

Structures Combinatoires et Simulation des Réseaux Radio Maillés

Christelle Molle

Réunion Mascotte - Le Boréon

12-13 Mars 2009

1 Contexte

- Les réseaux radio maillés
- Motivations

2 Résultats

- Modèles optimaux
- Relaxations
- Acknowledgment : une étude quantitative

3 Simulation

- La génération de colonnes
- Génération croisée de lignes et colonnes

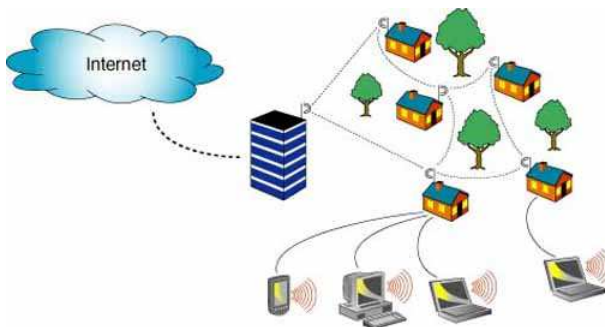
4 Travail en cours

- Clustering
- Glouton, feignant, mais optimal

Introduction

Les réseaux radio maillés (WMNs)

- Solution avantageuse, efficace et peu coûteuse
- Couverture large
- Facilement extensibles



Motivations

Optimiser les performances

- Maximiser la capacité
- Limiter les interférences
- Ordonnancer les communications : routage à un débit maximum

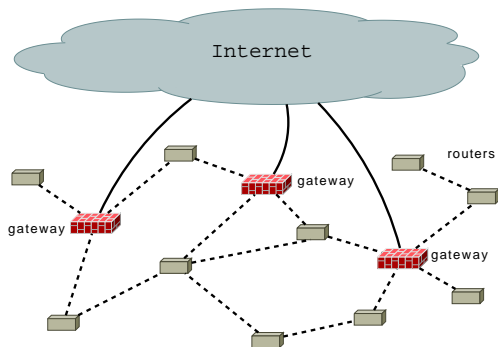
Travaux précédents

- Round Weighting Problem (*Klasing, Morales, Perennes*)

Modélisation du réseau

Graphe de connectivité $G = (V, E)$

- $\mathcal{AP} \subset V$: stations de base
- $V \setminus \mathcal{AP}$: routeurs
- E : connectivité entre deux nœuds



Joint gateway placement, routing and scheduling problem

Given connectivity graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ with links capacity \mathbf{c}_e and a subset of candidate nodes \mathbf{V}_c :

- \mathbf{d}_v : amount of packets router v has to send to the gateways,
- \mathbf{a}_e^t : 1 if link e is active during time slot t , 0 otherwise,
- \mathbf{x}_e^v : flow packets sent by router v going through link e ,
- \mathbf{s}_i : 1 if gateway i is selected (among the candidates),
- and $\mathbf{y}_{r,i}$: flow packets sent by router r received by gateway i .

Contraintes du problème

$$\sum_{e \in E_u \cup E_v} a_e^t \leq 1, \forall (u, v) \in E, t \leq T \quad (1)$$

$$\sum_{v \in V} x_e^v \leq c_e \cdot \sum_{t \leq T} a_e^t, \forall e \in E \quad (2)$$

$$\sum_{e=(u,v) \in E} x_e^r + \mathbb{1}_{\{v=r\}} d_v = \sum_{e=(v,u) \in E} x_e^r + y_{r,v}, \forall r, v \in V \quad (3)$$

$$\sum_{i \in V_c} y_{r,i} = d_r, \forall r \in V \quad (4)$$

$$y_{r,i} \leq s_i \cdot C, \forall i \in V_c, r \in V \quad (5)$$

Objectives

- Given node demands (d_v constants),
- Find minimum number of gateways needed such that every demand can be routed from sources to gateways.

