Ordonnancement du régime permanent (*Steady-State Scheduling*)

Frédéric Vivien

e-mail: Frederic.Vivien@ens-lyon.fr

14 octobre 2005

Plan du cours

Plan du cours

1. Le contexte
2. Routage de paquets à chemins fixés
3. Résolution du problème *fluidifié*
4. Construction d'un ordonnancement
5. Routage de paquets avec liberté sur les chemins

Plate-forme

Plate-forme : hétérogène et distribuée :

- des processeurs de caractéristiques différentes ;
- des liens réseaux de caractéristiques différentes.

Applications

Application composée d'un très (très) grand nombre de tâches, les tâches étant regroupées en un nombre fini de types, toutes les tâches d'un même type ayant les mêmes caractéristiques.

Principe

Quand on a un grand nombre de travaux identiques à exécuter, on imagine qu'après une phase d'initialisation, on atteindra un (long) régime permanent, avant une phase de terminaison.

Si le régime permanent est suffisamment long, les phases d'initialisation et de terminaison seront négligeables.

Plan du cours

Le problème

Problème : acheminement d'un ensemble de flots de messages.

Dans un réseau de communication, plusieurs flots de paquets doivent être acheminés, chaque flot de paquet doit être envoyé d'une source à une destination, tout en suivant un chemin donné, reliant la source à la destination.

Notations

- (V, A) un graphe orienté, représentant un réseau de communications.

Notations

- (V, A) un graphe orienté, représentant un réseau de communications.
- On a un total de n_c flots à acheminer.

Notations

- (V, A) un graphe orienté, représentant un réseau de communications.
- On a un total de n_c flots à acheminer.
- Le k^{e} flot est notée (s_k, t_k, P_k, n_k), où

Notations

- (V, A) un graphe orienté, représentant un réseau de communications.
- On a un total de n_c flots à acheminer.
- Le k^{e} flot est notée (s_k, t_k, P_k, n_k), où
  - s_k est la source des paquets ;

Notations

- (V, A) un graphe orienté, représentant un réseau de communications.
- On a un total de n_c flots à acheminer.
- Le k^{e} flot est notée (s_k, t_k, P_k, n_k), où
  - s_k est la source des paquets ;
  - t_k est la destination ;

Notations

- (V, A) un graphe orienté, représentant un réseau de communications.
- On a un total de n_c flots à acheminer.
- Le k^{e} flot est notée (s_k, t_k, P_k, n_k), où
  - s_k est la source des paquets ;
  - t_k est la destination ;
  - P_k est le chemin à suivre ;

Notations

- (V, A) un graphe orienté, représentant un réseau de communications.
- On a un total de n_c flots à acheminer.
- Le k^{e} flot est notée (s_k, t_k, P_k, n_k), où
  - s_k est la source des paquets ;
  - t_k est la destination ;
  - P_k est le chemin à suivre ;
  - n_k est le nombre de paquets à transférer.

Notations

- (V, A) un graphe orienté, représentant un réseau de communications.
- On a un total de n_c flots à acheminer.
- Le k^{e} flot est notée (s_k, t_k, P_k, n_k), où
  - s_k est la source des paquets ;
  - t_k est la destination ;
  - P_k est le chemin à suivre ;
  - n_k est le nombre de paquets à transférer.

  On note $a_{k,i}$ la i^{e} arête du chemin P_k.

Hypothèses

- Un paquet traverse une arête de A en un temps unitaire.

Hypothèses

- Un paquet traverse une arête de A en un temps unitaire.
- À un instant donné, un seul paquet traverse une arête donnée.

Objectif

Décider quel paquet doit passer quand par une arête donnée, de manière à minimiser le temps d'exécution global.

Borne inférieure sur la durée des ordonnancements

On appelle **congestion** de l'arête $a \in A$, et on note C_a, le nombre total de paquets qui traversent l'arête a :

$$C_a = \sum_{k \mid a \in P_k} n_k \qquad C_{\max} = \max_a C_a$$

Borne inférieure sur la durée des ordonnancements

On appelle **congestion** de l'arête $a \in A$, et on note C_a, le nombre total de paquets qui traversent l'arête a :

$$C_a = \sum_{k \mid a \in P_k} n_k \qquad C_{\max} = \max_a C_a$$

$C_{\max}$ est une borne inférieure sur le temps d'exécution de l'ordonnancement.

$$C^* \geq C_{\max}$$

Borne inférieure sur la durée des ordonnancements

On appelle **congestion** de l'arête $a \in A$, et on note C_a, le nombre total de paquets qui traversent l'arête a :

$$C_a = \sum_{k \mid a \in P_k} n_k \qquad C_{\max} = \max_a C_a$$

$C_{\max}$ est une borne inférieure sur le temps d'exécution de l'ordonnancement.

$$C^* \geq C_{\max}$$

Une résolution « fluide » (fractionnelle) du problème va nous donner une solution qui s'exécute en temps $C_{\max}$.

