

# Tentaexempel

---

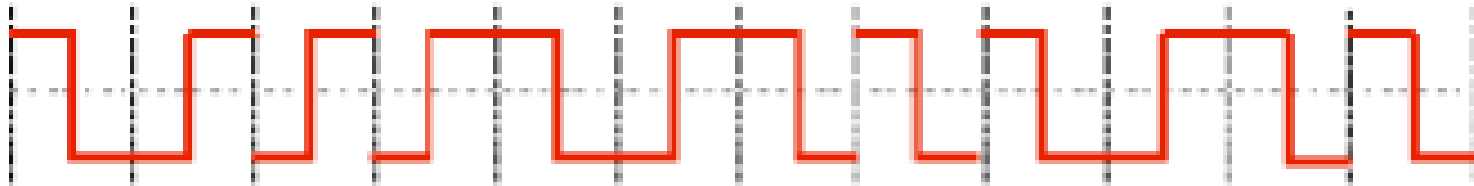
Maria Kihl



**LUND**  
UNIVERSITY

# Linjekodning

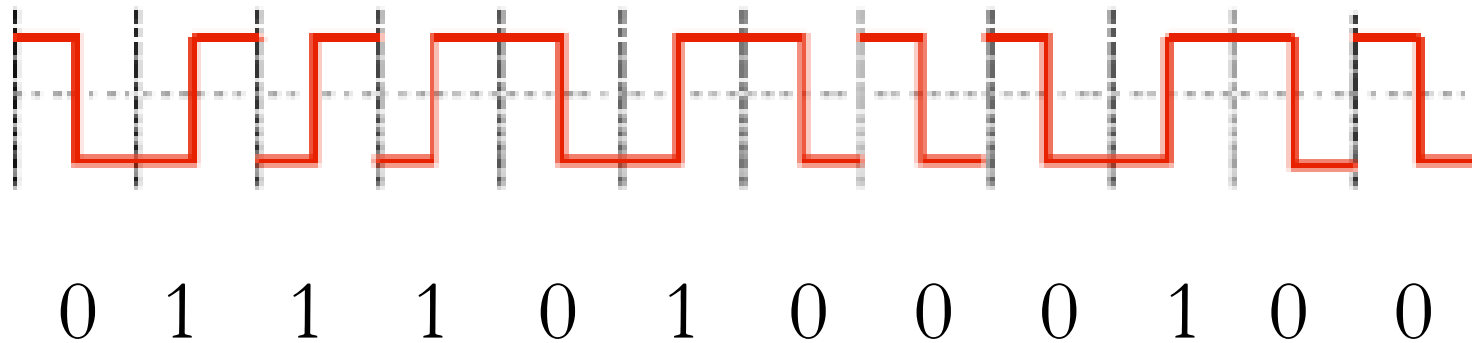
Följande signaler har kodats med Manchester. Hur ser bitströmmen ut om den inleds med en 0:a?



# Lösning

Övergång från hög-låg spänning = 0

Övergång från låg-hög spänning = 1



# Modulering

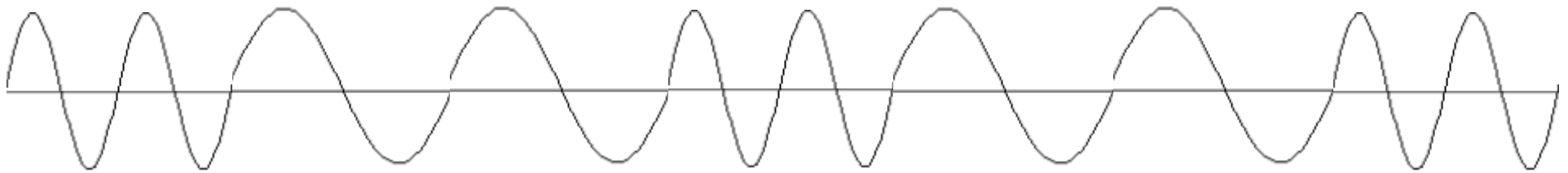
---

Koda bitsekvensen 1001001 med hjälp av  
Frekvensmodulering!

Tips: Definiera först hur du kodar 0 och 1.

# Lösning

Tex 0=låg frekvens, 1=hög frekvens (vad låg respektive hög är beror på frekvensband och standard)



# Pulse Code Modulation (PCM)

---

Anta att en ljudsignal använder frekvensområdet 0-10 kHz. Förklara hur denna ljudsignal kan kodas med 6 bitars datasegment på ett sätt som gör att mottagaren kan återskapa signalen korrekt.

Vad blir den minsta bithastigheten?

# Lösning

---

Frekvenser 0-10 kHz  $\Rightarrow$  Vi måste sampla med 20 kHz för att få med alla frekvenser.

