Datorkommunikation RV3 HT 2012

Johan Arvelius

8 december 2013

Det här dokumentet i senaste version:

PDF http://kyla.kiruna.se/~arvelius/dk.pdf

Läsanvisningar

Vi har ingen lärobok utan det blir mest hänvisningar till materiel på nätet och en del egen informationssökning på nätet.

Det här dokumentet är det tänkt skall växa fram under kursens gång så att det innan varje kurstillfälle ska finnas detaljerade anvisningar för det tillfället. Läsanvisningar utan särskild markering, övningar med nummer som Ö1 etc. och frågor med nummer som 1.1 etc som inte har någon annan särksild markering är instuderingsmaterial som det är tänkt att alla ska göra.

Moment markerade med • ska redovisas enligt instruktion. De moment som benäms **laboration** skall genomföras och redovisas genom inlämning av en laborationsrapport. De moment som benämns **praktisk övning** ska genomföras och resultatet av övningen är i sig redovisning för den och ingen vidare redovisning för utförandet behövs.

Moment markerade med * är extrauppgifter för de som hinner med, fler stjärnor är högre grad av överkurskaraktär.

Betygsunderlaget för kursen är laborationer inklusive rapporter, resultat av praktiska övningar, prov samt aktivt deltagande i diskussionsövningar.

Innehåll

Kursplanering

Vecka 35 Datorkommunikationens historia.

Vecka 36 Lagstiftning. Vi pratar om upphovsrätt, licenser, piratkopiering och personuppgiftslagen.

Vecka 37 Genomgång av TCP/IP stacken. Grundläggande struktur och exempel på vilka protokoll som finns i olika lager av stacken. Vi tar även upp portar och demoner. **Laboration** där vi tittar på hur data ser ut i praktiken för del övre lagren i stacken vid en enkel kommunikationssituation (HTML, HTTP, TCP).

Vecka 38 Kommunikationsmodeller. Vi kommer att ta upp de olika modellerna för att få två parter att byt information, peer-to-peer, torrents och server-klientmodellen. **Laboration** Vi sätter upp en apache webbserver.

Vecka 39 Webben. Vi tittar på webben a'la 90-tal med statiska hemsidor. **Praktisk övning** Vi skriver hemsidor i HTML.

Vecka 40 Stil på webben. Vi lär oss att skilja på innehåll och utformning på webben. **Praktisk övning** Vi sätter stilmallar till våra hemsidor i CSS.

Vecka 41-42 Web 2.0, den interaktiva webben vi talar om interaktiva webbtjänster, webbaserade projekt och sociala medier. **Praktisk övning** CGI-programmering.

Vecka 43 Distansvecka

Vecka 45 Kabelbundna anslutningar. Vi går igenom olika koncept som parallella och seriella kablar, koaxialkablar, skärmning, partvinning och fiberoptik. Vi tittar på vanliga standarder och jämför deras prestanda. **Praktisk övning** kontaktering av TP-kabel.

Vecka 46 Lokala nätverk och routing. Vi tittar på olika nätverkstopologier, adressrymder på kopplade nätverk, tar upp nätmasker och routing.

Vecka 47 Kryptering. Teori kring kryptering, symmetrisk och assymetrisk kryptering samt utbyte av symmetriska nycklar genom assymetrisk kryptering. Vi tittar på olika tillämpningar för signering och kryptering av epost, kryptering på webben, krypterad inloggning och tunnlade anslutningar samt DRM.

Vecka 48 Trådlösa nätverk. Vi tittar på hur trådlösa nätverk med WiFi går till och framför allt kryptering av nätverkstrafiken.

Vecka 49 Signalteori, vi tittar på hur signalerna är utformade i trådlösa och trådbundna system. Amplitudmodulering, frekvensmodulering, bärvågor diskuteras samt beroendet mellan frekvens, bandbredd och överföringskapacitet.

Vecka 50 Prov på Trådbundna och trådlösa överföringar, nätverk, kryptering och signalteori.

1 Historia

Läs

- Nedanstående,
- Christer Bergström, *Svensk knäckte nazisternas hemliga koder* http://www.svd.se/kultur/svensk-knackte-nazisternas-hemliga-koder_5549643.svd#enigma-blottade-br-hitlers-planer

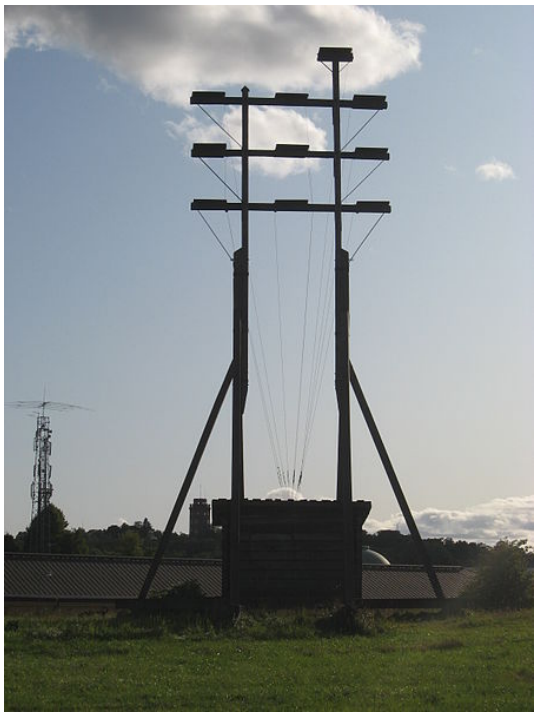
Se

- The Enigma Secret http://www.youtube.com/watch?v=IJToxIZMbZQ
- G som i hemlig - om svensken som knäckte tyskarnas kod http://www.youtube.com/watch?v=aXykQsvjLqs

Besvara Instuderingsfrågorna 1.12.