Plan du cours

Version *fluidifiée* (fractionnelle) : notations

Principe :

- on ne recherche pas une solution entière mais rationnelle.

Version *fluidifiée* (fractionnelle) : notations

Principe :

- on ne recherche pas une solution entière mais rationnelle.

- $n_{k,i}(t)$ nombre (fractionnel) de paquets en attente à l'entrée de la i^{e} arête du k^{e} chemin, au temps t.

Version *fluidifiée* (fractionnelle) : notations

Principe :

- on ne recherche pas une solution entière mais rationnelle.

- $n_{k,i}(t)$ nombre (fractionnel) de paquets en attente à l'entrée de la i^{e} arête du k^{e} chemin, au temps t.

- $T_{k,i}(t)$ est le temps total dévolu par l'arête $a_{k,i}$ aux paquets du k^{e} flot, pendant l'intervalle de temps $[0;t]$.

Version *fluidifiée* (fractionnelle) : mise en équations

1. Initiation des envois

$$n_{k,1}(t) = n_k - T_{k,1}(t), \qquad \text{pour tout } k$$

Version *fluidifiée* (fractionnelle) : mise en équations

1. Initiation des envois

$$n_{k,1}(t) = n_k - T_{k,1}(t), \qquad \text{pour tout } k$$

2. Loi de conservation

$$n_{k,i+1}(t) = T_{k,i}(t) - T_{k,i+1}(t), \qquad \text{pour tout } k$$

Version *fluidifiée* (fractionnelle) : mise en équations

1. Initiation des envois

$$n_{k,1}(t) = n_k - T_{k,1}(t), \qquad \text{pour tout } k$$

2. Loi de conservation

$$n_{k,i+1}(t) = T_{k,i}(t) - T_{k,i+1}(t), \qquad \text{pour tout } k$$

3. Contraintes de ressource

$$\sum_{(k,i) \ | \ a_{k,i}=a} T_{k,i}(t_2) - T_{k,i}(t_1) \leq t_2 - t_1, \forall a \in A, \forall t_2 \geq t_1 \geq 0$$

Version *fluidifiée* (fractionnelle) : mise en équations

1. Initiation des envois

$$n_{k,1}(t) = n_k - T_{k,1}(t), \qquad \text{pour tout } k$$

2. Loi de conservation

$$n_{k,i+1}(t) = T_{k,i}(t) - T_{k,i+1}(t), \qquad \text{pour tout } k$$

3. Contraintes de ressource

$$\sum_{(k,i)\ \mid\ a_{k,i}=a} T_{k,i}(t_2) - T_{k,i}(t_1) \leq t_2 - t_1, \forall a \in A, \forall t_2 \geq t_1 \geq 0$$

4. Objectif

$$\textsc{Minimiser } C_{\text{frac}} = \int_0^{\infty} \mathbb{1}\left(\sum_{k,i} n_{k,i}(t)\right) dt$$

Borne inférieure

- $n_{k,1}(t) = n_k - T_{k,1}(t), \qquad$ pour tout k
- $n_{k,i+1}(t) = T_{k,i}(t) - T_{k,i+1}(t), \qquad$ pour tout k

Borne inférieure

- $n_{k,1}(t) = n_k - T_{k,1}(t), \qquad$ pour tout k
- $n_{k,i+1}(t) = T_{k,i}(t) - T_{k,i+1}(t), \qquad$ pour tout k
- À tout instant t, $\displaystyle\sum_{j=1}^{i} n_{k,j}(t) = n_k - T_{k,i}(t)$

Borne inférieure

- $n_{k,1}(t) = n_k - T_{k,1}(t), \qquad$ pour tout k
- $n_{k,i+1}(t) = T_{k,i}(t) - T_{k,i+1}(t), \qquad$ pour tout k
- À tout instant t, $\displaystyle\sum_{j=1}^{i} n_{k,j}(t) = n_k - T_{k,i}(t)$
- Pour toute arête a :

$$\sum_{(k,i)|a_{k,i}=a} \sum_{j=1}^{i} n_{k,j}(t) = \sum_{(k,i)|a_{k,i}=a} n_k - \sum_{(k,i)|a_{k,i}=a} T_{k,i}(t)$$

