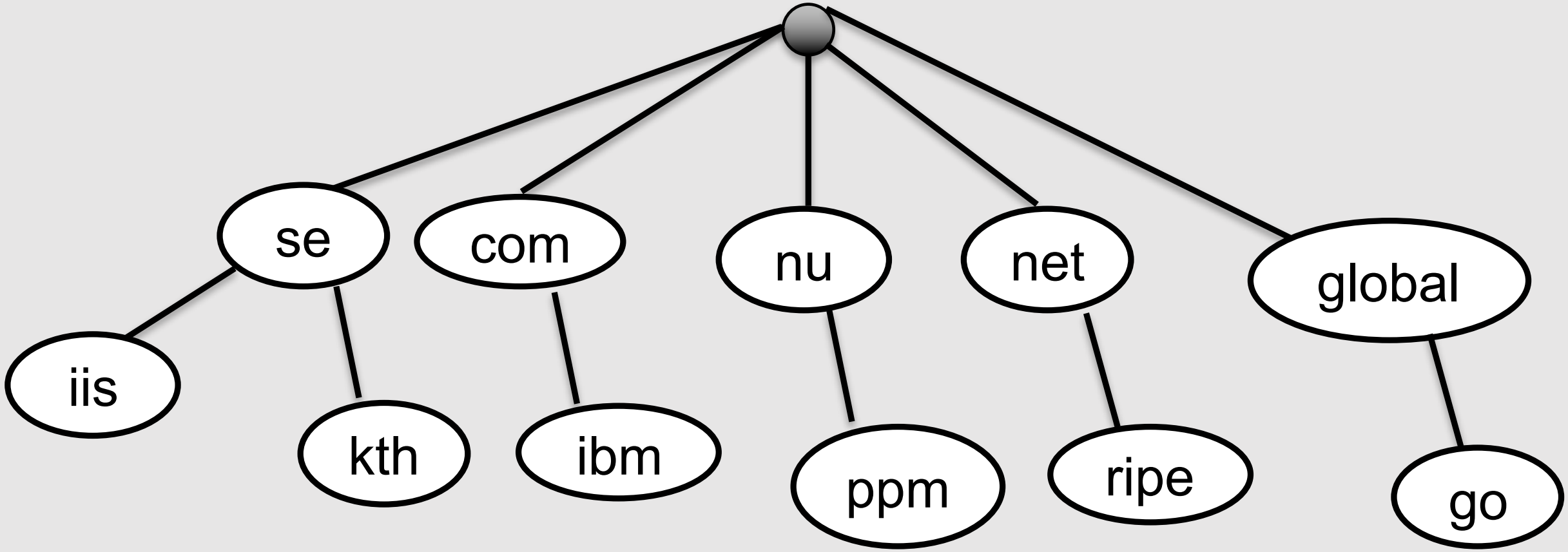

se
com
nu
net
global
iis
kth
ibm
ppm
ripe
go

INTERNET STIFTELSEN

Internets domännamnssystem*

Föreläsning FL04, HT 2023

Mats Dufberg

* Se “Internets domännamnssystem”

Innehåll

▶Zonfil och zonfilsformat

[Till Innehåll]

Zonfilsformat

Ett namnserverprogram (t.ex. Bind) använder en zonfil för att ladda innehållet i en zon. Zonfilen har ett visst format.

Zonfilsformatet är definierat i RFC 1035, vilket ger en standard som kan användas av olika namnserverprogram och som kan förstås av alla.

(RFC 1035 ingår i kurslitteraturen.)

Zonfilsformat

- Ett namnserverprogram kan ha utökningar som inte fungerar i andra namnserverprogram.
- Många namnserverprogram kan lagra zonen i en databas som alternativ till att lagra zonen i en vanlig zonfil.

Här ska vi gå igenom det gemensamma, standardiserade textformatet för zonfiler.

```
$TTL      3600
$ORIGIN namn.se.
@                   SOA       ns1 hostmaster (
                                   2020012104 ; serial
                                   14400      ; refresh
                                   900        ; retry
                                   604800     ; expire
                                   900        ; minimum
                              )
                    NS        ns1
                    NS        dns5.dnshosting.xa.
                    MX        10 mail1
                    MX        20 mail2
www                 A         192.0.2.5      ; www1
                    AAAA      2001:db8:a::5  ; www1
                    A         192.0.2.6      ; www2
                    AAAA      2001:db8:a::6  ; www2
mail1               A         192.0.2.20
                    AAAA      2001:db8:1::20
mail2               A         192.0.2.21
                    AAAA      2001:db8:1::21
ns1            1800 A         192.0.2.30
               1800 AAAA      2001:db8:1::5

; Development servers below
$ORIGIN development.namn.se.
$TTL      600
www                 A         192.0.2.7
                    AAAA      2001:db8:b::7
www.extra           A         192.0.2.8
                    AAAA      2001:db8:b::8
```

Denna exempelzonfil återkommer med kommentarer efter att vi har gått igenom delarna.

▶$TTL

[Till Innehåll]

$TTL

Varje DNS-post måste ha en definierad TTL. I alla utskrifter från ”dig” så finns TTL angivet för alla DNS-poster.

Det enklast och bästa sättet att sätta TLL för alla DNS-poster i en zonfil är att längst upp i zonfilen skriva ”$TTL” med lämpligt värde.

```
$TTL 3600
```

Det gör att alla DNS-poster, där man inte explicit skriver TTL, kommer att få TTL=3600 sekunder. *För en labbzon så kan värdet 300, eller mindre, vara lagom.*

$TTL

```
$TTL 3600
www.namn.se.          A  192.0.2.3
namn.se.              MX 10 mail.namn.se.
```

är samma sak som

```
www.namn.se.      3600  A 192.0.2.3
namn.se.          3600  MX 10 mail.namn.se.
```

TTL blir det som står i den $TTL som ovanför DNS-posten (ovanför i filen, behöver inte vara direkt ovanför), om TTL inte finns med i posten.

$TTL

```
$TTL 3600
www.namn.se.            A 192.0.2.3
namn.se.        1800    MX 10 mail.namn.se.
```

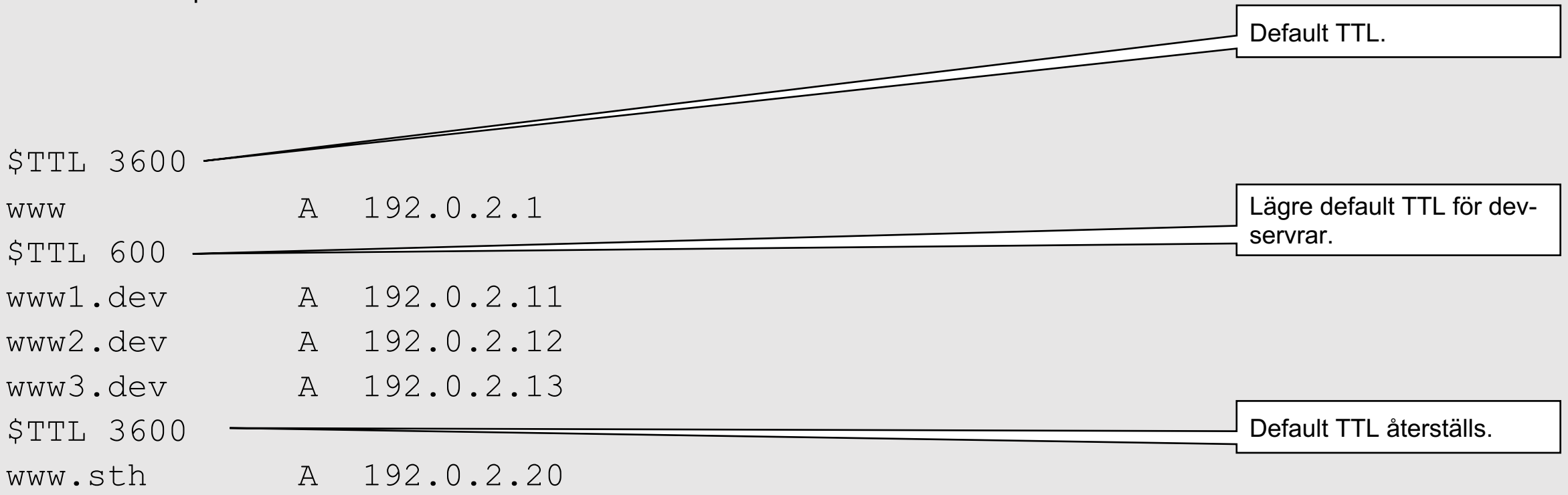
är samma sak som

```
www.namn.se.    3600    A 192.0.2.3
namn.se.        1800    MX 10 mail.namn.se.
```

Explicit värde på posten gäller.

