

# Rapport TP1 : Simulation de Routage Inter-domaine avec BGP



**MASTER 2 NETWORKING AND DEVOPS**

**NEVA**

**Stefano Secci**

**Créé par:**

Alaoui Belghiti Hanaa.  
Nasreddine Hammoudi.  
Amadou Samoura.  
Abdelali Sekhri.

## Sommaire

### Introduction

- Objectif du TP
- Présentation de la topologie ENST (AS1712)

### Traceroutes et Analyse de Chemins AS

- 2.1 Traceroute vers [www.elet.polimi.it](http://www.elet.polimi.it)
  - Chemin AS suivi
- 2.2 Traceroute vers [www.renater.fr](http://www.renater.fr)
  - Chemin AS suivi

### Simulation de Réseau avec C-BGP

- 3.1 Configuration et Commandes de Vérification
- 3.2 Explication du Filtrage BGP
  - Configuration des pairs (88.160.1.1 et 134.157.1.1)
- 3.3 Préférences Locales pour le Trafic Sortant
  - Configuration des valeurs de préférence locale
- 3.4 Préparation de la Route Entrante
  - Configuration du prépenage de l'AS

### Analyse des Temps de Réponse du Traceroute vers [www.enst.fr](http://www.enst.fr)

- Influence des filtres AS sur le temps de réponse

### Table de Routage et Impact des Filtres

- Résultats des commandes de routage

### Simulation de Panne entre AS

- Impact de la panne sur le routage

### Relations entre FAI (Peering/Transit)

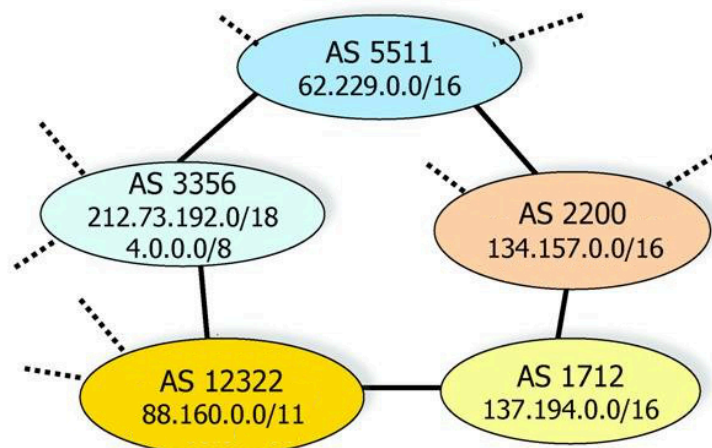
- 7.1 Analyse du chemin des paquets IP de l'AS-10 vers l'AS-1
- 7.2 Analyse du chemin des paquets IP de l'AS-1 vers l'AS-10

# I. Multi-homed network: exemple de ENST

## 1)Objectif :

Ce TP a pour but de simuler et comprendre les mécanismes de routage inter-domaine en utilisant le protocole BGP. Nous utilisons l'outil C-BGP pour simuler l'échange de routes entre différents Autonomous Systems (AS) et analyser les décisions de routage basées sur les configurations de filtres et préférences de chemin.

We study the configuration of the ENST (AS1712) showed in Figure 1:



Nous étudions la configuration de l'ENST (AS 1712) montrée dans la Figure 1 : Dans cette topologie, Level 3 (AS3356), FT Opentransit (AS5511), Renater (AS2200) et Free Proxad (AS12322) apparaissent.

**2)Question 1.1:** Nous rappelons qu'un traceroute fournit la liste des routeurs BGP par lesquels les paquets passent pour atteindre une destination donnée. En se référant à la topologie ci-dessus, commentez les résultats des traceroutes suivants, en expliquant uniquement le chemin AS suivi (utilisez [7] si nécessaire) :

### 1. Traceroute 1: [traceroute www.elet.polimi.it](http://www.elet.polimi.it)

Pour comprendre la configuration du réseau de l'ENST, concentrons-nous sur la façon dont la connexion a été initiée :

Elle a commencé avec la commande ``secci@alhambra.enst.fr:~$ traceroute www.elet.polimi.it``, probablement exécutée depuis un ordinateur au sein du réseau de l'ENST. Cette première étape se déroule au sein de **Renater (AS 1712)**.

Pour les premiers sauts du traceroute (1, 2 et 3), le paquet de données transite à travers des routeurs locaux de l'infrastructure de Renater, cherchant le meilleur chemin vers sa destination.

Au quatrième saut, le paquet atteint ``gw-free-th2 (137.194.4.2)``, qui sert de routeur passerelle. Après ce point, le réseau de Renater transfère le trafic à un fournisseur d'accès

Internet commercial, **Free Proxad (AS12322)**, comme indiqué au saut 5 ([th2-crs16-1-be1000.intf.routers.proxad.net](http://th2-crs16-1-be1000.intf.routers.proxad.net)).

