IP-nivån

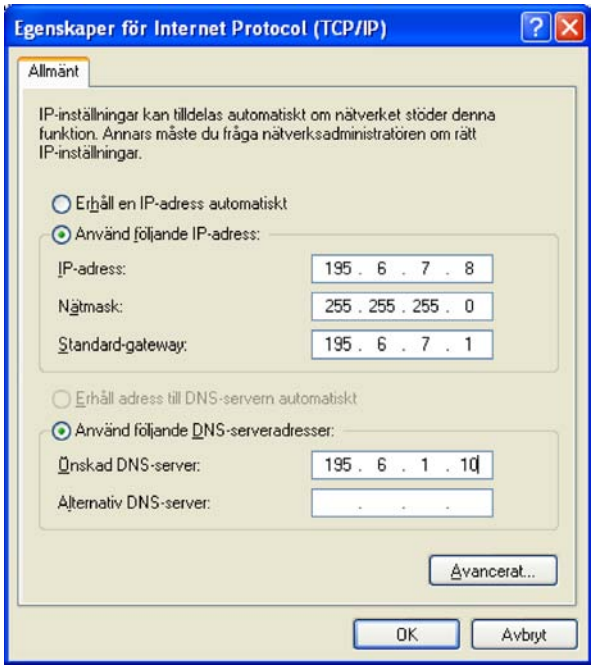
- Här beskrivs hur IP fungerar med statiska och dynamiska adresser (DHCP). Kapitlet behandlar grunderna för routing och hur IP-headern är uppbyggd. Subnätmaskens funktion, utseende och hur den används för att räkna ut nätnummer behandlas.
- Protokollet ICMP behandlas översiktligt.
- Kapitlet är tänkt att kunna läsas fristående.

För att din dator ska fungera på IP-nivån så behövs tre saker konfigureras:

- en IP-adress
- en standard gateway (default gateway)
- en subnätmask.

IP-adressen är datorns unika identitet på Internet, den som gör att andra hittar till din dator. Standard gateway talar om för IP-nivån vart den ska skicka paket den inte hittar till, tekniskt sett ”adresser som inte har en annan känd route”. Subnätmasken används för att ange hur stort nätet är. Men mer om allt detta nedan.

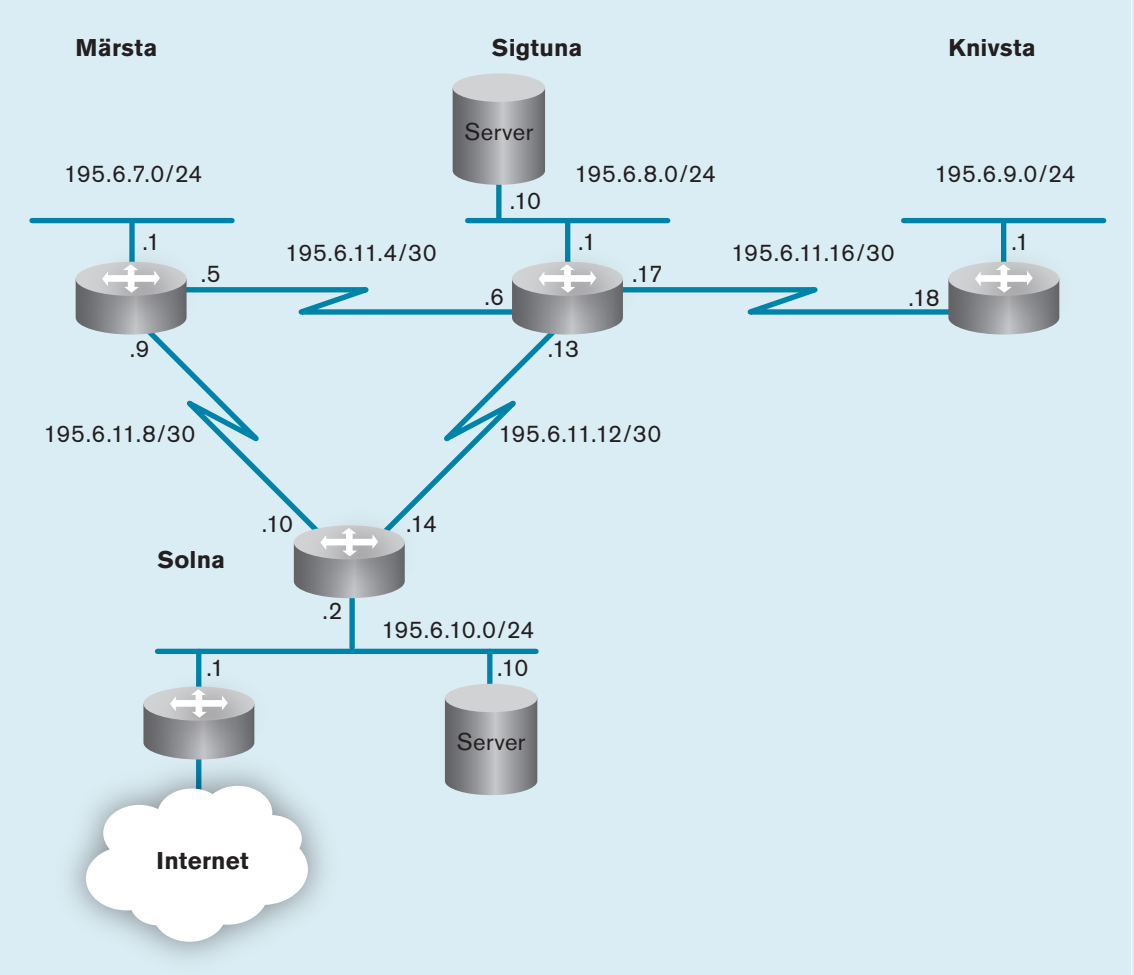
I Windows XP visas dessa tre inställningar ihop. För att du ska komma igång med del flesta applikationer krävs även IP-adressen till en namnserver (DNS står för Domain Name System), understa delen av inställningarna. IP-nivån fungerar dock utmärkt även utan namnserver.



IP-inställningar i Windows XP.

Eftersom IP-adressen ska vara unik kan inte var och en hitta på sin egen IP-adress utan de måste delas ut centralt. Idag delas en del adresser, genom att så kallade privata adresser används tillsammans med adressöversättning. En privat adress fungerar dock inte på Internet utan måste översättas till en publik adress. Adressöversättning hanteras i ett eget kapitel. Du kan erhålla IP-konfigurationen från en central server via protokollet DHCP (Dy-

namic Host Configuration Protocol), eller manuellt från IT-avdelningen eller din operatör. På Internet och i stora organisationer finns det en funktion för detta som kallas LIR (Local Internet Registry). Använder organisationen publika adresser så har man förbundit sig till att hålla reda på hur de används. Det finns inga formkrav på dokumentationen men vanligast är en IP-plan.



Exempel på en IP-plan.

