

Abstract

Door de huidige verschuiving naar cloud-gebaseerde technologieën zal het belang van netwerk-protocollen en beschikbaarheid van netwerken alleen maar toenemen. Om deze verschuiving vlot te laten verlopen is er meer en meer onderzoek nodig naar netwerktechnologieën. Voor dit onderzoek wordt er dan ook veelvuldig gebruik gemaakt van testbeds. Testbeds worden aangestuurd via jFed en worden gebruikt om netwerken te simuleren. De situatie heeft enkele nadelen. Het is voor een onderzoeker soms erg moeilijk om te bepalen of een bepaald gedrag in hun experiment te wijten is aan de eigen ontwikkelingen, of aan het testbed zelf.

Deze masterproef zal trachten dit probleem te verhelpen door de testbed monitoring te automatiseren. Via een webservice zal deze informatie dan aangeboden worden. Deze webservice zal weergeven hoe betrouwbaar een testbed is. Hiervoor komen verschillende ontwikkelingstools aan bod. Uiteindelijk zal deze service ervoor zorgen dat een onderzoeker via zijn primaire gebruikersinterface op de hoogte gebracht wordt van eventuele problemen.

Abstract

Due to the new cloud based technologies, network protocols and network reachability are now more important than ever. To make this change quick and clean, we need more and more network research. This research heavily relies on testbeds to simulate networks. These testbeds are controlled via software tools. In this thesis we will look at jFed, a java tool developed in the European FED4FIRE project. The current situation has a few downsides. It's very hard for researchers to determine if a certain behavior is caused by the testconfiguration or by the testbed.

This thesis will try to solve that matter, by automating the testbed monitoring. A webservice will then share this information. This webservice will show how reliable a testbed is. The webservice could then be used to notify the researcher of problems via his primary user interface.

Inhoudsopgave

1	Inleiding	1
1.1	Bestaande situatie	1
1.2	Probleemstelling	3
2	Uitwerking	4
2.1	monitoring service	4

Hoofdstuk 1

Inleiding

1.1 Bestaande situatie

iMinds is een onafhankelijk onderzoekscentrum dat opgericht werd door de Vlaamse overheid. Het is voornamelijk bezig met onderzoek omtrent ICT-innovatie. Bij dit onderzoek wordt veelvuldig gebruik gemaakt van testbeds. Een testbed is een verzameling van nodes waarmee netwerkopstellingen kunnen gesimuleerd worden.

Een goed voorbeeld van een testbed is de Virtual Wall. De Virtual Wall is ontworpen voor het experimenteren met nieuwe netwerkoplossingen voor de volgende generatie van het Internet. De kracht van deze infrastructuur is dat ze de mogelijkheid biedt om op een eenvoudige wijze specifieke netwerkopstellingen op te zetten volgens de noden van het onderzoek.

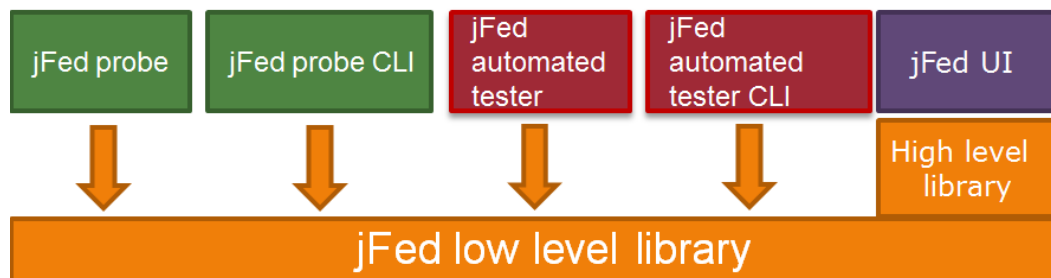
Een voorbeeld waarvoor dit testbed wordt gebruikt is dat men een server en een aantal cliënten definieert. Deze worden verbonden met een aantal tussenliggende routers. Vervolgens wordt een videostream opgestart. Op deze videostream kan men storing introduceren door pakketten te droppen. Deze storing zal ervoor zorgen dat het beeld aan de cliënt-side hapert. Er kunnen technieken ingebouwd worden aan cliënt-side om deze storing op te vangen. Zo kan er overgeschakeld worden naar een lagere kwaliteit indien blijkt dat de beschikbare bandbreedte onvoldoende is. Testen van dergelijke technieken verloopt dan ook aan de hand van testbeds.

Onderzoekers bij iMinds hebben niet alleen toegang tot hun Virtual Wall, maar door samenwerkingen met diverse onderzoekscentra hebben ze ook wereldwijd toegang tot andere en soms erg verschillende testbeds. Deze zijn elk apart ontwikkeld. Hierdoor maakt elk testbed gebruik van een eigen interface. Deze interface wordt aangeboden door middel van een federation API. Een federation API is een interface die alle mogelijk functionaliteiten van een specifiek testbed aanbiedt. Hij staat in voor de communicatie tussen een programma en het testbed. Onderzoekers die met meerdere testbeds werken, moeten dan ook de werking

van elk testbed apart bestuderen alvorens ermee aan de slag te kunnen. Ook rechtstreekse verbindingen leggen tussen verschillende testbeds wordt hierdoor moeilijker.