Borne inférieure

- $n_{k,1}(t) = n_k - T_{k,1}(t), \qquad$ pour tout k
- $n_{k,i+1}(t) = T_{k,i}(t) - T_{k,i+1}(t), \qquad$ pour tout k
- À tout instant t, $\displaystyle\sum_{j=1}^{i} n_{k,j}(t) = n_k - T_{k,i}(t)$
- Pour toute arête a :

$$\sum_{(k,i)|a_{k,i}=a} \sum_{j=1}^{i} n_{k,j}(t) = \sum_{(k,i)|a_{k,i}=a} n_k - \sum_{(k,i)|a_{k,i}=a} T_{k,i}(t) \geq C_a - t$$

Borne inférieure

- $n_{k,1}(t) = n_k - T_{k,1}(t)$, pour tout k
- $n_{k,i+1}(t) = T_{k,i}(t) - T_{k,i+1}(t)$, pour tout k
- À tout instant t, $\displaystyle\sum_{j=1}^{i} n_{k,j}(t) = n_k - T_{k,i}(t)$
- Pour toute arête a :

$$\sum_{(k,i)|a_{k,i}=a} \sum_{j=1}^{i} n_{k,j}(t) = \sum_{(k,i)|a_{k,i}=a} n_k - \sum_{(k,i)|a_{k,i}=a} T_{k,i}(t) \geq C_a - t$$

Tant que $t > C_a$, il y a des paquets dans le système.

Borne inférieure

- $n_{k,1}(t) = n_k - T_{k,1}(t), \qquad$ pour tout k
- $n_{k,i+1}(t) = T_{k,i}(t) - T_{k,i+1}(t), \qquad$ pour tout k
- À tout instant t, $\displaystyle\sum_{j=1}^{i} n_{k,j}(t) = n_k - T_{k,i}(t)$
- Pour toute arête a :

$$\sum_{(k,i)|a_{k,i}=a} \sum_{j=1}^{i} n_{k,j}(t) = \sum_{(k,i)|a_{k,i}=a} n_k - \sum_{(k,i)|a_{k,i}=a} T_{k,i}(t) \geq C_a - t$$

Tant que $t > C_a$, il y a des paquets dans le système.

Par conséquent $C_{\text{frac}} \geq \max_a C_a = C_{\max}$

Une solution candidate

Pour $t \leq C_{\max}$

Une solution candidate

Pour $t \leq C_{\max}$

- $T_{k,i}(t) = \dfrac{n_k}{C_{\max}} t$, pour tout k et i.

Une solution candidate

Pour $t \leq C_{\max}$

- $T_{k,i}(t) = \dfrac{n_k}{C_{\max}} t$, pour tout k et i.
- $n_{k,1}(t) = n_k - T_{k,1}(t) = n_k - \dfrac{n_k}{C_{\max}} t = n_k \left(1 - \dfrac{t}{C_{\max}}\right), \quad \forall k$

Une solution candidate

Pour $t \leq C_{\max}$

- $T_{k,i}(t) = \dfrac{n_k}{C_{\max}} t$, pour tout k et i.
- $n_{k,1}(t) = n_k - T_{k,1}(t) = n_k - \dfrac{n_k}{C_{\max}} t = n_k \left(1 - \dfrac{t}{C_{\max}}\right), \quad \forall k$
- $n_{k,i}(t) = 0$, pour tout k et $i \geq 2$.

Une solution candidate

Pour $t \leq C_{\max}$

- $T_{k,i}(t) = \frac{n_k}{C_{\max}} t$, pour tout k et i.
- $n_{k,1}(t) = n_k - T_{k,1}(t) = n_k - \frac{n_k}{C_{\max}} t = n_k \left(1 - \frac{t}{C_{\max}}\right), \quad \forall k$
- $n_{k,i}(t) = 0$, pour tout k et $i \geq 2$.

Pour $t \geq C_{\max}$

Une solution candidate

Pour $t \leq C_{\max}$

- $T_{k,i}(t) = \dfrac{n_k}{C_{\max}} t$, pour tout k et i.
- $n_{k,1}(t) = n_k - T_{k,1}(t) = n_k - \dfrac{n_k}{C_{\max}} t = n_k \left(1 - \dfrac{t}{C_{\max}}\right), \quad \forall k$
- $n_{k,i}(t) = 0$, pour tout k et $i \geq 2$.

