Haute école d'ingénierie et d'architecture Fribourg
Hochschule für Technik und Architektur Freiburg

Pôle TIC - Filière Télécommunication

Projet de semestre 6

2017-2018

Quel est l'effet de la QoS sur des petits réseaux de labo ?

Rapport du projet

Étudiant : Dufresne Loïc
(loic.dufresne@edu.hefr.ch)

Superviseurs : Jacques Robadey
(jacques.robadey@hefr.ch)
François Buntschu
(francois.buntschu@hefr.ch)

Hes·so FRIBOURG FREIBURG
Haute Ecole Spécialisée de Suisse occidentale
Fachhochschule Westschweiz
University of Applied Sciences and Arts Western Switzerland

Fribourg, 18.05.2018

Remerciements

J'adresse mes sincères remerciements à mes superviseurs, Monsieur Jacques Robadey et Monsieur François Buntschu qui m'ont soutenu, guidés et conseillés tout au long du projet et pour leur disponibilité. Merci beaucoup!

Un grand merci à Monsieur Simon Lièvre pour sa disponibilité à répondre à mes questions sur la configuration des équipements et à Monsieur Pascal Roulin qui m'a permis de tester mes catalogues d'architectures et de scénarios de tests sur son réseau de projet de semestre 6 "Remote Network Monitoring".

Et pour finir, merci à toutes les personnes qui m'ont accompagné afin de réaliser ce projet de semestre 6 du mieux possible.

Table des matières

Historique du document

Version	Étudiant	Tâches	Date
0.1	Loïc Dufresne	Création du document LaTeX	14.02.2018
0.2	Loïc Dufresne	Rendu de la partie analyse	27.03.2018
0.3	Loïc Dufresne	Rendu de la partie conception	16.05.2018
0.3	Loïc Dufresne	Rendu de la partie réalisation	16.05.2018
0.3	Loïc Dufresne	Rendu de la partie tests et validations	16.05.2018
1.0	Loïc Dufresne	Version finale	18.05.2018

1 Introduction

Ce rapport décrit le travail réalisé pour le projet de semestre 6: "Quel est l'effet de la QoS sur des petits réseaux de labo ?". Ce projet est réalisé par M. Loïc Dufresne et supervisé par M. Jacques Robadey et M. François Buntschu.

1.1 Contexte

En 2017, 51% de la population mondiale ont accès à Internet, soit 3,8 milliards d'usagers qui génèrent du trafic sur la toile. L'augmentation de ce trafic ne cesse de s'accroître de manière exponentielle grâce notamment à l'explosion du nombre d'appareils connectés, avec l'"Internet Of Things", plus de 20 milliards d'appareils à ce jour. La vidéo est aussi l'un des principaux facteurs de l'augmentation du trafic sur Internet, grâce notamment à la haute définition et de l'ultra-HD, et aux nouveaux services de "Video on Demand". Malgré cette forte hausse, les fournisseurs d'accès à Internet doivent garantir à leurs clients une certaine qualité de service.

Ce projet de semestre découle de deux travaux, le projet de semestre de M. Simon Lièvre "QoSLab" [1] et le projet de semestre 6 de M. Gabriel Python "Best Network Topology" [2]. Le premier travail consistait à mettre en place des réseaux de tests, dans le but d'analyser les différentes topologies afin d'en trouver l'optimale en fonction des services transportés. Le second projet consistait à l'étude de la qualité de service sur différents types de services, le côté topologies est quant à lui mis de côté.

Le projet de semestre "Quel est l'effet de la QoS sur des petits réseaux de labo ?" met en relation ces deux travaux en se focalisant sur la mise en place d'un réseau de tests permettant d'évaluer au mieux les stratégies de QoS. Différents flux de trafic simulant l'Internet, la télévision, la voix et le management de réseau seront analysés sur cette topologie réseau bien spécifique. Le projet consiste à étudier le trafic traversant ce réseau de bout en bout et à définir quel mécanisme de QoS est le plus adapté aux différents flux.

Dans un second temps, si le temps le permet, ces mêmes analyses seront effectuées sur un réseau émulé, le même réseau que celui mis en place précédemment, afin de comparer si les résultats obtenus par les scénarios sont semblables. Dans le cas où il y aurait peu de différence, nous aurons certifié l'émulateur. Ce serait un point très fort, car cela signifierait que de gros réseaux comme ceux des grands opérateurs suisses pourraient être émulés de manière très réaliste.

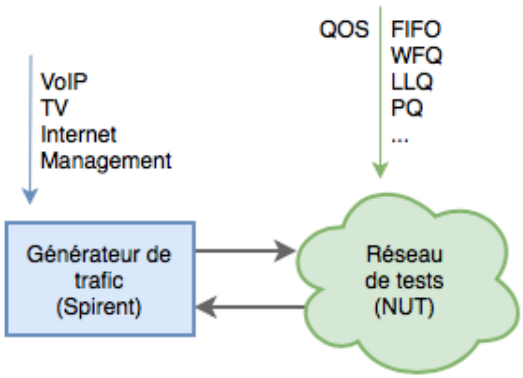


Figure 1 – Illustration de l'architecture

1.2 Objectifs

L'objectif principal est de proposer un catalogue d'architectures d'une ou de plusieurs topologies réseau afin de pouvoir tester au mieux les différentes stratégies de QoS en fonction des flux de trafic. Les différentes stratégies de QoS et les flux de trafic seront regroupés et définis dans un catalogue de scénarios de tests. Une série de mesures documentées sera fournie.

Les objectifs secondaires de ce projet sont:

1. Réaliser l'objectif principal au niveau "Layer 2"
2. Étudier l'émulateur "JAR" afin de définir ces limites et s'il est capable de réaliser les mêmes procédures que dans l'objectif principal
3. Interpréter les résultats obtenus afin déterminer si les réseaux physiques peuvent être émulés de manière très réaliste

La liste des tâches à réaliser pour ce projet est divisée en quatre étapes clés:

1. Analyse
2. Conception
3. Réalisation
4. Tests et validations

Pour l'orientation "Internet et Communication" de la filière "Télécommunications", une partie traitant de l'économie est à réaliser pour ce projet, deux points y seront traités:

1. Définir quel produit ou service peut être extrait de ce projet afin de tester les réseaux des planificateurs réseau
2. Définir la meilleure topologie en termes de QoS et de prix

Cette partie ne sera pas un point à part entière du rapport, mais sera décrite dans les quatre étapes clés du projet.

2 Analyse

Ce chapitre traite de l'analyse de notre projet. Cette analyse permettra de bien comprendre le but du projet, et la manière dont il va être mis sur pieds.

Ce projet se base sur 3 thématiques essentielles à la bonne mise en oeuvre de ce dernier:

1. La QoS et ces mécanismes
2. Les flux de trafic
3. Les topologies réseau

Pour rappel, 4 flux de trafic (Internet, télévision, téléphonie et management de réseau) seront injectés sur un réseau défini par une ou plusieurs topologies bien spécifiques. Ce réseau sera implémenté grâce aux différents mécanismes de QoS. Le but étant de proposer un catalogue d'architectures d'une ou de plusieurs topologies réseau afin de pouvoir tester au mieux ces différentes stratégies de QoS en fonction des flux de trafic.

Nous commencerons cette partie d'analyse avec un état de l'art des différents moyens ou produits permettant de réaliser ces tests. Avant de poursuivre le projet, nous définirons dans les grandes lignes la QoS et ces mécanismes ainsi que les différents flux de trafic et leurs caractéristiques.

Une partie de l'analyse sera aussi consacrée aux différentes topologies réseau afin d'en définir lesquelles sont les plus adéquates aux différents types de flux.

2.1 État de l'art

Pour rappel, l'objectif principal de ce projet est de proposer un catalogue d'architectures d'une ou de plusieurs topologies réseau afin de pouvoir tester au mieux les différentes stratégies de QoS en fonction des flux de trafic.

Pour réaliser ce projet, nous avons 2 points de départ:

1. Le projet de semestre 5 de Monsieur Simon Lièvre "QoSLab" [1]
2. Le projet de semestre 6 de M. Gabriel Python "Best Network Topology" [2]

Nous allons sortir les points de conclusion des deux projets précédemment réalisés, du langage technique est déjà utilisé ci-dessous, mais l'analyse définit tous ces termes. Nous allons, aussi, comparer certains produits ou services mis à disposition afin de concevoir et tester des réseaux.

2.1.1 "QoSLab"

L'objectif de ce projet était de: *"mettre en place un laboratoire de test permettant aux collaborateurs et étudiants de vérifier leurs stratégies de QoS sous différentes conditions de trafic, et démontrer la mise en application d'une stratégie de qualité de service permettant de répondre aux exigences de trafic des flux : VoIP, vidéo, internet et management."* [1]

Au niveau de la QoS, ce travail était essentiellement centré sur les types de files d'attente. Plusieurs types de files d'attente ont été configurés et testés afin d'analyser si ces dernières répondent aux exigences des différents flux.

Nous ressortons les informations concernant la couche 3 du modèle OSI uniquement, vu que notre travail est centré sur cette couche.

Lors de surcharge du réseau, la file d'attente FIFO (Firt In First Out) ne répond pas aux exigences des flux voix et management. La file d'attente LLQ (Low-Latency Queuing) offre de meilleurs résultats en termes de latence du flux de voix, que la file d'attente CBWFQ (Class-Based Weighted Fair-Queuingt). La file d'attente PQ (Priority Queuing) offre également de très bons résultats, mais le trafic Internet ne passe plus ou presque. Ces files d'attente décrites ci-dessus sont les 4 types de files d'attente analysées et testées lors de ce travail.

"Si l'on prend le cas d'un réseau couche 3, que l'on maîtrise de bout en bout, dans lequel l'on dispose d'outils d'analyse offrant des statistiques avancées et que l'on souhaite y appliquer une stratégie de QoS, alors l'on va peut-être pencher, soit pour du CBWFQ, soit pour du LLQ en fonction de si l'on transporte de la voix (ou un autre flux très sensible à la latence) ou non. Dans les 2 cas, la configuration se fait par classes de trafic pour lesquelles l'on réserve de la bande passante. On comprend alors l'utilité d'outils d'analyse du réseau afin de déterminer précisément les quantités de bande passante à réserver. Ces stratégies auront l'avantage de garantir une certaine bande passante minimale et donc aucun flux n'est complètement bloqué. Toujours, pour un réseau couche 3, dans le cas où l'on ne possède pas de tels outils ou que la charge du réseau varie fortement d'un moment de la journée à l'autre, sans pouvoir prédire l'utilisation future d'un flux en particulier, alors le PQ semble être plus adapté. La configuration se fait en attribuant des niveaux de priorité prédéfinis. Les résultats sont bons, mais il n'y a aucune garantie de bande passante." [1]

Ces définitions montrent quels types de files d'attente correspondent le mieux aux différents flux. Notre travail se base sur l'analyse de ces types de files d'attente, nous testerons nos réseaux avec les mêmes files. Voici notre point de départ concernant ce projet.

2.1.2 "Best Network Topology"

L'objectif de ce projet était de: *"pouvoir conseiller une topologie en fonction des besoins réseau et des services transportés. Afin de pouvoir fournir ces conseils, sept topologies différentes sont définies dans la donnée du projet et sont les suivantes : ligne, étoile avec gros nœud central, étoile avec petit nœud central, anneau, double anneau (2 anneaux de 4 nœuds), maillé (multitriangles), entièrement maillé (point à point)."* [2]

Au niveau des topologies réseau, 3 des 7 topologies ont été réalisées et testées. Nous pouvons en tirer qu'il n'existe pas vraiment de "meilleure topologie", car tout dépend du type de trafic. En effet, pour un trafic de type téléphonique, la meilleure topologie est la topologie entièrement maillée, car le trafic téléphonique va de chaque point à chaque point, contrairement aux autres trafics.

"Le trafic Internet et TV sont générés depuis un nœud central étant relié au serveur TV ainsi qu'à l'« Internet Peering ». Une topologie en étoile est plus intéressante pour ces types de trafic, car tout partirait du nœud central. Il faut néanmoins faire attention à la redondance du nœud central en le dupliquant ou en mettant en place un système avec différentes couleurs de fibre permettant de rediriger le trafic vers un nœud Backup. En ce qui concerne la fiabilité, une topologie de type entièrement maillée est la topologie la plus fiable, car c'est celle qui comporte le plus de liens entre les nœuds donc permet une panne de plusieurs liens. Pour les topologies qui ont été simulées, le double anneau est le plus fiable, car il permet une perte de 2 liens en moyenne." [2]

Ces définitions montrent quelles topologies de réseaux correspondent le mieux aux différents flux. Notre travail se base sur l'analyse de ces différentes topologies, nous définirons une topologie de tests en nous basant sur ces informations. Voici notre point de départ concernant ce projet.

2.1.3 Produits/Services de tests

Pour la partie économique consacrée à ce projet, nous définirons quel produit ou service peut être extrait de ce projet afin de tester les réseaux des planificateurs réseau. Bien entendu, nous définirons cela lorsque le projet sera réalisé. Mais dans un premier temps, nous allons faire le tour des produits ou services existants, permettant de tester différentes topologies réseau et leurs QoS.

Il existe plusieurs manières de tester une topologie spécifique ainsi que sa QoS.

Spirent

Spirent est le premier fournisseur, dans le monde, de solutions de test, d'analyse et de sécurité pour les réseaux de fournisseurs de services physiques ou virtuels, pour les centres de données d'entreprise et tous autres types de domaines.

Spirent fournit le Spirent TestCenter qui est un software proposant une solution de test de bout en bout offrant des performances élevées. Les fournisseurs de services et les entreprises l'utilisent pour tester, mesurer, valider leurs réseaux et déployer des services en toute sécurité. Il permet de réaliser des tests de haute performance avec des analyses très poussées de la virtualisation, du cloud Computing, du Backhaul mobile et de l'Ethernet à haut débit. Ce software de tests peut facilement automatiser les tâches répétitives afin d'étendre la couverture des tests correspondant à notre environnement.

Ce software est adéquat pour tester les réseaux et des infrastructures de nouvelle génération, en prenant en charge une multitude de protocoles émulés en réseau. Il réduit le temps des tests et des coûts avec des réponses en temps réel et pas seulement des résultats. Il est aussi capable d'émuler des topologies de réseaux complexes et les conditions de trafic réelles.

Le Spirent TestCenter permet de configurer des appareils de tests, comme le Spirent C1, qui est un puissant générateur permettant de créer des flux très variés avec une multitude de paramètres, comme la classification des paquets au niveau de la QoS. Spirent propose une large palette d'autres softwares de tests et de générations de trafic, pouvant être associé à plusieurs équipements.

Nous avons l'avantage à la HEIA-FR d'être en possession du Sirent TestCenter et du générateur de flux Spirent. Afin d'en profiter, le projet sera justement orienté autour de ces éléments.

Pour plus d'informations: `https://spirent.com`

Siama Systems

Siama Systems propose des solutions de tests suivant la même direction que Spirent. Cette entreprise propose des solutions de tests afin de garantir la conformité et la performance des réseaux testés. Ici aussi, les fournisseurs de services physiques ou virtuels, les entreprises, les centres de données et les fabricants de matériel de réseau peuvent utiliser les services Siama pour vérifier leurs produits et services, réduire les coûts de déploiement de leur infrastructure.

Siama Systems propose deux produits.

1. **PROVA-X**: est un appareil permettant de tester et d'analyser le comportement et les performances d'un réseau, d'un Cloud public ou privé ou d'une interface virtualisée. Cet instrument permet de simuler une multitude de flux et de capturer des flux spécifiques afin de les décoder très simplement.
2. **GENEM-X**: est un software permettant la génération et la capture de trafic fonctionnant sur l'appareil PROVA-X. Les fonctions d'émulation de réseau permettent d'ajouter divers paramètres comme des dégradations (retards, gigue, pertes de paquets aux flux) pour imiter au mieux des réseaux complexes. Il peut être utilisé dans un environnement de production ou de laboratoire. GENEM-X permet de manipuler l'appareil PROVA-X en instrument de tests de haute performance.

Contrairement à Spirent qui offre plusieurs softwares et instruments de tests, Siama Systems propose un software de paramétrages et de mesures, et un instrument de génération de trafic, mais offrant plus ou moins les mêmes services. Ces deux entreprises offrent la possibilité de tester des réseaux physiques ou virtualisés. Pour valider une architecture avant sa mise en production, avec ces deux solutions, il faut du matériel, de l'espace et du temps, le problème est que cela coûte cher, une alternative, est de réaliser ces tests avec des outils virtualisant les réseaux.

Pour plus d'informations: `https://siamasystems.com`

VIRL (Virtual Internet Routing Lab)

VIRL est une plateforme de virtualisation proposée par Cisco, elle permet de créer, suivant des modèles très précis, des réseaux existants ou planifiés afin de pouvoir les tester.

Grâce à cette plateforme, les fournisseurs de services physiques ou virtuels, les entreprises et les fabricants peuvent concevoir, réaliser, visualiser, tester leurs réseaux en créant des simulations d'équipement Cisco ou autres dans un environnement virtuel. Des modèles et des scénarios de simulation de réseaux réels et futurs peuvent être définis, des configurations automatiques peuvent être générées, une multitude de protocoles peuvent être visualisés.

VIRL propose des services ressemblant à ceux proposés par Spirent ou Siama Systems mais dans un environnement virtuel. Bien entendu avec les instruments proposés par Spirent ou Siama Systems, les tests et les paramétrages sont bien plus poussés, performants et réalistes. L'avantage de l'environnement virtuel est le côté économique au niveau du temps, de l'espace et des coûts des instruments.

Pour plus d'informations: `http://virl.cisco.com`

GNS3 (Graphical Network Simulator 3)

GNS3 est un software qui permet de concevoir, construire et tester des réseaux dans un environnement virtuel, suivant la ligne proposée par VIRL.

Ce software propose des simulations en temps réel pour les tests avant le déploiement d'un réseau, toutes les solutions décrites précédemment proposent, au final, des services semblables. Mais au point en plus pour GNS3 qui est gratuit.

Pour plus d'informations: `https://gns3.com`

Packet Tracer

Packet Tracer est un software proposé par Cisco permettant de simuler des réseaux. Cet outil est plus orienté sur l'apprentissage de la conception réseau contrairement à VIRL, aussi développé par Cisco, qui lui est un outil professionnel.

Ce software permet de compléter les équipements physiques et permet de créer un réseau avec un nombre pratiquement illimité d'appareils, il encourage l'apprentissage dans la manière de concevoir et de dépanner.

Pour plus d'informations: `https://netacad.com/fr/courses/packettracer/`

CNLab

CNLAB est un groupe suisse constitué de trois sociétés d'ingénierie informatique offrant des solutions et des services dans les domaines de la performance Internet, du développement de logiciels et de la sécurité informatique. CNLAB a mis au point un outil permettant de réaliser des "speedtest" très performants.

Cet outil ne suit pas la même lignée que ceux présentés précédemment, mais avec ce dernier, nous pouvons réaliser des tests afin d'analyser les débits réels mis à disposition par les opérateurs afin de se faire une idée lors de tests de conception de réseau.

Le "speedtest" CNLAB est conseillé par Swisscom ou encore UPC Cablecom, il s'agit d'un très bon outil. Une grande quantité d'informations est traitée, analysée et donnée par cet outil. Des séquences de tests journalières peuvent être paramétrées et un historique de mesures est disponible.

Pour plus d'informations: `https://www.cnlab.ch`

2.1.4 Synthèse

Comme nous avons pu le voir, le projet que nous allons réaliser est la suite des deux projets "QoSLab" [1] et "Best Network Topology" [2]. Nous avons deux points de départ, d'un côté un travail sur la QoS et de l'autre un travail sur les topologies réseaux. Nous allons réaliser un travail en assemblant ces deux notions afin d'en tirer un produit ou service. Une analyse de la QoS et des topologies est, bien entendu, nécessaire, mais sans "refaire ce qui a été déjà fait". Ces notions seront analysées dans leur base pour notre compréhension personnelle, mais pour la suite, nous nous dirigerons plus vers des éléments peut-être moins bien décrits afin de creuser vers d'autres directions.

Concernant les produits et services sur le marché permettant de créer et tester les réseaux. Nous avons pu voir qu'il existe plusieurs variantes. D'un côté, les solutions permettant de travailler avec des réseaux physiques et des équipements puissants et professionnels, mais impliquant un certain coût au niveau de l'espace à disposition et du prix des équipements, ces solutions sont faites pour les grandes entreprises de télécommunications. Et d'un autre côté les solutions permettant de travailler avec des réseaux virtuels, cette variante est peut-être plus dirigée vers l'apprentissage, bien que ces produits soient très professionnels.

2.2 QoS

"La qualité de service (QDS) ou quality of service (QoS) est la capacité à véhiculer, dans de bonnes conditions, un type de trafic donné, en termes de disponibilité, débit, délais de transmission, gigue, taux de perte de paquets." [2]

Voici une définition de la QoS, proposée sur Wikipedia, qui donne un bon aperçu de l'utilité de ce concept.

Tout d'abord, imaginons un réseau IP basique, sans la notion de QoS. Ce réseau IP va transporter les paquets des différents flux, de la source vers la destination, sans faire de distinction entre ces flux. Cette manière de faire peut fonctionner tant que les liens ne sont pas surchargés. Mais lorsque la bande passante n'est plus suffisante, les liens commencent à surcharger et les routeurs rejettent les paquets, ce qui est problématique. Les performances du réseau baissent drastiquement.

Maintenant, imaginons ce même réseau IP implémentant une des notions de QoS. La capacité du réseau sera partagée le plus équitablement entre les différents flux. Lorsque le réseau commence à surcharger, au lieu de rejeter des paquets, les routeurs vont garder ces paquets en mémoire, dans une file d'attente, et selon le type de mécanisme de QoS, les routeurs vont envoyer, d'abord, les paquets les plus anciens, ou selon un type flux ou même une priorité, et si cette file d'attente commence à saturer, rejeter les paquets en fonction des priorités. Cet exemple de mécanisme de QoS est connu sous le nom d'ordonnancement, nous reviendrons sur cette notion plus tard.

L'architecture actuelle d'Internet reste inchangée depuis ses débuts. Vu l'explosion du trafic Internet ces dernières années, la charge sur les réseaux évoluant de manière exponentielle, les notions de QoS dans le réseau est maintenant primordiale pour garantir les performances du réseau, optimiser les ressources de ce réseau et surtout la satisfaction du client qui paie son service.

Pour des informations plus détaillées sur la QoS, ces caractéristiques et mécanismes, veuillez vous référer au projet de semestre 5 de Monsieur Simon Lièvre "QoSLab" [1].

2.2.1 Architectures et protocoles

La gestion de la QoS sur un réseau peut être travaillée à différents niveaux des couches du modèle OSI, voici son architecture et ces protocoles:

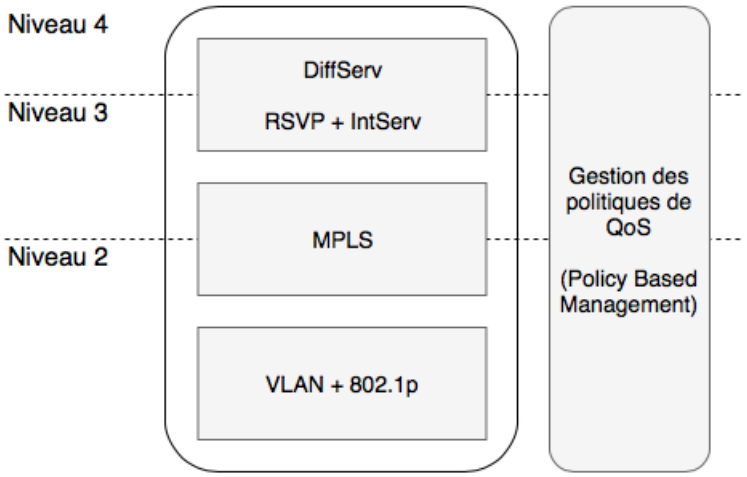


Figure 2 – L'architecture et les protocoles de gestion de la QoS

Nous allons mettre des mots sur ces terminologies:

- **DiffServ** (Differentiated Services): ce modèle définit une architecture de réseau qui permet de différencier les services des données, il a été mis en place afin de pallier aux difficultés de déploiement du modèle définissant l'architecture IntServ. Le mécanisme qui en découle classe et contrôle le trafic, le trafic est séparé en classe de trafic et ces derniers peuvent être identifié grâce à une valeur codée dans l'en-tête IP. Ces tâches sont réalisées en périphérie du réseau et les routeurs du coeur de ce réseau traitent les paquets en fonction de la classe qui les définissent. Une combinaison de ce modèle avec le protocole MPLS fournit un niveau de QoS optimal. Nous reviendrons sur ce modèle, une fois les différents mécanismes de QoS défini.
- **IntServ** (Integrated Services): ce modèle définit une architecture de réseau qui permet de prendre en charge la QoS en temps réel. Il a été conçu sur le principe de la réservation de ressources, les applications établissent un chemin à travers le réseau et réservent leurs ressources sur ce dernier.
- **RSVP** (Resource Reservation Protocol): ce protocole est utilisé avec le modèle IntServ afin de gérer les réservations des ressources.
- **MPLS** (MultiProtocol Label Switching): ce mécanisme de routage est *"basé sur des labels. Le but de MPLS est d'étendre les services fournis par le réseau IP en offrant un cadre pour l'ingénierie de trafic, la QoS et les Virtual Private Network (VPN). Il fonctionne en parallèle aux technologies de routage existantes et offre aux réseaux IP un mécanisme permettant le contrôle explicite des chemins de routage."* [1] Voici une bonne définition proposée dans le projet de semestre 5 de Monsieur Simon Lièvre "QoSLab" [1]. À noter que ce mécanisme peut être appliqué avec le modèle DiffServ et IntServ.
- **VLAN** (Virtual Local Area Network) **+ 802.1p**: ce réseau virtuel est défini par la norme 802.1p, elle permet la différenciation des services au niveau de la couche 2 du modèle OSI grâce à un identifiant VLAN.

Ces modèles de gestion de la QoS sont directement liés au niveau de services, définis ci-dessous.

2.2.2 Niveaux de service

Nous allons voir qu'il existe 3 niveaux de service de la QoS de réseau. Par niveau de service, nous voulons dire, le niveau d'exigence de QoS à mettre en place pour une infrastructure. Dans un réseau, nous avons du trafic point à point généré par différents services d'applications, le niveau d'exigence de QoS est coordonné par rapport aux besoins de ces applications à fournir un bon service. Les niveaux de services se basent sur les différents flux afin de mettre en place cette exigence au niveau de la QoS, si QoS il y a. Ces 3 niveaux de service sont:

1. **Les services au mieux** (Best Effort): ce niveau de service de QoS, ne propose pas de QoS. En effet, il propose uniquement un service permettant la connexion entre 2 points sur le réseau. Les flux ne sont pas différenciés sur le réseau. Il n'y a donc aucune garantie pour les services.
2. **Les services différenciés** (Differentated Services): ce niveau de service permet de différencier les flux afin de mettre en place des niveaux de priorité en fonction de ces flux. Les flux prioritaires sont donc acheminés avec de meilleurs délais à travers le réseau, mais il n'y a, ici aussi, aucune garantie.
3. **Les services garantis** (Guaranted Services): ce niveau de service se base sur le fait de réserver des ressources sur le réseau en fonction des types de flux. Plusieurs mécanismes peuvent être mis en place afin de réaliser ce niveau de service, mais le protocole RSVP (Resource reSerVation Protocol) est le plus utilisé. Ce type de services définit le trafic prioritaire sur le réseau.

En général, les entreprises et les opérateurs mettent en place ces 3 types de service, et leur attribution à des applications se fait en fonction de ces dernières.

Nous pouvons mettre, maintenant, en relation les niveaux de services et les modèles de gestion de la QoS présentés précédemment:

Niveaux de services	**Modèles de gestion de la QoS**
Les services au mieux (Best Effort):	Réseau IP traditionnel (Internet)
Les services différenciés (Differentated Services)	DiffServ (Differentiated Services)
Les services garantis (Guaranted Services)	IntServ (Integrated Services)
Optimisation du trafic	MPLS (MultiProtocol Label Switching)

2.2.3 Caractéristiques

Les caractéristiques techniques suivantes définissent les notions de QoS:

- **Fiabilité ou disponibilité**: le réseau et ces services doivent être fiables et disponibles afin de pouvoir acheminer les paquets
- **Bande passante**: le réseau et ces services doivent mettre à disposition suffisamment de bande passante afin de pouvoir absorber tout le trafic, on parle ici du débit du réseau
- **Le délai**: le réseau et ces services doivent assurer un acheminement rapide du trafic de la source vers la destination, il est synonyme de latence
- **La régularité**: le réseau et ces services doivent assurer un acheminement régulier du trafic, on parle ici de gigue
- **Le taux de pertes**: le réseau et ces services doivent assurer un acheminement sans perte du trafic, c'est un rapport entre le nombre de paquets émis et le nombre de paquets perdus

Ces caractéristiques techniques de QoS peuvent être appliquées à un trafic précis découlant d'une application ou d'un utilisateur spécifique. Elles peuvent aussi être appliquées à un ensemble de trafics précis découlant de plusieurs applications ou plusieurs utilisateurs ayant en commun des exigences semblables.

2.2.4 Mécanismes

Pour mieux illustrer les mécanismes utilisés par un routeur implémentant la QoS, voici une figure:

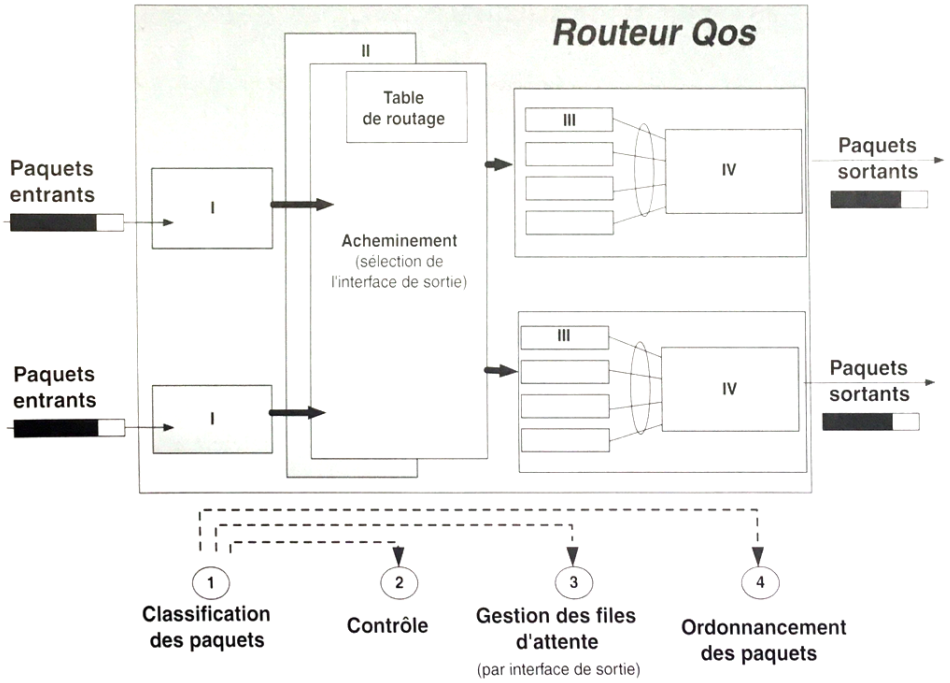


Figure 3 – Les mécanismes d'un routeur avec QoS [3]

Nous pouvons voir que 4 étapes sont définies dans ce routeur, ces étapes représentent les mécanismes mis en oeuvre afin de garantir la QoS, voici ces 4 étapes:

1. **La classification**: classe le paquet en fonction de leur contenu. Cette étape se passe avant l'entrée dans la table de routage des paquets. Elle permet d'analyser les caractéristiques d'un paquet grâce à son contenu dans le but de trouver, pour ce paquet, une entrée dans la table de routage qui s'occupe de déterminer son interface de sortie du routeur et sa file d'attente parmi les files d'attente de cette interface. Une priorité leur est définie.
2. **Le contrôle et marquage**: contrôle et, si besoin, marque le paquet. Cette étape permet de contrôler si le paquet reçu remplit les conditions de conformité au profil de trafic prévu. Si le paquet est non conforme au profil de trafic prévu, ce dernier peut soit être éliminé, soit , si le réseau n'est pas saturé, être marqué et réacheminé.
3. **La gestion des files d'attente**: gère les files d'attente afin de réduire la taille de ces dernières pour que le traitement du trafic soit optimal. Le nombre de paquets contenus dans une file d'attente doit être minimal, pour le faire, cette étape influence les émetteurs afin qu'ils réduisent leurs émissions en cas de congestion.
4. **L'ordonnancement**: achemine les paquets contenus dans les files d'attente vers l'interface de sortie du routeur en fonction des étapes de classification et de contrôle. Cette étape permet d'optimiser la perception de l'utilisateur face au délai et à la régularité du service attendu. Afin de réaliser cette tâche, l'ordonnanceur peut utiliser plusieurs sortes de files d'attente.

Un routeur avec QoS doit pouvoir mettre en place toute une logique de traitement du trafic. Cette logique de traitement est définie par des algorithmes de traitement et de plusieurs files d'attente par interfaces de sorties des routeurs.

Ordonnancement

Pour mettre en place la QoS, l'ordonnanceur utilise plusieurs modes de files d'attente. Le type de files d'attente est important pour nous, car il faut déterminer laquelle utiliser en fonction du type de flux de trafic.

Voici les différents modes de files d'attente:

- **FIFO** (First In First Out): comme son nom l'indique, premier arrivé, premier sorti. Ce mode de file d'attente est le plus utilisé, il s'agit aussi du monde par défaut sur les routeurs. Les flux ne sont pas différenciés, ils n'ont pas de priorité entre les paquets. Si le trafic est important, les applications sensibles au délai d'acheminement peuvent être fortement pénalisées.
- **PQ** (Priority Queuing): *"la différenciation des services est limitée à un nombre restreint de classes de trafic. Les flux de la même classe ne sont pas isolés les uns des autres. Nécessite un contrôle de trafic pour éviter que le trafic prioritaire monopolise la ligne."* [1] Le nombre de files d'attente peut être de maximum 4 et cela nécessite une configuration. Nous pouvons retenir que les files d'attente de haute priorité sont servies en premier et le trafic de haute priorité est protégé.
- **WFQ** (Weighted Fair Queuing): *"chaque flux obtient un poids. L'allocation de la bande passante est proportionnelle au poids. Si un flux ne consomme pas toute sa bande passante, elle est réallouée aux autres flux actifs. Un flux est certain d'obtenir sa bande passante et peut même en obtenir plus. Il a été démontré que WFQ est capable de garantir des limites de délai. Il en existe plusieurs variantes comme : Worst Case Fair Queuing (WF2Q), Hierarchical Weighted Fair Queuing (HWFQ)."* [1] Le nombre de files d'attente est configurable, mais par défaut aucune configuration n'est nécessaire et le nombre maximal de files d'attente est de 256. Nous pouvons retenir que cette méthode assure une équité entre tous les flux de trafic en fonction des poids fixés.
- **WRR** (Weighted Round Robin): *"chaque flux possède un poids. L'algorithme les sert l'un après l'autre et envoie un certain nombre de paquets en fonction de son poids. WRR fonctionne particulièrement bien quand la taille des paquets est fixe ou que la taille moyenne d'un paquet est connue. Dans le cas contraire, l'allocation de la bande passante n'est pas équitable."* [1]
- **DRR** (Deficit Round Robin): *"similaire à WRR mais prend en compte les paquets de tailles variables. Chaque flux possède en plus de son poids un compteur de déficit initialisé à 0. L'algorithme transmet un nombre fixe de bits appelé le « quantum ». Lors de chaque transmission, le quantum est décrémenté de la taille du paquet envoyé. L'algorithme transmet autant de bits que le quantum d'un flux le lui permet. Le quantum inutilisé est comptabilisé dans le compteur de déficit du flux. Il représente le quantum que lui doit l'ordonnanceur. Au prochain tour, ce déficit est ajouté au quantum du flux qui peut alors transmettre plus de paquets. Le quantum peut être attribué en fonction du poids d'un flux pour se rapprocher du WFQ."* [1]
- **CBWFQ** (Class-Based Weighted Fair-Queuing): étend la file d'attente WFQ pour prendre en charge des classes de trafic définies. Ces classes de trafic sont basées sur des critères de correspondance, comme des protocoles, des listes de contrôle d'accès. Une file d'attente est réservée pour chaque classe et le trafic correspondant à une de ces classes est dirigé vers la file d'attente correspondante. Une fois la classe définie, des critères sont attribués à cette classe comme la bande passante, le poids et la limite maximale de paquets. La bande passante attribuée à une classe est la bande passante garantie délivrée à la classe pendant la congestion. La limite maximale de paquets dans la file d'attente d'une classe est le nombre maximal de paquets autorisés à s'accumuler dans cette file d'attente. Lorsque la limite de la file d'attente est atteinte, la mise en file d'attente de paquets supplémentaires dans la classe entraîne l'effacement de la queue ou du paquet, en fonction de la configuration de la classe.
- **LLQ** (Low-Latency Queuing): variante de la file d'attente CBWFQ comportant une queue spéciale pour le trafic temps réel. Elle permet aux paquets sensibles au retard, comme la voix envoyée en premier, ce qui donne un traitement préférentiel aux données sensibles au retard.

Il n'existe pas de meilleur type de files d'attente à mettre en place, tout dépend des besoins. Si le réseau est maîtrisé de bout en bout, on va se pencher, soit pour du CBWFQ, soit pour du LLQ en fonction des types de flux transportés comme la voix ou un autre flux très sensible à la latence. Si le réseau n'est pas maîtrisé de bout en bout et que le trafic varie fortement, le PQ est plus adapté, car des niveaux de priorité sont définis, mais il n'y a aucune garantie de bande passante.

2.3 DiffServ (Differentiated Services)

Comme nous avons pu le voir précédemment, dans un modèle d'architecture DiffServ, les tâches de QoS, qui sont le contrôle et la classification du trafic sont réalisées en périphérie du réseau sur les routeurs Edge. Une fois ces tâches réalisées, les routeurs "Core" n'ont plus qu'à traiter ces flux en fonction de leur classification. Voici une figure représentant ces mécanismes:

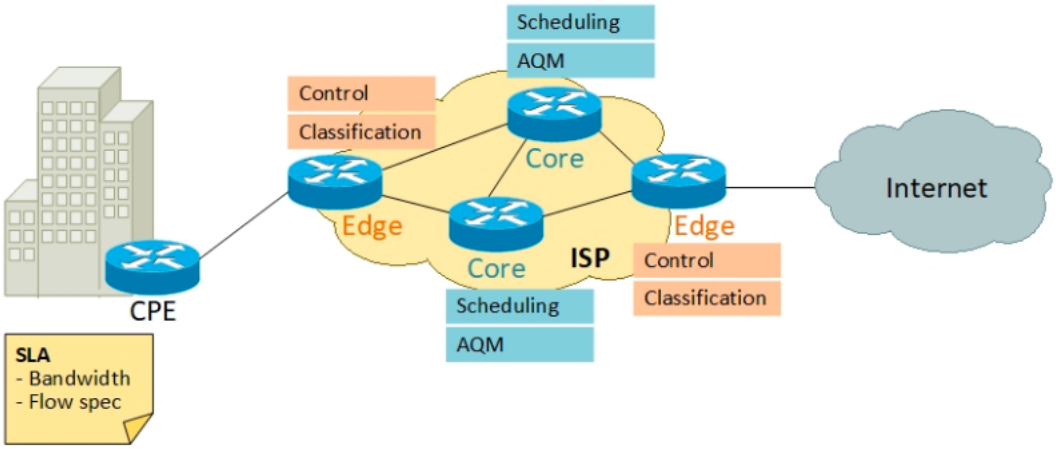


Figure 4 – Les tâches réalisées dans un modèle d'architecture DiffServ [1]

La classification des paquets est le premier mécanisme réalisé afin de garantir une certaine QoS. Mais comment cette classification est elle visible dans un paquet?

2.3.1 DSCP (Differentiated Services Code Point)

Historiquement, la QoS a vu le jour avec le champ ToS (Type of Service) codé sur 8 bits de l'en-tête IPv4, ce champ définit 2 sous-champs:

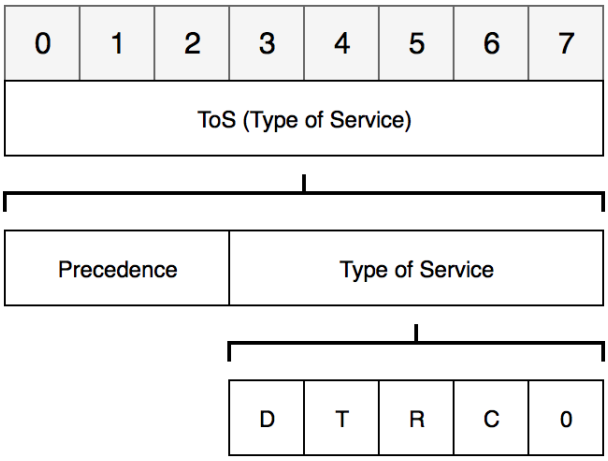


Figure 5 – Champ ToS (Type of Service)

Le champ Precedence permet d'indiquer 8 niveaux de priorités (000 pour la plus faible et 111 pour la plus forte).

Les 4 premiers bits qui composent le champ Type of Service permettent de préciser le mode de transport du paquet et le dernier bit est à 0:

Bits	Signification
D	Délai court
T	Délai élevé
R	Transport fiable
C	Coût maximal

Figure 6 – Bits du champ type of Service

Il n'y avait pas vraiment de règles à suivre pour l'utilisation de ce champ et ces sous-champs et il n'était que très rarement utilisé. Par exemple, Cisco utilisait le champ Precedence avec certains protocoles comme RIP, OSPF ou EIGRP, ou même les flux Telnet pour marquer les paquets entrants ou sortants des routeurs avec une valeur de 110 afin de définir un paquet de Internetwork Control.

Pour résumer, chacun pouvait en faire ce qu'il en voulait de ce champ ToS, ce qui n'était pas vraiment adéquat pour la garantie d'une bonne QoS. C'est pourquoi une recommandation a été proposée par l'IETF quant à l'utilisation de ce champ et ces sous-champs. Cette recommandation définit le champ DC (Differentiated Services) qui remplace le champ ToS sur ces 8 bits, bien entendu, ici nous parlons d'IPv4:

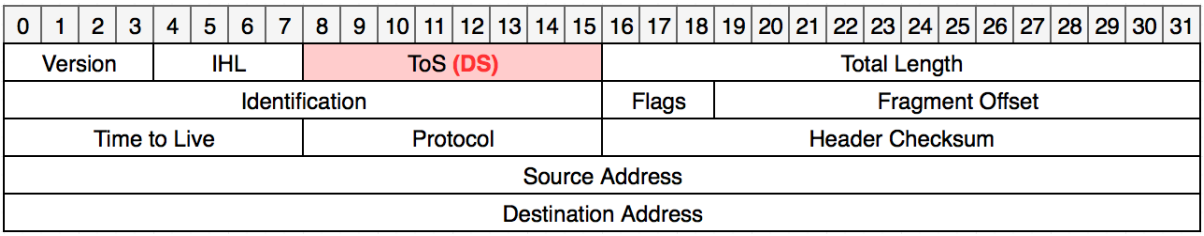


Figure 7 – En-tête IPv4

Au niveau d'IPv6, ce même champ DC est spécifié en utilisant le champ Trafic Class:

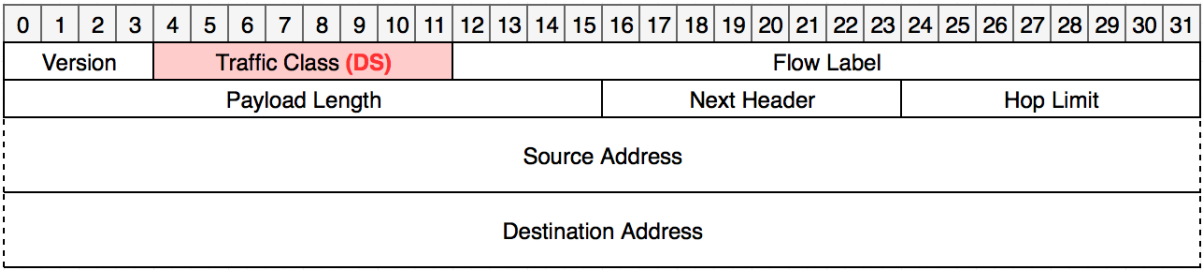


Figure 8 – En-tête IPv6

Le champ DC, dans le modèle DiffServ, définit 2 sous-champs:

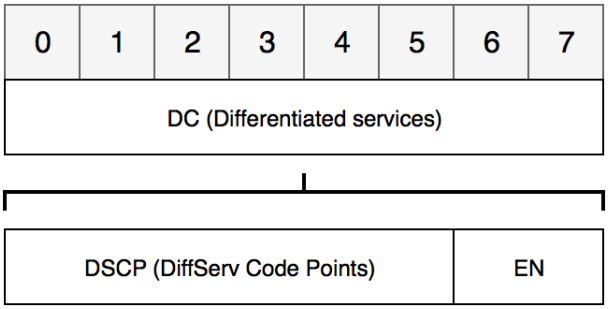


Figure 9 – Champ DC (Differentiated services)

Le champ ENC (Explicit Congestion Notification) permet de signaler une congestion sur le réseau, avant que la perte de paquets ne se produise, avec 4 valeurs:

Bits	Signification
00	Ne supporte pas les notifications de congestions
01	La source supporte les notifications de congestion
10	La destination supporte les notifications de congestion
11	Une congestion est survenue sur le réseau

Figure 10 – Champ ECN (Explicit Congestion Notification)

Pour résumer, la classification ou priorisation des paquets se définit dans le champ DSCP qui est la nouvelle dénomination du champ ToS dans un contexte de réseau DiffServ.

Ce champ étant codé sur 6 bits, il permet, en théorie, d'appliquer jusqu'à 64 valeurs différentes à un paquet contre 8 avec le champ Precedence défini précédemment. Ce qui permet aux opérateurs des réseaux une grande flexibilité dans la définition des classes de trafic, car les RFCs recommandent à ces derniers des définitions de classes, mais n'exigent rien.

Les 3 premiers bits du champ DSCP indiquent la classe de trafic et les 3 derniers bits indiquent la probabilité de suppression du paquet. Les routeurs suivant le modèle DiffServ appliquent des comportements par bond ou PHB (Per-Hop Behaviors), qui définissent les propriétés de transfert de paquets associées à une classe de trafic. Différents PHB peuvent être définis pour offrir, par exemple, une faible perte ou une faible latence.

Dans la plupart des réseaux, les classes suivantes, définies par les PHB, sont le plus souvent utilisées:

- **BE** (Best Effort): dédié aux flux meilleurs effort, aucun traitement particulier n'est appliqué à ces paquets de la part des noeuds du réseau
- **EF** (Expedited Forwarding): dédié aux flux sensibles aux délais, à faible perte et à faible latence
- **AF** (Assured Forwarding): dédié aux flux sensibles aux variations de la bande passante dans le réseau
- **CS** (Class Selector): dédié au maintien de la compatibilité avec le champ Precedence

Valeur	Classification Layer 3		Probabilité de suppression	Equivalence du champ Precedence
	PHB	DSCP		
101 110	EF (haute priority)	46	-	101 (critique)
000 000	BE (défaut)	0	-	000 (routine)
001 010	AF11	10	Faible	001 (priorité)
001 100	AF12	12	Moyen	
001 110	AF13	14	Fort	
010 010	AF21	18	Faible	010 (immédiat)
010 100	AF22	20	Moyen	
010 110	AF23	22	Fort	
011 010	AF31	26	Faible	011 (flash)
011 100	AF32	28	Moyen	
011 110	AF33	30	Fort	
100 010	AF41	34	Faible	100 (flash override)
100 100	AF42	36	Moyen	
100 110	AF43	38	Fort	
001 000	CS1	8	-	1
010 000	CS2	16	-	2
011 000	CS3	24	-	3
100 000	CS4	32	-	4
101 000	CS5	40	-	5
110 000	CS6	48	-	6
111 000	CS7	56	-	7

Figure 11 – Comportement par bond et les classes [4]

Les classes PHB et le champ DSCP classifient les paquets au niveau Layer 3.

"En résumé, DiffServ propose de diviser le trafic en un nombre restreint de classes et les ressources sont affectées en fonction de celles-ci. L'architecture DiffServ normalise un ensemble de PHB. Ces PHB sont encodés dans le champ "DS" (corresponds au ToS d'IPv4 et au "Trafic Class" d'IPv6) de l'en-tête IP. Le trafic est associé à une classe de traitement par les mécanismes de classification et de contrôle implémentés sur les routeurs en bordure de domaine. Le contrôleur de trafic compare le trafic entrant au profil négocié avec le client. Les paquets non conformes sont sujets à des mesures de répression comme le marquage, le retardement et la suppression. À l'intérieur du domaine, les paquets sont acheminés en fonction de leur "DSCP". 3 PHB (AF, BE, EF) ont étés normalisés. Le PHB AF spécifie 4 sous-classes d'acheminement, chacune avec sa propre bande passante. À l'intérieur de chaque sous-classe, AF dispose de 3 niveaux de probabilité de suppression. Le PHB EF offre un traitement des paquets avec des délais et des pertes minimums." [1]

2.4 Flux de trafic

"Un flux est un stream individuel et unidirectionnel de données, entre 2 applications identifiées par 5 paramètres:

- *Le protocole de transport*
- *L'adresse de la source*
- *Le numéro de port de la source*
- *L'adresse de la destination*
- *Le numéro de port de la destination"*

Voici une définition d'un flux, proposée par Jean-Louis Mélin. [5]

Dans ce travail, nous nous intéresserons à 4 types de flux de trafic:

1. Téléphone IP (VoIP)
2. Télévision (multicast et unicast)
3. Internet
4. Management

Ces flux sont sensibles à 3 facteurs principaux:

1. **Le délai ou latence**: indique le temps que parcourt un bit entre la source et la destination
2. **La gigue**: indique la variation du temps de latence entre le temps le plus long et le temps le plus court
3. **Le taux de pertes**: indique le rapport entre le nombre de paquets émis et le nombre de paquets perdus

Nous allons définir et analyser ces différents flux.