Dans la sortie du traceroute, une séquence de `\*\*\*` apparaît, indiquant qu'un routeur n'a pas répondu à cette requête spécifique, mais cela est dû au fait que le routeur priorise d'autres tâches.

Malgré l'absence de réponse, ce routeur continue à transmettre le trafic vers le saut suivant, qui, dans ce cas, est **the-3-2.car1.Paris1.Level3.net (AS 3356)**. Cette transition montre que le trafic a été transféré au réseau **Level 3 (AS 3356)**, qui est un grand fournisseur de dorsale Internet.

Les derniers sauts (9, 10 et 11) impliquent **rt-rm2-rt-mi2.mi2.garr.net**, qui font partie du réseau GARR.

Les dernières étapes (12, 13, 14 et 15) indiquent que le paquet a atteint le réseau souhaité, qui le livre ensuite à sa destination finale à **web0.elet.polimi.it**.

### Sommaire:

Le chemin AS suivi était le suivant :

**AS Path:** AS 1712 (Renater) → AS 12322 (Free Proxad) → AS 3356 (Level 3) → GARR Network.

## 2. Traceroute 2: [traceroute www.renater.fr](http://www.renater.fr)

Les résultats du traceroute fournissent des informations sur le chemin suivi par les paquets lorsqu'ils tentent d'atteindre la destination [www.renater.fr](http://www.renater.fr) (134.157.159.10) :

1. **Point de départ - AS 1712 (Renater)** : Le traceroute commence à AS 1712 (Renater), ce qui indique que les paquets initiaux proviennent d'un poste de travail situé dans cet AS. Les premiers sauts montrent des connexions réussies à l'intérieur de cet AS, confirmant son bon fonctionnement et sa capacité à transmettre les paquets efficacement.
2. **Transition vers AS12322 (Free Proxad)** : Les paquets atteignent ensuite AS 12322 (Free Proxad) au saut 6. Cette transition indique que les paquets sont routés du réseau Renater vers celui d'un autre fournisseur. Cependant, la réponse de ce saut est manquante, probablement en raison de certains problèmes.
3. **Sauts suivants - AS 3356 (Level 3 Communications)** : Les paquets entrent ensuite dans AS 3356 (Level 3 Communications) au saut 7. Le routage réussi à travers cet AS montre que les paquets continuent de se diriger vers la destination, bien que les réponses de certains routeurs manquent toujours.
4. **AS 5511 (FT Opentransit)** : Après Level 3, le routeur atteint [france.telecom-level3-10ge.Paris1.Level3.net](http://france.telecom-level3-10ge.Paris1.Level3.net), ce qui suggère une connexion entre Level 3 (AS 3356) et FT Opentransit (AS 5511). Cette connexion fait partie du peering BGP entre ces réseaux.

5. **Destination - AS 2200 (Renater)** : La destination finale se trouve dans AS 2200, correspondant à l'adresse IP 134.157.159.10. Cependant, le traceroute n'atteint pas cette destination avec succès, comme l'indiquent plusieurs astérisques (\*\*\*) dans les derniers sauts. Cela suggère que les paquets sont soit bloqués, soit que la destination n'est pas configurée pour répondre aux requêtes traceroute.

### **Sommaire:**

Le chemin AS suivi était le suivant :

**AS Path:** AS 1712 (Renater) → AS 12322 (Free Proxad) → AS 3356 (Level 3) → AS 5511 (FT Opentransit) → (Potentially Renater, AS 2200).

## II) Simulation de réseau ENST avec CBGP

Nous avons téléchargé et installé C-BGP depuis le site officiel.

Le fichier enst.cli a été utilisé pour charger la topologie de l'ENST (AS1712) et les routes statiques nécessaires à la simulation. La commande `cbgp -i` a été utilisée pour entrer dans le mode interactif de C-BGP, et le fichier `enst.cli` a été inclus avec la commande :

**cbgp> include enst.cli**

```
cbgp> init.  
cbgp> include enst.cli  
*** ENST AS connectivity ***
```

### Commandes de vérification :

Pour observer les voisins BGP :

`bgp router x.x.x.x show peers`

```
stats  
cbgp> bgp router 134.157.1.1 show peers  
62.229.1.1 AS5511 ESTABLISHED 62.229.1.1 snd-seq:3 rcv-seq:5  
137.194.1.1 AS1712 ESTABLISHED 137.194.1.1 snd-seq:6 rcv-seq:2  
cbgp> bgp router 137.194.1.1 show peers  
88.160.1.1 AS12322 ESTABLISHED 88.160.1.1 snd-seq:2 rcv-seq:6  
134.157.1.1 AS2200 ESTABLISHED 134.157.1.1 snd-seq:2 rcv-seq:6  
cbgp> bgp router 88.160.1.1 show peers  
4.1.1.1 AS3356 ESTABLISHED 4.1.1.1 snd-seq:3 rcv-seq:5  
137.194.1.1 AS1712 ESTABLISHED 137.194.1.1 snd-seq:6 rcv-seq:2  
cbgp>
```