Pour $t \geq C_{\max}$

- $T_{k,i}(t) = n_k$

Une solution candidate

Pour $t \leq C_{\max}$

- $T_{k,i}(t) = \dfrac{n_k}{C_{\max}} t$, pour tout k et i.
- $n_{k,1}(t) = n_k - T_{k,1}(t) = n_k - \dfrac{n_k}{C_{\max}} t = n_k \left(1 - \dfrac{t}{C_{\max}}\right), \quad \forall k$
- $n_{k,i}(t) = 0$, pour tout k et $i \geq 2$.

Pour $t \geq C_{\max}$

- $T_{k,i}(t) = n_k$
- $n_{k,i}(t) = 0$

Une solution candidate

Pour $t \leq C_{\max}$

- $T_{k,i}(t) = \dfrac{n_k}{C_{\max}} t$, pour tout k et i.
- $n_{k,1}(t) = n_k - T_{k,1}(t) = n_k - \dfrac{n_k}{C_{\max}} t = n_k \left(1 - \dfrac{t}{C_{\max}}\right), \quad \forall k$
- $n_{k,i}(t) = 0$, pour tout k et $i \geq 2$.

Pour $t \geq C_{\max}$

- $T_{k,i}(t) = n_k$
- $n_{k,i}(t) = 0$

Cette solution est un ordonnancement de durée $C_{\max}$. Il reste à montrer qu'elle est faisable.

Vérification de la solution (pour $t \leq C_{\max}$)

1. $n_{k,1}(t) = n_k - T_{k,1}(t), \qquad$ pour tout k
   Vérifiée par définition.

Vérification de la solution (pour $t \leq C_{\max}$)

1. $n_{k,1}(t) = n_k - T_{k,1}(t),$ pour tout k
   Vérifiée par définition.

2. $n_{k,i+1}(t) = T_{k,i}(t) - T_{k,i+1}(t),$ pour tout k
   $T_{k,i}(t) - T_{k,i+1}(t) = \frac{n_k}{C_{\max}}t - \frac{n_k}{C_{\max}}t = 0 = n_{k,i+1}(t)$

Vérification de la solution (pour $t \leq C_{\max}$)

1. $n_{k,1}(t) = n_k - T_{k,1}(t), \qquad$ pour tout k
   Vérifiée par définition.

2. $n_{k,i+1}(t) = T_{k,i}(t) - T_{k,i+1}(t), \qquad$ pour tout k
   $T_{k,i}(t) - T_{k,i+1}(t) = \frac{n_k}{C_{\max}}t - \frac{n_k}{C_{\max}}t = 0 = n_{k,i+1}(t)$

3. $$\sum_{(k,i)\ \mid\ a_{k,i}=a} T_{k,i}(t_2) - T_{k,i}(t_1) \leq t_2 - t_1, \forall a \in A, \forall t_2 \geq t_1 \geq 0$$
   $$\sum_{(k,i)\ \mid\ a_{k,i}=a} T_{k,i}(t_2) - T_{k,i}(t_1) = \sum_{(k,i)\ \mid\ a_{k,i}=a} \frac{n_k}{C_{\max}}(t_2 - t_1) =$$
   $$\frac{C_a}{C_{\max}}(t_2 - t_1) \leq t_2 - t_1$$

Plan du cours

Définition d'une tournée

- $\Omega \approx$ durée d'une tournée (déterminé ultérieurement).

Définition d'une tournée

- $\Omega \approx$ durée d'une tournée (déterminé ultérieurement).
- m_k : nombre de paquets du k^{e} flot distribués dans une tournée.

$$m_k = \left\lceil \frac{n_k \Omega}{C_{\max}} \right\rceil .$$

Définition d'une tournée

- $\Omega \approx$ durée d'une tournée (déterminé ultérieurement).
- m_k : nombre de paquets du k^{e} flot distribués dans une tournée.

$$m_k = \left\lceil \frac{n_k \Omega}{C_{\max}} \right\rceil .$$

- $D_a = \sum_{(k,i)|a_{k,i}=a} 1 = |\{k|a \in P_k\}|$

$$D_{\max} = \max_a D_a \leq n_c$$

Définition d'une tournée

- $\Omega \approx$ durée d'une tournée (déterminé ultérieurement).
- m_k : nombre de paquets du k^{e} flot distribués dans une tournée.

$$m_k = \left\lceil \frac{n_k \Omega}{C_{\max}} \right\rceil .$$

- $D_a = \sum_{(k,i)|a_{k,i}=a} 1 = |\{k|a \in P_k\}|$

$$D_{\max} = \max_a D_a \leq n_c$$

- Période de l'ordonnancement : $\Omega + D_{\max}$.