$TTL

$TTL kan upprepas i zonfilen med nytt värde. Det värde man sätter gäller tills nästa $TTL.



Hur bestäms TTL utan $TTL?

Om det inte finns någon $TTL och ingen TTL i posten så går vi till SOA-posten:

```
namn.se.        1800 SOA a.ns.namn.se. hostmaster.namn.se. (
                        2019012804 ; serial
                        14400      ; refresh (4 hours)
                        900        ; retry (15 minutes)
                        604800     ; expire (1 week)
                        86400      ; minimum (1 day)
                )
```

Default TTL om inget annat ges.

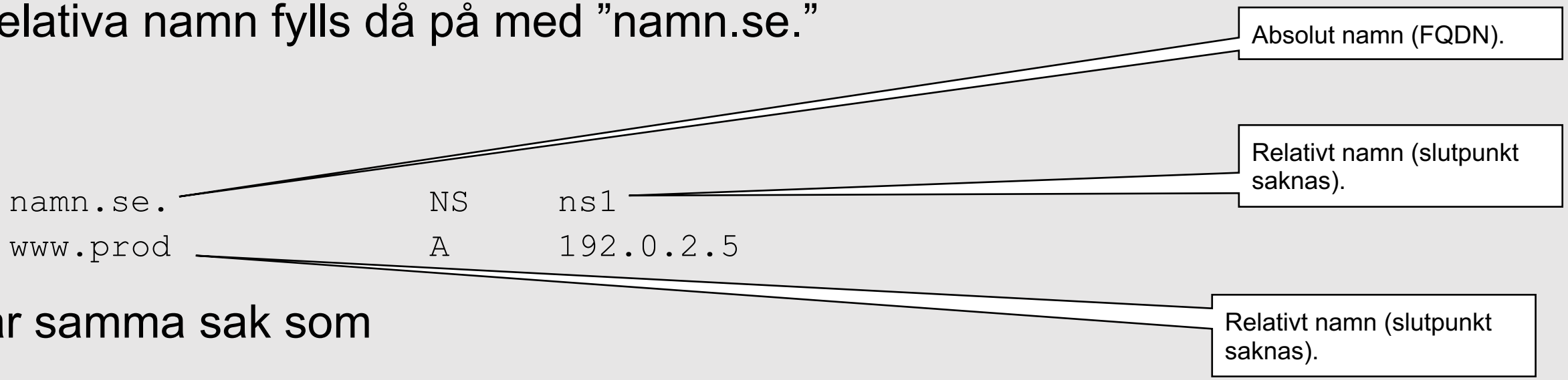
I brist på annan information så används det som finns i SOA-postens RDATA (***minimum***) som även används för negativ cachning. Man bör alltid sätta $TTL först i zonen för att skilja dem åt.

▶$ORIGIN och default domän

[Till Innehåll]

Default domän

När zonfilen laddas så kommer zonens namn att vara default domän. När zonen ”namn.se” laddas så kommer ”namn.se.” att vara default domän. Alla relativa namn fylls då på med ”namn.se.”



är samma sak som

```
namn.se.              NS     ns1.namn.se.
www.prod.namn.se.     A      192.0.2.5
```

$ORIGIN och default domän

Med $ORIGIN så kan vi ställa om default domän. Vi har laddat ”namn.se.” och får ”namn.se.” som default domän.

```
www                         A       192.0.2.5
$ORIGIN  sth.namn.se.
www                         A       192.0.2.51
```

Default domän sätts om och gäller resten av zonfilen eller tills det kommer en ny $ORIGIN.

är samma sak som

```
www.namn.se.                A       192.0.2.5
www.sth.namn.se.            A       192.0.2.51
```

Ibland är det bra, men ska inte överanvändas.

$ORIGIN och apex

Man kan sätta $ORIGIN före alla DNS-poster och ge den zonnamnet:

```
$ORIGIN  namn.se.
@          SOA      ns1   hostmaster (...)
```

Det är inte obligatoriskt, men en bra rutin för att markera vilken zon det är.

$ORIGIN sätter default domän till ”namn.se.” som redan är satt till ”namn.se.” genom att det är den zonen vi har laddat.

$ORIGIN och apex

Om zonfilen ska lagras på annan plats eller kopieras till annan plats, eller innehållet publiceras utanför namnservern, t.ex. som i denna presentation, så blir det entydigt vilken zon det gäller om zonfilen börjar med $ORIGIN och SOA-posten enligt nedan.

```
$ORIGIN  namn.se.
@          SOA      ns1   hostmaster (...)
```

@ och default domän

Istället för att skriva ut ”owner name” så kan man skriva @ ifall ”owner name” ska vara default domän. Normalt används @ för SOA-posten, men den kan användas för fler poster (här ”namn.se”):

```
@       SOA       ns1       hostmaster (...)
@       NS        ns1
@       NS        dns5.dnshosting.xa.
```

@ används oftast bara i ***apex***, men kan användas för andra namn om vi har ställt om default domän med $ORIGIN.

@ och default domän

@ måste stå ensam i domännamnsfältet. Följande är **inte** giltigt

```
www.@   A      192.0.2.5 ; OGILTIGT
```

Istället så kan man ändra default domän.

```
$ORIGIN www.namn.se.
@       A      192.0.2.5 ; GILTIGT
```

Samma owner name för flera poster

Om flera poster i rad har samma ”owner name” så räcker det med att skriva det i första DNS-posten (här ”namn.se”) och sedan göra indrag för följande poster:

```
@       SOA       ns1   hostmaster (...)
        NS        ns1
        NS        dns5.dnshosting.xa.
www     A         192.0.2.5
        AAAA      2001:db8::5
```

Samma owner name för flera poster

Förra bilden är alltså samma sak som

```
namn.se.          SOA       ns1   hostmaster (...)
namn.se.          NS        ns1
namn.se.          NS        dns5.dnshosting.xa.
www.namn.se.      A         192.0.2.5
www.namn.se.      AAAA      2001:db8::5
```

Men det kan vara lättare att läsa om man utnyttjar att flera poster har samma ”owner name”, men båda är rätt.

Relativa och absoluta namn

Vi antar att default domän och zon är "namn.se.".

```
mail.namn.se        A        192.0.2.20
www.                A        192.0.2.30
```

Ser ut som ett *absolut* namn men är *relativt* – ingen slutpunkt.

Ser ut som ett *relativt* namn men är *absolut* – har slutpunkt.

Första posten är samma sak som följande, vilket vi troligen inte vill ha:

```
mail.namn.se.namn.se.        A   192.0.2.20
```

Default domän läggs alltid på relativa namn.

Andra posten är "www.", d.v.s. direkt under root, och går inte att ladda – "out of zone data". I zonen "namn.se" så kan vi inte ha ***owner name*** som inte är i eller under den noden.

