SAT EST IN BONO LOCO Oui mais combien ? et où ?

Rohan Fossé, Mohamed Lamine Lamali, Paul Ouvrard

SAT EST IN BONO LOCO
Oui mais combien ? et où ?

Rohan Fossé[1] et Mohamed Lamine Lamali[1 †] et Paul Ouvrard[1]

[1] *LaBRI - Univ. de Bordeaux, France*

Dans un réseau hétérogène, plusieurs protocoles de communication coexistent, par exemple IPv4 et IPv6. Afin d'assurer l'interopérabilité et la communication, des convertisseurs (e.g., une configuration Dual Stack) sont placés sur certaines machines. Nous nous intéressons au problème de minimiser le nombre de convertisseurs à placer dans le réseau tout en assurant la communication entre chaque paire de machines. Dans ce travail préliminaire, nous montrons que le problème est NP-complet. Nous proposons également une réduction vers SAT qui permettra d'utiliser des solveurs pour résoudre le problème.

Mots-clefs : Conversion de protocoles, Interopérabilité IPv4/IPv6, Complexité, Réduction SAT.

---

1 Introduction

Les réseaux actuels contiennent plusieurs protocoles[‡], et certains équipements n'utilisent pas forcément des protocoles compatibles. L'exemple le plus commun est celui d'IPv4/IPv6. En effet, pour que deux machines utilisant ces deux protocoles puissent communiquer, il faut qu'il y ait *conversion* du format des paquets sur le chemin. Cette situation risque de devenir de plus en plus courante. En effet, les objets connectés, dont le nombre est amené à croître drastiquement, n'utilisent pas forcément les mêmes protocoles. Pour les réseaux de capteurs, par exemple, on peut citer ZigBee et 6LoWPAN, qui nécessitent une conversion pour communiquer [LLW11].

Pour permettre une communication globale, il faut placer des convertisseurs sur certains nœuds du réseau. Par exemple, pour le cas IPv4/IPv6, on dispose de plusieurs technologies de conversion (e.g., Dual Stack, Stateless IP/ICMP Translation, etc.). Or, installer des convertisseurs a un coût de configuration, en temps et en ressources. Le problème traité par cet article est celui de la minimisation du nombre de convertisseurs nécessaire pour assurer la communication globale dans le réseau, i.e., entre chaque paire de nœuds. Nous prouvons que le problème est NP-complet, même sur une classe particulière de graphes : les graphes scindés (*Split Graphs*, en anglais). Puis nous proposons une réduction de notre problème vers le problème SAT, afin de pouvoir utiliser des solveurs pour le résoudre. Ceci est un travail préliminaire qui a pour but d'être complété, notamment par l'évaluation de l'efficacité des solveurs SAT sur ce problème via des simulations.

2 Description du problème et formalisation

On considère un réseau où il existe deux protocoles de communications a et b (IPv4 et IPv6 par exemple). Certains nœuds supportent le protocole a, et d'autres le protocole b.

On modélise un tel réseau par un graphe non orienté $G = (U \cup V, E)$ où U est l'ensemble des nœuds qui supportent le protocole a, et V l'ensemble des nœuds qui supportent le protocole b. Il est à noter que $U \cap V = \emptyset$ (i.e., U et V forment une partition de l'ensemble des nœuds) mais

---

† Financé par le projet HÉRA ANR-18-CE25-0002.

‡. Dans cet article, le terme *protocole* signifie *protocole de communication*, à ne pas confondre avec un *protocole de routage*.

le graphe n'est pas forcément biparti. Pour faciliter la compréhension, on notera $u, u_i, u', \dots$ (resp. v, v_i, v'...) les nœuds de U (resp. de V).

Même s'il existe un lien $uv \in E$, u et v ne peuvent pas communiquer car ils n'utilisent pas le même protocole de communication. Ainsi, pour permettre au réseau d'être *réellement connexe*, et que tous les nœuds puissent communiquer entre eux, il faudra placer des *convertisseurs* de protocoles sur certains nœuds. Un convertisseur est un système (généralement une configuration, comme Dual-Stack) permettant de convertir les paquets reçus dans un certains protocole de communication en un autre (typiquement de IPv4 à IPv6 et inversement).

