Internetprotokollen

Maria Kihl

LUND
UNIVERSITY

Läsanvisningar

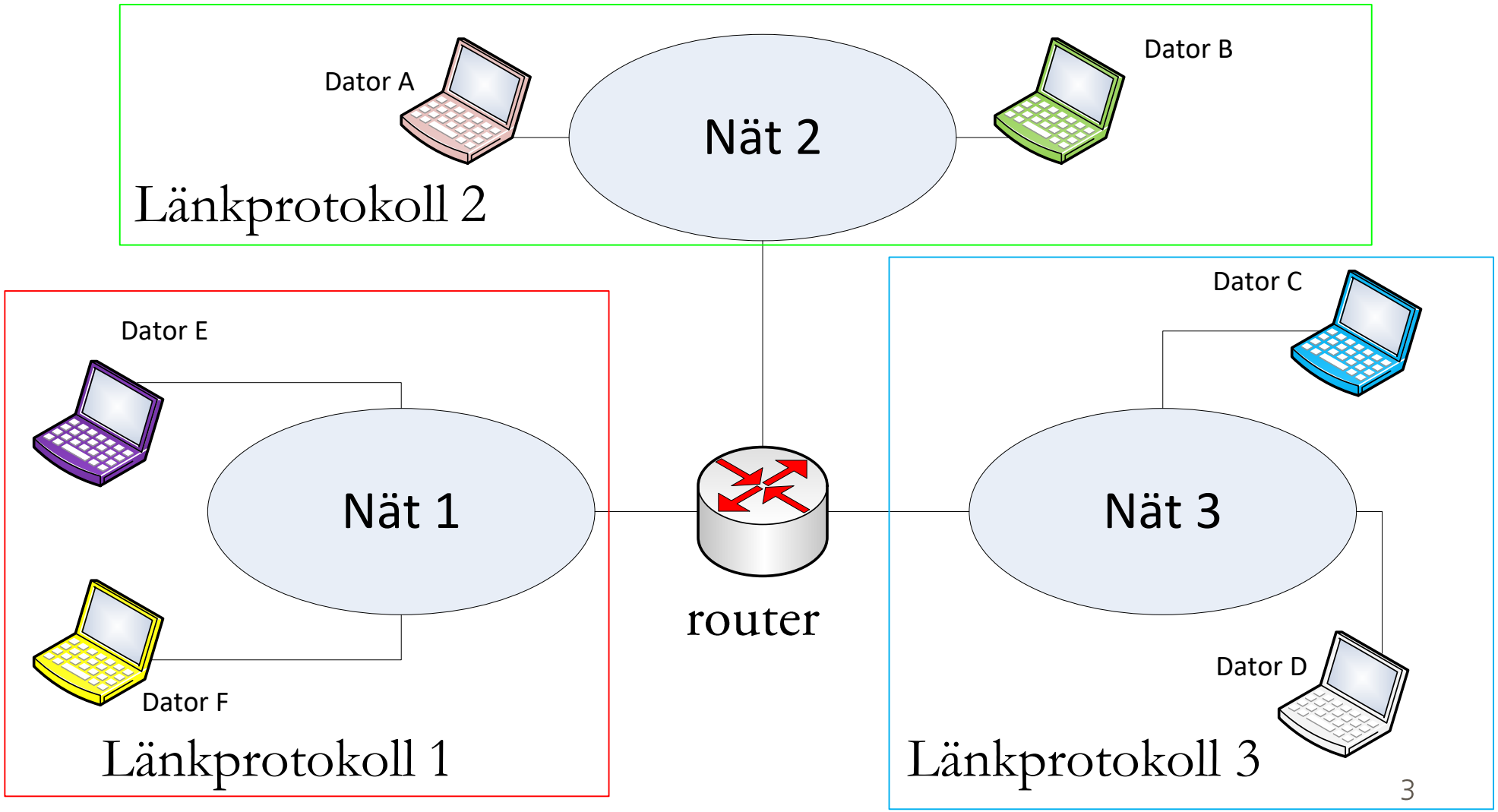
Kihl & Andersson: 7.1-7.6, 10.1-3

Stallings: 14.1-4, 15.1-3, 21.5

DHCP beskrivs även bra på

https://sv.wikipedia.org/wiki/Dynamic_Host_Configuration_Protocol

Repetition


Dator A
Dator B
Nät 2
Länkprotokoll 2
Dator E
Dator C
Nät 1
Nät 3
router
Dator F
Dator D
Länkprotokoll 1
Länkprotokoll 3

Internet-adresser

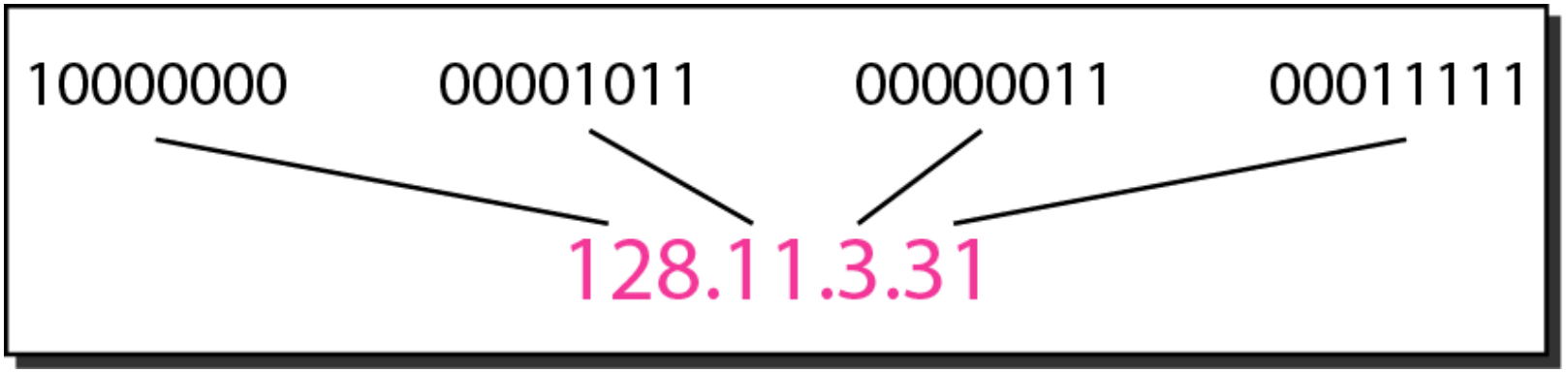
- Olika nät kan använda olika länkprotokoll och olika system för fysiska adresser. Varje host i ett specifikt nät måste ha en **fysisk adress (länkadress, MAC-adress)** som matchar just det länkprotokollet som används i det nätet.
- Alla nät på Internet måste använda IP som nätprotokoll, och varje host måste ha en **IP-adress** för att kunna kommunicera med en host på ett annat nät.
- Det måste finnas en mappning mellan fysisk adress och IP-adress (beskrivs snart).

Fysisk adress (MAC-adress) för 802.x nät

Alla terminaler med ett nätverkskort för IEEE 802.x har en fysisk adress, kallad MAC-adress. Har terminalen flera nätverkskort har den flera MAC-adresser.

IPv4-adresser

Varje värddator och routrar som är ansluten till Internet har en unik IP-adress. Om IPv4 används så är adressen på 32 bitar.



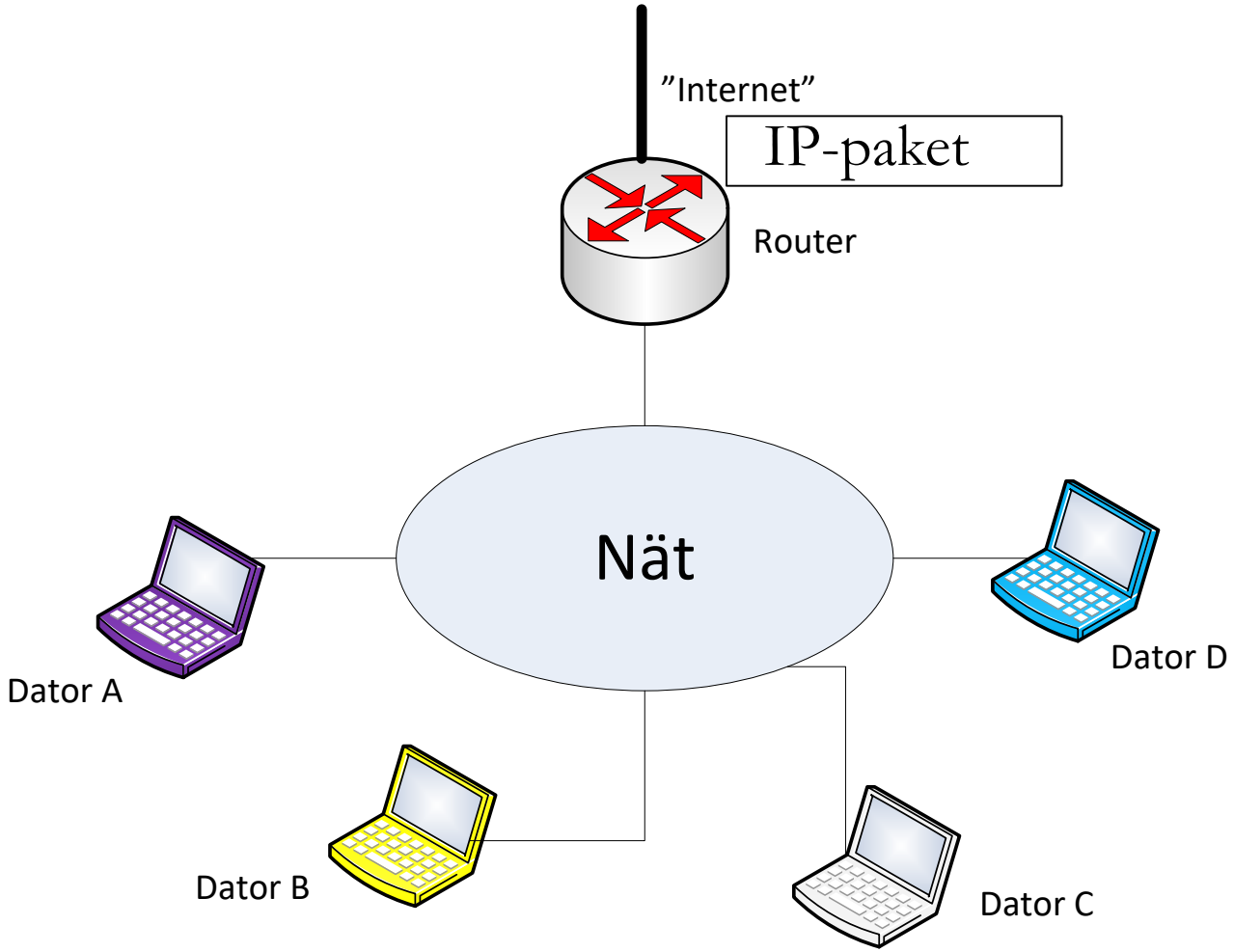
Adressen skrivs i så kallat dotted-decimal format.

Tentaexempel

Följande Ethernet-ram bär ett IP-paket. Preamble och SFD är borttagna. Identifiera sändarens MAC-adress samt IP-adress.

```
00 00 0c 07 ac 01 00 00 39 51 90 37 08 00 45 00
05 dc 48 00 20 00 20 01 94 67 82 eb 12 7f 82 eb
80 64 08 00 e3 fb 03 00 0c 00 61 62 63 64 65 66
67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76
77 61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f
```

Hur mappar man IP-adress till fysisk adress?



Address Resolution Protocol (ARP)

För att hitta en specifik host inom ett nät krävs det att alla enheter kopplade till nätet kan mappa IP-adresser till de fysiska adresser som används inom nätet.

Adress Resolution Protocol (ARP) används för detta inom nät som bygger på IEEE 802.x standarder.

Notering: En host i ett nät vet alltid IP-adressen till den router som är kopplad till ”resten av” Internet. Denna router kallas Default router/gateway.

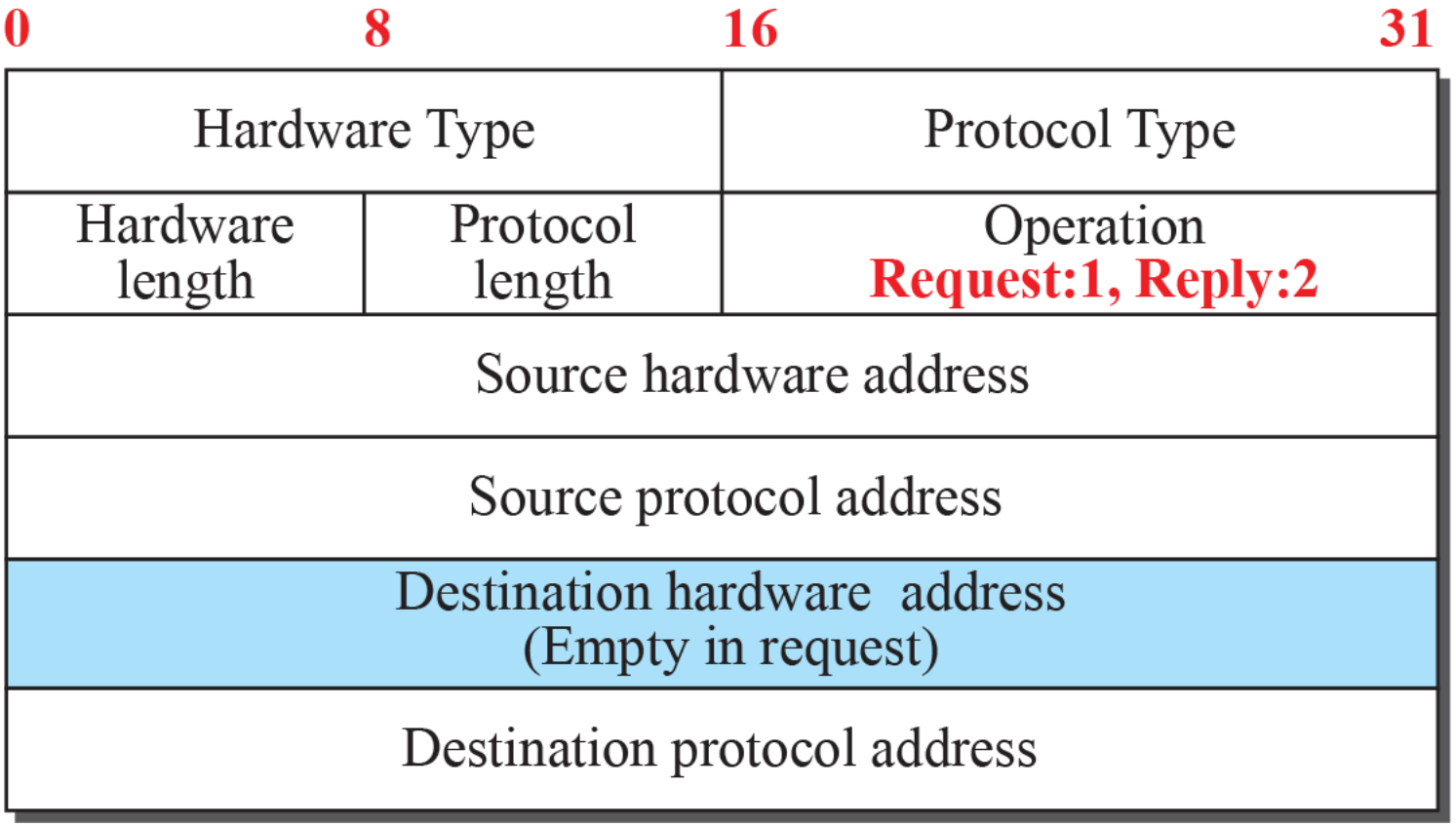
ARP-funktioner

- Varje host/router har en ARP-cache (tabell) som används för att registrera MAC/IP-adresspar.
- En ARP query broadcastas varje gång en host/router behöver mappa en IP-adress till en MAC-adress (**ARP broadcasts stoppas vid varje router**).
- Den host som har den efterfrågade IP-adressen skickar tillbaka en ARP response med sin MAC-adress i unicast.

