INF431

X-2006 2 juillet 2008: Contrôle Classant
CORRIGÉ

Les réponses devront être claires et précises. Il sera tenu compte de la concision. Les parties I et II sont indépendantes.

Partie I : Preuves de l'algorithme de Floyd-Warshall

On s'intéresse à un algorithme de calcul des plus courts chemins dans un graphe orienté, où chaque arête a un poids réel positif ou nul. Les sommets du graphe sont numérotés de 1 à n. L'ensemble des arêtes du graphe est représenté par une matrice d'adjacence, matrice carrée, de côté n, à coefficients dans $\mathbb{R} \cup \{\infty\}$ dans laquelle $A_{ij} = \infty$ signifie qu'il n'y a pas d'arête allant du sommet i au sommet j, et sinon A_{ij} est le poids de cette arête.

On note $A \times B$ la multiplication de deux matrices A et B selon l'algèbre « min-plus ». En d'autres termes, la matrice $A \times B$ est définie par :

$$\forall i, j \qquad (A \times B)_{ij} = \min_k (A_{ik} + B_{kj})$$

On remarque que l'opération $\times$ est associative mais non commutative en général. Pour k entier supérieur à 1, on note A^k la puissance k-ème de A avec la multiplication $\times$.

Une matrice est dite *positive* si et seulement si tous ses coefficients sont positifs ou nuls. On note $A \geq B$ le cas où pour tout i, j $A_{ij} \geq B_{ij}$.

On suppose que A est positive et à diagonale nulle. On admettra sans démonstration que la condition A positive à diagonale nulle est équivalente à la condition A positive et $A \geq A \times A$.

Question 1 Montrer que A^ℓ est à diagonale nulle pour tout entier $\ell > 0$. Montrer que le coefficient ij de A^ℓ est le poids minimum de tous les chemins entre le sommet i et j passant par au plus $\ell - 1$ sommets intermédiaires. ◇

Solution. A^ℓ est trivialement positive et comme $A \geq A \times A$ et que la multiplication est monotone : $A^\ell \geq A^\ell \times A^\ell$, donc A^ℓ est à diagonale nulle. On prouve par récurrence que le coefficient ij dans A^ℓ est le poids minimum des chemins du sommet i au sommet j passant par au plus $\ell - 1$ sommets intermédiaires. La proposition est vraie pour $\ell = 1$, et $A^{\ell+1}_{ij} = \min_k\{A_{ik} + A^\ell_{kj}$ on a manifestement A_{ij} égal à la borne inférieure du poids de tout chemin partant de i passant par un voisin k de i (ou i lui-même si $k = i$ mais le poids $A_{ii} = 0$) suivi d'un chemin de k à j passant par au plus $\ell - 1$ sommets intermédiaires, c'est à dire le poids d'un chemin de i à j passant par au plus ℓ sommets intermédiaires.□

Question 2 Démontrer que, si A est positive et à diagonale nulle, alors la suite (A^ℓ) est stationnaire à partir d'un certain rang. Donner une borne supérieure sur ce rang. ◇

Solution. L'entrée ij de la matrice A^ℓ donne la longueur du plus court chemin parmi ceux reliant le sommet i au sommet j et formés d'au plus $\ell - 1$ sommets intermédiaires.

Si un chemin π emprunte au moins $n + 1$ sommets, donc emprunte au moins deux fois un certain sommet, donc contient un cycle. La matrice A étant positive, ce cycle est de longueur positive ou nulle, donc le chemin π privé de ce cycle est de longueur inférieure ou égale à celle du chemin π. Un plus court chemin admet donc au maximum $n - 2$ sommets intermédiaires distincts.

Il découle de cela que $A^{n-1} = A^n$. La suite est stationnaire au plus à partir du rang $n - 1$. □

L'algorithme de calcul des plus courts chemins est le suivant :

— précondition : A est positive et à diagonale nulle
fonction PLUSCOURTSCHEMINS(A)
$C \leftarrow A$
$B \leftarrow C \times C$
tant que $B \neq C$
faire $\begin{cases} C \leftarrow B \\ B \leftarrow C \times C \end{cases}$
renvoyer B
— postcondition : $\exists k \geq 1 \quad (B = A^k \wedge B = B \times B)$

Question 3 Montrer que la postcondition $\exists k \geq 1 \quad (B = A^k \wedge B = B \times B)$ implique que la matrice B obtenue à la fin de PLUSCOURTSCHEMINS donne les poids des plus courts chemins dans le graphe valué par la matrice A. ◇

