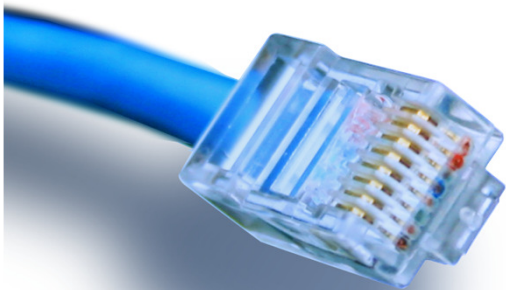


HOUSE OF  
TECHNOLOGY



- en del af **mercantec**<sup>+</sup>



# TCP & UDP

- de 'transportansvarlige' på lag 4

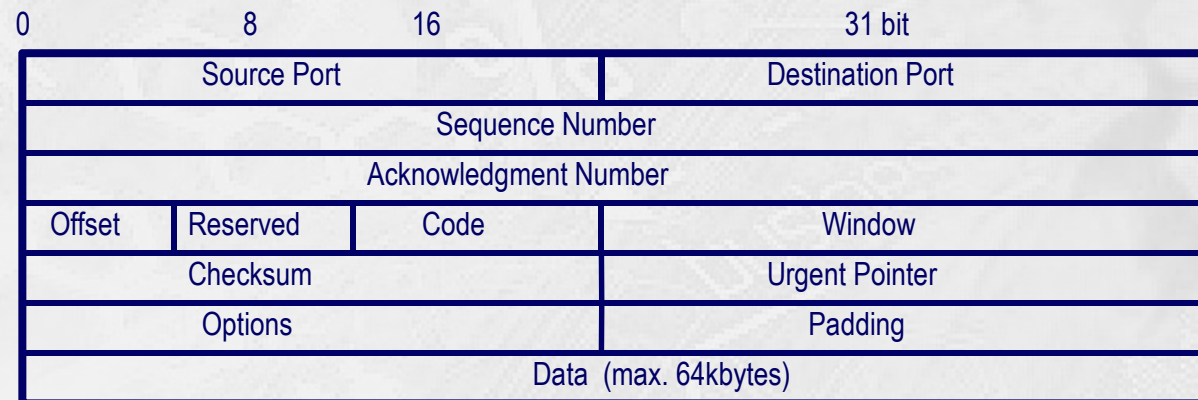
## Netteknik 1

# TCP & UDP

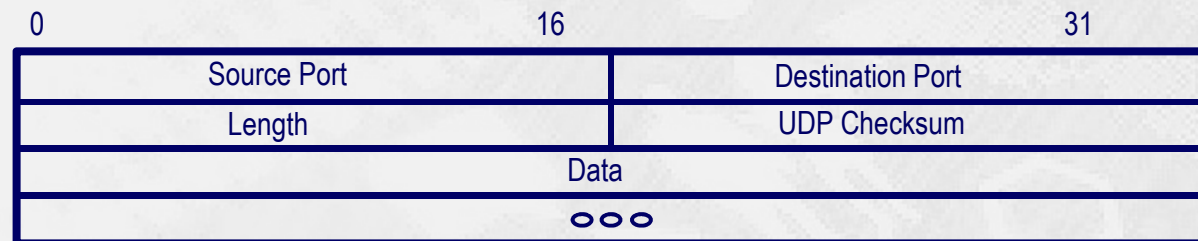


TCP og UDP er begge netværksprotokoller til transport, med hver deres header-information i pakken (segmentet):

TCP:



UDP:



- **Transport lagets** funktioner:

- Opbygger data transport forbindelser, som er transparente for de øvre lag.
- Forbindelserne kan betragtes virtuelle, punkt-til-punkt forbindelser.
- De 2 almindeligste protokoller er
  - TCP (Transmission Control Protocol)
  - UDP (User Datagram Protocol)
- De adresserer begge applikationer eller services vha. portnumre

Lag 7	Applikation
Lag 6	Præsentation
Lag 5	Session
Lag 4	Transport
Lag 3	Netværk
Lag 2	Data Link
Lag 1	Fysisk

- **Transmission Control Protocol**
  - Opbygger **virtuelle forbindelser** mellem slutbruger applikationer
- Karakteristisk for TCP er:
  - Forbindelsesorienteret, med virtuelle forbindelser
  - Sikker transport
  - Opdeler data som skal sendes i segmenter
  - Samler data segmenterne igen hos modtageren
  - Der kvitteres for alle data og hvis data ikke er modtaget sendes de igen