2.4.1 Téléphone IP (VoIP)

Le flux de trafic représentant la téléphonie sur IP (VoIP) est très sensible aux 3 facteurs présentés ci-dessus (le délai, la gigue et le taux de perte).

Des aspects déterminent la qualité de la voix au travers d'un réseau. Ces aspects sont la clarté de la voix, le traitement de la voix, le délai de bout en bout et l'écho de cette dernière. Ces aspects sont influencés par leur paramétrage, l'architecture du réseau et les autres flux concurrents.

Au niveau du paramétrage de la voix, un aspect important est le codec défini. Ce codec ou paramètre d'échantillonnage est utilisé pour la compression et la décompression des flux. Cette notion définit la vitesse à laquelle la voix est échantillonnée et permet de dimensionner le flux de données numériques que va générer la transformation d'un échantillon temporel de voix analogique. Le choix du codec est un compromis entre la QoS que l'on veut attribuée au flux et la capacité réseau à fournir une bande passante.

Les différents codecs et leurs aspects:

Codec	Débit [kbps]	Paquet par secondes	Volume de données de voix [B]	Volume de données de voix [B]	Bande passante [kbps]	Intervalle d'échantil-lonnage [ms]	Délais d'échantil-lonnage [ms]
G.711 PCM	64	50	160	160	80	20	1
G.726 ADPCM	32	50	80	80	48	20	1
G.726 ADPCM	24	50	60	60	40	20	1
G.728 LD-CELP	16	33	40	40	26.7	20	25
G.729 CS-ACELP	8	50	20	20	40	20	25
G.723.1 MP-MLQ	6.3	33	24	24	17.1	30	67.5
G.723.1 ACELP	5.3	33	20	20	16	30	67.5

Figure 12 – Codec et leurs aspects

Nous pouvons voir que le codec G.711 permet de garantir la meilleure qualité de service. Les autres types de codecs définis précédemment apportent une diminution de la QoS.

La voix sur IP étant un flux très sensible, nous allons définir ces limites, pour le codec G.711, au niveau des 3 facteurs auxquels elle est sensible, qui sont pour rappel le délai, la gigue et le taux de pertes de données:

	Bon	Moyen	Mauvais
Délais [ms]	D < 150	150 < D < 400	400 < D
Gigue [ms]	G < 20	20 < G < 50	50 < G
Perte [%]	P < 1	1 < P < 3	3 < P

Figure 13 – Limites du codec G.711

2.4.2 Télévision IP

Le flux de trafic représentant la télévision sur IP peut être diffusé de 2 manières:

1. Unicast
2. Multicast

Ces dernières années, il y a eu une forte augmentation des flux vidéos diffusés de manière unicast, car il y a de plus en plus de services de "replay" et de VoD (Video on Demand). Pour tout ce qui est en "live", les flux vidéo diffusés de manière multicast sont utilisés. Les flux vidéo sur IP sont moins sensibles que les flux téléphoniques sur IP:

		Moyen
Délais [s]		D < 4-5
Gigue [ms]		-
Perte [%]		P < 5
Bande passante [Mbit/s]	**SD**	2-5
	HD	4-10

Figure 14 – Exigences du flux de trafic vidéo sur IP

2.4.3 Internet

Le flux de trafic représentant Internet représente une multitude de services et protocole différents, ces flux sont répartis avec différentes classes:

- **Best Effort Data**: pour répondre aux besoins de QoS, ce type de trafic doit être marqué avec le champ DSCP 0. Une bande passante adéquate doit être affectée à la classe Best Effort dans son ensemble, car la majorité des applications utilisent cette classe par défaut. Il est recommandé de réserver au moins 25% pour le trafic Best Effort. Cette classe représente tout le trafic non critique.
- **Bulk Data**: pour répondre aux besoins de QoS, ce type de trafic doit être marqué avec une classe PHB de AF11, voir AF12 ou AF13. La bande passante doit être garantie, mais ne doit pas dominer un lien. Cette classe est destinée aux opérations qui ne sont pas interactives et qui ne sont pas vraiment sensibles aux pertes, elle est destinée aux applications qui font leurs opérations sur une longue période en arrière-plan.
- **Transactional Data/Interactive Data**: pour répondre aux besoins de QoS, ce type de trafic doit être marqué avec une classe PHB de AF21, voir AF22 ou AF23. Une bande passante adéquate doit être attribuée pour les opérations interactives de premier plan. Cette classe combine 2 types d'applications similaires: les applications client/serveur (Transactional Data) et les applications de messagerie (Interactive Data).
- **Locally Defined Mission-Critical Data**: pour répondre aux besoins de QoS, ce type de trafic doit être marqué avec une classe PHB de AF31, voir AF32 ou AF33. Une bande passante suffisante doit être attribuée pour les opérations interactives de premier plan. Cette classe définit le trafic critique localement.

Classe	PHB	DSCP	Exemples
Best Effort Data	EF	32	trafic non critique, recherches HTTP Web
Transactional Data	AF11 (AF12-AF13)	10 (12-14)	FTP, Email, sauvegardes, synchronisation de bases de données, distribution de contenu, transfert de longs fichiers, tâches en arrière plan
Bulk Data	AF21 (AF22-AF23)	18 (20-22)	base de données, messages, partage de fichiers, réseaux sociaux
Interactive Data	AF21 (AF22-AF23)	18 (20-22)	telnet, messages instantanés, conférences
Locally Defined Mission-Critical Data	AF31 (AF32-AF33)	26 (28-30)	

Figure 15 – Classification des flux de trafic Internet

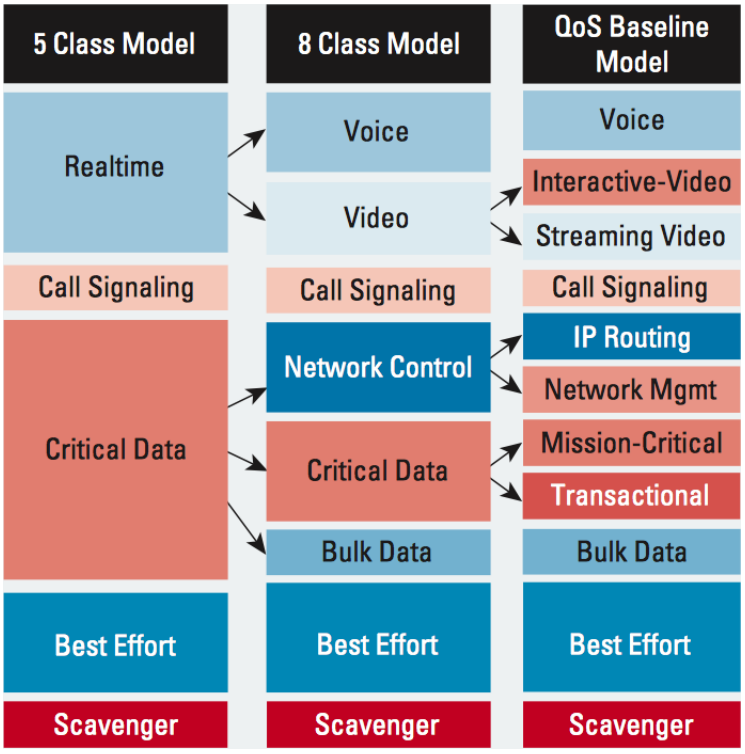
La majorité du trafic Internet est de type Best Effort et nous avons vu que pour cette classe, il est recommandé de réserver au moins 25% de la bande passante. Les autres classes n'ont pas d'exigences pour la bande passante, mais il faut penser à en réserver suffisamment pour que le flux ne soit pas interrompu.

2.4.4 Management

"Le management est un besoin vital pour assurer le bon fonctionnement d'un réseau. Le trafic généré par le management est faible. Il ne requiert pas d'allocation de bande passante particulière. En revanche, il est essentiel que ce flux puisse circuler en tout temps de manière à toujours avoir une vue globale de l'état du réseau." [1]

2.4.5 Synthèse

Cisco définit la ligne à suivre au niveau de la QoS, nous pouvons voir les différentes classes de flux répartis selon un modèle bien précis afin de les diviser en sous-classes, puis depuis ces sous-classes, l'attribution aux classes PHB et à une valeur DSCP:



Application	L3 Classification PHB	DSCP	Referencing Standard	Recommended Configuration
IP Routing	CS6	48	RFC 2474-4.2.2	Rate-Based Queuing + RED
Voice	EF	46	RFC 3246	RSVP Admission Control + Priority Queuing
Interactive-Video	AF41	34	RFC 2597	RSVP + Rate-Based Queuing + DSCP-WRED
Streaming Video	CS4	32	RFC 2474-4.2.2	RSVP + Rate-Based Queuing + RED
Mission-Critical	AF31	26	RFC 2597	Rate-Based Queuing + DSCP-WRED
Call-Signaling	CS3	24	RFC 2474-4.2.2	Rate-Based Queuing + RED
Transactional Data	AF21	18	RFC 2597	Rate-Based Queuing + DSCP-WRED
Network Mgmt	CS2	16	RFC 2474-4.2.2	Rate-Based Queuing + RED
Bulk Data	AF11	10	RFC 2597	Rate-Based Queuing + DSCP-WRED
Scavenger	CS1	8	Internet 2	No BW Guarantee + RED
Best Effort	0	0	RFC 2474-4.1	BW Guarantee Rate-Based Queuing + RED

Figure 16 – Classes de trafic et classification définies par Cisco [6]

2.5 Conception réseau

La conception d'un réseau permet de développer une infrastructure de services réseau performante, elle permet d'analyser les objectifs et de créer des stratégies de conception afin d'y répondre. Un réseau doit être conceptualisé de bout en bout.

Ce chapitre est basé sur la théorie du cours "Conception et exploitation des réseaux" [7].

2.5.1 Réseau Core

Le réseau Core permet le transport d'informations entre les grands points nodaux d'un réseau national, il est les autoroutes de l'information. Le réseau Core est interconnecté avec le réseau métro et avec les autres réseaux backbone nationaux, étrangers et internationaux, et avec les grands Datacenters, et les serveurs d'applications clefs. En cas de panne de réseau, la disponibilité du Core est essentielle. Le réseau Core est transparent à tous les services:

- Les transporter de manière encapsulée
- Les traiter dans la mesure du possible sans limitation
- Avoir suffisamment de bande passante
- Utiliser de la qualité de services

Notre projet est basé sur le réseau Core, nous définirons ce point dans la partie conception.

2.5.2 Topologies réseaux physiques

Le choix de la topologie réseau est important, car une fois cette dernière mise en place par les opérateurs réseau, elle ne peut pas être changée, il faudrait tout détruire et reconstruire.

Nous allons analyser les 6 topologies suivantes possédant 12 noeuds terminaux, ces analyses sont basées sur un système de trafic "any-to-any".

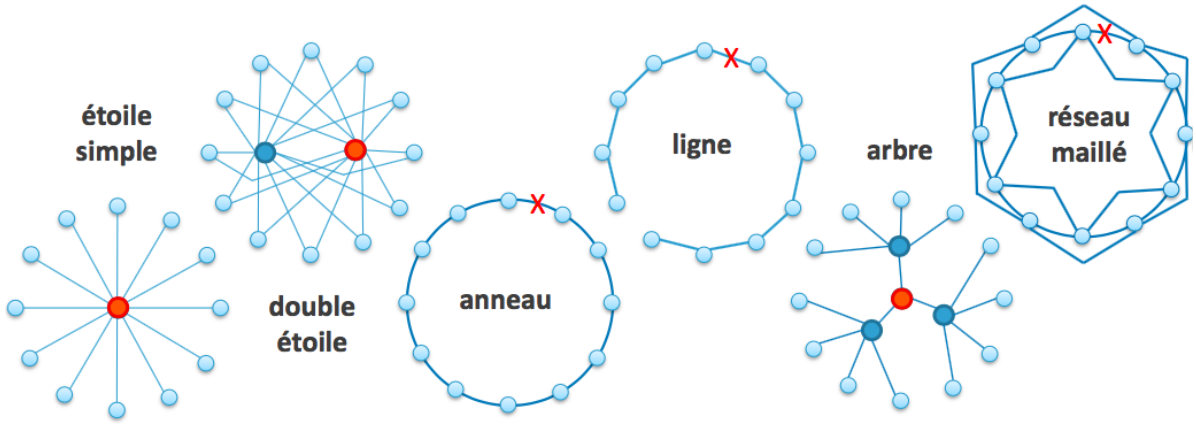


Figure 17 – 6 topologies réseaux [7]

Topologie	Noeuds	Liens	Longueur	Bonds	Fiabilité	Prix	Qualité
Etoile simple	13	12	12	2	non	24	2
Double étoile	14	24	29	2	oui	53	10
Anneau	12	12	6.28	3.3	oui	18.3	4
Ligne	12	11	5.8	4.3	non	16.8	2
Arbre	16	15	10	3.6	non	25	2
Réseau maillé	12	24	26	2	oui	50	10

Figure 18 – Comparaison des topologies [7]

Dans ce tableau, nous pouvons voir que certaines topologies de réseau ne sont vraiment pas fiables, les topologies en étoile simple, en ligne et en arbre sont délicates, car lorsqu'un lien tombe, le réseau est plombé, il est vraiment déconseillé de mettre en place ce type de réseau.

La conception d'un réseau est indispensable afin de calculer la capacité totale du réseau et de calculer son coût total. Avec ces 2 points, nous pouvons calculer le coût du réseau par capacité.

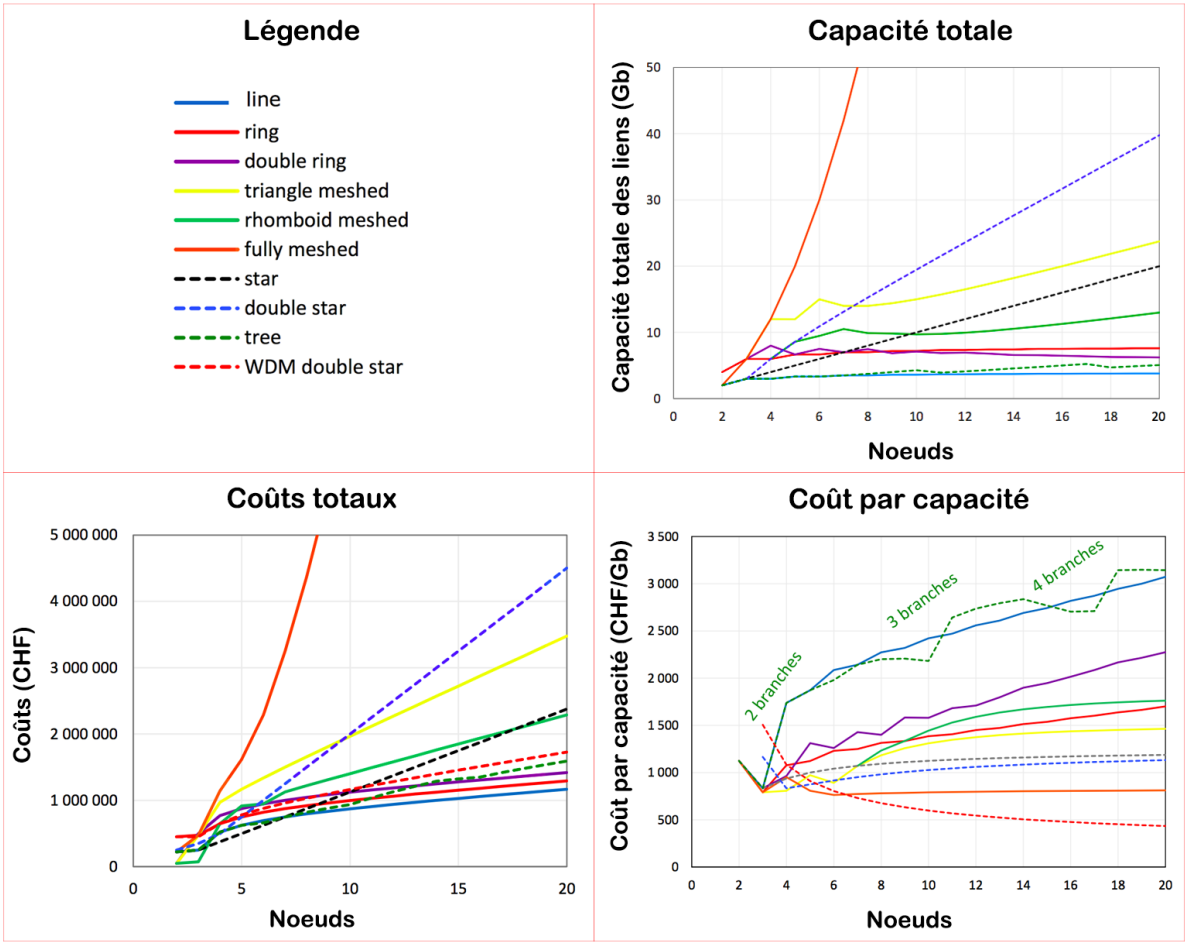


Figure 19 – Coûts et capacités des réseaux basés sur une communication "any-to-any" [7]

Ces calculs sont basés sur des flux "any-to-any", comme un trafic téléphonique sur IP. Ces tableaux comparatifs, comme présentés ci-dessus, doivent être mis en place et analyser lors du choix de la topologie réseau. Ces tableaux se basent sur la capacité et les coûts, ces 2 critères sont très importants lors de la conception du réseau. Bien entendu, on ne parle ici que de 2 critères, d'autres critères vont être définis et analysés, mais avec ces 2 critères, nous pouvons déjà avoir une bonne vue d'ensemble, et effecteur un premier tri des topologies réseaux.

Comme nous pouvons le voir, plus la capacité de la topologie est élevée, plus le coût est élevé, comme pour la topologie complètement maillée. Au niveau du rapport de coût par capacité, nous pouvons voir que les topologies en arbre ou en ligne ne sont pas vraiment intéressantes pour un trafic "any-to-any". C'est intéressant, car nous pouvons voir que la topologie en anneau a une meilleure capacité totale et un meilleur coût que la topologie en double anneau, cela vient du fait que le lien fermant le double anneau devient rapidement saturé. Pour des connexions "any-to-any", les topologies ayant le meilleur rapport de coût par capacité, en réduisant au maximum ce coût, sont les topologies en double étoile et en étoile simple, bien entendu la topologie complètement maillée est la plus adéquate, mais les coûts explosent.

Nous avons vu, précédemment, une évaluation des topologies pour des transferts de flux "any-to-any". Nous allons maintenant voir ces mêmes topologies, mais évaluer en fonction d'un trafic "datacentric" afin d'analyser les différences.

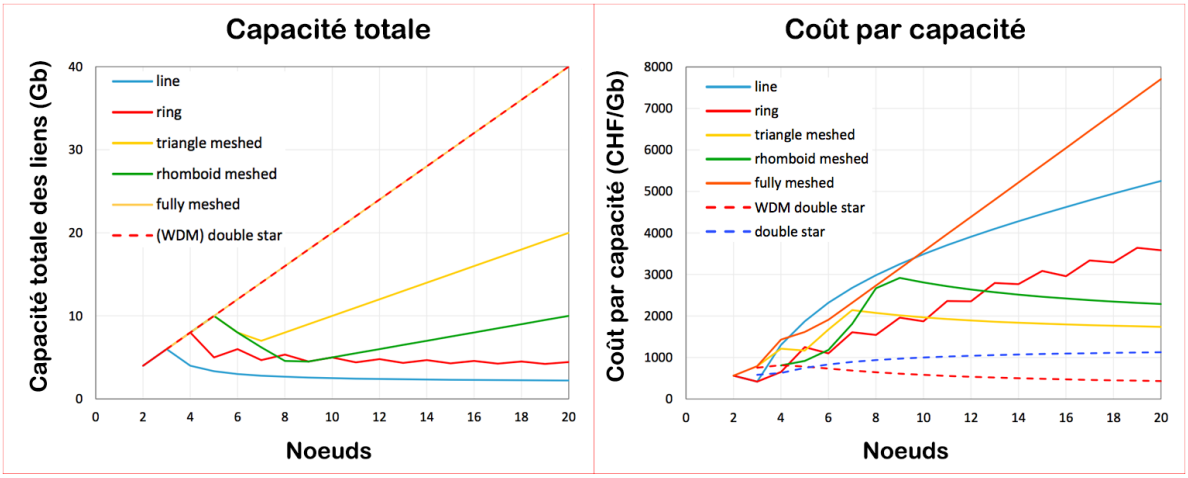


Figure 20 – Coûts et capacité des réseaux basés sur une communication "datacentric" [7]

Ces calculs sont basés sur des flux "datacentric", comme un trafic Internet ou TV sur IP. Ces mesures ont été réalisées en admettant que 2 datacenters envoient du trafic vers tous les autres noeuds.

Ici aussi, des topologies comme la topologie en ligne ou en arbres ne sont pas intéressantes. Pour ce cas, la topologie la plus intéressante pour les opérateurs est celle en double étoile. La topologie en double étoile possède la plus grande capacité totale et le coût par capacité le moins élevé. En effet, il s'agit du meilleur compromis, les 2 noeuds centraux génèrent le trafic qui est envoyé vers les noeuds directement reliés, le nombre de bonds est minimum et la redondance est garantie. Bien entendu, cette mise en place à un certain coût.

2.5.3 Critères de sélection des topologies réseau

Il n'existe pas de meilleure topologie réseau, tout dépend des besoins. Pour un trafic de type téléphonique, la topologie entièrement maillée est la meilleure topologie, car le trafic va de chaque noeud vers tous les autres noeuds. Pour un trafic de type Internet ou télévision (multicast ou unicast), la topologie en étoile est la meilleure topologie, car tout est géré depuis un noeud central. Bien entendu, au niveau de la fiabilité, la topologie entièrement maillée est la meilleure topologie.

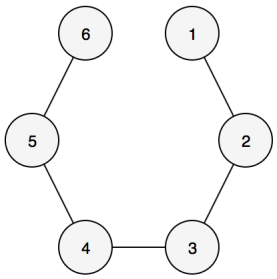
Comme nous l'avons expliqué précédemment, lors de la conception du réseau, des critères doivent être définis afin de sélectionner au mieux la topologie à mettre en place. Nous avons défini un certain nombre de critères, en plus de ceux vus précédemment:

- Coûts
- Capacité
- Coûts par capacité
- Redondance
- Nombre moyen de bonds
- Bond maximum
- Nombre de noeuds
- Nombre de liens
- Type de flux
- Choix des opérateurs (Core)

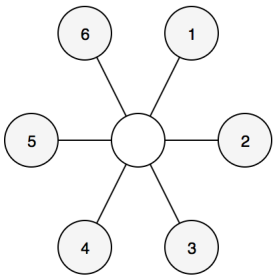
À noter aussi qu'il ne faut pas confondre la topologie réseau physique, qui définit la structure physique du réseau, et la topologie réseau logique, qui définit de quelle manière se passe la communication dans la topologie physique.

D'autres topologies que celles présentées peuvent être intéressantes. Nous avons repris celles précédemment définies et d'autres, potentiellement intéressantes. Nous nous sommes basées sur 6 noeuds afin de les définir.

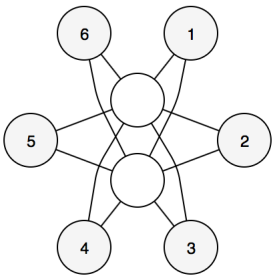
Ligne



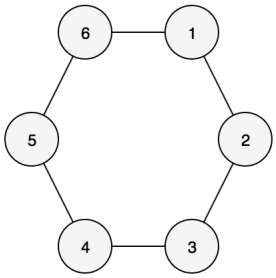
Etoile simple



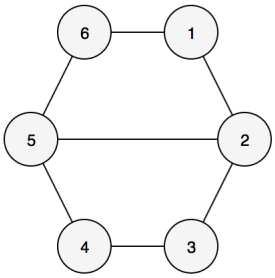
Etoile double



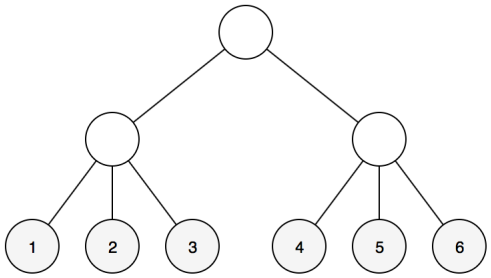
Anneau



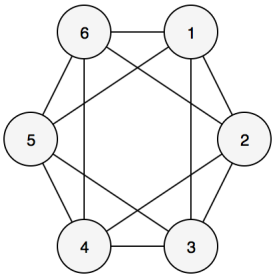
Double anneau



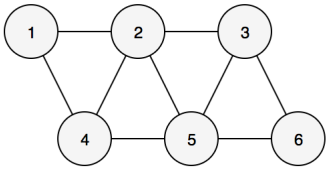
Arbre



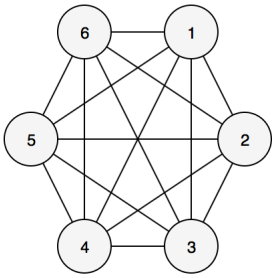
Maillé



Maillé en triangle



Entièrement maillé



Une fois les topologies présentées, nous nous basons sur chacun des critères définis précédemment afin d'en analyser les topologies grâce à une grille d'évaluation.

Topologie	Coûts	Capacité	Coûts par capacité	Redondance	Nombre de bonds moyens	Bond maximum	Nombre de noeuds	Nombre de liens	Choix des opérateurs (Core)	Flux de trafic		
										Voix	TV	Internet
Ligne	++	-	-	-	-	- (5)	++ (6)	++ (5)	-	-	-	-
Etoile simple	+	+	++	-	++	++ (2)	+ (7)	++ (6)	-	-	-	-
Double étoile	-	++	++	+	++	++ (2)	- (8)	+ (12)	-		+	+
Anneau	++	-	+	+	+	+ (3)	++ (6)	++ (6)	+			
Double anneau	++	-	+	++	+	+ (3)	++ (6)	++ (7)	+			
Arbre	-	-	-	-	-	- (4)	- (9)	+ (8)	-	-	-	-
Maillé	-	++	++	++	++	++ (2)	++ (6)	+ (12)	+			
Maillé en triangle	+	+	++	++	+	+ (3)	++ (6)	+ (9)	+			
Entièrement maillé	-	++	++	++	++	++ (1)	++ (6)	- (15)	-	+		

Figure 21 – Topologies et critères

Comme nous l'avons déjà vu, des topologies en ligne, en arbre ou entièrement maillées ne sont pas intéressantes pour les réseaux de coeur, d'après nos analyses. D'après les critères définis et analysés, des topologies comme le double anneau, la double étoile, partiellement maillé ou maillé en triangle sont plus intéressantes pour les opérateurs de réseau de coeur. Il faut trouver un juste milieu afin de respecter au mieux les critères. À noter que les topologies les plus intéressantes par rapport aux différents flux, ne sont pas forcément toujours celles mises en place.

Nous avons défini des critères techniques, mais d'autres éléments importants ne sont pas à négliger lors de la conception d'un réseau, comme la géographie de la zone où l'on veut mettre en place le réseau. La topologie réseau physique est influencée par le terrain. En effet, des barrières naturelles comme des lacs où des montagnes peuvent influencer sur la conception de la topologie. Les noeuds principaux d'un réseau de coeur doivent aussi être positionnés en fonction des agglomérations en fonction du nombre d'habitants. Il ne faut pas s'arrêter aux éléments techniques, mais voir plus large.

C'est toujours intéressant de prendre un cas concret, si nous prenons l'exemple de Switch, son coeur de réseau, appelé SWITCHlan, recouvre l'ensemble du territoire Suisse:

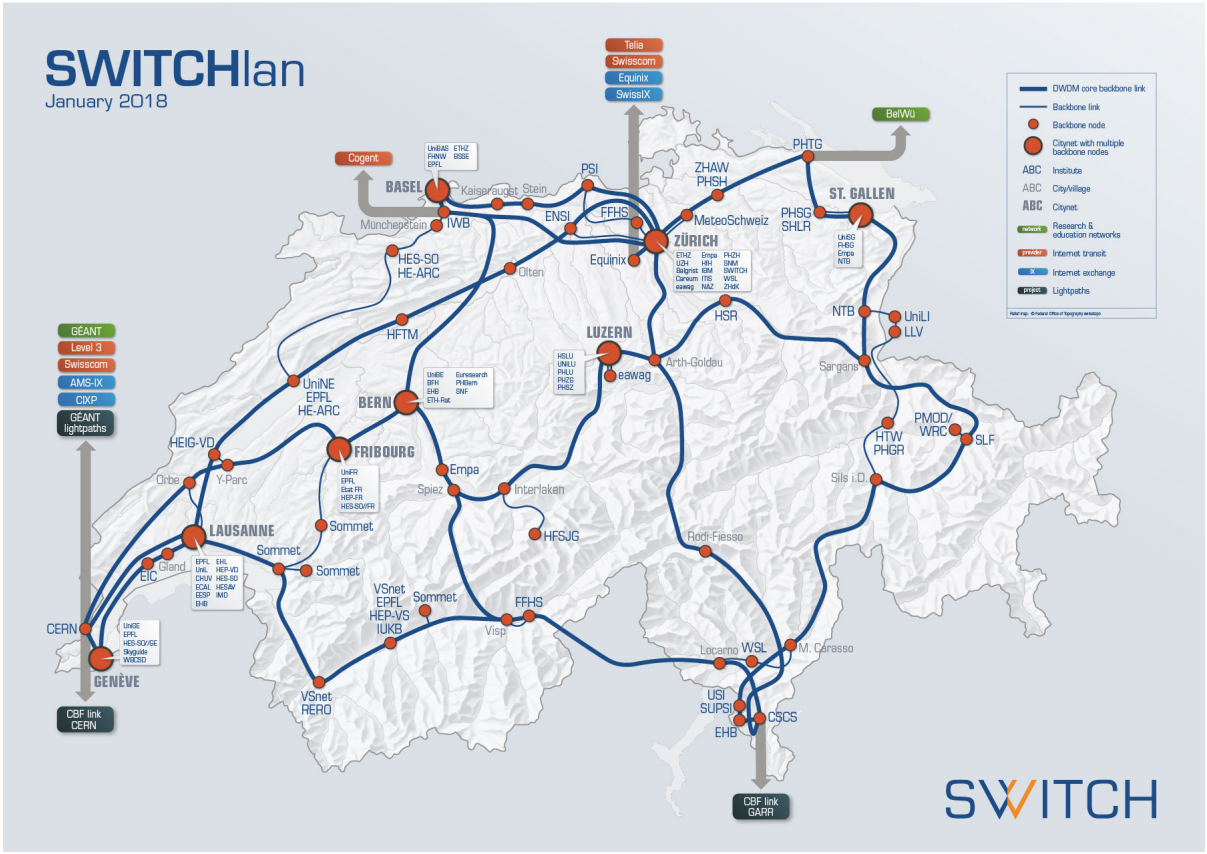


Figure 22 – Coeur de réseau Switch en 2018 [8]

Nous ne pouvons pas dire précisément de quelle topologie réseau il s'agit. En effet, le coeur de réseau a été construit en fonction de la demande. Nous pouvons voir que les principaux noeuds (Citynet with multiple backbone nodes) se trouvent répartis dans les plus grandes agglomérations suisses.

Si nous analysons ce réseau, nous pouvons voir qu'une association de plusieurs topologies réseau constitue le réseau de coeur Switch. En effet, il s'agit d'un réseau partiellement maillé composé de plusieurs anneaux ou doubles anneaux. Ce réseau d'une longueur d'environ 2900 km a été pensé de manière à garantir une bonne redondance et une haute qualité de transmission, un haut niveau de stabilité et de disponibilité et de bonnes liaisons vers les centres internationaux de recherche, des liaisons Internet par 4 fournisseurs différents et de nombreux points de Peering. C'est intéressant, car si nous comparons ce réseau au système autoroutier suisse, chacun des liens de fibres optiques peut se référer à une autoroute. Ce qui confirme nos dires, les principales agglomérations sont reliées sans respecter une topologie bien précise.

Il est intéressant aussi de noter que le réseau SWITCHlan, lors de sa création en 1989, possédait une bande passante limitée à 128 kbit/s, ce qui suffisait amplement pour l'époque. Et actuellement, la bande passante est de 100 Gbit/s, ce qui suffit aussi amplement aux besoins. En 1995, ce réseau était totalement différent.

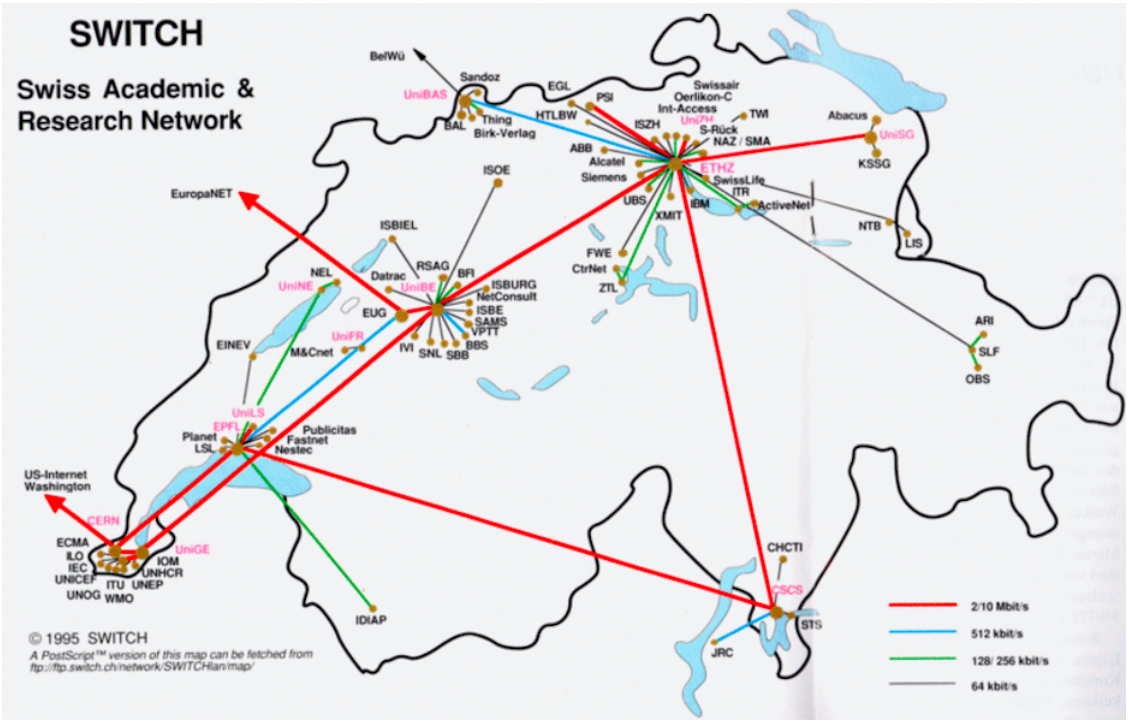


Figure 23 – Coeur de réseau Switch en 1995 [9]

Nous pouvons noter l'évolution du réseau SWITCHlan, pensé et conçu pour évoluer et durer. À ces débuts ce réseau était un simple anneau reliant Lausanne, Genève, Berne, Zurich, Lugano et retour sur Lausanne, mais pour répondre aux besoins et dans une optique d'évolutivité, d'autres anneaux ont été construits autour de cette base. Nous pouvons voir un détail intéressant, le fait qu'en 1995, une ligne directe entre le CERN et Washington était déjà existante.

2.5.4 Conclusion de l'analyse

Cette partie d'analyse nous a, tout d'abord, permis de bien comprendre les bases et le fonctionnement des méthodes de QoS. Ce domaine restait très vague, le fait d'avoir eu l'occasion d'étudier cette matière est un point très positif.

Premièrement, grâce à l'état de l'art, nous avons pu analyser les différents produits ou services permettant de simuler et de tester un réseau, ce qui nous donne une idée du type de service ou de produit à proposer pour la partie économique de ce travail.

Nous avons analysé les différentes étapes de la mise en place de la QoS, la définition des différents types de flux de trafic à mettre en place. Ces deux points permettent de définir les méthodes de QoS à mettre en place sur le réseau de tests, de la classification des paquets en fonction des différents types de flux de trafic aux types d'ordonnancement à mettre en place sur les équipements du réseau, nous avons de bonnes bases afin de définir la conception de ce projet.

Avec toutes ces notions, nous sommes prêts à définir dans la partie conception du projet, la classification des paquets à mettre en place lors de leur génération au niveau du Spirent et les méthodes d'ordonnancement à mettre en place sur les routeurs.

Une autre partie de cette analyse était l'étude des différentes topologies réseau, différents critères ont été définis afin de choisir au mieux cette topologie réseau. Ces critères sont importants, car nous avons pu voir que si certains critères étaient remplis au mieux, cela pouvait peser sur certains autres critères. Par exemple, une topologie réseau très fiable, à un coût très élevé, ce qui n'est pas forcement la meilleure solution. Il faut donc peser le pour et les contre, afin de trouver un équilibre entre ces critères pour que ces derniers soient tous remplis au mieux. Bien entendu, ces critères constituent une bonne base, mais lors de la conception d'un réseau de coeur, il faut aussi prendre en compte les aspects démographiques et géographies afin de garantir une bonne qualité de réseau.

À noter aussi que les topologies réseau définies comme étant les plus fiables pour un type de trafic, ne sont pas forcément celles mises en place. En effet, il faut trouver un compromis entre les différents flux afin de mettre en place une topologie réseau et non une topologie réseau pour chacun des types de flux. La topologie réseau la plus fiable peut aussi peser sur certains autres critères, voilà aussi pourquoi la topologie la plus fiable n'est pas forcément celle mise en place, tout à un coût.

Comme nous avons pu le voir avec le réseau SWITCHlan, lors de la conception d'un réseau, ce dernier doit être pensé pour évoluer et durer. Lors de la conception d'un réseau, les opérateurs ne se basent pas sur une topologie précise, le réseau est adapté en fonction des besoins et des demandes.

3 Conception

Ce chapitre traite de la conception de notre projet. Cette conception permettra aux superviseurs ou aux personnes qui reprendront le projet de bien comprendre son fonctionnement et la manière dont il a été pensé.

Nous commençons cette partie de conception avec le choix du modèle d'architecture qui constitue la base de notre infrastructure. Puis, nous définissons la génération des flux de trafic sur le Spirent qui sont Internet, la télévision, la téléphonie et management de réseau. Afin que ces flux de trafic puissent être générés et injectés sur un réseau de tests, nous définissons les topologies réseau à mettre en place en fonction de l'analyse précédemment réalisée. Pour finir, nous définissons quel type de QoS nous mettons en place sur les équipements du réseau. Tous ces éléments définissent notre scénario de tests.

3.1 Modèle d'architecture

Pour le choix du modèle d'architecture, nous suivons la même direction que dans le projet de semestre 5 de Monsieur Simon Lièvre "QoSLab" [1]. Nous travaillons avec le modèle d'architecture DiffServ, pour les raisons suivantes:

- Plus adaptée aux grands réseaux comme internet
- Souvent mise en oeuvre chez les opérateurs, popularité
- Norme plus récente
- Peut fonctionner en parallèle avec MPLS et les VLANs

Comme nous l'avons déjà défini, nous nous basons sur le coeur du réseau. Dans ce travail, nous n'aborderons pas les notions de MPLS.

3.2 Réseaux de tests

Nous allons définir les réseaux de tests que nous allons mettre en place afin de réaliser ce projet.

Ce travail se divise en plusieurs étapes, tout d'abord, la mise en place de deux réseaux qui permettront la prise en main des instruments et la mise en place d'une méthodologie de tests. Cette première étape permettra aussi de comprendre la mise en place de la QoS sur les routeurs et la génération du trafic sur le Spirent.

Les étapes suivantes permettent de pousser les tests afin de mieux comprendre certains phénomènes dans les réseaux et de mettre en place un réseau de tests plus conséquent.

Il est important de noter qu'il faut connecter le routeur et le Spirent, et les routeurs entre eux, avec des câbles croisés pour que la négociation entre ces équipements se passe correctement.

3.2.1 Étape 1

Dans un premier temps, nous réalisons deux réseaux de tests afin de valider quelques hypothèses, comme le fait qu'un réseau de tests comportant un routeur crée un délai de 20 millisecondes et, logiquement, un réseau de tests comportant trois routeurs crée un délai de 60 millisecondes. Cette étape permet aussi la prise en main des instruments comme le Spirent. Nous avons pu voir que dans le projet de semestre 5 de Monsieur Simon Lièvre "QoSLab" [1], ce dernier avait réalisé un réseau comportant un switch (Layer 2 et 3), dans notre réseau, nous nous concentrons sur le Layer 3, donc nous n'y mettrons pas de switch, la latence du switch étant négligeable.

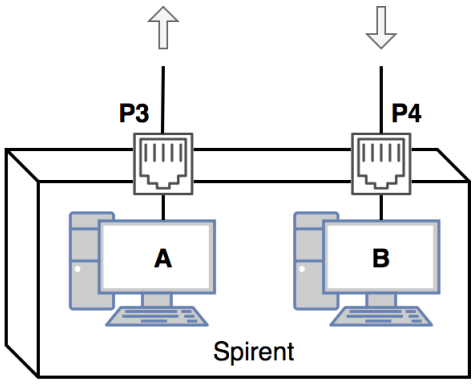


Figure 24 – Hôtes virtuels sur le Spirent

Sur le Spirent, nous définissons 2 hôtes virtuels, A (sur le port 3 du Spirent) et B (sur le port 4 du Spirent), ces hôtes virtuels sont placés dans 2 sous-réseaux distincts. Les flux sont générés depuis l'hôte virtuel A et réceptionnés par l'hôte virtuel B. Pour la phase de tests, nous jouerons avec le débit des liens sur le routeur, par exemple en bridant le lien de sortie du routeur à destination de l'hôte B afin d'analyser le comportement du réseau sous certaines conditions.

Les deux topologies définies ci-dessous permettent d'injecter au maximum 100% de trafic généré dans le réseau, nous reviendrons sur ce phénomène plus tard, dans la partie de génération des flux de trafic. À noter que dans notre projet, 100% de trafic est égal à 100Mb/s de trafic.

Topologie 1

La première topologie est composée du Spirent et d'un routeur Cisco 2800 Series, cette topologie réseau permet de mettre en place l'infrastructure du projet de semestre 5 de Monsieur Simon Lièvre "QoSLab" [1].

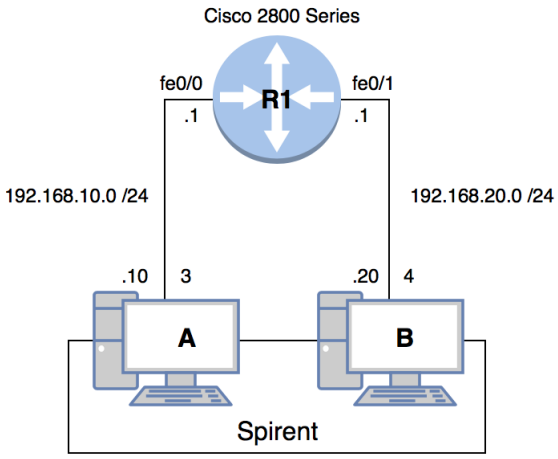


Figure 25 – Étape 1: réseau physique 1

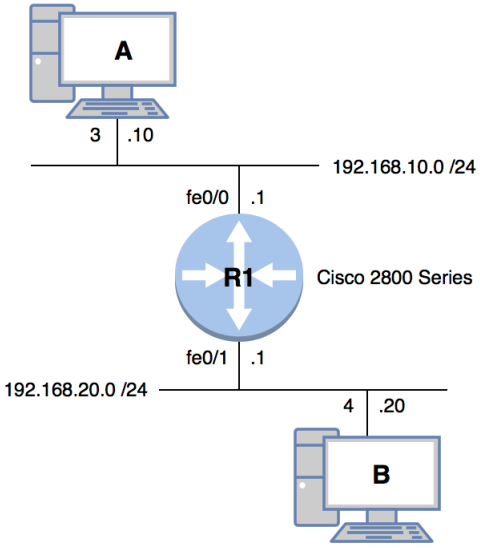


Figure 26 – Étape 1: réseau logique 1

Topologie 2

La seconde topologie est composée du Spirent et de 3 routeurs Cisco 2800 Series en ligne.

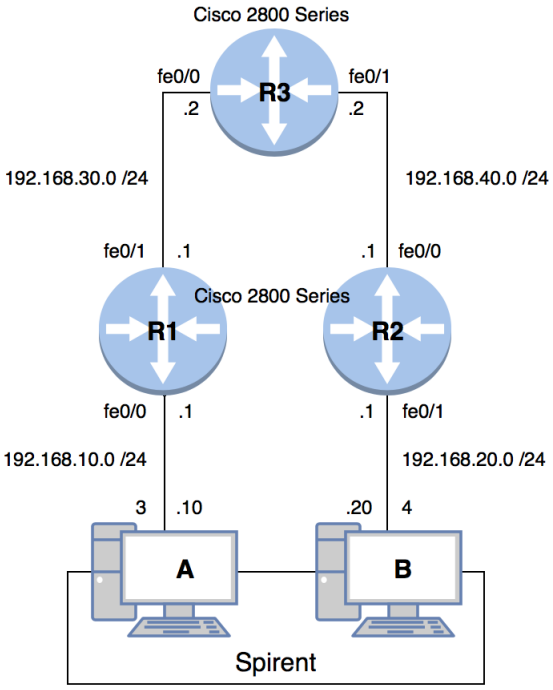


Figure 27 – Étape 1: réseau physique 2

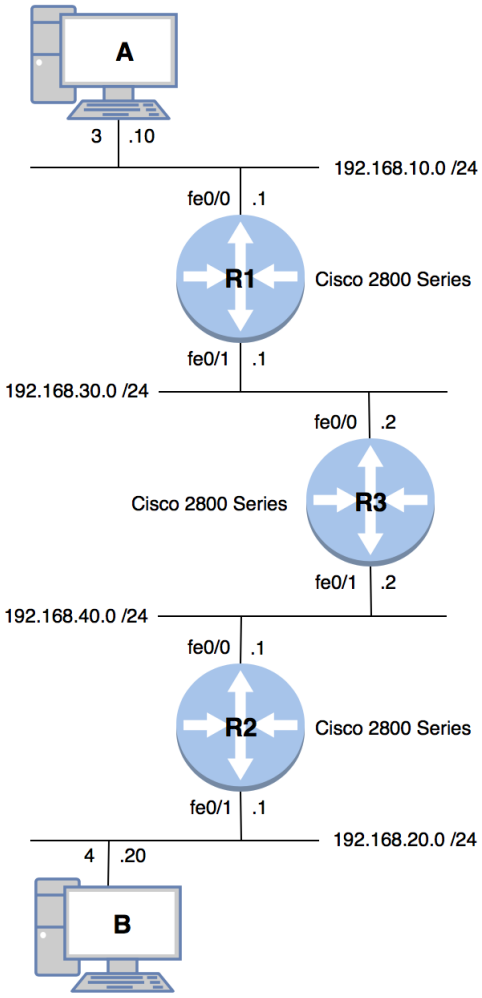


Figure 28 – Étape 1: réseau logique 2

3.2.2 Étape 2

Pour cette deuxième étape, nous réalisons deux réseaux de tests suivant la même logique que les deux réseaux mis en place précédemment. Comme nous avons pu le voir précédemment, dans les deux réseaux décrits, nous pouvions injecter au maximum 100% de trafic généré dans le réseau. Ces deux nouveaux réseaux permettent d'ajouter entre le Spirent et le routeur R1 afin d'injecter plus de 100% de trafic dans le réseau et au maximum 200% (200Mb/s), pour en analyser son comportement.

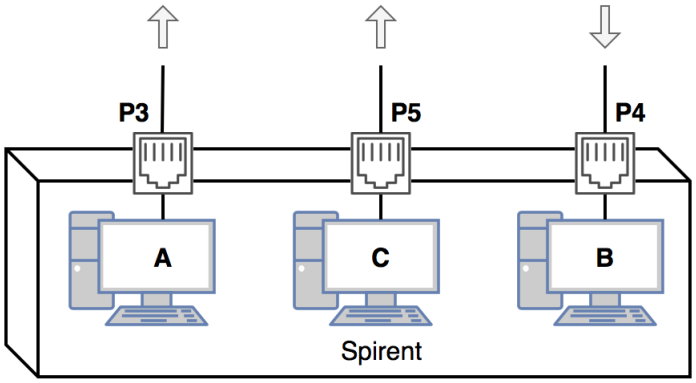


Figure 29 – Hôtes virtuels sur le Spirent

Sur le Spirent, nous définissons 3 hôtes virtuels, A (sur le port 3 du Spirent), B (sur le port 4 du Spirent) et C (sur le port 5 du Spirent), ces hôtes virtuels sont placés dans 3 sous-réseaux distincts. Les flux sont générés depuis les hôtes virtuels A et C, et réceptionnés par l'hôte virtuel B. Pour la phase de tests, nous jouerons avec le débit des liens sur le routeur, par exemple en bridant le lien de sortie du routeur à destination de l'hôte B afin d'analyser le comportement du réseau sous certaines conditions.

