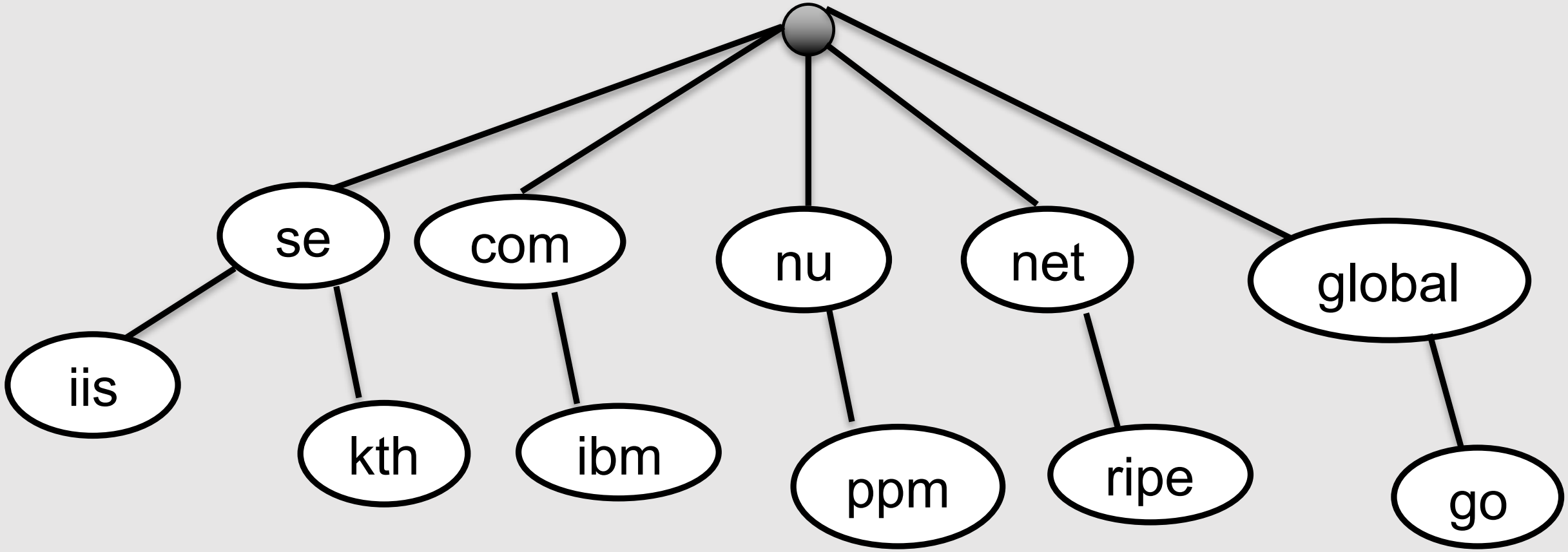

se
com
nu
net
global
iis
kth
ibm
ppm
ripe
go

INTERNET STIFTELSEN

Internets domännamnssystem*

Föreläsning FL06, VT 2024

Mats Dufberg

* Se “Internets domännamnssystem”

Innehåll

▶CNAME resp HTTP redirect

[Till Innehåll]

Första namnet bibehålls vid CNAME

Vi såg i tidigare föreläsning att ”dig” slår upp ”www.dn.se” till

```
;; ANSWER SECTION:
www.dn.se.                  CNAME   www.dn.se.edgekey.net.
www.dn.se.edgekey.net.      CNAME   e12723.a.akamaiedge.net.
e12723.a.akamaiedge.net.    A       2.22.42.116
```

Om vi använder webbläsaren och skriver in "www.dn.se" så kommer URL:en i adressfältet inte visa namnen som CNAME ger utan ”www.dn.se” kommer att bibehållas.

Första namnet bibehålls vid CNAME

Det namn vi först slår upp, ”www.dn.se”, kommer att använda IP-adressen som namnet med A-post pekar på via CNAME.

Resolvern ger både CNAME-posterna och A-posten i svaret.

Webbläsaren kommer att använda IP-adressen från A-posten för att koppla upp sig mot webbservern.

Webbläsaren kommer att använda det första namnet, "www.dn.se", för att validera TLS- (SSL-) certificatet mot.

Är www.dn.se och dn.se samma sak?

Om vi låter ”dig” slå upp ”dn.se” istället så får vi ett annat svar:

```
;; ANSWER SECTION:
dn.se. A 18.185.183.111
dn.se. A 18.196.149.61
dn.se. A 18.159.57.206
```

Inget CNAME här (inte tillåtet i apex). Istället är det helt andra IP-adresser som namnet pekar mot.

Om vi använder webbläsaren (skriver in http://dn.se i adressfältet) så kommer domännamnet i adressfältet att bytas ut till en med ”www.dn.se”. Men det bytet beror inte på DNS.

Är www.dn.se och dn.se samma sak?

Med ”curl” så kan vi slå upp mot HTTP-servern. Först slår curl upp domänen dn.se i DNS och kopplar upp sig mot den servern:

```
$ curl -v http://dn.se/
(...)
* Connected to dn.se (35.156.226.79) port 80 (#0)
(...)
< HTTP/1.1 301 Moved Permanently
< Content-length: 0
< Location: https://www.dn.se/
```

Detta är en ***HTTP redirect*** som gör att webbläsaren byter URL, och för att kunna gå mot ”www.dn.se” så måste webbläsaren göra en ny DNS-uppslagning mot sin resolver – av domänen www.dn.se.

▶Moderzon, dotterzon och delegering

[Till Innehåll]

Moderzon, dotterzon och delegering

Följande begrepp är viktiga för att förstå relationen mellan zoner som står i direkt relation till varandra:

- Moderzon (***parent zone***)
- Dotterzon (***child zone***)
- Delegering (***delegation***)

Moderzon och dotterzon är en relation mellan två zoner. En strikt hierarkisk relation.

Moderzon

Moderzonen är den “övre” zonen i relationen, den som står närmast rot i relationen (eller ev. är rotzonen). Rotzonen är alltid moderzon till zonerna i nästa nivå.

Det är i moderzonen som delegeringen placeras. Den placeras i delegeringspunkten (delegeringsnoden).

Moderzonen slutar där delegeringen finns. Där tar dotterzonen vid.

I moderzonen finns ingen annan data under delegeringspunkten utom möjligen gluedata för delegeringen.

Moderzon

Moderzonen slutar där delegeringen finns. Där tar dotterzonen vid. Tänk på DNS-trädet.

Moderzonen kan inte delegera ut två dotterzoner som står i relation till varandra.

Om .se har delegerat ut kth.se så kan .se inte samtidigt delegera ut nada.kth.se. Det skulle strida mot att kth.se och nedåt tillhör kth.se-zonen genom delegeringen av den.

Delegering till nada.kth.se kan göras i kth.se-zonen.

Delegeringen delar ut ansvar och kontroll

Namnrymden ”se” tillhör se-zonen, som se-zonen har fått genom delegeringen från root.

Genom delegering kan se-zonen **tilldela ansvaret och kontrollen för del av sin namnrymd** till en annan zon och dess servrar. Delegeringen görs genom att se-zonen pekar ut en subdomän under .se, t.ex. kth.se eller narnia.pp.se, och tilldelar den subdomänen till den utdelegerade zonen. Zonerna kth.se resp. narnia.pp.se.

När delegeringen har genomförts så har se-zonen **ingen kontroll över namnrymden under delegeringspunkten**.

Dotterzon

Dotterzonen är den “nedre” zonen i relationen, den som står längst från rot i relationen. Dotterzonen är en dotterzon i förhållande till sin moderzon. Det finns exakt en moderzon för varje dotterzon, men det kan finnas flera dotterzoner till en moderzon. (Rotzonen är ett undantag som inte har någon moderzon.)

Dotterzonen börjar i samma namn (apex) som motsvarande delegeringspunkt i moderzonen.

I dotterzonen kan det inte finnas några namn över apex. *Sådana namn skulle komma i konflikt med moderzonen.*

Delegering görs till namnservrar

Delegeringen pekar ut att dotterzonen finns (***owner name*** i NS-posterna) och var dotterzonen finns (RDATA i NS-posterna).

Delegeringen pekar ut de namnservrar (med namn) som har auktoritativ data för dotterzonen.

Det är NS-posterna som utgör den egentliga delegeringen. Glue-posterna (A, AAAA) är komplement till NS-posterna där glue-posterna måste eller kan finnas.

Moderzon-dotterzon är en relation

Moderzon och ditterzon är en relation mellan två zoner. Samma zon kan vara moderzon i en relation och dotterzon i en annan.

Moderzon	(rot)	.se	kth.se	dnskurs.se
Exempel på dotterzoner	.se, .com	kth.se, dnskurs.se	nada.kth.se	kth25.dnskurs.se

Hitta dotterzon från moderzon

Om man har tillgång till en zon (moderzon) så går det att hitta alla dotterzoner genom att ta fram alla delegeringar.

- NS-poster som finns i apex gäller zonen själv.
- NS-poster som finns längre ner är en delegering till en dotterzon.

För att vara mera korrekt – delegering av en dotterzon till dotterzonens namnservrar.

Hitta moderzon från dotterzon

Vilken är moderzonen? – Utgå från namnet på dotterzonen. Moderzonens namn kommer alltid att vara kortare än dotterzonens namn. Förkortningen är alltid hela ***labels*** till vänster, och kan vara en eller flera ***labels***.

För att hitta moderzonen så gör man en uppslagning, t.ex. genom att gå från rot och följa delegeringar.

Hitta moderzon från dotterzon

Dotterzon: www.sth.prod.exempel.xa

Tekniskt möjliga kandidater till moderzon:

- sth.prod.exempel.xa
- prod.exempel.xa
- exempel.xa
- .xa
- . (rot)

Hitta moderzon från dotterzon

Dotterzon: www.sth.prod.exempel.xa

Omöjliga kandidater till moderzon:

- gbg.prod.exempel.xa
- sth.prod.exempel.xb
- prod.example.xa
- example.xa

Hitta moderzon från dotterzon

Vilken är moderzonen till narnia.pp.se?