Définition 1. *Nous considérons deux types de liens :*

- *Un lien uu' où vv', c'est-à-dire entre deux nœuds utilisant le même protocole de communication, est dit homogène. L'ensemble de ces liens est noté H_m ;*
- *Un lien uv, dont les deux extrémités utilisent des protocoles de communication différents, est dit hétérogène. L'ensemble de ces liens est noté H_t.*

Soit $C \subset U \cup V$ l'ensemble de nœuds sur lesquels on va placer des convertisseurs.

Définition 2. *Un chemin $w_0 w_1 w_2 \dots w_{\ell-1} w_\ell$ (où $w_i \in U \cup V$ et $0 \le i \le \ell$) est dit valide si et seulement si pour chaque lien hétérogène $w_i w_{i+1}$ $(0 \le i \le \ell - 1)$ appartenant au chemin, soit $w_i \in C$, soit $w_{i+1} \in C$.*

Il est clair qu'un chemin valide permet à ses deux extrémités (et plus généralement à tous les nœuds qui le composent) de communiquer, i.e., qu'il permet une continuité protocolaire. Le problème consiste à minimiser le nombre de convertisseurs nécessaires afin que tous les nœuds puissent communiquer.

Problème 1.

$$\begin{aligned} &\min |C| \\ &t.q.\ \ \forall s,t \in U \cup V,\ \textit{Il existe un chemin valide entre } s \textit{ et } t \textit{ dans } G. \end{aligned}$$

3 Complexité du problème

Dans cette section, nous prouvons que le problème de décision associé à Problème 1, i.e., existe-t-il un ensemble de convertisseurs C, qui permet la communication globale, tel que $|C| \le k$, est NP-complet. Le problème est dans NP, puisqu'avec l'ensemble C nous pouvons vérifier que tous les nœuds communiquent. Ainsi, nous prouvons que la version décisionnelle du Problème 1 est NP-difficile en fournissant une réduction en temps polynomial à partir du problème MULTICUT.

Problème 2 (MULTICUT)**.** *Soit $G = (W,E)$ un graphe non orienté*[§]*, ℓ un ensemble de paires de nœuds distincts $\{(s_1,t_1),(s_2,t_2),\cdots,(s_\ell,t_\ell)\}$, et k un entier.*

Existe-t-il un ensemble de lien de taille au plus k, tel que la suppression de cet ensemble entraîne que pour tout $i = 1,2,\dots,\ell$, les nœuds s_i et t_i appartiennent à deux différentes composantes connexes ?

Rappelons qu'un graphe à n nœuds est un graphe *étoile* s'il contient $n-1$ nœuds de degré un $w_1, w_2, ..., w_n$ reliés au même nœuds central w_0. Évidemment, il n'existe qu'un seul chemin entre n'importe quelle paire de nœuds dans un graphe étoile. Garg *et al.* ont prouvé que MULTICUT est NP-complet, même si le graphe en entrée est une étoile [GVY97]. Un *graphe scindé* est un graphe $G = (W,E)$ tel que l'ensemble de ses nœuds W peut être partitionné en deux sous-ensembles A et B, tels que le graphe induit par A est une clique et le graphe induit par B est un stable.

Nous prouvons le résultat suivant :

Proposition 1. *Le problème de décision associé au Problème 1 est* NP*-difficile, même si le graphe en entrée $G = (U \cup V, E)$ est un graphe scindé.*

§. L'ensemble des nœuds est généralement noté V, néanmoins, nous le notons W pour éviter la confusion avec la notation définie dans le modèle. On notera ainsi par W l'ensemble des nœuds d'un graphe quelconque, i.e., non partitionné en deux sous-ensembles de nœuds.

(a) Un graphe étoile G.

(b) Le graphe scindé correspondant G'.

FIGURE 1: Réduction utilisée dans la preuve de la Proposition 1 avec (a) $I = (G, \{(w_1, w_3), (w_2, w_4), (w_3, w_5)\},\ k = 2)$; et (b) $I' = (G',\ k = 2)$.

Démonstration. Soit $G = (W, E)$ un graphe étoile avec $n+1$ nœuds, où $W = \{w_0, w_1, w_2, \cdots, w_{n-1}, w_n\}$, le nœud w_0 étant le nœud central. Soit $I = (G, \{(s_1, t_1), \cdots, (s_\ell, t_\ell)\},\ k)$ une instance de MULTICUT.

