Router et acheminer des informations dans un réseau

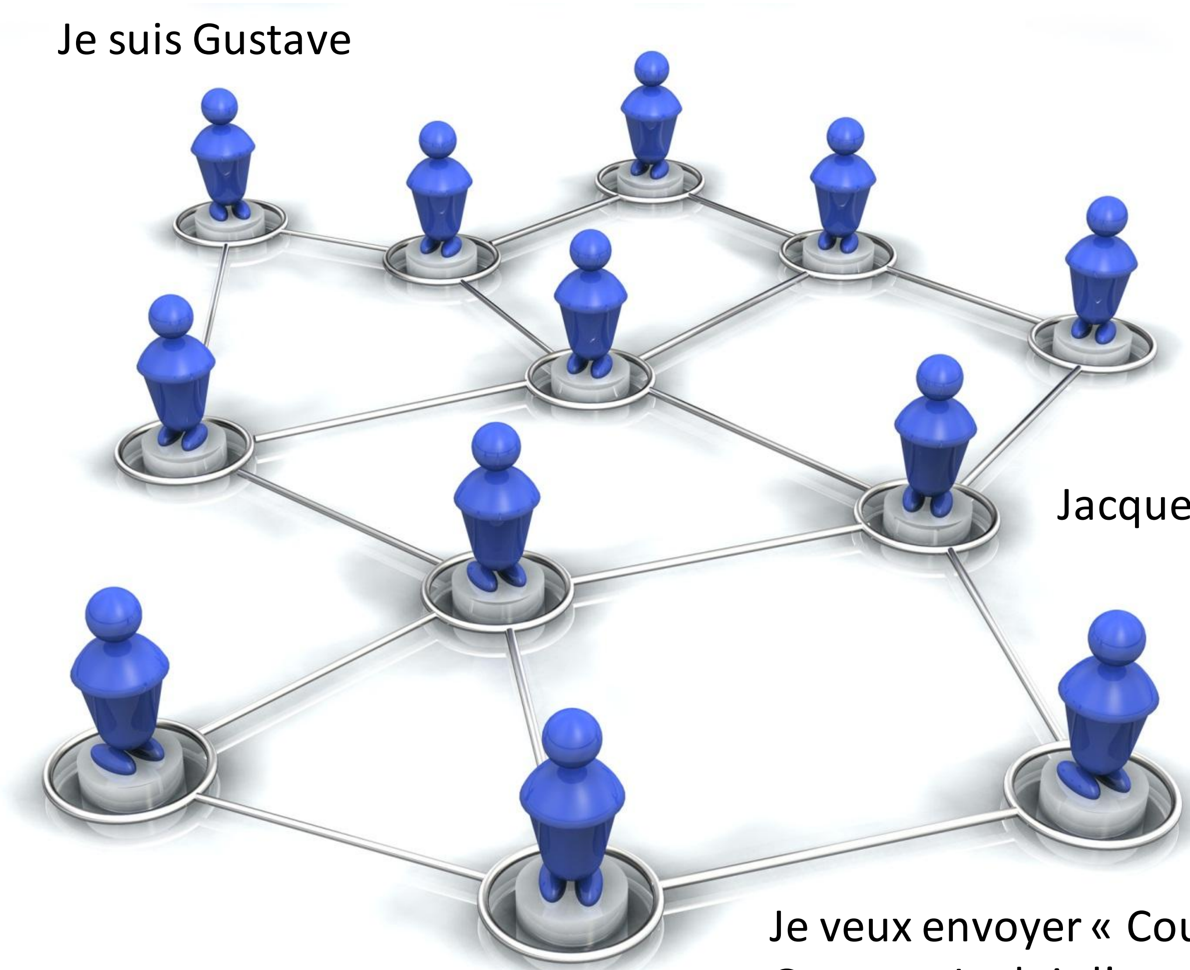
Alain Cournier

INTRODUCTION (AÏE)

Le problème

- Communiquer est un des fondements des systèmes distribués
- On souhaite acheminer une information i vers une destination d.
- Or un nœud ne connait que des informations sur son voisinage. Donc si l'information est à destination du nœud ou de l'un de ses voisins dans le réseau tout va bien. Que faire dans le cas contraire ?

Je suis Gustave



Jacques

Je veux envoyer « Coucou » à Gustave. Je dois l'envoyer à Jean ou à Jacques ?

Jean

Première Solution

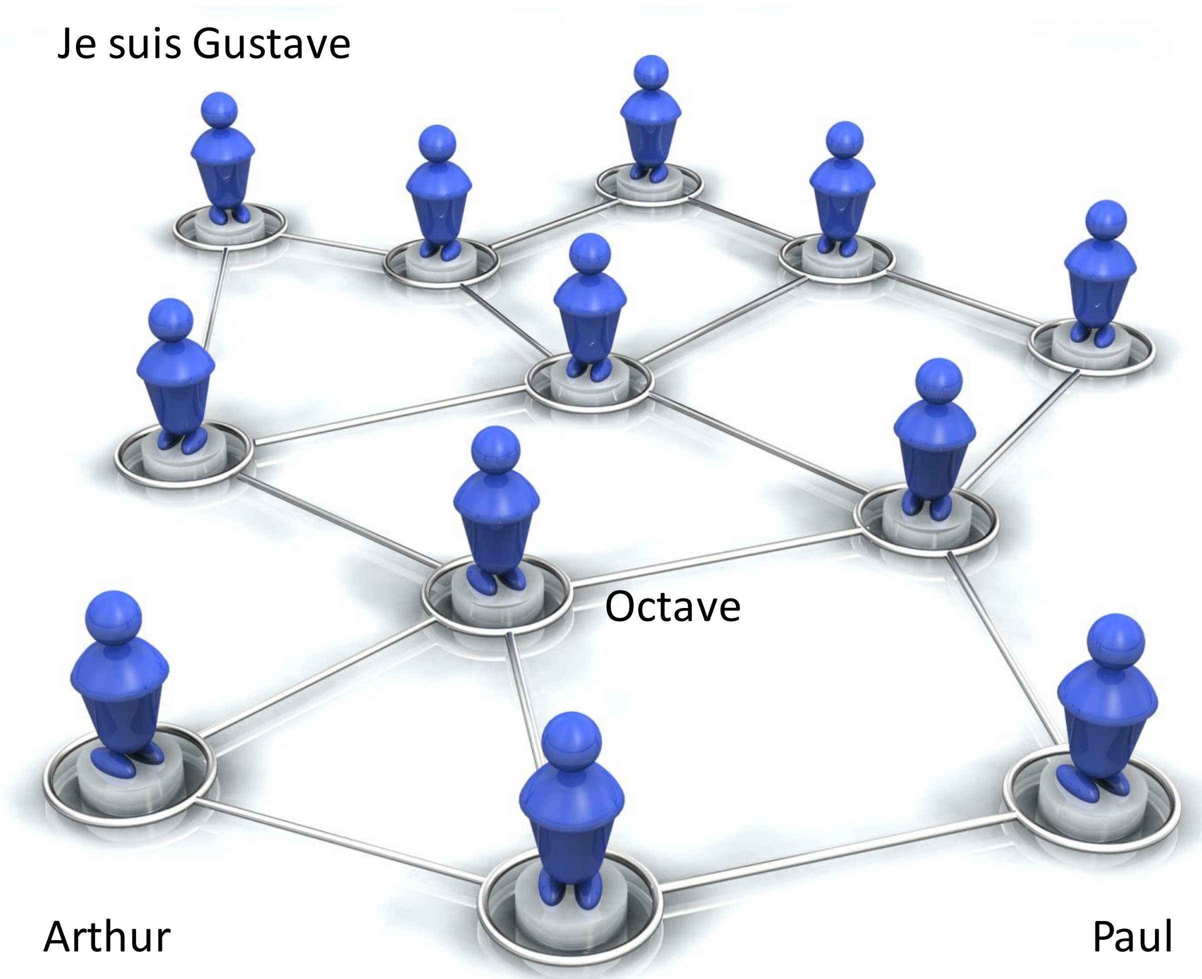
- On l'envoie à tout le monde.

