

Afstudeerscriptie

IPv6 implementatie in NetYCE

Derco Sportel

1548287



Colofon

| | |
|--------------------|---|
| | NetYCE B.V. Postbus 320 1380 AH Weesp, Nederland Business Centre Muiderpoort Leeuwendeldseweg 5n 1382 LV Weesp, Nederland |
| Auteur | Derco Sportel <i>Network Consultant</i> ☎ +31 6 1418 1361 ✉ derco.sportel@netyce.com 💻 derco.sportel |
| Docentbegeleider | Kees Uiterwijk <i>Hogeschooldocent, Systeembeheer</i> ☎ +31 88 481 8184 ✉ kees.uitewijk@hu.nl 💻 kees.uitewijk |
| Bedrijfsbegeleider | Ronny Lam <i>Solution Architect</i> ☎ +31 6 3976 6145 ✉ ronny.lam@netyce.com 💻 ronny_lam |
| Versie | 1.0 |

Document

Document geschiedenis

| Versie | Datum | Auteur | Notities |
|--------|--------------|---------------|--|
| 0.1 | 19 juni 2014 | Derco Sportel | Eerste opzet |
| 0.2 | 14 sept 2014 | Derco Sportel | Bijvoegen Bijlage I, II, III, IV en V Aanpassingen Hoofdstuk 2. Invoegen hoofdstuk 3, 4 en 5 Begin hoofdstuk 6 en 7 |
| 0.3 | 24 sept 2014 | Derco Sportel | Bijvoegen Bijlage VI en VII Aanpassingen Hoofdstuk 2, 4, 5, 6 en 7 Invoegen nieuw hoofdstuk 3 en henummering oude hoofdstukken. Invoegen voorwoord, management samenvatting, hoofdstuk 8, 9 en 10 |
| 0.4 | 28 sept 2014 | Derco Sportel | Tekstuele aanpassingen |
| 0.5 | 29 sept 2014 | Derco Sportel | Tekstuele aanpassingen |
| 0.6 | 5 okt 2014 | Derco Sportel | Opmaak en spellingscontroles Kleine tekstuele aanpassingen |
| 1.0 | 12 okt 2014 | Derco Sportel | Definitieve versie |

Distributie

| Name | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 1.0 |
|---------------------|-----|-----|-----|-----|-----|-----|-----|
| Derco Sportel | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Eric Yspeert | | | | | ✓ | | ✓ |
| Ronny Lam | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Geertrui Burgers | | | | | ✓ | | |
| Kees Uiterwijk (HU) | ✓ | ✓ | | | | ✓ | ✓ |
| Gerald Ovink (HU) | | | | | | | ✓ |
| J de Doelder (CvT) | | | | | | | ✓ |

Voorwoord

Voor u ligt de scriptie als resultaat van mijn afstudeerproject. In de periode van maart 2014 tot en met oktober 2014 heb ik bij NetYCE B.V. een onderzoek uitgevoerd over de invoering van IPv6 in de applicatie NetYCE en naar aanleiding van de onderzoeksresultaten een functioneel en technisch ontwerp gemaakt. Met dit eindresultaat wil ik mijn studie Deeltijd Informatica aan de Hogeschool Utrecht afronden.

Deze afstudeeropdracht was niet mogelijk geweest zonder een aantal personen. Allereerst wil ik Wim Gerrits en Eric Yspeert bedanken voor de gelegenheid deze opdracht uit te voeren. Daarnaast wil ik Ronny Lam en Kees Uiterwijk bedanken voor hun hulp en begeleiding gedurende de periode. Voor het doorlezen en corrigeren van deze scriptie wil ik Geertrui Burgers bedanken. Ten slotte wil ik mijn vriendin bedanken voor haar steun en begrip.

Derco Sportel
De Bilt, Oktober 2014

Management samenvatting

Dit document beschrijft het verloop van het afstudeerproject *IPv6 ondersteuning in de NetYCE applicatie* bij de NetYCE B.V. De projectopdracht was de volgende:

Realiseer IPv6 ondersteuning in netYCE waarmee het mogelijk wordt om netwerken van routers, switches en firewalls geautomatiseerd te configureren met een IPv6 of dual stack configuratie.

Dit project is uitgevoerd in de periode maart 2014 tot en met september 2014. Hierbij is de volledige ondersteuning nog niet gerealiseerd echter deze wordt op dit moment ontwikkeld.

Het project is verdeeld in 5 fases:

- Oriëntatiefase: Hierin is het projectvoorstel gemaakt.
- Analysefase: Hierin is het onderzoek gedaan wat de benodigheden zijn om de opdracht te realiseren.
- Architectuurfase: Hierin is een functionele beschrijving gemaakt voor de ondersteuning van IPv6 in de applicatie.
- Ontwerpfase: Hierin is een technische beschrijving gemaakt om de ondersteuning van IPv6 te realiseren in de applicatie.
- Implementatiefase

Dit document is opgedeeld in 11 hoofdstukken en 7 bijlagen. De eerste 3 hoofdstukken hebben betrekking op het bedrijf NetYCE en de applicatie NetYCE. Vervolgens wordt elke fase in een eigen hoofdstuk beschreven. De laatste 3 hoofdstukken hebben betrekking op de evaluatie van het project en de bronvermeldingen.

Inhoudsopgave

| | |
|---|-----|
| Colofon..... | ii |
| Document..... | iii |
| Document geschiedenis..... | iii |
| Distributie..... | iii |
| Voorwoord..... | iv |
| Management samenvatting..... | v |
| | |
| 1 Inleiding..... | 1 |
| | |
| 2 De organisatie..... | 2 |
| 2.1 NetYCE B.V..... | 2 |
| 2.2 De organisatie structuur..... | 2 |
| 2.3 Plaats van de student in de organisatie..... | 2 |
| | |
| 3 De NetYCE applicatie..... | 3 |
| 3.1 Algemene beschrijving..... | 3 |
| 3.2 Design fase..... | 3 |
| 3.3 Build fase..... | 4 |
| 3.4 Operate fase..... | 5 |
| 3.5 Resultaat van het gebruik van NetYCE..... | 5 |
| | |
| 4 De afstudeeropdracht..... | 6 |
| 4.1 Probleemstelling..... | 6 |
| 4.2 Opdracht omschrijving..... | 6 |
| 4.3 Op te leveren producten..... | 6 |
| 4.4 Afbakening..... | 6 |
| 4.4.1 Binnen de scope van het project..... | 6 |
| 4.4.2 Buiten de scope van het project..... | 6 |
| 4.5 Relatie met andere projecten..... | 7 |
| 4.6 Planning..... | 7 |
| | |
| 5 Analysefase..... | 8 |
| 5.1 Inleiding..... | 8 |
| 5.2 Marktontwikkeling..... | 8 |
| 5.3 Huidige architectuur van de netYCE applicatie..... | 8 |
| 5.4 Huidige functionaliteit van de netYCE applicatie..... | 8 |
| 5.4.1 Algemene functionaliteit..... | 8 |
| 5.4.2 IPv4 specifieke functionaliteit..... | 9 |
| 5.5 Gewenste situatie..... | 9 |
| 5.6 Conclusie..... | 9 |
| | |
| 6 Architectuurfase..... | 10 |
| 6.1 Inleiding..... | 10 |
| 6.2 IPv4 vs IPv6..... | 10 |
| 6.3 IPv6 adresnotatie en opbouw..... | 10 |
| 6.4 NetYCE Design fase..... | 11 |
| 6.4.1 IPv6 nummerplan opstellen..... | 11 |
| 6.4.2 Toegestane subnet en prefix grootte..... | 13 |
| 6.4.3 Gebruik van offset waarden..... | 13 |

| | |
|---|-----------------------|
| 6.5 NetYCE Build fase..... | 13 |
| 6.5.1 IPv6 prefix en adres validatie..... | 13 |
| 6.5.2 Integratie met externe systemen..... | 14 |
| 6.5.3 Overige aanpassingen..... | 15 |
| 6.6 NetYCE Operate fase..... | 15 |
| 6.7 Applicatie aanpassingen..... | 16 |
| 7 Ontwerpfase..... | 17 |
| 7.1 Inleiding..... | 17 |
| 7.2 Use cases..... | 17 |
| 7.3 Service task aanpassingen..... | 18 |
| 7.4 Database aanpassing..... | 19 |
| 8 Implementatiefase..... | 20 |
| 8.1 Inleiding..... | 20 |
| 8.2 Fasering..... | 20 |
| 8.3 Git IPv6 branch..... | 20 |
| 8.4 Proof of Concept fase..... | 20 |
| 8.4.1 Database aanpassing..... | 20 |
| 8.4.2 IPv6 communicatie naar de netwerkkapapparaat..... | 21 |
| 8.4.3 Frontend aanpassing..... | 21 |
| 8.5 Volledige implementatiefase..... | 21 |
| 8.5.1 Frontend aanpassingen..... | 21 |
| 8.5.2 Backend aanpassingen..... | 21 |
| 8.6 Proof of Concept opstelling..... | 22 |
| 8.7 Documentatie..... | 22 |
| 9 Evaluatie..... | 23 |
| 9.1 Opdracht..... | 23 |
| 9.2 Persoonlijk..... | 23 |
| 10 Conclusies en aanbevelingen..... | 24 |
| 11 Bronvermelding en literatuurlijst..... | 25 |
| 11.1 Bronnen..... | 25 |
| 11.2 Referenties..... | 26 |
| Bijlagen | |
| Bijlage I: | Het afstudeervoorstel |
| Bijlage II: | Analyse rapport |
| Bijlage III: | Plan van Aanpak |
| Bijlage IV: | Afstudeercontract |
| Bijlage V: | Functioneel ontwerp |
| Bijlage VI: | Technisch ontwerp |
| Bijlage VII: | Implementatievoorstel |



1 Inleiding

Sinds in 1993 het Internet opengesteld is voor bedrijven en particulieren is het aantal gebruikers explosief toegenomen. Al snel is onderkend dat het in 1981 ontworpen communicatieprotocol IP^[RFC781] onvoldoende schaalbaar bleek en de hoeveelheid adressen beperkt was. In december 1995 is met de publicatie van [RFC1883] IP versie 6 geïntroduceerd. Door adressen van 128 bits te gebruiken, in plaats van de in IP versie 4 gebruikte 32 bits, is voldoende ruimte gecreëerd om het Internet weer te kunnen laten groeien.

Hoewel het lijkt alsof IP versie 5 (IPv5) is overgeslagen, heeft deze wel bestaan. Het Internet Stream Protocol versie 2 (ST2 [RFC1819]) was een op IPv4 gebaseerd experimenteel protocol, wat bestaan heeft als IPv5. Dit protocol is nooit in productienetwerken gebruikt, maar veel van de concepten van ST2 zijn gebruikt in andere technieken.

Op 6 juni 2012 is met de World IPv6 Launch day, IPv6 gelanceerd als opvolger van IPv4. Voorafgaand aan 6 juni 2012 zijn er veel voorbereidende werkzaamheden geweest, waaronder het publiek beschikbaar maken van IPv6 DNS in 2004 en de World IPv6 day op 8 juni 2011. Op de World IPv6 Launch day hebben verschillende Internet Service Providers en Hosting providers IPv6 geconfigureerd in hun netwerk en dit niet meer uitgezet.

De invoer van IPv6 in netwerken begint inmiddels meer en meer vorm te krijgen. Naast ISP's en hosting providers wordt IPv6 nu ook ingevoerd bij bedrijfsnetwerken. Om het beheer van die netwerken mogelijk te maken is het belangrijk dat Netwerk Management Systemen ook IPv6 functionaliteit gaan bieden aan de gebruiker. De NetYCE applicatie, welke in het NMS spectrum valt, biedt deze functionaliteit nog niet. In de periode maart 2014 tot en met september 2014 is de benodigde functionaliteit beschreven en is begonnen met de implementatie hiervan.



2 De organisatie

2.1 NetYCE B.V.

NetYCE B.V. is een klein bedrijf dat opgericht is in 2011 en zich specialiseert in het geautomatiseerd doorvoeren van wijzigingen voor middelgrote en grote netwerken van routers, switches en in beperkte mate firewalls. Eindapparatuur, zoals servers, werkstations, printers etc. worden buiten beschouwing gelaten.

Het product dat hiervoor gebruikt wordt heet netYCE. Dit product is in 2001 ontwikkeld voor de Rabobank ICT groep als tool om een grootschalige migratie te realiseren. Om de continuïteit te waarborgen heeft de Rabobank de auteursrechten overgedragen en geholpen NetYCE B.V. op te richten.

Naast netYCE wordt op dit moment een nieuw product ontwikkeld, genaamd labYCE. Dit product richt zich voornamelijk op het configuratiebeheer en de gebruikersplanning van test- en acceptatie netwerken.

2.2 De organisatie structuur

NetYCE B.V. werkt volgens een platte organisatiestructuur. In totaal zijn er 9 werknemers en wordt er op projectbasis af en toe een freelancer ingehuurd. De directie bestaat uit 1 algemeen directeur en 1 technisch directeur. De software developers rapporteren aan de technisch directeur. De rest van de medewerkers rapporteert aan de algemeen directeur.

2.3 Plaats van de student in de organisatie

Als Network Consultant ben ik binnen NetYCE B.V. verantwoordelijk voor de implementatie en het gebruik van netYCE bij de klanten. Tevens maak ik de gebruikershandleidingen en test ik de software.

Bij het implementeren van de software bij de klanten komen er soms verzoeken en vragen over de software en nieuwe of aan te passen functionaliteiten. Deze vertaal ik dan naar use cases richting de software developers.



3 De NetYCE applicatie

3.1 Algemene beschrijving

NetYCE is een applicatie die gebruikt kan worden om grote router en switch netwerken te ontwerpen en te configureren. Hierbij is het doel een hoge mate van standaardisatie te realiseren in het netwerk.

Netwerk ontwerpers kunnen met NetYCE door een abstract model te maken een beoogd netwerkontwerp vastleggen. Daarbij kunnen zij vastleggen welke functies het netwerk moet bieden.

Dit abstracte model van een netwerk wordt opgeslagen in een database. Door configuratie templates te combineren met de gegevens uit het abstracte model wordt de configuratie voor een netwerkkapparaat gegenereerd. Hiermee kan het netwerkkapparaat vervolgens geconfigureerd worden.

Een netwerk wordt in 3 fases gebouwd in NetYCE. De Design fase, de Build fase en de Operate fase.

3.2 Design fase

In de Design fase wordt een netwerk functioneel ontworpen. De ontwerpers van een netwerk creëren hier een IP-indeling voor adresblokken (een IP-plan), locatie types, node types en netwerkdiensten. Bij locatie types moet worden gedacht aan hoofdkantoren, bijkantoren, datacentra etc. Node types zijn bijvoorbeeld core routers, distributie routers, aggregatie switches en access switches. Bij de netwerkdiensten kan gedacht worden aan VoIP VLAN's, Quality of Service (QoS), WiFi functionaliteit, VPN's en

Site types

| Client type | Site type | Caption |
|-------------|-------------|-------------|
| HU | Hoofdgebouw | Hoofdgebouw |
| HU | Bijgebouw | Bijgebouw |

Afbeelding 1: Een voorbeeld van verschillende locatie types

Node types

| Client type | Node type | Template |
|-------------|------------|--------------|
| HU | | |
| HU | Access | IanAccess |
| HU | Aggregatie | IanAggregate |
| HU | Core | IanCore |

Afbeelding 2: Een voorbeeld van verschillende node types

Bij het functionele ontwerp wordt nog geen rekening gehouden met routeringsprotocollen of hardware en fabrikant keuze. Hier wordt wel vastgelegd hoe routers en switches met elkaar verbonden moeten worden en welke VLAN's standaard op een switch aanwezig moeten zijn.



3.3 Build fase

In de Build fase wordt het functionele ontwerp gebruikt om een technisch ontwerp voor een specifieke klant en locatie te maken. Hier wordt de keuze gemaakt voor bepaalde fabrikanten en hardware modellen. Hier worden ook de gebruikte IP-adressen en VLAN's aangemaakt en toegewezen aan specifieke interfaces van de routers en switches.

Met dit technisch ontwerp wordt een afspiegeling van het beoogde netwerk in de database gebouwd. Benodigde wijzigingen op het netwerk worden eerst in de Build fase aangepast en vervolgens vanuit de Operate fase op de routers en switches geconfigureerd.

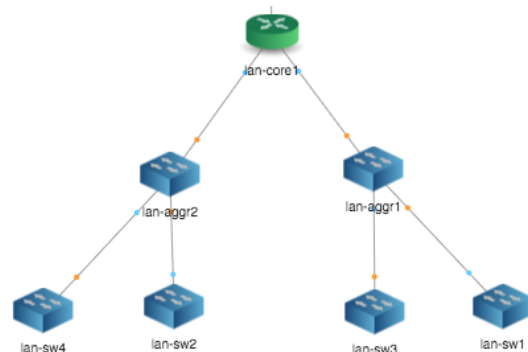
Binnen de Build fase worden allereerst 1 of meerdere klanten aangemaakt. Aan elke klant worden IP-adresblokken gekoppeld. Voor elke klant worden vervolgens locaties aangemaakt. Op de locaties worden vervolgens de routers en de switches aangemaakt.

Een klant kan bijvoorbeeld een scholengemeenschap zijn, waarbij de verschillende schoolgebouwen de locaties zijn.

6000

| Site | Type | Address |
|------|-------------|---------------|
| 6001 | Hoofdgebouw | Nijenoord 1 |
| 6500 | Bijgebouw | Oudenoord 700 |

Afbeelding 3: Een locatie overzicht voor een klant in NetYCE



Afbeelding 4: Een overzicht van de netwerkapparatuur op een locatie

In de Build fase worden ook configuratietemplates gemaakt voor de netwerkapparatuur. Per fabrikant worden deze templates gemaakt. Door middel van boomstructuur van main templates, subtemplates, port templates en het gebruik van variabelen en condities kunnen de templates gemakkelijk hergebruikt worden voor verschillende node types. In de volgende voorbeelden is het gebruik van deze variabelen, subtemplates en condities te zien.

```
! Template voor Cisco 6509
{ciscoBanners}
!
{ciscoL2VLANs}
!
{ciscoL3VLANs}
!
{ciscoInterfaces}
!
{ciscoMangement}
!
{ciscoRouting}
```

Een voorbeeld van een main template.

```
! Cisco Banners
banner login @\n
Welcome to <Node_fqdn>
Authorized users only!
@
```

Een voorbeeld van de sub template ciscoBanners.

```
! Cisco L2 access port
interface <Port_name>
switchport
switchport mode access
switchport access vlan <Vlan_id>
no cdp enable
no lldp enable
|<Port_shut> = 'N'| no shutdown
|<Port_shut> = 'Y'| shutdown
```

Een voorbeeld van een port template, deze wordt aangeroepen vanuit ciscoInterfaces



3.4 Operate fase

Door middel van change en configuratie management kunnen gebruikers in de Operate fase het netwerk configureren zoals dit in de Build fase is ontworpen. Dit geldt voor zowel de initiële uitrol van de netwerkapparatuur als het latere beheer.

Gebruikers van de applicatie kunnen in de Operate fase de volgende taken uitvoeren:

- Configuraties voor netwerkapparatuur genereren en bekijken;
- Bekijken of netwerkapparatuur bereikbaar is door middel van ping;
- Bekijken welke templates door welke netwerkapparatuur in gebruik is;
- Het OS van netwerkapparatuur bijwerken naar een gewenste versie;
- Gestandaardiseerde taken uitvoeren om netwerkapparatuur te configureren;
- Poortwijzigingen doorvoeren;
- Een volledige configuratie naar een netwerkapparaat sturen;
- Een netwerkapparaat opnieuw opstarten;
- Netwerkchanges voorbereiden en uitvoeren;
- Configuraties van een netwerkapparaat vergelijken.

```
line vty 0 4
exec-timeout 15 0
privilege level 15
transport preferred ssh
transport input telnet ssh
transport input ssh
transport output none
!
```

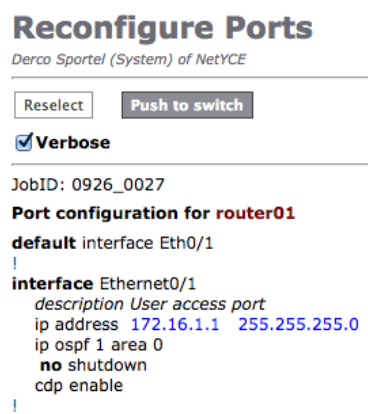
Afbeelding 5: Verschil tussen 2 configuraties

3.5 Resultaat van het gebruik van NetYCE

Door het netwerk in hoge mate te standaardiseren door middel van NetYCE zal het eindresultaat een kostenreductie realiseren. Het aantal storingen ten gevolg van een wijziging op het netwerk zal afnemen doordat er geen typefouten meer gemaakt worden door de engineer die de wijziging doorvoert.

Doordat alle informatie centraal opgeslagen is in NetYCE en de configuratie van de netwerkapparatuur is vastgelegd zijn netwerk- ontwerpers en engineers minder tijd kwijt aan het in stand houden van het netwerk. De tijd die hierdoor overblijft wordt bij de huidige klanten van NetYCE gebruikt om ontwikkeling van nieuwe diensten te realiseren.

Het invoeren van deze nieuw ontwikkelde diensten kan door het gebruik van NetYCE ook sneller plaatsvinden doordat alle informatie bekend is. Wanneer de configuratie eenmaal getest is in NetYCE kan met behulp van een change de configuratie-wijziging snel op alle relevante apparatuur uitgevoerd worden. Configuratie wijzigingen duren gemiddeld tussen de 15 seconden en 2 minuten. Hierdoor is het mogelijk om in een paar uur tijd duizenden apparaten te configureren.



Afbeelding 6: Herconfiguratie van een poort



4 De afstudeeropdracht

4.1 Probleemstelling

De huidige versie van netYCE heeft geen IPv6 ondersteuning. Vanwege de verschuiving van IPv4 naar IPv6 in netwerken is het belangrijk dat netYCE dit gaat ondersteunen.

4.2 Opdracht omschrijving

De opdracht is: Realiseer IPv6 ondersteuning in netYCE waarmee het mogelijk wordt om netwerken van routers, switches en firewalls geautomatiseerd te configureren met een IPv6 of dual stack configuratie.

De opdracht bestaat uit de volgende deelopdrachten:

- Onderzoek welke functionaliteit netYCE moet bieden voor IPv6 ondersteuning;
- Onderzoek hoe IPv6 op dit moment in de markt gebruikt wordt;
- Maak naar aanleiding van de uitkomst van het onderzoek een functioneel ontwerp;
- Maak naar aanleiding van het functioneel ontwerp een technisch ontwerp;
- Laat in een Proof of Concept zien, dat het mogelijk is om:
 - Een functioneel netwerkontwerp, zoals beschreven in hoofdstuk 3.2, op basis van een dual stack configuratie te ontwerpen;
 - Een technisch netwerkontwerp, zoals beschreven in hoofdstuk 3.3, van dit functioneel ontwerp te creëren als netwerk;
 - Het fysieke netwerk geautomatiseerd te voorzien van een dual stack configuratie, op basis van het technische ontwerp.

4.3 Op te leveren producten

Het project zal de volgende deelproducten opleveren:

- Een analyse rapportage van de huidige situatie en de gewenste situatie;
- Een functioneel ontwerp benodigd om tot de gewenste situatie te komen;
- Een technisch ontwerp;
- Een Proof of Concept opstelling;
- Een implementatie voorstel.

4.4 Afbakening

4.4.1 Binnen de scope van het project

De scope van dit project beperkt zich tot alle in de projectopdracht en op te leveren producten genoemde zaken.

4.4.2 Buiten de scope van het project

Om het project overzichtelijk te houden, zullen alle niet-gerelateerde zaken buiten beschouwing worden gelaten. Uitgezonderd het implementatievoorstel, zal de implementatie expliciet buiten de scope van het project vallen.



4.5 Relatie met andere projecten

Er zijn op dit moment geen andere projecten, die gerelateerd zijn aan de implementatie van IPv6 in netYCE.

4.6 Planning

Het project is in 4 fases gerealiseerd. Een 5^e fase (de implementatiefase) valt gedeeltelijk buiten de scope van dit project.

In fase 1 is een oriëntatie over het onderwerp en de grove uitlijning van het project gemaakt. Deze fase is afgerond in maart 2014.

In fase 2 is een analyse van de behoeften, het onderwerp en de huidige situatie gemaakt. Als afronding van deze fase is een analyse rapportage ^[Bijlage II] het Plan van Aanpak ^[Bijlage III] opgeleverd. Deze fase is eind juli 2014 afgerond.

In fase 3 is de architectuur beschreven. Als eindresultaat van deze fase is een functioneel ontwerp gemaakt ^[Bijlage V]. Deze fase is half augustus 2014 afgerond.

In fase 4 is het functioneel ontwerp uit fase 3 gebruikt om een technisch ontwerp ^[Bijlage VI] te maken. Deze fase is eind september 2014 afgerond.

De implementatie van het technische ontwerp valt buiten de scope van het project. Als onderdeel van het project is wel een implementatievoorstel ^[Bijlage VII] opgeleverd. Ook is een Proof of Concept opstelling gemaakt.

Naast deze fases wordt als onderdeel van de evaluatie de afstudeerzitting voorbereid. Dit zal gebeuren na het opleveren van deze scriptie.



5 Analysefase

5.1 Inleiding

Tijdens de analysefase en het schrijven van het analyserapport ^[Bijlage II] is gekeken naar de huidige marktontwikkeling, de huidige architectuur van de netYCE applicatie en de huidige functionaliteit van de netYCE applicatie.

Door middel van een literatuurstudie en het analyseren van de gewenste functionaliteiten binnen netYCE is een gewenste situatie beschreven. Het analyserapport is opgenomen als Bijlage II van dit document.

Aangezien de vraag voor NetYCE voornamelijk was hoe IPv6 te ondersteunen en niet of, heeft de analyse zich hier voornamelijk op gefocust.

5.2 Marktontwikkeling

In de periode na 6 juni 2012 (world IPv6 launch-day) is het aandeel IPv6 verkeer op internet toegenomen van 0,65% op 6-6-2012 naar 4,56% op 28-9-2014^[1] en de voorspelling van Internet Society uit januari 2014 is dat dit eind 2014 10% zal zijn^[2]. Bij gesprekken met (toekomstige) klanten wordt inmiddels vaker gevraagd naar IPv6 ondersteuning. Vooral bij de zakelijke service- en hosting providers wordt IPv6 inmiddels standaard geboden.

5.3 Huidige architectuur van de netYCE applicatie

Op dit moment wordt de netYCE applicatie geïnstalleerd op Red Hat Enterprise Linux 6.5 of de open source variant CentOS 6.5. Er wordt gebruik gemaakt van de volgende componenten:

- MySQL (5.6) of MariaDB (5.5 of 10.x) database;
- Apache (2.2.x);
- Een eigen gecompileerde Perl versie;
- OpenSSH;
- Telnet.

Alle hierboven genoemde software ondersteunt allemaal het gebruik van IPv6. IPv6 moet mogelijk nog wel geconfigureerd worden op de interfaces van het besturingssysteem.

5.4 Huidige functionaliteit van de netYCE applicatie

5.4.1 Algemene functionaliteit

De huidige netYCE applicatie wordt gebruikt om netwerken van routers, switches en firewalls geautomatiseerd te configureren. Hierbij wordt gebruik gemaakt van het Design, Build, Operate proces zoals beschreven in hoofdstuk 3 van dit document.



5.4.2 IPv4 specifieke functionaliteit

In de huidige situatie kan er door middel van IP-plannen gebruik gemaakt worden van standaard indelingen van een IP-reeks. Hierbij wordt een supernet ingedeeld in kleinere subnetten. Deze subnetten worden per interface bij gehouden in de netYCE applicatie, waardoor er een actueel overzicht bestaat van de IP informatie. Wanneer er geen gebruik gemaakt wordt van een IP-indeling dan is het ook mogelijk om subnetten los toe te voegen zonder vooraf ingedeeld plan.

5.5 Gewenste situatie

Voor NetYCE is het wenselijk dat door de gehele Design-Build-Operate cyclus gebruik van IPv6 gemaakt kan worden. De huidige IPv4 functionaliteit moet feitelijk overgenomen worden voor IPv6. Dit houdt in dat op alle vlakken in de software veranderingen doorgevoerd moeten gaan worden.

NetYCE maakt voornamelijk gebruik van een vooraf gedefinieerde architectuur en IP-indeling. Met name de IP-indeling, welke nu al binnen NetYCE gebruikt wordt, wordt aangeraden te gebruiken voor IPv6. Voorbeelden van dit soort indelingen staan duidelijk beschreven in [SURF-IPv6], [Cisco-IPv6] en [RFC5375]. De detailuitwerking van de gewenste situatie staat beschreven in Bijlage II hoofdstuk 3.

Omdat bepaalde functionaliteit belangrijker is dan andere, is een MoSCoW analyse gemaakt om de prioriteit van de functionaliteit te bepalen. Deze analyse is uitgewerkt in Bijlage II hoofdstuk 4.

5.6 Conclusie

Om IPv6 snel en afdoende te kunnen gebruiken in NetYCE is het belangrijk dat de Must have's nog voor het einde van 2014 ingevoerd zijn.

Doordat de functionaliteit van de applicatie uitgebreid wordt, maar de huidige functionaliteit niet aangepast wordt, zijn de risico's erg laag. De ontwikkeling kan parallel aan het huidige ontwikkel traject van NetYCE.



6 Architectuurfase

6.1 Inleiding

In de architectuurfase van dit project is gekeken naar de huidige architectuur en functionaliteit van de netYCE applicatie in relatie tot de gewenste situatie. Naar aanleiding hiervan is een literatuurstudie gedaan om te kijken welke IPv6 functionaliteit ontwikkeld moet worden voor NetYCE. Het eindresultaat hiervan is verwerkt tot een functioneel ontwerp dat is bijgevoegd in Bijlage V.

6.2 IPv4 vs IPv6

IPv6 is niet backwards compatible met IPv4. Door de opbouw van de adressen en het verschil in werking tussen deze 2 protocollen is communicatie tussen een IPv4 host en een IPv6 host niet mogelijk. Wanneer een host, alleen maar gebruik kan maken van IPv6 en toch met een IPv4 host moet communiceren is dit alleen mogelijk door middel van een NAT64 gateway.

IPv4 adressen bestaan uit 32 bits. Hierdoor zijn er theoretisch ongeveer 4,3 miljard adressen mogelijk. Dit aantal is theoretisch, omdat er grote blokken gereserveerd zijn voor speciaal gebruik of experimentele doeleinden. Hierdoor blijven er iets meer dan 3,7 miljard adressen over. Een schatting van de Cisco uit 2013 geeft aan dat er ongeveer 8,7 miljard computers, smartphones, tablets en andere netwerkkapparatuur aangesloten is op internet in 2012^[1]. Dit is meer dan het aantal beschikbare adressen. Door het gebruik van NAT (Network Address Translation) is het nu mogelijk om al deze apparaten toch te verbinden met internet.

