

# Configuration d'un routeur de bordure sous JUNOS 7+

---

Prise en main des matériels Juniper Networks  
Stratégies de configuration pour OSPF et BGP

Florian Le Goff - [Florian.Le-Goff@rp.lip6.fr](mailto:Florian.Le-Goff@rp.lip6.fr)

Version 1.0 - 30 mai 2007

# Introduction

---

- Cette présentation a pour but de vous présenter des exemples de stratégies de configuration de routeurs Juniper (versions récentes de JUNOS)
  - ➔ **Routage intra-domaine** : redistribution des routes en OSPF
  - ➔ **Routage inter-domaine** : échange et filtrage des routes BGP4+
- Ce ne sont que des **exemples**
- Vous pouvez contribuer à ce document; envoyez moi vos propositions et/ou corrections

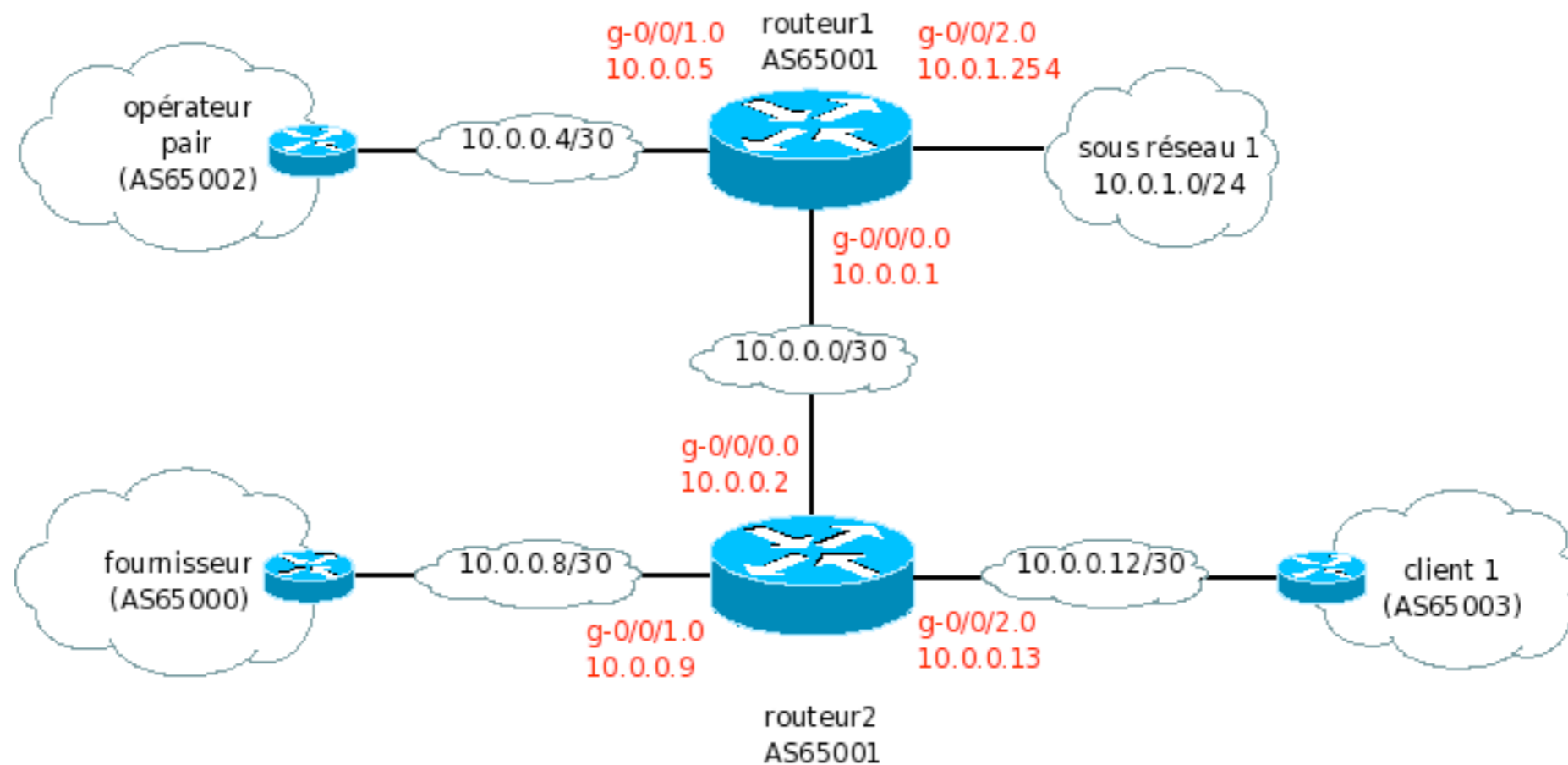
# Où se documenter ?