- On réalise une diffusion vers tous les nœuds du réseau :
  - Avantage : si le destinataire est dans le réseau il reçoit l'information
  - Inconvénient : Problèmes de confidentialité puisque tous les nœuds ont l'information.

Routage et plus courts chemins

- Le problème est complexe car si un nœud x veut envoyer une information à un nœud y, il ne sait pas obligatoirement où se trouve notre nœud y dans le réseau.

- Il doit donc calculer un chemin à partir du nœud x afin que l'information atteigne y



Je veux envoyer « Coucou » à Gustave. Je dois l'envoyer à Paul à Arthur ou à Octave?

Seconde solution

- Seconde solution : choisir un chemin de x vers y. L'information circulera seulement sur ce chemin ; C'est la notion de routage.

- Avantage : Gain de confidentialité.

Seconde solution

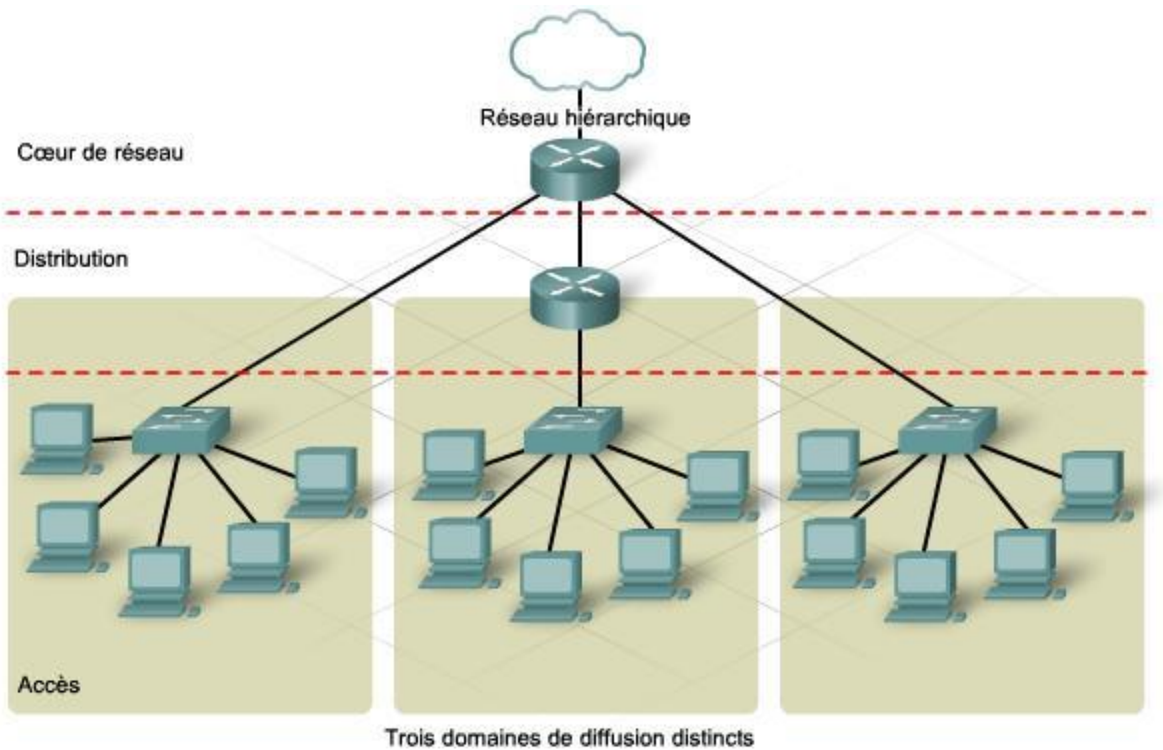
- Le routage est le moyen qui permet à un nœud de choisir un lien vers un de ses voisins pour envoyer une information vers une destination du réseau.
- Pour cela, on utilise souvent une table de routage.

Premiers problèmes liés au routage

- Comment choisir un chemin vers une destination ?
  - Le plus court ?
  - Le chemin de plus forte capacité ?
  - Le moins cher ?
  - Le plus sûr ?
- Comment construire les tables de routage ?
- Comment réduire la taille de ces tables ?

Routage

- Le routage IP est une réponse partielle à ces questions :
  - Réseau hiérarchique
  - Machines spécifiques : Routeur
  - Liens entre divers sous réseaux utilisant des passerelles gérées par ces routeurs.


Réseau hiérarchique
Cœur de réseau
Distribution
Accès
Trois domaines de diffusion distincts

Comment représenter ce chemin

- On peut calculer ce chemin
  - à la demande (lorsqu'on veut envoyer une information) : Routage réactif
  - Une fois pour toute : Routage Pro-actif.
- Dans le cas du routage pro-actif, une fois calculés ces chemins sont conservés sur les nœuds du réseau.
  - Sous forme de fonctions
  - Sous forme de tables

COMMENT REPRÉSENTER LES INFORMATIONS NÉCESSAIRES AU ROUTAGE ?

Le problème : Et pour Amiens ?

Le problème

- Un nœud lorsqu'il reçoit un message doit être capable de déterminer par quel lien le cheminement du message vers sa destination devra se poursuivre.
- Il existe au moins trois façon de résoudre ce problème

Solution 1 : Le message contient la route qu'il doit suivre

- C'est la solution du GPS qui calcule une fois la route entre le point de départ et le point d'arriver. Après quoi il sait quelle route emprunter à chaque carrefour.
- Inconvénients :
  – l'expéditeur connait tous les chemins vers tous les destinataires.
  – La longueur du message.

Solution 2 : Le nœud connait la direction à prendre pour chaque destination

- Idéal pour le message qui ne doit transporter que sa destination.
- Inconvénient : Chaque nœud connait toutes les destinations du réseau.

Discution

- La solution 1 est utilisé dans les réseaux fortement dynamiques

- La solution 2 est utilisé dans les réseaux fixes

- Il existe une solution mixte par exemple dans les réseaux de téléphonie cellulaire.

CODER UNE TABLE DE ROUTAGE

Codage d’une table

- Coder une table de routage, c’est établir sur chaque nœud du réseau une correspondance entre une identité (la destination) et un canal.

