

Project Telecom: RSVP Implementatie in Click

Johan Bergs – Jeremy Van den Eynde

2020 - 2021

1 Opgave

1.1 RFC

Implementeer in userlevel Click RSVP volgens de RFCs 2205 en 2210 die beschikbaar zijn op Blackboard. Houd je daarbij aan onderstaande richtlijnen en alle vereiste opmerkingen uit de RFC. De versie van de RFCs die beschikbaar zijn op Blackboard, is aangepast zodat de stukken die niet dienen te worden geïmplementeerd, zijn doorstreept. Houd er ook rekening mee dat *admission control* niet moet worden ondersteund.

2 Richtlijnen

2.1 Groepswerk

Dit project los je per twee op. Schrijf je in in een groepje op Blackboard vóór 13 maart. Problemen met de samenwerking meld je zo snel mogelijk, anders kunnen we daar geen rekening mee houden.

2.2 Referentieplatform

Het referentieplatform is de virtuele machine die ter beschikking is gesteld. Op de evaluatie wordt verwacht dat jullie code compileert en draait op deze VM.

2.3 Referentie-implementatie

Op de referentie-VM staat ook een referentie-implementatie geïnstalleerd. Deze referentie-implementatie implementeert alle componenten volgens de geannoteerde RFC. Er is documentatie voorzien hoe je deze implementatie kan gebruiken om bijvoorbeeld je eigen implementatie tegen te testen.

2.4 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 `/home/student/click/elements/local/rsvp/` worden geplaatst op de standaard click versie voorzien op de VM.

2.5 Vragen

In geval van vragen over Click of de opgave kan je die steeds aan ons stellen. Stuur je vraag bij voorkeur via e-mail naar ons beiden zodat we ze zo snel mogelijk kunnen behandelen. De meeste vragen zijn per mail wel te beantwoorden, maar als je liever toch uitleg via audio en/of video wil, stuur dan op voorhand een mail om een afspraak te maken.

Dinsdag's zullen we in het voorziene Telecom tijdslot (16u-18u) normaal gezien steeds beschikbaar zijn. Als er iets aan deze regeling verandert (eenmalig of structureel), zal dit via Blackboard worden aangekondigd.

We sturen mededelingen enkel via Blackboard. Lees dus je studentenmail.

3 Evaluatie

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 is op tijd (vóór het verstrijken van de deadline) ingediend via Blackboard.
- Je code moet compileren op het referentieplatform.
- Je code moet draaien op het referentieplatform door het daarvoor voorzien scriptje te starten (*start_click.sh*).
- Je elementen worden in de map */home/student/click/elements/local/rsvp/* 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. Broadcast adressen (bijvoorbeeld 255.255.255.255, 224.0.0.1 of ff:ff:ff:ff:ff) kunnen hier een uitzondering op vormen.
- Is je eigen implementatie interoperabel met de referentie-implementatie, wanneer deze wordt gebruikt in een “alles verloopt normaal” scenario? Hiermee wordt bedoeld dat er geen verwachte of onverwachte fouten optreden, dus dat de referentie-implementatie zich “gedraagt zoals het hoort”. In dit geval dient alles in jouw implementatie ook correct te werken!

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?
- Is je eigen implementatie interoperabel met de referentie-implementatie, ook als deze (verwacht of onverwacht) foutieve pakketten genereert? Hiermee wordt bedoeld dat, in geval er foutieve pakketten toekomen, je hier gepast (d.w.z. volgens de RFC) op reageert. Crashende applicaties vallen uiteraard niet onder “gepast reageren”.

3.3 Demonstraties

We verwachten dat je voor de demonstratie een README meelevert waarin duidelijk staat hoe we jouw code moeten gebruiken om aan te tonen dat die doet wat wordt verwacht. Je mag uiteraard zelf geschreven scriptjes mee indienen als dat het aantonen van bepaalde functionaliteit vergemakkelijkt.

3.4 Indienen code

De code moet tegen de deadline via Blackboard worden opgestuurd. Zorg ervoor dat alle benodigde elementen en scripts aanwezig zijn, want enkel en alleen deze code wordt gebruikt op je evaluatie. Verzeker je er van dat je de juiste versie hebt ingediend van je code, en test al je code met de ingediende zip file!

3.5 Tussentijdse evaluatie

Indienen via Blackboard kan tot 18 april 2021 23u59. We verwachten dat je volgende zaken 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. Jullie maken zelf een afspraak voor een evaluatie nadat de lijst met mogelijke momenten is doorgegeven. Uiteraard mag je al extra functionaliteit insturen en tonen, als je al verder staat.

3.6 Eindevaluatie

De definitieve versie van jullie code moet binnen zijn vóór 24 mei 2021 23u59. De eindevaluatie vindt plaats in de examenreeks. Jullie maken zelf een afspraak voor een evaluatie nadat de lijst met mogelijke momenten is doorgegeven. 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.

3.7 Tweede zittijd

Als je niet geslaagd zou zijn in de eerste zittijd, krijg je een nieuwe kans in de tweede zittijd om je project in te dienen. Daarbij gelden volgende regels:

- In tegenstelling tot de eerste zittijd, is de tweede zittijd een **individueel project**. Je werkt niet meer per twee, maar alleen.
- De opgave is dezelfde als die voor de groepjes in eerste zittijd. Dit geldt voor iedereen, ook voor degenen die om één of andere reden de opgave, mogelijks met stukken die wegvielen, in eerste zittijd alleen hebben ingediend. Ook voor hen geldt: de opgave voor de tweede zittijd is de **volledige opgave, individueel ingediend**.
- Uiteraard mag je vertrekken van wat je in eerste zittijd hebt ingediend. Dit is de “gemeenschappelijke basis” van waar beide groepsleden mogen vertrekken.