Föreläsning 1

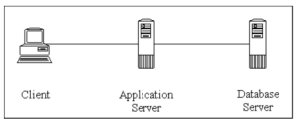
DD2390
Internetprogrammering
6 hp

Kursintroduktion

- Presentation
  - Kursledare: Sten Andersson, stene@kth.se
- Anmälningslista
  - Besök "Mina sidor" för att bekräfta att kursen är registrerad
  - https://rapp.csc.kth.se/rapp/
- Kursmaterial
  - Skrivs ut från webben
  - Ingen obligatorisk kurslitteratur
- Kurshemsida
  - https://www.kth.se/social/course/DD2390/
  - Nyhetsbrev
- Schema
- Föreläsningar
- Laborationer (datorsalar, konton, handledare)
  - OBS!!! Kontrollera att ni har fungerande UNIX-konton **innan** första laborationstillfället.
- Betyg

Allmänt

- Kursens mål: Att få kännedom om och kunna utveckla klient och serverbaserade program där klienten (vanligtvis) utgörs av en webläsare och servern av en webserver.
- Man brukar ofta tala om s k treskiktslösningar där klienten utgör presentationsskiktet av tillämpningen medans den bakomliggande logiken hanteras av servern. Databasskiktet utgör också ett eget skikt men behandlas endast i ringa omfattning i denna kurs (förkunskap).
- Främst java kommer att användas som språk och mycket av kursen rör java.net paketet.



På klientsidan...

- I presentationsskiktet behandlas främst layout men även enklare logiska funktioner t e x validering av inmatning. Vi kommer främst att använda HTML, CSS och javascript (=DHTML).

På serversidan...

- I denna del av kursen behandlas tillämpningar som tillhandahåller klienter (= webbläsare) **dynamiska** HTML-sidor. Dessa tillämpningar kräver tillgång till (körs under) en webserver.
- En dynamisk HTML-sida genereras med en utformning skräddarsydd för varje klient som efterfrågar den. PHP/Java EE används som språk här.
- En webserver är ett program som kommunicerar med http (https) och vanligtvis använder port 80 för kommunikation. En webbserver är även passiv, d v s den svarar endast på förfrågningar

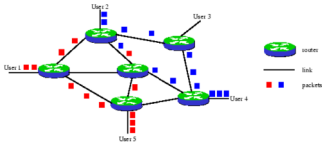
Och dessutom...

- Grundläggande nätverksterminologi
- Socket:ar
- Kryptering (JSSE)

Laborationer

- Laboration 1: Socketar & Trådar (Chat)
- Laboration 2: HTTP-protokollet (NumberGuess)
- Laboration 3: DHTML (Memoryspel)
- Laboration 4: PHP (Bovision)
- Laboration 5: Java EE (ny)
- Laboration 6: JSSE (Kryptering)
  - Dessa sex labbar utgör momentet LAB1 (4.5 hp)
- Projekt
  - Utgör momentet PRO1 (1.5 hp)

Paketväxlande nätverk



Informationen styckas upp i delar (paket) och skickas ut till nästa s k router.
Vilken som är nästa nod i kommunikationen bestäms av en routingtabell unik för varje router.
Egenskaper:
•Ingen uppkoppling krävs
•Nätet kan inte bli upptaget, bara långsammare.
•Paketen kan komma fram i fel ordning
•Om en länk går sönder beräknas routingtabellen om för närliggande noder och trafiken tar en annan väg