Afin de pouvoir connecter 3 lignes sur un routeur, nous devons remplacer un routeur Cisco 2800 Series (2 ports) avec un routeur Cisco 2900 Series (3 ports).

Topologie 3

La troisième topologie est composée du Spirent et de 1 routeur Cisco 2900 Series.

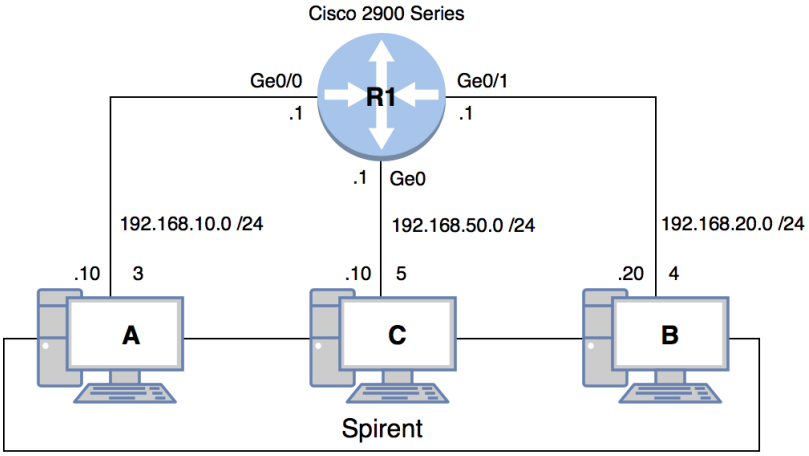


Figure 30 – Étape 2: réseau physique 3

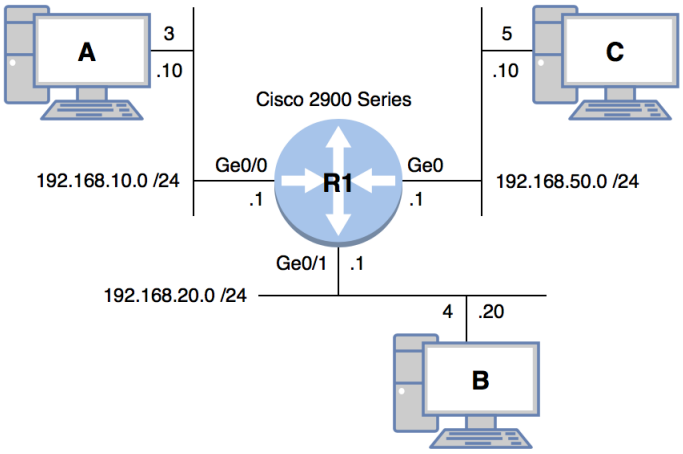


Figure 31 – Étape 2: réseau logique 3

3.2.3 Étape 3

Cette troisième étape permet la réalisation de tests sur un réseau plus conséquent. Nous réaliserons des tests afin de tester toutes nos stratégies de QoS au niveau des files d'attente. Ce réseau permet de simuler un réseau de laboratoires. Des switchs et des VLANs seront inclus dans ce réseau de tests.

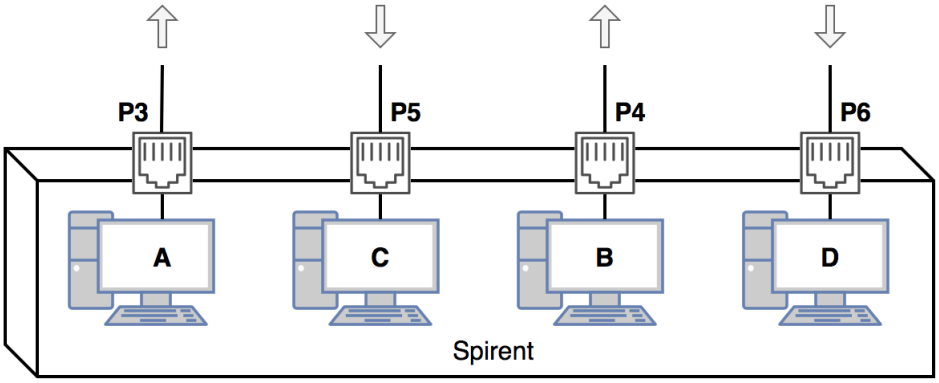


Figure 32 – Hôtes virtuels sur le Spirent

Sur le Spirent, nous définissons 4 hôtes virtuels, A (sur le port 3 du Spirent), B (sur le port 4 du Spirent), C (sur le port 5 du Spirent) et D (sur le port 6 du Spirent), ces derniers sont placés dans 4 sous-réseaux distincts. Les flux sont générés depuis les hôtes virtuels A et B, et réceptionnés par les hôtes virtuels D pour le trafic en provenance de A, et C pour le trafic en provenance de B. Pour la phase de tests, nous jouerons avec le débit des liens sur le routeur, par exemple en bridant les liens à l'intérieur du réseau de tests.

Afin de pouvoir connecter 4 lignes sur un routeur, nous devons ajouter au réseau des switchs Cisco Catalyst 3560-CG Series et définir des VLANs dans le réseau.

Topologie 4

Cette quatrième topologie est composée du Spirent, de deux routeurs Cisco 2800 Series, de deux routeurs Cisco 2900 Series et de deux switchs Cisco Catalyst 3560-CG Series.

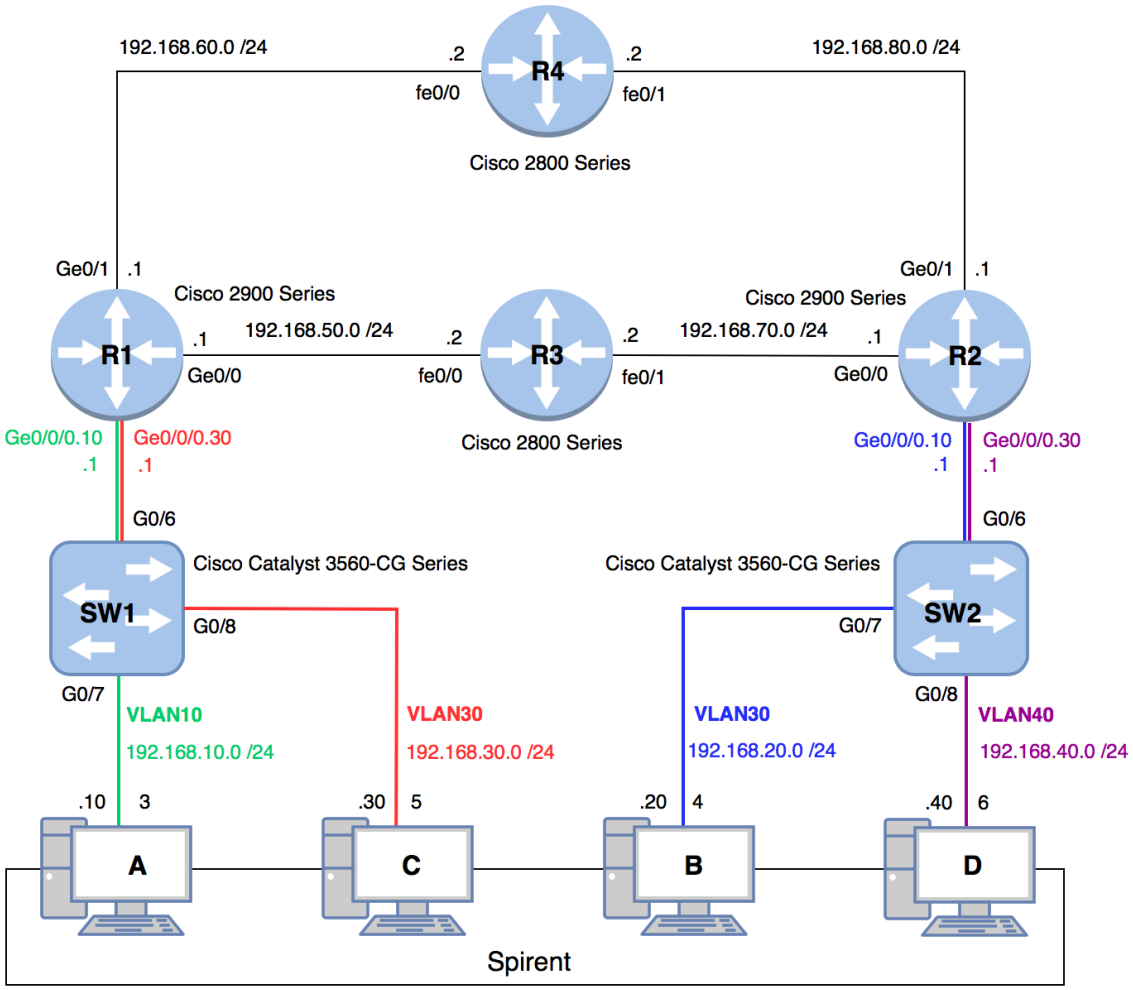


Figure 33 – Étape 3: réseau physique 4

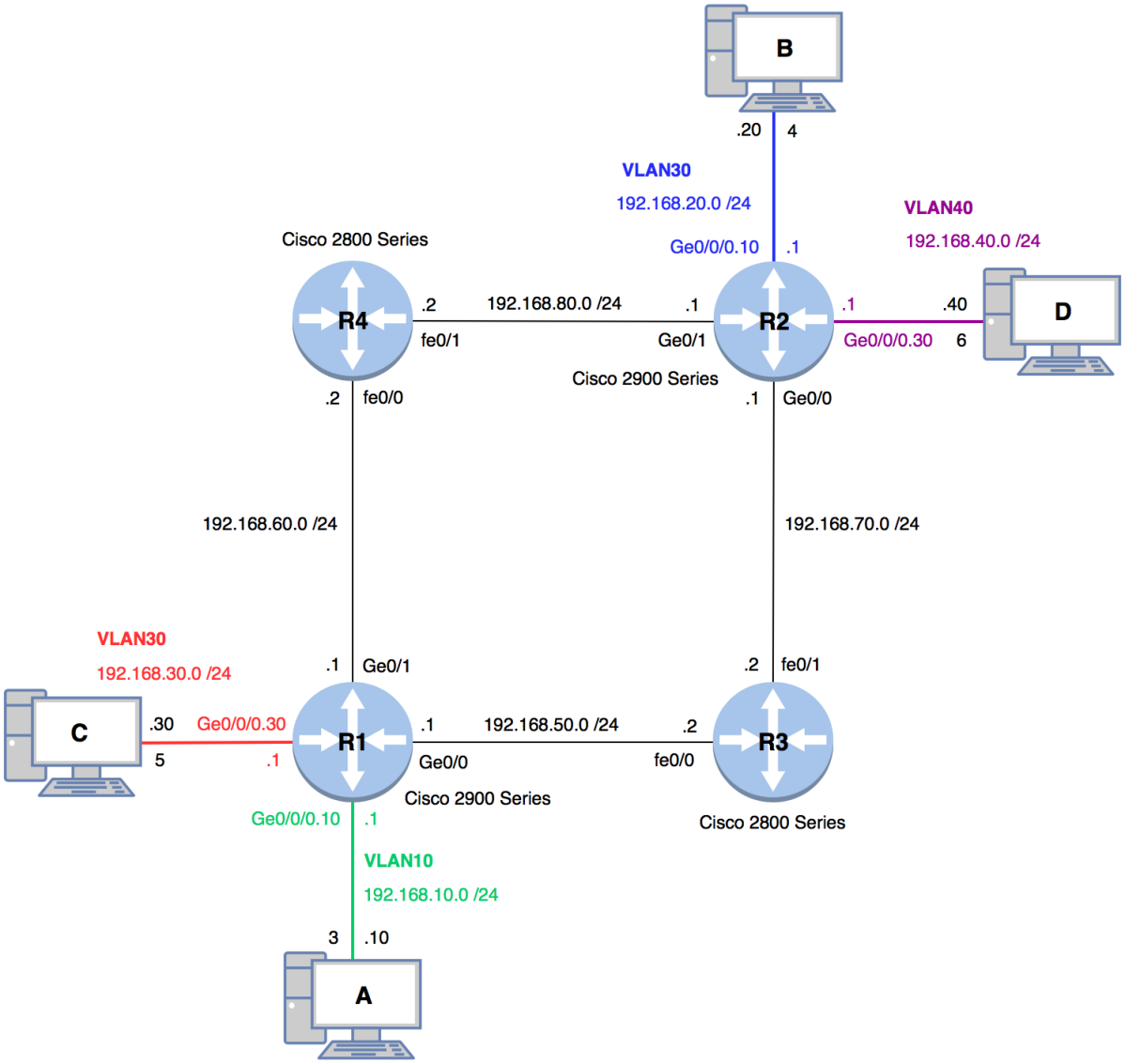


Figure 34 – Étape 3: réseau logique 4

3.2.4 Étape 4

Cette quatrième étape suit la logique de l'étape 3, les topologies réseau sont les mêmes, la différence est que les flux ne sont pas générés et réceptionnés par les mêmes hôtes virtuels.

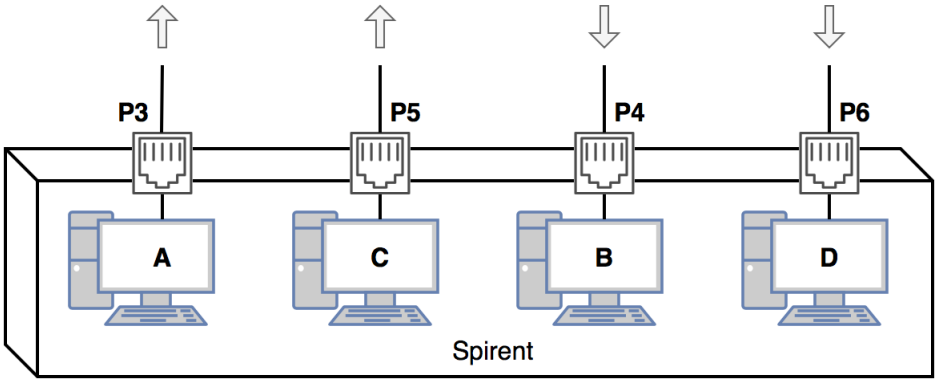


Figure 35 – Hôtes virtuels sur le Spirent

Sur le Spirent, nous définissons 4 hôtes virtuels, A (sur le port 3 du Spirent), B (sur le port 4 du Spirent), C (sur le port 5 du Spirent) et D (sur le port 6 du Spirent), ces derniers sont placés dans 4 sous-réseaux distincts. Les flux sont générés depuis les hôtes virtuels A et C, et réceptionnés par les hôtes virtuels D pour le trafic en provenance de A, et B pour le trafic en provenance de C. Pour la phase de tests, nous jouerons avec le débit des liens sur le routeur, par exemple en bridant les liens à l'intérieur du réseau de tests.

Afin de pouvoir connecter 4 lignes sur un routeur, nous devons ajouter au réseau des switchs Cisco Catalyst 3560-CG Series et définir des VLANs dans le réseau.

Les topologies réseaux physiques et logiques sont les mêmes que dans l'étape précédente, veuillez vous référer au point 3.2.3.

3.3 Génération des flux de trafic

Comme nous l'avons défini dans le cahier des charges, la définition des flux de trafic est basée sur le projet Monsieur Simon Lièvre "QoSLab" [1], nous reprenons, pour ce projet, ses informations. Nous ne nous penchons donc pas sur l'analyse des flux de trafic, pour des informations plus détaillées sur les flux de trafic, veuillez-vous référer au travail de Bachelor de Monsieur Gabriel Python "Cloud Topology for multiple services" [10].

Les flux de trafic généré par le Spirent seront les suivants:

Trafic	Transport	DSCP	QoS Byte	DSCP (Hex)	Volume [Mbit/s]
Téléphonie	UDP	EF	B8	2E	1
Télévision multicast	UDP	AF41	88	22	1
Télévision unicast	TCP	AF41	88	22	56
Internet	TCP	AF13	38	0E	41
Mangement	TCP	CS6	C0	30	1

Figure 36 – Génération des flux de trafic

Ces informations sont basées sur les résultats de l'opérateur Swisscom qui représentent la majeure partie du trafic Internet suisse. Pour la classification des flux, ces valeurs sont proposées par Cisco.

Les valeurs correspondant au volume représentent au total une génération de trafic à 100% (100% = 100Mb/s). Durant le projet nous modifierons ces valeurs afin de générer du trafic à différents pourcentages. À noter que nous ne pouvons pas générer plus de 100% de trafic sur une ligne avec le Spirent car la ligne tombe. Pour générer notre flux de trafic à plus de 100%, nous utiliserons 2 lignes afin d'injecter le trafic dans le réseau.

Nous nous focalisons uniquement sur le coeur de réseau, c'est pourquoi les paquets sont marqués lors de leur génération dans le Spirent, car cet instrument réalise ce mécanisme. Ce qui nous permet d'analyser plus en détail les types de files d'attente et leurs effets sur les paquets.

La configuration du Spirent au niveau des hôtes virtuels et de la génération du trafic sera décrite dans la partie réalisation.

3.4 Choix de la QoS

Nous mettrons en place 3 types de files d'attente QoS sur les routeurs, CBWFQ, LLQ et PQ. Ces 3 types de files d'attente sont les plus adaptés d'après notre analyse. Nous savons d'après l'analyse que la file d'attente FIFO (défaut) ne répond pas aux différentes attentes de QoS, nous mettrons quand même en place cette file d'attente pour des tests concernant la latence. Les types de files d'attente FIFO, CBWFQ, LLQ, PQ, WFQ et CQ gèrent les congestions sur le routeur Cisco 2800 Series, contrairement aux types de files d'attente DCBWFQ, WF2Q, HWFQ, WRR, DRR et SRR qui ne gèrent pas les congestions sur le routeur Cisco 2800 Series.

3.5 Scénarios de tests

Le scénario de tests général est le suivant:

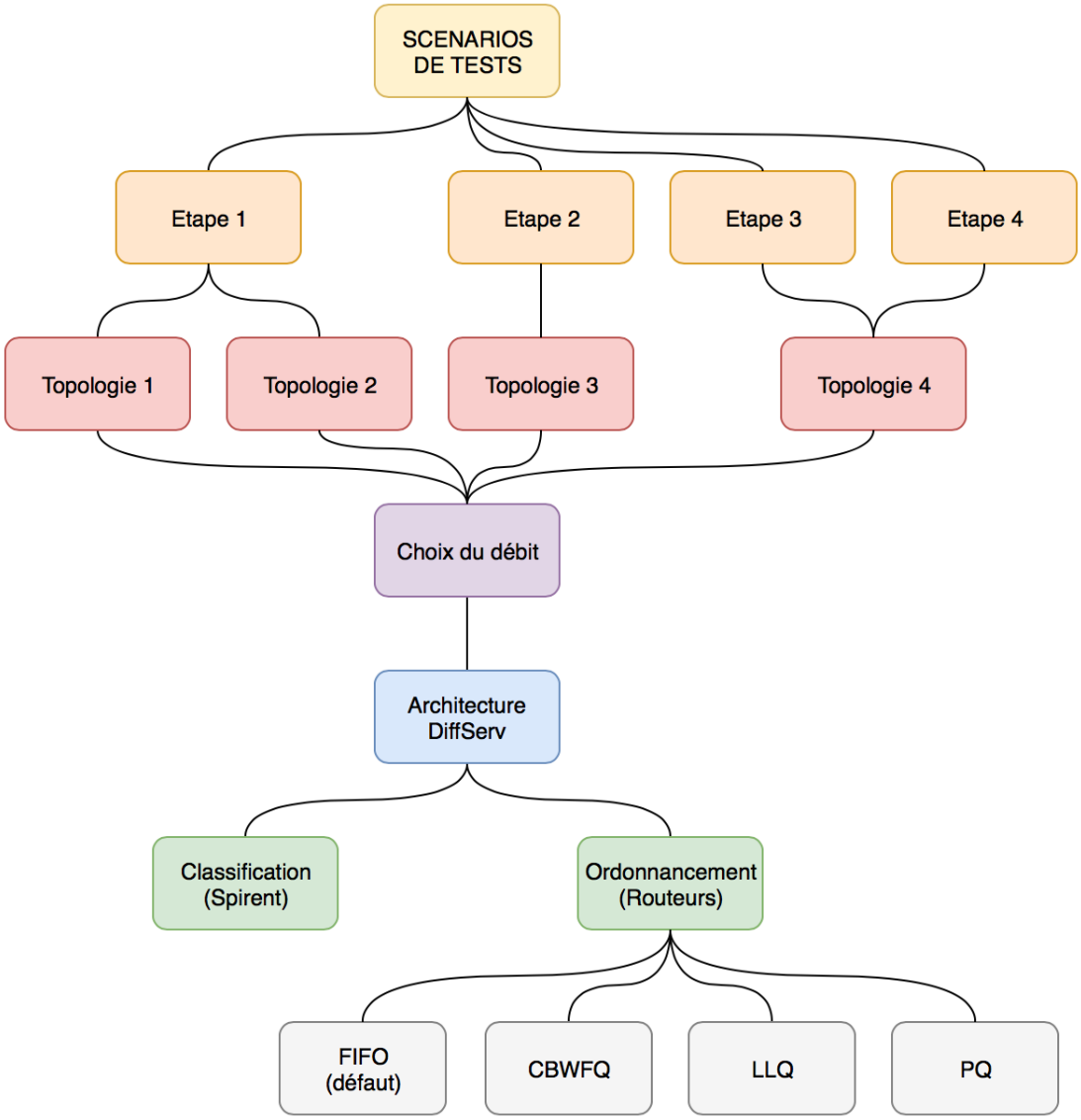


Figure 37 – Scénario de tests

Tout comme les topologies réseau, les scénarios de tests sont définis et propres aux différentes étapes.

3.5.1 Étape 1

Voici un résumé de la mise en place des stratégies de QoS:

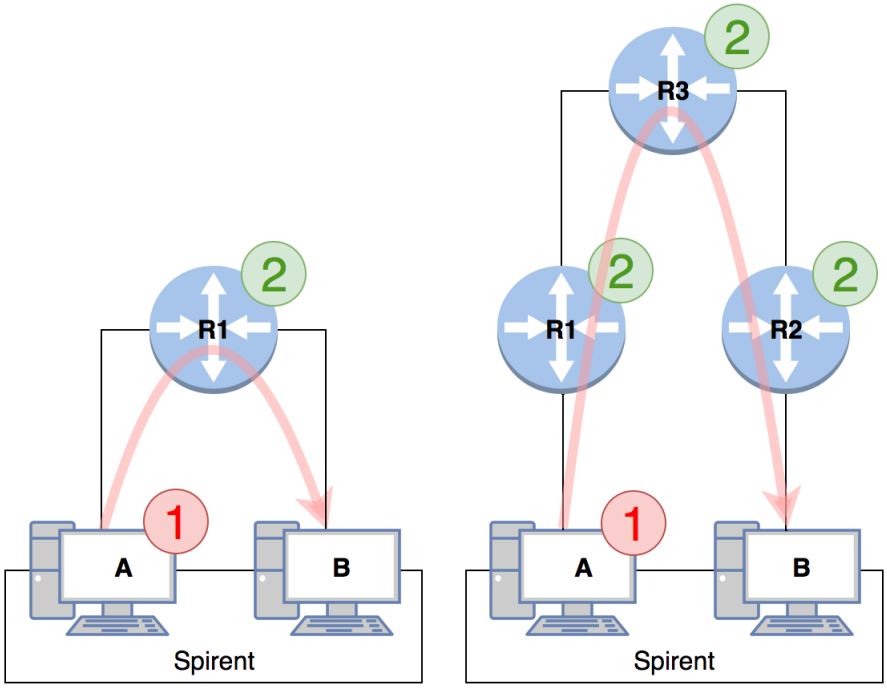


Figure 38 – Stratégies de QoS pour l'étape 1 sur les topologies 1 et 2

1. Flux marqués sur le Spirent selon la figure 36
2. Type de files d'attente par défaut ou PQ

La flèche représente le flux de trafic.

Les tests réalisables sur ces 2 topologies peuvent être très variés, c'est pourquoi nous allons fixer certaines limites.

Nous rappelons que les routeurs R1, R2 et R3 sont des Cisco 2800 Series.

Test 1: latence et perte en FIFO

Les deux topologies définies dans l'étape 1 vont servir à réaliser des comparaisons de comportement avec un réseau à 1 routeur et un réseau à 3 routeurs. La première question que nous nous posons est sur la latence générale des flux au niveau de l'émission depuis le Spirent et la réception sur le Spirent, comment la latence évolue-t-elle ? Comment se comporte-t-elle ? Nous pousserons les tests afin de définir le seuil auquel la latence reste stable et celui auquel la latence augmente avec le temps.

Pour réaliser ce test, nous jouerons avec le pourcentage de génération de volume de trafic sur le Spirent, comme défini précédemment, ce pourcentage ne peut pas dépasser 100% (100Mb/s).

Ce test nous permettra aussi d'analyser s'il y a des paquets perdus ou non.

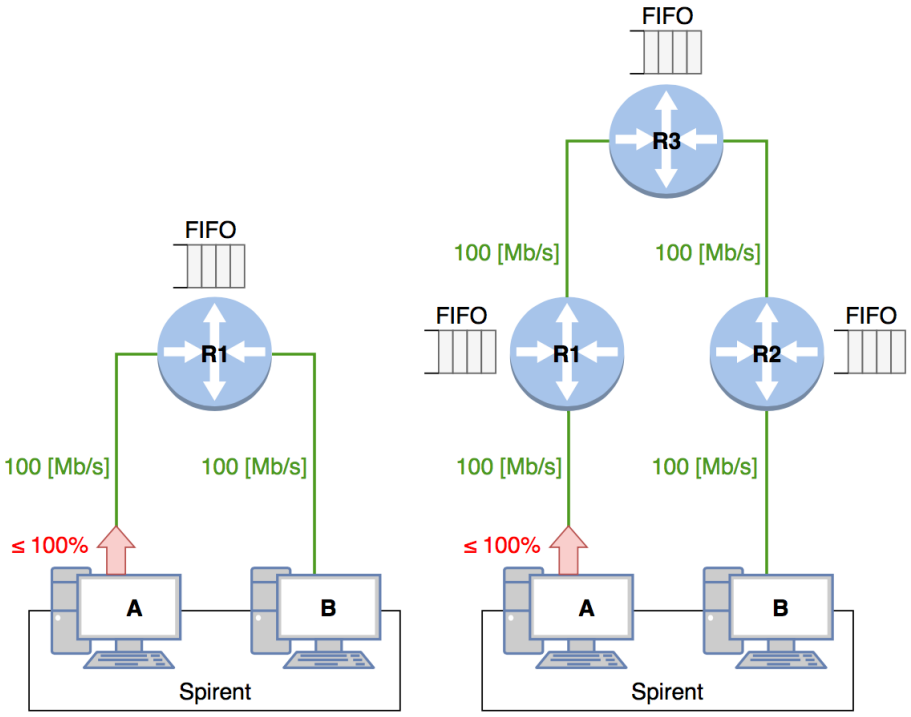


Figure 39 – Test 1: latence et perte en FIFO avec des réseaux de 1 et 3 routeurs

Test 2: latence et perte en PQ

Nous analyserons aussi, dans cette étape 1, la file d'attente PQ afin de comparer nos résultats avec les résultats présentés dans le projet de semestre 5 de Monsieur Simon Lièvre "QoSLab" [1]. Nous analyserons la latence générale et la latence pour chaque type de flux en bridant la ligne de sortie du routeur. Nous nous pencherons aussi sur la perte de paquets, car avec une ligne à 10Mb/s, il y aura forcément une perte de paquets importante.

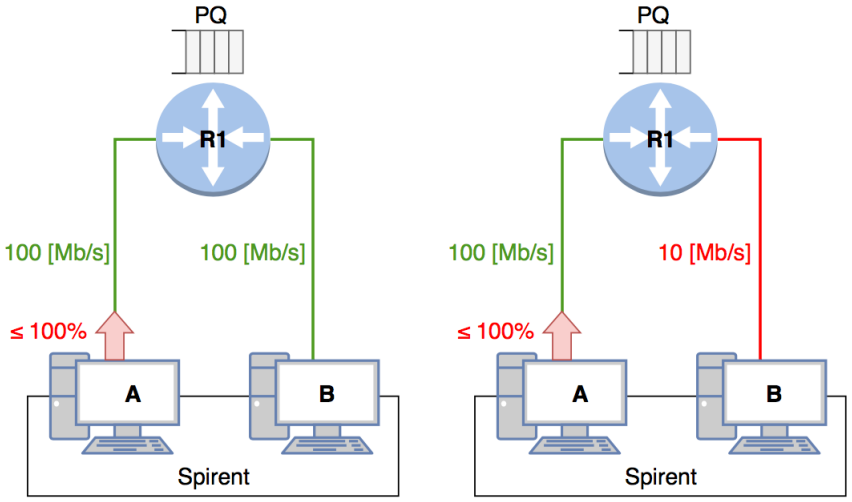


Figure 40 – Test 2: latence et perte en PQ avec des liens à 100 et 10Mb/s

3.5.2 Étape 2

Voici un résumé de la mise en place des stratégies de QoS:

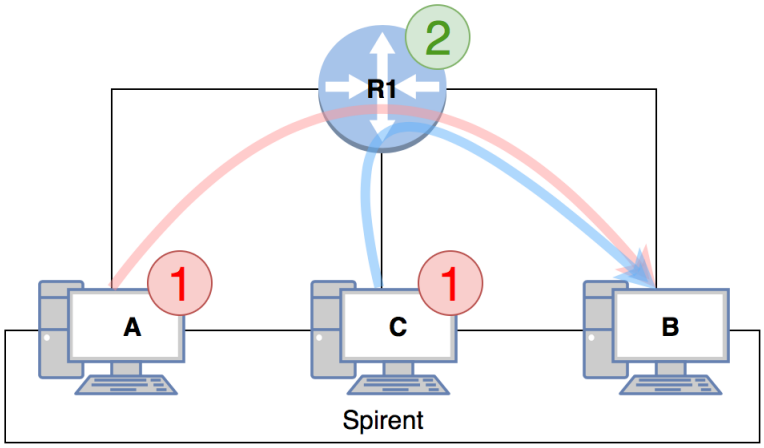


Figure 41 – Stratégies de QoS pour l'étape 2 sur la topologie 3

1. Flux marqués sur le Spirent selon la figure 36
2. Type de files d'attente par défaut ou PQ

Les flèches représentent le flux de trafic.

Les tests réalisables sur cette topologie peuvent être très variés, c'est pourquoi nous allons fixer certaines limites.

Nous rappelons que le routeur R1 est un Cisco 2900 Series.

À noter que sur la nouvelle source C, les flux générés sont les mêmes que sur la source A, la seule différence est le pourcentage de trafic injecté dans le réseau.

Test 3: latence et perte en FIFO

Même démarche que dans l'étape 1, la topologie définie dans cette étape va servir à réaliser des comparaisons de comportement au niveau de la latence. Avec cette étape, les tests iront plus loin, car nous pourrons injecter dans le réseau plus de 100% de volume de trafic dans le réseau.

Dans un premier temps, nous réaliserons un test préalable avec une génération à 100% afin de comparer les routeurs Cisco 2800 Series et Cisco 2900 Series pour déterminer s'ils ont le même comportement au niveau de la latence. Ce test se réalisera avec 1 routeur Cisco 2900 Series.

Dans un second temps, nous réaliserons une génération de trafic à 101% pour pouvoir analyser la latence et les pertes de paquets s'il y en a. Nous pousserons ensuite les tests avec un plus grand pourcentage d'injection de trafic dans le réseau afin de déterminer quand la perte de paquets apparaît, s'il n'y a pas encore de perte de paquets, et afin de déterminer quand le seuil maximal de latence apparaît.

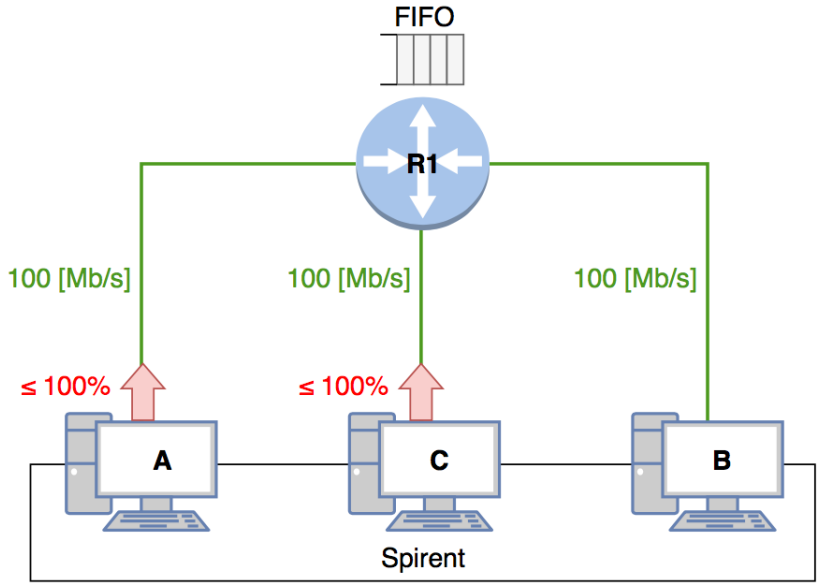


Figure 42 – Test 3: latence et perte en FIFO avec une génération de trafic de 200Mb/s

3.5.3 Étape 3

Voici un résumé de la mise en place des stratégies de QoS:

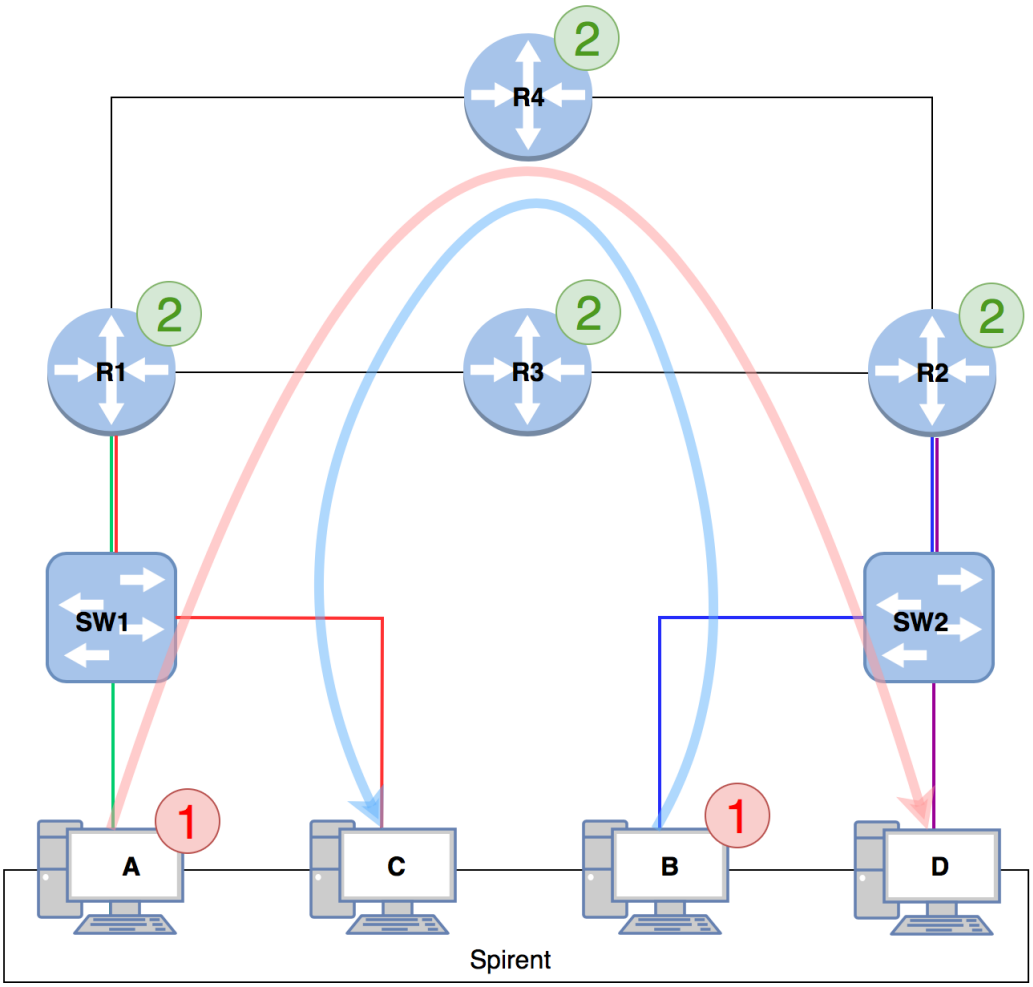


Figure 43 – Stratégies de QoS pour l'étape 3 sur la topologie 4

1. Flux marqués sur le Spirent selon la figure 36
2. Type de files d'attente par défaut ou PQ

Les flèches représentent le flux de trafic.

Les tests réalisables sur cette topologie peuvent être très variés, c'est pourquoi nous allons fixer certaines limites.

Dans cette troisième, nous présentons la latence par flux et la perte de paquets. Pour commencer, nous réaliserons un test de base avec des liens dans le réseau à 100Mb/s, ce réseau même étant capable de gérer une génération de trafic à 100%. Ensuite, nous définirons un lien du réseau, dans l'anneau à 10Mb/s et pour finir, nous romprons un lien afin d'en analyser les conséquences.

Nous rappelons que les routeurs R3 et R4 sont des Cisco 2800 Series, les routeurs R1 et R2 sont des Cisco 2900 Series et les switchs SW1 et SW2 sont des Cisco Catalyst 3560-CG Series.

Test 4: latence et perte en FIFO

Le test 4 permet d'analyser la latence des flux et la perte de paquets sur un réseau en forme d'anneau, avec des routeurs implémentant des files d'attente FIFO.

Premièrement, nous ferons notre analyse en nous basant sur le réseau suivant. Normalement, la latence devrait augmenter linéairement, mais rester relativement basse et il ne devrait pas y avoir de perte de paquets.

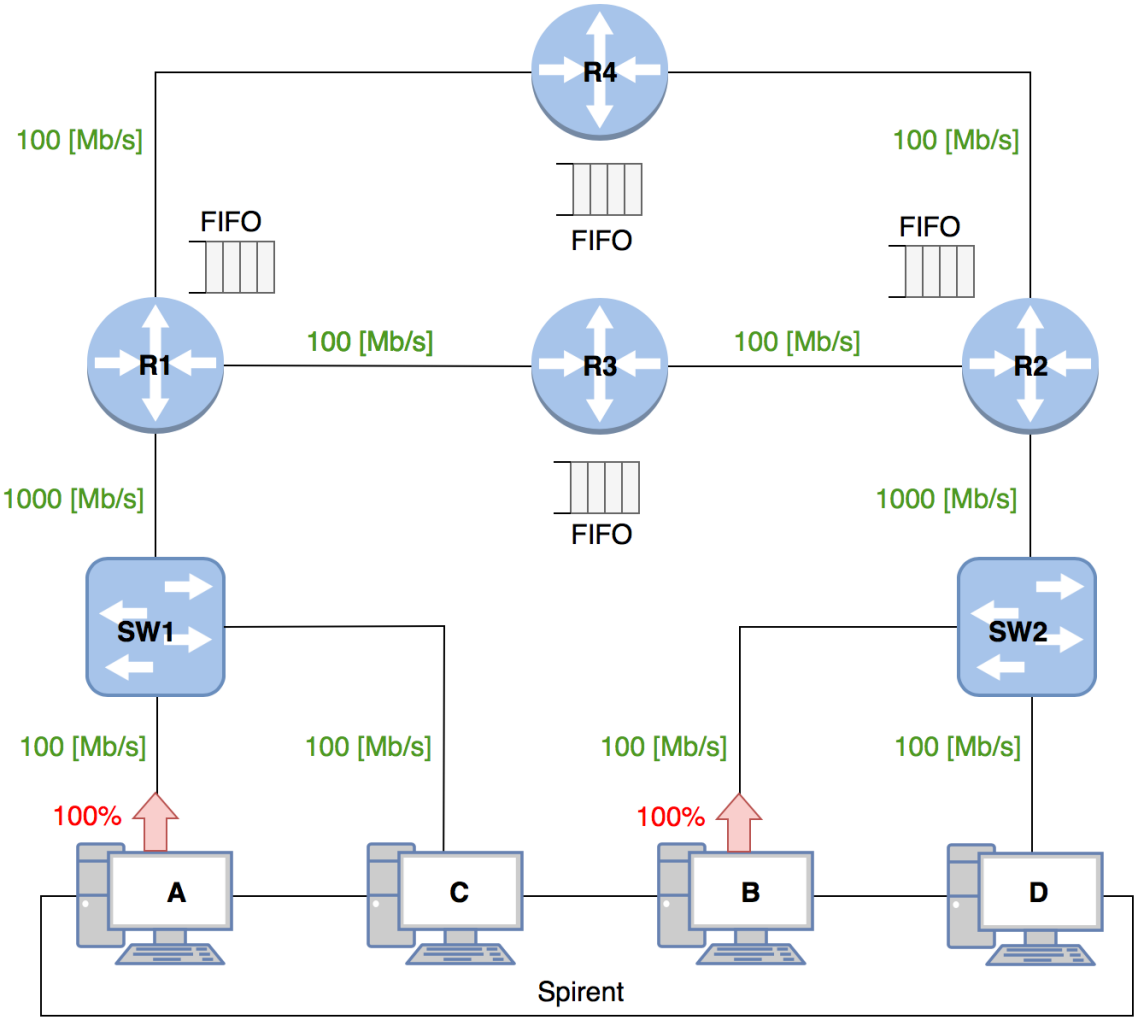


Figure 44 – Test 4: latence et perte en FIFO

Deuxièmement, nous analyserons la latence générale, la latence par flux et les pertes de paquets sur le réseau selon le schéma suivant. Une ligne au coeur du réseau sera bridée à 10Mb/s. Nous nous attendons à une perte de trafic de 50% et une latence atteignant un seuil maximum.

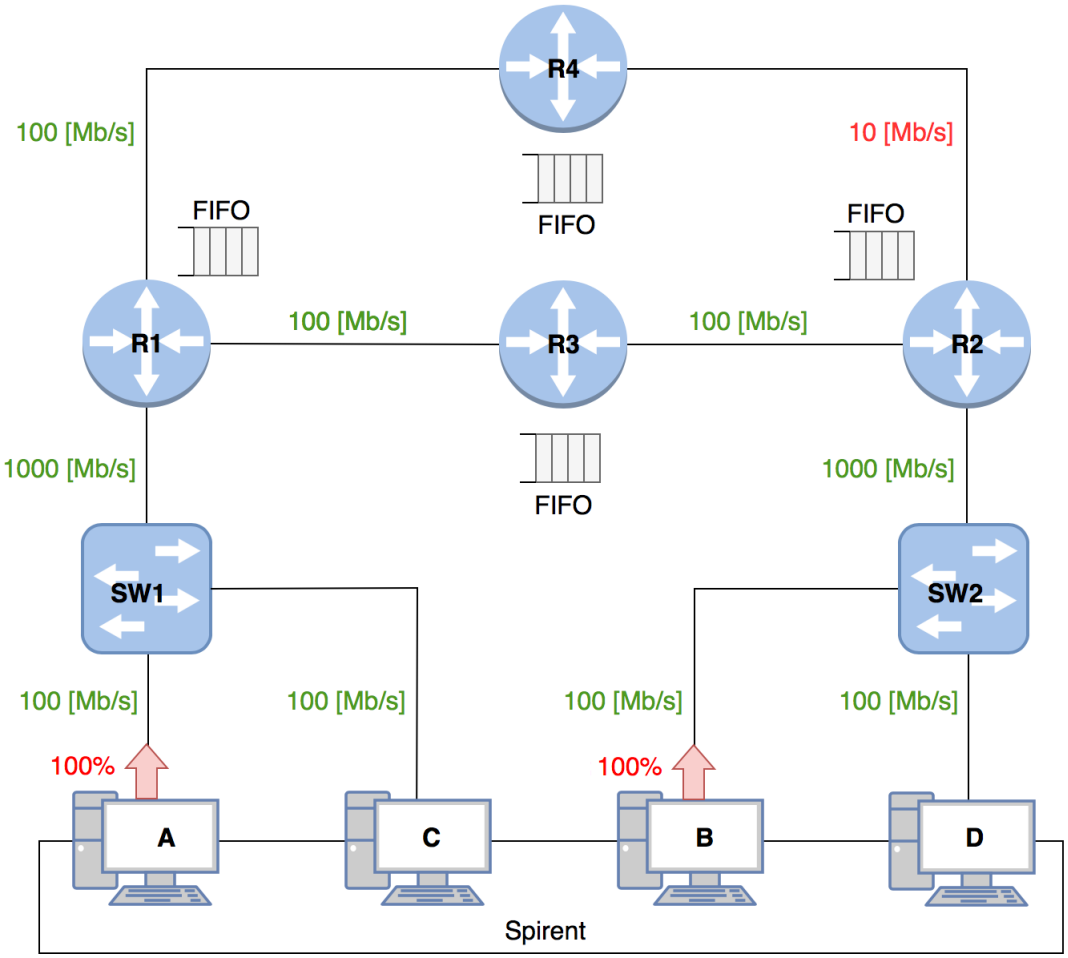


Figure 45 – Test 4: latence et perte en FIFO avec un lien à 100Mb/s

Pour terminer ce test, en nous basant sur les résultats du test précédent, nous couperons le lien par lequel le trafic transite au niveau de l'anneau afin de pouvoir analyser la redéfinition des routes. Une perte de paquet est à prévoir lors de la rupture du lien, mais cette réaction devrait vite être endiguée avec la redéfinition des routes.

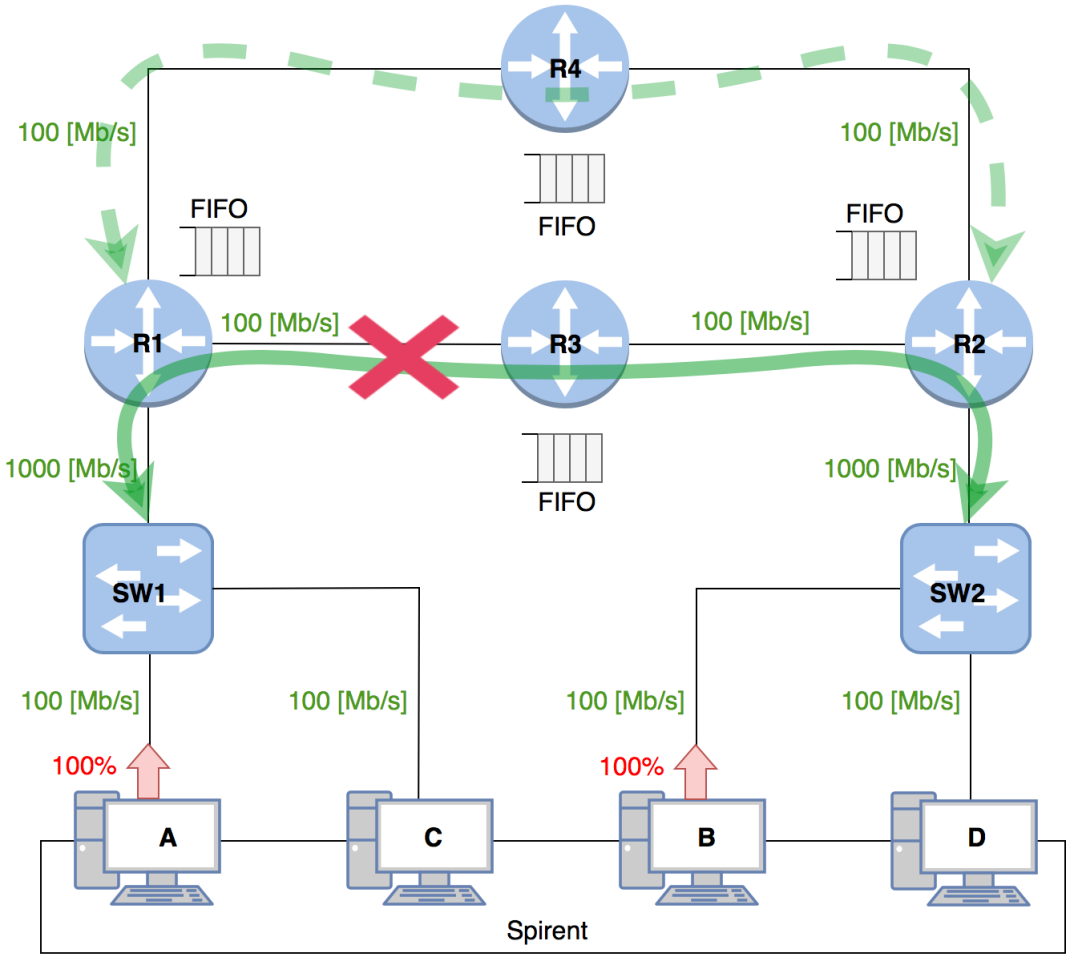


Figure 46 – Test 4: latence et perte en FIFO avec une rupture de lien

3.5.4 Étape 4

Voici un résumé de la mise en place des stratégies de QoS:

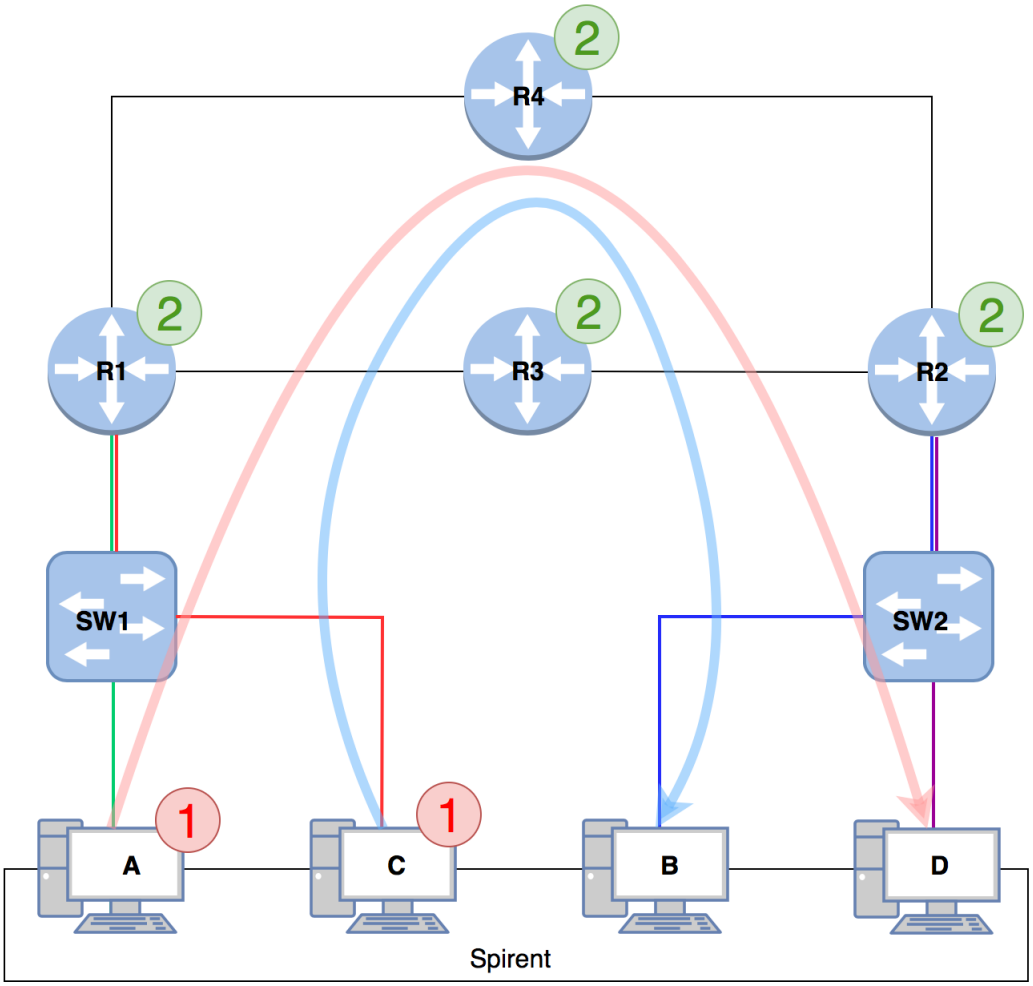


Figure 47 – Stratégies de QoS pour l'étape 3 sur la topologie 4

1. Flux marqués sur le Spirent selon la figure 36
2. Type de files d'attente par défaut ou PQ

Les flèches représentent le flux de trafic.