Solution. En d'autres termes, on doit vérifier que $\exists k \geq 1 \quad (B = A^k \wedge B = B \times B)$ implique que B est la limite de la suite des puissances de A. C'est le bien cas, car si $B = A^k$ et si $A \times B \neq B$ alors, comme A^k décroit (à partir de l'inégalité $A \geq A \times A$) alors $A^k > A^{k+1}$ et par conséquent $A^k > A^{2k}$ ce qui contredit $B = A^k = A^{2k}$. Donc la suite est stationnaire au moins à partir du rang k, et admet la valeur stationnaire B. □

Question 4 Quel invariant peut-on attribuer à la boucle, au sein de la fonction PLUSCOURTSCHEMINS, pour pouvoir établir, selon la méthode de Hoare, la postcondition espérée ? On rappelle que la règle de Hoare associée aux boucles impose trois conditions : l'invariant doit être satisfait initialement, préservé par chaque itération, et doit impliquer la propriété souhaitée à la sortie. Énoncer ces trois conditions sous forme de formules logiques, et vérifier leur validité. ◇

Solution. Nous allons utiliser l'invariant suivant, que nous noterons I :

$$\exists k \geq 1 \quad (C = A^k \wedge B = C \times C)$$

L'invariant est clairement vrai initialement. Si on souhaite appliquer à la lettre la méthode de Hoare, on applique deux fois successivement la règle d'affectation : pour cela, on remplace B par $C \times C$ puis C par A dans la formule ci-dessus, ce qui donne :

$$\exists k \geq 1 \quad (A = A^k \wedge A \times A = A \times A)$$

On vérifie alors que cette formule est vraie à l'entrée dans la procédure, ce qui immédiat (choisir $k = 1$).

Pour vérifier que l'invariant est préservé, on doit vérifier le triplet

$$\{I \wedge B \neq C\}\ C \leftarrow B; B \leftarrow C \times C\ \{I\}$$

L'hypothèse $B \neq C$ étant ici inutile, on l'abandonne. En appliquant à nouveau deux fois successivement la règle d'affectation, on constate que cette obligation revient à vérifier que I implique

$$\exists k \geq 1 \quad (B = A^k \wedge C \times C = C \times C)$$

Le premier membre de cette conjonction découle bien de I : en effet, si $C = A^k$ et $B = C \times C$, alors $B = A^{2k}$. Le second membre est une tautologie.

Pour vérifier que la postcondition est satisfaite, il reste à vérifier que $\neg(B \neq C) \wedge I$ implique bien la postcondition, ce qui est trivial. □

On rappelle qu'un variant est une expression entière, dont on doit pouvoir prouver qu'elle est positive ou nulle et qu'elle décroît strictement à chaque itération.

Question 5 Quel variant peut-on attribuer à la boucle, au sein de la fonction PLUSCOURTSCHEMINS, pour pouvoir établir, selon la méthode de Hoare, la terminaison de la fonction ? On pourra s'appuyer sur le résultat obtenu lors de la question 2. ◇

Solution. D'après la question 2, la suite est stationnaire à partir du rang $n-1$. Imaginons que l'on matérialise, dans le code, une variable k telle que $C = A^k$. (On initialise donc k à 1 avant la boucle, et on ajoute l'instruction $k \leftarrow 2 \times k$ dans la boucle.) L'invariant de boucle devient $1 \leq k \leq 2n \wedge C = A^k \wedge B = C \times C$, sans quantification existentielle. Cet invariant, avec la condition $B \neq C$, implique $k < n$. Un variant approprié est donc $2n - k$. La valeur de cette expression est bien positive ou nulle, et décroît strictement à chaque itération, puisqu'elle passe de $2n - k$ à $2n - 2k$, pour $k \geq 1$.

Bien sûr, le fait de modifier le code pour faire apparaître une variable k, qui ne joue aucun rôle dans l'algorithme, est artificiel. En pratique, certains outils basés sur la méthode de Hoare, comme l'outil Krakatoa/Why que nous avons utilisé en TD, permettent de faire ces modifications sous forme de « code fantôme ».

On pourrait noter que la valeur de la matrice B (ou C) décroît à chaque itération, pour l'ordre point par point. Cependant, cette matrice étant à coefficient réels, cet ordre n'est pas bien fondé, et il n'en découle donc pas directement que la boucle termine. □

Partie II : Routage à la demande et super-diffusion

1 Eléments de base

1.1 Un réseau de télécommunication

On considère un réseau de télécommunication constitué d'un ensemble V de n postes numérotés de 1 à n reliés par des liens de télécommunication qui peuvent être des câbles, des liens sans fil, etc. On considère le graphe $G(V, E)$ non-orienté où les postes constituent les sommets et chaque arête $\{i, j\}$ signifie qu'il existe un lien de télécommunication direct entre i et j. Soit un poste i, on appelle $N(i)$ l'ensemble des voisins du sommet i dans le graphe $G(V, E)$. On suppose de plus que le graphe $G(V, E)$ est connexe.

