

---

# HTTP & WEBSERVERAR

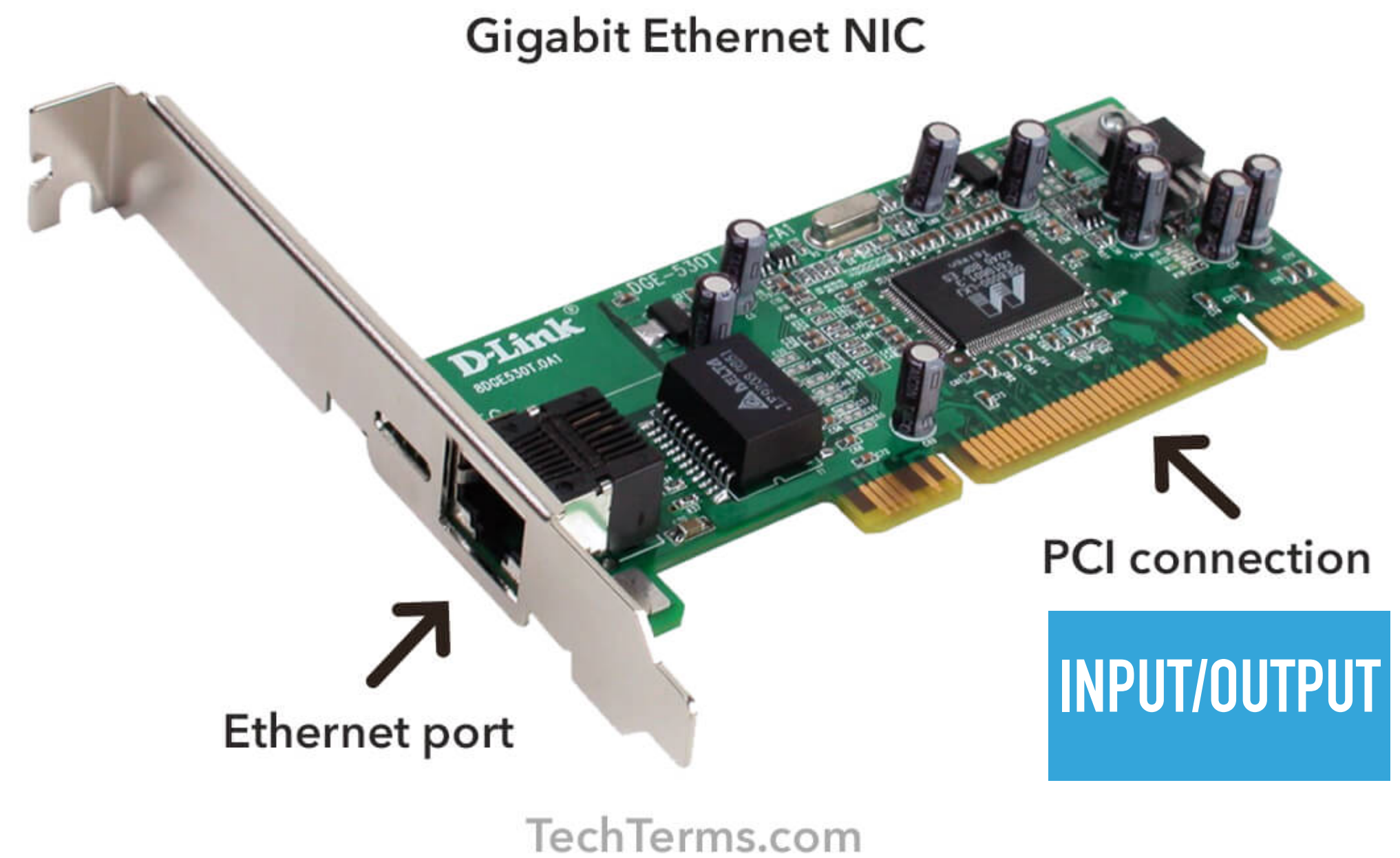
---

# AGENDA

- ▶ Networking TCP/IP protokoll
- ▶ Webb HTTP protokoll
- ▶ Localhost
- ▶ Klient i JavaScript
- ▶ Server i JavaScript

# NETWORKING – TCP/IP

- ▶ NIC (Network Interface Card) tillhandahåller nätverks kommunikation
  - ▶ Via kabel (Ethernet)
  - ▶ Trådlöst (WiFi genom LAN)