$$\min \sum_{i \in V_c} s_i, \quad (6)$$

$$\text{subject to} \quad (1) - (5) \quad (7)$$

Objectives

- Given complete topology with gateways (s_i constants),
- Compute the maximum throughput with fairness among routers.

$$\max \min_{v \in V} d_v, \quad (8)$$

$$\textit{subject to} \quad (1) - (5) \quad (9)$$

Objectives

- Given a fixed number n of gateways to deploy,
- Find the location of the gateways while ensuring maximum fair throughput to the network.

$$\max \min_{v \in V} d_v, \quad (10)$$

$$\text{subject to} \quad (1) - (5) \quad (11)$$

$$\text{and} \quad \sum_{i \in V_c} s_i = n \quad (12)$$

Change the formulation

$$a_e^t + a_{e'}^t \leq 1, \forall e \in E, e' \in \mathcal{I}(e), t \in [1, T]$$

- order of the link activation does not matter
 - cost of the optimal solution does not change
 - not a real scheduling
 - Scheduling \Rightarrow Weighting (Round Weighting Problem [KLA 04])
-
- Scheduling \Leftrightarrow round activation at each slot of $[1, T]$
 - Routing \Leftrightarrow path activation between V_r and V_g

Path/Round formulation

- Variables: combinatorial objects (paths, rounds) already constructed
- Reduction of the number of constraints
- But exponential number of variables: use of column generation

$$\min \sum_{s \in \mathcal{S}} a_s \quad (13)$$

$$\text{Link capacity} \quad \sum_{p \ni e} f_p \leq \sum_{s \ni e} a_s, \quad \forall e \in E \quad (14)$$

$$\text{Router demand} \quad \sum_{p \in \mathcal{P}_r} f_p = d_r, \quad \forall r \in V_r \quad (15)$$

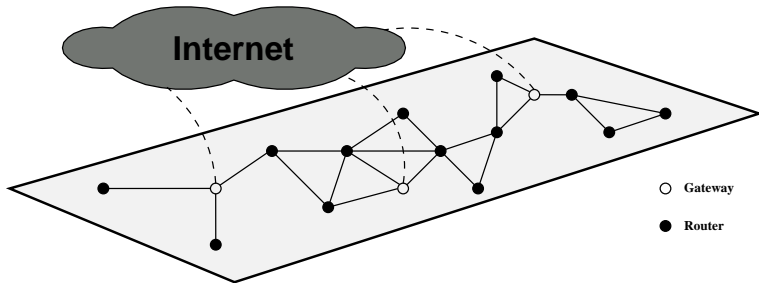
Results

- Random networks from 10 to 100 nodes
 - For each topology, from 1 to n/d gateways randomly chosen
 - n : network size
 - d : mean degree of the graph
 - For each gateway density, 5 random trials
-
- Per-node flow rate decreases with the network size
 - Detection of a critical area surrounding each gateway
 - Linear gain with a gateway addition
 - Minimal distance required between two gateways in the network

Max-flow & Min-Cut

Given a round weighting w :

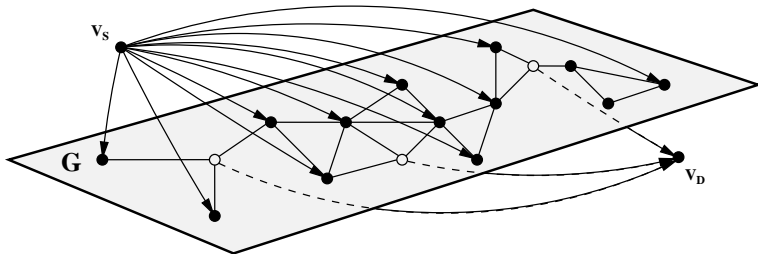
- Routing = maximum flow
- Feasible flow = flow between v_S and v_D of value $\sum_{v \in V_r} d_v$
- A cut S separating v_S and v_D has capacity $C_w(S) \geq \sum_{v \in V_r} d_v$
- Induced cut capacity $C_w(S) = \sum_{R \in \mathcal{R}} \delta(R, S) w(R)$



