

Project Telecommunicatiesystemen: Mobile IP Implementatie in Click

Johan Berghs – Jeremy Van den Eynde

2017 - 2018

1 Opgave

1.1 RFC

Implementeer in userlevel Click MobileIP volgens de RFC 5944 die beschikbaar is op Blackboard. Houd je daarbij aan onderstaande richtlijnen en alle vereiste opmerkingen uit de RFC. De versie van RFC 5944 die beschikbaar is op Blackboard, is aangepast zodat de stukken die niet dienen te worden geïmplementeerd, zijn doorstreept. De paragrafen die zijn gemarkeerd, worden hieronder verduidelijkt.

1.1.1 Co-located care-of addresses

Deze moeten niet worden ondersteund. Indien de Mobile Node dit aanvraagt, moet een gepaste weigering worden gestuurd. Deze weigering staat uitgelegd in de RFC.

1.1.2 Fragmentatie

Fragmentatie moet niet worden ondersteund. Je mag veronderstellen dat geen enkel pakket groter is dan de MTU.

1.1.3 Security

Security moet niet worden ondersteund. De specifieke hoofdstukken uit de RFC staan gemarkeerd in de PDF.

1.1.4 Meerdere agents op een netwerk

Je mag er vanuit gaan dat er maximaal 1 home agent en 1 foreign agent per netwerk aanwezig is.

1.1.5 Andere RFC's

Om de opdracht helemaal juist af te werken moet je nog twee andere (kleine) RFC's doorlezen, namelijk voor IP-in-IP encapsulatie en ICMP router advertisements. Een deel van de opdracht is uitzoeken welke RFC's dit zijn.

1.1.6 Hoofdstuk 2.3 pagina 25

In plaats van het adres 224.0.0.11 (Mobile-Agents multicast group), gebruik je 255.255.255.255.

1.1.7 Hoofdstuk 3.4 pagina 37 & 38

De aangeduide replies voor *registration denied* moeten zeker worden ondersteund. Uiteraard moet een succesvolle registratie een correcte reply teruggeven.

1.1.8 Hoofdstuk 3.6 pagina 42

Je mag ervan uitgaan dat de Mobile Node geconfigureerd is met een geldig IP adres.

1.1.9 Hoofdstuk 3.6.1.1 pagina 43

In plaats van het adres 224.0.0.11 (Mobile-Agents multicast group), gebruik je 255.255.255.255.

1.1.10 Hoofdstuk 3.6.1.2 pagina 44

Zowel de Mobile Node als de Home Agent moeten geen ondersteuning bieden voor *multiple simultaneous mobility bindings*. Uiteraard moet de HA wel correct reageren als een Mobile Node hier om vraagt in een registration request.

1.1.11 Hoofdstuk 3.6.1.2 pagina 46

Zorg er voor dat de waarde *ICMP Router Advertisement Lifetime* ergens instelbaar is.

1.1.12 Hoofdstuk 4.6 pagina 76

De eerste paragraaf op deze pagina specificeert dat een mobile node die niet thuis is geen ARP requests mag sturen en ook geen ARP replies mag beantwoorden. Deze opmerking mag je negeren. In de eenvoudige situaties waarin je gaat werken levert dit toch geen problemen op. Proxy ARP hoeft je niet te implementeren.

2 Richtlijnen

2.1 Groepswork

Dit project los je per twee op. Schrijf je in in een groepje op Blackboard vóór 21 oktober. 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 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* (of in een submap daarvan) worden geplaatst op de standaard click versie voorzien op de PCs.

2.4 Vragen

In geval van vragen over Click stel 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 even langskomt, stuur dan op voorhand een mail om een afspraak te maken. We sturen mededelingen enkel via Blackboard. Lees dus je studentenmail.

2.5 Referentie-implementatie

Op de referentie-VM staat ook een referentie-implementatie genstalleerd. Deze referentie-implementatie implementeert home agent, foreign agent en mobile node volgens de geannoteerde RFC. Er is documentatie voorzien hoe je deze implementatie kan gebruiken om bijvoorbeeld je eigen implementatie tegen te testen.

3 Evaluaties

3.0.1 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 */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. Broadcast adressen (bijvoorbeeld 255.255.255.255 of ff: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!

3.0.2 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”.

3.1 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.

3.2 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. Test alles op voorhand!

3.3 Tussentijdse evaluatie

Indienen via Blackboard kan tot 26 november 2017 23u59. We verwachten dat je de volgende zaken hebt geïmplementeerd en kan demonstreren:

- Tunneling van pakketten voor mobile nodes.
- Registration requests en replies opsturen met vast gekozen vlaggen.

- Een agent is ofwel home agent ofwel foreign agent maar niet beiden.

Uiteraard mag je al extra functionaliteit insturen en tonen, als je al verder staat.

3.4 Eindevaluatie

De definitieve versie van jullie code moet binnen zijn vóór 7 januari 2018 23u59. De eindevaluatie vindt plaats in de examenreeks. Jullie maken zelf een afspraak voor een evaluatie nadat de lijst met mogelijke datums is doorgegeven. We verwachten dat je, bovenop de vereisten voor de tussentijdse evaluatie, alle opgegeven stukken uit de RFC geïmplementeerd hebt en volgende zaken kan demonstreren:

- Alles is dynamisch, afhankelijk van de situatie dus.
- Signalisatieberichten worden periodiek herhaald.
- Registraties vervallen als ze niet worden vernieuwd.
- De mobile node ontdekt zelf dat hij op een nieuw netwerk zit aan de hand van agent advertisements.

3.5 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.

3.6 Puntenverdeling

De puntenverdeling is als volgt: 10 punten staan op het theorie-examen, 10 punten op deze opgave. Je moet op beide onderdelen slagen om in totaal geslaagd te kunnen zijn. Je eindresultaat wordt daarom als volgt berekend:

- Indien je op beide onderdelen bent geslaagd, is je eindresultaat de som van beide onderdelen. Je haalt bijvoorbeeld 7/10 op de theorie en 6/10 op de praktijk. In totaal behaal je dan 13/20. Met een 5/10 op theorie en een 5/10 op de praktijk, eindig je met een 10/20.
- Indien je op één van beide onderdelen (of op beide onderdelen) niet bent geslaagd, is je eindresultaat de gehalveerde som van beide onderdelen. Concreet: $2/10 + 4/10$ levert je in totaal 3/20 op. Een ander voorbeeld: $10/10 + 4/10$ levert je een 7/20 op.