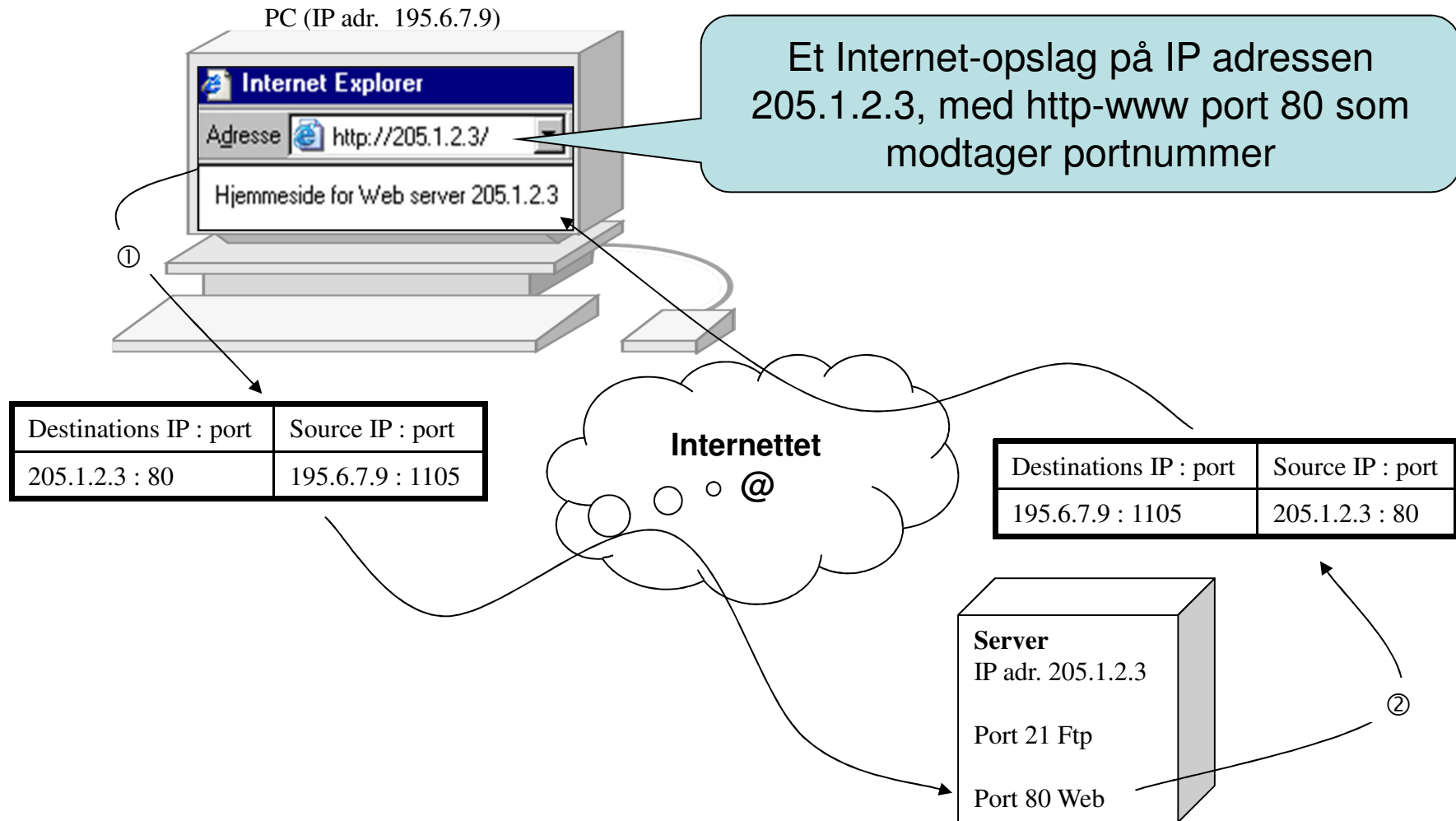
- **User Datagram Protocol**
  - Transportere data **hurtigt og effektivt** mellem hosts, men der **garanteres ikke** for om data afleveres
- Karakteristisk for UDP er:
  - Forbindelsesløs, men hurtig i forhold til TCP
  - Usikker - ingen garanti for at data afleveres
  - Samler ikke data segmenterne igen hos modtageren
  - Ingen flowkontrol
  - Hverken fejlkontrol eller fejlkorrektion