I IP-planer skrivs oftast IP-adresser i kortform. Den router som i Solna sitter mot Internet har IP-adressen 195.6.10.1. Men eftersom subnätmasken och de tre första byten är gemensamma för hela nätet skrivs de på ett ställe.

Ur IP-planen ovan kan man läsa ut att en fungerande IP-adress i Knivsta skulle vara till exempel 195.6.9.25, subnätmask 255.255.255.0 och standard gateway 195.6.9.1. Standard gateway är IP-adressen till en router som leder ut till Internet. I Knivsta kommer vi bara ut på Internet via en router, som i sin tur går via Sigtuna och sedan Solna. Av historiska skäl kallas router i vis-

sa sammanhang för gateway, detta är alltså den router man ska använda standardmässigt om man inte har en bättre idé om hur man ska nå den adress man försöker skicka data till. Även begreppet default route är vanligt.

Att subnätmasken är 255.255.255.0 kan man utläsa ur beteckningen /24 (uttalas slash 24). Ofta skriver man inte enstaka noders adresser i en IP-plan utan man noterar nätnummer som 195.6.9.0. Vi ska titta på hur IP-adresser, subnätmasker och nätnummer hänger ihop lite längre fram.

Begreppet gateway har historiskt fått flera betydelser inom datakommunikation. I vissa sammanhang kan gateway och router användas som synonymer.

Var kommer IP-adresserna ifrån?

IP-adresserna kan anges manuellt under /Inställningar/Nätverksanslutningar/LAN/Internet Protocol (TCP/IP) eller liknande. Du kan kontrollera inställningarna med kommandot ”ipconfig/all” i Windows. På Mac eller Linux skriver du ifconfig. Resultatet ska bli liknande:

```
Ethernet-kort Local Area Connection:

        Anslutningsspecifika DNS-suffix . . :
        Beskrivning . . . . . . . . . . . . : Intel(R) LAN 2435
        Fysisk adress . . . . . . . . . . . : 00-13-CE-26-F9-18
        DHCP aktiverat  . . . . . . . . . . : Ja
        Autokonfiguration aktiverat  . . . . : Ja
        IP-adress . . . . . . . . . . . . . : 192.168.1.2
        Nätmask . . . . . . . . . . . . . . : 255.255.255.0
        Standard-gateway  . . . . . . . . . : 192.168.1.1
        DHCP-server . . . . . . . . . . . . : 192.168.1.1
        DNS-servrar . . . . . . . . . . . . : 192.168.1.1
        Lånet erhölls . . . . . . . . . . . : den 2 maj 2007 16:41:41
        Lånet upphör  . . . . . . . . . . . : den 3 maj 2007 04:41:41
```

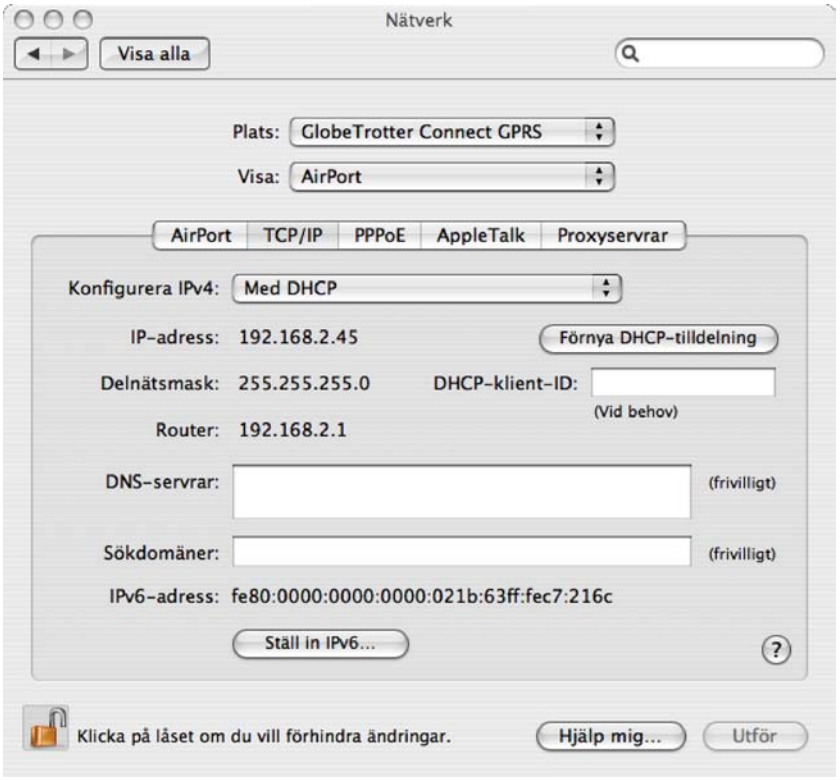
IP-adresser hämtade från en server.

I exemplet ovan har IP-adresserna hämtas från en server, de rader som har med DHCP att göra har markerats med fetstil. För att IP ska fungera behöver IP-parametrarna anges korrekt och det finns mycket att vinna på att centralisera denna funktion i en DHCP-server. Dessutom flyttar många noder runt mellan olika nätverk. Med statiska IP-adresser får man gå in och ändra dem hela tiden, och för inte så många år sedan ledde ändrade IP-adresser alltid till en omstart. Med DHCP blir detta mycket lättare, noderna frågar efter IP-inställningar för varje nät. Det har också blivit vanligt att

moderna operativsystem kontrollerar DHCP-lånet så fort länknivån ändras, det vill säga så fort man drar ut en nätverkskabel eller byter anslutning via WLAN.

Dynamisk utdelning av IP-adresser

Den vanligaste metoden för dynamisk utdelning av adresser är DHCP (RFC 1541 och 2131 med flera). DHCP är en funktionell förbättring av det äldre protokollet Bootp. DHCP använder också samma portnummer som Bootp, se protokollanalysen på nästa sida. Bootp var ett enkelt protokoll där en server huvudsakligen höll reda på vilken IP-adress som skulle delas ut till vilken MAC-adress. Sådana statiska mappningar kan även DHCP erbjuda men DHCP erbjuder flera förbättringar.

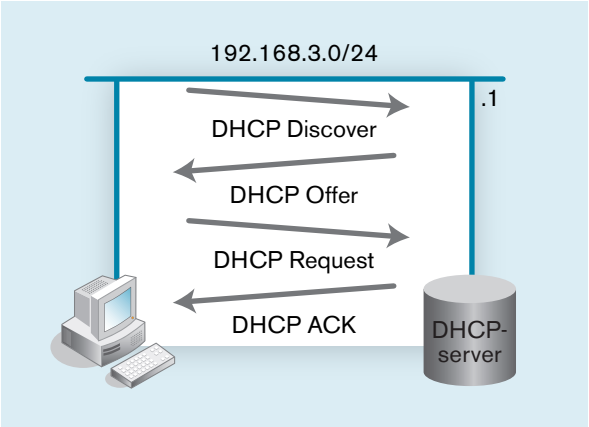


Inställningar för DHCP i MAC OS X.

