

Självständigt arbete på grundnivå

Independent degree project - first cycle

Datateknik

Computer science

Konsten att delegera prefix

En jämförelse av automatiserad och statisk konfiguration



Mittuniversitetet

MID SWEDEN UNIVERSITY

MITTUNIVERSITETET

Institutionen för informations- och kommunikationssystem (IKS)

Examinator: Magnus Eriksson, magnus.eriksson@miun.se
Handledare: Lennart Franked, lennart.franked@miun.se
Författare: Andreas Eriksson, aner0918@student.miun.se
Joakim Andersson, joan1409@student.miun.se
Utbildningsprogram: Nätverksdrift, 120 hp
Huvudområde: Nätverksteknik
Termin, år: vt, 2016

Sammanfattning

När det stod klart att antalet adresser i ipv4 inte skulle räcka, togs ipv6 fram. Med nya funktioner som hade anpassats till den infrastruktur som hade byggts upp världen runt så fanns det nu en ersättare till ipv4. Men skillnaderna är stora och protokollen är i sig inte lika varandra alls, framför allt inte när man tittar på inbygda funktioner. En av dessa nya funktioner i ipv6 är det som kallas för Prefix delegation, ett enkelt och smidigt sätt att per automatik dela ut delar av ett större nät till mindre nät. Även om det finns enkla och smidiga sätt att genomföra detta på så är inte all infrastruktur så enkel. Det gör att denna typ av funktioner kommer varvas med routes, både statiska och dynamiska. För att få en förståelse för hur detta kan skilja sig åt har vi tittat på en översikt av hur infrastrukturen kan se ut och sedan testat hur man skulle kunna lösa adresseringen genom en automatiserad process och en statisk process. Resultatet visar att den automatiserade lösningen kräver lite mer förarbete och specificerad utrustning, men att arbetet därefter blir lätt att kontrollera, felsöka och dokumentera. Den statiska processen är mer rakt på sak och kräver ingen speciell utrustning, däremot kommer det bli svårt att kontrollera, felsöka och dokumentera då minsta förändring kan kräva mycket arbete där risken för komplikationer är stor.

Nyckelord: Nätverksteknik, Datakommunikation, ipv6, Prefix delegation, Cisco

Abstract

When it became clear that the number of ipv4 addresses would not be enough, ipv6 was developed. With new features that had been adapted to the infrastructure that had been built up around the world, there was now a replacement for ipv4. But the differences are many and the protocols are not alike at all, especially when looking at the built-in functions. One of these new features in ipv6 is called Prefix Delegation, a simple and convenient way to automatically distribute parts of a larger network into smaller networks. Although there are simple and flexible way to implement this, the existing infrastructure of the Internet may not be so simple. The complexity of the infrastructure forces us to use a variety of routes, both static and dynamic. To get an understanding of how this can differ, we looked at an overview of how the infrastructure might look like and then tested how to solve the addressing through an automated process and a static process. The results shows that the automated solution requires a little more preparation and specified equipment, but the work then becomes easy to control, debug, and document. The static process is more straightforward and requires no special equipment, however, it will be difficult to control, debug, and document where the slightest change can require a lot of work where the risk of complications is great.

Keywords: Network technology, Computer communication, ipv6, Prefix delegation, Cisco

Förord

Detta projektarbetet har genomförts våren 2016 inom ramen av programmet Nätverksdrift på Mittuniversitet, Sundsvall. För att förstå innehållet som finns i rapporten bör man ha grundläggande förståelse inom följande delar; IP, ipv6, DHCP samt datortekniska begrepp som kan återfinnas inom denna bransch.

Vi har genomfört detta arbete på företaget TDC Sverige, med handledare Karl-Erik Svedin som vi vill uttrycka ett stort tack för det stöd och den stora kunskap inom just detta ämne. Vi tackar även Lennart Franked som var vår handledare på Mittuniversitet som vi har kunnat ställa frågor angående själva rapportens utförande samt vägledning framåt. Samt ett bra samarbete med vår examinator Magnus Eriksson.

Vi tackar även vänner och familjer som ställt upp under detta arbete.

Mittuniversitet, Sundsvall 2016-06-07

Innehåll

Sammanfattning	3
Abstract	4
Förord	5
1 Inledning	7
1.1 Bakgrund	7
1.2 Syfte	7
1.3 Avgränsningar	7
1.4 Konkreta och verifierbara mål	7
1.5 Författarens bidrag	7
2 Teori	8
2.1 Definition av termer och förkortningar	8
2.2 Teori	9
2.3 Funktioner i ipv6	9
2.4 ipv6 Adressering	9
2.5 NDP - Neighbor Discovery Protocol	10
2.6 DHCPv6	11
3 Metod	12
3.1 Testplattform	12
3.2 Scenario 1: Prefix delegation	13
3.3 Scenario 2: Statisk konfiguration	14
4 Resultat	15
4.1 Scenario 1	15
4.2 Scenario 2	18
4.3 Resultat av andra lösningar	20
5 Slutsatser	21
5.1 För och nackdelar	21
5.2 Diskussion	21
5.3 Slutsats	22
5.4 Fortsatt forskning	22
6 Referenser	23
7 Bilagor	24
7.1 Konfigurationsfiler: Scenario 1	24
7.2 Konfigurationsfiler: Scenario 2	29

1 Inledning

1.1 Bakgrund

Året är 1981 och ipv4 får sin första RFC. RFC 791 fastställer en adress på 32 bitar som ger oss ca 4,3 miljarder adresser. Tyvärr räcker inte dessa adresser i dag, vilket man insåg redan i början på 1990-talet och skapade därför ipv6, vars 128 bitar långa adress kan ge oss ca $3,4 * 10^{38}$ adresser. ipv6 har en helt annan uppbyggnad än ipv4 vilket gör att flera funktioner finns inbyggt i adressen, där de 64 första bitarna bildar ett prefix och de sista 64 bitarna en unik adress. En mycket viktig del av den nya adressen är funktionen i den 64 bitar långa prefixet, vilket bland annat ersatt nätmasken som är livsviktig för ipv4. För att kunna dela ut nät och adresser så måste en server kunna dela upp prefixet och dela ut delar av det till sina kunder. Detta görs genom Prefix Delegation och är en mekanism som finns i NDP. Vi har, tillsammans med TDC Sverige, kommit fram till att detta är en intressant del att lära sig samtidigt som TDC Sverige ser ett framtida behov av att kunna leverera ipv6 som standard till privatkunder, något de enbart levererar till företag idag, och då har vi genomfört denna jämförelse för att belysa de för- och nackdelar som finns med tekniken.