Pour visualiser les tables de routage :

`bgp router x.x.x.x show rib *`

```
stats  
cbgp> bgp router 88.160.1.1 show rib *  
*> 4.0.0.0/8 4.1.1.1 0 4294967295 3356 i  
*> 62.229.0.0/16 4.1.1.1 0 4294967295 3356 5511 i  
i> 88.160.0.0/11 88.160.1.1 0 0 null i  
*> 134.157.0.0/16 4.1.1.1 0 4294967295 3356 5511 2200 i  
*> 137.194.0.0/16 137.194.1.1 0 4294967295 1712 i  
*> 212.73.192.0/18 4.1.1.1 0 4294967295 3356 i  
cbgp> bgp router 137.194.1.1 show rib *  
*> 4.0.0.0/8 88.160.1.1 0 4294967295 12322 3356 i  
*> 62.229.0.0/16 134.157.1.1 0 4294967295 2200 5511 i  
*> 88.160.0.0/11 88.160.1.1 0 4294967295 12322 i  
*> 134.157.0.0/16 134.157.1.1 0 4294967295 2200 i  
i> 137.194.0.0/16 137.194.1.1 0 0 null i  
*> 212.73.192.0/18 88.160.1.1 0 4294967295 12322 3356 i  
cbgp>
```

Pour observer les routes reçues d'un voisin particulier :

`bgp router x.x.x.x show adj-rib in y.y.y.y *`

```

cbgp> bgp router 137.194.1.1 show adj-rib in 88.160.1.1 *
*> 4.0.0.0/8      88.160.1.1      0      4294967295      12322 3356      i
* 62.229.0.0/16   88.160.1.1      0      4294967295      12322 3356 5511 i
*> 88.160.0.0/11   88.160.1.1      0      4294967295      12322 i
* 134.157.0.0/16   88.160.1.1      0      4294967295      12322 3356 5511 2200 i
*> 212.73.192.0/18 88.160.1.1      0      4294967295      12322 3356      i
cbgp>

```

1)Question 1.2): expliquer le fonctionnement du code:

```

peer 88.160.1.1
  filter in
  add-rule
    match any
    action "community add 1"
  exit
  filter out
  add-rule
    match "community is 1"
    action deny
  exit
exit
peer 134.157.1.1
  filter in
  add-rule
    match any
    action "community add 1"
  exit
  filter out
  add-rule
    match "community is 1"
    action deny
  exit
exit

```

Cette partie du code implique la configuration des règles de filtrage BGP (Border Gateway Protocol) sur deux homologues (88.160.1.1 et 134.157.1.1). afin de contrôler les routes acceptées (entrantes) et les routes envoyées (sortantes).

#### Peer 88.160.1.1:

##### Filter in:

```

peer 88.160.1.1
  filter in
  add-rule
    match any
    action "community add 1"
  exit
exit

```

Il s'agit d'un filtre entrant appliqué aux itinéraires reçus du pair **88.160.1.1**.

La règle indique de faire correspondre n'importe quel itinéraire et d'ajouter la communauté 1 à tous les itinéraires reçus de ce pair.

##### Filter out:

```
filter out
  add-rule
    match "community is 1"
    action deny
  exit
exit
```

Il s'agit d'un filtre sortant appliqué aux routes annoncées sur le pair **88.160.1.1** et qui correspond aux routes avec la **community 1** (précédemment ajoutée dans le filtre entrant). L'action consiste à refuser ces routes, ce qui signifie qu'elles ne seront pas annoncées au pair **88.160.1.1**.

**Peer 134.157.1.1:**

**Filter in:**

```
peer 134.157.1.1
  filter in
    add-rule
      match any
      action "community add 1"
    exit
  exit
```

Il s'agit du filtre entrant pour les itinéraires reçus du peer 134.157.1.1. La règle correspond à n'importe quel itinéraire et ajoutez la communauté 1 à ces itinéraires.

**Filter out:**

```
filter out
  add-rule
    match "community is 1"
    action deny
  exit
exit
```

Il s'agit du filtre sortant pour les itinéraires annoncés sur le pair 134.157.1.1. Il correspond à tout itinéraire avec la communauté 1 et refuse de publier ces itinéraires sur ce pair.

**Pourquoi ces filtres? conclusion:**

Si une route provient de **88.160.1.1**, la renvoyer à 88.160.1.1 créerait une boucle, dans laquelle le routeur continue à envoyer les mêmes routes dans les deux sens, ce qui serait inefficace ou problématique.

Ce filtre garantit qu'une fois qu'une route est apprise à partir d'un peer, elle n'est pas renvoyée à ce même peer. -----> éviter les cycles ou les boucles dans les chemins de routage.



## 2)Question 1.3):

Pour garantir que le routeur **AS1712** envoie le trafic Internet sortant via Proxad (**AS12322**) comme chemin principal et ne bascule vers **Renater (AS2200)** que si la liaison avec Proxad échoue, je dois modifier les valeurs de préférence locale attribuées aux routes reçues de chacun de ces peers.