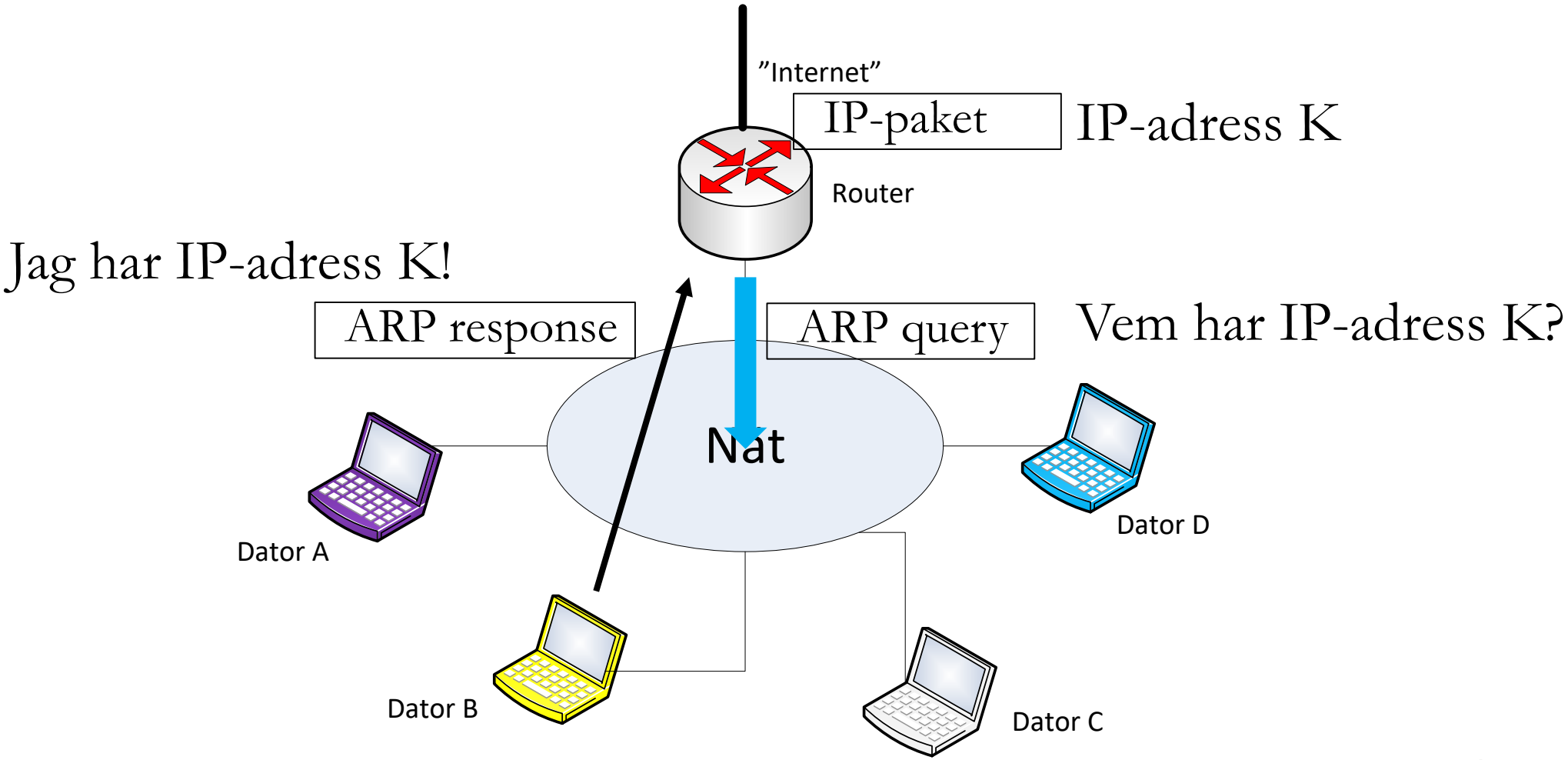
ARP-paket

Hardware: LAN or WAN protocol (Ethernet = 1)

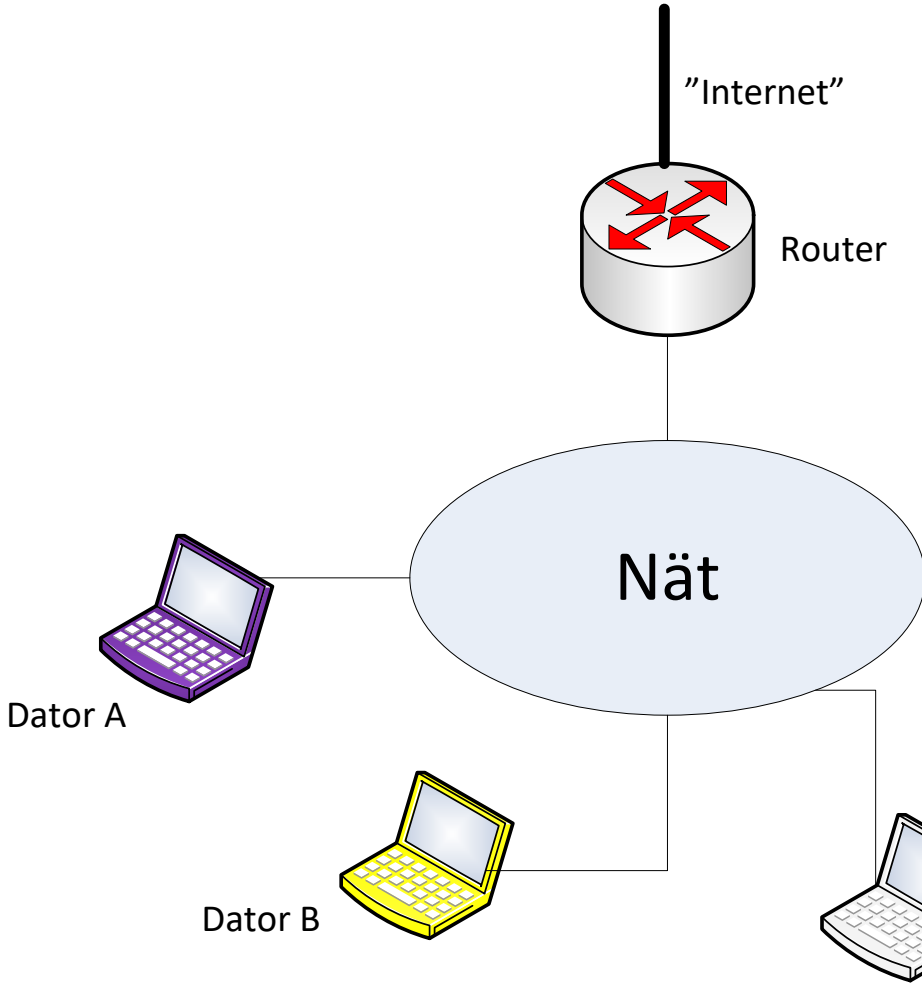
Protocol: Network-layer protocol (IPv4 = $(0800)_{16}$)



ARP-funktion (1)



ARP-funktion (2)

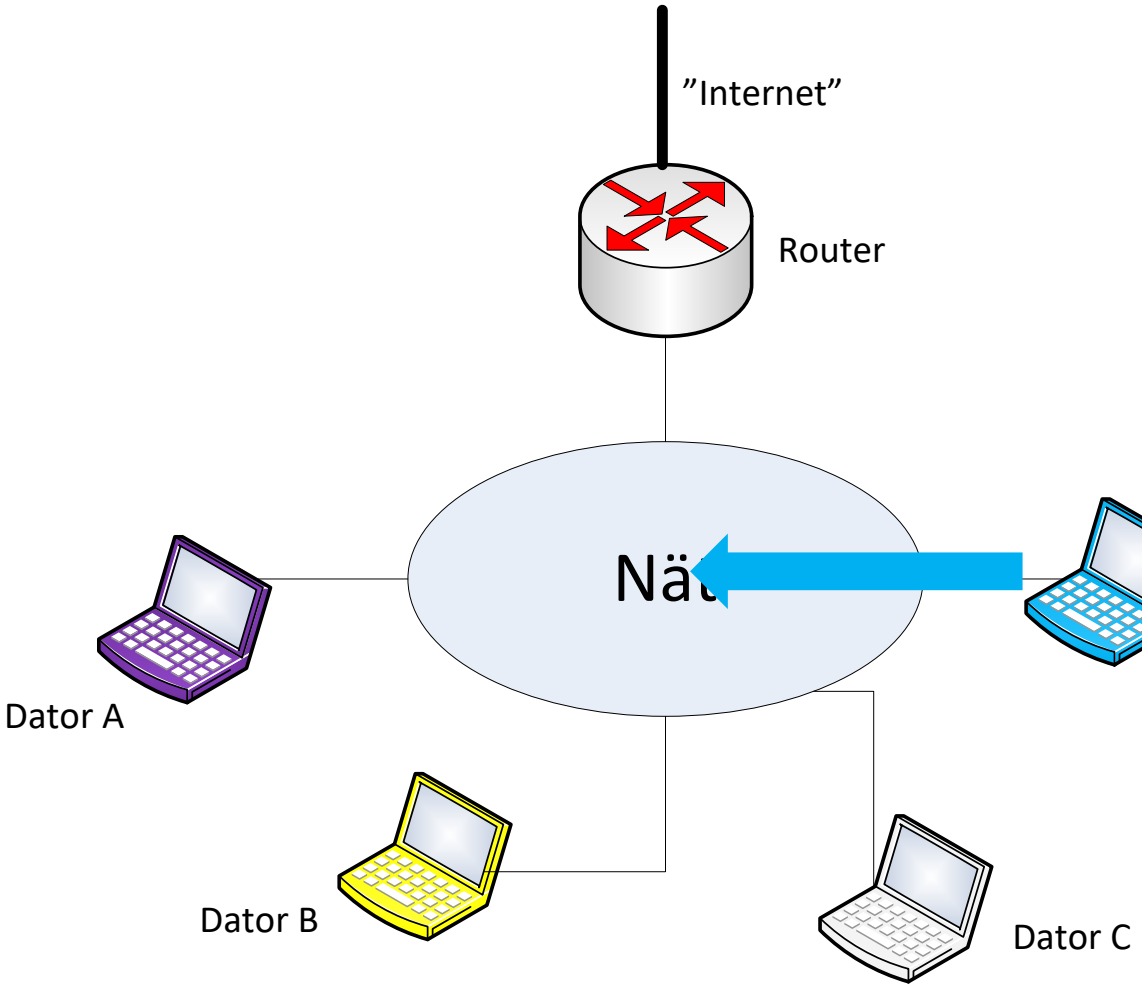


1. Host finns på samma nät
2. Host finns på annat nät

(identifieras av IP-adressen)

IP-paket IP-adress K

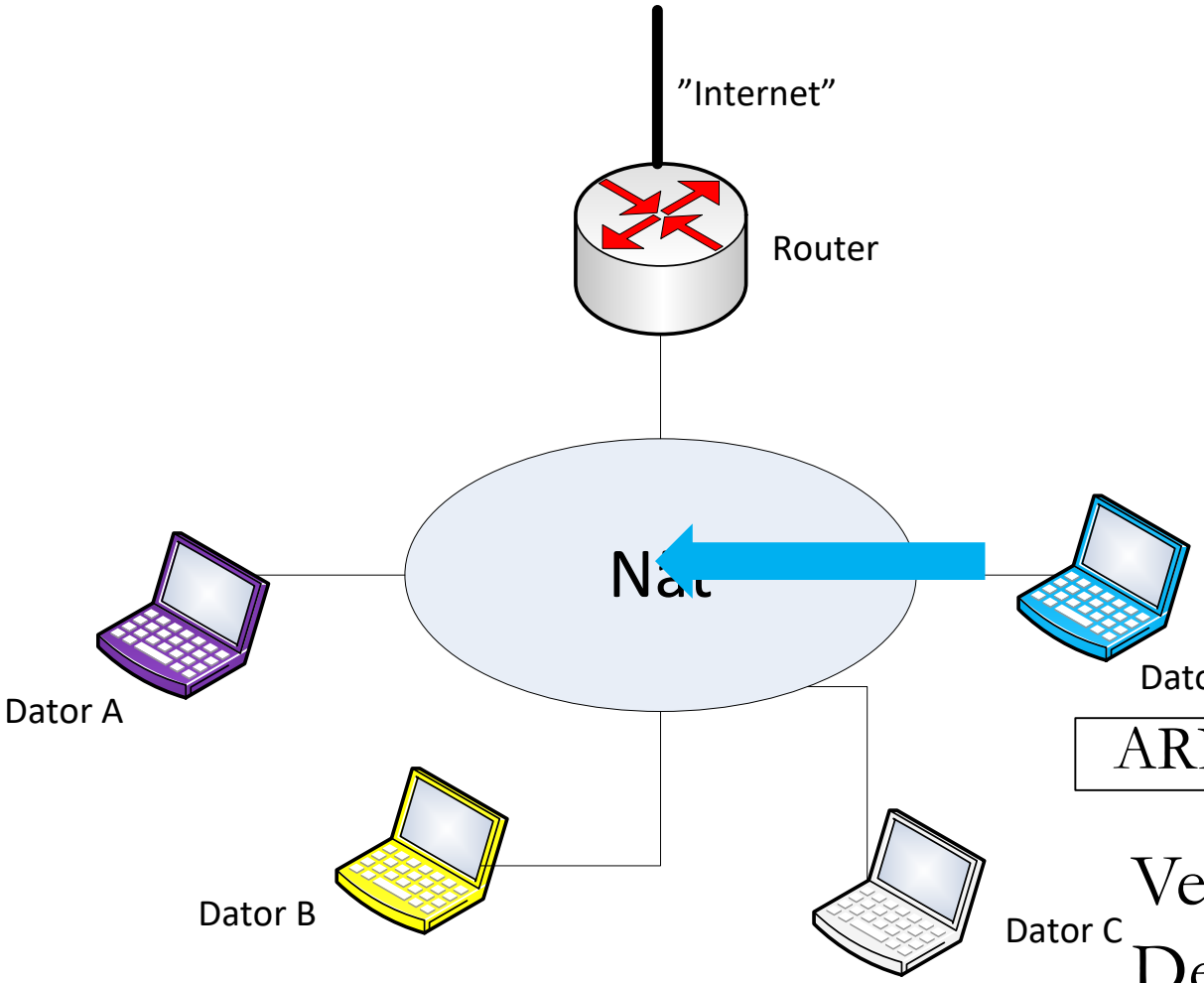
1. Host finns på samma nät



Om en host sitter på samma nät skickas en ARP ut som vanligt.

Vem har IP-adress K?

1. Host finns på annat nät



Om en host sitter på ett annat nät skickas paketet till Default gateway. ARP skickas om det behövs.

Vem har IP-adress Default gateway?

Hur får en host sin IP-adress?

I denna kurs ingår inte hur en host får sin IP-adress. Vi antar bara följande:

1. Varje host kan sin egen IP-adress (nät-id och värd-id) och MAC-adress.
2. Varje host kan IP-adressen till sin Default Gateway.

Oftast används protokollet **DHCP (Dynamic Host Configuration Protocol)** för dessa funktioner.

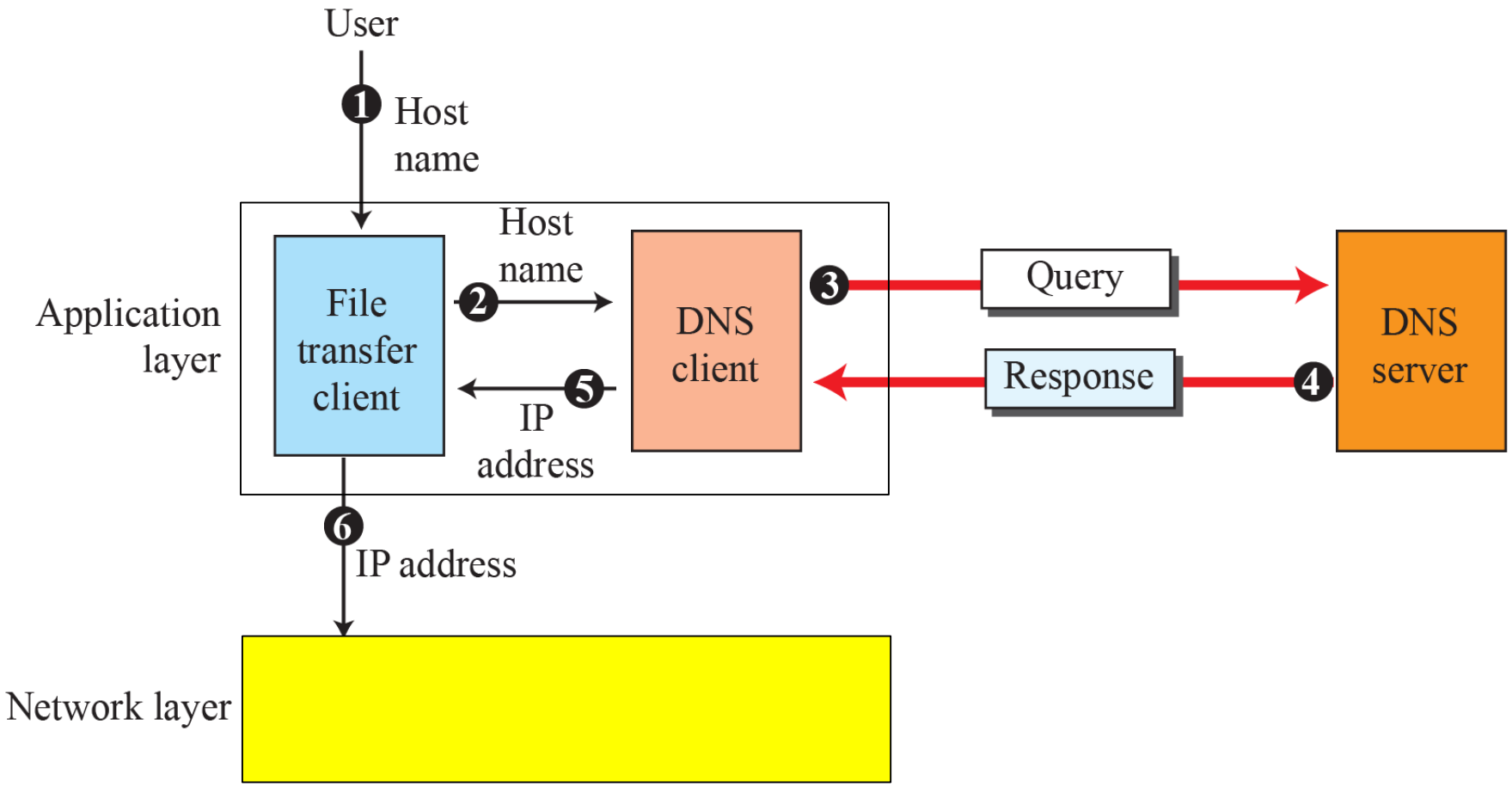
Dynamic Host Configuration Protocol (DHCP) funktioner

1. När en terminal kopplas in i ett nät skickar terminalens DHCP-klient ett broadcast meddelande med en **DHCP-förfrågan**.
2. DHCP-servern i nätet (tex defautlt router) svarar med ett **erbjudande om IP-adress** och annan information (tex nätverksadress, Default router, DNS-server etc.).
3. Terminalen kan tacka ja till detta erbjudande och får därmed en IP-adress som gäller en viss tid.