Ordonnancement

Pendant l'intervalle de temps $[j(\Omega+D_{\max}); (j+1)(\Omega+D_{\max})]$:

Ordonnancement

Pendant l'intervalle de temps $[j(\Omega+D_{\max}); (j+1)(\Omega+D_{\max})]$:

Le lien a transmet m_k paquets du k^{e} flot s'il existe i tel que $a_{k,i} = a$.

Ordonnancement

Pendant l'intervalle de temps $[j(\Omega{+}D_{\max}); (j{+}1)(\Omega{+}D_{\max})]$:

Le lien a transmet m_k paquets du k^{e} flot s'il existe i tel que $a_{k,i} = a$.

Le lien a reste inutilisé pendant une durée :

$$\Omega + D_{\max} - \sum_{(k,i) | a_{k,i} = a} m_k$$

Ordonnancement

Pendant l'intervalle de temps $[j(\Omega{+}D_{\max}); (j{+}1)(\Omega{+}D_{\max})]$:

Le lien a transmet m_k paquets du k^{e} flot s'il existe i tel que $a_{k,i} = a$.

Le lien a reste inutilisé pendant une durée :

$$\Omega + D_{\max} - \sum_{(k,i) | a_{k,i}=a} m_k$$

(Si moins de m_k paquets sont en attente en entrée de a au temps $j(\Omega + D_{\max})$, a transmet ce qui est disponible et reste inactif plus longtemps.)

Faisabilité de l'ordonnancement

$$\sum_{(k,i) \mid a_{k,i} = a} m_k$$

Faisabilité de l'ordonnancement

$$\sum_{(k,i)|a_{k,i}=a} m_k = \sum_{(k,i)|a_{k,i}=a} \left\lceil \frac{n_k \Omega}{C_{\max}} \right\rceil$$

Faisabilité de l'ordonnancement

$$
\begin{aligned}
\sum_{(k,i)|a_{k,i}=a} m_k &= \sum_{(k,i)|a_{k,i}=a} \left\lceil \frac{n_k \Omega}{C_{\max}} \right\rceil \\
&\leq \sum_{(k,i)|a_{k,i}=a} \left(\frac{n_k \Omega}{C_{\max}} + 1 \right)
\end{aligned}
$$

Faisabilité de l'ordonnancement

$$
\begin{aligned}
\sum_{(k,i)|a_{k,i}=a} m_k &= \sum_{(k,i)|a_{k,i}=a} \left\lceil \frac{n_k \Omega}{C_{\max}} \right\rceil \\
&\leq \sum_{(k,i)|a_{k,i}=a} \left(\frac{n_k \Omega}{C_{\max}} + 1 \right) \\
&\leq \frac{C_a}{C_{\max}} \Omega + D_a
\end{aligned}
$$

Faisabilité de l'ordonnancement

$$\begin{aligned}
\sum_{(k,i)|a_{k,i}=a} m_k &= \sum_{(k,i)|a_{k,i}=a} \left\lceil \frac{n_k \Omega}{C_{\max}} \right\rceil \\
&\leq \sum_{(k,i)|a_{k,i}=a} \left(\frac{n_k \Omega}{C_{\max}} + 1 \right) \\
&\leq \frac{C_a}{C_{\max}} \Omega + D_a \\
&\leq \Omega + D_{\max}
\end{aligned}$$

Comportement des sources

- $N_{k,i}(t)$: nombre de paquets du k^{e} flot en attente en entrée de la i^{e} arête, à l'instant t.