1.1 Telegrafer

För att kommunicera över större avstånd har människan använt telegrafer som kom i praktiskt bruk under 1700-talet. De tidigare var optiska telegrafer som opererades av en telegrafist som ändrade utseendet på telegrafen genom att fälla luckor som bildade mönster eller genom att ändra läget på telegrafens armar. De kunde användas när det var fri sikt mellan telegraferna. Så småningom kom den elektriska telegrafen som användes med elektriska impulser över ledningar mellan telegrafstationerna. Så småningom kom även telefonen för att överföra ljud direkt över samma ledningar. Telegrafen utvecklades parallellt med telefontrafiken och gjordes mer tekniska och oberoende av telegrafister. Omkodningen från skriven text till elektriska signaler som sändes över telenätet och tolkningen av sådana signaler till skriven text med så kallade fjärrskrivare eller teleprinter utvecklades under första delen på 1900-talet. Till dessa teleprintrar utvecklades 7 bitars ASCII-systemet för att representera skrift som sju bitars (ettor och nollor) för 2^7 tecken.



Figur 1: Replika av en optisk telegraf Foto: Wikimedia commons användare Liftarn Från Wikimedia Commons.

1.2 Kryptering

Ända sedan Caesars tid har man använt krypton för att hemlighålla information. Det har historiskt främst varit militären som varit intresserad av kryptografi, en idag är den största utbredningen på internet med banker och internethandel.

Under Andra världskriget spelade kryptering en avgörande betydelse. Båda sidor använde kryptering för både trådbunden och radioburen telegrafi. Avgörande för utvecklingen i alla fall i krigets slutskede var att de allierade lyckades dechiffreara tyska meddelanden.

Tyskarna använde framför allt krypteringsmaskinen Enigma i många olika versioner. Tre polska matematiker, Marian Rejewski, Jerzy Różycki och Henryk Zygalski knäckte i princip hur maskinen fungerade redan före kriget och överlämnade sitt arbete till brittisk underrättelsetjänst. För att knäcka de nya kodnycklar som användes hela tiden utvecklade britterna speciella maskiner “Bombe”. De var stora replikor av Enigma som provade flera

olika krypteringsnycklar i taget och körde meddelanden gång på gång tills man fick fram klartext. De byggdes i flera varianter både i Storbrittannien och USA.

Ett av de första avgörande användandet av informationen från den var vid slaget om Kreta då Luftwaffes enigmatrafik övervakats ett långt tag och de allierade väl kände tyskaras planer. Det var Nazisternas första stora luftlandsättningsmanöver och de tyska fallskärmsjägarna blev överrumplade av motståndet som både berodde på att de alierade hunnit landsätta mycket trupper i hemlighet för tyskarna och till en del kom från den Kretesiska civilbefolkningen som låg förberedda i bakhåll och stred med alla tillgängliga tillhyggen. Tyskarna insåg inte till en början att de blivit avlyssnade och trodde det var taktiken med luftlandsättning som misslyckats. Tyskarna vars flygvapen var överlägset intog till slut Kreta trots allt men till ett mycket högre pris än de räknat med. De allierade var imponerade över manövern som lyckades trots avlyssningen och satsade sedan på egna fallskärmstrupper. (*Prestidge-King*, 2007)

Svenska matematikprofessorn Arne Beurling knäckte tyskarnas G-skrivare (en annan krypteringsmaskin) och Sverige hade tillgång till den tyska kommunikationen genom Sverige till Norge efter Tyskarnas ockupation av Norge. Denna information delgavs inte de allierade i någon större omfattning. Finland fick en del avlyssnad information om ryssarna under vinterkriget.

Figur 2: G-skrivaren Från Wikimedia Commons.

1.3 Mekaniska beräkningsmaskiner

I början av 1800-talet fanns mekaniska räknemaskiner för att göra vissa beräkningar. Charles Babbage konstruerade 1820 differensmaskinen som var den mest avancerade dittills. Den visade sig vara för komplicerad för att byggas fungerande. Trots det gjorde han en förbättrad och mer generell maskin som han kallade den analytiska maskinen. Den byggde på samma mekaniska ide som den tidigare maskinen med kugghjul som växlades i och ur styrda från ett hålkort med instruktioner som matades till maskinen. Lady Ada Lovelace skrev program till maskinen som skulle få den att göra beräkningar. Det visade sig att inte heller denna maskin kunde byggas till fungerande skick med dåtidens mekaniska precision och den gjorde aldrig några beräkningar och Ada fick aldrig se sina program exekverade.

Den första som lyckades bygga en fungerande modell av differensmaskinen efter Babbage idéer var svenskarna Georg och Edward Scheutz. Hans maskiner som kunde både beräkna värden av olika uträkningar och trycka dem direkt till tabeller blev prisbelönt men också väldigt dyr att tillverka så de gjordes bara i några få exemplar. Dessa maskiner användes för att trycka matematiska tabeller som användes som räknehjälpmedel vid beräkningar av vetenskaplig art och försäkringar.

