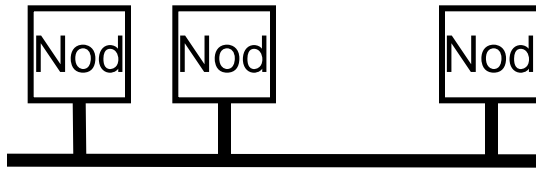


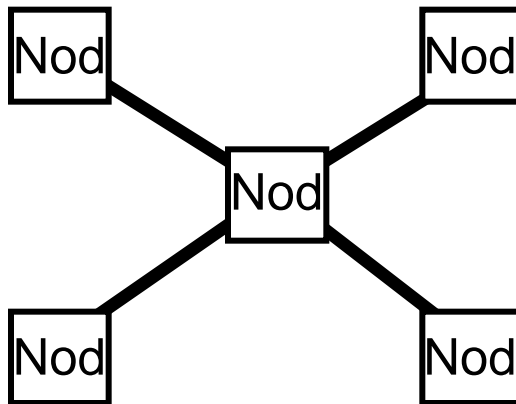
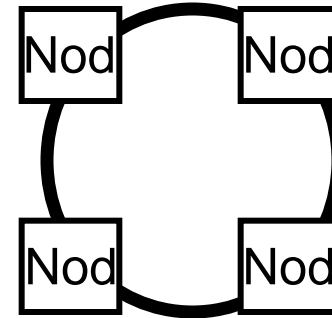
CAN – ett kommunikationsprotokoll för realtidssystem

Seriekomunikation- Datanät- Topologi

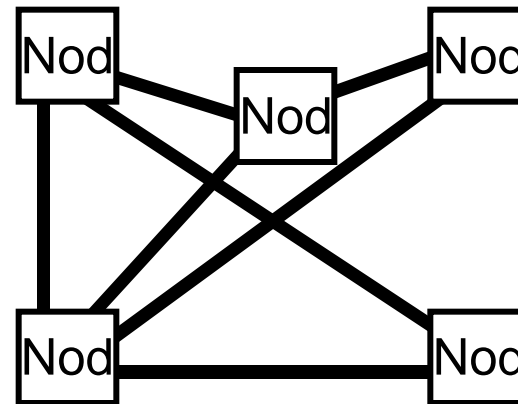
Buss



Ring



Stjärna



Masknät

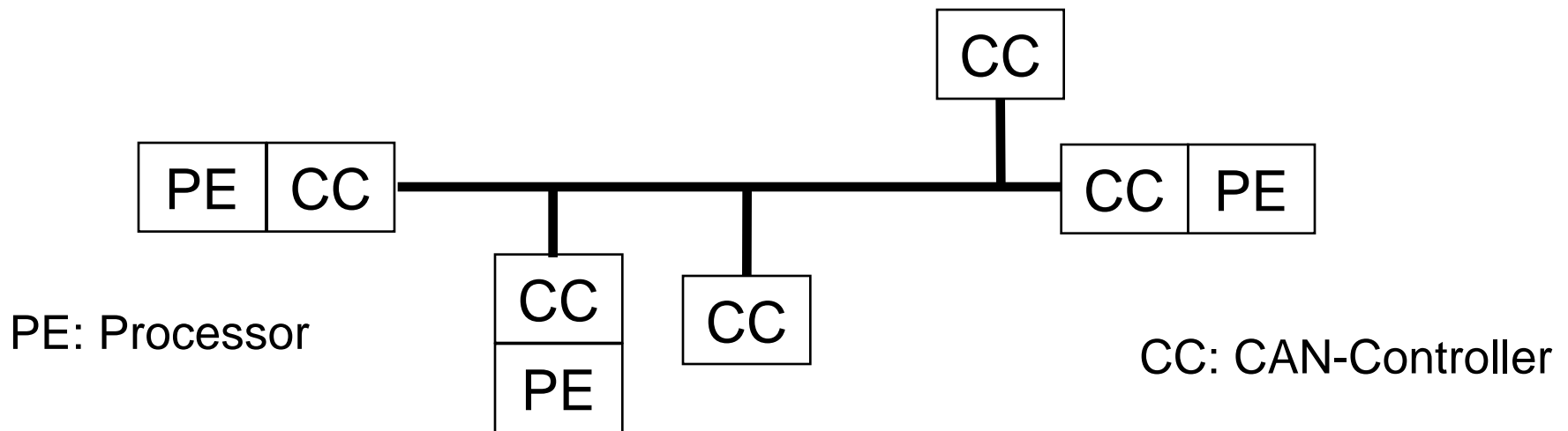
Seriekomunikation- Datanät- Accessmetoder

När ska någon få skicka data ut på nätet?

