

Internetprogrammering

DD1386

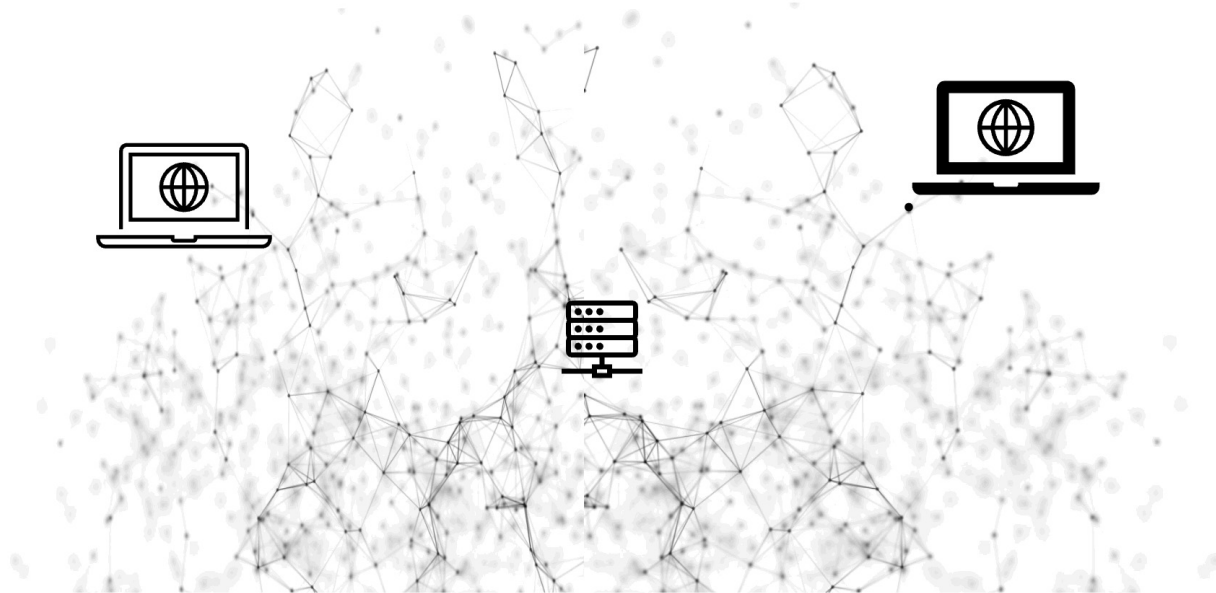
Föreläsning 1

- Kursinformation
- På klientsidan
- På serversidan
- Olika nätverkstyper
- Nätverksprotokoll
- HTTP- protokollet
- OSI-modellen
- Socket

Kursinformation

- Kursledare och examinator: Vahid Mosavat, vahid@kth.se
- Kursaktiviteter: föreläsningar, räkne-stuga, laborationer
- Kursmaterial: labbinstruktioner, föreläsningsanteckningar, länkar till externa webbsidor
- Examination:
 - LAB1 (3.0 hp): 5 obligatoriska labbar labb 1-5 (grupp av 2)
 - PRO1(3.0hp): Obligatoriskt projekt (grupp av 2) (Betyg E-A)
 - ÖVN1 (1.5 hp): övningsuppgifter i form av Quiz.
 - 3 bonusgivande labbuppgifter (frivilliga men påverkar betyget)
- Kurshemsidan: <https://canvas.kth.se/courses/36975>

Internet

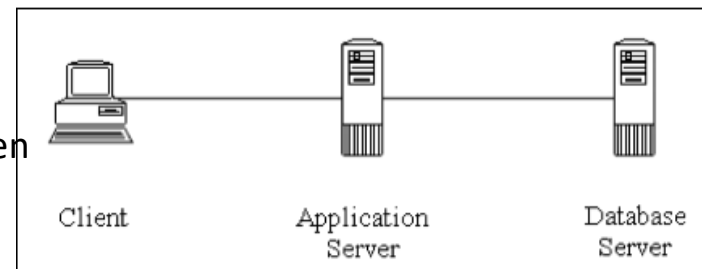


Allmänt

- Applikationer baserad på Client/Server modellen där klienten (vanligtvis) utgörs av en webbläsare och servern av en webbapplikation