### Explication :

Proxad comme principal : pour que le routeur choisisse **Proxad (AS12322)** comme route principale pour le trafic sortant, je dois augmenter la valeur de préférence locale pour les routes reçues de Proxad. Une préférence locale plus élevée permet au routeur de donner la priorité à ces routes lors de la sélection du chemin pour le trafic sortant.

Renater comme sauvegarde : pour faire de **Renater (AS2200)** un chemin de sauvegarde, je définirai une valeur de préférence locale inférieure pour les routes reçues de Renater. Cela signifie que le routeur n'utilisera ce chemin que si la route principale via Proxad devient indisponible (par exemple, si la liaison échoue).

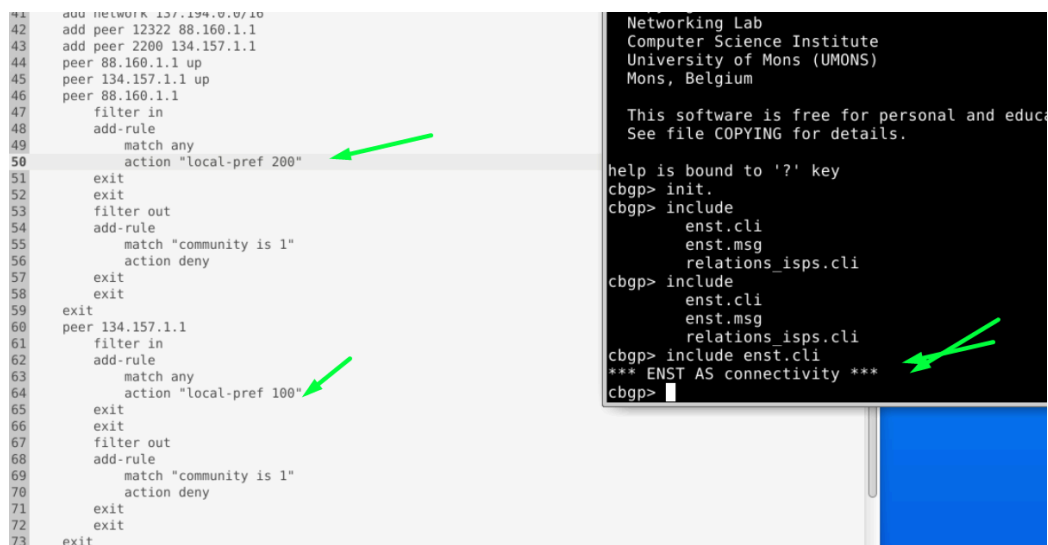
### Modifications nécessaires :

**J'ai défini une préférence locale de 200 pour Proxad** : cela permet au routeur de privilégier Proxad lorsque les deux chemins sont disponibles.

**J'ai défini une préférence locale de 100 pour Renater** : cela garantit que le routeur ne choisira Renater qu'en cas de défaillance de la liaison Proxad, ce qui en fait une option de secours.

En ajustant les valeurs de préférence locale de cette manière, le routeur tentera d'abord d'utiliser Proxad pour le trafic sortant et basculera automatiquement vers Renater uniquement si la connexion Proxad est interrompue.

comme vous voyez ici:



```
42 add peer 12322 88.160.1.1
43 add peer 2200 134.157.1.1
44 peer 88.160.1.1 up
45 peer 134.157.1.1 up
46 peer 88.160.1.1
47 filter in
48 add-rule
49 match any
50 action "local-pref 200"
51 exit
52 exit
53 filter out
54 add-rule
55 match "community is 1"
56 action deny
57 exit
58 exit
59 exit
60 peer 134.157.1.1
61 filter in
62 add-rule
63 match any
64 action "local-pref 100"
65 exit
66 exit
67 filter out
68 add-rule
69 match "community is 1"
70 action deny
71 exit
72 exit
73 exit
```

```
Networking Lab
Computer Science Institute
University of Mons (UMONS)
Mons, Belgium

This software is free for personal and educational use.
See file COPYING for details.

help is bound to '?' key
cbgp> init.
cbgp> include
  enst.cli
  enst.msg
  relations_isps.cli
cbgp> include
  enst.cli
  enst.msg
  relations_isps.cli
cbgp> include enst.cli
*** ENST AS connectivity ***
cbgp>
```

**Résultat:**

```

K  <prefix|address|*>
cbgp> bgp router 137.194.1.1 show rib *
*> 4.0.0.0/8      88.160.1.1      200      4294967295      12322 3356      i
*> 62.229.0.0/16  88.160.1.1      200      4294967295      12322 3356 5511 i
*> 88.160.0.0/11  88.160.1.1      200      4294967295      12322 i
*> 134.157.0.0/16 134.157.1.1     100      4294967295      2200 i
i> 137.194.0.0/16 137.194.1.1     0        0        null      i
*> 212.73.192.0/18 88.160.1.1      200      4294967295      12322 3356      i
cbgp>

```

**3)Question 1.4:** Configurer AS1712 afin d'acheminer le trafic entrant par l'AS12322 et non par l'AS2200.

Afin d'influencer le routage du trafic entrant pour **AS1712**, j'ai mis en œuvre le prépenage de l'AS dans la configuration pour la connexion de pair avec Renater (**AS2200**).