- *Polling*
- *CSMA*
- *TDMA*
- *Token*

CAN (Controller Area Network)

- Meddelanden, allmänt
- Bussåtkomst
- Synkronisering
- Olika ramar
- Felhantering
- Identifierare



CAN (forts)

– 1984 CAN C-Spec, Bosch och Intel

- Kommunikationsstandard för bilindustrin
- Standard CAN, senare Extended CAN

– I dag, Leverantörer

- INTEL, FREESCALE, PHILIPS, NCS, SGS, etc,etc

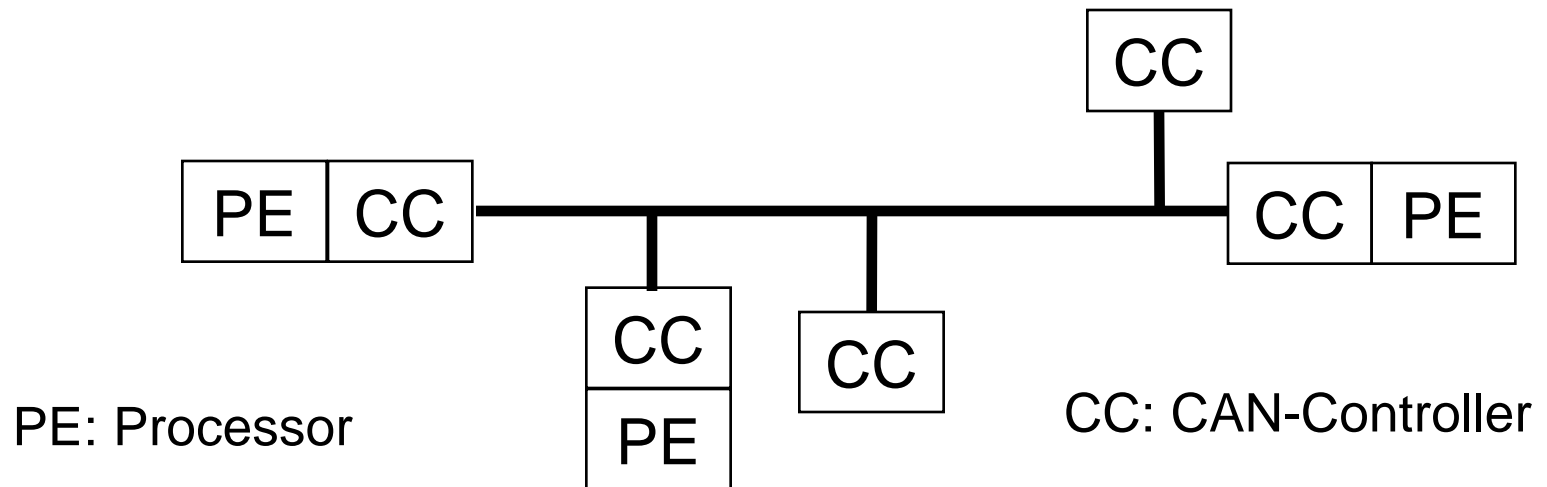
Utmärkt protokoll för distribuerad styrning av händelsestyrda system där man önskar att skicka små mängder data (1-8 bytes) i ett datanät med fysisk liten utsträckning .

CAN-protokollets egenskaper

- Flexibelt
- Multimaster protokoll
- Multicast protokoll
- Hög överföringshastighet (1 Mbit/s)
- Kort svarstid (200 μ s)
- Automatisk omsändning av störda överföringar
- ”Atomic Broadcast”
- Stöd för synkroniserad exekvering i olika noder
- Avlastar processorn/användaren med meddelandeöverföringen

Ett typiskt CAN nät

- Busstopologi
- Media: twisted pair, koaxial, fiber
- Utsträckning: 40m vid 1Mbit/s

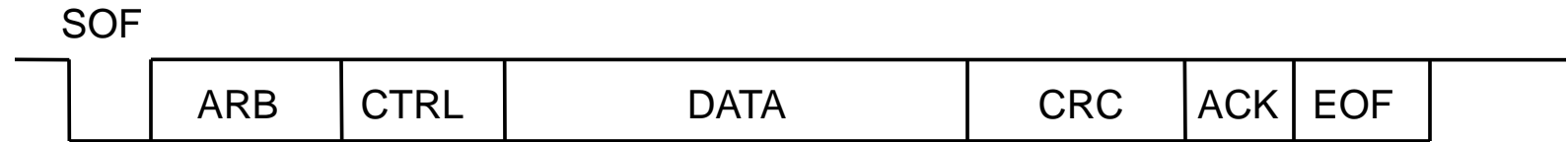


Flexibelt att ansluta nya noder

Meddelanden i ett CAN-nät

- Objekt (Meddelanden)
 - Ex: broms, ljus, fönsterhiss
 - Skicka/efterfråga ett objekt
- Prioriteter knutna till objekten (meddelanden)
 - Specificeras innan “run-time”
 - Prioriteten kan tolkas som en “adress”

Meddelanden i ett CAN-nät forts



- ARB Arbitrering (tävlan om bussen)
Identifierare anger objekttyp; även prioritet
- CTRL Control
Antal byte; Standard/Extended CAN
- Data 0-8 bytes
- CRC Kontrollsumma
Tillsammans med felhantering \Rightarrow Samtidig exekvering
- ACK Kvitto
Inbyggd handskakning
- EOF Slutmarkering

Bussåtkomst

CSMA/CD (Ethernet)

