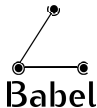


Quelques commentaires sur le protocole de routage Babel

Juliusz Chroboczek
IRIF
Université Paris-Diderot (Paris 7)

15 mars 2018



Conclusion

Babel est un protocole de routage à vecteurs de distance

- qui marche ;
- simple et facile à implémenter ;
- robuste ;
- extensible ;
- inclus dans FRR (Quagga) et BIRD ;
- documenté (RFC 6126, expérimentale) ;
- en cours de normalisation à l'IETF.

<https://www.irif.fr/~jch/software/babel/>



Algorithmes de routage

Les protocoles de routage sont basés sur un de deux algorithmes :

Les protocoles à **état de lien** (OSPF, IS-IS) inondent la topologie, puis calculent les chemins les plus courts localement (Dijkstra) :

- difficiles à implémenter ;
- subtils et donc difficiles à étendre ;
- **marchent très bien.**

Les protocoles à **vecteur de distances** (RIP) utilisent une variante distribuée de l'algorithme de Bellman-Ford :

- faciles à comprendre et implémenter ;
- robustes et donc faciles à étendre ;
- **ne marchent pas.**

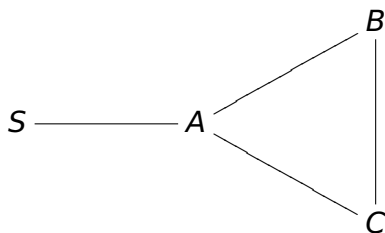
Le vecteur de distances, ça marche

Le vecteur de distances pur, **ça ne marche pas**.

Pour faire marcher le vecteur de distances, il faut un **mécanisme supplémentaire** :

- BGP : AS-path ;
- EIGRP : faisabilité + phase active ;
- DSDV : faisabilité + routes séquencées ;
- **Babel** : faisabilité + routes séquencées + requêtes explicites.

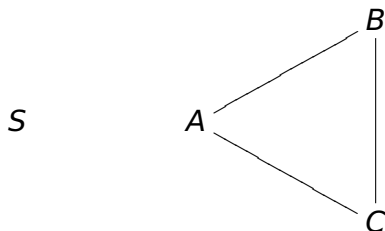
Vecteur de distances



S	0	0	0	0
A	∞	1, nh = S	1, nh = S	1, nh = S
B	∞	∞	2, nh = A	2, nh = A
C	∞	∞	2, nh = A	2, nh = A

Converge en $O(\Delta)$.

Comptage à l'infini



A	1, nh = S	3, nh = B	3, nh = B	3, nh = B
B	2, nh = A	2, nh = A	3, nh = C	3, nh = C
C	2, nh = A	2, nh = A	2, nh = A	4, nh = A

Converge en $O(\infty)$. Il faut choisir un petit ∞ .
Avant convergence, il y a une **boucle de routage**.

Vecteur de distances sans boucles : faisabilité

On définit la **distance de faisabilité** de X comme la plus petite métrique que X a annoncée :

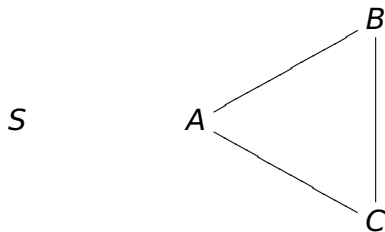
$$fd(X) = \min_{t \leq \text{now}} d(X).$$

Lorsqu'un voisin Y annonce une métrique $d(Y)$, la route est **faisable** pour X si

$$d(Y) < fd(X).$$

Si **chaque routeur ignore les routes infaisables**, il n'y aura pas de boucles.

Faisabilité : exemple

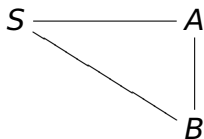


A	1, fd = 1	∞ , fd = 1	∞ , fd = 1	∞ , fd = 1
B	2, fd = 2	2, fd = 2	∞ , fd = 2	∞ , fd = 2
C	2, fd = 2	2, fd = 2	∞ , fd = 2	∞ , fd = 2

Converge en $O(\Delta)$.