1.2 Les files d'attente parallèles et bases de données locales

Sur chaque poste i, il existe deux types de files d'attente gérées en parallèle :
- une file d'attente de réception qui collecte les paquets reçus des voisins de i ;
- des files d'attentes d'émission, une pour chaque voisin de i, qui collectent les paquets mis en attente d'émission vers ces voisins.

1.3 Le paquet et ses champs

Un paquet P est un 5-uplet (destination,source,type,sequence,contenu).
- le champ DESTINATION(P) indique l'identificateur du poste concerné par le paquet P ;
- le champ SOURCE(P) indique l'identificateur du poste qui a créé le paquet P ;
- le champ TYPE(P) indique le type du paquet P. Les types des paquets sont :
  - le type *Donnée* qui désigne les paquets de données ;
  - le type *Ack* qui désigne les paquets d'acquittement, émis en retour par tout poste après chaque réception de paquets de type autre que *Ack* ;
  - le type *Requête* qui désigne les paquets de recherche de route pour la construction de la table de routage et qui sera décrit en section 4 ;
  - le type *Réponse* qui désigne les paquets de réponse à la recherche de route et qui sera décrit en section 4 ;
- le champ SEQUENCE(P) indique le numéro de séquence du paquet P.
- le champ CONTENU(P) contient les données du message.

Si le paquet P n'est pas du type *Ack* alors le numéro de séquence du paquet P est un numéro qui le différencie de tous les autres paquets émis par la source du paquet, par exemple un numéro incrémenté à chaque nouveau paquet créé (la fonction de création du numéro de séquence ne sera pas traitée). En conséquence le couple (SEQUENCE(P), SOURCE(P)) est un identificateur unique du paquet P. Il est

important de noter que les relais ne modifient pas le numéro de séquence du paquet P qui est seul créé par SOURCE(P).

Si le paquet P est du type *Ack*, alors son numéro de séquence est égal au numéro de séquence du paquet qu'il acquitte. Le contenu du paquet P est alors l'identifiant de la source du paquet qu'il acquitte.

1.4 Primitives

On définit les procédures et fonctions suivantes dans chaque i poste du réseau.

La procédure EMISSION(P, S) fait que le poste i met le paquet P dans ses files d'émission vers les postes j qui sont dans S, où S est un sous ensemble de $N(i)$.

La fonction RECEPTION() enlève le premier paquet en attente dans la file de réception du poste i et le retourne. S'il n'y a pas de paquet en attente alors la fonction retourne **nil**.

La fonction LASTRELAY(P) retourne le numéro du poste voisin de i qui a envoyé le paquet P.

Un paquet est sensé être routé de sa source vers sa destination, ce qui fait que les champs SOURCE(P) et DESTINATION(P) sont généralement différents des identifiants des émetteurs et récepteurs qui font relais.

2 Emission et réception de base

Tous les paquets doivent être acquittés à l'exception des paquets d'acquittement. Un paquet qui n'est pas de type *Ack* est dit "acquittable".

Chaque poste i gère un ensemble appelé "*Archive*" qui contient les couples (SEQUENCE(P), SOURCE(P)) de tous les paquets acquittables P reçus par le poste i et un ensemble "*Attente*" qui contient les paquets émis par le poste i en attente d'acquittement.

Un poste i qui reçoit un paquet P de type *Donnée* exécute la procédure SENDACK suivante :

procédure SENDACK(P)
$s \leftarrow$ SEQUENCE(P)
$j \leftarrow$ SOURCE(P)
si $(s, j) \notin$ *Archive* **alors** *Archive* $\leftarrow$ *Archive* $\cup \{(s, j)\}$
$\ell \leftarrow$ LASTRELAY(P)
$PA \leftarrow (\ell, i, \mathit{Ack}, s, j)$
EMISSION($PA, \{\ell\}$)

Le poste i qui désire envoyer un paquet P acquittable de numéro de séquence s et de source j, vers un sous-ensemble de voisins S, maintient un ensemble *AckRestants*(s, j) constitué des voisins qui n'ont pas encore acquitté le paquet P. La transmission est initialisée par la procédure INITTRANSMIT(P, S) :

procédure INITTRANSMIT(P, S)
AckRestants(SEQUENCE(P), SOURCE(P)) $\leftarrow S$
TRANSMIT(P, S)
Attente $\leftarrow$ *Attente* $\cup \{P\}$

Noter que le paquet P est inséré dans un ensemble *Attente* et en sera retiré quand le poste i aura reçu au moins un acquittement du paquet P de la part de chacun des membres de S.