Le graphe correspondant $G' = (U \cup V, E')$ ayant $n + \ell$ nœuds est construit comme suit : pour chaque nœud w_i de G (à l'exception de w_0), on crée un nœud correspondant u_i dans G', puis nous relions tous ces nœuds par des liens afin de former une clique de taille n. Ensuite, pour chaque paire (s_i, t_i), on crée un nœud v_i relié aux nœuds correspondant à s_i et t_i. Tous les v_i sont deux à deux non adjacents. Ainsi, G' est un graphe scindé. La Figure 1 donne un exemple où les liens rouges en pointillé sur la Figure 1a constituent une solution de MULTICUT et les nœuds rouges sur la Figure 1b constituent la solution correspondante du Problème 1. Ici, nous posons que chaque $u \in U$ (resp. $v \in V$) utilise le protocole de routage a (resp. b). Ainsi, tous les liens hétérogènes relient les nœuds de la clique (induite par U) à ceux du stable (induit par V). Cette réduction est polynomiale. Soit $I' = (G' = (U \cup V, E'),\ k)$ l'instance ainsi construite de Problème 1. Nous donnons une idée de la preuve que $G = (W, E)$ a un MULTICUT de taille k, i.e., I est une instance positive de MULTICUT, si et seulement si $G' = (U \cup V, E')$ a un ensemble de convertisseurs C tel que $|C| = k$ et qu'il existe un chemin valide entre chaque couple de nœuds, i.e., I' est une instance positive de Problème 1.

Pour le « seulement si », considérons $F \subset E$ un sous-ensemble de liens déconnectant les paires (s_i, t_i) de G de taille k, i.e., une solution pour l'instance de MULTICUT. Tous ces liens sont de la forme $w_0 w_i$ où $w_i = s_i$ ou $w_i = t_i$, vu que G est une étoile. Si pour chaque $w_0 w_i \in F$, on place un convertisseur sur le nœud u_i dans G' (correspondant à w_i dans G), i.e., $u_i \in C$, alors tous les nœuds v_i ont pour voisin un convertisseur (puisqu'ils correspondent aux paires à déconnecter dans G). Ainsi, ils peuvent communiquer avec tous les autres nœuds.

Pour le « si », considérons un ensemble de convertisseurs C de taille k dans G, i.e., une solution au Problème 1. On peut supposer sans perte de généralité que $C \subset U$, autrement, il suffirait de déplacer le convertisseur d'un sommet v vers l'un de ses voisins dans U tout en gardant les chemins valides. Or, pour chaque $u_i \in C$, on peut déconnecter le lien $w_0 w_i$ dans G, et ainsi séparer la paire (s_i, t_i). On peut donc construire une solution de MULTICUT $F = \{w_0 w_i \mid u_i \in C\}$. □

4 Réduction vers le problème SAT

Le problème `SAT` est un problème central en informatique. C'est le premier à avoir été prouvé NP-complet, et par conséquent tous les problèmes appartenant à NP s'y réduisent. Les solveurs `SAT` sont de plus en plus utilisés dans l'industrie pour résoudre des problèmes d'optimisation difficiles. Ces solveurs ont montré une très grande efficacité, même sur des problèmes comportant des millions de variables [Knu15]. D'où l'idée de les utiliser pour résoudre notre problème.

Rappelons qu'une formule `SAT` est une expression avec des variables booléennes et des opérateurs logiques sur ces variables (*et* ou *conjonction* $\wedge$, *ou* ou *disjonction* $\vee$, *négation* $\neg$, etc.).

Nous construisons une formule `SAT` à partir d'une instance de notre problème $I = (G = (U \cup V, E),\ k)$. Notons tout d'abord n le nombre de nœuds du graphe, et $N(w)$ l'ensemble des voisins d'un nœud w. Pour faciliter la réduction vers `SAT`, nous attribuons un ordre arbitraire aux nœuds contenant un convertisseur. La variable $y_{w,i}$ ($w \in U \cup V$ et $1 \leq i \leq k$) est définie comme étant à *vrai* si et seulement si w est le i^{e} nœud de C, i.e., w contient un convertisseur.