- pp.se?
- .se?
- . (rot)?

Om vi vet att zonen .se finns, så rot kan inte vara moderzon till narnia.pp.se, men är det .se eller pp.se? Utan att slå upp så vet vi inte säkert.

(Det är .se.)

▶Ordning i ett RRset

[Till Innehåll]

Vilken ordning har DNS-posterna?

”dig” ger t.ex. följande svar på frågan ”dn.se. A”:

```
;; ANSWER SECTION:
dn.se. A 35.156.226.79
dn.se. A 52.57.46.29
dn.se. A 18.194.204.241
```

”dig” ger samma svar som namnservern. Namnservern måste skicka A-posterna i någon ordning. Normalt så ändras ordningen varje gång man frågar.

Posterna inom ett RRset har ingen ordning i sig utan det är bara hur de presenteras. Ett RRset är en **oordnad** lista.

Vilken ordning har DNS-posterna?

Följande är exakt samma svar ur DNS-synvinkel (3 av 6 möjliga):

```
;; ANSWER SECTION:
dn.se. A 35.156.226.79
dn.se. A 52.57.46.29
dn.se. A 18.194.204.241

;; ANSWER SECTION:
dn.se. A 52.57.46.29
dn.se. A 35.156.226.79
dn.se. A 18.194.204.241

;; ANSWER SECTION:
dn.se. A 18.194.204.241
dn.se. A 35.156.226.79
dn.se. A 52.57.46.29
```

Vilken DNS-post ska klienten välja?

Klienten, t.ex. en webbläsare, får följande svar. Vilken IP-adress ska webbläsaren välja för uppkopplingen?

```
;; ANSWER SECTION:
dn.se. A 35.156.226.79
dn.se. A 52.57.46.29
dn.se. A 18.194.204.241
```

Klienten avgör själv vilken IP-adress den ska välja. Den kan t.ex. ha en preferens för IP-adress som ligger nära. Men...

Vilken DNS-post ska klienten välja?

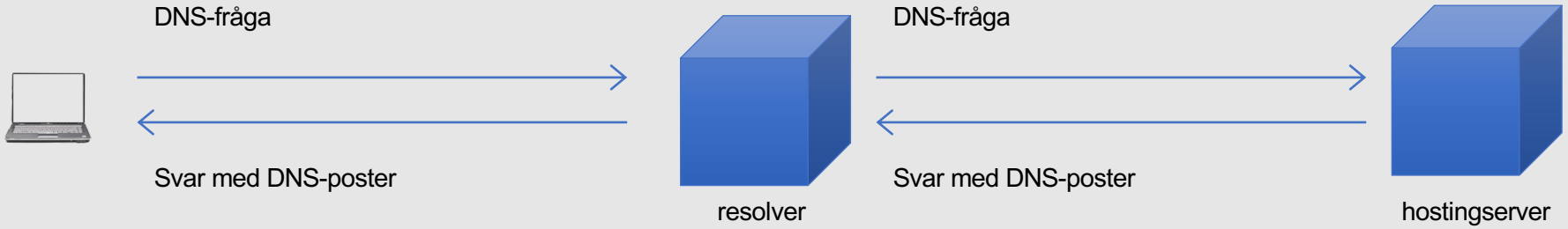
I normalfallet:

- Klienten kommer att ta den första adressen för uppkopplingen.
- Om den första adressen inte fungerar, så tar den adress nummer två.

Detta betyder att det kan ha betydelse i vilken ordning som multipla DNS-poster kommer till klienten.

Vilken namnserver sätter ordningen?

Det är alltid resolvern som avgör vilken ordning som DNS-posterna presenteras för klienten. Hostingservern kan inte kontrollera detta. Om svaret kommer från resolverns cache så är inte ens hostingservern inblandad i svarandet.



Ordning

Det finns olika sätt som resolvern kommer att presentera DNS-posterna i upprepade frågor om samma RRset:

- fixerad (”fixed”)
- slumpmässig (”random”)
- round robin (”cyclic”)

Fixerad (”fixed”)

Upprepad fråga om samma sak ger samma ordning på DNS-posterna i en RRset.

Det är i praktiken bara möjligt att uppnå i en sluten miljö eftersom det är resolvrarna som måste konfigureras.

Troligen väljer man en annan lösning istället, t.ex. bara svara med en DNS-post med kort TTL, men ge olika svar beroende på var frågan kommer från. Servrarna bakom övervakas så att en IP-adress till ”bra” server lämnas ut.

Slumpmässig (”random”)

Det ger den bästa balanseringen mellan alla svar i ett RRset. Om resolvern är satt för slumpmässig presentation så blir det så för dess användare, men även hostingservern kan påverka. Om hostingservern ger slumpmässig ordning och har en kort TTL så kommer fördelningen bli jämn.

Om användarna använder många olika resolvrar så spelar TTL mindre roll för fördelningen.

Numera så är det slumpmässig ordning från Bind om man inte konfigurerar något annat.

Round robin (”cyclic”)

Ordningen mellan DNS-posterna är fast, men vid varje svar så flyttas den i toppen ned ett steg.

```
;; ANSWER SECTION:
dn.se. A 35.156.226.79
dn.se. A 52.57.46.29
dn.se. A 18.194.204.241

;; ANSWER SECTION:
dn.se. A 52.57.46.29
dn.se. A 18.194.204.241
dn.se. A 35.156.226.79

;; ANSWER SECTION:
dn.se. A 18.194.204.241
dn.se. A 35.156.226.79
dn.se. A 52.57.46.29
```

Round robin (”cyclic”)

Om 35.156.226.79 inte svarar så kommer 52.57.46.29 att vara nästa i tur, vilket ger sämre balansering.

```
;; ANSWER SECTION:
dn.se. A 35.156.226.79
dn.se. A 52.57.46.29
dn.se. A 18.194.204.241
```

Tidigare så var ”round robin” det normala vid DNS-svar, och default i Bind. Nu är slumpmässig ordning default i Bind.

▶Cache poisoning

[Till Innehåll]

Cache poisoning

Cache poisoning betyder att man lyckas få in ond och medvetet felaktig data i en resolvers cache. Denna data kommer sedan att förmedlas till de som frågar efter den.

Om TTL är hög så kommer den att ligga kvar länge i cachen.

Cache poisoning

Ofta kan det handla om en felaktig A- eller AAAA-post med en IP-adress som styr offret till t.ex. falsk webbplats eller liknande, men det kan vara andra DNS-poster och gälla annan trafik.

Det kan vara falsk data som t.ex. styr mailen mellan mailservrar eller var uppdatering av datorns programvara ska hämtas.

Klassiskt trick för cache poisoning

```
; <<>> DiG 9.10.6 <<>> @ns.namn.se www.namn.se
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 32925
;; flags: qr rd; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;www.namn.se.                   IN  A

;; ANSWER SECTION:
www.namn.se.            600     IN  CNAME   www.kth.se.
www.kth.se.           86400     IN  A       192.0.2.40

;; Query time: 66 msec
;; SERVER: 192.0.2.246#53(192.0.2.246)
;; WHEN: Wed Jan 30 22:03:14 CET 2019
;; MSG SIZE  rcvd: 217
```

Vi får användaren att fråga efter www.namn.se.

Inskjuten information.

Cache poisoning

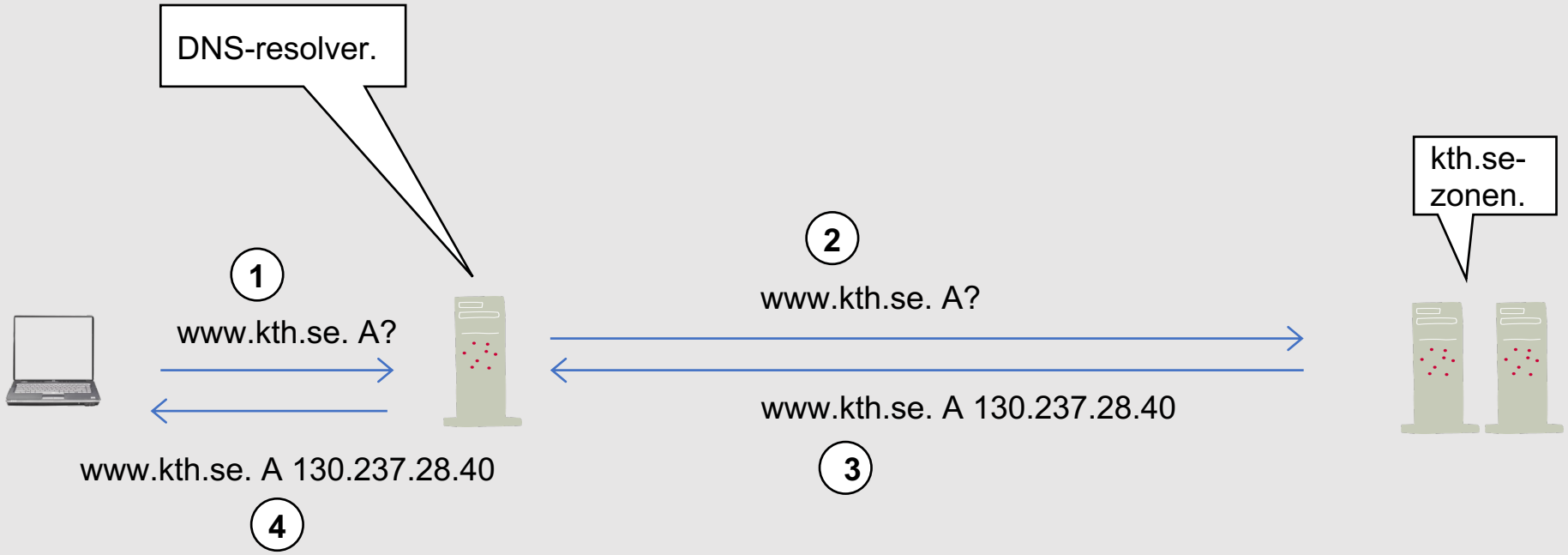
I DNS:s barndom så skulle resolvern glatt lägga in allt den fick i cachen.