Med DHCP kan flera noder dela på ett fåtal adresser. På servern lägger vi bara in hur stor adressrymd som ska delas ut. Servern håller sedan reda på vilken nod som fått dem. Då lånet av IP-adress är tidsbegränsat kan IP-adressen sedan erbjudas en ny nod. Vi kan använda flera DHCP-servers för att öka driftsäkerheten. Klienterna börjar med att fråga efter DHCP-servers (DHCP Dis-

cover) och får då ett erbjudande (DHCP Offer). Kommer det flera erbjudanden så gör det inget, klienten tar det första och bekräftar det till servern (DHCP request) varpå servern bekräftar (DHCP ack).

Man kan starta denna handskakning genom att ge kommandot ”ipconfig /renew” i Windows. En protokollanalys inspelad med Wireshark eller liknande program ger följande resultat (viss information utelämnad vid markeringar ”…”). Det första paketet skickas som en broadcast på både IP och Ethernet-nivå.



Utväxling av IP-information via DHCP.

```
Paket 1
Ethernet II, Src: Fujitsu_c3:2f:98 Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Internet Protocol, Src: 0.0.0.0, Dst: 255.255.255.255
User Datagram Protocol, Src Port: bootpc (68), Dst Port: bootps (67)
Bootstrap Protocol
    …
    Bootp flags: 0x0000 (Unicast)
    Client IP address: 0.0.0.0
    Your (client) IP address: 0.0.0.0
    Next server IP address: 0.0.0.0
    Relay agent IP address: 0.0.0.0
    Client MAC address: Fujitsu_c3:2f:98 (00:0b:5d:c3:2f:98)
    Server host name not given
    Boot file name not given
    Magic cookie: (OK)
    Option: (t=53,l=1) DHCP Message Type = DHCP Discover
    Option: (t=116,l=1) DHCP Auto-Configuration
    Option: (t=61,l=7) Client identifier
    Option: (t=50,l=4) Requested IP Address = 10.36.21.159
    Option: (t=12,l=14) Host Name = “lifebook-p7010”
    Option: (t=60,l=8) Vendor class identifier = “MSFT 5.0”
    Option: (t=55,l=11) Parameter Request List
    End Option
```

Del ur en protokollanalys. Här visas Paket 1 – DHCP Discover.

Lägg märke till att klienten egentligen vill ha en annan adress, 10.36.21.159, än vad den erbjuds senare. Klienten skickar över sin MAC-adress till servern, som kan kontrollera eventuellt befintligt lån.

I erbjudandet från servern kommer nu en ny IP-adress tillsam-

```
Paket 2
Ethernet II, Src: 3comEuro_9f:4a:fa , Dst: Fujitsu_c3:2f:98
Internet Protocol, Src: 192.168.3.1, Dst: 192.168.3.245
User Datagram Protocol, Src Port: bootps (67), Dst Port: bootpc (68)
Bootstrap Protocol
   Message type: Boot Reply (2)
   …
   Client IP address: 0.0.0.0
   Your (client) IP address: 192.168.3.245
   Next server IP address: 0.0.0.0
   Relay agent IP address: 0.0.0.0
   Client MAC address: Fujitsu_c3:2f:98 (00:0b:5d:c3:2f:98)
   Server host name: \377
   Boot file name: \377
   Magic cookie: (OK)
   Option: (t=53,l=1) DHCP Message Type = DHCP Offer
   Option: (t=54,l=4) Server Identifier = 192.168.3.1
   Option: (t=1,l=4) Subnet Mask = 255.255.255.0
   Option: (t=51,l=4) IP Address Lease Time = 12 hours
   Option: (t=52,l=1) Option Overload = Boot file and server host names
       hold options
   Option: (t=3,l=4) Router = 192.168.3.1
   Option: (t=6,l=4) Domain Name Server = 192.168.3.1
   End Option
   Padding
```

Från servern har en ny IP-adress erbjudits. Klienten får behålla den i 12 timmar.

mans med övrig konfiguration (nätmask, standard gateway och DNS). Det framgår även att klienten får låna IP-adressen i tolv timmar. Klienten skickar en begäran om denna IP-adress till severn. Detta är fortfarande en broadcast och lägg märke till att klienten fortfarande skickar från IP-adress 0.0.0.0. Paket fyra är sedan servern kvittens där all IP-konfiguration överförs.

```
Paket 3
Ethernet II, Src: Fujitsu_c3:2f:98, Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Internet Protocol, Src: 0.0.0.0, Dst: 255.255.255.255
User Datagram Protocol, Src Port: bootpc (68), Dst Port: bootps (67)
Bootstrap Protocol
   …
   Client IP address: 0.0.0.0
   Your (client) IP address: 0.0.0.0
```

```
   Next server IP address: 0.0.0.0
   Relay agent IP address: 0.0.0.0
   Client MAC address: Fujitsu_c3:2f:98 (00:0b:5d:c3:2f:98)
   Server host name not given
   Boot file name not given
   Magic cookie: (OK)
   Option: (t=53,l=1) DHCP Message Type = DHCP Request
   Option: (t=61,l=7) Client identifier
   Option: (t=50,l=4) Requested IP Address = 192.168.3.245
   Option: (t=54,l=4) Server Identifier = 192.168.3.1
   Option: (t=12,l=14) Host Name = "lifebook-p7010"
   Option: (t=81,l=18) Client Fully Qualified Domain Name
   Option: (t=60,l=8) Vendor class identifier = "MSFT 5.0"
   Option: (t=55,l=11) Parameter Request List
   End Option
```

Paket 3 – Klienten meddelar att den accepterar erbjudandet via DHCP Request.

```
Paket 4
Ethernet II, Src: 3comEuro_9f:4a:fa, Dst: Fujitsu_c3:2f:98
Internet Protocol, Src: 192.168.3.1, Dst: 192.168.3.245
User Datagram Protocol, Src Port: bootps (67), Dst Port: bootpc (68)
Bootstrap Protocol
   Message type: Boot Reply (2)
   …
   Client IP address: 0.0.0.0
   Your (client) IP address: 192.168.3.245
   Next server IP address: 0.0.0.0
   Relay agent IP address: 0.0.0.0
   Client MAC address: Fujitsu_c3:2f:98 (00:0b:5d:c3:2f:98)
   Server host name: \377
   Boot file name: \377
   Magic cookie: (OK)
   Option: (t=53,l=1) DHCP Message Type = DHCP ACK
   Option: (t=54,l=4) Server Identifier = 192.168.3.1
   Option: (t=1,l=4) Subnet Mask = 255.255.255.0
   Option: (t=51,l=4) IP Address Lease Time = 12 hours
   Option: (t=52,l=1) Option Overload = Boot file and server host names
       hold options
   Option: (t=3,l=4) Router = 192.168.3.1
   Option: (t=6,l=4) Domain Name Server = 192.168.3.2
   End Option
   Padding
```

Paket 4 – Serverns kvittens att IP-konfigurationen överförts.