Les tests réalisables sur cette topologie peuvent être très variés, c'est pourquoi nous allons fixer certaines limites.

Cette quatrième étape présente un test composé de 3 sous-tests, les caractéristiques du réseau sont semblables dans les 3 sous-tests, la seule différence est le fait que les routeurs implémentent chacun leur tour 3 types de files d'attente différentes.

Nous rappelons que les routeurs R3 et R4 sont des Cisco 2800 Series, les routeurs R1 et R2 sont des Cisco 2900 Series et les switchs SW1 et SW2 sont des Cisco Catalyst 3560-CG Series.

Test 5: latence et perte en FIFO/CBWFQ/LLQ

Le test 5 permet d'analyser la latence des flux et la perte de paquets sur un réseau en forme d'anneau avec des routeurs implémentant des files d'attente FIFO, CBWFQ et LLQ.

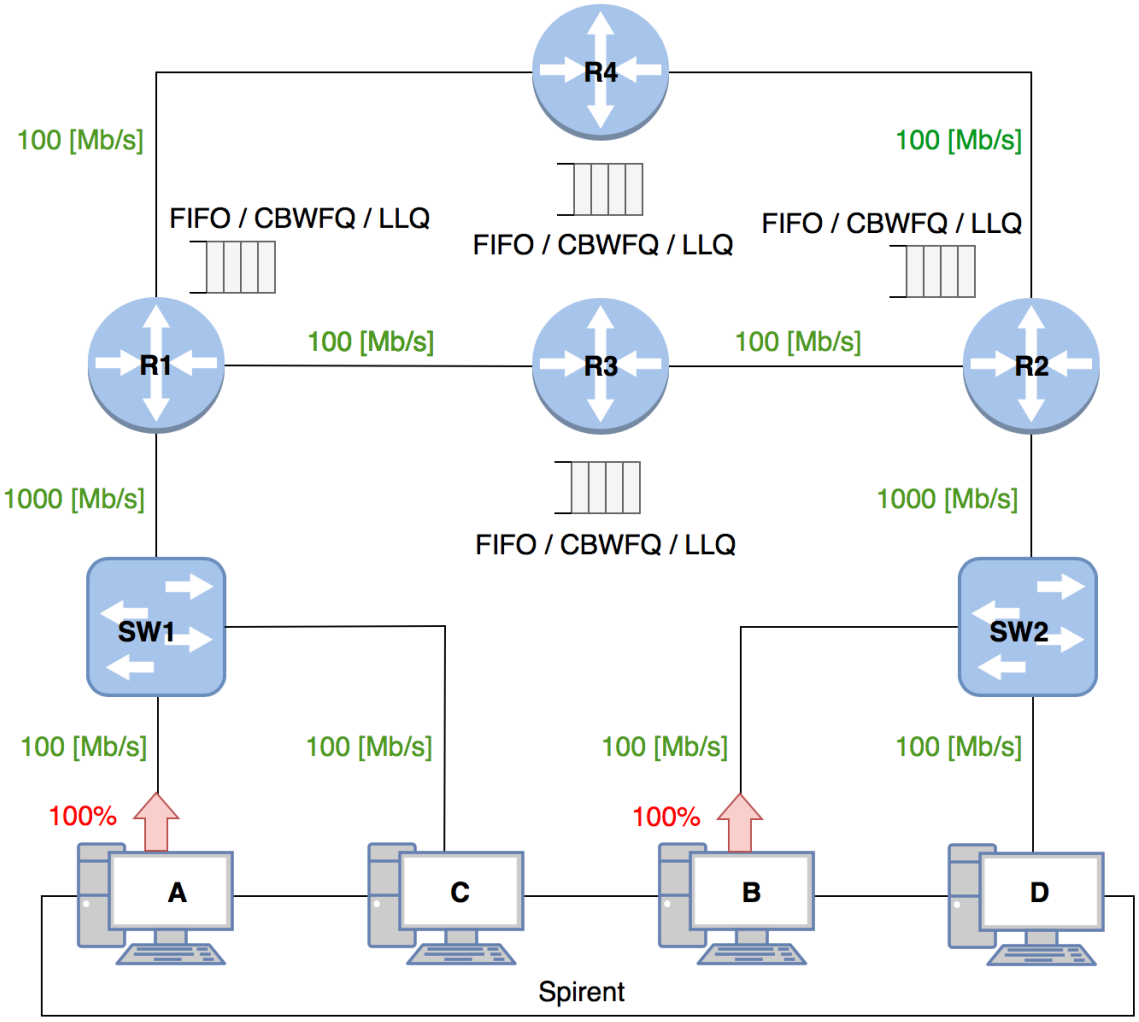


Figure 48 – Test 5: latence et perte en FIFO/CBWFQ/LLQ

Nous analyserons les résultats et définirons quel type de files d'attente correspond le mieux au besoin de ce réseau de tests.

Nous nous pencherons aussi sur le mécanisme de répartition ou partage des charges.

3.6 Configuration des routeurs

Nous travaillons avec des routeurs Cisco 2800 Series et Cisco Series 2900 Series, sur ces routeurs, nous devons configurer les interfaces, les files d'attente et les VLANs pour les topologies qui en requièrent.

Pour plus d'informations sur les configurations des routeurs, vous trouverez en annexe, dans le catalogue, les configurations complètes.

3.6.1 Interfaces

Pour configurer les interfaces sur un routeur Cisco, il faut suivre les étapes suivantes:

1. **Interface:** Définir l'interface à configurer:

```
Cmd: R1(config)# interface "nom de l'interface"
Ex: R1(config)# interface fast0/0
```

2. **Interface:** Définir l'adresse IP et le masque de sous-réseau de cette interface:

```
Cmd: R1(config-if)# ip address "adresse IP" "masque de sous-réseau"
Ex: R1(config-if)# ip address 192.168.10.1 255.255.255.0
```

3. **Interface:** Empêcher le shutdown sur cette l'interface:

```
Cmd: R1(config-if)# no shutdown
```

4. **Interface:** Définir la vitesse du port:

```
Cmd: R1(config-if)# speed "vitesse en Mb/s"
Ex: R1(config-if)# speed 100
```

3.6.2 VLANs

Pour la mise en place du réseau de tests relatif à l'étape 3 et 4 du projet, nous devons mettre en place des VLANs. À noter que la configuration des switchs doit être réalisée avant la configuration des routeurs, au niveau VLAN, donc au niveau des sous-interfaces des routeurs.

1. **Interface:** Définir la sous interface à configurer:

```
Cmd: R1(config)# interface "nom de l'interface et numéro du VLAN"
Ex: R1(config)# interface GigabitEthernet0/0/0.20
```

2. **Interface:** Définir l'adresse IP et le masque de sous-réseau de cette sous-interface:

```
Cmd: R1(config-subif)# ip address "adresse IP" "masque de sous-réseau"
Ex: R1(config-subif)# ip address 192.168.10.1 255.255.255.0
```

3. **Interface:** Définir le type d'encapsulation:

```
Cmd: R1(config-subif)# encapsulation dot1Q 20
```

La vitesse de l'interface est à définir au niveau de l'interface physique.

3.6.3 Files d'attente

CBWFQ

Pour configurer la file d'attente CBWFQ sur un routeur Cisco, il faut suivre les étapes suivantes:

1. **Classe de trafic:** Définir les classes de trafic:

```
Cmd: R1(config)# class-map "nom de la classe de trafic"
Ex: R1(config)# class-map class-voice
```

Le trafic ne correspondant à aucune classe est placé dans la classe par défaut "class-default".

2. **Classe de trafic:** Définir les règles de correspondance entre la classe de trafic et le PHB définit lors de la génération des flux sur le Spirent :

```
Cmd: R1(config-cmap)# match ip dscp "nom du PHB"
Ex: R1(config-cmap)# match ip dscp ef
```

Par défaut, les classes sont de type "match all" ce qui signifie que ce trafic doit remplir toutes les conditions fixées par les règles de correspondances.

3. **Politique de trafic:** Définir les politiques de trafic:

```
Cmd: R1(config)# policy-map "nom de la politique de trafic"
Ex: R1(config)# policy-map policy-cbwfq
```

4. **Politique de trafic:** Configurer les politiques de trafic:

```
Cmd: R1(config-pmap)# class "nom de la classe de trafic"
Ex: R1(config-pmap)# class class-voice
```

5. **Politique de trafic:** Réserver la bande passante de cette classe de trafic:

```
Cmd: R1(config-pmap-c)# bandwidth "bande passante en kb/s"
Ex: R1(config-pmap-c)# bandwidth 1000
```

6. **Politique de trafic:** Définir l'algorithme WRED pour la gestion active des files d'attente:

```
Cmd: R1(config-pmap-c)# random-detect dscp-based
```

Les poids sont calculés en fonction du DSCP.

Ces 6 premières étapes sont à faire pour chaque type de trafic. Par contre, le trafic Internet ira dans la classe par défaut "class-default".

7. **Interfaces:** Définir l'interface à configurer au niveau de la sortie du routeur:

```
Cmd: R1(config)# interface "nom de l'interface"
Ex: R1(config)# interface fast0/1
```

8. **Interfaces:** Définir la bande passante que l'on peut réserver sur cette interface:

```
Cmd: R1(config-if)# max-reserved-bandwidth "bande passante en pourcent"
Ex: R1(config-if)# max-reserved-bandwidth 95
```

Par défaut, le pourcentage de bande passante réservé est de 75%, dans notre cas, nous réservons 95% de la bande passante, les 5% restants sont réservés par les protocoles de routage et le management.

9. **Interfaces:** Appliquer la politique de trafic à cette interface:

```
Cmd: R1(config-if)# service-policy output "nom de la politique de trafic"
Ex: R1(config-if)# service-policy output policy-cbwfq
```

LLQ

Pour rappel, LLQ est une variante de la file d'attente CBWFQ, comportant une queue spéciale pour le trafic en temps réel. Les classes de trafic à mettre en place sont donc les mêmes que pour la file d'attente CBWFQ, nous pouvons utiliser celles déjà définies.

Pour configurer la file d'attente LLQ sur un routeur Cisco, il faut suivre les étapes suivantes:

3. **Politique de trafic:** Définir les politiques de trafic:

```
Cmd: R1(config)# policy-map "nom de la politique de trafic"
Ex: R1(config)# policy-map policy-llq
```

4. **Politique de trafic:** Configurer les politiques de trafic:

```
Cmd: R1(config-pmap)# class "nom de la classe de trafic"
Ex: R1(config-pmap)# class class-voice
```

5. **Politique de trafic:** Réserver la bande passante de cette classe de trafic (à définir seulement pour les queues normales):

```
Cmd: R1(config-pmap-c)# bandwidth "bande passante en kb/s"
Ex: R1(config-pmap-c)# bandwidth 1000
```

6. **Politique de trafic:** Définir la priorité de la classe de trafic sensible (à définir seulement pour la queue spéciale pour le trafic en temps réel:

```
Cmd: R1(config-pmap-c)# priority "bande passante en kb/s"
Ex: R1(config-pmap-c)# priority 1000
```

7. **Politique de trafic:** Définir l'algorithme WRED pour la gestion active des files d'attente:

```
Cmd: R1(config-pmap-c)# random-detect dscp-based
```

Les poids sont calculés en fonction du DSCP.

Ces 7 premières étapes (sauf la 6) sont à faire pour chaque type de trafic. Par contre, le trafic Internet ira dans la classe par défaut "class-default".

8. **Interfaces:** Définir l'interface à configurer au niveau de la sortie du routeur:

```
Cmd: R1(config)# interface "nom de l'interface"
Ex: R1(config)# interface fast0/1
```

9. **Interfaces:** Définir la bande passante que l'on peut réserver sur cette interface:

```
Cmd: R1(config-if)# max-reserved-bandwidth "bande passante en pourcent"
Ex: R1(config-if)# max-reserved-bandwidth 95
```

Par défaut, le pourcentage de bande passante réservé est de 75%, dans notre cas, nous réservons 95% de la bande passante, les 5% restants sont réservés par les protocoles de routage et le management.

10. **Interfaces:** Appliquer la politique de trafic à cette interface:

```
Cmd: R1(config-if)# service-policy output "nom de la politique de trafic"
Ex: R1(config-if)# service-policy output policy-llq
```

PQ

Pour configurer la file d'attente PQ sur un routeur Cisco, il faut suivre les étapes suivantes:

1. **Liste d'accès:** Définir les listes d'accès pour que les paquets IP en provenance de n'importe quelle source vers n'importe quelle destination correspondant au PHB définit lors de la génération des flux sur le Spirent:

Cmd: R1(config)# access-list "numèro de la liste d'accès" permit ip any any dscp "nom du PHB" Ex: R1(config)# access-list 101 permit ip any any dscp ef

Le trafic ne correspondant à aucune classe est placé dans la classe par défaut "class-default".

2. **Liste de priorité:** Définir les listes de priorités correspondant aux listes d'accès:

Cmd: R1(config)# priority-list "numéro de la liste de priorité" protocol ip "niveau de priorité" "numéro de la liste d'accès" Ex: R1(config)# priority-list 1 protocol ip high list 101

Pour rappel, lorsqu'on met en place la file d'attente PQ, il y a 4 types de files d'attente (High, Medium, Normal et Low), elles correspondent chacune à un niveau de priorité différent.

3. **Interfaces:** Définir l'interface à configurer au niveau de la sortie du routeur:

Cmd: R1(config)# interface "nom de l'interface" Ex: R1(config)# interface fast0/1

4. **Interfaces:** Appliquer la politique de trafic à cette interface:

Cmd: R1(config-if)# priority-group "numéro de la liste de priorité" Ex: R1(config-if)# priority-group 1

À noter que cette configuration des files d'attente PQ n'est possible que sur les routeurs Cisco 2800 Series.

3.6.4 Routage

Pour la mise en place de la première topologie de l'étape 1, nous n'avons pas besoin de protocole de routage, car il n'y a qu'un routeur dans le réseau. En revanche, pour la mise en place de la seconde topologie de l'étape 1, et les autres étapes, nous devons mettre en place un protocole de routage, car nous avons plusieurs routeurs et ces derniers doivent connaître leurs voisins pour acheminer les données de l'expéditeur à la destination.

Nous avons décidé de mettre en place OSPF (Open Shortest Path First) comme protocole de routage, pour plusieurs raisons:

- convergence très rapide
- adapté aux grands réseaux (aucune limitation sur le nombre de sauts)
- faible utilisation de la bande passante
- standard (interopérable)
- meilleur équilibrage de charge
- définition logique des réseaux où les routeurs peuvent être répartis en zones
- authentification du routage à l'aide de différentes méthodes d'authentification par mot de passe

Pour configurer le protocole de routage OSPF sur un routeur Cisco, il faut suivre les étapes suivantes:

1. **OSPF:** Définir le protocole de routage OSPF:

```
Cmd: R1(config)# router ospf "numéro de processus"
Ex: R1(config)# router ospf 1
```

Ce numéro de processus est le même pour tous les routeurs du réseau utilisant le protocole de routage OSPF.

2. **OSPF:** Définir l'identifiant du routeur OSPF:

```
Cmd: R1(config-router)# router ospf "numéro de l'identifiant"
Ex: R1(config-router)# router-id 1.1.1.1
```

Cet identifiant est unique pour chaque routeur du réseau utilisant le protocole de routage OSPF.

3. **OSPF:** Définir les réseaux et la zone:

```
Cmd: R1(config-router)# network "adresse IP du réseau" "masque de sous-réseau" "zone OSPF"
Ex: R1(config-router)# network 192.168.10.0 0.0.0.255 area 0
```

Ces indications de réseau sont à configurer sur tous les routeurs, mais en indiquant seulement les réseaux directement connectés au routeur en question, pour le reste, le protocole OSPF, une fois activé, s'en charge.

Pour la réalisation de différents tests, nous pouvons aussi définir le coût des interfaces pour les calculs des routes au niveau OSPF.

4. **Interfaces:** Définir l'interface à configurer au niveau de la sortie du routeur:

```
Cmd: R1(config)# interface "nom de l'interface"
Ex: R1(config)# interface fast0/1
```

5. **OSPF:** Définir les coûts des interfaces:

```
Cmd: R1(config-if)# ip ospf cost "priorité"
Ex: R1(config-if)# ip ospf cost 1
```

À noter que dans notre cas, pour une ligne à 100Mb/s le coût de l'interface est de 1, pour une ligne à 10Mb/s le coût est de 10. Le coût le plus bas est prioritaire et le calcul des routes se fait en fonction de ces coûts.

3.7 Configuration des switchs

Nous travaillons avec des switchs Cisco Catalyst 3560-CG Series, sur ces switchs, nous devons configurer les interfaces. Les switchs sont uniquement mis en place pour le réseau de tests relatif aux étapes 3 et4.

Pour plus d'informations sur les configurations des switchs, vous trouverez en annexe, dans le catalogue, les configurations complètes.

3.7.1 Interfaces

Pour configurer les interfaces sur un switch Cisco, il faut suivre les étapes suivantes:

1. **Interface:** Définir l'interface à configurer:

```
Cmd: R1(config)# interface "nom de l'interface"
Ex: R1(config)# interface GigabitEthernet0/7
```

2. **Interface:** Configurer l'encapsulation trunk en tant que dot1q:

```
Cmd: R1(config-if)# switchport trunk encapsulation dot1q
```

3. **Interface:** Activer le mode trunk sur l'interface:

```
Cmd: R1(config-if)# switchport mode trunk
```

4. **Interface:** Ajouter des VLANs à la liste des VLANs autorisés sur l'interface:

```
Cmd: R1(config-if)# speed "numéro des VLANs"
Ex: R1(config-if)# switchport trunk allowed vlan 1,20
```

5. **Interface:** Activer la QoS automatique sur les interfaces en fonction du champ DSCP:

```
Cmd: R1(config-if)# auto qos trust dscp
```

6. **Interface:** Définir la vitesse du port:

```
Cmd: R1(config-if)# speed "vitesse en Mb/s"
Ex: R1(config-if)# speed 100
```

Ces configurations sont à réaliser pour chacune des interfaces utilisées.

3.8 Méthode d'évaluation

La méthode d'évaluation nous permet de tenir une certaine cohérence entre les présentations des différents résultats des scénarios de tests.

L'infrastructure de chacun des scénarios de tests sera préalablement mise en place. Une fois cette dernière opérationnelle, nous pourrons démarrer nos tests.

Le Spirent fournit une grande quantité de diverses mesures, ces mesures seront tout au long du déroulement des phases de tests collectées afin d'en sortir des graphiques ou des tableaux. Ces graphiques ou ces tableaux représenteront divers éléments comme la latence ou la perte de paquets, tout dépendra des attentes du scénario de tests. Pour afficher les mesures dont nous aurons besoin, nous utiliserons plusieurs outils mis à disposition par le Spirent.

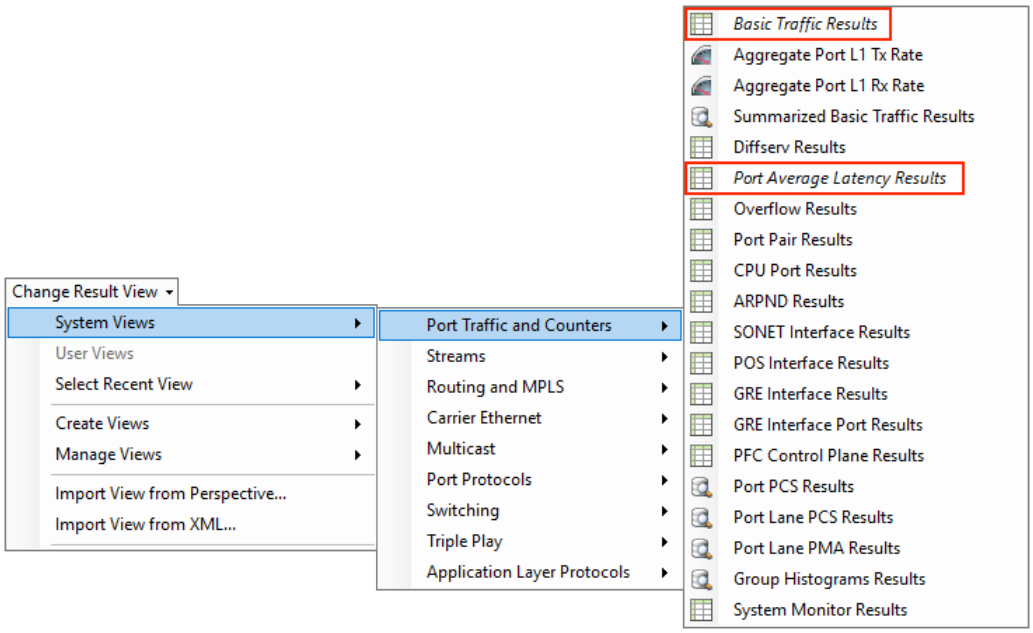


Figure 49 – Interfaces de visualisation des résultats des mesures pour le trafic général

Ces deux premières interfaces permettent de visualiser les résultats de mesures pour le trafic général.

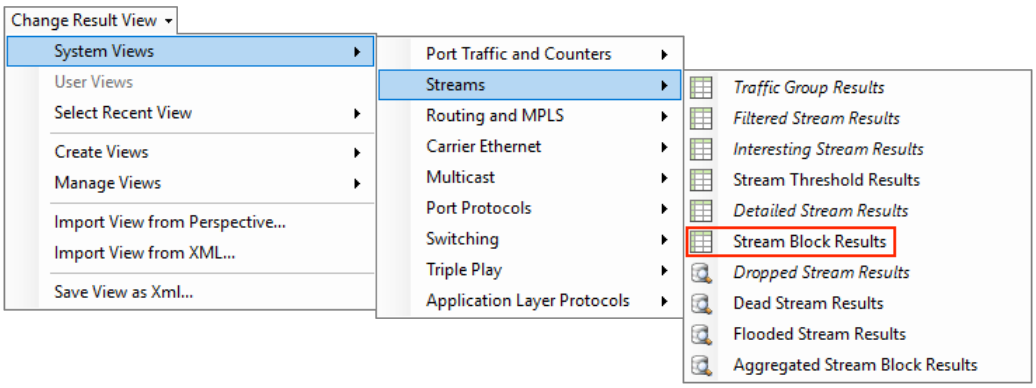


Figure 50 – Interfaces de visualisation des résultats des mesures pour le trafic par flux

Cette interface permet de visualiser les résultats de mesures pour les différents flux de trafic.

Un autre moyen de visualiser les résultats directement au niveau des files d'attente configurées est possible, nous utiliserons certaines commandes sur le routeur comme « show int fastethx/x" sur l'interface implémentant la file d'attente, qui nous permettra d'avoir une vision sur les trames reçues, émises et supprimées, et d'autres informations sur les files d'attente. À ne pas oublier de remettre les compteurs à 0 avec la commande "clear count" sur les routeurs.

Avec ces interfaces, nous pourrons visualiser les compteurs de trames émises et réceptionnées, les compteurs de bits, la latence moyenne, maximum et minimum, et beaucoup d'autres mesures.

Une fois ces mesures récoltées, nous créerons nos graphiques avec Excel ou divers tableaux afin de rendre l'interprétation des résultats agréables et explicites.

Lors de la présentation des résultats des scénarios des tests dans la partie tests et validation du projet, nous commencerons par citer les spécificités du réseau de tests, comme le nombre de routeurs dans le réseau et le modèle de ces derniers, le type de file d'attente mis en place, les particularités de la génération du flux de trafic, la vitesse des lignes de sortie des routeurs et le temps de la mesure.

Le graphique où les tableaux seront ensuite présentés et une conclusion, sur les résultats, sera rédigeé.

Nous allons essayer de rester dans une représentation très visuelle des résultats des scénarios de tests.

3.9 Conclusion de la conception

La partie de conception de ce projet a été d'une grande utilité. En effet, elle a permis de mettre sur pieds les différents réseaux de tests (NUT) utiles à la suite du projet. Au début du projet, nous n'avions pas pensé à mettre en place autant de réseaux de tests, mais au fil du projet les idées ont jailli, ce qui nous a permis de réfléchir aux différents scénarios tests que nous pourrions réaliser. Ces différents scénarios tests ne se rapportaient pas forcément à un seul réseau, mais l'idée de comparer les résultats d'un type de scénario de tests à plusieurs types de réseaux nous a semblé être une bonne chose. Ces différents réseaux ont été définis dans cette partie et leurs configurations au niveau du Spirent et des routeurs ont aussi été définis.

Grâce à la conception du projet, nous avons une bonne vue d'ensemble de tous les scénarios de tests mis en place et ces mêmes scénarios de tests se rapportent dans la partie de tests et validations du projet. Nous avons essayé de garder une certaine cohérence entre ce qui est défini, au niveau de la présentation des résultats des tests afin que les lecteurs se retrouvent sans problème. C'est pourquoi nous avons défini une méthodologie d'évaluation des scénarios de tests et nous nous y tiendrons.

Bien entendu, comme nous l'avons déjà dit, des scénarios de tests qui ont été définis dans cette partie de conception, mais durant les phases de réalisation et de validation, suivant les résultats, d'autres cas peuvent être pensés et testés afin de répondre à certaines questions. Nous allons donc faire des itérations entre la partie concession et la partie de tests et validation.

4 Réalisation

Ce chapitre traite de la réalisation de ce projet. Il est important de préciser que les routeurs permettant la réalisation de ce projet sont d'anciens routeurs d'entreprises et non des routeurs de coeur de réseau. Leurs performances peuvent donc être réduites.

La partie de réalisation traite de la configuration globale du Spirent. À noter que pour la mise en place d'un réseau de tests, comprenant la configuration du Spirent, des routeurs et des siwtchs, une documentation a été crée en annexe. Ces configurations sont aussi disponibles, dans le catalogue, en annexe.

4.1 Configuration du Spirent

La configuration du Spirent se décompose en 2 parties, les hôtes virtuels et la génération du trafic.

Pour la configuration du Spirent, nous nous sommes basés sur la documentation de configuration du Spirent réalisée en annexe au projet de Bachelor de Monsieur Gabriel Python "Cloud Topology Lab for multiple services" [11]. Dans cette partie du projet, nous ne décrivons que les spécificités et la vue d'ensemble des éléments configurés sur le Spirent.

Pour toutes informations sur la configuration du Spirent, veuillez vous référer à la documentation en annexe "Quel est l'effet de la QoS sur des petits réseaux de labo ? - Mise en place réseau de tests et stratégies de QoS - Configuration Spirent C1 et routeur Cisco 2800 Series" réalisée durant ce projet de semestre 6 "Quel est l'effet de la QoS sur des petits réseaux de labo ?".

4.1.1 Hôtes virtuels

Comme décrit dans la partie conception, nous définissons:

- **Étape 1**: 2 hôtes virtuels, A (sur le port 3 du Spirent) et B (sur le port 4 du Spirent). Le flux de trafic est généré depuis l'hôte virtuel A et réceptionné par l'hôte virtuel B.
- **Étape 2**: 3 hôtes virtuels, A (sur le port 3 du Spirent), B (sur le port 4 du Spirent) et C (sur le port 5 du Spirent. Les flux de trafic sont générés depuis les hôtes virtuels A et C et réceptionné par l'hôte virtuel B.
- **Étape 3**: 4 hôtes virtuels, A (sur le port 3 du Spirent), B (sur le port 4 du Spirent), C (sur le port 5 du Spirent) et D (sur le port 6 du Spirent. Le premier flux de trafic est généré depuis l'hôte virtuel A et réceptionné sur l'hôte virtuel D, le second flux de trafic est généré depuis l'hôte virtuel B et réceptionné sur l'hôte virtuel C.
- **Étape 5**: 4 hôtes virtuels, A (sur le port 3 du Spirent), B (sur le port 4 du Spirent), C (sur le port 5 du Spirent) et D (sur le port 6 du Spirent. Le premier flux de trafic est généré depuis l'hôte virtuel A et réceptionné sur l'hôte virtuel D, le second flux de trafic est généré depuis l'hôte virtuel C et réceptionné sur l'hôte virtuel B.

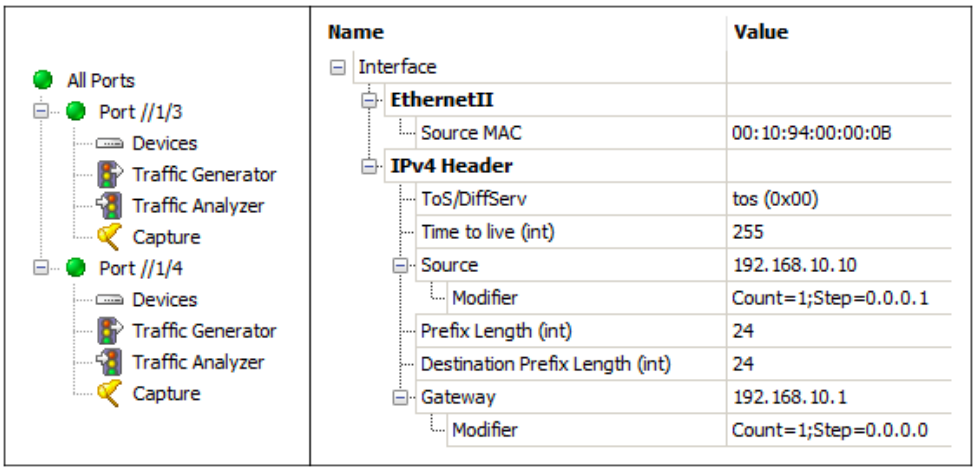


Figure 51 – Configuration des hôtes virtuels sur le Spirent

Sur la gauche de la figure précédente, nous pouvons voir les hôtes virtuels A et B (étape 1) configurés sur les ports 3 et 4. Sur la droite de la figure précédente, nous pouvons voir les configurations de l'hôte virtuel A.

Il est intéressant de noter que nous pouvons "jouer" avec la vitesse des ports du Spirent afin de mieux analyser le comportement des files d'attente sur le routeur. Bien entendu, cette spécification de vitesse des liens doit aussi être paramétrée sur les routeurs Cisco afin que la négociation soit faite correctement et que la ligne soit opérationnelle.

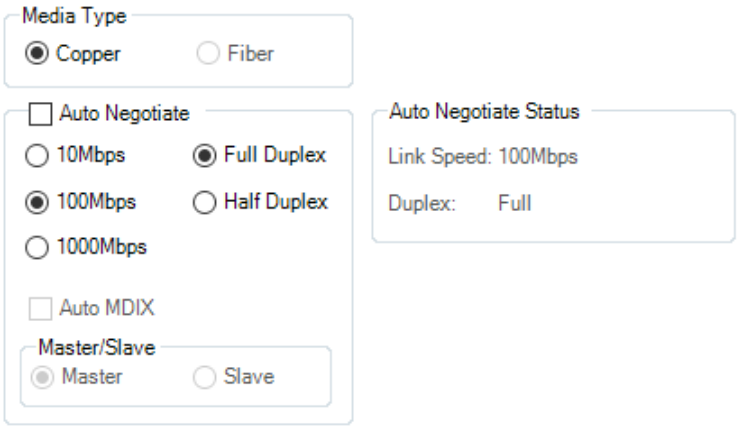


Figure 52 – Configuration de la vitesse des ports sur le Spirent

À noter que pour l'étape 1 et 2, aucun VLAN n'est défini au niveau des hôtes virtuels, en revanche pour l'étape 3 et l'étape 4, des VLANs sont définis dans le réseau de tests. Nous devons définir un VLAN pour chaque hôte virtuel afin que le trafic généré soit dirigé dans le réseau de tests vers les bons VLANs.

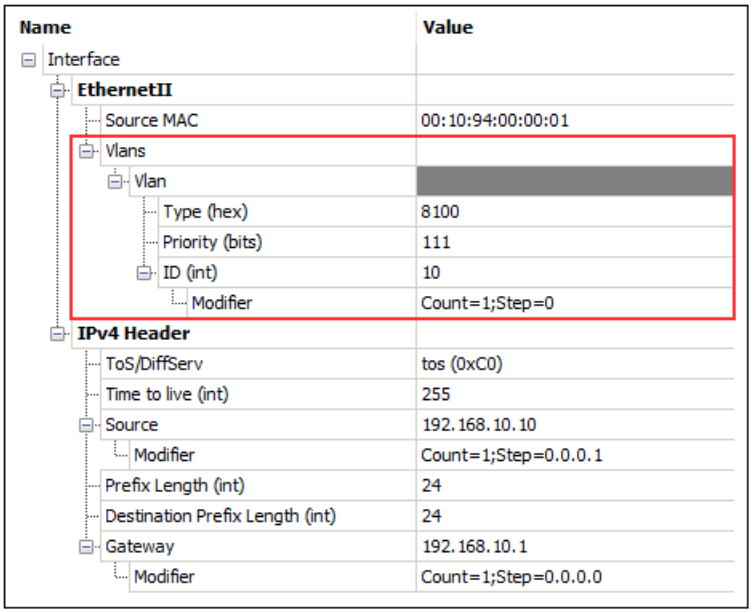


Figure 53 – Attribution d'une VLAN à l'hôte virtuel sur le Spirent

Contrairement à la figure 51, nous pouvons voir, dans la figure précédente, qu'un VLAN est défini avec son numéro dans l'hôte virtuel et cette notion est primordiale pour l'étape 3 et l'étape 4.

4.1.2 Générateur de trafic

Comme décrit dans la partie conception, nous définissons 5 types de trafic, la téléphonie, la télévision unicast, la télévision multicast, l'Internet et le management. Les informations suivantes vont nous aider à définir nos flux de trafic:

Trafic	Transport	DSCP	QoS Byte	DSCP (Hex)	Volume [Mbit/s]
Téléphonie	UDP	EF	B8	2E	1
Télévision multicast	UDP	AF41	88	22	1
Télévision unicast	TCP	AF41	88	22	56
Internet	TCP	AF13	38	0E	41
Mangement	TCP	CS6	C0	30	1

Figure 54 – Configuration des flux de trafic

Ces flux de trafic sont à définir:

- **Étape 1**: sur l'hôte virtuel A
- **Étape 2**: sur les hôtes virtuels A et C
- **Étape 3**: sur les hôtes virtuels A et B
- **Étape 4**: sur les hôtes virtuels A et C

Nous avons défini nos 5 types de flux de trafic, nous pouvons voir que pour chaque flux, la source et la réception sont chaque fois décrites avec l'adresse IP de l'hôte virtuel, et leur port avec le numéro de port su Spirent. Le volume de données pour chacun des flux de trafic y est aussi spécifié en se basant sur les chiffres relevés lors de l'analyse.

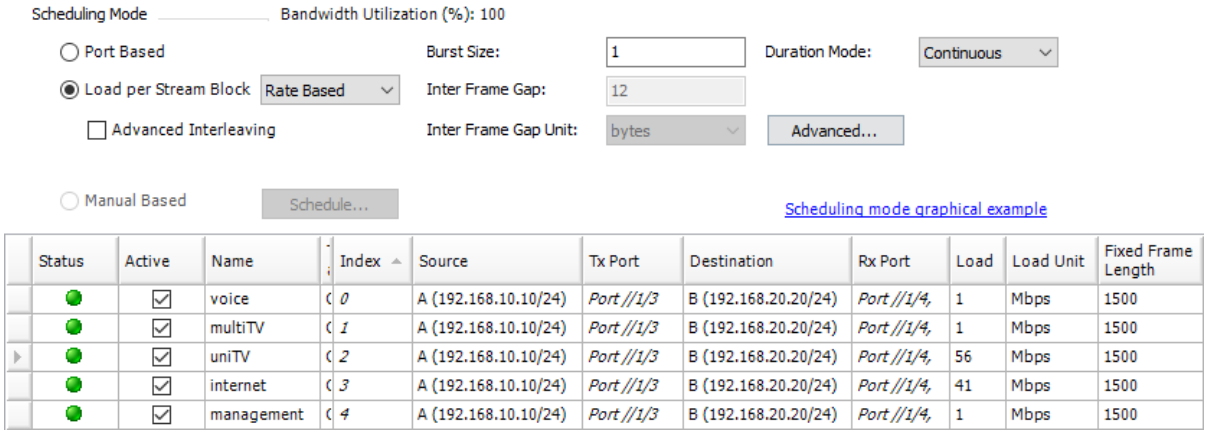


Figure 55 – Flux de trafic sur le Spirent avec une génération à 100%

Nous pouvons voir que l'utilisation de la bande passante est de 100%.

Comme décrit dans la conception, les valeurs correspondant au volume représentent au total une génération de trafic à 100%. Ces valeurs peuvent être modifiées afin de créer une génération de trafic à x%. Cette modification des valeurs se fait grâce à une simple règle de trois. Par exemple, dans la figure suivante, nous représentons une génération de trafic à 99%.

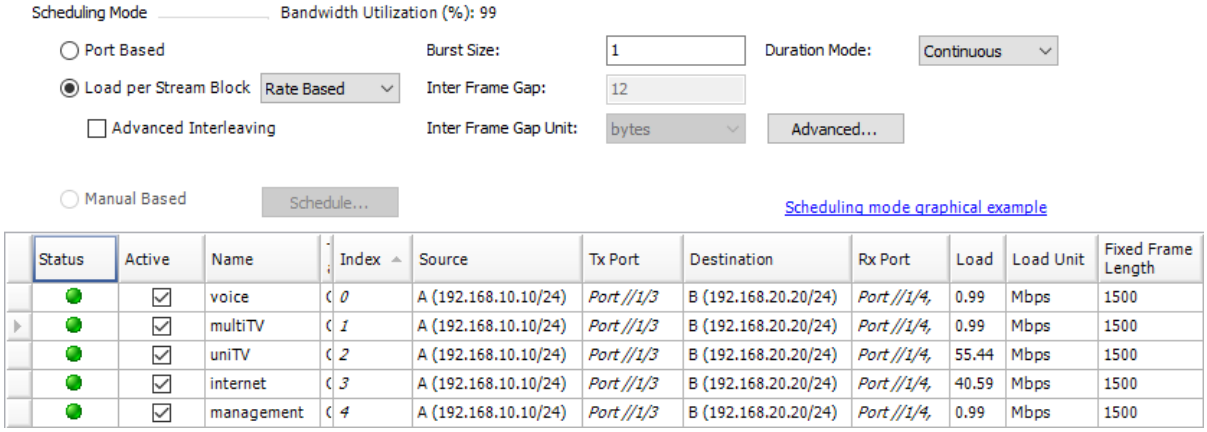


Figure 56 – Flux de trafic sur le Spirent avec une génération à 99%

Nous pouvons voir que l'utilisation de la bande passante est de 99%.

À noter que dans notre projet, les lignes reliant le Spirent aux routeurs (étape 1 et étape 2) ou les lignes reliant le Spirent aux switchs (étape 3 et étape 4) sont paramétrées à une vitesse de 100Mb/s, ce qui veut dire qu'une génération de trafic à 100% correspond à une génération de trafic à 100Mb/s ou une génération de trafic à 99% correspond à une génération de trafic à 99Mb/s.

Dans la vue d'ensemble des paramètres des différents flux de trafic, nous pouvons voir les différents headers des frames. Chaque frame contient un EthernetII Header, un IPv4 Header et un UDP Header ou TCP Header, en fonction du choix réalisé précédemment.

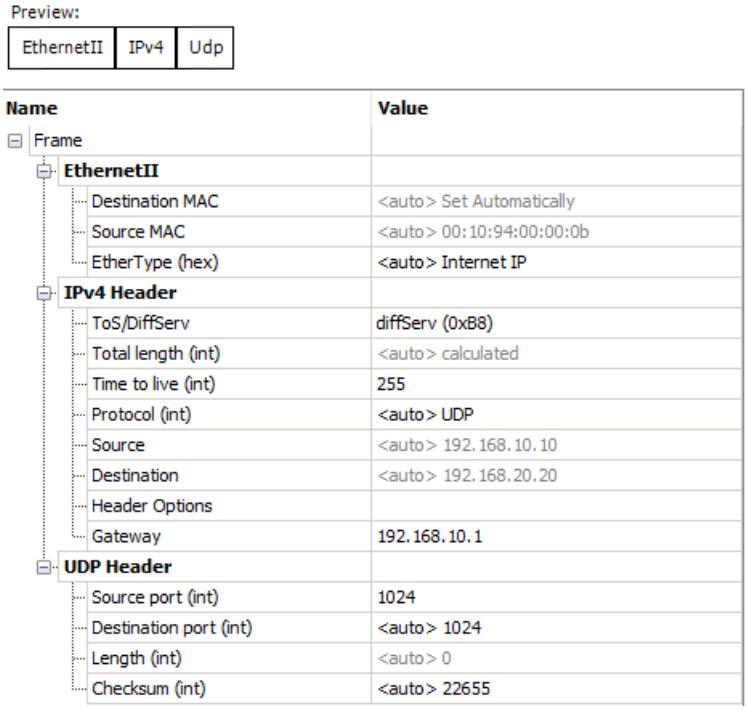


Figure 57 – Configuration des flux de trafic UDP sur le Spirent pour le trafic téléphonique

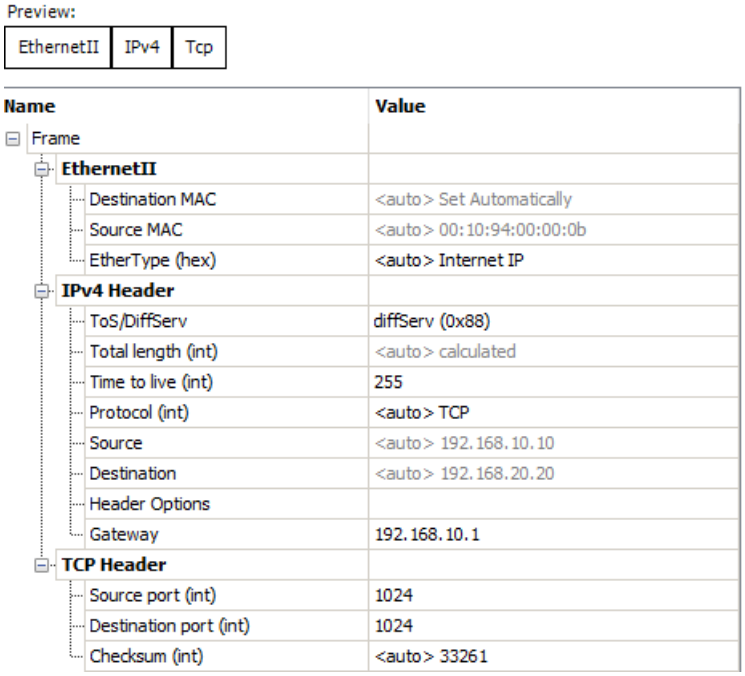


Figure 58 – Configuration des flux de trafic TCP sur le Spirent pour le trafic de télévision unicast

Pour chacun des types de flux de trafic, il faut mettre en place notre première stratégie de QoS qui est le fait de marquer les paquets lors de leur génération. Nous pouvons voir dans les 2 figures précédentes le champ "diffServ", dans le IPv4 Header, avec une valeur comme nous l'avons défini à la figure 54, qui permet de définir le PHB comme nous avons pu le voir dans la partie analyse.

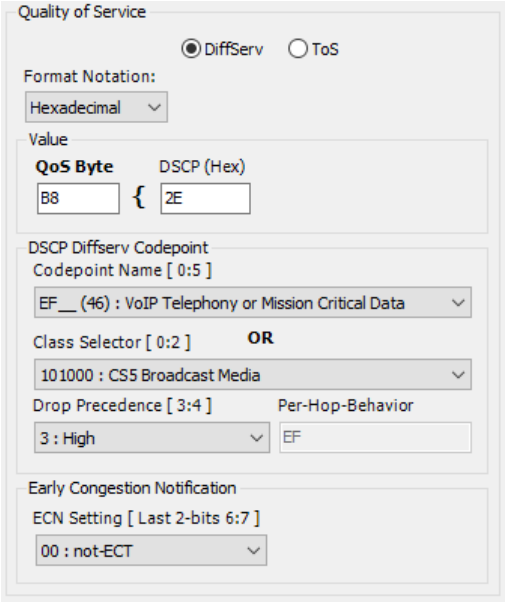


Figure 59 – Classification des paquets de voix

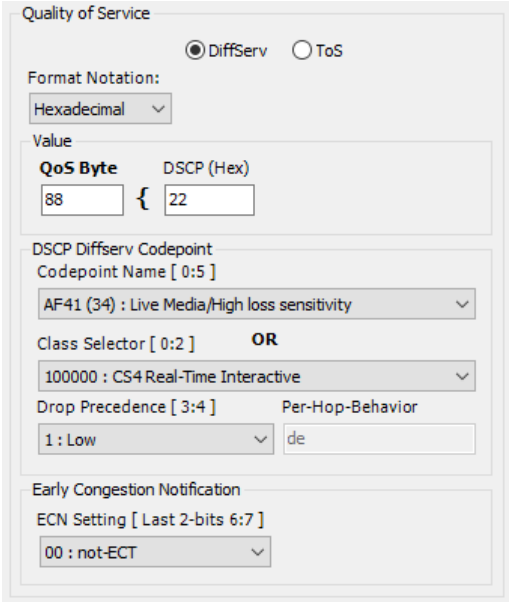


Figure 60 – Classification des paquets de télévision

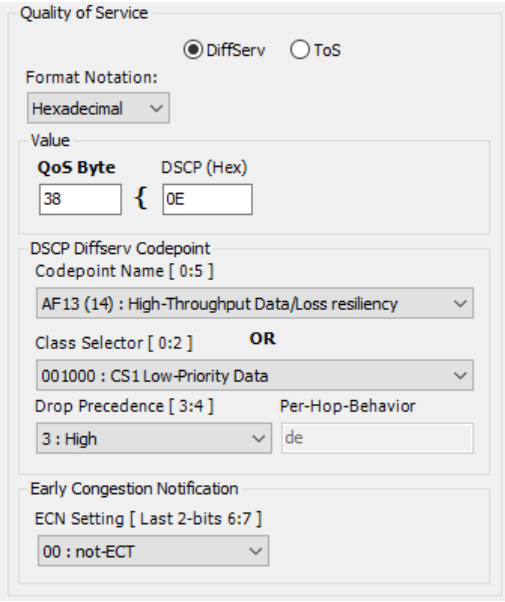


Figure 61 – Classification des paquets Internet

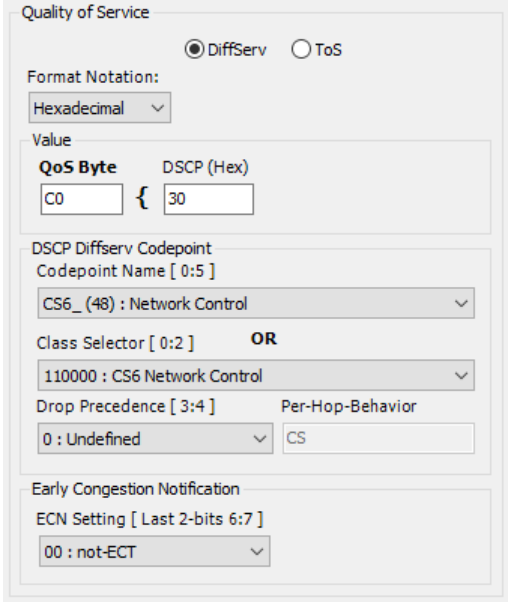


Figure 62 – Classification des paquets de management

Nous avons défini la classification des paquets en fonction de la figure 54, nous pouvons voir qu'une fois les champs "QoS Byte" et "DSCP (Hex)" spécifiés, le PHB est affiché et une indication sur son rôle est décrite, par exemple, pour le trafic téléphonique, le PHB "EF" est affiché suivi de l'indication "VoIP Telephony or Mission Critical Data", ce qui correspond à nos attentes.