Max-flow & Min-Cut

Given a round weighting w :

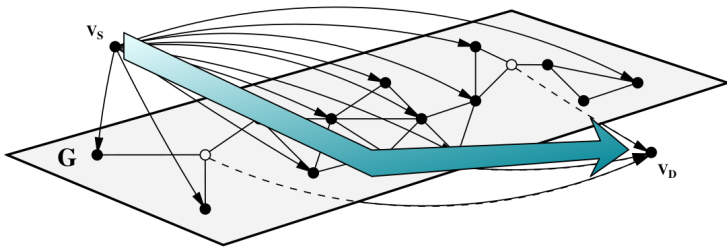
- Routing = maximum flow
- Feasible flow = flow between v_S and v_D of value $\sum_{v \in V_r} d_v$
- A cut S separating v_S and v_D has capacity $C_w(S) \geq \sum_{v \in V_r} d_v$
- Induced cut capacity $C_w(S) = \sum_{R \in \mathcal{R}} \delta(R, S) w(R)$



Max-flow & Min-Cut

Given a round weighting w :

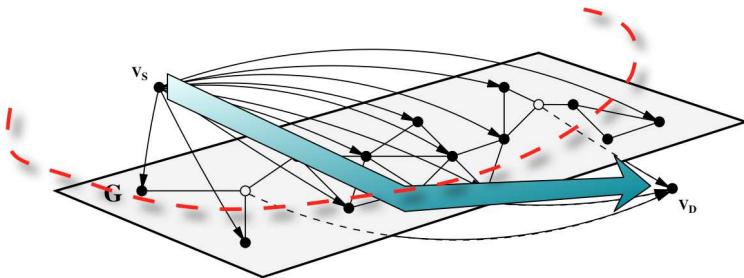
- Routing = maximum flow
- Feasible flow = flow between v_S and v_D of value $\sum_{v \in V_r} d_v$
- A cut S separating v_S and v_D has capacity $C_w(S) \geq \sum_{v \in V_r} d_v$
- Induced cut capacity $C_w(S) = \sum_{R \in \mathcal{R}} \delta(R, S) w(R)$



Max-flow & Min-Cut

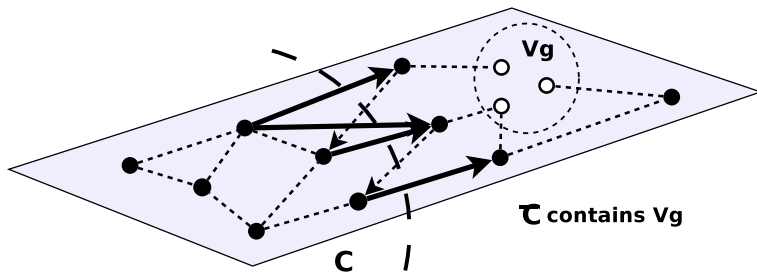
Given a round weighting w :

- Routing = maximum flow
- Feasible flow = flow between v_S and v_D of value $\sum_{v \in V_r} d_v$
- A cut S separating v_S and v_D has capacity $C_w(S) \geq \sum_{v \in V_r} d_v$
- Induced cut capacity $C_w(S) = \sum_{R \in \mathcal{R}} \delta(R, S) w(R)$



Round Weighting Problem \Leftrightarrow Cut Covering Problem

- \mathcal{C} : set of cuts of G separating some routers from the gateways
- Variables: combinatorial objects (cuts \mathcal{C} , rounds \mathcal{R}) already constructed
- Constraints: ensure that the traffic can cross the cut



