

Internetprotokollen

Maria Kihl

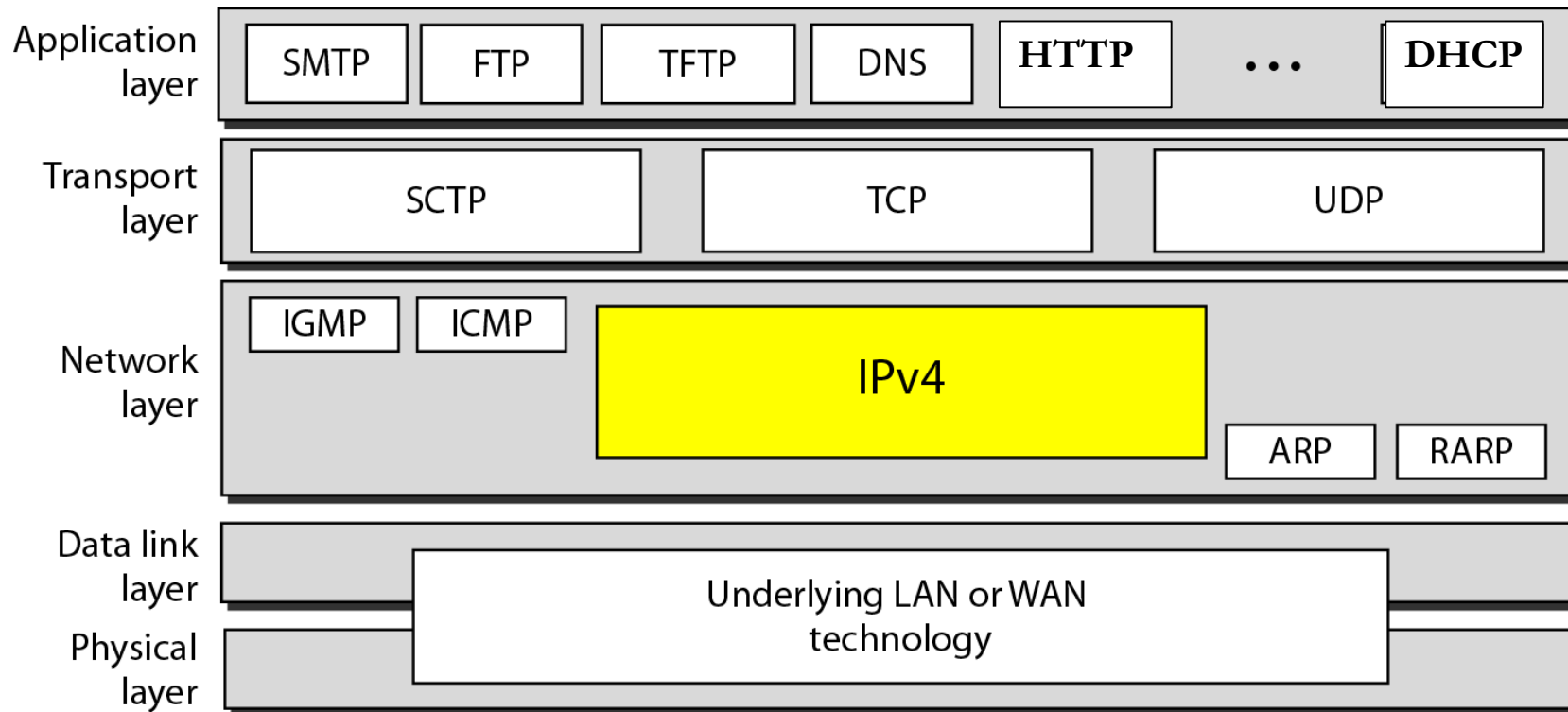


LUND
UNIVERSITY

TCP/IP-modellen

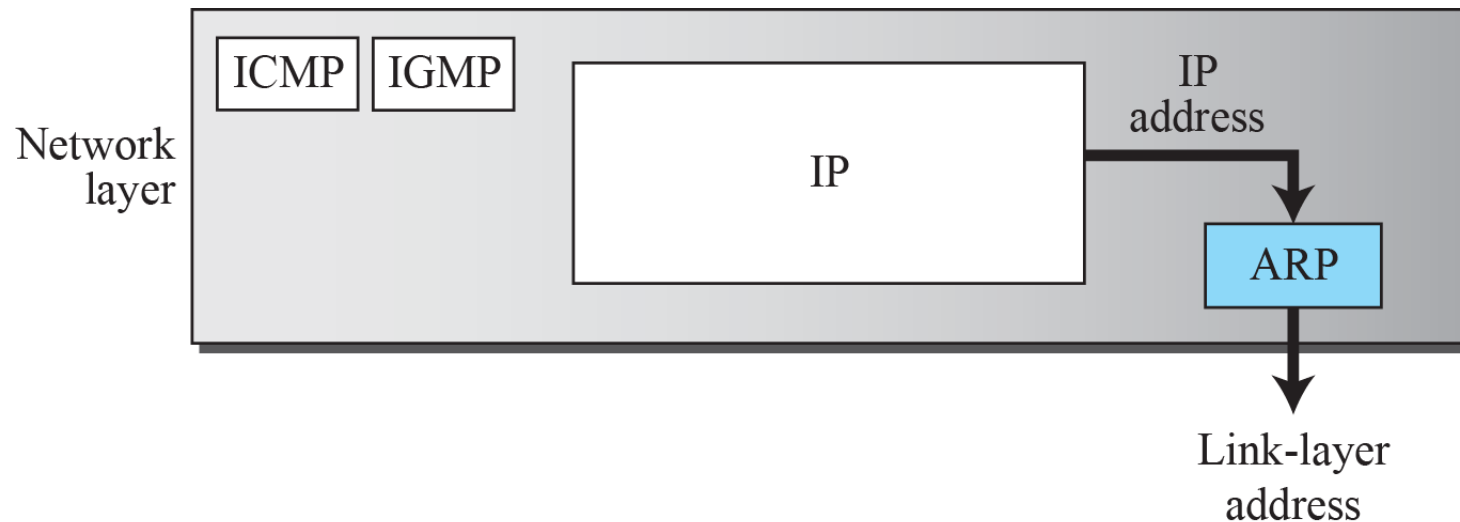
OSI-modellen	TCP/IP-modellen
Applikation	Applikation
Presentation	
Session	
Transport	Transport
Nät	Nät
Länk	Underliggande nät
Fysiska skiktet	

Internetprotokollen

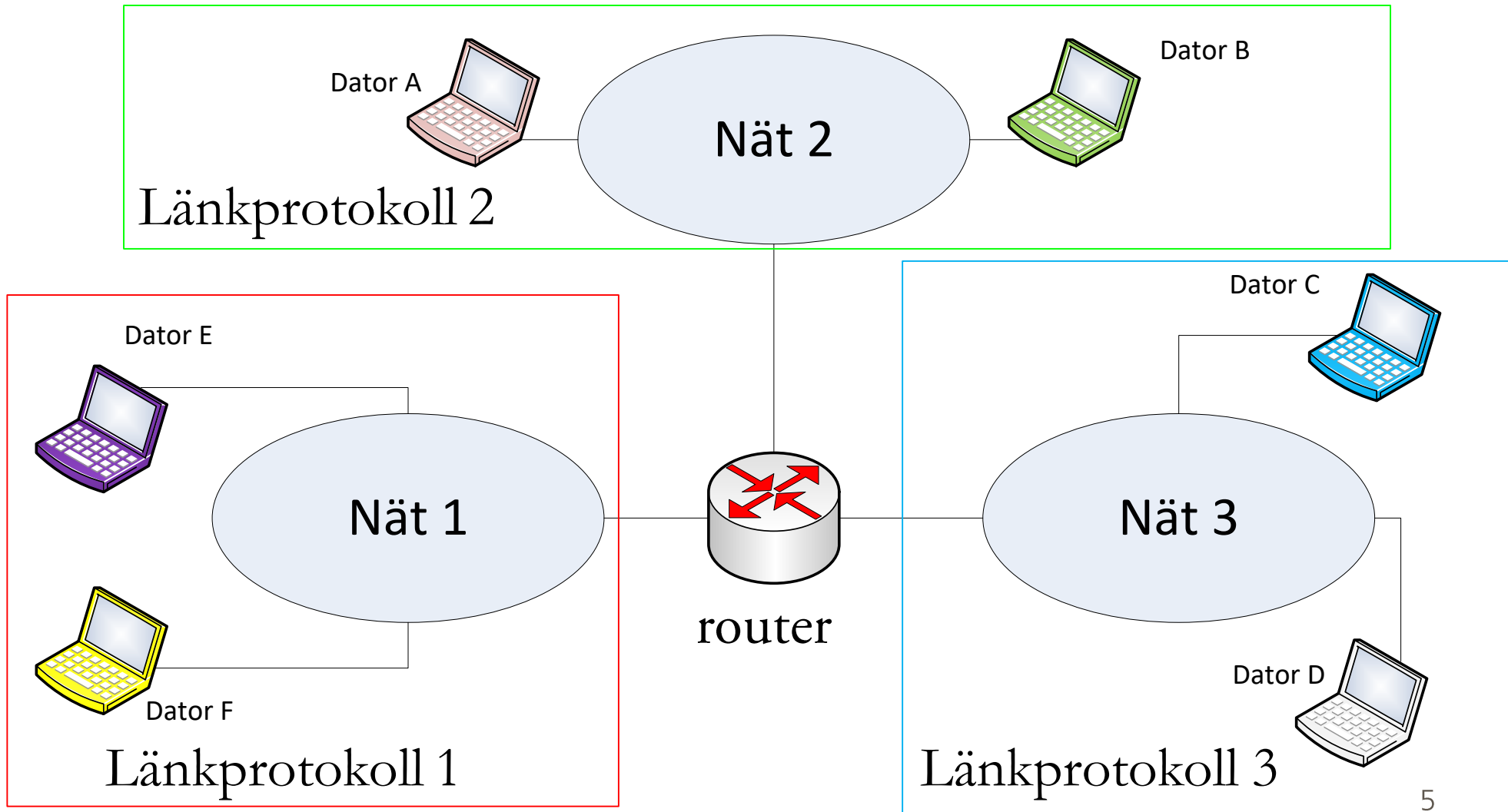


ARP i TCP/IP-modellen

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



Repetition



Repetition forts.

- 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.
- För att mappa en IP-adress med tillhörande MAC-adress (i 802-nät) används protokollet ARP.

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

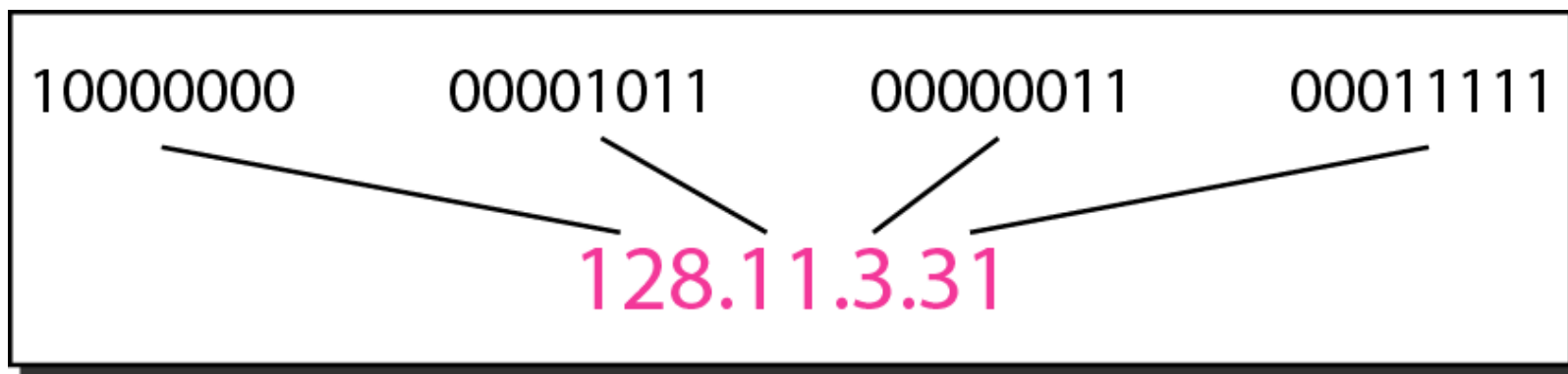
06 : 01 : 02 : 01 : 2C : 4B

6 bytes = 12 hex digits = 48 bits

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

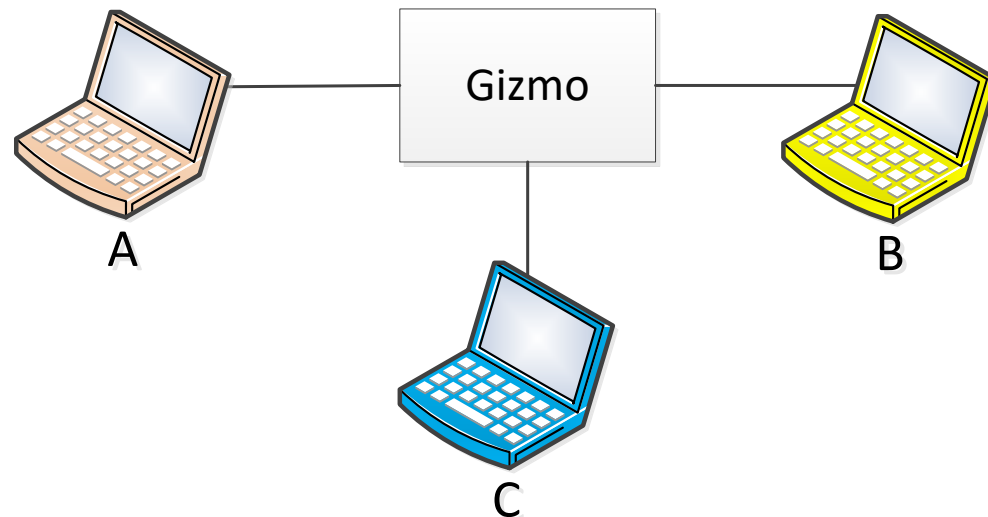
Varje värddator och routrar som är ansluten till Internet har en unik **IP-adress**. Om IPv4 används så är adressen på 32 bitar.