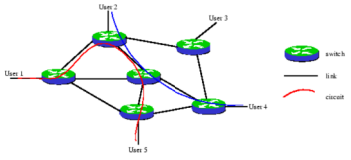
Grundläggande begrepp

- Nätverkstyper
- Protokoll
- Lager
- Sockets

Protokoll

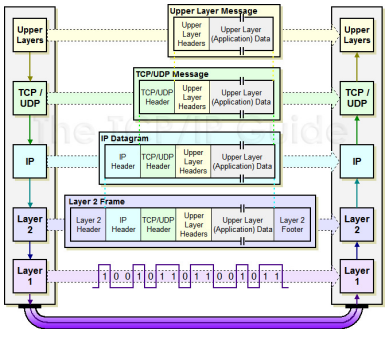
- Med protokoll avses de kommunikationsregler som måste följas inom trafiken mellan program på två enheter. Då det finns många olika tjänster (t e x http eller ftp) som ska stödjas, finns flera olika protokoll med olika uppgifter. Dessa är bl a
  - Skapa förbindelser
  - Vägval
  - Styckning och ihopkoppling av data
  - Upprättande av ordningsföljd vid överföring
  - Felkorrigering
- På lokala nätverk och även internet är dessa organiserade i fyra (fem) nivåer:
  - (Fysiskt lager t e x fiberoptik, seriell kabel etc)
  - Länklagret, skapar t e x förbindelser mellan enheter
  - Nätverkslagret, hanterar t e x vägval och flödesreglering
  - Transportlager, tar t e x hand om bl a ordningsföljd, felkorrigering
  - Tillämpningslager, högsta nivån, här finns tillämpningsprotokollet självt, t e x http eller ftp
- Datamängden som ska sändas delas upp i paket (datagram). Paketen från högre nivå förpackas i paket på lägre nivå. Denna struktur brukar benämnas protokollstacken.

Kretskopplade nätverk



En fysik förbindelse upprätthålls mellan två enheter med hjälp av växlar.
Egenskaper:
•Tar resurser i bruk under under hela uppkopplingstiden även om data inte sänds
•Det tar lång tid att upprätthålla en anslutning då hela kommunikationsvägen måste bestämmas i förväg
•Känsligt för trasiga länkar
•Om inga länkar finns kvar blir nätet upptaget

Protokollstacken



Datagram

- Innehåller följande information
  - Header
    - Avsändaradress
    - Destinationsadress
    - Typ av innehåll (t e x TCP)
    - Längd av data
    - Felkontroll
  - Data
    - Informationen självt, kan vara ett annat datagram
  - Footer
    - Endast för vissa protokoll
    - Felkontroll

IP-routing

- Medans en värd endast bryr sig om paket som är adresserade till sig själv så behandlar en router alla inkommande paket.
- Routern har en s k routingtabell med adresser på datorer den känner till och först görs en matchning mot dessa adresser. Om den inte hittar en exakt matchning görs en matchning mot netword-id och paketet skickas vidare till denna enhet (vanligtvis också en router). Om fortfarande ingen matchning finns skickas paketet till en standardadress (unik för routern). Om denna adress saknas skickar routern ett s k ICMP ”host unreachable” eller ”network” unreachable”

Länklagret (Ethernet)

- Länklagret representerar den fysiska nivån på vilken all verklig kommunikation sker.
- På LAN utgörs länklagret oftast av Ethernet. Varje nätverkskort har här ett unikt 6 byte hexadecimalt identifikationsnummer, s k MAC-adress (Media Access Control).
- På WAN utgörs länklagret av t ex ATM.

Transportlager: TCP, UDP, ICMP

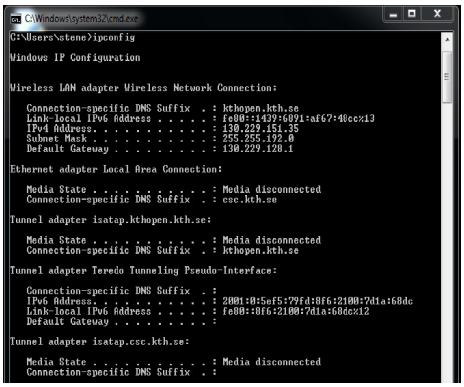
- Protokoll på transportlagret har hand om att se till att datan kommer fram säkert (om man vill) och sända om information om det är önskvärt, Om ett paket försvinner på väg till en dator så skickas det igen och TCP ser till att de kommer i rätt ordning. Paketen verifieras via en enkel kontrollsumma så de inte är korrupta. Skapar uppkoppling (session).
- UDP (User Datagram protocol) är bättre att använda än TCP om man vill streama något, dvs. att det viktigare att data alltid överförs än att det är in på byten korrekt. Till sådant här streamade radio- och tv-sändningar, IP-telefoni. Kommer datan bort och måste skickas om (som med TCP) så är den ändå värdelös när den kommer fram försenat. Adresseringen sker via portnummer. TCP:s kontroller slöar ner det jämfört med UDP. Kräver ingen uppkoppling.
- ICMP är egentligen ett protokoll på nätverkslagret men det är kopplat till TCP genom felkontrollen, har ett paket inte kommit fram eller har fel kontrollsumma skickas ett ICMP-meddelande till den dator som skickade paketet. Detta utnyttjas av programmet traceroute för att spåra en nätverkspath.