4.2 Configuration des routeurs et des switchs

Pour la configuration des routeurs et des switchs, les attentes ont été décrites dans la partie conception. Pour cette partie de réalisation, nous ne mentionnerons pas les configurations complètes des équipements, car elles sont trop volumineuses. Les configurations des routeurs et des switchs sont disponibles, dans le catalogue, en annexe.

Pour toutes informations sur la configuration des routeurs, veuillez vous référer à la documentation en annexe "Quel est l'effet de la QoS sur des petits réseaux de labo ? - Mise en place réseau de tests et stratégies de QoS - Configuration Spirent C1 et routeur Cisco 2800 Series" réalisée durant ce projet de semestre 6 "Quel est l'effet de la QoS sur des petits réseaux de labo ?".

4.3 Conclusion de la réalisation

La partie réalisation de ce projet traitait essentiellement de la configuration du Spirent. Pour cet équipement, nous avons défini ses attentes dans la partie conception et dans cette partie de réalisation, nous avons décrit dans les grandes lignes les configurations mises en place. En effet, nous n'avons pas défini toutes les commandes et les manipulations nécessaires, car cela aurait été beaucoup trop volumineux et dans la partie conception. C'est pourquoi, les configurations des routeurs et des switchs sont disponibles en annexe dans le catalogue et pour la mise en place du réseau de tests et des stratégies de QoS, une documentation supplémentaire a été réalisée, comme mentionné précédemment, et disponible en annexe du projet.

5 Tests et validations

Avant de présenter les différents tests et leurs résultats, pour une meilleure vue d'ensemble, nous allons présenter tous les tests réalisés durant ce projet.

Les tests mentionnés ci-dessous ne sont pas tous présentés dans cette partie. En effet ,certains tests se comportent de manière normale ou nous ont permis de penser à d'autres tests, mais ils sont disponibles dans le fichier Excel en annexe.

Voici le catalogue des différents scénarios des tests effectués sur les différents réseaux de tests:

Etape	Test	Nombre de routeurs / switchs dans le réseau	Mesure	File d'attente	Génération du flux de trafic [%]	Ligne de sortie [Mb/s]	Temps de mesures [min]
1	1	1 (1x 2800)	Latence + perte	FIFO	100	100	10
		3 (3x 2800)	Latence + perte	FIFO	100	100	10
		1 (1x 2800)	Latence	FIFO	80	100	10
		1 (1x 2800)	Latence	FIFO	90	100	10
		1 (1x 2800)	Latence	FIFO	96	100	10
		1 (1x 2800)	Latence	FIFO	97	100	10
		1 (1x 2800)	Latence	FIFO	99	100	10
		3 (3x 2800)	Latence	FIFO	99	100	10
	2	1 (1x 2800)	Latence + perte	PQ	100	100	10
		1 (1x 2800)	Latence + perte	PQ	100	10	10
2	3	1 (1x 2900)	Latence	FIFO	100	100	10
		1 (1x 2900)	Latence + perte	FIFO	101	100	10
		1 (1x 2900)	Latence + perte	FIFO	200	100	10
Etape	**Test**	**Nombre de routeurs / switchs dans le réseau**	**Mesure**	**File d'attente**	**Génération du flux de trafic [%]**	**Ligne dans le réseau [Mb/s]**	**Temps de mesures [min]**
3	4	6 (2x 2800 + 2x 2900 + 2x 3560)	Latence + perte	FIFO	200	100	10
		6 (2x 2800 + 2x 2900 + 2x 3560)	Latence + perte	FIFO	200	10	10
		6 (2x 2800 + 2x 2900 + 2x 3560)	Latence + perte	FIFO	200	100 -> 0	10
4	5	6 (2x 2800 + 2x 2900 + 2x 3560)	Latence + perte	FIFO	200	100	10
		6 (2x 2800 + 2x 2900 + 2x 3560)	Latence + perte	CBWFQ	200	100	10
		6 (2x 2800 + 2x 2900 + 2x 3560)	Latence + perte	LLQ	200	100	10

Figure 63 – Catalogue des scénarios de tests effectués

À noter que dans notre cas, le pourcentage de génération du flux de trafic de 100% correspond à un débit de 100Mb/s, une génération de flux de trafic à 99% correspond donc à un débit de 99Mb/s et une génération de flux de trafic à 101% à un débit de 101Mb/s.

5.1 Étape 1

Pour la conception de cette étape, veuillez vous référer à la section 3.2.1 et pour la définissions des scénarios de tests, veuillez vous référer à la section 3.5.1.

5.1.1 Test 1: latence et perte en FIFO

Nombre de routeurs dans le réseau	File d'attente	Génération du flux de trafic [%]	Ligne du réseau [Mb/s]	Ligne de sortie (routeur à Spirent) [Mb/s]	Temps de mesures [min]
1 (1x Cisco 2800 Series)	FIFO	100	100	100	10
3 (3x Cisco 2800 Series)	FIFO	100	100	100	10

Figure 64 – Test 1.1: Configuration du réseau

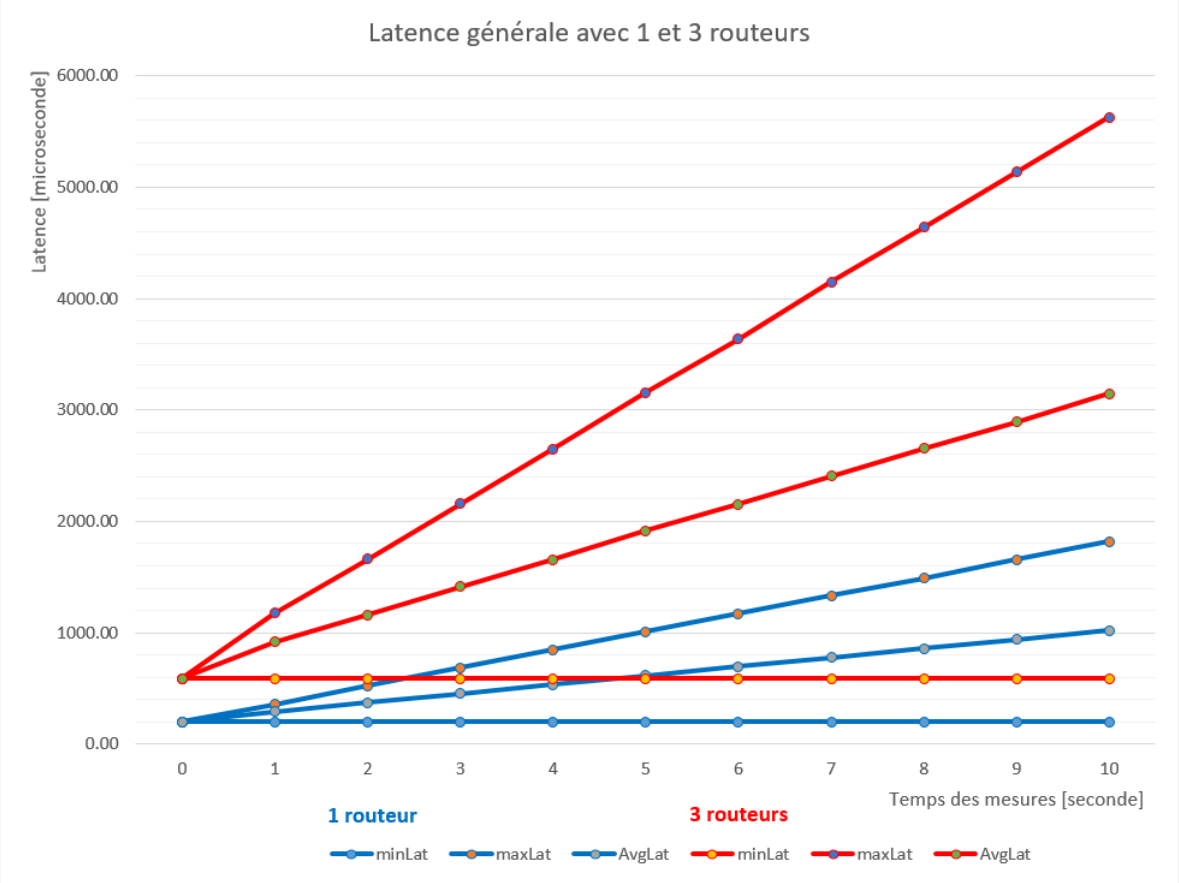


Figure 65 – Test 1.1: Latence générale sur 1 et 3 routeurs avec une génération de trafic à 100%, une ligne de sortie à 100Mb/s et une file d'attente FIFO

Avec ce précédent graphique, nous pouvons valider la première question posée lors de ce travail, est-ce que la latence augmente de manière proportionnelle au nombre de routeurs? Nous pouvons le confirmer. En effet, nous pouvons voir que la latence sur un réseau de 1 routeur est en moyenne de 1 milliseconde après 10 minutes de mesure et cette même latence est en moyenne de 3 millisecondes sur un réseau de 3 routeurs.

Par contre, ce comportement est étonnant, car en injectant 100% du trafic dans un réseau où toutes les lignes sont à 100Mb/s, nous nous attendions à ce que la latence reste stable à un seuil très bas, mais les files d'attente se remplissent quant même et la latence augmente.

Nous nous sommes aussi penchés sur la perte de paquets, les résultats sont excellents.

Port	Tx (compteur de trames)	Rx (compteur de trames)	Tx (compteur de bits) [Mb]	Rx (compteur de bits) [Mb]	Perte (trafic généré par le Spirent) [%]
3 (source)	**4'952'136**	73	**59'424,632**	0,061	**0**
4 (réception)	0	**4'952'209**	0	**59'425,693**	
3 (source)	**4'944'168**	149	**59'330,016**	0,120	**0**
4 (réception)	0	**4'944'318**	0	**59'330,504**	

Figure 66 – Test 1.1: Perte de paquets générale sur 1 et 3 routeurs avec une génération de trafic à 100%, une ligne de sortie à 100Mb/s et une file d'attente FIFO

Nous pouvons voir que dans les deux cas, sur un réseau à 1 routeur (en bleu) et sur un réseau à 3 routeurs (en rouge), nous enregistrons 0% du taux de perte, après 10 minutes de mesure.

Si nous prenons l'exemple du réseau de tests avec 3 routeurs, il est intéressant de noter que sur le port 3, qui correspond à la source du trafic (génération du trafic par le Spirent), nous enregistrons 149 trames au niveau de la réception, ces trames proviennent sûrement du management du réseau, car du côté de la source au port 4, qui correspond à la réception du trafic, aucune trame n'a été générée. Ce sont probablement des trames générées par les routeurs.

En admettant que 149 trames, en surplus du trafic générés par le Spirent, aient aussi été reçues à la réception sur le port 4, nous soustrayons ces 149 trames aux trames totales et nous obtenons 4'944'168 trames, ce qui correspond à une trame prête à la génération de trafic du Sirent, ce qui consolide encore plus notre 0% de perte.

Bien entendu, la perte de paquets arrivera lorsque les filles d'attentes arriveront à saturation maximum.

Au vu des résultats obtenus précédemment, nous avons injecté moins de pourcent de flux trafic afin de trouver le seuil où la latence reste stable. Nous avons injecté 80%, 90%, 96%, 97% et pour finir 99%.

Nombre de routeurs dans le réseau	File d'attente	Génération du flux de trafic [%]	Ligne du réseau [Mb/s]	Ligne de sortie (routeur à Spirent) [Mb/s]	Temps de mesures [min]
1 (1x Cisco 2800 Series)	FIFO	99	100	100	10
3 (1x Cisco 2800 Series)	FIFO	99	100	100	10

Figure 67 – Test 1.2: Configuration du réseau

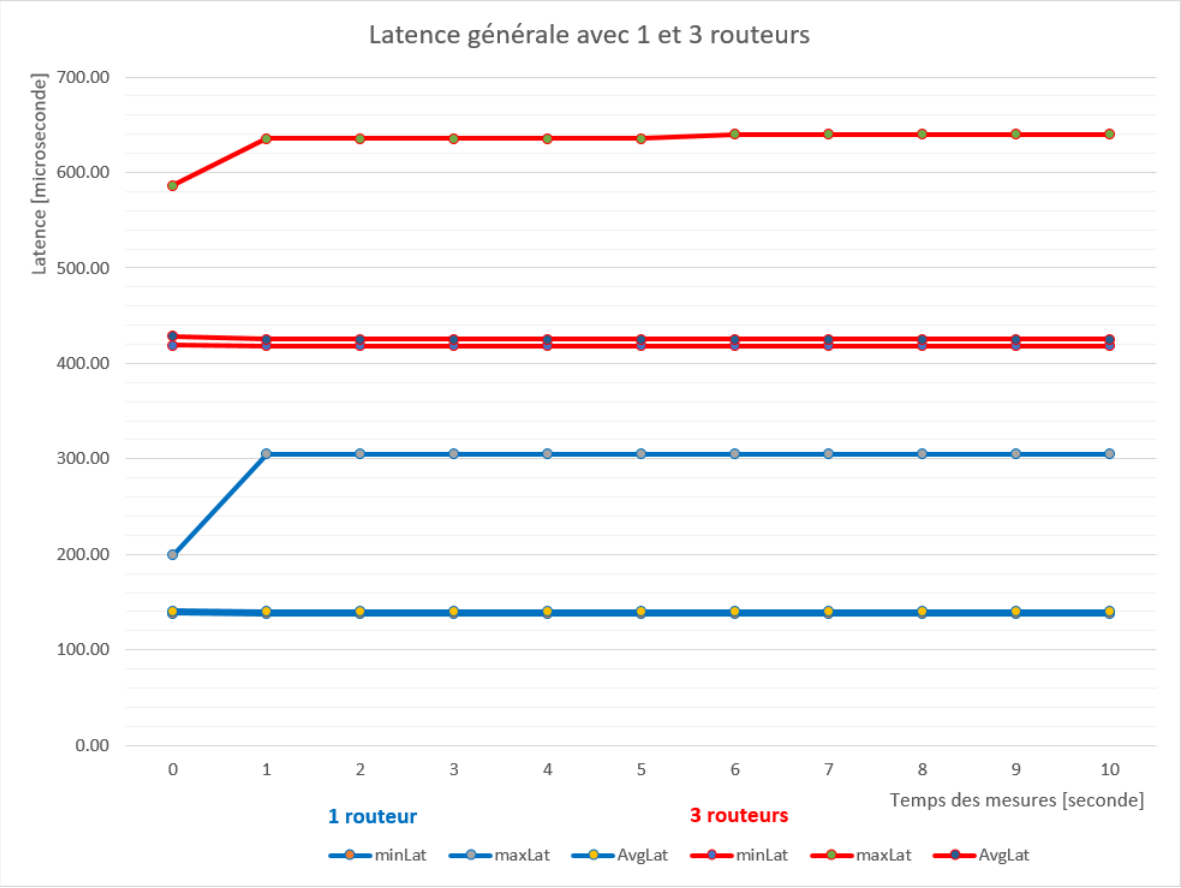


Figure 68 – Test 1.2: Latence générale sur 1 et 3 routeurs avec une génération de trafic à 99%, une ligne de sortie à 100Mb/s et une file d'attente FIFO

Précédemment, la latence augmentait avec un flux de trafic généré à 100%, maintenant avec un flux de trafic généré à 99%, nous pouvons voir que la latence se stabilise en moyenne à 0.14 milliseconde pour un réseau à 1 routeur et en moyenne à 0.42 milliseconde pour un réseau à 3 routeurs en ligne.

Cette mesure confirme une fois de plus que la latence augmente de manière proportionnelle au nombre de routeurs.

Naturellement, comme dans le test précédent, nous avons 0% du perte de paquets, après 10 minutes de mesure.

Port	Tx (compteur de trames)	Rx (compteur de trames)	Tx (compteur de bits) [Mb]	Rx (compteur de bits) [Mb]	Perte (trafic généré par le Spirent) [%]
3 (source)	4'894'029	71	58'728,348	0,060	0
4 (réception)	0	4'894'100	0	59'728,348	
3 (source)	4'898'540	132	58'782,480	0,105	0
4 (réception)	0	4'898'673	0	58'782,586	

Figure 69 – Test 1.2: Perte de paquets générale sur 1 et 3 routeurs avec une génération de trafic à 99%, une ligne de sortie à 100Mb/s et une file d'attente FIFO

Si nous faisons la relation entre le nombre de paquets émis sur le port 3 du test précédent, voir le tableau à la figure 66, et le nombre de paquets émis sur le port 3 du tableau ci-dessus en figure 69, nous arrivons bien à pourcentage de paquets émis de 99%.

Au passage, nous avons capturé les paquets afin de valider que le fait que les paquets soient bien classifié à la source:

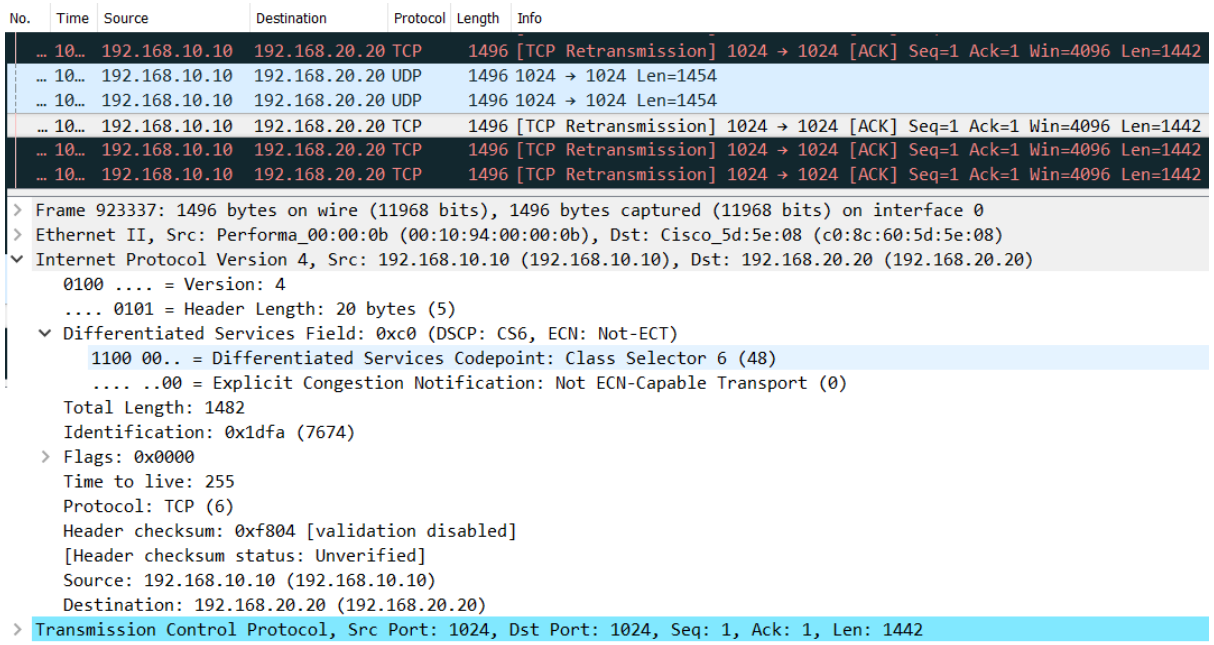


Figure 70 – Test 1.2: Analyse Wireshark des paquets générés

Nous pouvons voir nos paquets TCP et UDP, de la source 192.168.10.10 (hôte virtuel A du Spirent) à la réception 192.168.20.20 (hôte virtuel B du Spirent). En inspectant l'entête IPv4, le paquet TCP présenté ci-dessus est bien marqué avec le champ DSCP et une valeur "CS6", ce paquet fait parti du flux de trafic de management généré depuis le Spirent.

5.1.2 Test 2: latence et perte en PQ

Nombre de routeurs dans le réseau	File d'attente	Génération du flux de trafic [%]	Ligne du réseau [Mb/s]	Ligne de sortie (routeur à Spirent) [Mb/s]	Temps de mesures [min]
1 (1x Cisco 2800 Series)	PQ	100	100	100	10

Figure 71 – Test 2.1: Configuration du réseau

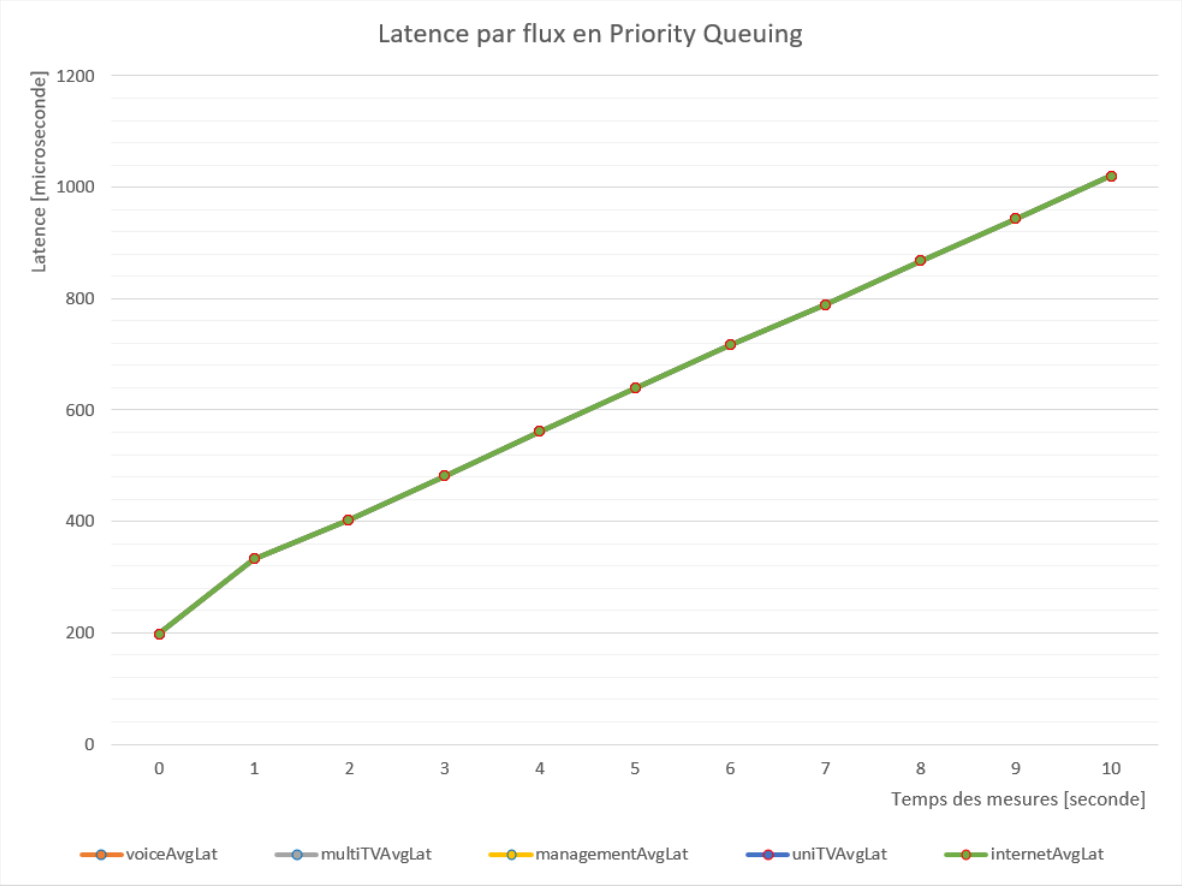


Figure 72 – Test 2.1: Latence par flux sur 1 routeurs avec une génération de trafic à 100%, une ligne de sortie à 100Mb/s et une file d'attente PQ

Nous obtenons le même effet que dans le test 1 de l'étape 1, la latence augmente régulièrement malgré le fait que les lignes soient toutes à 100Mb/s. Après 10 minutes de mesures, la latence moyenne générale est d'un peu plus de 1 milliseconde, exactement comme dans le test précédent, mais avec une file d'attente FIFO. Ce qui est intéressant, c'est que la stratégie de QoS mise en place, c'est-à-dire la file d'attente PQ, n'entre pas en action. En effet, nous pouvons voir que la latence moyenne reste la même pour les 5 types de flux. Apparemment tant que la file d'attente n'est pas saturée à un certain seuil ou que la latence n'atteint pas un certain temps, la stratégie PQ implémentée pour les files d'attente n'a aucun effet.

Une fois encore, il n'y a pas de perte de paquets.

Port	Tx (compteur de trames)	Rx (compteur de trames)	Tx (compteur de bits) [Mb]	Rx (compteur de bits) [Mb]	Perte (trafic généré par le Spirent) [%]
3 (source)	**4'957'761**	73	**59'493,132**	0,061	**0**
4 (réception)	0	**4'957'834**	0	**59'493,193**	

Figure 73 – Test 2.1: Perte de paquets générale sur 1 routeur avec une génération de trafic à 100%, une ligne de sortie à 100Mb/s et une file d'attente PQ

Flux de trafic	Tx (compteur de trames)	Rx (compteur de trames)	Tx (compteur de bits) [Mb]	Rx (compteur de bits) [Mb]	Perte (trafic généré par le Spirent) [%]
Téléphonie	39'982	39'982	479,784	479,784	0
TV multicast	39'982	39'982	479,784	479,784	0
TV unicast	2'798'735	2'798'735	33'584,820	33'584,820	0
Internet	2'039'080	2'039'080	24'468,960	24'468,960	0
Management	39'982	39'982	479,784	479,784	0

Figure 74 – Test 2.1: Perte de paquets par flux sur 1 routeur avec une génération de trafic à 100%, une ligne de sortie à 100Mb/s et une file d'attente PQ

Nous allons maintenant brider la ligne de sortie du routeur à 10Mb/s afin d'observer si la file d'attente PQ s'active bien lors d'une congestion.

Nombre de routeurs dans le réseau	File d'attente	Génération du flux de trafic [%]	Ligne du réseau [Mb/s]	Ligne de sortie (routeur à Spirent) [Mb/s]	Temps de mesures [min]
1 (1x Cisco 2800 Series)	PQ	100	100	10	10

Figure 75 – Test 2.2: Configuration du réseau

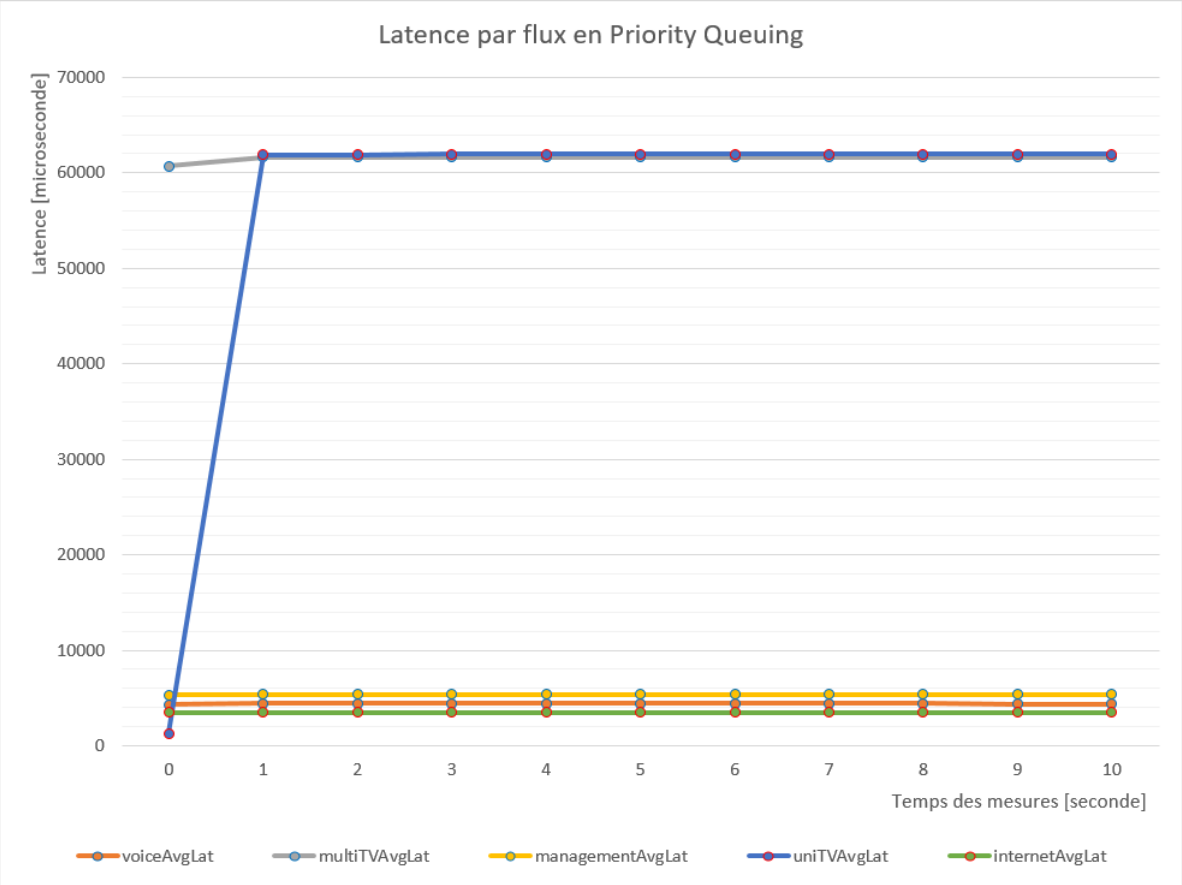


Figure 76 – Test 2.2: Latence par flux sur 1 routeurs avec une génération de trafic à 100%, une ligne de sortie à 10Mb/s et une file d'attente PQ

Première remarque, nous pouvons voir que notre file d'attente PQ s'est activée et fait son effet sur les différents flux de trafic, en effet les flux de trafic n'ont plus la même latence moyenne. Nous allons passer en revue les résultats et les comparer aux exigences minimales:

- La latence moyenne de la téléphonie (voiceAvgLat) est d'environ 4 millisecondes, si nous nous référons à la figure 13, pour une bonne qualité, la latence doit être en dessous de 150 millisecondes, pour ce test la latence est très bonne.
- Les latences moyennes de la télévision multicast et unicast (multiTVAvgLat et uniTVAvgLat) sont d'environ 62 millisecondes, si nous nous référons à la figure 14, pour une bonne qualité, la latence doit être en dessous de 4 secondes, pour ce test la latence est très bonne.
- La latence moyenne du management (managementAvgLat) est d'environ 5 millisecondes, ce qui est bon aussi.
- La latence moyenne du trafic Internet (internetAvgLat) est, à notre grand étonnement, aussi très bas, d'environ 4 millisecondes, ce qui est étrange vu que ce flux est le type de trafic avec le moins de priorité.

Nous avons vu que nous obtenons de bonnes latences en général, mais est-ce qu'une bonne latence veut forcément dire que tous les paquets ont atteint leur destination? Ce qui serait étrange pour le trafic Internet malgré une congestion que nous pouvons bien deviner vu le lien à 10Mb/s. Qui dit congestion, dit paquets supprimés, nous allons donc observer si nous avons des paquets supprimés et pour quel type de flux.

Port	Tx (compteur de trames)	Rx (compteur de trames)	Tx (compteur de bits) [Mb]	Rx (compteur de bits) [Mb]	Perte (trafic généré par le Spirent) [%]
3 (source)	**4'977'407**	72	**59'368,884**	0,060	**90,05**
4 (réception)	1	**494'910**	512	**5'938,105'432**	

Figure 77 – Test 2.2: Perte de paquets générale sur 1 routeur avec une génération de trafic à 100%, une ligne de sortie à 10Mb/s et une file d'attente PQ

Un taux de perte en général de 90%, ce qui est logique avec la ligne à 10Mb/s et le Spirent générant 100% de trafic, seuls 10% du trafic peut rejoindre la destination, les 90 autres pourcent sont donc supprimés.

Flux de trafic	Tx (compteur de trames)	Rx (compteur de trames)	Tx (compteur de bits) [Mb]	Rx (compteur de bits) [Mb]	Perte (trafic généré par le Spirent) [%]
Téléphonie	39'899	39'898	478,788	478'776	0,002
TV multicast	39'899	3'962	478,788	477,544	90,069
TV unicast	2'792'890	411'017	33'514,680	4'932,204	85,283
Internet	2'034'820	62	478'788	478'776	99,997
Management	39'899	39'898	479'784	0,744	0,002

Figure 78 – Test 2.2: Perte de paquets par flux sur 1 routeur avec une génération de trafic à 100%, une ligne de sortie à 10Mb/s et une file d'attente PQ

Pour rappel, lors de la configuration de la file d'attente PQ, nous avons défini 3 files d'attente, une priorité haute pour les flux téléphoniques et de management, une priorité moyenne pour les deux flux de télévision et une priorité normale pour le flux Internet.

Nous pouvons voir que les deux flux définis dans la file d'attente de hautes priorités ne subissent pratiquement aucune perte. En effet, dans les deux cas, téléphonie et management, sur une mesure de 10 minutes, 1 trame est perdue pour chacun, pratiquement 100% du trafic arrive à destination. Pour être de bonne qualité si nous nous référons à la figure 13, la téléphonie doit perdre moins d'un pourcent de paquets, les conditions sont ici remplies.

Les trafics de télévision subissent des pertes de 85 à 90%, ce qui n'est pas acceptable, car les exigences minimales sont une perte de moins de 5% du trafic pour une bonne qualité si nous nous référons à la figure 14.

Malgré une bonne latence, le trafic Internet, qui est le trafic avec le moins de priorité est perdu à pratiquement 100%, ce qui est compréhensible.

Nous pouvons nous pencher sur l'état de ces files d'attente sur le routeur en question.

```
Last clearing of "show interface" counters 00:10:12
Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 4452545
Queueing strategy: priority-list 1
Output queue (queue priority: size/max/drops):
   high: 0/20/0, medium: 0/40/2417797, normal: 0/60/2034748, low: 0/80/0
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 9443000 bits/sec, 804 packets/sec
```

Figure 79 – Test 2.2: Perte des paquets par file d'attente "show int fast0/1"

Nous retrouvons nos 3 files d'attente (high, medium et normal) et notre stratégie "priority-list 1" ou PQ, sur cette figure nous pouvons voir que le nombre total de paquets supprimés à l'entrée de la queue est de 4'452'545, ces paquets provenant des deux files d'attente moins prioritaires. Dans la file d'attente à haute priorité, le routeur indique 0 paquet perdu, contrairement au Spirent qui en présentait 2, mais ce sont de petits détails de précision.

À noter que les files d'attentes de sortir pouvant contenir le plus de paquets sont les moins prioritaires. Les files d'attente ne peuvent pas contenir énormément de paquets (80 paquets maximum pour la moins prioritaire et 20 paquets maximum pour la plus prioritaire). C'est intéressant, car en imaginant le nombre de paquets arrivant par seconde dans ces files d'attente, cela nous donne une idée de la puissance d'un routeur et de sa vitesse de traitement.

Les tests présentés dans le projet de semestre 5 de Monsieur Simon Lièvre "QoSLab" [1] sont identiques aux données mesurées dans ce travail, ces mesures sont cohérentes et nous pouvons en conclure que ces résultats sont logiques et corrects. À noter aussi que, la latence peut être faible, mais cela ne veut pas dire que tous les paquets arrivent à destination, nous l'avons vu avec le trafic Internet qui une obtient une bonne latence, mais qui atteint un taux de 100% de perte, cela vient du fait que les paquets ne sont pas gardés dans la file d'attente et sont supprimés directement.

5.2 Étape 2

Pour la conception de cette étape, veuillez vous référer à la section 3.2.2 et pour la définissions des scénarios de tests, veuillez vous référer à la section 3.5.2.

5.2.1 Test 3: latence et perte en FIFO

Nombre de routeurs dans le réseau	File d'attente	Génération du flux de trafic [%]	Ligne du réseau [Mb/s]	Ligne de sortie (routeur à Spirent) [Mb/s]	Temps de mesures [min]
1 (1x Cisco 2800 Series)	FIFO	100	100	100	10
1 (1x Cisco 2900 Series)	FIFO	100	100	100	10

Figure 80 – Test 3.1: Configuration du réseau

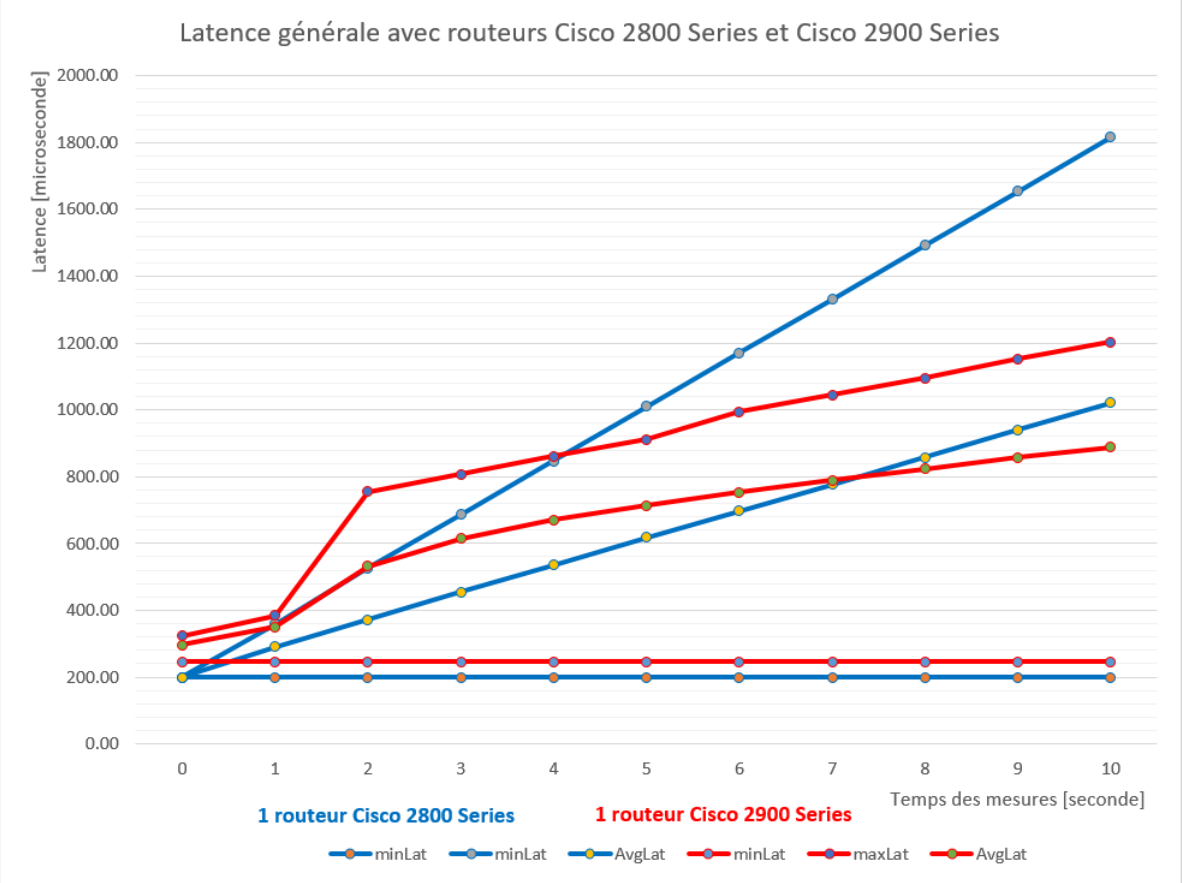


Figure 81 – Test 3.1: Latence générale sur 1 routeur Cisco 2800 Series et 1 routeur Cisco 2900 Series avec une génération de trafic à 100%, une ligne de sortie à 100Mb/s et une file d'attente FIFO

C'est intéressant, car ce précédent graphique représente la latence sur deux routeurs différents, le trafic injecté sur le réseau est exactement le même pour les deux cas, mais la latence ne se comporte pas de la même manière. Sur le routeur Cisco 800 Series, la latence augmente de manière constante sans écart, par contre sur le routeur Cisco 2900 Series, la latence n'augmente pas de manière constante, elle augmente rapidement les deux premières minutes, plus que sur le routeur Cisco 2800 Series, et ensuite elle augmente toujours, mais de manière beaucoup moins rapide. La latence moyenne du routeur Cisco 2900 Series repasse même en dessous de la latence moyenne du routeur Cisco 2800 Series après 7 minutes. On pourrait croire que le routeur Cisco 2900 Series ne travaille pas de manière régulière alors que le routeur Cisco 2800 Series fournit le même travail régulier. Pour rappel, le flux est injecté de manière constante, et ceci au maximum dès le début des mesures, il n'y a pas de variation.

Cette différence de latence provient du fait que les deux routeurs n'ont pas le même hardware. Le routeur Cisco 2800 Series (2801) intègre 2 ports FastEthernet (FE) 100BASE-T (RJ-45), ce qui permet de prendre en charge les implémentations suivantes:

- Transmission 100BASE-T à 100Mb/s en duplex intégral, prend en charge la longueur maximale des câbles Ethernet qui est de 100 mètres.
- Transmission 10BASE-T à 10Mb/s en duplex intégral, prend en charge la longueur maximale des câbles Ethernet qui est de 100 mètres.

Le routeur Cisco 2900 Series (2901) intègre 2 ports GigaEthernet (GE) 1000BASE-T (RJ-45) plus un module incluant 1 port GigaEthernet (GE) 1000BASE-T (RJ-45) en plus, ce qui permet de prendre en charge les implémentations suivantes:

- Transmission 1000BASE-T à 1Gb/s en duplex intégral, prend en charge la longueur maximale des câbles Ethernet qui est de 100 mètres
- Transmission 100BASE-T à 100Mb/s en duplex intégral, prend en charge la longueur maximale des câbles Ethernet qui est de 100 mètres
- Transmission 10BASE-T à 10Mb/s en duplex intégral, prend en charge la longueur maximale des câbles Ethernet qui est de 100 mètres

Comme nous pouvons voir, le routeur Cisco Series 2900 à une capacité supérieure, ce qui implique une vitesse de traitement des paquets supérieure, voilà pourquoi avec les mêmes débits, la latence est moins élevée sur ce routeur.

Les deux routeurs ne se comportent pas de la même manière, il faut tenir compte de ce fait lors de la comparaison des différents graphiques de résultats des tests menés dans ce projet.

À noter que nous avons appeler le service technique Cisco en Suisse pour avoir des informations sur les routeurs Cisco 2800 Series (2801) et Cisco 2900 Series (2901). Mais ces routeurs n'étant plus en vente, les techniciens n'ont pas pu répondre à nos questions.

Nous allons analyser les résultats d'une injection de trafic à plus de 100% dans le réseau.

Nombre de routeurs dans le réseau	File d'attente	Génération du flux de trafic [%]	Ligne du réseau [Mb/s]	Ligne de sortie (routeur à Spirent) [Mb/s]	Temps de mesures [min]
1 (1x Cisco 2900 Series)	FIFO	101	100	100	10

Figure 82 – Test 3.2: Configuration du réseau

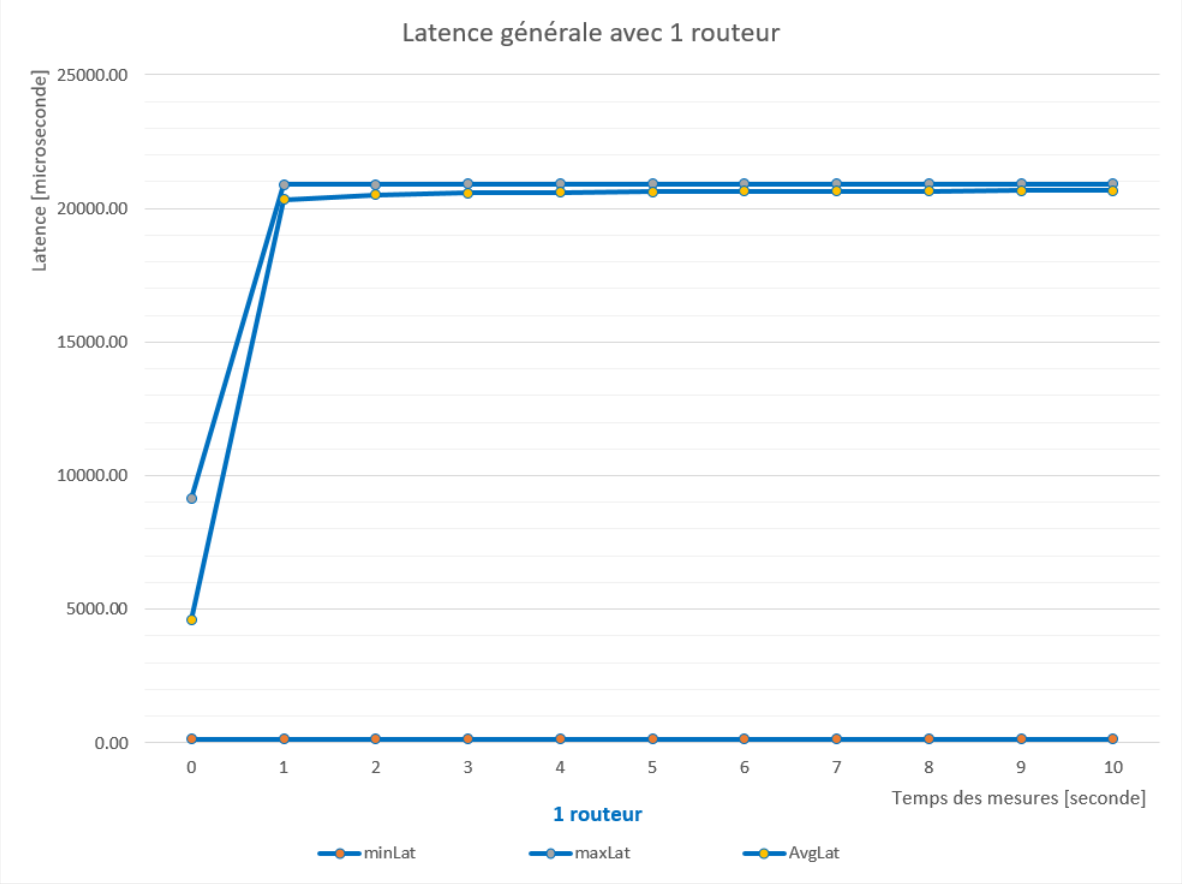


Figure 83 – Test 3.2: Latence générale sur 1 routeur Cisco 2900 Series avec une génération de trafic à 101%, une ligne de sortie à 100Mb/s et une file d'attente FIFO

Ces résultats sont assez étonnants, car en injectant 101% de trafic dans le réseau (50% avec l'hôte A du Spirent et 51% avec l'hôte B du Spirent), nous obtenons une latence pour le moins élevée, cette dernière atteint en moyenne 25 millisecondes après 10 minutes de mesures.

Nous pouvons donc supposer, d'après l'augmentation de la latence moyenne en 1 minute, puis sa stabilité après 1 minute, que le routeur à commencer le processus de suppression des paquets.

Port	Tx (compteur de trames)	Rx (compteur de trames)	Tx (compteur de bits) [Mb]	Rx (compteur de bits) [Mb]	Perte (trafic généré par le Spirent) [%]
3 (source)	**2'479'136**	73	**29'749,632**	0,064	**1**
5 (source)	**2'528'741**	73	**30'344,892**	0,065	
4 (réception)	0	**4'958'528**	0	**59'501,581**	

Figure 84 – Test 3.2: Perte de paquets générale sur 1 routeur avec une génération de trafic à 101%, une ligne de sortie à 100Mb/s et une file d'attente FIFO

Nous atteignons un taux de perte en général de 1%, une fois encore, ce résultat est logique avec la ligne à 100Mb/s et le Spirent générant 101% de trafic, seul 100% du trafic peut rejoindre la destination, les 1 autre pourcent sont donc supprimés.

Nous avons ou observer les cohérences suivantes.

- Génération de trafic du Spirent à 100% et ligne de sortie (routeur à Spirent) à 100Mb/s, nous obtenons 0% de perte
- Génération de trafic du Spirent à 99% et ligne de sortie (routeur à Spirent) à 100Mb/s, nous obtenons 0% de perte
- Génération de trafic du Spirent à 100% et ligne de sortie (routeur à Spirent) à 10Mb/s, nous obtenons 10% de perte
- Génération de trafic du Spirent à 101% et ligne de sortie (routeur à Spirent) à 100Mb/s, nous obtenons 1% de perte

Afin de confirmer les cohérences que nous avons pu observer, nous allons analyser la latence générale et la perte de paquets avec une injection de trafic du Spirent à 200% et une ligne de sortie (routeur à Spirent) à 100Mb/s, si nous suivons la logique des résultats obtenus, nous obtiendrons 50% de perte de paquets.