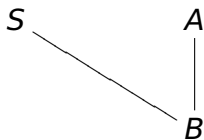
Faisabilité : famine

La condition de faisabilité cause une **famine**.



$$d(A) = 1, fd(A) = 1$$

$$d(B) = 1, fd(B) = 1$$



$$fd(A) = 1$$

$$d(B) = 1$$

La seule route disponible n'est **pas faisable**.

Résoudre la famine : routes séquencées

Les annonces de routes sont équipées d'un **numéro de séquence** en plus de la métrique :

$$(s, d(B))$$

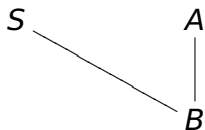
où $s \in \mathbf{N}$ est incrémenté **par la source**.

On définit

$$(s, m) \leq (s', m') \text{ lorsque } s > s' \text{ ou} \\ s = s' \text{ et } m \leq m'$$

$$\text{feasible}(s, m) \equiv (s, m) < \text{fd.}$$

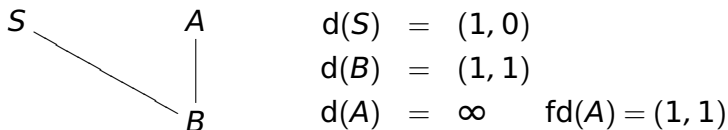
Routes séquencées : exemple



S	$(1, 0)$	$(2, 0)$	$(2, 0)$
A	$\infty, \text{fd} = (1, 1)$	$\infty, \text{fd} = (1, 1)$	$(2, 2), \text{fd} = (2, 2)$
B	$(1, 1), \text{fd} = (1, 1)$	$(2, 1), \text{fd} = (2, 1)$	$(2, 1), \text{fd} = (2, 1)$

Famine temporaire

Requêtes explicites



A doit **attendre** que S génère un nouveau seqno, et que celui-ci se propage.

Comment S sait-il qu'il faut générer un nouveau seqno ?

A envoie une **requête explicite** à B qui la fait suivre à S pour demander un **nouveau numéro de séquence**.

Robustesse et extensibilité

La preuve de correction de Babel repose sur des hypothèses très faibles et bien définies :

- **causalité** : un message n'est pas reçu avant d'avoir été envoyé ;
- **monotonie stricte** de la métrique :

$$m < c + m;$$

- **distributivité à gauche** :

$$m \leq m' \longrightarrow c + m \leq c + m'.$$

Robustesse

La correction ne dépend que de quelques hypothèses très faibles ce qui rend Babel **robuste** :

- **robuste** face aux bugs :
votre bug ne viole probablement pas les hypothèses ;
- **robuste** face aux réseaux bizarres :
votre réseau ne viole probablement pas les hypothèses.

Babel marche bien dans les **réseaux maillés**, où il a des performances comparables aux protocoles dédiés.

Extensibilité

La correction ne dépend que de quelques hypothèses très faibles ce qui rend Babel **extensible** :

- votre extension ne viole probablement pas les hypothèses.

Au cours des années, plusieurs extensions ont été développées :

- **routage sensible à la source** (SADR) (Matthieu Boutier);
- **routage sensible au délai**, pour réseaux *overlay* (Baptiste Jonglez);
- **routage sensible au ToS** (Gwendoline Chouasne);
- **routage sensible à la fréquence radio** (moi).

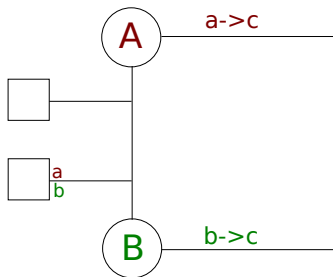
Le routage sensible à la source

Une extension importante

En routage *next-hop* classique, la décision de routage dépend de la **destination**.

En routage **sensible à la source** la décision de routage dépend de la paire **destination, source**.

Application : *multihoming* sans collaboration du fournisseur de service.



Conclusion

Babel est un protocole de routage à vecteurs de distance

- qui marche ;
- simple et facile à implémenter ;
- robuste ;
- extensible ;
- inclus dans FRR (Quagga) et BIRD ;
- documenté (RFC 6126, expérimentale) ;
- en cours de normalisation à l'IETF.

<https://www.irif.fr/~jch/software/babel/>