6 bitars datasegment  $\Rightarrow$  64 kvantiseringsnivåer

Varje sampel avrundas till närmaste kvantiseringsnivå och kodas med 6 bitar.

$20 \text{ kHz} * 6 \text{ bitar} = 120 \text{ kbitar/sekund}$

# Multiplexering

---

Följande bitsekvens kommer in till en demultiplexor för STDM (siffran till vänster kommer in först):

**1 1 0 0 0 0 0 1 0 1 1 1 0 1 1**

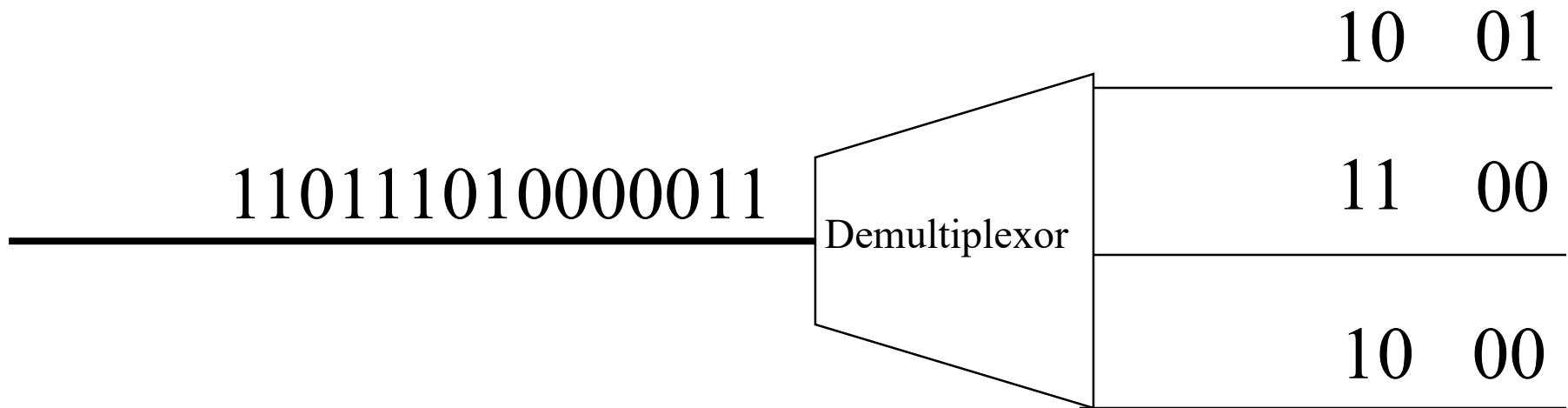
Systemet har tre kanaler, 2 bitar i varje tidslucka och en synkroniseringsbit (en 1:a) mellan varje ram.

Beskriv bitströmmarna ut från demultiplexorn!



# Lösning

---



# Bitstuffing

---

Följande bitström har tagits emot. Flaggan 01111110 används. Identifiera eventuella flaggor och bitstuffade 0:or.

**00010111110000101111110111100011111000001**

# Lösning

---

0001011111000010111110111100011111000001

0 = Bitstuffed 0:a

0111110 = Flagga

# Paritetsbit

---

Följande meddelande har tagits emot. Protokollet använder en paritetsbit (jämn paritet). Har meddelandet tagits emot korrekt?

**000101111100001**

# Lösning

---

Jämn paritet = Jämmt antal 1:or i meddelandet.

**000101111100001**

7 stycken 1:or =>

meddelandet är inte mottaget korrekt.

# CRC

---

Beräkna en 3-bitars CRC för sekvensen 101101 med generatortalet 1011.

# Lösning

Generatortal 1011  $\Rightarrow C(x) = x^3 + x + 1$  (Grad 3)

Meddelande 101101  $\Rightarrow M(x) = x^5 + x^3 + x^2 + 1$

$$M(x) * x^3 = x^8 + x^6 + x^5 + x^3$$

Dividera  $M(x) * x^3$  med  $C(x)$

Resten av denna division,  $x+1$ , blir CRC!

$$\Rightarrow \text{CRC} = 011$$

# Kontrollsumma (Checksum)

---

Beräkna en 4-bitars kontrollsumma för bitsekvensen

**1100 0010 1111**



# Lösning

---

Addera de tre segmenten med varandra.  
Överskjutande bitar adderas också.