- Lura resolvern att slå upp www.namn.se (ond domän)
- Namnservrarna fyllde cachen med data som kom.

Modern DNS-programvara tar inte in extradata som inte passar in i frågan. Det kan det finnas resolvrar som inte kontrollerar.

Men det finns andra tricks som är mera sofistikerade.

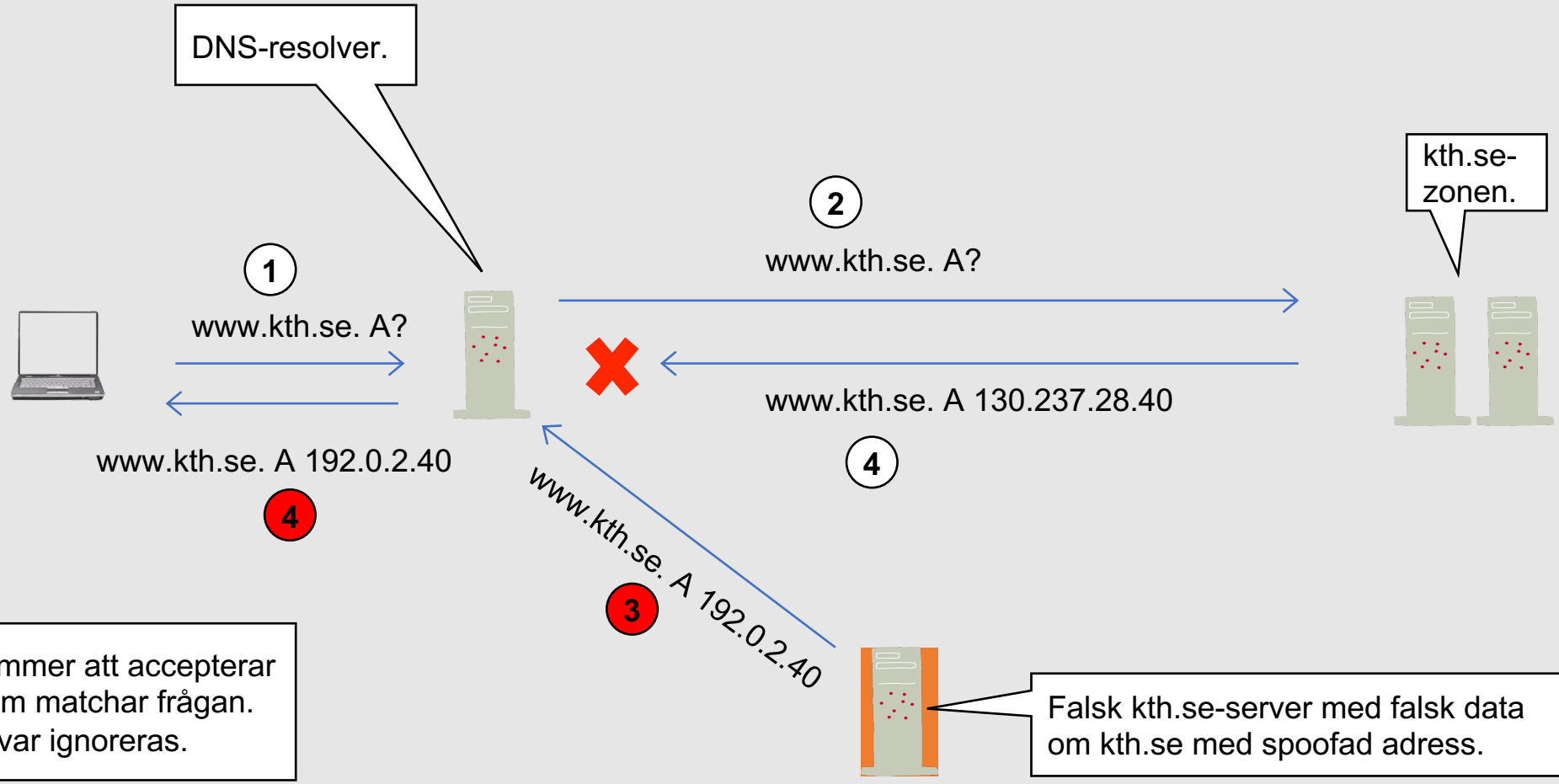
Korrekt uppslagning



▶Man-in-the-middle-attack

[Till Innehåll]

Uppslagning med ”man-in-the-middle” (1)



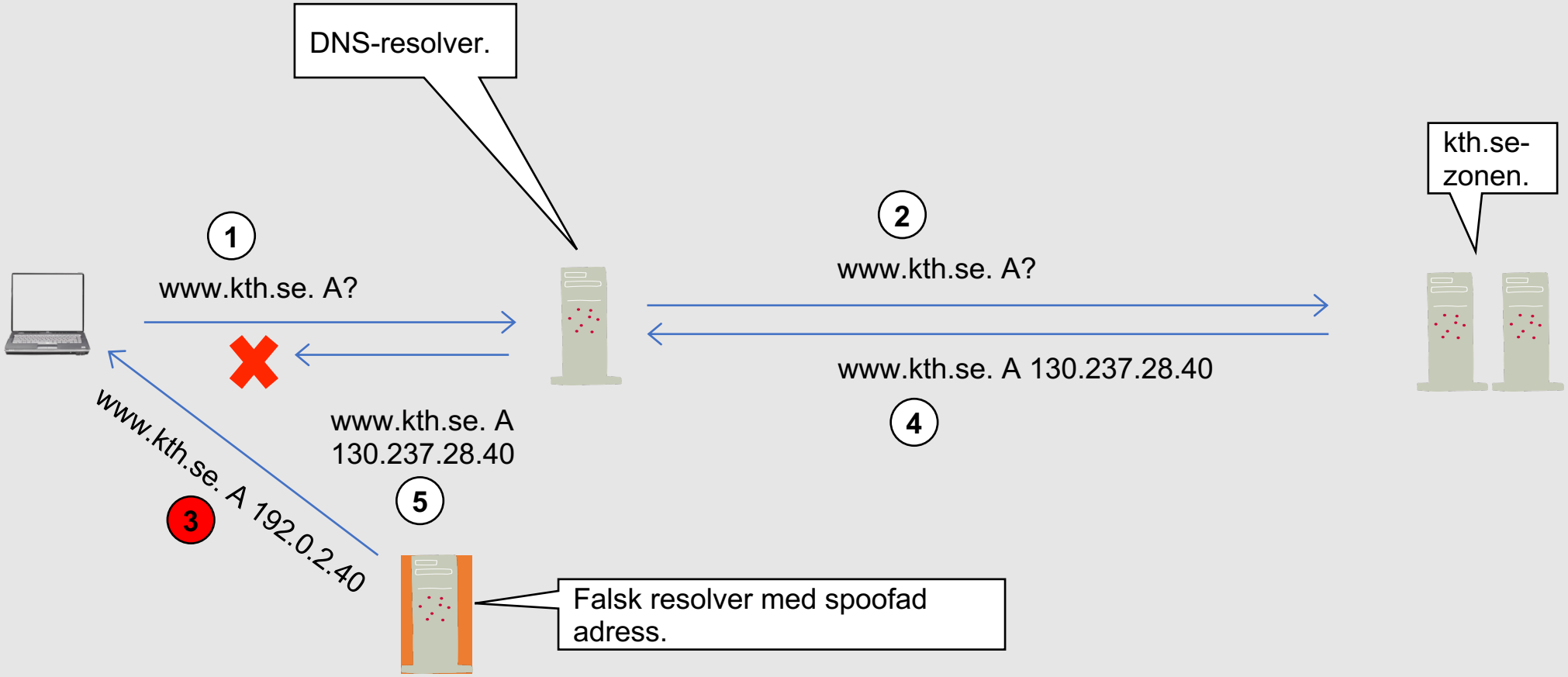
Resolvern kommer att accepterar första svar som matchar frågan. Kommande svar ignoreras.

Svar matchar fråga

För att ett DNS-paket från servern ska accepteras som svar (***response***) på frågan (***query***) som skickades så krävs följande:

1. Svaret ska komma från samma IP-adress och port som frågan skickades till.
2. "Question section" i svaret ska stämma med dito i frågan.
3. "Message ID" i svaret ska stämma med dito i frågan.

Uppslagning med ”man-in-the-middle” (2)



Skyddet i DNS mot falska svar

För att ett svar ska accepteras så måste det ha samma ID som frågan.
ID är ett heltal med 16 bitar, vilket ger 65.536 olika möjliga värden.

- Det är klienten som skapar ID, och första steget till skydd är att slumpa ID istället för att skapa dem i ordning.

