

# Intégration Voix / Données

## TP 2 - Differentiated Services

### Objectifs du TP

- Comprendre le fonctionnement d'une architecture à Qualité de Service (caractérisation des flux, signalisation et réservation de flux), différencier les types de réservations de flux, implanter les solutions techniques découvertes en cours (classification, marquage, polissage, prévention de congestion et ordonnancement de paquets).
- Mettre en place une solution de Qualité de Service basé sur Differentiated Services.
- Observer les performances du réseau lors de transferts (vidéo, audio et données) suivant la QoS mise en place.
- Comprendre les paramètres influant sur la QoS et savoir les estimer.

### Configuration

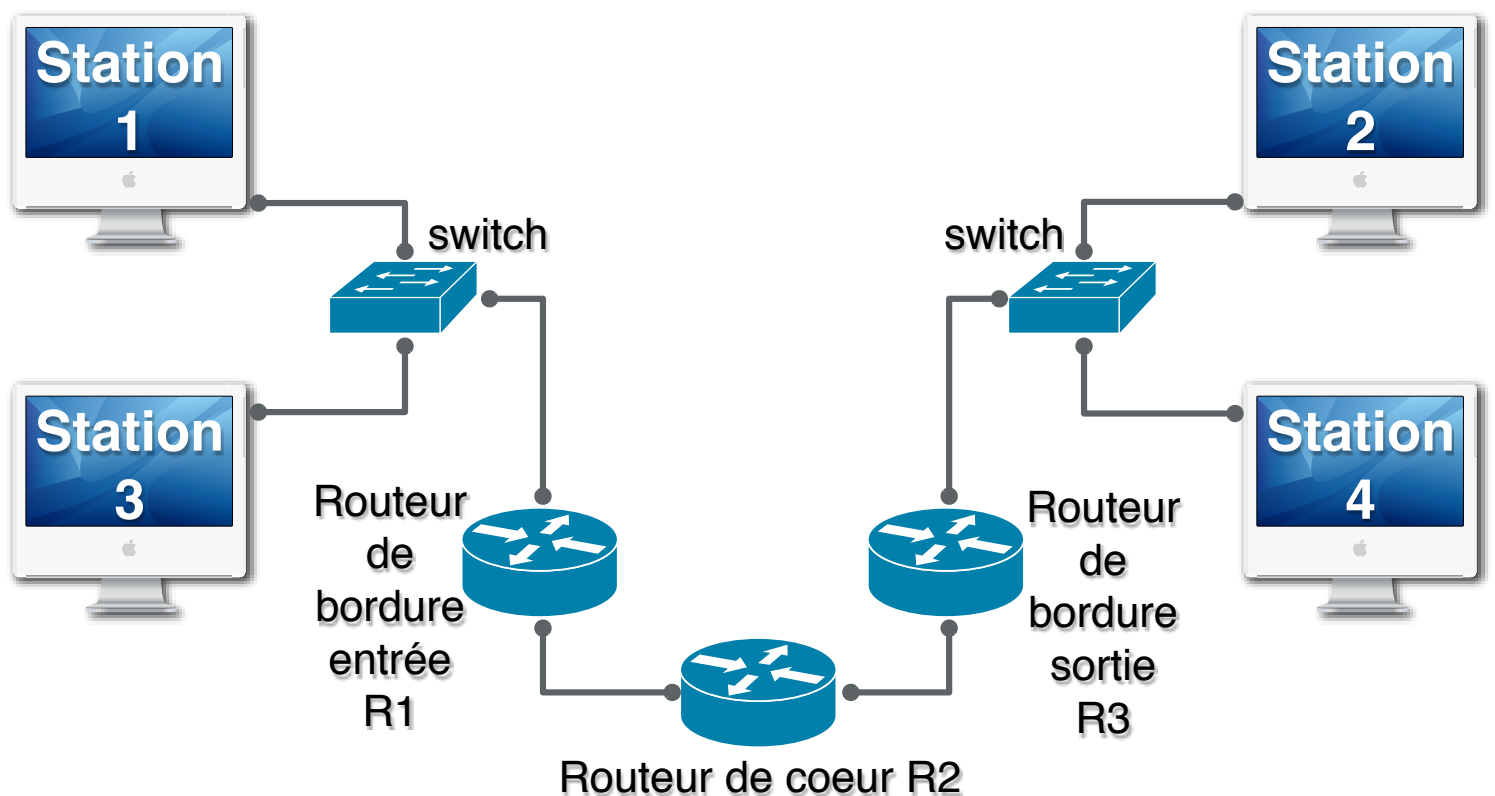
#### Configuration matérielle

- 4 PC stations par groupe
- 4 routeurs interconnectés par groupe

#### Configuration logicielle

- Wireshark
- VLC, FTP
- Iperf
- Utilitaires réseaux habituels (ifconfig, ping...)