Round Weighting Problem \Leftrightarrow Cut Covering Problem

- \mathcal{C} : set of cuts of G separating some routers from the gateways
- Variables: combinatorial objects (cuts \mathcal{C} , rounds \mathcal{R}) already constructed
- Constraints: ensure that the traffic can cross the cut

$$\min \sum_{R \in \mathcal{R}} w(R)$$

Cut covering by rounds $\sum_{R \ni \mathcal{C}} \delta(R, C) w(R) \geq t_C, \forall C \in \mathcal{C}$

Models analysis

Solutions: optimal w for flow = optimal w for cut covering

- No routing is computed
- Global constraints
- Exponential round set (as before) + exponential cut set

Models analysis

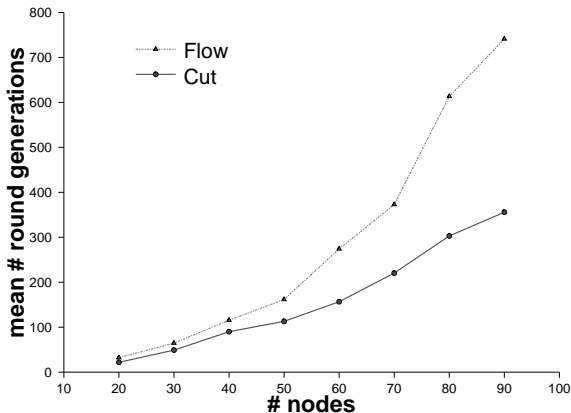
Solutions: optimal w for flow = optimal w for cut covering

- No routing is computed
- Global constraints
- Exponential round set (as before) + exponential cut set

Cross line and column generation process

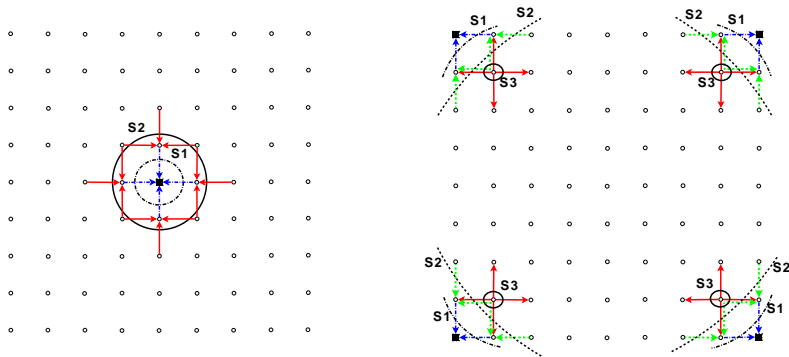
Results (1): # rounds generated

- Same order than flow formulation with CG
- But less rounds computed



Results (2): active cuts on grid topologies

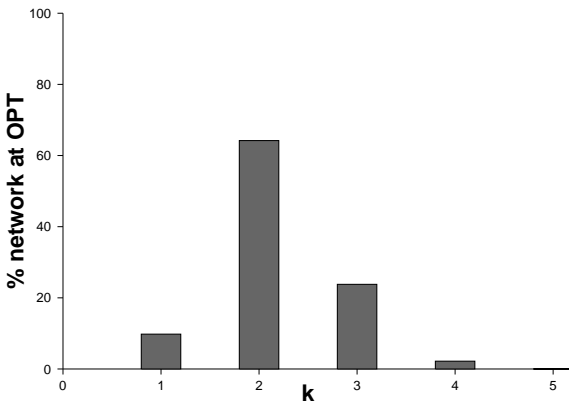
- $n \times n$ grid, gateway(s) in the center or at the corners.
- Important cuts: located in the neighborhood of the gtws



- Only a small number of cuts gives the bound on the capacity
- These cuts are located into the bottleneck area

Results (3): highlighting the bottleneck area

- Only generate cuts and rounds in the k -neighborhood of the gateways
- Complexity improved



avec MEV : influence des acquittements sur la capacité

Une étude quantitative

Comparaison de deux modèles :