Skyddet i DNS mot falska svar

Säkerhetsexperten Dan Kaminsky visade att det var otillräckligt med slumpat ID och efter det finns det oftast ett skydd till:

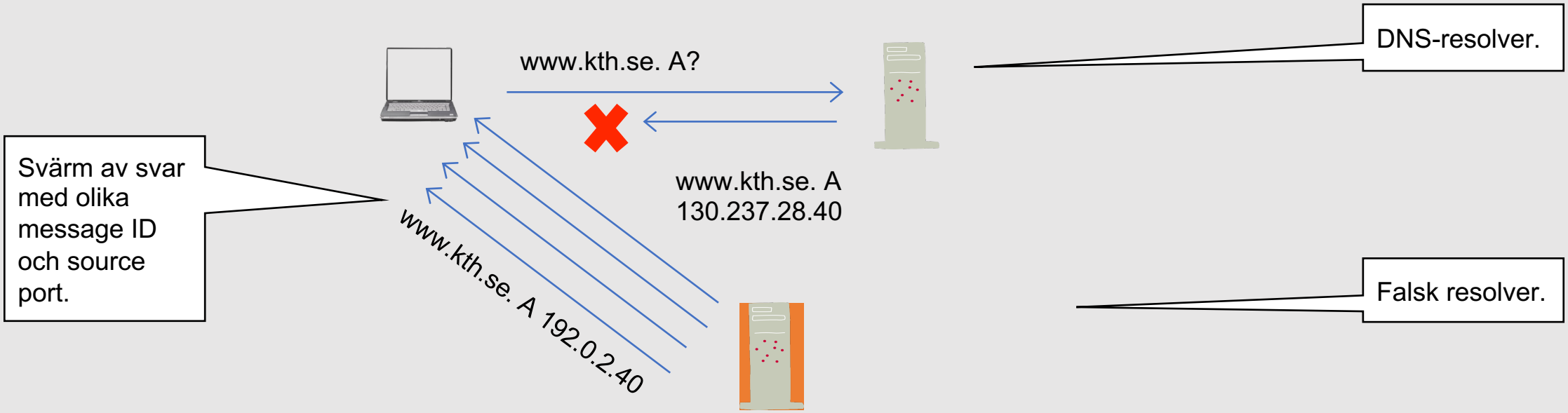
- ”Source port” slumpas av klienten när den ska skicka frågan.

Inte ens tillsammans ger dessa ett fullgott skydd.

Dessa två steg är vad som kan tas i vanlig DNS. Nästa steg är DNSSEC.

Många svar på en gång

Svaret som skjuts in måste stämma med både ID och port, men det finns inget som hindrar att den som attackerar skickar många svar med olika ID och port för att något svar ska stämma.



Kräver ”man-in-the-middle” UDP?

Scenariena med ”man-in-the-middle” kräver tillgång till nätet mellan resolvern och hostingservern resp. mellan dator och resolver.

Lättare med UDP än TCP, men UDP är inget krav.

▶Elak resolver

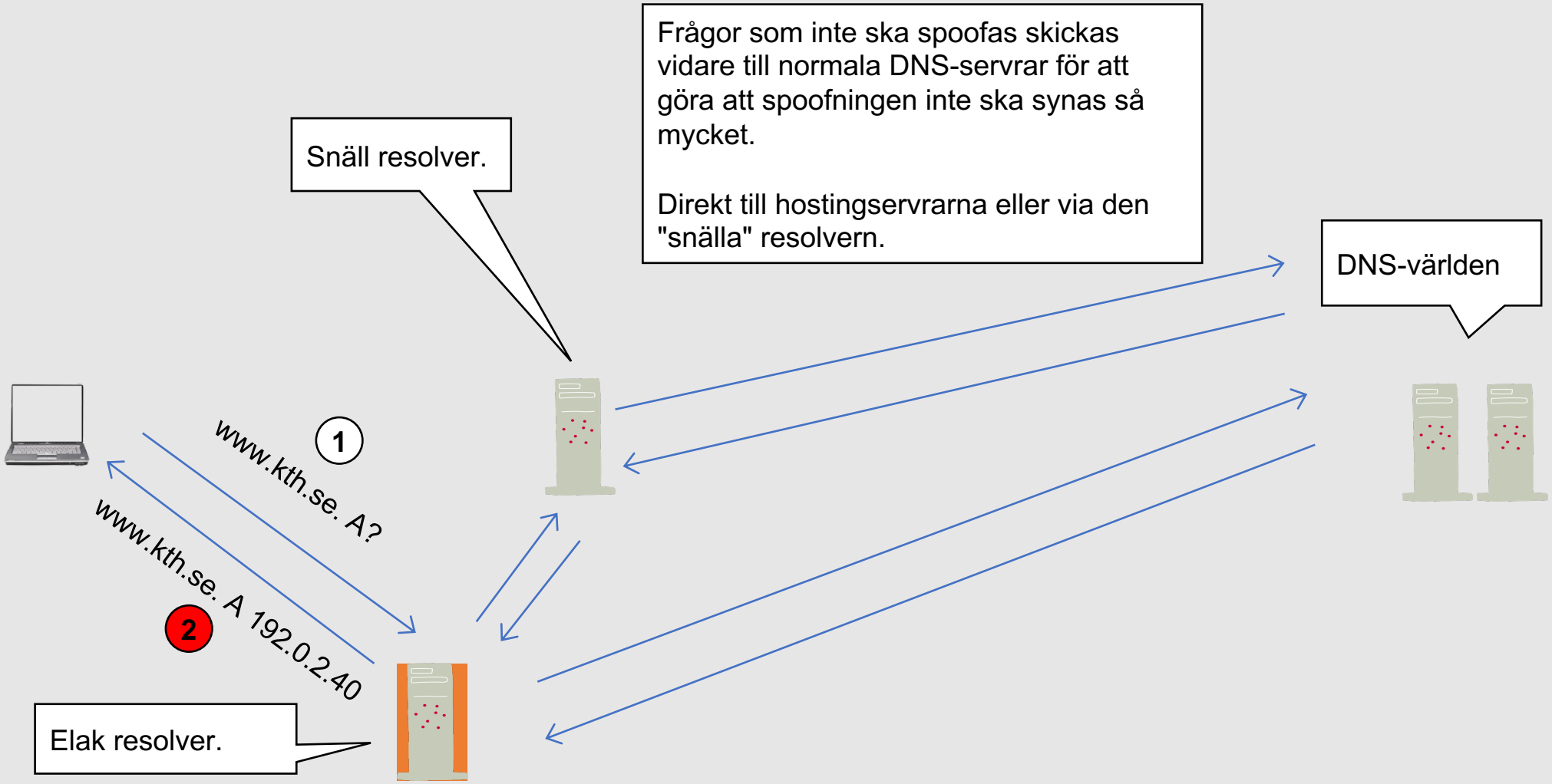
[Till Innehåll]

Elak resolver

Om man kan få klienten att skicka sina frågor direkt till en ”elak resover” så finns det inga begränsningar på vilka frågor som man kan ge falska svar.

- ARP-spoofning för att styra om trafiken till resolvern till en elak resolver.
- Extra, elak DHCP-server som delar ut adresser ur samma spann, men ger en elak resolver.
- Alternativ wifi-accesspunkt som är ”proxy” till den riktiga, men som delar ut en elak resolver.

Uppslagning med elak resolver



▶Modifierande resolver

[Till Innehåll]

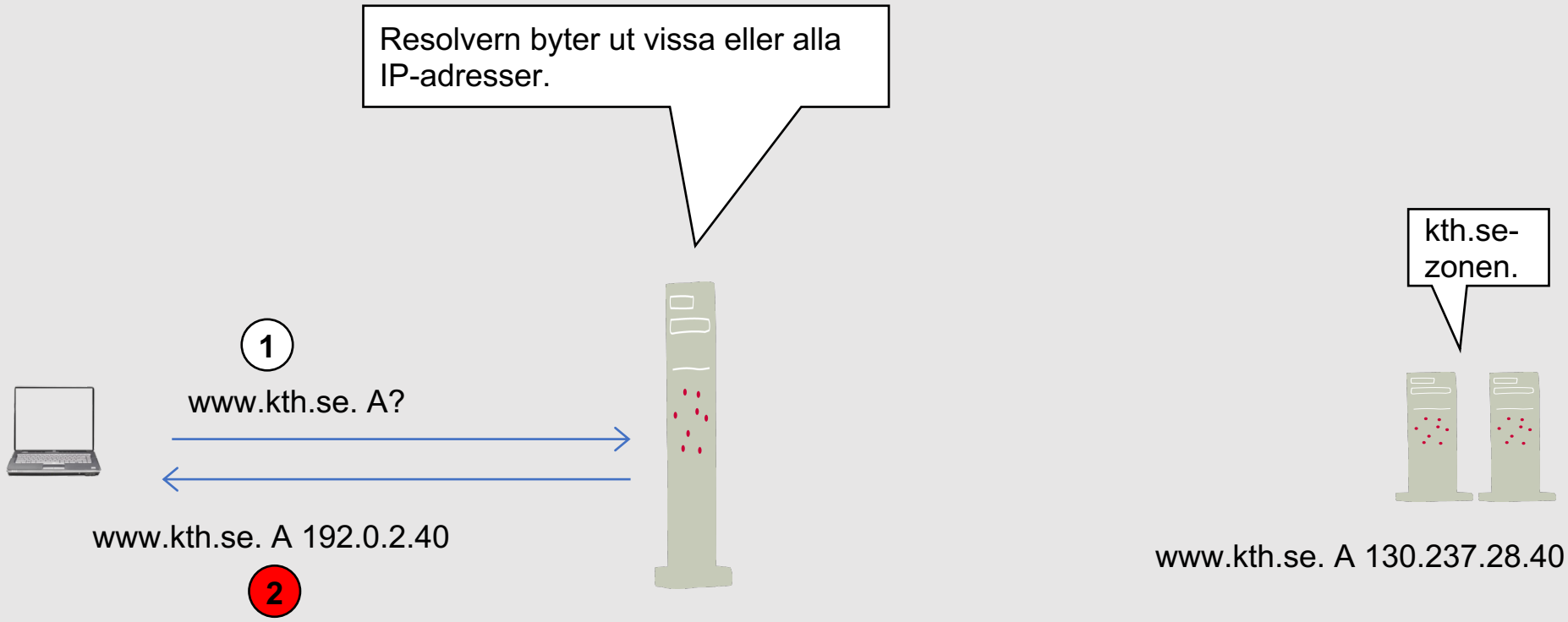
Modifierande resolver

En falsk resolver är en resolver som i lönndom tar över uppslagningarna, men även en vanlig resolver som tillhandhålls av en skola, ett företag, en myndighet eller en ISP kan tänkas modifiera vissa svar.