### Idée de Topologie d'étude



## I. Mise en place

Le fonctionnement de *Differentiated Services* suppose une ingénierie de l'architecture réseau du domaine considéré. Cette étape doit renseigner l'administrateur sur les choix à faire pour la configuration des différents équipements. Contrairement à l'approche *Integrated Services*, *Differentiated Services* se base sur l'ingénierie de trafic. Cela implique que la (ou les) Qualité(s) de Service(s) délivrée(s) par chaque équipement est configurée par l'administrateur (ou l'architecte) et non par une application cliente.

Dans le cadre des TP, nous simulerons l'existence d'applications clientes (et serveur) de type FTP et Vidéo en *streaming*.

Le but de ces TP est de définir 4 classes de trafic : or, argent, bronze et be (best effort).

La mise en place de *Differentiated Services* sur des équipements de type CISCO obéit aux étapes suivantes :

1. Classification de paquet ;
2. Marquage de paquets ;
3. Contrôle d'accès et polissage ;
4. Ordonnancement de paquets ;
5. Gestion de congestion.

## II. Classification

La philosophie de fonctionnement de DiffServ suppose une classification systématique des paquets par chaque équipement d'interconnexion de niveau 3. Cette classification peut par contre être différente (mais amener aux mêmes résultats...) selon que l'équipement considéré soit un équipement de bordure du domaine ou un équipement de coeur.

La classification DiffServ CISCO repose sur l'établissement de classes (*class-map*). Une *class-map* permet de définir le trafic à analyser. Chaque *class-map* peut définir un ou plusieurs critères de sélection de trafic.

### II.a) Listes d'accès

Dans ce TP, nous allons utiliser les listes d'accès (ACL) afin de caractériser le trafic entrant :

```
[config] access-list numero_liste [deny|permit] protocole source masque_source
                        destination masque_destination
```

où *numero\_liste* représente le numéro de la classification (de 0 à 99 pour une liste standard et de 100 à 199 pour une liste étendue); *deny* et *permit* indiquent si le trafic doit être rejeté ou accepté, *protocole* précise le type de protocole employé (*ip*, *udp*, *tcp* ou tout autre protocole connu du routeur). Selon que le *protocole* soit *ip*, *udp*, *tcp*, on pourra aussi filtrer les paquets selon les ports utilisés (ports source et ports destination). Voir l'aide CISCO... *source* et *destination* les adresses réseau ou hôte considérées et *masque\_source*, *masque\_destination* sont les masques binaires permettant d'évaluer l'étendue des adresses *source* et *destination*. **Attention, ces masques ne sont pas des masques de sous-réseau !!!**

Exemple d'emploi des masques : @réseau : 172.16.0.0. On souhaite désigner les adresses 172.16.0.x. Le masque sera 0.0.0.255.

☒ **Note** : La classification CISCO peut utiliser d'autres moyens de classier les paquet et notamment NBAR qui permet de filtrer les paquets selon leur contenu (un peu à la manière des *firewalls statefull*).

\* Exemple :

```
access-list 101 permit tcp 172.16.0.0 0.0.0.255 192.168.0.0 0.0.0.255 eq www définit une liste d'accès étendue de numéro
101 qui autorise le trafic tcp en partance des hôtes du réseau 172.16.0.0/24 et à destination des stations du réseau 192.168.0.0/24 pour le
trafic www.
```

### II.b) Classes de trafic

Nous devons maintenant définir les *class-map* associées aux *access-list* définies :

```
[config] class-map      match-all nom_classe
                        match access-group numero_liste
```

où *nom\_classe* représente le nom de la classe de trafic DiffServ et *numero\_liste* est la liste d'accès à appliquer.