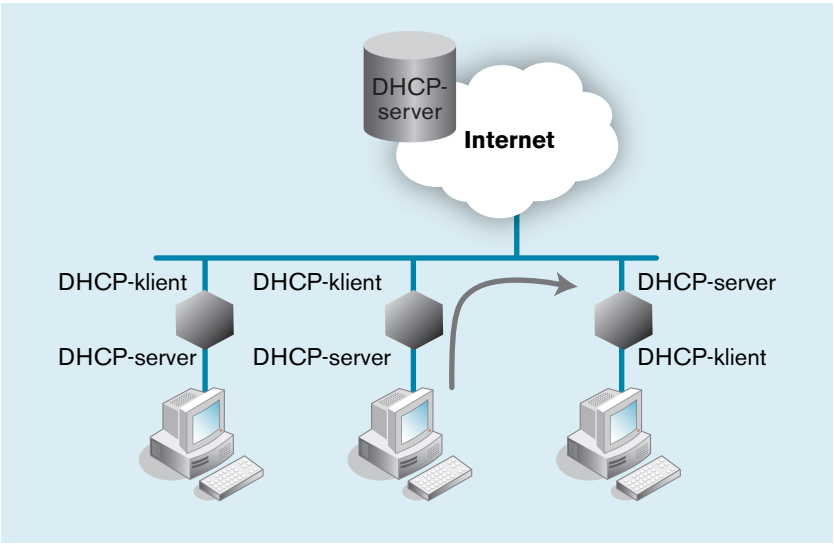
Hur länge klienten erbjuds att låna IP-adressen är en viktig parameter. Efter halva den tiden kommer klienten försöka att förnya lånet. Detta för att hela tiden försäkra sig om att ha en fungerande IP-adress. Om detta inte lyckas kommer klienten att göra en helt ny förfrågan (DHCP Discover) efter 87 procent av tiden.

Hur långa lånetider man ska ha varierar. I ett företagsnät har man ofta lånetider på en vecka, antalet maskiner som flyttar sig är inte så stort. På ett publikt trådlöst nätverk där användare kommer och går kan man ha lånetider på fem minuter. I ett operatörsnät låter man kunderna typiskt ha lånetider mellan 20 och 60 minuter. På mer avancerad nätverksutrustning är därför lånetiden alltid ställbar.

DHCP designades i början av 1990-talet och är idag en självklar del av modern nätverksteknik. Microsoft var snabba med använda tekniken, kanske lite för snabba men idag fungerar DHCP väldigt stabilt. Även i Unix/Linux finns fullgott stöd för DHCP både som server och klient. Själv har jag dock haft en del problem med distribution av DNS-information via DHCP till Linux-klienter.

DHCP gör IP-kommunikation lättare att komma igång med men också osäkrare.

Det är ofta svårt att kombinera säkerhet med användbarhet. Detta gäller även DHCP. DHCP är gjort för att inte krångla eller konstra. Varken klienten eller servern har i grundutförandet någon slags autentisering. Hur vet vi att vi pratar med rätt eller ens en godkänd server? En möjlig attack från en förövare är se till att svara på DHCP Discover från en nod och sedan ange sig själv som standard gateway. På så vis kan man göra kopior på all information till och från klienten. Det finns även andra möjligheter att ange fel subnätmask och på så sätt öppna nätet. Attacken bygger



En användare kan felaktigt koppla in en DHCP-server till Internet.

i princip på att förövaren sitter på samma lokala nätverk som den som attackeras. DHCP är tänkt att fungera lokalt för att minska riskerna, och man bör ha god kontroll ifall man ska låta det passera routrar.

En annan aspekt på problem med DHCP är välkänd hos operatörer. Användare kopplar in sig via små gateways som använder DHCP-klienter mot operatörer (WAN-anslutningen) och som sedan ska agera DHCP-server mot det egna nätet.

Någon användare vänder sin enhet fel och besvarar effektivt andra användares DHCP-frågor med sina egna normalt privata adresser. Det fungerar inte bara dåligt för den felvända enheten utan påverkar även andra. Utan onda avsikter får vi en effektiv Denial-Of-Service attack. Protokollet DHCP kan inte skydda mot detta utan det får operatören bygga funktioner för.

Fyra miljarder adresser

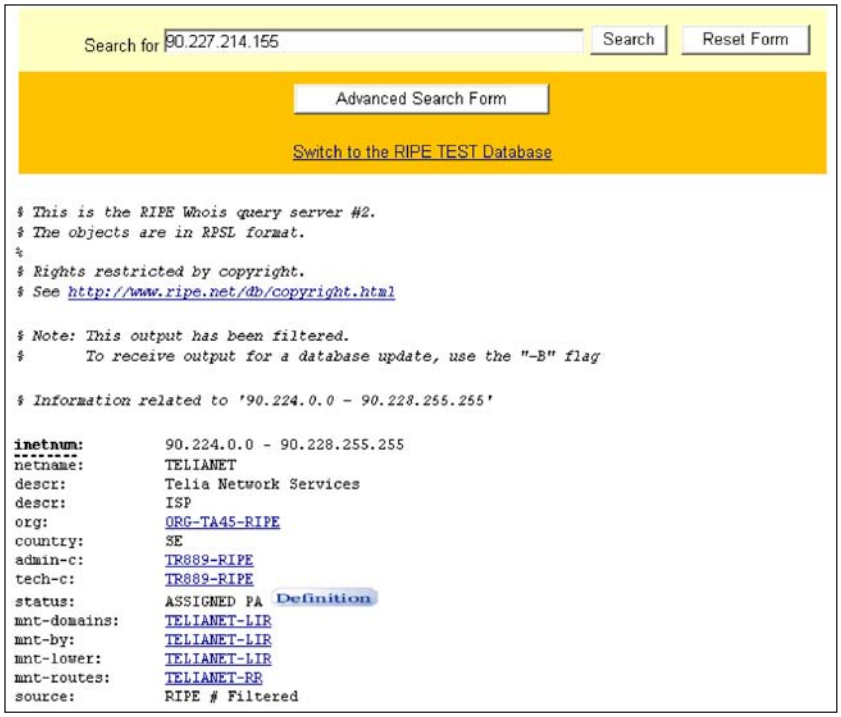
En IP-adress består av 32 bitar eller fyra byte. Det är en konvention att skriva dem på så kallad decimal punktnotation, där vi sätter in en punkt mellan varje byte. Detta gör att vi inte behöver skriva ut inledande nollor i varje byte.

195.006.007.010 kan skrivas som 195.6.7.10

Detta noteringssätt är helt förhärskande men det förekommer att IP-adresser skrivs i hexadecimalt format som C306070A.