Comportement des sources

- $N_{k,i}(t)$: nombre de paquets du k^{e} flot en attente en entrée de la i^{e} arête, à l'instant t.
- $a_{k,1}$ envoie m_k paquets pendant $[0, \Omega + D_{\max}]$.
  $N_{k,1}(\Omega + D_{\max}) = n_k - m_k$

Comportement des sources

- $N_{k,i}(t)$: nombre de paquets du k^{e} flot en attente en entrée de la i^{e} arête, à l'instant t.
- $a_{k,1}$ envoie m_k paquets pendant $[0, \Omega + D_{\max}]$.
  $N_{k,1}(\Omega + D_{\max}) = n_k - m_k$
- $a_{k,1}$ envoie m_k paquets pendant $[\Omega + D_{\max}, 2(\Omega + D_{\max})]$.
  $N_{k,1}(2(\Omega + D_{\max})) = n_k - 2m_k$

Comportement des sources

- $N_{k,i}(t)$: nombre de paquets du k^{e} flot en attente en entrée de la i^{e} arête, à l'instant t.
- $a_{k,1}$ envoie m_k paquets pendant $[0, \Omega + D_{\max}]$.
  $N_{k,1}(\Omega + D_{\max}) = n_k - m_k$
- $a_{k,1}$ envoie m_k paquets pendant $[\Omega + D_{\max}, 2(\Omega + D_{\max})]$.
  $N_{k,1}(2(\Omega + D_{\max})) = n_k - 2m_k$
- On pose $T = \left\lceil \frac{C_{\max}}{\Omega} \right\rceil (\Omega + D_{\max})$

$$N_{k,1}(T) \leq n_k - \frac{T}{\Omega + D_{\max}} m_k \leq n_k - \frac{n_k \Omega}{C_{\max}} \frac{C_{\max}}{\Omega} = 0$$

Délai de propagation

- $a_{k,1}$ envoie m_k paquets pendant $[0, \Omega + D_{\max}]$.
  $N_{k,1}(\Omega + D_{\max}) = n_k - m_k$ $\quad$ $N_{k,2}(\Omega + D_{\max}) = m_k$
  $N_{k,i\geq 3}(\Omega + D_{\max}) = 0$

- $a_{k,1}$ envoie m_k paquets pendant $[\Omega + D_{\max}, 2(\Omega + D_{\max})]$.
  $N_{k,1}(2(\Omega + D_{\max})) = n_k - 2m_k$ $\quad$ $N_{k,2}(2(\Omega + D_{\max})) = m_k$
  $N_{k,3}(2(\Omega + D_{\max})) = m_k$ $\quad$ $N_{k,i\geq 4}(2(\Omega + D_{\max})) = 0$

- Le délai entre le moment où un paquet traverse la première arête du chemin P_k et le moment où il traverse la dernière arête est au pire :
  $$(|P_k| - 1)(\Omega + D_{\max})$$
  On pose $L = \max_k |P_k|$.

Durée de l'ordonnancement

$$C_{\mathsf{total}} \leq T + (L-1)(\Omega + D_{\max})$$

Durée de l'ordonnancement

$$
\begin{aligned}
C_{\mathsf{total}} &\leq T + (L-1)(\Omega + D_{\max}) \\
&= \left\lceil \frac{C_{\max}}{\Omega} \right\rceil (\Omega + D_{\max}) + (L-1)(\Omega + D_{\max})
\end{aligned}
$$

Durée de l'ordonnancement

$$
\begin{aligned}
C_{\mathsf{total}} &\le T + (L-1)(\Omega + D_{\max}) \\
&= \left\lceil \frac{C_{\max}}{\Omega} \right\rceil (\Omega + D_{\max}) + (L-1)(\Omega + D_{\max}) \\
&\le \left(\frac{C_{\max}}{\Omega} + 1 \right) (\Omega + D_{\max}) + (L-1)(\Omega + D_{\max})
\end{aligned}
$$

Durée de l'ordonnancement

$$
\begin{aligned}
C_{\text{total}} &\leq T + (L-1)(\Omega + D_{\max}) \\
&= \left\lceil \frac{C_{\max}}{\Omega} \right\rceil (\Omega + D_{\max}) + (L-1)(\Omega + D_{\max}) \\
&\leq \left(\frac{C_{\max}}{\Omega} + 1 \right) (\Omega + D_{\max}) + (L-1)(\Omega + D_{\max}) \\
&= C_{\max} + LD_{\max} + \frac{D_{\max} C_{\max}}{\Omega} + L\Omega
\end{aligned}
$$

Durée de l'ordonnancement

$$
\begin{aligned}
C_{\mathsf{total}} &\leq T + (L-1)(\Omega + D_{\max}) \\
&= \left\lceil \frac{C_{\max}}{\Omega} \right\rceil (\Omega + D_{\max}) + (L-1)(\Omega + D_{\max}) \\
&\leq \left(\frac{C_{\max}}{\Omega} + 1 \right) (\Omega + D_{\max}) + (L-1)(\Omega + D_{\max}) \\
&= C_{\max} + L D_{\max} + \frac{D_{\max} C_{\max}}{\Omega} + L\Omega
\end{aligned}
$$