Adressen skrivs i så kallat **dotted-decimal format**.

Tentaexempel: Hub/Switch/Router

Antag det enkla nätet nedan. A ska skicka ett paket till C. Vilka adresser behöver A veta för att paketet ska hitta rätt om Gizmo är en (a) hub (b) switch (c) router ?

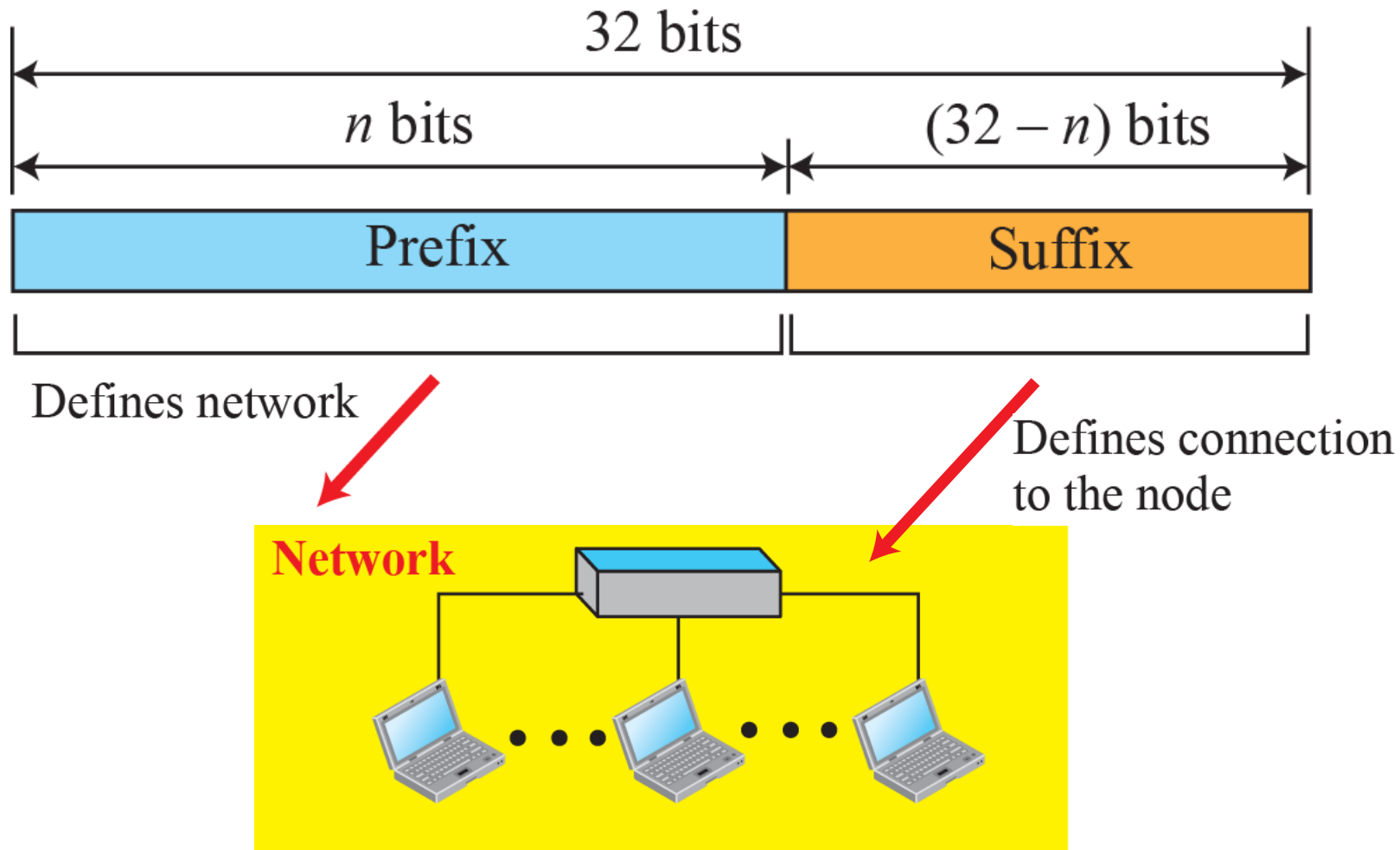


Tentaexempel

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
```

IPv4-adresser



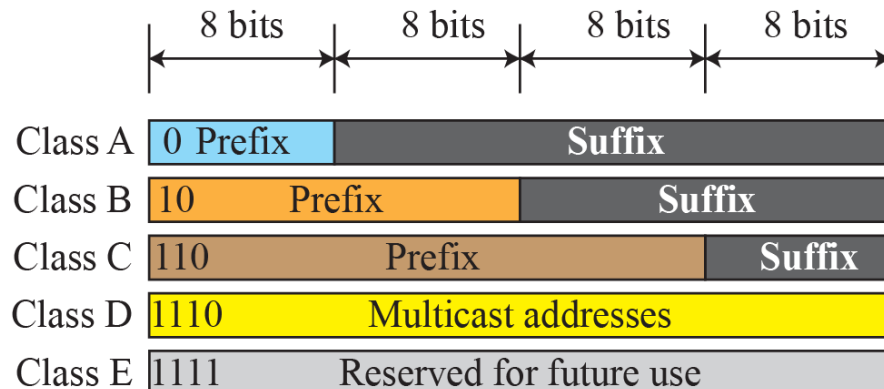
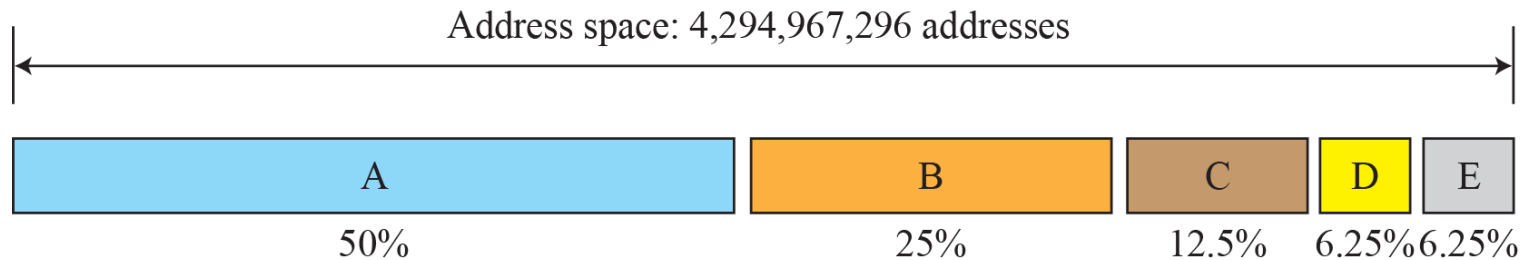
Adress-metoder för IPv4

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

Klassindelad adressering

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

<i>Class</i>	<i>Number of Blocks</i>	<i>Block Size</i>	<i>Application</i>
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

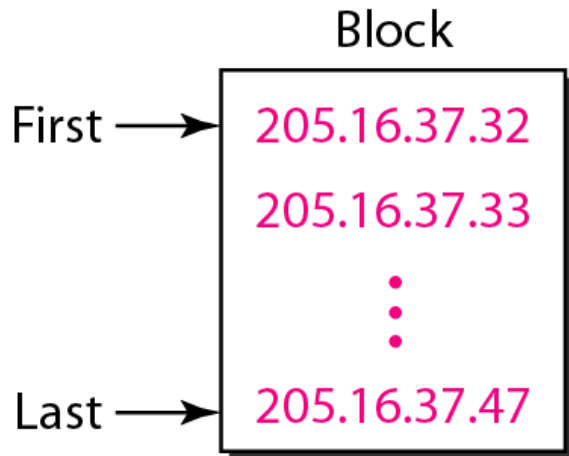
I klasslös adressering, blir en organisation (liten eller stor) tilldelad ett block med adresser, med följande restriktioner:

Adresserna i blocket måste följa på varandra.

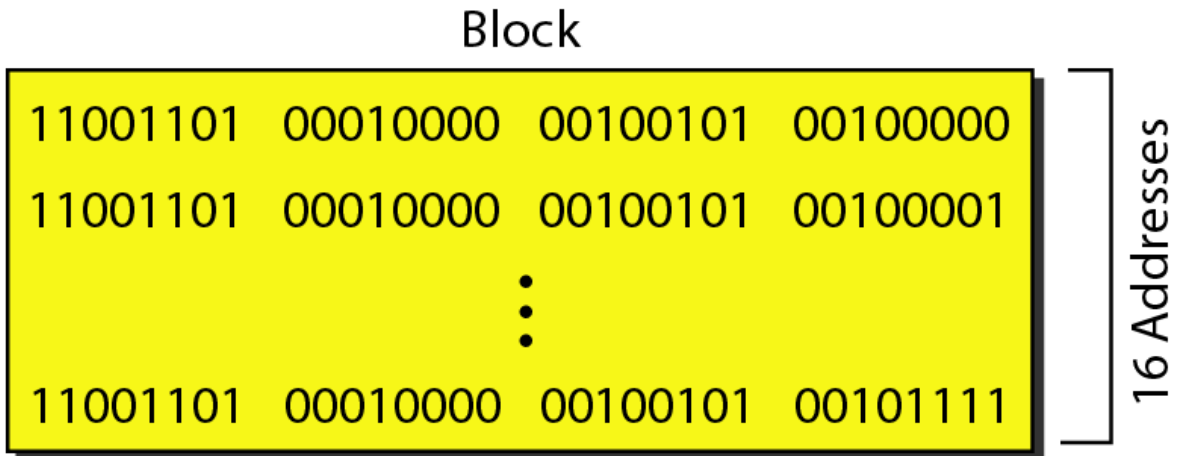
Antalet adresser måste vara av formen 2^x (1, 2, 4, 8, etc.).

Kallas även för **Classless Interdomain Routing (CIDR)**

Klasslös adressering, exempel



a. Decimal



b. Binary

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

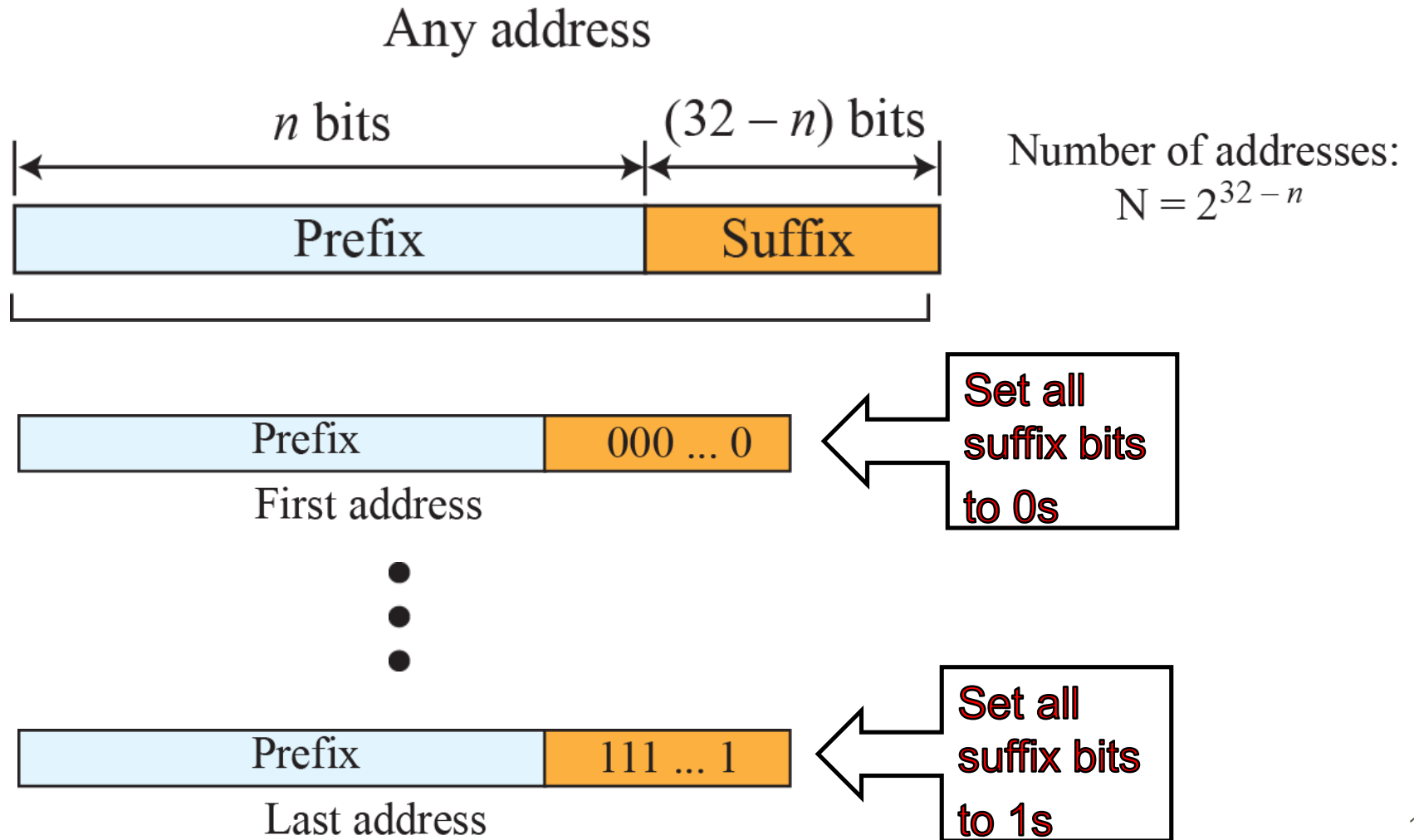
Nät-id: 11011110 00010111 01000000 00000000

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 address-block



Tentaexempel

Identifiera nät-id och värd-id för adressen

160.184.66.53/28

Identifiera även adressblocket som adressen ingår i.

Hur får en host sin IP-adress?

I denna kurs antar vi 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 (routern som kopplar till resten av Internet).

Oftast används applikationsprotokollet **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 default 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.

Exempel: ipconfig /all på min laptop på mitt kontor (kopplad till Ethernet)

Connection-specific DNS Suffix . : eit.lth.se

Description : Intel(R) Ethernet Connection (4) I219-LM

Physical Address. : 9C-7B-EF-9A-50-C8

DHCP Enabled. : Yes

Nätet använder DHCP.

Autoconfiguration Enabled : Yes

Link-local IPv6 Address : fe80::6595:206:97c4:1a89%18(Preferred)

IPv4 Address. : 130.235.201.210(Preferred)

Subnet Mask : 255.255.252.0

Lease Obtained. : den 14 november 2019 08:50:27

När fick jag min IP-adress,