▶DNS-poster och kommentarer i zonfilen

[Till Innehåll]

Långa DNS-poster på flera rader

Långa DNS-poster kan vi skriva på flera rader med hjälp av parenteser ”( )”. Vi bör normalt använda tekniken för SOA-posten och behålla samma kommentarer som ”dig” ger:

```
@    SOA ns1 hostmaster (
                          2019012804 ; serial
                          14400      ; refresh
                          900        ; retry
                          604800     ; expire
                          900        ; minimum
                          )
```

Praktiskt med dessa kommentarer på SOA-posten. Kan plockas från "dig kth.se SOA +mult"

Klass behövs aldrig

Det finns aldrig anledning till att skriva ”IN” i DNS-posterna eftersom klassen alltid är internetklassen.

Man kan skriva

```
www     IN    A       192.0.2.5
        IN    AAAA    2001:db8::5
```

men det är bättre att skriva

```
www     A       192.0.2.5
        AAAA    2001:db8::5
```

Alltid domän efter NS och MX

Man kan aldrig ange IP-adressen direkt efter NS och MX. Följande betyder förmodligen inte det man tror:

```
$ORIGIN   namn.se.
@         SOA         ns1 hostmaster (...)
          NS          192.0.2.5
          MX          10      192.0.2.10
```

Relativa domännamn även om det ser ut som IP-adresser. Ingen punkt på slutet.

utan det korrekta är förmodligen

```
$ORIGIN   namn.se.
@         SOA         ns1 hostmaster (...)
          NS          ns1
          MX          10      mail
ns1       A           192.0.2.5
mail      A           192.0.2.10
```

Alltid domän efter NS och MX

Om vi expanderar det felaktiga i förra bilden med default domän så får vi:

```
namn.se.  SOA     ns1.namn.se.   hostmaster.namn.se. (...)
          NS      192.0.2.5.namn.se.
          MX      10      192.0.2.10.namn.se.
```

Förmodligen felaktiga namn.

vilket troligen inte är det vi vill ha.

Alltid domän efter NS och MX

Om det felaktiga istället är:

```
$ORIGIN  namn.se.
@       SOA     ns1  hostmaster (...)
        NS      192.0.2.5.
        MX      10   192.0.2.10.
```

Avslutande punkt ger absolut domän (FQDN).

så betyder det att vi har något under toppdomänen ”.5” resp. ”.10”, och de finns inte.

CNAME kan inte kombineras med annat

Följande kommer inte att gå eftersom CNAME måste vara ensam i noden (för domännamnet):

```
www     A     192.0.2.5
www     CNAME www.webhosting.xa.
```

men följande är helt OK (olika noder/namn):

```
web     A     192.0.2.5
www     CNAME www.webhosting.xa.
```

Kommentarer i zonfilen

Det går bra att skriva in kommentarer i zonfilen. Både på egen rad och på samma rad som posten. Kommentar börjar med semikolon (”;”) och varar raden ut:

```
@       SOA     ns1     hostmaster (...)
        NS      ns1                     ; master server
        NS      dns5.dnshosting.xa.     ; slave server
; Web servers:
www     A       192.0.2.5               ; IPv4
        AAAA    2001:db8::5             ; IPv6
```

Kommentarer i zonfilen

Om man skriver kolon (”:”) eller brädgård (”#”) så blir det inte vad man tror. Följande har fel tecken:

```
@       SOA     ns1   hostmaster (...)
        NS      ns1                       : master server
        NS      dns5.dnshosting.xa.       ; slave server
: Old web server not used.
#www    A       192.0.2.5                 ; IPv4
#       AAAA    2001:db8::5               ; IPv6
; New web server outsourced
www     CNAME   web.webhosting.xa.        ; Has both v4 and v6
```

▶Kommenterad zonfil

[Till Innehåll]

```
$TTL      3600
$ORIGIN namn.se.
@                 SOA  ns1 hostmaster (
                              2020012104 ; serial
                              14400      ; refresh
                              900        ; retry
                              604800     ; expire
                              900        ; minimum
                             )
                  NS         ns1
                  NS         dns5.dnshosting.xa.
                  MX         10 mail1
                  MX         20 mail2
www               A          192.0.2.5       ; www1
                  AAAA       2001:db8:a::5   ; www1
                  A          192.0.2.6       ; www2
                  AAAA       2001:db8:a::6   ; www2
mail1             A          192.0.2.20
                  AAAA       2001:db8:1::20
mail2             A          192.0.2.21
                  AAAA       2001:db8:1::21
ns1          1800 A          192.0.2.30
             1800 AAAA       2001:db8:1::5

; Development servers below
$ORIGIN development.namn.se.
$TTL      600
www               A          192.0.2.7
                  AAAA       2001:db8:b::7
www.extra         A          192.0.2.8
                  AAAA       2001:db8:b::8
```

TTL för alla följande poster i filen, om inte specifik TTL är angiven.

Sätter default domän för relativa namn. På denna plats i zonfilen normalt zonens namn.

Praktiskt med dessa kommentarer på SOA-posten. Kan plockas från "dig kth.se SOA +mult"

Ersätts med default domän.

Efter ";" är kommentar.

Tom ***owner name*** betyder att den är samma som föregående post.

Relativt namn. Kompletteras med default domän

$ORIGIN och $TTL sätts till nya värden. Påverkar följande poster.

Använd relativa ***owner name.*** Gör alltid indrag i SOA-postens RDATA. Sätt posttyp (A, AAAA etc) över varandra så att det blir lätt att läsa zonfilen.

▶Master och slav

[Till Innehåll]

Master och slav

En hostingserver – där zonen hostas – är master eller slav.
Distinktionen gäller bara hostingservrar, inte resolvrar.

Master (eller primär, ”primary server”):

- Auktoritativ namnserver där zonfilen skapas/uppdateras.
- Auktoritativ namnserver som slavserver hämtar zonfilen från.

Slav (eller sekundär, ”seconday server”):

- Auktoritativ namnserver som hämtar zonfilen med zonöverföring (”zone transfer”, AXFR) från en master.

Master och slav

Master (primär) och slav (sekundär) är roller för en zon.

- En server kan vara master för en zon och slav för en annan.

Master och slav är roller mellan servrar.

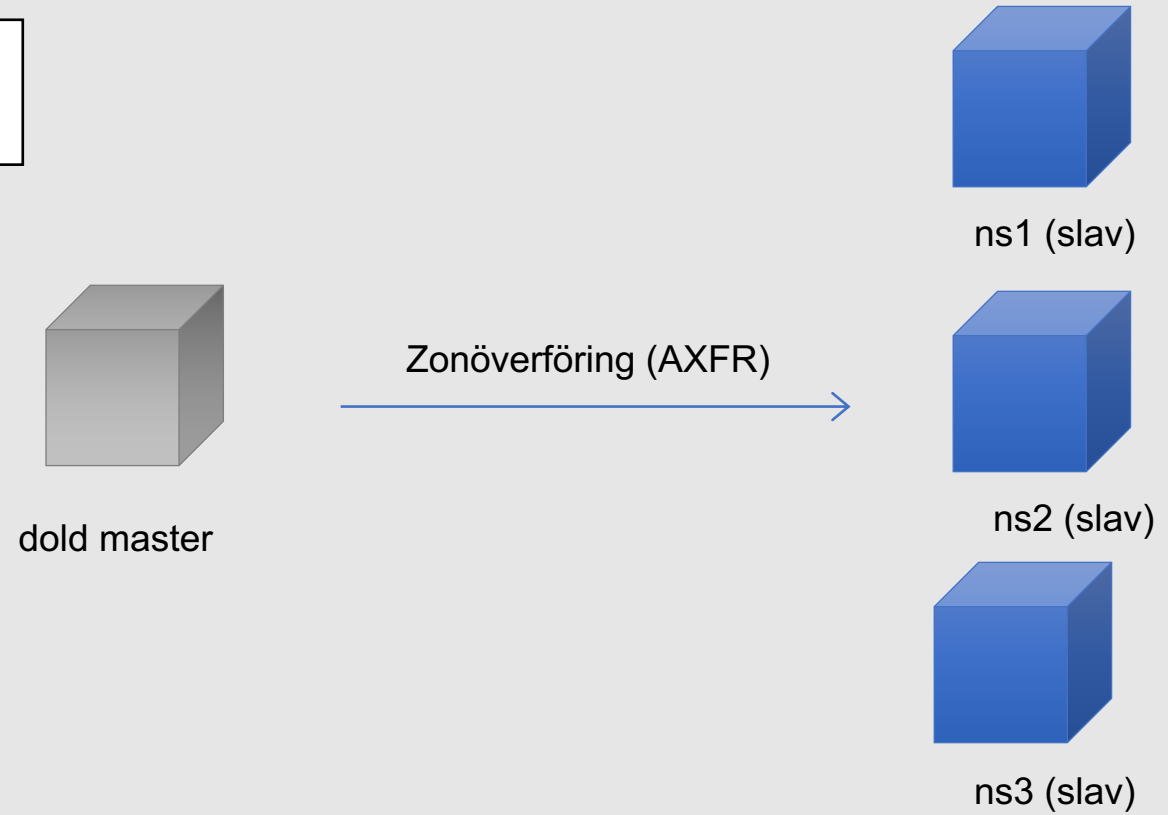
- *Server2* kan vara slav mot *server1* och master mot *server3*.