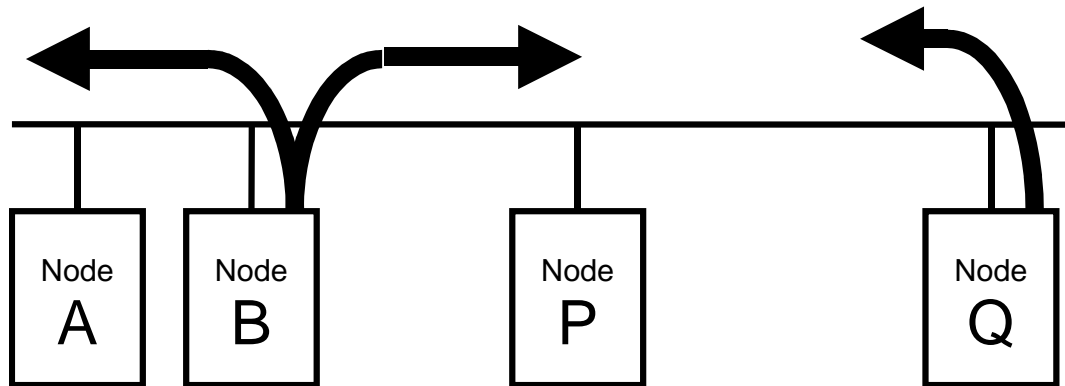
- **Carrier Sense:** Lyssna Först
- **Multiple Access:** Vem som helst kan skicka ett meddelande när bussen är ledig.
- **Collision Detect:** Buss-krock upptäcks.

Algoritm: Börja om vid fel

Bussåtkomst forts

Algoritm (CSMA/CD)

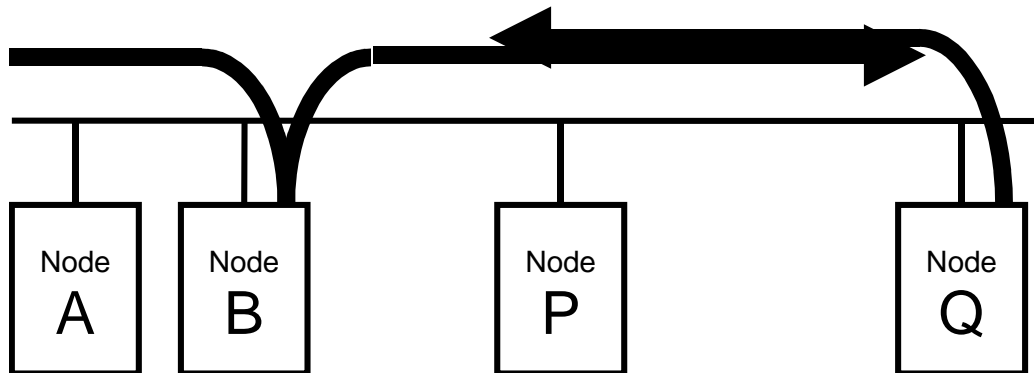
- Lyssna på bussen;
- Om bussen är ledig starta sändning av ett meddelande



Bussåtkomst forts

Algoritm (CSMA/CD)

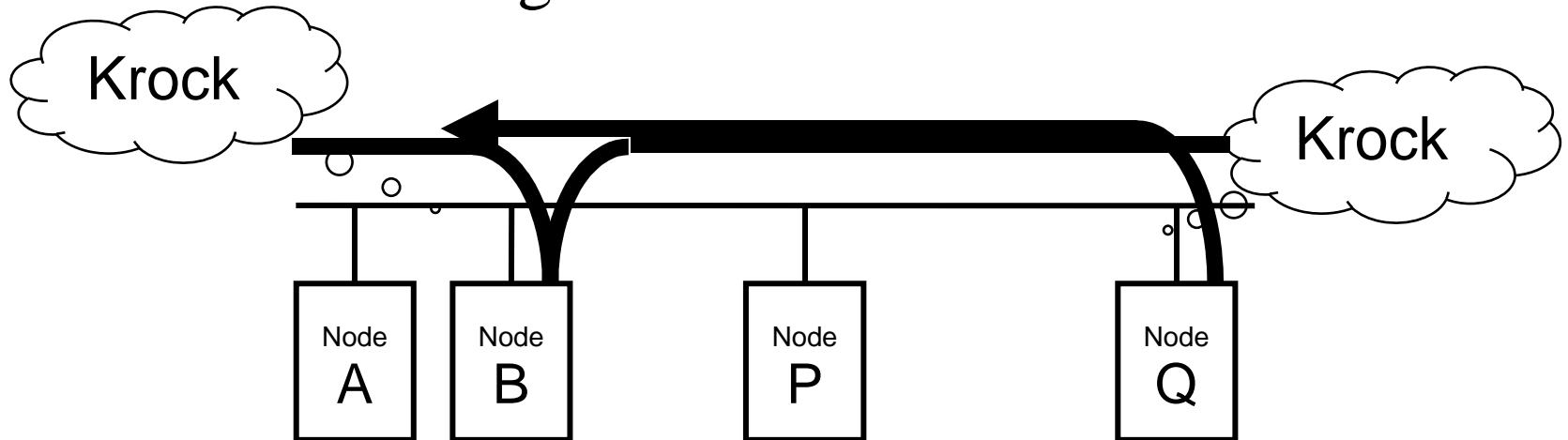
- Lyssna på bussen och kontrollera om busskrock uppstår.