32 bitar gör att vi har 2^{32} adresser till förfogande vilket motsvarar drygt fyra miljarder. I slutet av 1970-talet var det framsynt att ta till med ett så högt tal, på den tiden fanns inga persondatorer och lokala nätverk hade knappt sett dagens ljus. Idag kan vi beklaga att man inte tog till lite mer och att man inte begränsade utdelningen av IP-adresser initialt. Var man bara tidig var det lätt att få stora adresstilldelningar. Dessutom är IP-adresser som börjar med 224–255 reserverade för bland annat multicast, så i praktiken har vi under fyra miljarder IP-adresser som kan användas.

För att IP-adresserna ska vara unika behöver de administreras centralt. Detta görs av IANA (Internet Assigned Numbers Authority) som drivs av ICANN (Internet Corporation for Assigned Names and Numbers). Ansvaret är sedan delegerat till tre organisationer och i Europa är det RIPE som hanterar detta (Réseaux IP Européens se www.ripe.net). RIPE delar sedan ut block med IP-adresser till europeiska operatörer. På RIPE:s hemsida kan man söka upp vem som administrerar en europeisk IP-adress. Oftast sköter operatörer i denna administration.


Search for 90.227.214.155 Search Reset Form

Advanced Search Form

Switch to the RIPE TEST Database

```
% This is the RIPE Whois query server #2.
% The objects are in RPSL format.
%
% Rights restricted by copyright.
% See http://www.ripe.net/db/copyright.html

% Note: This output has been filtered.
%       To receive output for a database update, use the "-B" flag

% Information related to '90.224.0.0 - 90.228.255.255'

inetnum:        90.224.0.0 - 90.228.255.255
netname:        TELIANET
descr:          Telia Network Services
descr:          ISP
org:            ORG-TA45-RIPE
country:        SE
admin-c:        TR889-RIPE
tech-c:         TR889-RIPE
status:         ASSIGNED PA  Definition
mnt-domains:    TELIANET-LIR
mnt-by:         TELIANET-LIR
mnt-lower:      TELIANET-LIR
mnt-routes:     TELIANET-RR
source:         RIPE # Filtered
```

På www.ripe.net kan man kontrollera vem som registrerat en IP-adress. Databasen innehåller även kontaktinformation lite längre ner.

En publik adress är alltså spårbar. Detta behövs för att man ska kunna kontakta operatören eller organisationen ifall IP-adressen är inblandad i något som strider mot god ”netiquette” eller ifall det finns problem med till exempel e-post eller hantering av domäner. Netiquette, eller på svenska nätvett, är ett brett begrepp innefattande allt från goda råd hur man bör bete sig på nätet för att underlätta samvaron med de övriga användarna, till hur man ska undvika rena olagligheter.

RFC 1855 är en god start för god netiquette.

Om du tittar igenom avtalet med din Internetoperatör finns det ofta klausuler om att operatören förbehåller sig rätten att stänga av din anslutning ifall ditt användande strider mot god netiquette. Detta trots att begreppet är ytterst fritt för tolkningar. Men syftet men att använda netiquette är gott. På Internet ska det finnas plats för olika åsikter men vi behöver formulera hur vi ska dela med oss av dem och hur vi ska bemöta dem som har andra åsikter.

Enkel routing

Varje IP-nod har en enkel routingtabell. Du kan få fram den genom kommandot netstat -rn eller ”route print” i Windows. I Unix, Linux och Mac OS X används kommandot netstat -r.

```
===========================================================================
Aktiva vägar:
   Nätverksadress          Nätmask   Gateway-adress      Gränssnitt    Mått
          0.0.0.0          0.0.0.0     192.168.3.1   192.168.3.174       25
        127.0.0.0        255.0.0.0       127.0.0.1       127.0.0.1        1
      192.168.3.0    255.255.255.0   192.168.3.174   192.168.3.174       25
    192.168.3.174  255.255.255.255       127.0.0.1       127.0.0.1       25
    192.168.3.255  255.255.255.255   192.168.3.174   192.168.3.174       25
        224.0.0.0        240.0.0.0   192.168.3.174   192.168.3.174       25
  255.255.255.255  255.255.255.255   192.168.3.174   192.168.3.174        1
Standard-gateway:      192.168.3.1
===========================================================================
```

Routingtabell. Först kontrolleras om mottagaren sitter på samma nät som vi, i så fall behöver vi inte blanda in routrar och operatörer. Sedan kontrolleras om vi vet något om nätet eller just denna adress. I sista hand skickas paketet till en router som förhoppningsvis vet mer.

För att förstå tabellen behöver vi förstå subnätmaskens roll. Och vi behöver titta på hur grundfunktionen för routing fungerar:

1. Är mottagaren direkt adresserbar?
2. Är mottagaren en känd nod?
3. Tillhör mottagaren ett känt nät?
4. Använd ”default route”

Routingprocessen består av fyra huvudsakliga steg.

Routing-processen testar i denna ordning tills den hittar ett giltigt villkor. Och om den inte hittar något villkor som är applicerbart så kastas paketet och protokollet ICMP får se till så att ett felmeddelande skickas. Ordningen är i sig rätt självklar. Först kontrollerar vi om mottagaren sitter på samma nät som vi, i så fall behöver vi inte blanda in routrar och operatörer. Sedan kontrollerar vi om vi vet något om nätet eller just denna adress. I sista hand får vi skicka paketet till en router som förhoppningsvis vet mer.

Routrarna arbetar på samma sätt. Antingen sitter de på samma nät som mottagaren och då handlar det om LAN-teknik hur paketet ska vidarebefordras. Annars undersöker routern om den vet något om just denna nod eller detta nät. Annars skickas paketet till routerns standardlösning: default route. På så sätt routas paketen högre upp i hierarkin. Till slut kommer paketet hamna på en knutpunkt där anslutna routrar känner till alla nätnummer på Internet. Paketet leds då till en väg där routrarna känner till detta specifika nät. De här Internetknutpunkterna kallas idag för IXP, Internet eXchange Points. I Sverige finns cirka tio stycken.

Netnod (www.netnod.se) sköter driften på fem stycken IXP:s i Sverige. På www.ep.net finns en lista över aktuella IXP:s.

För att kunna avgöra om en mottagare är direkt adresserbar behöver noder och routrar förstå hur stort nät de hanterar. Detta är

subnätmaskens uppgift och vi kan uppfatta nätets storlek olika. Många routrar i världen arbetar med nätet 17.0.0.0/8. Detta är ett stort nät om drygt 16 miljoner noder. Men internt på Apple kan de arbeta med 17.0.0.0/24 som bara avser drygt 250 adresser. Det här är ett fundament inom IP. Adressrymder kan aggregeras till större nät. Detta gör routingtabellerna mindre och routingprocessen effektivare.

Första steget, direkt adresserbar, avgör om paketet behöver adresseras till ett lokalt nät. Med hjälp av subnätmasken /24 i detta fall så kan noden räkna ut att nätets storlek är 256 adresser.