1.2 Syfte

Tillsammans med TDC Sverige har vi valt att titta närmare på och jämföra olika lösningar för hur t.ex. en internetleverantör kan dela upp stora nätsegment till mindre nätsegment i ett nätverk som baseras på ipv6. Denna jämförelse ska ge en översiktlig bild av för- och nackdelar, som är tänkt att användas som grund för TDC Sverige att arbeta vidare på vid eventuell implementering av ipv6 för privatkunder.

1.3 Avgränsningar

- Projektet kommer utgå från Cisco och Juniper i form av utrustning och testplattform
- Projektet kommer enbart fokuseras på lösningar för ipv6

1.4 Konkreta och verifierbara mål

- Prefix Delegation
 - För / nackdelar prefix delegation och statisk konfiguration inom ipv6.
 - Skapa scenario med Prefix Delegation samt scenario med Statisk konfiguration
- Test av konfiguration
 - Skapa en testplattform som är översiktlig och skalbar
 - Skapa en testplattform som baseras på skarp driftmiljö
 - Verifiera att CPE får korrekt antal nät från PE
- Jämföra statisk och automatisk konfiguration
 - Översiktligt jämföra de båda lösningarna
 - Tabell med för- och nackdelar
- Undersöka andra lösningar
 - Undersöka om det finns andra sätt än statisk konfiguration och PD för att lösa samma funktion
 - Om det finns andra lösningar, göra en kort presentation av den/de lösningen-/arna

1.5 Författarens bidrag

Arbetet kan delas in i två olika arbetssätt: Teoretisk och praktisk. Den praktiska delen har varit utformningen och arbetet i testplattformen. Denna del av arbetet har vi hela tiden, från att skapa den till att genomföra tänkt konfiguration och test av konfiguration, genomfört tillsammans på TDC Sverige i Sundsvall.

Den teoretiska delen har till viss del genomförts tillsammans på plats på TDC Sverige, men en majoritet har genomförts på varsitt håll. Vi har inte haft en spikad uppdelning utan har tagit del för del och arbetat med att hela tiden fördela det som ska göras mellan varandra. Vi har därefter läst igenom varandras resultat och kommit med förslag om eventuella förändringar. Vi har även samarbetat mycket genom Skype och andra sociala medier.

2 Teori

2.1 Definition av termer och förkortningar

- IP, ipv4, ipv6
IP, förkortning för Internet Protocol, är en protokollstack som innehåller de protokoll vi behöver för att kunna skicka data mellan olika system. ipv4 betyder IP version 4 och är egentligen samma sak som förkortningen IP. ipv6 står för IP version 6 och är en vidareutveckling av ipv4. I detta arbete kommer ipv4 och ipv6 att användas för att inte skapa förvirring vilket av protokollen som används.
- DHCP, DHCPv6
Förkortning för Dynamic Host Configuration Protocol. En funktion för att automatiskt dela ut adresser och annan information från en server till anslutande enheter för att slippa manuellt konfigurera alla nya enheter som ansluts till ett nätverk. DHCPv6 är en anpassad version för de funktioner som finns i ipv6.
- PD
Förkortning för Prefix delegation. En funktion för att låta en enhet, t.ex. en router, fördela ett större nätsegment till mindre nätsegment och delegera detta till andra routrar. Funktionen är en del av Neighbor Discovery Protocol.
- Cisco, Juniper
Både Cisco och Juniper är tillverkare av utrustning för nätverk. De har även sin egen mjukvara för konfiguration av utrustning. Cisco har IOS och Juniper Junos.
- RFC
Förkortning för Request for comments. Officiell dokumentation som beskriver t.ex. funktioner, protokoll eller ny teknik.
- Route
En route är en rutt mellan 2 routrar.
- Prefix
Prefixet är de 64 första bitarna i en ipv6 adress. Denna del av adressen specificerar vilket nät som används och därmed så gör adressen routingsbar.
- ISP
Förkortning för Internet Service Provider. Leverantör av uppkoppling mot internet.
- PE, CPE
Förkortning för Provider Edge och Customer Premesis Equipment. PE är den punkt dit t.ex. en ISP har ansvaret för utrustningen. CPE är den utrustning som kunden själv har ansvar för.
- NDP
Förkortning för Neighbor Discovery Protocol. Protokoll i ipv6 för att upptäcka grannar i samma nätverk.
- SLAAC
Förkortning för Stateless Address Autoconfiguration. Funktion för en enhet att automatiska skapa sin egen unika ipv6 adress.
- DNS
Förkortning för Domain Name System. Ett system för att associera domännamn och dess IP-adress, samt att förmedla denna information till klienter vid begäran.
- ICMP, ICMPv6
Förkortning för Internet Control Message Protocol. Ett protokoll som generellt används för t.ex. felmeddelanden. Finns även som anpassad version för ipv6 och kallas då ICMPv6.
- DUID
Förkortning för DHCPv6 Unique Identifier. En identifierare för varje enhet som aktivt använder DHCPv6.
- ARP
Förkortning för Address Resolution Protocol. Protokoll för kartläggning av IP adresser samt deras hårdvaruspecifika adress (MAC Adress).

- IPsec
Förkortning för Internet Protocol Security. Protokollstack för autentisering och kryptering av kommunikation inom IP-protokollet.

2.2 Teori

För att kunna delegera nät och adresser från t.ex. en ISP till slutkund så går det att använda ett par olika tekniker. För att efterlikna verkligheten så långt som möjligt har vi valt en metod där vi använder NDP (Neighbor Discovery Protocol) för tilldelning av prefix och SLAAC för den unika adresseringen. DNS och Domain-name kommer att hämtas från DHCPv6 server.