\* Exemple :

```
class-map match-all Ma_classe match access-group 101 défini une classe de trafic nommée Ma_classe et qui utilise la liste d'accès
numéro 101.
```

### III. Marquage des paquets

Une fois les paquets classés, il est nécessaire de les marquer. Cette opération se déroule en deux-temps. Il faut en premier lieu définir une politique de marquage puis ensuite affecter cette politique aux interfaces d'entrées concernées.

#### III.a) Politique de marquage

Une politique de marquage sert à affecter la même valeur de DSCP à tous les paquets d'une classe. Nous utilisons la définition des politiques de services afin d'appliquer un traitement à nos paquets :

```
[config] policy-map    nom_politique
                    class nom_classe set ip dscp valeur_dscp
```

où `nom_politique` représente le nom de la politique de service; `nom_classe` est le nom de la classe de filtrage choisie et `valeur_dscp` est la valeur du champ `dscp` (*DiffServ Code Point*) à positionner.



**Attention** : les valeurs du DSCP peuvent varier entre 0 et 63. Les mnémoniques suivantes sont utilisables : `af11`, `af12`, `af13`, `af21`, `af22`, `af23`, `af31`, `af32`, `af33`, `af41`, `af42`, `af43`, `cs1`, `cs2`, `cs3`, `cs4`, `cs5`, `cs7`, `ef`, `rsvp`.

\* Exemple :

```
policy-map politik class Ma_classe set ip dscp 40
```

définit une politique de service nommée `politik` et qui positionne le DSCP à 40 pour chaque paquet de la classe de filtrage `Ma_classe`.

#### III.b) Affectation de la politique aux interfaces

La définition d'une politique de service ne précise pas quels sont les paquets qui lui seront soumis. Pour cela, une fois les classes et les politiques définies, il faut lier ces politiques aux interfaces d'entrées. Cette étape se déroule de la manière suivante :

```
[config-if]    service-policy input nom_politique
```

où `nom_politique` représente le nom de la politique de service et `nom_classe` est le nom de la classe de filtrage choisie.

\* Exemple :

```
interface E0/0 <- service-policy input politik
```

applique la politique de service nommée `politik` aux paquets entrants par l'interface `E0/0`.

### IV. Contrôle d'accès et polissage

Le marquage des paquets ne protège pas le router et le réseau de flux ne respectant pas leur contrat de QoS. Pour se prémunir de cela, il convient de déployer contrôle d'accès (*policing*) ET polissage (*shaping*) dès l'entrée des paquets. Ces deux traitements se définissent de la même manière que le marquage des paquets (politique et affectation aux interfaces). Nous allons donc modifier la configuration précédente et faire dépendre le marquage des paquets par leur respect de la QoS qui leur a été affectée (SLA).

#### IV.a) Politique de service

➡ Contrôle d'accès simple

```
[config-pmap-c]    police debit [burst-ok] [burst-max] conform-action action1
                    exceed-action action2 [violate-action action3]
```

où `action1`, `action2` et `action3` sont les actions possibles selon que le paquet est indiqué comme conforme (`conform-action`), en dépassement (`exceed-action`) ou en violation (`violate-action`) du contrat de QoS.

Le contrôle d'accès se base sur la définition d'un *Token Bucket* dont les paramètres sont `debit`, `burst-ok` et `burst-max`.

Dans ce TP, nous nous limiterons aux types d'action suivants :

- `drop` : indique que le paquet doit être détruit ;
- `set-dscp-transmit valeur-dscp` : permet de marquer (ou de re-marquer) le DSCP du paquet avec `valeur-dscp` ;
- `transmit` : indique de transmettre le paquet sans changement.

\* Exemple :

```
police 10000 conform-action transmit exceed-action set-dscp-transmit 0 violate-action drop
```

définit une politique de service fixant le débit nominal à 10 000 b/s. Les paquets respectant ce contrat de QoS seront transmis sans changement, ceux excédant le contrat seront remarqués vers la discipline Meilleur Effort. Enfin, ceux dépassant fortement le contrat seront supprimés.