Nätverkslagret (IP)

- Ett protokoll för att överföra data mellan olika nätverk (därav ordet inter-net). Är oberoende av det underliggande nätets implementation (t e x Ethernet, ATM).
- Då den verkliga kommunikationen endast sker på länklagret finns en teknik för att översätta mellan IP-adress och fysisk (länk) adress, ARP (adress resolution protocol).
- IP är ett tillståndslöst protokoll, varje paket som skickas är ett individuellt jobb i sig och ingen felhantering görs. Ej heller bryr sig IP om i vilken ordning paketen kommer fram.
- En ip adress består av 4 bytes och har formatet 130.237.225.94.
- I datagrammet finns ett kontrollfält som benämnds TTL (Time To Live) : det är ett tal som initieras till 64 och som vid varje router minskas med 1. När talet blir 0 kastas paketet. Detta för att inte paket ska cirkulera på internet i all evighet.
- Dagens IPv4 håller på att ersättas med IPv6 som har en adressrymd på 128 bitar.

Applikationslager

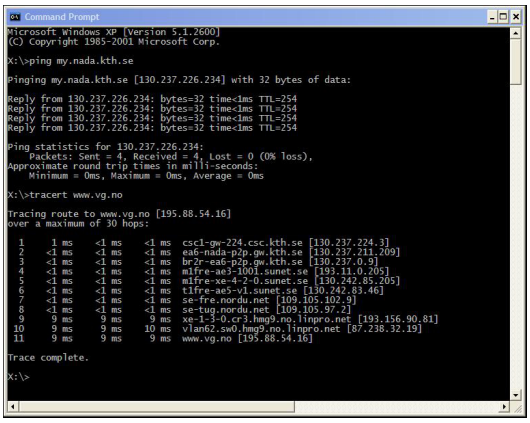
- I applikationslagret finns en mängd olika protokoll beroende på vilken tjänst som används:
  - HTTP
    - surfning
  - HTTPS
    - krypterad HTTP
  - FTP
    - filöverföring
  - POP3
    - epostmottangande
  - SMTP
    - epostskickande
  - DNS
    - IP adress <--> DNS-namn översättning
  - Telnet
    - 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”
  - Socket
    - En generell anslutning på detta lager



Server.java

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

public class Server{
   public static void main(String[] args) throws Exception{
      ServerSocket ss = new ServerSocket(1234);
      Socket s = null;
      String text = "";
      while( (s = ss.accept()) != null){
         BufferedReader indata =
        new BufferedReader(new InputStreamReader(s.getInputStream()));
          while( (text = indata.readLine()) != null){
        System.out.println(text);
         }
         s.shutdownInput();
      }
   }
}
```



Client.java

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

public class Client{
   public static void main(String[] args) throws Exception{
Socket s = new Socket(args[0],1234);
PrintStream out = new PrintStream(s.getOutputStream());
BufferedReader indata =
   new BufferedReader(new InputStreamReader(System.in));
      String text = "";
      while( (text = indata.readLine()) != null){
   out.println(text);
}
s.shutdownOutput();
   }
}
```

Socket

- Om man vill skriva till / läsa från en tcp/ip (udp/ip) anslutning på ett sätt liknande vanlig filhantering kan man använda sockets. En socketanslutning 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