Mappning från host namn till IP-address

- Applikationsprotokoll använder symboliska host-namn (exempel, www.lth.se).
- Men, TCP/IP använder IP-adresser.
- Mappning från host-namn till IP-adresser genomförs av Domain Name System (DNS).

DNS grundläggande funktion

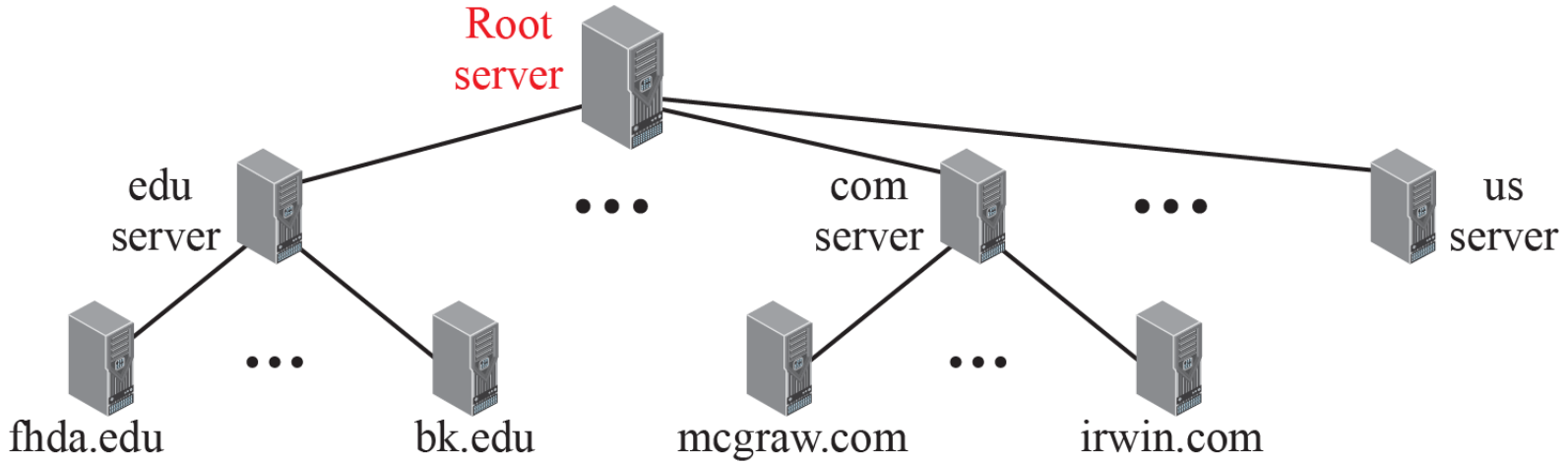


Domän-namn (Domain Name Space)

- DNS använder ett hierarkiska host-namn och hela Internet delas in i domäner och subdomäner (domains och subdomains).
- Ett domän-namn är en sekvens av labels separerade med punkter, e.g. www.eit.lth.se.

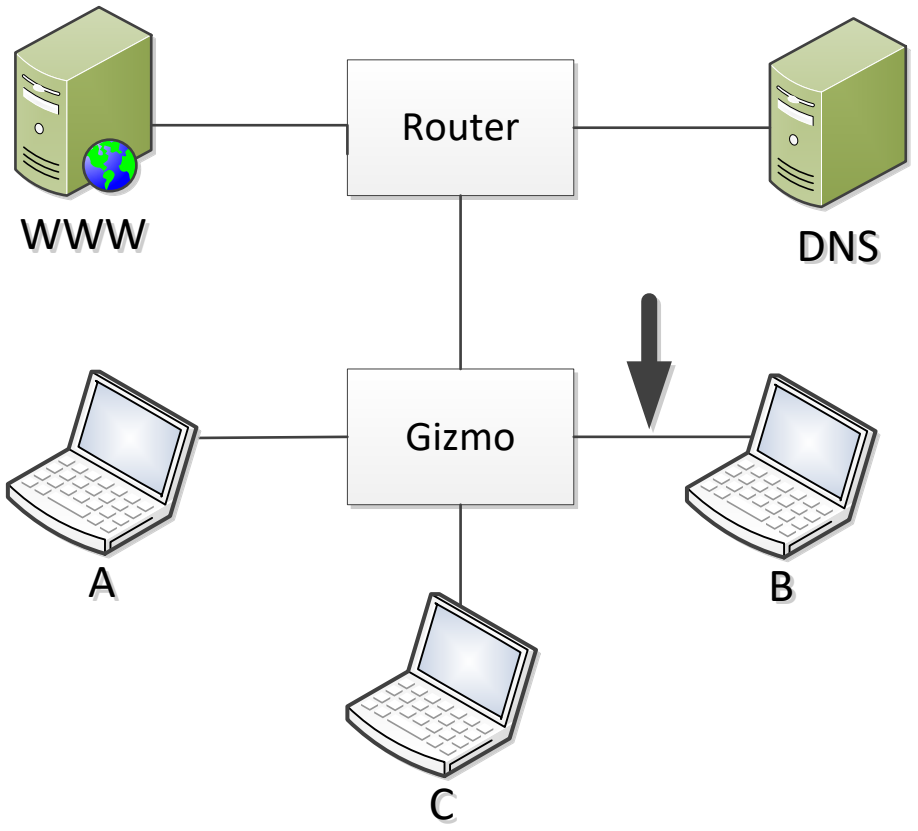
Domain Name Servers

Domännamnen registreras i särskilda DNS-servrar. Dessa servrar är distribuerade och varje domän eller subdomän har sina egna servers.



En host vet alltid IP-adressen till sin närmaste DNS-server.

Tentaexempel



Anta att A vill skicka ett IP-paket till C och vet C:s IP-adress. Alla adress-cacher är tomma.

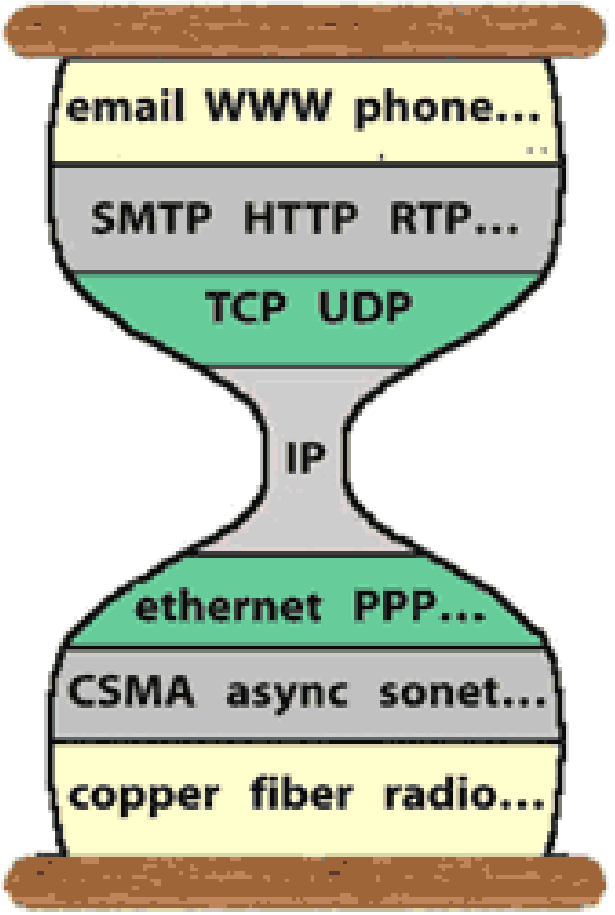
Beskriv vilka meddelanden som skickas på länken till B (vid pilen) om Gizmo är en (i) hub; (ii) switch; (iii) router. Motivera dina svar!

TCP/IP-modellen

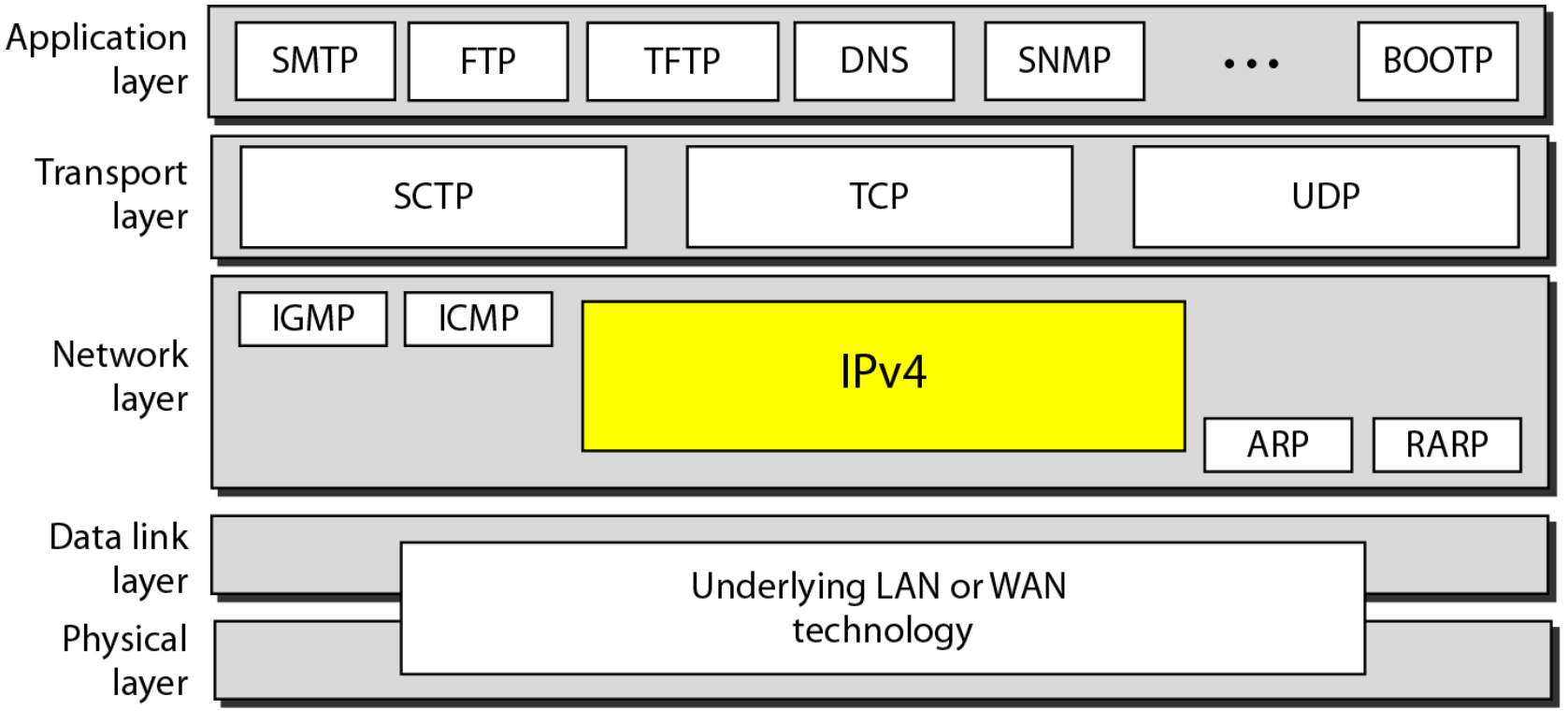
OSI-modellen	TCP/IP-modellen
Applikation	Applikation
Presentation	
Session	
Transport	Transport
Nät	Nät
Länk	Underliggande nät
Fysiska skiktet	

TCP/IP-modellen

TCP/IP-modellen illustreras ibland med ett timglas.