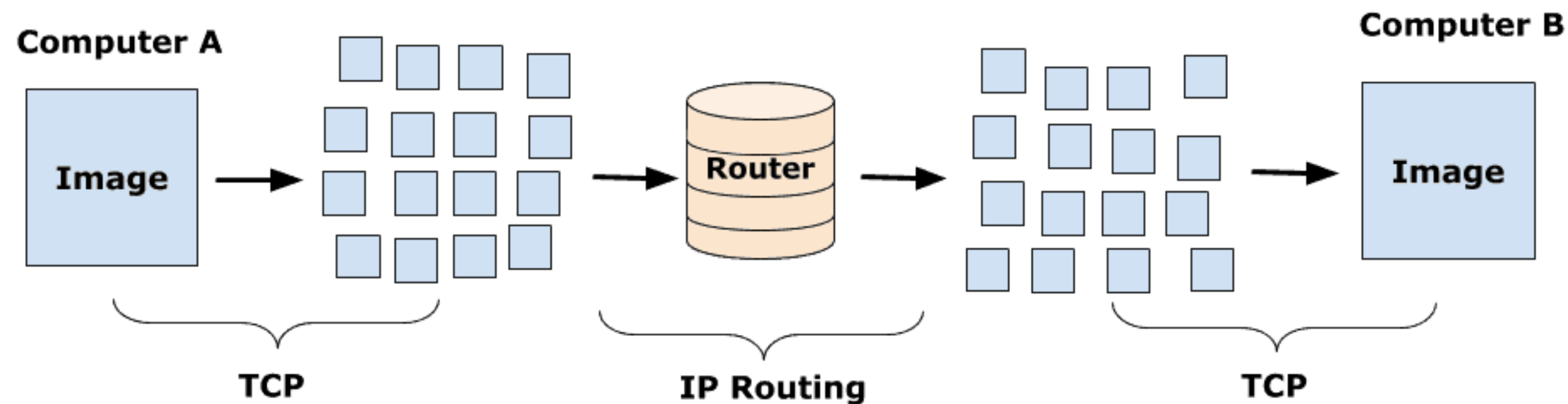
---

# NETWORKING – TCP/IP

- ▶ Datorer sänder hela tiden meddelanden till varandra via NIC
  - ▶ En fil
  - ▶ En hel webbsida "<html>....</html>"
  - ▶ Ett email "From: 'Bob Example' <bob@example.org>... Hej Alice,..."

# NETWORKING – TCP/IP

- ▶ Det finns fysiska begränsningar på hur mycket som kan skickas över "nätet"
- ▶ Meddelanden delas upp i IP-paket