2.1 Timer et période d'émission

Périodiquement un timer met une variable *TimeOut* à **vrai**, ce qui déclenche la retransmission des paquets de l'ensemble *Attente*. Le timer et sa période sont gérés indépendamment et en parallèle et ne seront pas étudiés ici. L'intervalle de temps entre deux passages de la variable *TimeOut* à **vrai** est appelé période d'émission.

A la fin de chaque période d'émission, indiquée par le passage de la variable *TimeOut* à **vrai**, il faut procéder à la réémission de tous les paquets de l'ensemble *Attente*.

On donne un exemple de tableau d'émission et de réception d'un paquet $P = (i, j, \mathit{Donnée}, s, \mathit{data})$ de séquence s par le poste 1 vers ses voisins 2 et 3. On suppose que tous les paquets sont perdus lors de la première période, qu'ensuite il n'y a plus de perte et que les acquittements sont reçus pendant la même période d'émission que le paquet acquitté.

poste	transmissions en période 0	transmissions en période 1
1	$(i, j, \textit{Donnée}, s, \textit{data})$	$(i, j, \textit{Donnée}, s, \textit{data})$
2	-	$(1, 2, \textit{Ack}, s, j)$
3	-	$(1, 3, \textit{Ack}, s, j)$

Question 6 Le poste 1 a deux voisins : 2 et 3. Le poste 1 doit transmettre un paquet $P = (i, j, \textit{Donnée}, s, \textit{data})$ à ses deux voisins. On numérote les périodes entre deux expirations de timer 0, 1, 2, *etc*. Les paquets émis vers le poste 2 pendant les périodes de rang pair sont perdus, alors que ceux émis pendant les périodes de rang impair sont reçus de manière instantanée. Les paquets vers le poste 3 sont perdus pendant les périodes impaires et reçus instantanément pendant les périodes de rangs pairs. On suppose que le premier paquet d'acquittement du poste 2 et du poste 3 sont systématiquement perdu. Quand ils ne sont pas perdus les paquets d'acquittement sont reçus instantanément et sans perte pendant la même période d'émission. Donner le tableau des émissions et des réceptions entre les postes 1, 2 et 3 jusqu'à ce que le paquet soit correctement reçu. ◇

Solution.

poste	période 0	période 1	période 2	période 3
1	$(i, j, \textit{Donnée}, s, \textit{data})$	$(i, j, \textit{Donnée}, s, \textit{data})$	$(i, j, \textit{Donnée}, s, \textit{data})$	$(i, j, \textit{Donnée}, s, \textit{data})$
2	-	$(1, 2, \textit{Ack}, s, j)$	-	$(1, 2, \textit{Ack}, s, j)$
3	$(1, 3, \textit{Ack}, s, j)$	-	$(1, 3, \textit{Ack}, s, j)$	-

2.2 Traitement des acquittements

Quand un poste i reçoit un acquittement (type *Ack*), il doit remettre à jour l'ensemble $\textit{AckRestants}(s, j)$ correspondant au paquet acquitté.

Question 7 Ecrire la procédure RECEIVEACK(A) que doit exécuter le poste i quand il reçoit un acquittement A. Si le paquet A acquitte le paquet P de numéro de séquence s et de source j, on prendra soin de retirer de l'ensemble *Attente* le paquet P quand l'ensemble associé $\textit{AckRestants}(s, j)$ devient vide, en utilisant la procédure ELIMINE(s, j) qu'on ne demande pas d'écrire. ◇

Solution. **procédure** RECEIVEACK(A)
$s \leftarrow$ SEQUENCE(A)
$j \leftarrow$ CONTENU(A)
si $(\textit{AckRestants}(s) \neq \emptyset)$
alors $\begin{cases} \ell \leftarrow \text{LASTRELAY}(A) \\ \textit{AckRestants}(s) \leftarrow \textit{AckRestants}(s) \setminus \{\ell\} \\ \textbf{si } (\textit{AckRestants}(s, j) = \emptyset) \textbf{ alors } \text{ELIMINE}(s, j) \end{cases}$

Noter la condition $\textit{AckRestants}(s, j) \neq \emptyset$ pour éviter d'activer la procédure ELIMINE(s, j) plusieurs fois en cas de répétition d'acquittement. □

Si à la fin de la période d'émission l'ensemble *Attente* n'est pas vide, il faut procéder à la retransmission de tous ses éléments.

