

# Rapport de TP : CISCO

Benjamin André  
Alexis Lecocq  
Waelkens Dimitri

November 6, 2017

## 1 Structure

Nous avons un fichier `main.cbgp` qui appelle à tour de rôle :

- `config.cbgp`, qui crée les noeuds et génère la topologie intra-domaine
- `genroutes.cbgp`, qui crée statiquement les routes entre 2 AS, et initialise le protocole BGP sur les routeurs de bordure.
- `filterS.cbgp`, qui définit les filtres en fonctions des politiques locales à respecter.

## 2 Les noeuds

Chaque router se voit attribuer une IP selon une règle simple : s'il est le router X dans le préfixe A.B.C.0, il aura l'adresse A.B.C.X. Cette règle simplifie l'interprétation du réseau au complet, ainsi que les tests.

## 3 Les routes statiques

Des routes statiques sont définies entre des domaines différents. Il y est notamment renseigné le next-hop et les adresses joignables par celle-ci. On part du principe que lorsqu'un lien est établi entre deux entités, elles ont au moins communiqué leur propre adresse au technicien concerné chez l'autre.

## 4 Les filtres

### 4.1 Les stub

Pour les stub, aucun filtre est nécessaire. En effet, ceux-ci doivent annoncer toutes leurs routes vers leur providers.

## 4.2 Les autres

Pour les AS Spring, BigCarrier, GEANT, Abilene, nous avons veillé à ce qu'ils respectent les règles énoncées dans le cours. Ces règles sont résumées dans le tableau suivant:

De/Vers	Providers	Peer	Clients
Providers	Non	Non	Oui
Peer	Non	Non	Oui
Clients	Oui	Oui	Oui

Pour ce faire nous avons créé trois communautés. Une pour les clients, une pour les peers et une pour les providers. Pour chaque lien entrant de chacun des routeurs des AS, nous avons un filtre qui ajoute les routes à la bonne communauté.

Et pour chaque lien sortant, nous avons ajouter un filtre qui ignore (deny) les routes venant des communautés providers et peers. Les routes venant des client ne sont jamais ignorées.

Nous avons également remarqué qu'un AS pouvait avoir une relation commerciale et une relation non-commerciale avec différents providers. Nous avons donc ajouté des préférences sur les liens entrant venant de providers non-commerciaux. En effet, ceux-ci sont à préférer.

## 4.3 Local prefs

Dans certains cas, la même adresse peut avoir deux routes différentes. Une préférence local de 10 est attribuée au paires, contre 2 pour un provider. De cette façon, on préférera utiliser un lien à coût partagé qu'un lien pour lequel on paye plein pot. Une préférence de 100 est attribuée au routes venant d'un éventuel client assez maladroit pour communiquer ses routes.

## Difficultés rencontrées

La quantité de configuration et la très forte redondance de celle-ci représente un temps considérable d'écriture. Pour en limiter l'impact, nous avons écrit deux scripts en python générant la configuration globale de tiny internet.

Certains ajustements sont cependant faits à la main sur la sortie de ces scripts.