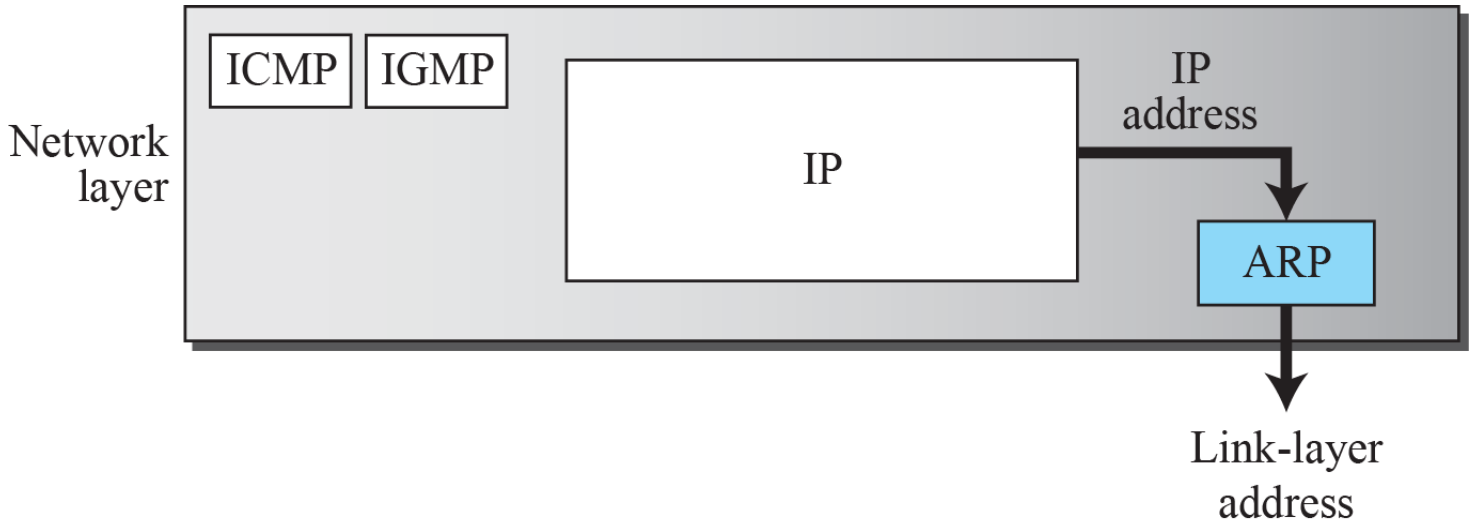


Internetprotokollen

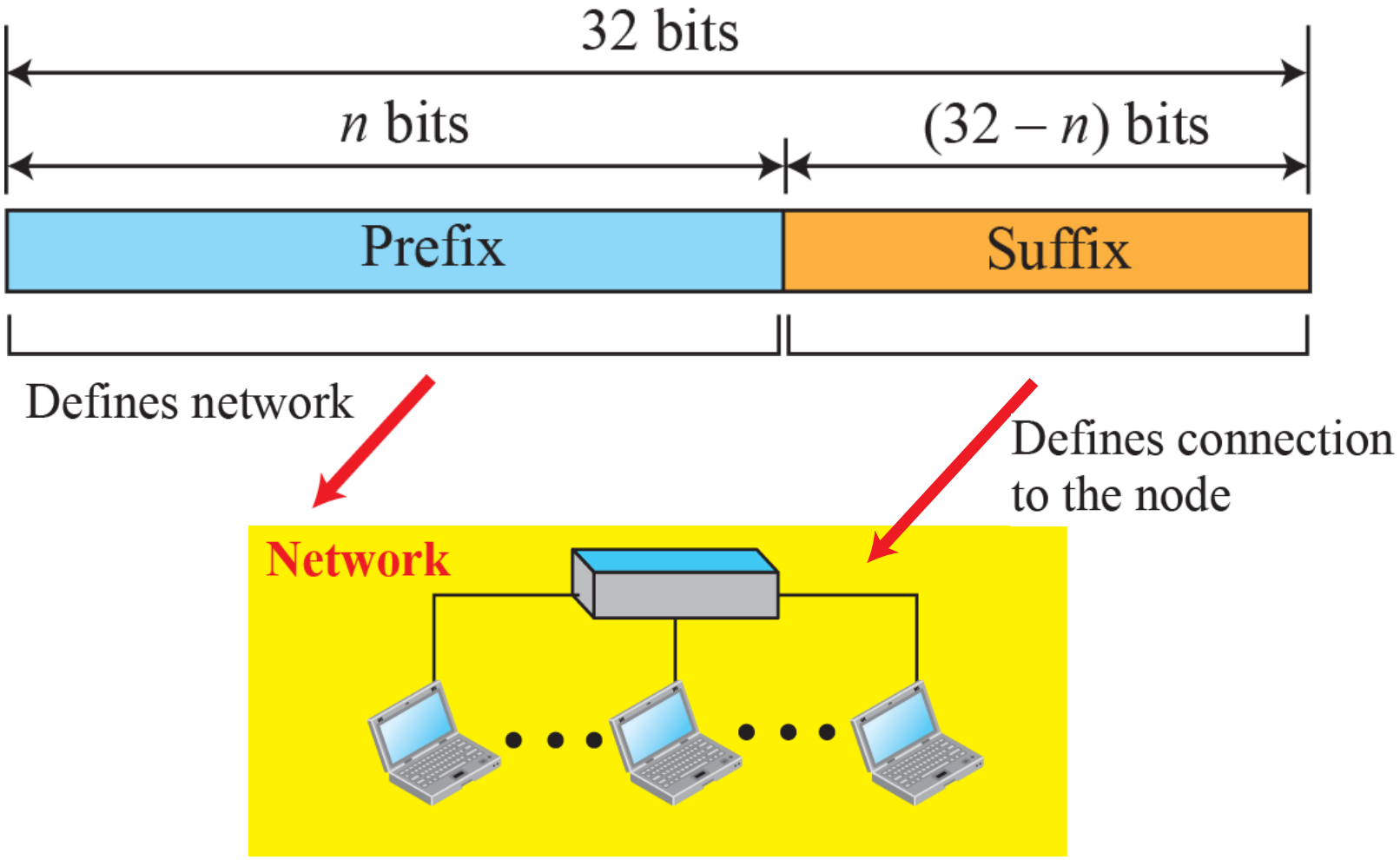


ARP i TCP/IP-modellen

ARP brukar placeras mellan lager 2 och lager 3 eftersom ARP hanterar både IP-adresser och MAC-adresser.



IPv4-adresser



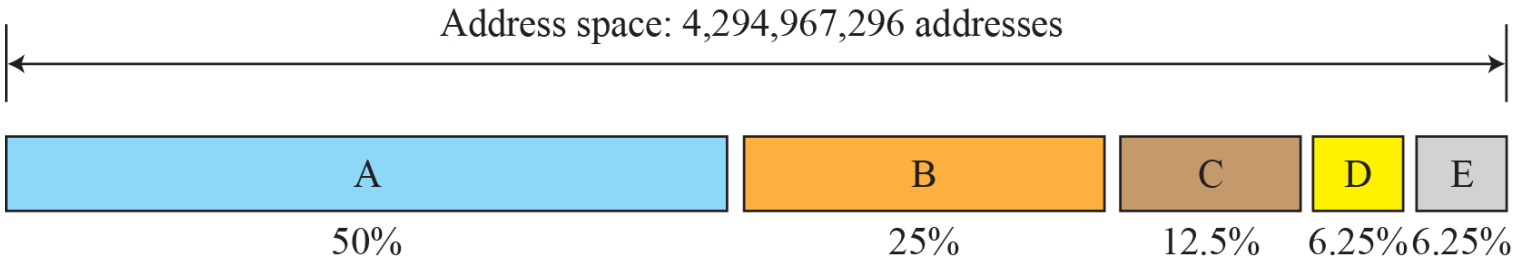
Adress-metoder för IPv4

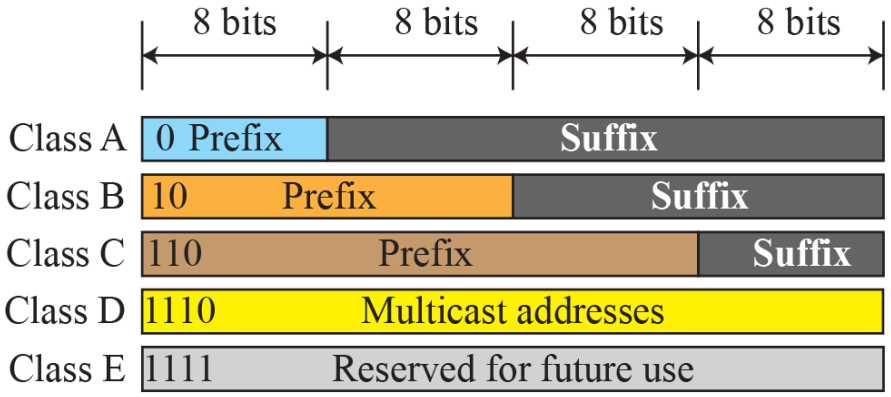
Det finns två sätt att definiera adresser:

- Klassindelad adressering (Classful addressing)
- Klasslös adressering (Classless addressing)

Klassindelad adressering

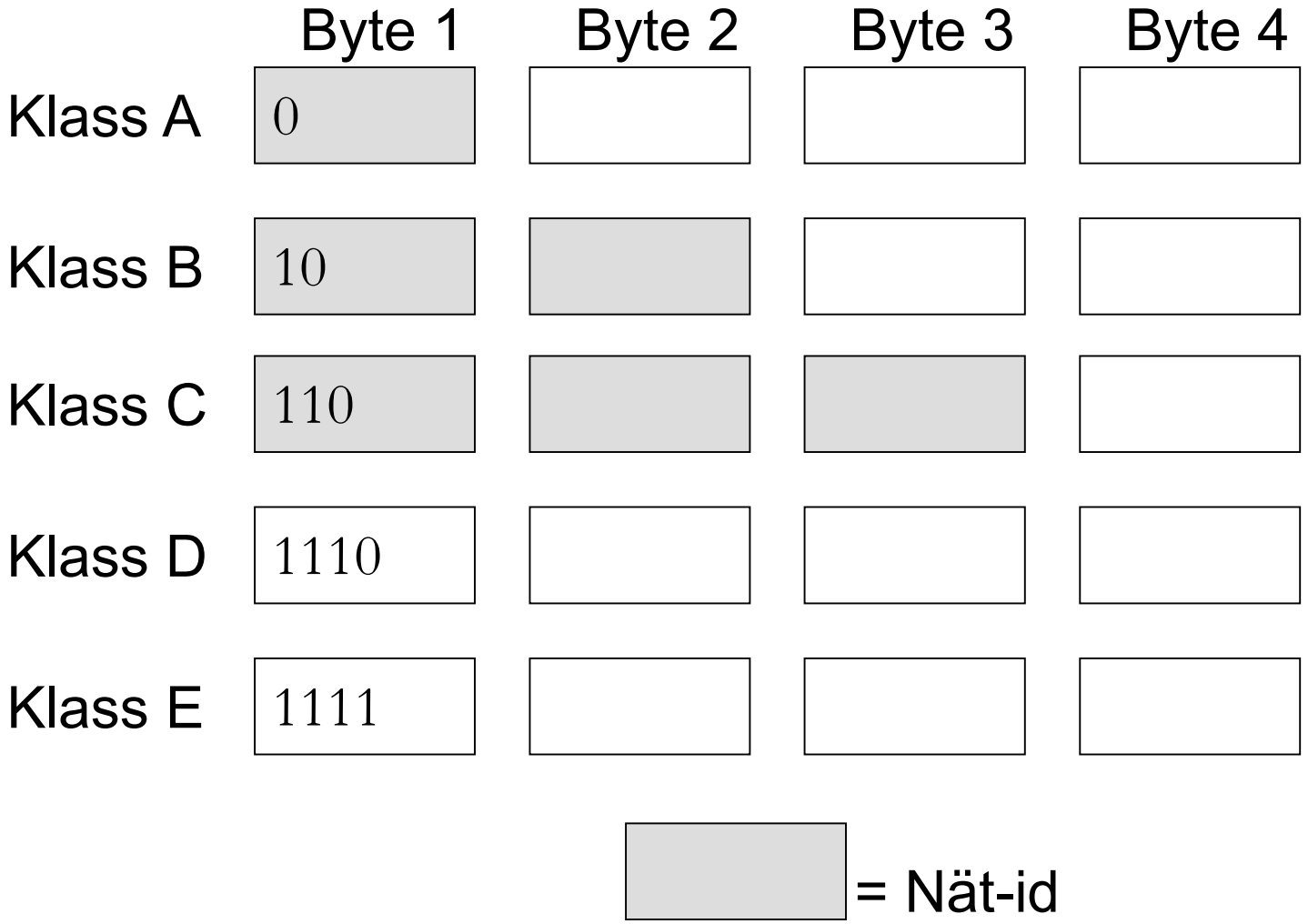
Fem adressklasser: A, B, C, (D, and E)





Class	Prefixes	First byte
A	$n = 8$ bits	0 to 127
B	$n = 16$ bits	128 to 191
C	$n = 24$ bits	192 to 223
D	Not applicable	224 to 239
E	Not applicable	240 to 255

Nät-id och värd-id i klassindelad adressering



Klassindelad adressering

Det största problemet med klassindelad adressering var att en organisation bara kunde få ett block med adresser.

Class	*Number of Blocks*	*Block Size*	*Application*
A	128	16,777,216	Unicast
B	16,384	65,536	Unicast
C	2,097,152	256	Unicast

Därför, 1996, infördes klasslös adressering.

Klasslös adressering

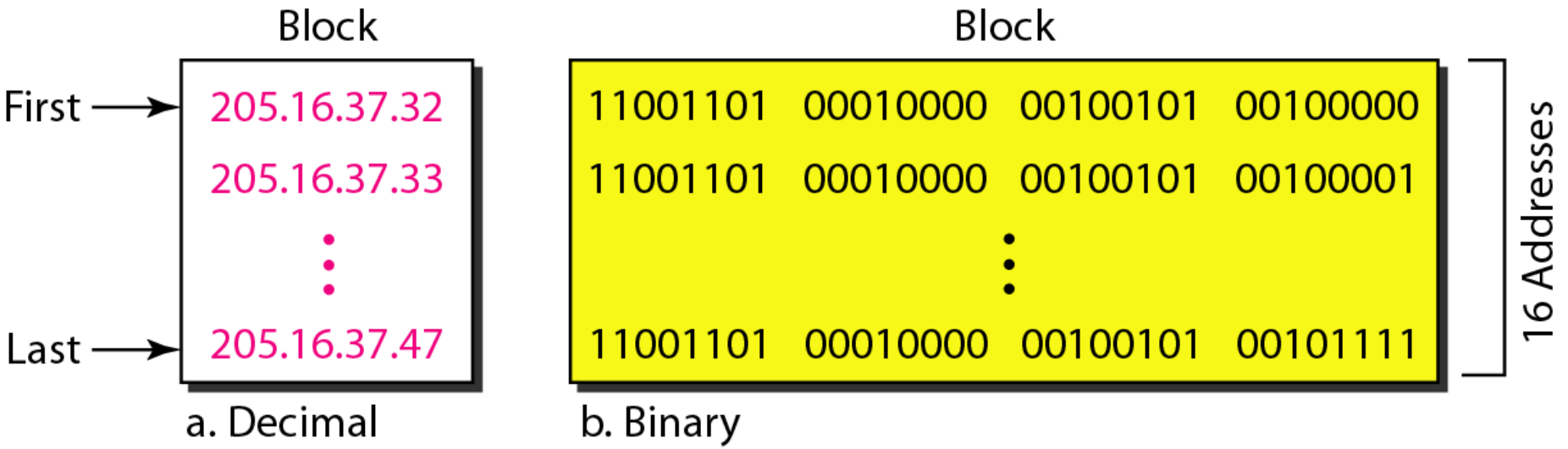
I klasslös adressering, blir en organisation (liten eller stor) tilldelad ett block med adresser, med följande restriktioner:

Adresserna i blocket måste följa på varandra.

Antalet adresser måste vara av formen 2^x (1, 2, 4, 8, etc.).

Kallas även för Classless Interdomain Routing (CIDR)

Klasslös adressering, exempel



Mask

Ett adress-block definieras av sin mask.

En mask består av 32 bitar där en etta indikerar att adressbiten på motsvarande position ingår i nät-id.

Ett block med adresser kan då definieras som:
x.y.z.t /*n*
där x.y.z.t definierar en av adresserna och /*n* definierar masken.

Nätmasken används endast av hosts och routrar, och skickas inte med i IP-headern.

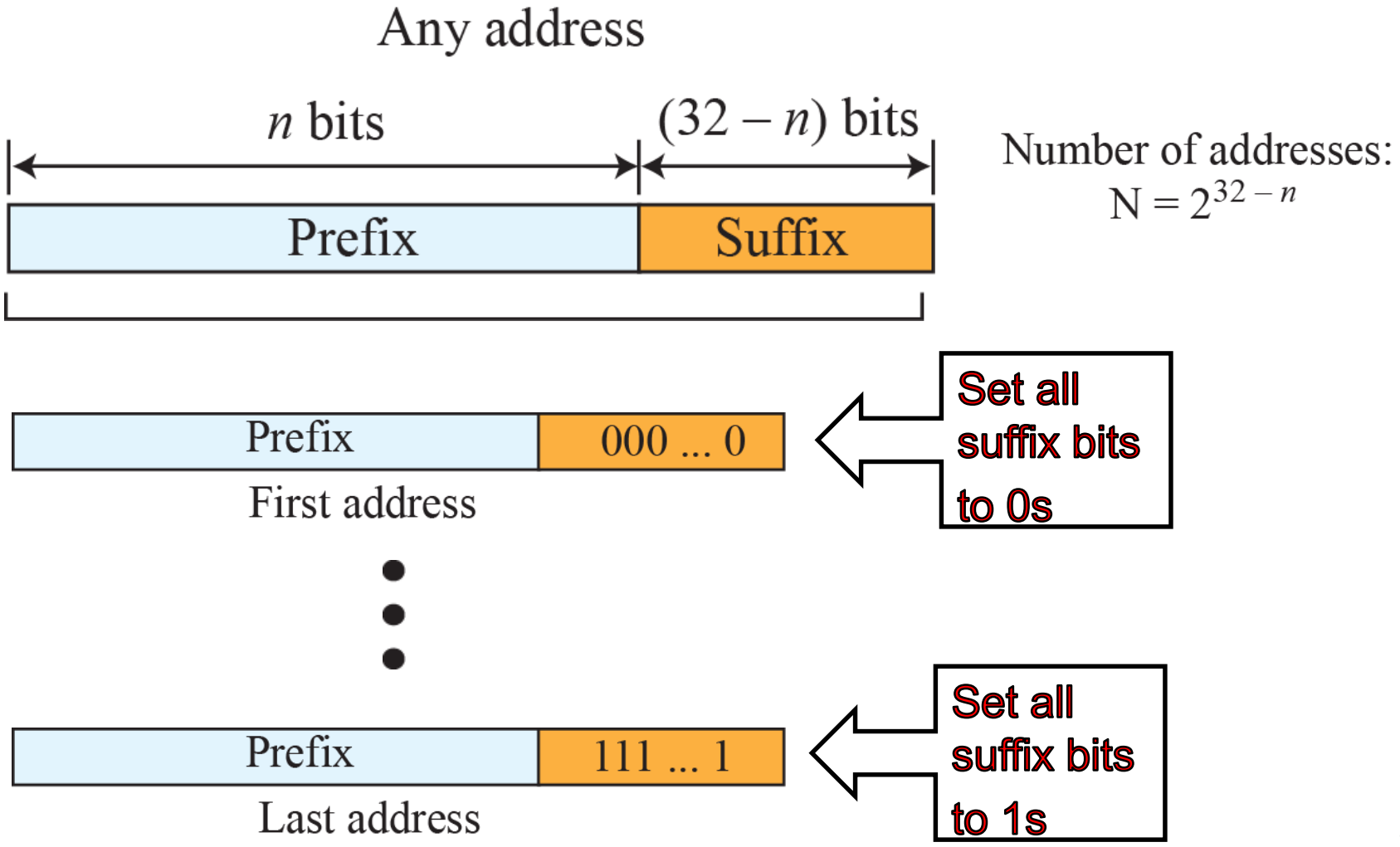
CIDR, exempel

Adress:	11011110 00010111 01000011 01000100
Mask:	11111111 11111111 11000000 00000000
Nät-id:	11011110 00010111 01000000 00000000
Värd-id:	00000000 00000000 00000011 01000100

Decimal-dotted format: 232.23.67.68/18

- Adress med värd-id satt till bara 0:or representerar nätet
- Adress med värd-id satt till bara 1:or är broadcast

CIDR adress-block



Klasslös v. Klassindelad adressering

Klassindelad adressering kan också representeras av masker:

Class	*Binary*	*Dotted-Decimal*	*CIDR*
A	**11111111** 00000000 00000000 00000000	**255**.0.0.0	/8
B	**11111111 11111111** 00000000 00000000	**255.255**.0.0	/16
C	**11111111 11111111 11111111** 00000000	**255.255.255**.0	/24

Tentaexempel

Identifiera nät-id och värd-id för adressen

160.184.66.53/28

Identifiera även adressblocket som adressen ingår i.