$$1100 + 0010 + 1111 \Rightarrow 1110$$

Komplementet till summan blir kontrollsumman alltså

$$\text{Kontrollsumma} = 0001$$

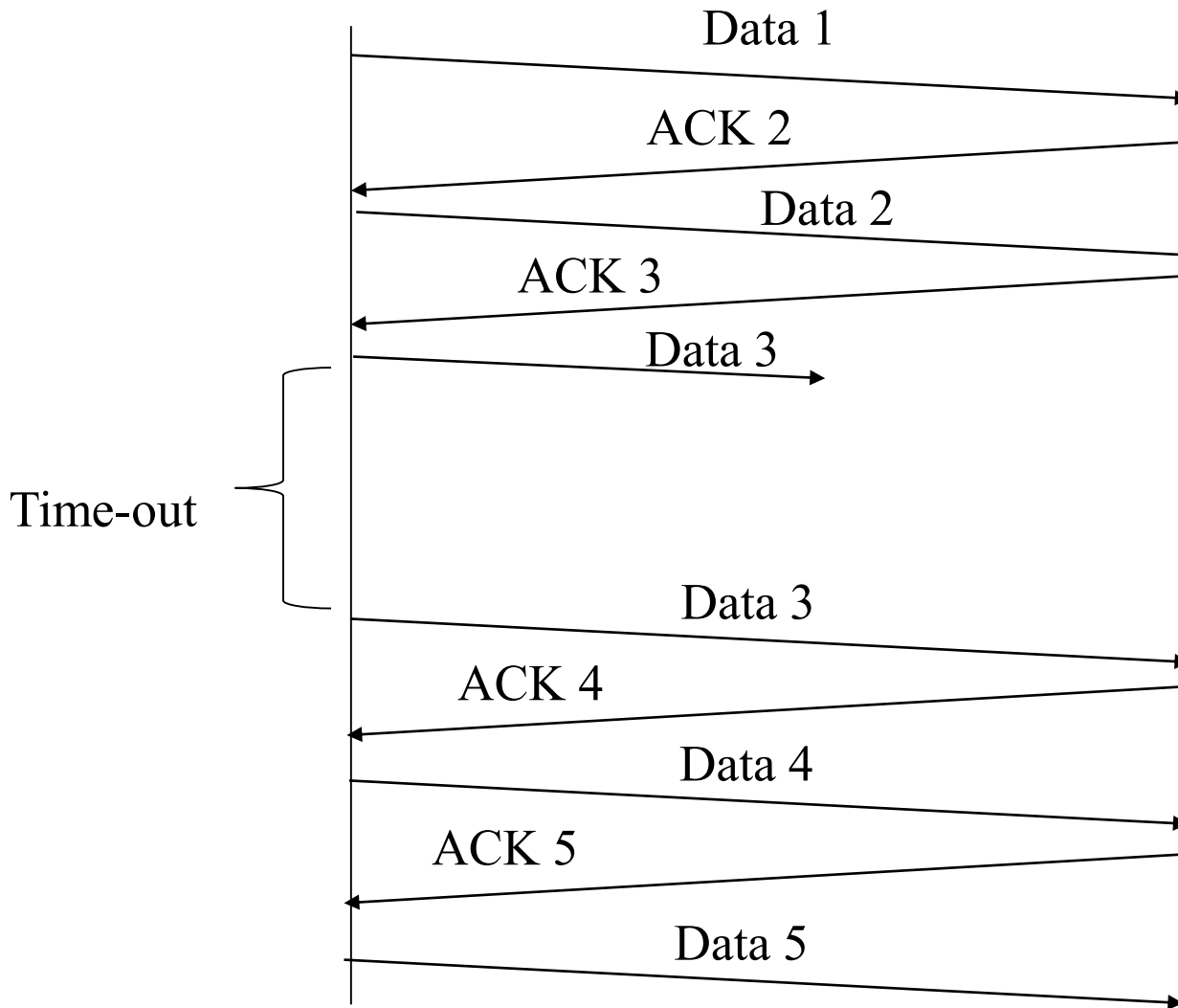
# Stop-and-wait ARQ

---

En dator skickar 5 paket (DATA 1-5) till en annan dator. DATA 3 försvinner (men alla andra paket kommer fram korrekt).

Hur många paket (DATA och ACK) skickas mellan de två datorerna om de använder Stop-and-wait ARQ?

# Lösning



Totalt 9 paket om vi inte räknar det som kastas och inte tar med ACK på sista paketet.