- Treskiktslösningar

- Klienten utgör presentations skiktet av tillämpningen
- Servern hanterar den bakomliggande logiken
- Databassystem utgör också ett eget skikt men behandlas endast i ringa omfattning i denna kurs



- Programspråket Java (Socket) och JavaScript (Node.js och andra ramverk så som Express, Vue, Vite) användas i labbarna och projektet.

Klientsidan

- Presentations skiktet behandlar främst layout men även enklare logiska funktioner som servern inte behöver blandas in t.ex. validering av inmatning. Vi kommer främst använda HTML, CSS och JavaScript.
- Modern webb använder sig av Ajax-teknik och liknande för att gå runt "problemet" med tillståndslöshet (stateless) som är grund till HTTP.

Serversidan

- Webbserver
 - HTTP, HTTPS (Hypertext Transfer Protocol)
 - Port 80
 - Tillståndslös
- Server-side scripting (kräver webbserver)
 - Genererar dynamisk HTML-sida
 - *PHP (5.3 och äldre versioner),*
 - *CGI- skript: c, perl, osv (egentligen nästan alla programspråk)*
- Webbapplikation
 - Genererar dynamisk HTML-sida med en utformning skräddarsydd för ett visst ändamål, t.ex webbmail, instant messaging, osv...
 - Node.js

... och dessutom

- Grundläggande nätverksterminologi
- Socket
- Kryptering (TSL/SSL)

Kursfordringar

Obligatoriska examinationer	Frivilliga uppgifter
<ul style="list-style-type: none">• Ladok-moment LAB2 (3.0 hp) (P/F)<ul style="list-style-type: none">• Labb 1: Gissa tal (HTTP-protokollet, Java, Socket)• Labb 2: Webbaserat Fyra-i-rad (DHTML, JavaScript)• Labb 3: Webbaserat inloggnings sekvens (NodeJs)• Labb 4 (2 delar): Bokningssystem (NodeJs, Vue, Express)• Labb 5: Krypterad anslutning (TSL/SSL)• Ladok-moment PRO2 (3.0 hp)<ul style="list-style-type: none">• Projekt (F-A)• Ladok-moment ÖVN1: Quiz (1.5hp) (P/F)	<ul style="list-style-type: none">• Bonusgivande uppgifter: Laborationerna 1,3 och 4 har var sin bonusgivande labbuppgift, X1, X3 och X4.• Bonus ges endast om de redovisas innan deadline.

Deadlines

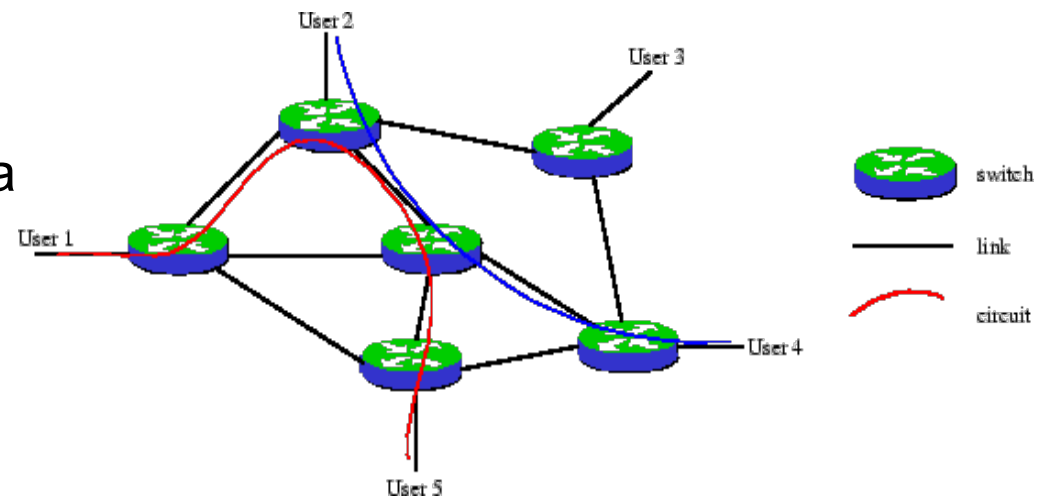
Uppgift	Deadline
Labb1X	2023-02-03
Labb3X	2023-02-24
Labb4X	2023-04-06
Projekt	2023-05-11