2.3 Funktioner i ipv6

- Adress:
Med en 128-bitar lång adress skapar vi möjlighet till globalt unika adresser till otaliga enheter. Det finns även ett flertal olika typer av adresser vilket gör att det blir smidigare och effektivare hantering av adresseringen.
- Nivåer av adressering:
Med ett prefix på 64-bitar går det att dela upp adresser i väldigt många nivåer, t.ex. en nivå för ISP i viss storlek medan större/mindre ISP kan tilldelas en annan nivå. Det blir betydligt lättare att bygga upp en hierarki på detta sätt.
- Delegering:
Stöd för automatisk delegering av prefix.
- Automatisk konfiguration:
Funktion för att skapa unika adresser, utan inblandning av t.ex. en DHCP server, genom SLAAC.
- Renumbering:
Automatisk funktion för att ändra tilldelade nät, t.ex. en router kan meddela en host i nätverket att nätverks prefix ändrats och därmed måste hostens adress ändras, vilket kommer ske automatiskt.
- Multicast:
Istället för att använda ARP som skickade broadcasts till hela subnätet går det att använda egna multicast adresser vilket gör att vi kan välja vilka som ska ta del av viss information.
- Header:
Effektivare header genom att förlänga headers storlek (40 bytes) och enbart 8 fält.
- Säkerhet:
Stöd för bland annat IPsec finns inbyggt i protokollet.[1, kap.3] [2]

2.4 ipv6 Adressering

ipv6 har en adress som är uppbyggd av 128 bitar. För att underlätta för oss människor att läsa och komma ihåg adresserna så delas den upp i 8 grupper om 4 hexadecimala tecken i varje grupp. För att ytterligare förenkla adresserna så finns en funktion som gör att adressen kan kortas ner genom komprimering av nollorna i adressen. [1, kap.2] [3] [4]

Full adress: 2001:0DB8:1001:0001:0000:0000:0000:0034

Kort adress: 2001:DB8:1001:1::34

Förkortningen görs genom att vi tar bort första nollan/nollorna i varje grupp, t.ex. 0DB8 blir DB8, däremot går det inte att ta bort nollor i en grupp som har ett tecken framför sig, t.ex. 1001 går inte att förkorta eftersom det finns en 1 innan nollorna i gruppen. Har vi flera grupper med enbart nollor går det att förkorta detta genom att vi använder :: (dubbla kolon), t.ex. 1001:0000:0000:2002 blir 1001::2002. Denna typ av förkortning kan dock bara användas en gång.

Adressen med sina 128 bitar är uppdelad i prefix och interface ID. Prefixet bestämmer vilket nät som används och Interface ID utgör den del som gör att varje adress blir unik. [1, kap.2] [3]

2001:0DB8:1000:2000:3000:ABCD:EF01:2345

|-----| |---| |-----|

Röd text: Globalt prefix

Grön text: Subnät

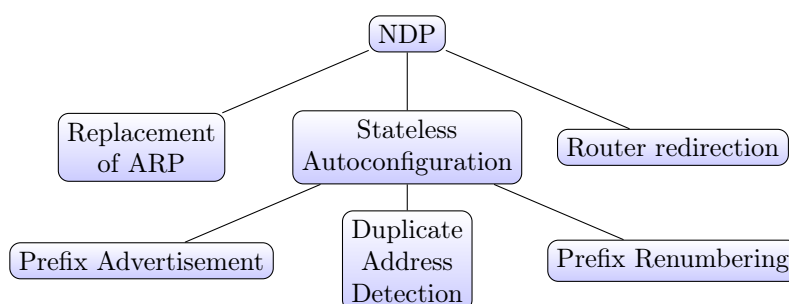
Blå text: Interface ID

Förutom uppdelning mellan nät och adress så finns en uppdelning av prefixet. Hur prefixet är uppdelat finns definierat i Generally Accepted Assignment Policy från RIPE. Uppdelningen ser ut på följande sätt: [5]

- **ISP-Prefix:**
/29 till /32, beroende på omfattningen av den verksamhet man kommer bedriva. Enbart till för ISP som uppfyller vissa krav.
- **Site-Prefix:**
/48. ISP kan tilldela denna nivå till större siter, t.ex. företag.
- **End-site:**
/56. Den tilldelning som är tänkt till slutkund, vilket ger 8 bitar som slutkunden själv kan dela upp sitt nät med, vilket blir totalt 256 nät.

2.5 NDP - Neighbor Discovery Protocol

Figur 1: NDP och dess mekanismer



NDP är ett protokoll som inte enbart används för att hitta sina grannar, vilket namnet ger sken av, det finns ett flertal mekanismer i protokollet. En del av dessa mekanismer är till för att ersätta eller samla protokoll som inom ramen för ipv4 var åtskilda protokoll.

NDP är uppdelat i tre delar; Replacement of ARP vilket är en funktion som ersatt ipv4's funktion ARP genom att använda NS (Neighbor solicitation) och NA (Neighbor advertisement) istället för broadcasts så att enheterna kan hålla koll på MAC-adresserna till sina grannar, Router redirection som kan meddela en sändande enhet att det finns en mer effektiv route att använda och stateless autoconfiguration som, tillsammans med 3 mekanismer, låter enheter själva skapa unika unicast ipv6 adresser. [1, kap.3] [6] De delar som specifikt kommer användas för att uppnå våra mål är stateless autoconfiguration och prefix delegation. NDP använder 5 typer av ICMPv6 för att kommunicera diverse information mellan enheterna.[1, kap.3] [7]

Tabell 1: ICMPv6 typer i NDP

Typ av ICMPv6	Funktion
133	RS, Router Solicitation
134	RA, Router Advertisement
135	NS, Neighbor Solicitation
136	NA, Neighbor Advertisement
137	Redirect message

Stateless Autoconfiguration

Stateless autoconfiguration, även förkortat SLAAC, är en mekanism, som tillsammans med 3 andra mekanismer (prefix delegation, duplicate address detection och prefix renumbering), gör att en enhet på egen hand kan skapa en unicast ipv6 adress. För att kunna skapa sig en egen adress så kommer en enhet som ansluter sig till ett nätverk att skicka ut en Router solicitation där den efterfrågar vilka inställningar, specifika för just det nätverket, som den ska använda. Routers i nätverket kommer då att svara med en Router advertisement som kommer innehålla information om prefix, prefix lifetime, flag information och

information om default router samt dess lifetime. Med denna information kan enheten skapa sitt prefix (de 64 första bitarna av adressen). För att skapa ett unikt interface ID används funktionen EUI-64. EUI-64 använder enhetens 48 bitar långa MAC adress, som är helt unik, för att skapa en 64 bitar lång interface ID. Detta görs genom att 4 stycken F (hexadecimala värdet F) läggs till mitt i MAC adressen. Om vi har följande MAC adress 00:00:02:45:3c:4a så kommer vi få en EUI-64 adress som ser ut på följande sätt 00:00:02:ff:ff:45:3c:4a. För att skilja mellan om det är en unik adress eller ej så används värdet i den 7:e biten i adressen. Om adressen är unik så skall denna bit ha värdet 1 och om den ej är unik så ska den ha värdet 0. [1, kap.3] [8]