# TCP og UDP portnumre

- Et datagram finder modtageren og modtagernetet vha. IP adressen.
- Adressering af applikationer i den enkelte host udføres vha. port numre i TCP og UDP protokollerne.
- En IP adresse med tilhørende port numre kaldes en socket.
- Et port nummer kan have en værdi fra 0–65535, hvor porte fra 0–1023 kaldes kendte porte hvilket vil sige at deres brug er fastlagt (fx 21 ftp port).
- Porte kan være destinations og source porte:
  - En destinations port beskriver hvor i modtager maskinen datagrammet skal hen, dvs. hvilket program eller service det skal adresseres til.
  - Source porte anvendes til at beskrive afsenderens sessions nummer og betyder at man fx kan se på flere hjemmesider (sessioner) på en gang. Source porten vælges af afsenderen og kan være alle ubrugte / ukendte porte, minus destinations porten.
- Den officielle liste over portnumre administreres af IANA og kan ses på hjemmesiden [www.iana.org/assignments/port-numbers](http://www.iana.org/assignments/port-numbers)

# Eksempel

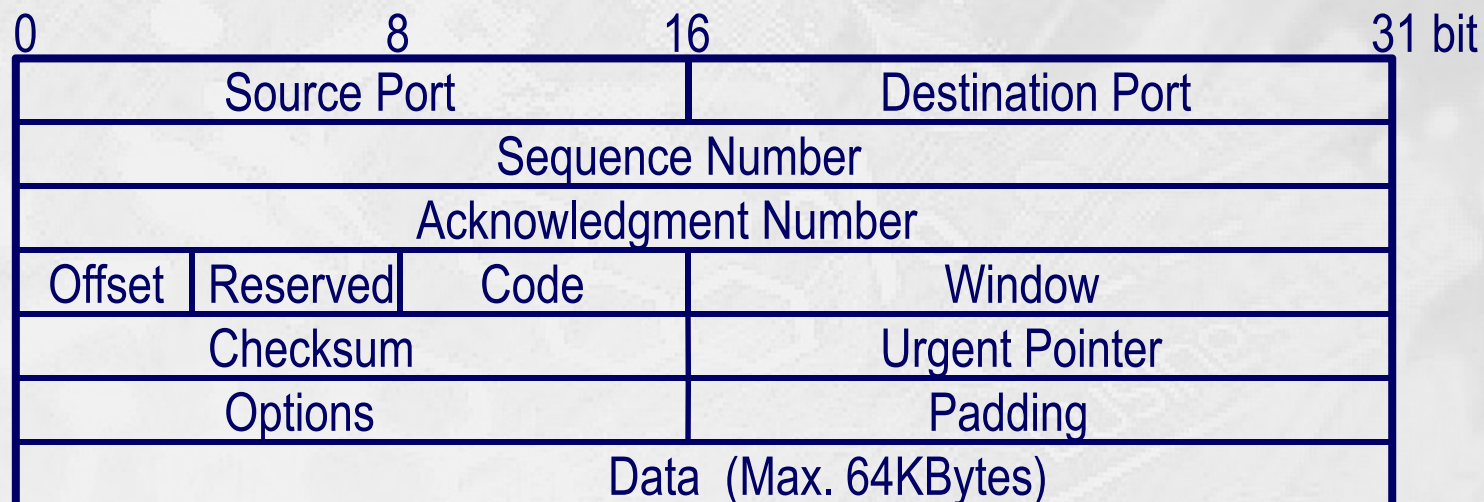


# De mest kendte portnumre

Navn	Port nr.	Protokol	Beskrivelse af service
Ftp	21	tcp	File Transfer Protokol
Telnet	23	tcp	Telnet remote login
Smtpt	25	tcp	Simple Mail Transfer Protokol
Domain	53	udp	Domain Name Server
Bootps	67	udp	Bootstrap Protocol Server / DHCP server
Bootpc	68	udp	Bootstrap Protocol Client / DHCP client
Tftp	69	udp	Trivial File Transfer Protocol
Www-http	80	tcp	World Wide Web http
Pop3	110	tcp	Post Office Protocol – Version 3
Nntp	119	tcp	Network News Transfer Protocol
Netbios-ns	137	tcp	NETBIOS Name Service
Netbios-ns	137	udp	NETBIOS Name Service
Netbios-dgm	138	tcp	NETBIOS Datagram Service
Netbios-dgm	138	udp	NETBIOS Datagram Service
Netbios-ssn	139	tcp	NETBIOS Session Service
Netbios-ssn	139	udp	NETBIOS Session Service
	1023		
Frie porte	1024-65535		Kan bruges af alle