---

- La source principale d'information reste **le vendeur** (elle est ici extrêmement complète, ce qui peut également être un problème)
  - Index : <http://www.juniper.net/techpubs>
  - JUNOS : <http://www.juniper.net/techpubs/software/junos/>
- Deja News...^W **Google Groups**
  - <http://groups.google.com/>
- Support vendeur si vous achetez un contrat

# Topologie des exemples

- 2 routeurs de bordure
- Regroupés dans 1 AS (AS65001) et reliés à d'autres réseaux de hiérarchie différente : 1 peer, 1 client et 1 fournisseur



# Traduction Conf / CLI

---

- Commandes

```
florian@routeur1> configure
florian@routeur1# edit protocols
```

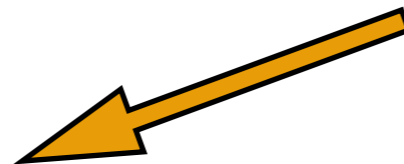
```
[edit protocols]
florian@routeur1# edit rsvp
```

```
[edit protocols rsvp]
florian@routeur1# set interface ge-0/0/1.0
florian@routeur1# set interface ge-0/0/3.0
florian@routeur1# delete interface ge-0/0/3.0
florian@routeur1# set ge-0/0/2.0
florian@routeur1# show
interface ge-0/0/1.0;
interface ge-0/0/2.0;
```

```
[edit protocols rsvp]
florian@routeur1# commit
```

- Résultat

```
florian@routeur1> show configuration
(...)
protocols {
  rsvp {
    interface ge-0/0/1.0;
    interface ge-0/0/2.0;
  }
}
(...)
```



# Routage intra-domaine exemple de OSPF

# Objectifs de l'IGP

---

- Nos routeurs gèrent des sous réseaux différents au sein d'un même système autonome...
- Comment savoir **où se trouve tel sous réseau** ?
  - ➔ En échangeant nos routes avec les autres routeurs
- Quel protocole ?
  - **Etat de lien** : vue globale du réseau ? OSPF / OSPFv3 (IPv6)
  - **Vecteur de distance** : routage des voisins ? RIP / RIPng (IPv6)

# OSPF : Etapes

---

- **Première étape**, définir à quelle area appartient chaque interface du routeur (suivant les sous réseaux dans lesquels elle est adressée)
  - <http://www.cisco.com/warp/public/104/8.html>
- **Deuxième étape**, rajouter chaque interface dans sa zone
- **Troisième étape**, vérifier que les interfaces connectées à un réseau de bordure sont marquées comme étant “**passive**”
  - ➔ Evite de parler OSPF aux autres AS (c'est le rôle de l'EGP)

# OSPF : Configuration

---

- Propositions de configuration

```
florian@routeur1> show configuration protocols
ospf {
  area 0.0.0.0 {
    interface ge-0/0/0.0;
  }
  area 3.3.3.3 {
    interface ge-0/0/1.0 {
      passive; /* pas d'échanges OSPF sur ce lien */
    }
  }
  area 4.4.4.4 {
    interface ge-0/0/2.0;
  }
}
```

```
florian@routeur2> show configuration protocols
ospf {
  area 0.0.0.0 {
    interface ge-0/0/0.0;
  }
  area 1.1.1.1 {
    interface ge-0/0/1.0 {
      passive; /* pas d'échanges OSPF sur ce lien */
    }
  }
  area 2.2.2.2 {
    interface ge-0/0/2.0 {
      passive; /* pas d'échanges OSPF sur ce lien */
    }
  }
}
```