Alan Turing var en brittisk matematiker som i sina studier om lösbarhet för matematiska problem använde sig av en tankemodell med en hypotetisk apparat som kunde göra några olika logiska operationer. Han visade att med några logiska operationer gick det att göra en sån maskin som kunde beräkna alla beräkningsbara problem, den kallas efter honom Turingmaskinen. Han visade också matematiskt att varje annan maskin som kan beräkna alla beräkningsbara problem kan emulera och emuleras av Turingmaskinen. Sådana maskiner kallas Turingkompletta. Det följer av Turings slutsatser att alla Turingkompletta maskiner kan lösa samma problem och emulera varandra. Det har i efterhand visat sig att Babbage utan vetskap om Turings senare upptäckt gjort en turingkomplett maskin med sin analytiska maskin.

1.4 40-talet, De första datorerna

Efter Babbage gjordes inga lika framstående försök bygga generella räknemaskiner på 100 år, förrens andra världskriget bröt ut. Med kriget som hade många nya vapen följde helt nya strategier och nya krav. Mycket beräkningar behövdes till att beräkna projektilbanor till artilleriet, knäcka krypton från krypteringsapparater och göra väderprognoser för flygvapnet.

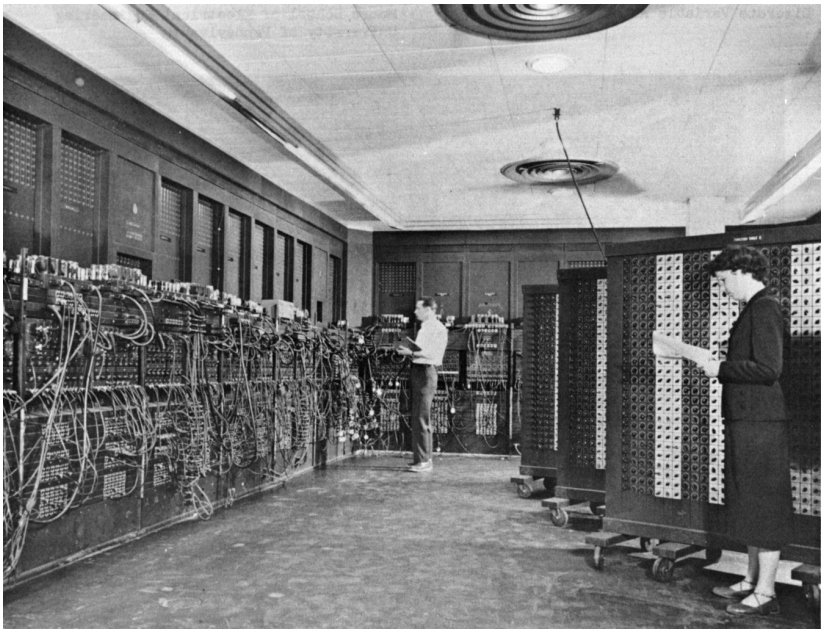
Figur 3: Replika av Z3 på Deutsches Museum. Foto: Wikipedia de användare Venusianer Från Wikimedia Commons.

Först ut var Konrad Zuse i Tyskland. Han byggde helt med egen finansiering en mekanisk maskin som bestod av en stor mängd stansade plåtbleck som flyttades runt och utförde logiska operationer i sina föräldrars vardagsrum. Den kallade han Z1, den drevs av en elmotor och kunde programmeras med en hålremsa. Den var inte turingkomplett och i praktiken inte användbar för att den ofta stannade i mekansikt krångel. Med maskinen kunde dock Zuse visa att hans koncept till räkneapparat fungerade, nazisterna fick upp ögonen för iden och han fick göra en förbättrad och stabilare version med 2600 reläer, Z3. Den drog 4 kW kontinuerlig effekt i klockfrekvensen 5,3 Hz och stod klar 1941. Zuse insåg från början att en maskin som använde elektronrör skulle vara mycket snabbare än tekniken med telefonreläer och ville direkt bygga en ny konstruktion med elektronrör men nazistregimen ansåg den “nicht Kriegswiktig”, alltså inte viktig för kriget och han fick ingen finansiering för det. Tyskarna använde Z3 för att beräkna projektilbanor tills

den förstördes i bombningarna av Berlin 1943. Konrad Zuse fortsatte sedan som en av efterkrigstidens främsta datorkonstruktörer. Ur rent datahistoriskt perspektiv kan man i efterhand konstatera att Z3 i princip var turingkomplett. Som den användes i praktiken var den inte en generell dator och det var först 1998 Raúl Rojas kunde visa att det går att skriva program som gör den i alla fall i teorin turingkomplett även om de kan bli ohanterbart stora.(*Rojas*, 1998)

Zuse som inte fick finansiering för att bygga en elektronrörsmaskin gjorde dock en utveckling av sin dator till Z4 som var den första i praktiken turingkompletta datorn. Den såldes till Eidgenössische Technische Hochschule Zürich och kan sägas vara den första dator som såldes och fungerade hos sin köpare.

I Storbritannien byggdes vid sidan av Bombe ett helt kluster av snabba beräkningsmaskiner (Colossi) under kriget för att knäcka tyskarnas krypterade radiotrafik krypterad med Lorentz-skrivaren som var mer avancerad än Enigma. De var avgörande för landsättningen i Normandie 1944. Turing var med som en av konstruktörerna men maskinerna var inte turingkompletta för att det inte behövdes för det specifika problemet så de brukar inte betraktas som regelrätta datorer. Efter kriget bestämde Churchill att både maskinerna och ritningarna skulle förstöras och de inblandade fick långvarig tystnadsplikt om projektet.