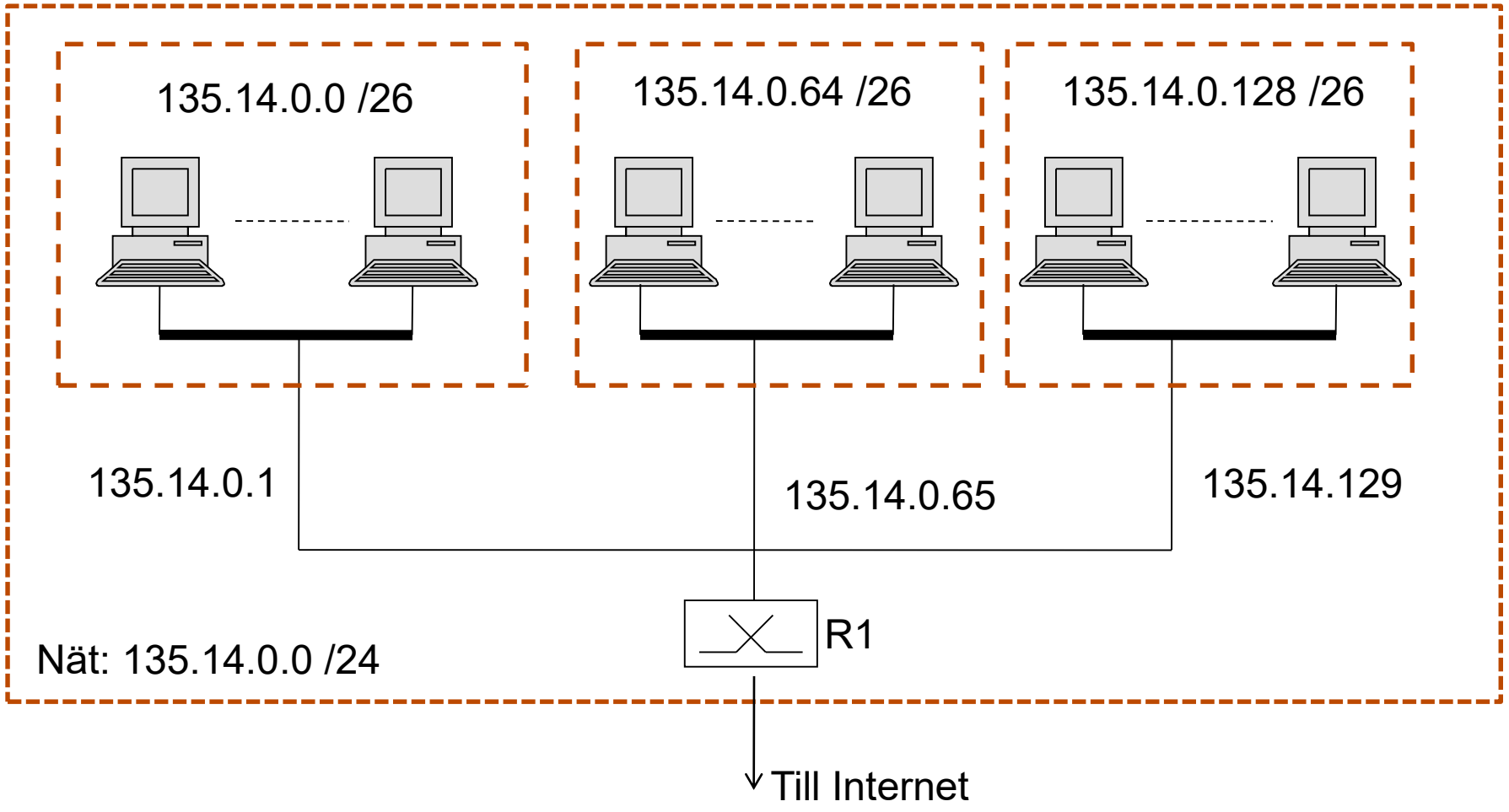
Subnetting

Idén med klasslös adressering kom av tekniken subnetting som användes för klass A- och B-nät.

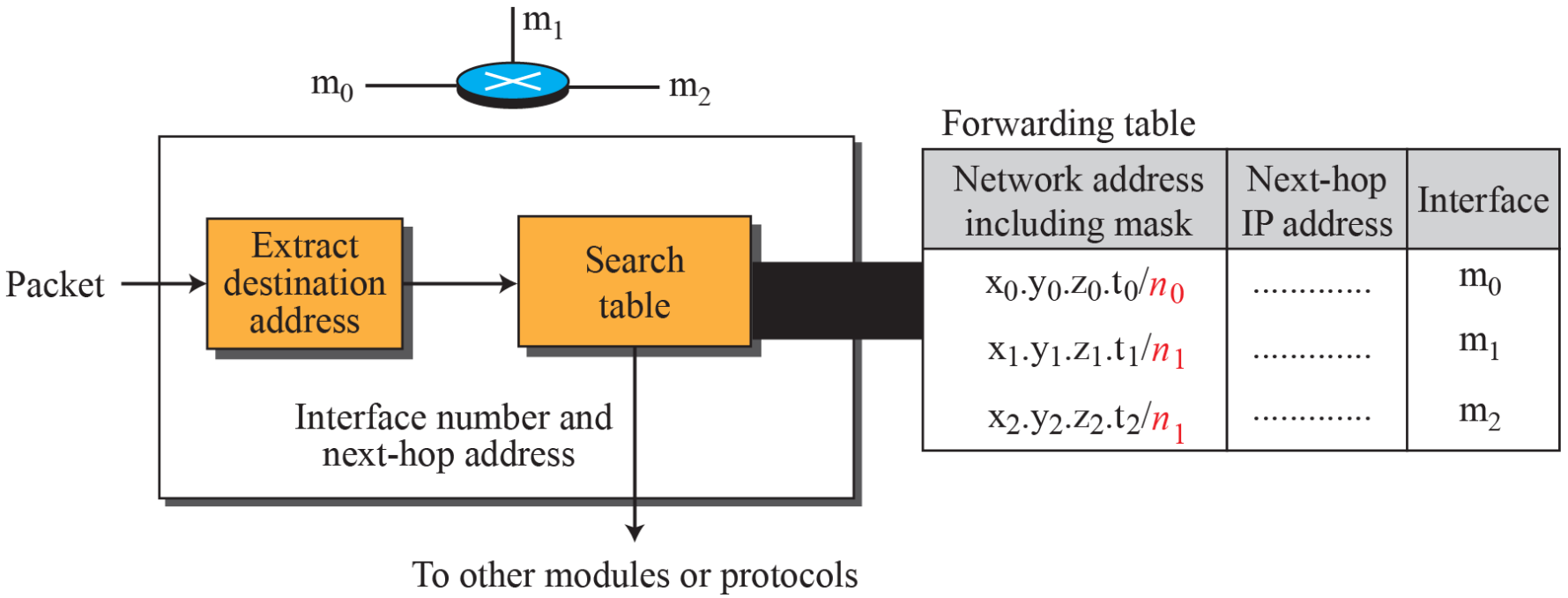
De var för stora att hantera som ett nät och för att kunna dela upp dem i flera mindre nät så infördes en adressmask.

En organisation kan få ett adressblock enligt reglerna för klassindelad adressering, och sedan internt dela upp nätet i flera med tillhörande mask.

Subnetting, exempel



Routrar använder nätadressen



Alla routrar måste kunna så kallad forwarding, dvs skicka vidare paket baserat på nätadressen.

Forwarding-tabell

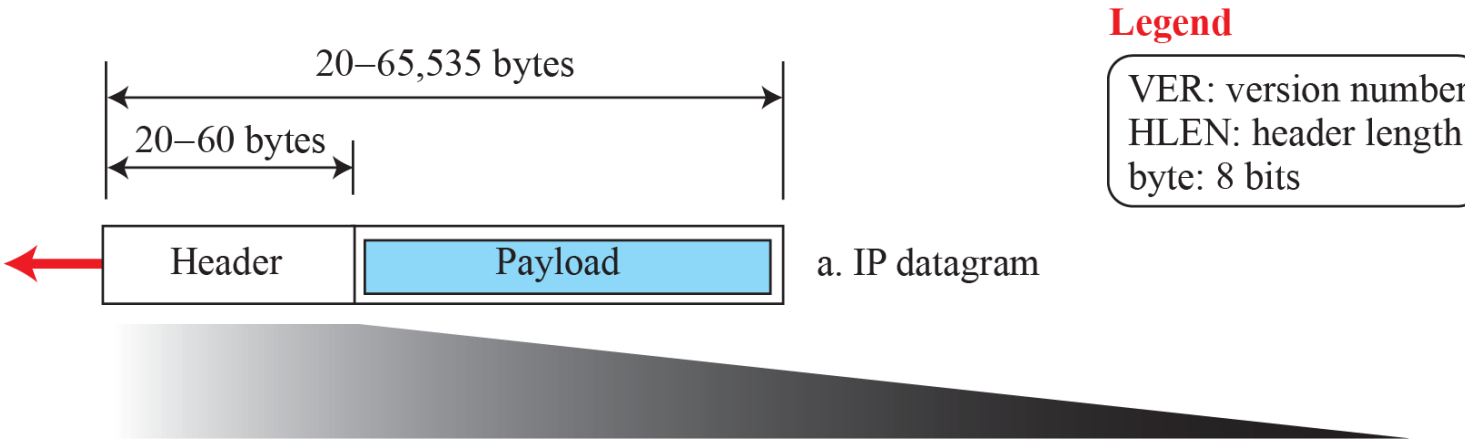
De *n* högsta bitarna för destinationsadressen (prefix) sparas och resten av bitarna (suffix) sätts till noll innan destinationsadressen jämförs med forwarding-tabellen.

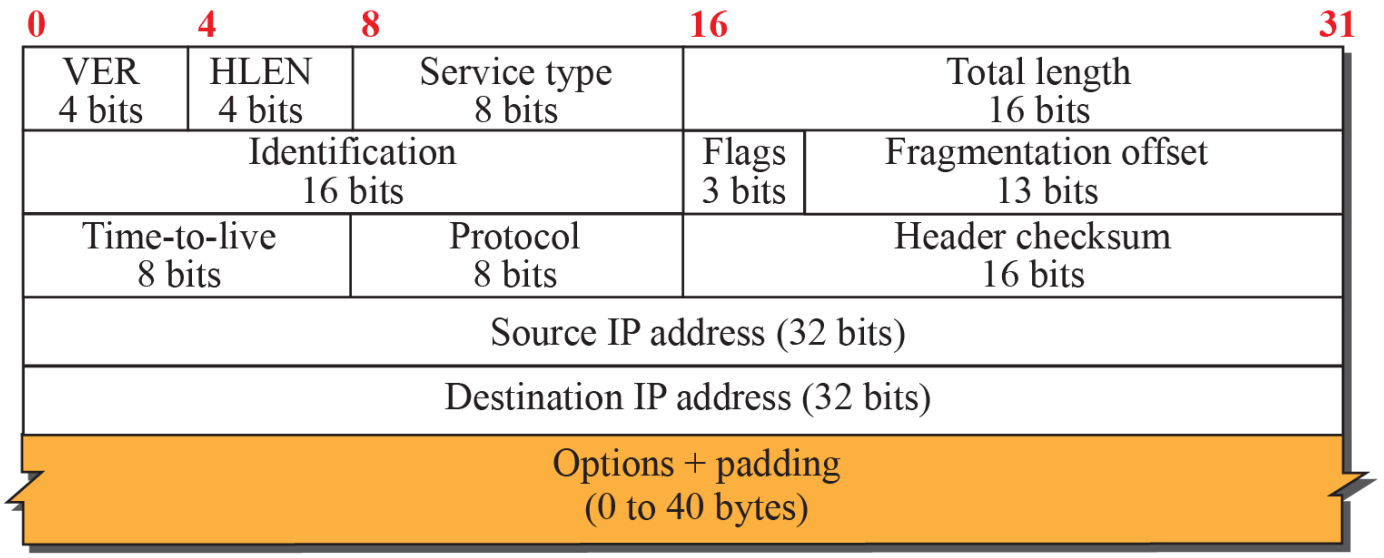
Network address/mask	*Next hop*	*Interface*
180.70.65.192/**26**	—	m2
180.70.65.128/**25**	—	m0
201.4.22.0/**24**	—	m3
201.4.16.0/**22**	—	m1
Default	180.70.65.200	m2

Fragmentering

- IP tillåter 65.535 bytes data (payload). Men IP-paket kan fragmenteras om den data som kommer från transportprotokollet inte kommer att få plats i en ram på länklagret. Tex. Ethernet tillåter max 1500 bytes (IEEE 802).
- Det är sändaren som fragmenterar data, och mottagaren (den host som ska ha IP-paketet) som sätter ihop data igen.
- Header-fälten *identification*, *flags* och *fragmentation offset* används

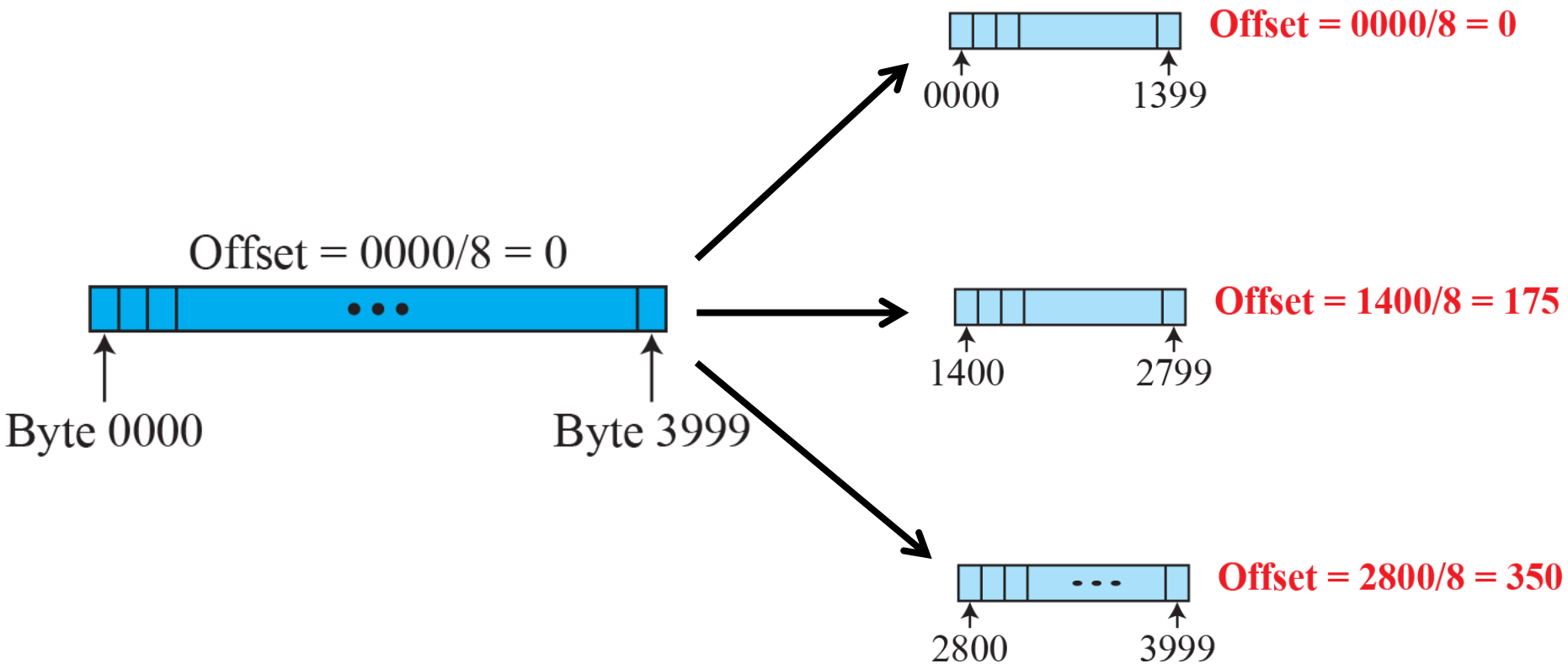
Header för IPv4-datagram





b. Header format

Fragmentering offset exempel



Motivering för IPv6

IPv4 har följande stora problem:

- Det finns inte tillräckligt med IPv4-adresser.
- IPv4 var inte utvecklat för realtidsapplikationer.
- IPv4 har inga funktioner för säkerhet.