Lease Expires : den 15 november 2019 08:50:26

och hur länge gäller den.

Default Gateway : 130.235.200.1

DHCP Server : 130.235.63.228

IP-adress till min DHCP-server.

DHCPv6 IAID : 110918639

DHCPv6 Client DUID. : 00-01-00-01-24-6B-18-EE-9C-7B-EF-9A-50-C8

DNS Servers : 130.235.63.228

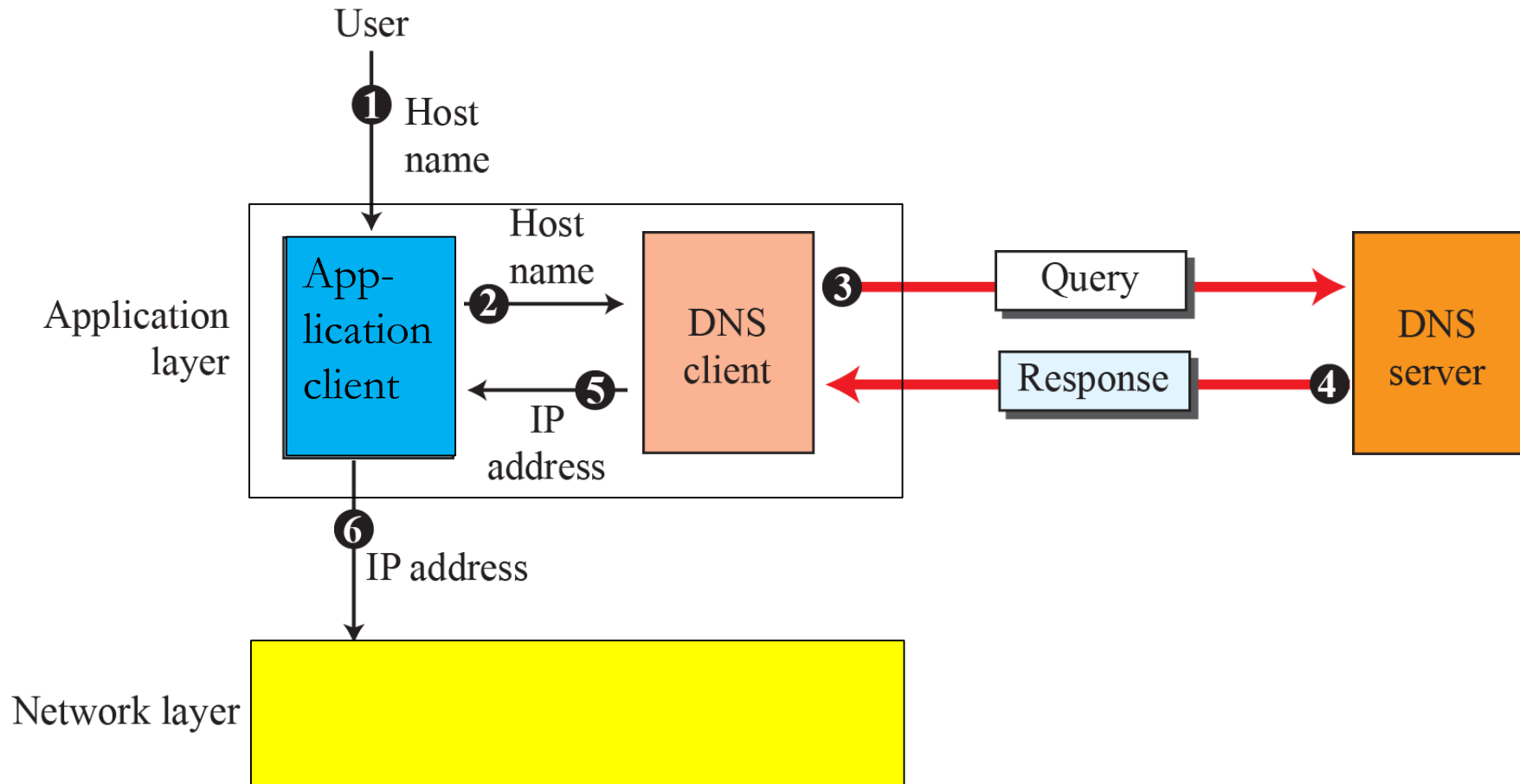
130.235.63.232

NetBIOS over Tcpip. : Enabled

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 applikationsprotokollet **Domain Name System (DNS)**.

DNS grundläggande funktion

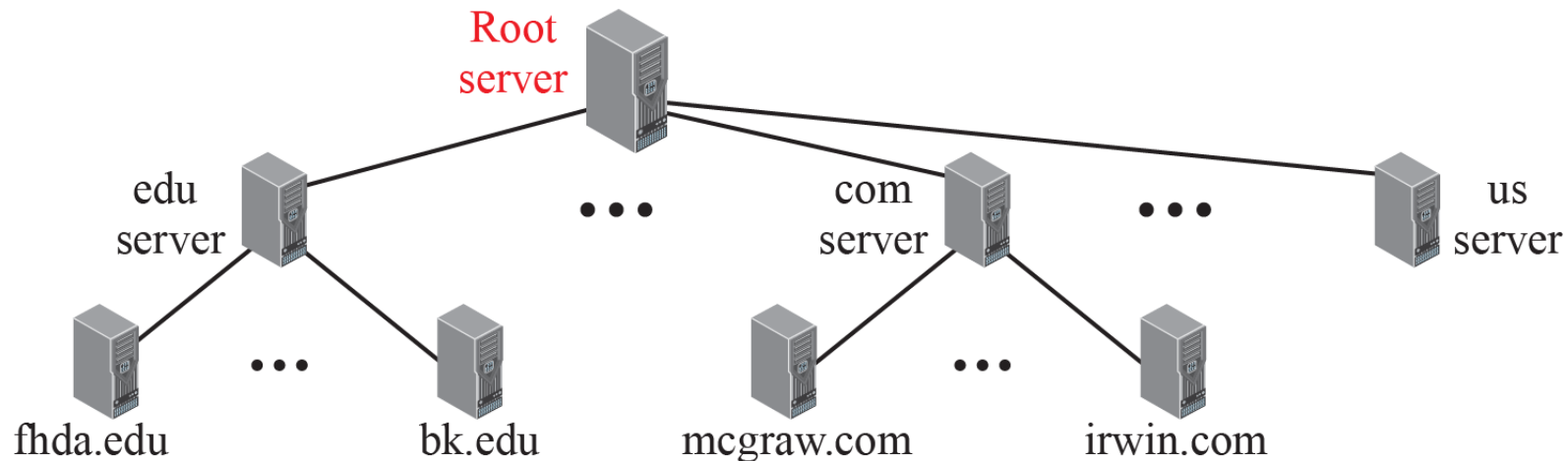


Domän-namn (Domain Name Space)

- 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.

Exempel: ipconfig /all på min laptop på mitt kontor (kopplad till Ethernet)

Connection-specific DNS Suffix . : eit.lth.se

Min dator tillhör nätet eit.lth.se

Description : Intel(R) Ethernet Connection (4) I219-LM

Physical Address. : 9C-7B-EF-9A-50-C8

DHCP Enabled. : Yes

Autoconfiguration Enabled : Yes

Link-local IPv6 Address : fe80::6595:206:97c4:1a89%18(Preferred)

IPv4 Address. : 130.235.201.210(Preferred)

Subnet Mask : 255.255.252.0

Lease Obtained. : den 14 november 2019 08:50:27

Lease Expires : den 15 november 2019 08:50:26

Default Gateway : 130.235.200.1

DHCP Server : 130.235.63.228

DHCPv6 IAID : 110918639

DHCPv6 Client DUID. : 00-01-00-01-24-6B-18-EE-9C-7B-EF-9A-50-C8

DNS Servers : 130.235.63.228

IP-adress till DNS-servers

130.235.63.232

NetBIOS over Tcpip. : Enabled

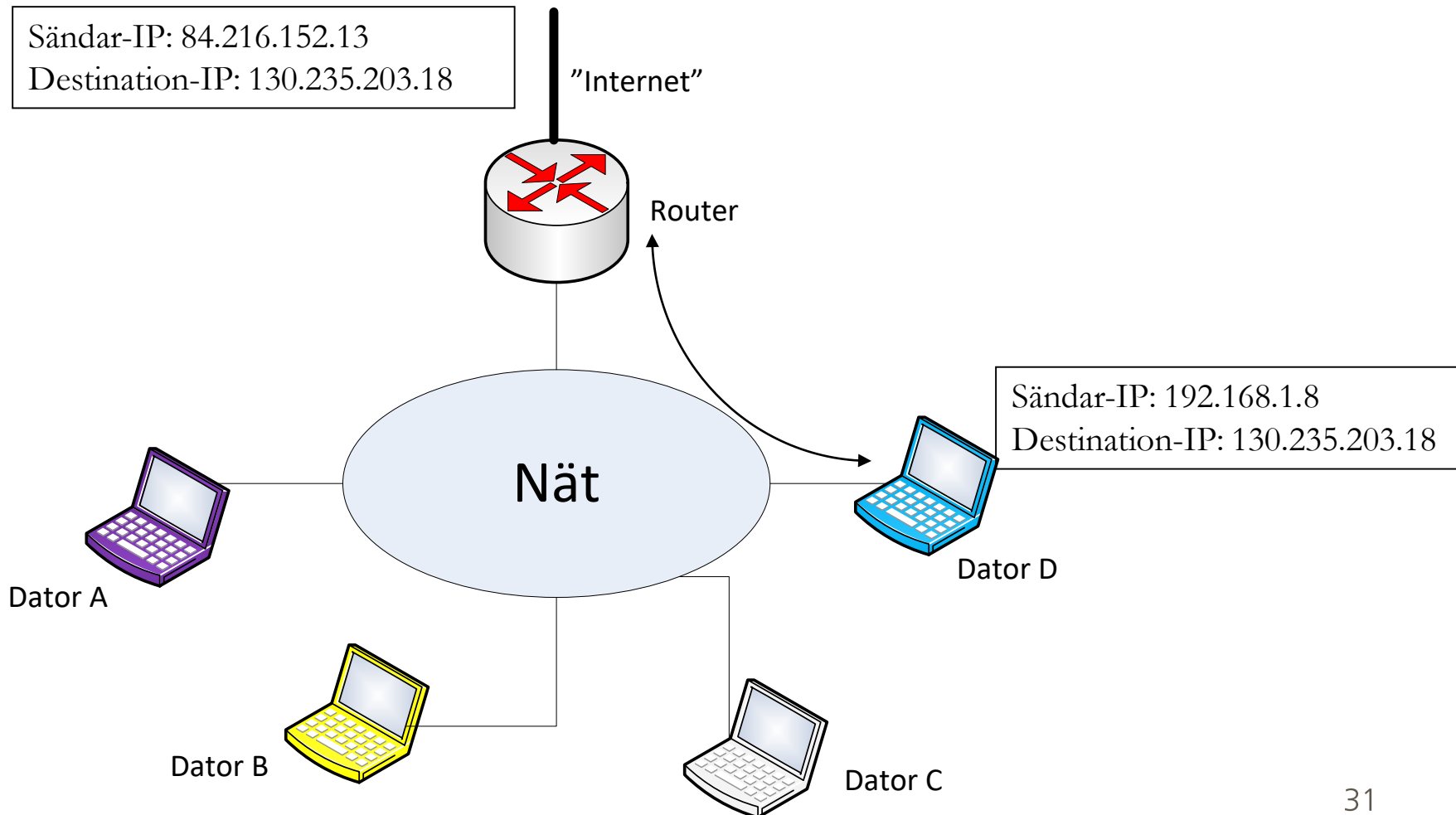
Privata och publika IP-adresser

- Idag är IPv4-adresserna slut eftersom det finns fler anslutna enheter än det finns möjliga adresser.
- Därför används ett system med **privata** och **publika** IP-adresser.
- Inom ett lokalt nät (tex hemma) kan en privat adress inom följande adressområden användas:
10.0.0.0 - 10.255.255.255 (10/8 prefix)
172.16.0.0 - 172.31.255.255 (172.16/12 prefix)
192.168.0.0 - 192.168.255.255 (192.168/16 prefix)

Privata och publika adresser forts.

- När en enhet med en privat adress ska kommunicera utanför det lokala nätet sker en adressöversättning till en publik IP-adress i routern.
- Eftersom inte alla enheter på det lokala nätet samtidigt är ute på Internet så minskas antalet publika IP-adresser.
- Denna teknik kallas **Network Address Translation (NAT)**.
- <https://whatismyipaddress.com/>

Network Address Translation (NAT)



Exempel: ipconfig /all hemma med WiFi