## ➡ Contrôle d'accès à deux taux

Ce contrôle d'accès diffère de précédent de par son utilisation de deux taux différents pour juger d'une conformité de trafic. Le premier taux est le taux instantané (CIR) et le second taux est le taux de crête (PIR). Le CIR dénote le débit qui doit être garanti dans la durée. Le PIR indique lui une possibilité passagère de dépasser le CIR. En ce sens, le contrôle d'accès à deux taux est plus précis que le contrôle d'accès simple. Le contrôle d'accès à deux taux se base sur l'utilisation d'un *Token Bucket* par taux.

```
[config-pmap-c]      police cir debit-moyen [bc burst-moyen] pir debit-max [be burst-max]
                      conform-action action1 exceed-action action2 [violate-action action3]
```

où *debit-moyen* est le débit moyen souhaité (*cir*), *bc* indique le *burst-moyen*, *debit-max* (*pir*) est le débit de crête et *burst-max* la taille maximale d'une rafale admissible (*be*). Les actions possibles *action1*, *action2* et *action3* sont identiques au contrôle d'accès simple.



**Note** : Il est aussi possible d'utiliser le mot clé *percent* après *cir* ou *pir* pour exprimer un pourcentage de bande passante auquel cas, *be* et *bc* seront exprimés en ms (mot clé aussi obligatoire). Exemple : *cir percent 10 bc 200 ms pir 12 be 200 ms*.

## IV.b) Polissage

Le polissage consiste à lisser le trafic sur une période de temps. Le but est de se rapprocher d'un modèle fluide (donc moins sujets aux rafales) et donc plus facilement prédictible. Dans l'architecture CISCO, plusieurs mécanismes de polissage ont été définis. Dans le cadre de ce TP, nous nous limiterons aux mécanismes *Class-Based* et GTS.

## ➡ Polissage par classe

Ce type de polissage permet de polir les trafics avec une approche par classe de service. Cela permet de polir tout le trafic entrant dans les différentes classes et s'adresse à toutes les interfaces désignées. CBWFQ (*Class-Based Weighted Fair Queuing*) est l'algorithme utilisé. Pour rappel, les mécanismes de polissage se basent pour la plupart sur le principe du seau à jetons (*Token Bucket*).

```
[config-pmap-c]      shape {average|peak} debit [burst-ok] [burst-max]
```

Où *average* indique le débit moyen de polissage, *peak* indiquant le débit de crête. Il est aussi possible de préciser la taille en bit des rafales conformes au trafic et celles le dépassant.

\* Exemple :

```
shape average 500000 définit une politique de polissage qui limite le trafic à 500 Kb/s.
```

Combiné à la commande *bandwidth*, il est possible d'indiquer une bande passante souhaitée pour la classe et de polir le trafic dépassant cette valeur.

\* Exemple :

```
bandwidth 256000 ↵ définit une limite de trafic à 256 Kb/s.
```

```
shape peak 500000 définit une politique de polissage qui autorise un débit jusqu'à 500 Kb/s si la bande passante du réseau le permet.
```

Le polissage par classe permet de définir un polissage général (*parent*) qui sera raffiné par différentes politiques de polissage (*child*) par l'utilisation de politiques hiérarchiques (voir documentation CISCO IOS QoS).

## ➡ GTS : Generic Traffic Shaping

Contrairement au polissage par classe, GTS fonctionne sur la base des interfaces, soit directement, soit par l'intermédiaire d'ACL. Le principe de fonctionnement se base sur un seau à jetons.

```
[config-if]      traffic-shape rate debit [burst-ok [burst-max]]
```

Où *rate* indique le débit moyen de polissage. Il est aussi possible de préciser la taille en bit des rafales conformes au trafic et celles le dépassant. Le débit moyen doit être compris entre 8 000 et 100 000 000 b/s.

## IV.c) Affectation de la politique aux interfaces

Comme les politiques ont été affectées aux interfaces à l'étape III, ici, elle est inutile.

## V. Ordonnancement de paquets

Nous allons étudier trois types de traitements applicables : WFQ (Weithed Fair Queuing), CBWFQ (Class Based WFQ) et Low-Latency Queuing.

### V.a) WFQ

WFQ assure un comportement équitable (« *fair* ») des différents flux de données en se basant sur des priorités. WFQ se configure pour chaque interface.

```
[config-if] fair-queue [congestive-discard-threshold [dynamic-queues [reservable-queues]]]
```

WFQ utilise une classification basée sur les en-têtes des paquets (adresses source et destination...).

### V.b) CBWFQ

Le choix du traitement effectué aux paquets d'une classe de service suppose la création de classes de trafic (class-map) classant les paquets uniquement selon leur DSCP et la création de politique de service (policy-map) appliquant le traitement de QoS souhaité. Bien que CBWFQ puisse être appliqué à plusieurs critères dépendant de la classification, nous nous limitons ici à leur application dans le cadre de DiffServ.