Default route (default gateway är samma sak men en vanlig beteckning för en enstaka nod) talar som sagt om vart enstaka paket ska skickas. Default route betecknas med nätnumret 0.0.0.0 och mask 0.0.0.0.

I routingtabellen på föregående sida finns nätet 127.0.0.0. Detta nät med nodadressen 127.0.0.1 används för felsökning. 127.0.0.1 kallas för localhost och gör det bland annat enkelt att felsöka. Vill man undersöka hur IP fungerar så startar man gärna med localhost. Vid felsökning kan man gärna starta med att pinga localhost. Adressen används även i accesslistor och brandväggar. Om man startar en servertjänst är det inte ovanligt att den bara svarar på localhost innan man uttryckligen konfigurerat programmet eller brandväggen att svara på andra adresser.

Vad gör subnätmasken?

Subnätmasken består av 32 bitar, som var och en kan ha värdet ett eller noll. På samma sätt som med IP-adresser skrivs varje byte decimalt med en särskiljande punkt. 255.255.255.0 motsvarar alltså 11111111 11111111 11111111 00000000. Ett annat sätt att notera subnätmasken är att notera antal ettor, i detta fall 24 stycken vilket noteras ”/24”. Denna kortform är vanlig i IP-planer. På samma sätt kan man lätt få fram att 255.255.0.0 kan noteras som ”/16”. Däremot kräver det räknande via binära tal för att se hur till exempel 255.255.255.248 kan skrivas som ”/29”. På nästa sida följer en tabell med bägge noteringssätten. Det finns andra möjliga värden på subnätmasken men detta är de absolut vanligaste.

Subnätmasken anger nätets storlek.

Subnätmasken anger nätets storlek. En nod som har subnätmasken 255.255.255.0 kan då räkna ut att den sitter på ett nät som har 254 noder direkt adresserbara. Om den istället har nätmasken 255.255.255.248 så är 6 noder direkt adresserbara.

Med hjälp av subnätmasken räknar noden ut vilket nätnummer den sitter på. Beräkningen går till så att IP-adressen multipliceras med subnätmasken bit för bit (programmeringsmässigt används

en logisk AND-operation: 0 AND 0 = 0, 0 AND 1 = 0, 1 AND 0 = 0, 1 AND 1 = 1). Att använda en AND-operation är ett vanligt sätt i programmering för att ta bort värden vi inte vill behandla. Vi kallar detta för att maska bort bitar och därav kommer namnet subnätmask.

En nod med IP-adressen 195.6.7.25 och subnätmask 255.255.255.0 utför följande beräkning:

IP-adress	195.006.007.025
Nätmask	255.255.255.000
Nätnummer	195.006.007.000

En nod med IP-adressen 195.6.7.25 och subnätmask 255.255.255.248 utför följande beräkning, nu med binär notering:

IP-adress	11000011 00000110 00000111 00011001
Nätmask	11111111 11111111 11111111 11110000
Nätnummer	11000011 00000110 00000111 00011000

Om vi omvandlar denna siffra till decimal punktnotation får vi 195.6.7.16. Noden kan alltså räkna ut nätnumret och dessutom räkna ut att nätets storlek är 14 noder, Första tillåtna värde blir 195.6.7.17 och det sista 195.6.7.30. Även IP-adressen 195.6.7.31 tillhör detta nät men detta värde används för IP-broadcast, det vill säga när alla noder ska adresseras. Totalt ingår IP-adresserna 195.6.7.16–195.6.7.31 i adressrymden men det lägsta numret används som nätnummer och nätnumret behövs i routingtabeller. Det högsta numret används för broadcast. Det är därför vi inte kan adressera 16 noder utan 14.

På varje nät är den högsta och lägsta adressen reserverade. Den lägsta används för att adressera nätet och den högsta används för IP-broadcast.

Subnätmasken används alltså för att räkna ut nätets storlek och det nätnummer som noden sitter på. Noderna 195.6.7.25 och 195.6.7.30 kommer att räkna ut att de sitter på samma nät. Och om vi tittar på hur routing fungerar är detta första villkoret, kontrollera om noden är direkt adresserbar, i så fall kan noderna utbyta data direkt. Om vi återvänder till IP-planen i början på detta kapitel kan alltså noderna i Sigtuna pratat direkt med sin server medan noderna i Märsta och Knivsta räknar ut att de måste gå via sin router.

En felaktig subnätmask gör att noden inte kan skicka paketen vidare på rätt sätt.

Felaktig subnätmask får till konsekvens att noderna misstolkar både nätnummer och nätstorlek och då fungerar inte routing. De kan då råka ut för att de försöker skicka paket till en dator som

Subnätmask	/n	Antal noder
255.0.0.0	/8	16 777 214
255.255.255.0	/16	65 534
255.255.128.0	/17	32 766
255.255.192.0	/18	16 382
255.255.224.0	/19	8 190
255.255.240.0	/20	4 094
255.255.248.0	/21	2 046
255.255.252.0	/22	1 022
255.255.254.0	/23	510
255.255.255.0	/24	254
255.255.255.128	/25	126
255.255.255.192	/26	62
255.255.255.224	/27	30
255.255.255.240	/28	14
255.255.255.248	/29	6
255.255.255.252	/30	2

inte sitter på samma nät eller att de inte når sin default gateway. Alla tre IP-parametrarna IP-adress, mask och default gateway måste alltså vara rätt.

I tidig IP-teknik hade subnätmasken bara värdena /8, /16 och /24. Men detta var en väldigt grov indelning som snabbt ledde till brist på IP-adresser. Själv arbetade jag under tidigt 1990-tal på ett företag med cirka 30 noder, men vi behövde på den tiden dela upp nätet. Därför erhöll vi två stycken publika nät med nätmask /24 vilket gav oss adresseringsmöjlighet upp till drygt 500 noder. Andra som var tidigt ute och lite större än oss fick ett helt /16-nät. När variabel subnätmask infördes runt mitten av 1990-talet så blev både användning och utdelning av publika adresser mycket effektivare. Det enda pris vi fått betala är att det blivit lite krångligare att räkna ut nätstorlek och nätnummer.