Plus précisément, j'ai appliqué la commande action "as-path-prepend 1712 1712" dans le filtre entrant pour les routes reçues d'**AS2200**. Cela ajoute efficacement **AS1712** deux fois au chemin AS, le faisant apparaître plus long par rapport aux routes provenant de Proxad (**AS12322**), qui n'ont pas cette information supplémentaire sur le chemin AS.

En conséquence, cette configuration garantit que, lorsque les deux routes sont disponibles, le BGP préférera le chemin AS plus court de Proxad, orientant ainsi le trafic entrant principalement à travers **AS12322**. De plus, j'ai maintenu une préférence locale élevée pour les routes de Proxad, renforçant davantage leur priorité dans le processus de décision de routage.

Cette approche assure qu'**AS1712** dirigera le trafic entrant à travers AS12322 dans des conditions normales, ne basculant vers **AS2200** qu'en cas de défaillance de la liaison avec Proxad.

Voici le screen de config:

```

58      exit
59      exit
60      peer 134.157.1.1
61          filter in
62          add-rule
63              match any
64              action "local-pref 100"
65          exit
66      exit
67      filter out
68      add-rule
69          match "community is 1"
70          action "as-path prepend 100"
71          action deny
72      exit
73      exit
74      exit

```



**Question 1.5):**

Dans le traceroute vers [www.enst.fr](http://www.enst.fr) (137.194.2.39), le filtrage appliqué par les systèmes autonomes (AS) est essentiel pour comprendre les temps de réponse.

1. **Sauts 1 à 4 :** Les premiers sauts montrent des temps de réponse très bas (moins de 1 ms), car les paquets circulent au sein du même réseau local, sans filtrage significatif.

```

"1 131.175.126.189 (131.175.126.189) 0.464 ms 0.356 ms 0.349 ms 2
131.175.174.185 (131.175.174.185) 0.399 ms 0.313 ms 0.303 ms 3

```

131.175.174.222 (131.175.174.222) 0.423 ms 0.532 ms 0.313 ms 4

131.175.174.122 (131.175.174.122) 0.345 ms 0.327 ms 0.326 ms"

Nous sommes toujours dans le même réseau.

2. **Saut 5 (GARR)** : À ce stade, le trafic reste rapide, mais le filtrage commence à se faire sentir, bien que les temps de réponse restent bas.
3. **Saut 6 (Géant2)** : Les paquets continuent de circuler efficacement, mais l'absence de filtrage strict commence à changer.
4. **Saut 8 (Géant2)** : À partir de ce saut, le temps de réponse augmente à 7.894 ms. Cela indique que le réseau applique des filtres, ralentissant le traitement des paquets en vérifiant leur conformité aux règles de sécurité.
5. **Saut 9 (Renater)** : Les temps de réponse montent à 16.839 ms, suggérant que Renater applique des contrôles supplémentaires sur le trafic, augmentant ainsi les délais.
6. **Sauts 10 à 12** : Les temps de réponse restent élevés, montrant que les politiques de filtrage de Renater continuent d'influencer le trafic. Chaque saut impose des vérifications qui ralentissent le passage des paquets.
7. **Sauts 13 à 16 (Destination finale)** : En atteignant [www.enst.fr](http://www.enst.fr), l'impact du filtrage est évident. Bien que le trafic ait été initialement rapide, chaque AS a imposé des politiques qui ont affecté le temps de transit global.

En conclusion, le traceroute vers [www.enst.fr](http://www.enst.fr) révèle l'influence des filtres appliqués par les systèmes autonomes (AS) sur le temps de réponse des paquets. Les paquets passent du chemin suivant :

**AS1712 → AS12322 (Proxad) → AS2200 (Renater) → AS193 (GARR) → AS62 (GÉANT) → AS193 (GARR) → AS2200 (Renater) → AS195 (Jussieu) → AS137194 (enst.fr).**

On observe que le temps de réponse est de 10 ms entre GARR et GÉANT, mais augmente à partir du hop 8. Cela indique que des filtres externes appliqués par d'autres AS peuvent affecter le routage des paquets.