Question 8 Ecrire la procédure RETRANSMIT() qui renvoie tous les paquets contenus dans la file d'attente *Attente* vers les voisins qui ne les ont pas encore acquittés. ◇

Solution. **procédure** RETRANSMIT()
pour $P \in \textit{Attente}$
faire $\begin{cases} s \leftarrow \text{SEQUENCE}(P) \\ \textbf{si } (\textit{AckRestants}(s) \neq \emptyset) \\ \textbf{alors } \{\text{TRANSMIT}(P, \textit{AckRestants}(s)) \end{cases}$ □

Question 9 On suppose que le poste i n'a qu'un seul paquet P de type *Donnée* à transmettre et qu'il ne reçoit que des acquittements. Ecrire le pseudo-code de la procédure SINGLETRANSMIT(P) que le poste i doit exécuter quand le paquet P doit être transmis à tous ses voisins.

Solution. **procédure** SINGLETRANSMIT(P)
$s \leftarrow$ SEQUENCE(P)
INITTRANSMIT(P)N(i)
tant que (*AckRestants*(s) $\neq \emptyset$)
faire { $PA \leftarrow$ RECEPTION()
tant que $PA \neq$ **nil**
faire { **si** TYPE(PA) = *Ack* **alors** RECEIVEACK(PA)
$PA \leftarrow$ RECEPTION()
si (*TimeOut* = **vrai**)
alors { RETRANSMIT()
TimeOut $\leftarrow$ **faux** □

3 Usage de la table de routage

On suppose que chaque arête $\{j, \ell\}$ a un coût $\omega_{j\ell}$ qu'on suppose symétrique $\omega_{\ell j} = \omega_{j\ell}$ et strictement supérieur à zéro. Le poids de l'arête $\{\ell, j\}$ est supposé connu des postes ℓ et j.

La table de routage d'un poste j est un tableau *Route* tel que *Route*$(k) = (j, d)$ où $\ell =$ NEXTRELAY(k) est le premier voisin de j sur la route de j à k, et cette route a pour poids cumulé $d =$ ROUTEDISTANCE(k) qui est égal à la somme des poids des arêtes de la route. Si cette route n'a pas encore été découverte alors *Route*$(k) =$ **nil**, sauf si $j = k$ alors *Route*$(k) = (k, 0)$.

Pour l'instant on ne cherche pas à optimiser le poids cumulé de la route. On introduit dans chaque paquet un nouveau champ appelé distance qui donne la distance cumulée depuis l'origine du paquet. Donc un paquet est maintenant un 6-uplet (destination, source, type, séquence, contenu, distance).

Question 10 Pour cette question on suppose que le graphe du réseau est $V = \{1, 2, 3, 4\}$ avec $E = \{\{1, 2\}, \{2, 3\}, \{3, 4\}\}$. Sur le poste 1 : *Route*$(4) = (2, 3.5)$, sur 2 *Route*$(4) = (3, 1.1)$ sur 3 *Route*$(4) = (4, 0.7)$ et sur 4 *Route*$(4) = (4, 0)$. Donner les valeurs des poids des liens du réseau, et donner l'évolution du champ distance d'un paquet P émis à partir de 1 vers 4, et lors de ses réémissions sur 2 et 3. ◇

Solution. $\omega_{1,2} = 2.4$, $\omega_{2,3} = 0.4$, $\omega_{3,4} = 0.7$, et le champ distance du paquet prend les valeurs successives de 0, 2.4 et 2.8. □

Question 11 Ecrire la procédure FORWARD(P) que doit exécuter le poste j pour acheminer un paquet P, qu'il vient de recevoir, vers son prochain relai. Si le poste j est la destination du message, on ne fait rien. On n'oubliera pas la mise à jour du champ distance dans le corps du message à retransmettre.