Nombre de routeurs dans le réseau	File d'attente	Génération du flux de trafic [%]	Ligne du réseau [Mb/s]	Ligne de sortie (routeur à Spirent) [Mb/s]	Temps de mesures [min]
1 (1x Cisco 2900 Series)	FIFO	200	100	100	10

Figure 85 – Test 3.3: Configuration du réseau

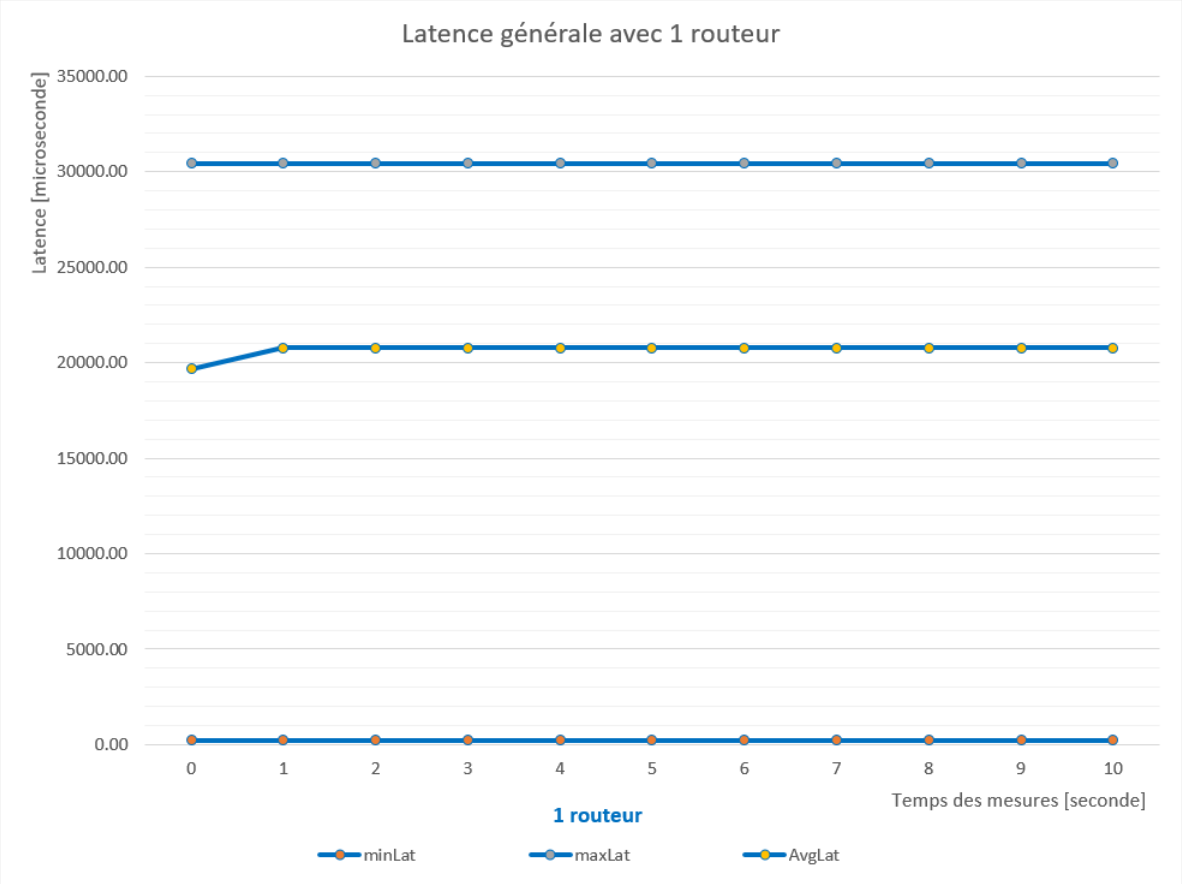


Figure 86 – Test 3.3: Latence générale sur 1 routeur Cisco 2900 Series avec une génération de trafic à 200%, une ligne de sortie à 100Mb/s et une file d'attente FIFO

Avec une injection de trafic à 200% dans le réseau, la latence moyenne est égale à la latence moyenne d'une injection de trafic à 101%. Nous pouvons en conclure qu'à partir de 101% de trafic injecté, la latence atteint son seuil maximum et n'augmente pas plus, tout se joue au niveau de la perte de paquets maintenant.

Pour rappel, d'après les résultats précédents, la logique voudrait que le taux de perte de paquets soit de 50% dans ce test.

Port	Tx (compteur de trames)	Rx (compteur de trames)	Tx (compteur de bits) [Mb]	Rx (compteur de bits) [Mb]	Perte (trafic généré par le Spirent) [%]
3 (source)	**4'947'921**	74	**59'375,052**	0,067	**50**
5 (source)	**4'947'800**	74	**59'373,600**	0,068	
4 (réception)	1	**4'948'099**	512	**59'376,464**	

Figure 87 – Test 3.3: Perte de paquets générale sur 1 routeur avec une génération de trafic à 200%, une ligne de sortie à 100Mb/s et une file d'attente FIFO

Le taux de perte de paquets est bien de 50%, nous allons analyser le routeur afin de voir s'il donne les mêmes chiffres.

```
Last clearing of "show interface" counters 00:10:40
Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 4947447
Queueing strategy: fifo
Output queue: 0/40 (size/max)
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 76270000 bits/sec, 6389 packets/sec
```

Figure 88 – Test 3.3: Perte des paquets par file d'attente "show int fast0/1"

Nous retrouvons notre stratégie de QoS avec une file d'attente FIFO et nous pouvons voir que le nombre total de paquets supprimés à l'entrée de la queue est de 4'947'447, ce qui correspond bien à 50% du trafic généré.

À noter qu'en FIFO nous avons bien une seule file d'attente en sortie pouvant contenir au maximum 40 paquets, contrairement aux 4 files d'attente configurées en PQ pouvant contenir réunies au maximum 200 paquets.

Avec ce test, nous pouvons conclure que la perte de paquets est proportionnelle au taux de trafic injecté dans le réseau et à la vitesse des liens dans le réseau.

5.3 Étape 3

Pour la conception de cette étape, veuillez vous référer à la section 3.2.3 et pour la définissions des scénarios de tests, veuillez vous référer à la section 3.5.3.

À noter que dans l'étape 3, le trafic généré est de 2x 100% soit 200Mb/s, mais notre nouveau réseau de tests est capable de recevoir et gérer une telle génération de trafic.

5.3.1 Test 4: latence et perte en FIFO

Nombre de routeurs dans le réseau	File d'attente	Génération du flux de trafic [%]	Ligne du réseau [Mb/s]	Ligne dans le réseau [Mb/s]	Temps de mesures [min]
6 (2x 2800 + 2x 2900 + 2x 3560)	FIFO	2X 100	100	100	10

Figure 89 – Test 4.1: Configuration du réseau

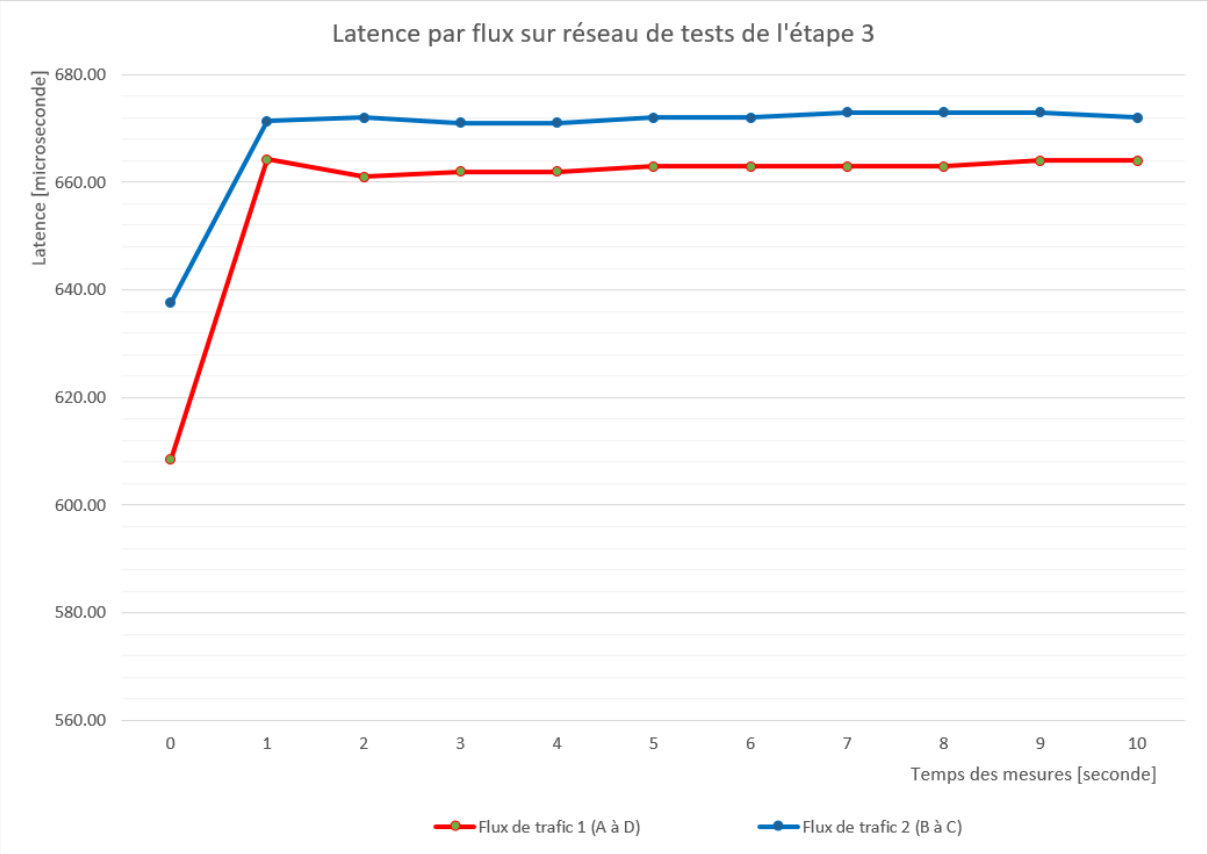


Figure 90 – Test 4.1: Latence par flux sur le réseau de tests de l'étape 3 avec une génération de trafic à 2x 100%, les lignes du réseau à 100Mb/s et des files d'attente FIFO

Ce graphique présente 2 courbes de mesures, en rouge, la latence moyenne par flux des flux allant de l'hôte virtuel A à D et en bleu, la latence par flux des flux allant de l'hôte virtuel B à C. Dans chacun des cas, la latence moyenne par flux est la même pour chacun des flux.

Ce graphique est particulièrement intéressant, car nous pouvons voir que la latence moyenne des flux allant de l'hôte virtuel A à D est moins élevée que la latence moyenne des flux allant de l'hôte virtuel B à C, pourtant sur les 2 hôtes virtuels, la configuration de la génération des flux est la même, mais nous obtenons environ 10 microsecondes de différences.

Un autre point est à relever, la latence moyenne devient stable très rapidement contrairement au graphique que nous avons pu observer à la figure 65.

Nous pouvons donc supposer, comme nous avons déjà pu l'observer, que d'après la stabilité de la latence moyenne, les routeurs atteignent un seuil maximal et suppriment des paquets. Nous allons analyser cette supposition.

Destination / Source	Flux de trafic	Tx (compteur de trames)	Rx (compteur de trames)	Tx (compteur de bits) [Mb]	Rx (compteur de bits) [Mb]	Perte (trafic généré par le Spirent) [%]
A -> D	Téléphonie	40'061	40'061	480,732	480,732	0
	TV multicast	40'061	40'061	480,732	480,732	0
	TV unicast	2'804'216	2'804'216	33'650,592	33'650,592	0
	Internet	2'043'072	2'043'072	24'516,864	24'516,828	0
	Management	40'061	40'061	480'732	480'732	0
B -> C	Téléphonie	40'061	40'061	480,732	480,732	0
	TV multicast	40'061	40'061	480,732	480,732	0
	TV unicast	2'804'241	2'804'241	33'650,892	33'650,892	0
	Internet	2'043'091	2'043'091	24'517,092	24'517,092	0
	Management	40'061	40'061	480,732	480,732	0

Figure 91 – Test 4.1: Perte de paquets par flux sur le réseau de tests de l'étape 3 avec une génération de trafic à 2x 100%, les lignes du réseau à 100Mb/s et des files d'attente FIFO

Nous atteignons un taux de perte totale de 0%. Le réseau de tests 3 a largement les moyens d'absorber le trafic généré depuis le Spirent. En effet, la latence est stable, elle n'augmente pas comme nous avons déjà pu l'observer et aucun paquet n'est supprimé.

Afin d'approfondir l'analyse des résultats précédemment obtenus, nous allons maintenant brider une ligne du réseau, plus précisément dans la boucle du réseau de tests 3 à 10Mb/s afin de voir si le réseau arrive toujours à gérer les 2X 100Mb/s, autrement dit 2X 100%, généré par le Spirent.

Nombre de routeurs dans le réseau	File d'attente	Génération du flux de trafic [%]	Ligne du réseau [Mb/s]	1 ligne dans le réseau [Mb/s]	Temps de mesures [min]
6 (2x 2800 + 2x 2900 + 2x 3560)	FIFO	2X 100	100	10	10

Figure 92 – Test 4.2: Configuration du réseau

C'est intéressant, car nous avons analysé les mesures au niveau de la latence par flux de trafic et de la perte de paquets, et nous obtenons exactement les mêmes résultats que le précédent test 5.3.1 (les résultats étant semblables, nous ne représentons pas le graphique). La latence est stable à, en moyenne, 670 microsecondes pour tous les flux et la perte de paquets est de 0%. Ces résultats sont étonnants, car comme notre réseau de tests peut absorber et gérer 200Mb/s de trafic alors que ces lignes sont toutes à 100Mb/s et même un lien est à 10Mb/s. Nous nous attendions à une perte de trafic de 45%.

Nous avons, en parallèle de ce test, tagué nos flux de trafic avec plusieurs VLANs afin d'inspecter, dans une capture de trafic, si ces VLANs sont bien associées aux paquets:

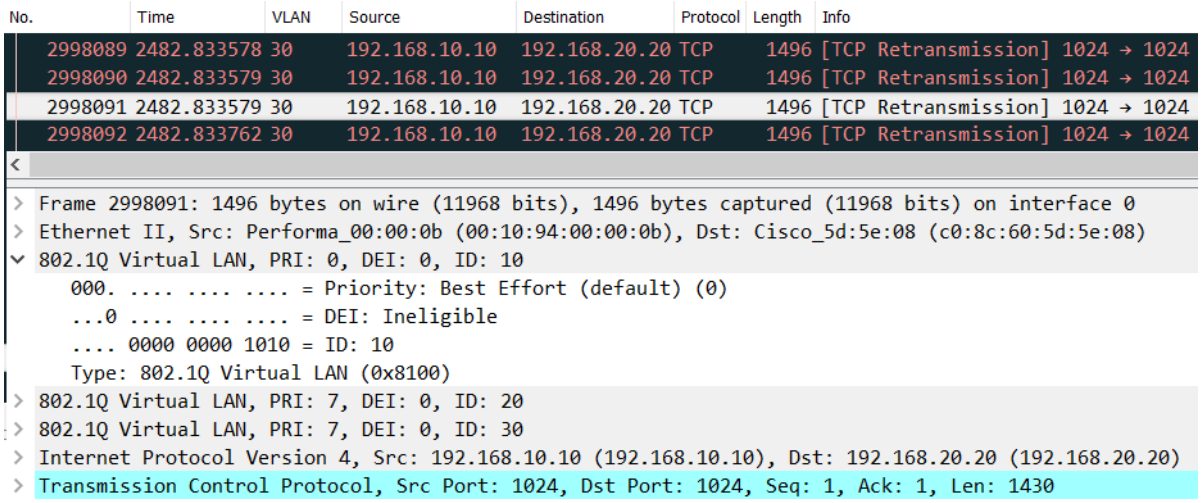


Figure 93 – Test 4.2: Analyse Wireshark des paquets générés

Nous pouvons voir nos paquets représentants nos différents flux de trafic. Une entête 802.1Q Virtual LAN est ajoutée pour chacune des VLANs associées au trafic. Dans notre cas, nous avons tagué nos paquets avec 3 VLANs et nous retrouvons ces 3 VLANs (10, 20, 30), associées à nos paquets.

Nous nous sommes donc penchés sur les tables de routage des routeurs R1 et R2, par quel chemin le trafic à destination de R1 ou R2 transite? Via R3 ou R4? Le lien étant bridé à 10Mb/s entre R2 et R4, tout le trafic passerait par R3? Mais comment R3 pourrait gérer 200Mb/s de trafic alors que ces liens sont à 100Mb/s?

```
R1#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static rout
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set

      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/24 is directly connected, GigabitEthernet0/0/0.1
L        192.168.1.1/32 is directly connected, GigabitEthernet0/0/0.1
      192.168.10.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.10.0/24 is directly connected, GigabitEthernet0/0/0.10
L        192.168.10.1/32 is directly connected, GigabitEthernet0/0/0.10
O     192.168.20.0/24 [110/3] via 192.168.50.2, 00:00:12, GigabitEthernet0/0
      192.168.30.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.30.0/24 is directly connected, GigabitEthernet0/0/0.30
L        192.168.30.1/32 is directly connected, GigabitEthernet0/0/0.30
O     192.168.40.0/24 [110/3] via 192.168.50.2, 00:00:12, GigabitEthernet0/0
      192.168.50.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.50.0/24 is directly connected, GigabitEthernet0/0
L        192.168.50.1/32 is directly connected, GigabitEthernet0/0
      192.168.60.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.60.0/24 is directly connected, GigabitEthernet0/1
L        192.168.60.1/32 is directly connected, GigabitEthernet0/1
O     192.168.70.0/24 [110/2] via 192.168.50.2, 00:01:07, GigabitEthernet0/0
O     192.168.80.0/24 [110/11] via 192.168.60.2, 00:00:12, GigabitEthernet0/1
```

Figure 94 – Test 4.2: Table de routage du routeur R1

```
R2#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set

      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/24 is directly connected, GigabitEthernet0/0/0.1
L        192.168.1.1/32 is directly connected, GigabitEthernet0/0/0.1
O     192.168.10.0/24 [110/3] via 192.168.70.2, 00:25:11, GigabitEthernet0/0
      192.168.20.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.20.0/24 is directly connected, GigabitEthernet0/0/0.20
L        192.168.20.1/32 is directly connected, GigabitEthernet0/0/0.20
O     192.168.30.0/24 [110/3] via 192.168.70.2, 00:25:11, GigabitEthernet0/0
      192.168.40.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.40.0/24 is directly connected, GigabitEthernet0/0/0.40
L        192.168.40.1/32 is directly connected, GigabitEthernet0/0/0.40
O     192.168.50.0/24 [110/2] via 192.168.70.2, 00:25:11, GigabitEthernet0/0
O     192.168.60.0/24 [110/3] via 192.168.70.2, 00:25:11, GigabitEthernet0/0
      192.168.70.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.70.0/24 is directly connected, GigabitEthernet0/0
L        192.168.70.1/32 is directly connected, GigabitEthernet0/0
      192.168.80.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.80.0/24 is directly connected, GigabitEthernet0/1
L        192.168.80.1/32 is directly connected, GigabitEthernet0/1
```

Figure 95 – Test 4.2: Table de routage du routeur R2

Sur les tables de routage précédentes, en nous référant aux adresses IP de sous-réseaux du schéma suivant, nous pouvons voir que sur les routeurs R1 et R2, le protocole de routage OSPF fait transiter les trafics en direction des sous-réseaux, dans lesquels ce trouve les hôtes virtuels, par le routeur R3. L'autre lien via le routeur R4 n'est pas viable du fait que le lien entre les routeurs R2 et R4 soit limité à 10Mb/s. Le fait que le trafic passe via le routeur R3 pour les communications entre les hôtes virtuels permet de comprendre pourquoi la ligne à 10Mb/s entre le routeur R2 et R4 n'a aucune influence sur le trafic.

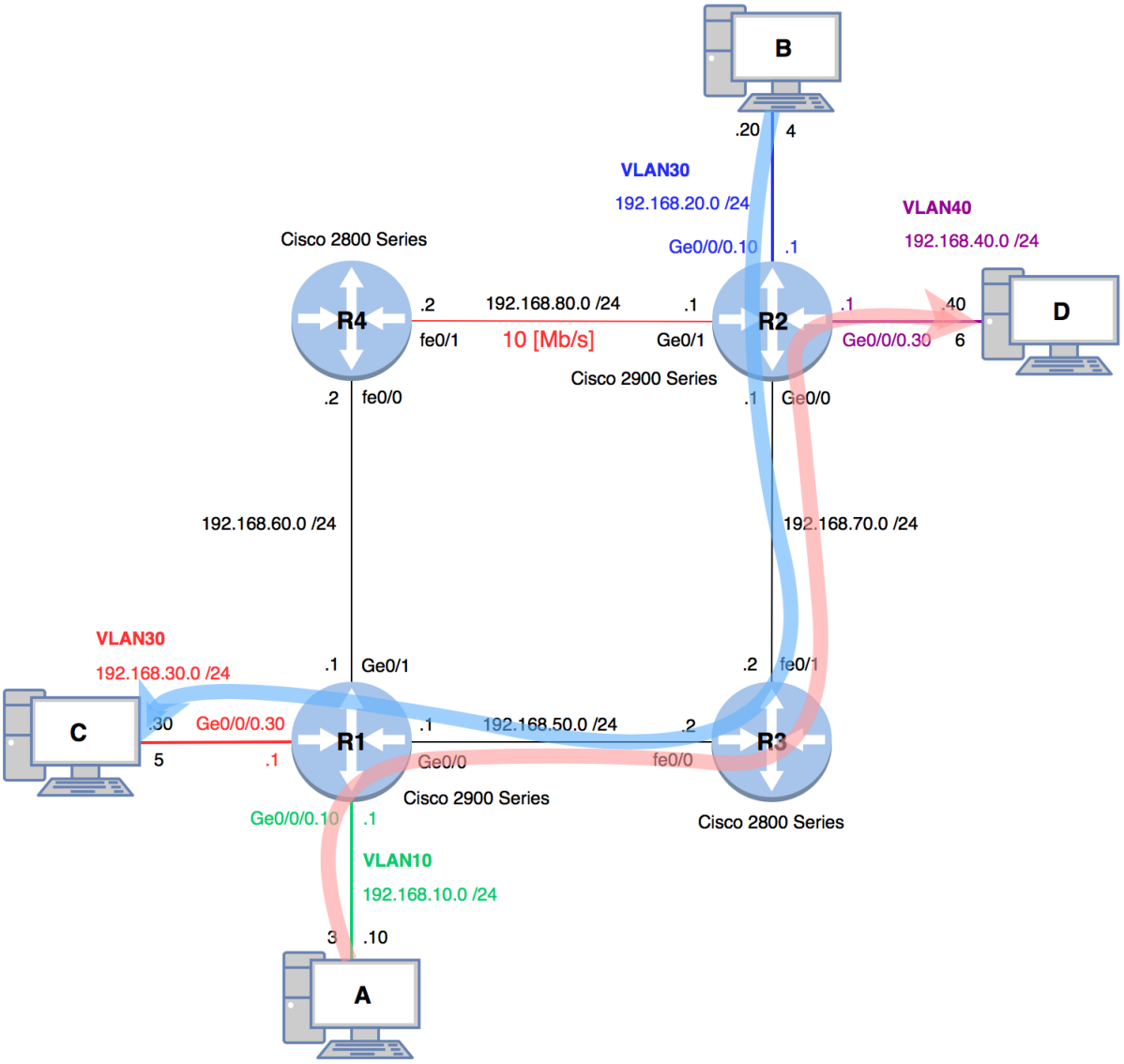


Figure 96 – Test 4.2: Trajet des flux sur le réseau de test 3

Maintenant pour la dernière question que nous nous posions sur le fait que le routeur R3 arrive à gérer 200Mb/s de trafic et que les liens du réseau sont à 100Mb/s. La réponse est simple, les interfaces des routeurs sont en mode "Full-Duplex", ce qui permet à ces dernières d'émettre à 100Mb/s et de recevoir à 100Mb/s simultanément. Les 200Mb/s de trafic se croisent et ne vont pas dans la même direction, c'est pourquoi les liens sont capables de supporter 100Mb/s dans un sens et 100Mb/s dans l'autre. Nous savons maintenant pourquoi la latence est stable, et pourquoi il n'y a pas de perte de paquets lors des tests à 200Mb/s de génération de trafic et avec un réseau de tests ayant une ligne à 10Mb/s.

C'est intéressant, car si nous analysons les tables de routage lorsque le lien entre les routeurs R2 et R4 est à 100Mb/s, le protocole OSPF propose deux routes pour le trafic entre les hôtes virtuels, soit via le routeur R3, soit via le routeur R4.

```
R1#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set

      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/24 is directly connected, GigabitEthernet0/0/0.1
L        192.168.1.1/32 is directly connected, GigabitEthernet0/0/0.1
      192.168.10.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.10.0/24 is directly connected, GigabitEthernet0/0/0.10
L        192.168.10.1/32 is directly connected, GigabitEthernet0/0/0.10
O     192.168.20.0/24 [110/3] via 192.168.60.2, 00:00:14, GigabitEthernet0/1
                      [110/3] via 192.168.50.2, 2d02h, GigabitEthernet0/0
      192.168.30.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.30.0/24 is directly connected, GigabitEthernet0/0/0.30
L        192.168.30.1/32 is directly connected, GigabitEthernet0/0/0.30
O     192.168.40.0/24 [110/3] via 192.168.60.2, 00:00:14, GigabitEthernet0/1
                      [110/3] via 192.168.50.2, 2d02h, GigabitEthernet0/0
      192.168.50.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.50.0/24 is directly connected, GigabitEthernet0/0
L        192.168.50.1/32 is directly connected, GigabitEthernet0/0
      192.168.60.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.60.0/24 is directly connected, GigabitEthernet0/1
L        192.168.60.1/32 is directly connected, GigabitEthernet0/1
O     192.168.70.0/24 [110/2] via 192.168.50.2, 2d02h, GigabitEthernet0/0
O     192.168.80.0/24 [110/2] via 192.168.60.2, 00:00:57, GigabitEthernet0/1
```

Figure 97 – Test 4.2: Table de routage du routeur R1

```
R2#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set

      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/24 is directly connected, GigabitEthernet0/0/0.1
L        192.168.1.1/32 is directly connected, GigabitEthernet0/0/0.1
O     192.168.10.0/24 [110/3] via 192.168.80.2, 00:05:23, GigabitEthernet0/1
                      [110/3] via 192.168.70.2, 2d02h, GigabitEthernet0/0
      192.168.20.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.20.0/24 is directly connected, GigabitEthernet0/0/0.20
L        192.168.20.1/32 is directly connected, GigabitEthernet0/0/0.20
O     192.168.30.0/24 [110/3] via 192.168.80.2, 00:05:23, GigabitEthernet0/1
                      [110/3] via 192.168.70.2, 2d02h, GigabitEthernet0/0
      192.168.40.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.40.0/24 is directly connected, GigabitEthernet0/0/0.40
L        192.168.40.1/32 is directly connected, GigabitEthernet0/0/0.40
O     192.168.50.0/24 [110/2] via 192.168.70.2, 2d02h, GigabitEthernet0/0
O     192.168.60.0/24 [110/2] via 192.168.80.2, 00:05:23, GigabitEthernet0/1
      192.168.70.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.70.0/24 is directly connected, GigabitEthernet0/0
L        192.168.70.1/32 is directly connected, GigabitEthernet0/0
      192.168.80.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.80.0/24 is directly connected, GigabitEthernet0/1
L        192.168.80.1/32 is directly connected, GigabitEthernet0/1
```

Figure 98 – Test 4.2: Table de routage du routeur R2

Le protocole de routage OSPF utilise l'algorithme Djikstra afin de calculer le chemin le plus court vers toutes les destinations connues. Chacun des routeurs découvre ces voisins et stocke ces informations dans leur table de voisins ("Neighbor Table"), puis ils échangent ces informations avec leurs voisins qui stockent ces informations reçues dans leurs bases de données de topologie du réseau ("Topology Database"). L'algorithme est exécuté afin de calculer le chemin ou la route la plus courte vers tous les sous-réseaux connus. Ces meilleures routes sont ensuite placées dans la table de routage des routeurs ("Routing Table"). Chaque routeur possède donc une "Neighbor Table", une "Topology Database" et une "Routing Table".

Avec OSPF, le chemin le plus court vers tous les sous-réseaux est calculé en fonction des coûts des interfaces. Ce coût d'une interface est inversement proportionnel à la bande passante de cette interface. Plus la bande passante d'une interface est élevée, plus le coût calculé est faible.

Dans notre réseau de tests, lorsque tous liens sont à 100Mb/s, les meilleures routes calculées entre les routeurs R1 (figure 97) et R2 (figure 98) peuvent donc soit être via le routeur R3 ou soit via le routeur R4. Cela vient du fait que les coûts des interfaces sont tous les mêmes. Nous pouvons voir sur la figure suivante que le coût des interfaces du lien entre les routeurs R2 et R4 est de 1, d'ailleurs pour toutes les interfaces du réseau, le coût est de 1 car toutes les interfaces sont à 100Mb/s.

```
R2#show ip ospf interface br
Interface    PID   Area            IP Address/Mask    Cost  State Nbrs F/C
Gi0/1        1     0               192.168.80.1/24    1     BDR   1/1
Gi0/0        1     0               192.168.70.1/24    1     BDR   1/1
Gi0/0/0.40   1     0               192.168.40.1/24    1     DR    0/0
Gi0/0/0.20   1     0               192.168.20.1/24    1     DR    0/0
R2#
R4#show ip ospf interface br
Interface    PID   Area            IP Address/Mask    Cost  State Nbrs F/C
Fa0/1        1     0               192.168.80.2/24    1     DR    1/1
Fa0/0        1     0               192.168.60.2/24    1     DR    1/1
```

Figure 99 – Test 4.2: Coûts des interfaces sur le routeur R2 et R4 avec un lien entre ces routeurs à 100Mb/s

Par contre, lorsque nous avons le lien entre les routeurs R2 et R4 à 10Mb/s, le coût de ces interfaces augmente, il passe à 10, c'est pourquoi cette route ne figure plus comme meilleure route entre les routeurs R1 (figure 94) et R2 (figure 95).

```
R2#show ip ospf interface br
Interface    PID   Area            IP Address/Mask    Cost  State Nbrs F/C
Gi0/1        1     0               192.168.80.1/24    10    WAIT  0/0
Gi0/0        1     0               192.168.70.1/24    1     BDR   1/1
Gi0/0/0.40   1     0               192.168.40.1/24    1     DR    0/0
Gi0/0/0.20   1     0               192.168.20.1/24    1     DR    0/0
R2#
R4#show ip ospf interface br
Interface    PID   Area            IP Address/Mask    Cost  State Nbrs F/C
Fa0/1        1     0               192.168.80.2/24    10    WAIT  0/1
Fa0/0        1     0               192.168.60.2/24    1     DR    1/1
```

Figure 100 – Test 4.2: Coûts des interfaces sur le routeur R2 et R4 avec un lien entre ces routeurs à 10Mb/s

Nous allons maintenant analyser les pertes de paquets dans un réseau avec des liens entièrement à 100Mb/s, donc avec un réseau pouvant absorber entièrement le trafic générer. Par contre sur notre mesure de 10 minutes, après 5 minutes, nous allons rompre un lien, dans l'anneau, par lequel le trafic transite. Nous pourrons analyser le mécanisme de redéfinition des routes.

Pendant la phase de conception correspondant à ce test, nous n'avions pas pensé qu'il faudrait jouer avec les coûts des interfaces, c'est pourquoi nous allons redéfinir le schéma correspondant à cette partie du test. Pour faire transiter le trafic via une seule route, nous allons modifier les coûts des interfaces d'un lien par lequel la seconde route transite. Par défaut toutes les interfaces auront un coût de 1 comme nous avons pu le voir, mais nous allons attribuer un coût de 10 sur le lien entre les routeurs R2 et R4.

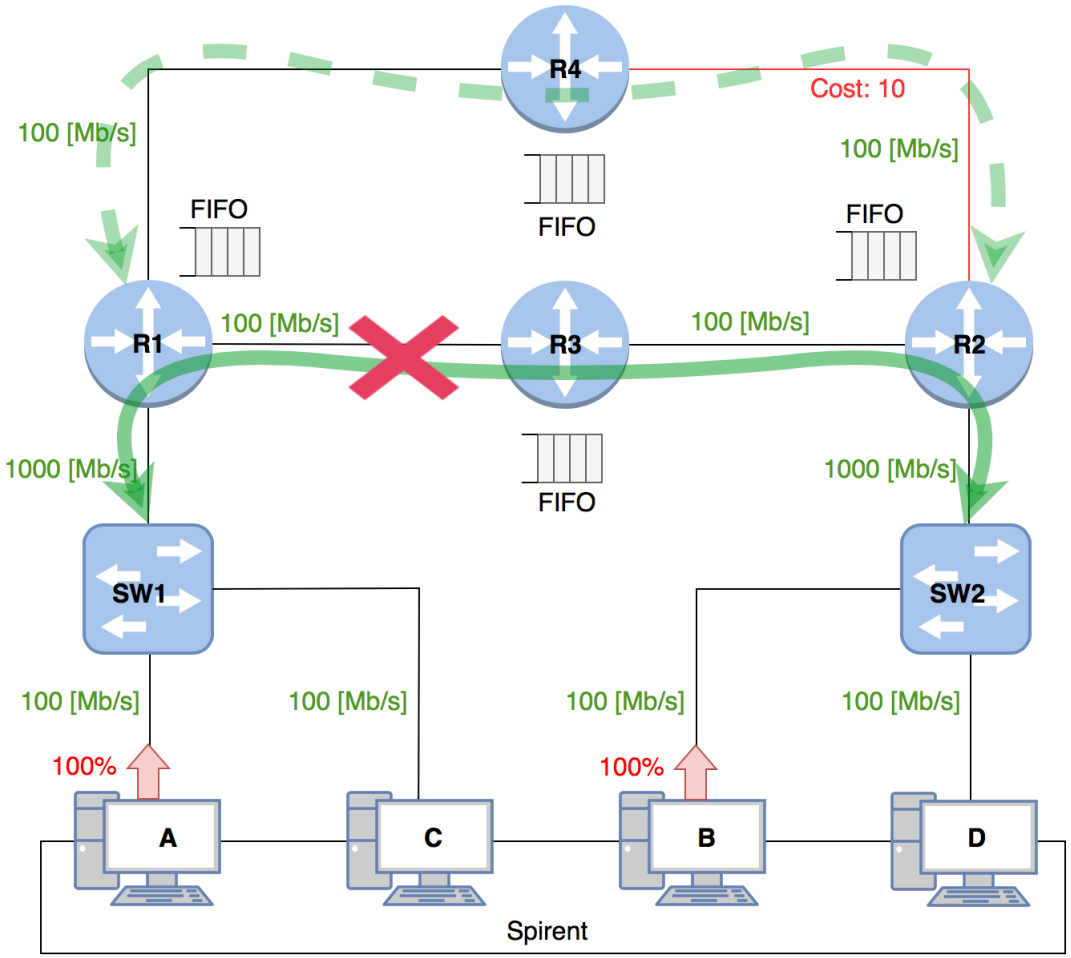


Figure 101 – Test 4.2: Modification du coût de l'interface entre les routeurs R2 et R4

Grâce aux commandes suivantes, nous pouvons changer les coûts OSPF sur les interfaces:

```
R2(config)# interface GigabitEthernet0/1
R2(config-if)# ip ospf cost 10
```

Nous avons entré cette commande sur les interfaces du lien entre les routeurs R2 et R4.

```
R2#show ip ospf interface br
Interface    PID   Area            IP Address/Mask    Cost  State Nbrs F/C
Gi0/1        1     0               192.168.80.1/24    10    WAIT  0/0
Gi0/0        1     0               192.168.70.1/24    1     BDR   1/1
Gi0/0/0.40   1     0               192.168.40.1/24    1     DR    0/0
Gi0/0/0.20   1     0               192.168.20.1/24    1     DR    0/0
R2#
R4#show ip ospf interface br
Interface    PID   Area            IP Address/Mask    Cost  State Nbrs F/C
Fa0/1        1     0               192.168.80.2/24    10    WAIT  0/1
Fa0/0        1     0               192.168.60.2/24    1     DR    1/1
```

Figure 102 – Test 4.2: Coûts des interfaces sur le routeur R2 et R4 avec un lien entre ces routeurs à 100Mb/s et une modification des coûts à 10

Les coûts des interfaces entre les routeurs R2 et R4 ont bien été modifiés à 10.

```
L        192.168.10.1/32 is directly connected, GigabitEthernet0/0/0.10
O     192.168.20.0/24 [110/3] via 192.168.50.2, 00:00:12, GigabitEthernet0/0
      192.168.30.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.30.0/24 is directly connected, GigabitEthernet0/0/0.30
L        192.168.30.1/32 is directly connected, GigabitEthernet0/0/0.30
O     192.168.40.0/24 [110/3] via 192.168.50.2, 00:00:12, GigabitEthernet0/0
      192.168.50.0/24 is variably subnetted, 2 subnets, 2 masks
```

Figure 103 – Test 4.2: Table de routage du routeur R1

```
L        192.168.1.1/32 is directly connected, GigabitEthernet0/0/0.1
O     192.168.10.0/24 [110/3] via 192.168.70.2, 00:25:11, GigabitEthernet0/0
      192.168.20.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.20.0/24 is directly connected, GigabitEthernet0/0/0.20
L        192.168.20.1/32 is directly connected, GigabitEthernet0/0/0.20
O     192.168.30.0/24 [110/3] via 192.168.70.2, 00:25:11, GigabitEthernet0/0
      192.168.40.0/24 is variably subnetted, 2 subnets, 2 masks
```

Figure 104 – Test 4.2: Table de routage du routeur R2

Grâce à cette modification des coûts, nous gardons un réseau à 100Mb/s et il n'y a qu'une route pour la communication entre les routeurs R1 et R2 ou entre les hôtes virtuels. Nous pouvons commencer notre test de rupture de lien afin d'analyser les conséquences sur les flux d'une redéfinition des routes.

Nombre de routeurs dans le réseau	File d'attente	Génération du flux de trafic [%]	Ligne du réseau [Mb/s]	1 ligne dans le réseau [Mb/s]	Temps de mesures [min]
6 (2x 2800 + 2x 2900 + 2x 3560)	FIFO	2X 100	100	100 -> 0	10

Figure 105 – Test 4.3: Configuration du réseau

Fait étonnant, la latence n'a pas augmenté après la rupture du lien et la redéfinition des routes. Nous ne présentons pas cette latence moyenne par flux, car elle se rapporte à la figure 90 présentée précédemment.

À 5 minutes de tests, les résultats concernant les pertes de paquets sont les mêmes qu'à la figure 91 présentée précédemment, nous obtenons 0% de perte de trafic. À 5 minutes, nous rompons le lien entre les routeurs R1 et R3 (lien par lequel le trafic transite) afin que la redéfinition des routes se fasse et que le trafic transite via le routeur R4.

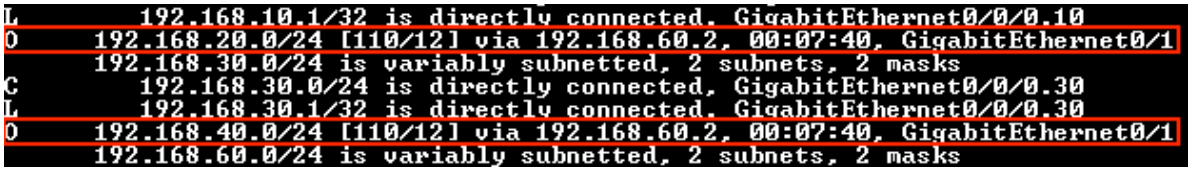


Figure 106 – Test 4.2: Table de routage du routeur R1 après rupture du lien entre les routeurs R1 et R3

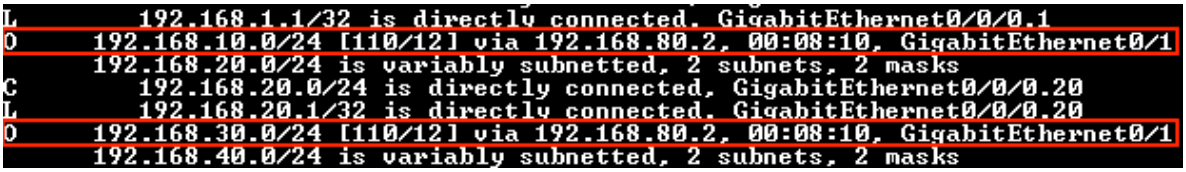


Figure 107 – Test 4.2: Table de routage du routeur R2 après rupture du lien entre les routeurs R1 et R3

Nous pouvons voir sur les tables de routage des routeurs R1 et R2 que les routes ont été redéfinies, et que cette fois, contrairement aux figures et où le trafic transite via le routeur R3, que le trafic transite maintenant via le routeur R4. Durant cette transition, la génération du flux de trafic n'a pas été arrêtée, nous nous attendons donc à une perte de paquets.

Destination / Source	Flux de trafic	Tx (compteur de trames)	Rx (compteur de trames)	Tx (compteur de bits) [Mb]	Rx (compteur de bits) [Mb]	Perte (trafic généré par le Spirent) [%]
A -> D	Téléphonie	39'868	39'319	478,416	471,828	1,377
	TV multicast	39'868	39'319	478,416	471,828	1,377
	TV unicast	2'790'743	2'752'290	33'488,916	33'027,480	1,377
	Internet	2'033'256	2'005'237	24'399,072	24'062,844	1,378
	Management	39'868	39'319	478,416	471,828	1,377
B -> C	Téléphonie	39'869	39'320	478,428	471,840	1,377
	TV multicast	39'869	39'320	478,428	471,840	1,377
	TV unicast	2'790'823	2'752'372	33'489,876	33'028,464	1,377
	Internet	2'033'314	2'005'300	24'399,768	24'063,600	1,377
	Management	39'869	39'320	478,428	471,840	1,377

Figure 108 – Test 4.1: Perte de paquets par flux sur le réseau de tests de l'étape 3 avec une génération de trafic à 2x 100%, une rupture de lien et une redéfinition des routes

Après 10 minutes de tests, nous atteignons un taux de perte totale de 1,377% et cette perte est de 1,377% pour tous les flux de trafic. Cette perte est due uniquement à la rupture du lien et à la redéfinition des routes, car ce même test effectué sans rupture de lien donnait 0% de perte de trafic. Cette perte de trafic de 1,377% est relativement élevée, avec un calcul, nous pouvons affirmer que la perte de trafic s'est effectuée durant 8,262 secondes. Nous pouvons donc supposé qu'entre la rupture du lien et le moment où la ligne redondante soit opérationnelle, il s'est passé 8,262 secondes, ce qui est relativement élevé.

Nous pouvons en conclure qu'une redéfinition des routes sur un réseau peut être délicate lorsque ce dernier est en mode production, un bon nombre de données peut être perdu.

Nous allons maintenant générer les flux de trafic depuis les hôtes virtuels A et C du Spirent.

5.4 Étape 4

Pour la conception de cette étape, veuillez vous référer à la section 3.2.4 et pour la définissions des scénarios de tests, veuillez vous référer à la section 3.5.4.

À noter que dans l'étape 4, le trafic généré est de 2x 100% soit 200Mb/s, mais notre nouveau réseau de tests est capable de recevoir et gérer une telle génération de trafic. .

5.4.1 Test 5: latence et perte en FIFO

Avant de commencer ce test, précisions que les priorités ajoutées précédemment ont été supprimées, nos tables de routage sont les suivantes sur les routeurs R1 et R2:

```
      192.168.10.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.10.0/24 is directly connected, GigabitEthernet0/0/0.10
L        192.168.10.1/32 is directly connected, GigabitEthernet0/0/0.10
O     192.168.20.0/24 [110/3] via 192.168.60.2, 00:00:14, GigabitEthernet0/1
                      [110/3] via 192.168.50.2, 2d02h, GigabitEthernet0/0
      192.168.30.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.30.0/24 is directly connected, GigabitEthernet0/0/0.30
L        192.168.30.1/32 is directly connected, GigabitEthernet0/0/0.30
O     192.168.40.0/24 [110/3] via 192.168.60.2, 00:00:14, GigabitEthernet0/1
                      [110/3] via 192.168.50.2, 2d02h, GigabitEthernet0/0
      192.168.50.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.50.0/24 is directly connected, GigabitEthernet0/0
L        192.168.50.1/32 is directly connected, GigabitEthernet0/0
```

Figure 109 – Test 5.1: Table de routage du routeur R1

```
      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/24 is directly connected, GigabitEthernet0/0/0.1
L        192.168.1.1/32 is directly connected, GigabitEthernet0/0/0.1
O     192.168.10.0/24 [110/3] via 192.168.80.2, 00:05:23, GigabitEthernet0/1
                      [110/3] via 192.168.70.2, 2d02h, GigabitEthernet0/0
      192.168.20.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.20.0/24 is directly connected, GigabitEthernet0/0/0.20
L        192.168.20.1/32 is directly connected, GigabitEthernet0/0/0.20
O     192.168.30.0/24 [110/3] via 192.168.80.2, 00:05:23, GigabitEthernet0/1
                      [110/3] via 192.168.70.2, 2d02h, GigabitEthernet0/0
      192.168.40.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.40.0/24 is directly connected, GigabitEthernet0/0/0.40
L        192.168.40.1/32 is directly connected, GigabitEthernet0/0/0.40
```

Figure 110 – Test 5.1: Table de routage du routeur R2

Pour la communication entre les hôtes virtuels les routes via les routeurs R3 et R4 sont disponibles. Bien entendu, les flux de trafic n'emprunteront qu'une seul de ces routes pour atteindre leur destination, nous nous attendons à une perte de 50% de trafic. Mais avec les files d'attente adaptée, nous allons réduire la perte de trafic pour les flux prioritaires.

Nombre de routeurs dans le réseau	File d'attente	Génération du flux de trafic [%]	Ligne du réseau [Mb/s]	Temps de mesures [min]
6 (2x 2800 + 2x 2900 + 2x 3560)	FIFO	2X 100	100	10
6 (2x 2800 + 2x 2900 + 2x 3560)	CBWFQ	2X 100	100	10
6 (2x 2800 + 2x 2900 + 2x 3560)	LLQ	2X 100	100	10

Figure 111 – Test 5.1: Configuration du réseau

Pour cette partie du test sur la présentation de la latence par flux, nous avons décidé de présenter des graphiques par types de flux en fonction des 3 types de files d'attente, ce qui facilitera la comparaison de latence entre les files d'attente pour un flux donné.

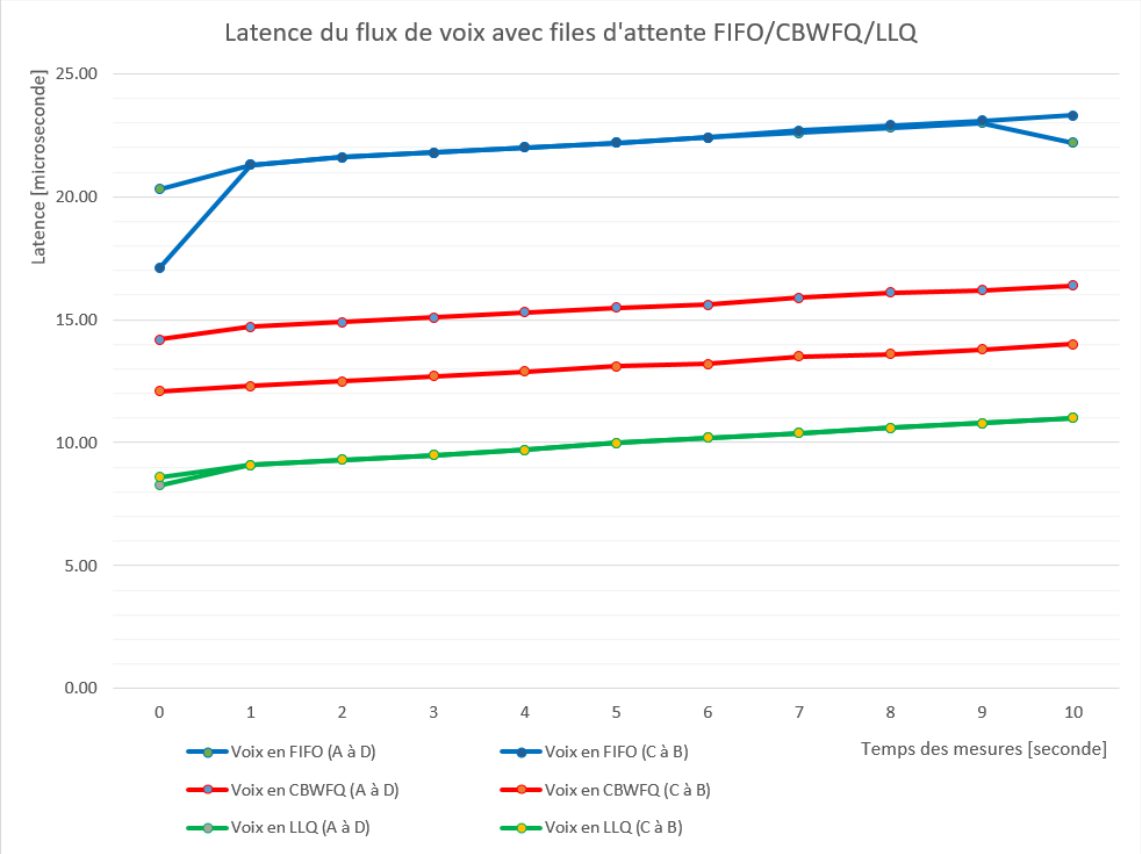


Figure 112 – Test 5.1: Latence du flux de voix sur le réseau de tests de l'étape 4 avec une génération de trafic à 2x 100%, les lignes du réseau à 100Mb/s et des files d'attente FIFO/CBWFQ/LLQ

Pour rappel, nous générons chaque fois 2 flux de trafic par catégorie, dans ce graphique, nous avons exécuté 3 mesures différentes, une fois avec un réseau implémentant des files d'attente FIFO, une fois des files d'attente CBWFQ et une fois des files d'attente LLQ. Par type de files d'attente, les 2 flux de voix indiquent la même latence, ce qui est logique et prouve que notre réseau est bien construit.