Prefix advertisement

Prefix advertisement är en mekanism som har till uppgift att, med hjälp av RA (Router advertisement) "utdela" information till det lokala nätverkets enheter. Denna information består av ipv6 Prefix, lifetime, default router information och flags/options. Genom denna mekanism sprids då det prefix som enheterna skall använda när de skapar sina adresser med hjälp av SLAAC. [1, kap.3] [9]

2.6 DHCPv6

DHCPv6 är en utveckling av DHCP anpassat för ipv6's funktioner. Detta gör att vi har flera olika sätt att använda DHCPv6 på. *Stateful DHCPv6* innebär att en klient kommer att få all nödvändig information från DHCPv6 servern: ipv6 adresser, DNS och domain. *Stateless DHCPv6* innebär att en klient använder SLAAC för att skapa sin unika adress, men att den får viss information från DHCPv6 servern. För att bestämma vilken typ av information som ska hämtas från DHCPv6 så används flaggor i RA ICMPv6 (typ 134). Flaggor som används är M-flaggan (Managed address configuration) och O-flaggan (Other stateful configuration). Om M-flaggan används så begär klienten att få samtlig information från DHCPv6 servern (Stateful DHCP), men om O-flaggan används kommer enbart viss information att hämtas genom DHCP, vanligtvis DNS och domännamn. [10] [11]

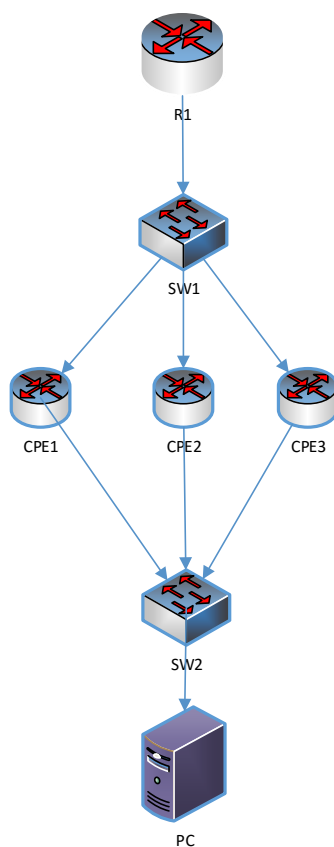
3 Metod

3.1 Testplattform

Tabell 2: Testplattform

Testplattform		
Namn	Typ av utrustning	Information
R1	Cisco 7206vxr	IOS 12.2(33)SRC2
Sw	Cisco 3550	IOS 15.2
CPE1	Cisco 3825	IOS 12.4(15)T9
CPE2	Juniper SXR210BE	Junos 12.1X46-D45.4
CPE3	Cisco 891	IOS 15.2
PC	Dator	MacBook Pro med OSX El Capitan ver 10.11.4

Figur 2: Testmiljö: Logisk topologi



3.2 Scenario 1: Prefix delegation

I detta scenario skall R1 fungera som delegerande enhet. Den skall dela upp ett /48 nät till /56 nät till CPE. CPE ska sedan skapa egna adresser med SLAAC men få både domain och DNS från DHCPv6. Syftet med detta scenario är att undersöka för- och nackdelar med en automatiserad lösning av nätsegmenteringen från stort till mindre nät.

Följande punkter skall genomföras i detta scenario:

- Konfigurera R1 att agera delegerande router för PD
- Konfigurera R1 att agera som DHCPv6 server för DNS och Domain
- Konfigurera CPE1-3 att ta emot prefix från delegerande router
- Konfigurera CPE1-3 att genom DHCPv6 hämta information om DNS (2001:DB8:3::67) och Domain (testdomain.com) från delegerande router
- Verifiera konfiguration och funktionalitet

Uppdelningen av näten ska ske enligt följande schema:

Tabell 3: Fördelning nät

Namn	Nät	DUID
R1	2001:DB8:3::/48	
CPE1	2001:DB8:3:100::/56	00030001001DE64EC630
CPE2	2001:DB8:3:500::/56	0003000188E0F32DAD80
CPE3	2001:DB8:3:300::/56	00030001CCEF48BB93AC
Ospecificerad CPE	2001:DB8:3::/56	

3.3 Scenario 2: Statisk konfiguration

I detta scenario skall samtliga enheter vara statiskt konfigurerat. Uppdelningen av nätet skall gå från ett /48 nät till ett /56 nät per slutkund. CPE1-2 ska skapa egna adresser med SLAAC men få både domain och DNS från DHCPv6. Syftet med detta scenario är att undersöka för- och nackdelar med en statisk lösning av nätsegmenteringen från stort till mindre nät.

Följande punkter skall genomföras i detta scenario:

- Konfigurera statiska routes från R1 till CPE1-2
- Konfigurera statiska routes från CPE1-3 till R1
- Konfigurera CPE1-3 att agera DHCP server för DNS (2001:DB8:1000::1) och Domain-name (Cisco.com)
- Verifiera konfiguration och funktionalitet

Tabell 4: Statiska routes för Scenario 2

Enhet	Från / Adress	Till
R1	FastEthernet0/0, 2001:DB8:1::1	2001:DB8:2::/64 via 2001:DB8:1::2 2001:DB8:3::/64 via 2001:DB8:1::3
R1	Loopback0, 2001:DB8:666::1	Loopback som får agera annat nät
CPE1	GigabitEthernet0/0, 2001:DB8:1::2	::/0 via 2001:DB8:1::1
CPE3	GigabitEthernet0, 2001:DB8:1::3	::/0 via 2001:DB8:1::1

4 Resultat

4.1 Scenario 1

Tabell 5: Översikt, Resultat scenario 1

Resultat, Översikt Scenario 1		
Förväntat resultat	Utfall	Verifierat genom
Korrekt prefix ut från R1 Korrekt DNS och domain-name från R1	Fungerar	show ipv6 dhcp pool på R1
Korrekt prefix till CPE1 - 3 Korrekt DNS och domain-name	Fungerar	show ipv6 dhcp interface (Cisco) Show configuration display set (Juniper)
Korrekt skapad adress på End-host	Fungerar	ipconfig en0 grep inet6 på PC

För att kunna delegera prefix till CPE1-3 där varje CPE ska ha ett specifikt nät, förutom CPE som ej är specificerade i konfigurationsfilen, måste varje nät bindas till det specifika DUID som varje DHCPv6 server och klient har.