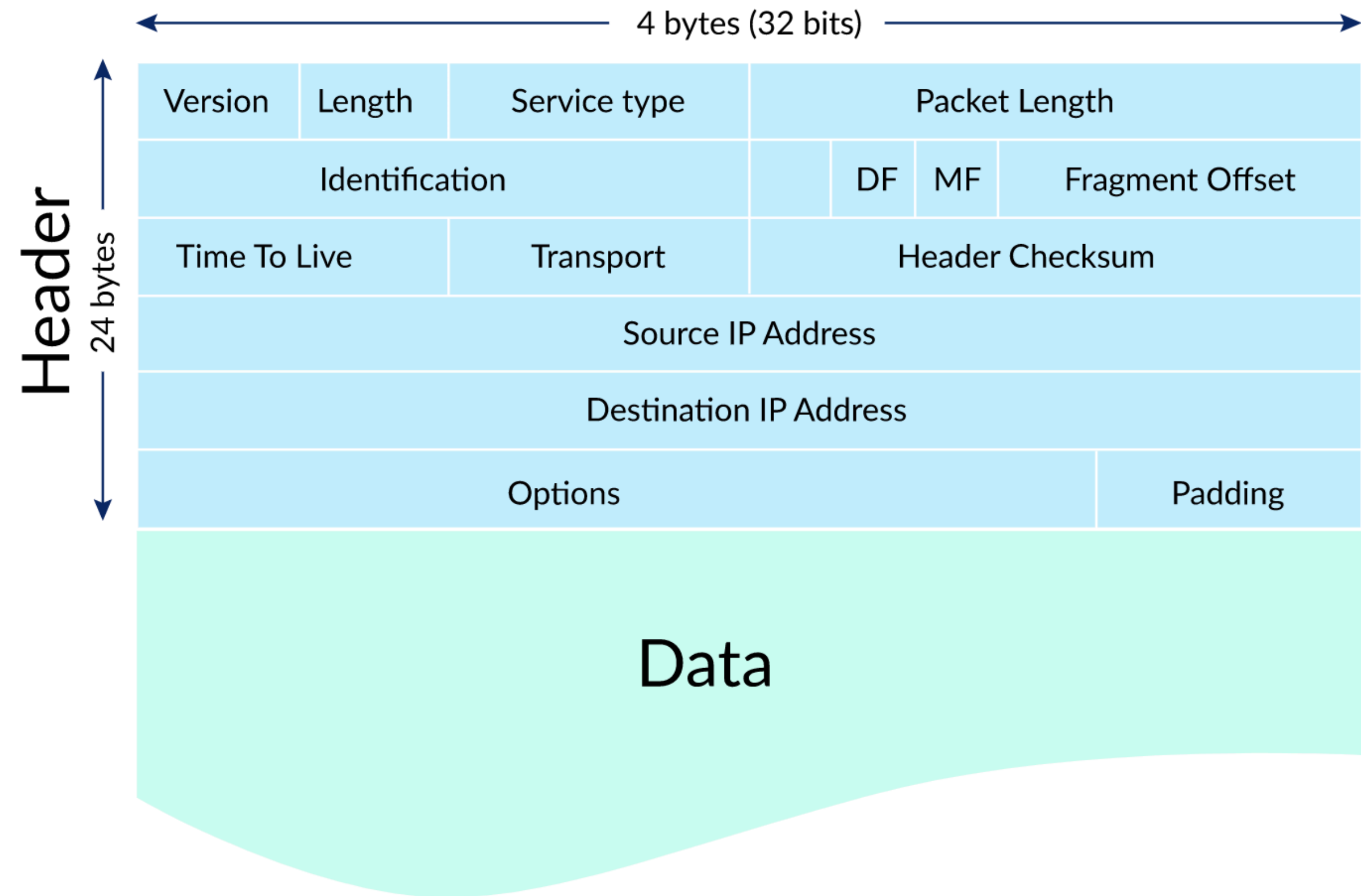
---

## NETWORKING – TCP/IP

- ▶ Det är svårt överskatta vikten av TCP/IP protokollen
- ▶ Underliggande protokoll för internet
