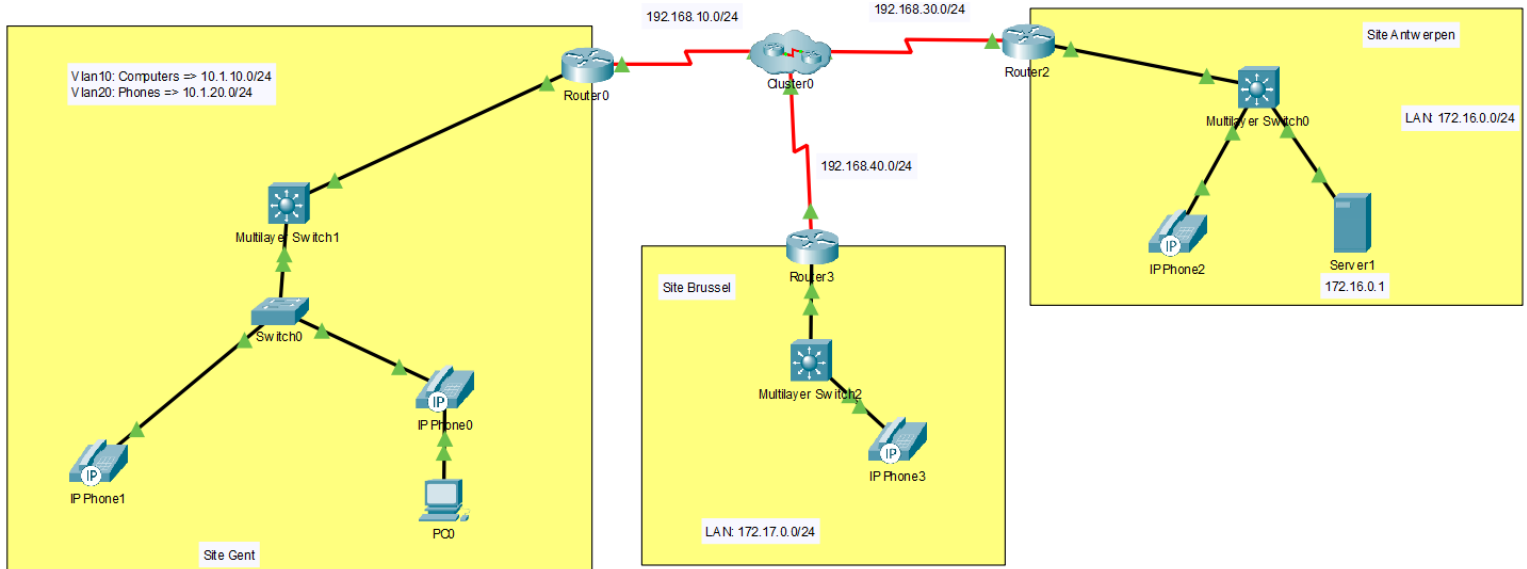


claLab QoS

Topologie



Doelstellingen

Qos implementeren: In dit lab is het de bedoeling om enkele basisconcepten van QoS toe te passen. Het classificeren, markeren, queuen en scheduleren komt aan bod tijdens dit lab.

Achtergrond / Scenario

Je bent netwerkeningenieur bij een groot bedrijf met meerdere vestigingen.

Dit bedrijf heeft een VoIP infrastructuur om telefonie tussen de verschillende sites mogelijk te maken.

De telefooncentrale die alle signaleringsverkeer (via SCCP) moet regelen is geïnstalleerd op de router in Gent. Verder werd in Antwerpen een webserver voorzien. Brussel is eerder een kleine site die enkel van de telefoniediensten in Gent en de webserver in Antwerpen gebruik maakt.

Meer info over de concepten van Qos vind je in het netacad curriculum of specifiek voor CBWFQ

<https://www.ciscopress.com/articles/article.asp?p=102233&seqNum=4> en voor de configuratie:

https://www.cisco.com/en/US/docs/ios/12_0t/12_0t5/feature/guide/cbwfq.html

Opdracht

Voorbereiding: Het netwerk zou reeds volledig functioneel moeten zijn. Dat betekent dat er volledige connectiviteit tussen elk toestel mogelijk is. Ook de individuele telefoons kunnen over de sites heen met elkaar communiceren.