D'après notre analyse, pour les flux de voix, la latence maximum pour une bonne qualité est de 150 millisecondes. Un réseau avec des files d'attente FIFO fournit une latence moyenne de 0,02 à 0,025 milliseconde, ce qui est excellent. Pour un réseau implémentant des files d'attente CBWFQ ou LLQ, la latence moyenne est de 0,01 à 0.015 milliseconde, ce qui est encore mieux.

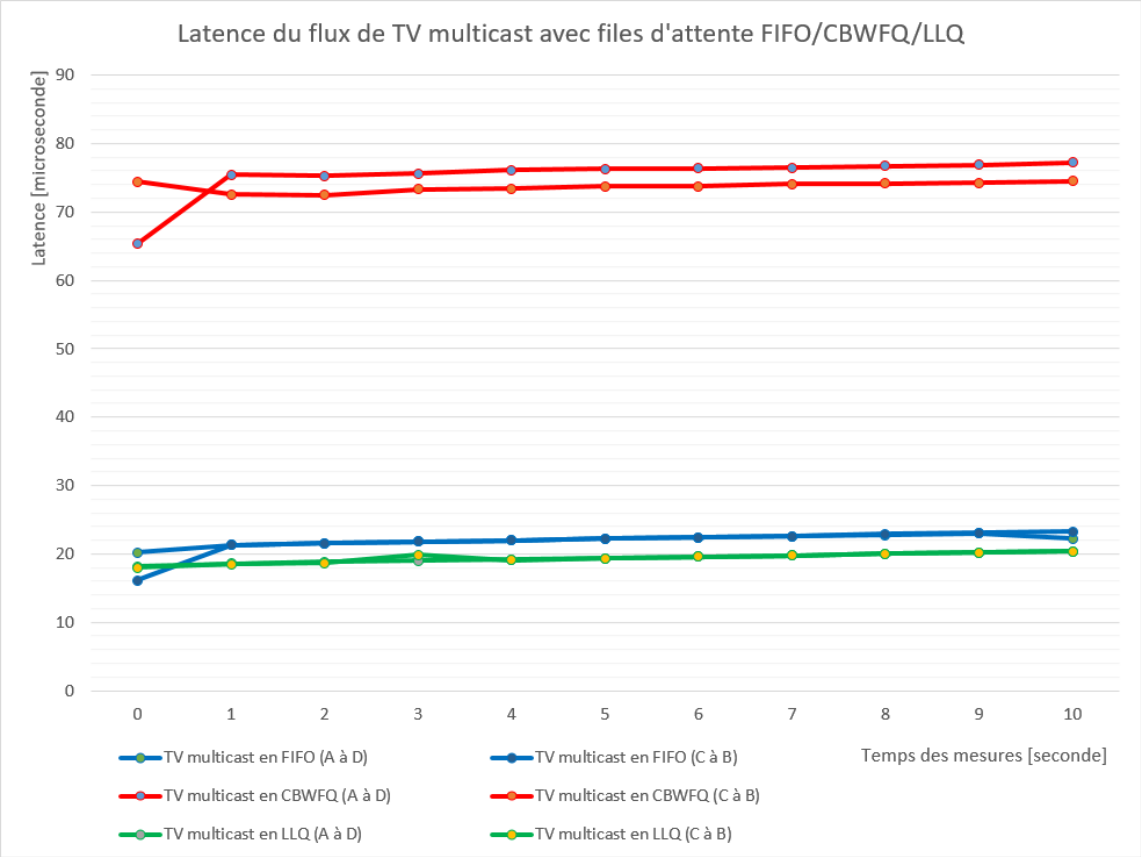


Figure 113 – Test 5.1: Latence du flux TV multicast sur le réseau de tests de l'étape 4 avec une génération de trafic à 2x 100%, les lignes du réseau à 100Mb/s et des files d'attente FIFO/CBWFQ/LLQ

D'après notre analyse, pour les flux vidéo, la latence maximum pour une bonne qualité est de 4000 à 5000 millisecondes. Un réseau avec des files d'attente FIFO ou LLQ fournit une latence moyenne pour les flux de télévision multicast d'environ 0,02 milliseconde, ce qui est excellent. Pour un réseau implémentant des files d'attente CBWFQ, la latence moyenne est de 0,07 à 0.08 milliseconde, ce qui un peu moins bien, mais tout de même excellent.

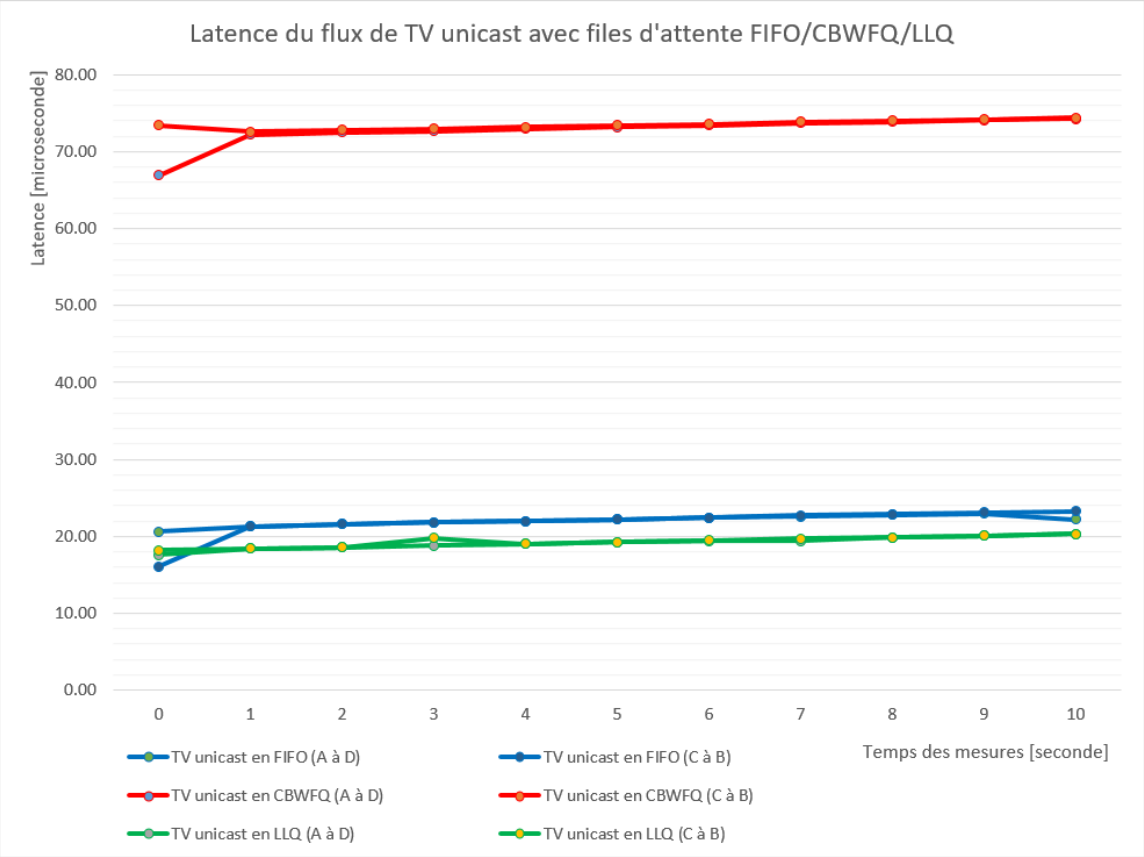


Figure 114 – Test 5.1: Latence du flux TV unicast sur le réseau de tests de l'étape 4 avec une génération de trafic à 2x 100%, les lignes du réseau à 100Mb/s et des files d'attente FIFO/CBWFQ/LLQ

Même chose pour les flux de télévision unicast, un réseau avec des files d'attente FIFO ou LLQ fournit une latence moyenne d'environ 0,02 milliseconde, ce qui est excellent. Pour un réseau implémentant des files d'attente CBWFQ, la latence moyenne est de 0,07 à 0.08 milliseconde, ce qui un peu moins bien, mais tout de même excellent. Les paquets des flux de télévision unicast et multicast étant marqués du même champ à la source, le même traitement leur est attribué, d'où cette même latence moyenne.

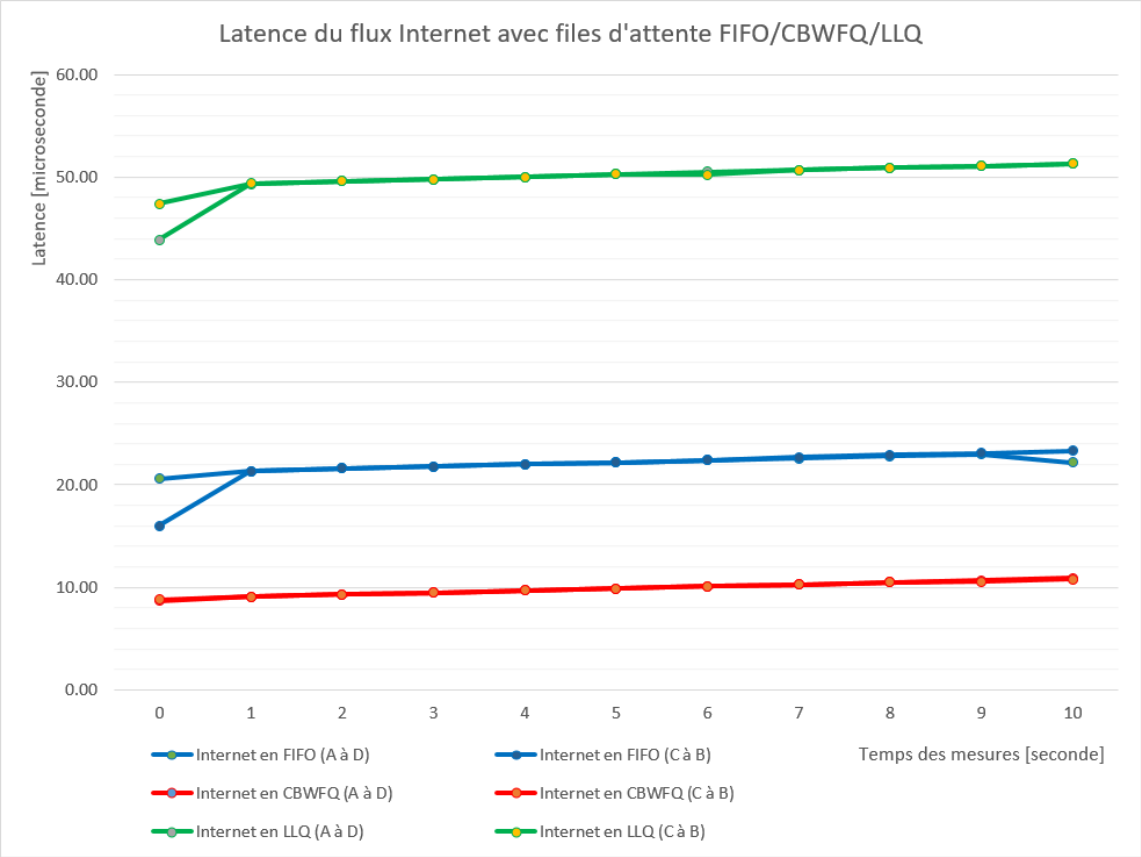


Figure 115 – Test 5.1: Latence du flux Internet sur le réseau de tests de l'étape 4 avec une génération de trafic à 2x 100%, les lignes du réseau à 100Mb/s et des files d'attente FIFO/CBWFQ/LLQ

Pour les flux Internet, dans notre réseau implémentant des files d'attente FIFO, la latence moyenne est de 0.02 milliseconde, pour des files d'attente CBWFQ, la latence moyenne est de 0.01 milliseconde et pour des files d'attente LLQ, la latence moyenne est de 0.05 milliseconde.

C'est intéressant, car ce n'est jamais le même type de files d'attente qui présente la meilleure latence.

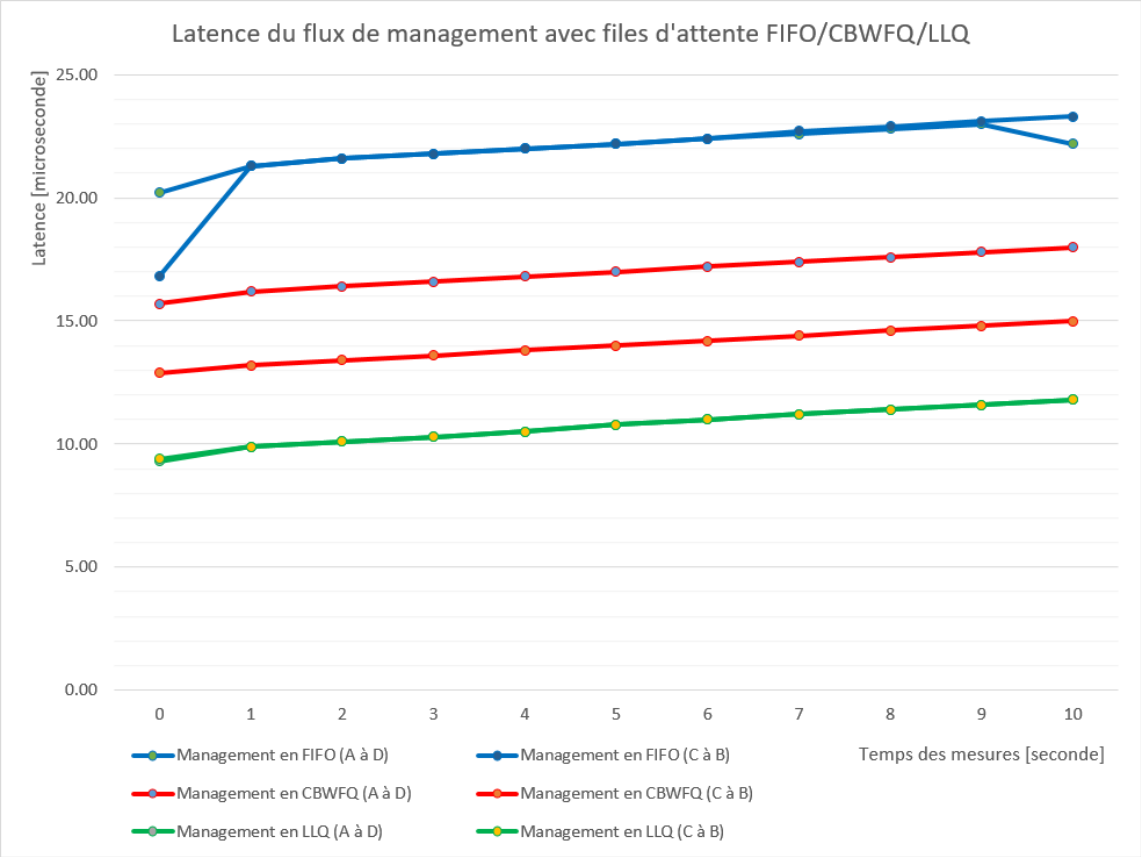


Figure 116 – Test 5.1: Latence du flux de management sur le réseau de tests de l'étape 4 avec une génération de trafic à 2x 100%, les lignes du réseau à 100Mb/s et des files d'attente FIFO/CBWFQ/LLQ

Pour les flux de management, dans notre réseau implémentant des files d'attente FIFO, la latence moyenne est de 0.02 à 0.025 milliseconde, pour des files d'attente CBWFQ, la latence moyenne est de plus ou moins 0.015 milliseconde et pour des files d'attente LLQ, la latence moyenne est de 0.01 milliseconde.

Ces latences sont excellentes pour tous les types de flux, mais nous devons maintenant analyser la perte de paquets, car une bonne latence pour un flux qui voit la moitié de ces paquets supprimés, ce n'est vraiment pas performant.

Destination / Source	Flux de trafic	Tx (compteur de trames)	Rx (compteur de trames)	Tx (compteur de bits) [Mb]	Rx (compteur de bits) [Mb]	Perte (trafic généré par le Spirent) [%]
A -> D	Téléphonie	39'953	32'484	479,436	389,808	18,694
	TV multicast	39'953	32'444	479,436	389,328	18,794
	TV unicast	2'769'640	2'274'916	33'559,680	27'298,992	17,862
	Internet	2'037'552	1'657'967	24'516,864	19'895,604	18,629
	Management	39'953	32'492	479,436	389,904	18,674
C -> B	Téléphonie	39'953	7'472	479,436	89,664	81,298
	TV multicast	39'953	7'503	479,436	90,036	81,220
	TV unicast	2'796'648	528'894	33'650,892	6'346,728	81,088
	Internet	2'037'558	385'694	24'517,092	4'628,328	81,070
	Management	39'953	7'640	479,436	91,680	80,8775

Perte totale moyenne : 50%

Figure 117 – Test 5.1: Perte de paquets par flux sur le réseau de tests de l'étape 4 avec une génération de trafic à 2x 100% et des files d'attente FIFO

Pour ce premier sous-test en mode file d'attente FIFO, nous remarquons bien que la perte moyenne totale entre tous les flux est de 50%.

Avec nos files d'attente FIFO, nous retrouvons le même phénomène que nous avons pu voir précédemment. Pour tous les flux de trafic des hôtes virtuels A à D, la perte est la même, dans notre cas en moyenne 18% et pour tous les flux de trafic des hôtes virtuels C à B la perte est la même, dans notre cas en moyenne 81%. Ce que nous pouvons souligner est cette énorme différence entre la perte de paquets des flux de trafic entre les hôtes virtuels A à D et C à B, pour le premier 18% et le second 81%, nous nous attendions à une perte équivalente de 50% et 50%, pourtant la configuration de tous les flux est la même.

Apparemment, les paquets des flux entre les hôtes virtuels C à B sont beaucoup plus supprimés, le type de file d'attente étant "premier arrivé, premier sorti", peut être que la cadence des flux entre les hôtes virtuels A à D est tout bonnement plus parfaite et évite la suppression.

Ce test ayant été réalisé plusieurs fois, nous sommes aussi arrivés à des résultats complètement différents où les paquets des flux entre les hôtes virtuels entre A à D étaient plus supprimés, ou des résultats où les paquets des flux entre les hôtes virtuels entre A à D et C à B étaient supprimées à 50% et 50%. Nous pouvons en conclure que ce phénomène est purement aléatoire.

Destination / Source	Flux de trafic	Tx (compteur de trames)	Rx (compteur de trames)	Tx (compteur de bits) [Mb]	Rx (compteur de bits) [Mb]	Perte (trafic généré par le Spirent) [%]
A -> D	Téléphonie	39'972	39'972	479,664	479,664	0
	TV multicast	39'972	4'958	479,664	59,820	87,596
	TV unicast	2'797'974	361'597	33'575,688	4'339,164	87,076
	Internet	2'038'525	2'038'523	24'462,300	24'462,276	0
	Management	39'972	39'972	479,664	479,664	0
C -> B	Téléphonie	39'972	39'971	479,664	479,652	0,002
	TV multicast	39'972	5'428	479,664	65,136	86,420
	TV unicast	2'797'976	360'774	33'575,712	4'329,288	87,105
	Internet	2'038'526	2'038'416	24'462,312	24'460,992	0,005
	Management	39'972	39'970	479,664	479,640	0,005

Perte totale moyenne : 50%

Figure 118 – Test 5.1: Perte de paquets par flux sur le réseau de tests de l'étape 4 avec une génération de trafic à 2x 100% et des files d'attente CBWFQ

Nous avons effectué le même test avec des files d'attente CBWFQ, la perte moyenne totale entre tous les flux est toujours de 50% du trafic.

Contrairement au premier sous-test, les différents mêmes flux de trafic présentent les mêmes pourcentages de perte de paquets. Nous pouvons voir que les flux de trafic correspondant à la voix et au management présentent 0% de perte de paquets, ce qui prouve que pour les flux sensibles, ce type de file d'attente fonctionne parfaitement. Pour les flux TV multicast et unicast, une perte de 87% est enregistrée, ce qui est médiocre vu que, d'après notre analyse, le maximum acceptable est de 5%, mais le réseau n'étant pas adapté, le flux sensible est prioritaire au détriment des flux vidéo. En revanche, les flux correspondant à Internet présentent aussi une perte de paquets de 0%, ce qui est étonnant.

Des réglages sont peut-être envisageables concernant l'implémentation de cette file d'attente.

Destination / Source	Flux de trafic	Tx (compteur de trames)	Rx (compteur de trames)	Tx (compteur de bits) [Mb]	Rx (compteur de bits) [Mb]	Perte (trafic généré par le Spirent) [%]
A -> D	Téléphonie	39'928	39'928	479,136	479,136	0
	TV multicast	39'928	27'156	479,136	325,872	31,987
	TV unicast	2'794'895	1'906'462	33'538,740	22'877,544	31,787
	Internet	2'036'282	471'342	24'435,384	5'656,104	76,852
	Management	39'928	39'928	479,136	479,136	0
C -> B	Téléphonie	39'928	39'928	479,136	479,136	0
	TV multicast	39'928	27'495	479,136	329,940	31,138
	TV unicast	2'797'976	1'905'806	33'539,172	22'869,672	31,886
	Internet	2'036'307	495'866	24'435,684	5'950,392	75,648
	Management	39'928	39'928	479,136	479,136	0

Perte totale moyenne : 50%

Figure 119 – Test 5.1: Perte de paquets par flux sur le réseau de tests de l'étape 4 avec une génération de trafic à 2x 100% et des files d'attente LLQ

Avec des files d'attente LLQ implémentées dans notre réseau, encore une fois nous obtenons une perte de trafic de 50%.

Chaque type de flux de trafic obtient, comme avec les files d'attente CBWFQ, le même pourcentage de perte de trafic. Pour la téléphonie et le management, nous obtenons encore 0% de perte de paquets ce qui est optimal pour ce type de trafic. Cette fois-ci la perte est plus élevée pour le trafic Internet qui est de 75%, contrairement au trafic vidéo, dont la perte de de 31% pour le trafic de TV multicast et unicast, ce qui est plus logique.

Ce type de files d'attente est plus optimal concernant le trafic vidéo, en effet le trafic Internet est davantage supprimé, contrairement à ce que l'on a pu voir avec les files d'attente CBWFQ, au détriment du trafic Internet. Ce type de files d'attente est optimal pour un réseau de tests de notre type.

Nous allons observer via quelle route le trafic transite et où ces 50% de trafic sont supprimés.

Port	Tx (compteur de trames)	Rx (compteur de trames)	Tx (compteur de bits) [Mb]	Rx (compteur de bits) [Mb]	Perte (trafic généré par le Spirent) [%]
A -> D C -> B	9'890'977	4'965'073	118'691,701	59'505,606	**50**

Figure 120 – Test 5.1: Perte de paquets totale pour les 3 types de files d'attente

En faisant le rapport entre le nombre total de paquets envoyés par la source et reçus par la destination.

```
GigabitEthernet0/0 is up, line protocol is up
  Hardware is CN Gigabit Ethernet, address is c08c.605d.5e08 (bia c08c.605d.5e08)
  Internet address is 192.168.50.1/24
  MTU 1500 bytes, BW 100000 Kbit/sec, DLY 100 usec,
     reliability 255/255, txload 217/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Full Duplex, 100Mbps, media type is RJ45
  output flow-control is unsupported, input flow-control is unsupported
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:06, output 00:00:01, output hang never
  Last clearing of "show interface" counters 00:10:35
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 4932085
```

Figure 121 – Test 5.1: Perte des paquets "show int fast0/0" sur le routeur R1

Nous avons inspecté tous toutes les suppressions de paquets sur les tables de routage des 4 routeurs du réseau, les paquets sont uniquement supprimés sur le routeur R1 et dans la file d'attente de l'interface "GigabitEthernet0/0", ce qui démontre que le trafic transite via le routeur R3 (figure 34). Dans cette file d'attente, 4'932'085 paquets sont supprimés, ce qui représente environ les 50% de paquets supprimés. Le trafic injecté étant de 200Mb/s et les lignes du réseau étant de 100Mb/s, 50% du trafic est logiquement supprimé sur le premier routeur, qui est dans notre cas le routeur R1.

Notre réseau de tests formant une boucle avec des liens de 100Mb/s pourrait potentiellement gérer une injection de trafic de 200Mb/s, mais le trafic n'empruntant qu'une route, sur les 2 routes disponibles, seuls 100Mb/s sont acheminés vers la destination d'où notre perte de 100%. Nous allons implémenter une technique permettant une répartition des charges vers les 2 routes via les routeurs R3 et R4 afin d'essayer de réduire la perte de trafic.

Le partage de charge Cisco Express Forwarding permet d'effectuer un équilibrage de charge grâce à l'utilisation de tables de partage de charge. Cet équilibrage de charge est implémenté sur les interfaces de sortie, dans notre cas sur les interfaces "GigabitEthernet0/0" (via le routeur R3) et "GigabitEthernet0/1" du routeur R1 (via le routeur R4). Nous pouvons répartir ces charges de 2 manières, par destinations (distribution des paquets en fonction de la destination) et par paquets (distribution des paquets aléatoirement). L'équilibrage de charge par destinations assure que le chemin destiné à une seule source ne devienne pas congestionné et l'équilibrage de charge par paquets n'est pas optimal pour certains flux sensibles qui doivent arriver à destination dans un ordre séquentiel.

1. **Interface:** Définir l'interface à configurer:

```
Cmd: R1(config)# interface "nom de l'interface"
Ex: R1(config)# interface GigabitEthernet0/0
```

2. **Interface:** Configurer la répartition des charges par destinations:

```
Cmd: R1(config-if)# ip load-sharing per-destination
```

2. **Interface:** Ou configurer la répartition des charges par paquets:

```
Cmd: R1(config-if)# ip load-sharing per-packets
```

Nous n'avons pas réussi à implémenter l'équilibrage de charge par destinations, les configurations ne sont pas prises en compte sur les routeurs Cisco 2900 Series. Nous avons donc implémenté sur les 2 interfaces "GigabitEthernet0/0" et "GigabitEthernet0/1" du router R1 l'équilibrage de charge par paquets et nous avons démarré le trafic sur le réseau de tests.

Port	Tx (compteur de trames)	Rx (compteur de trames)	Tx (compteur de bits) [Mb]	Rx (compteur de bits) [Mb]	Perte (trafic généré par le Spirent) [%]
A -> D C -> B	9'884'863	9'888'172	118'618,356	118'609,219	0

Figure 122 – Test 5.1: Perte de paquets totale pour les 3 types de files d'attente

Avec cette répartition des charges par paquets, notre réseau peut gérer un trafic de 200Mb/s, car le trafic est réparti sur les 2 routes disponibles, soit 50% du trafic via le routeur R3 et 50% du trafic via le routeur R4. Le partage de charge Cisco Express Forwarding répartit le trafic sur les 2 routes à part égales.

Le seul inconvénient que nous avons pu observer est le fait que la latence explose. En effet, nous avons réalisé ce test sur un réseau implémentant des files d'attente LLQ. Nous avions pu observer que pour un réseau composé de files d'attente LLQ, la latence moyenne pour tous les types flux était entre 10 et 50 microsecondes. Avec la répartition des charges, la latence est la même pour tous les types de flux et atteint en moyenne 820 microsecondes, ce qui n'est vraiment pas performant.

5.4.2 Test 5: synthèse

Cette partie ne traite pas de la répartition des charges, mais des résultats obtenus précédemment. Nous avons obtenu de bonnes latences pour tous les flux transitant dans un réseau implémentant soit des files d'attente FIFO, soit des files d'attente CBWFQ où soit des files d'attente LLQ. Mais pour affirmer que ce réseau est bien de bonne qualité, nous devons faire le rapport avec le pourcentage de paquets perdus.

FIFO

Un réseau avec des files d'attente FIFO fournit une latence moyenne de 0,02 à 0,025 milliseconde pour les flux de voix, ce qui est excellent, mais entre 18 et 81% de ces paquets sont perdus, ce qui n'est pas optimal du tout pour un flux sensible. Pour les flux vidéo, Internet et de management, là aussi entre 18 et 81% de ces paquets sont perdus, malgré une bonne latence d'en moyenne entre 0,02 et 0,025 milliseconde, ce n'est vraiment pas optimal.

Comme nous avons pu le voir, les pertes de paquets en FIFO sont vraiment aléatoires, pour un réseau performant, il n'est vraiment pas recommandé de se fier à ce type de files d'attente, car les flux sensibles sont traités de la même manière que tous les autres flux.

CBWFQ

Un réseau avec des files d'attente CBWFQ fournit une latence moyenne de 0,02 à 0,025 milliseconde pour les flux de voix, ce qui est excellent et tous les paquets arrivent à destination, ce qui n'est optimal pour un flux sensible. Les flux vidéo ont une latence moyenne de 0,07 à 0.08 milliseconde, ce qui est bon, mais environ 87% de ces paquets sont supprimés, ce qui n'est pas optimal. Les flux Internet ont une latence moyenne de 0,01 milliseconde, ce qui est excellent, et 0% de ces paquets sont supprimés, ce qui est vraiment performant pour ce flux uniquement, mais le flux Internet est supposé être le moins prioritaire, ce qui nous laisse perplexes. Les flux de management ont une latence moyenne d'environ 0,015 milliseconde et 0% de ces paquets sont supprimés, ce qui est vraiment performant pour le réseau, c'est ce que nous cherchons comme résultat.

Comme nous avons pu le voir, les flux Internet arrivent à pratiquement 100% à la destination au détriment des flux vidéo, ce qui n'est pas optimal malgré le fait que les flux sensibles ont une suppression de paquets nuls.

LLQ

Un réseau avec des files d'attente LLQ fournit une latence moyenne de 0,01 milliseconde pour les flux de voix, ce qui est excellent, car c'est la meilleure latence entre les 3 types de files d'attente et tous les paquets arrivent à destination, avec un taux de suppression nul, ce qui n'est optimal pour un flux sensible.

Les flux vidéo ont une latence moyenne de 0,01 milliseconde, ce qui est très bon, là aussi, il s'agit de la meilleure latence entre les 3 types de files d'attente et environ 31% de ces paquets sont supprimés, ce qui est mieux que les 87% que nous avons pu voir avec la file d'attente CBWFQ.

Les flux Internet ont une latence moyenne de 0,05 milliseconde, ce qui est aussi bon, mais il s'agit de la moins bonne latence entre les 3 types de files d'attente, et environ 76% de ces paquets sont supprimés, ce qui n'est pas performant, mais c'est un résultat de ce type que nous attendions contrairement au 0% des paquets supprimés pour ce flux en CBWFQ.

Les flux de management ont une latence moyenne d'environ 0,01 milliseconde, ce qui est aussi excellent et présente là aussi la meilleure latence pour les 3 types de files d'attente et 0% de ces paquets sont supprimés, ce qui est vraiment performant pour le réseau, c'est là aussi ce que nous cherchons comme résultat.

Comme nous avons pu le voir, les files d'attente LLQ présentent les meilleurs résultats pour un réseau de notre type. C'est pourquoi nous recommandons cette stratégie de QoS au détriment des 2 autres types de files d'attente que nous avons pu observer.

5.5 Conclusion des tests

Tous les tests que nous avons réalisés durant ce projet nous ont permis d'apprendre énormément sur le fonctionnement d'un réseau tant au niveau stratégie de QoS avec la classification de nos paquets et l'ordonnancement, et tant au niveau du protocole de routage OSPF. Nous pouvons sortir plusieurs points intéressants de la réalisation de tous ces tests.

Nous avons pu voir que l'implémentation des files d'attente CBWFQ, LLQ ou PQ sur un réseau est vraiment indispensable au bon fonctionnement et à la performance de ce dernier. En effet, par défaut, une file d'attente FIFO ne fait pas la différence entre les flux sensibles et les autres flux moins sensibles, le premier arrivé est le premier à sortir n'est pas une bonne façon de penser pour mettre en place un réseau performant. Nous avons aussi pu voir que ces files d'attente sont entièrement paramétrables et nous les configurons afin de décider quel flux passe avec un autre, dans notre cas, la voix et le management étaient définis comme flux sensible et prioritaire.

Comme nous avons pu l'observer dans le dernier test (chapitre 5.4), les files d'attente ne se comportent pas toujours de la manière dont nous nous attendions, comme le fait qu'un réseau implémentant les files d'attente CBWFQ traite avec une plus grande priorité les flux Internet que les flux vidéo, alors que le flux Internet est défini lors de sa classification moins prioritaire. Il est important de bien réfléchir et de bien tester ces stratégies de QoS sur un réseau de tests afin d'en définir la meilleure pour que le réseau soit performant en mode production.

Contrairement à ce que nous supposions avant de tester les files d'attente CBWFQ, LLQ ou PQ, ces files d'attente ne permettent pas d'injecter sur l'interface plus de paquets ou de les stocker afin de les injecter plus tard, non. Comme nous avons pu le voir dans le test précédent 5.4, un réseau composé de lignes à 100Mb/s implémentant par défaut des files d'attente FIFO recevant 200Mb/s de trafic aura, à la réception, une perte de paquets de 50%. Si nous implémentons dans ce même réseau des files d'attente CBWFQ, LLQ ou PQ, la perte de paquets ne sera pas diminuée, mais toujours de 50%, la perte de paquets sera drastiquement réduite pour les flux sensibles. Nous avons pu voir que la taille des files d'attente n'était pas vraiment grande, les files d'attente ne peuvent donc pas stocker pour un certain temps les paquets afin de les envoyer plus tard. Même effet avec un réseau en ligne ou bus, en injectant 100Mb/s de trafic et ce réseau contenant une ligne à 10Mb/s, la perte est dans tous les cas de 90%. Nous avons vu que ces pertes peuvent se calculer avec de simples règles de trois.

Nous avons aussi pu observer un phénomène pour le moins intéressant, lors de la réalisation de nos tests, nous nous penchions premièrement sur la latence générale ou des différents flux, lorsque nous obtenions de bonnes latences, au premier coup d'oeil nous étions satisfaits. Mais lorsque nous nous penchions sur la perte de paquets, elle était souvent élevée. En effet, nous avons pu observer que la plupart du temps, lorsque la latence était relativement élevée, les paquets des différents flux arrivaient sans perte à la destination et lorsque la latence était relativement basse, la perte de paquets était souvent élevée. Cette relation provient du fait que le paquet est souvent directement supprimé en arrivant dans la file d'attente lors des congestions, voilà pourquoi la latence est faible. Il est bien de toujours faire la relation entre latence et paquets perdus.

Un fait intéressant que nous avons pu analyser, dans un test précédent 5.3, est le fait qu'une redéfinition de routes lorsqu'un réseau est en mode production peut avoir des conséquences. En effet, sur un test de 10 minutes, une rupture d'un lien et un recalcul de route, nous a amené à une perte de trafic de 1,377%, ce qui n'est pas énorme, mais même en implémentant des files d'attente adéquates, les flux sensibles seront touchés autant que les autres flux. Nous supposons, d'après quelque calcul qu'entre le moment où le lien a été débranché et le moment où le réseau était de nouveau opérationnel, 8,262 secondes se soient écoulées, ce qui est relativement élevé. Nous avons aussi pu jouer avec les routes en forçant le trafic à passer par un chemin précis en implémentant des coûts OSPF sur les interfaces.

Nous avons implémenté une technique, de répartition des charges permettant à notre réseau de travailler à plein régime en absorbant et acheminant 200Mb/s de trafic, ce qui fut un succès au niveau du taux nul de perte de paquets, mais au niveau de la latence moyenne, ce n'étant pas satisfaisant. Ces méthodes de partage des charges sont encore à analyser et approfondir afin d'obtenir de meilleurs résultats.

Nous avons réalisé un tableau récapitulatif des différentes stratégies de QoS au niveau des files d'attente en rapport avec les différents flux. Pour rappel, lors de tous les tests que nous avons réalisés sur les différentes topologies, la classification des paquets et leur rapport entre eux au niveau du pourcentage généré et injecté dans le réseau sont restés les mêmes. Les files d'attente ne donneront pas les mêmes résultats en fonction d'autres configurations et des classifications des flux, c'est pourquoi il est bien de toutes les tester en fonction du type de réseaux et de ce que le réseau transporte comme trafic.

<table>
<tr><th></th><th>Voix</th><th>TV multicast</th><th>TV unicast</th><th>Internet</th><th>Management</th></tr>
<tr><td>FIFO</td><td>pas de priorité</td><td>pas de priorité</td><td>pas de priorité</td><td>pas de priorité</td><td>pas de priorité</td></tr>
<tr><td rowspan="2">CBWFQ</td><td rowspan="2">perte de paquet de 0% + très basse latence</td><td rowspan="2">haute latence comparé au autres files</td><td rowspan="2">haute latence comparé au autres files</td><td>très basse latence</td><td rowspan="2">perte de paquet de 0% + très basse latence</td></tr>
<tr><td>perte de paquet de 0%</td></tr>
<tr><td colspan="4">mauvais compromis des pertes de paquets</td></tr>
<tr><td rowspan="2">LLQ</td><td rowspan="2">perte de paquet de 0% + très basse latence</td><td>très basse latence</td><td>très basse latence</td><td>haute latence comparé au autres files</td><td rowspan="2">perte de paquet de 0% + très basse latence</td></tr>
<tr><td colspan="3">bon compromis des pertes de paquets</td></tr>
<tr><td rowspan="2">PQ</td><td rowspan="2">perte de paquet de 0% + très basse latence</td><td>trop grande latence</td><td>trop grande latence</td><td>perte de paquets de 100%</td><td rowspan="2">perte de paquet de 0% + très basse latence</td></tr>
<tr><td colspan="3">mauvais compromis des pertes de paquets</td></tr>
</table>

Figure 123 – Performance des files d'attente en fonction des différents types de flux de trafic

Ce tableau est réalisé en fonction des résultats obtenus au niveau de la latence et des pertes de paquets dans un réseau de tests congestionnés.

Pour nos réseaux de tests, nous avons pu confirmer que les files d'attente CBWFQ, LLQ ou PQ, d'après nos configurations, se comportent très bien avec les flux sensibles comme la voix ou le management. Avec le flux Internet, les files d'attente CBWFQ et PQ se comportent mal, car soit tous les paquets sont supprimés, soit tous les paquets arrivent à destination au détriment d'autres flux. Nous avons clairement pu voir que la file d'attente LLQ remplissait au mieux les exigences au niveau des pertes de paquets et de la latence.

Nous avons pu observer, au niveau de la latence, dans le premier test réalisé qu'un routeur Cisco 2800 Series donne une latence moyenne générale de 140 microsecondes et 3 routeurs Cisco 2800 Series donnent une latence moyenne générale de 420 microsecondes. Cette augmentation de latence est proportionnelle aux nombres d'équipements traversés. Et cette latence n'est pas forcement réduite en implémentant des stratégies de QoS. En effet, comme nous l'avons déjà cité, nous avons pu observer que dans les cas où pour un flux la latence était relativement basse, le nombre de paquets supprimés pour ce flux était relativement élevé. Un autre cas où nous avons aussi pu observer une latence relativement haute est lorsqu'un routeur effectue de la répartition de charge. C'est au concepteur réseau de bien penser son réseau pour réduire au maximum la latence.

Pour terminer cette partie de tests, voici un tableau récapitulatif des résultats intéressants des tests réalisés sur nos architectures:

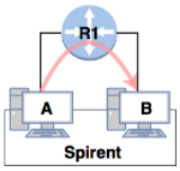
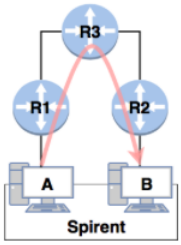
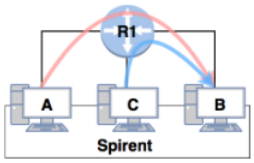
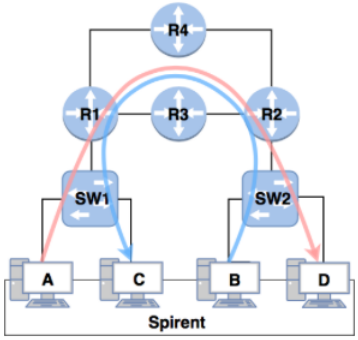
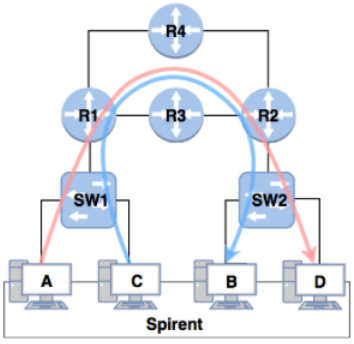
Etape 1	Topologie 1		**Réseau sans congestion avec injection 100Mb/s (FIFO):** bonne latence, même latence pour tous les flux, augmente de manière linéaire (1 routeur: latence fois 1, 3 routeurs: latence fois 3), aucune perte de paquets. **Réseau sans congestion avec injection 99Mb/s (FIFO):** latence bonne et stable, même latence pour tous les flux (1 routeur: latence fois 1, 3 routeurs: latence fois 3), aucune perte de paquets. **Réseau sans congestion avec injection 100Mb/s (PQ):** latence latence bonne, même latence pour tous les flux, augmente de manière linéaire (1 routeur: latence fois 1, 3 routeurs: latence fois 3), aucune perte de paquets, même effet qu'en FIFO. **Réseau avec congestion (ligne à 10Mb/s) avec injection 100Mb/s (PQ):** latence bonne et stable pour flux voix, Internet et management, très mauvaise latence pour flux TV, aucune perte pour flux voix et management, 100% perte pour flux Internet .
	Topologie 2		
Etape 2	Topologie 3		**Réseau sans congestion avec injection 100Mb/s (FIFO):** latence bonne et stable pour Cisco 2800 Series, latence moins élevée mais irrégulière pour Cisco 2900 Series. **Réseau avec congestion (ligne à 100Mb/s) avec injection 101Mb/s (FIFO):** latence très élevée, même latence pour tous les flux, 1% de perte, même perte pour tous les flux.
Etape 3	Topologie 4		**Réseau sans congestion (ligne à 100Mb/s) avec injection 200Mb/s (FIFO):** latence bonne et stable, même latence pour tous les flux, aucune perte de paquets, car ligne en Full-Duplex. **Réseau sans congestion (ligne à 100Mb/s) avec injection 200Mb/s (FIFO), coupure du lien et recalcul des routes:** latence bonne et stable, perte de 1,37% pour tous les flux, le recalcul des routes prends 8,262 secondes.
Etape 4	Topologie 5		**Réseau avec congestion (ligne à 100Mb/s) avec injection 200Mb/s (FIFO):** latence bonne, augmente légèrement, même latence pour tous les flux, perte de 50%, même perte pour tous les flux. **Réseau avec congestion (ligne à 100Mb/s) avec injection 200Mb/s (CBWFQ):** latence bonne, augmente légèrement, 50% de perte totale, 0% de perte pour flux voix, management et Internet, 86% de perte pour flux TV. **Réseau avec congestion (ligne à 100Mb/s) avec injection 200Mb/s (CBWFQ):** latence bonne, augmente légèrement, 50% de perte totale, 0% de perte pour flux voix, management, 75% de perte pour flux Internet, 31% de perte pour flux TV. **Réseau sans congestion (ligne à 100Mb/s) avec injection 200Mb/s et partage de charge:** 0% de perte, latence extrêmement élevée pour tous les flux.

Figure 124 – Synthèse de tous les résultats intéressants réalisés sur nos architectures

Les configurations du Spirent, des routeurs et des switchs sont disponibles, en annexe, dans un catalogue d'architectures afin de mettre en places ces 5 architectures et le catalogue des scénarios de tests est disponible à la figure 63. Ces catalogues permettent de reproduire ces tests aisément.

6 Problèmes

Les problèmes présentés ci-dessous sont mentionnés dans la documentation en annexe "Quel est l'effet de la QoS sur des petits réseaux de labo ? - Mise en place réseau de tests et stratégies de QoS - Configuration Spirent C1 et routeur Cisco 2800 Series", afin de les éviter.

6.1 Négociation Spirent-routeurs

Le premier problème que nous avons rencontré est un problème de négociation entre le Spirent et le routeur Cisco 2800 Series durant la mise en place du réseau de tests de l'étape 1. Nous avions donc un seul routeur connecté au Spirent.

La négociation entre les interfaces du Spirent et du routeur ne se faisait pas, les lignes restaient "Down". Premièrement, nous avons cru que la négociation ne s'effectuait pas correctement, car l'autonégociation au niveau de vitesse des liens pouvait causer problème. Nous avons donc fait des tests en définissant manuellement, du côté du Spirent et du côté de routeur Cisco 2800 Series, une vitesse aux liens de 100Mb/s sans succès, puis de 10Mb/s, sans succès non plus. Les interfaces restaient en état "Down".

```
R3#show ip int br
Interface                  IP-Address      OK? Method Status                Protocol
FastEthernet0/0            192.168.50.2    YES manual up                    down
FastEthernet0/1            192.168.70.2    YES manual up                    down
Serial0/1/0                unassigned      YES NVRAM  administratively down down
Serial0/1/1                unassigned      YES NVRAM  administratively down down
```

Figure 125 – Interfaces en état "Down"

En suite, nous avons remis en question notre plan d'adressage IP, en effet nous avons analysé les adresses IP définies sur le Spirent et sur le routeur, mais tout semblait correct et l'était.

Sans grande conviction et ne comprenant pas d'où pouvait provenir ce problème de négociation entre le Spirent et le routeur, nous avons remplacé les câbles Ethernet non croisés par des câbles Ethernet croisés. Cintre toutes attentes, le problème provenait des câbles, en effet il fallait mettre des câbles croisés entre le Spirent et le routeur. Les routeurs Cisco 2800 Series étant trop vieux pour gérer ce cas. Après cette modification, la négociation entre le Spirent et le routeur s'est faite sans problème.

À noter que si nous plaçons un switch entre le Spirent et le routeur, des câbles non croisés doivent être utilisés.

6.2 Caprices du Spirent

Un problème qui nous a causé beaucoup de soucis est au niveau du Spirent. Le Spirent est très capricieux aux modifications de configuration sur lui-même ou au niveau du réseau de tests. En effet, lorsque nous modifions les configurations des hôtes virtuels ou de la génération des flux de trafic, les ports passaient en état "Down". Dans ce cas, il faut fermer le programme "Spirent TestCenter" et le redémarrer.

```
R1#
*Jan 15 22:58:27.277: %GT96K_FE-5-EXCESSCOLL: Excessive Collision on int  FastEthernet0/0
*Jan 15 22:58:30.969: %GT96K_FE-5-EXCESSCOLL: Excessive Collision on int  FastEthernet0/0
*Jan 15 22:58:37.245: %GT96K_FE-5-EXCESSCOLL: Excessive Collision on int  FastEthernet0/0
*Jan 15 22:58:40.937: %GT96K_FE-5-EXCESSCOLL: Excessive Collision on int  FastEthernet0/0
*Jan 15 22:58:41.601: %GT96K_FE-5-EXCESSCOLL: Excessive Collision on int  FastEthernet0/0
*Jan 15 22:58:47.213: %GT96K_FE-5-EXCESSCOLL: Excessive Collision on int  FastEthernet0/0
*Jan 15 22:58:50.829: %GT96K_FE-5-EXCESSCOLL: Excessive Collision on int  FastEthernet0/0
*Jan 15 22:58:57.253: %GT96K_FE-5-EXCESSCOLL: Excessive Collision on int  FastEthernet0/0
*Jan 15 22:59:00.797: %GT96K_FE-5-EXCESSCOLL: Excessive Collision on int  FastEthernet0/0
*Jan 15 22:59:07.297: %GT96K_FE-5-EXCESSCOLL: Excessive Collision on int  FastEthernet0/0
```

Figure 126 – Collisions excessives sur l'interface

Lors de modifications du réseau de tests, le Spirent s'est, des fois, emballé et de nombreuses collisions étaient visibles sur les routeurs. Dans ce cas aussi, il faut fermer le programme "Spirent TestCenter" et le redémarrer.

7 Network Topology & Quality TestCenter

Cette partie introduit une partie économique au projet en construisant une société experte dans l'analyse et la conception de réseau de télécommunication. Cette société est construite autour des éléments abordés durant ce projet.

Network Topology & Quality TestCenter est une société de télécommunication, basée à la Haute École d'Ingénierie et d'Architecture de Fribourg, qui fournit des solutions de tests, d'analyse et de conception de réseaux de tests pour les sociétés de télécommunication et les concepteurs de réseaux.

Network Topology & Quality
TestCenter

Nous travaillons en étroite collaboration avec la Haute École d'Ingénierie et d'Architecture de Fribourg, Spirent et Cisco.