Master och slav (alla listas som NS)

Klassisk konfiguration med en master och en eller flera slavar där alla finns med som NS för zonen.

Master och slav med dold master

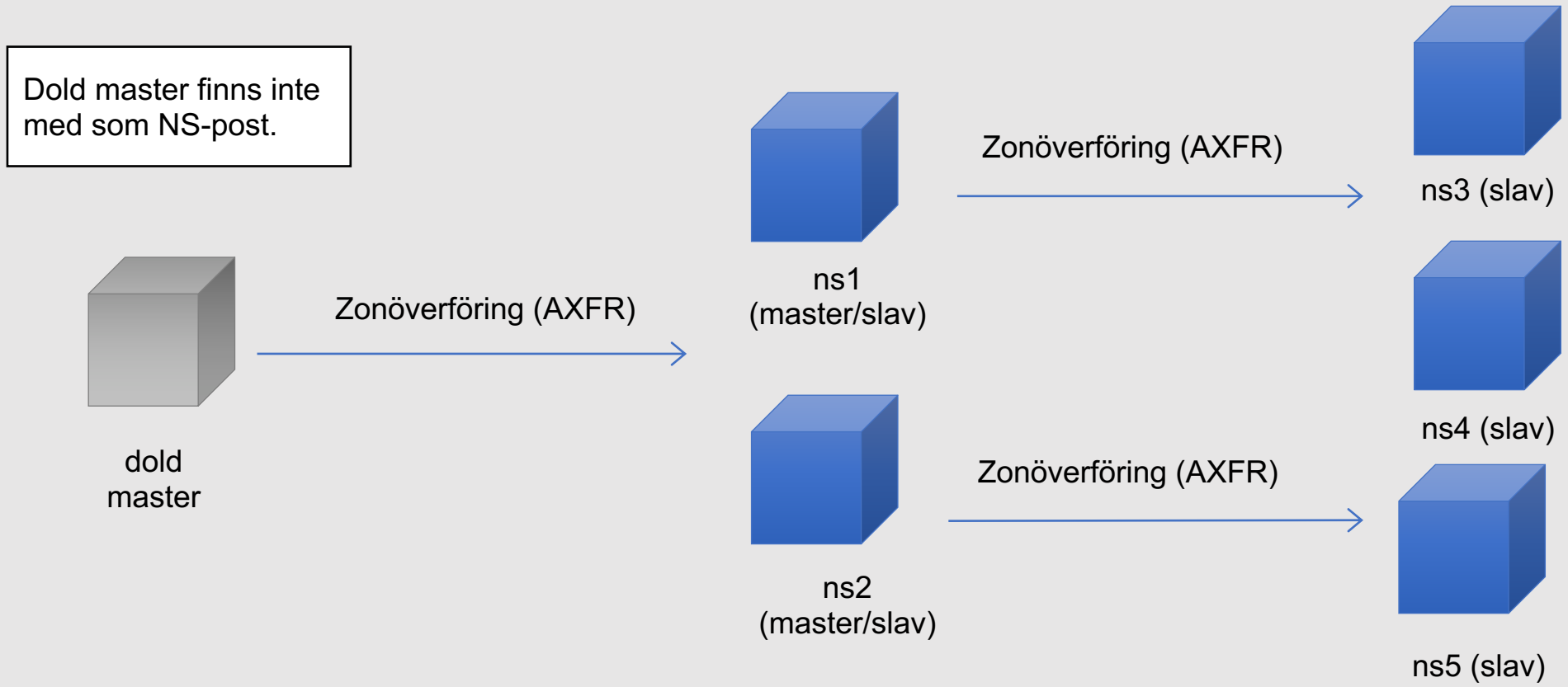
Dold master finns inte med som NS-post.



Smart lösning också för små zoner eftersom alla NS i zonen uppdateras på samma sätt. Mastern kan vara relitivt svag eftersom den inte får frågor från Internet.

Konfiguration med en dold master, d.v.s. endast slavarna finns med som NS för zonen. Väldigt vanligt idag för större zoner.

Dubbla mastrar

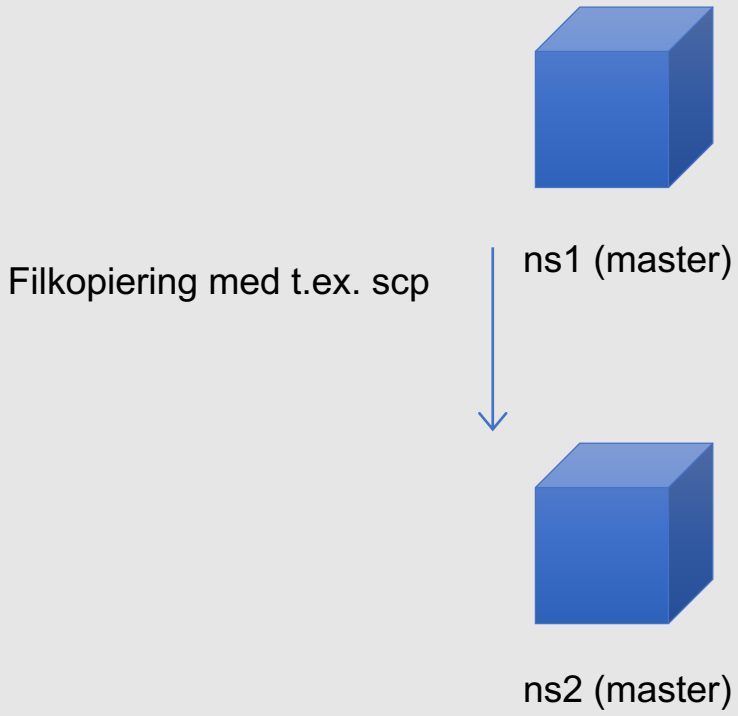


Konfiguration med en dold master, och två slavar som är mastrar för resten (dubbla mastrar).

Bara masterservrar

Det går att ha två (eller fler) masterservrar som alla är som NS. Hur ska man hålla zonfilen identisk på båda?