Figur 4: Programmering av ENIAC gjordes genom att slå om vippströmbrytarna till höger i bilden och koppla om kablarna till vänster i bilden. Från Wikimedia Commons.

I USA satte man målet högt på en gång och började bygga en stor dator, Electronic Numerical Integrator And Calculator (ENIAC), som skulle klara av att både beräkna både projektilbanor och ställa väderprognoser. Maskinen blev mycket komplicerad den byggde istället för det binära på det decimala talsystemet som gör det lättare att läsa ut resultatet av beräkningarna men det gjorde den mer komplicerad att bygga. ENIAC programmerades genom att slå om en mängd strömvippor för hand och koppla om en mängd kopplingstrådar mellan olika kontakter. Den stod klar 1946 när krigets krav var slut. Det visade sig att kapaciteten inte räckte för att göra väderprognoser, den användes sedan bl.a. för beräkningar när man konstruerade vätebomben. I ENIAC fanns 18 000 radiorör som gick sönder ganska ofta. Man kom snart på att många rör gick sönder när

de värmdes upp eller kyldes ner av att maskinen startades eller stängdes av. Man lät den därför gå kontinuerligt och gjorde byten under tiden sk hot-swap.

Under diskussioner om hur man skulle kunna bygga en bättre dator kom konceptet fram att datorn skulle kunna lagra sitt program i ett minne och inte behöva programmeras med vippor eller läsa in programmet kontinuerligt. John von Neumann skissade fram den datorkonstruktion som kommit att bli den förhärskande med beräkningsenhet, kontrollenhet och minne.

I Manchester, England, byggdes den första datorn med helelektroniskt primärminne efter von Neumanns arkitektur 1948. Den kallades Manchester Baby och var helt praktiskt oanvändbar för att den var hopplös att programmera och resultatet fick man läsa ut genom att det gick att avläsa status i minnet som prickmönster på skärmar. Men den var bara ämnad att visa på att konceptet fungerade och stod till grund för den första kommersiella datorn i historien Ferranti Mark I vars första exemplar köptes av universitetet i Manchester 1951.

I Sverige bildades matematikmaskinnämnden för att utreda möjligheterna för Sverige att skaffa en dator. Man försökte förhandla med amerikanska universitet att få köpa en dator men stoppades av amerikanska exportrestriktioner. Man skickade över några forskare på forskarutbyte till USA och de fick efter sin återkomst leda arbetet med att utveckla en egen svensk dator. Man började parallellt att utveckla en dator som byggdes med telefonreläer, vilket gick snabbare och resulterade i Binär Aritmetisk ReläKalkylator (BARK) 1950, och en med elektronrör, Binär Elektronisk SekvensKalkylator (BESK), som stod klar 1953 men var snabbare.

1.5 Första generationen datorer: Elektronrör

Den första generationen av kommersiella datorer byggdes med radiorör. Den första datorn med större kommersiell framgång var var UNIVAC I som producerades av skrivmaskinstillverkaren Remington Rand som köpt upp en avknoppning från ENIAC-projektet. Efterfrågan på datorer växte mycket mer än man kunde beräkna. En UNIVAC I såldes i början för $159 000 och efter några år hade stigit i pris till $1 500 000 utan modifieringar. De första UNIVAC modellerna var precis som ENIAC decimala.

Denna generation datorer programmerades i assembler som är en enkel omskrivning av maskinkoden till kommandon. Det allra första abstrakta programmeringsspråket kom på en av de sista radiorörsdatorerna. Fortran var språket som kom i första version till IBM 704 år 1956. Data lagrades på magnetband och interminnet utgjordes oftast av trumminnen.

Figur 5: Elektronrör Foto: John Rehwinkel Från Wikimedia Commons.

1.6 Andra generationen datorer: Transistorer

Den andra generationen datorer kom genom att transistorn som uppfanns 1947 började användas. Den första datorn med transistorer byggdes även den i Manchester 1953. Denna generation datorer programmerades med abstrakta språk som ALGOL, COBOL, Fortran eller PL1. Datorerna krymper i storlek och minidatorerna kommer som kan få plats i ett kontor och inte behöver en egen maskinhall.

Hårddiskar började användas för datalagring och ferritkärnminnen till interminne. Fjärranslutning med flera terminaler till samma dator blev vanligt även över telenätet. Som terminaler till dessa datorer användes samma teknik som redan var utvecklad för fjärrskrivare som i en del fall kunde dubblera som datorterminal och fjärrskrivare. Det var därför inget stort steg att flytta terminalerna ut över telefonnätet.

International Buisness Machines (IBM) dominerade världsmarknaden starkt. Svenska Aeroplan AktieBolaget (SAAB) hade en dataavdelning, Datasaab, som var stora på da-

(a) DEC PDP-10 Foto: Michael L. Umbricht Från Wikimedia Commons.

(b) hålremsa till PDP-11 Foto: Wikimedia commons användare Poil Från Wikimedia Commons.

Figur 6: Andra generationen. Även om både magnetband och hårddisk kom som lagringsmedel för hålremsor fortfarande vanligt, framför allt till in och utmatning. av data.