```
Connection-specific DNS Suffix . : lan
Description . . . . . : Intel(R) Dual Band Wireless-AC 8265
Physical Address. . . . . : 48-89-E7-1B-E4-FA
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . : Yes
Link-local IPv6 Address . . . . . : fe80::4997:ffde:f946:f6d6%6(Preferred)
IPv4 Address. . . . . : 192.168.10.144(Preferred)
Subnet Mask . . . . . : 255.255.255.0
Lease Obtained. . . . . : den 19 november 2019 08:28:50
Lease Expires . . . . . : den 19 november 2019 20:28:50
Default Gateway . . . . . : 192.168.10.1
DHCP Server . . . . . : 192.168.10.1
DHCPv6 IAID . . . . . : 105417191
DHCPv6 Client DUID. . . . . : 00-01-00-01-24-6B-18-EE-9C-7B-EF-9A-50-C8
DNS Servers . . . . . : 192.168.10.1
```


whatismyipaddress.com

My IP Address Is:


IPv4: **83.226.63.144**

IPv6: Not detected

My IP Information:

ISP: Bredbandsbolaget AB
City: Kristianstad
Region: Skåne
Country: Sweden

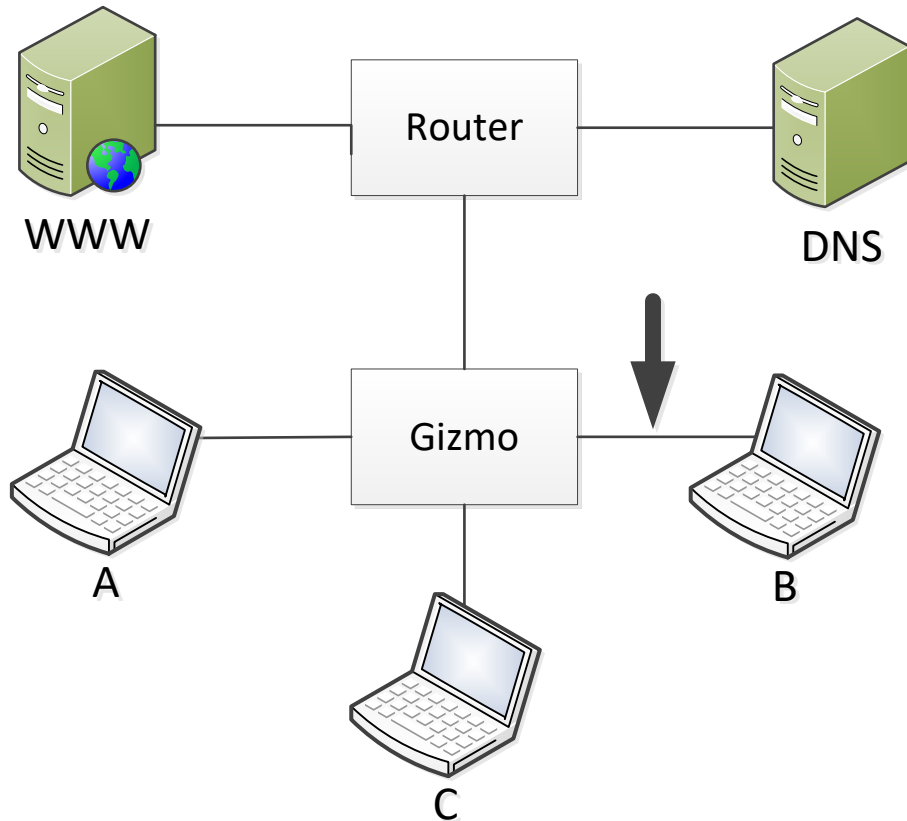
Make My IP Address Private
[Click Here](#)



Location not accurate? [Update my IP location](#)

[Show Complete IP Details](#)

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!

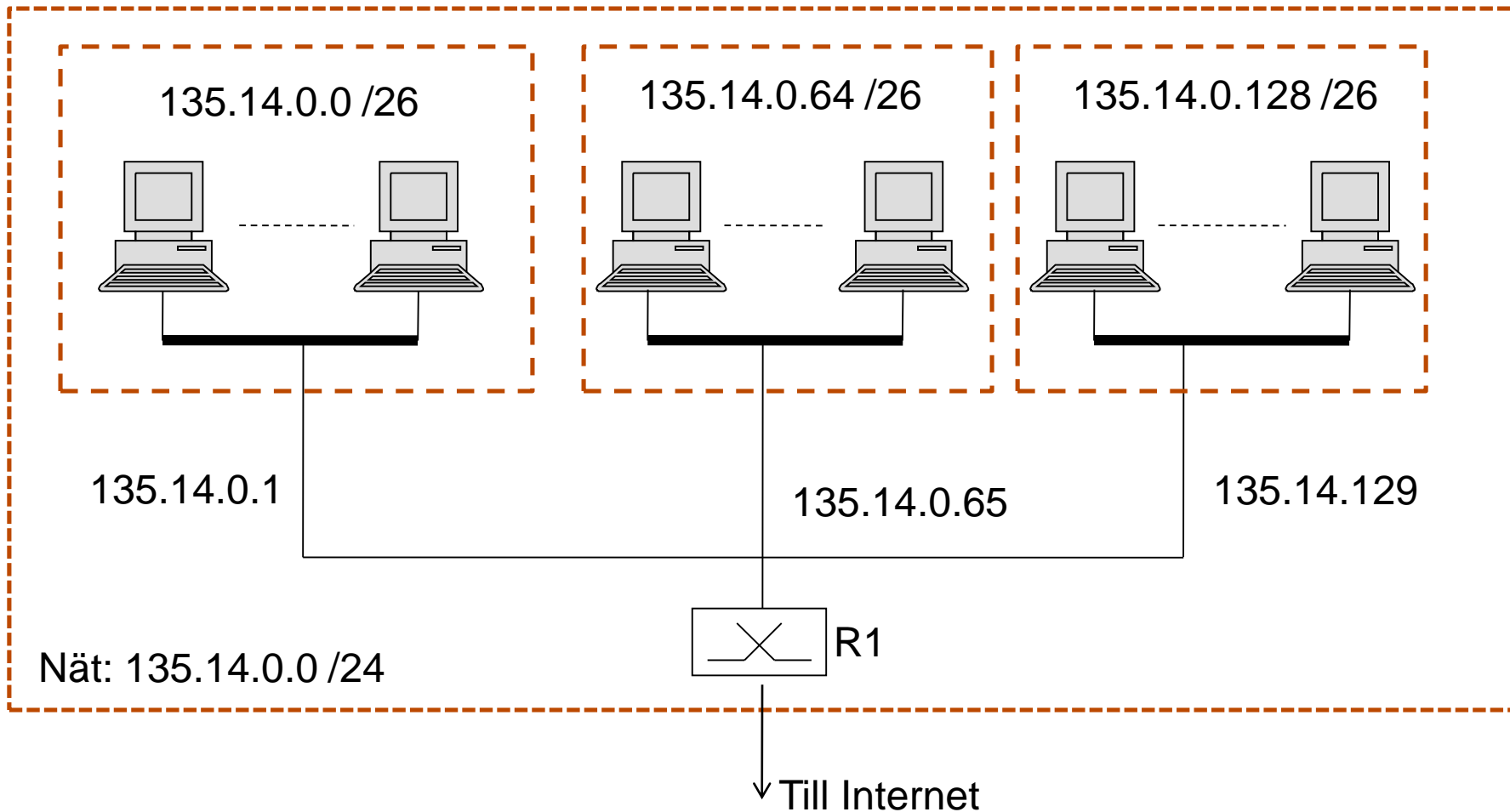
Subnetting

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

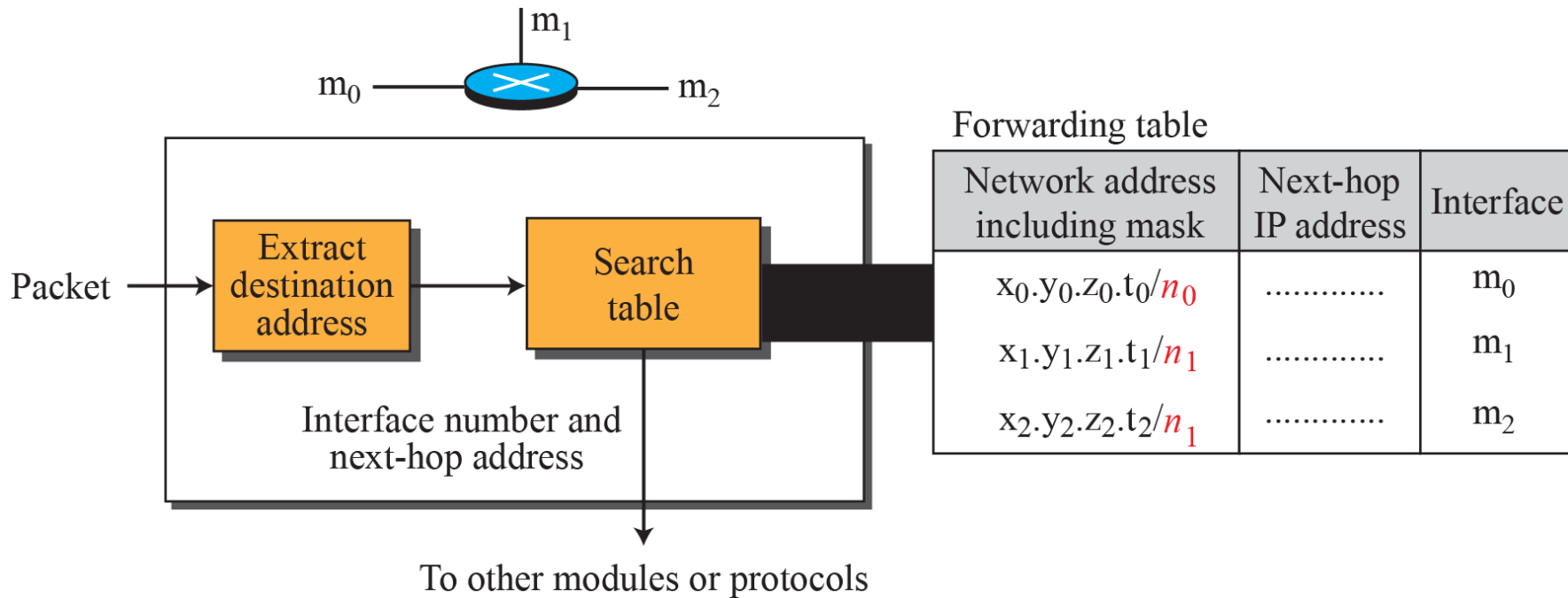
De var för stora att hantera som ett nät och för att kunna dela upp dem i flera mindre nät så infördes en adressmask.

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.

<i>Network address/mask</i>	<i>Next hop</i>	<i>Interface</i>
