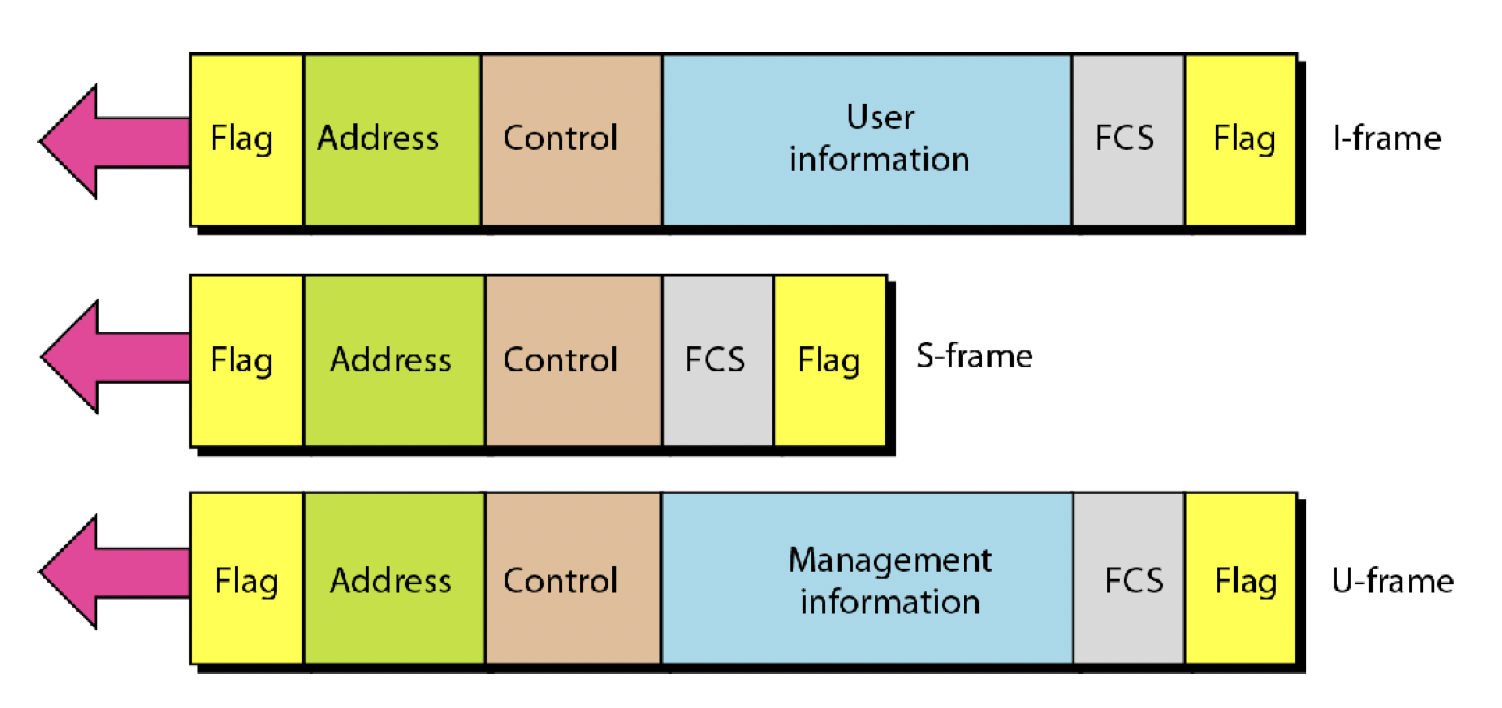
*Föreläsning 8*

*(Punkt-till-punkt-access)*

**HDLC - High Level Data Link Control**

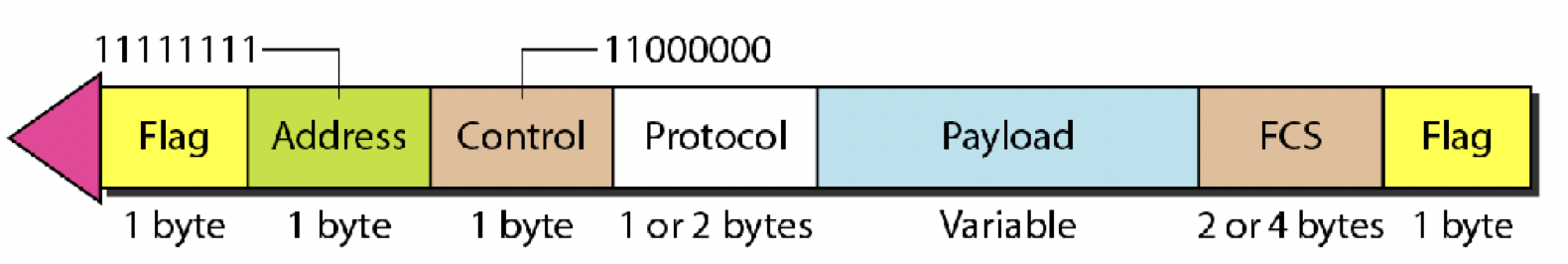
* OSI-nivå - Datalänk lagret
* Både för punkt-till-punkt- och multipunkt-förbindelser
* **Primär** (skickar **kommando**) och **sekundär** station (ger ett **response** i gensvar)
* 3 olika ramtyper (kallas ramar i OSi-nivå 2)
  + **I-frame**, data
    - Förmedlar datan vi vill skicka
    - Dvs datan som kommer från OSI-nivå 3 (ex. ett IP-paket)