Pour une paire donnée de nœuds (s,t), nous définissons la variable booléenne $x_{w,i,s,t}$ ($1 \leq i \leq \ell$, où ℓ est la longueur du chemin entre s et t en nombre de liens) comme étant à *vrai*, si et seulement si w est le i^{e} nœud du chemin entre s et t. Ainsi, nous aurons toujours $x_{s,0,s,t} = vrai$ et $x_{t,\ell,s,t} = vrai$. Pour une paire (s,t) donnée, nous devons nous assurer qu'il existe un chemin de longueur ℓ entre s et t, que s est le nœud numéro 0 du chemin et que t est le numéro ℓ. Et que pour chaque w, i^{e} nœud du chemin, w', $i+1^{e}$ nœud du chemin, w' est un voisin de w, i.e., $w' \in N(w)$. Cette contrainte s'exprime ainsi :

$$Chemin_{s,t} = \bigvee_{1 \leq \ell \leq n-1} \left[x_{s,0,s,t} \wedge x_{t,\ell,s,t} \wedge \left(\bigwedge_{0 \leq i \leq \ell} \left(\bigvee_{w \in U \cup V} x_{w,i,s,t} \wedge \left(\bigvee_{w' \in N(w)} x_{w',i+1,s,t} \right) \right) \right) \right] \tag{1}$$

Néanmoins, cette formule ne garantit pas que le chemin soit valide. En effet, il faudrait que pour chaque lien ww' *hétérogène* ($ww' \in H_t$) du chemin, un des deux nœuds contiennent un convertisseur. Cette formule sera de la forme « si $ww' \in H_t$ alors w ou w' ont un convertisseur », c'est-à-dire

$$conv_{w,w'} = \left(\bigvee_{1 \leq j \leq k} y_{w,j} \right) \vee \left(\bigvee_{1 \leq j \leq k} y_{w',j} \right) \tag{2}$$

La partie « si $ww' \in H_t$ » s'exprime par un *et* ($\wedge$) sur le lien $ww' \in H_t$. En effet, si cette appartenance est fausse, la formule *et* sera vraie car vide. Autrement, la contrainte doit être satisfaite. Ainsi, formule (2) devient

$$\bigvee_{1 \leq \ell \leq n-1} \left[x_{s,0,s,t} \wedge x_{t,\ell,s,t} \wedge \left(\bigwedge_{0 \leq i \leq \ell} \left(\bigvee_{w \in U \cup V} x_{w,i,s,t} \wedge \left(\left(\bigvee_{w' \in N(w)} x_{w',i+1,s,t} \right) \wedge \bigwedge_{ww' \in H_t} conv_{w,w'} \right) \right) \right) \right] \tag{3}$$

Bien entendu, cette formule ne sera équivalente à une instance du Problème 1 que si certaines règles d'unicité sont respectées : par exemple, un nœud ne peut pas être le i^{e} et le j^{e} nœud de C si $i \neq j$, i.e., $y_{w,i}$ et $y_{w,j}$ ne doivent pas être à *vrai* toutes les deux si $i \neq j$. De même, deux nœuds ne doivent pas avoir la même position dans C, ni dans un chemin quelconque. La contrainte d'unicité d'un ensemble de variable $\{x_1, \ldots, x_n\}$ s'exprime facilement par une formule `SAT` par $Unique(\{x_1, \ldots, x_n\}) = (\bigvee_{1 \leq i \leq n} x_i) \wedge (\bigwedge_{1 \leq i < j \leq n} \neg x_i \vee x_j)$ La première partie de la formule peut-être supprimée pour obtenir une contrainte *au plus un*. Il suffira ensuite d'appliquer ces formules à toutes les ensembles de variables sus-cités. Ainsi, cette formule sera à *vrai* si et seulement si on peut placer k convertisseurs dans le réseau tout en garantissant une communication globale. De plus, les variables $y_{w,i}$ à *vrai* nous indiqueront les nœuds w où placer les convertisseurs.

Ce travail étant préliminaire, beaucoup de pistes de travaux futurs sont envisagées. Premièrement, l'évaluation de l'efficacité de différents solveurs pour résoudre le problème. Nous envisageons également de concevoir des algorithmes FPT ¶ exacts paramétrés par le nombre de convertisseurs. Enfin, il serait intéressant d'étendre le travail au cas où il existe plus que deux protocoles.

Références

[GVY97] N. Garg, V. V. Vazirani, and M. Yannakakis. Primal-dual approximation algorithms for integral flow and multicut in trees. *Algorithmica*, 18(1) :3–20, May 1997.

[Knu15] Donald E Knuth. *The Art of Computer Programming, Volume 4, Fascicle 6 : Satisfiability*. Addison-Wesley Professional, 2015.

[LLW11] Chia-Wen Lu, Shu-Cheng Li, and Quincy Wu. Interconnecting zigbee and 6lowpan wireless sensor networks for smart grid applications. In *2011 Fifth International Conference on Sensing Technology*, pages 267–272. IEEE, 2011.

¶. *Fast Parameter Tractable.*