# Research en mogelijke oplossingen

Sedric Yaovi Lodonou

Opgesteld op 26/4/2020

Versie 1

**Probleem 1: Bittorrent**

## Wat is Bittorrent

Bittorrent is een filesharing systeem die ervoor zorgt dat een gebruiker een bestand kan downloaden via een van een andere host. Bij een traditionele download pc1 en pc2 hun bestanden rechtstreeks van de server. Elke client maakt een aparte connectie met de server en begint met downloaden. Bij bittorrent bestaat er geen aparte server. Elke host is zowel client als server.

Host1 zoekt een document die te vinden is op host 3 en host 3 zoekt een bestand die te vinden is op host 2. Host 1 verbindt zich met host 3 en begint te downloaden van host 3. Host 1 is de client en host 3 is de server. Host 3 verbindt zich met host2 en begint te downloaden van host 2. Hier is host 2 server en host 3 is client. In deze voorbeeld is host 3 zowel de client als de server.

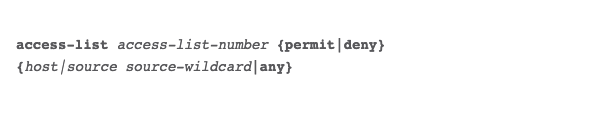
## Mogelijke oplossingen

### ACL

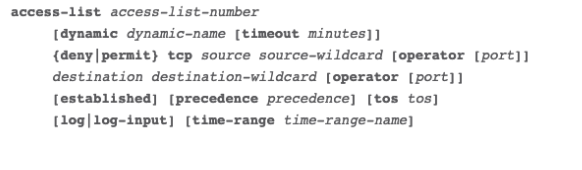
Access Control List (ACL) zijn wordt gebruikt om het netwerkverkeer te filter. Het is een soort firewall die geïmplementeerd kan worden in een netwerk om de inkomende en uitgaande verkeerstroom te filteren. Een ACL kan toegepast worden op een specifieke interface als een inbound of outbound.

Een Inbound Access List is wanneer de ACL toegepast wordt op het inkomend verkeer op de interface. Het inkomende verkeer wordt eerst gecontroleerd door de ACL voordat de veder naar buiten mag via de interface. Outbound Access List wanneer de ACL toegepast wordt op het uitgaand verkeer. Hier wordt het verkeer gecontroleerd voordat ze de uitgaande interface verlaten.

Er zijn 4 verschillende soorten ACL, maar gebruiken en focussen alleen op 2. Standaard en extended ACL. De standaard is een eenvoudige ACL die alleen de source adres gebruikt als criteria. Men kan eenvoudige toelating (Permits) of verbod (Deny) toepassen op een interface.



Een extended ACL is een beetje complexer. Bij een extended kan men zowel de scource als de destination address doorlaten of verbieden voor een host of alle hosten in het hele netwerk. Bij de extended kan er ook specifiek op een protocol worden gefilterd. Extended is een betere beveiliging en filter voor het netwerk.



### Sacavenger via Nbar

Een andere optie is het gebruiken van Nbar(Zie nbarConfig.pdf). Hiermee kunne we ervoor zorgen dat nbar een specifiek protocol deels- of niet toelaat. We gaan ervoor zorgen dat deze verkeerstype gemarkeerd wordt als Scavenger Class (Less-Than-Best-Service). Dit betekent dat p2p wel toegestaan wordt op het netwerk wanneer er genoeg bandbreedte is. Bij opstopping krijgt p2p heel weinig bandbreedte of soms worden de pakketten gedropt. Dit wordt toegepast bij verkeer die niet belangrijk is voor de dagdagelijkse prestaties. Dit kan door het verkeer een DSCP-waarde van SC1 te geven of via CBWFQ laag bandbreedte te geven.