  - ▶ Paketcentrerat (basenhet för information)

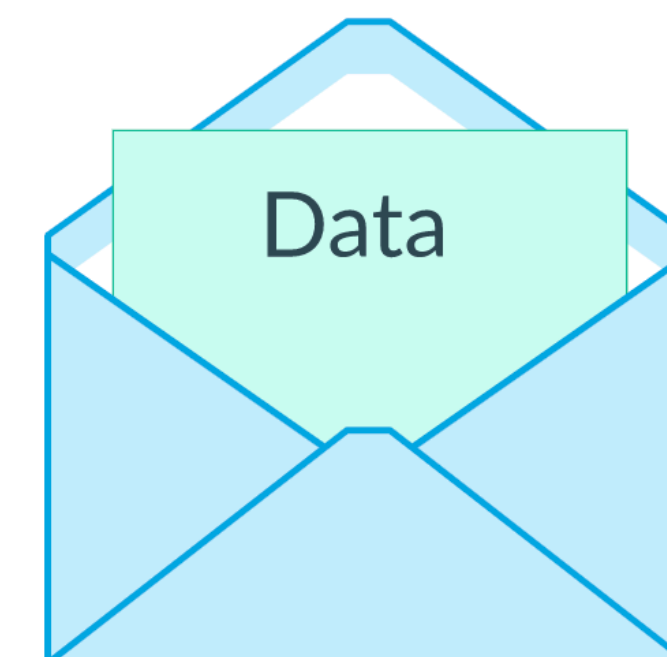
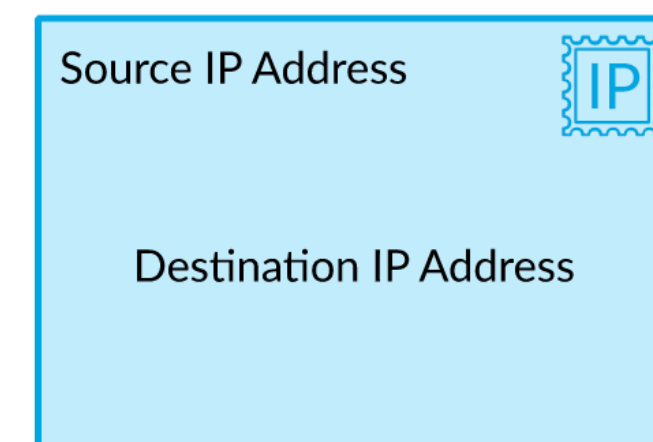
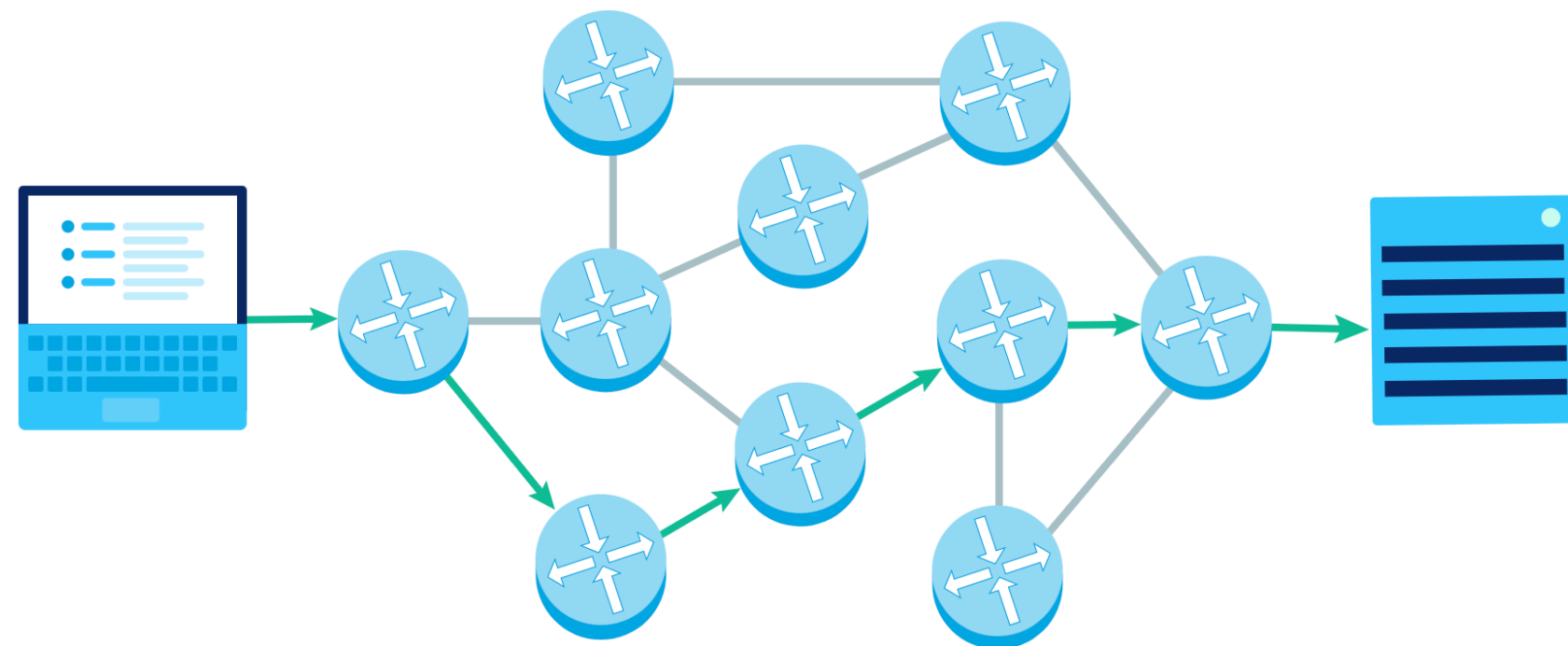
# NETWORKING – TCP/IP