    - Piggybacking (snålskjuts): Ett ack läggs in i paketet, för flödes- och felkontroll
  + **S-frame** (Supervisory), ack
    - För flödes- och felkontroll
    - Innehåller bara ett ack
  + **U-frame** (Unnumbered), intern nätverkskontroll



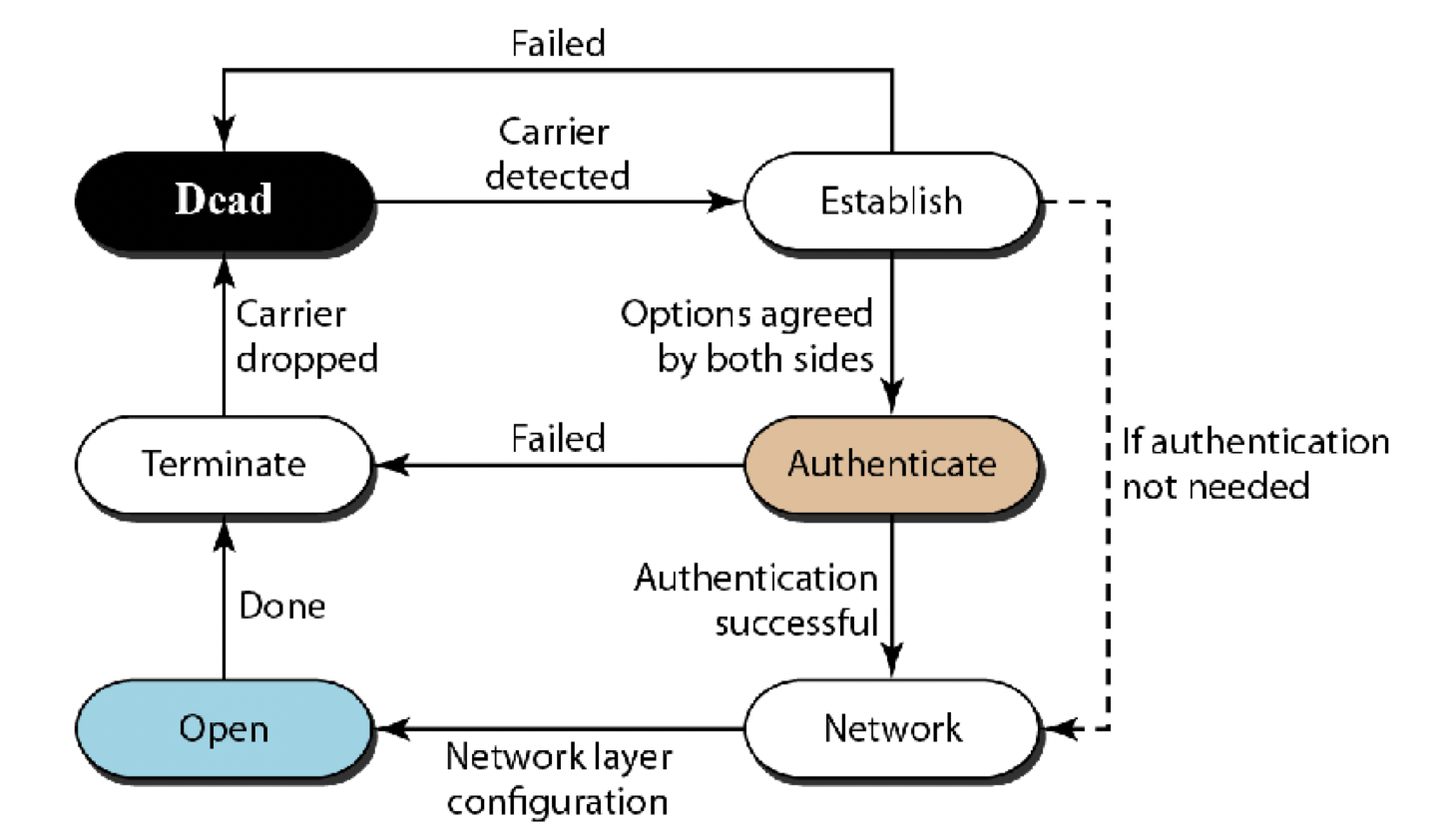
* Flagfält (8 bitar)
  + För synkronisering
  + Nu ankommer ett nytt paket till mottagaren
* Addressfält (till vem)
* Styrfält (paket 7, jag förväntar mig 8 av dig)
* Informationsfält - data
* CRC - felkontrolfält