Exemple : poids des arêtes 1

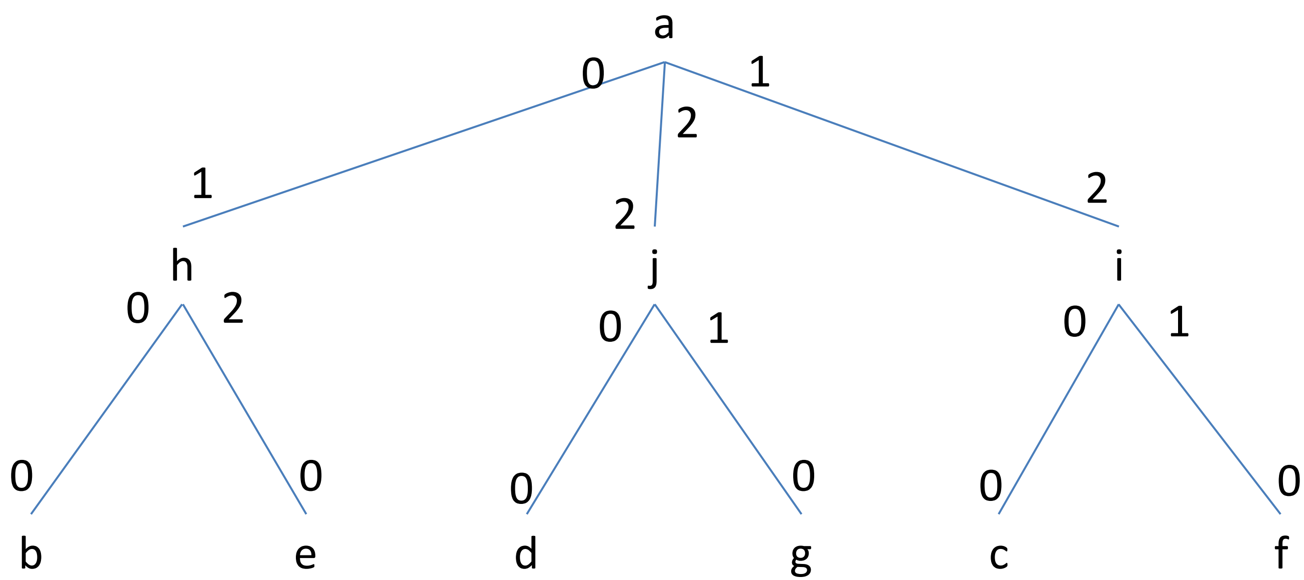


Table de routage de a

Dest	a	b	c	d	e	f	g	h	i	j
Canal	Loc.	0	1	2	0	1	2	1	2	3

Exemple : poids des arêtes 1

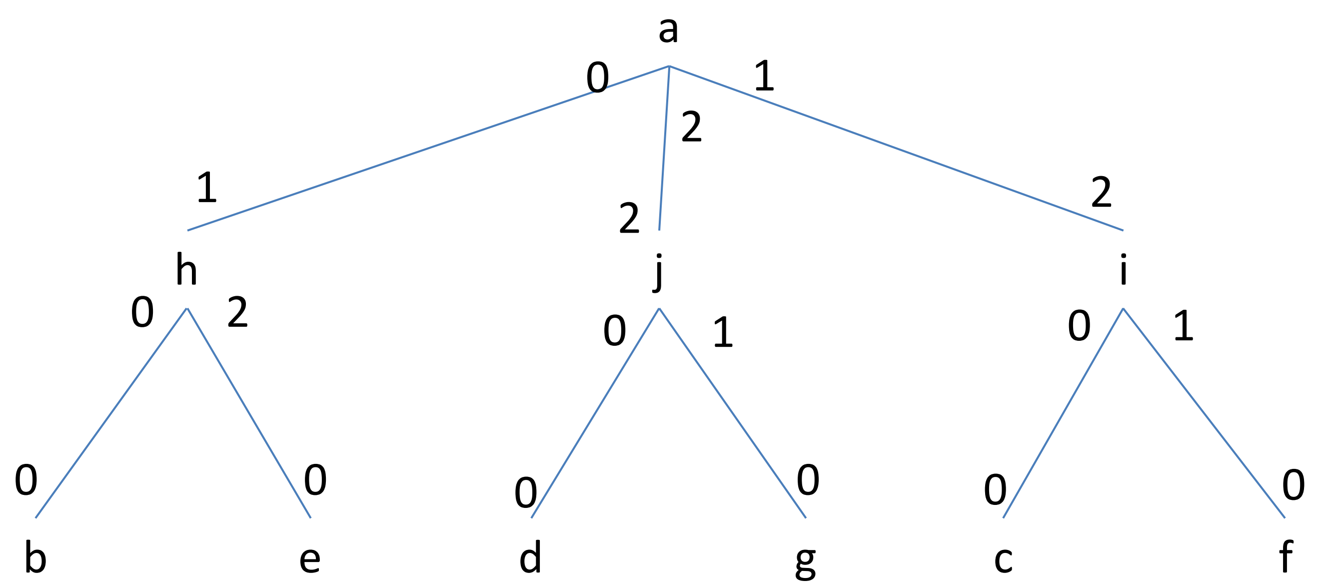


Table de routage de a

canal	**0**	**1**	**2**
Dest	<b, e, h>	<d, g, j>	<c, f, i>

LIMITER LA TAILLE DES TABLES DE ROUTAGE IMPLANTÉES SUR UN NŒUD

Taille de la table de routage

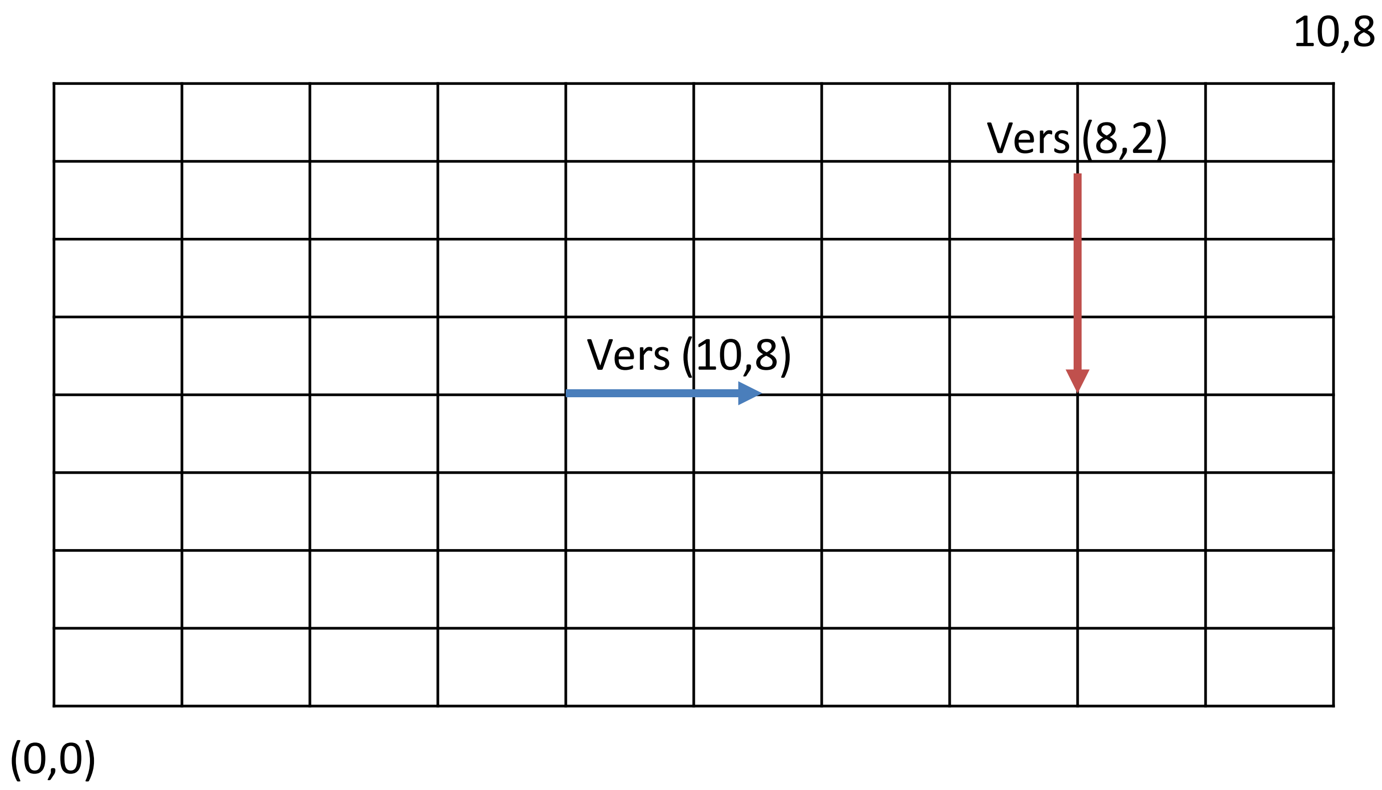
- Ici pour la taille d'une table de routage on retient l'espace utilisé pour représenter cette information.