La borne supérieure est minimisée pour $\Omega = \sqrt{\dfrac{D_{\max} C_{\max}}{L}}$

Durée de l'ordonnancement

$$
\begin{aligned}
C_{\textsf{total}} &\leq T + (L-1)(\Omega + D_{\max}) \\
&= \left\lceil \frac{C_{\max}}{\Omega} \right\rceil (\Omega + D_{\max}) + (L-1)(\Omega + D_{\max}) \\
&\leq \left(\frac{C_{\max}}{\Omega} + 1 \right) (\Omega + D_{\max}) + (L-1)(\Omega + D_{\max}) \\
&= C_{\max} + L D_{\max} + \frac{D_{\max} C_{\max}}{\Omega} + L\Omega
\end{aligned}
$$

La borne supérieure est minimisée pour $\Omega = \sqrt{\dfrac{D_{\max} C_{\max}}{L}}$

$$
C_{\textsf{total}} \leq C_{\max} + 2\sqrt{C_{\max} D_{\max} L} + D_{\max} L
$$

Optimalité asymptotique

$$C_{\max} \leq C^* \leq C_{\mathsf{total}} \leq C_{\max} + 2\sqrt{C_{\max} D_{\max} L} + D_{\max} L$$

$$1 \leq \frac{C_{\mathsf{total}}}{C_{\max}} \leq 1 + 2\sqrt{\frac{D_{\max} L}{C_{\max}}} + \frac{D_{\max} L}{C_{\max}}$$

Avec $\Omega = \sqrt{\frac{D_{\max} C_{\max}}{L}}$

Besoins en ressources

$$\sum_{(k,i)|a_{k,i}=a,k\geq 2} m_k \leq \sum_{(k,i)|a_{k,i}=a,k\geq 2} \left(\frac{n_k}{C_{\max}} \sqrt{\frac{D_{\max} C_{\max}}{L}} + 1 \right)$$

$$\leq \sqrt{\frac{D_{\max} C_{\max}}{L}} + D_{\max}$$

Conclusion

- Oubli des phases d'initialisation et de terminaison
- Résolution en rationnel du régime permanent
- Tournée de taille la racine carrée de la solution :
  - Chaque tournée « perd » un temps constant
  - La somme des temps perdus croit moins vite que l'ordonnancement
  - Tampons de taille racine carrée de la solution

Plan du cours

Problème

- Même problème que précédemment, mais les chemins ne sont pas fixés.

Problème

- Même problème que précédemment, mais les chemins ne sont pas fixés.
- On a un total de n_c collections de paquets à acheminer

Problème

- Même problème que précédemment, mais les chemins ne sont pas fixés.
- On a un total de n_c collections de paquets à acheminer
- Chaque collection est acheminée via un *ensemble* de flots (les différents paquets d'une même collection peuvent suivre des chemins différents).

Problème

- Même problème que précédemment, mais les chemins ne sont pas fixés.
- On a un total de n_c collections de paquets à acheminer
- Chaque collection est acheminée via un *ensemble* de flots (les différents paquets d'une même collection peuvent suivre des chemins différents).
- $n^{k,l}$ nombre total de paquets à acheminer de k à l.

Problème

- Même problème que précédemment, mais les chemins ne sont pas fixés.
- On a un total de n_c collections de paquets à acheminer
- Chaque collection est acheminée via un *ensemble* de flots (les différents paquets d'une même collection peuvent suivre des chemins différents).
- $n^{k,l}$ nombre total de paquets à acheminer de k à l.
- $n_{i,j}^{k,l}$: nombre total de paquets acheminés de k à l qui passent par l'arête (i, j).

Problème

- Même problème que précédemment, mais les chemins ne sont pas fixés.
- On a un total de n_c collections de paquets à acheminer
- Chaque collection est acheminée via un *ensemble* de flots (les différents paquets d'une même collection peuvent suivre des chemins différents).
- $n^{k,l}$ nombre total de paquets à acheminer de k à l.
- $n_{i,j}^{k,l}$: nombre total de paquets acheminés de k à l qui passent par l'arête (i,j).
  Congestion : $C_{i,j} = \displaystyle\sum_{(k,l)|n^{k,l}>0} n_{i,j}^{k,l} \qquad C_{\max} = \max_{i,j} C_{i,j}$.