**PPP - Point-to-point Protocol**

* OSI-nivå 2
* Vanligaste protokollet för p-p-förbindelser
* Stödjer **INTE** punk-till-multipunkt!!!
* Används för kontakt mellan användare och internetleverantör
* PPP-ram
  + Flagga: För synkronisering
  + Address: Behövs EJ, pga det endast finns en mottagare
    - Unicast
    - Multicast
    - **Broadcast ⇒ Alla bitar är 1:or i PPP**
  + Control: Behövs EJ
  + FCS, är CRC (cyclic redundancy check)



* Bitstuffing
  + Sändare: 0:a sätts in efter 5 stycken 1:or
  + Mottagare: ??
* Tillståndsgraf för PPP
  + Utgångspunkt i **dead**
  + Authenticate är valfritt
  + I **network** förhandlar mottagare och sändare, exempelvis om vilken hastighet



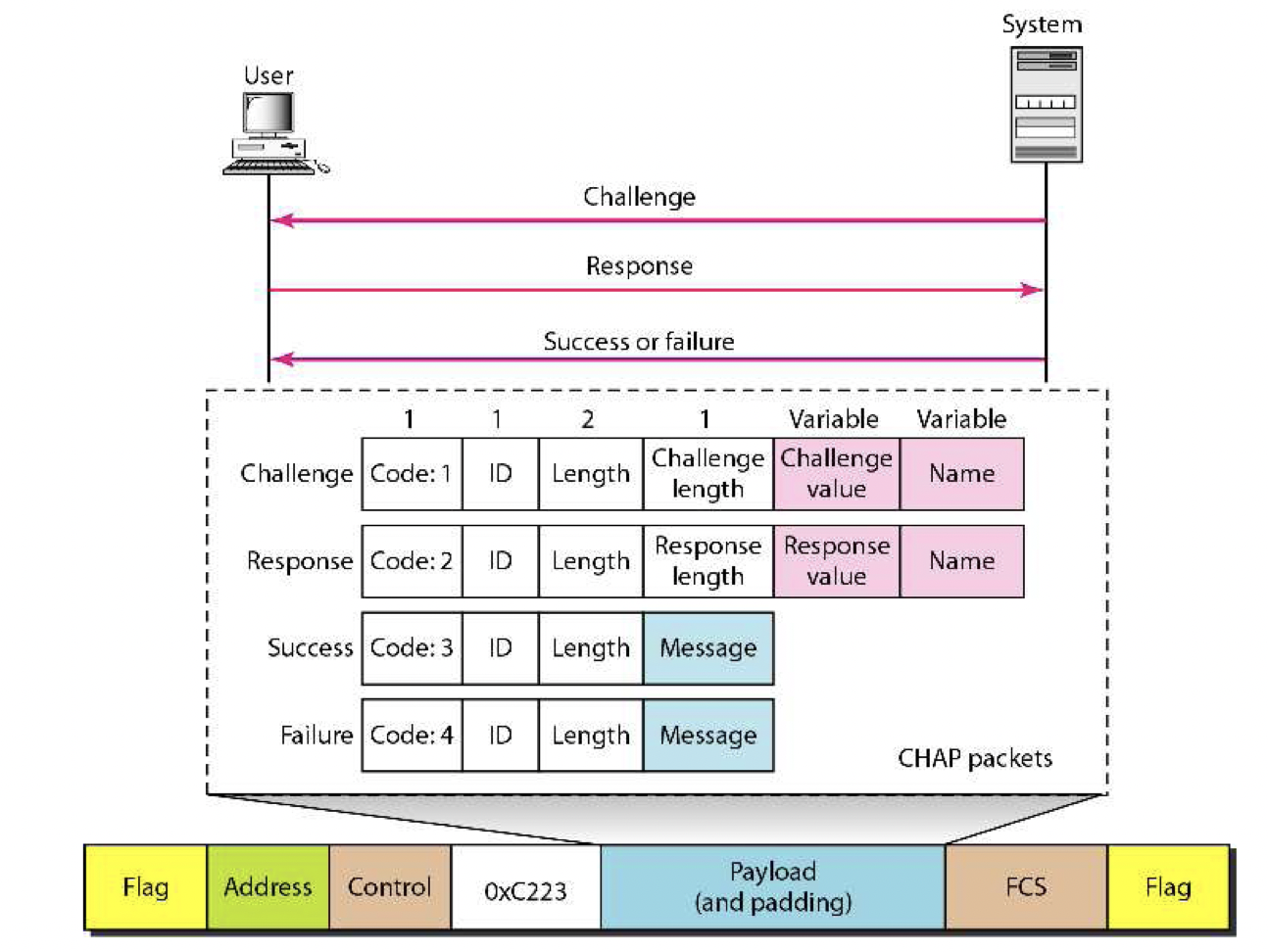
* PPP har 3 olika underprotokoll (varav 2 är för autentisering, vilka är valbara)
  + **1. LCP**
    - **Kod**: C021
    - **ID**: till för att matcha förfrågan och svar
    - **Längd**: LCP-paketets (totala) längd
    - **Information**: Extra info