Solution. **procédure** FORWARD(P)
$s \leftarrow$ SEQUENCE(P)
$i \leftarrow$ SOURCE(P)
si ($(s, i) \notin$ *Archive*)
alors { *Archive* $\leftarrow$ *Archive* $\cup \{(s, i)\}$
$k \leftarrow$ DESTINATION(P)
si ($k \neq j$)
alors { $d \leftarrow$ DISTANCE(P) + ω_{ij}
$\ell \leftarrow$ NEXTRELAY(k)
$PN \leftarrow (k, i,$ *Donnée*$, s,$ DATA$(P), d)$
INITTRANSMIT($PN, \{\ell\}$)

Question 12 Adapter le code de la question 9 et utiliser le résultat de la question 11 pour écrire le pseudocode de la procédure RELAI() que doit exécuter un poste qui ne fait que du relayage sans jamais créer de paquet de données et qui a sa table de routage fixée une fois pour toute. Ne pas oublier les acquittements. ◇

Solution. **procédure** RELAI()
tant que (**vrai**)
faire {
  $P \leftarrow$ RECEPTION()
  tant que ($P \neq$ **nil**)
  faire {
    si (TYPE(P) $\neq$ *Ack*)
    alors {
      SENDACK(P)
      si (TYPE(P) = *Donnée*) **alors** FORWARD(P)
    si (TYPE(P) = *Ack*) **alors** RECEIVEACK(P)
    $P \leftarrow$ RECEPTION()
  si (*TimeOut* = **vrai**)
  alors {
    RETRANSMIT()
    TimeOut $\leftarrow$ **faux**

4 Construction de la table de routage

Dans cette section on regarde un protocole de construction de table de routage basé sur un processus de découverte de route. Comme les communications sont souvent symétriques et nécessitent de mettre en place des routes dans les deux sens, le processus va donc se dérouler en deux phases :

- phase découverte : le poste i qui veut établir une connexion avec le poste k fait construire une route du poste k au poste i.
- phase réponse : le poste k fait construire une route du poste i au poste k en utilisant la route inverse.

Pour cela on introduit deux nouveaux types de paquet :

- Le type *Requête* qui permet une recherche de route quand elle est absente de la table de routage de la source pour une destination donnée. Le paquet *Requête* est diffusé dans l'ensemble du réseau ;
- le type *Réponse* qui achemine la connaissance de la route découverte à partir de la destination vers la source du paquet *Requête*.

Ces deux types de paquets ont un champ CONTENU mis à **nil**.

4.1 Découverte de route

Lorsqu'un poste i désire ouvrir une connexion avec un poste distant k, il crée un paquet Q_{ik} de type *Requête* avec source i, destination k, distance initialisée à zéro, et un numéro de séquence. Le paquet *Requête* est envoyé à tous les voisins de i.

Un poste $j \neq i$ qui reçoit la première fois le paquet Q_{ik}, supposons du poste ℓ, effectue les opérations suivantes :

- il met à jour sa table de routage vers la source i de Q_{ik} en faisant

$$\textit{Route}(i) = (\ell, \text{DISTANCE}(Q_{ik}) + \omega_{j\ell})$$

- il retransmet le message vers tous ses voisins sauf ℓ de qui il a reçu le paquet Q_{ik}.

Ce message est acquittable comme un message de type *Donnée*. Afin d'éviter les boucles, les autres réceptions du messages Q_{ik} par le poste j ne provoquent que l'acquittement du message vers son émetteur.

Question 13 On fait l'hypothèse que les liens entre les postes constituent des canaux finis (les messages finissent par passer au bout d'un nombre fini d'essais) et que les postes sont obstinés (on retransmet les messages jusqu'à ce qu'ils soient acquittés par tous les voisins concernés). Montrer que tous les postes du réseau finissent par recevoir le paquet Q_{ik}. ◇

Solution. Par l'absurde, soit un poste j qui n'a pas reçu le paquet, voisin d'un poste ℓ qui l'a reçu. Le poste ℓ répètera le paquet R_{ik} jusqu'à ce que le poste j l'acquitte. Donc le paquet finit par passer et l'acquittement de j finit par passer. Donc j a aussi reçu le paquet Q_{ik}. □

Dans cette section on ne cherche pas à optimiser les routes vers le poste i. On dit que la table de routage est cohérente si

- dans le cas où *Route*$(i) \neq$ **nil**, la route décrite par les NEXTRELAY(i) sur les relais successifs donne une route vers le poste i ;
- la distance indiquée par la table de routage est effectivement la somme des poids des arêtes de cette route.

On notera que la cohérence implique que les routes déduites des tables sont des chemins simples sans boucles, dans la mesure où les distances sont strictement décroissantes.

Question 14 On suppose que tous les tables de routages *Route*(i) sont à **nil** (sauf sur le poste i) au moment où le poste i commence la diffusion de son message Q_{ik}. Montrer que les tables de routage de tous les postes du réseau vers le poste i restent cohérentes. Quelle structure les tables de routage induisent-elle sur le graphe du réseau ?