- ▶ IP (Internet Protokoll)
  - ▶ IP-paket
    - ▶ Header (20/24 bytes)
      - ▶ Source/Destination
      - ▶ Fält för routing
    - ▶ Data



# NETWORKING – TCP/IP

- ▶ IP-paket
  - ▶ Precis som posten skickar mellan olika centraler och brevlådor
    - ▶ Header läses av flera routrar
    - ▶ Datan läses bara av mottagare





---

# NETWORKING – TCP/IP

- ▶ Paket har flera adresstyper som används tillsammans
  - ▶ Hostname - Människovänlig adress (översätts till IPv4/IPv6)
    - ▶ *codic.se*
  - ▶ IPv4/IPv6 - IP-adress som används av mjukvara (översätts till MAC)
    - ▶ *192.34.67.2*
  - ▶ MAC (Media Access Controll) - Fysisk adress som används av hårdvara (NIC)
    - ▶ *8C:87:91:49:7A:DE*

---

## NETWORKING – TCP/IP

- ▶ IP-adresser säger bara till vilken **host** ett paket ska (***vilket hus***)...
- ▶ Inget om vilken **process** (***person i hushållet***)?

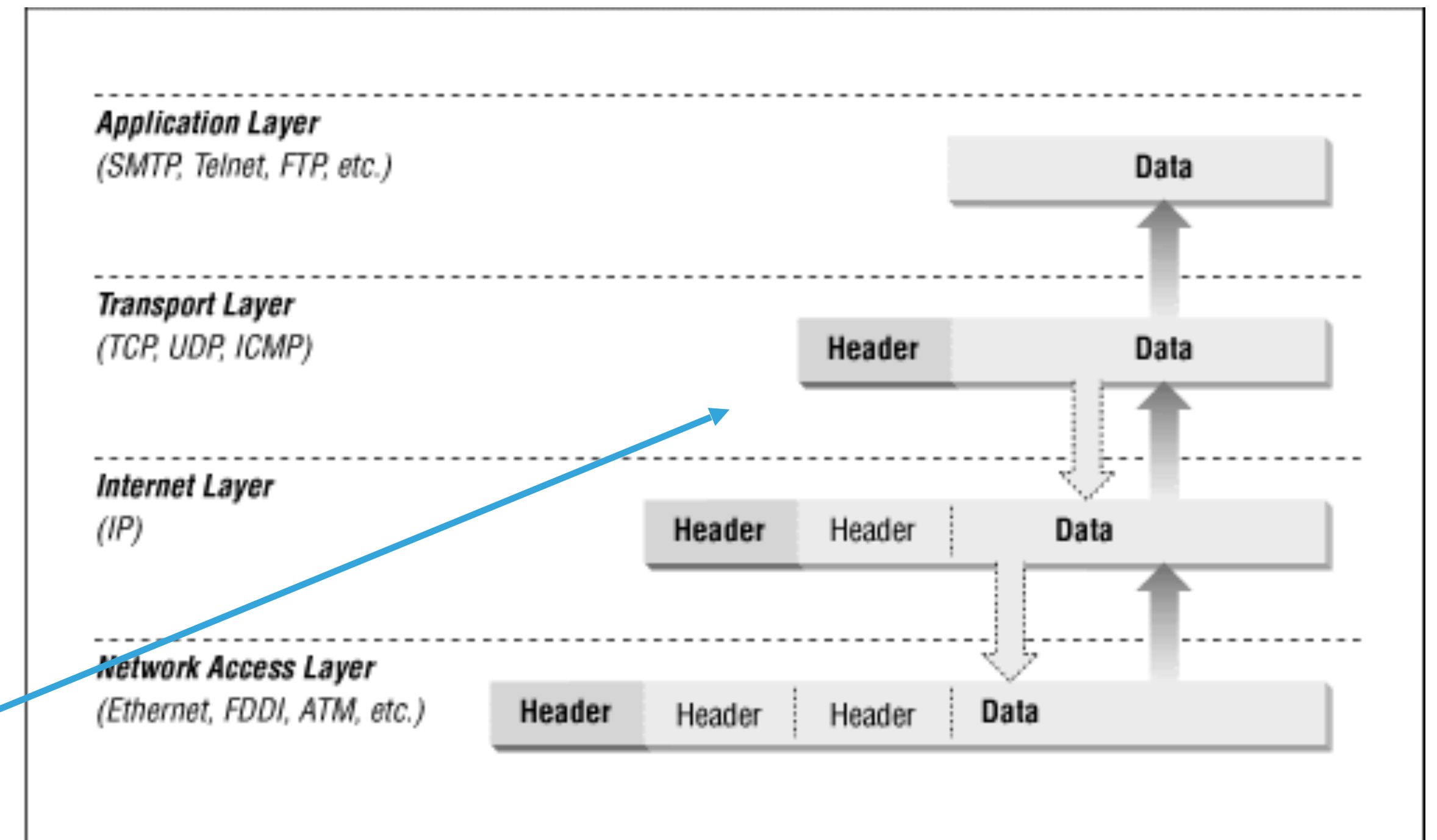
---

# NETWORKING – TCP/IP

- ▶ Portar bestämmer vilken nätverksprocess som ska nås (TCP)
  - ▶ SSH: 197.8.43.77:22
  - ▶ **HTTP: 197.8.43.77:80**
  - ▶ **HTTPS: 197.8.43.77:443**
  - ▶ SMTP: 197.8.43.77:25
  - ▶ FTP: 197.8.43.77:21
- ▶ Dessa är standardiserade program
  - ▶ Har därför dedikerade portar