(Via een demo wordt even aangetoond hoe je de telefonie kan gebruiken/simuleren.)

De verschillende sites zijn met relatief trage, seriële verbindingen met elkaar verbonden.

In deze oefening zal het heel belangrijk zijn om de “simulation mode” van Packet Tracer te gebruiken om de markering van de pakketten te bekijken.

De gekozen waardes zijn eerder indicatief en weerspiegelen niet altijd reële waardes. Deze oefening is dan ook vooral een methode om enkele QoS principes te leren begrijpen.

Stap 1: Site Gent

- **Doel:**

We willen enkele communicatie flows onderwerpen aan QoS op het moment dat ons verkeer de site Gent verlaat. We willen concreet er voor zorgen dat:

- **VoIP** verkeer **ten allen tijde prioriteit** krijgt t.o.v. ander uitgaand verkeer. Dit omdat VoIP zeer gevoelig is aan packet loss en delay. Verder moet VoIP verkeer een gegarandeerde bandbreedte krijgen van **200kbps**. VoIP verkeer wordt altijd als heel tijds-kritisch behandeld en krijgt dus de hoogste waarde voor markering: **Expedited Forwarding (EF of DSCP 46)**
- **HTTP** verkeer een minimaal gegarandeerde bandbreedte krijgt van **500kbps**. Voor de behandeling (prioritisering) krijgt dit **Assured Forwarding 31 (AF31 of DSCP 26)**.
- **ICMP** verkeer een minimaal gegarandeerde bandbreedte krijgt van **100kbps** en een **Assured Forwarding 11 (AF11 of DSCP 10)**

- **Uitwerking:**

- Het verkeer dat in Gent de site verlaat zal dus verschillende stappen van QoS moeten ondergaan.
 - ⇒ Classificeren, Markeren, Queuing en Scheduling
- Classificeren:
 - Dit kan op verschillende manieren. Je kan als netwerkbeheerder dus op verschillende manieren bepalen hoe je verkeer identificeert of in bepaalde categorieën steekt. Zonder classificatie worden alle pakketten op dezelfde manier behandeld.
 - Mogelijke parameters om verkeer te classificeren zijn:
 - ⇒ L1: De inkomende interface
 - ⇒ L2: Mac-adres, CoS, MPLS exp
 - ⇒ L3 : IP header fields (src-dst), DSCP, IP Precedence
 - ⇒ L4: TCP/UDP headers
 - ⇒ L7: Application signatures (NBAR)
 - In deze oefening kiezen we ervoor om het **HTTP en ICMP via NBAR** te laten herkennen. Het **VoIP verkeer** (hier wordt de spraak in een RTP-stream verstuurd) zullen we door **een access-list** identificeren (L3 headers). In dit laatste geval betekent dit veel minder

gedetailleerde classificatie aangezien alle verkeer vanaf het VoIP-verkeer in deze classificatie zal opgenomen worden.

○ **Configuratie classificeren:**

- Op een router of een switch kan verkeer geclassificeerd worden door een class-map aan te maken met “class-map”. Via een “match-any” of “match-all” kan respectievelijk één of alle onderliggende voorwaarden gecheckt worden om het verkeer te herkennen. Er worden hier dus 3 class-maps aangemaakt.

⇒ `Gent(config)#class-map match-any HTTP`
`Gent(config-cmap)# match protocol http`

⇒ `Gent(config)#class-map match-any ICMP`
`Gent(config-cmap)# match protocol icmp`

⇒ Maak zelf nog een derde class-map aan met de naam VOIP. Je zorgt er hiervoor dat het verkeer herkend wordt door een access-list die je aanmaakt (standard ACL). In de access-list neem je alle verkeer van het VoIP-netwerk op (10.1.20.0/24).