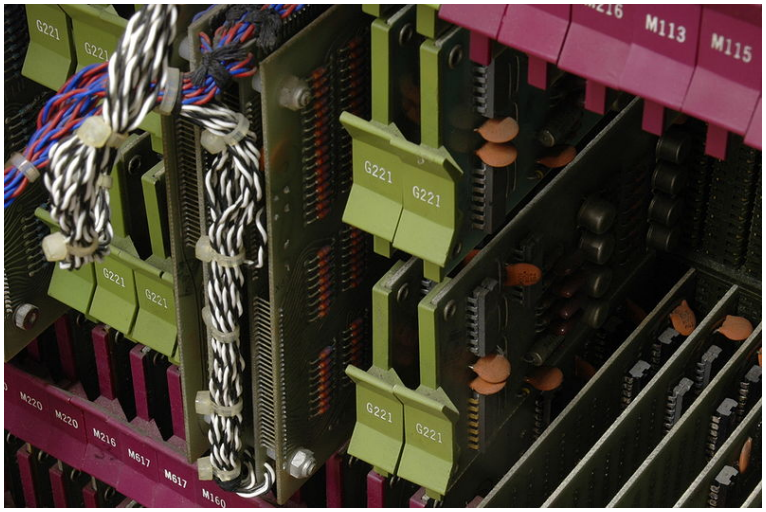
torsystem på svenska marknaden, deras primära intresse av att ge sig in i databranchen var att utveckla datorer för militära flygplan och för att användas i konstruktionen av bilar och flygplan.

1.7 Arpanet och internet

De första fjärranslutningarna till datorer gick som telegraftrafik. En telegraf liksom en telefon tar upp ett par trådar hela vägen från terminalen till datorn oavsett hur mycket signaler som går i uppkopplingen. En sårbarhet i telefonuppkopplingen är dessutom att om någon del i uppkopplingen, växelstation eller ledning, går sönder bryts förbindelsen. Den amerikanska armens förbindelseorganisation DARPA startade därför tillsammans med några universitet ett projekt för att bygga ett effektivare och säkrare system för förbindelser mellan datorer. Det resulterade 1969 i ett system med paketbaserad kommunikation. Det nya i systemet är att kommunikationen går på gemensamma ledningar i paket som i sig själva har adressen på den dator det ska till och skickas vidare från en dator till nästa om det inte är framme vid sin destination. Det löste båda problemen. Nätet som kallades Arpanet öppnades med att fyra datorer på olika amerikanska universietet kopplades samman med varandra. Nätet växte så småningom genom att datorer på nya universitet och försvarsanläggningar kopplades till. På Arpanet kom e-posten till 1971 och blev snabbt den mest använda tjänsten. Den första utvidgningen utanför USA var till NORSAR i Norge via satellit tätt följda av Tanum i Sverige och London, England 1973. Eftersom det växte mycket långt bortom DARPA bytte nätet så småningom namn till Internet.

1.8 Tredje generationen datorer: Integrerade kretsar

Med de integrerade kretsarna kunde datorerna göras än mindre och få plats vid ett skrivbord. Datorer i denna generation får operativsystem, dvs kör ett program när det startar som bara har till uppgift att hålla ordning på datorn och fördela resurser för andra program. På Xerox togs steget ett steg längre och man gjorde ett operativsystem som hade ett grafiskt gränssnitt och mus för att hantera det. Det användes i interna system på Xerox.



Figur 7: Processorkärnan i DEC PDP-8I, en tredje generationens minidator med en del integrerade krestsar. Foto: Dave Fischer Från Wikimedia Commons.

1.9 Fjärde generationen datorer: VLSI

När integrerade kretsarna utvecklats en del och tagit över de flesta funktionerna i en dator (Very Large Scale Integration (VLSI)) kunde datorer produceras på en konstnadsnivå så de kunde köpas av privatpersoner. De första datorerna som gjorde genomslag i hemmen vara Apple II och Commodore PET. När IBM som varit stora på stordatorsystem efter några år tog sig an mikrodatormarknaden med IBM-PC var det med ett helt nytt grepp. Tillsammans med datorn släppte man komplett specifikation till den och uppmuntrade andra att tillverka kompatibla delar. Det blev en succe och den standarden som därmed utveckades är idag efter en mängd uppgraderingar den totalt förhärskande på hemdatormarknaden. Standarden som gjorde datorn till en sådan succe gjorde snart att konkurrenter även började leverera kompletta datorer enligt samma standard som IBM och de blev ändå så småningom utkonkurrerade från marknaden.

Det första företag som gjorde kommers av Xerox koncept med grafiskt gränssnitt var Apple som presenterade sin Apple Lisa 1983 som inte gjorde någon kommersiell succe, men följdes upp av Macintosh 1984 som gjorde det.

Första generationen av hemdatorer hade kassettband eller floppydiskar som det vanligaste lagringsmediet, men varefter priset på hårddiskar föll kom de att bli vanliga i hemdatorer.

1.10 World Wide Web

I mitten på 80-talet anslöts forskningsanläggningen CERN till internet och kopplade upp en mängd datorer. Forskarna hade behov att dela information med varandra i sin forskning på den stora anläggningen. En av fysikerna, Tim Berners Lee, gjorde ett system där man kan skriva informationen i textfiler som har länkar till varandra, han skrev ett par program för systemet, en server som skickar ut filerna på begäran och en läsare som presenterade informationen på skärmen. Den första webbservern, webbläsaren och själva webben var födda 1989 alltså 18 år efter e-posten.

1.11 Standarder för radioöverföring

Trådlös dataöverföring genom radiovågor för data har utvecklats. De allra första mobiltelefonerna var i stort sett polisradio som hade ett system för att välja en kanal för

(a) Apple II Foto: Wikimedia commons användare Poil Från Wikimedia Commons.