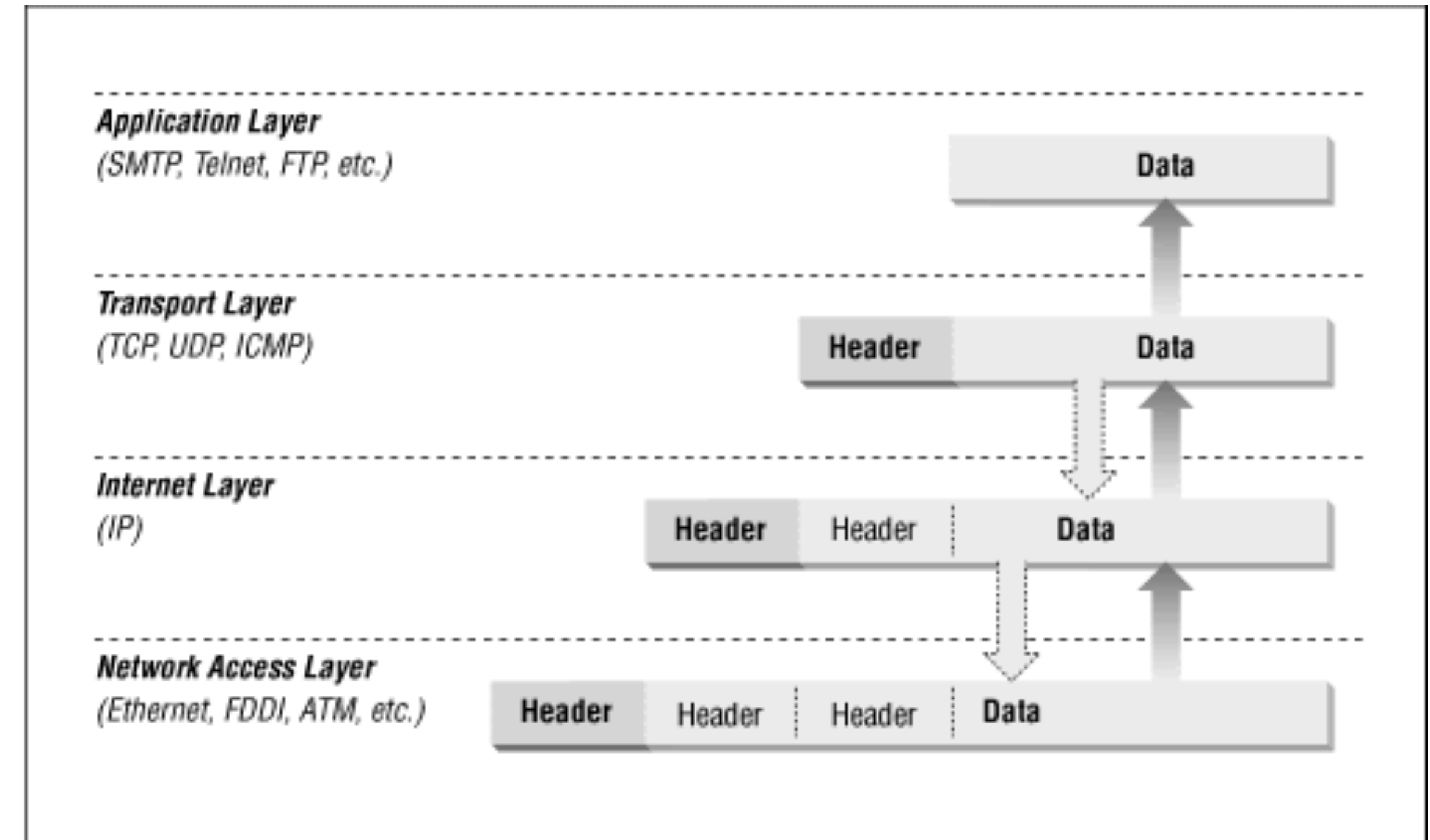
# NETWORKING – TCP/IP

- ▶ Paketen skickas till en process
  - ▶ Process tolkar data och betydelse
- ▶ IP - hanterar transport mellan routrar
- ▶ TCP - hanterar transport till process



# NETWORKING – TCP/IP

- ▶ TCP (Transmission Control Protocol)
  - ▶ Inter-process kommunikation
  - ▶ Hanterar att paketen kommer
    - ▶ I rätt ordning
    - ▶ Förlorade paket återfrågas



---

## NETWORKING – TCP/IP

- ▶ Så i exemplen när vi skickar data måste vi tala om till vilken process...
  - ▶ En fil (**ftp 21**)
  - ▶ En hel webbsida "<html>....</html>" (**http 80**)
  - ▶ Ett email "From: 'Bob Example' <bob@example.org>... Hej Alice" (**smtp 25**)

---

# NETWORKING – TCP/IP

## ► Frågor



---

# WEBB – HTTP

- ▶ HTTP protokoll (webbläsare <-> servrar)
  - ▶ Använder port 80
- ▶ En *klient* öppnar en connection till server, gör **request**, väntar...
- ▶ En *server* svarar med en **response**



---

# WEBB – HTTP

## ▶ Hur ser processen ut?

- ▶ 1. Klient öppnar TCP connection för att skicka paketen
- ▶ 2. Klienten skickar ett HTTP meddelande (request) som delas upp i paket

```
GET / HTTP/1.1  
Host: developer.mozilla.org  
Accept-Language: fr
```