IP-headern

Om vi använder en nätverksanalysator för att spela in en IP-header så skulle vi normalt se 20 byte, till exempel sekvensen:

45000056b1ae40003506586b5883 1e68c0a8 03f5

Vi lägger in mellanslag för att öka tydligheten:

4500 0056 b1ae 4000 3506 586b 5883 1e68 c0a8 03f5

För att kunna tolka detta brukar man granska fyra byte i taget, och läsa dem som en rad. (Kom ihåg att en byte motsvarar åtta bitar eller 2 hexadecimala siffror.) Vi får då fem rader enligt nedan:

4	IHL	Diffserv	Total längd	
ID			Flaggor	Fragment offset
TTL		Protokoll	Checksumma	
IP avsändare (S-IP)				
IP mottagare (D-IP)				
Optioner				

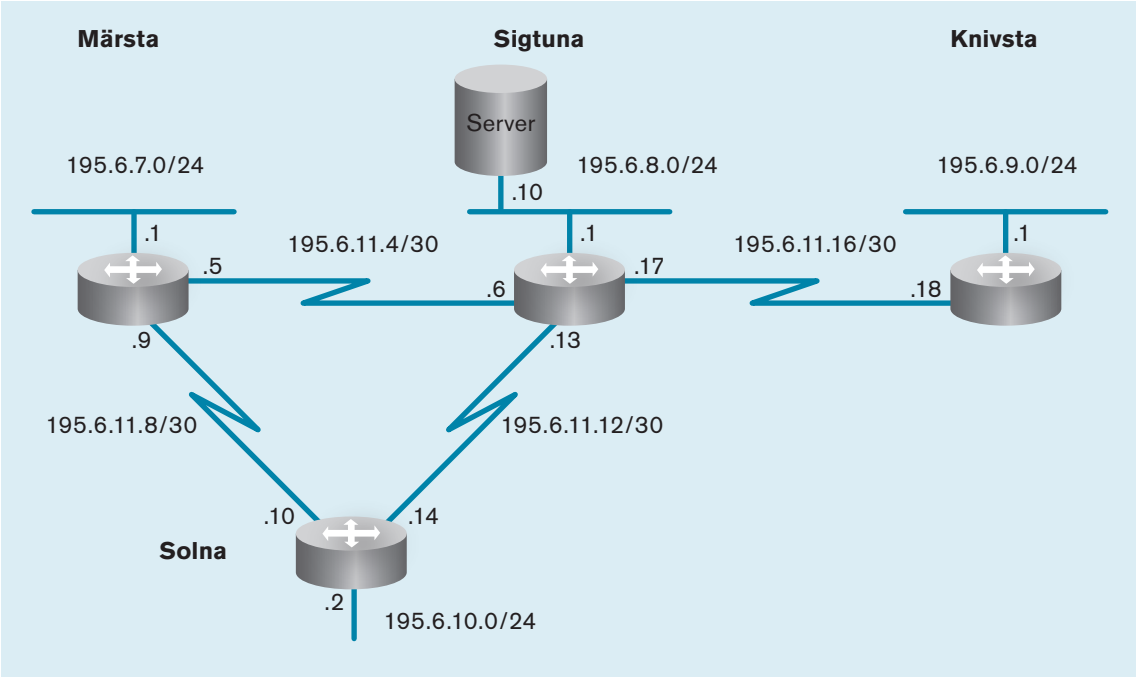
Figuren visar schematiskt hur man läser/analyserar en IP-header.

- Första siffran anger att detta är IP version 4.
- Nästa siffra (dvs 4 bitar) IHL står för IP Header Length och mäts i 32-bitars ord. Värdet 5 är vanligast och innebär att headern är 20 byte lång.
- Nästa byte är ofta just ”00”. Detta fält har historiskt tolkats som Type of Service, förkortat ToS. I modern IP tolkas detta fält enligt en standard som kallas Diffserv, en kort form av differentiated services. Fältet används av noder för att kunna skilja olika trafiktyper åt.

- Total längd består av 2 byte och anger paketets storlek inklusive header. Mäts i byte, 0x0056 motsvarar decimalt 86 byte. (Beteckningen 0x innebär att efterföljande format är hexadecimalt.)
- Varje paket förses med en unik identitet bestående av två byte. I detta fall 0xB1AE. IP-identiteten används bland annat vid fragmentering.
- Tre bitar är reserverade för flaggor. Den första används för närvarande inte. Den andra biten avser ”Don’t Fragment” den tredje avser ”More Fragments”. Dessa tre flaggor tillsammans med offsetvärdet bildar två byte. Värdet 4 (binärt 0100) innebär att flaggan Don’t Fragment är satt till ett. En router får alltså inte fragmentera detta paket.
- Fragment offset består av 13 bitar. Värdet är ofta noll som i exemplet ovan. Funktionen förklaras i nästa avsnitt.
- TTL, Time To Live motsvarar antal routerhopp paketet ska kunna passera. I detta fall 0x35=53 hopp innan paketet kastas. Totalt kan IP inte hantera ett Internet som skulle ha mer än 255 routerhopp, fältet är åtta bitar stort. På Internet idag ligger vi inte i närheten, ett typiskt värde idag är snarare 10–30 hopp innan paketet når sin mottagare. Av historiska skäl lever beteckningen med time kvar, i modern IP borde parametern kallas ”hops to live”.
- Protokollfältet med i detta fall innehåll 06 tolkas som att IP bär protokollet TCP.
- Checksumma 586b används för att kunna detektera överföringsfel eller fel i program. Checksumman i IP är av en enkel typ och detekterar bara fel i IP-headern, inte i nyttolasten. IP-nivån förutsätter inte att nivå två kontrollerar innehållet, därför kan överföringsfel uppstå och IP-nivån skulle då fatta routingbeslut på felaktiga paket. Behovet av denna funktion har minskat kraftigt sedan IP togs fram, moderna nivå två protokoll hanterar överföringsfel effektivare än IP.
- Funktionen för fältet IP-avsändare framgår av namnet. I den här guiden används ofta kortformen S-IP, IP-Source, men även beteckningen ”Src” förekommer. Det hexadecimala värdet 5883 1e68 motsvarar 88.131.30.104.
- IP mottagare, i den här guiden ofta betecknat som D-IP, D för destination. c0a8 03f5 motsvarar 192.168.3.245.

Fragmentering

I bästa fall kan vi helt undvika fragmentering. TCP ska ju dela upp dataflödet i lagom stora paket. Men en avsändare kan inte vara säker på hur paket hanteras av alla länkar som kan vara inblandade i en överföring mellan två noder. Dessutom kan ju förhållandena ändras. Därför kan en router få ett inkommande paket med en storlek som nästkommande länk inte kan hantera. Alltså behöver paketet delas upp – fragmenteras.