- Contrôle du fonctionnement : **show ospf database, show ospf route, show ospf, show route / show route table inet.0**

# Routage inter-domaine BGP et filtres

# Objectifs de l'EGP

---

- Savoir quels sont les **réseaux accessibles par l'intermédiaire** d'un réseau voisin (préfixe de son réseau, éventuellement de ses voisins et de ses clients)
- **Annonce** des réseaux accessibles sur et via notre réseau (préfixes de notre réseau, préfixe de nos clients)
- Protection contre la découverte de notre réseau et l'**annonce de réseaux indésirables** (martians, bogons, limitation de nos clients)
  - ➔ Objectif 1 : monter une sessions BGP entre deux voisins
  - ➔ Objectif 2 : filtrer des routes en **import** et en **export**

# BGP : Configuration système

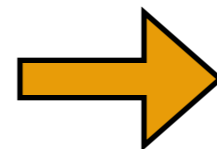
---

- Préciser à quel système autonome est relié ce routeur
  - Directive **routing-options autonomous-system**
- Choisir un identifiant unique pour ce routeur sur le réseau
  - Directive **routing-options router-id**

```
florian@routeur1> configure  
florian@routeur1# edit routing-options
```

```
[edit routing-options]  
florian@routeur1# set router-id 10.42.0.1  
florian@routeur1# set autonomous-system 65001  
florian@routeur1# show  
router-id 10.42.0.1;  
autonomous-system 65001;
```

```
[edit routing-options]  
florian@routeur1# commit
```



```
florian@routeur1> show configuration  
(...)  
routing-options {  
  router-id 10.42.0.1;  
  autonomous-system 65001;  
}  
(...)
```

# BGP : Connexion à un pair (I)

---

- Préciser l'adresse IP et le numéro de l'AS distant (le notre dans le cas d'une session iBGP). Il est possible de grouper des politiques.

```
florian@routeur1> show configuration
(...)
protocols {
  bgp {
    /* configuration minimale */
    neighbor 10.0.0.14 {
      peer-as 65003;
    }
  }
}
(...)
```

```
florian@routeur1> show configuration
(...)
protocols {
  bgp {
    group liaisons-client1 {
      import import-client1;
      export export-clients;
      peer-as 65003;
      neighbor 10.0.0.14;
      /* on peut également mettre plusieurs voisins */
      /* multipath;
      neighbor 10.0.10.12;*/
    }
  }
}
(...)
```

- Les attributs sont présentés sur la page suivante

# BGP : Pairs (2)

---

- **neighbor** : <@IPv4> | <@IPv6> adresse IP du voisin
- **peer-as** : <N°AS distant> adresse AS du voisin
- **import** : <nom> nom du script (dans policy-options policy-statement) à exécuter sur les routes reçues
- **export** : <nom> nom du script (dans policy-options policy-statement) à exécuter sur les routes à envoyer
- Pour tout le reste : <http://www.juniper.net/techpubs/software/junos/junos83/swconfig83-routing/id-13211253.html#id-13211253>

# BGP : Envoi des préfixes (I)

---

- Si le préfixe n'est pas déjà dans la table de routage (réseau + masque identique), il faut le rajouter en statique (comme sur IOS)

```
florian@routeur1> show configuration
(...)
routing-options {
  static {
    /* discard n'installe pas la route dans la table de routage */
    route 10.0.0.0/16 discard;
  }
}
(...)
```

- On précise ensuite un script d'export au niveau de notre session avec notre pair, appelé via la directive **export** dans la configuration d'une session BGP
  - Client : ré-envoi des routes apprises en BGP de nos fournisseurs
  - Fournisseur : on envoie seulement nos préfixes (et les clients)

# BGP : Envoi des préfixes (2)

---

- Stratégie d'export pour un **client**
  - Toutes les routes que l'on peut joindre
    - ➔ Routes reçues en BGP + notre route statique
- Stratégie d'export pour un **pair ou fournisseur**
  - Toutes les routes que l'on gère (nos préfixes + préfixes clients)
    - ➔ Route statique + préfixes clients
    - ➔ On ne devient pas AS de transit pour nos pairs !
- En général : moins on annonce de réseaux, moins il y a de trafic à venir vers nous et plus nos ressources sont préservées. Il faut donc faire un peu de politique avec les autres intervenants.