#### Question 1.6):

Après avoir exécuté la commande **bgp router x.x.x.x show rib \***, nous obtenons les résultats suivants :

- Dans la table de routage du **routeur 137.194.1.1**, les filtres appliqués montrent que la préférence est systématiquement de passer par **88.160...**

```
cbgp> bgp router 137.194.1.1 show rib *
*> 4.0.0.0/8      88.160.1.1      200      4294967295      12322 3356      i
*> 62.229.0.0/16  88.160.1.1      200      4294967295      12322 3356 5511i
*> 88.160.0.0/11  88.160.1.1      200      4294967295      12322      i
*> 134.157.0.0/16 88.160.1.1      200      4294967295      12322 3356 5511 2200      i
i> 137.194.0.0/16 137.194.1.1      0         0         null      i
*> 212.73.192.0/18 88.160.1.1      200      4294967295      12322 3356      i
```

- Pour la table de routage du **routeur 134.157.1.1**, on observe que tous les paquets transitent par **62.229.1.1**, à l'exception de ceux destinés à **137.194.1.1**.

```

cbgp> bgp router 134.157.1.1 show rib *
*> 4.0.0.0/8      62.229.1.1      0      4294967295      5511 3356      i
*> 62.229.0.0/16   62.229.1.1      0      4294967295      5511      i
*> 88.160.0.0/11   62.229.1.1      0      4294967295      5511 3356 12322i
i> 134.157.0.0/16   134.157.1.1     0      0      null      i
*> 137.194.0.0/16   137.194.1.1     100    4294967295      1712      i
*> 212.73.192.0/18 62.229.1.1      0      4294967295      5511 3356      i

```

- En ce qui concerne la table de routage du routeur **88.160...**, bien que des filtres aient été appliqués, pour envoyer un paquet vers l'AS 2200, la route ne passe pas par l'AS 1712, même si ce dernier représente le chemin le plus court.

```

cbgp> bgp router 88.160.1.1 show rib *
*> 4.0.0.0/8      4.1.1.1 0      4294967295      3356      i
*> 62.229.0.0/16   4.1.1.1 0      4294967295      3356 5511      i
i> 88.160.0.0/11   88.160.1.1     0      0      null      i
*> 134.157.0.0/16   4.1.1.1 0      4294967295      3356 5511 2200      i
*> 137.194.0.0/16   137.194.1.1     0      4294967295      1712      i
*> 212.73.192.0/18 4.1.1.1 0      4294967295      3356      i

```

- Enfin, les tables de routage des routeurs 62.229.1.1 et 4.1.1.1, aucune préférence n'est observée. Ils se basent simplement sur le critère des chemins les plus courts, car aucun filtre n'a été appliqué.

```

cbgp> bgp router 62.229.1.1 show rib *
*> 4.0.0.0/8      4.1.1.1 0      4294967295      3356      i
i> 62.229.0.0/16   62.229.1.1     0      0      null      i
*> 88.160.0.0/11   4.1.1.1 0      4294967295      3356 12322      i
*> 134.157.0.0/16   134.157.1.1     0      4294967295      2200      i
*> 137.194.0.0/16   134.157.1.1     0      4294967295      2200 1712      i
*> 212.73.192.0/18 4.1.1.1 0      4294967295      3356      i
cbgp>
cbgp> bgp router 4.1.1.1 show rib *
i> 4.0.0.0/8      4.1.1.1 0      0      null      i
*> 62.229.0.0/16   62.229.1.1     0      4294967295      5511      i
*> 88.160.0.0/11   88.160.1.1     0      4294967295      12322     i
*> 134.157.0.0/16   62.229.1.1     0      4294967295      5511 2200      i
*> 137.194.0.0/16   88.160.1.1     0      4294967295      12322 1712     i
i> 212.73.192.0/18 4.1.1.1 0      0      null      i

```

#### Question 1.7):

Après avoir simulé la panne entre l'AS 1712 et l'AS 12322 à l'aide de la commande **"bgp router 137.194.1.1 peer 88.160.1.1 down"**, suivie de la commande **"sim run"** pour appliquer ce changement :

- Les paquets provenant de **137.194.1.1** ne peuvent plus recevoir de routes de l'AS **12322**. Désormais, toutes les routes disponibles sont fournies par l'AS **2200**, comme l'indique le tableau ci-dessous.

```

cbgp> bgp router 137.194.1.1 peer 88.160.1.1 down
cbgp> sim run
cbgp> bgp router 137.194.1.1 show rib *
*> 4.0.0.0/8      134.157.1.1     100    4294967295      2200 5511 3356      i
*> 62.229.0.0/16   134.157.1.1     100    4294967295      2200 5511      i
*> 88.160.0.0/11   134.157.1.1     100    4294967295      2200 5511 3356 12322      i
*> 134.157.0.0/16   134.157.1.1     100    4294967295      2200      i
i> 137.194.0.0/16   137.194.1.1     0      0      null      i
*> 212.73.192.0/18 134.157.1.1     100    4294967295      2200 5511 3356      i