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

Tentaexempel

(Uppgift 6.a, 190118, värd 5p)

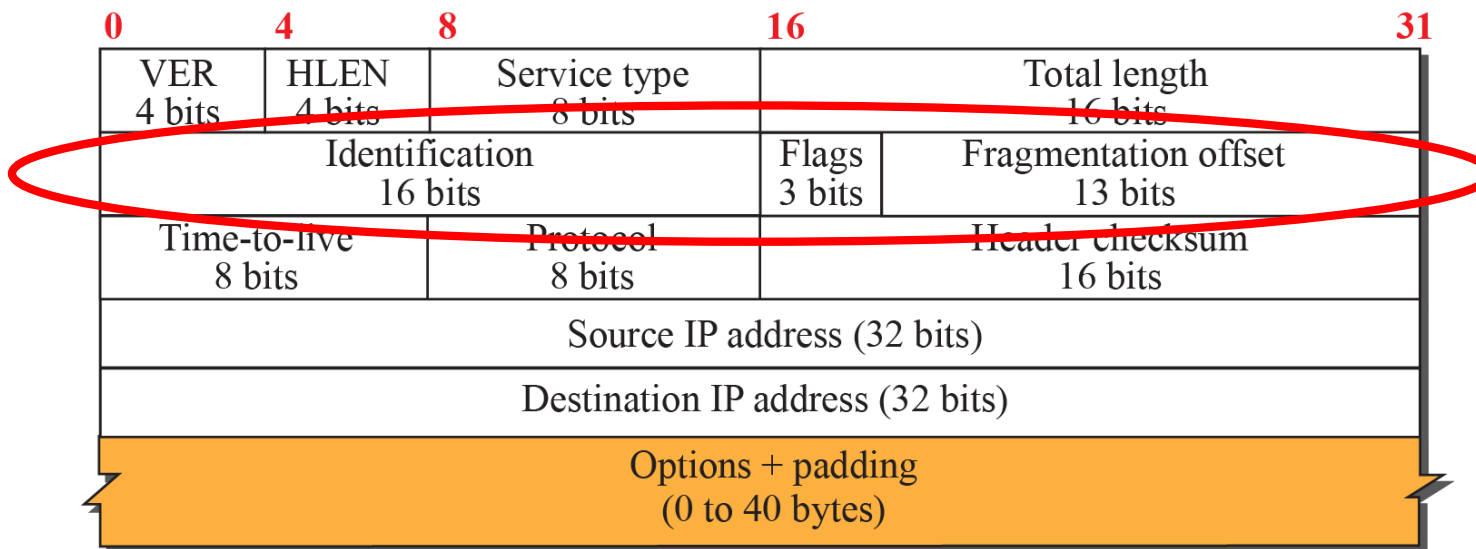
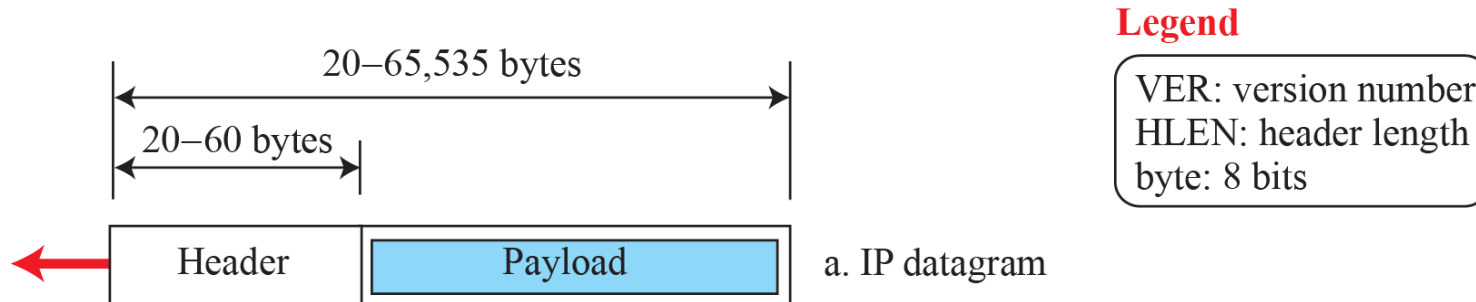
Ett block IPv4-adresser identifieras med hjälp av 123.123.128.0/18 (CIDR-beteckning). Antag att blocket delas i 8 lika stora subnät.

- (i) Vad är den gemensamma subnet-masken för alla subnäten?
- (ii) Ange adresserna till näten (med CIDR-beteckning).

Fragmentering

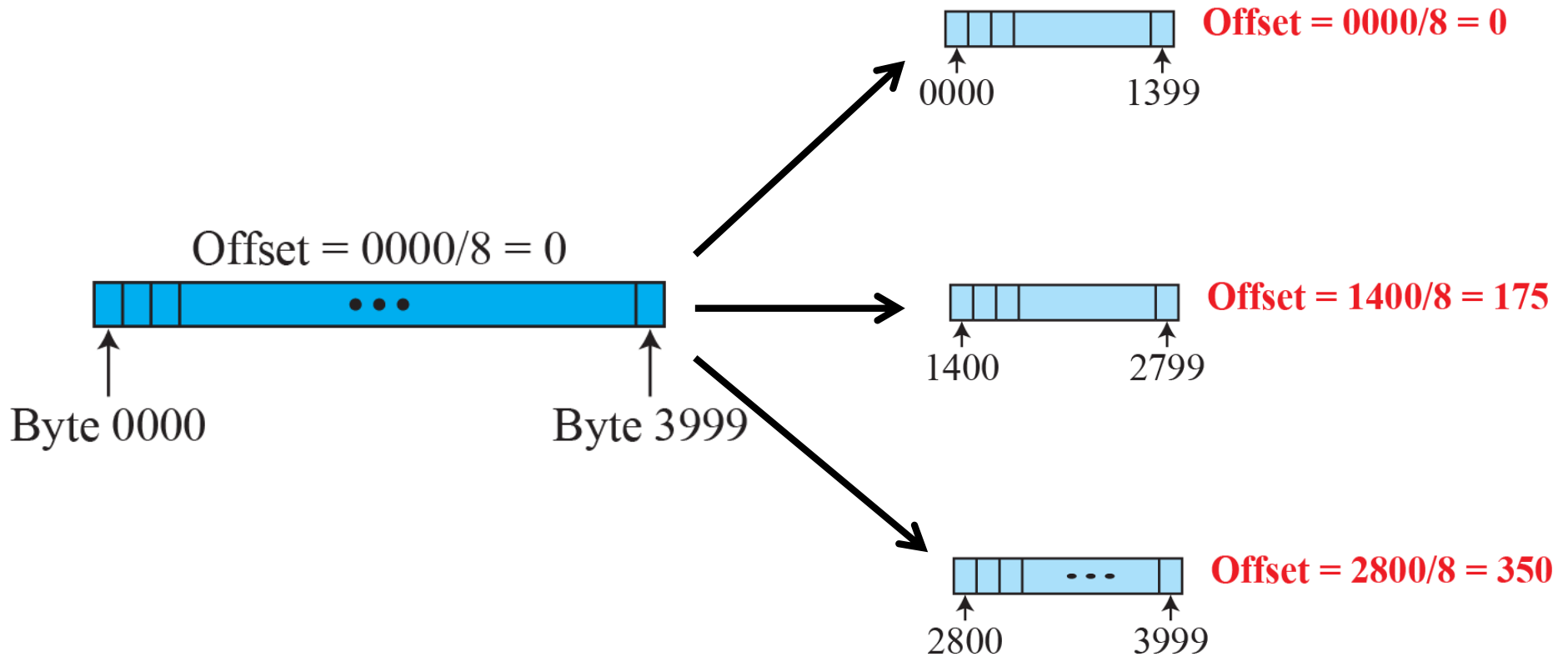
- IP tillåter 65.535 bytes data (payload). Men IP-paket kan fragmenteras om den data som kommer från transportprotokollet inte kommer att få plats i en ram på länklagret. Tex. Ethernet tillåter max 1500 bytes (IEEE 802).
- Det är sändaren som fragmenterar data, och mottagaren (den host som ska ha IP-paketet) 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 (alla tillgängliga IPv4-adresser är utdelade).
- IPv4 var inte utvecklat för realtidsapplikationer.
- IPv4 har inga funktioner för säkerhet.

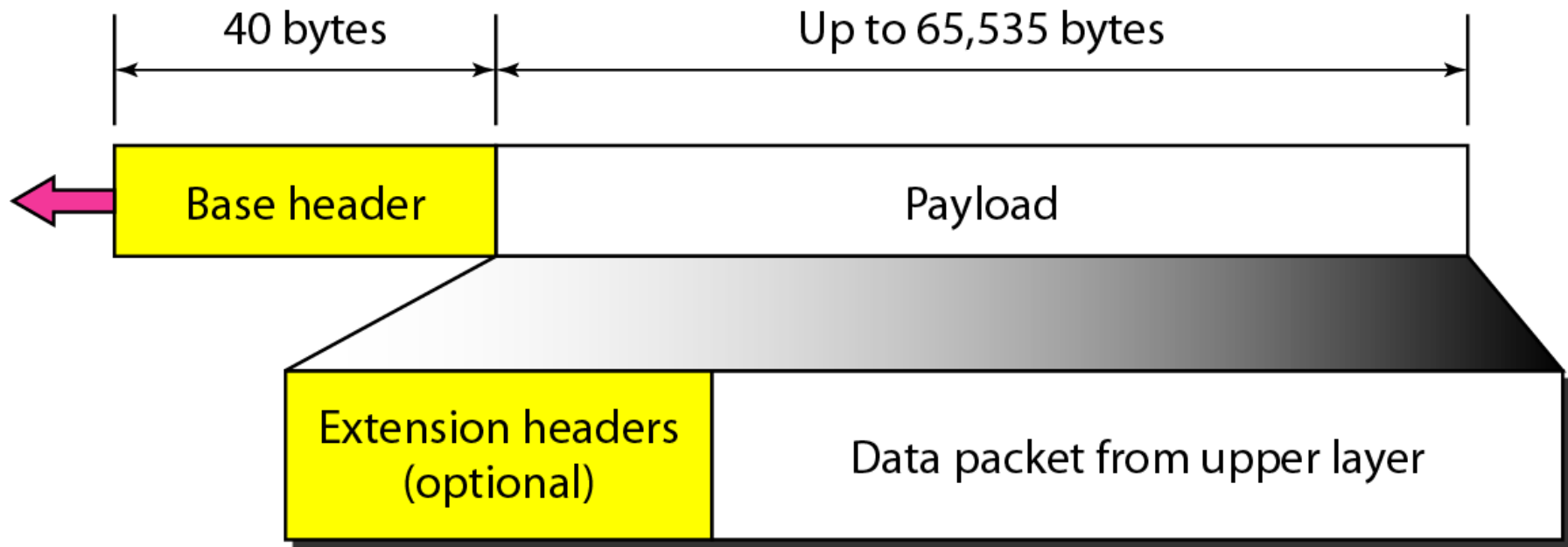
Detta är några anledningar till att **IPv6** utvecklades.

- Men övergången till IPv6 går långsamt.
- <https://computersweden.idg.se/2.2683/1.684063/ipv6-foretag>

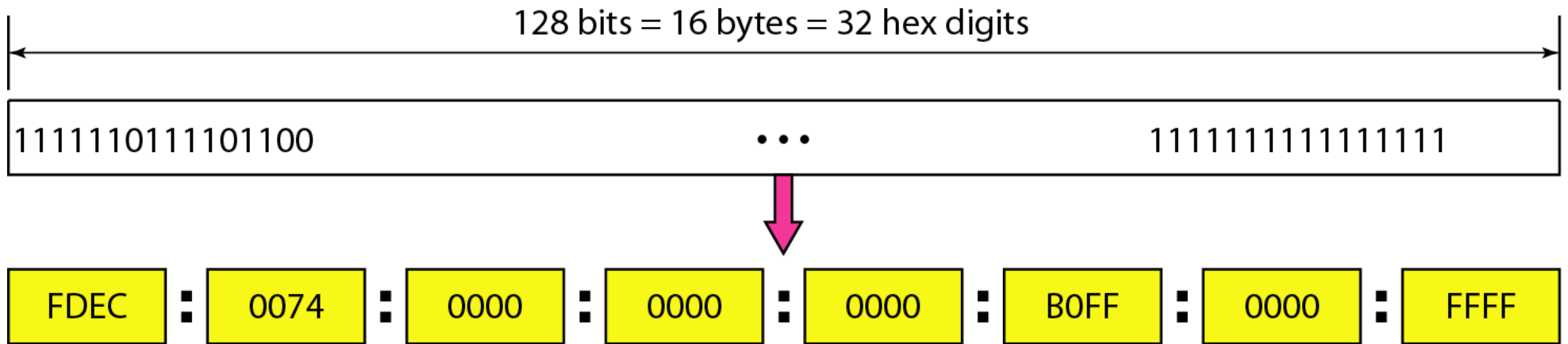
Några fördelar med IPv6

- **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

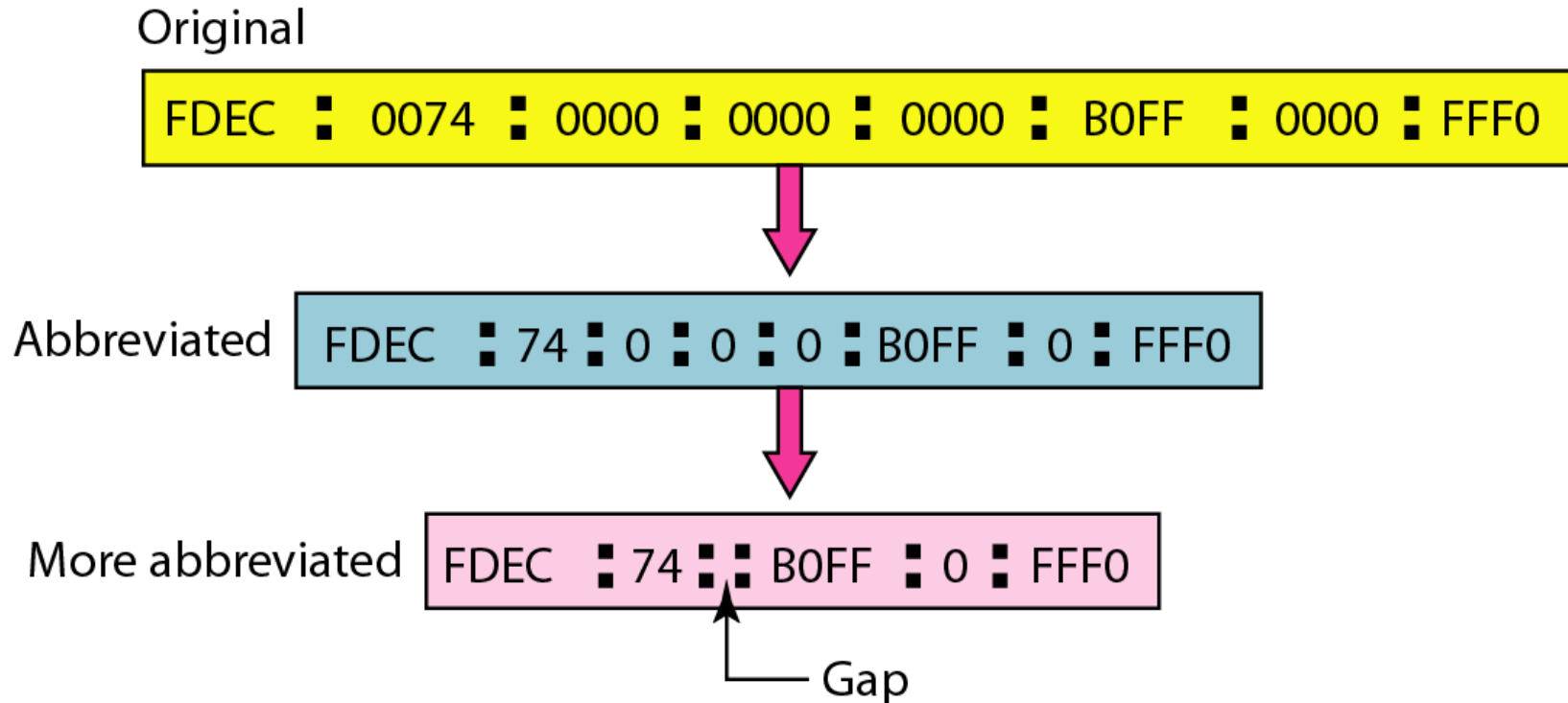


IPv6-adresser



Hexadecimal colon notation

Förkortade IPv6-adresser



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

Exempel: ipconfig /all hemma med WiFi