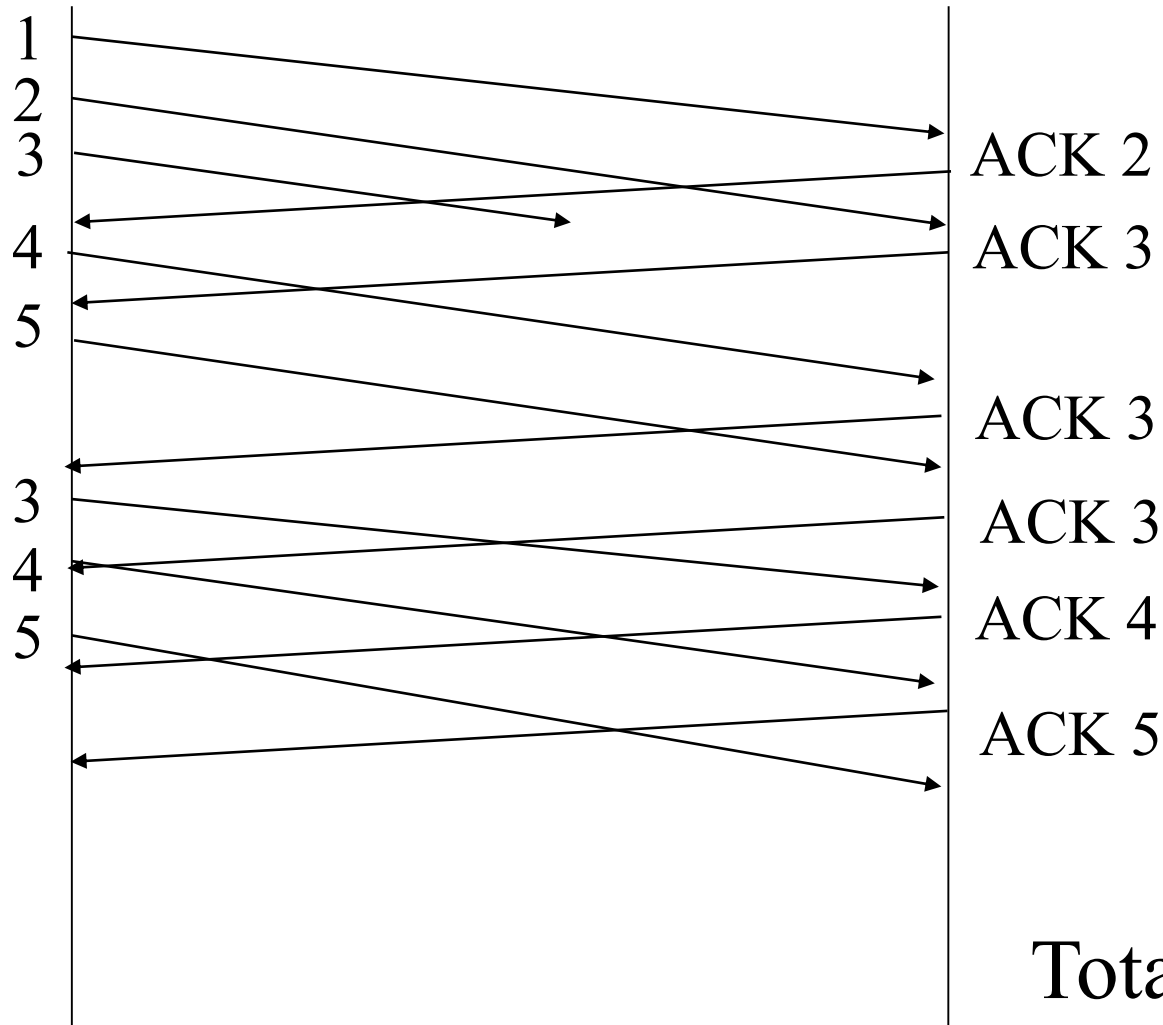
# Go-back-N ARQ

---

En dator skickar 5 paket (DATA 1-5) till en annan dator. DATA 3 försvinner (men alla andra paket kommer fram korrekt).

Hur många paket (DATA och ACK) skickas mellan de två datorerna om de använder Go-back-N ARQ med fönsterstorlek 3?

# Lösning



Totalt 13 paket

# Medium access control

---

Antag att vi ska bygga ett lokalt nät med 10 datorer kopplade till samma länk. Vi väljer mellan att använda en Token Ring baserad eller en CSMA/CD baserad accessmetod.

Vilka fördelar och nackdelar har dessa metoder i vårt nät med avseende på hur enkelt det ska vara att konfigurera och driva nätet samt hur mycket data varje dator kommer att skicka?

# Lösning

---

Exempel:

- Med Token Ring blir det mer rättvist när alla datorer har mycket att skicka.
- CSMA/CD är enklare att administrera eftersom alla datorer sköter sig själva.
- CSMA/CD är mer effektivt när ett fåtal datorer vill skicka data.