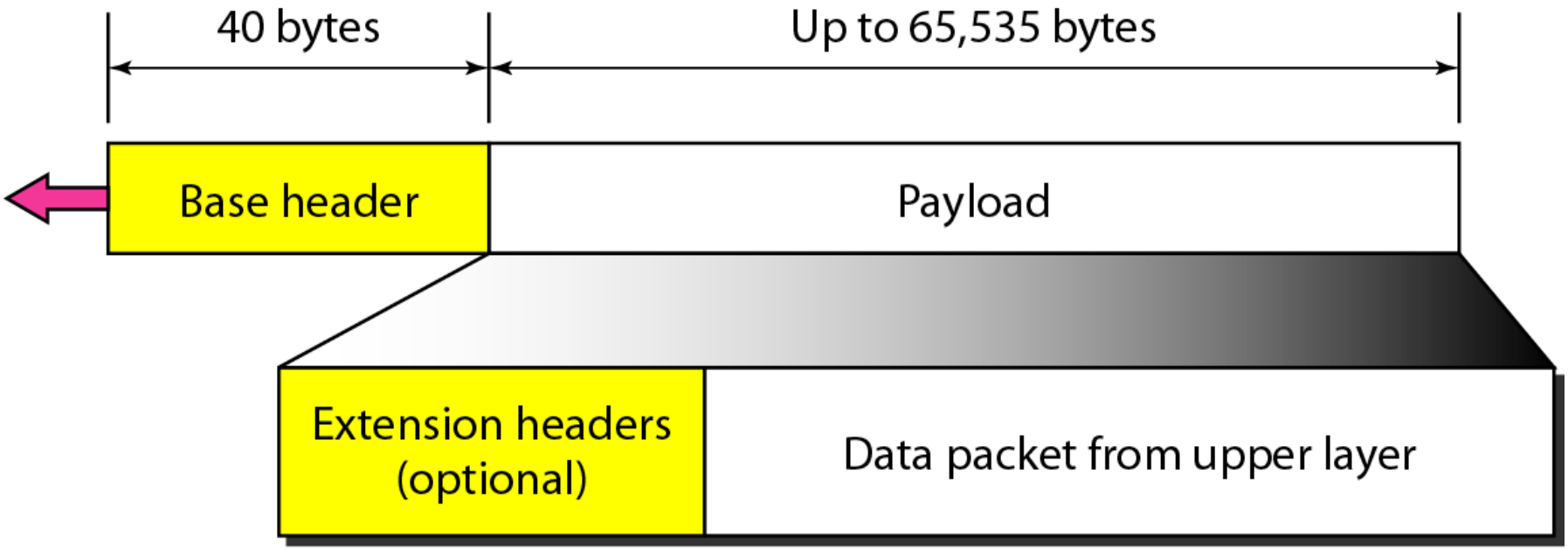
Detta är några anledningar till att IPv6 utvecklades.
Men övergången till IPv6 går långsamt.

- Några operatörer i Sverige använder IPv6 inom sina nät.

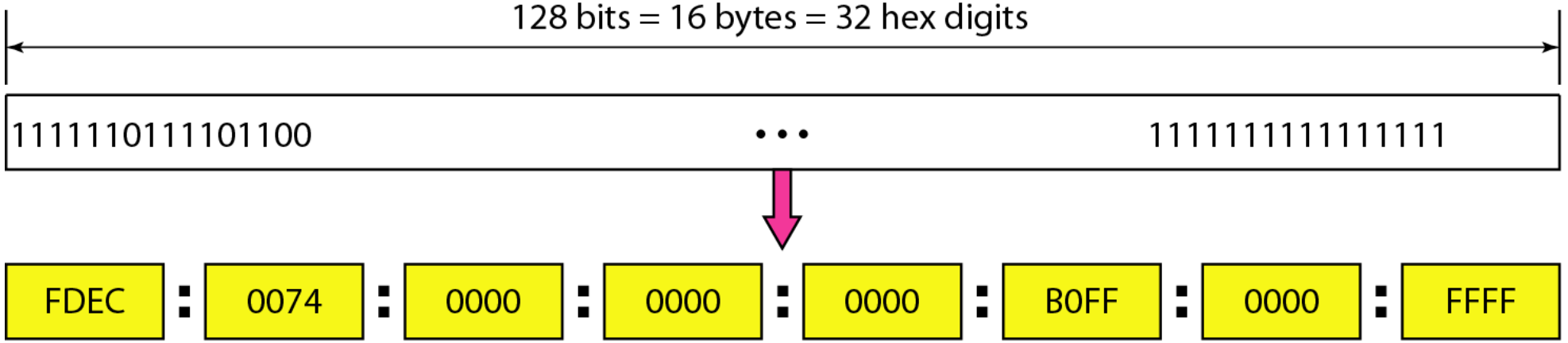
Några fördelar med IPv6

- Fler adresser: 128 bitars adresser.
- Bättre header format: IPv6 har en basheader konstant längd 40 bytes. Det går att lägga till options, men det finns regler för hur man gör det.
- Fler säkerhetsfunktioner: IPv6 har inbyggda säkerhetsfunktioner.
- Support för realtidsapplikationer: Routrar ska kunna specialhantera realtidsapplikationer för att dataöverföringen ska gå snabbare.

IPv6-paket

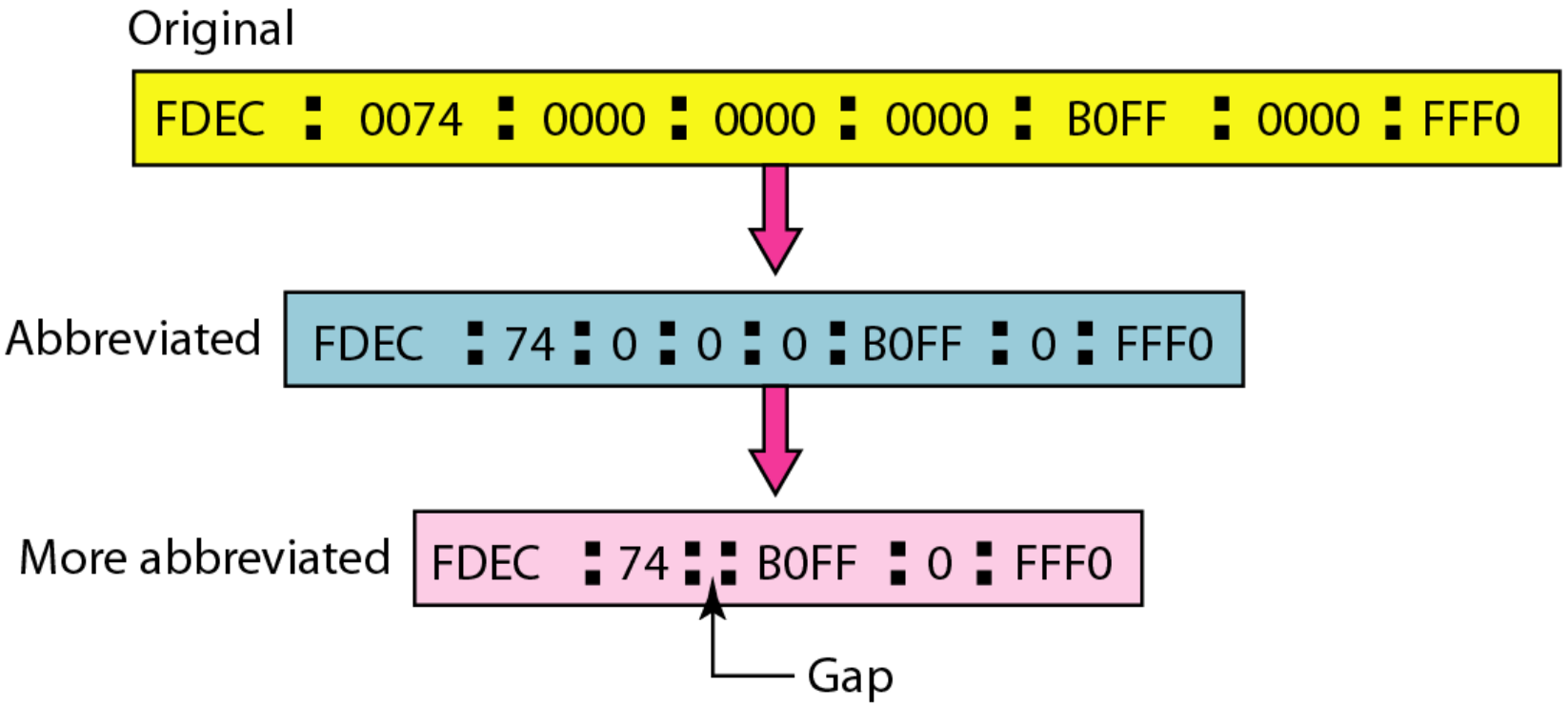

40 bytes
Up to 65,535 bytes
Base header
Payload
Extension headers (optional)
Data packet from upper layer

IPv6-adresser



Hexadecimal colon notation

Förkortade IPv6-adresser



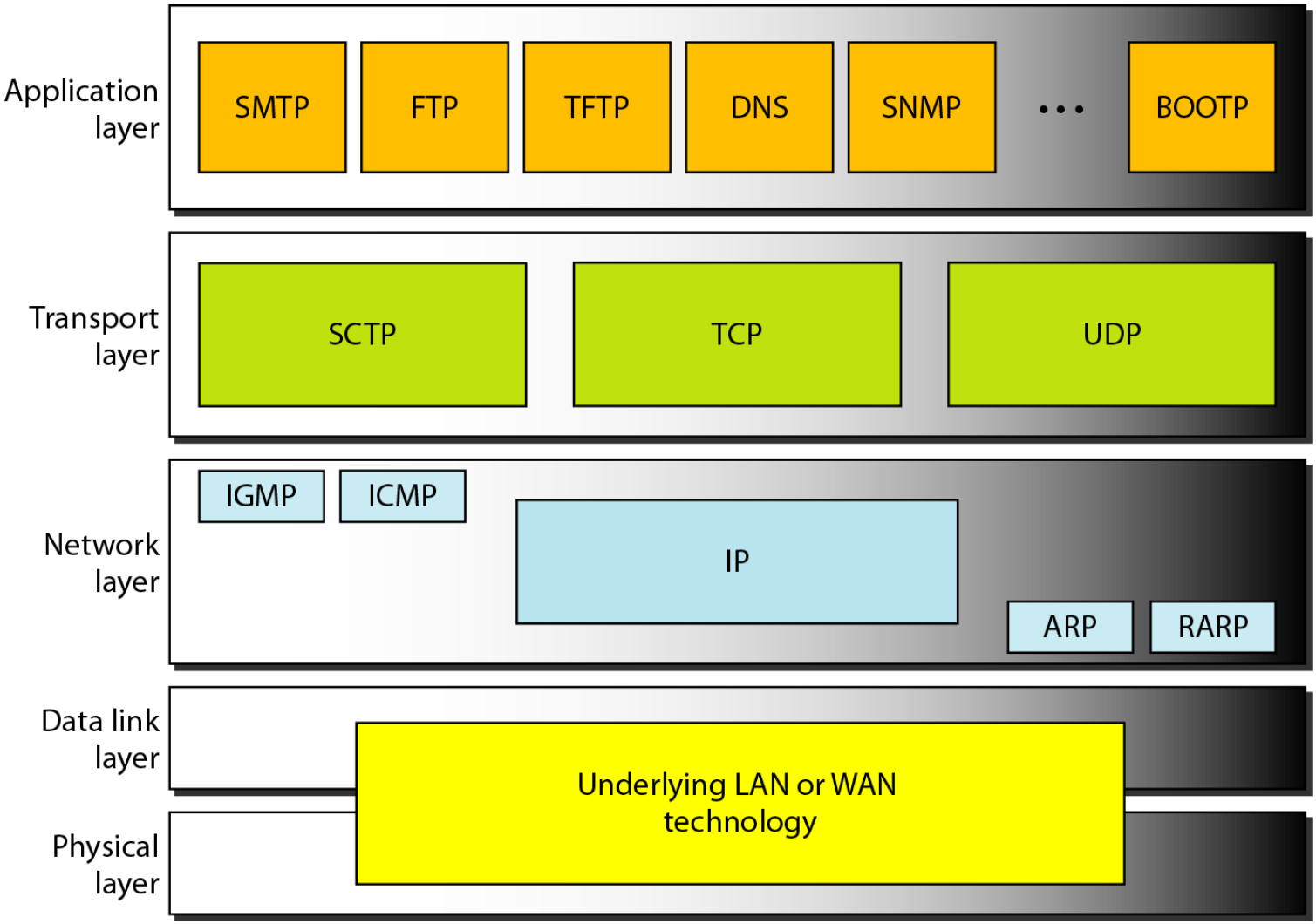
Man får bara ta bort hela sektioner med nollor en gång per paket.

Tentaexempel

Ange den kortaste formen på följande IPv6 adress:

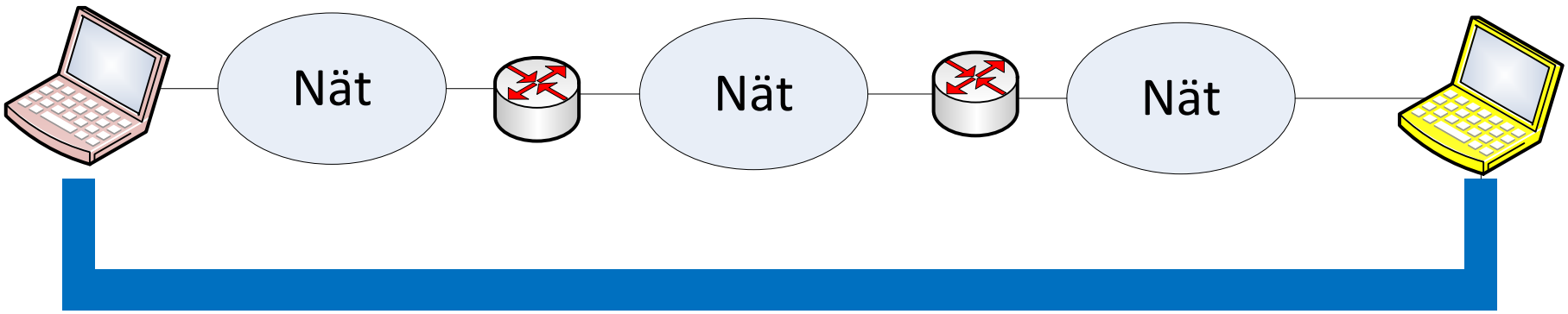
`0340:0000:0000:0000:000B:C003:0000:0234`

Transportprotokoll



Logisk förbindelse

Transportprotokollet skapar en logisk (virtual) förbindelse mellan sändare och mottagare.