7.1 Produits & solutions

Network Topology & Quality TestCenter propose plusieurs solutions pour les sociétés de télécommunication et les concepteurs de réseaux.

7.1.1 Analyse de réseaux

Grâce au puissant générateur de trafic Spirent, Network Topology & Quality TestCenter propose des tests complets de réseaux de tests avec des analyses très poussées.

Cette solution d'analyse de réseaux de tests comprend 4 étapes:

1. Attentes du client
2. Définition des scénarios de tests
3. Tests et analyses du réseau de tests
4. Présentation des résultats

Le client a conçu et réalisé lui-même son réseau. Nous partons dans l'optique que le réseau de tests est déjà configuré et nous ne le touchons plus, nous ne faisons qu'injecter du trafic dans ce dernier afin de pouvoir l'analyser.

Attentes du client

Nous rencontrons le client afin que ce dernier nous présente son réseau de tests et nous décrivent ces attentes. Nous devons connaître le but de la mise en place d'un tel réseau afin de pouvoir définir nos critères d'évaluation et nos scénarios de tests.

Cette étape nous permet tout d'abord de nous rendre compte, sans encore avoir effectué de tests, si le réseau de tests du client est, à la première impression, adapté à ces attentes. À cette étape, nous pouvons déjà nous faire une petite idée. Ensuite, grâce aux attentes du client, nous pourrons comprendre quel type de trafic ce réseau de tests devra transporter et à quel volume, ce qui nous permettra de définir nous scénarios de tests.

Définition des scénarios de tests

Une fois les attentes du client décrites, nous mettons en place nos scénarios de tests comme nous l'avons fait dans ce projet. Ces scénarios de tests doivent être respectés précisément. Ils permettront de valider ou non le réseau de tests du client en fonction de ces attentes.

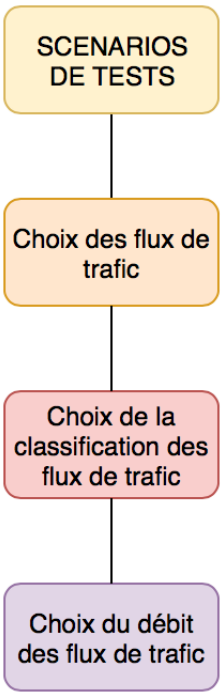


Figure 127 – Définission des scénarios de tests

La définissions des scénarios de tests est basé sur la génération de trafic. Nous devons définir quel type de flux de trafic nous allons injecter dans le réseau de tests grâce au générateur de trafic Spirent. Nous proposons de base 5 types de flux de trafic avec des classifications à la source, qui introduit notre première stratégie de QoS, et un débit pour chacun des flux en nous basant sur les chiffres de Swisscom.

Trafic	Transport	DSCP	QoS Byte	DSCP (Hex)	Volume [Mbit/s]
Téléphonie	UDP	EF	B8	2E	1
Télévision multicast	UDP	AF41	88	22	1
Télévision unicast	TCP	AF41	88	22	56
Internet	TCP	AF13	38	0E	41
Mangement	TCP	CS6	C0	30	1

Figure 128 – Génération des flux de trafic pour les scénarios de tests

Bien entendu, ces chiffres nous permettent d'avoir une base, mais la définition des flux de trafic, pour la mise en place de nos scénarios de tests, dépend aussi des attentes du client pour son réseau de tests. Nous nous adaptons afin de respecter au mieux les exigences du réseau de tests du client.

Les configurations pour ces scénarios de tests se passent uniquement au niveau du Spirent. Nous avons aussi à disposition un catalogue de scénarios de tests disponibles dont nous pourrons nous inspirer.

Tests et analyses du réseau de tests

Nous prenons la possession de son réseau de tests du client et nous appliquons nos scénarios de tests et nos analyses poussées sur son réseau de tests.

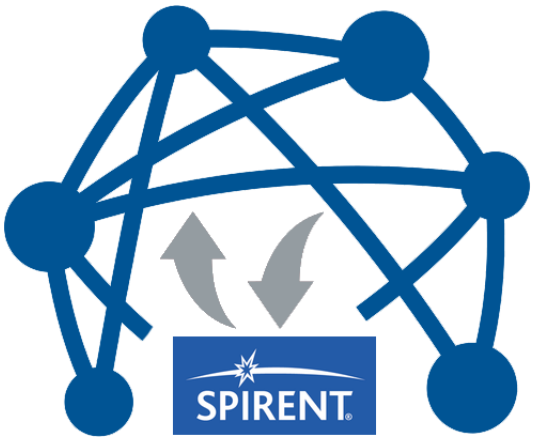


Figure 129 – Injection du trafic dans le réseau de tests

Network Topology & Quality TestCenter utilise le puissant générateur de trafic Spirent mis à disposition par la HEIA-FR. Le Spirent offre une large palette d'outils afin d'analyser le réseau en fonction du flux de trafic injecter dans ce dernier.

Nous analysons les performances du réseau de tests et rédigeons des rapports.

Présentation des résultats

Une fois les tests terminés, nous rendons notre rapport au client. Bien entendu, les tests peuvent être validés ou non. Si le réseau de tests est performant et remplit les critères fixés, le réseau de tests est validé et le client peut potentiellement mettre son réseau en production. Si le réseau de tests n'est pas validé en fonction de nos critères, nous proposons au client de nous occuper de sa modification, dans ce cas nous passons de la solution "Analyse de réseaux" à la solution "conception de réseaux". Le client peut aussi reprendre possession de son réseau de tests et s'occuper lui-même des modifications en fonction de notre rapport ou en faire ce qu'il en veut.

7.1.2 Conception de réseau

Network Topology & Quality TestCenter conçoit de réseaux de tests en fonction des besoins des clients. Cette solution de conception de réseaux de tests comprend la solution d'analyse de réseaux de tests.

Cette solution de conception de réseaux de tests comprend 6 étapes:

1. Attentes du client
2. Conception du réseau de tests
3. Réalisation du réseau de tests
4. Définition des scénarios de tests
5. Tests et analyses du réseau de tests
6. Présentation des résultats

Le client veut que nous lui livrions un réseau de tests performants répondant à ces exigences. Nous devons donc mettre en place un réseau de tests et le tester afin de valider ces performances.

Attentes du client

Nous rencontrons le client afin que ce dernier nous décrive ces attentes. Nous devons connaître le but de la mise en place d'un tel réseau afin de pouvoir définir et concevoir ce réseau de tests, nos critères d'évaluation et nos scénarios de tests.

Cette étape nous permet tout d'abord de nous rendre compte, des dimensions du réseau de tests à mettre en place, grâce aux attentes du client, nous pourrons aussi comprendre quel type de trafic ce réseau de tests devra transporter et à quel volume, afin de concevoir au mieux le réseau, ce qui nous permettra aussi de définir nous scénarios de tests.

Conception du réseau de tests

La conception du réseau est l'étape la plus importante de cette solution que nous proposons. Le réseau doit être conçu afin d'être évolutif en fonction de la demande. Cette étape peut être réalisée en collaboration avec le client, nous proposons différentes solutions avec des catalogues de topologies, de stratégie de QoS ou autres. Comme réalisé dans ce projet, nous nous basons sur un modèle d'architecture DiffServ.

Premièrement, nous avons à disposition un catalogue de topologies réseau:

- Ligne
- Étoile simple
- Étoile double
- Anneau
- Double anneau
- Arbre
- Maillé
- Maillé en triangle
- Entièrement maillé

Ces topologies de base permettent de nous faire une idée sur la topologie correspondant le mieux aux attentes du client. Bien entendu, ces topologies sont flexibles et nous adapterons au mieux la topologie du réseau de tests en fonction des critères. Ces critères sont les suivants:

- Coûts
- Capacité
- Coûts par capacité
- Redondance
- Nombre moyen de bonds
- Bond maximum
- Nombre de noeuds
- Nombre de liens
- Type de flux

Nous analyserons ces critères en fonction de ces attentes du réseau de tests. Pour le choix d'une topologie réseau en fonction de ces critères, vous pouvez vous référer au point 2.5.3. Le choix de la topologie réseau doit aussi se faire en fonction du type de réseau et son flux "datacebtric" ou "any-to-any". Dans les deux cas, une topologie en double étoile est optimale, car elle remplit au mieux les critères.

Deuxièmement, une fois la topologie réseau définie, nous devons concevoir les configurations du réseau de tests, surtout au niveau des routeurs ou switchs.

Nous définirons notre réseau de tests au niveau des plans d'adressage et en créant des schémas. Au niveau du protocole de routage, nous proposons de base OSPF (Open Shortest Path First), mais là aussi un catalogue de protocole de routage est disponible:

- OSPF (Open Shortest Path First)
- RIP (Routing Information Protocol)
- IGRP (Interior Gateway Routing Protocol)
- Integrated IS-IS (Integrated Intermediate System to Intermediate System)
- EIGRP (Enhanced Interior Gateway Routing Protocol)

Nous analyserons ce catalogue de protocole de routage afin de définir quelle est la meilleure solution.

Nous pouvons maintenant, définir notre première stratégie de QoS qui est l'ordonnancement. Pour les types de files d'attente, nous avons aussi à disposition un catalogue, en fonction des files d'attente vues durant ce projet:

- FIFO (First In First Out)
- CBWFQ (Class-Based Weighted Fair-Queuing)
- LLQ (Low-Latency Queuing)
- PQ (Priority Queuing)

Bien entendu, nous pourrons réaliser des tests avec tous ces différents types de files d'attente. Pour un réseau maîtrisé de bout en bout au flux de trafic constant, pour des flux sensibles, nous conseillons la file d'attente LLQ et pour des flux moins sensibles, nous conseillons la file d'attente CBWFQ.

Réalisation du réseau de tests

Cette étape se base sur l'étape précédente et nous mettons en place notre infrastructure dans nos laboratoires.

Définition des scénarios de tests

Cette étape est semblable à l'étape "Définition des scénarios de tests" de la solution "Analyse de réseaux" présentée précédemment, veuillez vous référer au point 7.1.1. La seule différence est que la stratégie de QoS au niveau de l'ordonnancement peut varier pour les scénarios de tests afin de trouver la meilleure solution.

Tests et analyses du réseau de tests

Cette étape est semblable à l'étape "Tests et analyses du réseau de tests" de la solution "Analyse de réseaux" présentée précédemment, veuillez-vous référer au point 7.1.1. La seule différence est que nous ne prenons pas possession du réseau du client, mais nous avons monté notre propre réseau de tests dans nos laboratoires. Contrairement à la solution "Analyse de réseaux", nous livrons au client un réseau performant et pouvant être mis en production, c'est pourquoi les étapes précédentes peuvent être répétées jusqu'à que les attentes soient remplies et que le client et nous-mêmes soyons satisfaits.

Présentation des résultats

Une fois, les tests terminés et le réseau étant validés pour une mise en production, nous présentons les résultats au client et nous lui remettons le réseau de tests. Bien entendu, si les résultats ne sont pas satisfaisants pour le client, nous pouvons recommencer les étapes précédentes jusqu'à satisfaction complète du client.

7.2 Premier mandat

Network Topology & Quality TestCenter été mandaté par Monsieur Pascal Roulin pour tester son infrastructure de projet de semestre 6 "Remote Network Monitoring". Son réseau étant déjà existant, nous lui proposons la solution "Analyse de réseaux".

Nous nous sommes donc déplacés chez lui afin de comprendre ces attentes au niveau des tests. Il veut analyser son réseau en injectant du trafic d'un côté du réseau et ressortir ce trafic de l'autre côté du réseau afin d'en analyser premièrement si le trafic passe bien à travers son réseau et deuxièmement la perte de paquets et la latence générale.

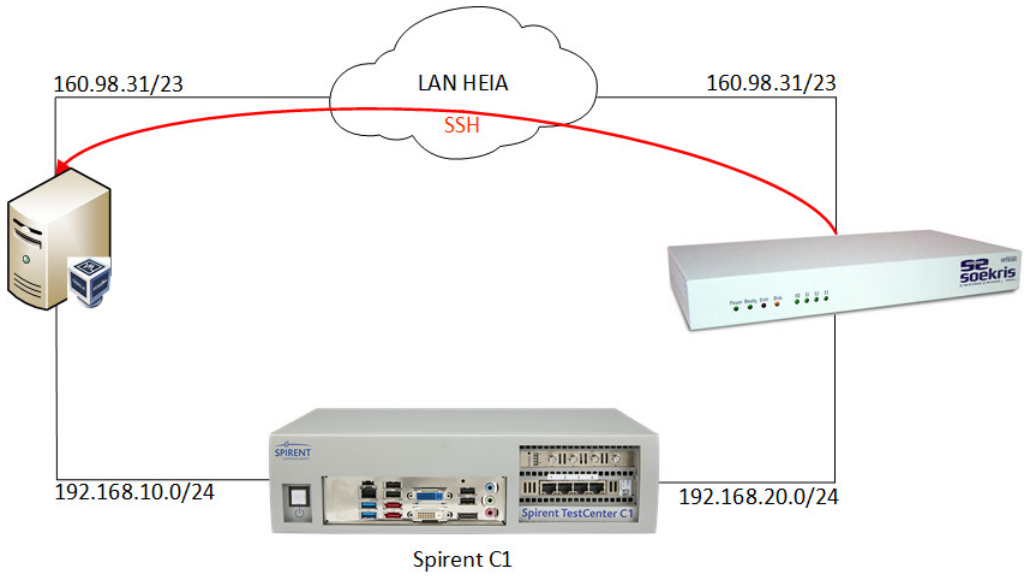


Figure 130 – Schéma du réseau de tests "Remote Network Monitoring" fourni par Monsieur Pascal Roulin

En nous référant à notre catalogue d'architecture, nous avons défini la configuration du Spirent correspondant au mieux au réseau de tests du client. Nous avons appliqué cette solution en associant son réseau de tests et notre Spirent afin de pouvoir générer et réceptionner du trafic sur nos équipements au travers de son réseau de tests.

Nous définissons ensuite, nos scénarios de tests pour analyse de la perte de paquets et de la latence, en nous référant à notre catalogue de scénarios de tests. La génération de trafic se base sur les 5 flux déjà connus et un paramétrage de base de ces flux comme nous avons déjà pu le voir selon la figure 128. Nous allons injecter ce trafic par étape, premièrement à 10Mb/s, puis à 50Mb/s et pour finir à 100Mb/s afin d'en analyser au mieux les réactions sur le réseau.

Le trafic a bel et bien traversé son réseau de la source vers la destination, ce test est un franc succès. Les résultats étant relatifs au projet de Monsieur Pascal Roulin, nous ne présentons pas les résultats. Nous avons fourni au client les différentes informations relatives aux tests effectués. Pour toutes informations supplémentaires, veuillez-vous référer chez Monsieur Pascal Roulin.

Cette étape fut très intéressante, car nous avons pu voir que les catalogues d'architectures et de scénarios de tests que nous avons définis ont été utiles pour un cas concret. En effet, nous avons pu mettre en commun le travail que nous avons réalisé avec un réseau de tests quelconques. De la combinaison de ces deux éléments, des résultats en sont sortis, et ont été de grande utilité pour le client. Je tiens d'ailleurs à remercier Monsieur Pascal Roulin pour sa collaboration.

8 Conclusion

Ce projet nous a permis de comprendre le fonctionnement et l'importance de différentes stratégies de QoS. En effet, nous avons implémenté ces stratégies de QoS sur différents réseaux de tests. Nous avons pu voir que ces notions de qualité de service sont indispensables pour garantir la qualité, la performance d'un réseau et le bon acheminement des différents flux vers l'utilisateur afin que ce dernier soit satisfait.

Comme nous avons pu le voir tout au long de la partie de tests et de validations du projet, en testant un réseau sur sa latence par type de flux de trafic et la perte de paquets que ce réseau engendre par flux, nous pouvons déjà nous faire une bonne idée de l'efficacité et de la qualité de ce réseau. En effet, pour ce projet, nous nous sommes plus concentrés sur la latence et la perte de paquets que sur l'utilisation de la bande passante et la gigue.

Ce qui a rendu ce projet encore plus intéressant est le fait que lorsque nous effectuions certains tests, nous nous attendions souvent à des résultats, qui pour nous paraissaient logiques. Mais souvent, ces résultats n'étaient pas ce que nous attendions et nous devions réfléchir, chercher afin de comprendre les effets de certaines configurations ou effectuer d'autres tests plus en profondeur afin de sentir le comportement du réseau de tests sous certaines contraintes. C'est pourquoi, nous effectuions un test, pour ensuite en redéfinir un autre complètement inattendu, tout au long du projet, nous travaillions à la fois sur la partie conception et la partie de tests et de validations. Le fait de chercher minutieusement pourquoi nous obtenions certains résultats nous a permis de mieux comprendre certaines notions, d'en sortir nos propres théories et surtout d'en apprendre davantage, ce qui est très important.

En général, tous les résultats des tests ont été analysés, expliqués et compris. Nous avons défini, pour chacun des tests, les résultats que nous attendions et les résultats obtenus. Au niveau des résultats de latence et de pertes de paquets obtenus dans les tests, nous faisons chaque fois la comparaison avec ce qui est acceptable afin de définir si les stratégies de QoS remplissent leurs objectifs. Grâce à cela, nous avons pu faire une synthèse sur les différentes stratégies de QoS définies dans ce projet afin de citer leurs points faibles et leurs points forts.

Dans l'ensemble, le planning initial a bien été respecté tout au long du projet. En revanche, nous avions fixé une date butoir pour chaque partie du projet, les parties de conception et de réalisation n'ont pas pu être rendues aux dates fixées, car nous avons effectué des itérations entre la conception, la réalisation et la partie de tests et validations, afin de définir chacun des nouveaux tests. En effet, lorsque nous terminions un test, en fonction des résultats, un nouveau test ou une nouvelle topologie réseau étaient pensés et mis en place, nous devions alors les conceptualiser et les réaliser. Voilà qui explique ce système d'itération que nous avons mis en place entre les parties et pourquoi certaines parties n'ont pas été rendues dans les temps.

Pour toutes informations sur la mise en place d'un réseau de tests et des stratégies de QoS vues dans ce projet au niveau de la configuration du Spirent C1 et des routeurs Cisco 2800 Series, veuillez-vous référer à la documentation en annexe "Quel est l'effet de la QoS sur des petits réseaux de labo ? - Mise en place réseau de tests et stratégies de QoS - Configuration Spirent C1 et routeur Cisco 2800 Series" réalisée durant le projet de semestre 6 par Monsieur Loïc Dufresne "Quel est l'effet de la QoS sur des petits réseaux de labo ?".

8.1 Atteinte des objectifs

L'objectif principal était de proposer un catalogue d'architectures d'une ou de plusieurs topologies réseau afin de pouvoir tester au mieux les différentes stratégies de QoS en fonction des flux de trafic. Au total, 5 architectures réseau ont été conçues, réalisées et testées, 2 stratégies de QoS dont la classification de paquets et l'ordonnancement avec 4 types de files d'attente ont été testés sur ces 5 réseaux, et 5 types de flux de trafic ont été configurés et testés sur les architectures. Tous ces éléments ont été regroupés dans un catalogue afin qu'une tierce personne puisse aisément reprendre ce projet et mettre en place ces infrastructures de tests. Ce catalogue est disponible en annexe.

Un second objectif était de définir un catalogue de scénarios de tests regroupant les stratégies de QoS et les flux de trafic. Ce catalogue est défini dans la partie conception du projet, et listé à la figure 63 de la partie tests et validations.

Le projet devait aussi fournir, une série de mesures documentées, elle est définie par le catalogue de scénarios de test. Cette série de tests est disponible dans la partie de tests et validations du projet, au chapitre 5.

Les objectifs principaux de ce projet ont été atteints. Malheureusement, les objectifs secondaires du projet qui était de réaliser l'objectif principal au niveau "Layer 2", étudier l'émulateur "JAR" afin de définir ces limites et s'il est capable de réaliser les mêmes procédures que dans l'objectif principal et interpréter les résultats obtenus afin déterminer si les réseaux physiques peuvent être émulés de manière très réaliste, n'ont pas pu être réalisés. Ces objectifs secondaires ont été mis de côté afin de pouvoir mettre en place et tester plusieurs architectures et pousser les tests plus profondément afin de proposer un catalogue d'architectures et de scénarios de tests plus garnis.

Un point à réaliser était de vérifier et tester l'implémentation des files d'attente PQ comme la réaliser Monsieur Simon Lièvre dans son projet de semestre 5 "QoSLab" [1]. Ces tests semblaient étonnants au niveau de la latence. Nous avons mis en place son infrastructure, implémenté la file d'attente en question. Au niveau de la latence, nous avons obtenu les mêmes résultats, nous nous sommes penchées sur la perte de paquets et cette perte de paquets était logique. Nous avons pu ensuite mettre en relation cette perte de paquets et la latence obtenue, une théorie a été mise au point et les tests réalisés par Monsieur Simon Lièvre ont été validés, car corrects.

Durant ce projet, nous avons aussi réalisé une documentation supplémentaire sur la mise en place d'un réseau de tests et des stratégies de QoS vues dans ce projet au niveau de la configuration du Spirent C1 et des routeurs Cisco 2800 Series. Cette documentation est disponible en annexe "Quel est l'effet de la QoS sur des petits réseaux de labo ? - Mise en place réseau de tests et stratégies de QoS - Configuration Spirent C1 et routeur Cisco 2800 Series" réalisée durant le projet de semestre 6 par Monsieur Loïc Dufresne "Quel est l'effet de la QoS sur des petits réseaux de labo ?".

8.2 Perspectives futures

Ce projet peut être approfondi vers de nombreuses directions. Beaucoup de perspectives futures peuvent être envisagées.

Tout d'abord, nos scénarios de tests ont été réalisés en fonction d'une seule et même classification de paquets, en nous basant sur les chiffres du réseau Swisscom. Un réseau peut transporter d'autres informations, son but peut être varié et différent de ce que nous avons réalisé. Nous pourrions nous pencher sur d'autres flux de trafic et d'autre classification afin de tester nos stratégies de QoS.

Nous avons testé 4 files d'attente différentes, mais comme nous avons pu le voir dans l'analyse, il en existe un bon nombre. Nous pourrions effectuer des tests avec d'autres files d'attente. Comme nous avons pu le voir, sur les routeurs, lors de l'implémentation des files d'attente, nous pouvons entièrement les configurer à notre guise, nous pourrions analyser plus en détail certaines files d'attente afin d'essayer de les optimiser pour certains flux de trafic. Encore une fois, tout dépend des informations transitant sur le réseau et des attentes des concepteurs.

Des réseaux plus complexes peuvent aussi être mis en place afin d'y réaliser des tests. Nous nous sommes penchés sur un réseau en anneau, mais nous pourrions tester nos stratégies de QoS sur un réseau en double étoile, par exemple. D'après notre analyse, nous avons pu voir qu'un réseau en double étoile était un des réseaux les plus optimaux et ayant un des meilleurs rapports capacité/prix. Il serait intéressant de mettre en place ce type de réseau afin d'y faire des tests.

Nous nous sommes aussi penchés sur la répartition des charges. Comme dans le cas d'un réseau en anneau, cette technique est intéressante, car elle permet d'utiliser la capacité entière du réseau et donc de réduire la perte de paquets. Nous avons utilisé la technique de partage de charge Cisco Express Forwarding, mais nous avons vu qu'il existe d'autres techniques. Sur notre réseau de tests, la répartition des charges augmentait énormément la latence des différents flux. Il faudrait analyser et tester ces différentes techniques de partage de charge afin de se faire une idée de laquelle est la plus optimale. Ce serait intéressant de trouver une technique qui activerait se partage de charges que lorsque le réseau devient congestionné et fixer un compromis pour les flux, entre la latence et la perte de paquets, car comme nous avons pu le voir plus la perte de paquets est faible, plus la latence est grande.

Un des objectifs secondaires intéressants est le fait de virtualiser ce que nous avons réalisé dans ce projet et d'y réaliser les mêmes tests afin de voir si les réseaux physiques et virtuels se comportent de la même manière.

Tout au long du projet, nous avons travaillé avec le puissant générateur de trafic Spirent. Nous avons pu voir que cet appareil propose une palette importante d'outils d'analyse, de paramètres et autres. Pour la réalisation de ce projet, nous avons travaillé avec les outils de base que propose le Spirent. Mais, par exemple, nous avons pu lire avec Monsieur Simon Lièvre, que le Spirent à la capacité de générer lui-même des PDF avec les résultats sous forme de tableaux ou graphiques. Cet outil est très intéressant, car il permet de gagner énormément de temps, en nous évitant de reporter toutes les mesures dans Excel afin de faire nos propres tableaux ou graphiques. Nous n'avons malheureusement pas réussi à utiliser cet outil. Il serait intéressant de faire un travail d'analyse du Spirent afin de découvrir ce que cet appareil est capable de faire.

8.3 Conclusion personnelle

Revenons au mois de septembre passé, lors des choix de projet de semestre 5. À ce moment-là, je ne me serais jamais imaginé choisir un projet de réseau, car je me sentais plus à l'aise avec l'environnement du développement. Après avoir passé plus de 5 mois à développer une application Web, il est vrai que j'avais envie de découvrir de nouvelles choses et dans un autre domaine. Lors des choix de projet de semestre 6, j'hésitais à repartir sur un projet de développement, mais l'envie de partir sur un projet plus consacré au réseau me trottait dans la tête, et c'est avec beaucoup d'appréhension que j'ai décidé, au dernier moment, d'opter pour ce projet.

Au début du projet, je n'étais vraiment pas à l'aise avec certaines notions comme les stratégies de QoS, qui représentent un des éléments principaux de ce projet. J'ai donc commencé à faire des recherches autant sur Internet que dans des livres. Au début, ce n'était vraiment pas facile, d'ailleurs je remercie encore Monsieur Simon Lièvre pour sa disponibilité à répondre à mes questions. Ensuite, plus les jours passaient, plus j'apprenais et plus ma motivation grandissait à l'idée de réaliser ce projet, d'ailleurs ce travail m'a permis d'apprendre énormément dans ce domaine, car il était vraiment intéressant et entraînant.

Un autre point important durant le projet était le fait que je voulais trouver seul les solutions à mes problèmes. Je m'explique, il m'est arrivé de me retrouver bloqué pendant plusieurs jours à cause de certains problèmes, comme la notion de génération des flux de trafic sur le Spirent qui m'a pris bien 2 semaines avant d'être acquise et opérationnelle, ou certaines réactions du réseau que je n'arrivais pas à expliquer. Dans tous les cas, j'ai cherché à régler seul mes problèmes en cherchant des explications et en essayant d'autres configurations comme sur le Spirent. Je trouvais important de me sortir de certaines situations tout seul, sans déranger tout le temps Monsieur Simon Lièvre ou mes superviseurs.

En tout cas, je ne regrette pas d'avoir choisi et réalisé ce projet, je suis satisfait du travail réalisé, comme je l'ai dit cela m'a permis d'apprendre énormément et d'approfondir mes connaissances, ce qui est pour moi le point essentiel. Comme tout projet, il y a une fin, il est vrai que j'aurais bien aimé pousser plus certains tests afin de mieux comprendre certaines notions, mais j'ai dû me mettre une limite, malheureusement. J'espère que ce projet connaîtra une suite et que ce rapport sera utile pour le futur. Je tiens encore à remercier mes superviseurs, Monsieur Jacques Robadey et Monsieur François Buntschu, pour m'avoir fait confiance et pour m'avoir motivé pour la réalisation de ce projet, merci.

Déclaration d'honneur

Je, soussigné, Loïc Dufresne, déclare sur l'honneur que le travail rendu est le fruit d'un travail personnel. Je certifie ne pas avoir eu recours au plagiat ou à toutes autres formes de fraudes. Toutes les sources d'information utilisées et les citations d'auteur ont été clairement mentionnées.

Fribourg, le 18.05.2018.

Loïc Dufresne

9 Figures

9.1 Liste des figures

10 Lexique

AF	Assured Forwarding
BE	Expedited Forwarding
CBWFQ	Class-Based Weighted Fair-Queuing
CS	Class Selector
DC	Differentiated Services
DiffServ	Differentiated Services
DRR	Deficit Round Robin
DSCP	DiffServ Code Points
ECN	Explicit Congestion Notification
EF	Expedited Forwarding
EIGRP	Enhanced Interior Gateway Routing Protocol
EITF	Internet Engineering Task Force
FIFO	First In First Out
FTP	File Transfer Protocol
GNS3	Graphical Network Simulator
HD	High Definition
HEIA-FR	Haute école d'ingénierie et d'architecture de Fribourg
HES-SO	Haute École spécialisée de Suisse occidentale
HTTP	Hypertext Transfer Protocol
IGRP	Interior Gateway Routing Protocol
IHL	Internet Header Lengh
Integrated IS-IS	Integrated Intermediate System to Intermediate System
IntServ	Integrated Services
IP	Internet Protocol
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
LLQ	Low-Latency Queuing
MPLS	MultiProtocol Label Switching
OSI	Open Systems Interconnection
OSPF	Open Shortest Path First
PHB	Per-Hop Behaviors
PQ	Priority Queuing
QoS	Quality of service
RIP	Routing Information Protocol
RSVP	Resource reSerVation Protocol
SD	Standard Definition
TCP	Transmission Control Protocol
ToS	Type of Service
TTL	Time to Live
UDP	User Datagram Protocol
VIRL	Virtual Internet Routing Lab

VLAN	Virtual Local Area Network
VoD	Video on Demand
VPN	Virtual Private Network
WFQ	Weighted Fair Queuing
WRR	Weighted Round Robin

11 Bibliographie

[1] Monsieur Simon Lièvre. Projet de semestre 5 - qos lab, 2018. [Sur le support Multidoc; Livre disponible le 01-mars-2018].

[2] Monsieur Gabriel Python. Projet de semestre 6 - best network topology, 2018. [Sur le support Multidoc; Livre disponible le 01-mars-2018].

[3] Jean-Louis Mélin. Solution réseaux - qualité de service sur ip. page 65, 2001. [A la bibliothèque; Livre disponible le 01-mars-2018].

[4] Cisco. Dscp and precedence values. https://www.cisco.com/c/en/us/td/docs/switches/datacenter/nexus1000/sw/4_0/qos/configuration/guide/nexus1000v_qos/qos_6dscp_val.pdf, -. [En ligne; Page disponible le 13-mars-2018].

[5] Wikipedia. Qualité de service. https://fr.wikipedia.org/wiki/Qualité_de_service, 2018. [En ligne; Page disponible le 01-mars-2018].

[6] Cisco. The qos baseline. https://www.cisco.com/en/US/technologies/tk543/tk759/technologies_white_paper0900aecd80295a9b.pdf, 2005. [En ligne; Page disponible le 20-mars-2018].

[7] Dr. Jacques Robadey. Conception, déploiement et exploitation de réseaux: 4. conception du core. 2018. [Sur le support de cours; Slides disponible le 20-mars-2018].

[8] Switch. Le réseau informatique génial. https://www.switch.ch/fr/services/network/, 2018. [En ligne; Page disponible le 26-mars-2018].

[9] NetConsult. Schweizer internet 1995. https://www.netconsult.ch/de/ueber-uns/geschichten/uebersicht/, 2016. [En ligne; Page disponible le 26-mars-2018].

[10] Monsieur Gabriel Python. Projet de bachelor - cloud topology lab for multiple services, 2018. [Sur le support Multidoc; Livre disponible le 01-mars-2018].

[11] Monsieur Gabriel Python. Projet de bachelor - configuration spirent, 2018. [Sur le support Multidoc; Documentation disponible le 01-mars-2018].

[12] Jean-Louis Mélin. Solution réseaux - qualité de service sur ip. page 103, 2001. [A la bibliothèque; Livre disponible le 01-mars-2018].

[13] Accellent Group. La qualité de service de la voix sur ip. http://wallu.pagesperso-orange.fr/VoIP.pdf, -. [En ligne; Page disponible le 20-mars-2018].

[14] Christina Hattingh Cisco Tim Szigeti. Quality of service design overview. http://www.ciscopress.com/articles/article.asp?p=357102&seqNum=3, -. [En ligne; Page disponible le 20-mars-2018].

[15] Spirent. Documentation v4 70. 2016. [Sur le support d'installation Spirent; Documentations disponible le 20-mars-2018].

[16] Cisco. Troubleshooting input queue drops and output queue drops. https://www.cisco.com/c/en/us/support/docs/routers/10000-series-routers/6343-queue-drops.html, 2016. [En ligne; Page disponible le 22-avril-2018].

[17] Cisco. Documents sur le matériel cisco 2800 : introduction et avertissements. https://www.cisco.com/c/fr_ca/td/docs/routers/access/1800/1841/hardware/installation/guide/hw.pdf, 2010. [En ligne; Page disponible le 25-avril-2018].

[18] Cisco. Routeurs à services intégrés cisco 2900. https://www.cisco.com/c/dam/global/fr_fr/assets/pdfs/isr/2900_data_sheet_c78_553896.pdf, 2009. [En ligne; Page disponible le 25-avril-2018].

[19] Cisco. Quality of service (qos) configuration guide, cisco ios xe everest 16.6.x (catalyst 9500 switches). https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst9500/software/release/16-6/configuration_guide/qos/b_166_qos_9500_cg/b_166_qos_9500_cg_chapter_00.html, -. [En ligne; Page disponible le 1-mai-2018].

[20] Cisco. switchport trunk allowed vlan. https://www.cisco.com/c/m/en_us/techdoc/dc/reference/cli/nxos/commands/l2/switchport-trunk-allowed-vlan.html, -. [En ligne; Page disponible le 1-mai-2018].

[21] Cisco. Configuring 802.1q trunking between a catalyst 3550/3560/3750 and catalyst switches that run cisco ios software. https://www.cisco.com/c/en/us/support/docs/switches/catalyst-6000-series-switches/10599-88.html, 2006. [En ligne; Page disponible le 1-mai-2018].

[22] Wikipedia. Routage. https://fr.wikipedia.org/wiki/Routage, 2018. [En ligne; Page disponible le 6-mai-2018].

[23] Valentin Weber. Ospf : Configuration basique. https://www.networklab.fr/ospf-configuration-basique/, 2013. [En ligne; Page disponible le 8-mai-2018].

[24] TCC2. How to configure ospf cost. https://supportforums.cisco.com/t5/network-infrastructure-documents/how-to-configure-ospf-cost/ta-p/3133153, 2009. [En ligne; Page disponible le 8-mai-2018].

[25] Cisco. Ospf design guide. https://www.cisco.com/c/en/us/support/docs/ip/open-shortest-path-first-ospf/7039-1.html, 2005. [En ligne; Page disponible le 8-mai-2018].

[26] Cisco. Dépannage de l'équilibrage de charge sur des liens parallèles utilisant cisco express forwarding. https://www.cisco.com/c/fr_ca/support/docs/ip/express-forwarding-cef/18285-loadbal-cef.html, 2005. [En ligne; Page disponible le 12-mai-2018].

[27] Cisco. Configuring a load balancing scheme. https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/ipswitch_cef/configuration/15-mt/isw-cef-15-mt-book/isw-cef-load-balancing.html, 2018. [En ligne; Page disponible le 13-mai-2018].

[28] Cisco. Configuring a load balancing scheme. https://www.cisco.com/c/en/us/support/docs/ip/border-gateway-protocol-bgp/5212-46.html, 2015. [En ligne; Page disponible le 13-mai-2018].

12 Annexes

Les annexes cités ci-dessous sont imprimés à part du rapport et fournis avec la version papier du rapport. Pour la version PDF du rapport, les annexes cités ci-dessous sont disponibles sur la Forge et sur le CD/DVD:

1. Le cahier des charges
2. Mise en place réseau de tests et stratégies de QoS - Configuration Spirent C1 et routeur Cisco 2800 Series

Pour les personnes ayant participé au projet, tous les documents, PV, plannings, informations relatives au projet, et les catalogues d'architectures et de scénarios de tests, sont disponibles sur la FORGE:
`https://forge.hefr.ch/projects/projetsesmestre6_dufresne_qos_2018`

12.1 Versions

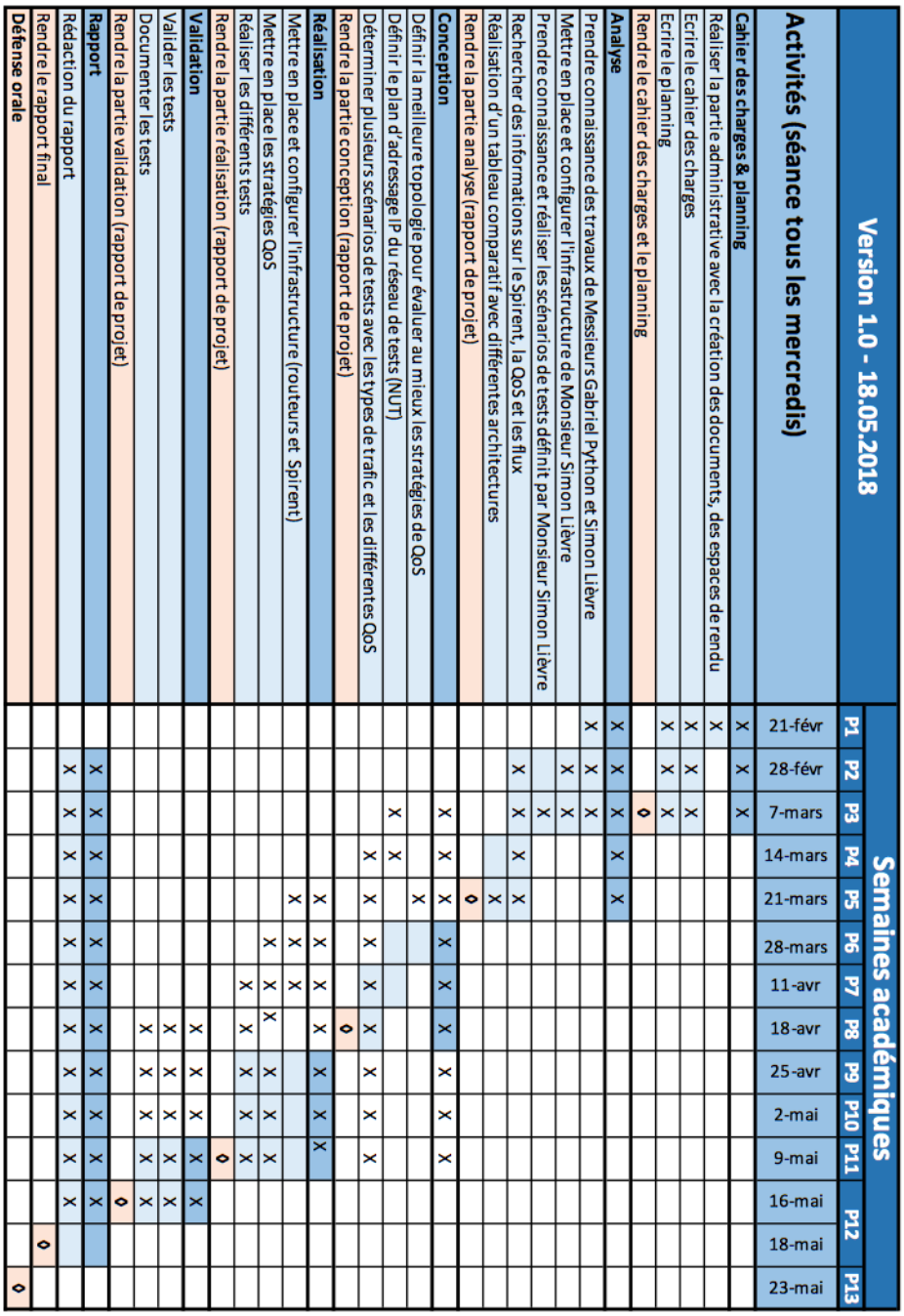
Ces versions sont celles avec lesquelles nous avons réalisé ce projet:

- **Spirent TestCenter:** version 4.70.9706.0000
- **Tera Term:** version 4.95
- **Word:** version 16.10
- **Excel:** version 16.10

12.2 Contenu du CD/DVD

```
Projet semestre 6 - QoS & réseaux de labo
   .
   |-- cahier_des_charges
   |   `-- cahier_des_charges_070318.pdf
   |-- catalogue
   |   |-- reseau_1
   |   |   |--configurations_routeurs
   |   |   |  `--*
   |   |   |--configurations_spirent
   |   |   |  `--*
   |   |   |--README_reseau_1.txt
   |   |   `--schema_reseau
   |   |      `--*
   |   |-- reseau_2
   |   |   `--*
   |   |-- reseau_3
   |   |   `--*
   |   |-- reseau_4
   |   |   `--*
   |   `-- reseau_5
   |       `--*
   |-- documentations
   |   `-- configurations_spirent_routeurs_020518
   |-- mesures
   |   `-- graphiques.pdf
   |-- planning
   |   |-- heures_180518.pdf
   |   `-- planning_final_180518.pdf
   |-- pv
   |   `-- *
   |-- rapport
   |   `-- rapport_final_180518.pdf
   `-- README.txt
```

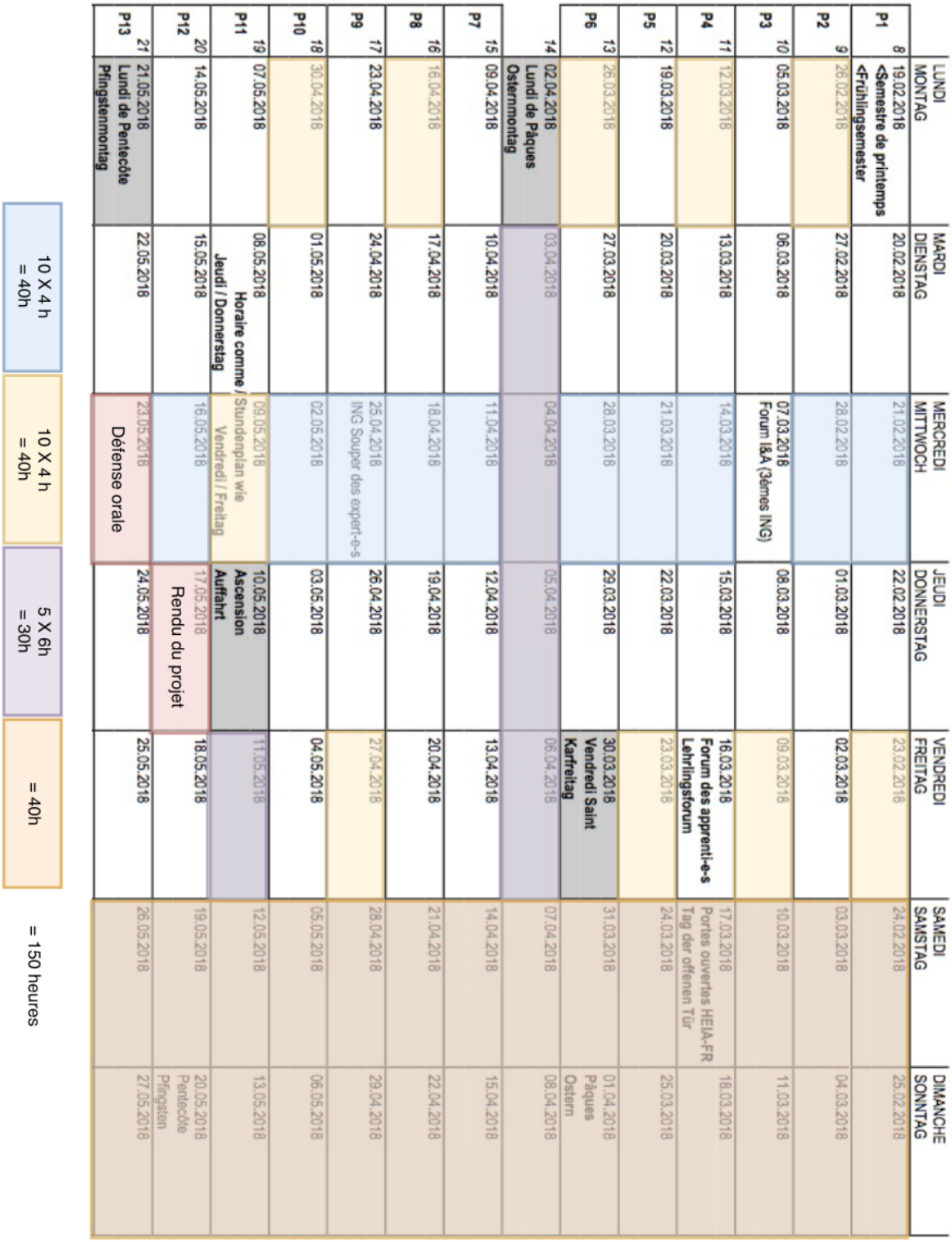
12.3 Planning



Version 1.0 - 18.05.2018	Semaines académiques													
Activités (séance tous les mercredis)	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	
	21-févr	28-févr	7-mars	14-mars	21-mars	28-mars	11-avr	18-avr	25-avr	2-mai	9-mai	16-mai	18-mai	23-mai
Cahier des charges & planning	X	X	X											
Réaliser la partie administrative avec la création des documents, des espaces de rendu	X													
Ecrire le cahier des charges	X	X	X											
Ecrire le planning	X	X	X											
Rendre le cahier des charges et le planning			O											
Analyse	X	X	X	X	X									
Prendre connaissance des travaux de Messieurs Gabriel Python et Simon Lièvre	X	X	X											
Mettre en place et configurer l'infrastructure de Monsieur Simon Lièvre		X	X											
Prendre connaissance et réaliser les scénarios de tests définit par Monsieur Simon Lièvre			X											
Rechercher des informations sur le Spirent, la QoS et les flux		X	X	X	X									
Réalisation d'un tableau comparatif avec différentes architectures				X	X									
Rendre la partie analyse (rapport de projet)				O										
Conception			X	X	X	X	X	X	X	X	X			
Définir la meilleure topologie pour évaluer au mieux les stratégies de QoS					X									
Définir le plan d'adressage IP du réseau de tests (NUT)			X	X										
Déterminer plusieurs scénarios de tests avec les types de trafic et les différentes QoS				X	X	X	X	X	X	X	X			
Rendre la partie conception (rapport de projet)							O							
Réalisation				X	X	X	X	X	X	X				
Mettre en place et configurer l'infrastructure (routeurs et Spirent)					X	X	X							
Mettre en place les stratégies QoS						X	X	X	X	X	X			
Réaliser les différents tests							X	X	X	X	X			
Rendre la partie réalisation (rapport de projet)											O			
Validation											X	X		
Valider les tests											X	X		
Documenter les tests							X	X	X	X	X	X		
Rendre la partie validation (rapport de projet)												O		
Rapport		X	X	X	X	X	X	X	X	X	X	X		
Rédaction du rapport		X	X	X	X	X	X	X	X	X	X	X		
Rendre le rapport final													O	
Défense orale														O

Légende	Cellule
A faire	
Fait (prévu)	X
Fait (pas prévu)	X
Jalons	O

12.3.1 Répartition des heures



	LUNDI MONTAG	MARDI DIENSTAG	MERCREDI MITTWOCH	JEUDI DONNERSTAG	VENDREDI FREITAG	SAMEDI SAMSTAG	DIMANCHE SONNTAG
P1 8	19.02.2018 **<Semestre de printemps <Frühlingsemester**	20.02.2018	21.02.2018	22.02.2018	23.02.2018	24.02.2018	25.02.2018
P2 9	26.02.2018	27.02.2018	28.02.2018	01.03.2018	02.03.2018	03.03.2018	04.03.2018
P3 10	05.03.2018	06.03.2018	07.03.2018 **Forum I&A (3èmes ING)**	08.03.2018	09.03.2018	10.03.2018	11.03.2018
P4 11	12.03.2018	13.03.2018	14.03.2018	15.03.2018	16.03.2018 **Forum des apprenti-e-s Lehrlingsforum**	17.03.2018 Portes ouvertes HEIA-FR Tag der offenen Tür	18.03.2018
P5 12	19.03.2018	20.03.2018	21.03.2018	22.03.2018	23.03.2018	24.03.2018	25.03.2018
P6 13	26.03.2018	27.03.2018	28.03.2018	29.03.2018	30.03.2018 **Vendredi Saint Karfreitag**	31.03.2018	01.04.2018 Pâques Ostern
14	02.04.2018 **Lundi de Pâques Ostermontag**	03.04.2018	04.04.2018	05.04.2018	06.04.2018	07.04.2018	08.04.2018
P7 15	09.04.2018	10.04.2018	11.04.2018	12.04.2018	13.04.2018	14.04.2018	15.04.2018
P8 16	16.04.2018	17.04.2018	18.04.2018	19.04.2018	20.04.2018	21.04.2018	22.04.2018
P9 17	23.04.2018	24.04.2018	25.04.2018 ING Souper des expert-e-s	26.04.2018	27.04.2018	28.04.2018	29.04.2018
P10 18	30.04.2018	01.05.2018	02.05.2018	03.05.2018	04.05.2018	05.05.2018	06.05.2018
P11 19	07.05.2018	08.05.2018 **Horaire comme / Stundenplan wie Jeudi / Donnerstag**	09.05.2018 Vendredi / Freitag	10.05.2018 **Ascension Auffahrt**	11.05.2018	12.05.2018	13.05.2018
P12 20	14.05.2018	15.05.2018	16.05.2018	17.05.2018 Rendu du projet	18.05.2018	19.05.2018	20.05.2018 Pentecôte Pfingsten
P13 21	21.05.2018 **Lundi de Pentecôte Pfingstenmontag**	22.05.2018	23.05.2018 Défense orale	24.05.2018	25.05.2018	26.05.2018	27.05.2018

10 X 4 h = 40h

10 X 4 h = 40h

5 X 6h = 30h

= 40h

= 150 heures