➔ Classes de trafic  
Cf. §II.b

➔ Politique de service  
Cf. §III.a

Pour chaque classe, il est possible de définir la quantité de bande passante ou le pourcentage de bande passante réservé :

```
[config-pmap-c] bandwidth {bp | percent pourcentage}
```


où **bp** exprime la bande passante en kb/s et **pourcentage** exprime le pourcentage de bande passante alloué. Attention, ces valeurs expriment la bande passante réservée et non la bande passante restante disponible.


La classe par défaut (principalement utilisée pour le trafic Meilleur Effort) et uniquement celle-ci peut être configurée avec WFQ (cf. V.a).

➔ Affectation de la politique aux interfaces  
Cf. §III.b

### V.c) LLQ

Low-Latency Queuing permet d'affecter une file de haute priorité à une classe de trafic. Cette classe de service est donc particulièrement adaptée au transport de la voix sur les réseaux.

 **Note** : Pour assurer un tel traitement, l'équipement assure un contrôle strict du contrat de QoS. Ainsi, tout paquet ne respectant pas son contrat sera détruit. Il ne peut donc être re-marqué vers une autre classe de trafic.

 **Note** : Même si LLQ peut être utilisé pour plusieurs classes de trafic, il est souhaitable de limiter son emploi au trafic correspondant au transport de la voix ou du trafic CBR (*Constant Bit Rate*).

La configuration LLQ est très similaire à celle de CBWFQ :


➔ Classes de trafic  
Cf. §II.b


➔ Politique de service  
Cf. §III.a

Pour chaque classe, il est possible de définir la quantité de bande passante ou le pourcentage de bande passante réservé :

```
[config-pmap-c] priority {bp | percent pourcentage} [taille-rafale]
```

où **bp** exprime la bande passante en kb/s et **pourcentage** exprime le pourcentage de bande passante alloué. Attention, ces valeurs expriment la bande passante réservée et non la bande passante restante disponible. **taille-rafale** indique la taille maximale admissible d'une rafale exprimée en octets (compris entre 32 et 2 000 000 octets).


 **Note** : Sauf mention contraire via le mot clé **max-reserved-bandwidth**, il n'est pas possible de réserver plus de 75% de la bande passante totale d'une interface. Combiné au respect strict du contrat de QoS, cela permet d'éviter que cette classe de trafic ne monopolise toute la bande passante d'une interface au détriment des autres classes.

 **Note** : LLQ ne peut être associé aux mots-clés **bandwidth**, **queue-limit** ou **random-detect** (voir §VI) dans la même classe de service.

➔ Affectation de la politique aux interfaces  
Cf. §III.b

## VI. Contrôle de congestion


Bien que les mécanismes précédents permettent un support de la QoS, les agrégats de trafic peuvent conduire à des congestion dans le réseau. Auquel cas, les équipements d'interconnexion suppriment des paquets dans leur ordre d'arrivée. De fait, il est intéressant de déployer des mécanismes de contrôle de congestion préventifs et curatifs permettant de d'ajuster la probabilité de destruction de paquets selon les classes de service. Nous nous limiterons ici à WRED dans une utilisation conjointe avec DiffServ.

 **Note** : Les protocoles de niveau 4 possédant un contrôle de congestion (ex: TCP) sont un bon complément à ces mesures.

WRED permet de définir 2 seuils à partir desquels WRED supprimera une partie des paquets. Ainsi, le premier seuil permet d'obtenir un comportement préventif tandis que le second seuil permet un comportement curatif. Utilisation de WRED :

```
[config-pmap-c]      random-detect dscp-based ←
                      random-detect dscp DSCODE seuil-bas seuil-haut [probabilite_drop]
```

Où DSCODE est la valeur du champ DSCP. seuil-bas et seuil-haut précisent les intervalles de fonctionnement de WRED (en nombre de paquets variant de 1 à 4 096). Probabilite\_drop indique la probabilité pour un paquet d'être supprimé une fois le seuil bas franchi. Cette probabilité est exprimé en fraction pour un paquet (voir exemple). Si aucune probabilité n'est indiquée, IOS applique une taux de 1/10. Les valeurs admissibles pour ce champ vont de 1 à 65 536.

