

Project Telecom/Gedistribueerde Systemen

Bart Braem – Johan Bergs – Tom De Schepper

2015-2016

1 Introductie

In het geïntegreerde project van de vakken Telecommunicatiesystemen en Gedistribueerde Systemen moet een gedistribueerde applicatie gebouwd worden bovenop een netwerk waarbij je een protocol implementeert. Dit laatste gebeurt aan de hand van de Click Modular Router. Dit document bespreekt zowel de opgave voor het Telecommunicatiesystemen deel als dat voor het luik Gedistribueerde Systemen.

2 Telecommunicatie Systemen

2.1 Opgave: RSVP in Click

Implementeer in userlevel Click RSVP volgens de geannoteerde RFCs 2205 en 2210 die beschikbaar zijn op Blackboard. Houd je daarbij aan onderstaande richtlijnen en alle vereiste opmerkingen uit de RFC. Houd er rekening mee dat *admission control* niet moet worden ondersteund.

2.2 Richtlijnen

2.2.1 Elementen

Je elementen moeten verantwoord *push*, *pull* of *agnostic* zijn. Je moet ons kunnen uitleggen waarom je welke keuze maakte. Je elementen moeten bovendien compileren en werken als ze in */elements/local* worden geplaatst op een standaard Click 2.0.1 distributie.

2.2.2 Vragen

In geval van vragen over Click stel je die NIET op de Click mailinglist. Op Blackboard staat een forum waarop je vragen kan stellen en een FAQ kan nalezen. We sturen mededelingen enkel via Blackboard. Lees dus je studentenmail.

Als je voor een vraag liever even langskomt, stuur dan op voorhand een mail naar ons.

2.3 Evaluatie

2.3.1 Minimumvereisten om te slagen

Er zijn een aantal *minimale vereisten* waaraan jullie project moet voldoen om te kunnen slagen voor dit deel van het practicum. Als aan één van deze vereisten niet is voldaan, kan je nooit de helft van de punten halen:

- Je code moet compileren op het referentieplatform.
- Je code moet draaien op het referentieplatform door de gecompileerde click binary op te roepen met als argument het scriptje, voorzien voor de evaluatie (*ipnet-work.click*).
- Je elementen worden in de map */elements/local* (of in een submap daarvan) geplaatst en moeten op die manier compileren en werken. Elementen die je hebt gewijzigd of toegevoegd buiten deze map, worden sowieso verwijderd of vervangen door de originele elementen.
- Al je pakketten dienen correct te zijn volgens Wireshark. Met “correct” wordt bedoeld dat op zijn minst:
 - Alle **checksums** correct zijn.
 - Er geen pakketten zijn die geheel of deels niet door wireshark worden herkend.
 - De pakketformaten overeen komen met de specificaties in de RFC.
- In je code komen geen hardcoded IP, MAC of andere adressen voor.

2.3.2 Waar moet je verder op letten?

Als je project voldoet aan de minimumvereisten, wordt er voornamelijk op de volgende zaken gelet om je resultaat te beoordelen:

- Heb je de RFC correct gevolgd? Dit wil zeggen: volg je de RFC en heb je geen eigen versie ervan geïmplementeerd? Als er een bepaald gedrag in de RFC wordt vermeld, houd je je daar dan aan?
- Zijn alle velden/vlaggen/... in al je pakketten correct gezet (zie ook vorig punt)?
- Heb je inzicht in de RFC en in je code? Kan je ons uitleggen waarom iets is zoals je het hebt geïmplementeerd?

2.3.3 Tussentijdse evaluatie

We verwachten dat je volgende features hebt geïmplementeerd en kan demonstreren:

- het sturen van alle benodigde RSVP berichten met behulp van handlers (je hoeft dus nog geen timers te starten of onderhouden)
- op pakketten kan de TOS byte gezet worden met een handler
- een apart Click script toont hoe priority scheduling is geïmplementeerd met enkel bestaande Click elementen

Bijkomend verwachten we dat je (beknopt) redelijke waarden voorstelt voor de TSpec en RSpec parameters, om het gevraagde scenario te ondersteunen. Uiteraard mag je al extra functionaliteit insturen en tonen, als je al verder staat.

2.3.4 Eindevaluatie

We verwachten dat je, bovenop de vereisten voor de tussentijdse evaluatie, alle opgegeven stukken uit de RFC hebt geïmplementeerd en volgende features kan demonstreren:

- Alle berichten triggeren nu ook de relevante timers
- De datastream die quality of service vereist krijgt dat ook
- DiffServ gebeurt enkel op basis van klassen, gedefinieerd door de TOS byte in de IP headers maar gereserveerd op basis van poort en protocolnummer.

5.2 Referentieplatform

Het referentieplatform zijn de computers in het computerlabo op de tweede verdieping. Op de evaluatie wordt verwacht dat jullie code compileert en draait op deze VM.

6 Inzending en deadlines

6.1 Inzending

We verwachten dat je bij elke inzending een manual of README meelevert waarin duidelijk staat hoe we jou code moeten gebruiken om aan te tonen dat die doet wat wordt verwacht. Voor het Click project mag je uiteraard zelf geschreven scriptjes mee indienen als dat het aantonen van bepaalde functionaliteit vergemakkelijkt. Naast een manual moet voor het Avro project ook een beknopte beschrijving van de architectuur en de gebruikte technologieën en libraries worden toegevoegd.

De code moet tegen de deadline via de uploadzone van het vak Telecommunicatie Systemen op Blackboard worden opgestuurd. Zorg ervoor dat alle benodigde elementen, scripts, files en libraries aanwezig zijn, want enkel en alleen deze code wordt gebruikt op je evaluatie. Test alles op voorhand!

6.2 Deadlines

Voor dit project zijn er volgende deadlines:

- **Zondag 22 november 2015 23u59:** Tussentijdse evaluatie.
- **Zondag 3 januari 2016 23u59:** Finale inzending.

De eindevaluatie vindt plaats in de examenreeks. Jullie maken zelf een afspraak voor een evaluatie nadat de lijst met mogelijke datums is doorgegeven.