# Ethernet-header

Hitta sändaradress och mottagaradress (MAC-adresser) i följande Ethernet-ram (preamble och SFD är borttagna).

```
00 25 22 81 dd 39 ac 81 12 1b 97 55 86 dd 60 00
00 00 00 28 3a 80 20 01 16 d8 cc 3a 0b f6 d4 91
66 c2 cf c2 02 71 20 01 09 b0 01 00 00 04 00 00
00 00 00 00 00 04 80 00 8d a6 00 01 00 07 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
```



# Lösning

Mottagare:

Sändare:

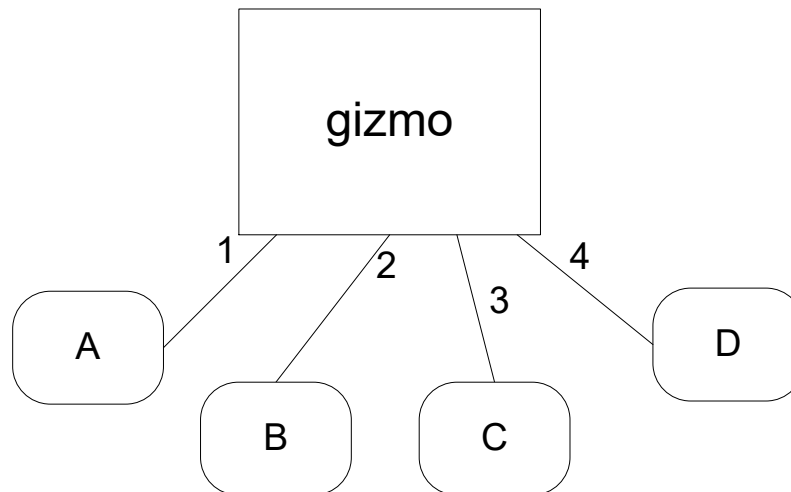
00	25	22	81	dd	39	ac	81	12	1b	97	55	86	dd	60	00
00	00	00	28	3a	80	20	01	16	d8	cc	3a	0b	f6	d4	91
66	c2	cf	c2	02	71	20	01	09	b0	01	00	00	04	00	00
00	00	00	00	00	04	80	00	8d	a6	00	01	00	07	61	62
63	64	65	66	67	68	69	6a	6b	6c	6d	6e	6f	70	71	72
73	74	75	76	77	61	62	63	64	65	66	67	68	69		

Mottagaradress: 00:25:22:81:dd:39

Sändaradress: ac:81:12:1b:97:55

# Hub/Switch

Följande "gizmo" kopplar ihop fyra terminaler (A-D) till ett nät. Antag att A skickar en ram adresserad till C. På vilka länkar (1-4) kommer denna ram att skickas om gizmo är en (a) hub (b) switch.



# Lösning

---

- (a) Hubbar broadcastar allt så ramen skickas på alla länkar.
- (b) Switchar har en adresstabell för att veta vilken länk en ram ska skickas vidare på. Om switchen inte har C i tabellen så broadcastar den ramen på alla länkar. Om den har C i tabellen så skickas ramen endast på länk 3.

# IPv4-header

Följande bitström börjar med en IPv4-header. Identifiera sändarens och mottagarens IP-adresser.

```
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 00 30 40 00 2f
a2 00 00 02 04 05 b4 01 01 04 02
```

# Lösning

```
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 00 30 40 00 2f
a2 00 00 02 04 05 b4 01 01 04 02
```

Sändare: 82 eb 12 bd => IP-adress = 130.235.18.189

Mottagare: 82.eb.84.43 => IP-adress = 130.235.132.67

# Protokoll i olika lager

Följande bitström börjar med en Ethernet-header (utan Preamble, SFD, CRC). Vilket nätprotokoll och transportprotokoll används?

```
0000: 00 09 0f ce 1f 59 e0 db 55 e4 6c 7e 08 00 45 00
0010: 00 28 48 4b 40 00 80 06 00 00 0a fe 01 e7 86 19
0020: 04 97 d5 34 00 50 31 40 0c e5 46 6b c1 43 50 10
0030: 02 99 97 af 00
```

# Lösning

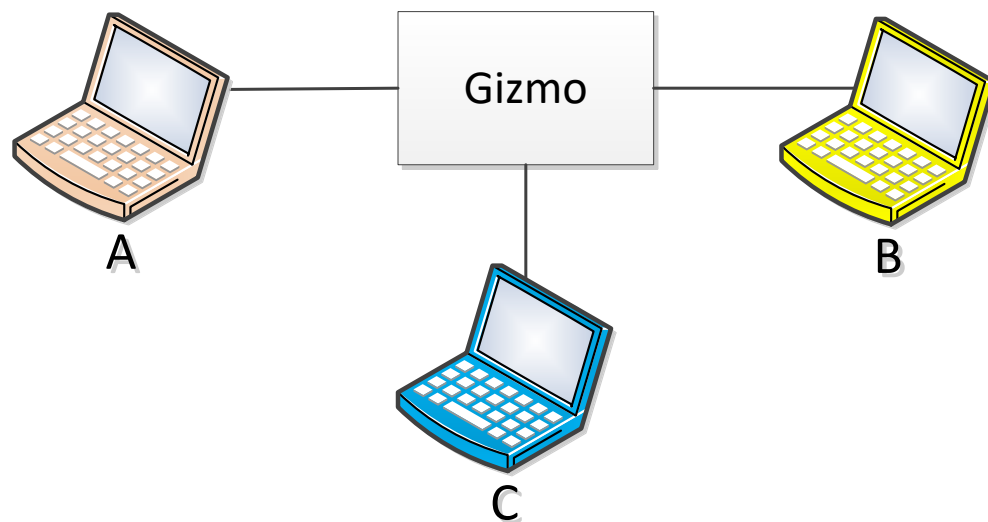
```
0000: 00 09 0f ce 1f 59 e0 db 55 e4 6c 7e 08 00 45 00
0010: 00 28 48 4b 40 00 80 06 00 00 0a fe 01 e7 86 19
0020: 04 97 d5 34 00 50 31 40 0c e5 46 6b c1 43 50 10
0030: 02 99 97 af 00
```