IPv4 adressen worden weergegeven in 4 numerieke blokken met punten er tussen. Elke 8 bits van een adres is zo'n numeriek blok. Een voorbeeld is 192.0.2.2

IPv6 adressen bestaan uit 128 bits, waar mee ongeveer $3,4 * 10^{38}$ adressen mogelijk zijn. Dit is voldoende om de komende jaren te groeien.

6.3 IPv6 adresnotatie en opbouw

IPv6 adressen worden weergegeven in 8 hexadecimale blokken van 16 bits. Een voorbeeld van een IPv6 adres is 2001:0db8:fc90:0001:0000:0000:0000:199a. Om de leesbaarheid te vergroten worden voorloop nullen in 16 bits velden niet geschreven. Ook mogen repeterende blokken met nullen vervangen worden voor 2x een dubbele punt. Dit mag echter maar één keer voorkomen in een adresblok. Een voorbeeld hiervan is het eerder genoemde adres 2001:0db8:fc90:0001:0000:0000:0000:199a wat geschreven wordt als 2001:db8:fc90:1::199a.

[RFC5952] beschrijft in detail hoe IPv6 adressen weergegeven moeten worden. De notatie van deze adressen moet op de juiste manier weergegeven worden binnen netYCE. Binnen de huidige implementatie van de software wordt gebruik gemaakt van onder andere Perl en MySQL of MariaDB. Zowel Perl als de gebruikte database software kennen functies die IPv6 adressen op de juiste manier weergeven. Hier hoeft verder geen extra functionaliteit voor ontwikkeld te worden.



Wanneer een klant gebruik wil maken van IPv6 krijgt deze per locatie een adresreeks toegewezen ^[RIPE589]. Dit wordt een prefix genoemd. In Europa krijgt elke locatie in beginsel een /48 prefix toegewezen. Dit houdt in dat de eerste 48 bits door de provider toegewezen worden. De overige 80 bits kunnen gebruikt worden door de klant. Een voorbeeld van een prefix is 2001:db8:fc90::/48. Deze prefix wordt standaard ingedeeld in /64 subnetten. De 16 bits die vervolgens beschikbaar zijn worden gebruikt om locatie en functiescheiding te realiseren. Een voorbeeld hiervan wordt gegeven in 6.4.1

6.4 NetYCE Design fase

6.4.1 IPv6 nummerplan opstellen

Binnen het ontwerp in NetYCE wordt sterk geleund op een vooraf gedefinieerd netwerkontwerp. Onderdeel van dit netwerkontwerp is onder andere het IP-nummerplan. Op voorhand wordt aangegeven hoe de toegekende IP-blokken ingedeeld worden voor een klant, hoe de VLAN's ingedeeld worden en wat de adressen zijn voor de netwerkapparatuur.

Voor IPv4 wordt deze methodiek slechts beperkt gebruikt. Bij gebruik van private adressen is de methode te gebruiken, wanneer er echter publieke adressen gebruikt worden is het vooraf indelen niet aan te raden. De kans op verspilling van IPv4 adressen is te groot. Voor IPv6 is het vooraf opstellen van een indeling gebruikelijker. [Surf-IPv6] en [Cisco-IPv6] beschrijven methodieken voor het opstellen van een IPv6 nummerplan.

Bij het opstellen van een IPv6 nummerplan kan worden gekeken naar een onderverdeling in regio's, locaties en gebruikerstypen. De de-facto standaard voor gebruikers-subnetten is een /64 subnet. Dit komt doordat diverse automatische adresconfiguratie methoden zoals Stateless Address Autoconfiguration (SLAAC, beschreven in [RFC4862]) en Secure Neighbor Discovery (SEND, beschreven in [RFC3971]) alleen werken met /64 subnetten. Wanneer een klant een /48 prefix toegewezen krijgt betekent dit dat er 16 bits beschikbaar zijn om een indeling te maken. De opbouw van een IPv6 adres ziet er dan als volgt uit:

| Toegekende prefix (48 bits) | Subnet (16 bits) | Host adres (64 bits) |
|-----------------------------|------------------|----------------------|
| 2001:db8:fc90: | 1234: | ba8d:12ff:fe3f:bc00 |

Tabel 1: Opbouw van een IPv6 adres

Het is echter mogelijk om af te wijken van de /64 subnetten. Dit is bijvoorbeeld zinvol voor adressen die bestemd zijn voor Loopback interfaces of point-to-point verbindingen tussen routers. In dat geval kan bijvoorbeeld gekozen worden om het eerste /64 subnet verder te verdelen in /128 loopback subnetten en /127 point-to-point subnetten.

De subnet verdeling kan als volgt aangepakt worden:

- De hoeveelheid locaties kiezen;
- De hoeveelheid gebruikerstypes kiezen;
- Schaalbaarheid kiezen.

Bij het kiezen van de hoeveelheid locaties wordt in eerste instantie gekeken naar de huidige hoeveelheid locaties. Zijn dit er bijvoorbeeld 10, dan zijn er 4 bits (2^4) nodig voor de locatie. Hetzelfde moet gedaan worden voor de gebruikers types. Wanneer dit bijvoorbeeld ook 10 zijn, zijn nogmaals 4 bits nodig. Er zijn dan 8 bits over voor de subnetten. Dit betekent dat er 256 subnetten per type beschikbaar zijn op alle locaties.



Omdat het omnummeren van IP-adressen vaak een behoorlijke inspanning met zich meebrengt is het zinvol om op voorhand te kijken naar de schaalbaarheid van het nummerplan. Wanneer er op dit moment 10 verschillende types subnetten zijn maar dit in de toekomst bijvoorbeeld kan verdubbelen, is het zinvol om hier direct rekening mee te houden en direct 5 bits voor de gebruikerstypen te reserveren. Hiermee kan gegroeid worden naar 32 typen. Het nadeel is echter wel dat de hoeveelheid beschikbare bits afneemt en er dan nog maar 128 subnetten per type beschikbaar zijn.

Bovenstaand model is niet flexibel. Om flexibeler om te kunnen gaan met de hoeveelheid subnetten per gebruikerstype is voor een iets andere methode gekozen. In plaats van een vaste hoeveelheid bits voor gebruikerstypen en locatie, is gekozen om gebruik te maken van een minimale en maximale offset vanaf de toegekende prefix. Hierdoor wordt het mogelijk om bijvoorbeeld 1 management subnet te kiezen, maar ook 300 data subnetten, 150 wifi subnetten en 100 VoIP subnetten. Een subnet indeling van een /48 prefix kan er dan als volgt uitzien:

| Gebruikerstype | Minimale offset | Maximale offset | #subnetten |
|---------------------|-----------------|-----------------|------------|
| Loopback /128 | :: | ::ffff | 65535 |
| Point-to-point /127 | ::11:0 | ::12:fffe | 65535 |
| Management /64 | 0:0:0:1:: | 0:0:0:5:: | 5 |
| Data /64 | 0:0:0:11:: | 0:0:0:13d:: | 300 |
| Wifi /64 | 0:0:0:200:: | 0:0:0:296:: | 150 |
| VoIP /64 | 0:0:0:400:: | 0:0:0:464:: | 100 |

Tabel 2: Voorbeeld indeling van een /48 prefix

De tabel laat zien dat er gaten zitten tussen de verschillende subnetten. Hierdoor is latere groei te realiseren. De methode is flexibeler dan het gebruik van een vastgestelde hoeveelheid bits voor het type en daarmee een vaste hoeveelheid subnetten.

Om een locatie-verdeling te maken wordt gebruik gemaakt van een aantal bits. In bovenstaand voorbeeld kan bijvoorbeeld gekozen worden voor 5 bits ten behoeve van een locatie verdeling. Om dit te realiseren wordt gebruik gemaakt van een site masker dat opgeslagen wordt als 128 bits waarde. Op deze manier is dit masker te gebruiken in de berekening van IPv6 adressen. Het binaire masker wordt op de volgende manier gebruikt:

| 48 bits Prefix | 16 bits voor verdeling van locatie en subnetten | | | | | | | | | | | | | | 64 host bits | |
|----------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|--------------|---------|
| 0:0:0 | L | L | L | L | L | S | S | S | S | S | S | S | S | S | S | 0:0:0:0 |

Tabel 3: Voorbeeld verdeling van locatie en subnet bits in een /48 prefix

Het gebruik van 5 bits voor een locatie verdeling betekent in bovenstaand voorbeeld dat er nog 11 bits beschikbaar zijn voor een subnet verdeling. In het voorbeeld van tabel 2 kan er dus nog uitgebreid worden naar een maximale offset van 0:0:0:3ff::.

In bovenstaand voorbeeld wordt uitgegaan van een /48 prefix dat aan een klant toegewezen wordt. Wanneer dit een kleinere of grotere prefix is zijn er natuurlijk meer of minder bits voor een locatie/subnet verdeling.



6.4.2 Toegestane subnet en prefix grootte

Omdat de de-facto standaard uitgaat van /64 subnetten is dit de maximale grootte van een subnet. Alles groter dan /64 (meer adressen per subnet) wordt gezien als een bad practice implementatie van IPv6 in een netwerk.

Subnetten kleiner dan /64 (minder adressen per subnet) zijn binnen IPv6 wel toegestaan. [RFC6164] is hier een voorbeeld van. Dit RFC behandelt het gebruik van /127 subnetten op point-to-point verbindingen tussen routers. Ook worden /128 subnetten gebruikt voor Loopback interfaces.

Om deze redenen zullen de subnetten binnen NetYCE niet groter zijn dan /64. Alle waarden tussen /64 en /128 zijn echter toegestaan.

Omdat NetYCE geïmplementeerd wordt bij zowel grote als kleinere bedrijven is het noodzakelijk om de prefix grootte hier ook op af te stemmen. [RIPE589] specificeert dat de kleinst mogelijke prefix die uitgedeeld mag worden een /64 is. [RIPE589] specificeert ook dat organisaties ten minste een /32 toegekend krijgen. Hierbij is een organisatie gedefinieerd als een Local Internet Registry. Omdat de /32 prefix ook precies op een 16 bits scheiding ligt wordt deze waarde gekozen als grootste mogelijke prefix waarde.

6.4.3 Gebruik van offset waarden

In de Design fase van NetYCE wordt gewerkt met offsets van IPv6 adressen.

Een voorbeeld hiervan is bijvoorbeeld een /127 point-to-point subnet. Binnen een /127 subnet zijn er 2 adressen beschikbaar. In de designfase wordt in dit geval een referentie gemaakt naar het 1^e adres welke een offset :: krijgt en het 2^e adres welke een offset van ::1 krijgt.

Voor subnet indeling wordt ook gebruik gemaakt van offsets. In 6.4.1 tabel 2 wordt hier een voorbeeld van gegeven. Voor een /127 point-to-point subnet uit voorbeeld 6.4.1 is de minimale offset ::11:0 en de maximale offset is ::12:fffe. Dit betekent dat het eerste beschikbare subnet moet beginnen met ::11:0

Wanneer in de Build fase de echte prefixen ingevoerd worden betekent dit dat een klant met de prefix 2001:db8:fc90::/48 het eerste point-to-point subnet 2001:db8:fc90::11:0/127 is. Het 1^e geldige adres is dan 2001:db8:fc90::11:0/127 en het 2^e is 2001:db8:fc90::11:1/127.

Het 3^e management subnet in ditzelfde voorbeeld is 2001:db8:fc90:3::/64

6.5 NetYCE Build fase

6.5.1 IPv6 prefix en adres validatie

In de Build fase worden de werkelijke prefixen ingevoerd en gebruikt. Hierbij is het belangrijk dat de ingevoerde adressen en prefixen geldig zijn. Op dit moment is er voor gekozen om alleen Unicast adressen te ondersteunen binnen NetYCE.

Link-Local adressen worden automatisch geconfigureerd op een node, ongeacht wat voor type node het betreft. Omdat Link-Local adressen alleen relevant zijn op de lokale interface en verder niet gerouteerd kunnen worden ^[RFC4291-2.5.6] worden deze niet opgeslagen en gebruikt binnen netYCE.

Multicast adressen worden voorlopig niet door netYCE beheerd. Dit betekent echter niet dat de configuraties geen multicast kunnen bevatten. De multicast adressen worden dan opgenomen in de configuratietemplates.



IPv6 kent naast Link-Local 2 typen unicast adressen, de Global Unicast en de Unique Local Unicastaddress (ULA). De Global Unicast adressen zijn globaal uniek en worden gereguleerd uitgegeven. De uitgifte vindt hierbij plaats via een hiërarchisch model. Het IANA beheert de totale hoeveelheid IP-adressen. Hierbij zijn blokken toegewezen aan de Regional Internet Registries (RIR), die op hun beurt de blokken verdelen en toekennen aan de Local Internet Registries (LIR) wat veelal Internet Service providers zijn. Deze adressen worden gerouteerd op internet.

De ULA adressen zijn niet bedoeld om gerouteerd te worden via internet. Deze adressen worden ook niet gereguleerd uitgegeven, al is SixXS^[3] wel een registratieproject begonnen omtrent de ULA adressen. Deze adresblokken kunnen door gebruikers vrij ingedeeld worden. Het gebruik van de ULA adressen is te vergelijken met de [RFC1918] private IPv4 adressen.

Het IANA heeft blokken met IPv6 adressen toegekend voor ULA en Global Unicast gebruik. Voor Global Unicast wordt op dit moment de reeks 2000::/3 gebruikt en voor ULA fc00::/7. Hoewel een adres wat uit de reeks 4000::/3 wel een geldig IPv6 adres kan zijn wordt dit op dit moment en in de nabije toekomst niet toegestaan. Het IANA houdt in [IANA-IPv6-AS] bij welke blokken in gebruik zijn en voor welk doel. In [IANA-IPv6-GUAA] wordt vervolgens bijgehouden welke Global Unicast blokken aan welke RIR toegekend zijn.

Om fouten te voorkomen moet er bij het invoeren van een IPv6 prefix gecontroleerd worden of deze prefix wel gebruikt mag worden voor unicast verkeer.

Wanneer het een globale prefix betreft zou een controle gedaan kunnen worden tegen de [IANA-IPv6-GUAA] administratie. Wanneer een prefix uit een ALLOCATED reeks komt zou deze geldig moeten zijn. Dit betekent echter dat NetYCE regelmatig aangepast moet worden omdat de lijst kan veranderen. Omdat de NetYCE servers meestal opgesteld worden in beheer segmenten van een netwerk, en daarmee geen directe verbinding met internet hebben, is dit niet praktisch. Om deze reden wordt ervoor gekozen een ingevoerde prefix uit de reeks 2000::/3 of fc00::/7 te laten komen. Wanneer in de toekomst de volgende reeks gebruikt gaat worden, naar verwachting 4000::/3, zal dit in de software aangepast moeten worden. De verwachting is dat dit in de nabije toekomst echter nog niet aan de orde is.

6.5.2 Integratie met externe systemen

Hoewel integratie met externe systemen buiten de scope van het project zit is het wel belangrijk om hier aandacht aan te besteden.

In de IPv6 Address Allocation and Assignment policy is bepaald dat een organisatie uitgedeelde prefixen moet registreren^[RIPE589-5.5]. Hiervoor heeft het Ripe NCC een REST API beschikbaar om toegekende IPv6 prefixen te registreren in de Ripe database. De documentatie voor deze API is beschikbaar via de Ripe NCC website. Omdat NetYCE voor een klant precies weet welke IPv6 prefixen waar uitgedeeld zijn is het denkbaar dat in de toekomst een integratie ontwikkeld wordt om vanuit NetYCE geautomatiseerd de Ripe database bij te werken. Het eerder gestelde probleem met directe internet toegang van de NetYCE servers zal dan wel ondervangen moeten worden.

Omdat IPv6 adressen een stuk langer zijn dan de IPv4 adressen en moeilijker te onthouden zijn zal het belang van een IPAM (IP Address Management) tool toenemen. Integratie met deze IPAM tooling vanuit NetYCE zal dan ook belangrijk zijn. Omdat er veel verschillende IPAM pakketten zijn zal deze integratie ontwikkeld moeten worden per pakket wanneer er behoefte is bij de klant.



6.5.3 Overige aanpassingen

Voor de Build fase moeten er binnen NetYCE een aantal aanpassingen gedaan worden om IPv6 op de juiste manier te kunnen ondersteunen. Een groot deel van deze aanpassingen is gelijk aan de huidige IPv4 functionaliteit. In Bijlage V sectie 3.3 staat de Build fase verder uitgewerkt.

6.6 NetYCE Operate fase

In de Operate fase van NetYCE wordt de netwerkkapppatuur geconfigureerd. Om de apparatuur te configureren wordt door middel van SSH of Telnet ingelogd en de configuratie naar het apparaat gestuurd. Om te verbinden naar een netwerkkappparaat wordt op dit moment gebruik gemaakt van het management IPv4 adres van het apparaat.

Zodra een netwerkkappparaat ook een management IPv6 adres heeft moet er een keuze gemaakt worden, hoe naar dit apparaat te verbinden. Voor deze keuze zijn er 2 opties:

- Gebruik maken van DNS en verbinden op naam in plaats van IP adres;
- Verbinden op IPv6 adres, wanneer dit niet lukt, proberen te verbinden op IPv4 adres.

Wanneer er gebruik gemaakt wordt van DNS moet er voor een apparaat een A record en een AAAA record aanwezig zijn in de DNS server. De hostname *router01* heeft bijvoorbeeld A record 192.0.2.1 en AAAA record 2001:db8::1. Door te verbinden naar *router01* zal de DNS server beide records teruggeven en maakt de server een keuze voor of IPv6 of IPv4. Een groot nadeel van deze methode is dat de DNS server wel bijgewerkt moet zijn. Dit houdt in dat er een DNS of IPAM integratie moet zijn, of dat de DNS server handmatig bijgewerkt moet worden. Er is hierdoor een externe afhankelijkheid. Deze zou ondervangen kunnen worden door de NetYCE server ook te voorzien van een DNS server implementatie en hier rechtstreeks met te integreren. Hierdoor kan de implementatie echter nodeloos complex worden.

De andere optie, het verbinden op IPv6 adres en wanneer dat niet werkt op IPv4 adres, is dan een betere optie. Dit kan echter een tijdsvertraging opleveren waardoor het lang kan duren voordat NetYCE inlogt op een apparaat. Deze tijdsproblematiek wordt uitgebreid besproken in [RFC6555].

Om te voorkomen dat het lang duurt voordat er ingelogd is op een apparaat, beschrijft [RFC6555] ook een algoritme om deze tijd te verkorten. In plaats van te verbinden via IPv6 en wanneer dat niet lukt te verbinden via IPv4, wordt verbinding gemaakt op IPv6 en met een zeer korte tijdsvertraging ook op IPv4. Het protocol dat als eerste reageert zal gebruikt worden. De tijdsvertraging tussen IPv6 en IPv4 kan eventueel ook weggelaten worden echter om de voorkeur voor IPv6 enigszins te forceren beschrijft [RFC6555] de tijdsvertraging als methode om dit te bereiken.

Er zijn 3 opties om een verbinding tussen de NetYCE server en de node tot stand te brengen:

1. De node is bereikbaar op zowel IPv4 als IPv6, IPv6 zal gekozen moeten worden;
2. De node is alleen bereikbaar op IPv4, IPv4 zal gekozen moeten worden;
3. De node is alleen bereikbaar op IPv6, IPv6 zal gekozen moeten worden.



De 3 scenario's zijn hieronder uitgewerkt.

| netYCE server | Node |
|--------------------------------|-----------------------|
| 1. =====TCP SYN, IPv6=====> | |
| 2. -----TCP SYN, IPv4-----> | (50 - 300 msec delay) |
| 3. <====TCP SYN+ACK, IPv6=== | |
| 4. <----TCP SYN+ACK, IPv4--- | |
| 5. =====TCP ACK, IPv6=====> | |
| 6. -----TCP ACK, IPv4-----> | |
| 7. -----TCP RST, IPv4-----> | |

Scenario 1: RFC6555 implementatie in netYCE, IPv4 en IPv6 bereikbaar

| netYCE server | Node |
|--------------------------------|-----------------------|
| 1. =====TCP SYN, IPv6=====X | |
| 2. -----TCP SYN, IPv4-----> | (50 - 300 msec delay) |
| 3. <----TCP SYN+ACK, IPv4--- | |
| 4. -----TCP ACK, IPv4-----> | |

Scenario 2: RFC6555 implementatie in netYCE, IPv4 bereikbaar

| netYCE server | Node |
|--------------------------------|-----------------------|
| 1. =====TCP SYN, IPv6=====> | |
| 2. -----TCP SYN, IPv4-----X | (50 - 300 msec delay) |
| 3. <====TCP SYN+ACK, IPv6=== | |
| 4. =====TCP ACK, IPv6=====> | |

Scenario 3: RFC6555 implementatie in netYCE, IPv6 bereikbaar

Bij scenario 1 is het belangrijk om de IPv4 verbinding netjes op te hangen. Dit vermindert de belasting op de server en de node.

Bij scenario 2 zou het kunnen zijn dat de node alsnog een SYN+ACK stuurt op de IPv6 SYN, echter dat er al een IPv4 verbinding tot stand is gekomen. In dat geval moet de IPv6 verbinding opgehangen worden door middel van een ACK en daarna een RST. Ditzelfde zou ook kunnen optreden bij scenario 3, echter dit lijkt op scenario 1 waarbij stap 4 en 5 omgedraaid zijn en stap 7 een TCP RST op IPv6 is.

6.7 Applicatie aanpassingen

De enige noodzakelijke aanpassing bij de NetYCE server is de configuratie van een IPv6 adres op de netwerkinterfaces van de server. De gebruikte software, Perl, Apache en MySQL ondersteunen allemaal al IPv6. Afhankelijk van de Apache configuratie moet hier nog wel het één en ander aangepast worden. Dit is echter alleen noodzakelijk als de gebruikersinterface ook op IPv6 bereikbaar moet zijn.



7 Ontwerpfase

7.1 Inleiding

Tijdens de ontwerpfase zijn de in het functioneel ontwerp gedefinieerde use cases^[4] verder uitgewerkt. Ook is hier de database aanpassing ontworpen. Dit is samengevoegd tot een technisch ontwerp welke is opgenomen als Bijlage VI van dit document.

7.2 Use cases

In de analysefase is een gewenste situatie beschreven. Met behulp van de MoSCoW lijst uit het analyserapport^[Bijlage II-4] is de gewenste situatie vertaald in een aantal use cases.

Met uitzondering van de ping functionaliteit zijn de Could haves en Won't haves items niet opgenomen als use case. De implementatie van de ping functionaliteit is eenvoudig te ontwikkelen. NetYCE maakt gebruik van fping^[5] om netwerkkapparatuur te pingen.

Fping is uitgerust met zowel functionaliteit voor het pingen van IPv4 als voor IPv6. Om ping functionaliteit te realiseren voor IPv6 in NetYCE, kan de huidige code eenvoudig gekopieerd en aangepast worden.

Tijdens het maken van het functioneel ontwerp is ondersteuning van IPv6 in service taken als use case opgenomen^[Bijlage V-3.2.2]. Tijdens het maken van het technisch ontwerp is echter besloten deze als apart hoofdstuk in het technisch ontwerp op te nemen^[Bijlage VI-3].

In de ontwerpfase zijn alle benodigde use cases verder uitgewerkt. Deze uitwerking bevat een korte tekstuele uitleg over wat de use case inhoudt. Vervolgens is er door een korte tabel een aantal kenmerken benoemd. Deze kenmerken zijn: de betrokken gebruikersrollen, de aanleiding van de use case, de samenhang met andere use cases en de wijze en frequentie van uitvoering.

Voor een aantal van deze uses cases zijn ook schermontwerpen gemaakt. Deze zijn gemaakt door screenshots van de huidige applicatie aan te vullen met nieuwe informatie.

Om de software ontwikkelaars inzicht te geven in de keuzes en de bedoeling van een use case is ook een activiteiten diagram toegevoegd per use case. In dit diagram staat sequentieel afgebeeld wat voor proces de gebruiker moet doorlopen voor de use case. Daarnaast is een scenario beschrijving toegevoegd per use case. Dit moet de ontwikkelaars inzicht geven in de vereiste beginstaat en het gewenste eindresultaat.

Er is voor gekozen om geen sequence diagrammen op te nemen bij de use cases. Deze keuze is gemaakt omdat er voornamelijk interactie plaats vindt tussen de gebruiker en de applicatie en er zeer weinig interactie plaats vindt tussen de applicatie en andere applicaties. Het activiteitendiagram geeft voldoende inzicht in de gewenste resultaten en procesgang.



7.3 Service task aanpassingen

De gestandaardiseerde ontwerpen binnen NetYCE worden gerealiseerd door middel van service taken. Deze servicetaken maken gebruik van een eigen syntax waarmee het netwerkontwerp gemaakt kan worden op een gestandaardiseerde manier.

De syntax is opgebouwd uit 4 typen opdrachten:

Zoek (Locate) opdrachten. Met de locate opdracht is het mogelijk objecten uit het netwerkontwerp op te zoeken. Deze objecten zijn onder andere poorten, nodes, IP-adressen en subnetten. Voor de IPv6 ondersteuning wordt deze uitgebreid met de mogelijkheid IPv6 subnetten te zoeken, IPv6 adressen te zoeken en het VLAN id van een bepaald IPv4 subnet te vinden.

Toewijzing-opdrachten (Assign). Met de assign opdracht is het mogelijk om objecten uit het netwerkontwerp een bepaalde waarde toe te kennen, of objecten toe te kennen aan andere objecten. Voorbeelden van het toekennen van een bepaalde waarde is bijvoorbeeld het toekennen van een VLAN id aan een subnet, of het toekennen van een poortstatus aan een poort. Voorbeelden van het toekennen van objecten aan een ander object zijn het toekennen van een IP adres aan een poort en het toekennen van een subnet aan een poort of een groep poorten. Voor de IPv6 ondersteuning wordt de assign opdracht uitgebreid met de mogelijkheid een IPv6 adres aan een poort te koppelen, de mogelijkheid een IPv6 subnet te voorzien van een VLAN id, een naam, een template en een VRF en de mogelijkheid een IPv6 subnet te koppelen aan een poort of een groep van poorten.

Toevoeg-opdrachten (Add). Met de add opdracht is het mogelijk om nieuwe objecten toe te voegen aan het netwerkontwerp. Deze objecten zijn bijvoorbeeld nieuwe subnetten, nieuwe poorten in een netwerkkapparaat, of een nieuw netwerkkapparaat. Voor de IPv6 ondersteuning wordt de add opdracht uitgebreid met de mogelijkheid een IPv6 subnet toe te voegen aan het netwerkontwerp.

Verwijder opdrachten (Delete). Met de delete opdrachten is het mogelijk om objecten uit het netwerkontwerp te verwijderen. Het verwijderen van de objecten is beperkt tot het verwijderen van netwerkkapparatuur, poorten en subnetten. Het verwijderen van poorten is onderverdeeld in een complete poort verwijdering, het verwijderen van een link tussen poorten, of het verwijderen van alle subnetten van een poort. Voor de IPv6 ondersteuning wordt de delete opdracht uitgebreid met de mogelijkheid een IPv6 subnet te verwijderen van een poort of een groep poorten, de mogelijkheid het IPv6 subnet volledig te verwijderen of het verwijderen van een VRF welke gekoppeld is aan het IPv6 subnet.



7.4 Database aanpassing

Om de IPv6 gegevens op te slaan is een database aanpassing nodig. Hoofdstuk 4 van Bijlage VI bevat de benodigde aanpassingen in detail.

Bij het ontwerpen van de tabellen is in eerste instantie gekeken naar de huidige tabellen voor IPv4. Hoewel het mogelijk is om zowel IPv6 als IPv4 gegevens in dezelfde database op te slaan vereist dit in de omliggende software erg veel controle op fouten. Tevens zijn IPv4 en IPv6 niet compatible met elkaar. Hierom is besloten om de gegevens ook apart op te slaan. De huidige IPv4 tabellen zijn wel als uitgangspunt genomen voor het ontwerp.

Een belangrijk verschil tussen de huidige IPv4 tabellen en de nieuwe IPv6 tabellen is de manier van opslag van de adressen. In de huidige tabellen worden adressen opgeslagen als tekst. Controle op geldigheid van de adressen moet op deze manier plaatsvinden in de software. Dit is vaak een werkbare methode, echter wanneer er gerekend moet worden met de adressen is het zinvoller om de adressen op te slaan als binaire waarde. Tijdens een presentatie van een software engineer van Facebook op het IPv6 world congres 2014 in Parijs^[6] werd dit ook aangeraden. Om deze reden worden alle IPv6 adresvelden opgeslagen als varbinary(16). Wanneer een veld zowel tekst als een IPv6 adres kan bevatten moet er gebruik gemaakt worden van een conversie naar HEX zodat er nog wel gerekend kan worden met de adressen. De conversie naar HEX wordt gedaan door gebruik van de hex() functie van mysql in combinatie met de inet6_aton() functie van mysql.

Het opslaan van de adressen in het varbinary formaat heeft als nadeel dat de adressen zonder conversie niet leesbaar zijn als adressen. Omdat gebruikers van NetYCE zelf toegang krijgen tot de database en eigen query's kunnen uitvoeren is gekozen om naast de tabellen ook een aantal views te creëren. De varbinary velden uit de tabellen worden in deze views omgezet naar een leesbaar IPv6 adres.

Een ander belangrijk verschil met de IPv4 tabellen zit in de opdeling van een prefix in subnetten. Bij de IPv4 tabellen wordt voor een type subnet elk subnet als offset opgeslagen. Wanneer er dus 100 subnetten voor VoIP diensten aangemaakt worden, worden er 100 records aangemaakt in de database. Bij IPv6 wordt gebruik gemaakt van een minimale en een maximale offset. Hierdoor is er 1 record per gebruikerstype nodig.



8 Implementatiefase

8.1 Inleiding

Hoewel de implementatiefase buiten de scope van het project valt is een onderdeel van dit project het opleveren van een implementatievoorstel. Dit voorstel is bijgevoegd als Bijlage VII van dit document.

8.2 Fasering

De implementatie wordt gefaseerd ontwikkeld. Hierbij wordt eerst een gedeelte van de applicatie aangepast voor een Proof of Concept. De implementatie hiervan moet eind oktober 2014 afgerond zijn.

Na de PoC en wanneer er tijd is voor de verdere ontwikkeling wordt de overige functionaliteit ontwikkeld. Deze wordt per use case ontwikkeld, getest en gedocumenteerd.

Hoewel in het originele plan de implementatie eind oktober 2014 afgerond zou moeten zijn, is dit inmiddels bijgesteld. Het uitlopen van een ander project, alsmede verhoogde prioriteit voor andere functionaliteit heeft hiertoe geleid. De nieuwe einddatum is nog niet bekend.