Bussåtkomst forts

Algoritm (CSMA/CD)

- Om busskrock uppstått, avsluta då sändningen och försök igen senare.



Bussåtkomst forts

CSMA/CR (CAN)

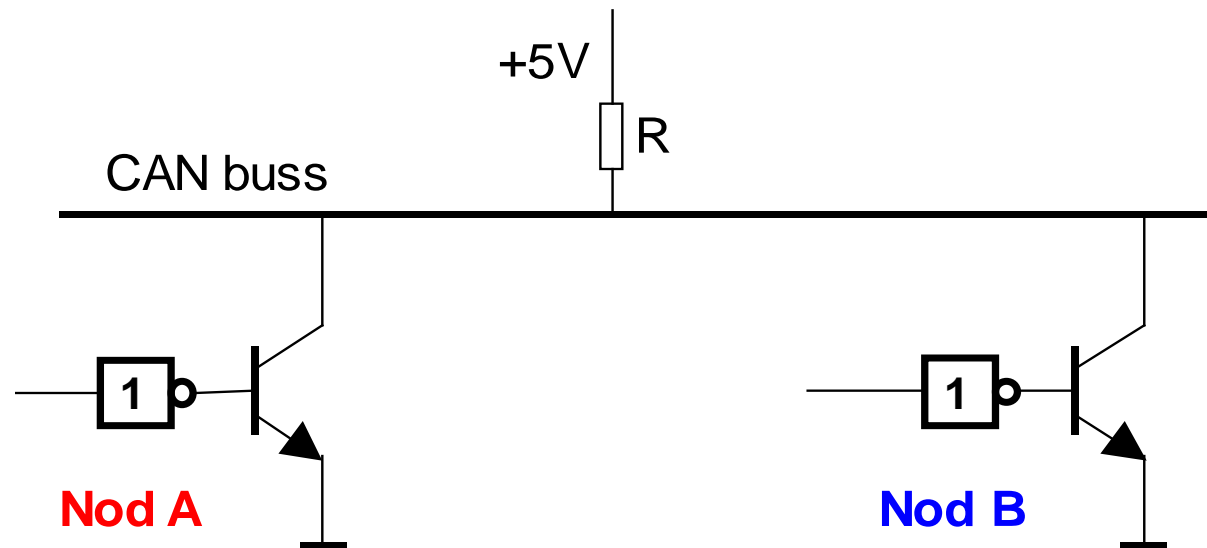
- **Carrier Sense:** Lyssna först
 - **Multiple Access:** Vem som helst kan skicka ett meddelande när bussen är ledig.
 - **Collision Resolution:** Buss-krocksupplösning.
-
- **Algorithm:** Starkast vinner
 - **Multimaster**

Bussåtkomst forts

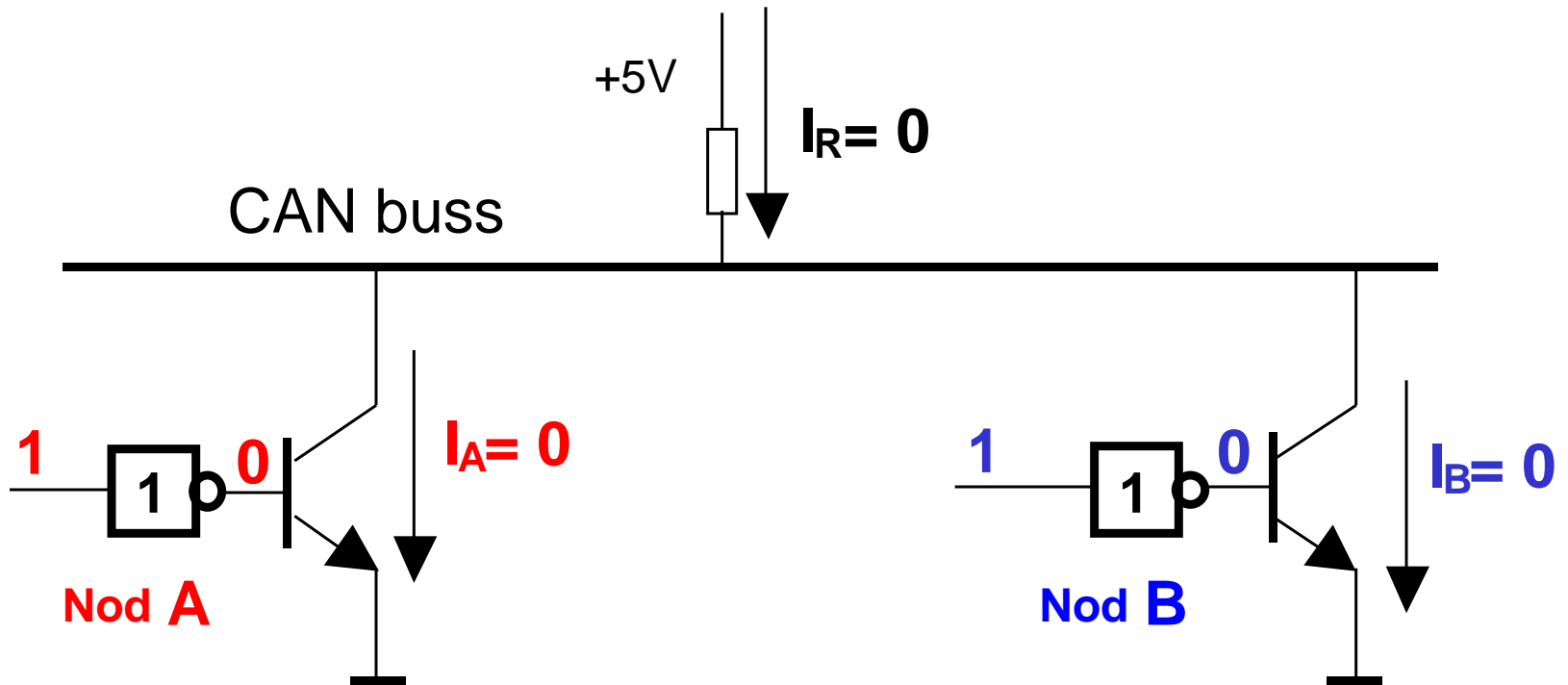
- Algoritm (CAN)
 - Lyssna på bussen. Om bussen är ledig starta sändning av ett meddelande.
 - Fortsätt lyssna på bussen och jämför bit för bit av sänd data och data på bussen.
 - Om läst bit skiljer sig från sänd bit indikerar detta att någon med högre prioritet skickar data samtidigt. Avsluta då sändningen och försök igen när “högprioritetsmeddelandet” har skickats i sin helhet.
- Kort svarstid (200 μ s vid 1Mbit)