## Conclusie

Voor deze probleem is de beste oplossing de nbar. De nbar is al actief voor het verzamelen van de verschillende statistieken, een extra functionaliteit eraan toevoegen zal geen impact hebben op het netwerk. Nbar is gemaakt om te helpen bij dit soort problemen en QoS te verbeteren dus het is van zelfspreken dat deze oplossing het beste optie is. Dit is de meeste flexibele oplossing voor deze probleem.

Bittorrent wordt niet heel vaak gebruikt, hierdoor is het wel beschikbaar wanneer het nodig. CVO-lethas is een school, tijdens de examenperiodes gebruiken leraren video’s voor bepaalde examen. Wanneer de statistieken opgevraagd wordt na het examen zal er zeker een protocol voor videostream aanwezig zijn. Dit kan ook het geval bij bittorent zijn.

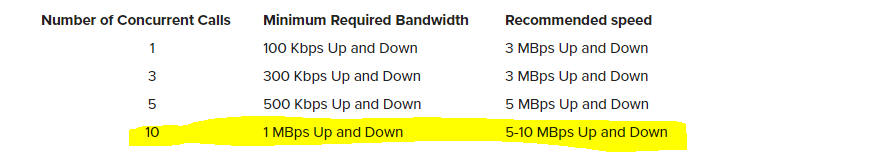
Het is dus beter Scavenger te gebruiken, zodat het toch gebruikt kan worden. Maar bij wanner het druk is in het verkeer zal de prioriteiten 1% zijn, dus het zal geen andere verkeer storen.

**Probleem 2:Telephonie/VOIP**

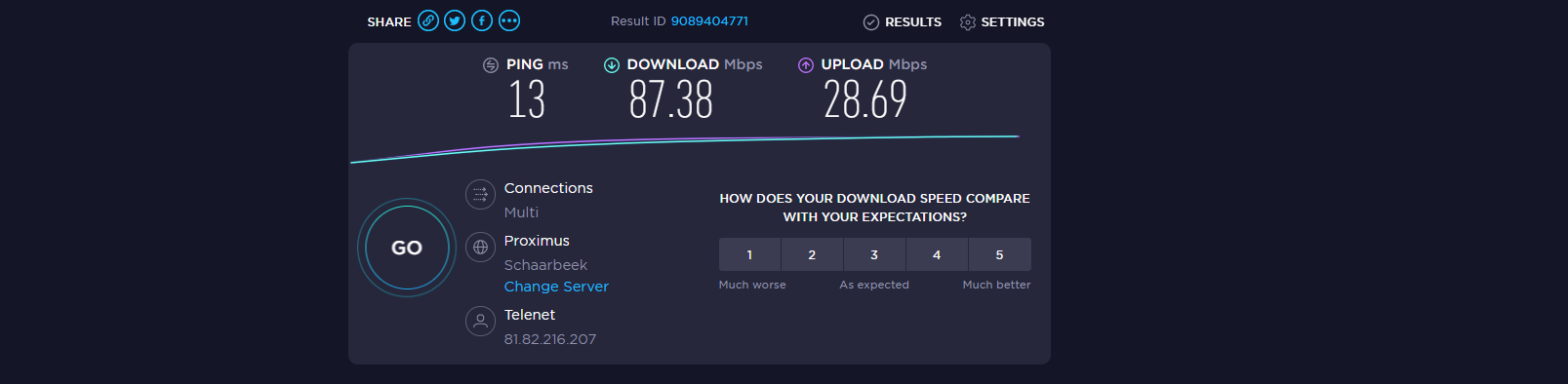
## Voip

## Voip gebruikt 2 belangrijke protocollen SIP en RTP. SIP (Session Initiation Protocol) wordt gebruikt voor het aanmaken en controleren van de connectie multimedia sessies zoals VOIP. RTP (Real-Time Transport Protocol) dit is een protocol die audio/media pakketten van VOIP vervoerd, RTP wordt gebruikt bij alle VOIP Communicaties. Om ervoor te zorgen dat er geen probleem is voor de VOIP zorgen ervoor dat dat VoIP een hoger prioriteit heeft en dat VoIP de minimum Kbps heeft.