8.3 Git IPv6 branch

Bij de huidige ontwikkeling van NetYCE wordt gebruik gemaakt van Git^[7]. Hiermee wordt versie beheer gedaan voor de software en wordt de code afgesplitst (een branch) om bepaalde functionaliteit te ontwikkelen. Eenmaal ontwikkeld, wordt de branch samengevoegd met de master branch.

Voor de ontwikkeling NetYCE worden op dit moment 2 branches gebruikt. De master- en de development branch. Voor de ontwikkeling van IPv6 functionaliteit is nieuwe branch afgesplitst van de development branch. In deze branch wordt de IPv6 functionaliteit ontwikkeld. Zodra deze productierijp is wordt de branch samengevoegd met de development branch, getest en vervolgens als productieversie naar de klanten gedistribueerd.

8.4 Proof of Concept fase

Om een Proof of Concept uit te voeren zal een beperkte hoeveelheid functionaliteit ontwikkeld worden in NetYCE. Deze functionaliteit zal beperkt zijn tot het tonen van gegevens en het aanpassen van de database. De invoer van nieuwe gegevens, alsmede de validatie van die gegevens wordt nog niet ontwikkeld.

8.4.1 Database aanpassing

Voor de Proof of Concept is het van belang dat de gegevens uitgelezen kunnen worden door de formulieren. Hiervoor dient de database in een vroeg stadium aangepast te worden voor de opslag van de IPv6 informatie. De aanpassing zal gedaan worden door middel van een patchfile. Deze methode wordt reeds gebruik voor database aanpassingen en hiervan is aangetoond dat het werkt. De patchfile is ontwikkeld in de IPv6 branch.



8.4.2 IPv6 communicatie naar de netwerkkaparaatuur

Voor de Proof of Concept is het van belang dat NetYCE kan communiceren met de netwerkkaparaatuur over het IPv6 protocol. Tijdens de PoC fase zal daarom de communicatiemodule aangepast moeten worden voor het gebruik van IPv6.

De implementatie van deze aanpassingen is geraamd op 2 tot 3 dagen werk.

8.4.3 Frontend aanpassing

Voor de Proof of Concept worden 3 formulieren aangepast om de gegevens te tonen. De betreffende formulieren zijn: Client details, Site details en Node details.

De implementatie van deze aanpassingen is geraamd op 2 tot 3 dagen werk.

8.5 Volledige implementatiefase

8.5.1 Frontend aanpassingen

De frontend aanpassingen worden gefaseerd ontwikkeld. Allereerst zal een beperkte implementatie van een aantal use cases ingevoerd worden. Deze implementatie beperkt zich tot het tonen van de gegevens en nog niet het invoeren, verwijderen, aanpassen en valideren van de gegevens. Het tonen van de gegevens wordt gebruikt in de Proof of Concept opstelling.

Wanneer er tijd is voor de ontwikkeling van de functionaliteit wordt deze per use case ontwikkeld. Voor de ontwikkeling van 1 use case zal tussen de 3 en 10 werkdagen benodigd zijn.

8.5.2 Backend aanpassingen

Het uitvoeren van de service taken is afhankelijk van de Service Task module in de backend. Tijdens de volledige implementatie moet deze module ontwikkeld worden. In hoofdstuk 3 van het Technisch Ontwerp^[Bijlage V1] staan de verschillende taken beschreven. De ontwikkeling van deze taken moet per taak worden ontwikkeld, getest en gedocumenteerd alvorens door te gaan naar de volgende taak.

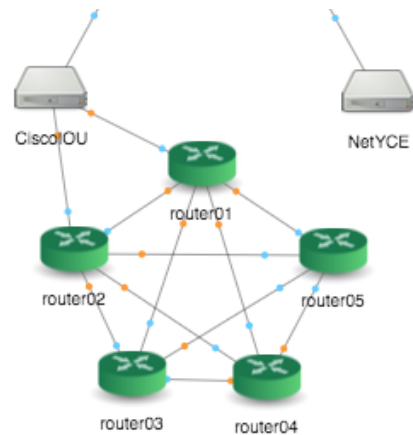
De ontwikkeling van deze module zal naar verwachting 10 tot 15 dagen in beslag nemen.



8.6 Proof of Concept opstelling

Voor de Proof of Concept opstelling wordt gebruik gemaakt van een NetYCE server welke gekoppeld is met de Git IPv6 branch. Deze server draait door middel van VirtualBox en heeft verbinding met een host only netwerk. Dit host only netwerk heeft vervolgens verbinding met een gevirtualiseerd netwerk van 5 routers door middel van Cisco IOU.

Cisco IOU maakt gebruik van Cisco IOS images die aangepast zijn om te gebruiken op x86 Unix computers. Elke virtuele router wordt opgestart als een VM op de IOU server. Het voordeel IOU is dat er geen hardware benodigd is, terwijl de IOS functionaliteit nagenoeg gelijk is aan de werkelijke Cisco IOS images. Hiermee is het aantonen van een Proof of Concept mogelijk.



Afbeelding 7: Proof of Concept opstelling

Met de Proof of Concept opstelling zal worden aangetoond dat NetYCE in staat is te verbinden op IPv6 naar het gevirtualiseerde netwerk en hier configuratie aanpassingen door te voeren.

8.7 Documentatie

Als onderdeel van de ontwikkeling wordt het aanleggen van documentatie over het gebruik van IPv6 binnen NetYCE, alsmede de werking van IPv6 binnen NetYCE beoogd. Per uitgewerkte use case wordt de documentatie bijgewerkt op de NetYCE documentatie site^[8].

Naast het documenteren van de functies moeten er ook howto's geschreven worden over het gebruik van IPv6 met NetYCE. Dit beperkt zich echter niet tot IPv6 maar heeft betrekking op de gehele applicatie. Inmiddels is de wens uitgesproken om de documentatie volledig te herzien. Dit zal als een apart project behandeld worden waar op dat moment de IPv6 handleidingen in meegenomen worden.



9 Evaluatie

9.1 Opdracht

Dit afstudeerproject is begonnen met de opdracht

Realiseer IPv6 ondersteuning in netYCE waarmee het mogelijk wordt om netwerken van routers, switches en firewalls geautomatiseerd te configureren met een IPv6 of dual stack configuratie.

Direct aan het begin van deze opdracht is erkend dat de opdracht te groot was en niet past binnen de tijd die staat voor het afstuderen.

Wanneer nu gekeken wordt naar de huidige stand van zaken ligt er een basis waarop verder ontwikkeld kan worden. De IPv6 ondersteuning is nog niet ten volle gerealiseerd maar begint wel vorm te krijgen. Er is inmiddels een Git branch gemaakt voor de IPv6 ondersteuning in NetYCE en wanneer er tijd is wordt daar functionaliteit in ontwikkeld. De eerste functionaliteit is reeds ontwikkeld en kan getoond worden in een Proof of Concept. Helaas is op dit moment nog niet bekend wanneer de implementatie volledig afgerond kan zijn.

9.2 Persoonlijk

Vooral in de aanloop naar het project ben ik er achter gekomen dat de opleiding niet aansloot bij de benodigde competenties voor het afstuderen. In de 4 jaar deeltijd opleiding heeft nagenoeg geen ruimte gezeten voor onderzoeksvaardigheden, die wel benodigd zijn voor het afstuderen. Ik liep onder andere tegen problemen aan bij het schrijven van het afstudeervoorstel en in de structuur en volgorde van een onderzoek. Terugkijkend op het curriculum is er slechts 1 vak geweest waar er kort aandacht besteed is aan onderzoeksvaardigheden. Hierdoor heeft het me veel moeite en extra tijd gekost om het afstuderen te voltooien.

De onderzoeksvaardigheden heb ik me gelukkig wel eigen kunnen maken. Naast de onderzoeksvaardigheden heb ik veel geleerd over het gebruik en de werking van IPv6.



10 Conclusies en aanbevelingen

Aan het einde van dit project is de conclusie dat IPv6 niet erg verschilt van IPv4. De grote hoeveelheid beschikbare adressen voor IPv6 maakt dat er flexibeler omgegaan kan worden met adresblokken. Waar bij het gebruik van IPv4 nog erg goed nagedacht moet worden over het voorkomen van verspilling is dit bij IPv6 niet aan de orde. Door de enorme hoeveelheid beschikbare adressen en het gebruik van voornamelijk /64 subnetten is het niet nodig hierover na te denken.

Door het gebruik van verschillende technieken in IPv6 zoals Neighbor Discovery en SLAAC wordt het beheer van netwerken gemakkelijker. Waar in IPv4 netwerken computers ofwel de juiste IP instellingen geconfigureerd moeten worden, ofwel de DHCP server(s) op de juiste manier ingesteld moeten zijn, zoekt een computer met een IPv6 verbinding de juiste instellingen zelf uit.

De ondersteuning van IPv6 in NetYCE kan bedrijven helpen om snel het huidige netwerk te voorzien van een dual stack configuratie. Daarom is aan te bevelen de ontwikkeling van IPv6 binnen NetYCE snel in te zetten. Zeker omdat de ontwikkeling relatief weinig tijd kost en weinig risico met zich meebrengt.



11 Bronvermelding en literatuurlijst

11.1 Bronnen

- [[ARIN-IPv6](#)] ARIN number resource policy manual
www.arin.net/policy/nrpm.html#six592
- [[Cisco-IPv6](#)] Cisco Validated Design, IPv6 Addressing White paper, August 2013
www.cisco.com/c/dam/en/us/td/docs/solutions/CVD/Aug2013/CVD-IPv6AddressingWhitePaper-AUG13.pdf
- [[IANA-IPv6-AS](#)] Internet Protocol Version 6 Address Space
www.iana.org/assignments/ipv6-address-space/ipv6-address-space.xml
- [[IANA-IPv6-GUAA](#)] IPv6 Global Unicast Address Assignments
www.iana.org/assignments/ipv6-unicast-address-assignments/ipv6-unicast-address-assignments.xhtml
- [[IANA-IPv6-Reg](#)] IANA IPv6 Special-Purpose Address Registry
www.iana.org/assignments/iana-ipv6-special-registry/iana-ipv6-special-registry.xhtml
- [[RFC3139](#)] Requirements for Configuration Management of IP-based Networks, June 2001
tools.ietf.org/pdf/rfc3139.pdf
- [[RFC3315](#)] Dynamic Host Configuration Protocol for IPv6 (DHCPv6), July 2003
tools.ietf.org/pdf/rfc3315.pdf
- [[RFC3849](#)] IPv6 Address Prefix Reserved for Documentation , July 2004
tools.ietf.org/pdf/rfc3849.pdf
- [[RFC3879](#)] Deprecating Site Local Addresses, September 2004
tools.ietf.org/pdf/rfc3879.pdf
- [[RFC4193](#)] Unique Local IPv6 Unicast Addresses, October 2005
tools.ietf.org/pdf/rfc4193.pdf
- [[RFC4291](#)] IP Version 6 Addressing Architecture, February 2006
tools.ietf.org/pdf/rfc4291.pdf
- [[RFC4773](#)] Administration of the IANA Special Purpose IPv6 Address Block, December 2006
tools.ietf.org/pdf/rfc4773.pdf
- [[RFC4862](#)] IPv6 Stateless Address Autoconfiguration, September 2007
tools.ietf.org/pdf/rfc.pdf
- [[RFC5375](#)] IPv6 Unicast Address Assignment Considerations, December 2008
tools.ietf.org/pdf/rfc5375.pdf
- [[RFC5737](#)] IPv4 Address Blocks Reserved for Documentation, January 2010
tools.ietf.org/pdf/rfc5737.pdf
- [[RFC5952](#)] IPv6 Text Representation, August 2010
tools.ietf.org/pdf/rfc5952.pdf
- [[RFC6164](#)] Using 127-Bit IPv6 Prefixes on Inter-Router Links, April 2011
tools.ietf.org/pdf/rfc6164.pdf



- [[RFC6555](#)] Happy Eyeballs: Success with Dual-Stack Hosts, April 2012
tools.ietf.org/pdf/rfc6555.pdf
- [[RIPE589](#)] IPv6 Address Allocation and Assignment Policy, May 2013
www.ripe.net/ripe/docs/ripe-589
- [[SURF-IPv6](#)] Surfnet Whitepaper IPv6-nummerplan opstellen.
www.surf.nl/kennis-en-innovatie/kennisbank/2013/whitepaper-ipv6-nummerplan-opstellen.html

11.2 Referenties

- [1] Google IPv6 statistics:
www.google.com/intl/nl/ipv6/statistics.html
- [2] Internet Society Tech Matters:
internetsociety.org/blog/tech-matters/2014/01/prediction-ipv6-adoption-hits-double-digits-2014
- [3] IPv6 ULA (Unique Local Address) registration:
www.sixxs.net/tools/grh/ula/
- [4] Use cases
http://nl.wikipedia.org/wiki/Use_case
- [5] Fping
http://fping.org/
- [6] Internet Society, Facebook Extremely impressive internal use of IPv6:
www.internetsociety.org/deploy360/blog/2014/03/facebooks-extremely-impressive-internal-use-of-ipv6/
- [7] Git
http://en.wikipedia.org/wiki/Git_(software)
http://git-scm.com/
- [8] NetYCE documentation:
https://wiki.netyce.com/
- [[RFC781](#)] Internet Protocol, September 1981
tools.ietf.org/pdf/rfc781.pdf
- [[RFC1883](#)] Internet Protocol, Version 6 (IPv6) Specification, December 1995
tools.ietf.org/pdf/rfc1883.pdf
- [[RFC1918](#)] Address Allocation for Private Internets, February 1996
tools.ietf.org/pdf/rfc1918.pdf
- [[RFC3971](#)] Secure Neighbor Discovery (SEND), March 2005
tools.ietf.org/pdf/rfc3971.pdf

Bijlagen

| | |
|--------------|-----------------------|
| Bijlage I: | Het afstudeervoorstel |
| Bijlage II: | Analyse rapport |
| Bijlage III: | Plan van Aanpak |
| Bijlage IV: | Afstudeercontract |
| Bijlage V: | Functioneel ontwerp |
| Bijlage VI: | Technisch ontwerp |
| Bijlage VII: | Implementatievoorstel |

Bijlage I: Het afstudeervoorstel

Project Proposal

Implementing IPv6 within NetYCE

Research proposal Derco Sportel

Student 1548287

Colophon

| | |
|--------------------|--|
| | NetYCE P.O. 320 1380 AH Weesp, The Netherlands Business Centre Muiderpoort Leeuwendveldseweg 5n 1382 LV Weesp, The Netherlands |
| Contact | Derco Sportel <i>Network Consultant</i> ☎ +31 6 1418 1361 ☎ +31 20 897 3412 ✉ derco.sportel@netyce.com 🌐 derco.sportel |
| Company supervisor | Ronny Lam <i>Solution Architect</i> ☎ +31 6 3976 6145 ☎ +31 20 897 3412 ✉ ronny.lam@netyce.com 🌐 ronny_lam |
| Version | 3.0 |
| Date | 4 march 2014 |

Table of Contents

| | |
|--|---|
| Colophon | 2 |
| Introduction | 4 |
| Document revision | 4 |
| NetYCE B.V. | 4 |
| Position of the student within NetYCE B.V. | 4 |
| The company supervisor | 5 |
| The netYCE platform | 5 |
| Bachelor assignment | 6 |
| Project business case | 6 |
| Bachelor assignment target | 6 |
| Bachelor assignment scope | 6 |
| Deliverables | 7 |
| Project scope | 7 |
| Relations with other projects | 7 |
| Risk assessment | 7 |
| The challenge of the assignment | 8 |
| Relevance of the project with the specialization | 8 |
| Projected timeline | 9 |

Introduction

Document revision

| Version | Date | Author | Notes |
|---------|-------------|---------------|---|
| 0.1 | 31 oct 2013 | Derco Sportel | First draft |
| 1.0 | 26 nov 2013 | Derco Sportel | Final version. |
| 1.1 | 17 dec 2013 | Derco Sportel | Processing comments |
| 2.0 | 19 dec 2013 | Derco Sportel | Final version. |
| 2.1 | 14 feb 2014 | Derco Sportel | Processing comments. |
| 2.2 | 25 feb 2014 | Derco Sportel | Processing comments, after meeting with A. Jongman. |
| 2.3 | 4 mar 2014 | Derco Sportel | Processing comments, after mail from A. Jongman. |
| 3.0 | 4 mar 2014 | Derco Sportel | Final version. |

NetYCE B.V.

NetYCE B.V. is a company founded in 2011 as a spinoff from the Rabobank group ICT and specializes in the automation of medium and large networks.

Although only being around for 3 years, NetYCE has over 12 years of experience in the automation of networks. At NetYCE we believe that all networks can be automated when adopting Design Driven Networking, reducing Total Cost of Ownership, reducing Time to Market and assuring quality.

NetYCE B.V. has become an independent company and is working hard to grow. Rabobank is now a customer as well as the Ministry of Defense. Also there are several companies interested in the product and are considering to implement the netYCE platform.

Currently NetYCE B.V. employs 10 people, also we hire some freelance specialists when we need them. The team now consists of 4 software developers, 2 sales, 2 network specialists and 2 support staff.

Position of the student within NetYCE B.V.

For the most part of my working career, I've worked at the Netherlands Ministry of Defense. Always in a IT section. The last 3 years I've worked for IVENT. The full service IT department for the MoD and some services for other ministries within the Netherlands government.

In 2012 IVENT decided to automate a large portion of the network, in order to reduce operational costs and reduce human error. In order to achieve this, the netYCE platform was acquired. I was appointed lead engineer for the implementation of the platform within IVENT. During that time NetYCE and myself developed a good working relationship and in september 2013 I made the step forward and decided to go work for NetYCE.

At NetYCE I work as a Network Consultant and I am responsible for advising the customers how to implement netYCE within their network(s). Also I am testing the new netYCE platform from a users perspective, identifying bugs and advising the software developers to further improve the netYCE platform. And also I am writing documentation on how to use the netYCE platform.

The company supervisor

Ronny Lam is the Solution Architect of NetYCE B.V. Ronny has worked for several ISP's including UPC and Casema, over the last 13 years. During this time he has gained a lot of experience in network engineering as well as programming. As Solution Architect, Ronny also is responsible for implementing the netYCE platform at the customers, including user and admin training. Ronny is also the operational manager of netYCE and has a role in the pre-sales process.

The netYCE platform

In 2001 the Rabobank needed a way to migrate their existing WAN infrastructure with more than 1600 device from 3Com to Cisco in just a short timeframe. In order to achieve this, they needed a product to automate this process. Because no product existed, Rabobank has asked their supplier to develop a network provisioning and automation tool to support their migration process.

Network provisioning tools are tools that configure network devices with the correct configuration for the specific device. Traditionally this is still done by a network engineer.

The result was essentially netYCE version 1. More than 12 years later, version 6 of the netYCE platform is almost ready for production.

With the netYCE platform, companies can fully design, maintain and operate their network. The platform is designed automate network changes, reduce time to market and reduce the total cost of ownership.

Bachelor assignment

Project business case

Because the netYCE platform was developed in 2001 as a network automation tool, IPv6 was not part of the original program. With the current market and new possible customers who are mainly large ISP's, IPv6 is now a must have. Basic IPv6 functionality should be implemented before the third quarter of 2014.

Bachelor assignment target

In order to successfully implement IPv6 within netYCE, we first need to know how IPv6 will be used within large enterprises and ISP's. The target of the project will be to identify several best practices of IPv6 implementations and to integrate IPv6 within the existing data model. The best practices need to be translated into use cases and sequence diagrams.

Once the studies are complete, the development team at NetYCE B.V. will implement the functionality in the netYCE platform.

Bachelor assignment scope

1. Research how IPv6 is used at enterprises and ISP's.
NetYCE B.V. has had no experience regarding implementation of IPv6 at their customers. There are no existing customers who are using IPv6. In order to provide the customer with a working solution, we first need to know what the use cases are regarding IPv6.
Because NetYCE is a provisioning platform for large networks, as described in [RFC3139](#), we need to have a generic approach of the problem.
Examples of the research area are:
 - 1.1. *The first thing we want to know is, if the usage of /127 subnets on inter router links ([RFC6164](#)) will be used, or that all subnets will be /64 ([RFC4291](#)). According to RFC4291, all subnets should be /64, however it is possible to deviate from that RFC. We will want to know, if and how the customer want's to deviate.*
 - 1.2. *Also we don't know if Stateless Address Autoconfiguration (SLAAC, [RFC4862](#)) will be used, Statefull Autoconfiguration (DHCPv6, [RFC3315](#)) or static addresses.*
 - 1.3. *There are about 30 relevant RFC's on IPv6, to be implemented in the platform. These RFC's need to be studied, to identify, if and how this RFC needs to be implemented. Furthermore, we need to analyze several more implementation reports and best practice documents from several companies and foundations. For instance [SurfNet](#) and [6deploy](#) publications on IPv6.*
The results of the research will be part of the thesis and will serve as input for the functional design and the datamodel design.
2. Design a data model to integrate IPv6 within the netYCE platform, based on the research.
The existing database model needs to be expanded to provide the necessary tables to store the IPv6 addresses. This must be done without changing the existing data-mode, with the exception of the necessary tables used to store IPv6 data.
The results will be used within the functional design.
3. Create a functional design for the needed programming.
Within the functional design of the programming, the use cases will be designed, as well as mock-ups of the forms.

When the functional design is complete, it will be verified with the customers, to check if the design is an accurate representation of their wishes.

The results will be used by the software developers to implement the functionality within the netYCE platform.

Deliverables

This assignment will deliver the following products for NetYCE B.V.:

1. Research rapport on the use of IPv6;
2. Data-model design, compatible with the current data-model;
3. Functional design of the needed programming

This assignment will deliver the following products for the Hogeschool Utrecht:

1. Project plan;
2. Thesis

Project scope

The scope of this project will limit to the implementation of the deliverables within netYCE platform. Other aspects on the development of the netYCE platform are out of scope.

Relations with other projects

At this time there are no other projects at NetYCE involved in the implementation of IPv6. Several companies however are considering implementing the netYCE platform. Most of these companies want to begin with providing IPv6 to their customers.

Risk assessment

Because IPv6 is a new functionality within the netYCE platform, and the modular design of the platform the risk of disturbing the current programming is limited.

Because of the limited resources within NetYCE, and several new possible customers, it may be possible that my attention can't always be with this project. The deadlines may not be met, because of this.

The challenge of the assignment

Because the netYCE platform is already in production, the main challenge is to successfully integrate the IPv6 functionality, without changing the existing programming. Especially the existing data-model.

The netYCE platform is a product which has evolved over the last 13 years, and although several improvements have been made over the years, the layered approach of software design has not yet been used. In the year 2014, NetYCE wants to implement the MVC framework, in order to be more flexible. This project must take that into account.

Although the concept of IPv6 is relatively simple, NetYCE needs to be able to provide IPv6 functionality to all their customers. And because IPv6 is relatively new, most companies don't know yet how to implement this within their network. This is probably the reason that only a few ISP's in The Netherlands support native IPv6 and implementation of IPv6 has been postponed at UPC, KPN and Ziggo.

In preliminary conversations with several companies, there are a lot of questions as to how IPv6 must be implemented, without disrupting the current operations and also making the implementation future proof. The implementation of IPv6 is likely to be different from an ISP in comparison with an enterprise environment. There is a challenge meeting both markets.

Relevance of the project with the specialization

The specialization I am following is mostly application development, where as my own specialty is network engineering. It is because of that combination, I was asked to join NetYCE. This project is basically a combination of both specialities. During this project I will have the opportunity to combine my passion for network engineering with the chosen specialization.

Projected timeline

| Orientation phase | Period | Projected hours |
|---|---------------|-----------------|
| Analyzing current situation | November 2013 | 8 |
| Formulation of the problem | November 2013 | 8 |
| Formulation of the project proposal | November 2013 | 40 |
| Adapt the project proposal to comment | February 2014 | 40 |
| Research phase | | |
| Describing the current situation | February 2014 | 30 |
| Formulate targets | March 2014 | 20 |
| Formulate project assignment with sub assignments | March 2014 | 20 |
| Drafting project plan | March 2014 | 50 |
| Execution phase | | |
| Market research | March 2014 | 80 |
| Literature research | March 2014 | 80 |
| Drafting of the report | March 2014 | 40 |
| <i>End of execution phase one (market research)</i> | | |
| Design the data-model | April 2014 | 40 |
| <i>End of execution phase two (data-model design)</i> | | |
| Design the functional specifications | April 2014 | 160 |
| <i>End of execution phase three (functional specifications)</i> | | |
| Writing the thesis | May 2014 | 160 |
| Adapt the thesis to comment | May 2014 | 40 |
| <i>End of execution phase four (Thesis)</i> | | |

Bijlage II: Analyse rapport

Analyse rapport

IPv6 implementatie in NetYCE

Derco Sportel

1548287

Colofon

| | |
|--------------------|--|
| | NetYCE B.V. Postbus 320 1380 AH Weesp, Nederland Business Centre Muiderpoort Leeuwendveldseweg 5n 1382 LV Weesp, Nederland |
| Auteur | Derco Sportel <i>Network Consultant</i> ☎ +31 6 1418 1361 ✉ derco.sportel@netyce.com 🌐 derco.sportel |
| Docentbegeleider | Kees Uiterwijk <i>Hogeschooldocent, Systeembeheer</i> ☎ +31 88 481 8184 ✉ kees.uitervijk@hu.nl 🌐 kees.uitervijk |
| Bedrijfsbegeleider | Ronny Lam <i>Solution Architect</i> ☎ +31 6 3976 6145 ✉ ronny.lam@netyce.com 🌐 ronny_lam |
| Versie | 1.0 |

Document

Document geschiedenis

| Versie | Datum | Auteur | Notities |
|--------|--------------|---------------|--|
| 0.1 | 8 juli 2014 | Derco Sportel | Eerste opzet |
| 0.2 | 14 juli 2014 | Derco Sportel | Aanpassingen n.a.v. commentaar |
| 0.3 | 24 juli 2014 | Derco Sportel | Conclusie en aanbevelingen, Design Build Operate toegelicht. |
| 0.4 | 25 juli 2014 | Derco Sportel | Tekstuele aanpassingen. Conclusie herschreven. |
| 1.0 | 4 aug 2014 | Derco Sportel | Conclusie herschreven. |

Distributie

| Name | 0.1 | 0.2 | 0.3 | 0.4 | 1.0 |
|---------------------|-----|-----|-----|-----|-----|
| Derco Sportel | ✓ | ✓ | ✓ | ✓ | ✓ |
| Eric Yspeert | | | | | ✓ |
| Ronny Lam | ✓ | ✓ | ✓ | ✓ | ✓ |
| Kees Uiterwijk (HU) | | | | | ✓ |
| Gerald Ovink (HU) | | | | | ✓ |
| J de Doelder (CvT) | | | | | ✓ |

Inhoudsopgave

| | |
|--|----|
| Colofon | 2 |
| Document | 3 |
| Document geschiedenis | 3 |
| Distributie | 3 |
| 1 Inleiding | 5 |
| 1.1 Aanleiding | 5 |
| 1.2 Gebruik van de documentatieprefix | 5 |
| 2 Huidige situatie | 6 |
| 2.1 Probleemstelling | 6 |
| 2.2 Marktontwikkeling | 6 |
| 2.3 Huidige architectuur van de netYCE applicatie | 6 |
| 2.4 Huidige functionaliteit van de netYCE applicatie | 6 |
| 2.4.1 Algemene functionaliteit | 6 |
| 2.4.2 IPv4 specifieke functionaliteit | 7 |
| 3 Gewenste situatie | 8 |
| 3.1 IPv6 ondersteuning in het Design proces | 8 |
| 3.1.1 IPv6 prefix plan | 8 |
| 3.1.2 IPv6 ondersteuning in service tasks. | 8 |
| 3.1.3 IPv6 ondersteuning in parameter groups | 9 |
| 3.1.4 Koppelen van IPv4 adressen met IPv6 adressen. | 9 |
| 3.2 IPv6 ondersteuning in het Build proces | 9 |
| 3.2.1 IPv6 prefix beheren in een client | 9 |
| 3.2.2 IPv6 prefix beheren in een site | 10 |
| 3.2.3 IPv6 subnet beheren in een service | 10 |
| 3.2.4 IPv6 adres veranderen op een interface | 10 |
| 3.2.5 IPv6 subnet beheren op een interfaces | 10 |
| 3.3 IPv6 ondersteuning in het Operate proces | 11 |
| 3.3.1 Verbinden naar een door netYCE beheert apparaat met IPv6 | 11 |
| 3.3.2 ping6 en traceroute6 | 11 |
| 3.3.3 Integraties met externe systemen | 11 |
| 3.4 Gevolgen voor de software | 12 |
| 3.4.1 Database aanpassing | 12 |
| 3.4.2 Frontend aanpassingen | 12 |
| 3.4.3 Backend aanpassingen | 12 |
| 3.5 Gevolgen voor de installatie | 12 |
| 4 MoSCoW analyse | 13 |
| 5 Conclusie en aanbevelingen | 14 |
| 6. Bronvermelding en literatuurlijst | 15 |
| 6.1 Bronnen | 15 |
| 6.2 Referenties | 15 |

1 Inleiding

1.1 Aanleiding

Sinds het begin van het internet is het aantal aansluitingen op internet jaarlijks gegroeid. Omdat steeds meer apparaten aangesloten kunnen worden op internet, is de behoefte voor meer adressen ook alleen maar toegenomen. Omdat er in het huidige IP protocol niet voldoende adressen beschikbaar zijn om aan alle eisen en wensen te voldoen is in 1998 een nieuw protocol ontwikkeld, welke hier wel in voorziet. Dit protocol is IP versie 6 of IPv6 genoemd.

Sinds 6-6-2012 is dit nieuwe protocol wereldwijd in gebruik en neemt het gebruik van IPv6 verkeer op internet maandelijks toe. Veel (internet) service providers zijn bezig hun netwerk geschikt te maken voor IPv6 en IPv6 uit te rollen naar hun klanten. Ook grote bedrijven zijn bezig, of gaan op korte termijn bezig met de uitrol van IPv6 in hun netwerk. Omdat netYCE te maken heeft met zowel de service provider markt, als de Enterprise markt is het belangrijk dat netYCE functionaliteit biedt om IPv6 netwerken te kunnen modelleren, bouwen en onderhouden.

Dit document beschrijft de analyse welke onderdelen van IPv6 voor netYCE belangrijk zijn om het IPv6 protocol te ondersteunen en klanten van NetYCE de mogelijkheid te bieden om IPv6 met behulp van netYCE te ontwerpen, bouwen en configureren.

Dat IPv6 ondersteuning binnen netYCE mogelijk moet worden staat al vast, maar hoe moet nog bepaald worden.

1.2 Gebruik van de documentatieprefix

Tijdens de analyse zijn verschillende voorbeelden van IPv6 adressen aangetroffen. De prefix van alle gebruikte voorbeelden is vervangen door de documentatie prefix 2001:db8::/32 [RFC3849]. Hiermee zijn de voorbeelden gedeclassificeerd.