Grundläggande begrepp

- Nätverkstyper
- Protokoll
- OSI-Lager
- Socket, enkel server-klient applikation

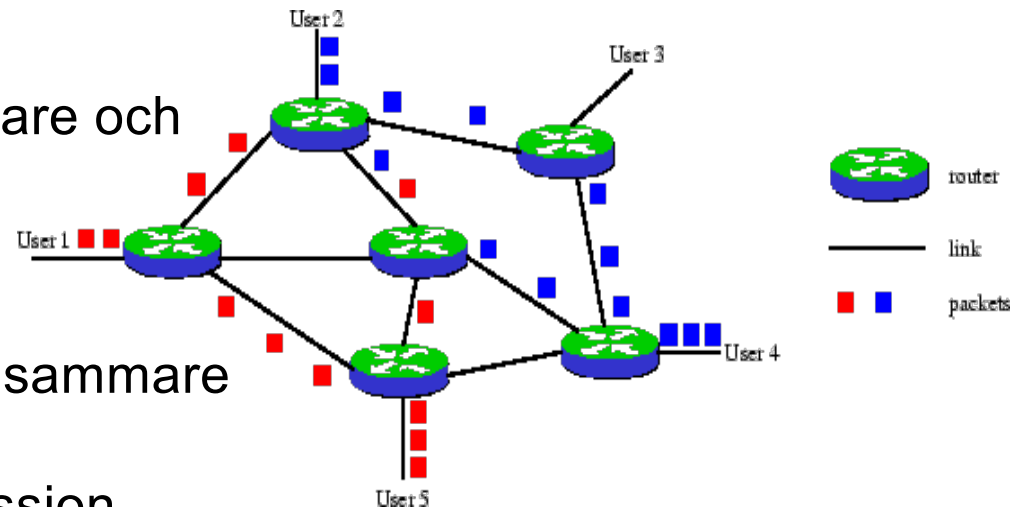
Kretskopplade nätverk

- En förbindelse mellan två enheter upprätts med hjälp av växlar innan överföringen kan starta.
- Alla inblandade resurser låses av denna uppkoppling
- Det kan ta lång tid att upprätthålla en anslutning
- Ömtålig, t.ex om en resurs går sönder under överföringen
- Begränsad kapacitet
- Analog med traditionellt telefon-samtal



Paketväxlande nätverk

- Förbindelse-lös kommunikation
- Paketen överförs oberoende av varandra
- Fullständig adressinformation om avsändare och mottagare
- Paketen kan ta olika vägar
- Paketen kan tappas bort eller dubblas
- Nätet kan inte bli upptaget utan bara långsammare
- Hållbar överföring
- T.ex: IP(Internet Protocol), TCP(Transmission Control Protocol), UDP (User Datagram Protocol)



Protokoll

- Ett protokoll är en samling av regler och överenskommelser kring kommunikation mellan två enheter som kommunicerar med varandra
- Innehåller bl.a. uppgifter om:
 - Förbindelser
 - Vägval
 - Sönderdelning av data
 - Sammansättning av data
 - Upprättande av ordningsföljd vid överföring
 - Felkorrigering

Exempel på HTTP protokoll

~> **telnet www.nada.kth.se 80**

Trying 130.237.227.116...