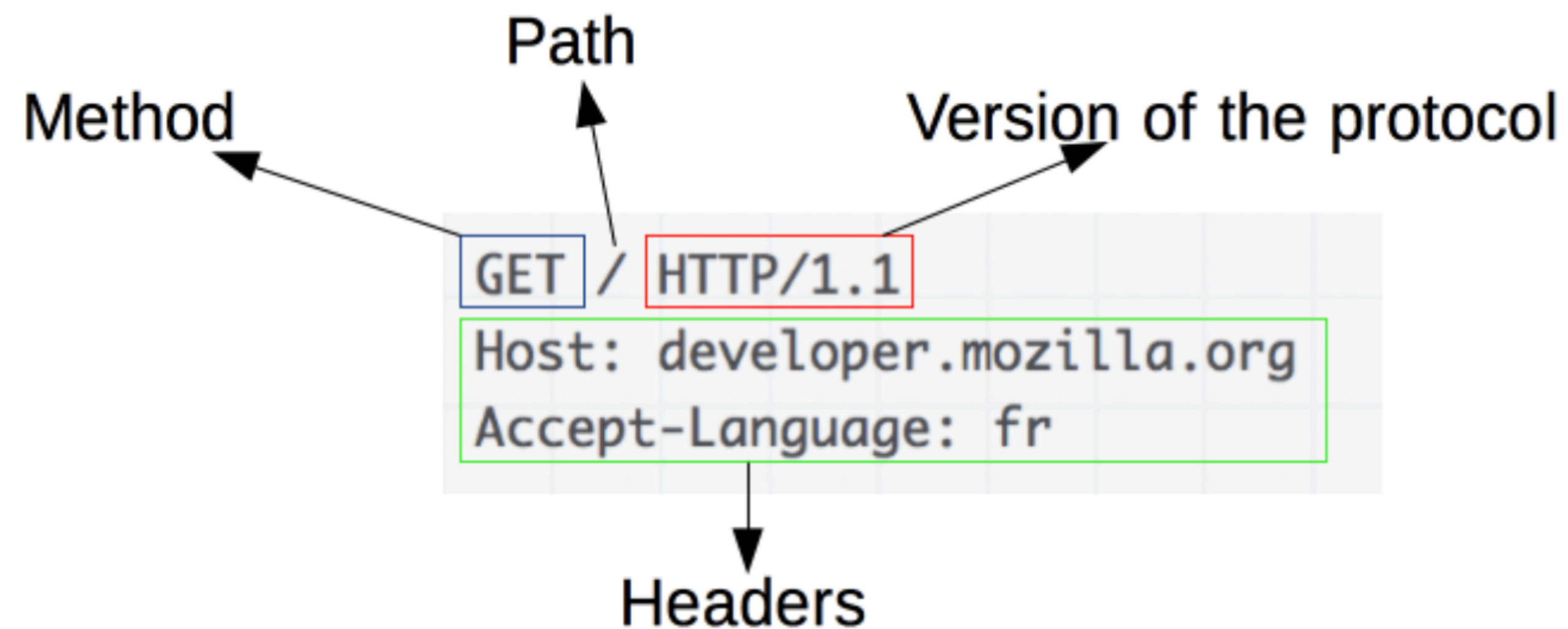
- ▶ 3. Server svarar med HTTP meddelande (response) som delas upp i paket

```
HTTP/1.1 200 OK  
Date: Sat, 09 Oct 2010 14:28:02 GMT  
Server: Apache  
Last-Modified: Tue, 01 Dec 2009 20:18:22 GMT  
ETag: "51142bc1-7449-479b075b2891b"  
Accept-Ranges: bytes  
Content-Length: 29769  
Content-Type: text/html  
  
<!DOCTYPE html>... (here come the 29769 bytes of the requested web page)
```