Detta har blivit verifierat genom kommandot `show ipv6 dhcp pool` på R1:

```
DHCPv6 pool: to-cpe
Static bindings:
  Binding for client 00030001001DE64EC630
    IA PD: IA ID not specified; being used by 00020001
    Prefix: 2001:DB3:3:100::/56
           preferred lifetime 604800, valid lifetime 2592000
  Binding for client 00030001CCEF48BB93AC
    IA PD: IA ID not specified; being used by 000D0001
    Prefix: 2001:DB3:3:300::/56
           preferred lifetime 604800, valid lifetime 2592000
  Binding for client 0003000188E0F32DAD80
    IA PD: IA ID not specified; being used by 00000000
    Prefix: 2001:DB8:3:500::/56
           preferred lifetime 604800, valid lifetime 2592000
Prefix pool: cpepool1
           preferred lifetime INFINITY, valid lifetime INFINITY
DNS server: 2001:DB8:3::67
Domain name: testdomain.com
Active clients: 3
```

Verifiering att CPE1 har fått korrekt prefix genom kommando: `show ipv6 dhcp interface`, rad 10 visar vilket prefix som delegerats från R1:

```
GigabitEthernet0/0 is in client mode
State is OPEN
Renew will be sent in 3d11h
List of known servers:
  Reachable via address: FE80::206:D6FF:FE1D:8000
  DUID: 000300010006D61D8000
  Preference: 0
Configuration parameters:
  IA PD: IA ID 0x00020001, T1 302400, T2 483840
  Prefix: 2001:DB8:3:100::/56
           preferred lifetime INFINITY, valid lifetime INFINITY
DNS server: 2001:DB8:3::67
Domain name: testdomain.com
Information refresh time: 0
```

```
Prefix name: from-7206
Rapid-Commit: disabled
GigabitEthernet0/1 is in server mode
Using pool: to-host
Preference value: 0
Hint from client: ignored
Rapid-Commit: disabled
```

Verifiering att CPE2 har fått korrekt prefix genom kommando: `Show configuration | display set`, rad 13 visar vilket prefix som delegerats från R1:

```
Client Interface: ge-0/0/0.0
  Hardware Address:          88:e0:f3:2d:a3:80
  State:                     BOUND(DHCPV6_CLIENT_STATE_BOUND)
  ClientType:                STATEFUL
  Lease Expires:             2016-04-27 09:16:08 UTC
  Lease Expires in:         85657 seconds
  Lease Start:               2016-04-26 09:16:08 UTC
  Bind Type:                 IA_PD
  Client DUID:                LL0x3-88:e0:f3:2d:a3:80
  Rapid Commit:              Off
  Server Ip Address:         fe80::206:d6ff:fe1d:8000
  Update Server:             Yes
  Client IP Prefix:          2001:db8:3:300::/56

DHCP options:
  Name: server-identifier , Value: LL0x1-00:06:d6:1d:80:00
  Name: dns-recursive-server , Value: 2001:db8:3::67
  Name: domain-serch-list , Value: testdomain.com
```

Verifiering att CPE3 har fått korrekt prefix genom kommando: `show ipv6 dhcp interface`, rad 17 visar vilket prefix som delegerats från R1:

```
FastEthernet8 is in server mode
Using pool: to-host
Preference value: 0
Hint from client: ignored
Rapid-Commit: disabled
GigabitEthernet0 is in client mode
Prefix State is OPEN (0)
Information refresh timer expires in 23:17:08
Renew will be sent in 3d11h
Address State is IDLE
List of known servers:
  Reachable via address: FE80::206:D6FF:FE1D:8000
  DUID: 000300010006D61D8000
  Preference: 0
Configuration parameters:
  IA PD: IA ID 0x000D0001, T1 302400, T2 483840
  Prefix: 2001:DB8:3:500::/56
  preferred lifetime INFINITY, valid lifetime INFINITY
  DNS server: 2001:DB8:3::67
  Domain name: testdomain.com
  Information refresh time: 0
Prefix name: from-7206
```



```
Prefix Rapid-Commit: disabled  
Address Rapid-Commit: disabled
```

Verifiering att PC har fått korrekt adress (inklusive prefix) från CPE1 med kommandot `ipconfig en0 | grep inet6`:

```
inet6 2001:db8:3:100:129a:ddff:fe6b:6b19 prefixlen 64 autoconf
```

Verifiering att PC har fått korrekt adress (inklusive prefix) från CPE2 med kommandot `ipconfig en0 | grep inet6`:

```
inet6 2001:db8:3:300:129a:ddff:fe6b:6b19 prefixlen 64 autoconf
```

Verifiering att PC har fått korrekt adress (inklusive prefix) från CPE3 med kommandot `ipconfig en0 | grep inet6`:

```
inet6 2001:db8:3:500:129a:ddff:fe6b:6b19 prefixlen 64 autoconf
```

Med de verifiering som blivit genomförda så har vi bevisat att prefixet har delegerats korrekt genom hela testplattformen, från R1, via CPE1-3 ner till end-host, som fått korrekta adress, skapat en egen adress med korrekt prefix och fått viss information från DHCPv6 servern.

