**Varför görs bit stuffing i CAN och hur går det till?**

**Vad innehåller fältet Payload Type i ett RTP-huvud? I vilket fält finns identifieraren SSRC? Vilken ordlängd har SSRC? Vad identifierar SSRC?**

**Beskriv hur noder styrs på fältbussar**

**Vilka är de tre versionerna av Profibus vad kännetecknar respektive?**

**Förklara kabelbeteckningen för Ethernet: [max. bithastighet] BASE/BROAD -Y [max. segmentlängd]**

**Vilka skikt ger Quality-of-Service (QoS)? Vilket skikt kan förbättra QoS? Vad innebär Best-effort Service?**

**Hur beräknas Slot time för en Ethernetversion?**

**Var placeras SSL i en Internet-stack? Vad ger SSL? Vad kallas SSLv3?**

**Vad är Controller Area Network (CAN)?**

**Vilka är de tre komponenterna (huvuddelarna, bl.a protokoll) i Internet-skiktet (nätverksskiktet) och vad gör dessa?**

**Vad bidrar till total nodfördröjning**

**(router och länk)?**

**Hur kontrolleras meddelandeintegriteten?**

* Payload Type innehåller **audio- och videoformat**, **samplingsfrekvens** och **bithastighet (kodningshastighet)**.
* SSRC finns i fältet som kallas **Synchronization Source Identifierer**. SSRC har ordlängden **32 bitar** och utgör **identifieringsnummer för mediet som avspelas**.
* För att skydda mot förväxling mellan dataramar och felramar. De senare innehåller felflaggor av typen **aktiv** (000000) eller **passiv** (111111).
* För att förbättra **synkroniseringen hos mottagarna**.
* Efter fem lika bitar insätts en extra bit med **motsatt värde** till de fem lika. Mottagarna räknar antal lika bitar och efter fem lika kastas den sjätte biten. (Den sjätte biten räknas förstås in i nästa sekvens av eventuellt lika bitar.)
* Profibus-FMS (Field Message Specification)  
  **Universell** fältbuss på **nivåerna fält och cell**.
* Profibus-DP (Decentral Peripheral)  
  **Snabb** fältbuss på **fältnivå**.
* Profibus-PA (**Process Automation**)  
  Ger egensäkert område, exempelvis inom kemisk och petrokemisk industri. På **fältnivå**.
* Masters styr slaves
* En passiv master kallas **monomastersystem**
* Två eller flera inaktiva masters kallas **multimastersystem**
* Masters kan utgöras av PC och PLC:er (på cellnivå)
* Slaves är typiskt I/O-enheter (för givare och ställdon) och PLC:er (på fältnivå)
* QoS bestäms av skikten **värd-till-nät (fysiskt och länk)** och **Internet-skiktet (nätverksskiktet)**. QoS kan förbättras i **transportskiktet**,**exempelvis genom att välja TCP istället för UDP**.
* Best-effort Service innebär att **routrarna distribuerar paket så snabbt som möjligt** och detta i kombination med att använda **det snabba UDP**. Best-effort Service ger **ingen garanti för tidsfördröjning och jitter**.
* [max. bithastighet] ges i **Mbps**
* BASE är **basband**, dvs. utan FDM
* BROAD är **bredband**, dvs. med FDM
* Y anger transmissionsmediet om inte **koaxialkabel** används
* [max. segmentlängd] ges i hundratals **meter**
* Eftersom SSL är en **socket**placeras den mellan **applikationsskiktet och transportskiktet**.
* SSL ger **krypterad kommunikation** exempelvis för e-handel och bankärenden.
* **Transport Layer Security (TLS) = SSLv3**

Slot time = **2**\*utbredningstiden + säkerhetsmarginal där utbredningstiden bestäms för den **längsta** nättypen för Ethernetversionen. Ett exempel är utbredningstiden över 5\*500 = 2500 **m tjock koaxialkabel (RG11)** för versionen med bithastigheten 10 Mbps. Multiplikationen med **2** innebär att Slot time beräknas för "tur och retur", tur för dataramen och retur av jam. Detta ger Slot time **= 51,2 μs.**

1. **IP**: Kommunikation mellan värden

Format på paket

Adresseringen

1. **ICMP**: Felmeddelanden

Routersignaler

1. Routingprotokoll, inkl routingtabeller t.ex. RIP, OSPF(Open Shortest Path First), BGP

* Utvecklades av **Bosch** för att ersätta kabelsystem i fordon.
* CAN används i Volvofordon och kallas **Volcano.**
* CAN används i ABB:s robotar och kallas **DeviceNet.**
* CAN används i Atlas Copcos **gruvmaskiner.**
* Bussen består av ett par ledare, CAN\_L och CAN\_H.
* Mottagen signal är differensen **CAN\_L – CAN\_H** som omvandlas för att matcha övrig elektronik

1. Sändaren beräknar H(m), där m är meddelandet och H(.) är **hashfunktionen**.
2. Sändaren skickar m + H(m), där + är **konkatenering**.
3. Mottagaren **separerar** inkommande meddelande (m') från **hashkoden**.
4. Mottagaren beräknar H(m').
5. Mottagaren testar om **H(m') = H(m)**