Om QoS toe te passen op onse VoIp moeten eerste een paar berekingen doen. Ten eerste kijken we of we genoeg minimum verreisde bandwindth voor VoIp afhanklijk van de aantal gelijktijdige gesprekken. Hier is er ongeveer 10 gelijktijdige gesprekken op een heel drukke dag

**

Hierna moet”en we een speedtest doen om te zien of we wel genoeg Bandwidth hebben voor al onze telefoons. Zoals we hieronder zien is dit geen probleem.



De min bandwidth voor ons is 1000kbps(1MBps), in de tabel hierboven wordt 5 tot10 MBps aangeraden. Beste zal iets zij tussen de 2 omdat we maar 28MBps upload hebben dus zal niet slim zijn om de helft te reserveren voor VOIP. 3 tot 5 MBps zal beste optie zijn voor ons.

## Mogelijke Oplossingen

### Auto-QoS

Dit is een manier voor de automatisering van QoS. Nadat auto-Qos aangemaakt wordt, gaat het in discovery mode. Tijdens deze discovery periode worden er informatie over hert netwerkverkeer bekeken. Door de auo-QoS kan er na de discovery periode een advies gevraagd worden voor het volledige netwerkverkeer.

Na de *discovery* periode wordt QoS toegepast door Auto-Qos. Het voordeel is dat deze QoS methode al alle informatie verzameld heeft over het verkeer van het volledige verkeer. Er wordt een weergave gemaakt voor de beste verdelingen van prioriteiten, bandbreedte van VoIP, video en transactions. De bandbreedte en prioriteit kan ingesteld worden voor alles, waardoor het mogelijk is om een QoS implementatie te creëren die perfect is voor het netwerk.

Er wordt een QoS formule gemaakt die optimaal is voor het netwerk. Dit is auto-QoS dus het kan automatische aangepast worden als er een verandering is in het netwerk waardoor de ingestelde QoS niet meer effectief is.

### MQC LLQ

Low-Latency Queing is een toepassing die in auto-QoS zit. LLC is een toepassing is prioriteit queuing biedt aan de protocol VoIP (RTP) verkeer. Het is dus mogelijk om in plaats van Auto-QoS , alleen LLQ te toepassen. Dit is dus allen voor de VoIP en het moet elke keer manueel aangepast worden bij veranderingen in het netwerkverkeer.

## Conclusie

De beste optie is Auto-Qos omdat wee hier niet alleen VoIP hebben maar het volledige netwerkverkeer. Hiermee zal de QoS van het bedrijf altijd het beste keuze zijn voor het netwerk. Doordat het werkt via discovery van het netwerkverkeer, is de aanpassing van de QoS bij netverkeer veranderingen of bij het vergroten het netwerk heel eenvoudige.

**Probleem 3: wachtwoorden**

## Ten eerste maken we een “password policy”, dit zijn een stel regels die zullen toegepast worden op de wachten om het ingegeven wachtwoord te controleren. Minstens 8 teken, niet hetzelfde als vorige wachtwoord, hoofdletters, klein letters, cijfers en speciale karakters.

## Daarna We zullen gebruik maken van de "Maximum Password Age Policy " van Windows om een tijdsduur in te stellen zodat de gebruikers verplicht zijn om hun wachtwoorden aan te passen. Kan gaan van 1 tot 999 dagen. Volgens BBB(Better Business Bureau) is het beter om het om de 30,60 tot 90 dagen te veranderen. 90 dagen is zeker de minimum voor ons, voor de personeel stel ik voor een "Max age" van 150 dagen (5 Maanden).