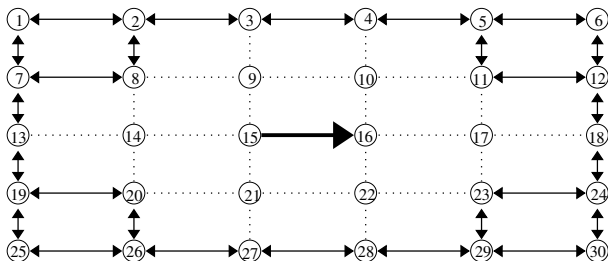
- ack au niveau MAC : modèle interférences binaires symétriques,
- ack niveau transport : interférences asymétriques

Evaluation de la capacité

Acquittements au niveau MAC

Interférences binaires symétriques

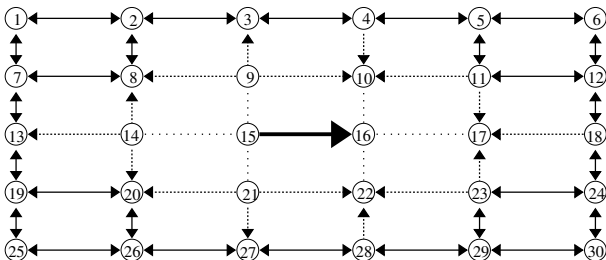
- Wifi 802.11
- Nœud à la fois émetteur et récepteur
- Interférences à deux sauts



Acquittements au niveau transport

Interférences binaires asymétriques

- Nœud émetteur : il n'attend pas d'acquittements
- Interférences à un saut



Programme linéaire

Variables

- a_s : durée d'activation du round s au cours d'une période
- f_p : flot envoyé sur le chemin p
- $x_e = \begin{cases} e & \text{dans le premier modèle} \\ \text{l'arc opposé à } e & \text{dans le second modèle} \end{cases}$

$$\min \sum_{s \in \mathcal{S}} a_s$$

Satisfaction de la demande

$$\sum_{p \in \mathcal{P}_k} f_p = d_k, \quad \forall k \in K$$

Contrainte de capacité

$$\sum_{p \in \mathcal{P}, p \ni e} f_p + \alpha \sum_{p \in \mathcal{P}, p \ni x_e} f_p \leq \sum_{s \ni e} a_s, \quad \forall e \in E$$

\mathcal{S} et \mathcal{P} exponentiels \Rightarrow résolution par génération de colonnes

Etude de l'influence des acquittements

Simulations

- Implémenté dans Mascot
- Instances aléatoires 10 à 100 nœuds
- Grilles régulières, réseaux aléatoires

Résultats

- Capacité meilleure en interférence asymétrique
- Gain de plus de 20% pour topologies en grille
- Charge des nœuds mieux répartie en interférence asymétrique
- Utilisation de plus de routes

Simulation

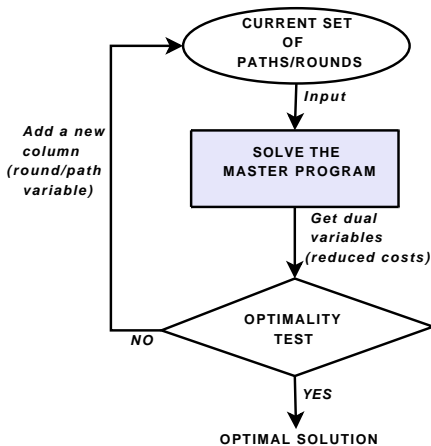
Dans Mascot :

- Implémentation des modèles linéaires
- Génération de colonnes
- Génération croisée de lignes et colonnes

The column generation process

Set of initial variables:

- paths = shortest paths in terms of hops,
- rounds = singletons



Optimality test: auxiliary program computation

Given reduced costs:

- either gives a new column that improves the solution of (MP),
- or certify that no constraints are violated.