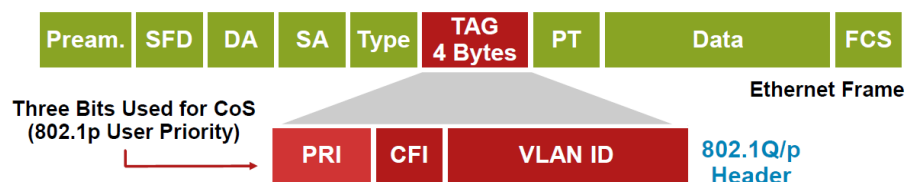
`Gent(config)#class-map match-any VOIP`
`Gent(config-cmap)# match.....`

- **Opgelet:** Verkeer dat niet specifiek in een class-map ondergebracht werd zal in een “default class” ondergebracht worden en gebruik mogen maken van de “resterende” resources. Er wordt dus automatisch een “class-map” met de naam “class-default” voorzien.

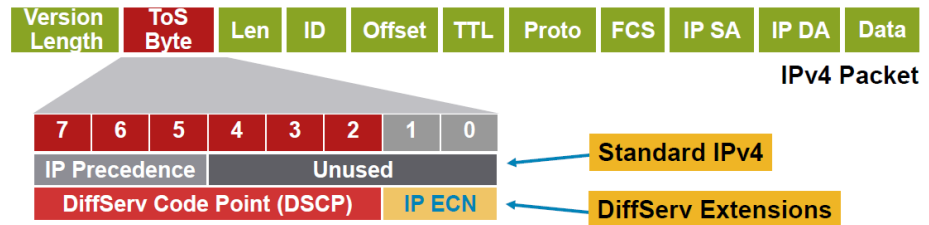
○ **Markeren:**

- Nadat je verkeer “herkend” werd is het de bedoeling om je pakketten ergens te “markeren” zodat verdere toestellen in je netwerk niet opnieuw het verkeer zouden moeten classificeren. In de theorie heb je gezien dat markeren van verkeer op 2 plaatsen kan: Ofwel op laag 2 ofwel op laag 3. Laag 3 heeft als voordeel dat de informatie in het pakket end-to-end behouden blijft, dus ook de QoS informatie. Laag 2 verliest na elke router (en dus her-encapsulatie) zijn QoS informatie die in het frame zit. Switches werken op laag 2. Dus in principe kan bij eenvoudige switches markering enkel op laag 2.

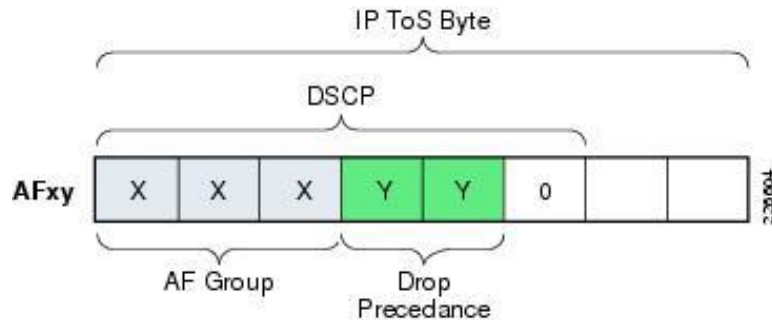
⇒ **Laag 2:** Bij Ethernet is er in een standaard frame geen plaats om nog QoS-info toe te voegen. Dat kan enkel bij een “Trunk” waar in het uitgebreidere frame, naast de Vlan-ID nog enkele bits over zijn om de CoS waardes (3bits, 8 mogelijke waardes) weg te schrijven.



- ⇒ **Laag 3:** Daar is het mogelijk om in het ToS veld 3 bits (IP Precedence) of 6 bits (DSCP) te gebruiken om de QoS waarden weg te schrijven.



IP Precedence (IPP) is analoog of compatibel aan CoS op laag2. DSCP geeft je veel meer granulariteit aangezien er 6 bits ter beschikking zijn.



In de figuur hierboven zie je duidelijk dat DSCP 6 bits gebruikt (de LSB staat altijd op "0"). Van die 6 bits worden de eerste drie gebruikt voor de classificatie in groepen (oa. Class Selectors) naar analogie IPP en CoS en de volgende 2 bits om te bepalen, indien er toch congestie optreedt, welke pakketten eerst mogen gedropt worden.