## Conclusie

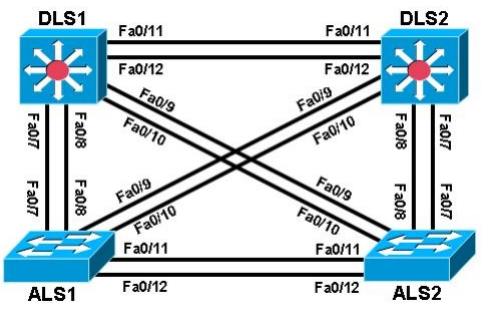
De beste optie is 5 maanden. Een artikel van 2019 zegt dat wanner werknemers te vaak hun wachtwoorden moeten veranderen kan dit voor een echte security risk zorgen. Werknemers kiezen wachtwoorden die gemakkelijk zijn, of een afwijking van het vorige wachtwoord. DE gekozen wachtwoord policy zorgt ervoor dat het wachtwoord een niveau van compliciteit heeft.

**Probleem 4:Redundantie**

## Mogeljke Oplossingen

### EtherCannel

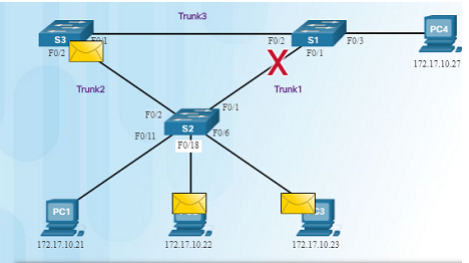
EtherChannel is een link die gecreëerd wordt door meer poorten met elkaar te groeperen. Bij EtherChannel worden er meerdere interfaces met elkaar verbonden om een logische interface van maken. Hierdoor is er load balance. Als er een link met vier poorten gemaakt heb, wordt de informatiestroom verdeeld door die vier kabels. Waardoor het verkeer sneller zal zijn, want poort is minder belast. Als er een probleem met een van de kabels is, zal de communicatie normaal veder gaan met de overgebleven kabels.



Hierboven is een voorbeeld van etherchannels op meerdere toestellen. Er lopen telkens twee kabels van een toestel naar de volgende, dit zijn etherchanels van twee kabels. Het is mogelijk om tot 8 kabels in een channel te zetten. Dit zorgt ook voor redundantie in het netwerk. Dus ook al valt 1 kabel uit of defect zal die link nog altijd bestaand omdat er andere kabels zijn die het werk veder verrichten. De link tussen de Core Router en de Core switch is een blangerijke link daar kan er etherchannel gemaakt worden.

### Spanning tree Protocol

Tweede oplossing die voor redudantie moet zorgen is spanningtree. Dit is een protocol die de snelste weg berekent en dan een andere weg aan duidt als back-up als primaire link uitvalt. We gaan de variatie Per-Vlan rapid Spanning tree gebruiken.



Eerst wordt er een root bridge uitgekozen. Het toestel met het laagste *bridge priority* wordt het rood bridge, als ze geen bridge priority hebben dan wordt de switch met het laagste IP-adres uitgekozen. De andere switches kiezen allenmaal een root port. Dit is de interface die direct geconnecteerd is met de root bridge, of de connectie die berekend is door de root bridge als beste connectie. De andere connecties van de switches worden om niet voor loops te zorgen. Alle andere interfaces worden uitgeschakeld, allen de interfaces van de root bridge worden allemaal in een forward state gezet.

Wanneer de primaire route die gekozen is door de root brigde uitvalt, dan worden er berekeningen gedaan op de overige connectie om een andere route te vinden. Het verkeer zal dan door de nieuwe connectie gaan, totdat er een de primaire connectie weer terug operationeel is. Dit zorgt voor redundantie maar verminderd ook de kans op loops in het netwerk.

## Conclusie

Voor deze probleem is de tweede oplossing het beste. Zoals boven besproken is Etherchannel niet van toepassing omdat er geen bottleneck is. STP zorgt voor redundantie die ervoor zorgt dat het netwerk altijd up zal zijn. Dit vermijdt het ontstaan van loops, loops kunnen voor grote vertraging in het netwerk zorgen. De vermelde pluspunten hierboven zorgen allebei voor een betere QoS voor het netwerk.