(b) IBM PC Foto: Wikimedia commons användare Bofy b Från Wikimedia Commons.

Figur 8: Fjärde generationens datorer. Apple II var en av de första populära persondatorerna men IBM PC var den som nästan tog över marknaden när den kom.

två parter. Den första generationen kommersiellt mobilnät (1G) var analoga och mycket dåliga för dataöverföring. I Sverige var Nordiskt mobiltelefonisystem (NMT) första generationens nät det invigdes 1981. Man införde parallellt en andra generation (2G) med ett system för digital mobiltelefontrafik, Groupe Spécial Mobile (GSM) 1991. Den första tjänsten som blev populär förutom röstöverföring var Short Message System (SMS). Ungefär samtidigt kom Wi-Fi för trådlösa nätverk och Blåtand 1994. Parallellt med andra generationens mobilnät infördes tredje (3G), i Europa är det Universal Mobile Telecommunications System (UMTS) som är det vanliga 3G systemet. I skrivande stund är utbyggnaden av fjärde generationen igång, baserad på olika tekniker.

1.12 Frågor

1.1 a) Vad är ASCII-kod?
b) Vilken var den första tillämpningen av ASCII-kod?

1.2 a) Vem knäckte först krypteringen av den tyska wehrmacht enigma och hur användes den informationen?
b) Vem knäckte först krypteringen av den tyska G-skrivaren och hur användes den informationen?

1.3 Under andra världskriget tog utvecklingen av beräkningsmaskiner och datorer fart. De stridande parterna hade till viss del samma problem men deras projekt förknippas i första hand med var sitt problem. Nämn dess för

a) Tyskland
b) USA
c) Storbritannien
d) Nämn ett namn på en framstående person i datorutveckningen i vart och ett av dessa länder.
e) Vad kallades de tre ländernas beräkningsmaskiner?

1.4 a) När och var utvecklades tekniken med paketbaserad datatrafik?

b) Hur hade dataanslutningar fungerat för dess?
c) Vad kallas det första nätverket som byggdes?
d) Hur har det nätverket utvecklats?

1.5 Av vem och för vilka uppfanns webben?

1.6 Man brukar ungefärligt dela upp de kommersiella datorerna i fyra generationer. Beskriv för dessa fyra

a) Vilken var den vanliga komponenten för att bygga logiska kretsar?
b) Vilket var det normala användarinterfacet?
c) Vilka språk var vanliga för att programmera datorerna?
d) Vilken ny typ av dator slog igenom i generationen?

1.7 Den tyska flottan och särskilt ubåtarna drog till sig det största intresset för kryptoforcerarna på Bletchley park av flera olika anledningar.

a) Vad var det som gjorde dessa mest intressanta för britterna? Nämn flera aspekter.
b) Tyskarna insåg också att flottan var mer utsatt för avlyssning. Nämn något de gjorde i respons på detta.
c) Vid landstigningen i Normandie var ett av de högst prioriterade målen för allierade bombflyg och luftlandsättningstrupper att slå ut tyskarnas trådbundna telegraflinjer, vad var den största vinsten med det.

1.8 När tyskarna ockuperat Norge upplät Sverige telegraflinjer längs västkusten till tyskarna.

a) Vilken var den svenska baktanken med det?
b) Vilken krypteringsmaskin använde tyskarna för telegrafin till Norge?
c) Vad hette den svensk som knäckte kryptot?

2 Upphovsrätt, Licenser och PUL

Läs

- Nilsson, *Upphovsrätt, en översikt* http://www.lub.lu.se/fileadmin/user_upload/pdf/Upphovsraett_-_en_oeversikt.pdf,
- *Personuppgiftslagen PUL* http://www.mah.se/medarbetare/Juridiska-fragor/Personuppgiftslagen-PUL/,
- nedanstående.

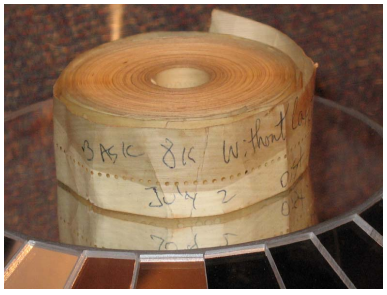
Diskutera Diskussionsfrågorna 2.7 i grupper om 3-5 personer.

Besvara Instuderingsfrågorna 2.8.

2.1 Introduktion

Under vilka förutsättningar man ska distribuera den mjukvara man har skrivit, vilka rättigheter man ska ge användaren och om och i så fall på vilket sätt man har tänkt få betalt för det arbete man lagt ner finns det många åsikter om. Eftersom datorområdet är så pass ungt är dess ursprungliga huvudkombatanter i högsta grad aktiva i denna dag. Båda har intagit sina ståndpunkter och handlat efter dem på radikalt olika sätt under de senaste decennierna sedan de känt sig lurade på sina verk.

(a) Altair 8800. Foto: Flickr användare euthman Från Flickr.