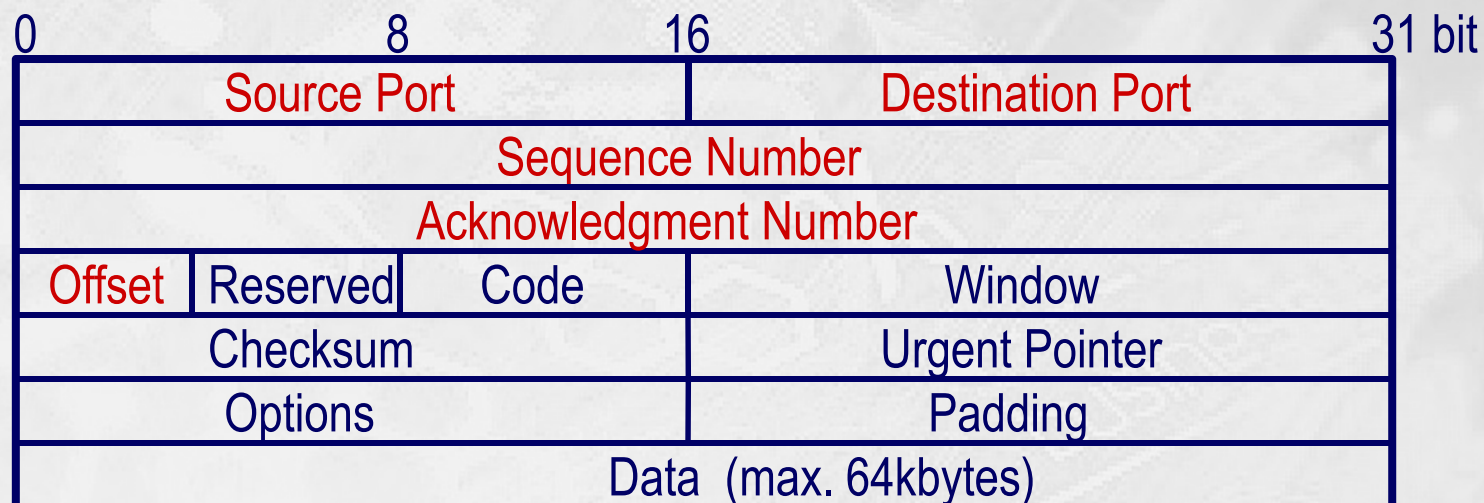


# Segmentet



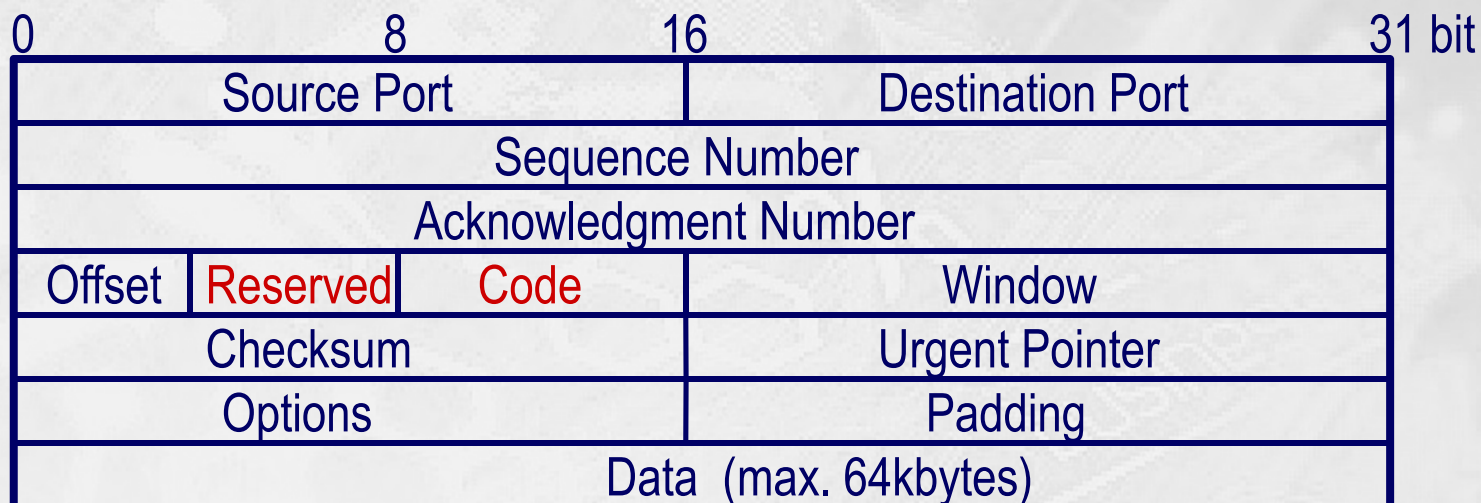
- Alle TCP segmenter opdeles i et antal header-felter, efterfulgt af et stort datafelt.
- Datafeltet er på max. 64KB, hvilket svarer til at et TCP segment præcis kan være i datafeltet i et IP-datagram.