**Probleem 5. Back-up**

## TFTPD

We hebben een applicatie nodig die ons gaat helpen om via de commande-line in onze switch of router een backup te maken en op te slaan naar de gewenste locatie. Zodat we geen copy-past manoeuvre moeten doen en tijd kunnen winnen. Hiervoor maken we gebruik van tftpd (64), dit is een opensource applicatie die een paar handige features heeft waaronder tpftpserver die we kunnen gebruiken. ierHiermee kunne we via de commande-line opslaan naar onze toestel of een externe opslagruimte.

## Conclusie

Dit is een snelle en eenvoudige manier om een back-up te maken naar een externe harde schijf of een aparte server die in hetzelfde netwerk bevindt.

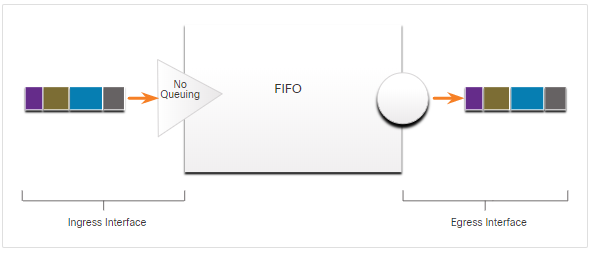
**Probleem 6 : Qos Design**

Om Qos te kunne toepassen moet er een onderzoek gedaan naar welke van de volgende oplossing best geschikt is voor het bedrijf. Om een goeie beslissing te kunne nemen moet er naar verschillende aspecten van QoS kijken. En de beste oplossing voor de werknemer zal een combinatie zijn. Voor de volledige onderzoek en uitleg van Qos zie document *“QpSConcepts.doc”*

## Queuing Algoritme

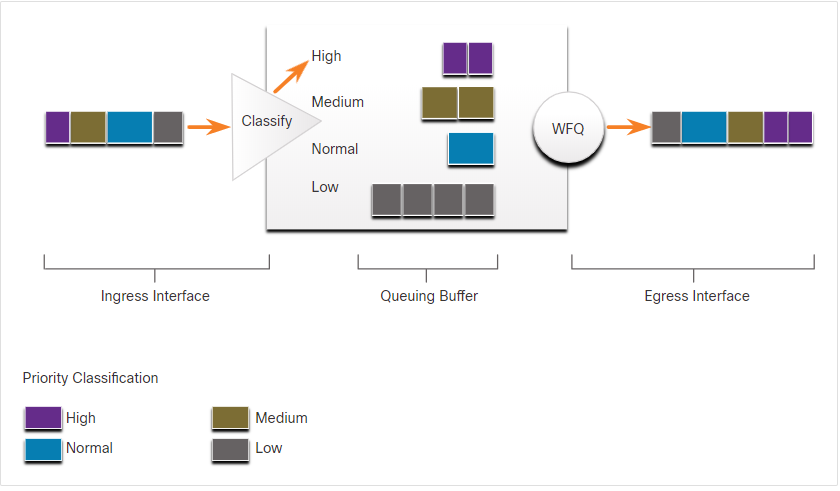
### FIFO

Dit is de eenvoudigste form van queuing. De pakketten worden in hetzelfde volgorde naar buiten gestuurd. Bij FIFO kan belangrijk en tijdsgevoelige verkeer gedropt worden bij een opstopping van de interface. FIFO is de snelste methode van queuing en wordt best gebruikt bij een grote link of een link met zeer weinig opstoppingen.



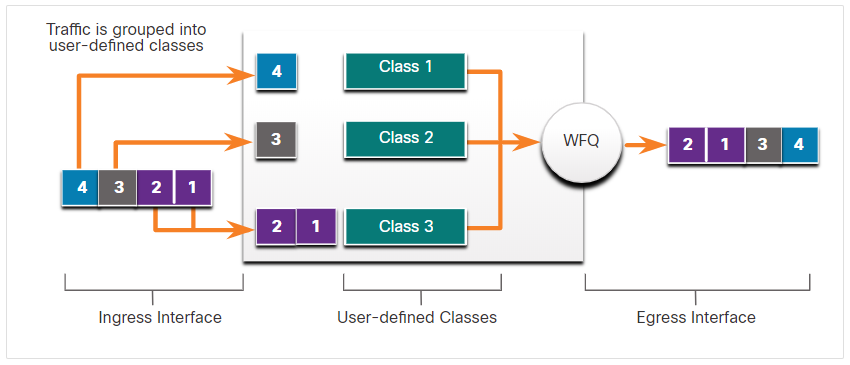
### Weighted Fair Queuing (WFQ)