Bussåtkomst forts

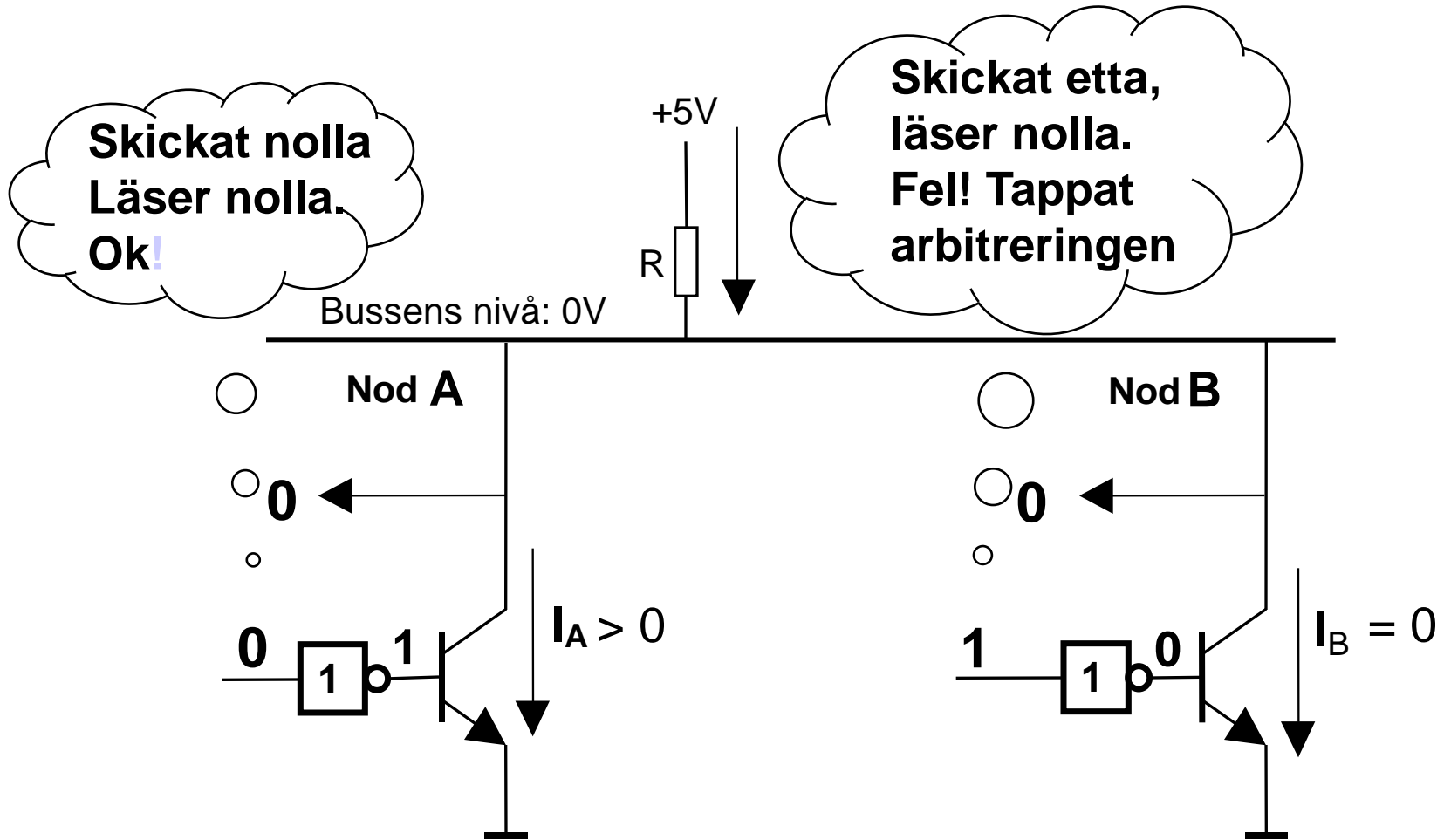
- Bussdrivare: Princip ”Open collector”
- Bussens nivå:
 - Recessiv (bit)
 - Dominant (bit)