2 Huidige situatie

2.1 Probleemstelling

In de netYCE applicatie is op dit moment op geen enkele IPv6 functionaliteit mogelijk. Modelleren en het creëren van het abstracte netwerkmodel, alsmede het onderhouden van router en switch netwerken op IPv6 is hierdoor onmogelijk.

2.2 Marktentwikkeling

In de periode na 6 juni 2012 (world IPv6 launchday) is het aandeel IPv6 verkeer op internet toegenomen van 0,65% op 6-6-2012 naar 4,13% op 12-7-2014 ^[1] en de voorspelling van Internet Society in januari 2014 is dat dit eind 2014 10% is ^[2]. Bij gesprekken met (toekomstige) klanten wordt inmiddels vaker gevraagd naar IPv6 ondersteuning. Vooral bij de zakelijke service- en hosting providers wordt IPv6 inmiddels standaard geboden.

2.3 Huidige architectuur van de netYCE applicatie

Op dit moment wordt de netYCE applicatie geïnstalleerd op Red Hat Enterprise Linux 6.5 of de open source variant CentOS 6.5. Er wordt gebruik gemaakt van de volgende componenten:

- MySQL (5.6) of MariaDB (5.5 of 10.x) database.
- Apache (2.2.x)
- Een eigen gecompileerde Perl versie
- OpenSSH
- Telnet

Deze software ondersteunt allemaal het gebruik van IPv6. IPv6 moet echter nog wel geconfigureerd worden op de interfaces. Wanneer het netwerk waar de NetYCE server op aangesloten is levert de communicatie voor wat betreft IPv6 van en naar de server geen probleem op.

2.4 Huidige functionaliteit van de netYCE applicatie

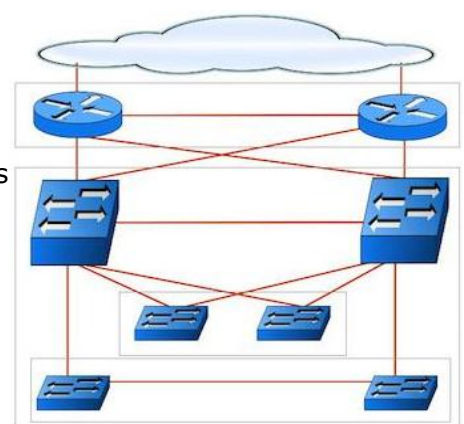
2.4.1 Algemene functionaliteit

Het huidige netYCE platform wordt gebruikt om netwerken van routers, switches en firewalls geautomatiseerd te configureren volgens vastgestelde ontwerp principes, configuraties en scenario's.

Dit wordt binnen netYCE het Design, Build, Operate proces genoemd. Hierbij wordt eerst een netwerkontwerp vastgelegd als model in de *Design* fase.

Vervolgens worden de configuratietemplates voor de benodigde apparatuur, alsmede een abstractie van het netwerk gemaakt in de *Build* fase. Hierbij worden klanten (Client) aangemaakt in de applicatie. Elke klant heeft 1 of meerdere locaties (Site) en per locatie wordt een netwerk gemaakt. Dit netwerk wordt volgens het vastgelegde model aangemaakt.

Voor een grote locatie kan dat bijvoorbeeld resulteren in een netwerk architectuur zoals in afbeelding 8. Hier is een redundant uitgevoerde routerlaag gekoppeld aan een distributielaag. Aan de distributielaag worden toegangslagen gekoppeld waar de eind apparatuur op gekoppeld wordt. De routerlaag komt 1



Afbeelding 8: Een redundante netwerk architectuur

keer voor, waar de distributie en toegangslaag meerdere keren voor kan komen. De manier van aansluiten blijft echter gelijk. Ook bijvoorbeeld de verdeling van de VLAN's kan hier op een gestandaardiseerde manier gedaan worden.

Door het gebruik van variabelen en logica in templates kunnen vervolgens configuraties voor de routers en de switches gegenereerd worden en kunnen deze geconfigureerd worden.

In de *Operate* fase worden alle routers en switches uitgevoerd met configuraties die gegenereerd worden met behulp van de templates.

Benodigde wijzigingen op het netwerk worden door middel van de Design Build Operate cyclus uitgevoerd, waardoor het werkelijke netwerk is, zoals het oorspronkelijk ontworpen is.

2.4.2 IPv4 specifieke functionaliteit

In de huidige situatie kan er door middel van IP plannen gebruik gemaakt worden van standaard indelingen van een IP reeks. Hierbij wordt een supernet ingedeeld in kleinere subnetten. Deze subnetten worden per interface bij gehouden in de netYCE applicatie, waardoor er een actueel overzicht bestaat van de IP informatie. Wanneer er geen gebruik gemaakt wordt van een IP-indeling dan is het ook mogelijk om subnetten los toe te voegen zonder vooraf ingedeeld plan.

3 Gewenste situatie

Het is wenselijk dat de Design Build Operate cyclus gebruik kan maken van IPv6. Dit houdt in dat er op verschillende plekken in het platform aanpassingen gedaan moeten worden aan de software.

3.1 IPv6 ondersteuning in het Design proces

3.1.1 IPv6 prefix plan

In de huidige situatie wordt een IPv4 plan gemaakt. Dit plan wordt tijdens de Design fase gemaakt en ingedeeld in verschillende subnetten. Deze subnetten krijgen vervolgens een functie, een VLAN indeling en er kan per functie in een VLAN een IP adres (reeks) aangewezen worden.

De ontwerpers en architecten welke gebruik maken van netYCE moeten in staat zijn om IPv6 prefixen in te delen volgens een door hen bepaald nummerplan. Voorbeelden van dit soort plannen staan onder andere beschreven in [SURF-IPv6]. Hierin wordt aangeraden om een verdeling per locatie en per gebruikstype te realiseren. Hierbij zijn er de opties locatie eerst en gebruikstype eerst.

De huidige functionaliteit van NetYCE biedt alleen een onderverdeling voor gebruikstype. Voor IPv6 moet het mogelijk zijn om een verdeling te maken voor een prefix op locatie en functie om ondersteuning te kunnen bieden aan beide scenario's. Vervolgens moet het mogelijk zijn om gestandaardiseerd een VLAN verdeling op basis van locatie en/of functie van een subnet. Ook moeten de subnetten volgens een bepaalde methode opgebouwd kunnen worden. Bijvoorbeeld gateway adressen van een subnet en ip adressen van apparatuur met een bepaalde functie moeten ingesteld kunnen worden.

3.1.2 IPv6 ondersteuning in service tasks.

Om op een standaard manier een netwerk te bouwen maakt netYCE gebruik van service tasks. In de huidige situatie worden er met behulp van deze service tasks onder andere IP-adressen, subnetten en VLAN's toegekend aan routers en switches.

De ondersteuning voor IPv6 in de service tasks kan relatief eenvoudig worden gerealiseerd omdat de functionaliteit weinig verschilt met de huidige situatie.

Om migratie stappen van IPv4 naar een dual stack omgeving mogelijk te maken moet het mogelijk zijn om door middel van service tasks IPv4 subnetten op te zoeken binnen huidige services en hier een IPv6 subnet naast te creëren met hetzelfde VLAN id.

3.1.3 IPv6 ondersteuning in parameter groups

Binnen de huidige functionaliteit is het mogelijk om Nodes, Regio's en beheerdomeinen een set van parameters toe te kennen. Hierin kunnen onder andere IP-adressen van bijvoorbeeld RADIUS of DHCP servers opgeslagen worden. Deze adressen kunnen vervolgens weer worden gebruikt als parameters in templates.

Om dual stack netwerken goed te ondersteunen moet er vanuit gegaan worden dat deze servers ook een IPv6 adres krijgen die geconfigureerd wordt op routers en switches.

Binnen de parameter groups is het daarom wenselijk om IPv6 parameters te kunnen modelleren. Voor elke IPv4 parameter moet er een IPv6 vermelding kunnen zijn.

3.1.4 Koppelen van IPv4 adressen met IPv6 adressen.

In gesprekken met de Rabobank is ter sprake gekomen dat het wellicht zinvol is om een koppeling te maken tussen de huidig gebruikte IPv4 adressen en de nieuw in te voeren IPv6 adressen. [SURF-IPv6] geeft dit ook als optie aan wanneer er geen gebruik gemaakt wordt van een nummerplan. Hoewel het op het eerste gezicht een optie lijkt, zal een IPv4 adres wat verwerkt is in een IPv6 adres niet als zodanig herkenbaar zijn. Omdat er in netYCE vaak gebruik gemaakt wordt van een nummerplan en om de complexiteit nietodeloos te verhogen wordt het koppelen van IPv4 en IPv6 adressen buiten de scope gelaten.

3.2 IPv6 ondersteuning in het Build proces

3.2.1 IPv6 prefix beheren in een client

In de huidige situatie worden IPv4 supernetten toegekend aan een Client. Deze supernetten kunnen vervolgens door middel van het toegekende IP plan verdeeld worden in 1 of meerdere subnetten.

Binnen IPv6 worden prefixen toegekend aan elke unieke aansluiting op internet. Deze prefixen worden voor Europa geregistreerd bij het Ripe NCC. Per [RIPE589] wordt er aan elke unieke aansluiting op internet een /48 prefix toegekend welke de eindgebruiker zelf kan verdelen. Voor de SoHo omgevingen kan dit worden gezien als een overkill maar voor grote bedrijven, die meerdere locaties hebben, echter met maar 1 internet aansluiting kan deze /48 prefix verdeeld worden naar functie en locatie (zie ook 3.1.1).

Het Internet Assigned Numbers Authority (IANA) is verantwoordelijk voor het uitdelen van IP-adressen zowel van IPv4 als van IPv6. Het IANA heeft voor IPv6 2 registraties.

In 1 registratie, [IANA-IPv6-GUAA], worden toegekende en gereserveerde adresblokken bijgehouden. Voor de toegekende blokken wordt alleen de Regional Internet Registry (RIR) geregistreerd. Binnen Europa is dit het RIPE NCC. Voor de functionaliteit binnen netYCE is het wenselijk dat alleen de publieke adresblokken (2000::/3) met de status ALLOCATED toegekend kunnen worden aan Clients. Dit voorkomt fouten ten aanzien van routeringen die niet werken op internet.

Ook wordt er bij het IANA een special purpose registratie, [IANA-IPv6-Reg], bijgehouden. Binnen deze registratie worden bepaalde speciale toepassingen met hun bijbehorende adresblokken geregistreerd. Om fouten te voorkomen is het wenselijk dat de publieke adresblokken uit deze registratie niet toegekend kunnen worden aan een client. Prefixen uit de private reeks (fc00::/7) moet wel kunnen.

Voor test, lab en opleiding situaties zou het mogelijk wenselijk kunnen zijn om de prefix 2001:db8::/32 [RFC3849] toe te kennen aan een client. Dit zou gerealiseerd kunnen worden door middel van een instelling van de software.

3.2.2 IPv6 prefix beheren in een site

Vanwege de hoeveelheid beschikbare adressen en de richtlijnen (zoals [RIPE589]) voor het toekennen van prefixen aan klanten is het niet ondenkbaar dat een prefix specifiek voor een bepaalde locatie bestemd is. Ook is het mogelijk dat een /48 prefix voor een klant verdeeld moet worden over verschillende locaties. [SURF-IPv6] gaat bijvoorbeeld uit van een /48 aan een organisatie. In dit White paper wordt vervolgens een onderverdeling gemaakt op basis van locatie en functie.

Vanwege de richtlijnen voor de toewijzing moet het mogelijk zijn om per site een prefix toe te kennen of te verwijderen. Deze prefix moet uit de aan de client toegekende prefix komen. Het toekennen of verwijderen van de prefix moet ofwel handmatig, ofwel door middel van een service task, ofwel op basis van het IPv6-nummerplan mogelijk zijn.

3.2.3 IPv6 subnet beheren in een service

Binnen de huidige versie van netYCE is het mogelijk om in een service een IPv4 subnet toe te voegen of te verwijderen. Ook is het mogelijk om deze subnetten te voorzien van een template en een VRF.

Binnen de huidige versie is het niet mogelijk de naam van een subnet te veranderen. De naam van een subnet binnen de huidige versie is gelijk aan de functie. Met name voor de service provider en managed hosting provider is dit echter wel wenselijk. Het is dus aan te raden om deze functionaliteit op te nemen in netYCE. Voor IPv4 subnetten valt dit buiten de scope van het project. Voor IPv6 kan dit meegenomen worden.

3.2.4 IPv6 adres veranderen op een interface

Met IPv6 is het mogelijk om meerdere IP-adressen per interface te hebben. Een voorbeeld hiervan is een router welke op de management interface een ULA ([RFC4193]) en een global IPv6 adres heeft. Wat in sommige gevallen ook aangeraden wordt is het toekennen van een verkort adres. [SURF-IPv6] sectie 5.6 geeft dit bijvoorbeeld aan voor de DNS servers. Het verkorte adres is in dit voorbeeld 2001:db8:1234:a::53. 53 is het poortnummer van het DNS protocol en hierdoor makkelijk te onthouden. Tijdens de analyse zijn adressen van routers gezien met een verkort adres van 2001:db8:xyz::22. Naar dit adres was SSH verkeer mogelijk.

Door de vele mogelijkheden is het wenselijk om elke interface die voorzien wordt van IPv6 adressen te kunnen wijzigen. Niet alleen de management interface maar ook gebruikersinterfaces.

3.2.5 IPv6 subnet beheren op een interfaces

Net als bij IPv4 worden bij IPv6 de laag 2 interfaces van een switch niet voorzien van een IP adres, maar transporteert een interface 1 of meerdere VLAN's.

Hoewel er geen verschil is tussen de IPv4 en IPv6 subnetten voor switches, is het wel aan te raden om iets flexibeler om te gaan met het koppelen van een subnet aan een interface. Nu is het met de grafische interface alleen maar mogelijk om een VLAN welke bestaat in een service te koppelen aan interface. Wanneer deze interface gekoppeld is met een andere interface van een andere router/switch kan het VLAN ook gekoppeld worden aan de gelinkte interface. Echter het is niet mogelijk om vanuit een service waarin het VLAN niet bestaat een VLAN toe te kennen. Met het oog op flexibiliteit is het wenselijk om, wanneer een VLAN al gekoppeld is aan een interface van een router/switch/firewall, deze ook op andere interfaces van dit apparaat te koppelen.

3.3 IPv6 ondersteuning in het Operate proces

3.3.1 Verbinden naar een door netYCE beheert apparaat met IPv6

Vanwege de langzame implementatie van IPv6 in het algemeen en de huidig in gebruik zijnde applicaties op netwerken is het aannemelijk dat de komende 5 tot 10 jaar dual stack netwerken in gebruik zijn. Dit houdt in dat alle netwerkapparatuur en end user devices deze periode met zowel IPv4 als IPv6 adressen geconfigureerd gaan worden.

Dit geldt dan ook voor het beheer over de routers, switches en firewalls. De management interface van deze apparatuur zal zowel een IPv4 als een IPv6 adres krijgen. Op beide adressen moet management mogelijk zijn en netYCE zal ook op beide adressen moeten proberen te verbinden. De meest eenvoudige methode hiervoor is verbinden op de DNS naam. Wanneer er in de DNS zowel een A record (IPv4) als een AAAA record (IPv6) bestaat zal geprobeerd worden via beide protocollen verbinding te maken. Door de implementatie van [RFC6555] zal de software zelf uitzoeken welk protocol gebruikt wordt. Hierbij heeft IPv6 de voorkeur echter wanneer IPv4 sneller reageert zal IPv4 gebruikt worden.

Hoewel het mogelijk is om zelf een methode te ontwerpen heeft het gebruik van DNS de voorkeur. Voorwaarde is hierbij wel dat de DNS goed bijgewerkt is. Het is wenselijk om in ieder geval een methode te ontwerpen welke gebruikt wordt wanneer de DNS niet beschikbaar is.

3.3.2 ping6 en traceroute6

Om te kijken of een netwerkapparaat bereikbaar is wordt veel gebruik gemaakt van ping en traceroute. Deze beide applicaties maken gebruik van ICMP. Binnen de huidige versie van netYCE is het mogelijk om te kijken of een apparaat te bereiken is met ping.

Voor IPv6 zijn ping en traceroute ook ontworpen en maken gebruik van ICMPv6. ICMPv6 is in wezen een aanpassing van ICMP.

Omdat de netYCE servers vaak in een apart segment van het beheer-netwerk gepositioneerd worden bij klanten kan het nuttig zijn om de mogelijkheid te bieden deze beide programma's vanuit netYCE te kunnen bedienen. De huidige versie van netYCE biedt alleen ondersteuning voor ping echter traceroute is hier een zinvolle aanvulling op.

3.3.3 Integraties met externe systemen

Uiteindelijk is het wenselijk dat er integraties komen met externe systemen over IPv6. Hierbij wordt voornamelijk gedacht aan IPAM systemen, zoals Infoblox. Communicatie voor de integratie is veelal geen probleem over IPv6. Echter om de scope van het project overzichtelijk te houden is het verstandig deze stap op dit moment buiten beschouwing te laten.

3.4 *Gevolgen voor de software*

3.4.1 *Database aanpassing*

Binnen de huidige database is niet voorzien in de opslag van de IPv6 prefixen, subnetten en adressen. Hiervoor zal een database aanpassing noodzakelijk zijn welke voorziet in de opslag van deze gegevens.

Net als bij de huidige database, zal veel van de business logic in de software worden opgelost en niet worden gecontroleerd in de database. Hierbij bestaat nog steeds het gevaar dat er corruptie optreedt bij de data. Het is echter niet logisch de IPv6 data wel te controleren in de database en de andere data niet.

3.4.2 *Frontend aanpassingen*

De gewenste functionaliteit zal resulteren in verschillende aanpassingen aan de huidige front end van de applicatie. Met name bij het Design- en het Build proces is dit aan de orde.

3.4.3 *Backend aanpassingen*

Bij de backend zal de methodiek voor het verbinden met apparatuur moeten veranderen.

Ook zal het gedeelte van de backend wat de service tasks uitvoert uitgebreid moeten worden met de IPv6 functionaliteit.

3.5 *Gevolgen voor de installatie*

Omdat IPv4 en IPv6 niet met elkaar kunnen communiceren is het noodzakelijk dat de netYCE server(s) in een implementatie bij de klant geconfigureerd worden met een IPv6 adres. Uiteraard moet het netwerk waar deze server(s) op aangesloten zijn ook voorzien zijn van IPv6 functionaliteit.

Verder kan het zijn dat de Apache configuratie aangepast moet worden om IPv6 te ondersteunen. Wanneer er op basis van virtual hosts gewerkt wordt met Apache is er geen probleem. Echter wanneer er op basis van IP adres gecommuniceerd wordt met de server moet de configuratie aangepast worden.

Het is ook mogelijk dat het netwerk dual stack uitgevoerd wordt maar dat het beheer van het netwerk alleen maar gebruik maakt van IPv4. In dit geval hoeft er niets te worden aangepast.

4 *MoSCoW analyse*

Must have:

- Toevoegen en verwijderen van IPv6 prefixen bij een client.
- Toevoegen en verwijderen van een IPv6 prefix plan.
- Toevoegen en verwijderen van een IPv6 subnet plan.
- Gebruik van IPv6 zaken in service tasks.
 - Toekennen van een IPv6 adres aan een interface.
 - Verwijderen van een IPv6 adres van een interface.
 - Toekennen van een IPv6 subnet aan een interface of link.
 - Verwijderen van een IPv6 subnet van een interface of link.
 - Opzoeken van een vrij IPv6 adres.
 - Opzoeken van het VLAN id van een IPv4 subnet.
 - Toekennen van een IPv4 VLAN id aan een IPv6 subnet.

Should have:

- Toevoegen en verwijderen van IPv6 prefixen bij een site.
- Mogelijkheden om IPv6 server adressen op te slaan.
- Mogelijkheid om IPv6 adressen van een node te veranderen.
- In het modelleren custom parameters op te nemen en deze op te slaan als geldig IPv6 adres.

Could have:

- Het wel of niet kunnen gebruiken van de documentatie reeks 2001:db8::/32 [RFC3849] als prefix.
- Gebruik van IPv6 zaken in service tasks
 - Veranderen van een IPv6 toegekend adres.
 - Opzoeken van een random vrij IPv6 adres.
- De huidige ping functie geschikt maken voor IPv6, alsmede ook gebruik kunnen maken van traceroute

Won't have:

- Integraties met externe systemen.
- Mappen van IPv4 adressen op IPv6 adressen.

5 Conclusie en aanbevelingen

Om de door de markt gevraagde ondersteuning van IPv6 in netYCE afdoende te ondersteunen zullen de Must Have's uit de MoSCoW analyse nog voor het einde van 2014 ingevoerd moeten zijn. Om flexibel genoeg om te kunnen gaan met de IPv6 ondersteuning is het noodzakelijk dat ook de Should Have's en Could have's geïmplementeerd worden. Q1 2015 zal dit gerealiseerd moeten zijn.

Doordat de functionaliteit uitgebreid wordt, maar de huidige functionaliteit niet aangepast wordt, zijn de risico's erg laag. De ontwikkeling kan parallel aan het huidige ontwikkel traject van netYCE. De huidige architectuur hoeft niet aangepast te worden.

Gezien de vraag vanuit de markt en de mogelijkheden van de ontwikkeling is het dan ook aan te bevelen, zo snel mogelijk te beginnen met de ontwikkeling en de invoer van IPv6.

6. Bronvermelding en literatuurlijst

6.1 Bronnen

- [1] Google IPv6 statistics: <http://www.google.com/intl/nl/ipv6/statistics.html>
- [2] Internet Society Tech Matters: <http://internetsociety.org/blog/...>

6.2 Referenties

- [[SURF-IPv6](#)] Surfnets Whitepaper IPv6-nummerplan opstellen.
- [[IANA-IPv6-GUAA](#)] IPv6 Global Unicast Address Assignments
- [[IANA-IPv6-Reg](#)] IANA IPv6 Special-Purpose Address Registry
- [[RIPE589](#)] IPv6 Address Allocation and Assignment Policy, May 2013
- [[RFC1918](#)] Address Allocation for Private Internets, Feb 1996
- [[RFC3849](#)] IPv6 Address Prefix Reserved for Documentation , July 2004
- [[RFC4193](#)] Unique Local IPv6 Unicast Addresses, Oct 2005
- [[RFC4291](#)] IP Version 6 Addressing Architecture, Feb 2006
- [[RFC6555](#)] Happy Eyeballs: Success with Dual-Stack Hosts, April 2012

Bijlage III: Plan van Aanpak

Plan van Aanpak

IPv6 implementatie in NetYCE

Derco Sportel

1548287

Colofon

| | |
|--------------------|--|
| | NetYCE B.V. Postbus 320 1380 AH Weesp, Nederland Business Centre Muiderpoort Leeuwendveldseweg 5n 1382 LV Weesp, Nederland |
| Auteur | Derco Sportel <i>Network Consultant</i> ☎ +31 6 1418 1361 ✉ derco.sportel@netyce.com 🌐 derco.sportel |
| Docentbegeleider | Kees Uiterwijk <i>Hogeschooldocent, Systeembeheer</i> ☎ +31 88 481 8184 ✉ kees.uitervijk@hu.nl 🌐 kees.uitervijk |
| Bedrijfsbegeleider | Ronny Lam <i>Solution Architect</i> ☎ +31 6 3976 6145 ✉ ronny.lam@netyce.com 🌐 ronny_lam |
| Versie | 1.0 |

Document

Document geschiedenis

| Versie | Datum | Auteur | Notities |
|--------|--------------|---------------|-------------------------------|
| 0.1 | 8 juli 2014 | Derco Sportel | Eerste opzet |
| 0.2 | 23 juli 2014 | Derco Sportel | Kleine tekstuele aanpassingen |
| 1.0 | 4 aug 2014 | Derco Sportel | Definitief. |

Distributie

| Name | 0.1 | 0.2 | 1.0 |
|---------------------|-----|-----|-----|
| Derco Sportel | ✓ | ✓ | ✓ |
| Eric Yspeert | | | ✓ |
| Ronny Lam | ✓ | ✓ | ✓ |
| Kees Uiterwijk (HU) | | | ✓ |
| Gerald Ovink (HU) | | | ✓ |
| J de Doelder (CvT) | | | ✓ |

Inhoudsopgave

| | |
|----------------------------------|-------|
| Colofon | 2 |
| Document | 3 |
| Document geschiedenis | 3 |
| Distributie | 3 |
| 1. Inleiding | 5 |
| 1.1 Bedrijfsachtergrond | 5 |
| 1.2 Aanleiding van het project | 5 |
| 1.3 Doel van het project | 5 |
| 2. Project definitie | 6 |
| 2.1 Probleembeschrijving | 6 |
| 2.2 Projectopdracht | 6 |
| 2.3 Op te leveren producten | 6 |
| 2.4 Scope van het project | 6 |
| 2.5 Succesfactoren en risico's | 6 |
| 3. Projectplan | 7 |
| 3.1 Aanpak | 7 |
| 3.2 Voorlopig verwachte planning | 8 |

1. Inleiding

Voor u ligt het Plan van Aanpak voor de uitvoering van de afstudeeropdracht voor IPv6 ondersteuning in netYCE voor de studie Deeltijd Informatica aan de Hogeschool Utrecht.

De opdracht wordt uitgevoerd voor het bedrijf NetYCE, die in hun gelijknamige product IPv6 ondersteuning wensen.

1.1 Bedrijfsachtergrond

NetYCE B.V. is een klein bedrijf dat opgericht is in 2011 en zich specialiseert in het geautomatiseerd doorvoeren van wijzigingen voor middelgrote en grote netwerken van routers, switches en in beperkte mate firewalls. Eindapparatuur, zoals servers, werkstations, printers etc. worden buiten beschouwing gelaten.

Het product dat hiervoor gebruikt wordt heet netYCE. Dit product is in 2001 ontwikkeld voor de Rabobank ICT groep als tool om een grootschalige migratie te realiseren. Om de continuïteit te waarborgen heeft de Rabobank de auteursrechten overgedragen en geholpen NetYCE B.V. op te richten.

Naast netYCE wordt op dit moment een nieuw product genaamd ontwikkeld, genaamd labYCE. Dit product richt zich op het configuratiebeheer en de gebruikersplanning van test- en acceptatie netwerken.

1.2 Aanleiding van het project

Sinds de World IPv6 Launch Day op 6 juni 2012 is het aandeel IPv6 verkeer op internet gegroeid van 0,65% naar inmiddels 4,1%^[1]. Bij NetYCE komt ook steeds vaker in de gesprekken met (toekomstige) klanten de vraag of en hoe IPv6 ondersteunt wordt.

Omdat netYCE op het moment geen IPv6 ondersteunt in het platform heeft de directie in november 2013 de wens uitgesproken om IPv6 ondersteuning in het 3^e kwartaal van 2014 gereed te hebben voor een beta release naar de klanten. Hiermee kan de toekomstvastheid van het product netYCE gewaarborgd blijven.

1.3 Doel van het project

Het doel van het project is, om de huidige functionaliteit van de netYCE applicatie uit te breiden met ondersteuning voor IPv6 en dualstack netwerken.

2. Project definitie

2.1 Probleembeschrijving

In de huidige versie van netYCE mist de ondersteuning voor het automatiseren van IPv6 en dual stack netwerken. Om de huidige markt te kunnen blijven bedienen, vindt de directie het noodzakelijk dat deze ondersteuning zo snel mogelijk, uiterlijk eind 2014, mogelijk is.

2.2 Projectopdracht

De opdracht is het realiseren van IPv6 ondersteuning in netYCE, waarmee het mogelijk wordt om netwerken van routers en switches geautomatiseerd te configureren met een IPv6 of dual stack configuratie.

De opdracht bestaat uit de volgende deelopdrachten:

- Onderzoek welke functionaliteit netYCE moet bieden voor IPv6 ondersteuning;
- Onderzoek hoe IPv6 op dit moment in de markt gebruikt wordt;
- Maak naar aanleiding van het onderzoek een functioneel ontwerp;
- Maak naar aanleiding van het functioneel ontwerp een technisch ontwerp;
- Laat in een Proof of Concept zien, dat het mogelijk is om:
 - Een functioneel netwerkontwerp op basis van een dual stack configuratie te ontwerpen;
 - Een technisch netwerkontwerp van dit functioneel ontwerp te creëren als netwerk;
 - Het fysieke netwerk geautomatiseerd te voorzien van een dual stack configuratie, op basis van de abstractie.

2.3 Op te leveren producten

Het project zal de volgende deelproducten opleveren:

- Een analyse rapportage van de huidige situatie en de gewenste situatie;
- Een functioneel ontwerp benodigd om tot de gewenste situatie te komen;
- Een technisch ontwerp;
- Een Proof of Concept opstelling;
- Een implementatie voorstel.

2.4 Scope van het project

De scope van dit project beperkt zich tot alle, in de projectopdracht en op te leveren producten genoemde zaken. Om het project overzichtelijk te houden zullen alle niet-gerelateerde zaken buiten beschouwing worden gelaten. Uitgezonderd het implementatievoorstel, zal de implementatie buiten de scope vallen.

2.5 Succesfactoren en risico's

Vanwege de positie van de opdrachtnemer binnen het bedrijf, bestaat het risico dat er te weinig tijd is voor dit project.

Het project wordt als een succes beschouwd wanneer de PoC geslaagd is.

3. Projectplan

3.1 Aanpak

Dit project zal worden opgebouwd uit 4 fases. Een 5^e fase (de implementatiefase) valt gedeeltelijk buiten de scope van dit project.

Fase 1 zal bestaan uit een oriëntatie over het onderwerp en de grove uitlijning van het project. Deze fase is reeds afgerond in maart 2014.

Fase 2 zal bestaan uit de analyse van de behoeften, het onderwerp en de huidige situatie. Als afronding van deze fase wordt dit document opgeleverd, alsmede een analyse rapportage. Hiervoor heeft inmiddels een literatuurstudie plaats gevonden, alsmede een analyse van de huidige werking van de applicatie. De beoogde einddatum voor deze fase is 31 juli 2014

In fase 3 wordt de architectuur beschreven. Hierin zullen de functionele eisen opgesteld worden, waaruit een usecase model volgt. Dit geheel wordt samengevoegd tot een functioneel ontwerp. De beoogde einddatum voor deze fase is medio augustus 2014.

In fase 4 zal het functioneel ontwerp uitgewerkt worden in een technisch ontwerp. Ook zal hier in een proof of concept opstelling de werking aangetoond worden. Medio september moet deze fase zijn afgerond.

Fase 5 is de implementatie van het technisch ontwerp in de applicatie. Hiertoe zal wel een implementatievoorstel binnen de scope van het project vallen, echter de werkelijke implementatie in de applicatie zal buiten de scope van het project vallen.

Uiterlijk 30 september moet het binnen scope gedeelte van deze fase afgerond zijn. Eind oktober moet een beta release klaar zijn, om richting de klanten te versturen.