- shortest path algorithm:

$$\text{find } p \text{ such that } \sum_{e \in p} y_e < x_{O(p)}.$$

- maximum independent set algorithm in the conflict graph :

$$\text{find } s \text{ such that } \sum_{e \in s} y_e > 1.$$

Primal-dual resolution method

- Initial restricted sets : $\mathcal{C}_0 = \{V_r\}$, $\mathcal{R}_0 = \{\{e\}, \forall e \in E\}$
- Column generation: compute a cut covering by the rounds
- Line generation: add non-covered cuts
- Optimal solution: when no rounds/cuts violate the primal/dual constraints

Master program (MP)

$$\min \sum_{R \in \mathcal{R}} w(R)$$

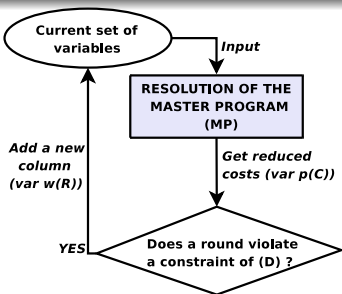
$$\sum_{R \ni C} \delta(R, C) w(R) \geq t_C, \forall C \in \mathcal{C}$$

Dual program (D)

$$\max \sum_{C \in \mathcal{C}} t_C p(C)$$

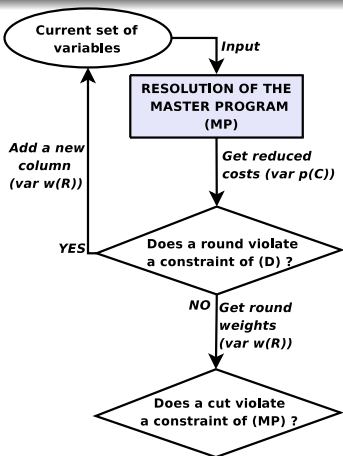
$$\sum_{C \in \mathcal{C}} \delta(R, C) p(C) \leq 1, \forall R \in \mathcal{R}$$

Primal-dual algorithm



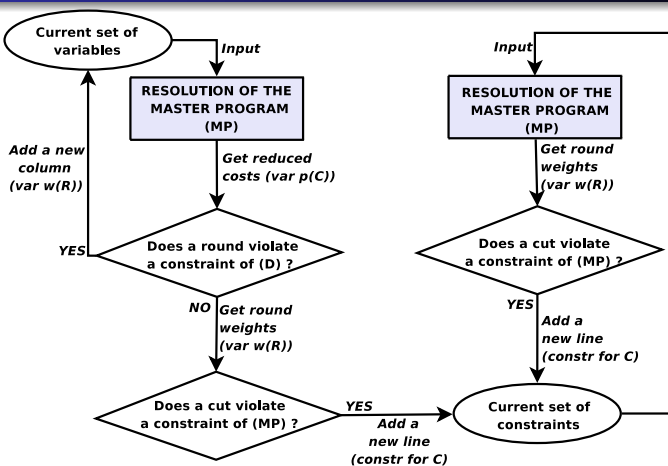
- 1 CG: add new rounds \rightarrow objective decreases

Primal-dual algorithm



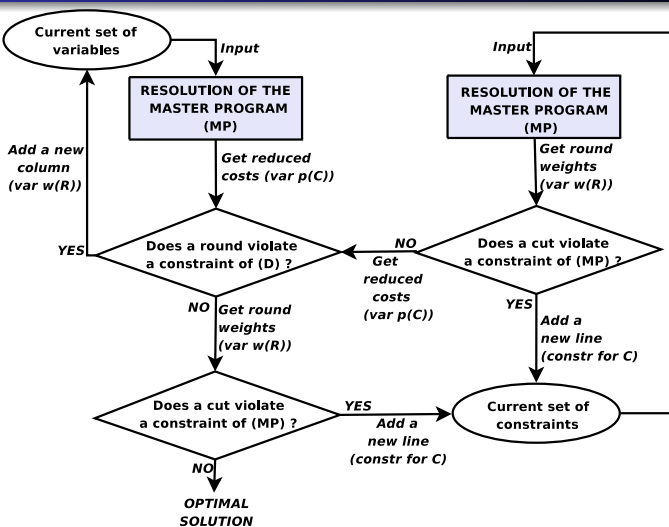
- 1 CG: add new rounds \rightarrow objective decreases