Bussåtkomst forts



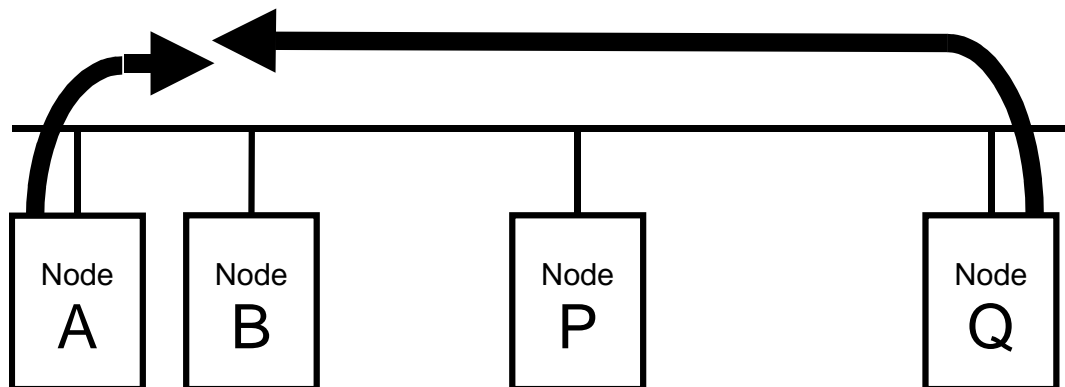
Bussåtkomst forts



Bussåtkomst forts

Inskränkningar

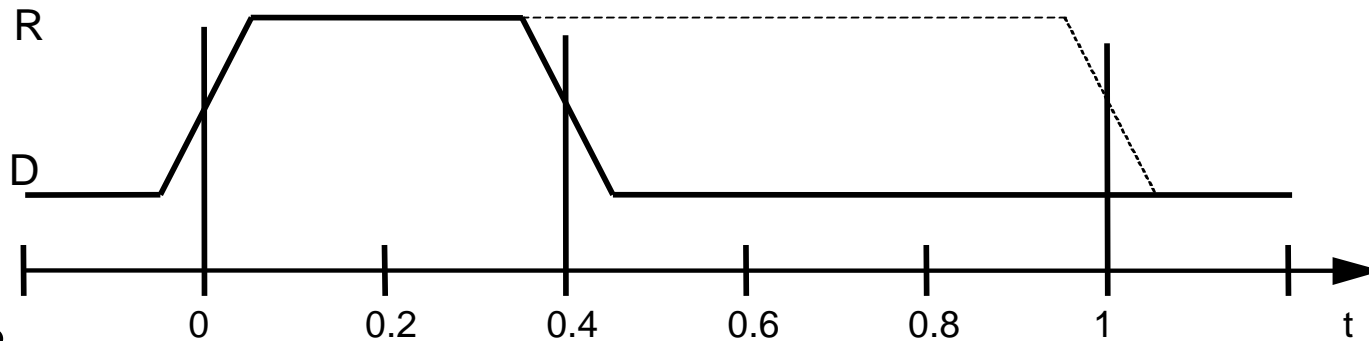
- Bitsynkronisering
- En enskild bit måste vara ”giltig i hela nätet”



Bussåtkomst forts

Inskränkningar

- Nätets utsträckning/bithastighet



•

•

$$\tau_{\text{vänta}} = 2 \cdot 1/v = 2 \cdot 40 / 2 \cdot 10^8 \text{ s} = 0,4 \mu\text{s}$$

•

Max 40m vid 1 Mbit

Bussåtkomst (arbitrering)

Tre noder vill sända samtidigt

Nod A identifierare: \$257 (010 0101 0111)
Nod B - " - : \$360 (011 0110 0000)
Nod C - " - : \$25F (010 0101 1111)

Bit nummer	SOF	1	2	3	4	5	6	7	8	9	10	11	12
	D	D	R	D	R	D	R	D	R	R	R	R	D
Bussens nivå	0	0	1	0	1	0	1	0	1	1	1	1	0
Nod A skickar	0	0	1	0	0	1	0	1	0	1	1	1	0
Nod B skickar	0	0	1	1	Slutar sända								
Nod C skickar	0	0	1	0	0	1	0	1	1	Slutar sända			

Bussåtkomst (arbitrering)

- Varje datameddelande har en unik identifierare som också anger prioritet
- **OBS!!!** Två noder får inte sända datameddelanden med samma identifierare