```
Connection-specific DNS Suffix . : lan
Description . . . . . : Intel(R) Dual Band Wireless-AC 8265
Physical Address. . . . . : 48-89-E7-1B-E4-FA
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . : Yes
Link-local IPv6 Address . . . . . : fe80::4997:ffde:f946:f6d6 %6(Preferred)
IPv4 Address. . . . . : 192.168.10.144(Preferred)
Subnet Mask . . . . . : 255.255.255.0
Lease Obtained. . . . . : den 19 november 2019 08:28:50
Lease Expires . . . . . : den 19 november 2019 20:28:50
Default Gateway . . . . . : 192.168.10.1
DHCP Server . . . . . : 192.168.10.1
DHCPv6 IAID . . . . . : 105417191
DHCPv6 Client DUID. . . . . : 00-01-00-01-24-6B-18-EE-9C-7B-EF-9A-50-C8
DNS Servers . . . . . : 192.168.10.1
```

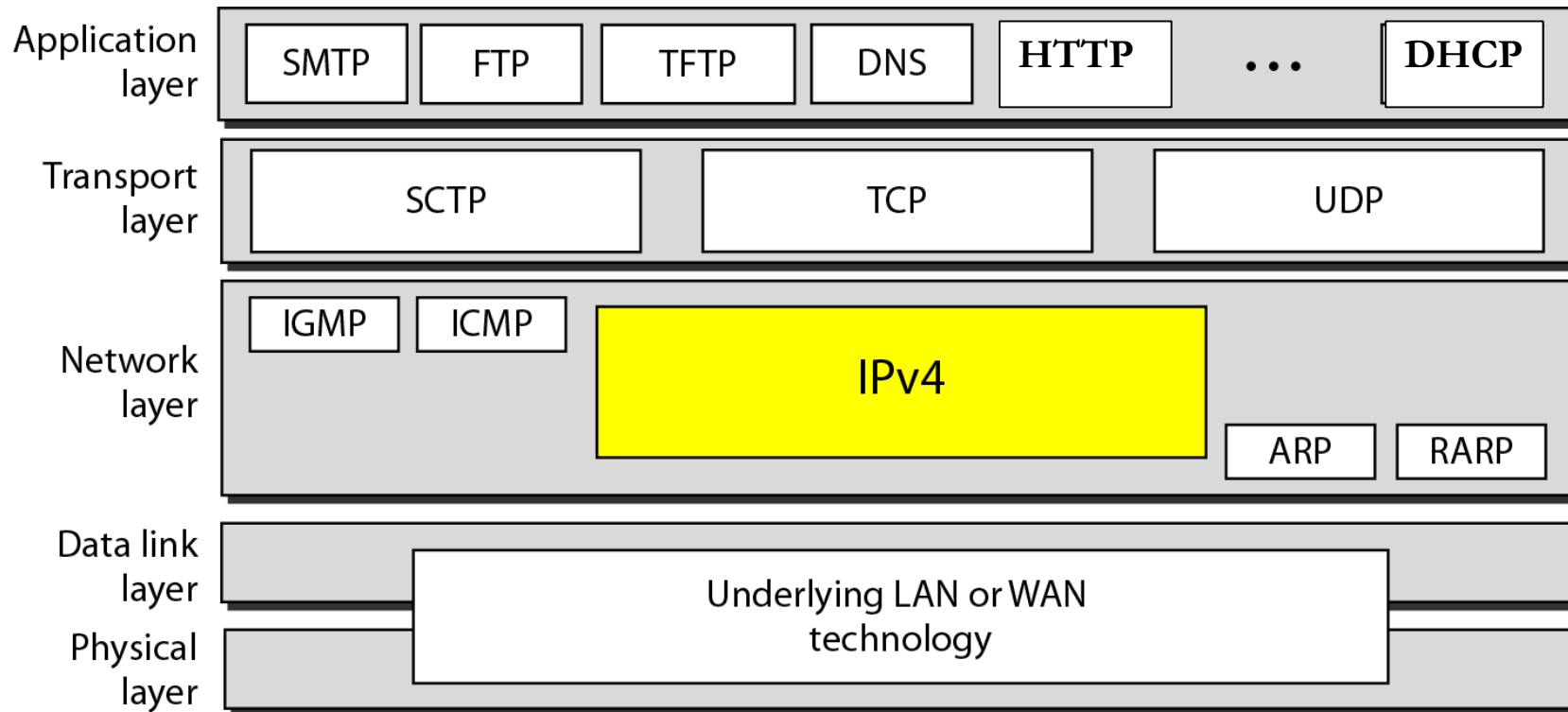
link-local => får endast användas inom det lokala nätet.