Aide par la topologie

- Dans une grille (à 1, 2 ou 3 dimensions) les coordonné du nœud (courant) et de la destination suffissent pour déterminer quel voisin prolongera le chemin (si toutes les arêtes ont le même poids)

- Il faut que l'étiquetage des nœuds soit cohérent

Exemple sur une grille



Aide par la topologie

- Il en est de même pour des topologies telles que :
  - Les anneaux
  - Les tores
  - Les arbres
  - Les hypergraphes

Exemple : poids des arêtes 1

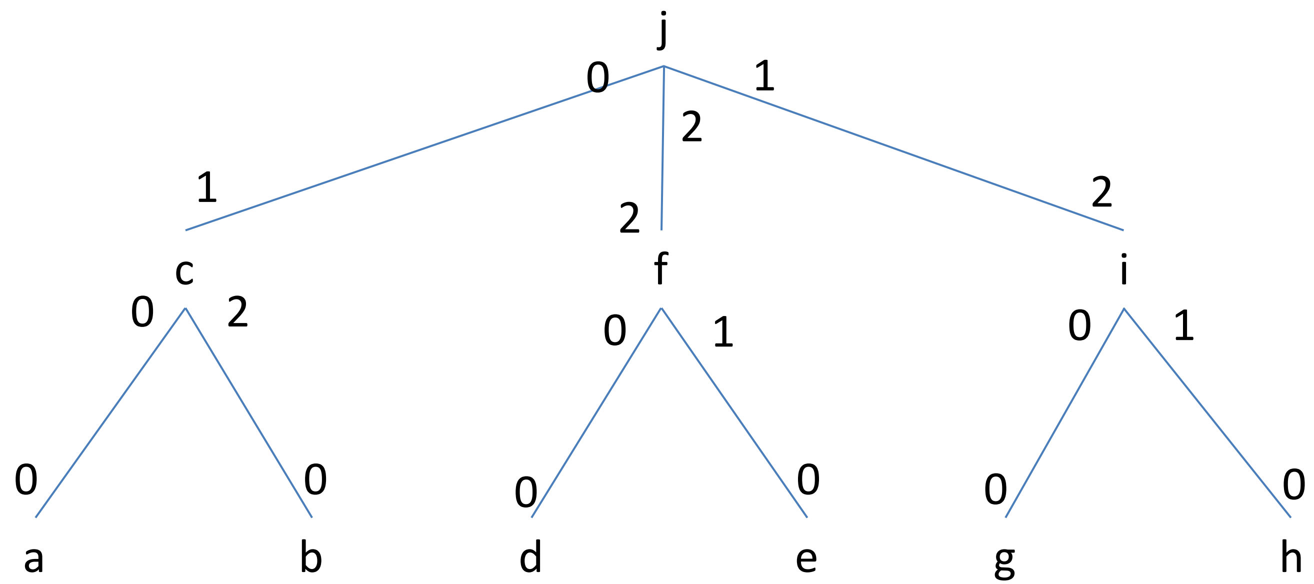


Table de routage de j

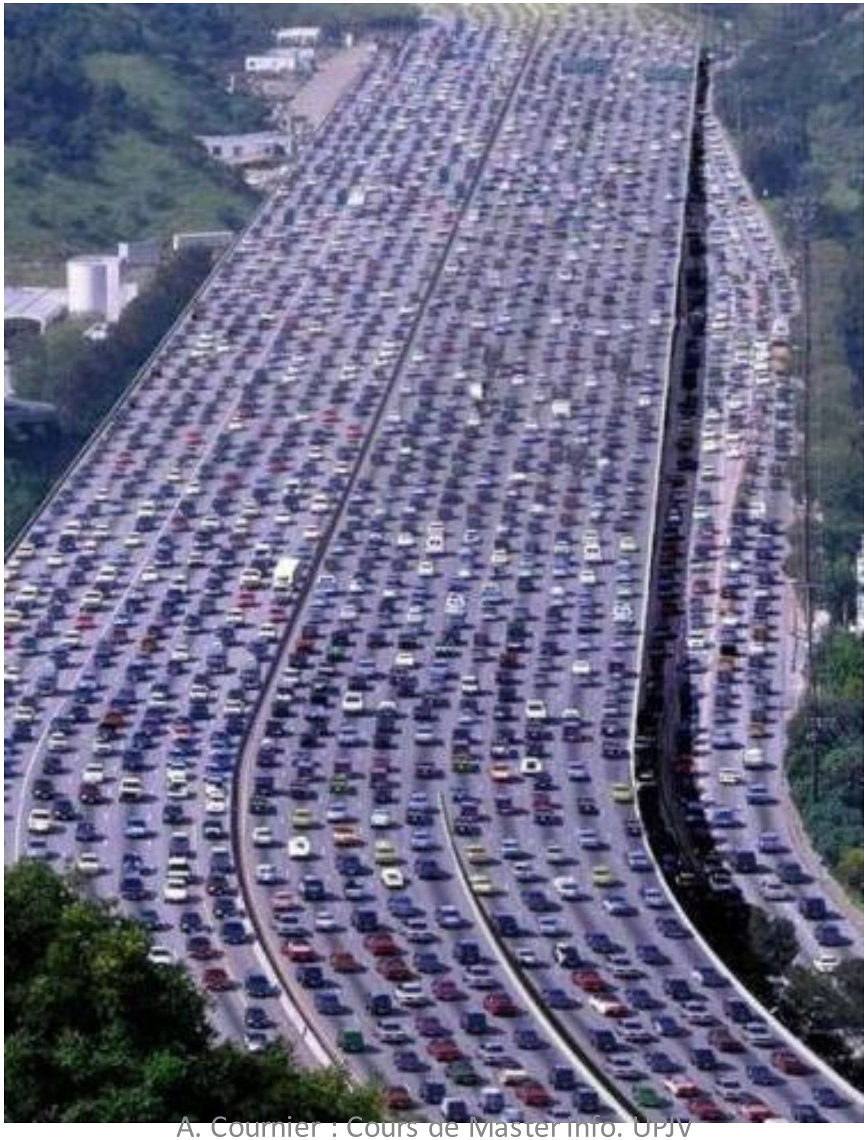
canal	**0**	**1**	**2**
Dest	[a..c]	[g..i]	[d..f]

Le nom du destinataire dépend-t-il de sa position dans le réseau ?