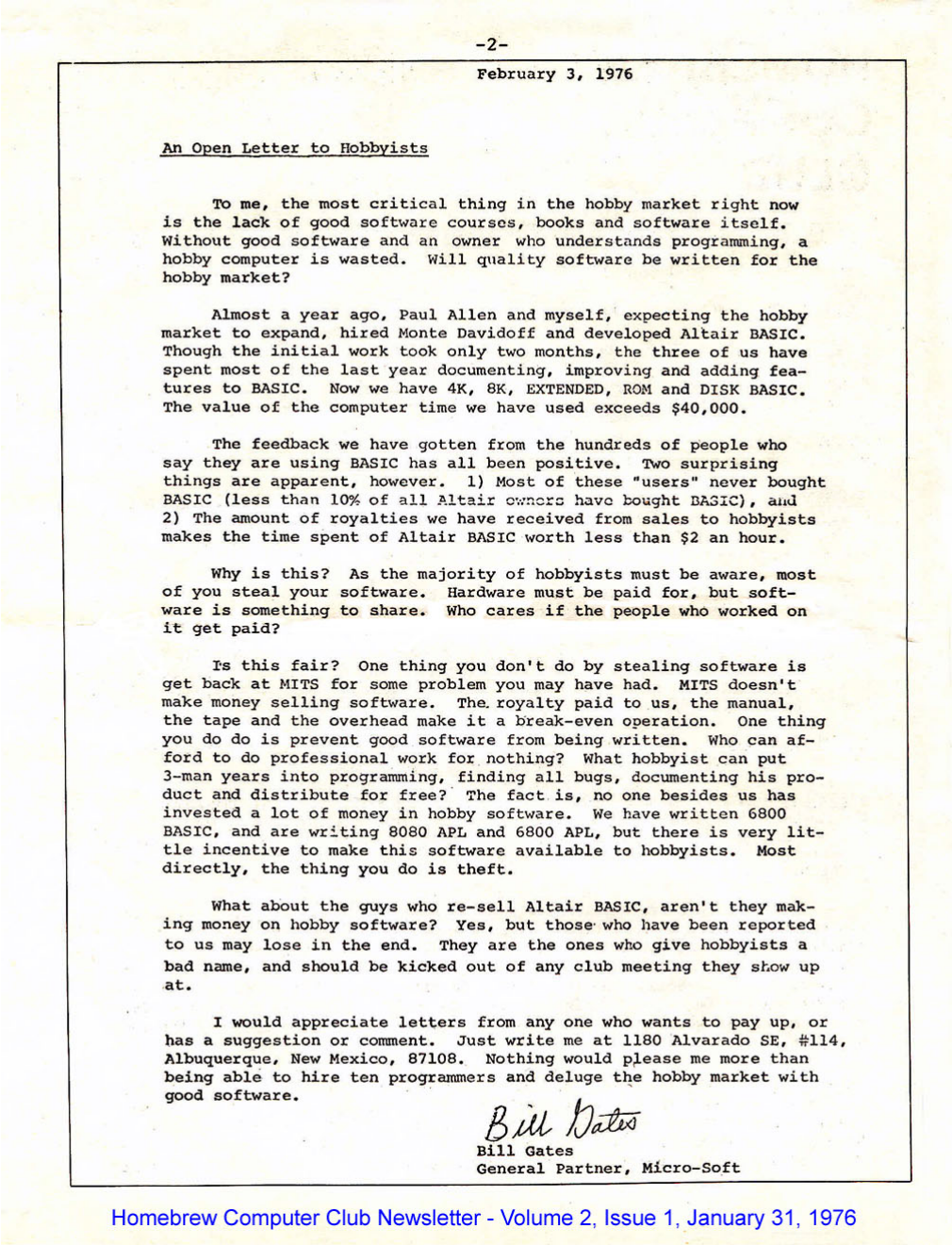


(b) Basic 8k på hålremsa. Foto: Wikipedia en användare Swtpc6800 Från Wikimedia Commons.

Figur 9: Datorn Altair 8800, såldes i 10 gånger fler exemplar än basictolken till den, här på hålremsa som var en vanlig distributionsform.

Figur 10: Bill Gates 1977.

I hemdatorernas allra tidigaste fas såg den unge Bill Gates på det lilla mjukvaruföretaget Microsoft en växande marknad. Företaget skrev en basictolk till hobbydatorn Altair 8800. Med tillverkaren MITS hade Microsoft ett avtal om royalty på de program som såldes till datorn. Det såldes dock endast en tiondel så många basictolkar som datorer och eftersom datorn utan denna var i praktiken obrukbar för gemene användaren kunde man tydligt förstå att majoriteten av datorerna körde på kopierad programvara. Han skrev "Open Letter to Hobbyists", se figur 11, som publicerades i flera amerikanska tidskrifter för hobbydatorentusiaster 1976. I dagens läge har Microsoft bundit upp i stort sett alla tillverkare av datorer för vilka deras operativsystem kan användas på att bara sälja datorer med Windows förinstallerat utan möjlighet att välja bort detta.



-2-

February 3, 1976

An Open Letter to Hobbyists

To me, the most critical thing in the hobby market right now is the lack of good software courses, books and software itself. Without good software and an owner who understands programming, a hobby computer is wasted. Will quality software be written for the hobby market?

Almost a year ago, Paul Allen and myself, expecting the hobby market to expand, hired Monte Davidoff and developed Altair BASIC. Though the initial work took only two months, the three of us have spent most of the last year documenting, improving and adding features to BASIC. Now we have 4K, 8K, EXTENDED, ROM and DISK BASIC. The value of the computer time we have used exceeds $40,000.

The feedback we have gotten from the hundreds of people who say they are using BASIC has all been positive. Two surprising things are apparent, however. 1) Most of these "users" never bought BASIC (less than 10% of all Altair owners have bought BASIC), and 2) The amount of royalties we have received from sales to hobbyists makes the time spent of Altair BASIC worth less than $2 an hour.