Daarom werd in het Europese onderzoeksproject Fed4FIRE beslist om een aantal federation API's vast te leggen die gelijk moeten zijn op elk testbed. Dit maakt het mogelijk om tools te ontwikkelen die bruikbaar zijn op alle testbeds verenigd in dit project. Hiervoor werd de tool jFed ontwikkeld door iMinds1. jFed biedt een wrapper aan rond de specifieke federation API's die in dat project worden toegepast. Daarenboven voorziet jFed in verschillende applicaties om enerzijds te kunnen testen of testbeds deze API's correct ondersteunen, en anderzijds om als onderzoeker op een eenvoudige wijze een experiment te kunnen opzetten.

De kernmodule waar alle andere modules gebruik van maken is de "jFed library". De voornaamste functie van deze library is om de communicatie met de testbeds te vereenvoudigen. De hiervoor bruikbare testbeds implementeren enkele gestandaardiseerde API's, die de benodigde functionaliteiten aanbieden. De jFed library implementeert de client kant van deze API's, en voorziet Java interfaces voor de API-calls. De gebruiker zal hierdoor niet geconfronteerd worden met deze details. Zo zal de library o.a. de achterliggend SSL-connectie en de bijhorende certificaten en private keys beheren. Verder zullen ook de nodige omzettingen gebeuren. Daarnaast bevat deze library vele hulpklassen en methoden om de API's aan te spreken.



Figuur 1.1: Opbouw van jFed

Een goed voorbeeld van een API die door jFed wordt geïmplementeerd is de Aggregate Manager API, versie 2 en 3. Een voorbeeld van een call, is de "CreateSliver" call in de AMv2 API. De AMv2 API specificeert dat alle calls gedaan worden over een SSL verbinding met client-side certificate. Verder specificeert de API de exacte argumenten die vereist zijn en het formaat van het resultaat. De jFed library biedt een Java methode "createSliver" in de klasse "AggregateManager2". De argumenten worden opgegeven in een Java formaat. Zo zal bijvoorbeeld een datum-argument als een Java Date-object doorgegeven worden. De jFed library zal de datum naar het RFC3339 formaat omzetten voor het naar de server verstuurd wordt. Ook het resultaat van de call wordt verwerkt door de jFed library terug omgezet.

Steunend op deze library zijn enkele tools zoals jFed probe en jFed compliance tester gebouwd. Deze hebben als doel het valideren of testbeds de federation API's correct ondersteunen. Tenslotte is er de jFed GUI tool. Deze tool maakt het mogelijk om als eindgebruiker een experiment op een intuïtieve manier op te zetten. De jFed GUI heeft momenteel echter een relatief complexe interface die gebruiksvriendelijker gemaakt kan worden.

1.2 Probleemstelling

Er is echter nog ruimte voor verbetering in jFed. Zo worden er op geregelde tijdstippen verschillende testen automatisch uitgevoerd op testbeds. Deze testen geven o.a. weer of testbeds online zijn en of het aanmelden via een SSH-verbinding gelukt is. Deze resultaten worden momenteel nog redelijk ad hoc bijgehouden. Men maakt geen gebruik van een databank, maar van bash scripts om de resultaten te archiveren. Resultaten op deze manier bijhouden, is uiteraard inefficiënt. Bovendien staat deze informatie momenteel enkel op een enkele specifieke website vermeld.

Een onderzoeker wordt dus niet via zijn primaire gebruikersinterface op de hoogte gebracht van eventuele storingen met de testbeds opgenomen in zijn experiment. Dit maakt het voor onderzoekers soms moeilijk om te identificeren of een bepaald gedrag in hun experiment te wijten is aan de eigen ontwikkelingen, of aan de testbeds zelf. Hierdoor kan veel tijd onnodig verloren gaan tijdens de uitvoering en de analyse van het experiment. Een betere oplossing zou zijn dat bij het selecteren van een testbed de betrouwbaarheid weergegeven wordt. Zo kan een betrouwbaar testbed gekozen worden, om dergelijke situaties zoveel mogelijk te vermijden.

Niet enkel de betrouwbaarheid is van belang, ook het 'comfort' is van belang. Een testbed met een snellere verbinding geniet de voorkeur op een testbed met een trage verbinding. Naast de betrouwbaarheid en comfort moet ook gekeken worden naar de vereisten. Indien een betrouwbaar testbed niet beschikt over voldoende resources kan het niet gebruikt worden voor een experiment.

Door rekening houden met de betrouwbaarheid, het comfort en de vereisten kan jFed de onderzoeker melden dat een gekozen testbed niet betrouwbaar is. Hiervoor is een monitoring service nodig die bijhoudt hoe vaak en hoelang een testbed offline is.

Hoofdstuk 2

Uitwerking

2.1 monitoring service

Er moeten een aantal zaken bijgehouden worden.

1. Hoe vaak is een testbed offline?
2. Hoe lang is een testbed offline?
3. Hoe snel is de verbinding met een testbed?
4. Hoeveel resources zijn er beschikbaar?

De eerste