Bara masterservrar



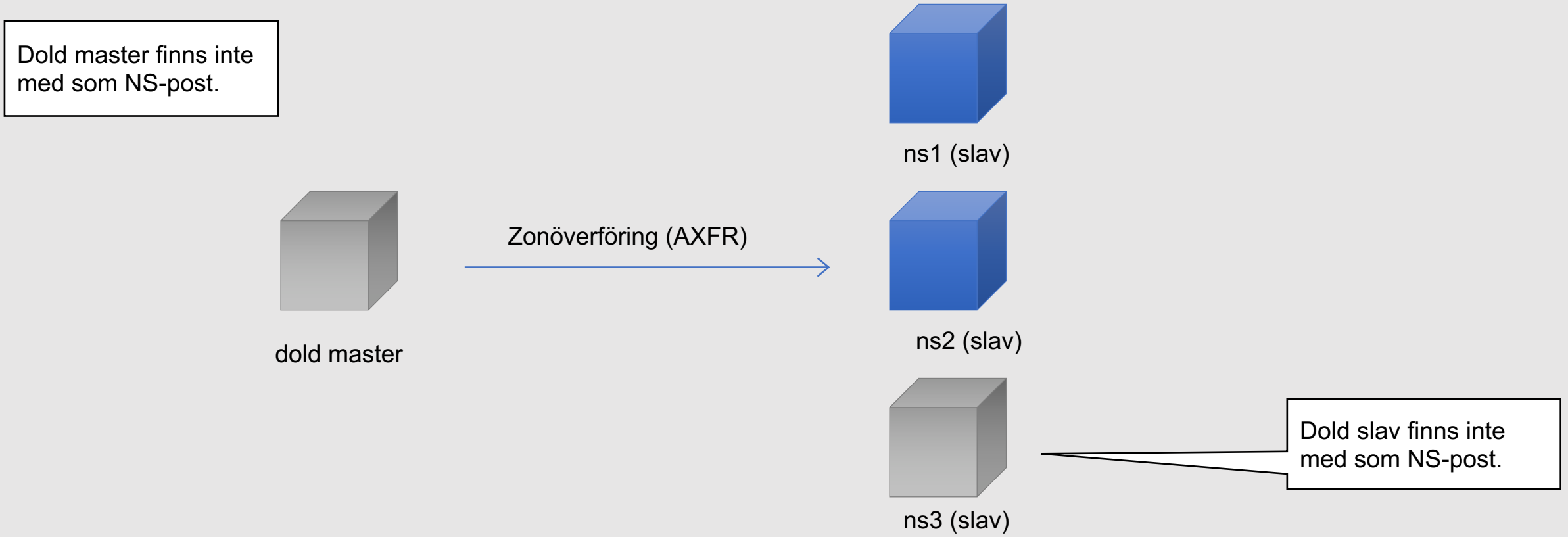
Det går att redigera filen på ena servern och kopiera till den andra.

DNS-tekniskt två mastrar, funktionellt en master och en slav.

Bara masterservrar

Zonfilen skapas från databasen och läggs på masterservrarna. DNS-tekniskt två mastrar, funktionellt två slavar med databasen som (dold) master.

Dold slav



Om zonöverföring tillåts så går det att sätta upp en dold slav. Om AXFR är öppen så kan vem som helst sätta upp en dold slav.

Dold master eller slav

Dold master eller dold slav får inga DNS-frågor via den vanliga delegeringsvägen. Resolvrar hittar namnservrarna genom NS-posterna, och finns det ingen NS-post så blir det inga frågor.

En dold master får fortfarande DNS-frågor från sina slavar i samband med zonöverföring. Normalt är avsikten att den dolda mastern inte ska få vanliga DNS-frågor.

En dold slav har annat syfte och kan få frågor via någon annan väg, t.ex. hårdkodat i en resolver. Eller kanske för att hålla koll på zonen.

Vilken är masterservern?

```
;; ANSWER SECTION:
kth.se.                 1800 IN   SOA a.ns.kth.se. hostmaster.kth.se. (
                        2019012045 ; serial
                        14400      ; refresh (4 hours)
                        900        ; retry (15 minutes)
                        604800     ; expire (1 week)
                        86400      ; minimum (1 day)
                        )

;; AUTHORITY SECTION:
kth.se.                 1800 IN   NS a.ns.kth.se.
kth.se.                 1800 IN   NS nic2.lth.se.
kth.se.                 1800 IN   NS ns2.chalmers.se.
kth.se.                 1800 IN   NS b.ns.kth.se.
```

Vi kan gissa, men vi kan inte veta. Master och slav har bara betydelse för uppdateringen av zonen. Alla är lika auktoritativa för zondatat.

Servern som listas i SOA MNAME? – Kanske, men så behöver det inte vara. Mastern kan vara helt dold.

Vilken är masterservern?

Den som kontrollerar namnservrarna vet vilken som är masterservern.

På masterservern så konfigureras zonen som ”master” och där finns en redigerbar zonfil.

Om zonfilen på masterservern uppdateras från en databas eller genom filkopiering så är det ändå, DNS-tekniskt sett, en masterserver.

Vilken är slavservern?

Den som kontrollerar namnservrarna vet vilken som är slavservern.

På slavservern så konfigureras zonen som ”slave” och med ett ”masters statement”. Slaven måste veta IP-adressen till mastern. Eller IP-adresserna till mastrarna.

Zonen lagras i en zonfil som namnservern skapar när den hämtar zonen med zonöverföring. Den zonfilen kan inte uppdateras manuellt, utan bara genom zonöverföringar.

Hur hittar man slavarna?

Normal så konfigurerar man inte mastern med vilka servrar som är slavservrar.

Ibland konfigurerar man mastern med vilka IP-adresser som får hämta zonen, och det är ungefär lika med slavarna, men om man konfigurerar med TSIG-nyckel istället för IP-adress så blir kopplingen oklar. (Mer om TSIG-nyckel i kommand bild.)

Hur hittar man slavarna?

När mastern vill ”veta”, så tar den alla NS-poster i zonen och drar bort sig själv. Kvar är de som mastern antar är slavar. Det är till dem mastern skickar NOTIFY om att zonen har uppdaterats. (Mer om NOTIFY i kommande bild.)

Ev. dold slavserver måste man hålla reda på. Ev. så kan det finnas konfiguration att NOTIFY ska skickas till ytterligare servrar.

▶Konfigurering av master och slav

[Till Innehåll]

Konfigurering av masterserver

I Bind så lägger vi in följande konfiguration för varje zon som servern är masterserver för:

```
# Master for namn.se
zone ”namn.se" {
  type master;
  file ”namn.se";
};
```

Man kan välja annat namn, bara det är unikt i katalogen. Utan sökväg så ska den finnas i default katalog.

Det finns fler inställningar som man kan göra, men ovanstående räcker. Vilken eller vilka servrar som är slav specificeras inte här.

Zonfilen på slavservern

Zonfilen på slavservern är innehållsmässigt identisk med zonfilen på masterservern (om zonöverföringen fungerar), men filmässigt annorlunda.

Per default så sparar Bind zonfilen på slavservern i ett binärt format, ”raw”. Behåll ”raw” på slavzoner för att få snabb laddning.

Sätt ”masterfile-format text” för att få ett läsbart format, t.ex. i samband med utbildning, tester, utveckling eller felsökning. Det går också att konvertera från ”raw” till ”text”.

Konfigurering av slavserver

I bind så lägger vi in följande konfiguration för varje zon som servern är slavserver för:

```
# Slave for namn.se
zone ”namn.se" {
  type slave;
  file ”namn.se";
  masterfile-format text;
  masters { 192.0.2.10;};
};
```

Man kan välja annat namn, bara det är unikt i katalogen. Utan sökväg så ska den finnas i default katalog.

Textformat, inte binärformat, för kursen. Annars behåller man normalt "raw".

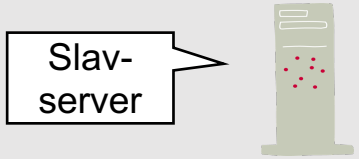
Man kan ha flera mastrar, och blandat IPv4 och IPv6.