3.2 Voorlopig verwachte planning

| Fase 1: Oriëntatie fase | Periode | |
|---|------------------------|--|
| Probleemformulering | Maart 2014 | |
| Opleveren Voorstel | Maart 2014 | |
| Fase 2: Analyse fase | | |
| Analyse en omschrijving van de huidige situatie | Mei 2014 | |
| Literatuur onderzoek | April/Mei 2014 | |
| Opstellen project doelstellingen | Juli 2014 | |
| Opstellen project deelopdrachten | Juli 2014 | |
| Opleveren analyse rapport | Juli 2014 | |
| Opleveren Plan van Aanpak | Juli 2014 | |
| Fase 3: Architectuur fase | | |
| Opstellen functionele eisen | Juli 2014 | |
| Uitwerken Use Case model | Juli 2014 | |
| Opleveren Functioneel ontwerp | Augustus 2014 | |
| Fase 4: Design fase | | |
| Uitwerken Usecases | Augustus 2014 | |
| Opstellen database model | Augustus 2014 | |
| Opleveren technisch ontwerp | Augustus 2014 | |
| Proof of concept opstelling realiseren | 15 september 2014 | |
| Fase 5: Implementatie fase | | |
| Opstellen implementatievoorstel | 30 september 2014 | |
| <i>Backend implementatie (buiten scope)</i> | <i>31 oktober 2014</i> | |
| <i>Frontend implementatie (buiten scope)</i> | <i>31 oktober 2014</i> | |
| | | |

Bijlage IV: Afstudeercontract

Contract afstudeeropdracht
Institute for ICT
Nijenoord 1, 3552 AS, UTRECHT



NB: Dit contract dient te worden opgenomen als vast onderdeel van het plan van aanpak

| | |
|-------------------------------|----------------------------|
| Datum | : __ september 2014 |
| Naam student | : Derco Sportel |
| Opleiding | : Deeltijd |
| Adres student | : Hessenweg 127A |
| Postcode / Woonplaats student | : 3731JG De Bilt |
| Studentnummer | : 1548287 |
| Telefoonnummer privé | : 06-14181361 |
| E-mailadres | : derco.sportel@netyce.com |

| | |
|-----------------------------------|------------------------|
| Naam bedrijf (afstuderen) | : NetYCE B.V. |
| Adres bedrijf | : Leeuwendeldseweg 5n |
| Postcode / Woonplaats bedrijf | : 1382LV Weesp |
| Naam bedrijfsbegeleider | : Ronny Lam |
| Telefoonnummer bedrijfsbegeleider | : 06-39766145 |
| E-mailadres bedrijfsbegeleider | : ronny.lam@netyce.com |

| | |
|---------------------------------------|-----------------------|
| Beoogde datum van afstuderen | : oktober 2014 |
| Geheimhouding geaccordeerd door HU op | : niet van toepassing |

Ondergetekenden verklaren akkoord te gaan met de inhoud van aangehecht plan van aanpak.

Handtekeningen

Student :
 Docentbegeleider :
 Bedrijfsbegeleider¹ :

Ondertekende versie op 3 september 2014 ingeleverd bij de Hogeschool Utrecht.

¹Door ondertekening van dit formulier verklaart de bedrijfsbegeleider (en eventuele mede-begeleiders) over voldoende kennis te beschikken, op minimaal HBO-niveau, om de afstudeerder te begeleiden.

Bijlage V: Functioneel ontwerp

Functioneel ontwerp

IPv6 implementatie in NetYCE

Derco Sportel

1548287

Colofon

NetYCE B.V.

Postbus 320
1380 AH Weesp, Nederland

Business Centre Muiderpoort
Leeuwendveldseweg 5n
1382 LV Weesp, Nederland

Auteur

Derco Sportel

Network Consultant

☎ +31 6 1418 1361

✉ derco.sportel@netyce.com

🌐 derco.sportel

Docentbegeleider

Kees Uiterwijk

Hogeschooldocent, Systeembeheer

☎ +31 88 481 8184

✉ kees.uitervijk@hu.nl

🌐 kees.uitervijk

Bedrijfsbegeleider

Ronny Lam

Solution Architect

☎ +31 6 3976 6145

✉ ronny.lam@netyce.com

🌐 ronny_lam

Versie

1.0

Document

Document geschiedenis

| Versie | Datum | Auteur | Notities |
|--------|--------------|---------------|------------------------|
| 0.1 | 25 juli 2014 | Derco Sportel | Eerste opzet. |
| 0.2 | 26 aug 2014 | Derco Sportel | Toevoegen usecases |
| 0.3 | 30 aug 2014 | Derco Sportel | Tekstuele aanpassingen |
| 0.4 | 2 sept 2014 | Derco Sportel | Tekstuele aanpassingen |
| 1.0 | 13 sept 2014 | Derco Sportel | Definitive versie |

Distributie

| Name | 0.1 | 0.2 | 0.3 | 0.4 | 1.0 |
|---------------------|-----|-----|-----|-----|-----|
| Derco Sportel | ✓ | ✓ | ✓ | ✓ | ✓ |
| Eric Yspeert | | | ✓ | ✓ | ✓ |
| Ronny Lam | ✓ | ✓ | ✓ | ✓ | ✓ |
| Kees Uiterwijk (HU) | | | | | ✓ |
| Gerald Ovink (HU) | | | | | ✓ |
| J de Doelder (CvT) | | | | | ✓ |

Inhoudsopgave

| | |
|--|----|
| Colofon..... | 2 |
| Document..... | 3 |
| Document geschiedenis..... | 3 |
| Distributie..... | 3 |
| 1 Inleiding..... | 5 |
| 2 IPv6 toepassingen en gebruik..... | 6 |
| 2.1 IPv4 vs IPv6..... | 6 |
| 2.2 IPv6 notatie en prefix..... | 6 |
| 3 Functionele eisen..... | 7 |
| 3.1 Architectuurbeschrijving..... | 7 |
| 3.2 IPv6 ondersteuning in het Design proces..... | 7 |
| 3.2.1 IPv6 prefix plan..... | 7 |
| 3.2.2 IPv6 ondersteuning in service tasks..... | 8 |
| 3.2.3 IPv6 ondersteuning in parameter groups..... | 8 |
| 3.3 IPv6 ondersteuning in het Build proces..... | 8 |
| 3.3.1 IPv6 prefix beheren in een client..... | 8 |
| 3.3.2 IPv6 prefix beheren in een site..... | 9 |
| 3.3.3 IPv6 subnet beheren in een service..... | 9 |
| 3.3.4 IPv6 adres veranderen op de management interface..... | 9 |
| 3.3.5 IPv6 subnet beheren op een interface..... | 9 |
| 3.4 IPv6 ondersteuning in het Operate proces..... | 10 |
| 3.4.1 Verbinden naar een door netYCE beheert apparaat met IPv6..... | 10 |
| 3.4.2 Het pingen van een door NetYCE beheert apparaat op een IPv6 adres (ping6)..... | 11 |
| 3.4.3 Het opzoeken van 1 of meerdere IPv6 adressen/prefixen/subnetten..... | 11 |
| 4 Use cases en gebruikersrollen..... | 12 |
| 4.1 Gebruikersrollen..... | 12 |
| 4.1.1 Ontwerper (Designer)..... | 12 |
| 4.1.2 Engineer..... | 12 |
| 4.1.3 Operator..... | 12 |
| 4.2 U-01: Beheren van een IPv6 prefix plan..... | 12 |
| 4.3 U-02: Beheren van een IPv6 subnet plan..... | 12 |
| 4.4 U-03: Gebruik van IPv6 in service tasks..... | 12 |
| 4.5 U-04: Beheren van IPv6 parameters in parameter groups..... | 13 |
| 4.6 U-05: Beheren van IPv6 prefixen in een client..... | 13 |
| 4.7 U-06: Beheren van IPv6 prefixen in een site..... | 13 |
| 4.8 U-07: Beheren van IPv6 subnetten in een service..... | 13 |
| 4.9 U-08: Beheren van het IPv6 adres op de management interface..... | 13 |
| 4.10 U-09: Beheren van een IPv6 subnet op een interface of link..... | 13 |
| 4.11 U-10: Pingen van 1 of meerdere IPv6 adressen..... | 13 |
| 5 Bronnen en referenties..... | 14 |

1 Inleiding

Dit document beschrijft het functionele ontwerp voor IPv6 ondersteuning in de netYCE applicatie. De functionele eisen en de usecase beschrijvingen worden hierin uitgewerkt.

In dit document wordt uitgegaan van dual stack implementaties en niet van een IPv6 only implementatie.

Tenzij expliciet anders vermeld zijn alle IP-adressen (IPv4 en IPv6) vervangen voor IP-adressen uit de documentatie prefix ([RFC5737] voor IPv4 en [RFC3849] voor IPv6).

2 IPv6 toepassingen en gebruik

In computer netwerken wordt het Internet Protocol gebruikt om te communiceren tussen verschillende nodes. In 1980 is de huidige versie van IP (IPv4) ontworpen. Ten tijde van dit ontwerp is nooit voorzien in de hoeveelheid apparaten die nu online zijn. Omdat IPv4 niet voldoende adressen bevat om alle nodes van een adres te voorzien, is in december 1998 een nieuwe versie van IP gelanceerd. Dit was IP versie 6 ofwel IPv6 ([RFC2460]). In oudere documentatie wordt er soms ook nog gesproken over IPng (IP next generation).

Het Internet Stream Protocol versie 2 (ST2 [RFC1819]) heeft als experimenteel protocol als IP versie 5 bestaan. Dit protocol is nooit in productie geweest, al zijn veel van de mogelijkheden van ST2 verwerkt in andere technieken. Hierdoor lijkt het alsof versie 5 is overgeslagen, echter dit is niet het geval.

2.1 IPv4 vs IPv6

IPv4 en IPv6 zijn niet in staat om met elkaar te communiceren. Door de opbouw van de adressen en het verschil in werking tussen deze 2 protocollen is dit niet mogelijk. Wanneer een host, alleen maar gebruik kan maken van IPv6 en toch met een IPv4 host moet communiceren is dit alleen maar mogelijk door middel van een NAT64 gateway.

IPv4 adressen bestaan uit 32 bits. Hier zijn er theoretisch ongeveer 4,3 miljard adressen mogelijk. Dit aantal is theoretisch, omdat er grote blokken gereserveerd zijn voor speciaal gebruik of experimentele doeleinden. Hierdoor blijven er iets meer dan 3,7 miljard adressen over. Een schatting van de Cisco uit 2013 geeft aan dat er ongeveer 8,7 miljard computers, smartphones, tablets en andere netwerkapparatuur aangesloten is op internet in 2012^[1]. Dit is meer dan het aantal beschikbare adressen. Door het gebruik van NAT (Network Address Translation) is het nu mogelijk om al deze apparaten toch te verbinden met internet.

IPv4 adressen worden weergegeven in 4 numerieke blokken met punten er tussen. Elke 8 bits van een adres is zo'n numeriek blok. Een voorbeeld is 192.0.2.2

IPv6 adressen bestaan uit 128 bits, waar mee er ongeveer $3,4 * 10^{38}$ adressen mogelijk zijn. Dit is voldoende om de komende jaren te groeien.

2.2 IPv6 notatie en prefix

IPv6 adressen worden weergegeven in 8 hexadecimale blokken van 16 bits. Een voorbeeld van een IPv6 adres is 2001:db8:fc90:1::199a. Vanwege de leesbaarheid worden voorloop nullen in 16 bits velden niet geschreven. [RFC5952] beschrijft in detail hoe IPv6 adressen weergegeven moeten worden.

Wanneer een klant gebruik wil maken van IPv6 krijgt deze afhankelijk van de hoeveelheid locaties een adresreeks toegewezen. Dit wordt een prefix genoemd. In Europa krijgt elke locatie in beginsel een /48 prefix toegewezen. Dit houdt in dat de eerste 48 bits door de provider toegewezen wordt. De overige 80 bits kunnen gebruikt worden door de klant. Een voorbeeld van een prefix is 2001:db8:fc90::/48. Deze prefix wordt standaard ingedeeld in /64 subnetten. De 16 bits die vervolgens beschikbaar zijn worden gebruikt om locatie en functiescheiding te realiseren.

3 Functionele eisen

3.1 Architectuurbeschrijving

de netYCE applicatie bestaat op dit moment uit een RedHat Enterprise Linux of CentOS server met een Apache webserver, een MySQL of MariaDB database server en Perl. Alle huidige releases van het OS en de software ondersteunen op dit moment IPv6. Voor het uitvoeren van bepaalde taken maakt netYCE gebruik van de aanwezige DNS servers in het netwerk. De DNS servers in deze netwerken ondersteunen allemaal het gebruik van IPv6.

Om netwerkkapparatuur met netYCE te beheren met behulp van IPv6 zullen de netYCE servers voorzien moeten worden van een IPv6 adres. De te beheren netwerkkapparatuur moet bereikt kunnen worden door de netYCE servers.

3.2 IPv6 ondersteuning in het Design proces

3.2.1 IPv6 prefix plan

Binnen een netwerk ontwerp is het planmatig inzetten van IP-adressen een must. Netwerk designers krijgen met IPv6 per verbinding naar internet, afhankelijk van de policy van de RIR, grote blokken adressen toegewezen. Het beleid van het RIPE NCC is bijvoorbeeld dat een End Site een /48 adresblok toegewezen krijgt. Een End Site is in deze een internet aansluiting van een klant. Dit kan een thuis gebruiker zijn, maar ook een bedrijf.

De huidige versie van netYCE maakt gebruik van IP plannen, waarin een blok adressen onderverdeeld wordt in 1 of meerdere subnetten. Deze subnetten worden vervolgens op een standaard manier ingedeeld, waarmee het bijvoorbeeld mogelijk is de default gateway te bepalen, maar ook de VLAN indeling voor dit type subnet.

Voor IPv4 wordt deze methodiek nog niet vaak toegepast. Zeker wanneer er sprake is van publieke IP-adressen kan deze methodiek er in resulteren dat er veel IP-adressen ongebruikt blijven. Voor IPv6 wordt deze methodiek echter aangeraden te gebruiken. [RFC5375], [SURF-IPv6] en [Cisco-IPv6] zijn enkele voorbeelden hiervan.

Om IPv6 goed te kunnen ondersteunen is het kunnen maken van een IPv6 prefix plan noodzakelijk. Hierbij moet het mogelijk zijn de prefix in te delen naar functie of locatie en functie.

Omdat er met de komst van IPv6 voldoende adressen beschikbaar zijn is het in veel gevallen niet nodig om af te wijken van een standaard subnet grote. Binnen IPv6 zijn de subnetten standaard een /64. Hier mag van afgeweken worden echter SLAAC [RFC4862] zal dan niet meer werken. Omdat afgeweken mag worden van de /64 subnet lengte is het noodzakelijk dat netYCE dit ook ondersteunt. Echter omdat het voor IPv6 onwaarschijnlijk lijkt dat de subnetten op grote schaal anders zijn dan de standaard lengte van /64 is het wenselijk dat dit aan en uitgezet kan worden met een instelling in de software. Lengtes die sowieso ondersteunt moeten worden zijn: /64, /127 (point-to-point) en /128 (loopback interfaces).

3.2.2 IPv6 ondersteuning in service tasks

Binnen netYCE wordt gebruik gemaakt van service tasks om op een eenduidige manier een netwerk ontwerp te realiseren. Door middel van deze service tasks is het bijvoorbeeld mogelijk om nodes toe te voegen aan een netwerk, verbindingen tussen de nodes te creëren en IP-adressen of gehele subnetten toe te kennen aan interfaces.

Binnen de huidige service task functionaliteit zijn de volgende IPv4 functies beschikbaar:

- Toevoegen, verwijderen of opzoeken van een subnet aan/in een service;
- Toekennen of verwijderen van een subnet aan een interface;
- Opzoeken van het eerste vrije adres in een subnet en deze toekennen aan een interface. Het vrije adres komt uit een reeds gedefinieerde reeks binnen dit subnet.
- Toekennen van een VLAN ID aan een subnet;
- Toekennen van adres reserveringen in een subnet.

Het doel IP adres in het algemeen is het uniek kunnen adresseren en vervolgens communiceren met een node op een netwerk. Omdat dit voor IPv4 en IPv6 geldt is het in ieder geval vereist de voor IPv4 reeds bestaande service task functionaliteit ook te implementeren voor IPv6.

Om migratie van IPv4 naar dual stack netwerken mogelijk te maken moet het mogelijk zijn in de service tasks IPv4 subnetten op te zoeken binnen de huidige services en hier een IPv6 subnet naast te creëren met hetzelfde VLAN id.

3.2.3 IPv6 ondersteuning in parameter groups

Om de gebruiker van netYCE de vrijheid te geven eigen gedefinieerde variabelen op te slaan in de applicatie maakt netYCE gebruik van parameter groups. De variabelen worden door de gebruiker gedefinieerd met naam en type. Er zijn verschillende types waaronder ook IPv4 adressen.

Om IPv6 afdoende te kunnen ondersteunen is het ook belangrijk dat IPv6 als type ondersteunt gaat worden. Bij het opslaan van deze variabelen is het belangrijk dat er alleen geldige IPv6 adressen opgeslagen kunnen worden.

3.3 IPv6 ondersteuning in het Build proces

3.3.1 IPv6 prefix beheren in een client

Op dit moment is het mogelijk om een IPv4 supernet toe te kennen aan- en te verwijderen van een client in de applicatie. Dit supernet wordt gekoppeld aan een IPv4 plan zoals beschreven in sectie 3.2.1. Voor IPv6 moet het mogelijk worden om aan een client 1 of meerdere prefixen toe te kennen. De prefixen welke gekoppeld worden aan een client moeten worden gekoppeld aan een IPv6 prefix plan.

Om fouten te voorkomen moeten de prefixen geldig zijn. De huidige geldige prefixen staan beschreven in [IANA-IPv6-AS]. Verdere detaillering staan beschreven in [IANA-IPv6-GUAA] en [IANA-IPv6-Reg]. De in [IANA-IPv6-AS] *Global Unicast* en *Unique Local Unicast* reeksen mogen uitgedeeld worden. Op dit moment en waarschijnlijk in de komende jaren gaat dit om de reeksen 2000::/3 (*Global Unicast*) en fc00::/7 (*Unique Local Unicast*).

3.3.2 IPv6 prefix beheren in een site

NetYCE maakt gebruik van een hiërarchisch model. In de applicatie bestaan klanten. Een klant heeft 1 of meerdere locaties en op een locatie bestaan 1 of meerdere nodes. Met IPv6 is het gebruikelijk om elke unieke aansluiting op internet een eigen IPv6 prefix toe te kennen ([RIPE589], [ARIN-IPv6]). 1 klant kan dus meerdere IPv6 prefixen hebben.

Om binnen 1 klant de juiste IPv6 prefix te koppelen aan een locatie moet het mogelijk zijn een IPv6 prefix toe te kennen aan-, of te verwijderen van een site. De in sectie 3.3.1 genoemde regels omtrent geldigheid van de prefixen gelden ook voor een site.

3.3.3 IPv6 subnet beheren in een service

Binnen de huidige functionaliteit worden IPv4 subnetten gekoppeld aan een service. Vanuit deze service is het vervolgens mogelijk de subnetten te gebruiken op interfaces en nodes. Vanuit het service formulier kan bekeken worden op welke interfaces het subnet gebruikt wordt.

Voor IPv6 is dezelfde functionaliteit benodigd als nu al aanwezig is voor IPv4.

3.3.4 IPv6 adres veranderen op de management interface

Binnen de huidige functionaliteit kan het IPv4 adres, waarop de node beheerd wordt, aangepast worden. Hierbij is het niet mogelijk een adres te kiezen buiten het gebruikte beheer subnet. Dit houdt in dat een interface met IPv4 adres 192.0.2.1 en een subnet masker van 255.255.255.224 wel het IPv4 adres 192.0.2.30 mag krijgen maar niet het IPv4 adres 192.0.2.33.

Voor IPv6 moet het mogelijk worden om het IPv6 beheer adres te veranderen. Hierbij moet ook rekening gehouden worden met het subnet.

Het veranderen van IPv6 adressen op andere interfaces dan de management-interface wordt in de eerste implementatie buiten beschouwing gelaten.

3.3.5 IPv6 subnet beheren op een interface

Op dit moment is het mogelijk om IPv4 subnetten toe te kennen aan-, of te verwijderen van interfaces. Deze functionaliteit is ook voor IPv6 subnetten gewenst.

In de huidige situatie is het met de grafische interface alleen mogelijk om subnetten op interfaces en links te plaatsen binnen dezelfde service. Dit moet voor IPv6 mogelijk zijn binnen dezelfde site. Met een service task is dit al mogelijk.

3.4 IPv6 ondersteuning in het Operate proces

3.4.1 Verbinden naar een door netYCE beheert apparaat met IPv6

Bij het beheren en configureren van een node maakt netYCE op dit moment gebruik van het IPv4 adres. Na een migratie van een IPv4 node naar een dual stack configuratie zal het beheren van een node bij voorkeur via IPv6 moeten gaan.

[RFC6555] beschrijft hiervoor een methode alsmede de problemen bij verbinden naar een dual stack configuratie. Om te bepalen welk protocol gebruikt gaat worden moet de volgende methodiek toegepast worden:

| netYCE server | Node |
|-------------------------------|-----------------------|
| 1. =====TCP SYN, IPv6=====> | |
| 2. -----TCP SYN, IPv4-----> | (50 - 300 msec delay) |
| 3. <===TCP SYN+ACK, IPv6=== | |
| 4. <---TCP SYN+ACK, IPv4--- | |
| 5. =====TCP ACK, IPv6=====> | |
| 6. -----TCP ACK, IPv4-----> | |
| 7. -----TCP RST, IPv4-----> | |

Scenario 1: RFC6555 implementatie in netYCE, IPv4 en IPv6 bereikbaar

| netYCE server | Node |
|-------------------------------|-----------------------|
| 1. =====TCP SYN, IPv6=====X | |
| 2. -----TCP SYN, IPv4-----> | (50 - 300 msec delay) |
| 3. <---TCP SYN+ACK, IPv4--- | |
| 4. -----TCP ACK, IPv4-----> | |

Scenario 2: RFC6555 implementatie in netYCE, IPv4 bereikbaar

| netYCE server | Node |
|-------------------------------|-----------------------|
| 1. =====TCP SYN, IPv6=====> | |
| 2. -----TCP SYN, IPv4-----X | (50 - 300 msec delay) |
| 3. <===TCP SYN+ACK, IPv6=== | |
| 4. =====TCP ACK, IPv6=====> | |

Scenario 3: RFC6555 implementatie in netYCE, IPv6 bereikbaar

Wanneer bovenstaande scenario's geïmplementeerd zijn zal IPv6 eerder gekozen worden dan IPv4. De 50 tot 300 milliseconden vertraging zal ingesteld moeten worden. Dit is afhankelijk van de gemiddelde response tijd van het netwerk. Waarschijnlijk is 50 milliseconden in de meeste gevallen voldoende echter bij het beheren van netwerken met een hoge latency zal de vertraging vergroot moeten worden.

3.4.2 Het pingen van een door NetYCE beheert apparaat op een IPv6 adres (ping6)

Het is op dit moment mogelijk om apparaten, die met netYCE worden beheerd, te pingen vanuit de applicatie. Het is wenselijk om deze functionaliteit ook voor IPv6 beschikbaar te hebben.

Met de huidige functionaliteit is het mogelijk om ofwel 1, ofwel alle apparaten op een locatie te pingen. Voor IPv6 is het wenselijk om hier een selectie van 1 of meerdere apparaten te pingen, met een maximum van alle devices.

3.4.3 Het opzoeken van 1 of meerdere IPv6 adressen/prefixen/subnetten

Het is op dit moment mogelijk om een stuk tekst in netYCE te analyseren voor ip adressen. Delen tekst welke lijken op IP-adressen worden opgezocht in de database van netYCE en er wordt vervolgens een resultaat getoond waar deze adressen gebruikt worden.

Het is wenselijk deze functionaliteit ook beschikbaar te hebben voor IPv6.

4 Use cases en gebruikersrollen

4.1 Gebruikersrollen

Er zijn 3 relevante gebruikerstypen in netYCE. De overige gebruikerstypen hebben rechten voor het systeembeheer van netYCE en niet voor gebruik van netYCE.

4.1.1 Ontwerper (Designer)

De ontwerpers in netYCE zijn verantwoordelijk voor het netwerk ontwerp. De ontwerpers bepalen hoe het netwerk gemodelleerd worden. Hoe services in een netwerk architectuur werken en hoe IP-adressen verdeeld worden. De ontwerper rol heeft naast de eigen verantwoordelijkheden ook de rechten van een engineer.

4.1.2 Engineer

De engineers zijn verantwoordelijk voor de implementatie van het ontwerp. De configuraties voor de routers en switches worden door de engineers gemaakt. Ook maken de engineers het abstracte netwerkmodel. De engineer rol heeft naast de eigen verantwoordelijkheden ook de rechten van een operator.

4.1.3 Operator

De operator is de rol die gebruikt wordt voor gebruikers die changes in het netwerk uitvoeren. Zij kunnen door middel van scenario's veranderingen uitvoeren in het netwerk. Ze kunnen geen IP-adressen aanpassen, VLAN's toevoegen, topologie aanpassen etc.

4.2 U-01: Beheren van een IPv6 prefix plan.

IPv6 prefix plannen worden door ontwerpers aangemaakt en verwijderd. IPv6 prefix plannen kunnen worden gebruikt voor meerdere klant typen. De ontwerpers bepalen welke IPv6 prefix plannen gekoppeld worden aan welke klant-typen.

4.3 U-02: Beheren van een IPv6 subnet plan.

Binnen een IPv6 prefix plan wordt een indeling gemaakt voor wat betreft de verdeling van de subnetten. Deze verdeling wordt gemaakt door de ontwerpers. De prefix wordt verdeeld in meerdere subnetten. Deze subnetten worden vervolgens gemodelleerd. Hier wordt vastgelegd welke VLAN id's gebruikt worden en hoe (reeksen van) IP-adressen verdeeld worden.

4.4 U-03: Gebruik van IPv6 in service tasks.

Ontwerpers moeten de volgende zaken in een service task ten aanzien van IPv6 kunnen doen:

- Toevoegen van een IPv6 subnet;
- Toekennen van een IPv6 adres aan een interface;
- Toekennen van een IPv6 subnet aan 1 of meerdere interfaces of een link tussen 2 nodes;
- Toekennen van een VLAN id aan een IPv6 subnet;
- Toekennen van een VRF aan een IPv6 subnet;
- Toekennen van een template aan een IPv6 subnet;
- Toekennen van adresreserveringen in een IPv6 subnet;
- Verwijderen van een IPv6 subnet;

- Verwijderen van een IPv6 subnet van 1 of meerdere poorten;
- Opzoeken van een vrij IPv6 adres;
- Opzoeken van een IPv6 subnet op service of VLAN id;
- Opzoeken van het VLAN id van een IPv4 subnet.

4.5 U-04: Beheren van IPv6 parameters in parameter groups.

De parameters in een parameter group worden beheerd door de ontwerpers. Elke parameter heeft een bepaald type. Bij het beheren van een IPv6 parameter kan de ontwerper voor het type IPv6 kiezen. Het lengte veld moet op dat moment automatisch gezet worden, zonder dat deze aan te passen is.

4.6 U-05: Beheren van IPv6 prefixen in een client.

Een engineer moet een IPv6 prefix kunnen toekennen aan-, of verwijderen van een client. NetYCE moet controleren of de prefix een geldige Global Unicast of Unique Local Unicast prefix is.

4.7 U-06: Beheren van IPv6 prefixen in een site.

Een engineer moet bij een site van een klant een IPv6 prefix kunnen toevoegen of verwijderen. Deze prefix kan nieuw toegevoegd worden of vanuit een lijst van reeds aan de client gekoppelde prefixen komen. Ook moet het mogelijk zijn een gedeelte van een client IPv6 prefix aan de site te koppelen.

4.8 U-07: Beheren van IPv6 subnetten in een service.

Een engineer moet IPv6 subnetten in een service kunnen toevoegen of verwijderen. Ook moet een overzicht getoond worden op welke interfaces het subnet gekoppeld is.

4.9 U-08: Beheren van het IPv6 adres op de management interface.

Een engineer moet in staat zijn het IPv6 adres van de management interface aan te passen. Dit moet vanuit het Node Details form gebeuren.

4.10 U-09: Beheren van een IPv6 subnet op een interface of link.

Een engineer moet in staat zijn om een IPv6 subnet op een interface of link te koppelen, of te verwijderen. De IPv6 subnetten waaruit gekozen kan worden moet gekoppeld zijn aan de service waar de node zich in bevindt of al gekoppeld zijn op 1 van de interfaces van de beoogde node.

4.11 U-10: Pingen van 1 of meerdere IPv6 adressen.

Een operator moet in staat zijn een selectie van nodes te pingen op het IPv6 adres. De output van deze ping moet getoond worden op het scherm.

5 *Bronnen en referenties*

| | |
|------------------|--|
| [1] | Forbes 1 juli 2013, How many things are currently connected to the IoT |
| [ARIN-IPv6] | ARIN number resource policy manual |
| [Cisco-IPv6] | Cisco Validated Design, IPv6 Addressing White paper, August 2013 |
| [IANA-IPv6-AS] | Internet Protocol Version 6 Address Space |
| [IANA-IPv6-GUAA] | IPv6 Global Unicast Address Assignments |
| [IANA-IPv6-Reg] | IANA IPv6 Special-Purpose Address Registry |
| [RFC3849] | IPv6 Address Prefix Reserved for Documentation , July 2004 |
| [RFC4291] | IP Version 6 Addressing Architecture, Feb 2006 |
| [RFC4862] | IPv6 Stateless Address Autoconfiguration, September 2007 |
| [RFC5375] | IPv6 Unicast Address Assignment Considerations, December 2008 |
| [RFC5737] | IPv4 Address Blocks Reserved for Documentation, January 2010 |
| [RFC5952] | IPv6 Text Representation, August 2010 |
| [RFC6555] | Happy Eyeballs: Success with Dual-Stack Hosts, April 2012 |
| [RIPE589] | IPv6 Address Allocation and Assignment Policy, May 2013 |
| [SURF-IPv6] | Surfnet Whitepaper IPv6-nummerplan opstellen. |

Bijlage VI: Technisch ontwerp

Technisch ontwerp

IPv6 implementatie in NetYCE

Derco Sportel

1548287

Colofon

NetYCE B.V.