- Vrai avec l'adresse IP par exemple
- Faux avec votre téléphone portable

- Pourtant dans les deux cas il faut acheminer l'information ?

Et pour les embouteillages ?



Le routage alternatif

Les solutions radicales ?

Des solutions plus douce

- Peut-on garantir que cela n’arrivera pas ?

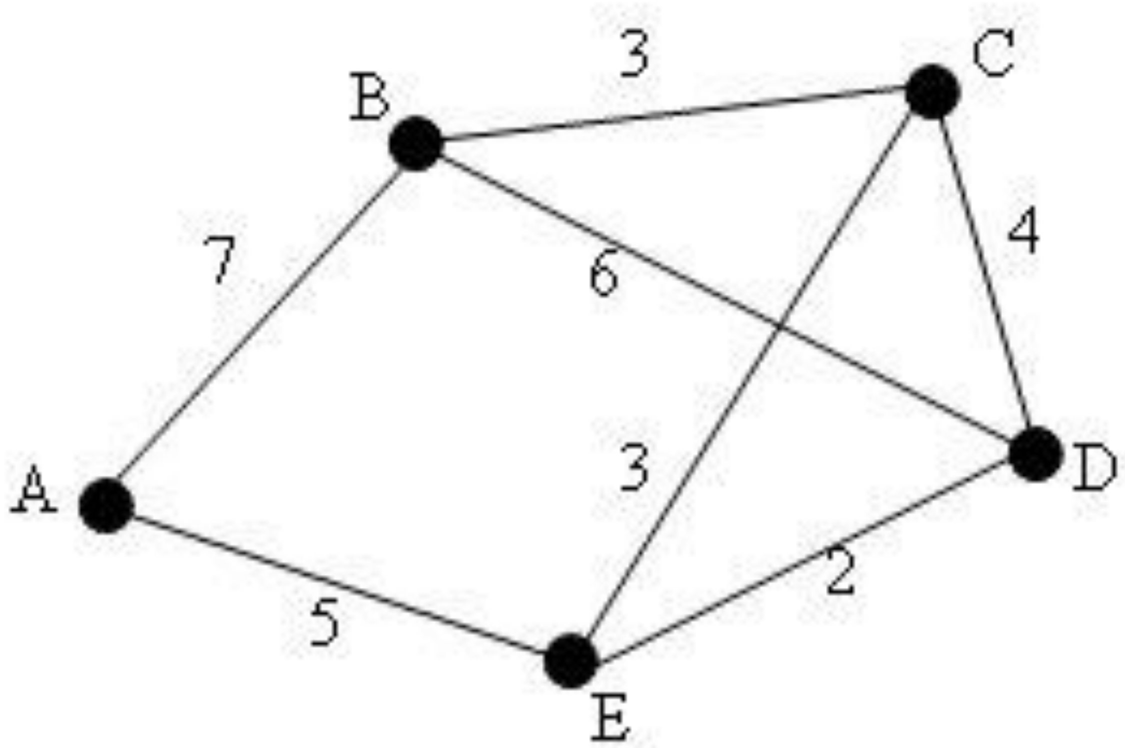
Que faire si le réseau est dynamique ?

Le problème

- Un réseau peut être représenté par un graphe étiqueté G=(X,U,V)
- X ensemble de nœuds du réseau
- U ensemble de liens du réseau
- V une application qui à chaque lien du réseau associe un poids.