4.2 Scenario 2

Tabell 6: Översikt, Resultat scenario 2

Resultat, Översikt Scenario 2		
Förväntat resultat	Utfall	Verifierat genom
Korrekta routes från R1 till CPE1-3	Fungerar	show ipv6 routes
Korrekta routes från CPE1-3 mot R1	Fungerar	show ipv6 routes
Åtkomst från CPE1-3 till loopback0 på R1 (loopback0 får agera ett utomstående nät)	Fungerar	ping till 2001:DB8:666:1 (loopback0 på R1) från PC inkopplat på CPE1-3

Den statiska lösningen har statiska routes från R1 till varje CPE, medan varje CPE enbart har en ::/0 route mot R1. De tänkta end-host i detta scenario skall skapa sin egen adress med SLAAC. Att samtliga routes är korrekta har verifierats med kommandot `show ipv6 routes` på R1, CPE1 och CPE3:

Routes på R1:

```
IPv6 Routing Table - default - 7 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
       B - BGP, R - RIP, I1 - ISIS L1, I2 - ISIS L2
       IA - ISIS interarea, IS - ISIS summary, D - EIGRP, EX - EIGRP external
       O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
C   2001:DB8:1::/64 [0/0]
    via FastEthernet0/0, directly connected
L   2001:DB8:1::1/128 [0/0]
    via FastEthernet0/0, receive
S   2001:DB8:2::/64 [1/0]
    via 2001:DB8:1::2
S   2001:DB8:3::/64 [1/0]
    via 2001:DB8:1::3
C   2001:DB8:666::/64 [0/0]
    via Loopback0, directly connected
L   2001:DB8:666::1/128 [0/0]
    via Loopback0, receive
L   FF00::/8 [0/0]
    via Null0, receive
```

Routes på CPE1:

```
IPv6 Routing Table - 4 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route, M - MIPv6
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
       D - EIGRP, EX - EIGRP external
S   ::/0 [1/0]
    via 2001:DB8:1::1
C   2001:DB8:1::/64 [0/0]
    via ::, GigabitEthernet0/0
L   2001:DB8:1::2/128 [0/0]
    via ::, GigabitEthernet0/0
L   FF00::/8 [0/0]
    via ::, Null0
```

Routes på CPE3:

```
IPv6 Routing Table – default – 4 entries
Codes: C – Connected, L – Local, S – Static, U – Per-user Static route
       B – BGP, HA – Home Agent, MR – Mobile Router, R – RIP
       H – NHRP, D – EIGRP, EX – EIGRP external, ND – ND Default
       NDp – ND Prefix, DCE – Destination, NDr – Redirect, O – OSPF Intra
       OI – OSPF Inter, OE1 – OSPF ext 1, OE2 – OSPF ext 2,
       ON1 – OSPF NSSA ext 1, ON2 – OSPF NSSA ext 2, I – LISP
S   ::/0 [1/0]
    via GigabitEthernet0, directly connected
    via 2001:DB8:1::1
C   2001:DB8:1::/64 [0/0]
    via GigabitEthernet0, directly connected
L   2001:DB8:1::3/128 [0/0]
    via GigabitEthernet0, receive
L   FF00::/8 [0/0]
    via Null0, receive
```

För att kontrollera om denna konfiguration fungerar har vi valt att använda ping från vår PC till loopbacken på R1, som i detta scenario får agera ett nät utanför vårt egna nät, till exempel internet.

Ping från CPE1 till loopback på R1:

```
PING6(56=40+8+8 bytes)2001:db8:2::78e8:51e6:114c:a647 -> 2001:db8:666::1
16 bytes from 2001:db8:666::1, icmp_seq=0 hlim=63 time=0.943 ms
16 bytes from 2001:db8:666::1, icmp_seq=1 hlim=63 time=1.039 ms
16 bytes from 2001:db8:666::1, icmp_seq=2 hlim=63 time=1.012 ms
16 bytes from 2001:db8:666::1, icmp_seq=3 hlim=63 time=1.074 ms
16 bytes from 2001:db8:666::1, icmp_seq=4 hlim=63 time=1.104 ms

— 2001:db8:666::1 ping6 statistics —
5 packets transmitted, 5 packets received, 0.0% packet loss
round-trip min/avg/max/std-dev = 0.943/1.034/1.104/0.055 ms
```

Ping från CPE3 till loopback på R1:

```
PING6(56=40+8+8 bytes)2001:db8:3::685f:341d:2544:18ba -> 2001:db8:666::1
16 bytes from 2001:db8:666::1, icmp_seq=0 hlim=63 time=0.859 ms
16 bytes from 2001:db8:666::1, icmp_seq=1 hlim=63 time=1.051 ms
16 bytes from 2001:db8:666::1, icmp_seq=2 hlim=63 time=1.143 ms
16 bytes from 2001:db8:666::1, icmp_seq=3 hlim=63 time=0.930 ms
16 bytes from 2001:db8:666::1, icmp_seq=4 hlim=63 time=0.937 ms

— 2001:db8:666::1 ping6 statistics —
5 packets transmitted, 5 packets received, 0.0% packet loss
round-trip min/avg/max/std-dev = 0.859/0.984/1.143/0.101 ms
```

Som vi ser i routing tabellerna så är samtliga routes korrekt konfigurerade. Tittar vi närmare på pingarna så ser vi att adressen vår PC fått är 2001:db8:2::78e8:51e6:114c:a647, vilket indikerar att prefixet (2001:db8:2::/64) är rätt och att adressen sedan är skapat med SLAAC. Pingens skickas mot 2001:db8:666::1 som är adressen till R1's loopback. Pingens går fram som den ska och visar då att vår konfiguration har fungerat.

4.3 Resultat av andra lösningar

Vi har genomfört undersökningar om det finns andra lösningar som uppfyller samma funktion som prefix delegation, men har enbart hittat applikationer, t.ex. Dibbler (<http://klub.com.pl/dhcpv6>), som använder redan befintliga funktioner i ipv6.

5 Slutsatser

5.1 För och nackdelar

Tabell 7: För och nackdelar med Prefix delegation

Prefix delegation	
Fördelar	Nackdelar
Bra översikt av nätverket	Mer konfiguration initialt
Automatisk tilldelning av prefix till nya enheter	Inte all utrustning har stöd för PD
Går att tilldela prefix till specifikt interface eller DUID	
Inbyggd funktion i ipv6	
Förändringar genomförs bara från delegerande server	

Tabell 8: För och nackdelar med Statisk routing

Statisk konfiguration	
Fördelar	Nackdelar
Lätt att lägga till nya routes	Förändringar är svåra att genomföra och risken för komplikationer och problem är stor
Lätt att initialt dokumentera	Har man många end-hosts eller end-sites blir det många routes
	Svårt att hålla dokumentation uppdaterad och korrekt
	Svårt att kontrollera utan att påverka andra delar av nätverkets konfiguration
	Svårt att felsöka
	Dålig översikt av nätverket

5.2 Diskussion

Som tidigare nämnts så finns det en svårighet att göra en sådan här jämförelse då infrastrukturen i stora nät oftast inte är så enkel att det går att genomföra i en simulerad testmiljö. Det som i verkligheten kan röra sig om flera lager av routrar och gränsdragningar mellan PE och CPE som kan se olika ut beroende på slutkunder och dess funktion skall i denna testmiljö bestå av 3 lager med utrustning.