Solution. On procède par récurrence sur l'ordre de réception du paquet Q_{ik}. A l'ordre zéro le poste i a sa table de routage cohérente, puisqu'elle pointe sur le poste i. Supposons que les tables de routage sont cohérente sur tous les routeurs qui ont reçu le paquet R_{ik} dans un ordre de réception inférieur ou égal à m. Considérons que le routeur ℓ reçoit le paquet Q_{ik} à l'ordre $m+1$. Supposons qu'il le reçoit du poste j qui ayant reçu R_{ik} à un ordre inférieur ou égal à m à sa table de routage cohérente.

La table de routage du poste ℓ est cohérente puisque la distance est augmentée de $\omega_{j\ell}$ avec celle du poste j et que la route ne peut pas reboucler sur ℓ puisque la distance y décroit strictement.

A la fin les tables de routage induisent une structure d'arbre couvrant enraciné sur le poste i. □

4.2 Réponse de route

A la première réception du paquet Q_{ik} le poste k, crée un message R_{ki} de type *Réponse* avec un numéro de séquence spécifique grâce à la fonction CRÉERRÉPONSE(i). Ce message R_{ki} est acheminé comme un message de type *Donnée* vers le poste i (voir section 3) à la différence que sur chaque relai ℓ, recevant le paquet R_{ki} du poste j, met à jour la table de routage vers k :

$$\textit{Route}(k) \leftarrow (j, \text{DISTANCE}(R_{ki}) + \omega_{\ell j})$$

Question 15 Montrer que les tables de routage créées par le paquet R_{ki} sont cohérentes. ◇

Solution. La preuve est la même que celle de la question 14. □

Question 16 Ecrire le pseudocode du traitement des paquets *Requête* et *Réponse* sur le poste ℓ. On ne cherchera pas à détailler la fonction CRÉERRÉPONSE(). Il faut écrire deux fonctions RECEIVEREQUETE(Q) et RECEIVEREPONSE(R) qui assurent le traitement et la retransmission des paquets Q et R respectivement. ◇

Solution. **procédure** RECEIVEREQUETE(Q)
$s \leftarrow$ SEQUENCE(Q)
$i \leftarrow$ SOURCE(Q)
si $((s,i) \notin$ *Archive*$)$
alors
- *Archive* $\leftarrow$ *Archive* $\cup \{(s,i)\}$
- $k \leftarrow$ DESTINATION(Q)
- $j \leftarrow$ LASTRELAY(Q)
- $d \leftarrow$ DISTANCE(Q) $+ \omega_{j\ell}$
- **si** $(\ell \neq i)$
  - **alors** *Route*$(i) \leftarrow (j,d)$
- $NQ \leftarrow (k, i,$ ***Requête***$,$ **nil**$, d)$
- INITTRANSMIT($NQ, N(\ell) \setminus \{j\}$)
- **si** $(\ell = k)$
  - **alors**
    - $R \leftarrow$ CRÉERRÉPONSE(i)
    - INITTRANSMIT($R, \{j\}$)

SENDACK(Q)
procédure RECEIVEREPONSE(R)
$s \leftarrow$ SEQUENCE(Q)
$i \leftarrow$ SOURCE(Q)
si $((s,i) \notin$ *Archive*$)$
alors
- $k \leftarrow$ DESTINATION(Q)
- $j \leftarrow$ LASTRELAY(Q)
- $d \leftarrow$ DISTANCE(Q) $+ \omega_{j\ell}$
- *Route*$(i) \leftarrow (j,d)$
- **si** $(k \neq \ell)$
  - **alors** FORWARD(R)
  - **sinon** *Archive* $\leftarrow$ *Archive* $\cup \{(s,i)\}$
  - SENDACK(R)

□

5 Super-diffusion

Dans la section précédente les postes ne relayaient qu'une seule fois le paquet Q_{ik}. Dans la super-diffusion le poste j relaie le paquet Q_{ik} chaque fois qu'il le reçoit avec une distance strictement inférieure à la précédente.

Pour éviter les confusions entre les acquittements des différentes versions du paquet Q_{ik}, le paquet Q_{ik} est acquitté par un paquet de type *SuperAck* identique au type *Ack* mais dont le champ contenu est le couple (SOURCE(Q_{ik}), DISTANCE(Q_{ik})) (on rajoute le champ distance du paquet Q_{ik}).

De même, le poste k renvoie un paquet R_{ki} chaque fois qu'il reçoit un paquet Q_{ik} avec une distance plus courte vers le poste i.