Tentaexempel

Ange den kortaste formen på följande IPv6 adress:

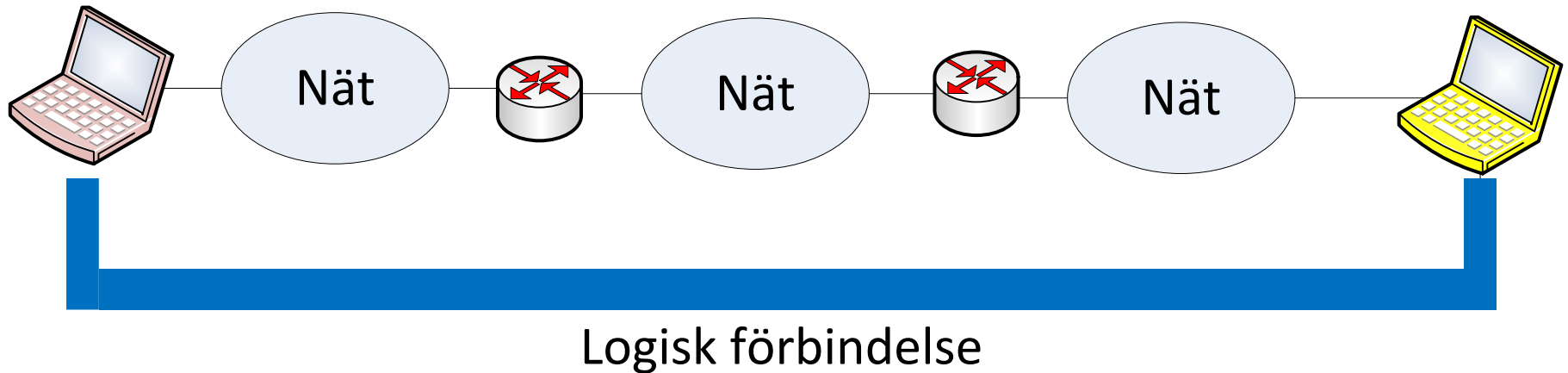
0340:0000:0000:0000:000B:C003:0000:0234

Transportprotokoll



Logisk förbindelse

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

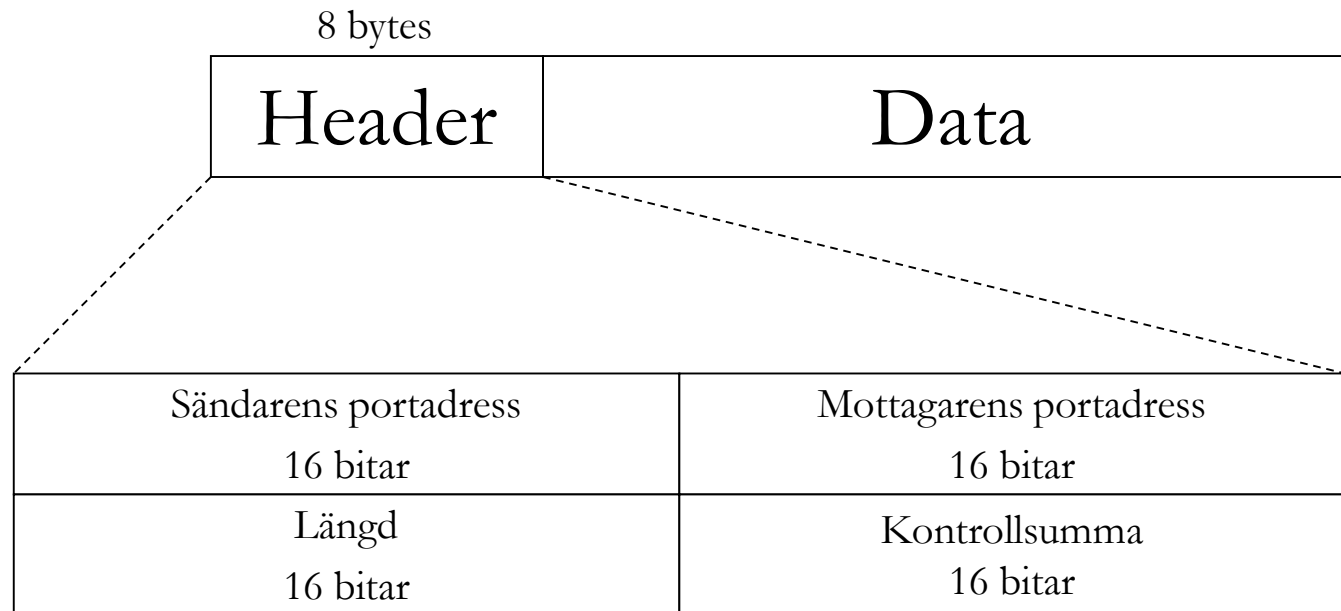


User Datagram Protocol (UDP)

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

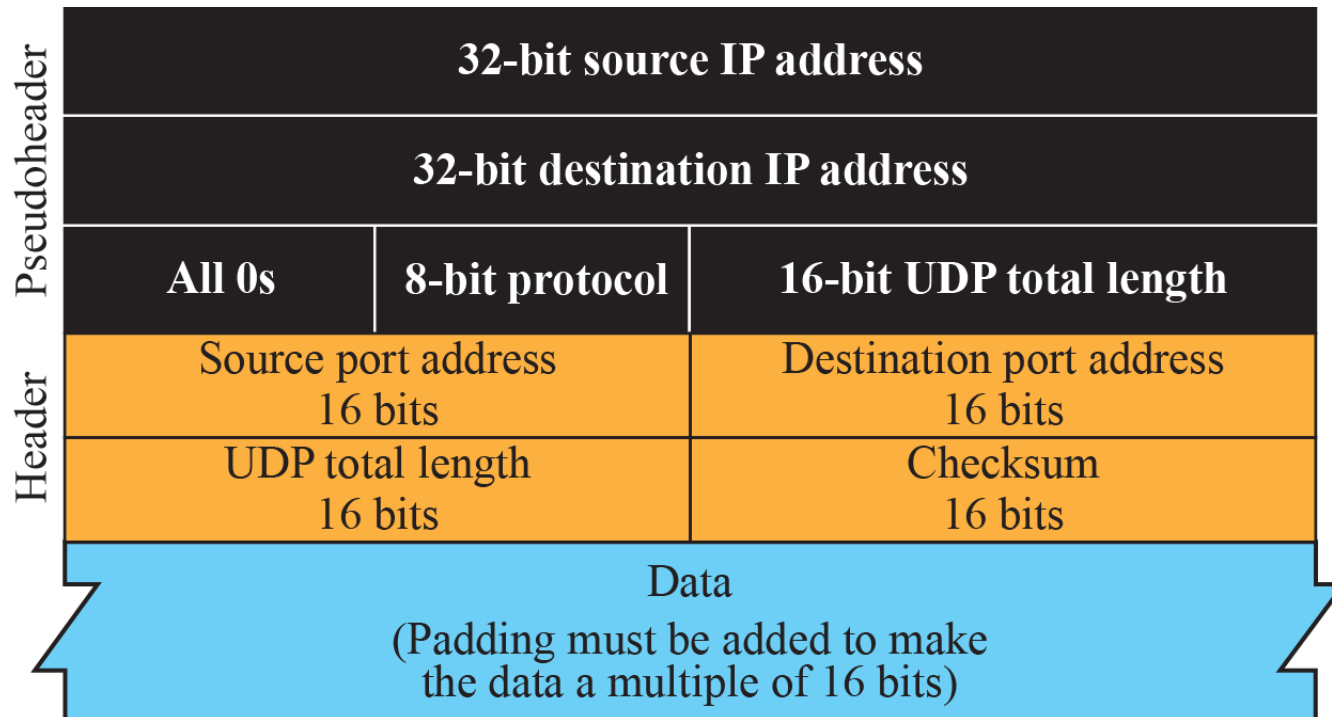
Fördel: UDP är väldigt enkelt och lägger till ett *minimum av overhead*.

UDP-headern



Checksum

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



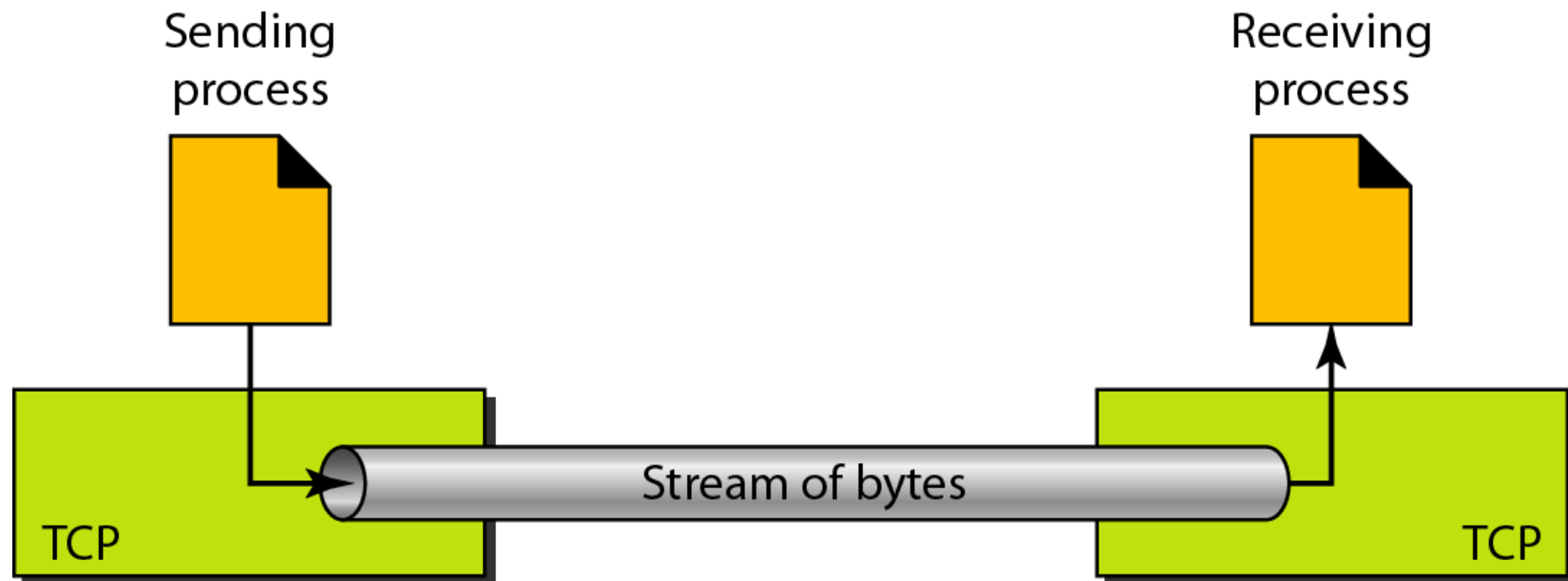
Transmission Control Protocol (TCP)

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

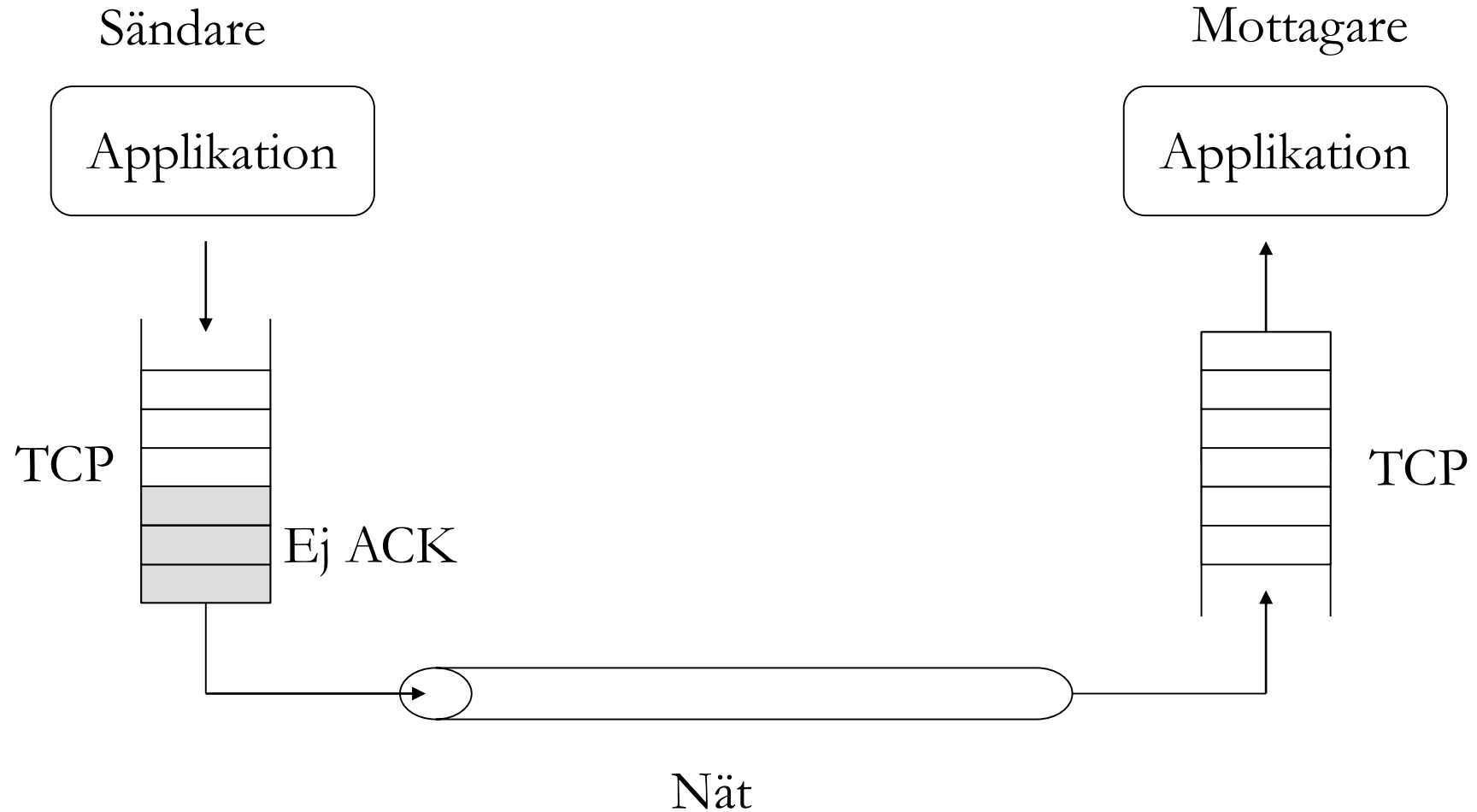
TCP packar in ett antal bytes i ett paket som kallas **TCP-segment**.