Tittar man närmare på för- och nackdelarna för de olika scenariorna så ser vi stora skillnader. Skillnaderna beror primärt i det grundläggande att det ena scenariot är en automatiserad process, vilket generellt sett är en lösning som förenklar och minskar mängden arbete som krävs för att upprätthålla funktionen, medan den statiska processen är mer stabil så länge den inte förändras. Det är här som problemet blir påtagligt, nätverk förändras ofta, särskilt större nätverk.

Prefix delegation ger inte bara möjligheten att dela upp ett nät till mindre, det ger även möjligheten att styra vilket DUID som ska bli tilldelad ett visst prefix. Detta ökar kontrollen över nätverket och underlättar dokumentation, men ger även en ökad nivå av säkerhet. Det går även att använda en funktion där man utgår från det unika ID som varje interface har, men den utrustning vi hade till vårt förfogande kunde inte hantera denna funktion. Med denna funktion kan vi underlätta förändringar i nätverket eftersom prefixet är bundet till ett interface, vilket innebär att samma prefix kommer delas ut oavsett vart i nätet t.ex. en router är, men fortfarande med samma nivå av kontroll och säkerhet.

Arvingen till ipv4 är ipv6, vilket även är en förbättring på många punkter då ett flertal av de protokoll vi använder i ipv4 nu har anpassats till ipv6. Detta gör att vi har en större möjlighet att styra hur vi använder protokollen. I just detta projekt har vi en tydlig användning för just detta när vi använder M- och O-flaggorna för att styra vilken information som vi vill ha från DHCPv6 servern. Även här får vi en större kontroll över vårt nätverk och dess funktioner.

5.3 Slutsats

Även om infrastrukturen i större nätverk alltid kommer kräva statiska routes för att fungera så är det inte hållbart att bygga ett helt nätverk på det viset, även om vi utnyttjar funktioner som stateless DHCPv6 och SLAAC. Skulle det fungera så skulle det behövas vara en mycket begränsad storlek på nätverket och utan förändringar. Fördelarna med statiska konfiguration övervägar inte nackdelarna annat än vid mycket små nätverk och med tanken att ipv6 ska vara grunden för "internet of everything" så kan vi inte räkna med att begränsa oss till små nätverk längre.

Eftersom vi vill ha översikt, kontroll och minimalt med arbete vid förändringar i vårt nätverk så fyller bara alternativet med prefix delegation våra krav.

5.4 Fortsatt forskning

Detta projekt har varit väldigt övergripande, både när det gäller teori och praktisk tillämpning. Så det finns flera delar att titta på för att fortsätta där vi nu slutar. Tanken här är att gå från teori och praktisk tillämpning i liten skala till en mer korrekt bild av verkligheten med funktioner anpassade för verklig drift.

Den första delen är att titta på liknande scenarion, men med ett betydligt större nät, kanske t.o.m. ett nätverk som är i drift. Det skulle ge en möjlighet att se om grunderna vi tittat på är överföringsbara till större nätverk.

Som tidigare nämnts så finns det vissa funktioner som vi inte kunnat titta närmare på då vår testplattform inte kunnat hantera. Att titta på hur dessa skulle kunna förbättra delegeringen av prefix skulle ge oss ännu större möjligheter att anpassa nätet efter våra behov.

Eftersom detta är just en översikt för att förstå funktionerna så skulle det vara bra att ha en implementering av säkerhet, vilket bland annat finns inbyggt i ipv6.

6 Referenser

- [1] R. Desmeules, *Cisco Self-Study: Implementing Cisco IPv6 Networks*. Indianapolis, USA: Cisco Press, 2005.
- [2] IETF, "Rfc 1883," "<https://www.rfc-editor.org/rfc/rfc1883.txt>", [Online; accessed 03-June-2016].
- [3] —, "Rfc 1924," "<https://www.rfc-editor.org/rfc/rfc1924.txt>", [Online; accessed 03-June-2016].
- [4] —, "Rfc 1884," "<https://www.rfc-editor.org/rfc/rfc1884.txt>", [Online; accessed 03-June-2016].
- [5] RIPE, "Ripe publication 655," "<https://www.ripe.net/publications/docs/ripe-655>", [Online; accessed 03-June-2016].
- [6] IETF, "Rfc 1970," "<https://www.rfc-editor.org/rfc/rfc1970.txt>", [Online; accessed 03-June-2016].
- [7] —, "Rfc 1885," "<https://www.rfc-editor.org/rfc/rfc1885.txt>", [Online; accessed 03-June-2016].
- [8] —, "Rfc 1971," "<https://www.rfc-editor.org/rfc/rfc1971.txt>", [Online; accessed 03-June-2016].
- [9] —, "Rfc 3633," "<https://www.rfc-editor.org/rfc/rfc3633.txt>", [Online; accessed 03-June-2016].
- [10] —, "Rfc 3315," "<https://www.rfc-editor.org/rfc/rfc3315.txt>", [Online; accessed 03-June-2016].
- [11] Cisco, "Dhcpv6 based ipv6 access services," "<http://tinyurl.com/nrkqv67>", [Online; accessed 03-June-2016].

7 Bilagor

7.1 Konfigurationsfiler: Scenario 1

CPE1, CPE3 och R1 är reviderade konfigurationsfiler som enbart innehåller de relevanta delarna av konfigurationen.

CPE2 är hela konfigurationsfilen.