# Porte og Sequence



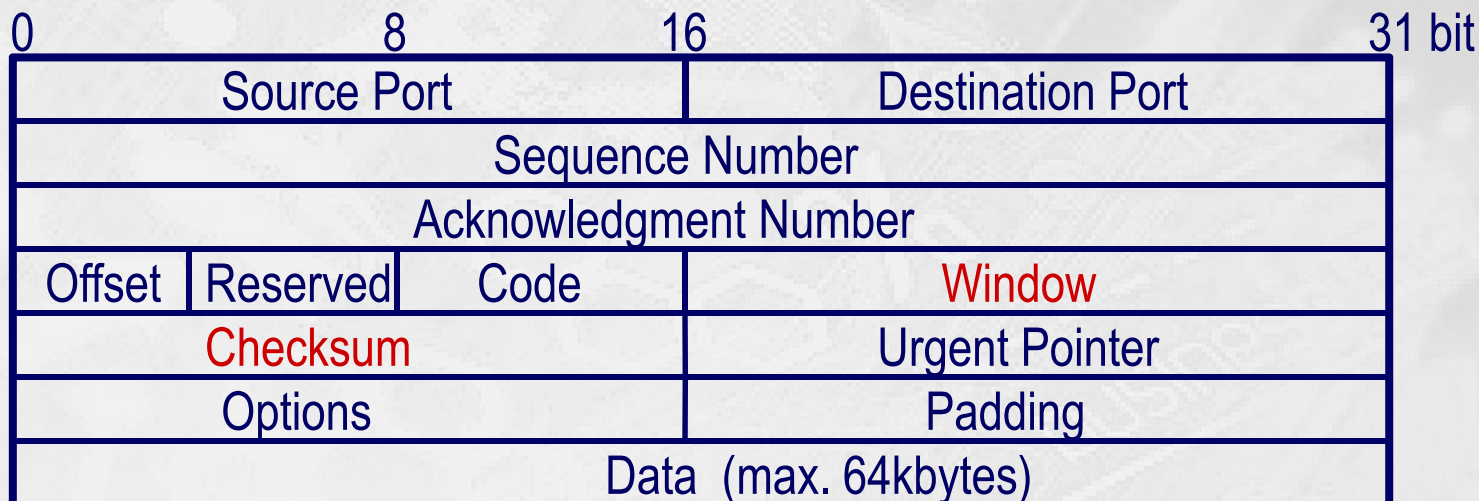
- **Source** og **Destination** porte identificerer applikationen i hver ende af forbindelsen.
- **Sequence Number** er sekvensnummeret på den første byte i det aktuelle data felt. Angiver segmentets nummer i rækkefølgen.
- **Acknowledgment Number** er et kvitteringsnummer som angiver nummeret på det sekvensnummer som senest er modtaget plus 1.
- **Offset** angiver størrelsen på TCP headeren, som er variabel alt efter options feltets længde.