Connected to sippans.csc.kth.se. Escape
character is '^['.

```
GET /~vahid/intnet18/f1.html HTTP/1.0↵
```

```
HTTP/1.1 200 OK↵
```

```
Date: Mon, 15 Jan 2020 22:14:15 GMT↵
```

```
Server: Apache↵
```

```
Last-Modified: Tue, 05 Sep 2017 12:58:38 GMT↵
```

```
ETag: "4eea4522-27-c9285b80"↵
```

```
Accept-Ranges: bytes↵
```

```
Content-Length: 39↵
```

```
Connection: close↵
```

```
Content-Type: text/html↵
```

```
↵
```

```
<b>Internetprogrammering är roligt!</b>
```

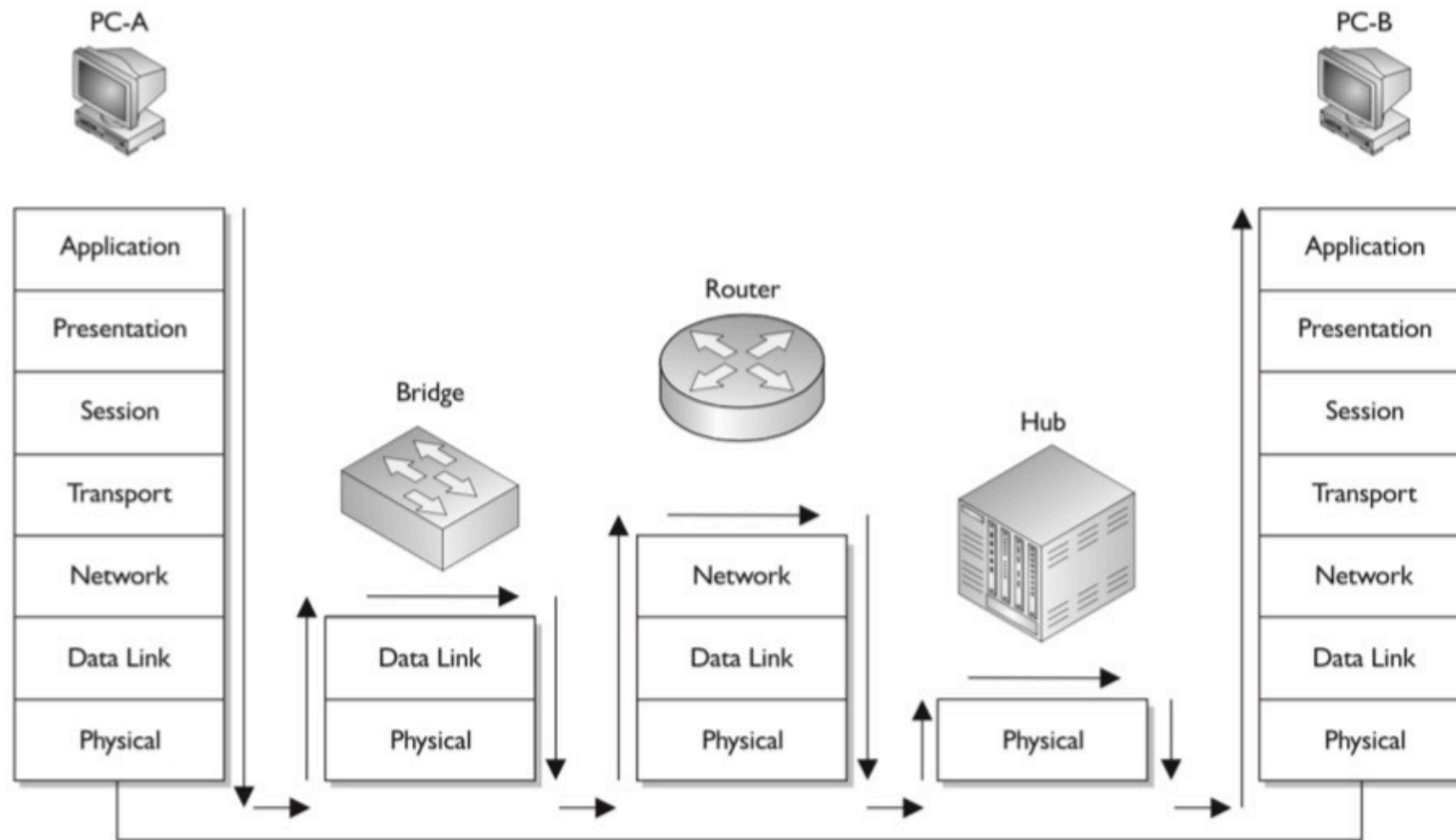
Connection closed by foreign host.