Synkronisering i CAN

- Asynkront protokoll
- Okodade bitar



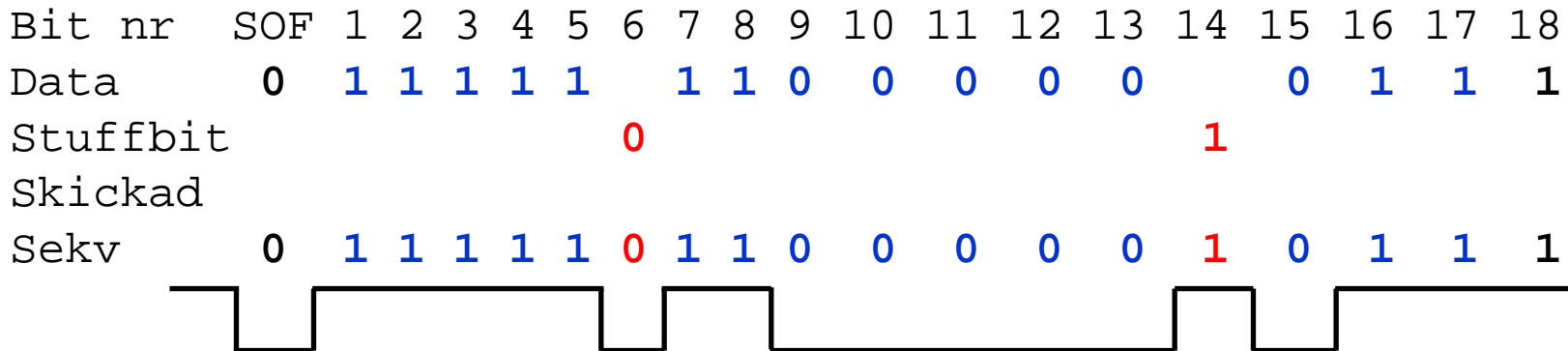
Synkronisering i CAN forts

- Synkroniseringsmekanism
 - Mottagare:
 - Hård synkronisering på SOF
 - Synkronisera på flanker i mottaget data
 - Sändaren:
 - Inför STUFF BITAR vid långa sekvenser av nollor / ettor

Synkronisering i CAN forts

- Stuffbitar

- Skickar ett meddelande som börjar med \$7F03 binärt 111 1111 0000 0011. (Bussen får innehålla högst 5 st lika bitar i rad, inklusive eventuell tidigare stuffbit.)



Olika ramar (frames)

- Data Frame (skickar data)
 - SOF Start Of Frame. 1 bit
 - ARB Arbitration. 12 bitar (Stand CAN)
 - CTRL Control. 6 bitar
 - DATA Användardata. 0-8 byte (max 64 b)
 - CRC Kontrollsumma 16 bitar
 - ACK Kvitto 2 bitar
 - EOF End Of Frame 7 bitar (Totalt 108 b)
 - IFS Inter Frame Space. 3 bitar

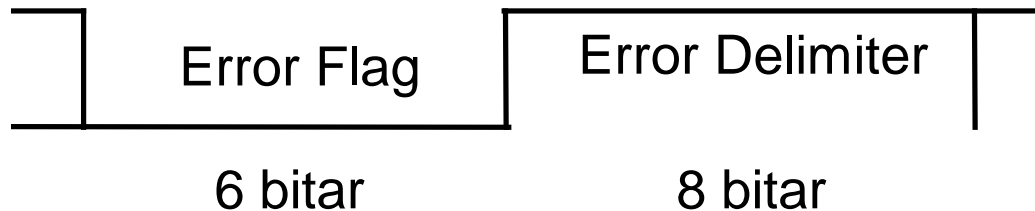


Olika ramar (forts.)

- Remote Frame (efterfråga en data frame)
 - En remoteframe saknar datafält (se data frame).
 - Remote- och dataframe har samma identifierare
 - Ex efterfråga hastighet; skicka hastighet
 - ID är 11 bitar, ARB-fältet är 12 bitar
 - RTR-bit (bit 12 i ARB-fältet)
 - Remote frame recessiv RTR-bit
 - Data frame dominant RTR-bit
 - Innebär att vem som helst kan skicka en remoteframe (men endast EN kan skicka en data-frame)

Olika ramar (forts.)

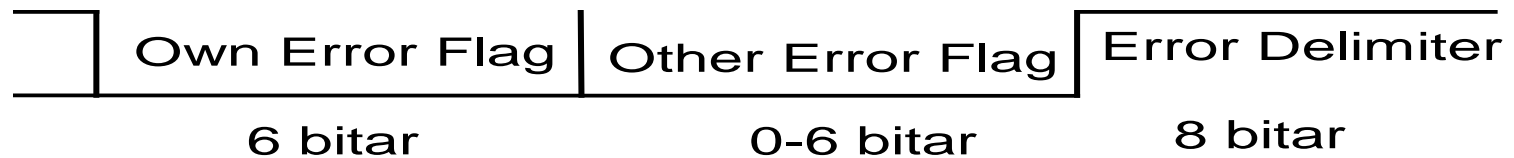
- Error frame (skriv sönder pågående utskick)
 - 6 st dominanta bitar (Error flag)
 - 8 recessiva bitar (Error delimiter)



- Alla kan skicka en Error frame
- Automatisk omsändning av störda överföringar
- Robust

Olika ramar (forts.)