# BGP : Envoi des préfixes (3)

---

- Filtre pour l'export vers **un client** de notre système autonome

```
florian@routeur1> show configuration
(...)
policy-options {
  policy-statement export-clients {
    term 1 { /* extraire notre préfixe de static */
      from {
        protocol static;
        route-filter 10.0.0.0/16 exact;
      }
      then accept;
    }
    term 2 { /* routes reçues de nos pairs, fournisseurs et clients */
      from protocol bgp;
      then accept;
    }
    term 3 { /* on ne tient pas compte des autres routes statiques */
      then reject;
    }
  }
}
(...)
```

- **advertise-peer-as** : ré-annonce ses routes à un AS (par défaut : non)

# BGP : Envoi des préfixes (4)

---

- Filtre pour l'export vers **un pair ou fournisseur** de notre système autonome :

```
florian@routeur1> show configuration
(...)
policy-options {
  policy-statement export-pairs-fourni {
    term 1 { /* extraire notre préfixe de static */
      from {
        protocol static;
        route-filter 10.0.0.0/16 exact; /* notre préfixe */
      }
      then accept;
    }
    term 2 { /* routes reçues en BGP */
      from {
        protocol bgp;
        route-filter 192.168.1.0/24 exact; /* route de notre client 1 */
        route-filter 172.16.0.0/12 exact; /* route de notre client 2 */
      }
      then accept;
    }
    term 3 { /* on ne tient pas compte des autres routes */
      then reject;
    }
  }
}
(...)
```

# BGP : Filtrer les préfixes entrants

---

- On ne veut pas recevoir de **préfixes invalides**
  - ➔ Risque de router des paquets au mauvais endroit
- On ne veut pas recevoir nos **propres préfixes**
  - ➔ Risque de router nos données à travers un autre réseau (coût, sécurité et confidentialité, garanties de fiabilité)
- Utilisation de policy-statements appelés dans la directive **import** relative à un voisin BGP (fonctionnement similaire aux exports)
- D'une manière générale : **ne jamais faire confiance à ses voisins** (un accident est... si vite arrivé)

# BGP : Réception des préfixes

---

- Exemple : filtrer les préfixes annoncés par un client : il n'a le droit d'annoncer que son préfixe

```
florian@routeur1> show configuration
(...)
policy-options {
  policy-statement import-client1 {
    term 1 { /* toutes les routes reçues en BGP */
      from {
        route-filter 192.168.1.0/24 exact; /* route du client 1 */
      }
      then accept;
    }
    term 2 { /* on ne tient pas compte des autres routes */
      then reject;
    }
  }
}
(...)
```

# BGP : Outils de contrôles

---

- Préfixes reçus pour un voisin
  - `show route receive-protocol bgp <IP voisin>`
- Préfixes envoyés à un voisin
  - `show route advertising-protocol bgp <IP voisin>`
- Préfixes reçus en BGP
  - `show route protocol bgp`
- Etat des sessions
  - `show bgp summary`

# EGPs : Important

---

- Votre réseau est à la merci des annonces qu'il **reçoit**
  - Attention à ne laisser passer que ce qui est nécessaire
  - Limite systématique du nombre de préfixes max
  - Obligatoirement filtrer les annonces reçues
  - Il faut au minimum la configuration Secure BGP :
    - ➔ <http://www.cymru.com/gillsr/documents/junos-bgp-template.pdf>
- Et celui de vos voisins, à ce que vous **envoyez**
  - Pas plus de réseaux annoncés que de réseaux gérés !

Fin !

# Remerciements

---

- ➔ Olivier Fourmaux, UPMC / LIP6
- ➔ Yesid Jarma, UPMC
- ➔ Rémi Gacogne, Aquaray