DSCP waarden kan je op 2 manieren gebruiken. Ofwel via hun decimale waarde of wel via hun AF classes (PHB-waarde) =>

PHB		DSCP		Maps to IP Precedence	
Default (Best Effort)		0	000000	0	
Assured Forwarding					
	Low Drop Pref.	Med Drop Pref.	High Drop Pref.		
Class 1	AF11	AF12	AF13	001 ¹⁰ 010 001 ¹² 100 001 ¹⁴ 110	1
Class 2	AF21	AF22	AF23	010 ¹⁸ 010 010 ²⁰ 100 010 ²² 110	2
Class 3	AF31	AF32	AF33	011 ²⁶ 010 011 ²⁸ 100 011 ³⁰ 110	3
Class 4	AF41	AF42	AF43	100 ³⁴ 010 100 ³⁶ 100 100 ³⁸ 110	4
Expedited Forwarding	EF			101 ⁴⁶ 110	5

○ **Configuratie Markeren:**

- Aangezien we QoS op het verkeer dat de router in Gent verlaat willen toepassen moeten we de markering op Laag 3 plaatsen.

- Verdere bepalingen (markering, shaping, scheduling,...) van de QoS acties worden nu ondergebracht in een “**policy-map**”. Een “policy-map” brengt alle acties van je QoS samen en kan op één of meerdere interfaces geplaatst worden. Per “class-map” bepaal je onder die policy-map de nodige parameters. Je verwijst telkens naar je “class-map” door dezelfde naamgeving te gebruiken:

```
⇒  Gent(config)#policy-map QoS_Gent
    Gent(config-pmap)#class HTTP
    Gent(config-pmap)#set ip dscp af31
```

```
⇒  Gent(config-pmap)#class ICMP
    Gent(config-pmap)#set ip dscp 10
```

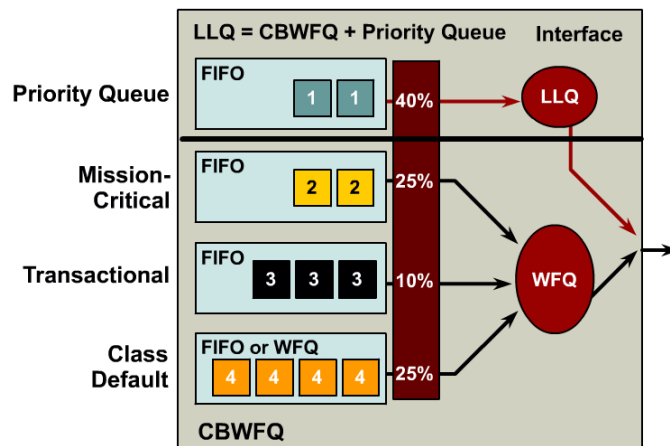
- Maak nu ook de 3de class aan voor VOIP die je de juiste DSCP-waarde geeft via de PHB-waarde

```
Gent(config-pmap)#class VOIP
Gent(config-pmap)#set.....
```

○ Queuing en Scheduling:

- Mochten we nu verder niets meer bepalen en gewoon de “policy-map” toepassen op een interface dan zal Class Based Weighted Fair Queuing (CBWFQ) enkel het verkeer in verschillende queues onderverdelen en uitsturen maar worden er nog geen garanties gegeven op het vlak van bandbreedte. De markering in verschillende klassen zorgt wel voor het gebruik van verschillende “gewichten” (Weighted Fair Queuing) bij het dispatchen van de pakketten op de uitgaande interface. Dus VoIP pakketten zouden meer gewicht krijgen dan http en icmp. Op zich hoeft dit niet want VoIP pakketten zijn vrij klein t.o.v. http maar moeten liefst wel absolute voorrang krijgen. Http mag ook niet tijdelijk de uitgaande interface “bezet” houden waardoor bv. voice pakketten niet tijdig buiten geraken.