* + **2. PAP** (autentiseringprotokoll) ⇒ Valbar
    - **Kod**: C023
    - Användarnamn och lösen
    - Bekräfta/avfärda



* + **3. CHAP** (autentiseringprotokoll) ⇒ Valbar
    - Liknelse: Inloggning med bankdosa
    - **Kod**: C223



**ADSL**

* **Uppströmskanaler**: Från oss till nät
* **Nedströmskanaler**: Från nät till oss
* ⇒ **Asymmetrisk**. Inte lika mycket trafik upp som ned
* För varje signal som skickas, skickas **(längd, vinkel)**
* **Splitter**, delar upp tal och data genom filtrering
  + Lågpassfilter, plockar ut de lägsta kanalerna, skickar ut till telefonnätet
  + Högpassfilter gör om till digitala signaler, skickar ut till internet

**Multimedia & QoS**

* Multimedia: bild, text, ljud, etc
* QoS: Hur pass bra tjänsten kan användas på nätet

**Multimedia networking applications**

* **Strömmande media**: Ljud/video börjar sändas **INNAN** **ALL** information är framme (vi börjar titta innan allt är framme)
* **Filnedladdning**: Ljud/video börjar sändas när **ALL** information är framme
* 3 applications
  + Stored streaming (e.g. netflix)
  + Live streaming (fotbollsmatch)
  + Interactive, real-time
* Problem:
  + känsliga för fördröjningar (end-to-end)
  + Delay jitter ⇒ Variability of packet delays within same packet stream

**Streaming stored media**

* Allt finns hos sändare
* Vi sparar i uppspelningsbuffert, sen kommer allt efterhand
* Pausa, rewind, FF