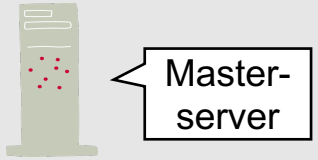
Det finns fler inställningar som man kan göra, men ovanstående räcker.

▶Zonsynkronisering och SOA

[Till Innehåll]

Synkronisering av master och slav





2 Fråga efter SOA för zonen (UDP).

3 Svar med SOA för zonen.

1 Zonen har nått ***refresh time*** på servern, d.v.s. tid att kontrollera.

4 Jämför serienumret i SOA med serienumret i laddad slavzon.

6 Begär hämtning av zonen med AXFR eller IXFR (TCP).

7 Ev. verifiera att slaven får hämta zonen, annars skicka REFUSED.

5 Om serienumret i mastern inte är högre så stoppar vi här (vi gör inget om det är lägre eller lika).

8 Alt 1: Skicka zonen.

Alt 2: Skicka REFUSED (slaven är inte godkänd).

9 Ladda den nya versionen av zonen på slavservern.

SOA-postens roll för zonsynkronisering

```
;; ANSWER SECTION:
kth.se.                 1800 IN   SOA a.ns.kth.se. hostmaster.kth.se. (
                        2019012045 ; serial
                        14400      ; refresh (4 hours)
                        900        ; retry (15 minutes)
                        604800     ; expire (1 week)
                        86400      ; minimum (1 day)
                        )
```

- **SERIAL** – Används för att avgöra om zonsynkronisering behöver göras. Om mastern har högre så behöver vi göra det.
- **REFRESH** – Så ofta ska slaven kontrollera mot mastern.
- **RETRY** – Om REFRESH misslyckades (kunde inte hämta SOA eller kunde inte göra zonöverföring) så ska vi vänta så länge innan vi försöker igen.
- **EXPIRE** – Om REFRESH misslyckas gång på gång så ger vi upp med att ”visa” zonen efter så lång tid (svarar med SERVFAIL istället).

▶Serienumret i SOA

[Till Innehåll]

Serienumret

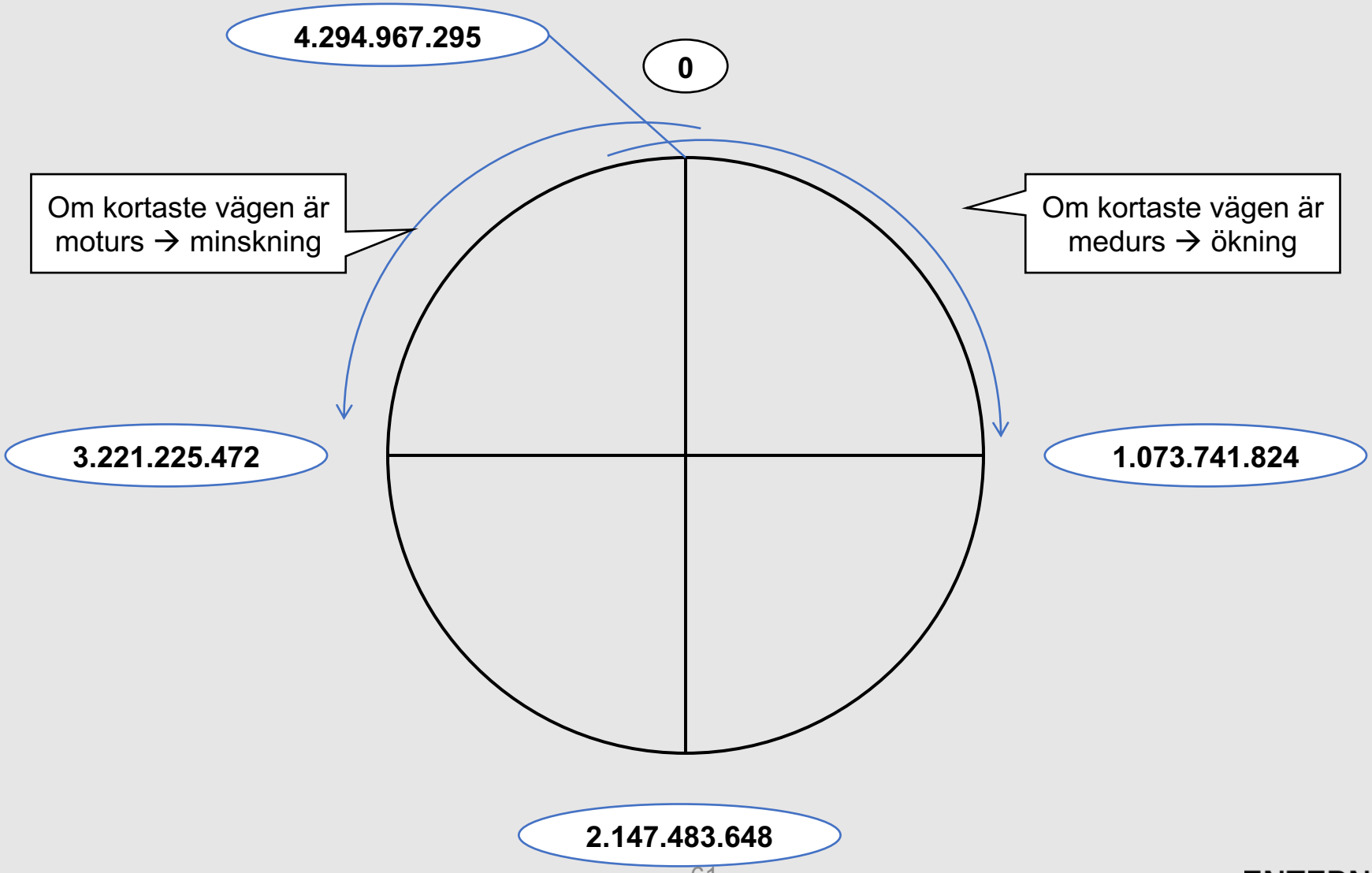
Serienumret är ett 32-bitars heltal, 0 till 4.294.967.295.

Serienumret jämförs som en klocka så att om kortaste sträckan är medurs så är det en ökning. Om det är moturs så är det en minskning. Om två tal är precis mitt emot varandra så är det odefinierat.

Serienummer

Serienummer	”Klockan”
0	12
1.073.741.824	3
2.147.483.648	6
3.221.225.472	9
4.294.967.295	precis före 12

Serienummer



Ändring av serienumret

Exempel 1:

Befintligt: 100
Nytt: 101
→ ökning (rätt)

Exempel 2:

Befintligt: 2021012901
Nytt: 2021012903
→ ökning (rätt)

Exempel 3:

Befintligt: 1000
Nytt: 3300200100
→ minskning (fel)

Exempel 4:

Befintligt: 4100200100
Nytt: 1
→ ökning (rätt)

Serienumret har ändrats åt fel håll

Om serienumret minskar istället för att öka så blir det ingen zonöverföring och slaven kan släpa efter.

T.ex.: man använder datum som serienummer, men byter sedan till att börja på 1. Man glömmer att nollställa slaven.

Man kan tvinga slaven att hämta ny zon genom att stoppa slaven, ta bort zonfilen på slaven och starta slaven igen.

▶AXFR och IXFR

[Till Innehåll]

Begära zonen – AXFR

AXFR är en frågetyp (men inte posttyp). Den kan alltså stå i frågan, men det finns ingen sådan post.

```
$ dig @a.ns.kth.se kth.se axfr

; <<>> DiG 9.10.6 <<>> @a.ns.kth.se kth.se axfr
; (1 server found)
;; global options: +cmd
; Transfer failed.
```

Det gick inte så bra. Hur ser det ut när det går bra?

