

# Project Telecom

## Implementatie van het IGMP protocol, versie 3.

Evert Heylen

### Structuur

- De `elements` map bevat alle code voor de elementen, en de inhoud hiervan kan dus in de `elements/local` map worden gezet.
- De `scripts` map bevat alle nodige Click scripts. Standaard heb ik de Multicast Server geconfigureerd op 224.4.4.4. Deze scripts worden ook gebruikt bij de tests.
- De `tests` map bevat alle tests, in Python 3.

### Compilatie

Om te compileren gebruik ik de volgende CXX flags om wél met C++11 te kunnen werken:

```
CXXFLAGS="-Wno-narrowing -std=c++11"
./configure --disable-linuxmodule --enable-local --enable-etherswitch
```

Of, voor de clang liefhebbers:

```
CXX=clang++ CXXFLAGS="-Wno-narrowing -Wno-string-plus-int -std=c++11 -g"
./configure --disable-linuxmodule --enable-local --enable-etherswitch
```

### Belangrijke opmerking

Om zowel de tests als de verdediging een beetje vlotter en sneller te laten verlopen, heb ik van sommige timers de default waardes verlaagd. Mocht het nodig zijn om ze terug op de default waarde te zetten zoals voorgeschreven staan ze in `elements/constants.hh` netjes opgelijst in de `defaults` namespace.

## Handlers

Elke router of client bevat een `igmp` en een `mc_table` element. Het `igmp` element heeft de volgende **write** handlers (enkel voor clients):

- `change_mode <INCLUDE/EXCLUDE> <group> <source>*`: Zal de modus aanpassen, en ook een Report sturen naar de router (indien de modus veranderde, anders zal er een ALLOW en een BLOCK worden gestuurd, zie p20 van de RFC).
  - `<group>` is een multicast group adres (class D)
  - `<source>` is een source adres
- `silent_change_mode <mode> <group> <source>*`: Doet het hetzelfde als hierboven, maar zonder hiervan de router op de hoogte te stellen. Nuttig om timers te testen.
- `change_sources <ALLOW/BLOCK> <group> <source>+`: Past enkel de sources aan en geeft dit door aan de router.
- `silent_change_sources <ALLOW/BLOCK> <group> <source>+`: Doet het hetzelfde als hierboven, maar zonder hiervan de router op de hoogte te stellen.

Een `mc_table` element heeft één **read** handler:

- `table`: print de multicast tabel af.

## Functionaliteit

Ik heb mij eigenlijk gebaseerd op de volledige RFC, en ben van pagina tot pagina gegaan. Omdat ik te weinig tijd had, is er nog wat functionaliteit over die ik *niet* heb:

- Hersturen van Group en Group-And-Source Queries (staat doorstreept in aangepaste RFC). De reden voor de afwezigheid hiervan is dat het vrij ver vanachter in RFC staat. Hersturen van antwoorden op deze queries gebeurt wel.
- De floating point modus van QQIC en `max_resp_code`. De code hiervoor was origineel geschreven door Stijn Vissers (als zijn enige bijdrage) en daardoor ben ik dat uit het oog verloren. Voor de eerlijkheid heb ik die code er dan toch uitgelaten.

Er is dus wel volledige support voor sources (inclusief Group-And-Source Queries), en ook voor routers als members. Voor sources heb ik een variant van `ipnetwork.click` gemaakt met een hele hoop meer sources en groups (3 groups, met 3 sources elks). Voor routers als members is er een script dat zoals `ipnetwork.click` werkt, maar elke client is vervangen door een netwerk zeer gelijkaardig als de originele `ipnetwork.click`. Er zijn dus in totaal 5 routers (1 in de root, en 4 in de netwerken eronder), en 16 clients (4 per netwerk).

## Tests

Ik heb mijn eigen test framework geschreven in Python, op basis van de modules `unittest` en `pexpect`. De tests testen voornamelijk op het aankomen van de pakketten (door de output van `click` na te gaan). Correcte checksums etc moeten manueel nagegaan worden in Wireshark. Om de tests ietwat snel te doen lopen zijn er bepaalde veronderstellingen gemaakt over lagere default waardes van timers (zie `constants.hh`). De tests werken niet tenzij de typfout in `ipnetwork.click` op lijn 70 wordt aangepast (een spatie te weinig). Er zijn ook extra scripts geschreven om te testen, die allemaal in de folder `scripts` staan.

Het uitvoeren van de tests zelf is niet getest op de referentie-PC's, maar na het installeren van de nodige libraries kunnen ze zo uitgevoerd worden:

```
cd tests
python3 alltests.py
```

Er zijn 4 tests voor de basisfunctionaliteit (gebruikt `ipnetwork.click`), 8 voor sources (gebruikt `sources_ipnetwork.click`), en 3 voor routers als members (gebruikt `routers_ipnetwork.click`).