Mise en équations (1)

1. Initiation des envois

$$\sum_{j|(k,j)\in A} n_{k,j}^{k,l} = n^{k,l}$$

Mise en équations (1)

1. Initiation des envois

$$\sum_{j|(k,j)\in A} n_{k,j}^{k,l} = n^{k,l}$$

2. Réception des envois

$$\sum_{i|(i,l)\in A} n_{i,l}^{k,l} = n^{k,l}$$

Mise en équations (1)

1. Initiation des envois

$$\sum_{j|(k,j)\in A} n_{k,j}^{k,l} = n^{k,l}$$

2. Réception des envois

$$\sum_{i|(i,l)\in A} n_{i,l}^{k,l} = n^{k,l}$$

3. Loi de conservation

$$\sum_{i|(i,j)\in A} n_{i,j}^{k,l} = \sum_{i|(j,i)\in A} n_{j,i}^{k,l} \quad \forall(k,l), j \neq k, j \neq l$$

Mise en équations (2)

4. Congestion

$$C_{i,j} = \sum_{(k,l) | n^{k,l} > 0} n_{i,j}^{k,l}$$

Mise en équations (2)

4. Congestion

$$C_{i,j} = \sum_{(k,l)|n^{k,l}>0} n_{i,j}^{k,l}$$

5. Définition de l'objectif

$$C_{\max} \geq C_{i,j}, \qquad \forall i,j$$

Mise en équations (2)

4. Congestion

$$C_{i,j} = \sum_{(k,l) | n^{k,l} > 0} n_{i,j}^{k,l}$$

5. Définition de l'objectif

$$C_{\max} \geq C_{i,j}, \qquad \forall i, j$$

6. Fonction objectif

Minimiser $C_{\max}$

Mise en équations (2)

4. Congestion

$$C_{i,j} = \sum_{(k,l)|n^{k,l}>0} n_{i,j}^{k,l}$$

5. Définition de l'objectif

$$C_{\max} \geq C_{i,j}, \qquad \forall i, j$$

6. Fonction objectif

$$\text{Minimiser } C_{\max}$$

Programme linéaire en nombre rationnel : résolu en temps polynomial par n'importe quel programme de résolution.

Algorithme de routage

1. Calculer la valeur optimale $C_{\max}$ du programme linéaire précédent.

Algorithme de routage

1. Calculer la valeur optimale $C_{\max}$ du programme linéaire précédent.
2. Soit Ω à définir ultérieurement.
   Pendant l'intervalle $[p\Omega, (p+1)\Omega]$, l'arête (i,j) transmet :

$$m_{i,j}^{k,l} = \left\lfloor \frac{n_{i,j}^{k,l}\Omega}{C_{\max}} \right\rfloor$$

   paquets qui vont de k vers l.

Algorithme de routage

1. Calculer la valeur optimale $C_{\max}$ du programme linéaire précédent.
2. Soit Ω à définir ultérieurement.
   Pendant l'intervalle $[p\Omega, (p+1)\Omega]$, l'arête (i, j) transmet :
   $$m_{i,j}^{k,l} = \left\lfloor \frac{n_{i,j}^{k,l}\Omega}{C_{\max}} \right\rfloor$$
   paquets qui vont de k vers l.
3. À partir du temps :
   $$T \equiv \left\lceil \frac{C_{\max}}{\Omega} \right\rceil \Omega \leq C_{\max} + \Omega$$
   on traite les M paquets résiduels séquentiellement, ce qui prend un temps ML (au pire) où L est la longueur maximale d'un chemin simple dans le réseau.

L'ordonnancement est faisable

$$\sum_{(k,l)} m_{i,j}^{k,l} \leq \sum_{(k,l)} \frac{n_{i,j}^{k,l}\Omega}{C_{\max}} = \frac{C_{i,j}\Omega}{C_{\max}} \leq \Omega$$

Durée de l'ordonnancement

- On défini Ω par : $\Omega = \sqrt{C_{\max} n_c}$.
- Le nombre total de paquets restants dans le réseau au temps T est au pire :
$$2|A|\sqrt{C_{\max} n_c} + |A| n_c$$
- La durée de l'ordonnancement est alors
$$C_{\max} \leq C^* \leq C_{\max} + \sqrt{C_{\max} n_c} + 2|A|\sqrt{C_{\max} n_c}|V| + |A| n_c |V|$$