 **Note** : une fois le seuil haut franchi, tous les paquets ayant un DSCP égal à DSCODE sont supprimés.

\* Exemple :

```
random-detect dscp-based ←

random-detect dscp 63 20 40 5 active le support de WRED pour DiffServ. WRED est configuré pour les paquets ayant un DSCP à 63. WRED
va supprimer 1 paquet sur 5 lorsque plus de 20 paquets (ayant ce DSCP) sont dans la file d'attente. Au delà de 40 paquets dans la file, tous les paquets
sont supprimés.
```

## VII. Observations

L'observation des différents éléments de ce TP peut être faite au moyen des commandes suivantes :

# <b>show interfaces</b>	affiche les informations IntServ concernant les interfaces
# <b>show access-lists</b>	Affiche le contenu des listes d'accès
# <b>show policy-map [interface [nom_interface [input   output] [class nom_classe]]]</b>	affiche la configuration et les statistiques des politiques de services et/ou classes de services de la politique et ce pour toutes(ou une) interface
# <b>show queue nom_interface</b>	Affiche le contenu de la files d'attente pour une interface donnée
# <b>show queuing fair</b>	Affiche le statut de la file WFQ
# <b>show queuing fair-queue</b>	Affiche le statut de la file WFQ
# <b>show policy interface nom_interface</b>	Affiche les politiques pour une interface donnée

## VIII.Documentation

Les documents suivants complètent les informations de ce sujet :

- Utilisation de CISCO IOS : <http://www.cisco.com/en/US/docs/ios/preface/usingios.html>
- CISCO QoS : [http://www.cisco.com/en/US/docs/ios/qos/configuration/guide/12\\_4t/qos\\_12\\_4t\\_book.html](http://www.cisco.com/en/US/docs/ios/qos/configuration/guide/12_4t/qos_12_4t_book.html)
- Commandes QoS pour IOS CISCO : [http://www.cisco.com/en/US/docs/ios/qos/command/reference/qos\\_book.html](http://www.cisco.com/en/US/docs/ios/qos/command/reference/qos_book.html)
- DiffServ CISCO : [http://www.cisco.com/en/US/docs/ios/qos/configuration/guide/dffsrv\\_for\\_qos\\_oview.html](http://www.cisco.com/en/US/docs/ios/qos/configuration/guide/dffsrv_for_qos_oview.html)
- IETF RFC sur DiffServ : 2474, 2475, 2597, 2598, 3260.

**IX. Exercice**

1. Proposer et déployer une architecture d'interconnexion.

☒ **Conseil #1** : Pour la suite (q7), nous vous conseillons de paramétrer les mécanismes un par un et d'observer l'effet (le cas échéant les effets) d'un mécanisme AVANT de configurer le mécanisme suivant.

☒ **Conseil #2** : Nous vous conseillons également de déployer les classes de services une à une...

2. Déployer les services de classification et de marquage à l'entrée de votre système autonome.
3. Déployer une politique de contrôle d'accès sur votre réseau en vous basant sur le marquage et la classification des paquets entrant.
4. Déployer une politique d'ordonnancement différencié de paquets en utilisant CBWFQ.
5. Déployer une stratégie de contrôle de congestion.
6. Déployer une politique de lissage de trafic.
7. Déployer un réseau DiffServ possédant les caractéristiques suivantes :

Classe de Service	Sous classe	DSCP	Utilisation	Paramètres
Or	-	46 (EF)	Vidéo	Service Premium. 50% du débit interface
Argent	-	26 (AF31)	Vidéo	30% du débit interface
Bronze	Vidéo	10 (AF11)	Vidéo	10% débit interface
	FTP	14 (AF13)	FTP	
Meilleur Effort	-	0	Tout trafic	Polissage à 20 Kb/s

8. Utiliser les stations 3 et 4 pour charger le réseau. Observer le débit obtenu.
9. Au moyen des outils à votre disposition (VLC, FTP, Wireshark, iperf...), observer les performances.

 **Un dossier doit être remis à la fin des enseignements de TP.**

 **Ce dossier doit contenir :**

- Une présentation de l'architecture d'interconnexion déployée ;
- Une justification des mécanismes à déployer ET de leurs paramètres de configuration pour les différentes classes de services à déployer ;
- Les configurations des équipements.

*Vos notes :*