Type-fältet i Ethernet-headern är satt till 0800 => IPv4 (rött)

Protocol-fältet i IP-headern är satt till 06 => TCP (grönt)

# 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 ?





# Lösning

---

- (a) En fysisk adress så att C vet att paketet ska dit.
- (b) En fysisk adress så att switchen vet vart den ska skicka paketet samt att C vet att paketet ska dit.
- (c) En IP-adress för att routern ska veta vart den ska skicka paketet samt en fysisk adress så att länkprotokollet ska fungera (även om det bara sitter en mottagare på varje länk).

# Protokoll i olika lager

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
```

# Lösning

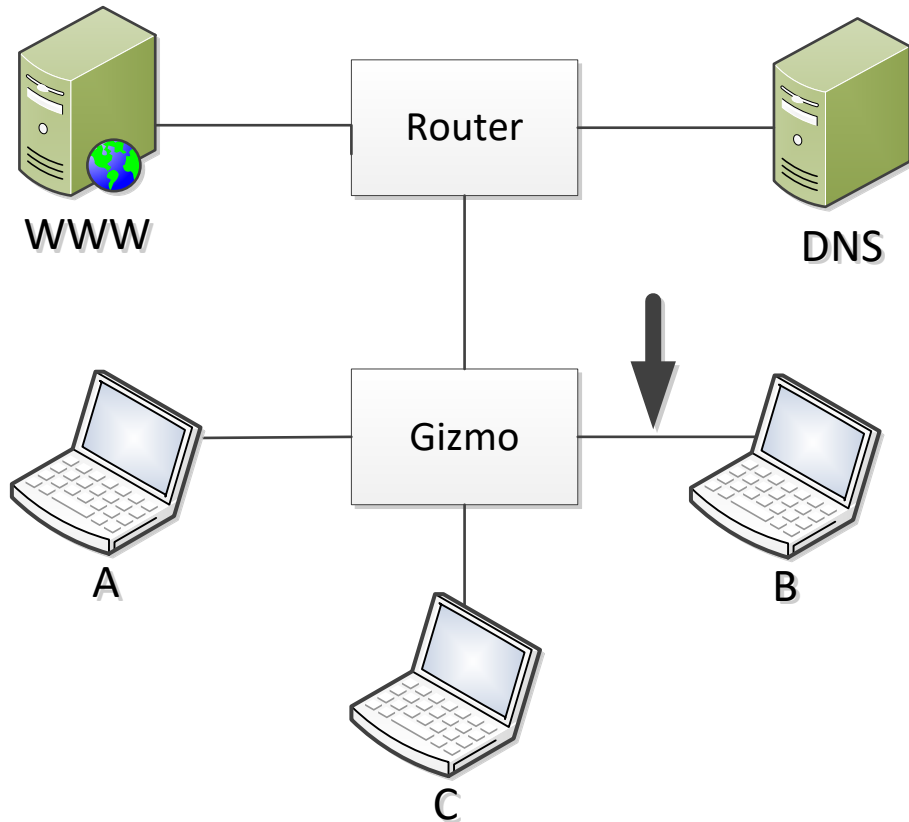
```
00 00 0c 07 ac 01 00 00 39 51 90 37 08 00 45 00
05 dc 48 00 20 00 20 01 94 67 82 eb 12 7f 82 eb
80 64 08 00 e3 fb 03 00 0c 00 61 62 63 64 65 66
67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76
77 61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f
```

Sändarens

MAC-adress: 00:00:39:51:90:37

IP-adress: 82 eb 12 7f => 130.235.18.127

# Networking



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!

# Lösning (a) Hub

Meddelande	Sändare	Mottagare	Innehåll
ARP Request	MAC(A), IP(A)	IP(C), MAC(Broadcast)	Vem har IP(C)?
ARP Reply	MAC(C), IP(C)	MAC(A), IP(A)	Jag har IP(C)
IP-paket	MAC(A), IP(A)	MAC(C), IP(C)	Data

Hubben bara broadcastar allt.

# Lösning (b) Switch

Meddelande	Sändare	Mottagare	Innehåll
ARP Request	MAC(A), IP(A)	IP(C), MAC(Broadcast)	Vem har IP(C)?

Switchen lär sig var A sitter så ARP Reply går inte till B.

# Lösning (c) Router

---

Inget skickas på länken till B eftersom alla hosts sitter på olika nät.

# IP-adresser

---

Identifiera nät-id och värd-id för adressen

160.184.66.53/28

Identifiera även adressblocket som adressen ingår i.



# Lösning

---

160.184.66.53/28

Adress: 10100000 10111000 01000010 00110101

Nätmask: 11111111 11111111 11111111 11110000

---

Nät-id: 10100000 10111000 01000010 00110000

Värd-id: 00000000 00000000 00000000 00000101

(kan sedan konverteras till decimal-dotted format)

Adressblock: 160.184.66.48 – 160.184.66.63

# IPv6-adress

---

Ange den kortaste formen på följande IPv6 adress:

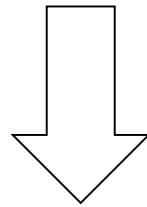
0340:0000:0000:0000:000B:C003:0000:0234

# Lösning

---

Följ reglerna för förkortning av IPv6-adresser:

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



340::B:C003:0:234

# Header-uppgift

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
```