ALGORITHME DE PLUS COURT CHEMIN ENTRE TOUT COUPLE DE SOMMETS

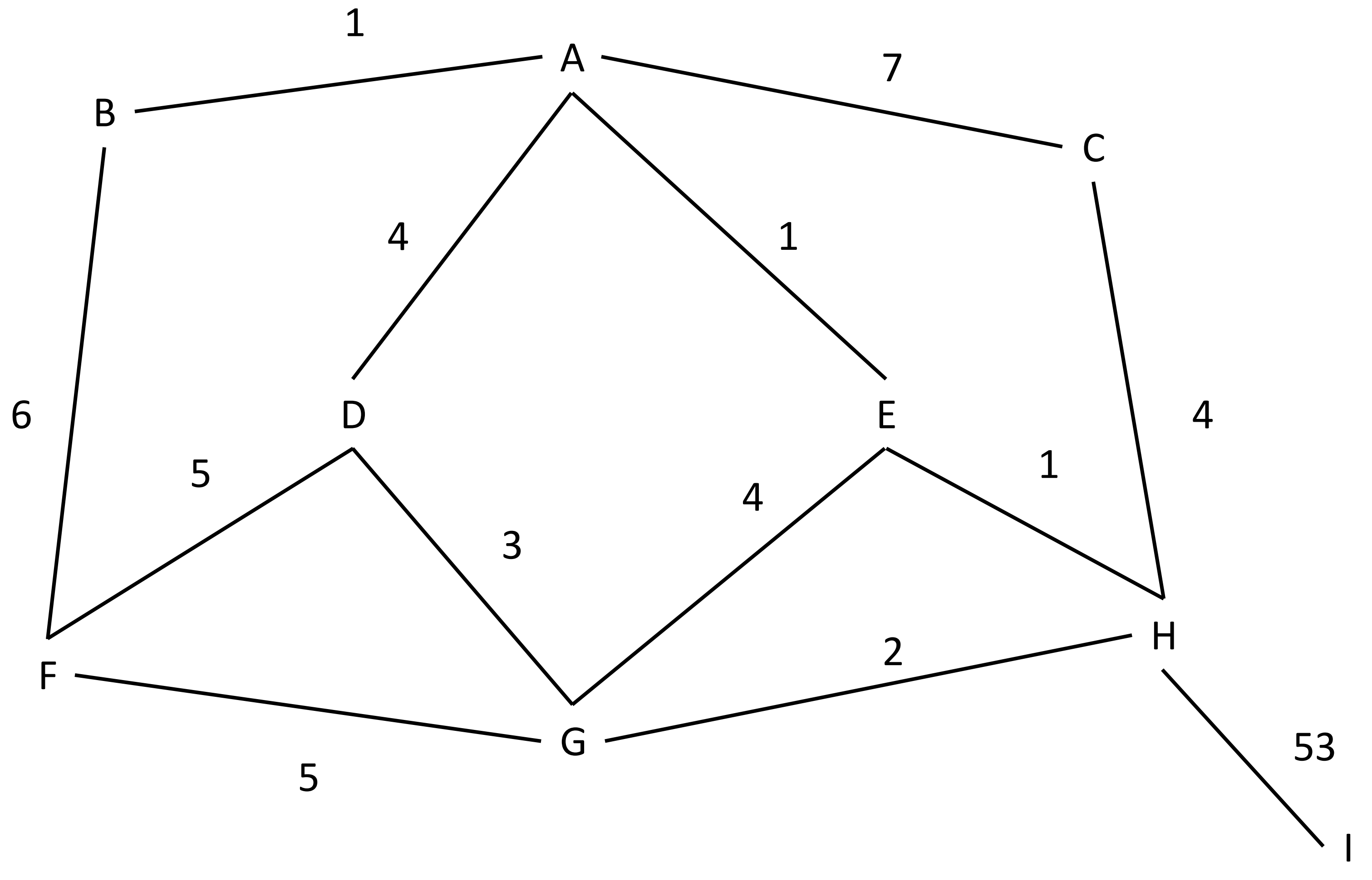
Exemple



Représentation

- La représentation en algorithmique centralisée se fait par une matrice M carrée indicée par les éléments de X
  - $M(x,x) = 0$
  - $M(x,y) = V(x,y)$ si $(x,y) \in U$
  - $M(x,y) = \infty$ si l'arête (x,y) n'est pas dans U

Exemple



Représentation par matrice

	A	B	C	D	E	F	G	H	I
A	0	1	7	4	1	∞	∞	∞	∞
B	1	0	∞	∞	∞	6	∞	∞	∞
C	7	∞	0	∞	∞	∞	∞	4	∞
D	4	∞	∞	0	∞	5	3	∞	∞
E	1	∞	∞	∞	0	∞	4	1	∞
F	∞	6	∞	5	∞	0	5	∞	∞
G	∞	∞	∞	3	4	5	0	2	∞
H	∞	∞	4	∞	1	∞	2	0	53
I	∞	∞	∞	∞	∞	∞	∞	53	0

But

- On souhaite connaitre le plus court chemin entre tous les couples de sommets du graphe.

- Utile pour l'acheminement d'information d'une source vers une destination dans un réseau

Représentation Table

	A	B	C	D	E	F	G	H	I
A	(0,_)	(1,B)	(7,C)	(4,D)	(1,E)	(∞,_)	(∞,_)	(∞,_)	(∞,_)
B	(1,A)	(0,_)	(∞,_)	(∞,_)	(∞,_)	(6,F)	(∞,_)	(∞,_)	(∞,_)
C	(7,A)	(∞,_)	(0,_)	(∞,_)	(∞,_)	(∞,_)	(∞,_)	(4,H)	(∞,_)
D	(4,A)	(∞,_)	(∞,_)	(0,_)	(∞,_)	(5,F)	(3,G)	(∞,_)	(∞,_)
E	(1,A)	(∞,_)	(∞,_)	(∞,_)	(0,_)	(∞,_)	(4,G)	(1,H)	(∞,_)
F	(∞,_)	(6,B)	(∞,_)	(5,D)	(∞,_)	(0,_)	(5,G)	(∞,_)	(∞,_)
G	(∞,_)	(∞,_)	(∞,_)	(3,D)	(4,E)	(5,F)	(0,_)	(2,H)	(∞,_)
H	(∞,_)	(∞,_)	(4,C)	(∞,_)	(1,E)	(∞,_)	(2,G)	(0,_)	(53,I)
I	(∞,_)	(∞,_)	(∞,_)	(∞,_)	(∞,_)	(∞,_)	(∞,_)	(53,H)	(0,_)

Par opérations sur les matrices

- Pour chaque couple de sommet (x,y) on calcule la valeur

$$v2(xy) = \mathrm{Min}_{z \in X}(v(xz)+v(zy))$$

- On obtient ainsi le chemin de poids minimal de x à y dont la longueur est au plus 2. Si on souhaite les chemins de longueur au plus 3 on calcule la valeur :

$$v3(xy) = \mathrm{Min}_{z \in X}(v2(xz)+v(zy))$$

Opération de Base (Entête)

- Algorithme OpMat
- Données :
  - M, N : deux matrices d'entiers indicées par les sommets
- Résultat :
  - Res : Une matrice d'entier indicées par les sommets

Opération de Base (code)

- DébutCode
  - Pour tout x ϵ X faire
    - Pour tout y ϵ X faire
      - Res[x,y] ← M[x,y] +N[y,y]
      - Pour tout z ϵ X faire
        - » Res[x,y] ← Min(Res[x,y], M[x,z] + N[z,y])
      - FinPour
    - FinPour
  - FinPour
- FinCode

Complexité

- $O(n^3)$ opérations

Algorithme des plus courts chemins : opération matrice (entête)

- Algorithme PCCOMOM
  - Donnée :
    - M la matrice d'un graphe pondéré
  - Résultat :
    - Res une matrice
  - Variables
    - i entier
    - Inter, Inter2 : deux matrices

Algorithme des plus courts chemins : opération matrice (code)

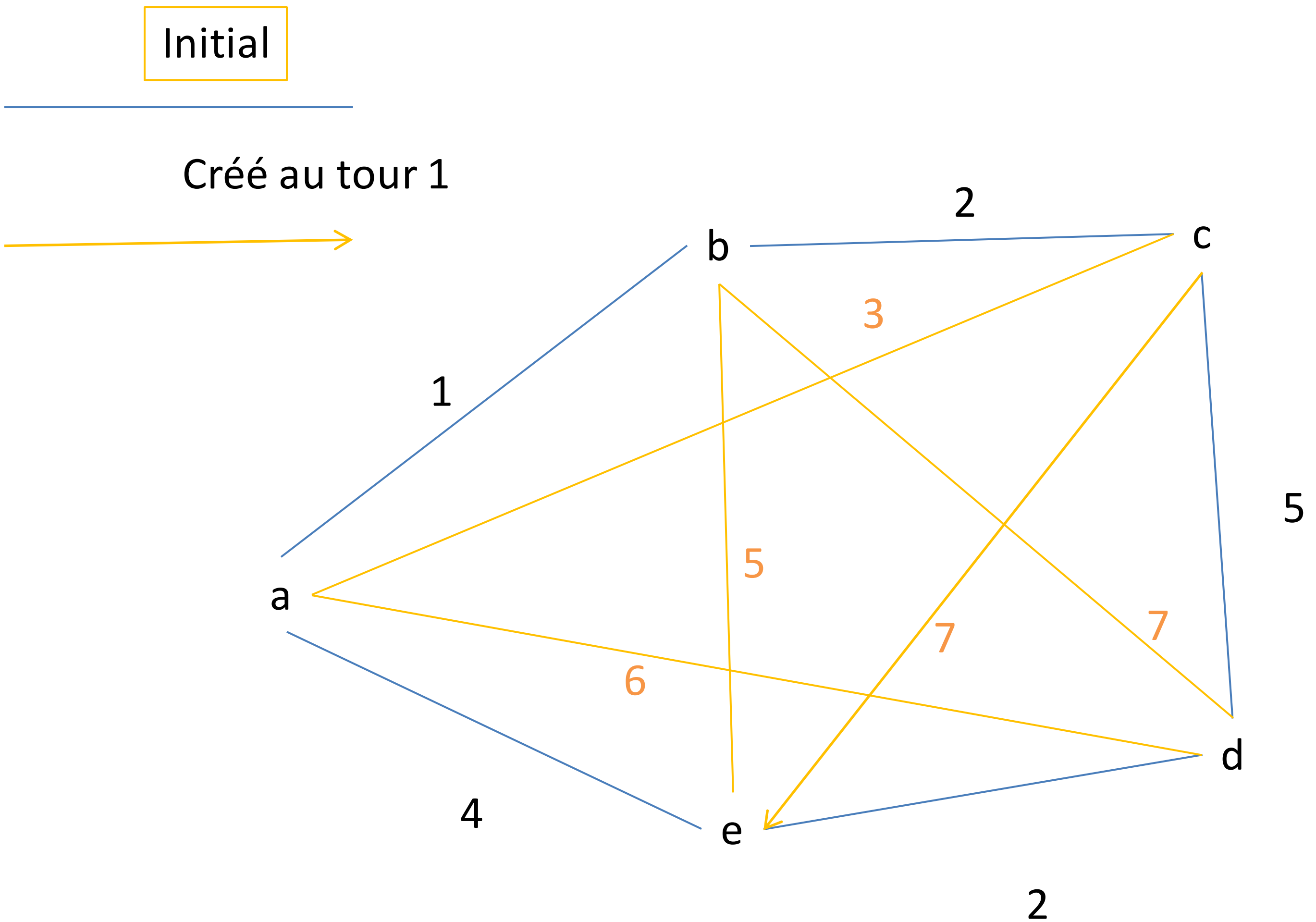
- DébutCode
  - Inter ← M; Inter2← M;// par duplication
  - Pour tout x ϵ X faire
    - Inter[x,x]← 0; Inter2[x,x] ← 0;
  - FinPour
  - Pour i ← 2 à n faire
    - OpMat(Inter, Inter2, Res); Inter ← Res;//Duplication
  - FinPour
- FinCode