AXFR – Query

```
; <<>> DiG 9.11.5-P1 <<>> @localhost pearldragon.se axfr +qr +mult +comments +question
; (1 server found)
;; global options: +cmd
;; Sending:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 57972
;; flags: ad; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 1b9d1912030b5caf
;; QUESTION SECTION:
;pearldragon.se.        IN AXFR

;; QUERY SIZE: 55

(forts)
```

Flaggorna behövs för att ”dig” ska visa full information när man väljer AXFR.

OPT-posten i ***query*** läggs i ***additional section***.

AXFR – Response 1 (2)

```
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 57972
;; flags: qr aa ra; QUERY: 1, ANSWER: 6, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1280
; COOKIE: 1b9d1912030b5caf093de27f5c48e1f06ba35e5caf45c84e (good)
;; QUESTION SECTION:
;pearldragon.se.        IN AXFR

;; ANSWER SECTION:
pearldragon.se.        43200 IN SOA ns.narnia.pp.se. postmaster.narnia.pp.se. (
                2016083000 ; serial
                28800      ; refresh (8 hours)
                7200       ; retry (2 hours)
                1209600    ; expire (2 weeks)
                43200      ; minimum (12 hours)
                )
(forts)
```

OPT-posten i ***response*** läggs i ***additional section***.

Zonens samtliga DNS-poster läggs i ***answer section***.

AXFR – Response 2 (2)

```
pearldragon.se.         43200 IN NS ns.narnia.pp.se.
pearldragon.se.         43200 IN NS kif.hildingsson.se.
pearldragon.se.         43200 IN NS lakridstrollet.regexp.se.
pearldragon.se.         43200 IN MX 10 mx.narnia.pp.se.
pearldragon.se.         43200 IN SOA ns.narnia.pp.se. postmaster.narnia.pp.se. (
                2016083000 ; serial
                28800      ; refresh (8 hours)
                7200       ; retry (2 hours)
                1209600    ; expire (2 weeks)
                43200      ; minimum (12 hours)
                )

;; Query time: 0 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: ons jan. 23 22:51:44 CET 2019
;; XFR size: 6 records (messages 1, bytes 266)
```

SOA-posten kommer två gånger, som först post (föregående bild) och sista post (denna bild) i zonöverföringen. Det var 6 poster enligt ***answer count*** (på förra bilden). SOA räknas som två. Totalt 5 DNS-poster i denna zon (zonfil).

Liten zon som skulle kunna rymmas i ett UDP-paket, men AXFR är alltid över TCP.

Begära zonen – AXFR

Försöka hämta där vi inte får.

Flaggorna behövs för att ”dig” ska visa full information när man väljer AXFR.

```
; <<>> DiG 9.10.6 <<>> @a.ns.kth.se kth.se axfr +comment +question
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: REFUSED, id: 42093
;; flags: qr; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;kth.se.                IN  AXFR

; Transfer failed.
```

Status blir REFUSED. I detta fall så är det policy som förbjuder AXFR att fullföljas.

AXFR hämtar hela zonen

Med AXFR så hämtas hela zonen varje gång. Det är inget problem om zonen är liten (som i vårt exempel).

Stora zoner ger stora zonfiler, t.ex. se-zonen mer än 1 GB på disk. Det finns betydligt större zoner.

Om en stor zonfil ändras ofta och man vill publicera ändringarna snabbt så blir det problem.

IXFR

IXFR (”Incremental Zone Transfer”)

Istället för att skicka hela zonen så skickas en ”diff” mellan den version som slaven har och den version som är den aktuella hos mastern.

IXFR kräver att zonfilen uppdateras så att Bind kan hålla redan på förändringarna. Med uppdateringar med en editor så fungerar det dåligt.

Vi går inte in på detaljerna.

▶NOTIFY

[Till Innehåll]

Notifiera uppdateringar – NOTIFY

Slaven ansvarar för att hålla sig uppdaterad, men mastern kan informera när zonen har ändrats.

Mastern skickar då ett speciellt NOTIFY-meddelande. Det är ett DNS-paket, men med speciella värden.

Notifiera uppdateringar – NOTIFY

1. Zonfilen har ändrats och masterservern har laddat den nya versionen.
2. Masterservern tar alla NS-poster och får fram alla (potentiella) slavar.
3. Bind kan konfigurera ytterligare slavar att skicka NOTIFY till.
4. Masterservern skickar NOTIFY till alla slavar.
5. Enligt standarden så ska slaven svara.

Slavar och NOTIFY

1. Slaven får ett NOTIFY.
2. Slaven kontrollerar att NOTIFY kom från en master (enligt dess lista över mastrar).
3. Enligt standarden så ska den svara.
4. Om NOTIFY kom från rätt master så kommer slaven att agera som om REFRESH har löpt ut, d.v.s. det är tid att kontrollera serienumret på mastern.

Slavar och NOTIFY

NOTIFY innebär alltså inte att zonöverföring begärs (eller startas) utan att slaven startar kontrollen om begäran om zonöverföring är nödvändig.

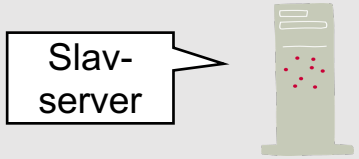
Konfigurera NOTIFY

I Bind är NOTIFY på per default. Alla slavar enligt zonen kommer att få NOTIFY. Man kan lägga till t.ex. dolda slavar (slavar som inte är listade i zonen) med ”also-notify” på masterservern.

```
# Master for namn.se
zone ”namn.se" {
  type master;
  file ”namn.se";
  also-notify { 192.0.2.200; };
};
```

Det går också att låta en slav skicka NOTIFY till dolda slavar.

Synkronisering med NOTIFY



1. Zonen har laddats om – skapa lista över slavar som ska få NOTIFY

2. Skicka NOTIFY för zonen (UDP) till slavarna.

3. Kommer NOTIFY från en av våra mastrar? Om inte, bryt.

4. Fråga efter SOA för zonen (UDP).

5. Svar med SOA för zonen.

6. Jämför serienumret i SOA med serienumret i laddad slavzon.

7. Om serienumret i mastern *inte* är högre så stoppar vi här.

8. Begär hämtning av zonen med AXFR eller IXFR (TCP).

9. Ev. verifiera att slaven får hämta zonen, annars skicka REFUSED.

10. Alt 1: Skicka zonen.
Alt 2: Skicka REFUSED.

11. Ladda den nya versionen av zonen på slavservern.

▶Begränsa zonåtkomst, ACL och TSIG

[Till Innehåll]

Begränsa zonåtkomst

De flesta zonägare vill begränsa vem som kan göra zonöverföring. Det finns i Bind två sätt att kontrollera om klienten är behörig:

1. Endast tillåta klienter på vissa IP-adresser att göra zonöverföring.
2. Använda kryptonyckel enligt TSIG med en delad hemlighet mellan master och slav.

Begränsa zonåtkomst med IP-adress

Enkel konfigurering som bara behöver göras i masterservern.

I slavservern kan zonöverföring stängas av eller bara tillåtas av ”localhost”.

Begränsa zonåtkomst med IP-adress (master)

Begränsa zonöverföring till godkänd eller godkända slavar. Inkludera även ”localhost” plus ev. andra datorer som ska få göra zonöverföring med ”dig” så att det går att testa vid behov.

```
# Master for namn.se
zone ”namn.se" {
  type master;
  file ”namn.se";
  allow-transfer { localhost; 192.0.2.200; 2001:db8:1::200; };
};
```

I bind så är ”localhost” samma sak som 127.0.0.1 plus ::1.

Det går också att lägga in generell konfiguration för alla zoner under ”options” i named.conf.

Begränsa zonåtkomst med IP-adress (slav)

Begränsa zonöverföring till ”localhost” plus ev. andra datorer som ska få göra zonöverföring med ”dig” så att det går att testa vid behov.

```
# Slave for namn.se
zone ”namn.se" {
  type slave;
  file ”namn.se";
  masterfile-format text;
  masters { 192.0.2.10;};
  allow-transfer { localhost; };
};
```

Det går också att lägga in generell konfiguration för alla zoner under ”options” i named.conf.