ISP:er i Sverige blockerar regelmässigt s.k. barnporrdomäner genom att i DNS ge ett ”falskt” svar.

Samma teknik som används för den falska resolvern kan även användas för modifieringar.

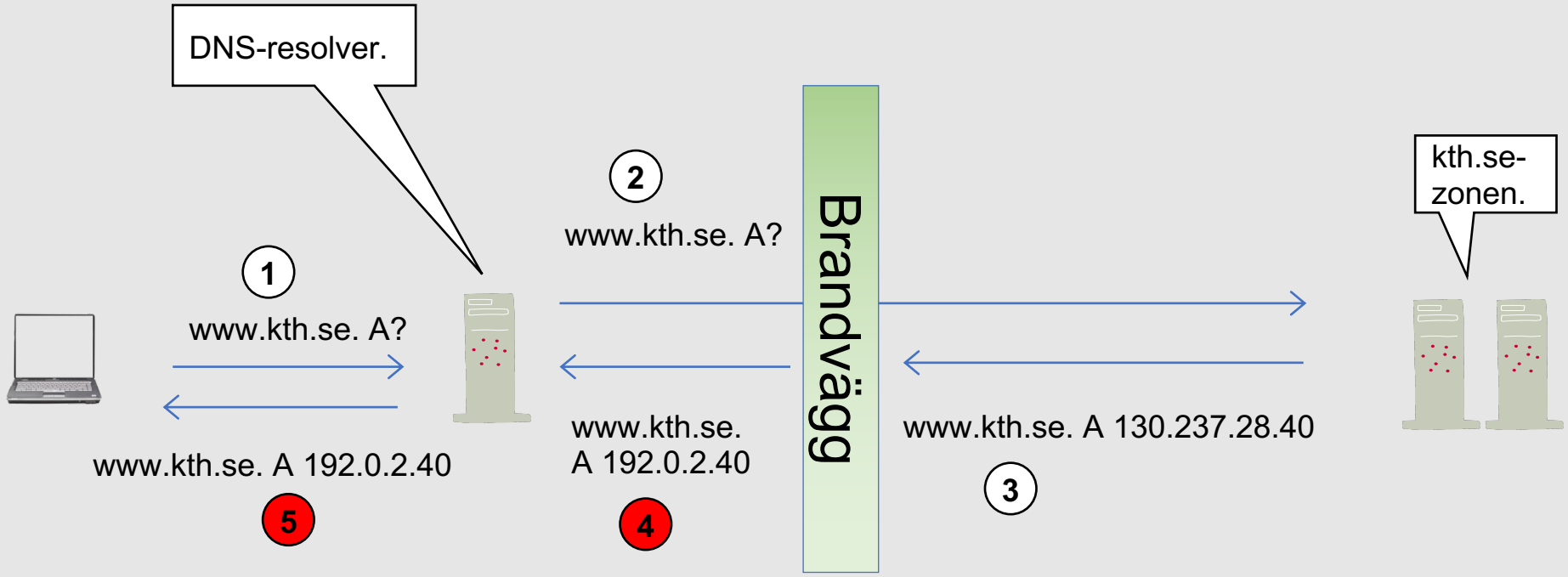
Uppslagning med modifierande resolver



Modifierande resolver

Många fall så styrs man om till en inloggningssida när man ansluter till ett ”öppet” wifi-nät, t.ex. på ett hotell. Den omstyrningen görs ofta med en modifierande resolver kombinerat med filter på IP-nivån.

Modifierande brandvägg, företag eller land



▶Bortglömd NS för en zon

[Till Innehåll]

Bortglömd NS

Vi har namn.se delegerat enligt följande:

```
namn.se.          NS   ns1.namn.se.    ; master
                  NS   ns1.red.xa.     ; slav
                  NS   ns1.blue.xa.    ; slav
                  NS   ns1.green.xa.   ; slav
ns1.namn.se.      A    192.0.2.37      ; glue
```

Man slutar använd red.xa och servern ns1.red.xa döps om till ns2.namn.se. Man slutar att betala för red.xa – red.xa avregistreras och upphör – men missar att ändra delegeringen.

Vad händer?

Bortglömd NS

NS i delegering:

```
namn.se.   NS   ns1.namn.se.
           NS   ns1.red.xa. ; Fungerar inte längre!
           NS   ns1.blue.xa.
           NS   ns1.green.xa.
```

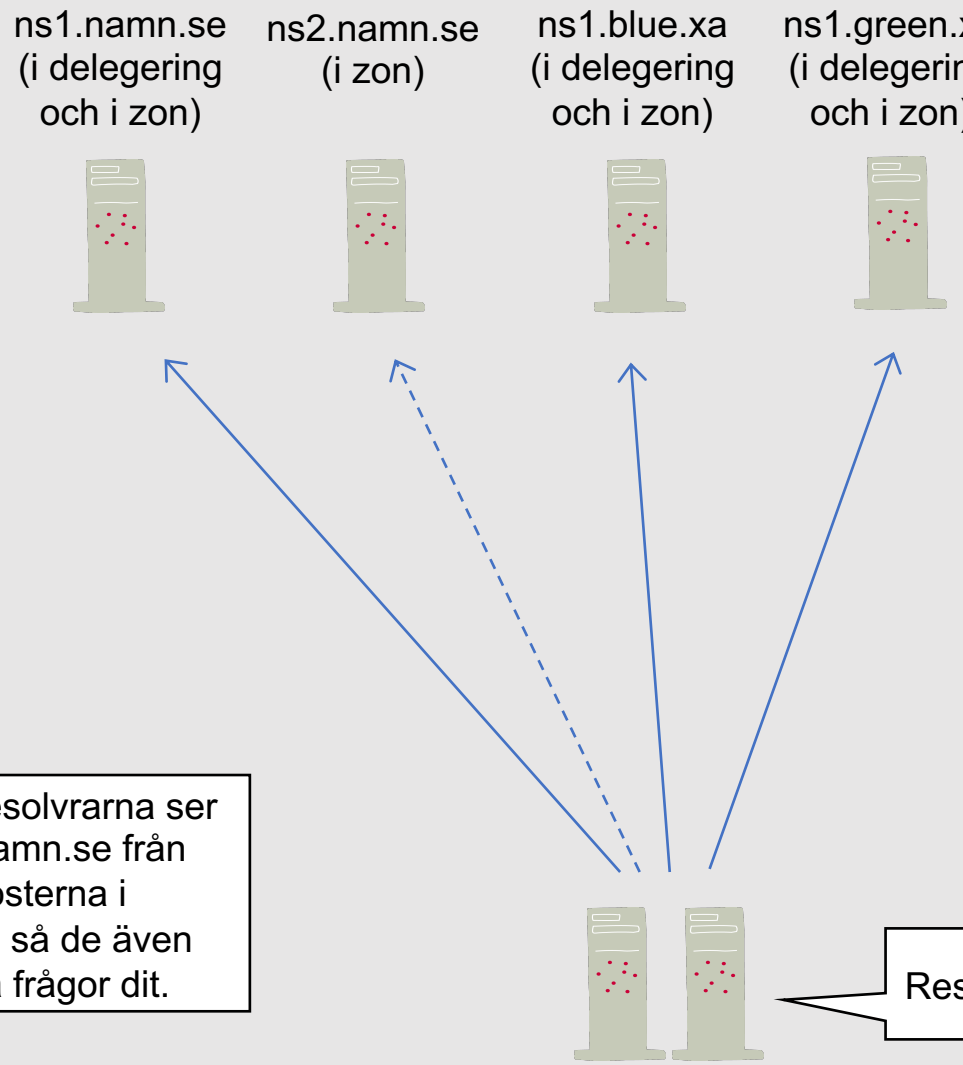
NS i zonen:

```
namn.se.   NS   ns1.namn.se.    ; master
           NS   ns2.namn.se.    ; slav
           NS   ns1.blue.xa.    ; slav
           NS   ns1.green.xa.   ; slav
```

DNS är förlåtande så ingen kommer att märka felet om man inte kör ett verktyg som Zonemaster och vet vad man ska titta efter. Eller kör egen uppslagning och vet hur man ska kontrollera.

(Glue tillkommer men listas inte här.)

Bortglömd NS



Resolvrarna kommer att se ns1.red.xa i delegeringen, men eftersom det inte går att slå upp den så finns det ingen IP-adress så de går snabbt över till annan server.

Detta märks inte i uppslagningstiden.