Propriétés

- A la fin du kème tour de boucle, Inter[u,v] contient le poids d'un chemin de u à v de poids minimal et de longueur au plus k+1. C'est l'invariant de l'algorithme

- Complexité $O(n^4)$

Exemple



Question

- Comment adapter cet algorithme en système distribué ?

Répartition des données

- Chaque nœud contiendra une ligne de la matrice le nœud i contiendra la ligne i de la matrice.
- Ci-dessous la table du nœud A Les liens sont étiqueté par le nom du nœud extrémité

TR	A	B	C	D	E	F	G	H	I
A	(0,_)	(1,B)	(7,C)	(4,D)	(1,E)	(∞,_)	(∞,_)	(∞,_)	(∞,_)

Hypothèses

- On considère que tous les nœuds connaissent les identités de tous les nœuds du réseau.
- C l'ensemble de tous les canaux d'un nœud
- Ces canaux sont numérotés de 0 à δ-1
- Pour chaque canal i : Coût(i) donne le poids du canal dans le chemin. Les poids sont tous positifs.

Constante et Variables

- Constante : IdLoc identité du nœud
- Variables
- TR un tableau de couple (poids,canal) indicé par les identités du réseau :
  - TR[IdLoc] $\leftarrow$ (0,_)
  - TR[Id] $\leftarrow$ (∞,_) quand Id $\neq$ IdLoc
- Mess(Identité,Poids) un message

Action1

Spontanée (dois être effectuer par tous avant toute réception de message)

Envoyer Mess(IdLoc, 0) sur C

FinSpontanée

Action2

A la réception de Mess(Id,p) sur le canal i

(ap,ac) ← TR[Id]

Si ap > p + Coût(i) alors

TR[Id] ← (p+Coût(i),i)

Envoyer Mess(TR[Id]) sur C // tous les voisins

FinSi

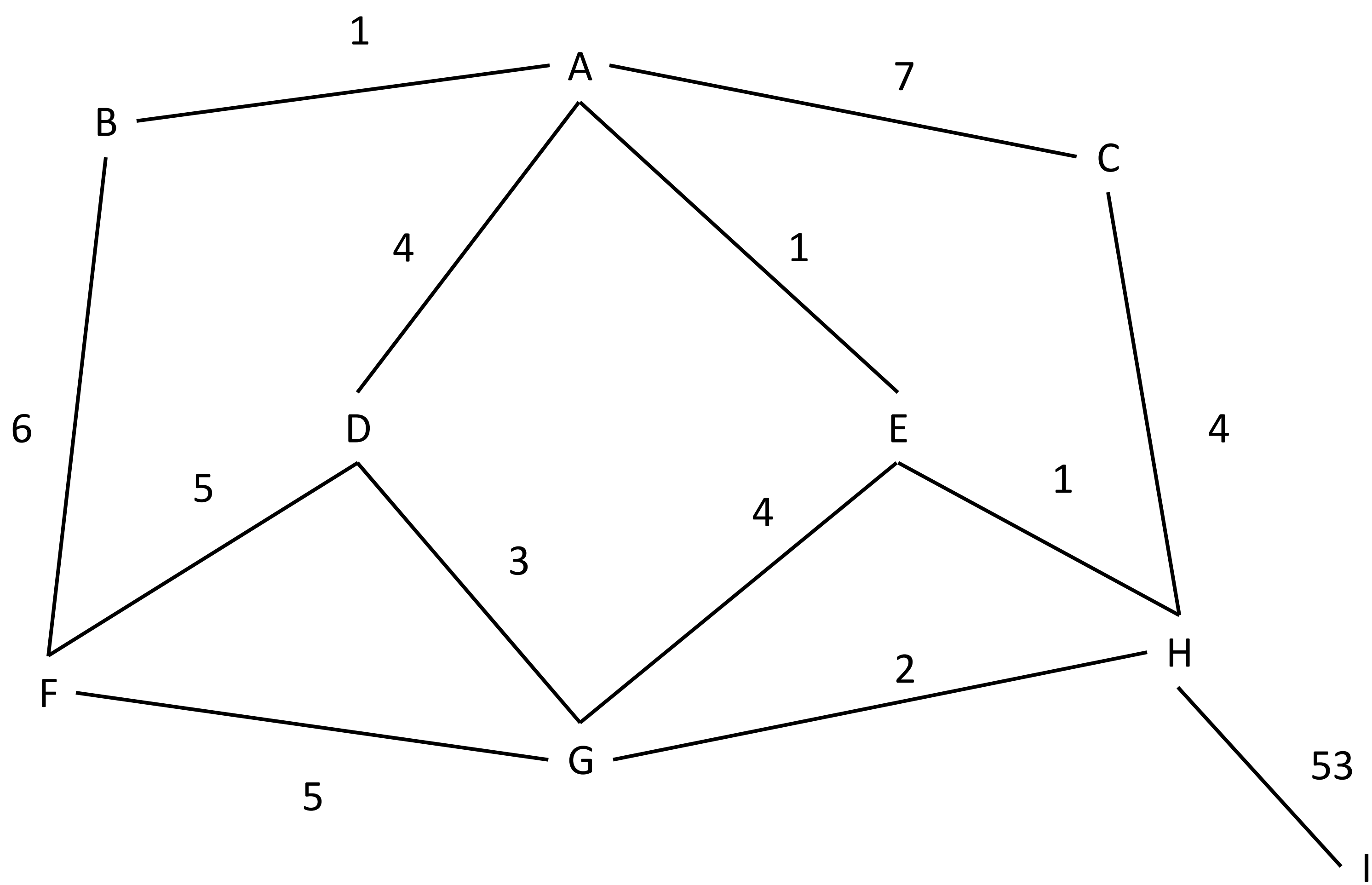
FinAction

Exercice

- Donnez une exécution de cet algorithme sur le graphe donné page suivante

- Calculez la complexité Chandy-Misra :
« Distributed computation on graphs: shortest path algorithms »

Exemple



Sujets de réflexion

- Comment faire en sorte que les chemins soient construits à la manière de BF, c'est-à-dire les chemins de longueur 0 puis 1 puis 2…

- Comment détecté la terminaison ?