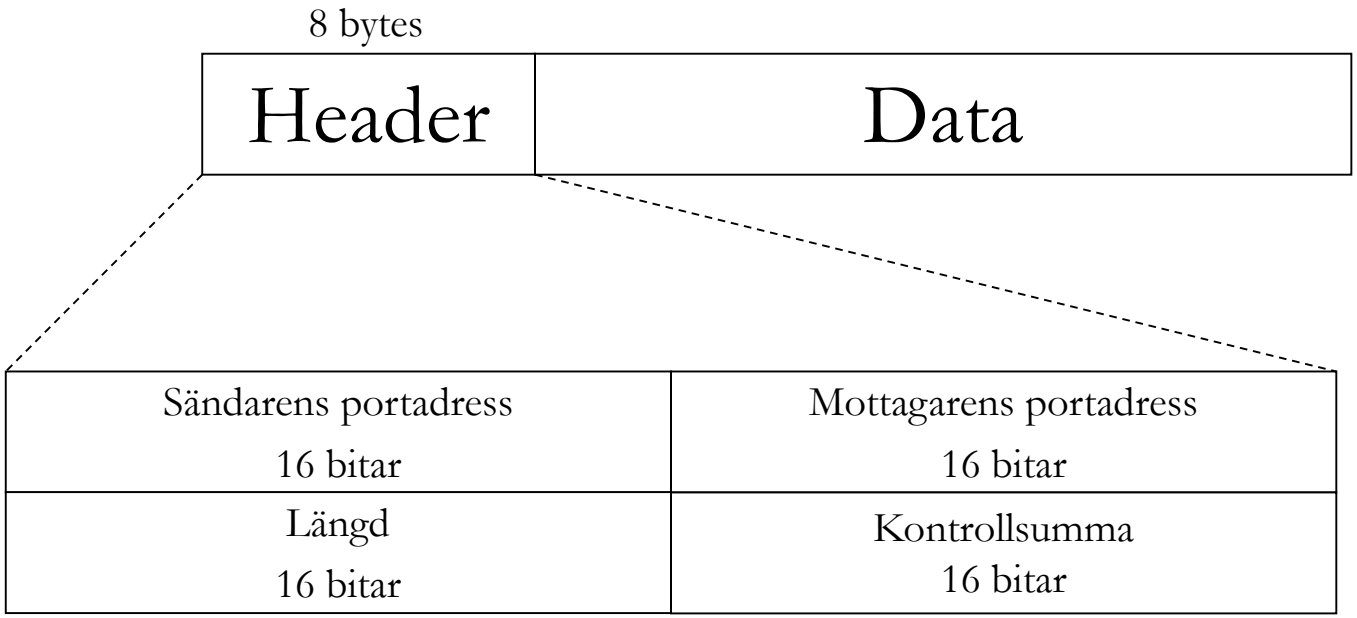


User Datagram Protocol (UDP)

UDP är ett förbindelsefritt transportprotokoll. Det enda UDP lägger till är en process-to-process kommunikation utöver IPs host-to-host kommunikation.

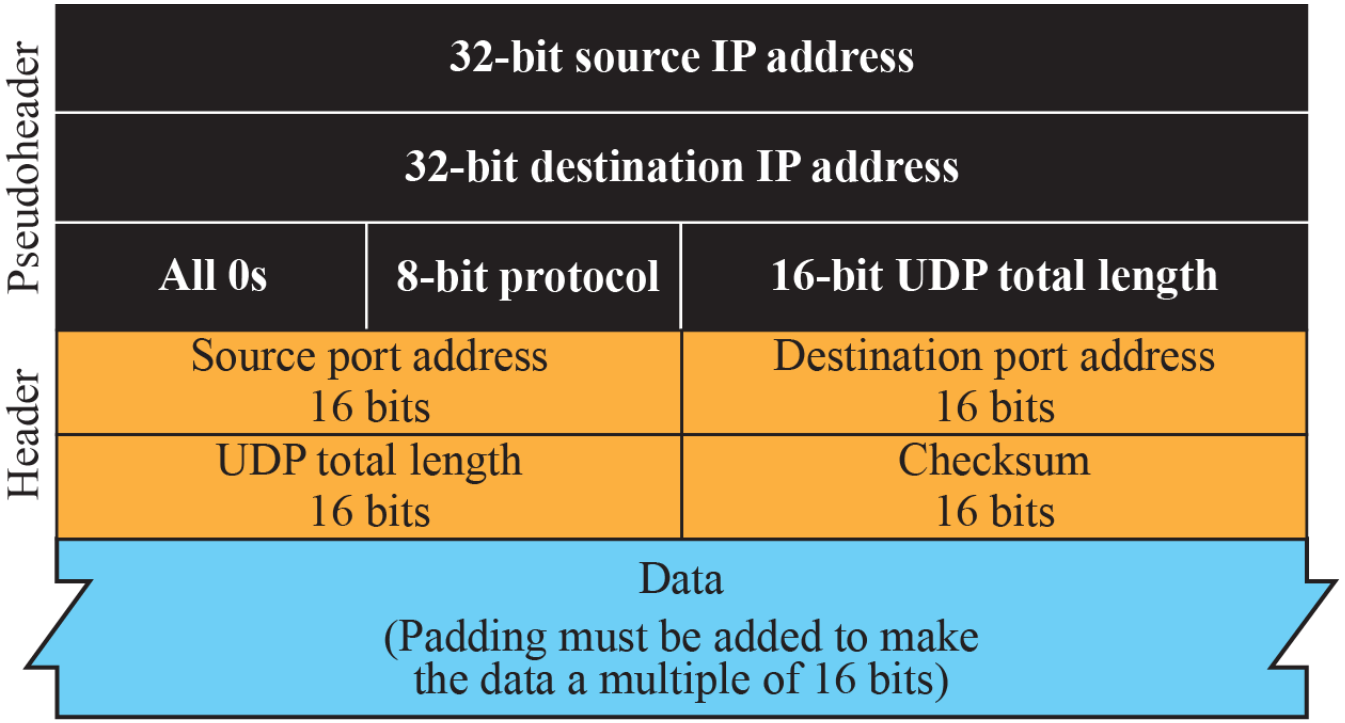
Fördel: UDP är väldigt enkelt och lägger till ett minimum av overhead.

UDP-headern


8 bytes
Header
Data
Sändarens portadress
16 bitar
Mottagarens portadress
16 bitar
Längd
16 bitar
Kontrollsumma
16 bitar

Checksum

UDP checksum beräknas på delar av IP-headern, UDP headern och data från applikationen (i multiplar av 16 bitar)



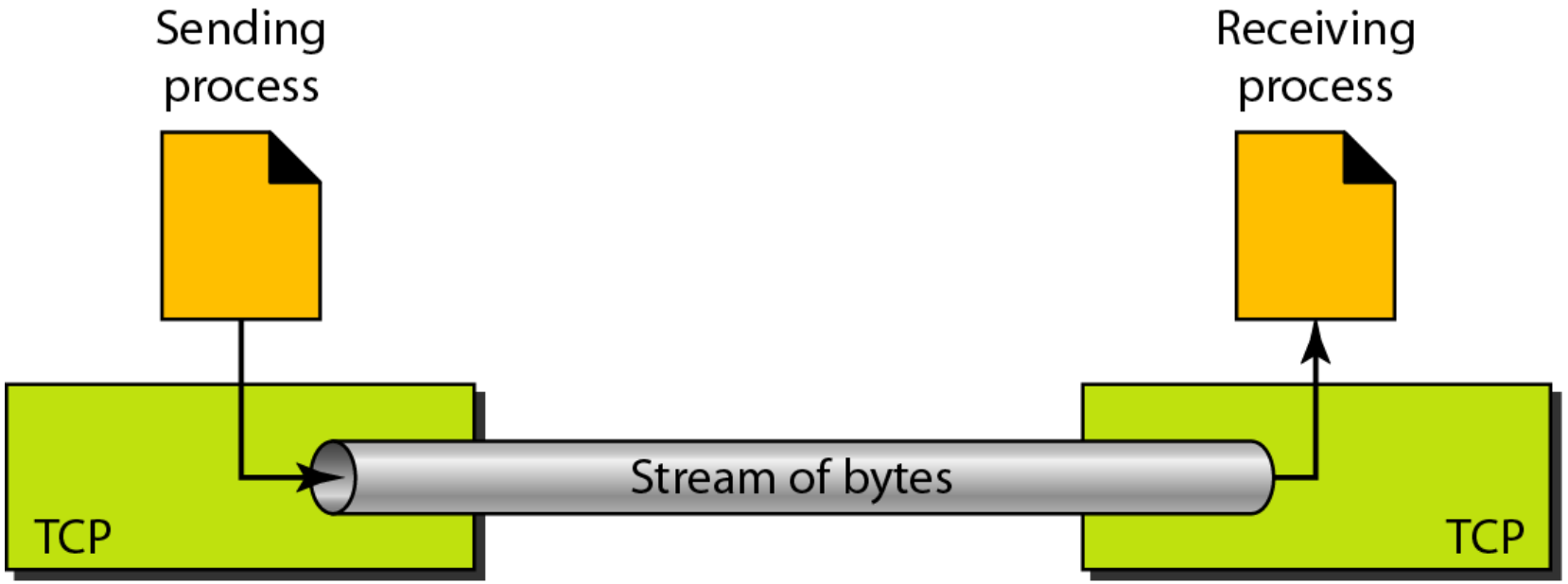
Transmission Control Protocol (TCP)

TCP är ett förbindelseorienterat transport protokoll som tillhandahåller en tillförlitlig dataöverföring.

TCP tillhandahåller funktioner för felhantering och flödeskontroll.

Stream delivery

TCP ser till att sändarens och mottagarens processer (applikation) kan skicka data som en ström av bytes.



TCP-funktioner

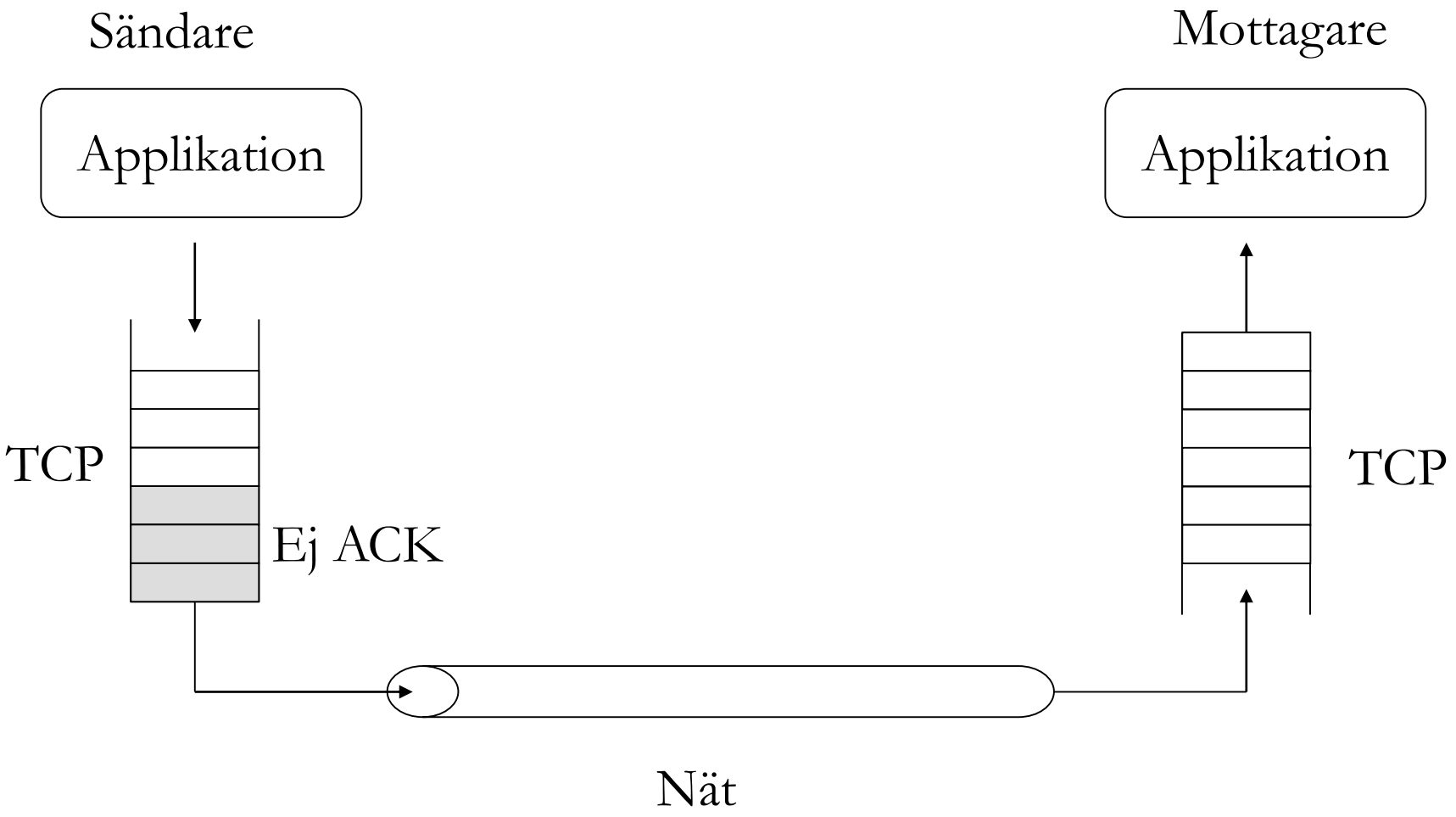
TCP packar in ett antal bytes i ett paket som kallas TCP-segment.

Både sändare och mottagare använder buffertar för att kunna genomföra felhantering och flödeskontroll.

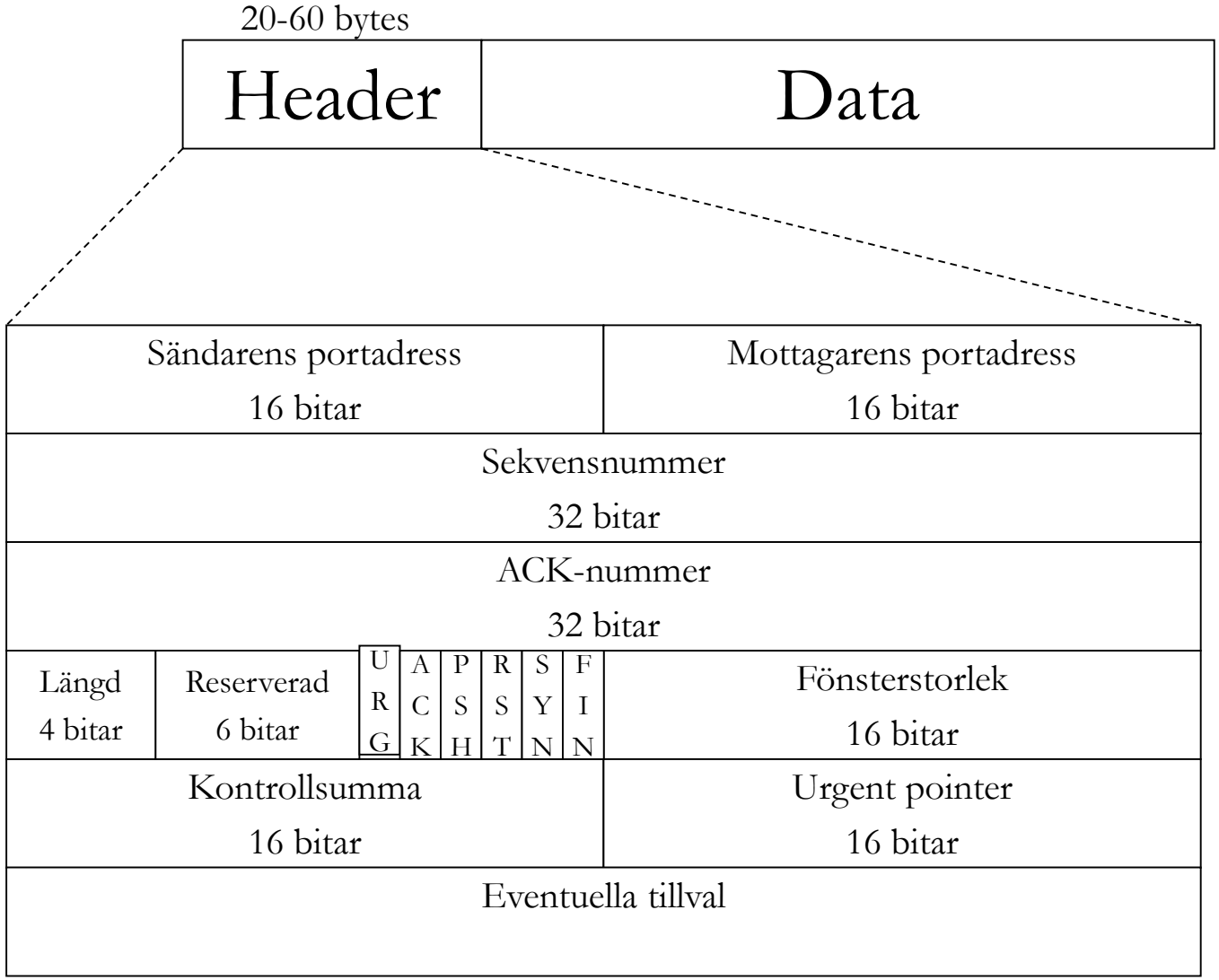
TCP innehåller ett Go-back-N protocol där sekvensnumret är den första byte som finns i segmentet. ACK:et innehåller numret på nästa byte som mottagaren förväntar sig.