~>

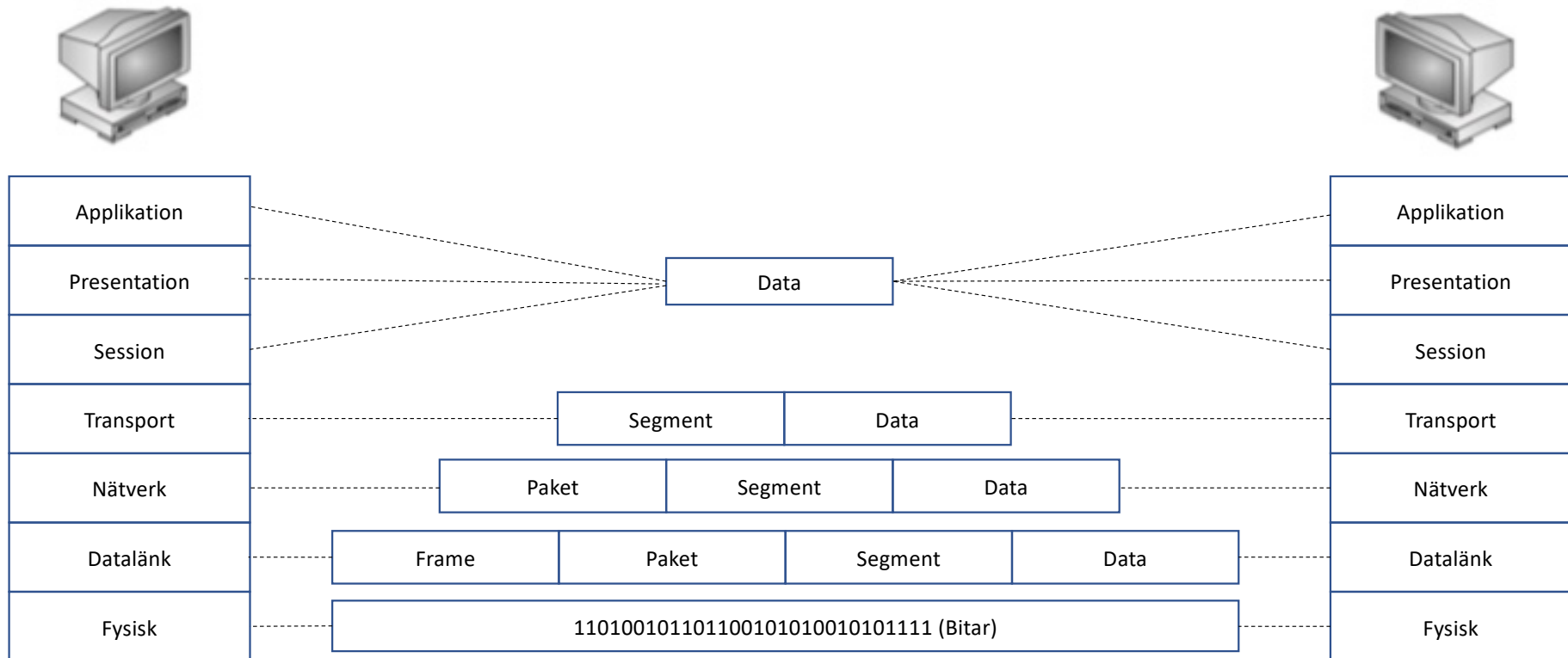
Förfråga

Respon
s

Protokoll stacken



Protokollstacken



Datagram

- Innehåller följande information

- Header

- Avsändaradress
 - Destinationsadress
 - Typ av innehåll (t.ex TCP, UDP etc)
 - Längd av data
 - Felkontroll

- Data

- Själva användarinformationen (payload)

Nätverkslagret (IP)

- Ett protokoll för att överföra data mellan olika nätverk (därför ordet internet). Är oberoende av det underliggande nätets implementation (t.ex Ethernet, ATM)
- Då den verkliga kommunikationen endast sker på länklagret finns en teknik för att översätta mellan IP-adress och fysisk (länk) adress, ARP (address resolution protocol).
- IP är ett tillståndslöst protokoll
- Paketet kan fördubblas eller försvinna
- En IP-adress (IPv4) består av 4 bytes och har formatet 130.237.225.94
- TTL (Time To Live) : kontrollfält
- Dagens IPv4 håller på att ersättas med IPv6 som har en adressrymd på 128 bitar.

Transportlager: TCP, UDP och ICMP

- TCP (Transmission Control Protocol)
 - Förbindelseorienterad (logiskt)
 - Säker transport (blanda inte med sekretess)
 - Data förloras inte (förlorad data skickas igen)
 - Data fördubblas inte (fördubblad data slängs)
 - Data kommer fram i rätt ordning
- UDP (User Datagram Protocol)
 - Förbindelselöst
 - Data kan förloras (kommer inte heller skickas igen)
 - Data kan fördubblas (fördubblad data slängs inte)
 - Data kan komma fram i fel ordning
- ICMP (Internet Control Message Protocol)
 - Kopplat till TCP
 - Skickas av mottagare som väntat på ett paket som inte kommit fram
 - `traceroute`

