel interdisant un sur-dimensionnement excessif
  - Jouer sur les métriques IP pour adapter les routes aux trafics transportés par le réseau.
- **Heuristique présentée :**
  - Technique de recherche locale
  - Originalité : structure de voisinage.
  - Utilisation d'un algo de PCC dynamique et propagation dynamique des trafics.