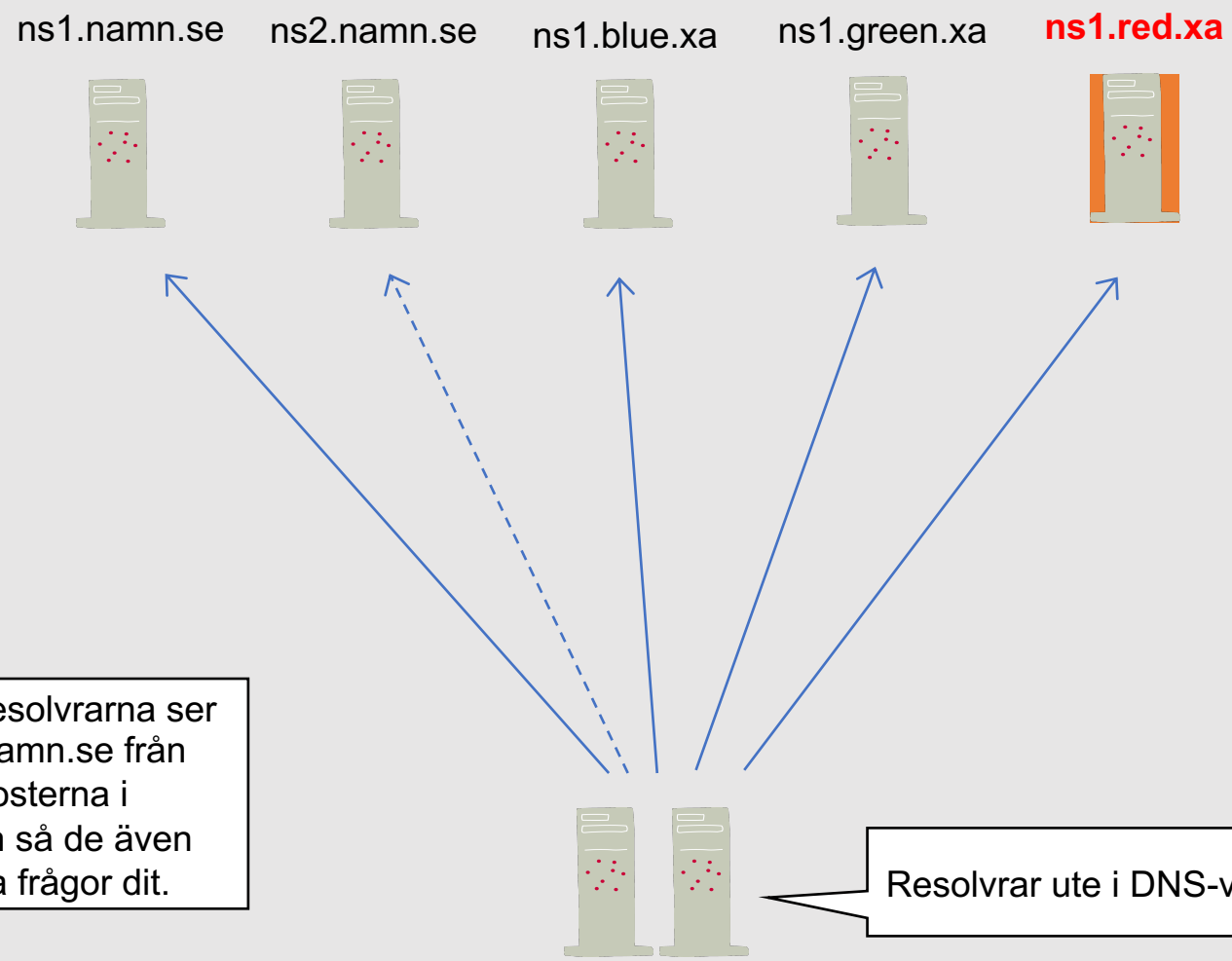
Bortglömd NS

Någon ”ond” person upptäcker att red.xa är ledig och registrerar den och ser DNS-frågor på namnet ns1.red.xa.

- Vad kan den göra?

Den kan tilldela IP-adress till ns1.red.xa (skapa adressposter, A, AAAA) och sätta upp en server som lyssnar på adressen eller adresserna för att se vilken typ av trafik det är.

Bortglömd NS



Alla NS kommer att få frågor, även ns1.red.xa. Initialt så kommer den kanske inte att kunna svara på frågor.

Men...

Bortglömd NS

Om namn.se är öppen för AXFR så kan ns1.red.xa hämta zonen och sedan modifiera valda DNS-poster.

Om namn.se inte är öppen för AXFR så är det inget hinder:

- När ns1.red.xa får en inkommande fråga så ställer den omedelbart samma fråga till ns1.namn.se och lägger svaret i sin cache.
- ns1.red.xa besvarar den inkommande frågan från cache.

Om man inte använder ett verktyg som Zonemaster så kommer man inte att märka det.

Bortglömd NS

Från cache så kan man hitta intressanta poster att manipulera med.

Nu tar hen ett steg till

- ns1.red.xa sätter ett falskt AA på vissa svar som skickas.
- Valda DNS-poster modifieras.

Zonemaster skulle inte märka detta ifall inte modifieringen är på en DNS-post som kontrolleras, t.ex. SOA.

Bortglömd NS

Vilka svar kan modifieras? Det finns många möjligheter.

- T.ex. MX.

Om MX modifieras så kan mailen styras till annan mailserver där mailen kopieras och sedan skickas de vidare till rätt MX så att det inte ska märkas att något händer. Det är få som kommer att upptäcka att mailen har gått en annan väg.

TTL kan sättas upp så att svaren från ns1.red.xa får större genomslag.

▶Öppen zonfil

[Till Innehåll]

Öppen zonfil

Även om det är fritt att fråga om vilket namn som helst så betyder inte det att det ska vara fritt fram att hämta zonen med AXFR.

Jfr. med att man kan fråga om vilket telefonnummer som helst hos https://www.hitta.se/ men man kan inte hämta hela databasen.

Är öppen zonfil ett problem?

Zonfilen kan avslöja nya system som har tillkommit, kanske system under installation. Det finns risk att dessa inte har säkrats ordentligt innan installationen har slutförts.

Zonfilen kan alltså, oavsiktligt, ge information till någon som vill göra intrång.

Samtidigt så ska man inte betrakta informationen i zonen som skyddad ifall man kan fråga efter den via en resolver.

Är öppen zonfil ett problem?

Fördelen med öppen zonfil är att det blir enklare att sätta upp och hantera slavservrar, men oftast är det bättre att sätta upp en övervakning, t.ex. med Zonemaster, för att upptäcka ifall synkroniseringen mellan master och slav har slutat att fungera.

För det ***är*** ett problem om synkroniseringen slutar att fungera.

▶Öppen resolver

[Till Innehåll]

Avsiktligt öppen resolver

De som tillhandahåller öppna resolvrar, t.ex. Google och Quad 9, förväntas ha metoder att motverka missbruk och berörs inte här.

Oavsiktligt öppen resolver

En öppen resolver kan missbrukas så att den inte fungerar för de som ska använda den.

- Om någon vill ha hela innehållet i en TLD (alla delegeringar) så kan man gissa och sedan ställa frågor. Om resolvern är öppen så kan man utnyttja den.

DOS- och DDOS-attack

Namnservrar kan både vara mål och verktyg för DOS- och DDOS-attacker.

- Olika domäner och rotzonen har varit utsatta för attacker av olika skäl. Förmodligen så finns det utpressningssyften med i bilden.
- Om man gör en ”man-in-the-middle attack” så kan man attackera de legitima namnservrarna så att de har svårt att svara.

DOS- och DDOS-attack

En resolver eller något som fungerar som resolver eller "forwarder" kan vara verktyg för attack.

- Ofta används ”botnet” eller dålig konsumentutrustning, t.ex. oskyddade bredbandsroutrar.
- Öppna resolvrar, speciellt oavsiktligt öppna resolvrar, kan vara verktyg för attacker.

▶Amplification-attack

[Till Innehåll]

Resolvrar som redskap för DOS-attacker

Resolvrar kan bli ofrivilliga redskap för DOS-attacker.

En liten DNS-fråga (***query***) kan ge ett stort svar (***response***), d.v.s. uppförstoring, ”amplification”.

Query

```
; <<>> DiG 9.10.6 <<>> kth.se txt +noedns +qr
;; global options: +cmd
;; Sending:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 63233
;; flags: rd ad; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;kth.se.                        IN  TXT
```

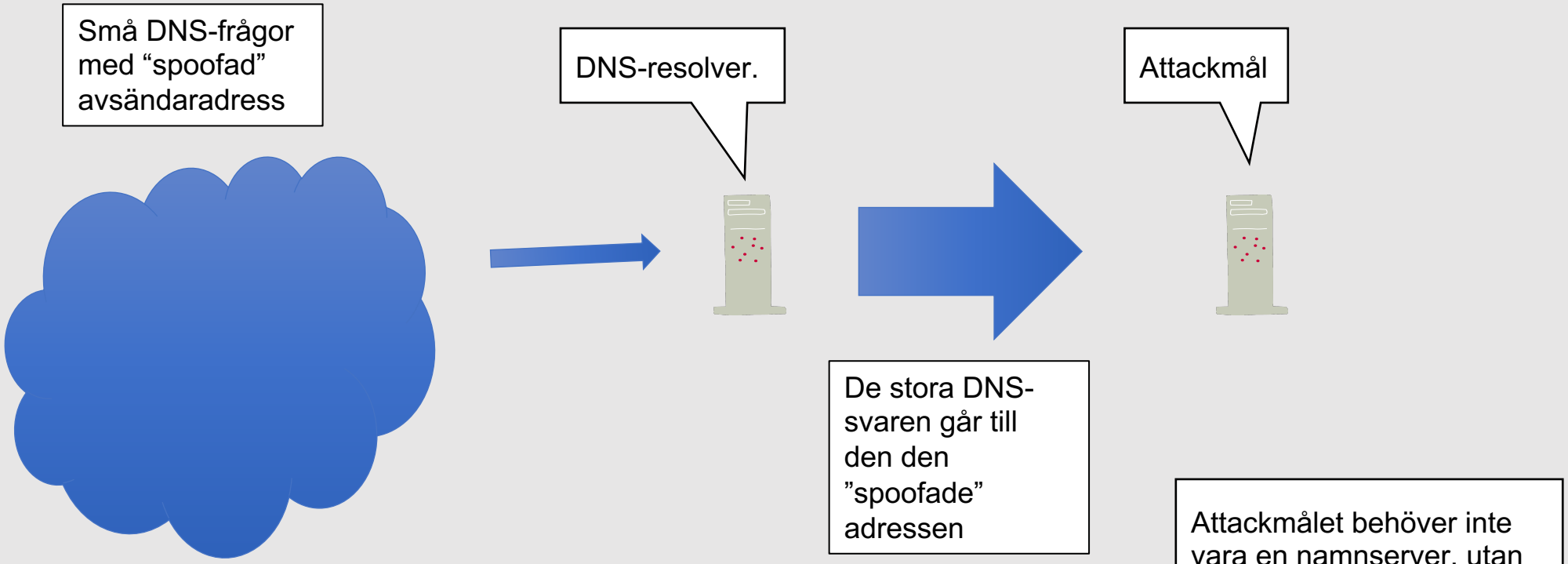
Paketet är på **24** byte inkl allt.