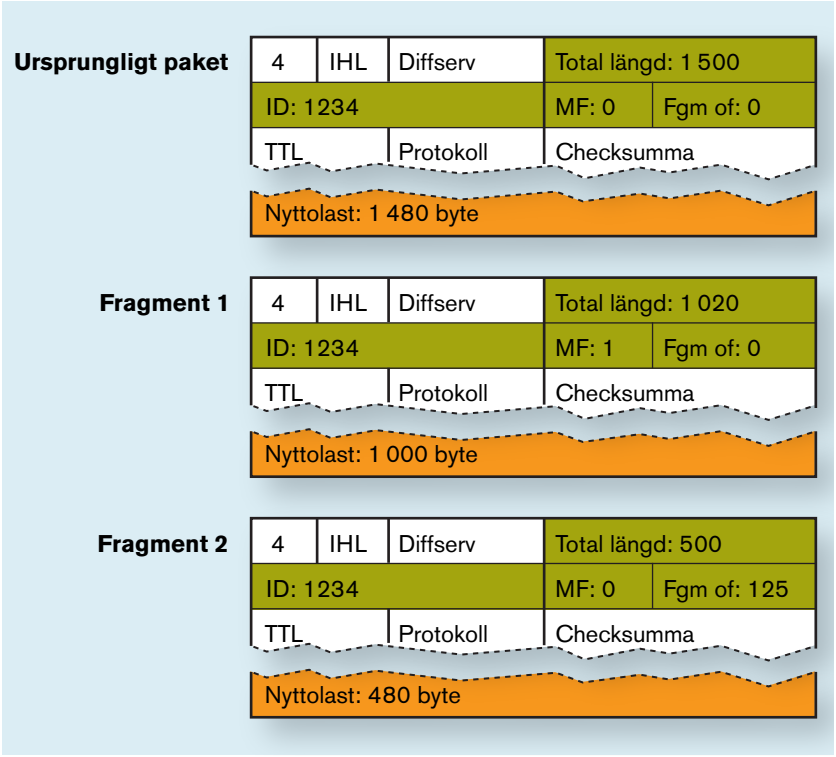


Om olika länkar kan hantera olika paketstorlekar finns stor risk för att routrar måste fragmentera.

Varje länk har ett maximal värde för hur stora paket den hanterar, värdet kallas för MTU Maximum Transmission Unit. För ett vanligt Ethernet är värdet 1 500 byte, men värden ned till 576 byte är tillåtna i IPv4.

Om vi tänker oss att servern i Sigtuna ska skicka paket till en nod på Märsta-nätet så kommer den försöka med paketlängden 1 500 byte. Den seriella länken mellan Sigtuna och Märsta har en MTU om 1 024. Routern har därför inget annat val än att dela upp paketet.

När routern fragmenterar får den göra det i sektioner om åtta byte, och längden på ett fragment mäts i ”enheter om åtta byte”. Ett giltigt värde skulle därför vara 1 000 byte nyttolast plus 20 byte header. Totallängden blir då 1 020 vilket inte överskrider MTU för länken. Routern behåller samma identitet som det ursprungliga paketet. Första fragmentet skickas iväg med flaggan



Ett paket kan delas upp av en router i fragment.

More Fragments = 1 samt Fragment Offset satt till 0. Nästa fragment är det sista så flaggan More Fragments sätts till 0. Fragment offset sätts till 1 000/8 = 125. Totallängden för varje paket sätts till nyttolast plus 20 byte header.

För mottagarens IP-stack, det vill säga slutstationen kommer sedan fragment tas emot med flaggan More Fragments satt till ett. Det vill säga mottagaren förstår att detta inte är den sista delen av ett paket så den får mellanlagra detta fragment. Sedan kommer fragment två, mottagaren ser på ID-värdet att dessa hör ihop. Flaggan More Fragments satt till noll gör att mottagaren kan avgöra att detta var det sista fragmentet.

Offsetvärdet tillsammans med flaggorna gör att pakten kan komma fram i godtycklig ordning. Om det andra fragmentet kommer först så skulle flaggan More Fragments vara satt till noll. Men Fragment Offset är satt till 125, mottagaren kan alltså avgöra att innehållet i detta fragment ska föregås av $8 \times 125 = 1\,000$ byte data.

Att undvika fragmentering

Fragmentering är något vi vill undvika. Det belastar routrar som helst ska ägna all kraft åt att vidareförmedla paket. Och vi har ju en

gång delat upp dataflödet i paket, varför inte göra rätt från början?

Fragmentering kombinerat med paketförluster är riktigt dåligt. Om ett fragment saknas kan mottagaren inte återskapa det ursprungliga datagrammet (paketet). Efter ett tag kommer TCP upptäcka att ett paket saknas och skicka om det men nu med ett annat ID, samtidigt som det ofullständiga paketet ligger och tar plats hos mottagaren. Alltså måste mottagaren regelbundet städa bort misslyckade refragmenteringar.

En avsändare kan hindra fragmentering genom att sätta flaggan Don't fragment till ett. En router som tar emot ett paket som behöver fragmenteras kommer då att skicka tillbaka ett felmeddelande. Avsändaren har då möjlighet att skicka om paketet men med minskad paketlängd.

Ofta kan man i operativsystem eller register sätta vilken MTU man vill använda. Om man misstänker att någon slags IP-tunnling kan förekomma kan det vara en god idé att ställa ner MTU till 1 480 eller 1 460 för att ge plats åt extra headers. (IP-tunnling innebär att paket förses med en ny IP-header, till exempel för att kunna routas på Internet med en publik adress.)

En sista möjlighet som vissa operativsystem använder är att använda det lokala nätets MTU då man adresserar noderna som sitter på samma nätverk medan man använder en mindre paketstorlek då paketen skickas via default gateway.

Kortfattat om ICMP

Ibland fungerar inte routing eller förmedling av paket på IP-nivån. Paket kommer inte fram eller en router hinner inte hantera paketet i tid. I sig är IP både förbindelselöst och baserat på den praktiska modellen går-det-så-går-det (best effort). Förbindelselös kommunikation innebär att paketet skickas utan att vi kontrollerat om mottagaren är redo att ta emot paket, eller att mottagaren över huvud taget existerar. IP behöver i princip inte ta hänsyn till de problem som kan uppstå. Problem som uppstår ska istället hanteras av protokollet ICMP, Internet Control Message Protocol.

I praktiken ska vi se ICMP som en del av IP. IP beskrivs i RFC 791 och ICMP i nummer 792. De har utvecklats för att vara beroende av varandra. När problem uppstår med att skicka ett paket vidare på IP-nivå så skickas normalt ett ICMP-paket tillbaka till den nod som står som avsändare på det paket som förorsakade problemet. Problemen kan till exempel vara:

- Vi hittar ingen bra väg till mottagaren, varken som nod eller nät ("no route to host" eller "destination unavailable").

- Paket har passerat för många routrar, vanligtvis beror detta på fel i routingtabeller så att paketet skickas runt i en loop. (Felet som uppstår kallas time exceed.)
- Parameter problem.

Det är även mekanismer i ICMP som används med felsökningskommandona traceroute och ping.

Referenser

RFC 791 Internet Protocol
RFC 792 Internet Control Message Protocol

I äldre dokument används bara klasser och begreppet subnät förekommer inte. Begreppet introduceras i:

RFC 917 Internet Subents
RFC 940 Toward an Internet Standard Scheme for Subnetting

RFC 1541 DHCP
RFC 2131 DHCP (en uppdatering av RFC 1541)

Protokollet DHCP använder samma portnummer som BootP och hanteras av routrar som BootP. För att läsa om hur routrar hanterar DHCP (en viktig fråga angående till exempel hur DHCP Discover ska skickas vidare) måste man alltså läsa äldre dokument om hur BootP hanteras.

RFC 1519 Classless Inter-Domain Routing (CIDR):
an Address Assignment and Aggregation Strategy