IDENTITÉS INCONNUES ?

Ajuster l’algorithme

- On ne peut plus avoir une matrice.
- A la place nous allons devoir gérer une liste de triplets (Identité, Distance, Canal)
- Initialement LRL = <(IdLoc, 0, null)>

Opération sur les listes

- Fonctions Classiques : ListeVide (), Premier(L), TestListeVide(L), Suite(L), AjoutTête(e, L)

- Fonction ExisteId (Id, L) indique si la liste L contient un triplet (a,b,c) tel que a = Id

- Fonction CapteTrip(Id, L) renvoie le triplet (a,b,c) de L tel que a = Id.

Constantes

- IdLoc : l'identité locale du nœud.
- C ensemble des canaux.

Variables

- Nous noterons LRL la Liste de Routage Locale

- Initialement LRL =<(IdLoc,0,null)>

Information à transmettre

- Nous devons transmettre un couple (Id, Dist).

Action1

Spontanée (dois être effectuer par tous avant toute réception de message)

Envoyer Mess(IdLoc, 0) sur C

FinSpontanée

Action2

A la réception de Mess(Id,p) sur le canal i

- Si ExisteId (Id, LRL) alors
  - (Id, Dist, can) ← CapteTrip (Id,LRL)
  - Si Dist > p + Coût(i) alors
    - Remplace dans LRL (Id, Dist, can) par (Id, p+Coût(i), i)
    - Envoyer Mess(Id, p+Coût(i)) sur C – {i} // tous les voisins
  - FinSi
- Sinon
  - AjoutTête ((Id, p+Coût(i), i), LRL)
  - Envoyer Mess(Id, p+Coût(i)) sur C – {i} // tous les voisins
- FinSi

FinAction

RÉSEAUX DYNAMIQUES ?

Chalenge

- Adapter cet algorithme dans le cas où un nœud peux entrer dans le réseau ou en sortir
- En particulier si un nœud entre tardivement ou se déconnecte trop rapidement
- Vous devrez considérer ces deux cas comme des actions spéciales initiées par ces nœuds.

Chalenge

- Quel bel exercice

Réduire la taille de la table de routage

- Jusqu'à présent un table de routage code sur chaque nœud tous les couples (Id, canal) utiles pour acheminer les messages. Nous avons donc une taille de la table de routage égale à : $\Theta(n\log(w))$ avec
  - n le nombre de nœud
  - w la valeur maximale d'une identité

Réduire la taille de la table de routage

- Une solution consiste à utiliser le routage par intervalles.
  - $[i,j] = \{i, i+1, i+2, \ldots, j-1, j\}$ quand $i < j$
  - $[i,j] = \{i, i+1, \ldots, n-1, 0, 1, \ldots, j\}$ dans le cas contraire.

Routage par intervalle

- Une table de routage telle que pour tout nœud x :
  - Les destinations associées à chaque canal peuvent être représentées par un intervalle
  - L'intersection des intervalles de deux canaux distincts est vide
  - Chaque destination est présente dans un canal.
- Est un routage par intervalle

Exemple : poids des arêtes 1

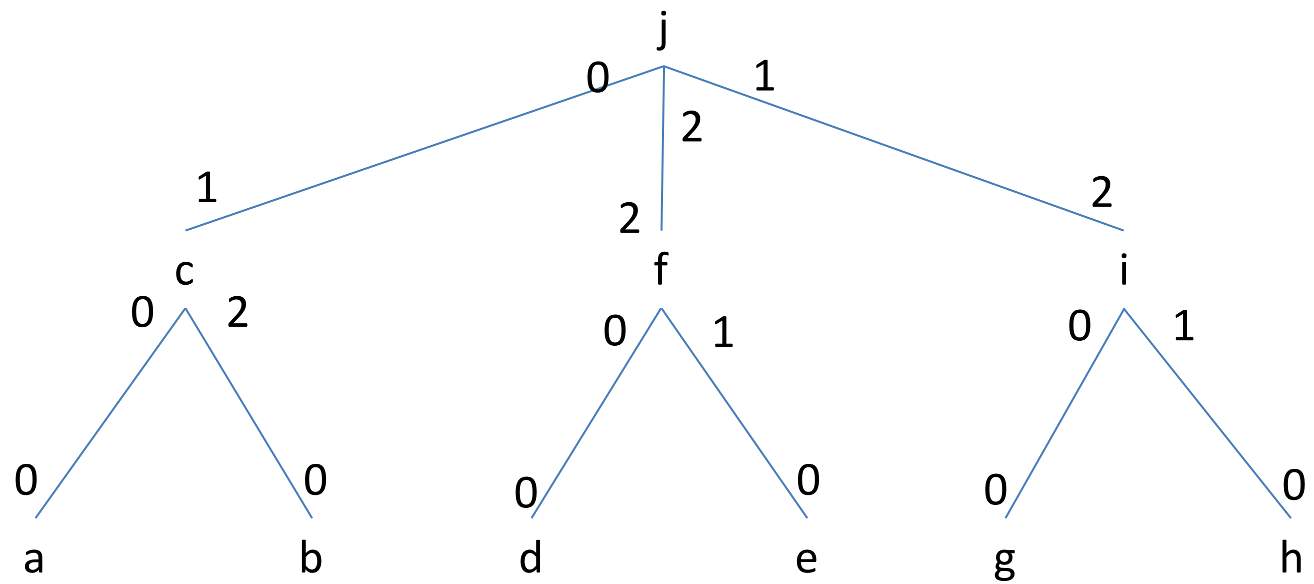


Table de routage de f

canal	**0**	**1**	**2**
Dest	[d]	[e]	[g..c]

Routage par k intervalles

- Une table de routage telle que pour tout nœud x :
  - Les destinations associées à chaque canal peuvent être représentées par k intervalles
  - L'intersection des intervalles de deux canaux distincts est vide
  - Chaque destination est présente dans un canal.
- Est un routage par k intervalles

Avantages/Inconvénient

- Rendre les tables locales plus compactes

- L’identité des nœuds doit être adaptée au réseau (Renommage).
- Dans le cas général on perd l’acheminement du message par le chemin de poids minimal.

Exercices

- Montrez que les parcours d’un arbre en préfixé, postfixé et infixé donne un routage par intervalle sur chaque nœud.

- Montrez que si x est un point d’articulation, alors il est possible de créer au moins un intervalle de routage pour chaque nœud. Du réseau.

Interblocage

- Est-il possible qu'une pénurie de resources rende l'acheminement des messages impossibles ?

- Peut-on prévenir ces difficultés ?

Interblocage