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

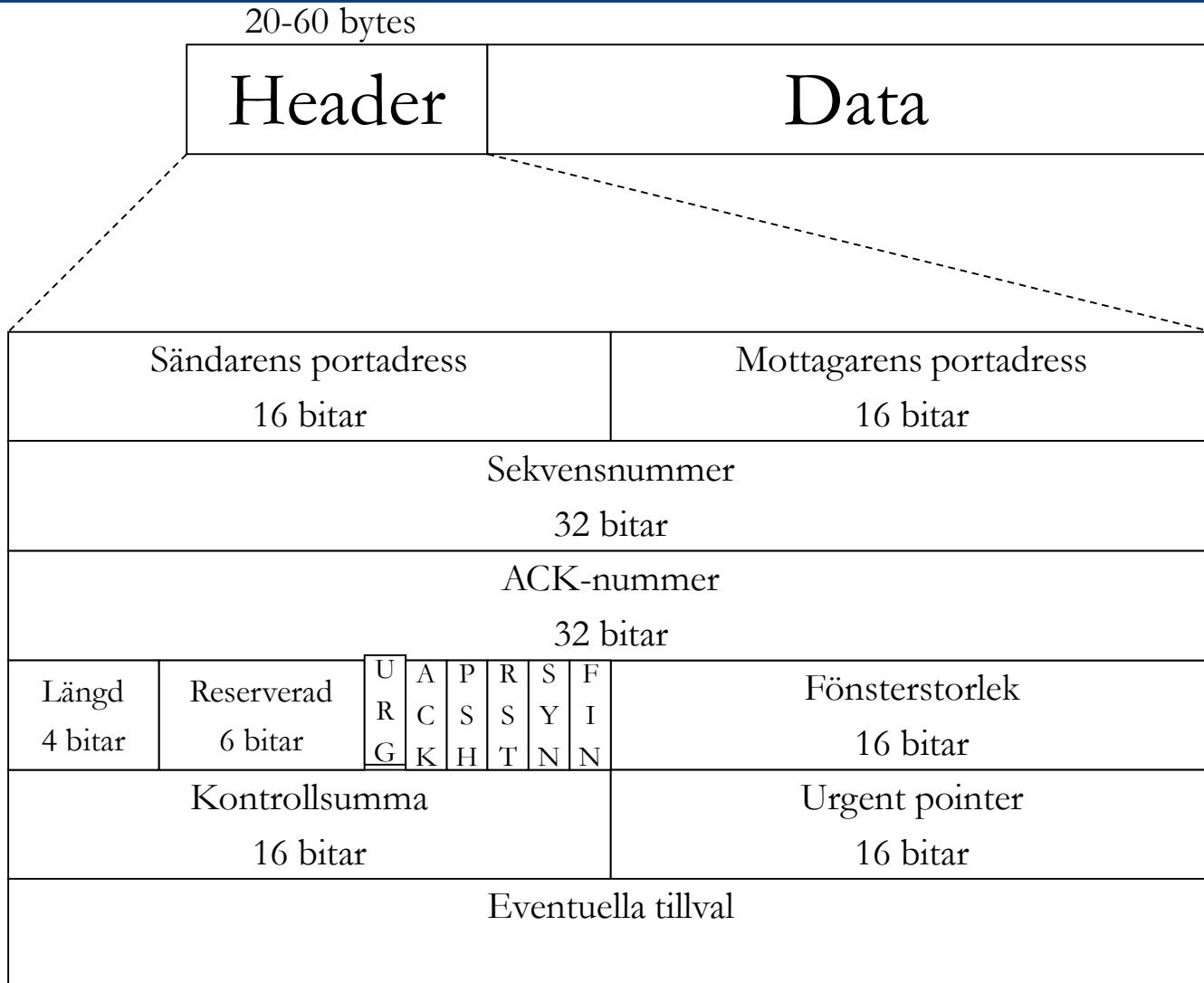
TCP innehåller ett **Go-back-N protocol** där sekvensnumret är den första **byte** som finns i segmentet. **ACK**:et innehåller numret på nästa byte som mottagaren förväntar sig.

ACKs kan vara **piggybacked**.

TCP buffert, exempel



TCP-header

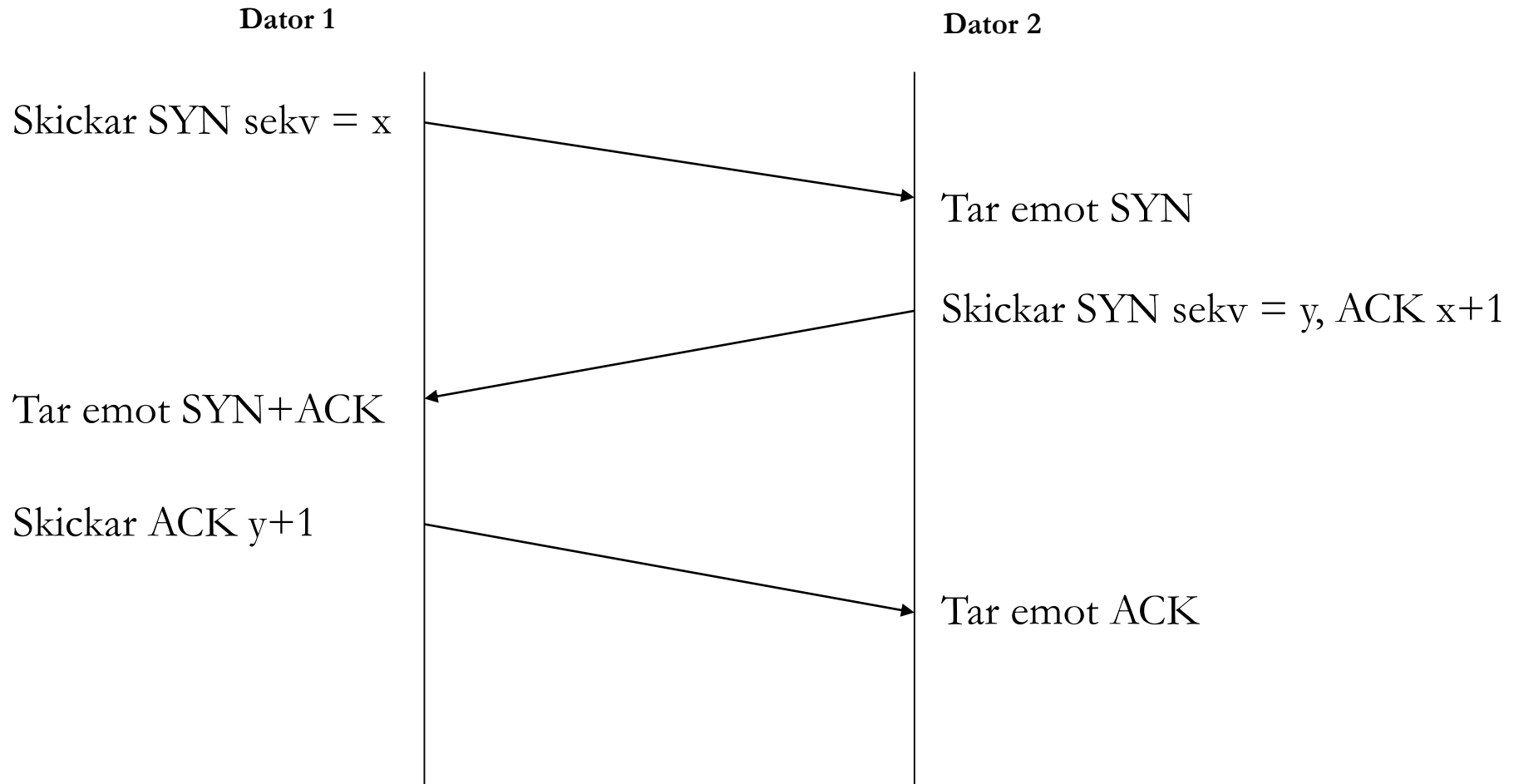


TCP sekvensnummer

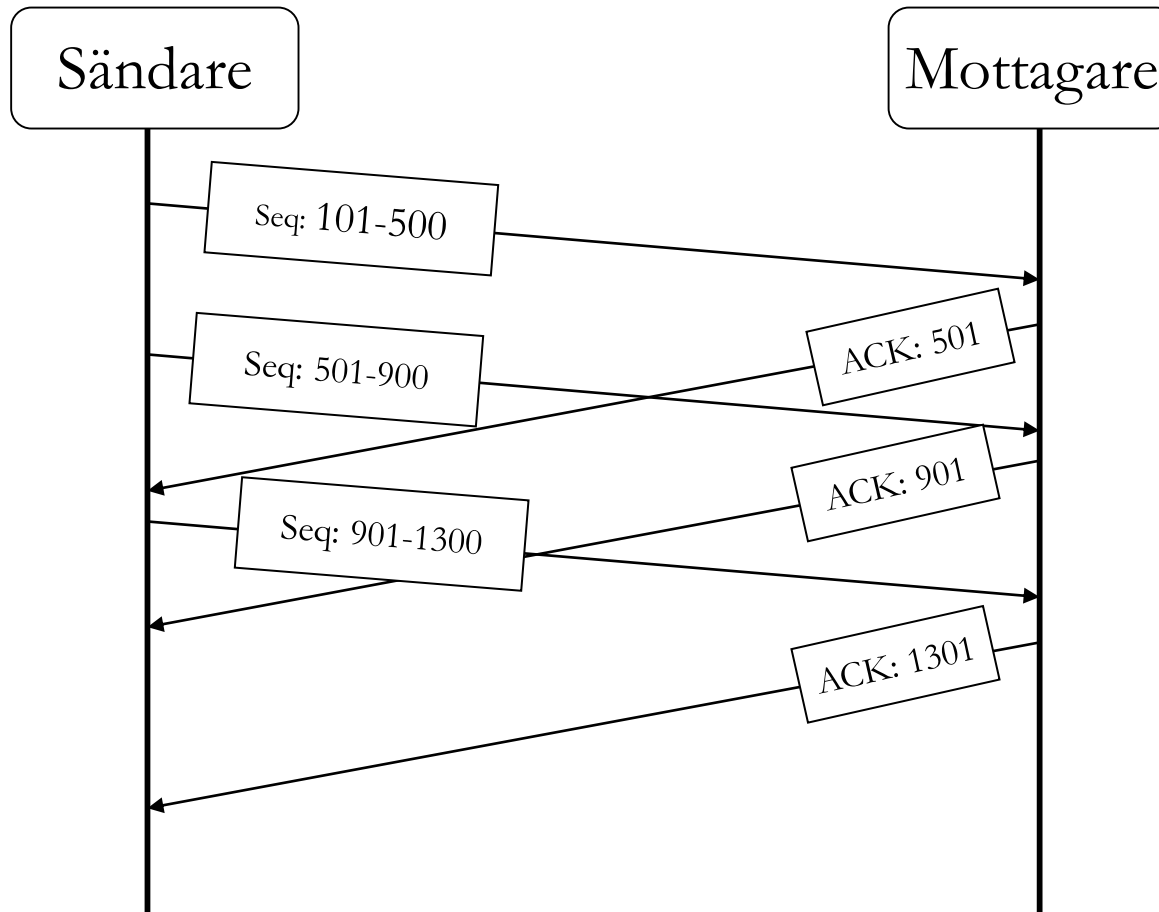
TCP tilldelar ett sekvensnummer till varje segment som skickas.

- 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.

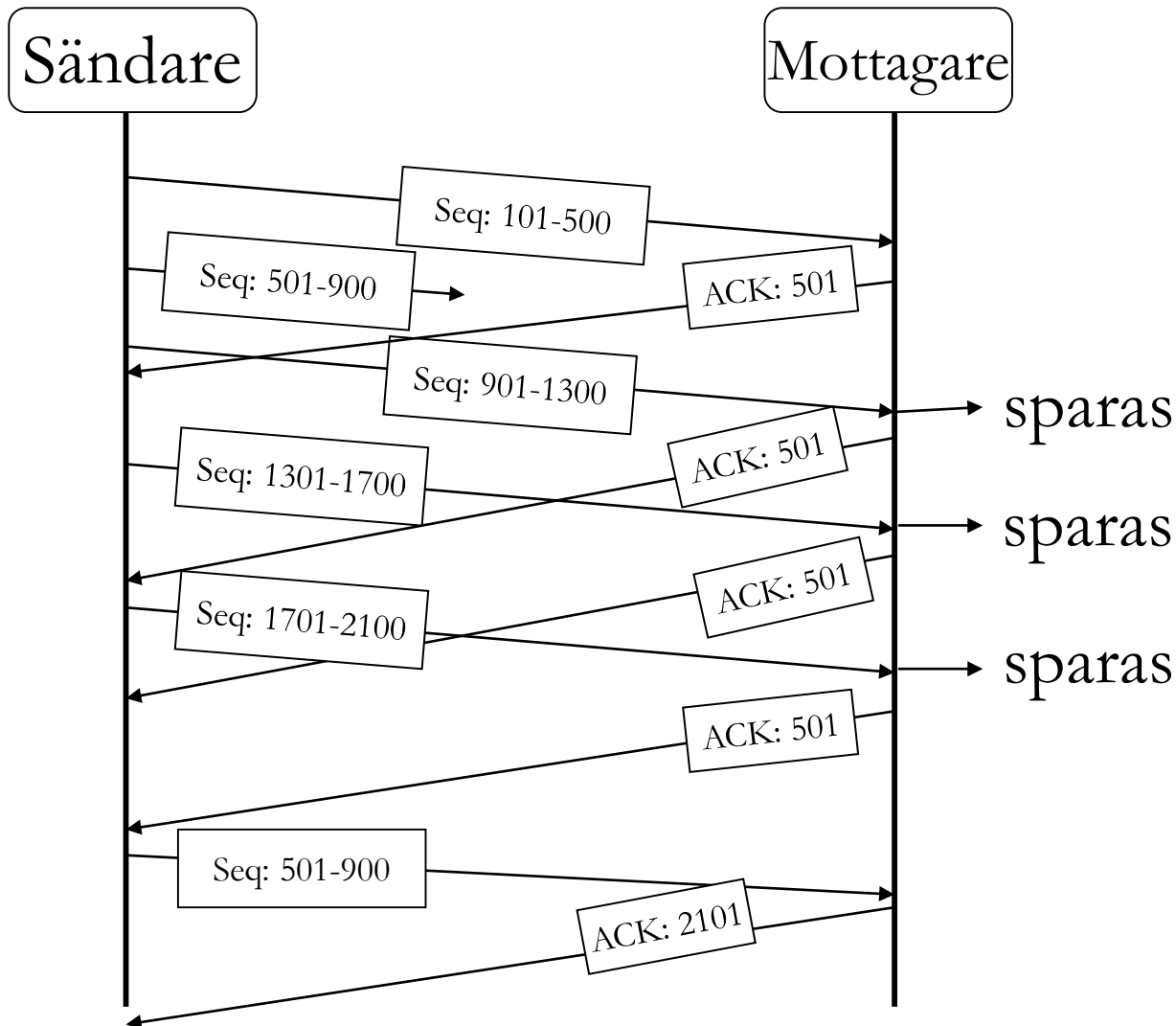
Uppkoppling av TCP-förbindelse



TCP dataöverföring om allt funkar



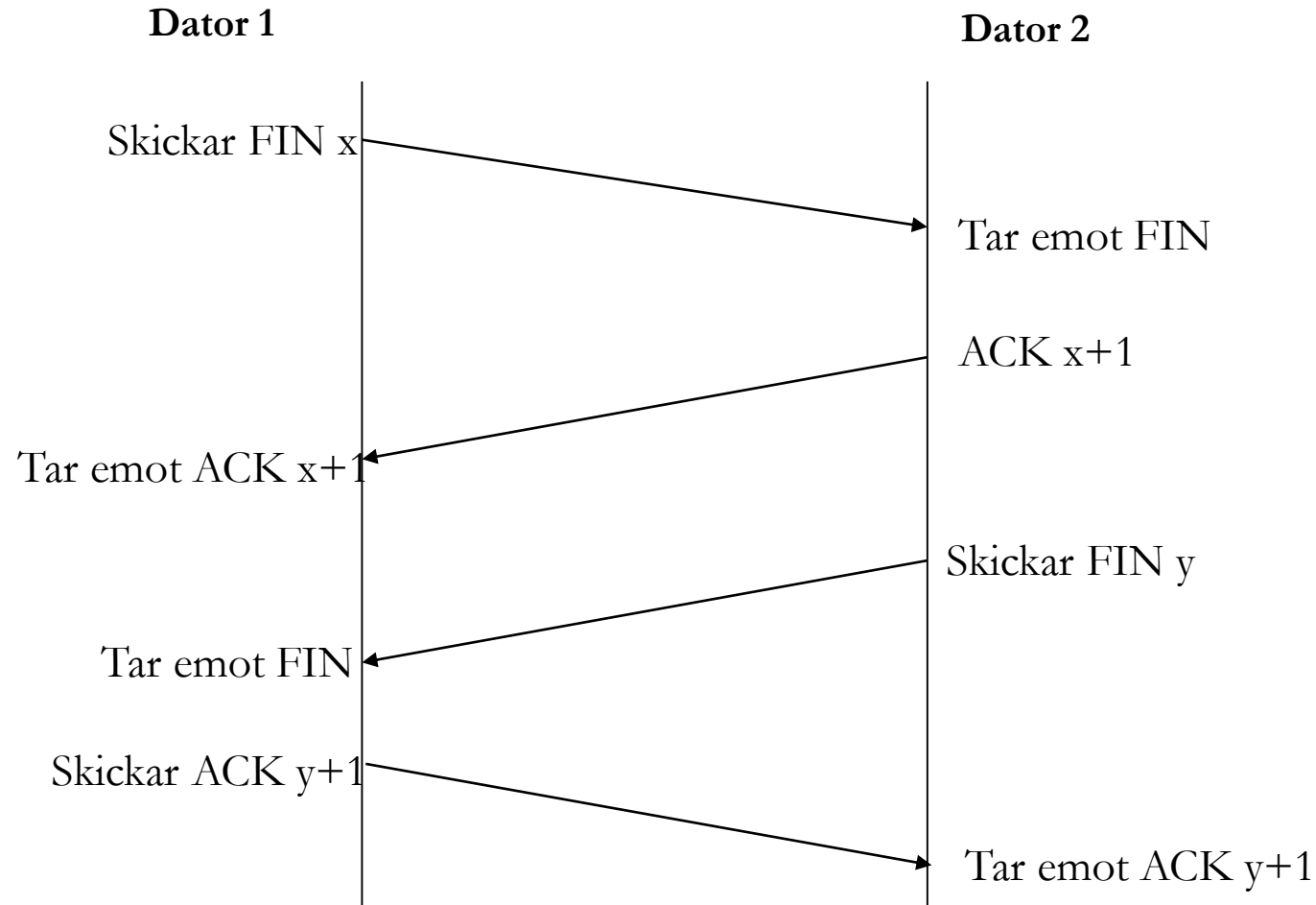
TCP omsändning



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
```