Postbus 320
1380 AH Weesp, Nederland

Business Centre Muiderpoort
Leeuwendveldseweg 5n
1382 LV Weesp, Nederland

Auteur**Derco Sportel***Network Consultant*

☎ +31 6 1418 1361

✉ derco.sportel@netyce.com

🌐 derco.sportel

Docentbegeleider**Kees Uiterwijk***Hogeschooldocent, Systeembeheer*

☎ +31 88 481 8184

✉ kees.uitervijk@hu.nl

🌐 kees.uitervijk

Bedrijfsbegeleider**Ronny Lam***Solution Architect*

☎ +31 6 3976 6145

✉ ronny.lam@netyce.com

🌐 ronny_lam

Versie**1.0**

Document

Document geschiedenis

| Versie | Datum | Auteur | Notities |
|--------|--------------|---------------|--|
| 0.1 | 26 aug 2014 | Derco Sportel | Eerste opzet. |
| 0.2 | 1 sept 2014 | Derco Sportel | Inleiding, tekstuele aanpassingen en verschillende schermontwerpen |
| 0.3 | 12 sept 2014 | Derco Sportel | Tekstuele aanpassingen |
| 1.0 | 24 sept 2014 | Derco Sportel | Definitief |

Distributie

| Name | 0.1 | 0.2 | 0.3 | 1.0 |
|---------------------|-----|-----|-----|-----|
| Derco Sportel | ✓ | ✓ | ✓ | ✓ |
| Eric Yspeert | | | ✓ | ✓ |
| Ronny Lam | ✓ | ✓ | ✓ | ✓ |
| Kees Uiterwijk (HU) | | | | ✓ |
| Gerald Ovink (HU) | | | | ✓ |
| J de Doelder (CvT) | | | | ✓ |

Inhoudsopgave

| | |
|--|----|
| Colofon..... | 2 |
| Document..... | 3 |
| Document geschiedenis..... | 3 |
| Distributie..... | 3 |
| 1 Inleiding..... | 6 |
| 2 Use cases..... | 7 |
| 2.1 U-01: Beheren van een IPv6 prefixplan..... | 7 |
| 2.1.1 Use case beschrijving..... | 7 |
| 2.1.2 Kenmerken..... | 7 |
| 2.1.3 Schermafbbeelding..... | 8 |
| 2.1.4 Activiteitendiagram..... | 9 |
| 2.1.5 Scenario 1: Invoeren van een nieuw IPv6 prefix plan..... | 10 |
| 2.1.6 Scenario 2: (Ont) koppelen van een bestaand IPv6 prefix plan aan een bestaande klant type..... | 10 |
| 2.1.6. Scenario 3: Verwijderen van een bestaand IPv6 prefix plan..... | 10 |
| 2.2 U-02: Beheren van een IPv6 subnetplan..... | 11 |
| 2.2.1 Use case beschrijving..... | 11 |
| 2.2.2 Kenmerken..... | 11 |
| 2.2.3 Activiteitendiagram..... | 12 |
| 2.2.4 Scenario 1: Toevoegen van een nieuw subnet..... | 13 |
| 2.2.5 Scenario 2: Modelleren van een subnet type..... | 13 |
| 2.2.6 Scenario 3: Verwijderen van een subnet type..... | 13 |
| 2.3 U-03: Gebruik van IPv6 in servicetasks..... | 14 |
| 2.3.1 Use case beschrijving..... | 14 |
| 2.4 U-04: Beheren van IPv6 parameters in parameter groups..... | 14 |
| 2.4.1 Use case beschrijving..... | 14 |
| 2.4.2 Kenmerken..... | 14 |
| 2.4.3 Activiteitendiagram..... | 15 |
| 2.4.4 Scenario 1: Toevoegen van een parameter..... | 16 |
| 2.4.5 Scenario 2: Verwijderen van een parameter..... | 16 |
| 2.5 U-05: Beheren van IPv6 prefixen in een client..... | 17 |
| 2.5.1 Use case beschrijving..... | 17 |
| 2.5.2 Kenmerken..... | 17 |
| 2.5.3 Schermontwerp..... | 17 |
| 2.5.4 Activiteitendiagram..... | 18 |
| 2.5.5 Scenario 1: Toevoegen van een IPv6 prefix aan een klant..... | 19 |
| 2.5.6 Scenario 2: Verwijderen van een IPv6 prefix van een klant..... | 19 |
| 2.6 U-06: Beheren van IPv6 prefixen in een site..... | 20 |
| 2.6.1 Use case beschrijving..... | 20 |
| 2.6.2 Kenmerken..... | 20 |
| 2.6.3 Schermontwerp..... | 20 |
| 2.6.3 Activiteitendiagram..... | 21 |
| 2.6.4 Scenario 1: Toevoegen van een IPv6 prefix aan een site..... | 22 |
| 2.6.5 Scenario 2: Verwijderen van een IPv6 prefix van een site..... | 22 |

| | |
|--|----|
| 2.7 U-07: Beheren van IPv6 subnetten in een service..... | 23 |
| 2.7.1 Use case beschrijving..... | 23 |
| 2.7.2 Kenmerken..... | 23 |
| 2.7.3 Schermontwerp..... | 24 |
| 2.7.4 Activiteitendiagram..... | 25 |
| 2.7.5 Scenario 1: Toevoegen van een IPv6 subnet..... | 26 |
| 2.7.6 Scenario 2: Verwijderen van een IPv6 subnet..... | 26 |
| 2.7.6 Scenario 3: Veranderen van de naam van een IPv6 subnet..... | 26 |
| 2.8 U-08: Beheren van het IPv6 adres op de management interface..... | 27 |
| 2.8.1 Use case beschrijving..... | 27 |
| 2.8.2 Kenmerken..... | 27 |
| 2.8.3 Schermontwerp..... | 28 |
| 2.8.4 Activiteitendiagram..... | 29 |
| 2.8.5 Scenario 1: Het IPv6 adres van de node wordt veranderd..... | 29 |
| 2.9 U-09: Beheren van een IPv6 subnet op een interface of link..... | 30 |
| 2.9.1 Use case beschrijving..... | 30 |
| 2.9.2 Kenmerken..... | 30 |
| 2.9.3 Schermontwerp..... | 31 |
| 2.9.4 Activiteitendiagram..... | 32 |
| 2.9.5 Scenario 1: IPv6 subnet toevoegen aan een interface of link..... | 33 |
| 2.9.6 Scenario 2: IPv6 subnet verwijderen van een interface of link..... | 33 |
| 2.10 U-10: Pingen van 1 of meerdere IPv6 adressen..... | 34 |
| 2.10.1 Use case beschrijving..... | 34 |
| 2.10.2 Kenmerken..... | 34 |
| 2.10.3 Activiteitendiagram..... | 35 |
| 2.10.4 Scenario 1: Ping6 van nodes..... | 35 |
| 3 Service tasks..... | 36 |
| 3.1 LOCATE taken..... | 36 |
| 3.1.1 LOCATE IPV6_NET..... | 36 |
| 3.1.2 LOCATE IPV6_ADDR..... | 36 |
| 3.1.3 LOCATE SUBNET..... | 36 |
| 3.2 ADD taken..... | 36 |
| 3.3 ASSIGN taken..... | 36 |
| 3.3.1 ASSIGN IPV6_ADDR..... | 36 |
| 3.3.2 ASSIGN IPV6_NET..... | 36 |
| 3.4 DELETE taken..... | 36 |
| 4 Database model..... | 37 |
| 4.1 Benodigde tabellen..... | 37 |
| 4.2 Tabel IPv6_client_type..... | 38 |
| 4.3 Tabel: IPv6_plan..... | 38 |
| 4.4 Tabel: IPv6_site_plan..... | 39 |
| 4.5 Tabel: IPv6_subnet_plan..... | 40 |
| 4.6 Tabel: IPv6_prefix..... | 41 |
| 4.7 Tabel: IPv6_subnet..... | 42 |
| 4.8 Tabel: IPv6_map..... | 43 |
| 4.9 View: IPv6_prefix_view..... | 44 |
| 4.10 View: IPv6_subnet_view..... | 44 |
| 4.11 View: IPv6_map_view..... | 44 |
| 4.12 Aanpassingen aan reeds bestaande tabellen..... | 44 |
| 5 Bronnen en referenties..... | 45 |

1 Inleiding

Dit document beschrijft het technische ontwerp voor IPv6 ondersteuning in de netYCE applicatie. De use cases worden hierin uitgewerkt, alsmede het technische ontwerp voor het database model.

Tenzij expliciet anders vermeld zijn alle IP-adressen (IPv4 en IPv6) vervangen voor IP-adressen uit de documentatie prefix ([RFC5737] voor IPv4 en [RFC3849] voor IPv6).

2 Use cases

2.1 U-01: Beheren van een IPv6 prefixplan.

2.1.1 Use case beschrijving

De ontwerper maakt of verwijderd een IPv6 prefix plan. Deze prefix plannen worden vervolgens gekoppeld aan 1 of meerdere klant typen. Bij het aanmaken van het plan kiest de ontwerper een uniek nummer voor het plan, een prefix grootte en een omschrijving voor het plan.

Het koppelen van een prefix plan gaat door middel van het selecteren van een klant type, het selecteren van een prefix plan en vervolgens met pijltjes koppelen.

2.1.2 Kenmerken

| Kenmerk | Omschrijving |
|--------------------------------|--|
| Aanleiding (Trigger) | Bij een nieuw ontwerp maakt de ontwerper een keuze voor wat betreft het IPv6 plan |
| Actor | Ontwerper |
| Wijze van uitvoering | Interactief |
| Samenhang met andere use cases | Geen in relatie met IPv6. Uitgangssituatie is dat er klant types bestaan of aangemaakt worden in de applicatie. Deze use cases vallen buiten de scope van het project. |
| Frequentie van uitvoering | Bij elk nieuw ontwerp en aanpassing van een huidig ontwerp. |

2.1.3 Schermafbeelding

Type Modeling

Client types
Site types
Parameter groups
Node types
Service types

Client types

| Client type | Description |
|-------------|-------------|
| YCE | NetYCE |

+

IP-plans assigned

| Description | Version |
|---------------------|---------|
| NetYCE Generic Demo | IPv4 |
| Fritz LAN | IPv4 |
| Wan | IPv4 |
| ULA Custom | IPv6 |

+

IPv4 Plans Defined

| Prefix | Description |
|--------|---------------------|
| 16 | NetYCE Generic Demo |
| 24 | Fritz LAN |
| 32 | Wan |

+

IPv6 Plans Defined

| Prefix | Description |
|--------|-------------|
| 48 | Generic /48 |
| 48 | ULA Custom |

+

Groups assigned

| Group id | Description | Permiss. | Global lvl |
|----------|------------------------------|----------|------------|
| NetYCE | NetYCE B.V. Vendor S Manager | System | |

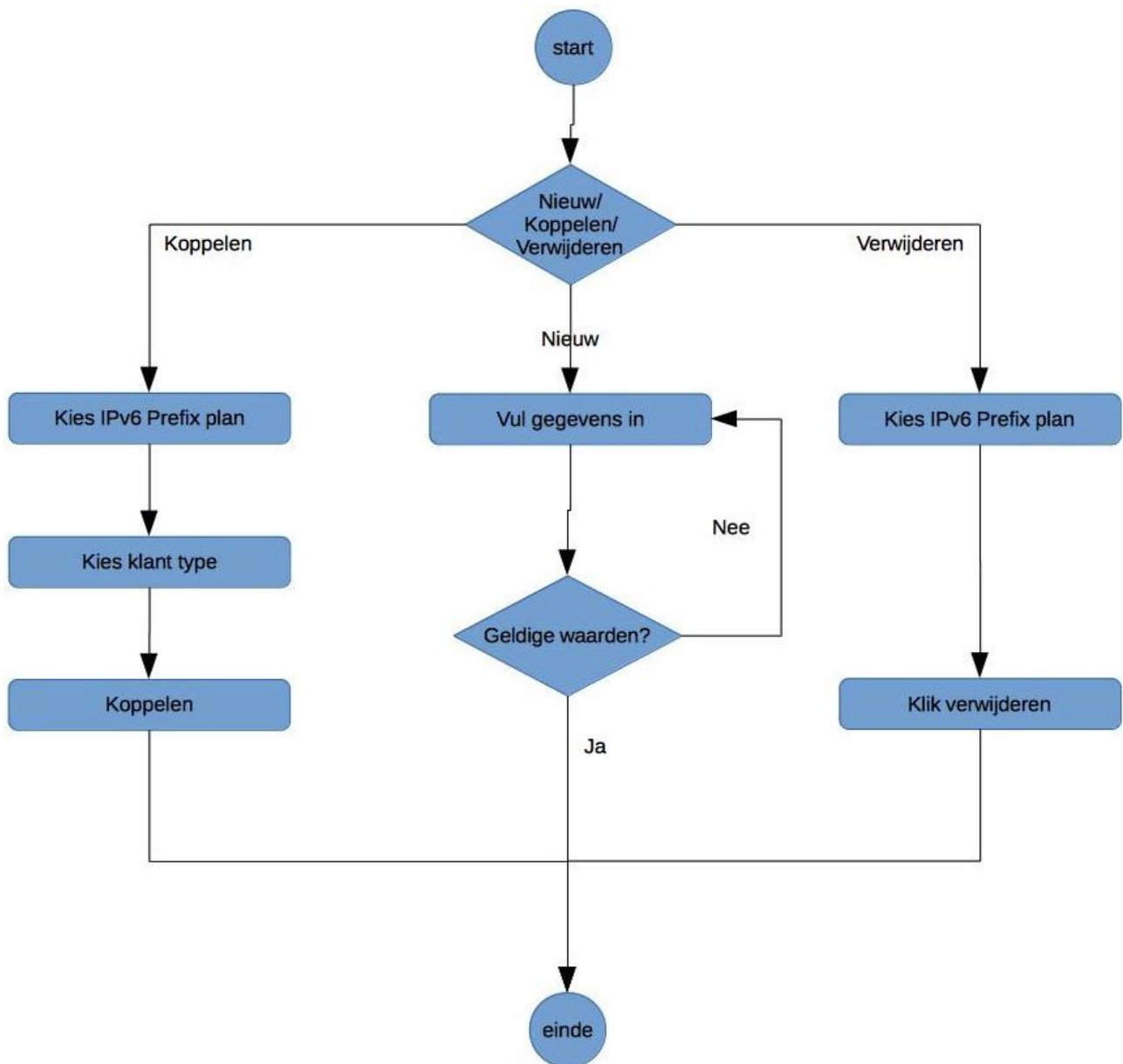
+

Groups available

| Group id | Description | Global lvl |
|----------|--|------------|
| NetYCE | NetYCE B.V. Vendor Suport | System |
| TenICT | http://www.tenict.nl/ | System |
| SSC | SSC DeSom | Operator |
| Default | Group permissions for users with Browser | |

Apply
Close

2.1.4 Activiteitendiagram



2.1.5 Scenario 1: Invoeren van een nieuw IPv6 prefix plan.

| | |
|-----------------------|--|
| Actor | Ontwerper |
| Pre conditie | De ontwerper is ingelogd en heeft het <i>Modeling form</i> geopend op het <i>Client types</i> tabblad |
| Scenario beschrijving | De ontwerper kiest voor een nieuw IPv6 prefix plan. De applicatie geeft het eerste vrije nummer in 10 tallen afgerond naar boven (wanneer 11 gebruikt is wordt 20 getoond). De ontwerper kiest voor een prefix size (standaard is deze 48) en geeft een omschrijving. Vervolges kiest de ontwerper voor opslaan. |
| Post conditie | Er is een nieuw IPv6 prefix plan beschikbaar. |

2.1.6 Scenario 2: (Ont) koppelen van een bestaand IPv6 prefix plan aan een bestaande klant type.

| | |
|-----------------------|--|
| Actor | Ontwerper |
| Pre conditie | De ontwerper is ingelogd en heeft het <i>Modeling form</i> geopend op het <i>Client types</i> tabblad. Er bestaat een IPv6 prefix plan en er bestaat een client type. Bij ontkoppelen is het IPv6 prefix plan al gekoppeld aan de client type. |
| Scenario beschrijving | De ontwerper kiest het client type waar een koppeling gelegd, of verwijderd moet worden. Vervolgens kiest de ontwerper het IPv6 prefixplan en kiest voor Koppelen [←] of ontkoppelen [→] |
| Foutmelding | Wanneer er aan een klant van dit type nog een IPv6 prefix gekoppeld is van het type wat verwijderd wordt, wordt een foutmelding getoond en wordt het prefixplan niet ontkoppeld. |
| Post conditie | Het IPv6 prefixplan is gekoppeld aan- of ontkoppeld van een client type. |

2.1.6. Scenario 3: Verwijderen van een bestaand IPv6 prefix plan.

| | |
|-----------------------|--|
| Actor | Ontwerper |
| Pre conditie | De ontwerper is ingelogd en heeft het <i>Modeling form</i> geopend op het <i>Client types</i> tabblad. Er bestaat een IPv6 prefix plan. |
| Scenario beschrijving | <p>De ontwerper kiest het IPv6 prefix plan wat verwijderd moet worden. Vervolgens wordt de verwijderknop ingedrukt. NetYCE controleert of er geen IPv6 prefixen bestaan welke gekoppeld zijn aan dit plan. Als dit niet zo is wordt om bevestiging gevraagd.</p> <p>Bij bevestiging wordt het prefix plan verwijderd uit de applicatie. Mocht het prefixplan nog gekoppeld zijn aan een klant type wordt deze referentie ook gelijk verwijderd. De bestaande IPv6 subnetplannen in dit prefixplan worden ook verwijderd.</p> |
| Post conditie | Het Client type scherm wordt getoond. Het verwijderde plan is niet meer zichtbaar. |

2.2 U-02: Beheren van een IPv6 subnetplan.

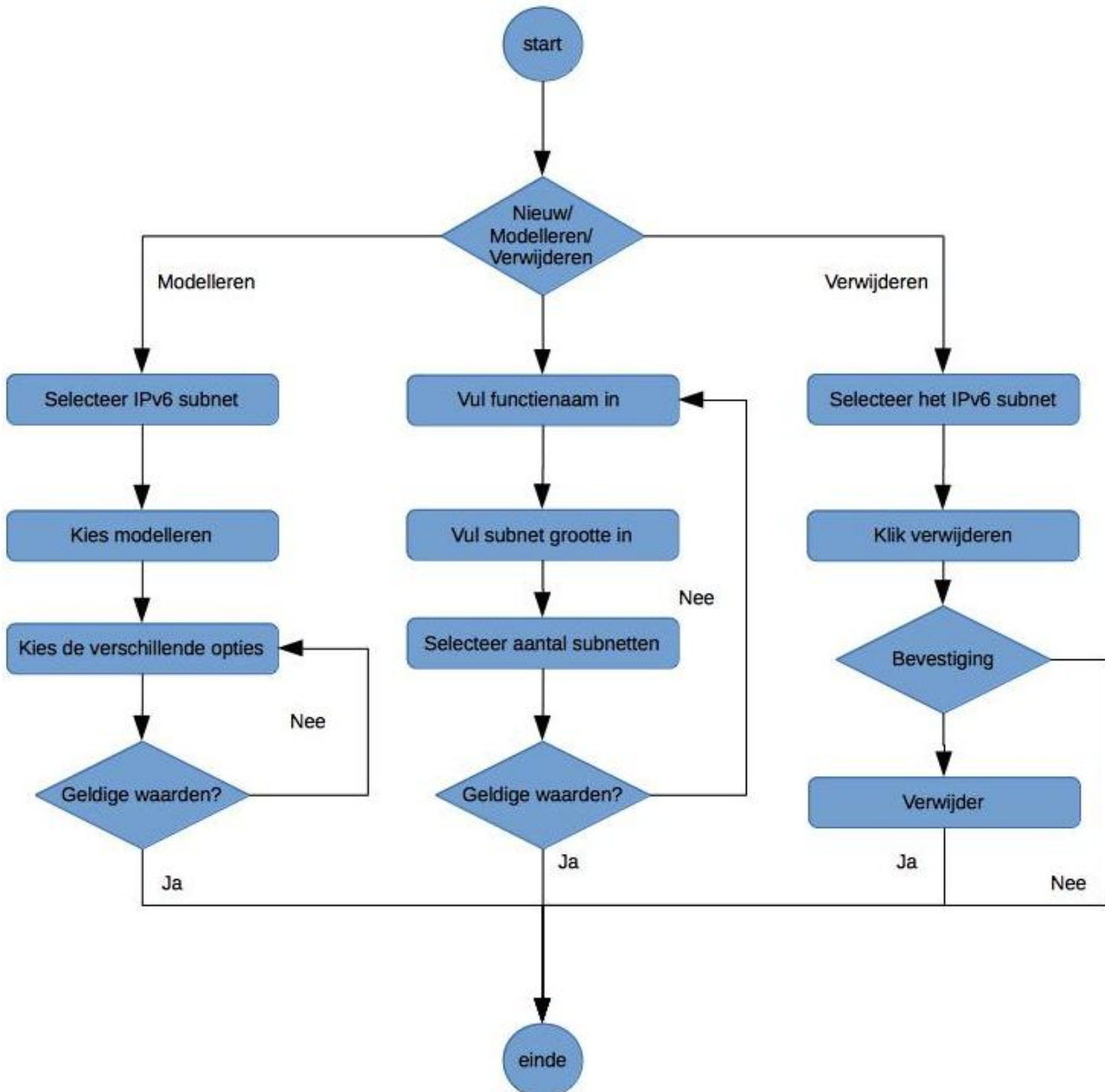
2.2.1 Use case beschrijving

De ontwerper deelt een IPv6 prefix in volgens een vooropgesteld subnetplan. De ontwerper kan kiezen om subnetten in te delen volgens functie en/of locatie.

2.2.2 Kenmerken

| Kenmerk | Omschrijving |
|--------------------------------|---|
| Aanleiding (Trigger) | Er is een IPv6 prefix plan aangemaakt. Dit plan kan ingedeeld worden in subnetten, of de bestaande subnetten kunnen worden aangepast. |
| Actor | Ontwerper |
| Wijze van uitvoering | Interactief |
| Samenhang met andere use cases | Er moet een IPv6 prefixplan bestaan. Deze wordt aangemaakt met U-01 scenario 1 |
| Frequentie van uitvoering | Bij het initiële ontwerp en bij aanpassingen op een bestaand ontwerp. |

2.2.3 Activiteitendiagram



2.2.4 Scenario 1: Toevoegen van een nieuw subnet.

| | |
|-----------------------|---|
| Actor | Ontwerper |
| Pre conditie | Er is een IPv6 prefix plan aangemaakt. De ontwerper is ingelogd en heeft het IPv6 subnetplan formulier geladen. |
| Scenario beschrijving | De ontwerper voegt een subnet toe. De subnet grootte wordt ingesteld en de naam van het subnet wordt ingevoerd door de ontwerper. De hoeveelheid subnetten wordt gekozen en toegevoegd. |
| Post conditie | De geselecteerde hoeveelheid subnetten is toegevoegd. |

2.2.5 Scenario 2: Modelleren van een subnet type.

| | |
|-----------------------|---|
| Actor | Ontwerper |
| Pre conditie | Er is een IPv6 prefix plan aangemaakt en het plan is ingedeeld in 1 of meerdere subnetten. De ontwerper is ingelogd en heeft het IPv6 subnetplan formulier geladen. |
| Scenario beschrijving | De ontwerper kiest het te modelleren subnet en kiest voor modelleren. De ontwerper kiest de gewenste functies en parameters en kiest voor opslaan. De gekozen parameters kunnen hierna desgewenst aangepast worden. |
| Post conditie | Het geselecteerde subnet is gemodelleerd. |

2.2.6 Scenario 3: Verwijderen van een subnet type

| | |
|-----------------------|---|
| Actor | Ontwerper |
| Pre conditie | Er is een IPv6 prefix plan aangemaakt en het plan is ingedeeld in 1 of meerdere subnetten. De ontwerper is ingelogd en heeft het IPv6 subnetplan formulier geladen. |
| Scenario beschrijving | De ontwerper kiest het subnet wat verwijderd moet worden en kiest voor verwijderen. Er wordt om bevestiging gevraagd. Wanneer de ontwerper bevestigd wordt het subnet verwijderd. |
| Post conditie | Het geselecteerde subnet is verwijderd. |

2.3 U-03: Gebruik van IPv6 in servicetasks.

2.3.1 Use case beschrijving

De ontwerper maakt in de service tasks gebruik van specifieke taken voor IPv6. De IPv6 gerelateerde taken worden in hoofdstuk 3 van dit document verder uitgewerkt.

2.4 U-04: Beheren van IPv6 parameters in parameter groups.

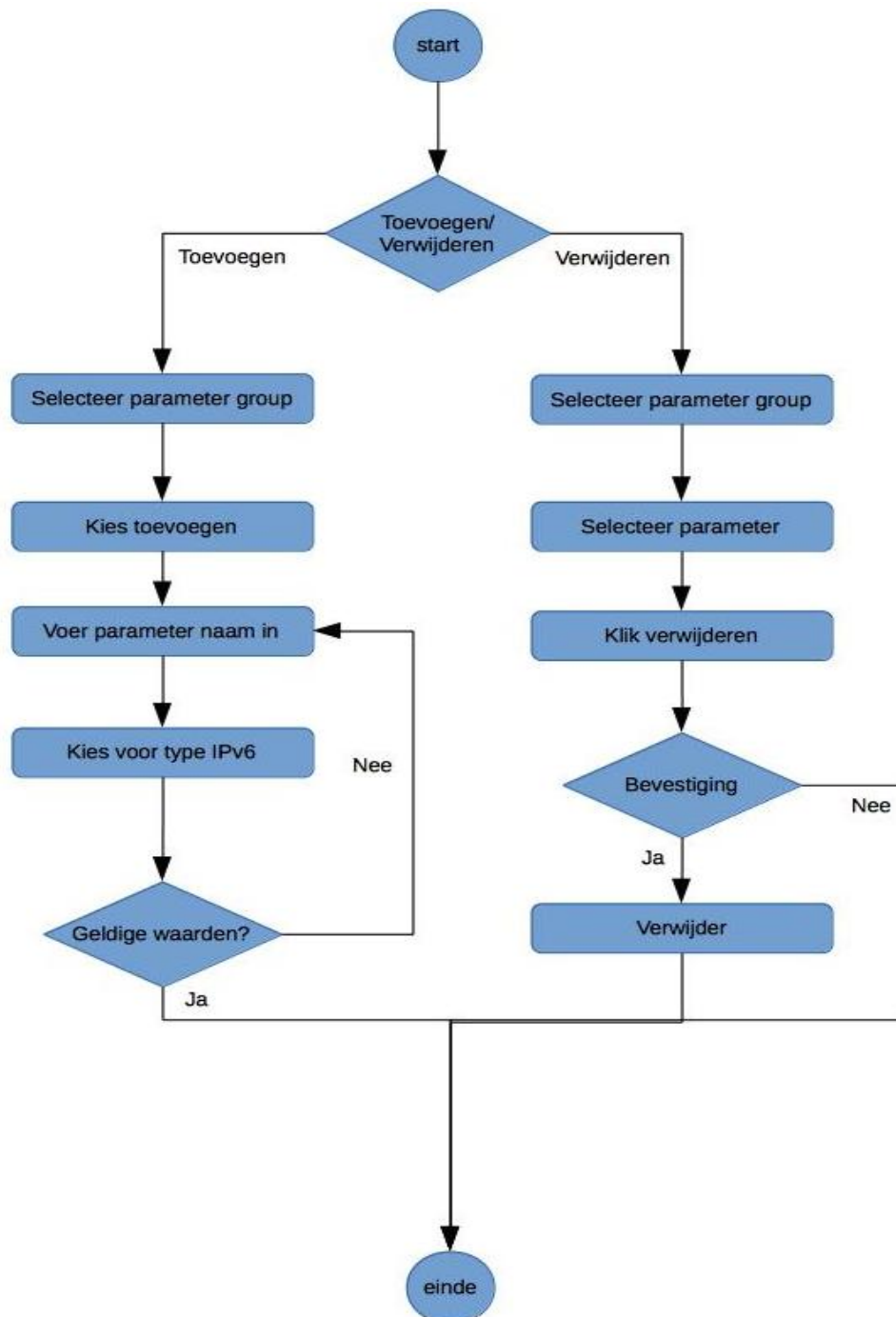
2.4.1 Use case beschrijving

De ontwerper maakt een custom variabele aan in een parameter groep. Engineers kunnen deze variabele dan later gebruiken in de ontwerpen.

2.4.2 Kenmerken

| Kenmerk | Omschrijving |
|--------------------------------|---|
| Aanleiding (Trigger) | Wanneer een IPv6 adres als variabele opgeslagen moet kunnen worden |
| Actor | Ontwerper |
| Wijze van uitvoering | Interactief |
| Samenhang met andere use cases | Geen |
| Frequentie van uitvoering | Incidenteel, bij nieuwe ontwerpen of aanpassingen op bestaande ontwerpen. |

2.4.3 Activiteitendiagram



2.4.4 Scenario 1: Toevoegen van een parameter

| | |
|-----------------------|--|
| Actor | Ontwerper |
| Pre conditie | De ontwerper is ingelogd in de applicatie. Het modeling formulier is geladen op het parameter groups tabblad. |
| Scenario beschrijving | De ontwerper kiest de parameter group waar de parameter benodigd is. De extra parameter kan toegevoegd worden. Uit de dropdown list wordt IPv6 gekozen. De overige waarden worden automatisch op de juiste instellingen gezet. |
| Post conditie | Binnen de geselecteerde parameter group is een nieuwe parameter beschikbaar. Opgeslagen parameters worden opgeslagen als HEX string van 32 karakters. |

2.4.5 Scenario 2: Verwijderen van een parameter

| | |
|-----------------------|---|
| Actor | Ontwerper |
| Pre conditie | De ontwerper is ingelogd in de applicatie. Het modeling formulier is geladen op het parameter groups tabblad. Er zijn IPv6 parameters aangemaakt in een parameter groep. |
| Scenario beschrijving | De ontwerper kiest de parameter groep waar de parameter verwijderd moet worden. De parameter wordt geselecteerd. De ontwerper kiest voor verwijderen. Er wordt om bevestiging gevraagd. Wanneer de ontwerper bevestigd wordt de parameter verwijderd. |
| Post conditie | Binnen de geselecteerde parameter group is een nieuwe parameter beschikbaar. Opgeslagen parameters worden opgeslagen als HEX string van 32 karakters. |

2.5 U-05: Beheren van IPv6 prefixen in een client.

2.5.1 Use case beschrijving

Een engineer of ontwerper voegt een IPv6 prefix aan een klant toe, of verwijderd deze van de klant. De prefix wordt aan de klant gekoppeld volgens een bepaald IPv6 plan.