```

- Les paquets de **134.157.1.1** ne peuvent plus passer par l'AS **1712** pour atteindre l'AS **12322** après la panne. Par conséquent, tous les paquets de l'AS **2200** transitent par l'AS **5511**, sauf ceux destinés à l'AS **1712**, comme illustré dans le tableau ci-dessous.

```

cbgp> bgp router 134.157.1.1 show rib *
*> 4.0.0.0/8      62.229.1.1  0      4294967295      5511 3356      i
*> 62.229.0.0/16  62.229.1.1  0      4294967295      5511      i
*> 88.160.0.0/11  62.229.1.1  0      4294967295      5511 3356 12322 i
i> 134.157.0.0/16  134.157.1.1 0      0      null      i
*> 137.194.0.0/16  137.194.1.1 100    4294967295      1712      i
*> 212.73.192.0/18 62.229.1.1  0      4294967295      5511 3356      i

```

- Tous les paquets envoyés depuis **88.160.1.1** vers l'**AS 1712** passent désormais par l'**AS 3356**, comme le montre également le tableau ci-dessous.

```

cbgp> bgp router 88.160.1.1 show rib *
*> 4.0.0.0/8      4.1.1.1 0      4294967295      3356      i
*> 62.229.0.0/16  4.1.1.1 0      4294967295      3356 5511      i
i> 88.160.0.0/11  88.160.1.1 0      0      null      i
*> 134.157.0.0/16  4.1.1.1 0      4294967295      3356 5511 2200 i
*> 137.194.0.0/16  4.1.1.1 0      4294967295      3356 5511 2200 1712 i
*> 212.73.192.0/18 4.1.1.1 0      4294967295      3356      i

```

### III) Relationships between ISPs (peering/transit)

#### Question 2.1):

Analyse du chemin des paquets IP de l'AS-10 vers l'AS-1 :

Pour les paquets IP provenant de l'AS-10 vers l'AS-1, un seul chemin est préféré :

- **AS10-AS2-AS1.**

La raison pour laquelle ce chemin est choisi est qu'il y a trois propositions de routes avec la même **Local-pref**. En appliquant le second critère, qui est la longueur du chemin, la route la plus courte est AS10-AS2-AS1.

```
cbgp> bgp router 10.10.1.1 show rib 10.1.1.1
*> 10.1.0.0/16 10.2.1.1 10 4294967295 2 1 i
cbgp> bgp router 10.2.1.1 show rib 10.1.1.1
*> 10.1.0.0/16 10.1.1.1 100 4294967295 1 i
```

Analyse du chemin des paquets IP de l'AS-1 vers l'AS-10 :

Pour les paquets IP envoyés de l'AS-1 vers l'AS-10, deux chemins ont la même préférence :

- Le premier est **AS1-AS4-AS7-AS9-AS10.**
- Le second est **AS1-AS5-AS8-AS9-AS10.**

Pourquoi ces deux chemins sont-ils utilisés ?

Les routes proposées par l'**AS-2** ont une préférence locale basse, celles proposées par l'**AS-3** ont une **Local-pref** moyenne, et celles offertes par l'**AS-4** et l'**AS-5** ont une **Local-pref** élevée.

Puisque les chemins **AS4** et **AS5** ont la même **Local-pref** élevée, le critère suivant, la longueur du chemin, est appliqué. Les deux chemins étant de longueur égale, ils peuvent être utilisés de manière interchangeable.

De la même manière on fait le chemin inverse :

```
cbgp> bgp router 10.1.1.1 show rib 10.10.1.1
*> 10.10.0.0/16 10.4.1.1 100 4294967295 4 7 9 10
cbgp> bgp router 10.4.1.1 show rib 10.10.1.1
*> 10.10.0.0/16 10.7.1.1 100 4294967295 7 9 10 i
cbgp> bgp router 10.7.1.1 show rib 10.10.1.1
*> 10.10.0.0/16 10.9.1.3 100 4294967295 9 10 i
cbgp> bgp router 10.9.1.1 show rib 10.10.1.1
*> 10.10.0.0/16 10.10.1.1 100 4294967295 10 i
```

On peut aussi utiliser la commande suivante mais cela nous affiche tous les routeurs traversés par les paquets.

```
cbgp> net node 10.1.1.1 record-route 10.10.1.1
10.1.1.1 10.10.1.1 SUCCESS 6 10.1.1.1 10.4.1.1 10.7.1.1 10.9.1.3 10.9.1.1 10.10.1.1
```

### Question 2.2):

```
cbgp> net node 10.9.1.1 record-route 10.1.1.1
10.9.1.1      10.1.1.1      SUCCESS 5      10.9.1.1 10.9.1.3 10.7.1.1 10.4.1.1 10.1.1.1
cbgp> net node 10.9.1.2 record-route 10.1.1.1
10.9.1.2      10.1.1.1      SUCCESS 4      10.9.1.2 10.8.1.1 10.5.1.1 10.1.1.1
cbgp> net node 10.9.1.3 record-route 10.1.1.1
10.9.1.3      10.1.1.1      SUCCESS 4      10.9.1.3 10.7.1.1 10.4.1.1 10.1.1.1
```

```
71 net node 10.9.1.1 route add --oif=10.9.1.3 10.7.0.0/16 12
72 net node 10.9.1.2 route add --oif=10.9.1.1 10.10.0.0/16 11
```

#### Routeur 1 (10.9.1.1) :

- Le routeur reçoit deux chemins équivalents en **Local Preference** et **AS-PATH**. La décision se fait sur la base de la **métrique IGP**.
  - Chemin via **10.9.1.2 (AS8 → AS5 → AS1)** : Métrique = 12.
  - Chemin via **10.9.1.3 (AS7 → AS4 → AS1)** : Métrique = 11.
- Conclusion** : Le routeur 1 choisit donc le chemin passant par **AS7** (via 10.9.1.3), car la métrique est plus basse (11).