# Code



- **Reserved** feltet består af 3 bits reserveret til fremtidige formål
- **Code** feltet består af 6 bits, med hver deres funktion:
  - URGeNT: Indikerer at data i dette segment er særlig vigtige
  - ACKnowledgement: Viser at der er tale om en kvittering
  - PuSH: Viser at data i dette segment skal sendes lige nu
  - ReSeT: Hvis der har været en fejl, viser bittet at sessionen skal genoprettes
  - SYNchronize: Anvendes under etablering af en forbindelse
  - FINish: Anvendes til at afslutte en forbindelse

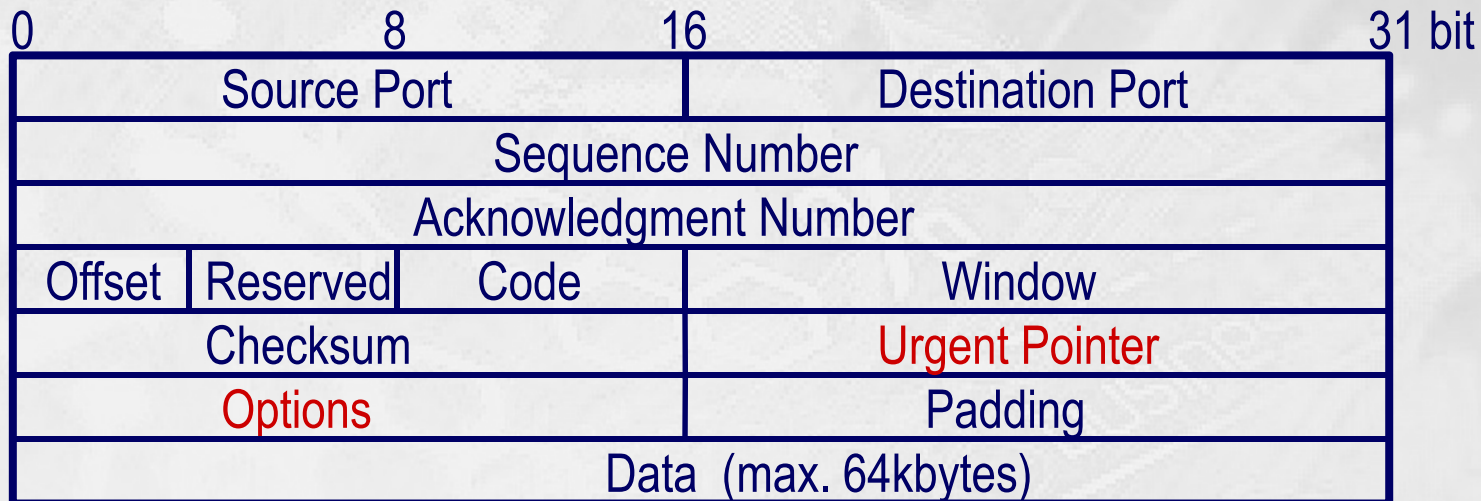
# Window & Checksum



- **Window** feltet:
  - Angiver hvor mange bytes afsenderen kan acceptere før den sender en kvittering.
  - Det er et mål for hvor stor modtagebufferen er på hosten som skal modtage data.
  - Vindues størrelsen kan ændres løbende i TCP, funktion kaldes sliding windows.
  - Window feltet er sammen med Sequence og Acknowledge nummeret TCP's flow kontrol.
- **Checksum** feltet er checksummen beregnet ud fra segmentet og headeren.