2.5.2 Kenmerken

| Kenmerk | Omschrijving |
|--------------------------------|---|
| Aanleiding (Trigger) | Wanneer er bij een klant een nieuwe prefix toegevoegd moet worden, of een bestaande verwijderd. |
| Actor | Engineer of Ontwerper |
| Wijze van uitvoering | Interactief |
| Samenhang met andere use cases | Er moet een IPv6 prefix plan bestaan. |
| Frequentie van uitvoering | Bij wijzigingen op een IPv6 prefix bij een klant. |

2.5.3 Schermontwerp

Client Details

ClientCode1000

Client_typeYCE

ClientStatusactive

RegionYCE

Client_nameNetYCE

| Supernet | Ip-plan type | IPv6 Prefix | IPv6 prefix plan |
|------------------|---------------------|--------------------|--------------------------|
| 172.16.0.0/16 | NetYCE Generic Demo | 2001:db8:981::/48 | NetYCE customer CPE demo |
| 172.19.84.0/24 | Fritz LAN | 2001:db8:fc90::/48 | NetYCE customer CPE demo |
| 172.19.85.0/24 | Fritz LAN | fd00:c0ff:ee::/48 | NetYCE ULA demo |
| 192.168.178.0/24 | Fritz LAN | fd00:deca:ff::/48 | NetYCE ULA demo |
| 192.168.179.0/24 | Fritz LAN | | |
| 192.168.180.0/24 | Fritz LAN | | |
| 192.168.181.0/24 | Fritz LAN | | |

+ / -

Reserve

Client_reserve1

Client_reserve2

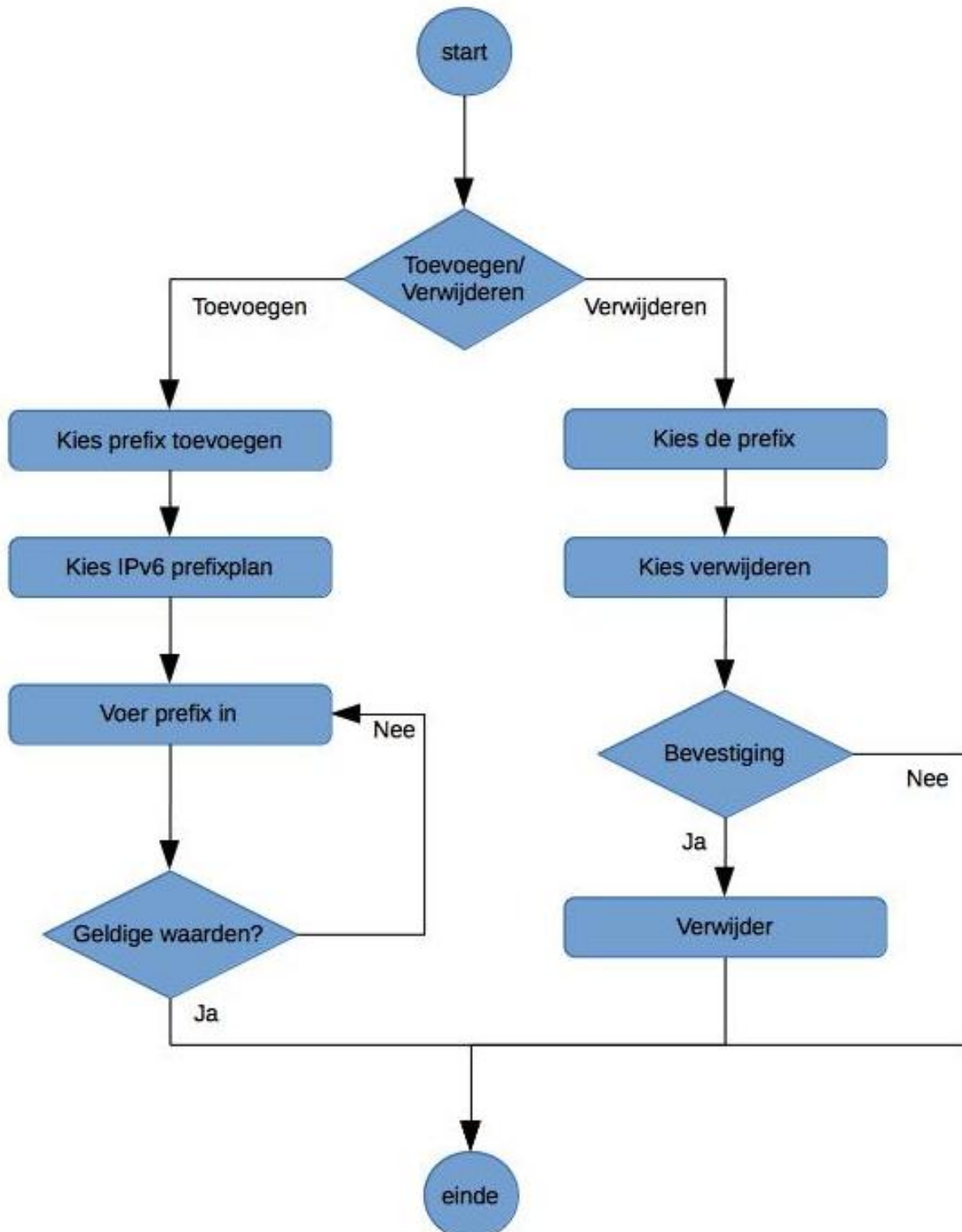
Client_reserve3

Client_reserve4

Client_notes

OKClose

2.5.4 Activiteitendiagram



2.5.5 Scenario 1: Toevoegen van een IPv6 prefix aan een klant.

| | |
|-----------------------|--|
| Actor | Engineer of Ontwerper |
| Pre conditie | De engineer is ingelogd. Er bestaat een klant in de applicatie en er is een IPv6 prefixplan aanwezig in de applicatie. |
| Scenario beschrijving | De engineer opent het Client details formulier en kiest voor IPv6 prefix toevoegen. De engineer kiest voor een vrije prefix in de applicatie, of voert een vrije prefix in. Het juiste IPv6 plan wordt gekoppeld en de prefix wordt gevalideerd. Wanneer de prefix gevalideerd is wordt deze opgeslagen. |
| Post conditie | Het client details formulier toont de details van de klant met de zojuist toegevoegde IPv6 prefix. |

2.5.6 Scenario 2: Verwijderen van een IPv6 prefix van een klant.

| | |
|-----------------------|---|
| Actor | Engineer of Ontwerper |
| Pre conditie | De engineer is ingelogd. Er bestaat een klant in de applicatie en er is een IPv6 prefix gekoppeld aan de klant. |
| Scenario beschrijving | De engineer opent het Client details formulier en kiest de IPv6 prefix die verwijderd moet worden. De engineer kiest voor verwijderen. Nadat de applicatie om bevestiging vraagt en heeft gekregen wordt de prefix verwijderd. Reeds bestaande IPv6 subnetten uit de prefix worden niet verwijderd. |
| Post conditie | Het client details formulier toont de details van de klant zonder de verwijderde IPv6 prefix. De prefix wordt niet verwijderd uit de applicatie, maar wordt gemarkeerd als vrij. |

2.6 U-06: Beheren van IPv6 prefixen in een site.

2.6.1 Use case beschrijving

De engineer of ontwerper voegt een IPv6 prefix toe aan een site, of verwijderd deze van de site. Wanneer er geen locatie masker beschikbaar is voor het IPv6 prefixplan wordt de gehele prefix gekoppeld, wanneer deze wel beschikbaar is krijgt de engineer een lijst met vrije IPv6 prefixen uit de gekoppelde IPv6 prefixen op de klant.

2.6.2 Kenmerken

| Kenmerk | Omschrijving |
|--------------------------------|--|
| Aanleiding (Trigger) | Bij wijzigingen op het netwerk van de klant. |
| Actor | Engineer of Ontwerper |
| Wijze van uitvoering | Interactief. |
| Samenhang met andere use cases | U-01, U-02 en U-05 |
| Frequentie van uitvoering | Wanneer nodig. |

2.6.3 Schermontwerp

Site Details

ClientCode1000SiteCode127A

Site Address

AddressHessenwegNr127A

CityDe BiltSt.UT

Zip code3731JGReference code

Site Keys

SiteStatusactive

Managed LanNo

Site_typeXS4ALL Fritz

OSPF area0

IPv6 Prefix

2001:db8:fc90::/54

fd00:c0ff:ee::/56

IPv6 prefix plan

NetYCE Customer CPE Demo

NetYCE ULA demo

Site Profiles

Site_profile10 - unknown

Site_profile20 - none

MigrationPhase0 - - onbekend -

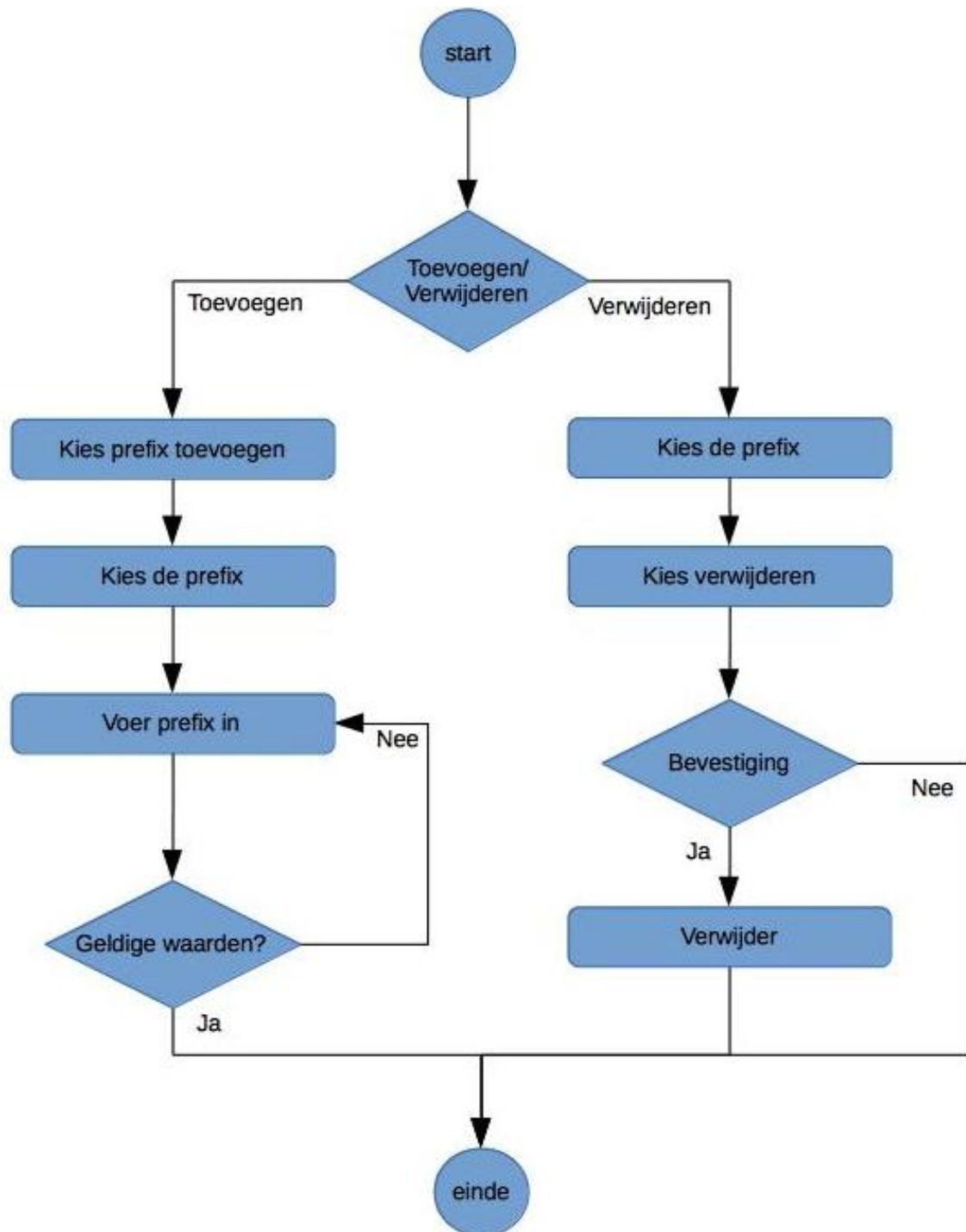
Site_notes

OK

Close

20

2.6.3 Activiteitendiagram



2.6.4 Scenario 1: Toevoegen van een IPv6 prefix aan een site.

| | |
|-----------------------|---|
| Actor | Engineer of Ontwerper |
| Pre conditie | De engineer is ingelogd. Er bestaat een klant in de applicatie en er is een IPv6 prefixplan aanwezig in de applicatie. |
| Scenario beschrijving | De engineer opent het Site details formulier en kiest voor IPv6 prefix toevoegen. De engineer kiest voor een vrije prefix in de applicatie. Vervolgens wordt de prefix opgeslagen voor de site. |
| Post conditie | Het site details formulier toont de details van de site met de zojuist toegevoegde IPv6 prefix. |

2.6.5 Scenario 2: Verwijderen van een IPv6 prefix van een site.

| | |
|-----------------------|--|
| Actor | Engineer of Ontwerper |
| Pre conditie | De engineer is ingelogd. Er bestaat een klant in de applicatie en er is een IPv6 prefix gekoppeld aan de site. |
| Scenario beschrijving | De engineer opent het Site details formulier en kiest de IPv6 prefix die verwijderd moet worden. De engineer kiest voor verwijderen. Nadat de applicatie om bevestiging vraagt en heeft gekregen wordt de prefix verwijderd van de locatie. Reeds bestaande IPv6 subnetten uit de prefix worden niet verwijderd. |
| Post conditie | Het site details formulier toont de details van de site zonder de verwijderde IPv6 prefix. |

2.7 U-07: Beheren van IPv6 subnetten in een service.

2.7.1 Use case beschrijving

De engineer voegt vanuit het service details formulier een IPv6 subnet toe of verwijderd deze. De engineer kan eventueel de naam van het subnet aanpassen.

2.7.2 Kenmerken

| Kenmerk | Omschrijving |
|--------------------------------|--|
| Aanleiding (Trigger) | Wanneer er wijzigingen nodig zijn op het netwerk van de klant. |
| Actor | Engineer of Ontwerper |
| Wijze van uitvoering | Interactief |
| Samenhang met andere use cases | Geen. |
| Frequentie van uitvoering | Wanneer nodig. |

2.7.3 Schermontwerp

Service Details

Client

1000

Site

127A-Vbox

Service

router

Subnets in service

| Function | Vlan | Vrf | Subnet | Supernet | Ip-plan | #Interf |
|----------|------|-----|----------------|------------|---------|---------|
| Loopback | | | 172.16.0.1 /32 | 172.16.0.0 | 10 | 0 |
| Loopback | | | 172.16.0.2 /32 | 172.16.0.0 | 10 | 1 |
| Loopback | | | 172.16.0.3 /32 | 172.16.0.0 | 10 | 1 |
| Loopback | | | 172.16.0.4 /32 | 172.16.0.0 | 10 | 1 |

+

IPv6 subnets in service

| Name | VLAN | VRF | IPv6 subnet | IPv6 prefix | #Interfaces |
|----------|------|-----|------------------------|-------------------|-------------|
| Loopback | | | fc00:deca:ff::1/128 | fc00:deca:ff::/48 | 1 |
| Data | 100 | | fc00:deca:ff:1::/64 | fc00:deca:ff::/48 | 10 |
| Voice | 200 | | fc00:deca:ff:c001::/64 | fc00:deca:ff::/48 | 6 |
| Server | 300 | | fc00:deca:ff:d001::/64 | fc00:deca:ff::/48 | 50 |

+

Interfaces in subnet

| Function | Topo | Port | Nodename |
|----------|------|------|----------|
|----------|------|------|----------|

Nodes in service

| Node | Type | Status | Topo. | Template |
|----------|--------|---------|-------|----------|
| router01 | router | planned | NA | router |
| router02 | router | planned | NA | router |
| router03 | router | planned | NA | router |
| router04 | router | planned | NA | router |
| router05 | router | planned | NA | router |

+

Topology

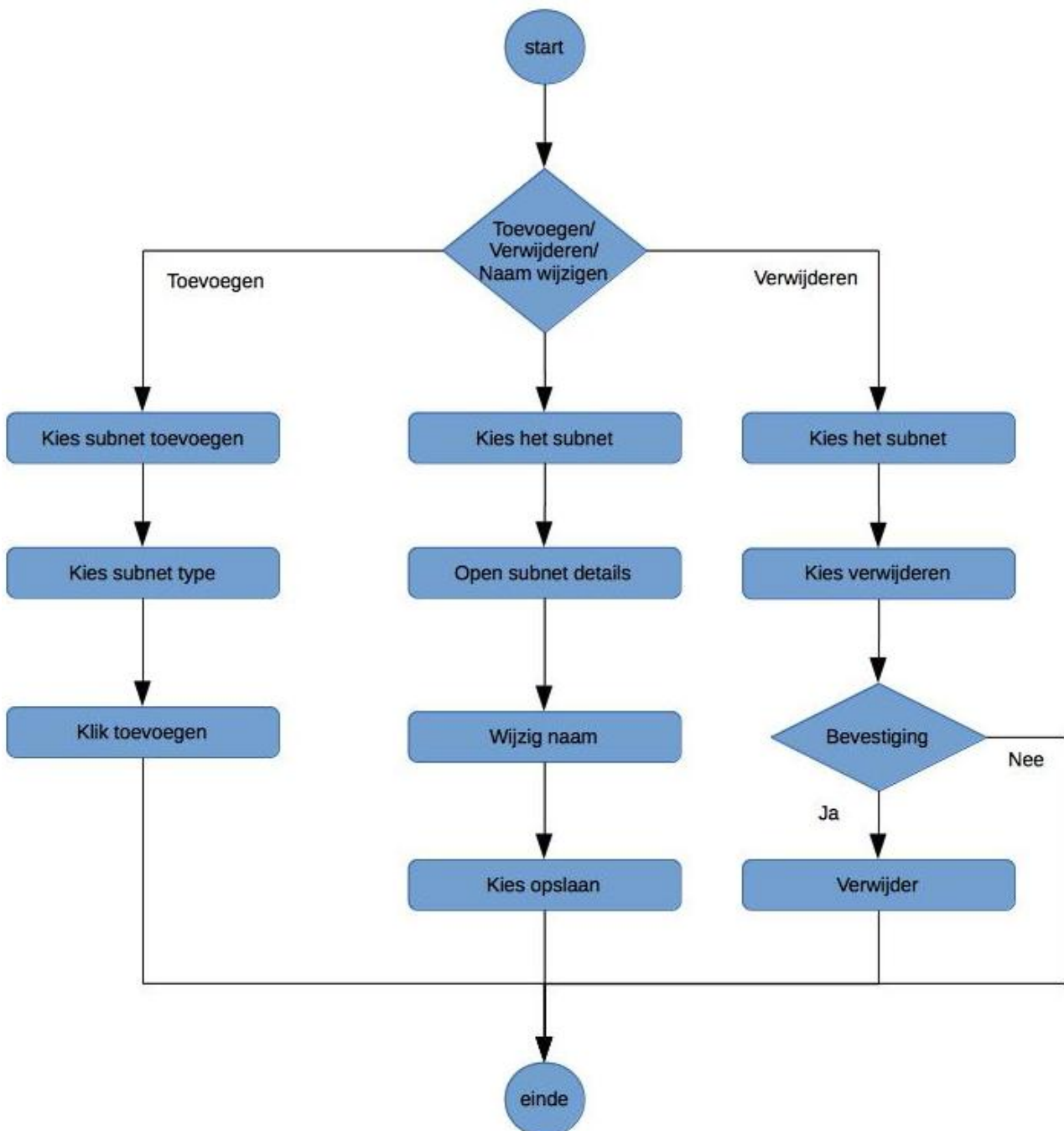
Ports

Details

OK

Close

2.7.4 Activiteitendiagram



2.7.5 Scenario 1: Toevoegen van een IPv6 subnet

| | |
|-----------------------|--|
| Actor | Engineer of Ontwerper |
| Pre conditie | Er bestaat reeds een service op een locatie. De engineer is ingelogd en heeft het Service details formulier geladen. |
| Scenario beschrijving | De engineer kiest voor IPv6 subnet toevoegen, kiest het gewenste subnet en kiest voor toevoegen. |
| Post conditie | Het IPv6 subnet is toegevoegd. |

2.7.6 Scenario 2: Verwijderen van een IPv6 subnet

| | |
|-----------------------|--|
| Actor | Engineer of Ontwerper |
| Pre conditie | De engineer is ingelogd en heeft het service details formulier geladen. Er bestaat een IPv6 subnet in de service. |
| Scenario beschrijving | Het doel IPv6 subnet wordt geselecteerd en de engineer kiest voor verwijderen. Er wordt om bevestiging gevraagd en na bevestiging wordt het subnet verwijderd. Het subnet wordt ook verwijderd van alle interfaces in het netwerk. |
| Post conditie | Het subnet is verwijderd. |

2.7.6 Scenario 3: Veranderen van de naam van een IPv6 subnet

| | |
|-----------------------|--|
| Actor | Engineer of Ontwerper |
| Pre conditie | De engineer is ingelogd en heeft het IPv6 subnet details formulier voor zich. Het IPv6 subnet is reeds aangemaakt. |
| Scenario beschrijving | De engineer verandert de naam van het subnet en kiest voor opslaan. |
| Post conditie | De naam van het subnet is veranderd. |

2.8 U-08: Beheren van het IPv6 adres op de management interface.

2.8.1 Use case beschrijving

De engineer of ontwerper verandert het IPv6 adres op de management interface. Het adres moet een geldig IPv6 adres zijn uit het IPv6 managementsubnet. Eventueel kan er gekozen worden voor een random IPv6 adres.

2.8.2 Kenmerken

| Kenmerk | Omschrijving |
|--------------------------------|--|
| Aanleiding (Trigger) | Wanneer het huidige adres niet klopt en veranderd moet worden. |
| Actor | Engineer of Ontwerper |
| Wijze van uitvoering | Interactief |
| Samenhang met andere use cases | Geen |
| Frequentie van uitvoering | Wanneer nodig. |

2.8.3 Schermontwerp

Node Details

ClientCode
1000

SiteCode
127A-Vbox

Hostname
router01
Change

Node_fqdn
router01.vbox.lan

Node_type
router
40

Service_type
router

Node_position
NA
Redundant
☐

Loopback
172.16.0.2

Loopback IPv6
fc00:deca:ff::1

Enable_secret

DeviceStatus
planned

Domain
YCE
Details

Original node

Boot_system
Reset

Boot_loader

SNMP_syscontact

SNMP_syslocation

SNMP_sysname

Template
router

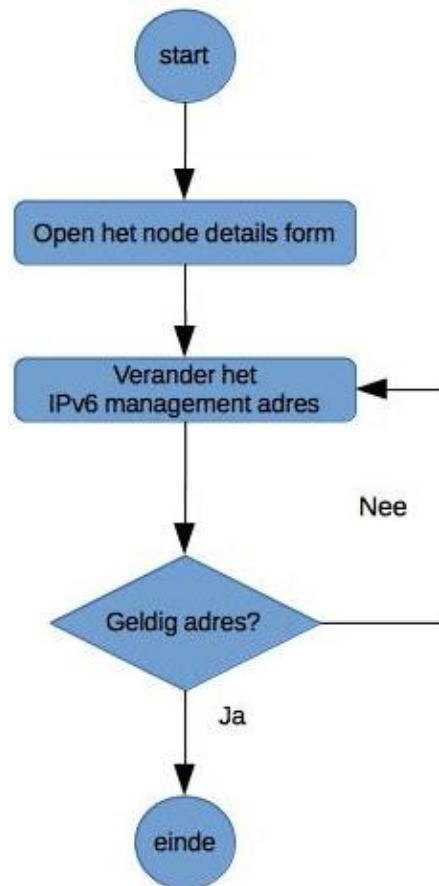
Active rev

Ports
Topology
VRFs
eVPN

Notes

Apply
Close

2.8.4 Activiteitendiagram



2.8.5 Scenario 1: Het IPv6 adres van de node wordt veranderd.

| | |
|-----------------------|--|
| Actor | Engineer of Ontwerper |
| Pre conditie | De engineer is ingelogd in de applicatie en heeft het node details scherm geladen voor de node waarvan het adres veranderd moet worden. |
| Scenario beschrijving | De engineer verandert het IPv6 adres in het invoerveld of kiest voor een nieuw random adres. Bij keuze voor een nieuw random adres wordt om bevestiging gevraagd. Vervolgens wordt met de apply knop het adres opgeslagen. |
| Post conditie | Het managementadres van de node is veranderd. |

2.9 U-09: Beheren van een IPv6 subnet op een interface of link.

2.9.1 Use case beschrijving

De engineer of ontwerper voegt een IPv6 subnet toe, of verwijderd deze van een interface of een link tussen 2 nodes.

2.9.2 Kenmerken

| Kenmerk | Omschrijving |
|--------------------------------|---|
| Aanleiding (Trigger) | Er moet een IPv6 subnet verwijderd van, of toegevoegd worden aan een interface of link. |
| Actor | Engineer of Ontwerper |
| Wijze van uitvoering | Interactief |
| Samenhang met andere use cases | Geen |
| Frequentie van uitvoering | Wanneer nodig |

2.9.3 Schermontwerp

Add IPv6 subnet to port

Add subnet to topology

For port

| Node | Portname | Template | Ch. Topo |
|----------|----------|----------|----------|
| router02 | Gi0/3 | Gi | |
| router02 | Gi1/0 | Gi | B |
| router02 | Gi1/1 | Gi | A |
| router02 | Gi1/2 | Gi | A |
| router02 | Gi1/3 | Gi | A |
| router02 | Lo0 | Loopback | |

Existing subnets on interface

| Node | Portname | Pos. | NetType | Vlan |
|------|----------|------|---------|------|
|------|----------|------|---------|------|

To subnet

| Net type | Subnet | Vlan | #Interfaces |
|----------|-----------------------|------|-------------|
| Loopback | fc00:deca:ff::1/128 | | |
| Loopback | fc00:deca:ff::2/128 | | |
| Loopback | fc00:deca:ff::3/128 | | |
| PPP | fc00:deca:ff::1:0/127 | | |
| PPP | fc00:deca:ff::1:2/127 | | |
| PPP | fc00:deca:ff::1:4/127 | | |

Existing interfaces in subnet

| Net type | Topo | Vlan | Port | Node |
|----------|------|------|------|------|
|----------|------|------|------|------|

Node
Port
=>
Type
Subnet

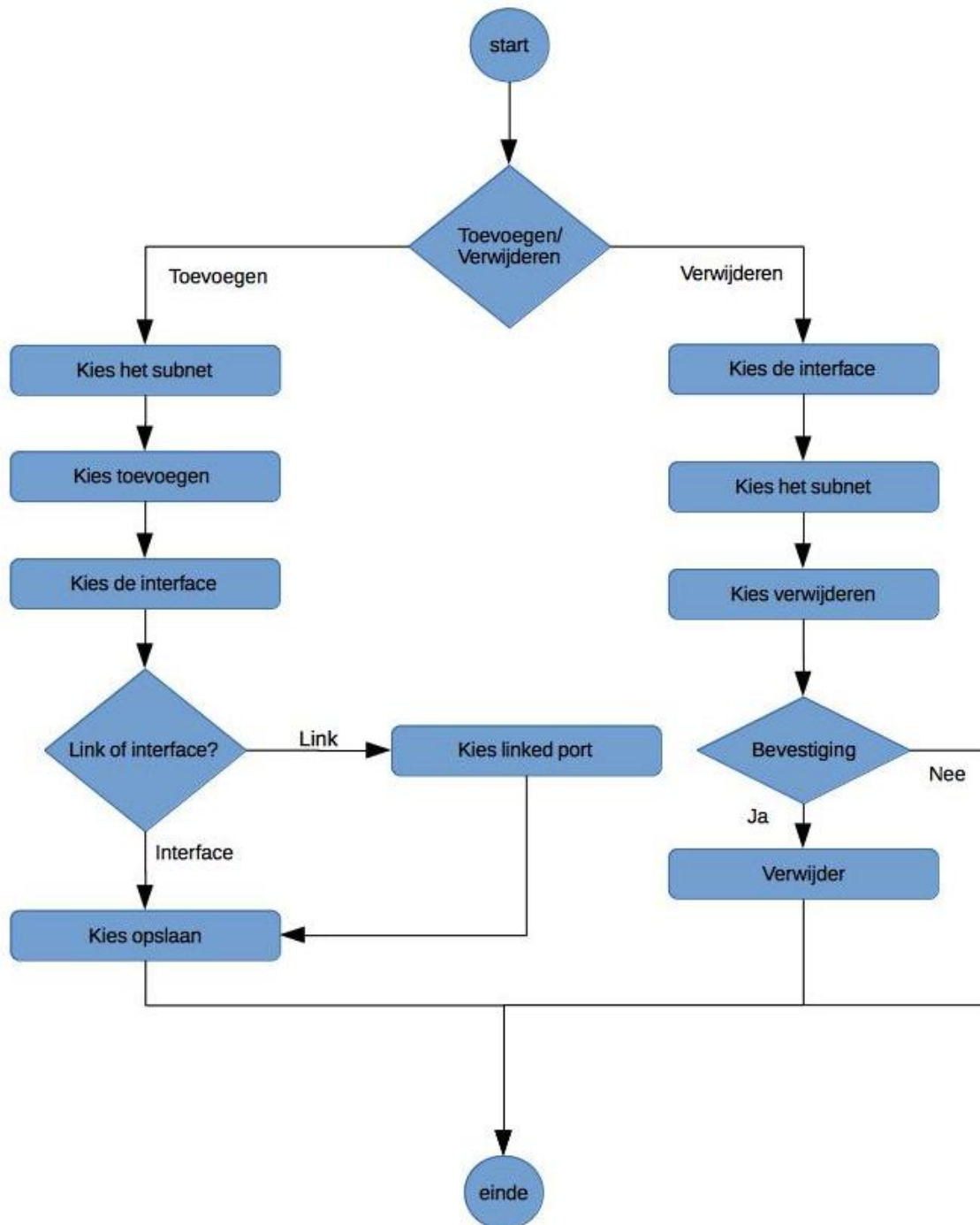
☒ Include linked port

Node
Port
Topo

Node
Port
Topo

OK
Close

2.9.4 Activiteitendiagram



2.9.5 Scenario 1: IPv6 subnet toevoegen aan een interface of link.

| | |
|-----------------------|---|
| Actor | Engineer |
| Pre conditie | De engineer is ingelogd en heeft het topology details formulier geladen. |
| Scenario beschrijving | De engineer kiest in het formulier het IPv6 subnet wat toegevoegd moet worden aan de link en kiest voor toevoegen. Vervolgens wordt de beoogde interface gekozen. Wanneer een link bestaat op de interface wordt de optie <i>Include linked port</i> getoond. Bij opslaan wordt het subnet toegevoegd aan de geselecteerde interface of link. |
| Post conditie | Het geselecteerde IPv6 adres is toegevoegd aan de geselecteerde interface of link. Dit is zichtbaar in het overzicht. |

2.9.6 Scenario 2: IPv6 subnet verwijderen van een interface of link.

| | |
|----------------------|---|
| Actor | Engineer |
| Pre conditie | De engineer is ingelogd en heeft het topology details formulier geladen. |
| Scenariobeschrijving | De engineer kiest in het formulier het IPv6 subnet wat toegevoegd moet worden aan de link en kiest voor toevoegen. Vervolgens wordt de beoogde interface gekozen. Wanneer een link bestaat op de interface wordt de optie <i>Include linked port</i> getoond. Bij opslaan wordt het subnet toegevoegd aan de geselecteerde interface of link. |
| Post conditie | Het geselecteerde IPv6 adres is toegevoegd aan de geselecteerde interface of link. Dit is zichtbaar in het overzicht. |

2.10 U-10: Pingen van 1 of meerdere IPv6 adressen.

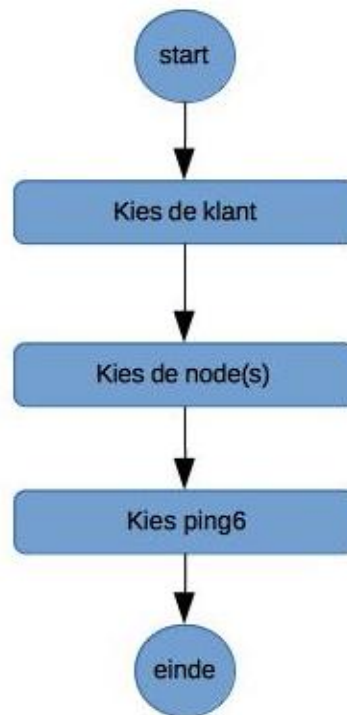
2.10.1 Use case beschrijving

De operator, engineer of ontwerper kiest de klant waarvan de nodes gepingd moeten worden. Vervolgens worden de nodes geselecteerd die gepingd moeten worden. Na selectie kiest de operator voor ping en gaat de applicatie de geselecteerde nodes proberen te pingen. De uitkomst wordt op het scherm getoond.