Response

```
;; ANSWER SECTION:
kth.se. 335 IN TXT "5 - Tel. +46 8 790 60 00"
kth.se. 335 IN TXT "2 - Kungliga Tekniska H|gskolan"
kth.se. 335 IN TXT "3 - SE-100 44  STOCKHOLM"
kth.se. 335 IN TXT "4 - SWEDEN"
kth.se. 335 IN TXT "facebook-domain-verification=ybxcywd9y5omfyzv9a4hi1122pxhys"
kth.se. 335 IN TXT "MS=ms86914267"
kth.se. 335 IN TXT "v=spf1 include:_spf.kth.se include:customers.clickdimensions.com ~all"
kth.se. 335 IN TXT "adobe-idp-site-
verification=9cb3a0e2a6197322d0dd6d29de0e1ac1ce9f990e3d5bd864aa4c19007f78ef91"
kth.se. 335 IN TXT "1 - Royal Inst of Technology"
```

Paketet är på **491** byte inkl allt.

DOS-attack via ”amplification”



Kräver attacken UDP?

"Spoofad" avsändaradress betyder att IP-adressen i IP-paketet inte är den som skickar paketet, utan den som man vill att det ska se som DNS-frågan kommer från.

Om man ”spoofar” avsändaradressen så kommer det normalt inte fungera med TCP eftersom TCP startar med "three-way handshake".

▶DOS-attack via resolver

[Till Innehåll]

DOS-attack via resolver

Målet för attacken är att hosting av en domän, t.ex. namn.se, blir otillgänglig.

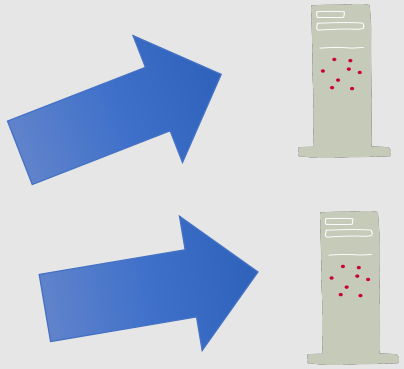
- Se till att namnservrarna för namn.se får så många frågor att den inte orkar svara.
- Ställ frågan XXX.namn.se till resolvrar där XXX är ett slumpat namn.

Resolvern kommer att cacha, men det kommer hela tiden nya namn i frågorna.

DOS-attack via resolver

DNS-frågor med “spoofad” avsändaradress eller från botnet

DNS-resolvrar, som ev. också får problem med alla obesvarade frågor.



Attackmålet är en viss domän som vi gör oåtkomlig genom att sänka dess NS.

DNS-frågorna går vidare till namnservrarna som inte orkar svara

Kräver attacken UDP?

Om man ”spoofar” avsändaradressen så kommer det normalt inte att gå med TCP, men med ett botnet så går det bra med TCP.

På en ISP:s nät kan det finnas botnet. Även konsumentutrustning kan användas.

Är DNSSEC en lösning?

Just för detta scenario (upprepade slumpade frågor) så kan DNSSEC göra det svårare att lyckas med attacken.

▶Är UDP ett problem?

[Till Innehåll]

Är UDP ett problem?

Attacker som kräver ”spoofade” avsändaradresser är svåra eller omöjliga att utföra med TCP.

- En övergång till TCP skulle å andra sidan göra namnservrarna mer belastade.

Oavsett, idag så är UDP en integrerad del av DNS och det är omöjligt att välja bort UDP.

▶DNS-trafik i klartext

[Till Innehåll]

DNS-trafik i klartext

All DNS-trafik går i klartext. Den som nätmässigt sitter så att den kan se DNS-trafiken kan också granska innehållet i alla DNS-frågor. Det kan vara en pusselbit för att samla information om en person eller ett företag.

Detta har länge inte setts som något problem eftersom DNS-informationen som sådan är publikt tillgänglig. DNSSEC, som vi kommer att titta på nästa gång, har ingen lösning på detta.

Det finns idag lösningar som helt eller delvis löser detta problem, när eller om det införs. Vi kommer att titta på det senare.

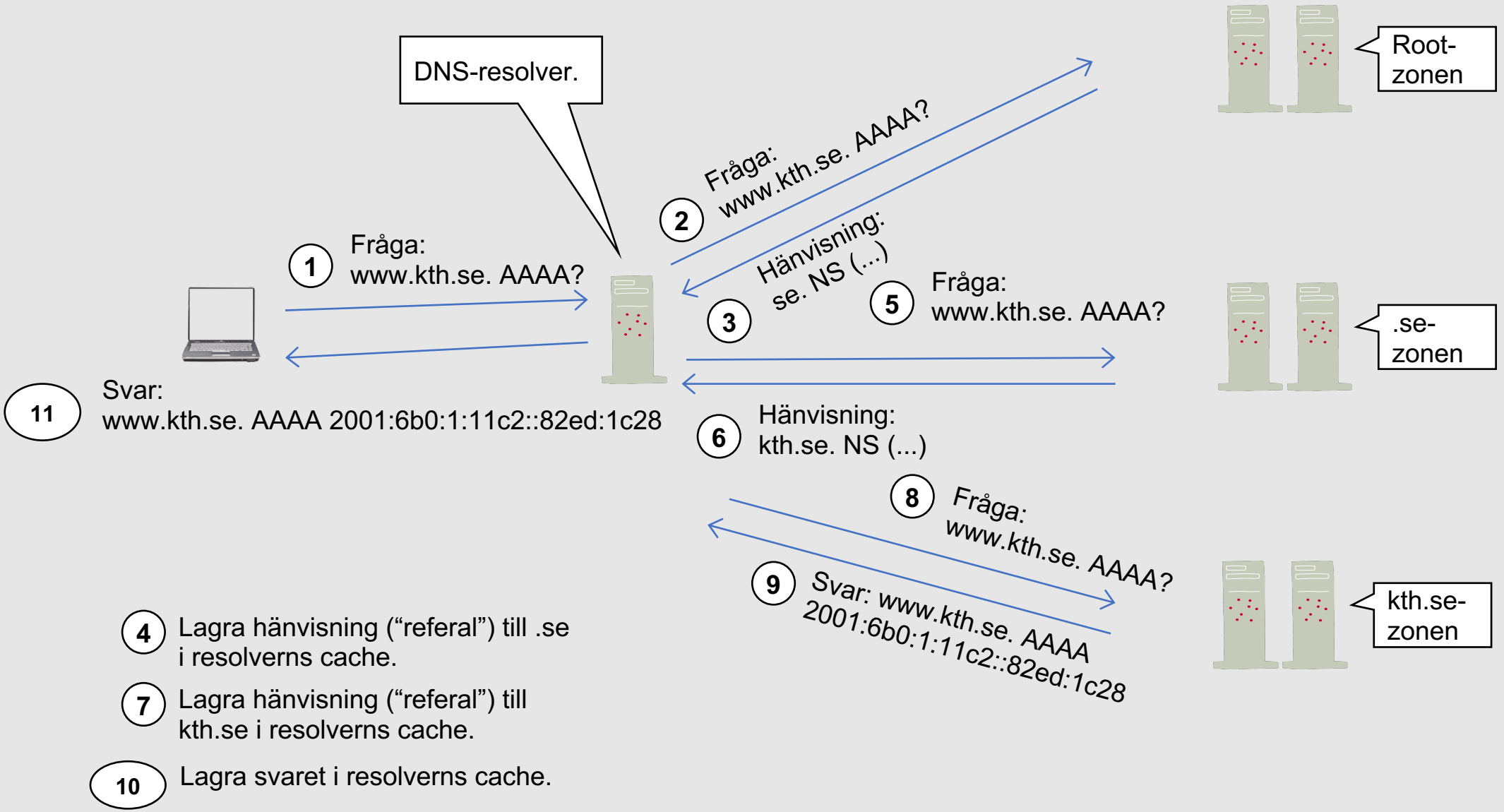
▶Query name minimisation

[Till Innehåll]

DNS-frågan visas överallt

Nästa bild, som vi har tittat på flera gånger, visar hur DNS-frågan upprepas på alla nivåer trots att den inte är relevant innan vi har hittat rätt zon.

Varför ska rotnamnservern kunna logga att någon frågar efter ”www.kth.se. AAAA”?


DNS-resolver.
Root-zonen
.se-zonen
kth.se-zonen
1 Fråga: www.kth.se. AAAA?
2 Fråga: www.kth.se. AAAA?
3 Hänvisning: se. NS (...)
5 Fråga: www.kth.se. AAAA?
6 Hänvisning: kth.se. NS (...)
8 Fråga: www.kth.se. AAAA?
9 Svar: www.kth.se. AAAA 2001:6b0:1:11c2::82ed:1c28
11 Svar: www.kth.se. AAAA 2001:6b0:1:11c2::82ed:1c28
4 Lagra hänvisning (“referal”) till .se i resolverns cache.
7 Lagra hänvisning (“referal”) till kth.se i resolverns cache.
10 Lagra svaret i resolverns cache.

Är det något problem?

Att frågan sprids till så många ställen kan vara integritetsproblem. Det ökar möjlighet till övervakning, och det ökar möjlighet att samla information om personer och företag som kan vara icke-önskvärd för de utsatta.

Måste DNS-frågan visas överallt?

Det finns inget krav på att resolvern upprepar den ursprungliga frågan ”www.kth.se. AAAA?” på alla nivåer. Det har varit ett enkelt sätt att hitta rätt zon och namnservrar, som vi visade när vi emulerade en resolver och slog upp ”www.dn.se. A”.