ACKs kan vara piggybacked.

TCP buffert, exempel



TCP-header

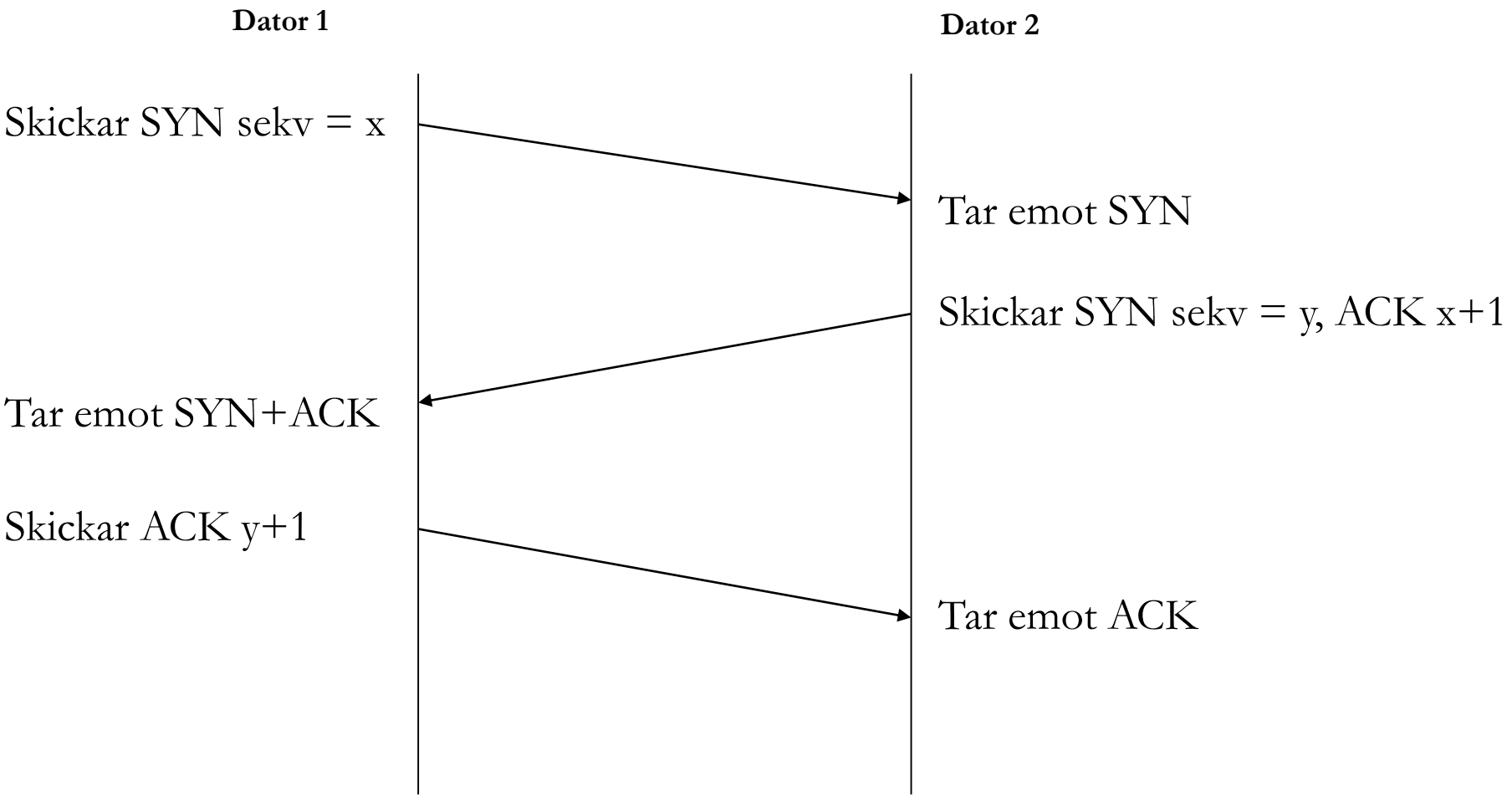

20-60 bytes
Header
Data
Sändarens portadress
16 bitar
Mottagarens portadress
16 bitar
Sekvensnummer
32 bitar
ACK-nummer
32 bitar
Längd
4 bitar
Reserverad
6 bitar
URG
ACK
PSH
RST
SYN
FIN
Fönsterstorlek
16 bitar
Kontrollsumma
16 bitar
Urgent pointer
16 bitar
Eventuella tillval

TCP sekvensnummer

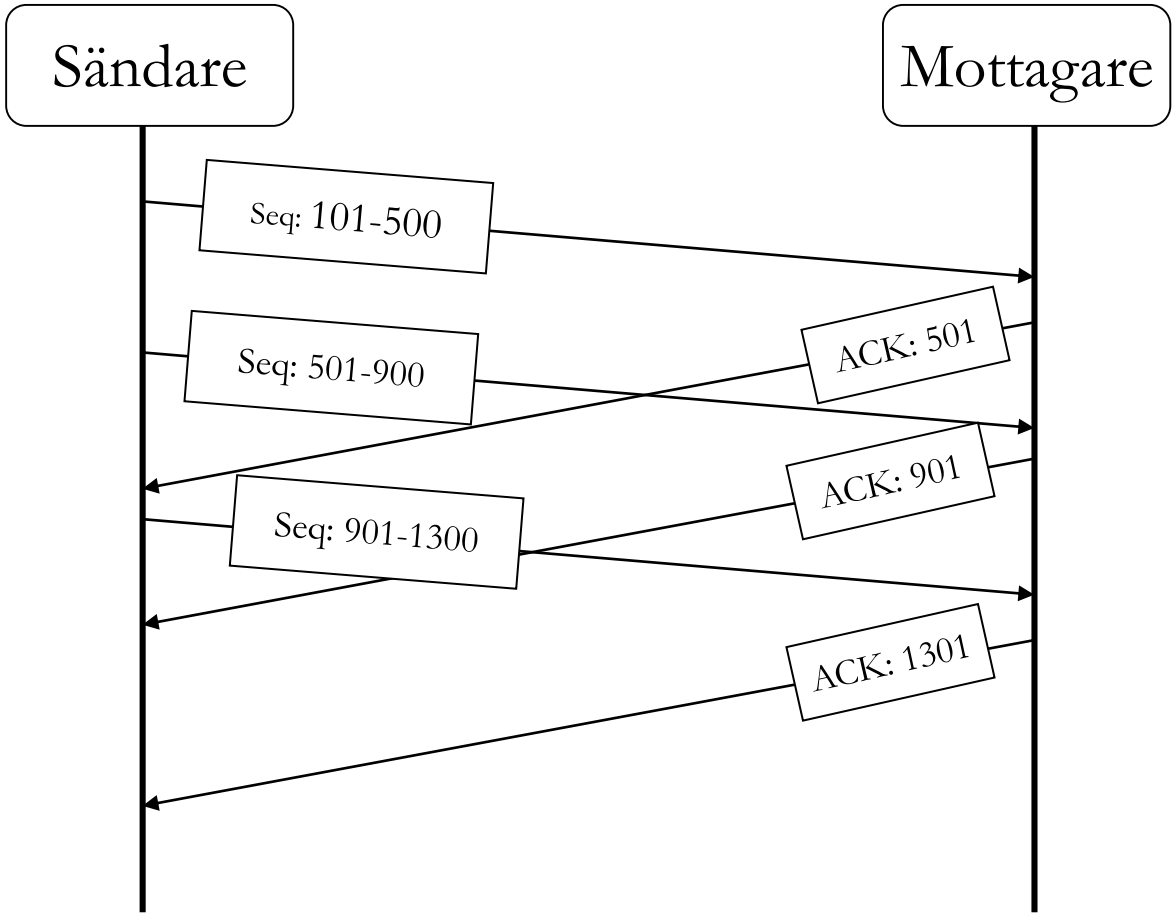
TCP tilldelar ett sekvensnummer till varje segment som skickas.

- Sekvensnummer för det första segmentet kallas Initial sequence number (ISN) och är ett slumpmässigt tal.
- Sekvensnummer för nästföljande segment är sekvensnummer för föregående segment + antal bytes som skickades i föregående segment.

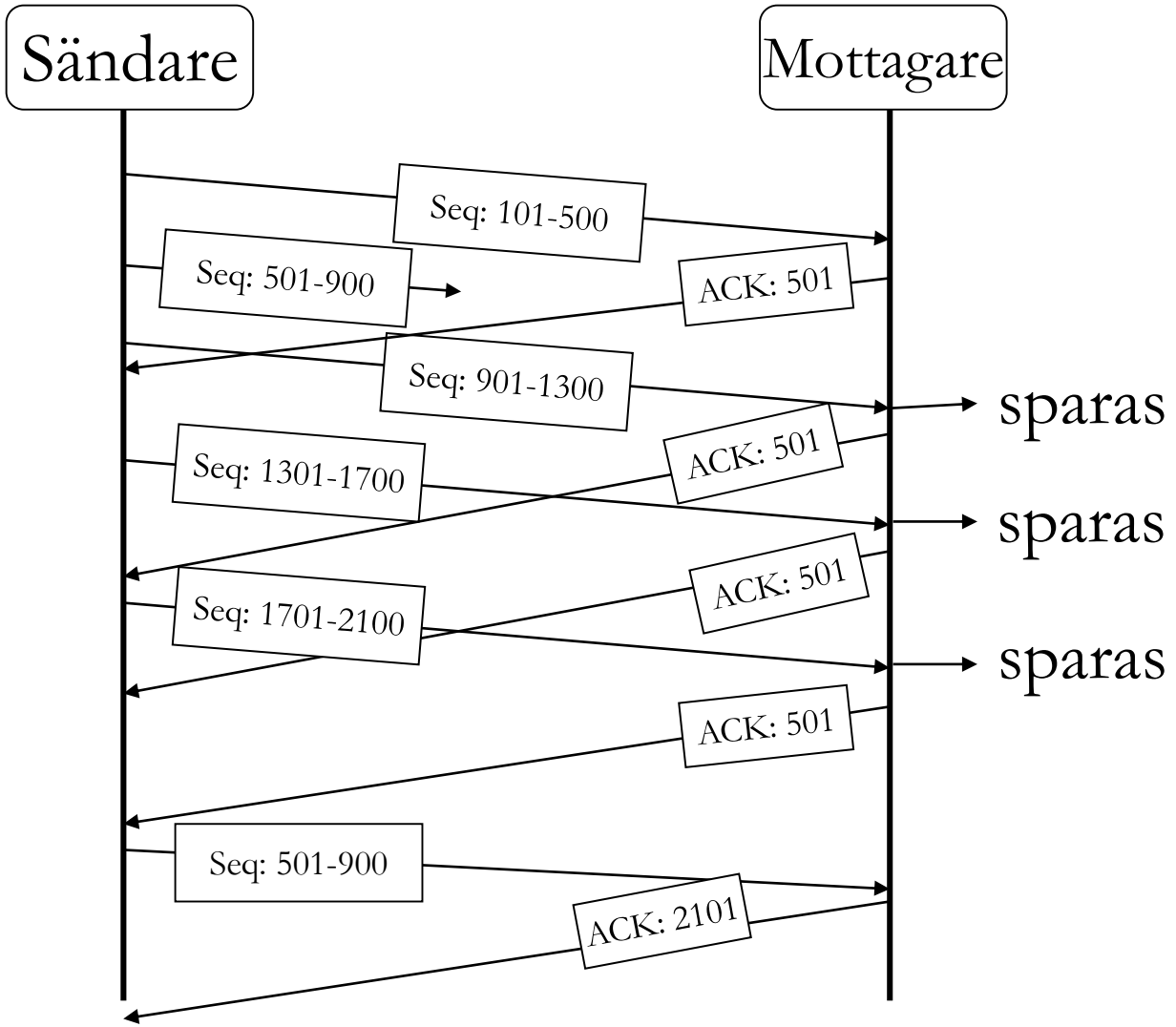
Uppkoppling av TCP-förbindelse



TCP dataöverföring om allt funkar



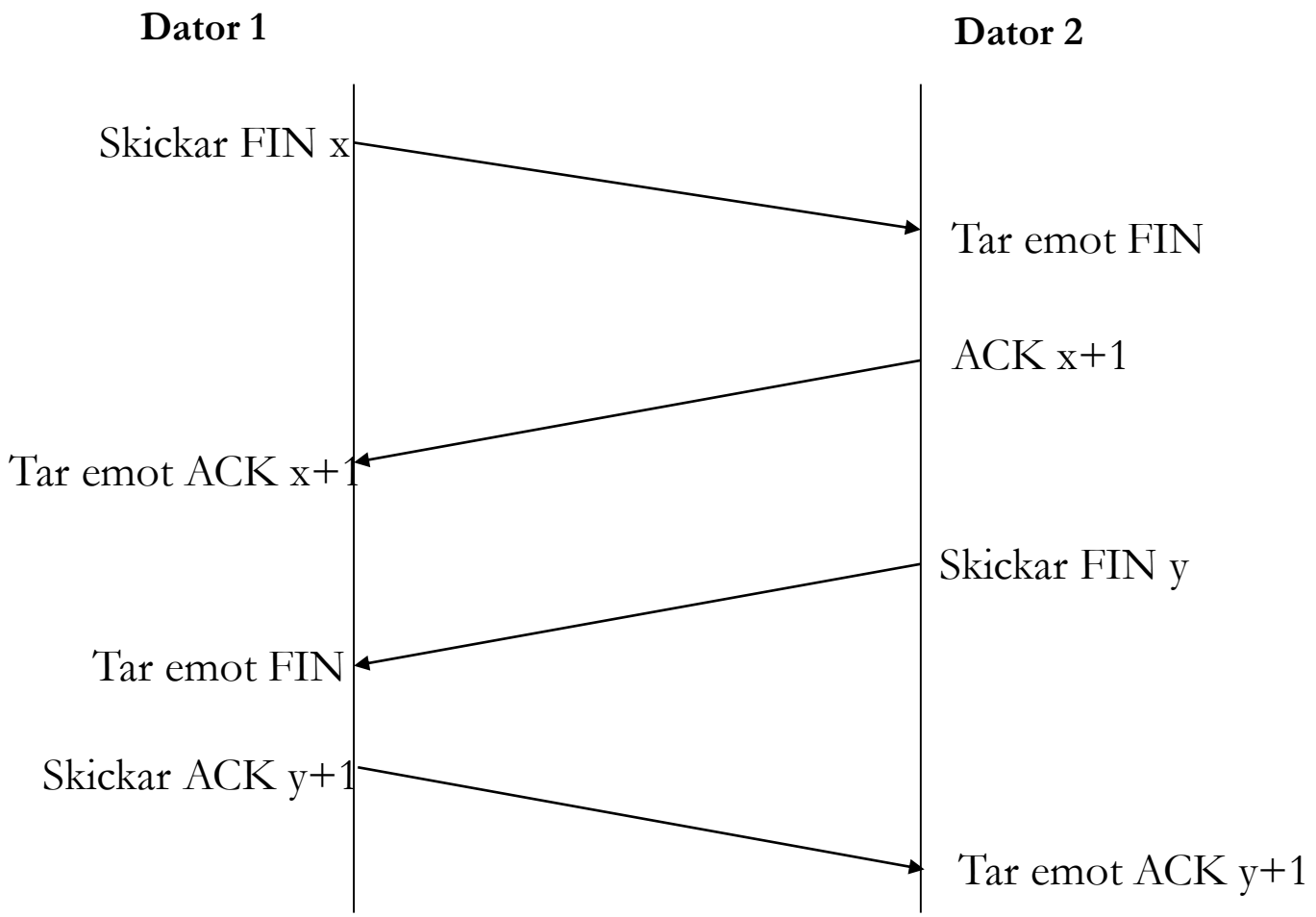
TCP omsändning



Omsändning triggas av:

1. Retransmission time-out (RTO). Dynamisk, beroende på round-trip-time.
2. Duplicerade ACK (3 stycken)

Nedkoppling av TCP-förbindelse



Flödeskontroll

- TCP har en avancerad flödes- och lastkontroll som inte ingår i denna kurs.

- De olika parametrar i Go-back-N-algoritmen är dynamiska och baseras på hur dataöverföringen fungerar.

Tentaexempel

Följande Ethernet-ram bär ett TCP-segment (Preamble, SFD och CRC borttagna). Vad är destinationens portnummer?

```
00 00 0c 07 ac 01 00 08 74 41 af a7 08 00 45 00
00 30 88 14 40 00 80 06 d5 dc 82 eb 12 bd 82 eb
84 43 09 93 00 17 f2 d2 7a 29 00 00 00 00 70 02
40 00 2f a2 00 00 02 04 05 b4 01 01 04 02
```