Question 17 On suppose que toutes les tables de routages *Route*(i) sont à **nil** (sauf sur le poste i) au moment où le poste i commence la diffusion de son message Q_{ik}. Montrer que les tables de routages vers le poste i restent cohérentes. ◇

Solution. On procède par récurrence sur l'ordre de réception du paquet Q_{ik}. A l'ordre zéro le poste i a sa table de routage cohérente, puisqu'elle pointe sur le poste i. Supposons que les tables de routage sont cohérentes lors des m premières réceptions du paquet Q_{ik}. Considérons que le routeur ℓ est le récepteur de la $m+1$-ème réception du paquet Q_{ik}. Supposons qu'il le reçoit du poste j. Comme le poste j avait sa table de routage cohérente lors de cette transmission du paquet Q_{ik} vers le poste j, alors la preuve termine comme à la question 14, après avoir néanmoins avoir observé que la route à partir du poste j ne peut pas boucler sur ℓ parce que cela impliquerait que une ancienne valeur de la distance sur ℓ serait plus petite que ce qu'elle est après la m-ème réception de Q_{ik}. □

Question 18 Montrer que le processus ne retransmet qu'un nombre fini de paquets Q_{ik} dans le réseau.◇

Solution. Par l'absurde : si un poste ℓ retransmet un nombre infini de versions du paquet Q_{ik} avec des distances à chaque fois strictement plus petite, comme les tables de routages sont cohérentes, les distances correspondent au cumul de poids de chemins simples existant entre le poste ℓ et le poste i. Or il existe un nombre fini de tels chemins, donc un nombre fini de distances strictement décroissantes. □

Question 19 Montrer qu'à la fin de toutes les retransmissions des paquets Q_{ik} et R_{ki} les tables de routages présentent des routes de poids minimal (c'est à dire que les tables de routages sont optimales).◇

Solution. Par récurrence sur le nombre de relais sur le chemin optimal vers i. Si le chemin optimal est de longueur 1 alors il est formé après la réception du message Q_{ik} à partir de i. Supposons maintenant que tous les postes dont il existe un chemin optimal de moins de m relais finissent par avoir une table de routage optimale. Montrons que cette propriété est vraie quand le nombre de relais est porté à $m+1$. Soit un poste ℓ dans ce cas. Soit j le poste qui est le prochain relai de ℓ vers i ; donc j a un chemin optimal avec m relais, donc j finit par avoir une table de routage optimale. Quand la distance dans la table de routage descend à la valeure optimale, le poste j envoie un paquet Q_{ik} vers ℓ. Quand le poste ℓ reçoit ce paquet Q_{ik} il a nécessairement une table de routage optimale. □

Question 20 Donner une borne supérieure du nombre total de transmissions du paquet Q_{ik} dans le cas où chaque arête a un poids égal à l'unité. On ne compte que les transmissions avec des distances différentes et on exclut les possibles retransmissions sur timeout. ◇

Solution. Il n'y a que $n-1$ valeurs possibles pour les distances, donc chaque poste n'émettra au plus que n paquets Q_{ik} différents. Donc la borne supérieure est $(n-1)^2$. □

Question 21 Même question mais avec des poids quelconques. ◇

Solution. Dans le pire cas il y a un cumul de poids différent pour chaque chemin. Dans le graphe complet il y a $\sum_{\ell \leq n-1} \frac{(n-1)!}{n-1-\ell} \leq e(n-1)!$ chemins différents, donc autant de paquets Q_{ik} différents par poste. La borne supérieure est donc un cumul de $en!$ paquets dans le réseau. □

La requête de route peut aussi s'accompagner d'une contrainte de capacité par lien. Par exemple si la route est demandée pour un flux vidéo il faut que la route soit constituée de liens dont la capacité est au dessus d'un certain seuil, par exemple 100 kbit par seconde. On suppose que chaque lien $\{i, j\}$ a une capacité spécifique b_{ij} connue des postes i et j. On suppose que le paquet de requête contient aussi un champ capacité qui indique la capacité minimale requise sur chacun des liens de la route.

Question 22 Donner brièvement quelques indications pour modifier le protocole de construction de route pour prendre en compte la contrainte de capacité minimale. On fera l'hypothèse qu'une telle route existe. ◇

Solution. Il suffit que chaque poste j ne fasse rien quand il reçoit le paquet Q_{ik} d'un poste ℓ tel que $b_{j\ell} < \text{CAPACITÉ}(Q_{ik})$. □