- Error frame (forts.)
- Vem detekterar en error frame?
 - Sändaren av data/remote-frame
 - Utskickad bit \neq Läst bit
 - Mottagare
 - Felupptäckt

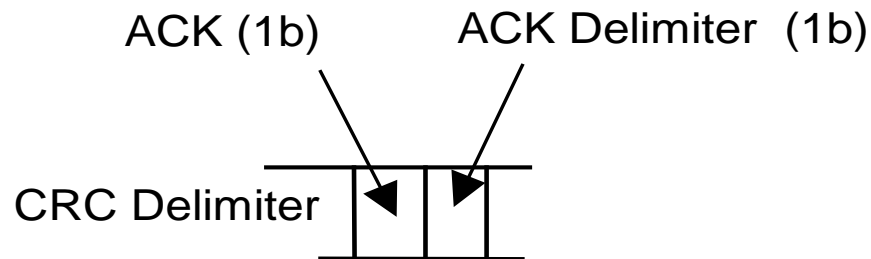


Feldetektering

- Skicka Error Frame vid felupptäckt
 - Bit-fel
 - Stuff-fel
 - CRC-fel
 - Fixed Form-fel
- Möjliggör samtidig exekvering i noder
- Atomic broadcast

Kvitto (ACK)

- En frame innehåller ett två-bitars ACK-fält för handskakning
- Sändaren av en frame skickar två recessiva bitar i detta fält
- En adresserad fungerande mottagare skriver över den första med en dominant bit
- Sändaren detekterar dominant eller recessiv bit i ACK-fältet



Kvitto forts

- Vad innebär ACK-funktionen?
 - Sändaren “ser” att hans frame tas emot
 - Sändaren detekterar att den inte är “ensam i nätet”
 - Observera att sändaren inte kan detektera *vem* som korrekt tagit emot utskickad meddelande utan endast att *någon* korrekt tagit emot utskickad meddelande.
- Underlättar konsistens

Identifierare

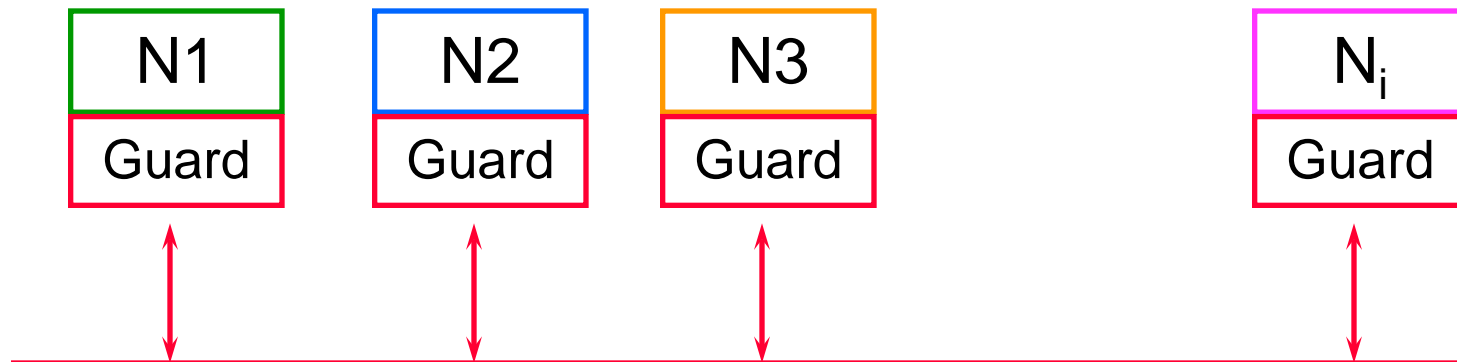
- Identifieraren “identifierar” meddelandet
 - ID x kan endast skickas av en viss nod pga arbitrering
- Standard CAN
 - 11 bitar- 2048 olika identifierare
- Enskilda meddelanden
 - **\$04B** Bromskommando
 - **\$1A2** Ljus meddelande, fram

Identifierare forts

- Extended CAN (29 bitar $\approx 0.5 \cdot 10^9$ identifierare)
- Försök till standardisering (anpassning J1939)
 - Standardiserade identifierare
 - \$1A29 92D0 HF parklampa
 - \$1A29 22D1 HF halvljuslampa
 - \$1A29 32D2 HF helljuslampa

Identifierare forts

- Filtrering
 - En CAN-krets initieras (i mask-registret) med ett antal identifierare som den skall känna igen



- Multicast
- Belastar ej processorn med oönskad data

Sammanfattning CAN

- Flexibelt
- Multimaster protokoll
- Multicast protokoll
- Hög överföringshastighet (1 Mbit/s)
- Kort svarstid (200 μ s)
- Automatisk omsändning av störda överföringar
- ”Atomic Broadcast”
- Stöd för synkroniserad exekvering i olika noder
- Avlastar processorn/användaren med meddelandeöverföringen