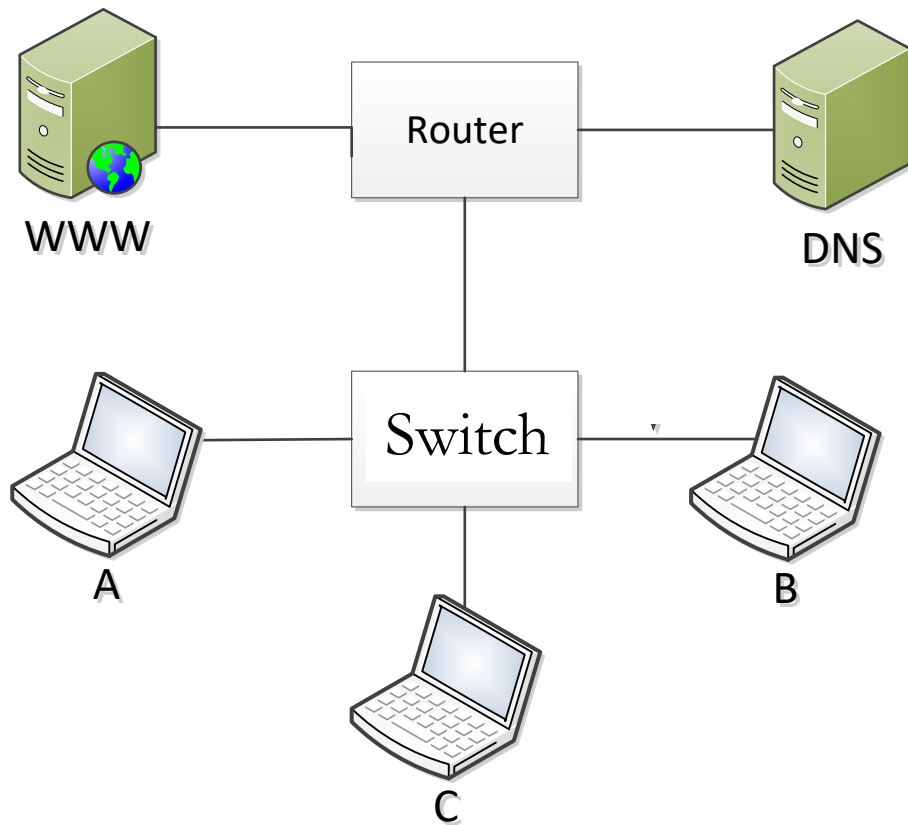
# Header-uppgift

```
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
```

Först måste du hitta var TCP-headern börjar, den är efter Ethernet- och IPv4-headers.

Destinationens portnummer är 0x0017 vilket decimalt är 23.