In komende verkeer wordt geclassificeerd en afhankelijk van de classificatie krijgen ze een prioriteitsniveau. Het verkeer met hoogste prioriteit mag eerst door. Elke inkomende pakket krijgt een prioriteit. Dit zorgt ervoor dat de volledige bandbreedte eerlijk verdeeld wordt tussen het verschillende verkeer in het netwerk. WFQ zet het verkeer in verschillend soort afhankelijke van de Type of Service (TOS). De ToS gegevens in de IP-header wordt gebruikt om deze classificaties te maken. WFQ zorgt ervoor dat kleine interactieve verkeer zoals Telnet sessie en voice een hogere prioriteit krijgen dan grote verkeer zoals een ftp-sessie.



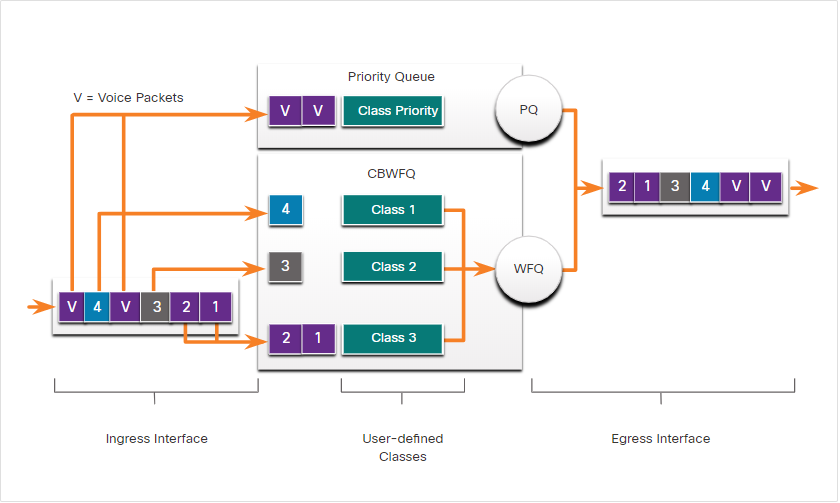
### Class-Based Weighted Fair Queuing

Dit is een verlenging van de WFQ. Hier moet gebruiker de verschillende klassen van het netwerkverkeer definiëren, doormidden van protocollen, Access control lists (ACLs)en de input interface. Elke class heeft zijn eigen FIFO-wachtrij. Elk class heeft zijn een paar karaktereigenschappen die geconfigureerd moeten worden. De karaktereigenschappen bandbreedte, weight en max pakketten in de wachtrij.



### LQoow Latency Queuing (LLQ)

LLQ zorgt ervoor dat er een extra priority Queuing bij de CBWFQ. Hiermee kunnen we specifieke ervoor zorgen dat belangrijke en tijdgevoelige verkeer zoals Voice eerst gestuurd wordt.

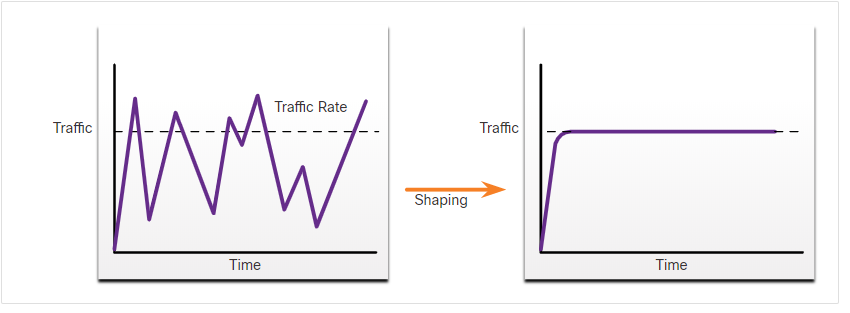


## Qos implemtation techniques:

Dit zijn twee mechanismes van Cisco QoS Software om opstopping tegen te houden.