# WEBB - HTTP

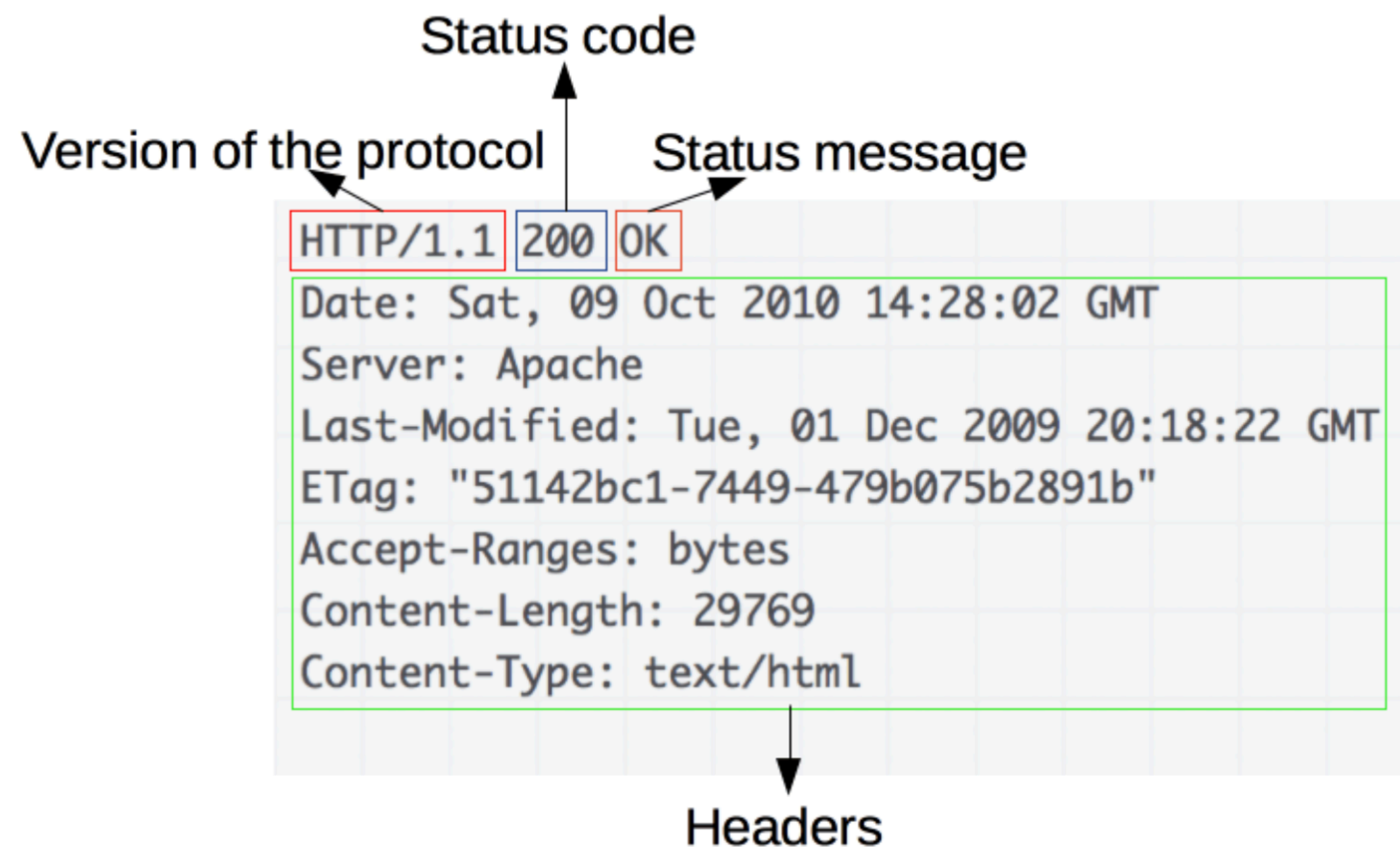
## ► Request Headers

### ► Methods: **GET, POST, PUT, DELETE**



# WEBB – HTTP

- ▶ Response Headers
  - ▶ Status codes: **200, 301, 400, 404, 500**



---

# WEBB – HTTP

## ▶ curl

▶ <https://reqbin.com/req/c-1n4ljxb9/curl-get-request-example>

▶ GET, POST, PUT, DELETE (exempel)

▶ curl -I <https://kvibergsmarknad.se/index.html>

▶ curl -v <https://kvibergsmarknad.se/index.html>

▶ curl -o index.html <https://kvibergsmarknad.se/index.html>

---

# LOCALHOST – HTTP

## ▶ Localhost

- ▶ Ibland vill vi använda vår dator som klient och server för att testa en applikation
- ▶ **localhost** är en dedikerat hostname -> adress **127.0.0.1**
- ▶ Denna adress är loopback adress
  - ▶ Vi kan ha en mailserver på porten **smtp://localhost:25**
  - ▶ Vi kan ha en filserver på porten **ftp://localhost:21**
  - ▶ Vi kan ha en webserver på porten **http://localhost:80**

---

# WEBB – HTTP

## ► Frågor



---

# KLIENT I JS

- ▶ <https://www.digitalocean.com/community/tutorials/how-to-create-an-http-client-with-core-http-in-node-js>
- ▶ GET, POST, PUT, DELETE

---

# SERVER I JS

- ▶ <https://www.digitalocean.com/community/tutorials/how-to-create-a-web-server-in-node-js-with-the-http-module>
- ▶ GET, POST, PUT, DELETE



---

# LÄXA

- ▶ Prova att ändra klienten till att använda tredjeparts paket istället för https
  - ▶ <https://blog.logrocket.com/5-ways-to-make-http-requests-in-node-js/>
    - ▶ Prova antingen: axios, node-fetch, got eller superagent
- ▶ Prova att ändra server för att kunna ta emot POST, PUT och DELETE requests också