2.10.2 Kenmerken

| Kenmerk | Omschrijving |
|--------------------------------|--|
| Aanleiding (Trigger) | Wanneer de operator wil bepalen of een bepaalde set van nodes bereikbaar is voor de NetYCE applicatie. |
| Actor | Operator, Engineer of Ontwerper |
| Wijze van uitvoering | Interactief |
| Samenhang met andere use cases | Geen |
| Frequentie van uitvoering | Incidenteel |

2.10.3 Activiteitendiagram



2.10.4 Scenario 1: Ping6 van nodes.

| | |
|----------------------|---|
| Actor | Operator |
| Pre conditie | De operator is ingelogd in de applicatie en heeft het formulier om nodes te pingen voor zich. |
| Scenariobeschrijving | De operator zoekt de klant en selecteert deze. Vervolgens kiest hij voor alle doel nodes en kiest ping. |
| Post conditie | De uitkomst van de ping wordt getoond. |

3 Service tasks

3.1 LOCATE taken

3.1.1 LOCATE IPV6_NET

Via LOCATE IPV6_NET kan een IPv6 subnet opgezocht worden. Dit kan op basis van een functie of subnetnaam, een VLAN id of een geselecteerd IPv6 subnet in het service details formulier.

3.1.2 LOCATE IPV6_ADDR

Via LOCATE IPV6_ADDR kan een IPv6 adres opgezocht worden. Dit kan een random adres uit een IPv6 subnet zijn, of een adres uit een vooraf gedefinieerde reeks.

3.1.3 LOCATE SUBNET

Via LOCATE SUBNET kan het VLAN id van een IPv4 subnet opgezocht worden. Dit VLAN id kan vervolgens toegekend worden aan een IPv6 subnet via een ASSIGN taak.

3.2 ADD taken

Via de add opdracht kan een IPv6 subnet toegevoegd worden. Hier in zijn 2 mogelijkheden, er kan een gemodelleerd IPv6 subnet toegevoegd worden op IPv6 subnet naam en een custom IPv6 subnet.

3.3 ASSIGN taken

3.3.1 ASSIGN IPV6_ADDR

Via ASSIGN IPV6_ADDR kan een IPv6 adres toegevoegd worden aan een interface.

3.3.2 ASSIGN IPV6_NET

Via ASSIGN IPV6_NET kunnen de volgende taken uitgevoerd worden:

- Toekennen van het IPv6 subnet aan een interface, lijst van interfaces en een link.
- Toekennen van een VLAN id aan het IPv6 subnet.
- Toekennen van een naam aan een IPv6 subnet.
- Toekennen van een VRF aan een IPv6 subnet.
- Toekennen van een template aan het IPv6 subnet

3.4 DELETE taken

Met de DELETE taken kan een IPv6 subnet verwijderd worden van een interface, een lijst met interfaces, volledig of het VRF kan verwijderd worden van een IPv6 subnet.

4 Database model

4.1 Benodigde tabellen

Om de benodigde informatie op te slaan zijn de volgende tabellen benodigd in netYCE:

- IPv6_client_type
- IPv6_plan
- IPv6_site_plan
- IPv6_subnet_plan
- IPv6_prefix
- IPv6_subnet
- IPv6_map

Het huidige database model is een relationeel model echter er zijn geen foreign keys en constraints gedefinieerd. Alle logica en data integriteit wordt opgelost en gevalideerd in de software.

Alle records in alle benodigde tabellen worden voorzien van een timestamp. Het tijdstip van aanmaken of laatste update van een record wordt hierin opgeslagen.

Omdat de uiteindelijke IPv6 adressen opgeslagen worden als varbinary formaat is het niet mogelijk om, zonder gebruik van SQL functies, de adressen weer te geven als een geldig IPv6 adres. Voor de gebruikersvriendelijkheid worden er nog 3 views gemaakt:

- IPv6_prefix_view
- IPv6_subnet_view
- IPv6_map_view

Deze views worden alleen gebruikt om data weer te geven. INSERT, UPDATE en DELETE query's moeten op de tabellen worden gedaan.

Vanwege de scope van het project zullen alleen de IPv6 gerelateerde tabellen en views in dit hoofdstuk uitgewerkt worden. Reeds bestaande tabellen in netYCE worden niet verder uitgewerkt.

4.2 Tabel IPv6_client_type

De tabel IPv6_client_type wordt gebruikt om de koppeling te maken tussen een IPv6 prefix plan en een client type.

| Tabel: IPv6_client_type | | |
|-------------------------|--------------|--|
| Kolom | Type | Attributen |
| Client_type | VARCHAR(30) | NOT NULL |
| IPv6_plan_id | Integer (11) | NOT NULL |
| IPv6_plan_size | Integer(3) | NOT NULL |
| IPv6_plan_description | VARCHAR(100) | DEFAULT NULL |
| Timestamp | timestamp | NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP |

Client_type: De referentie naar het Client type.

IPv6_plan_id: Het IPv6 plan id.

IPv6_plan_size: De grootte van het IPv6 plan.

IPv6_plan_description: De omschrijving van het IPv6 prefix plan.

4.3 Tabel: IPv6_plan

De IPv6_plan tabel wordt gebruikt om alle subnetten binnen een prefixplan op te slaan. De tabel wordt niet gebruikt om werkelijke IPv6 adressen op te slaan maar om de offset voor de prefix op te slaan.

IPv6 plannen kunnen voor meerdere klant typen gebruikt worden. Er wordt dus geen specifieke klant type informatie opgeslagen in deze tabel.

| Tabel: IPv6_plan | | |
|------------------------|---------------|--|
| Kolom | Type | Attributen |
| IPv6_plan_id | Integer (11) | NOT NULL |
| IPv6_subnet_type | Integer (10) | NOT NULL |
| IPv6_subnet_size | Integer (3) | NOT NULL DEFAULT 64 |
| IPv6_subnet_min_offset | varbinary(16) | NOT NULL |
| IPv6_subnet_max_offset | varbinary(16) | NOT NULL |
| IPv6_subnet_name | varchar(30) | NOT NULL |
| IPv6_subnet_remark | varchar(100) | DEFAULT NULL |
| Timestamp | timestamp | NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP |

IPv6_plan_id: Een administratief nummer. Dit nummer wordt door de ontwerper gekozen in het Design proces. Het is mogelijk om nummer tot 11 cijfers te kiezen echter praktisch is om tot 4 cijfers te gebruiken.

IPv6_subnet_type: Het typenummer wordt gebruikt om de type subnetten bij te houden. Dit nummer wordt niet getoond aan de gebruikers en wordt alleen maar intern gebruikt.

IPv6_subnet_size: De hoeveelheid bits die gebruikt worden om de subnet grote aan te geven. IPv6 gebruikt standaard een /64 subnet [RFC4291, sectie 2.5.1]. De default waarde staat daarom op 64. Het is toegestaan om afwijkende grootte in te stellen.

IPv6_subnet_min_offset: De minimale offset waarmee gestart wordt voor dit subnet. Bijvoorbeeld wanneer een klant de prefix 2001:db8:fc90::/48 toegewezen krijgt en het eerste subnet voor gebruikers moet beginnen met 2001:db8:fc90:1000::/64 wordt de minimale offset 0:0:0:1000::

IPv6_subnet_max_offset: De maximale offset waarmee gestart wordt voor dit subnet. Bijvoorbeeld wanneer het laatste gebruikers subnet 2001:db8:fc90:3ffe::/64 zou moeten zijn, dan is de waarde van de maximale offset 0:0:0:3ffe::

IPv6_subnet_name: De toegewezen naam voor het subnet. Wanneer een subnet aangemaakt wordt, krijgt dit subnet standaard deze naam. Binnen het ontwerp proces wordt deze naam als referentie gebruikt. De uiteindelijke subnetten kunnen van naam veranderen.

IPv6_subnet_remark: Een optioneel veld waarin een ontwerper notities kan maken of keuzes kan toelichten voor het betreffende subnet.

4.4 Tabel: IPv6_site_plan

De IPv6_site_plan tabel wordt gebruikt voor het site masker. Het masker wordt als een IPv6 adres opgeslagen. De tabel maakt gebruik van de kolom IPv6_plan_id uit de tabel IPv6_plan.

| Tabel: IPv6_site_plan | | |
|-----------------------|---------------|--|
| Kolom | Type | Attributen |
| IPv6_plan_id | Integer (11) | NOT NULL |
| IPv6_site_mask | varbinary(16) | NOT NULL |
| Timestamp | timestamp | NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP |

IPv6_plan_id: Gebruikt als referentie vanuit de tabel IPv6_plan.

IPv6_site_mask: Het site masker. Opgeslagen als IPv6 adres.

4.5 Tabel: IPv6_subnet_plan

De IPv6_subnet_plan tabel wordt gebruikt om de specifieke informatie voor elk type IPv6 subnet uit een IPv6 plan. In deze tabel wordt bijvoorbeeld opgeslagen welke VLAN id('s) gebruikt worden voor een subnet, welke gateway adressen gebruikt worden etc.

De tabel maakt gebruik van de kolommen IPv6_plan_id en IPv6_subnet_type uit de tabel IPv6_plan.

| Tabel: IPv6_subnet_plan | | |
|-------------------------|--------------|--|
| Kolom | Type | Attributen |
| Key_id | Integer (11) | NOT NULL AUTO_INCREMENT UNIQUE |
| IPv6_plan_id | Integer (11) | NOT NULL |
| IPv6_subnet_index | Integer (10) | NOT NULL |
| Net_parameter | VARCHAR (20) | NOT NULL DEFAULT 64 |
| Net_key | VARCHAR (30) | NOT NULL |
| Net_value | VARCHAR (32) | NOT NULL |
| Net_notes | mediumtext | NOT NULL |
| Timestamp | timestamp | NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP |

Key_id: Een autonummer veld voor intern gebruik.

IPv6_plan_id: Gebruikt als referentie vanuit de tabel IPv6_plan.

IPv6_subnet_type: Gebruikt als referentie vanuit de tabel IPv6_plan.

Net_parameters: De gebruikte parameter. Deze waarden zijn dezelfde als de parameterwaarden vanuit de IPv4 subnet plannen.

Net_key: Beschrijft de functie van de Net_value. Bijvoorbeeld een IPv6 offset, VLAN reeks, of een scope. Deze waarden zijn dezelfde als de key waarden vanuit de IPv4 subnet plannen.

Net_value: De waarde van de parameter. Wanneer een IPv6 offset opgeslagen wordt, moet deze een geldige waarde hebben. De IPv6 offsets en adressen moeten opgeslagen worden als hex waarde.

Net_notes: Notities voor het record. Ontwerpers kunnen hier notities maken of hun keuzes toelichten.

4.6 Tabel: IPv6_prefix

In de IPv6_prefix tabel worden de IPv6 prefixen opgeslagen die toegewezen zijn aan een klant. De prefix kan ook toegewezen worden aan een locatie van een klant. Dit wordt ook in deze tabel opgeslagen.

De tabel maakt gebruik van kolommen ClientCode uit de Client tabel, SiteCode uit de Site tabel en IPv6_plan_id uit de IPv6_plan tabel.

| Tabel: IPv6_prefix | | |
|--------------------|---------------|--|
| Kolom | Type | Attributen |
| <u>IPv6_prefix</u> | varbinary(16) | NOT NULL UNIQUE |
| ClientCode | VARCHAR (30) | NOT NULL |
| SiteCode | VARCHAR (30) | DEFAULT " |
| Net_status | Integer (1) | NOT NULL DEFAULT 64 |
| IPv6_plan_id | Integer (11) | NOT NULL |
| IPv6_prefix_size | Integer (3) | NOT NULL DEFAULT 48 |
| Scope | Integer (1) | DEFAULT 0 |
| Timestamp | timestamp | NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP |

IPv6_prefix: De toegewezen prefix. Alle waarden moeten uniek zijn.

ClientCode: Referentie naar de Client tabel. Wanneer de prefix niet aan een klant is toegewezen wordt er geen waarde ingevuld.

SiteCode: Referentie naar de Site tabel. Wanneer de prefix niet aan een site is toegewezen wordt er geen waarde ingevuld.

Net_status: De status van een prefix. Geldige statussen worden ingevoerd vanuit de Lookup tabel.

IPv6_plan_id: Referentie naar naar de IPv6_plan tabel.

IPv6_prefix_size: De grootte van de prefix. Standaard is deze 48.

Scope: De scope van de prefix. Geldige waarden worden uit de Lookup tabel gehaald.

4.7 Tabel: IPv6_subnet

De IPv6_subnet tabel wordt gebruikt om uitgedeelde IPv6 subnetten op te slaan. De tabel maakt gebruik van de kolommen IPv6_plan_id, IPv6_prefix en ClientCode vanuit de IPv6_prefix tabel. IPv6_subnet_type vanuit de IPv6_plan tabel. SiteCode vanuit de Site tabel en Service_key uit de Service_keys tabel.

| Tabel: IPv6_subnet | | |
|--------------------|---------------|--|
| Kolom | Type | Attributen |
| IPv6_subnet_id | Integer (11) | NOT NULL AUTO_INCREMENT |
| IPv6_plan_id | Integer (11) | NOT NULL |
| IPv6_subnet_type | Integer (10) | NOT NULL |
| IPv6_subnet_name | VARCHAR (50) | DEFAULT '' |
| IPv6_prefix | varbinary(16) | DEFAULT '' |
| IPv6_subnet_offset | varbinary(16) | NOT NULL |
| IPv6_subnet_size | Integer (3) | NOT NULL DEFAULT '64' |
| Vlan_id | Integer (4) | DEFAULT NULL |
| Vlan_template | VARCHAR (30) | DEFAULT '' |
| Vrf_id | Integer (6) | DEFAULT '' |
| ClientCode | VARCHAR (30) | DEFAULT '' |
| SiteCode | VARCHAR (20) | DEFAULT '' |
| IPv6_gateway | varbinary(16) | DEFAULT '' |
| IPv6_loopback | varbinary(16) | DEFAULT '' |
| Net_status | Integer (1) | NOT NULL DEFAULT '0' |
| Net_notes | mediumtext | NOT NULL DEFAULT '0' |
| Service_key | Integer (11) | NOT NULL DEFAULT '0' |
| Sid | Integer (11) | |
| Timestamp | Timestamp | NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP |

IPv6_subnet_id: Een auto nummer veld voor interne referentie.

IPv6_plan_id: Het IPv6 plan id uit de IPv6_prefix tabel.

IPv6_subnet_type: Het IPv6 subnet type.

IPv6_subnet_name: De naam voor het subnet. De standaard naam is de naam vanuit het subnetplan. De naam kan aangepast worden door een engineer.

IPv6_prefix: De prefix waar het subnet uit gecreëerd is.

IPv6_subnet_offset: De offset voor dit subnet.

IPv6_subnet_size: De grootte van het subnet

Vlan_id: Het VLAN id voor dit subnet. Standaard is de waarde NULL. Wanneer de waarde gezet wordt, moet deze tussen de 1 en de 4094 zitten.

Vlan_template: Optionele waarde voor een template van dit subnet.

Vrf_id: Optionele waarde. Referentie naar een MPLS VRF wanneer dit gebruikt wordt.

ClientCode: De klant code.

SiteCode: De site code

IPv6_gateway: Het IPv6 adres wat gebruikt wordt als de gateway.

IPv6_loopback: Het IPv6 adres wat gebruikt wordt als loopback adres.

Net_status: De status van dit subnet.

Net_notes: Optioneel veld waarin ontwerpers en engineers notities kunnen maken en keuzes kunnen toelichten.

Service_key: Referentiewaarde naar de Service_keys tabel.

Sid: Een intern nummer om bij te kunnen houden waar het subnet gebruikt wordt. Deze functie wordt op dit moment nog ontwikkeld. De functionaliteit kan in de toekomst dus nog veranderen.

4.8 Tabel: IPv6_map

De IPv6_map tabel wordt gebruikt om IPv6 adressen of een IPv6 subnet te koppelen aan een interface van een node.

De tabel maakt gebruik van de kolommen Interface_id uit de tabel Port_map en IPv6_subnet_id uit de IPv6_subnet tabel.

| Tabel: IPv6_map | | |
|-----------------------|-------------|--|
| Kolom | Type | Attributen |
| <u>Interface_id</u> | Integer(11) | NOT NULL |
| <u>IPv6_subnet_id</u> | Integer(11) | NOT NULL |
| Topo_pos | VARCHAR (2) | DEFAULT NULL |
| <u>IPv6_parameter</u> | VARCHAR(32) | DEFAULT NULL |
| Sid | Integer(11) | NOT NULL |
| Timestamp | timestamp | NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP |

Interface_id: De Interface_id uit de Port_map tabel.

IPv6_subnet_id: De IPv6_subnet_id uit de IPv6_subnet_tabel.

Topo_pos: De topologische positie voor de interface. Mogelijke waarden zijn M (Management), L (Loopback), NA, NB, ZA, ZB, ZV, A, B, N en Z. Met de uitzondering van M en L refereren alle waarden naar de hiërarchie van de topologie.

IPv6_parameter: De parameter welke toegewezen is aan de interface. Dit kan een IPv6 adres zijn of een te berekenen waarde (te herkennen aan een variabele tussen <>).

Sid: Een intern nummer om bij te kunnen houden waar het subnet gebruikt wordt. Deze functie wordt op dit moment nog ontwikkeld. De functionaliteit kan in de toekomst dus nog veranderen.

4.9 View: IPv6_prefix_view

De IPv6_prefix_view toont alle kolommen van de IPv6_prefix tabel. De IPv6 prefix wordt weergegeven als een IPv6 adres door gebruik van de functie `inet6_ntoa()`.

4.10 View: IPv6_subnet_view

De IPv6_subnet_view toont alle kolommen van de IPv6_subnet tabel met uitzondering van de kolommen IPv6_subnet_id, IPv6_plan_id en IPv6_subnet_type. De varbinary(16) kolommen uit de tabel worden weergegeven als IPv6 adres.

4.11 View: IPv6_map_view

De IPv6_map_view toont de interface naam, de topologische positie van de interface en de IPv6 parameter. Wanneer de IPv6 parameter een IPv6 adres is, wordt deze als een IPv6 adres getoond. De interface naam komt uit de tabel Port_map. De overige gegevens komen uit de tabel IPv6_map.

4.12 Aanpassingen aan reeds bestaande tabellen.

In de Domain tabel worden op dit moment adressen opgeslagen van NMS servers in het netwerk. Deze kolommen zijn nu van het type VARCHAR(20). Om IPv6 adressen op te kunnen slaan moet dit aangepast worden naar VARCHAR(39). Het gaat om de volgende kolommen:

- Syslog_server1
- Syslog_server2
- Snmp_traphost1
- Snmp_traphost2
- Tacacs_server1
- Tacacs_server2

5 *Bronnen en referenties*

| | |
|------------------------------------|--|
| [1] | Overview of new features in Apache HTTP Server 2.0 |
| [ARIN-IPv6] | ARIN number resource policy manual |
| [Cisco-IPv6] | Cisco Validated Design, IPv6 Addressing White paper, August 2013 |
| [IANA-IPv6-AS] | Internet Protocol Version 6 Address Space |
| [IANA-IPv6-GUAA] | IPv6 Global Unicast Address Assignments |
| [IANA-IPv6-Reg] | IANA IPv6 Special-Purpose Address Registry |
| [RFC3849] | IPv6 Address Prefix Reserved for Documentation , July 2004 |
| [RFC4291] | IP Version 6 Addressing Architecture, Feb 2006 |
| [RFC4862] | IPv6 Stateless Address Autoconfiguration, September 2007 |
| [RFC5737] | IPv4 Address Blocks Reserved for Documentation, January 2010 |
| [RFC5952] | IPv6 Text Representation, August 2010 |
| [RIPE589] | IPv6 Address Allocation and Assignment Policy, May 2013 |

Bijlage VII: Implementatievoorstel

Implementatievoorstel

IPv6 implementatie in NetYCE

Derco Sportel

1548287

Colofon

| | |
|--------------------|--|
| | NetYCE B.V. Postbus 320 1380 AH Weesp, Nederland Business Centre Muiderpoort Leeuwendveldseweg 5n 1382 LV Weesp, Nederland |
| Auteur | Derco Sportel <i>Network Consultant</i> ☎ +31 6 1418 1361 ✉ derco.sportel@netyce.com 🌐 derco.sportel |
| Docentbegeleider | Kees Uiterwijk <i>Hogeschooldocent, Systeembeheer</i> ☎ +31 88 481 8184 ✉ kees.uitervijk@hu.nl 🌐 kees.uitervijk |
| Bedrijfsbegeleider | Ronny Lam <i>Solution Architect</i> ☎ +31 6 3976 6145 ✉ ronny.lam@netyce.com 🌐 ronny_lam |
| Versie | 1.0 |

Document

Document geschiedenis

| Versie | Datum | Auteur | Notities |
|--------|--------------|---------------|---------------------------|
| 0.1 | 2 sept 2014 | Derco Sportel | Eerste opzet. |
| 0.2 | 17 sept 2014 | Derco Sportel | Aanpassingen nav feedback |
| 0.3 | 17 sept 2014 | Derco Sportel | Aanpassingen nav feedback |
| 1.0 | 24 sept 2014 | Derco Sportel | Definitieve versies |

Distributie

| Name | 0.1 | 0.2 | 0.3 | 1.0 |
|---------------------|-----|-----|-----|-----|
| Derco Sportel | ✓ | ✓ | ✓ | ✓ |
| Eric Yspeert | | ✓ | ✓ | ✓ |
| Ronny Lam | ✓ | ✓ | ✓ | ✓ |
| Kees Uiterwijk (HU) | | | | ✓ |
| Gerald Ovink (HU) | | | | ✓ |
| J de Doelder (CvT) | | | | ✓ |

Inhoudsopgave

| | |
|---|----|
| Colofon..... | 2 |
| Document..... | 3 |
| Document geschiedenis..... | 3 |
| Distributie..... | 3 |
| 1 Inleiding..... | 5 |
| 2 Aanpak en tijdslijn..... | 6 |
| 2.1 Fasering..... | 6 |
| 2.2 Git IPv6 branch..... | 6 |
| 3 Fase 1: Proof of Concept opstelling..... | 7 |
| 3.1 Database aanpassing..... | 7 |
| 3.2 IPv6 communicatie naar de netwerkkapapparaat..... | 7 |
| 3.3 Client details formulier..... | 7 |
| 3.4 Site details formulier..... | 7 |
| 3.5 Node details formulier..... | 7 |
| 3.6 Relaties voor het gebruik in templates..... | 7 |
| 4 Fase 2: Volledige implementatie..... | 8 |
| 4.1 Frontend..... | 8 |
| 4.2 Backend..... | 8 |
| 5 Documentatie..... | 9 |
| 6 Bronnen en referenties..... | 10 |

1 Inleiding

Dit document beschrijft de implementatie van IPv6 ondersteuning in netYCE, alsmede de instandhouding van de software en de documentatie.

Als basis voor de implementatie worden het Functioneel^[1] en Technisch^[2] ontwerp genomen

2 Aanpak en tijdslijn

2.1 Fasering

De ontwikkeling zal in 2 fases uitgevoerd worden. De Proof of Concept fase en de volledige implementatiefase.

De PoC fase zal zich beperken tot de database aanpassing, het tonen van gegevens in de formulieren en de communicatie tussen de NetYCE server en de netwerkapparatuur.

In de volledige implementatiefase zal alle functionaliteit ontwikkeld worden. Bij het ontwikkelen van de IPv6 functionaliteit moet er per use case gewerkt worden. Een use case wordt ontwikkeld, getest en gedocumenteerd alvorens door te gaan naar de volgende use case.

Hoewel de originele ontwikkeling bedacht was om eind oktober 2014 te hebben afgerond moet deze planning naar achteren geschoven worden. Een nieuwe einddatum is nog niet bekend. De PoC fase moet wel eind oktober 2014 gereed zijn.

2.2 Git IPv6 branch

Bij de huidige ontwikkeling van NetYCE wordt gebruik gemaakt van Git. Hiermee wordt versie beheer gedaan voor de software en wordt de code afgesplitst (een branch) om bepaalde functionaliteit te ontwikkelen. Eenmaal ontwikkeld, wordt de branch samengevoegd met de master branch.

Voor de ontwikkeling NetYCE worden op dit moment 2 branches gebruikt. De master- en de development branch. Voor de ontwikkeling van IPv6 functionaliteit is nieuwe branch afgesplitst van de development branch. In deze branch wordt de IPv6 functionaliteit ontwikkeld. Zodra deze productierijp is wordt de branch samengevoegd met de development branch, getest en vervolgens als productieversie naar de klanten gedistribueerd.

3 Fase 1: Proof of Concept opstelling

Voor de PoC opstelling zal een beperkt gedeelte van de functionaliteit geïmplementeerd worden. Deze functionaliteit beperkt zich tot het tonen van de gegevens en nog niet de invoer van nieuwe gegevens en de validatie van de ingevoerde gegevens.

3.1 Database aanpassing

Voor de Proof of Concept is het van belang dat de gegevens uitgelezen kunnen worden door de formulieren. Hierom dient de database in een vroeg stadium aangepast te worden voor de opslag van de IPv6 informatie. De aanpassing zal gedaan worden door middel van een patchfile. Deze methode wordt reeds gebruik voor database aanpassingen en hier is van aangetoond dat dit werkt. De patchfile is ontwikkeld in de IPv6 branch.

3.2 IPv6 communicatie naar de netwerkkaparaatuur

In de PoC is het van belang dat er met IPv6 gecommuniceerd kan worden met de netwerkkaparaatuur. Hiervoor dient sectie 3.4.1 uit het FO^[1] geïmplementeerd te worden.

3.3 Client details formulier

In het client details formulier zullen de toegewezen IPv6 prefixen getoond worden. De toegewezen IPv6 prefixen worden naast de huidige IPv4 supernetten getoond in een apart grid.

Het tonen van deze gegevens heeft een relatie met use case 5 uit het TO^[2].

3.4 Site details formulier

In het site details formulier worden de grids iets aangepast om ruimte te maken voor de site IPv6 prefix. Toevoegen en verwijderen van de prefixen wordt in deze fase nog niet geïmplementeerd.

Het tonen van deze gegevens heeft een relatie met use case 6 uit het TO^[2].

3.5 Node details formulier

In het node details formulier wordt het management IPv6 adres getoond onder het management IPv4 adres. Wijzigen van het IPv6 adres is in deze fase nog niet mogelijk.

Het tonen van deze gegevens heeft een relatie met Use case 8 uit het TO^[2].

3.6 Relaties voor het gebruik in templates

Om op de juiste manier configuraties te kunnen genereren maakt NetYCE gebruik van relaties. Dit zijn SQL query's waarmee een eindgebruiker in templates een variable naam kan gebruiken. De variable wordt tijdens het genereren van een template vervangen voor de werkelijke waarde.

Zowel in de PoC fase als in de volledige implementatiefase zullen er verschillende relaties gecreëerd worden, zodat eindgebruikers deze relaties direct kunnen gebruiken. Het is voor de eindgebruikers ook mogelijk hun eigen relaties te ontwikkelen.

4 Fase 2: Volledige implementatie

4.1 Frontend

Bij het verder ontwikkelen van de frontend moet per use case gewerkt worden. Net als bij de backend ontwikkeling moet elke use case getest en gedocumenteerd worden voordat er doorgedaan wordt naar de volgende use case. Voor de ontwikkeling van 1 use case zal tussen de 3 en 10 werkdagen benodigd zijn.

Het ontwikkelen van de use cases moet gebeuren op volgorde zoals deze staan in het TO^[2]. Dit vanwege de samenhang tussen de use cases.

Use case 3 is afhankelijk van de servicetask module in de backend en de database aanpassingen die worden gedaan in PoC fase. Deze use case heeft verder geen raakvlak met de frontend.

4.2 Backend

Het uitvoeren van de service tasks is afhankelijk van de Service Task module in de backend. Tijdens de volledige implementatie moet deze module ontwikkeld worden. In hoofdstuk 3 van het TO^[2] staan de verschillende taken beschreven. De ontwikkeling van deze taken moet per taak worden ontwikkeld, getest en gedocumenteerd alvorens door te gaan aan de volgende taak.

De ontwikkeling van deze module zal naar verwachting 10 tot 15 dagen in beslag nemen.

5 Documentatie

De gebruikersdocumentatie moet tijdens de ontwikkeling geschreven worden. Elke functie moet hierbij worden gedocumenteerd. Hiervoor wordt het documentatieplatform^[3] van NetYCE gebruikt.

Naast dat de functies gedocumenteerd worden, moeten er ook howto's geschreven worden over het gebruik van IPv6 met NetYCE. Dit beperkt zich echter niet tot IPv6 maar heeft betrekking op de gehele applicatie. Inmiddels is de wens uitgesproken om de documentatie volledig te herzien. Naast de functionele documentatie moeten er howto's en andere handleidingen ontwikkeld worden. Dit zal als een apart project behandeld worden waar op dat moment de IPv6 handleidingen in meegenomen worden.

6 *Bronnen en referenties*

- [1] Functioneel ontwerp IPv6 in NetYCE versie 1.0
- [2] Technisch ontwerp IPv6 in NetYCE versie 1.0
- [3] NetYCE Documentatie: <https://wiki.netyce.com/>