#### Routeur 2 (10.9.1.2) :

- Le routeur choisit le chemin avec le **AS-PATH** le plus court.
  - Chemin direct via **AS8** : **AS9 → AS8 → AS5 → AS1**.
- Conclusion** : Le routeur 2 choisit le chemin direct passant par **AS8** car c'est le chemin le plus court en termes de nombre d'AS.

#### Routeur 3 (10.9.1.3) :

- Le choix se fait en faveur des routes apprises via **eBGP** plutôt que **iBGP**.
  - Le chemin passant par **AS7** est appris via eBGP (prioritaire).
- Conclusion** : Le routeur 3 choisit le chemin passant par **AS7 → AS4 → AS1**.

### Question 3.1):

```
cbgp> bgp router 10.9.1.1 show adj-rib in * 10.1.0.0
*> 10.1.0.0/16 10.8.1.1 10 4294967295 8 5 1 i
* 10.1.0.0/16 10.7.1.1 10 4294967295 7 4 1 i
cbgp> bgp router 10.9.1.2 show adj-rib in * 10.1.0.0
*> 10.1.0.0/16 10.8.1.1 10 4294967295 8 5 1 i
* 10.1.0.0/16 10.7.1.1 10 4294967295 7 4 1 i
cbgp> bgp router 10.9.1.3 show adj-rib in * 10.1.0.0
*> 10.1.0.0/16 10.7.1.1 10 4294967295 7 4 1 i
* 10.1.0.0/16 10.8.1.1 10 4294967295 8 5 1 i
```

Cela montre que les routes via **AS7** et **AS8** sont disponibles, mais qu'il est nécessaire de configurer le **MED (Multiple Exit Discriminator)** pour influencer la sélection de route et forcer les paquets à passer par **AS7** plutôt que **AS8**.

```
cbgp> net node 10.9.1.1 route del 10.7.0.0/16 10.9.1.3
cbgp> net node 10.9.1.1 route del 10.8.0.0/16 10.9.1.2
cbgp>
cbgp> net node 10.9.1.1 route add --oif=10.9.1.3 10.7.0.0/16 50
cbgp> net node 10.9.1.1 route add --oif=10.9.1.2 10.8.0.0/16 200
cbgp>
cbgp> bgp router 10.9.1.1 show rib *
```

*> 10.1.0.0/16	10.8.1.1	10	4294967295	8 5 1	i
*> 10.2.0.0/16	10.8.1.1	10	4294967295	8 5 1 2	i
*> 10.3.0.0/16	10.8.1.1	10	4294967295	8 5 1 3	i
*> 10.4.0.0/16	10.7.1.1	10	4294967295	7 4	i
*> 10.5.0.0/16	10.8.1.1	10	4294967295	8 5	i
*> 10.6.0.0/16	10.8.1.1	10	4294967295	8 5 6	i
*> 10.7.0.0/16	10.7.1.1	10	4294967295	7	i
*> 10.8.0.0/16	10.8.1.1	10	4294967295	8	i
i> 10.9.0.0/16	10.9.1.1	0	0 null	i	
*> 10.10.0.0/16	10.10.1.1	100	4294967295	10	i

Pour répondre à cette question sans modifier les **local-preferences**, j'ai utilisé le **Multiple Exit Discriminator (MED)**. J'ai d'abord supprimé les routes existantes via **AS7** et **AS8**, puis j'ai ajouté de nouvelles routes avec un **MED faible** de **50** pour **10.9.1.3 (AS7)** et un **MED élevé** de **200** pour **10.9.1.2 (AS8)**. Cette configuration garantit que les paquets envoyés de **10.9.1.1** vers **AS1** passent par **AS7** plutôt que par **AS8**.



## **Conclusion TP:**

Ce travail pratique nous a permis d'analyser la topologie du réseau ENST (AS1712) et d'étudier les chemins de routage empruntés par les paquets. Grâce aux traceroutes, nous avons identifié les routes AS et observé l'impact des filtres BGP sur le routage.

La simulation avec C-BGP a montré l'importance des préférences locales et du filtrage des routes pour optimiser le routage. En analysant les temps de réponse, nous avons constaté comment ces filtres influencent les performances du réseau.

Enfin, l'étude des pannes a mis en évidence la résilience du système, soulignant l'importance d'une gestion proactive pour garantir la disponibilité et la performance des services.