Om voice-verkeer absolute voorrang te geven (kleine pakketten maar tijdsgevoelig) zullen we dit in een andere soort Queue brengen, een **priority queue** (PQ). Het andere verkeer blijft volgens Class Based Weighted Fair Queuing (CBWFQ) werken. De combinatie van **CBWFQ** en **PQ** wordt in de praktijk heel veel toegepast en heet **Low Latency Queuing**.



- **Configuratie Queuing en Scheduling:**
 - Ingrijpen op de queues en bandbreedtes bepalen doen we opnieuw per “class” in de policy-map.
 - ⇒ `Gent(config)#policy-map QoS_Gent`
`Gent(config-pmap)#class HTTP`
`Gent(config-pmap)#bandwidth 500`
 - ⇒ `Gent(config-pmap)#class ICMP`
`Gent(config-pmap)#bandwidth 100`
 - ⇒ **Opgelet:** VoIP-verkeer willen we in een andere type Queue brengen (PQ). De bandbreedte geven we dus op een andere manier op:

`Gent(config-pmap)#class VOIP`
`Gent(config-pmap)#priority 200`
 - Om nu effectief de QoS vereisten toe te passen op de router moet de policy-map gekoppeld worden aan een interface en dat voor een bepaalde richting (inbound of outbound)
 - ⇒ `Gent(config)#int s0/0/0`
`Gent(config-if)#service-policy output QoS_Gent`
- **Controle:**
 - Naast de simulation mode in Packet Tracer (zie demo en tips onderaan) zijn er enkele commando's in de CLI die je kunnen helpen om je QoS implementatie te analyseren
 - ⇒ `show policy-map <naam vd policymap>` geeft je de geconfigureerde policies terug.
 - ⇒ `show policy-map interface <interface>` geeft je aan hoeveel pakketten werden geclassificeerd en hoeveel pakketten er per seconde doorgelaten worden of hoeveel pakketten gedropt werden.

Stap 2: Site Antwerpen

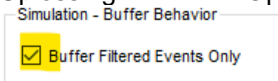
- **Doel:**

Het verkeer dat in Antwerpen binnenkomt willen we opnieuw onderwerpen aan QoS. In dit voorbeeld zullen we opnieuw ons beperken tot VoIP, Http en ICMP verkeer. Het idee is dat we het reeds gemarkeerde verkeer (door de router in Gent) niet vertrouwen en dus een hermarkering gaan uitvoeren (remark). Verder geven we geen extra prioriteiten of bandbreedtes mee voor deze types verkeer.

- **VoIP** Expedited Forwarding (EF of DSCP 46) hermarkeren naar **IPP 5**
- **HTTP** Assured Forwarding 31 (AF31 of DSCP 26) hermarkeren naar **IPP 3**
-
- **ICMP** Assured Forwarding 11 (AF11 of DSCP 10) hermarkeren naar IPP0
- **Uitwerking:**
 - Classificeren:
 - Het verkeer dat Antwerpen binnenkomt zullen we opnieuw moeten classificeren.

- Maak de “**class-maps**” (en een bijhorende **policy-map** met de naam “Remark”) aan voor:
 - ⇒ VOIP
 - identificeer op basis van dscp-waarde “ef”
 - markeer met IPP5
 - ⇒ HTTP
 - identificeer op basis van dscp-waarde “AF31”
 - markeer met IPP3
 - ⇒ ICMP
 - identificeer op basis van dscp-waarde “AF11”
 - markeer met IPP0
- Queuing en Scheduling:
 - Koppel de policy-map “Remark” aan de inkomende serial interface.
- **Controle:**
 - Bekijk opnieuw met de simulation mode en **show policy-map interface <interface>** of je pakketten inderdaad opnieuw gemarkeerd worden.

TIP:

- Packet Tracer Buffer Full tijdens simulation
Oplossing => Menu “Options” > “Preferences” > Tab “Miscellaneous” >

- Controle van de oefening: Controleren of pakketten (en frames) correct gemarkeerd werden kan enkel door in Packet Tracer in simulation mode te kijken naar het ToS veld en de hex-waarde van de DSCP te bekijken. In de animatie zou een gemarkeerd pakket ook een gekleurd blokje moeten krijgen.