### Shaping

Shaping is een concept voor verkeer die naar buiten(outbound) gaan Shaping neemt het overschot van pakketten en zet ze in een wachtrij. Deze pakketten worden dan geplant om later te versturen over een bepaalde periode. De planning aspect van shaping zorgt ervoor dat pakketten die vertraging hebben, in een wachtrij gaan en wachten om later opnieuw opgestuurd te worden. Voorbeeld van planning functies voor shaping zijn CBWFQ en LLQ.



### Policing

Policing is voor het verkeer die binnen(inbound) de interface komt. Wanneer het netwerkverkeer de maximun rate behaald, dan worden het overbodige verkeer gedropt. Policing wordt gebruikt door ISP.



## Qos Models:

### Integrated Services (IntServ).

Deze model is specifiek gemaakt voor de behoeften van de Real-Time Applicaties. Een applicatie stuurt een request van het netwerk voor een speciefiek service. De applicatie informeert het netwerk over zijn verkeersprofiel en vraagt om een bepaalde service die de vereisten bandbreedte en delay heeft. IntServ stuurt Resource Reservation Protocol (RSVP) naar de toestellen in het netwerk. Wanneer zij de nodige bandbreedte en delay kunnen reserveren, dan kan de applicatie beginnen te versturen.

### Differentiated Services (DiffServ)

Elke pakket wordt gemarkeerd voordat ze het netwerk binnen komen, deze markering wordt gedaan op basis van verkeerstype. Het netwerk gebruikt die markering om te weten hoe het pakket behandeld moet worden. Dit betekent ook dat elke netwerk device de nodige vereisten aan, Dit is geen end-to-end Qos Strategie.

## Conclusie

Eerst moesten er een QoS Model gekozen worden, de beste optie is DiffServ. Ten eerste is de IntServ een oud model. Ten tweede moeten alle toestellen in het netwerk RSVP ondersteunen en een verzoekt terug kunnen sturen. Wanneer dit niet gebeurt dan mislukt het proces. Ten slotte moeten alle toestellen in het netwerk alle verzoeken ven de verschillende applicaties apart bijhouden, dit kan heel verweldigend zijn op de toestellen.

Pakketten in het verkeer van de werkgever mogen niet gedropt worden, dus de beste optie is Shaping. Goeie voorbeelden van QoS algoritmes die hetzelfde soorten planningsmethode gebruiken zijn CBWFQ en LLQ.

Voor de Queuing algoritmes zijn de CBWFQ en LLQ de beste opties. Met deze twee algoritmes kunnen de verschillende pakketten geclassificeerd worden waardoor ze altijd herkenbaar zijn in het netwerk. Dit betekent ok dat de prioriteiten die een bepaalde Classen gegeven worden toegepast kunne worden in het hele netwerk. En hierdoor kan er oor voorrang gegeven worden aan VoIP en specifieke verkeerstypes vertragen of tegen houden.

De Qos Concept die geïmplementeerd zal worden is een combinatie van de boven gevonden oplossingen. Het wordt een DiffServ model, met een Shaping type implementatie voor de queuing en opstoppingen te vermijden. DiffServ is een Model die door markeringen prioriteiten kan geven aan pakketten. Hierdoor zijn de kunne we met LLQ en CBWFQ specifiek prioriteiten geven een aan bepaalde klassen en andere Classen tegenhouden of vertragen.

*[[1]](#footnote-1)*

*ip cef*

1. [NetCource]p.8-13 [↑](#footnote-ref-1)