Primal-dual algorithm



- 1 CG: add new rounds \rightarrow objective decreases
- 2 LG: add new cuts \rightarrow objective increases

Primal-dual algorithm



- 1 CG: add new rounds \rightarrow objective decreases
- 2 LG: add new cuts \rightarrow objective increases

Algorithme distribué de Clustering

En fonction des demandes sur les nœuds, décomposer le réseau radio maillé en clusters, chaque point d'accès est un clusterhead
⇒ équilibrer la charge, routage distribué

Métrique des nœuds

$$\mathcal{I}(u) = \sum_{v \in \Gamma_2(u)} d(v)$$

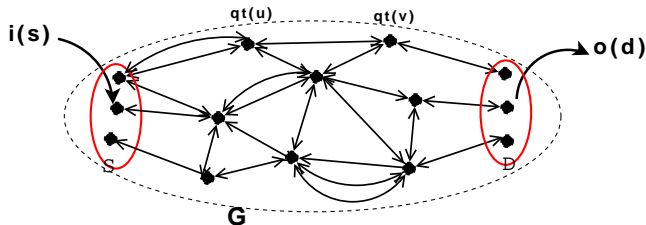
Algorithme distribué

Chaque nœud se rattache à son voisin de plus petite métrique.

Stabilité d'un algorithme glouton distribué

Modèle de réseau

- un réseau $G = (V, E)$ synchrone,
- chaque nœud possède une file d'attente $q_t(v)$,
- un sous-ensemble de sources \mathcal{S} et de destinations \mathcal{D} tels que à chaque étape :
 - chaque source s injecte $in(s)$ paquets dans sa file d'attente,
 - chaque lien peut transporter au plus 1 paquet,
 - chaque destination d extrait $out(d)$ paquets.



Algorithme distribué de type gradient local

A chaque instant t :

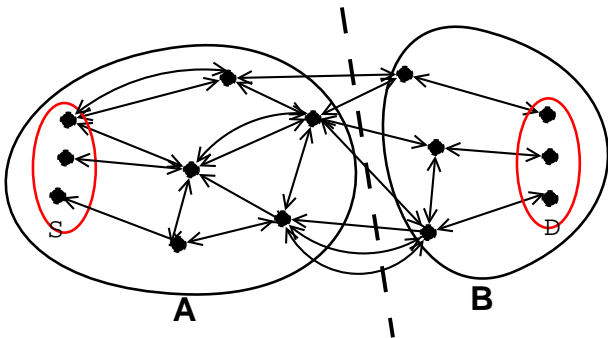
- chaque s injecte $in(s)$ paquets,
- $\forall u \in V, v \in \Gamma(u)$, si $q_t(v) < q_t(u)$, alors u envoie un paquet à v (s'il le peut),
- chaque d extrait $out(d)$ paquets.

Théorème

Si le taux d'arrivée est réalisable, i.e. $\sum_{s \in \mathcal{S}} in(s) \leq f^*$ avec f^* valeur du flot maximum entre \mathcal{S} et \mathcal{D} , alors le système est stable. Sinon, le nombre de paquets en transit peut diverger avec le temps quel que soit l'algorithme.

Idée de la preuve

- Etat du réseau $P(t) = \sum_{v \in V} q_t^2(v)$,
- Taux d'arrivée strictement faisable : comparaison au flot,
- $\sum_{s \in S} in(s) = f^*$: preuve par induction sur la taille du réseau.



Idée de la preuve

- Etat du réseau $P(t) = \sum_{v \in V} q_t^2(v)$,
- Taux d'arrivée strictement faisable : comparaison au flot,
- $\sum_{s \in S} in(s) = f^*$: preuve par induction sur la taille du réseau.