Query name minimisation

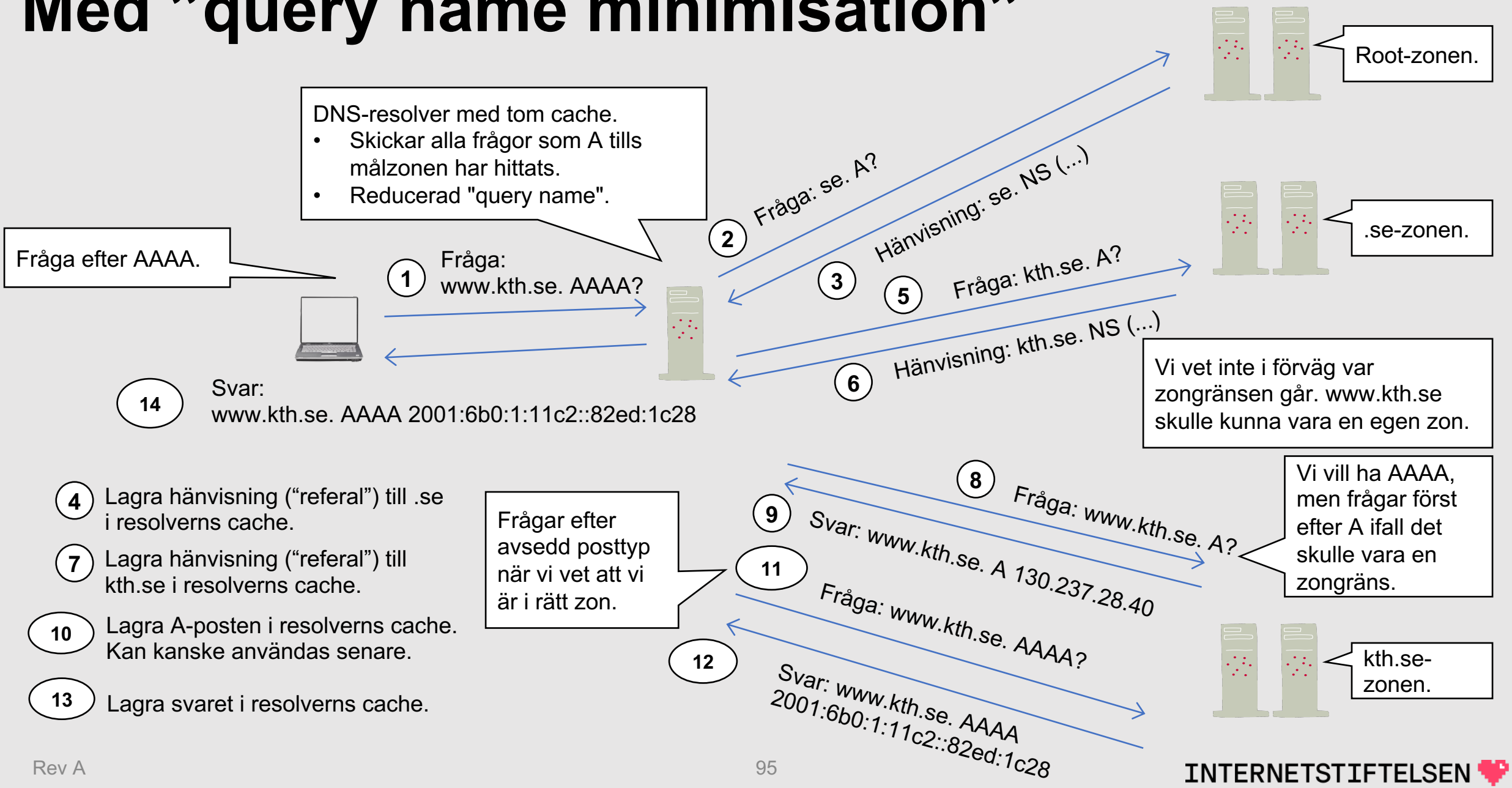
Hittills så har alla resolvrar betett sig som bilden visade och upprepat *hela* frågan överallt.

Nu finns det en alternativ metod som heter ”query name minimisation” (”QNAME minimisation”). Det krävs inga förändringar i DNS-standarden för att införa den. Metoden definieras i RFC 9156 .

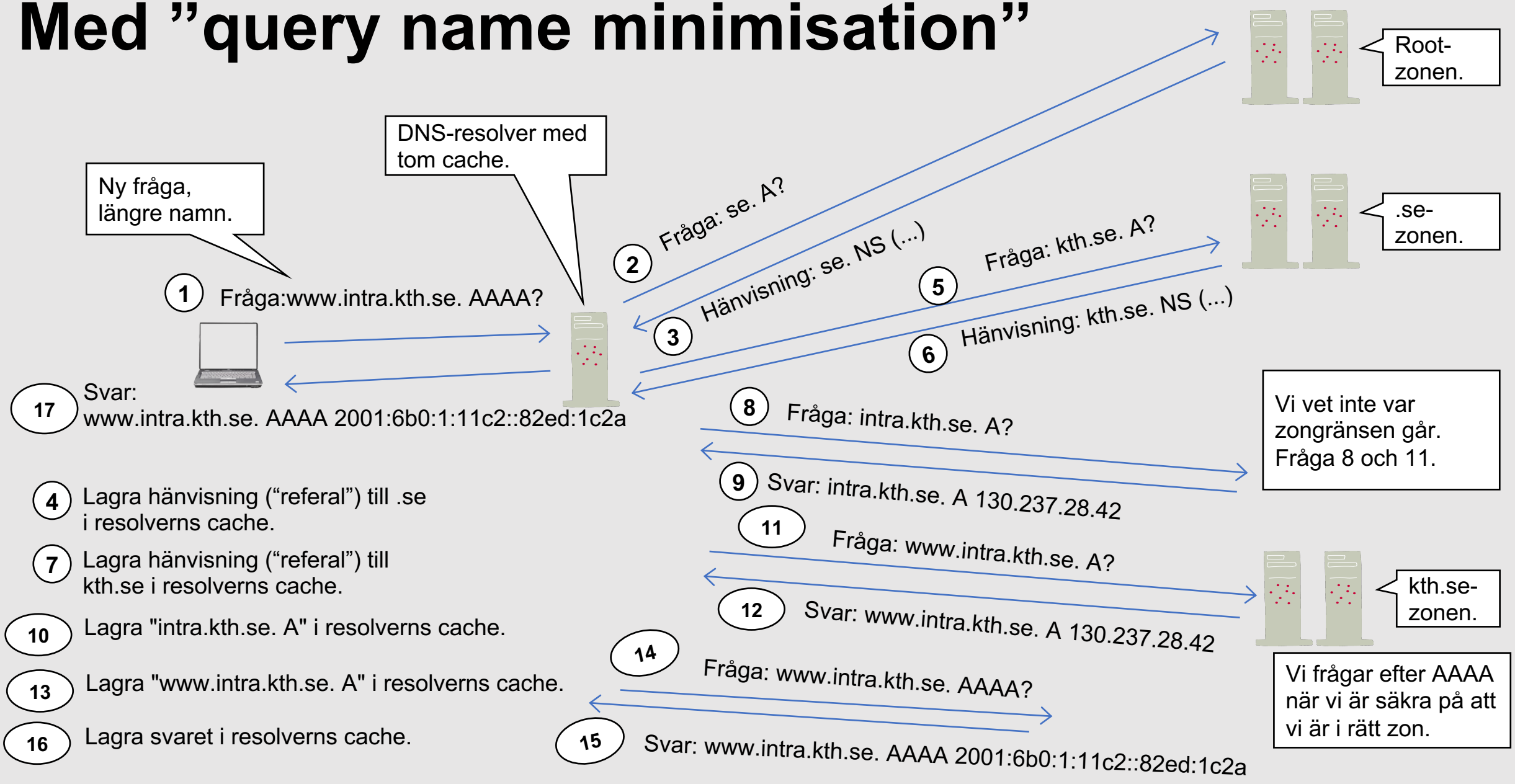
Query name minimisation

- Frågar inte efter hela namnet, utan börjar med en ***label*** mer än zonen man skickar frågan till.
  - Fråga bara efter TLD till rotzonen.
  - Fråga bara efter två nivåer till TLD.
- Fråga alltid efter samma posttyp, förslagsvis A, vilket gör att posttyp inte avslöjas.
- Fråga steg för steg för att hitta zongränserna och målzonen.
  - Lägg till en nivå i taget för att hitta delegering, om det finns någon.

Med ”query name minimisation”



Med ”query name minimisation”



Fler "queries"

Som framgick av bilderna ovan så blir det fler frågor. I vissa fall så blir det många fler frågor. Ökningen är oftast hanterlig.

Det finns mekanismer som tillåts enligt principen där man tar större steg vid domännamn med många ”*labels*”

Query name minimisation

Allt är dock inte frid och fröjd.

Det finns flera implementationer som inte helt har följt DNS-standarden, men som ändå fungerar beroende på det sätt som resolvningen alltid har gjorts. Resultatet blir att viss DNS-data blir oåtkomlig med "Query name minimisation".

Query name minimisation

Bind har infört en modifierad version av ”query name minimisation” för att komma runt en del av de felaktiga implementationerna som finns.

Bind har slagit på ”query name minimisation” som standard, men med sin modifiering.

Det är alltså dags att rita om hur resolvning går till. Detta kan bli standard för hur resolvning går till i framtiden.

▶Om presentationen

[Till Innehåll]

Internets domännamnssystem

Denna presentation är framtagen 2019–2024 av Mats Dufberg (mats.dufberg@internetstiftelsen.se) på Internetstiftelsen (https://internetstiftelsen.se/). Den är en del av undervisningsmaterialet för kursen ”Internets domännamnssystem” vid Kungliga tekniska högskolan, KTH (kurskod HI1037) resp. Karlstads universitet, KAU (kurskod DVGC28).

Licens

Dokumenthistorik

- Rev A: Ursprunglig version VT 2024

Slut.