# Networking



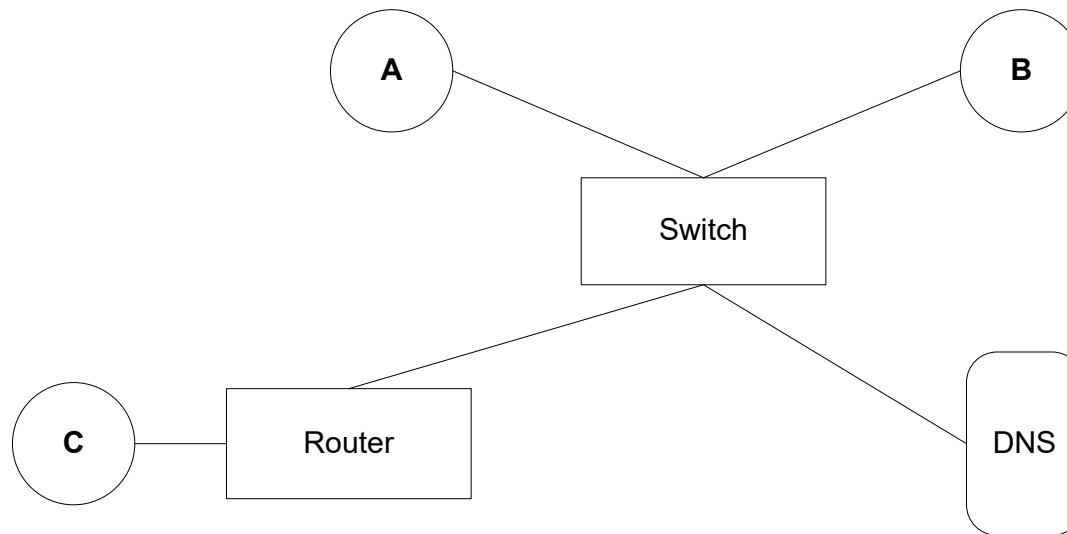
Antag att A vill hämta en websida på WWW-servern och A känner endast till WWW-serverns symboliska namn (samt de IP-adresser som förutsätts i kursen). Antag att alla adress-cacher är tomma.

Beskriv vilka meddelanden som skickas i nätet samt MAC-adresser och IP-adresser för varje meddelande.

# Lösning

Meddelande	Sändare	Mottagare	Innehåll
ARP Request	MAC(A), IP(A)	IP(Router), MAC(Broadcast)	Vem har IP(Router)?
ARP Reply	MAC(Router), IP(Router)	MAC(A), IP(A)	MAC(Router) har IP(Router)
DNS Request	MAC(A), IP(A)	MAC(Router), IP(DNS)	Vem är c.citynetwork.se?
DNS Request	MAC(Router), IP(A)	MAC(DNS), IP(DNS)	Forward av router (ARP först om Router inte har MAC(DNS))
DNS Reply	MAC(DNS), IP(DNS)	MAC(Router), IP(A)	c.citynetwork.se har IP(WWW)
DNS Reply	MAC(Router), IP(DNS)	MAC(A), IP(A)	c.citynetwork.se har IP(WWW)
HTTP Request	MAC(A), IP(A)	MAC(Router), IP(WWW)	Request för websida
HTTP Request	MAC(Router), IP(A)	MAC(WWW), IP(WWW)	Forward av router (ARP först om router inte har MAC(WWW) )

# Networking



Host A vill skicka en ICMP echo request till host C. Host A kan bara C:s symboliska namn `c.citynetwork.se`. Förutsätt att alla adress-cacher är tomma. Beskriv vilka meddelanden som skickas i nätet.



# Lösning

Meddelande	Sändare	Mottagare	Innehåll
ARP Request	MAC(A), IP(A)	IP(DNS), MAC(Broadcast)	Vem har IP(DNS)?
ARP Reply	MAC(DNS), IP(DNS)	MAC(A), IP(A)	MAC(DNS) har IP(DNS)
DNS Request	MAC(A), IP(A)	MAC(DNS), IP(DNS)	Vem är c.citynetwork.se?
DNS Reply	MAC(DNS), IP(DNS)	MAC(A), IP(A)	c.citynetwork.se har IP(C)
ARP Request	MAC(A), IP(A)	IP(Router), MAC(Broadcast)	Vem har IP(Router)?
ARP Reply	MAC(Router), IP(Router)	MAC(A), IP(A)	MAC(Router) har IP(Router)
ICMP Request	MAC(A), IP(A)	MAC(Router), IP(C)	ICMP echo request
ICMP Request	MAC(Router), IP(A)	MAC(C), IP(C)	Forward av router (ARP först om router inte har MAC(C) )

# OSI-modellen

---

Förklara var i OSI-modellen följande protokoll hör hemma:

**HTTP, 802.3, TCP, PPP, IP, UDP, ICMP, ARP**

Dina svar ska vara motiverade!

# Lösning

Protokoll eller standard	OSI lager
HTTP	7 Applikation
802.3	1 Fysiska + 2 Länk
TCP	4 Transport
PPP	2 Länk
IP	3 Nät
UDP	4 Transport
ICMP	3 Nät
ARP	2/3 Länk/Nät