Begränsa zonåtkomst med IP-adress

En nackdel är att det passar dåligt för mera dynamiska miljöer och att det kräver uppdatering i mastern om slavservern byter IP-adress.

Begränsa zonåtkomst med TSIG

TSIG är en kryptonyckel som kan användas för att signera DNS-kummunikationen (”query” eller ”response”) mellan t.ex. master och slav.

Både master och slav måste konfigureras med en hemlig nyckel initialt. När nyckeln är inlagd så behöver inget göras innan nyckeln ev. måste bytas.

Slaven kan fortsätta att använda samma nyckel efter ev. IP-adressbyte.

Begränsa zonåtkomst med TSIG

Förutom att TSIG kan användas för att begränsa accessen till zonöverföring så ger TSIG ytterligare fördel:

- När en slav hämtar zonen med TSIG så kan den säkerställa att zonfilen inte har förvanskats på vägen genom att svaren (***responses***) är signerade med TSIG-nyckeln.

Begränsa zonåtkomst med TSIG

Begränsning:

- TSIG ger inte något avlyssningsskydd. Om tredjepart kan lyssna på förbindelsen så kan den ta del av den överförda zonen.

Oftast är detta inget allvarligt problem.

TSIG för alla typer av DNS-frågor

Det är inte bara AXFR/IXFR som kan kontrolleras med TSIG. Det gäller alla typer av DNS-frågor.

- En resolver kan tillåta DNS-frågor från klienter med rätt TSIG-nyckel. Fördelen är att klienten då får ett signerat svar (***response***).
- TSIG ger en identitet eller grupptillhörighet åt klienten, och servern kan därför ge olika svar beroende på klienten baserat på TSIG-nyckeln
- För avlyssningsskydd krävs andra mekanismer (kommer senare).

Skapa TSIG-nyckel

TSIG-nyckel skapas med kommandot ”tsig-keygen. Kör kommandot plus det namn som ni ska ha på nyckeln, t.ex. ”grupp19.tsig.” (med punkt på slutet).

```
$ tsig-keygen grupp19.tsig.

key "grupp19.tsig." {
  algorithm hmac-sha256;
  secret "QpldMAPwHäoJ71uW8Vb3GaXdqOOjåiWVA7lXdCz4eu4=";
};
```

Punkt på slutet av nyckelnamnet.

Varje gång kommando körs så får man en ny ”secret”.

Använd inte exemplet ovan, som är ”trasigt”, utan generera en ny nyckel.

Kopiera TSIG-nyckel till named.conf

Lägg in nyckeln i named.conf (eller named.conf.local) på både master och slav:

```
key "grupp19.tsig." {
  algorithm hmac-sha256;
  secret "QpldMAPwHäoJ71uW8Vb3GaXdqOOjåiWVA7lXdCz4eu4=";
};
```

Punkt på slutet av nyckelnamnet.

Använd inte exemplet ovan, som är ”trasigt”, utan generera en ny nyckel.

Namn och ”secret” kommer sedan också att kunna användas av ”dig” för att ställa frågor och göra zoneöverföring med TSIG.

Använd TSIG-nyckel i slav

Lägg in motsvarande i named.conf på slaven med er nyckel och rätt IP-adress så att den använder TSIG-nyckeln vid förfrågningar (inkl. AXFR) till mastern:

```
server 192.0.2.10 {
  keys { grupp19.tsig. ;};
};
```

IP-adress till mastern.

Punkt på slutet av nyckelnamnet.

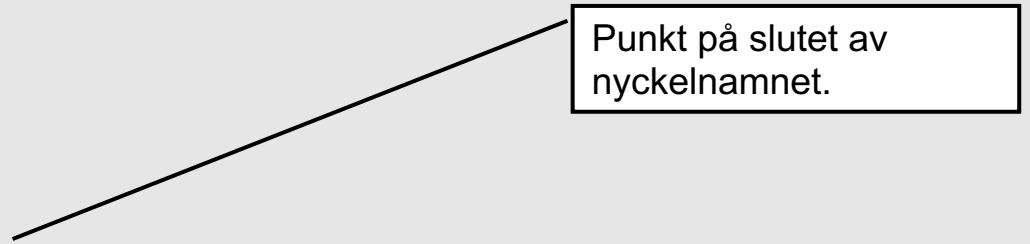
IP-adressen ska vara till mastern. Om det finns flera mastrar så blir det flera ”server” med referens till samma nyckel. Nu kommer slaven att använda nyckeln för alla förfrågningar (***query***) till mastern, inte bara AXFR, utan även t.ex. SOA-frågan.

Kräv TSIG-nyckel i master

Uppdatera named.conf på mastern så att den kräver er TSIG-nyckel för zonöverföringar:

```
# Master for namn.se
zone "namn.se" {
  type master;
  file "namn.se";
  allow-transfer { localhost; key "grupp19.tsig."; };

};
```

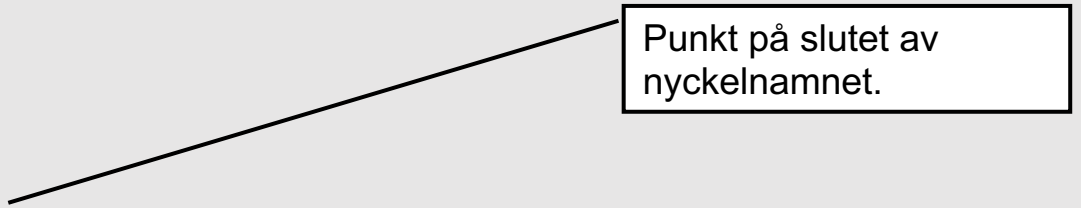


Det går att kombinera nycklar och adresser. Det är lämpligt att tillåta ”localhost” att göra zonöverföring utan nyckel.

Kräv TSIG-nyckel i slav

Uppdatera named.conf på slaven så att åtkomsten till zonen är begränsad:

```
# Slave for namn.se
zone "namn.se" {
  type slave;
  file "namn.se";
  masterfile-format text;
  masters { 192.0.2.10;};
  allow-transfer { localhost; key "grupp19.tsig."; };
  # allow-transfer { localhost; };
  # allow-transfer { none; };
};
```



Det går att begränsa olika mycket. Huvudsaken är att det inte ska gå att komma åt zonen via slaven för den som inte kan komma åt den via mastern.

Begränsa zonåtkomst

Mekanismerna kan kombineras på så sätt att vissa slavar godkännes på IP-adressen och andra med TSIG. Det räcker med att det ena är uppfyllt för att zonöverföringen ska accepteras.

Om man vill både begränsa från vilka IP-adresser som zonöverföring får begäras och kräva nyckel så måste man lägga upp någon lösning med en brandvägg utanför Bind.

▶Ska zonåtkomst begränsas?

[Till Innehåll]

Ska man begränsa zonöverföring?

Det är ett policybeslut. I vissa fall kan en zonfil innehålla information som kan användas i onda syften.

Om zonfilen bara innehåller "www" och MX (förutom NS och SOA),

```
namn.se.        MX      10 some.mail.hosting.
www.namn.se.    CNAME   some.web.hosting.
```

så är det svårt att se vitsen med begränsningen.

▶Om presentationen

[Till Innehåll]

Internets domännamnssystem

Denna presentation är framtagen 2019–2023 av Mats Dufberg på Internetstiftelsen (https://internetstiftelsen.se/). Den är en del av undervisningsmaterialet för kursen ”Internet domännamnssystem” vid Kungliga tekniska högskolan, KTH (kurskod HI1037) resp. Karlstads universitet, KAU (kurskod DVGC28).

Licens

Dokumenthistorik

- Rev A: Ursprunglig version HT 2023.

Slut.