Applikationslager

– HTTP

- Surfning

– HTTPS

- Krypterad surfning

– FTP

- Filöverföring

– POP3

- Epost-mottagande

– SMTP

- Epost-skickande

– DNS

- IP adress <--> DNS-namn översättning

– Telnet

- Okrypterad fjärrinloggning

– SSH

- krypterad överföring, ofta för fjärrinloggning mot ett unix-skal

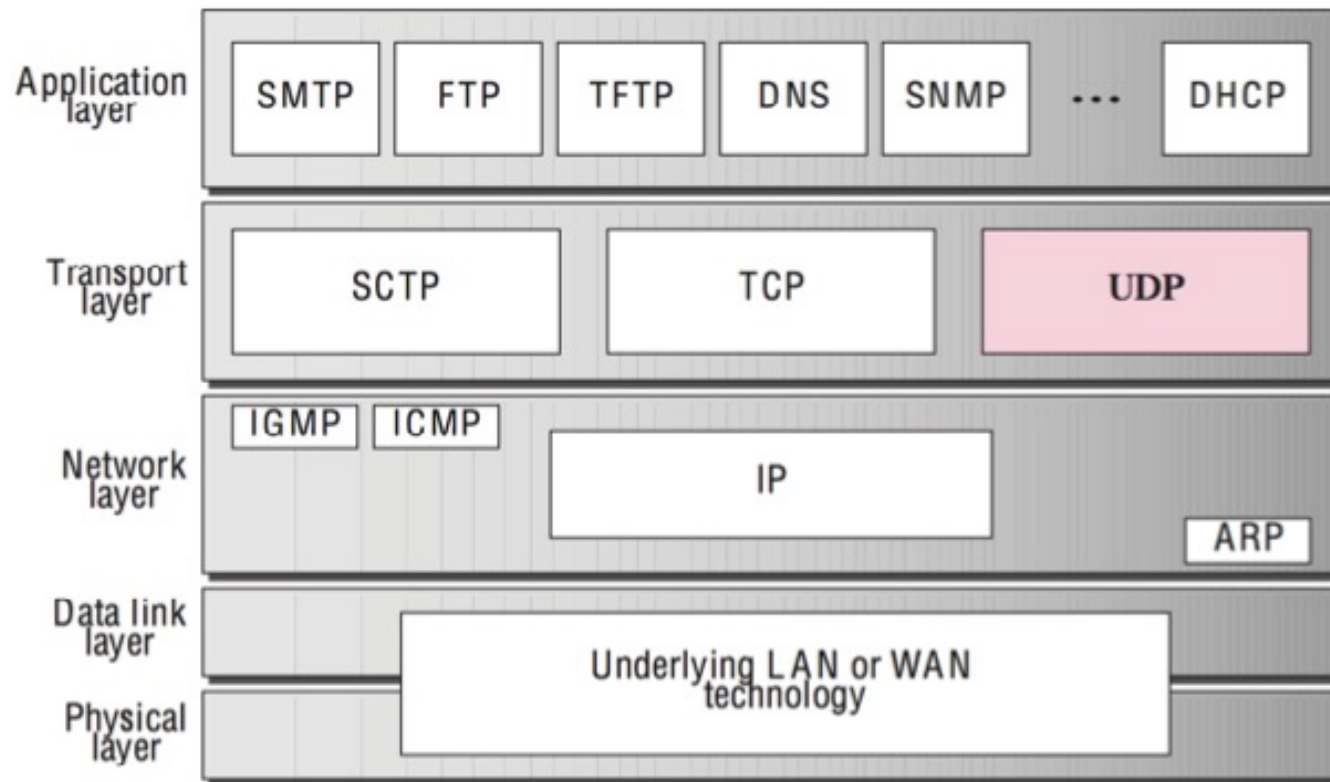
– DHCP

- För att tilldela klienter ett dynamisk ip-adress i ett nätverk.

– Ping

- För att kolla om en värd är "uppe"

Förhållanden mellan protokollen



Socket

- Socket är gränssnitt mot TCP/IP och UDP/IP och används alltså för att upprätta en anslutning baserat på IP mellan två maskiner
- En socketanslutning som exempelvis använder port 22 för att initieras men den fortsatta kommunikationen kan ske över valfri port på servern
- I java finns följande klasser implementerade
 - `java.net.Socket`
 - `java.net.ServerSocket`
- Klientens port slumpas

Server.java

```
import java.net.*;
import java.io.*;

public class Server{
    public static void main(String[] args) throws Exception{
        ServerSocket serverSckt = new ServerSocket(1234);

        Socket sckt = serverSckt.accept();
        while( (sckt != null){

            BufferedReader indata = new BufferedReader(
                new InputStreamReader(sckt.getInputStream()));

            String text = null;
            while( (text = indata.readLine()) != null){

                System.out.println(text);

            }

            sckt.shutdownInput();
            sckt = serverSckt.accept();

        }

    }
}
```

Client.java

```
import java.net.*;
import java.io.*;
public class Client{
    public static void main(String[] args){
        try{
            Socket sckt = new Socket("share-02.csc.kth.se",1234);
            PrintStream out = new PrintStream(sckt.getOutputStream());
            BufferedReader indata = new BufferedReader(
                new InputStreamReader(System.in));

            String text;
            while((text = indata.readLine()) != null)
                out.println(text);
            sckt.shutdownOutput();
        }catch (Exception e){
            System.err.println("Ett fel intraffade!");
        }
    }
}
```