R1

Till [Scenario 1](#)

```
version 12.2
!
hostname R1
!
ipv6 dhcp pool to-cpe
 prefix-delegation 2001:DB8:3:100::/56 00030001001DE64EC630
 prefix-delegation 2001:DB8:3:300::/56 00030001CCEF48BB93AC
 prefix-delegation 2001:DB8:3:500::/56 0003000188E0F32DAD80
 prefix-delegation pool cpepool1 lifetime infinite infinite
 dns-server 2001:DB8:3::67
 domain-name testdomain.com
!
!
interface FastEthernet0/0
 no ip address
 no ip route-cache cef
 no ip route-cache
 duplex half
 ipv6 address 2001:DB8:3::1/64
 ipv6 nd other-config-flag
 ipv6 dhcp server to-cpe
!
ipv6 local pool cpepool1 2001:DB8:3::/48 56
```

CPE1

Till [Scenario 1](#)

```
version 12.4
!
hostname CPE1
!
ip domain list testdomain.com
!
ipv6 unicast-routing
ipv6 dhcp pool to-host
 import dns-server
 import domain-name
!
interface GigabitEthernet0/0
 no ip address
 duplex half
 speed 100
 media-type rj45
 ipv6 address autoconfig default
 ipv6 enable
 ipv6 nd other-config-flag
 ipv6 dhcp client pd from-7206
!
interface GigabitEthernet0/1
 no ip address
```



```
duplex auto
speed auto
media-type rj45
ipv6 address from-7206 ::1/64
ipv6 enable
ipv6 nd other-config-flag
ipv6 dhcp server to-host
!
ip forward-protocol nd
```

CPE2

Till Scenario 1

```
version 12.1X46-D45.4;
system {
    root-authentication {
        encrypted-password "/*****"; ## SECRET-DATA
    }
    name-server {
        208.67.222.222;
        208.67.220.220;
    }
    services {
        ssh;
        telnet;
        xnm-clear-text;
        dhcp-local-server {
            dhcpv6 {
                overrides {
                    interface-client-limit 10;
                    process-inform {
                        pool from-isp;
                    }
                }
                group ipv6 {
                    interface ge-0/0/1.0;
                }
            }
        }
        inactive: web-management {
            http {
                interface vlan.0;
            }
            https {
                system-generated-certificate;
                interface vlan.0;
            }
        }
        inactive: dhcp {
            router {
                192.168.1.1;
            }
            pool 192.168.1.0/24 {
                address-range low 192.168.1.2 high 192.168.1.254;
            }
            propagate-settings ge-0/0/0.0;
        }
    }
    syslog {
        archive size 100k files 3;
    }
}
```

```
    user * {
        any emergency;
    }
    file messages {
        any critical;
        authorization info;
    }
    file interactive-commands {
        interactive-commands error;
    }
}
max-configurations-on-flash 5;
max-configuration-rollbacks 5;
license {
    autoupdate {
        url https://ae1.juniper.net/junos/key_retrieval;
    }
}
processes {
    dhcp-service {
        traceoptions {
            file dhcp-trace size 10m;
            level all;
            flag all;
        }
    }
}
}
interfaces {
    ge-0/0/0 {
        unit 0 {
            family inet6 {
                dhcpv6-client {
                    client-type statefull;
                    client-ia-type ia-pd;
                    update-router-advertisement {
                        interface ge-0/0/1.0 {
                            other-stateful-configuration;
                            max-advertisement-interval 20;
                            min-advertisement-interval 10;
                        }
                    }
                }
                client-identifier duid-type duid-ll;
                req-option dns-server;
                update-server;
            }
        }
    }
}
}
ge-0/0/1 {
    unit 0 {
        family inet6;
    }
}
}
}
protocols {
    stp;
}
}
security {
    forwarding-options {
        family {
```

```
        inet6 {
            mode flow-based;
        }
    }
}
screen {
    ids-option untrust-screen {
        icmp {
            ping-death;
        }
        ip {
            source-route-option;
            tear-drop;
        }
        tcp {
            syn-flood {
                alarm-threshold 1024;
                attack-threshold 200;
                source-threshold 1024;
                destination-threshold 2048;
                timeout 20;
            }
            land;
        }
    }
}
nat {
    source {
        rule-set trust-to-untrust {
            from zone trust;
            to zone untrust;
            rule source-nat-rule {
                match {
                    source-address 0.0.0.0/0;
                }
                then {
                    source-nat {
                        interface;
                    }
                }
            }
        }
    }
}
policies {
    from-zone trust to-zone untrust {
        policy trust-to-untrust {
            match {
                source-address any;
                destination-address any;
                application any;
            }
            host-inbound-traffic {
                system-services {
                    all;
                }
                protocols {
                    all;
                }
            }
        }
    }
    interfaces {
```



```
no ip address
duplex auto
speed auto
ipv6 address autoconfig default
ipv6 enable
ipv6 nd other-config-flag
ipv6 dhcp client pd from-7206
!
ip forward-protocol nd
```

7.2 Konfigurationsfiler: Scenario 2

CPE1, CPE3 och R1 är reviderade konfigurationsfiler som enbart innehåller de relevanta delarna av konfigurationen.

R1

Till [Scenario 2](#)

```
Building configuration...
!
version 12.2
!
hostname R1
!
ipv6 address 2001:DB8:666::1/64
!
interface FastEthernet0/0
no ip address
no ip route-cache cef
no ip route-cache
duplex half
ipv6 address 2001:DB8:1::1/64
!
ipv6 route 2001:DB8:2::/64 2001:DB8:1::2
ipv6 route 2001:DB8:3::/64 2001:DB8:1::3
```

CPE1

Till [Scenario 2](#)

```
Building configuration...
!
version 12.4
!
hostname CPE1
!
no ip domain lookup
!
ipv6 unicast-routing
ipv6 dhcp pool KLIENT_POOL
dns-server 2001:DB8:1000::1
domain-name cisco.com
!
interface GigabitEthernet0/0
no ip address
duplex auto
speed auto
media-type rj45
ipv6 address 2001:DB8:1::2/64
ipv6 enable
!
```

```
interface GigabitEthernet0/1
  no ip address
  duplex auto
  speed auto
  media-type rj45
  ipv6 address 2001:DB8:2::1/64
  ipv6 enable
  ipv6 nd other-config-flag
  ipv6 dhcp server KLENT_POOL
!
ip forward-protocol nd
!
ipv6 route ::/0 2001:DB8:1::1
```

CPE3

Till [Scenario 2](#)

```
Building configuration...
!
version 15.2
!
hostname CPE3
!
no ip domain lookup
ipv6 unicast-routing
ipv6 cef
ipv6 dhcp pool KLENT_POOL
  dns-server 2001:DB8:1000::1
  domain-name cisco.com
!
interface FastEthernet8
  no ip address
  duplex auto
  speed auto
  ipv6 address 2001:DB8:3::1/64
  ipv6 enable
  ipv6 nd other-config-flag
  ipv6 dhcp server KLENT_POOL
!
interface GigabitEthernet0
  no ip address
  duplex auto
  speed auto
  ipv6 address 2001:DB8:1::3/64
  ipv6 enable
!
ipv6 route ::/0 GigabitEthernet0
ipv6 route ::/0 2001:DB8:1::1
```