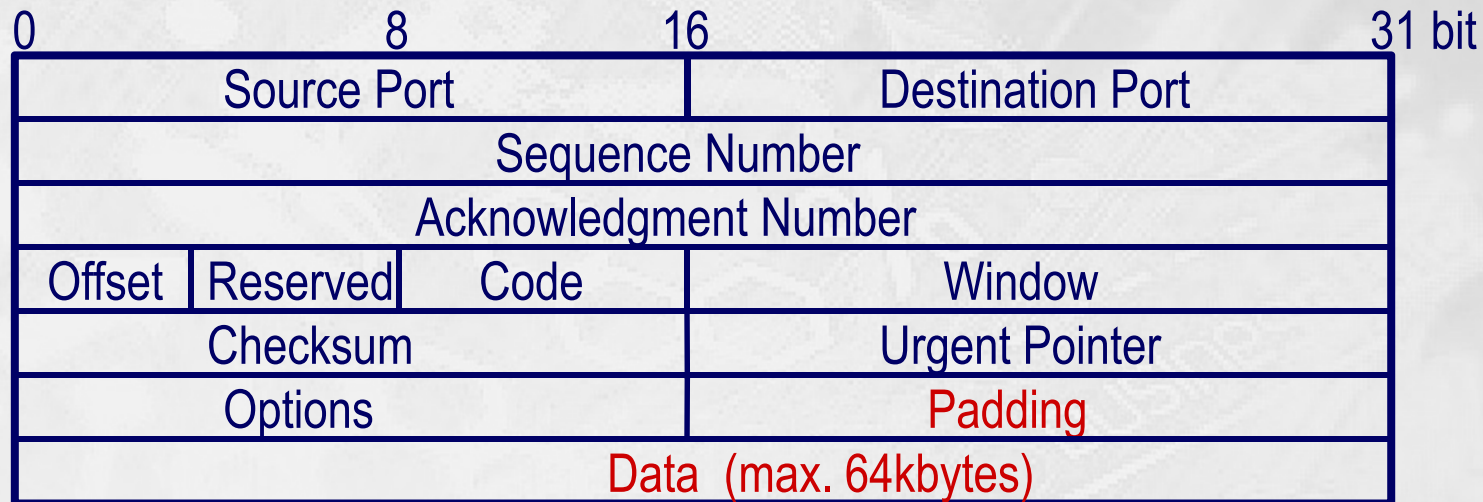


# Urgent Pointer & Options



- **Urgent Pointer** anvendes sammen med sekvens nummeret til at angive hvor vigtige data fx tastatur tryk starter og slutter
- **Options** er et felt af variable længde og bruges fx til at åbne for muligheden for vinduer der er større end 65536 bytes og måle latenstiden på forbindelsen

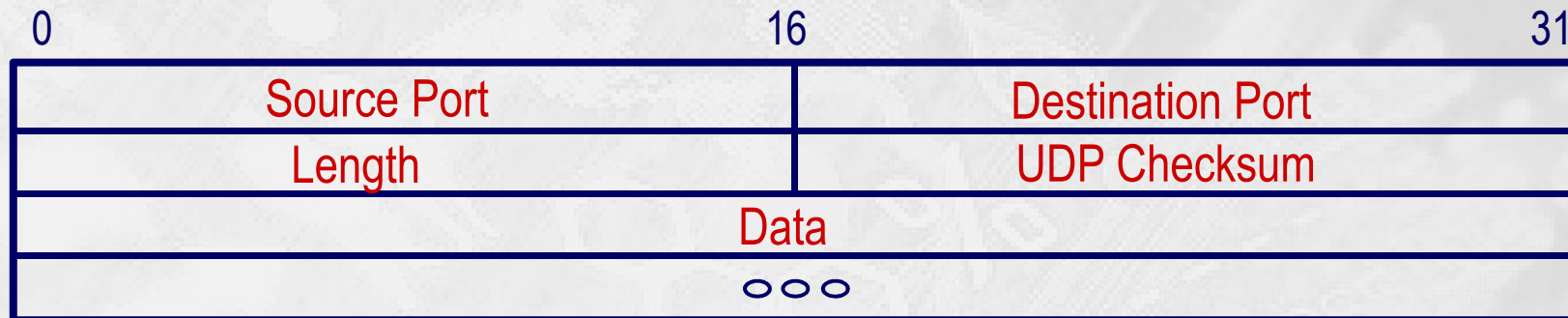
# Padding & Data



- **Padding** anvendes til at 'fylde efter' med 'dummy data' så headerfeltets længde bliver præcis et helt antal 32 bits felter (words).
- **Data** indeholder brugerdata fra et eller andet program på en maskine, som har haft behov for at sende information ud over nettet via TCP/IP systemet. Maksimal datamængde er 64 Kbytes.



# UDP Segmentet



- Source og destination porte identificere applikationen i hver ende af forbindelsen:
  - **Destination Port** angiver protokollen i laget over som skal modtage datagrammet.
  - **Source Port** kan anvendes (sættes til 0 hvis den ikke bruges) og viser hvor datagrammet kommer fra.
- **Length** er antallet af bytes datagrammet fylder incl. headeren.
- **Checksum** kan anvendes efter behov, men sættes blot til 0 hvis den ikke bruges.
- **Data** indeholder brugerdata fra et eller andet program på en maskine, som har haft behov for at sende information ud over nettet via TCP/IP systemet. Maksimal datamængde er 64 Kbytes.