**Streaming Live media**

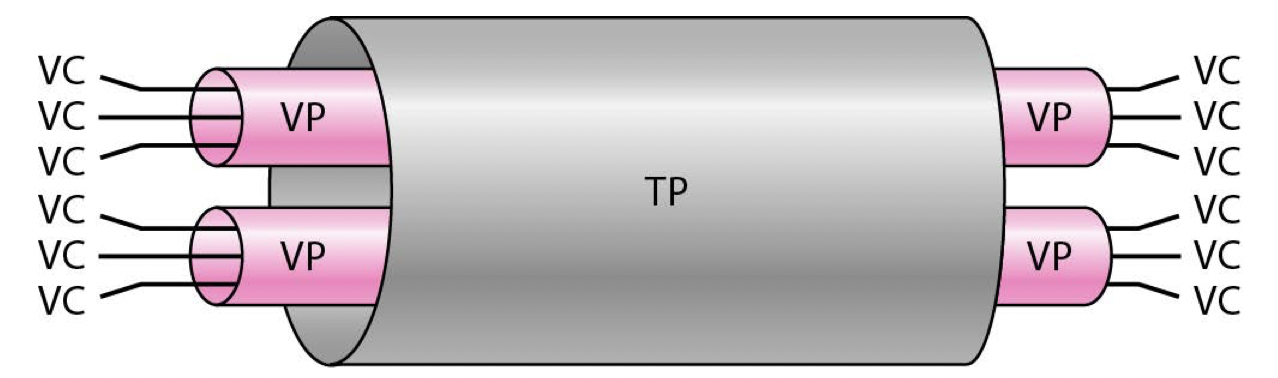
* Vi sparar i uppspelningsbuffert
* FF omöjligt

**Real-time interactive media**

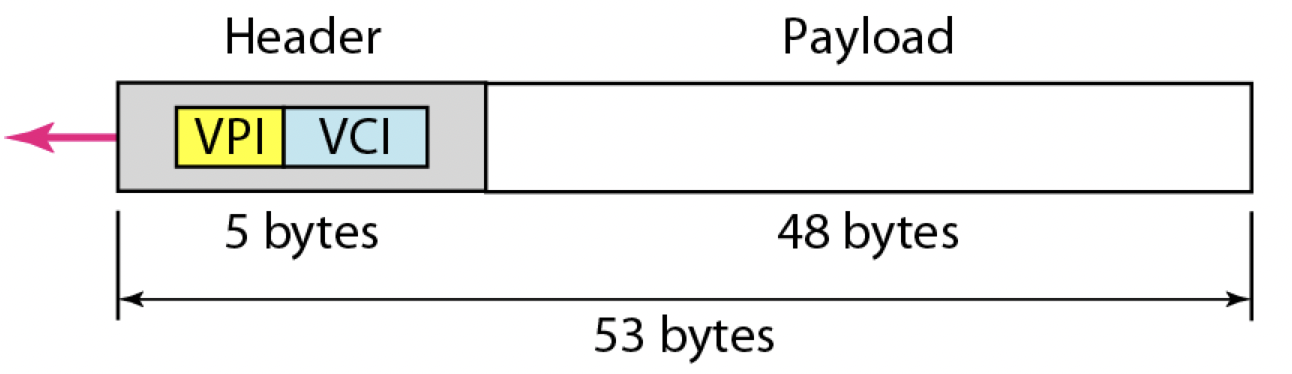
* End-to-end delay requirements
  + Audio delay, a
    - 150msec < a < 400msec

**ATM - Asynchronous Transfer Mode**

* Bara en ren informationsmotorväg
* Man skickar små paket (=celler) med en “fixed” längd, istället för ramar
* Multiplexering med ATM
  + Cellerna har en fix storlek
  + Man plockar den första i varje “kö”
  + Varje lucka (slot) behöver inte fyllas upp
  + Varje paket har head + data
  + **⇒ EFFEKTIVARE** än TMD (tidsmultiplexering)
* Indelning av förbindelser
  + **Transmission path**
    - **Virtual path**
      * **Virtual circuit**

****

* Varje virtual circuit identifieras med 2 tal
  + VPI
  + VCI
* Cell
* Totalt: 53 bytes
* Huvud (VPI + VCI): 5 bytes



* 3 lager
  + **1. Application Adaption layer (AAL)**
    - Beroende på applikation (text, vild, ljud känsligt för fördröjningar)
    - ⇒ Gör om till ATM
    - **AAL - 5 olika varianter**
      * AALl - Tar emot konstant bitström, ljud och video
      * AAL2 - Trafik med låg bithastighet
      * AAL3/4 - För förbindelseorienterade respektive förbindelselösa tjänster
      * AAL5 - Som AAL3/4 men enklare kontroll
  + **2. ATM-lagret**
    - Här läggs cellens header på
  + **3. Fysiska lagret**
* **Växlar**
  + Endast de 2 nedre lagren utnyttjas i växlarna
* Operatörer tar betalt på 3 olika vis (sortering: dyrast först)
  + **CBR** = Constant bit rate
  + **VBR** = Variable bit rate
  + **ABR** = Available bit rate
    - Det som är ”kvar”