Why is this? As the majority of hobbyists must be aware, most of you steal your software. Hardware must be paid for, but software is something to share. Who cares if the people who worked on it get paid?

Is this fair? One thing you don't do by stealing software is get back at MITS for some problem you may have had. MITS doesn't make money selling software. The royalty paid to us, the manual, the tape and the overhead make it a break-even operation. One thing you do do is prevent good software from being written. Who can afford to do professional work for nothing? What hobbyist can put 3-man years into programming, finding all bugs, documenting his product and distribute for free? The fact is, no one besides us has invested a lot of money in hobby software. We have written 6800 BASIC, and are writing 8080 APL and 6800 APL, but there is very little incentive to make this software available to hobbyists. Most directly, the thing you do is theft.

What about the guys who re-sell Altair BASIC, aren't they making money on hobby software? Yes, but those who have been reported to us may lose in the end. They are the ones who give hobbyists a bad name, and should be kicked out of any club meeting they show up at.

I would appreciate letters from any one who wants to pay up, or has a suggestion or comment. Just write me at 1180 Alvarado SE, #114, Albuquerque, New Mexico, 87108. Nothing would please me more than being able to hire ten programmers and deluge the hobby market with good software.

Bill Gates

Bill Gates
General Partner, Micro-Soft

Homebrew Computer Club Newsletter - Volume 2, Issue 1, January 31, 1976

Figur 11: Bill Gates open letter to hobbyists.

1984 jobbade en annan man, Richard Stallman (RMS), på en liknande tolk för språket Lisp. Han blev tillfrågad om den av datortillverkaren Symbolics. Han gav upp sin upphovsrätt och släppte sin tolk till Public Domain (PD) och försåg Symbolics med koden. När sedan Symbolics vidareutvecklat den nekade de RMS tillgång till förbättringarna. Han beslöt då att undvika detta i fortsättningen och skrev den första juridiska licensen för copyleft till sin texteditor EMACS. Det tände också Stallman att starta GNU-projektet för att få till ett helt operativsystem enligt copyleft. Copylefttanken var inte helt ny men hade inte tidigare skrivits som en juridiskt hållbar licens. Grundtanken är att upphovsrättsinnehavaren till ett verk av något slag ger andra rätten att göra vad de vill med det så länge de ger samma rätt vidare och avkräver även kommande användare att göra detsamma.

2.2 Upphovsrätt och patenträtt

Figur 12: Richard Stallman. Foto: Flickr användare chrys Från Flickr.

Datorprogram innefattas av upphovsrätt. Den som skrivit ett program har med automatik upphovsrätt till detta, precis som en författare eller journalist har upphovsrätt till sin text eller en konstnär eller fotograf till sina verk. Juridiska innebörden skiljer lite mellan olika delar av världen. I anglosaxiska världen gäller copyright som från början var just rätten att framställa kopior men som i de flesta rättssystem i dag har en innebörd som mer liknar den svenska upphovsrätten. Olika rättigheter som ingår i upphovsrätten kan upphovsrättsinnehavaren överlämna till någon annan, sälja eller överge. I svensk rätt gäller det de ekonomiska rättigheterna men inte de ideella. Upphovsrätt gäller under en begränsad tid, i svensk lagstiftning är det i de flesta fall till 70 år efter upphovsmannens död, efter det hamnar verken i de flesta fall i PD.

Upphovsrätten ger upphovsrättsinnehavaren skydd mot kopiering av verket men behandlar inte alls iden eller funktionen. Detta till skillnad från tekniska innovationer som behandlas av patenträtten där man ansöker om patent på en speciell funktion oavsett hur den implementeras. Huruvida patenträtten är fungerande och fyller sitt syfte är väl upp till var och en att avgöra. Det som är klart är att den är framtagen för en industriell värld där det krävs att man för att kunna göra intrång på patentet har möjlighet att framställa varor i en industriell skala. Eftersom framställan av fysiska produkter alltid medför kostnader är dessa alltid av kommersiell karaktär och således är patent något som reglerar affärer mellan kommersiella industrier och inte av direkt betydelse för enskilda personer. Att söka patent är kostsamt och de kostnaderna ska tas hem genom att man kan göra större vinst genom sin ensamrätt.

Eftersom programvaror behandlas av upphovsrätt men inte patenträtt är hela mjukvaruvärlden till stor del uppbyggd av kloner, som i det här fallet menas ett program som gör samma sak som ett annat program men inte är en kopia av detta. I själva verket består större delen av POSIX-system av kloner. POSIX beskriver detaljerat vad en mängd program ska göra och systemen innehåller därför massor med klonade program.

Den kommersiella mjukvaruindustrin driver en lobbyverksamhet för att få programvaror att också omfattas av patenträtten. Det skulle på sikt effektivt göra det omöjligt att utveckla programvara för ekonomiskt svaga aktörer som inte har råd att varken bevaka en snårig patentdjungel eller själv söka kostsamma patent på sina egna lösningar.

2.3 BSD och MIT, fria licenser

I USA finns en stark tradition av att immateriella värden som är framtagna för skattepengar lämnas direkt till PD. Det ursprungliga Unix utvecklades vid American Telephone & Telegraph (AT&T) men snart togs den upp av amerikanska universitet framför allt University of California, Berkeley. De utökningar som skrevs på universitetet kallades Berkeley Software Distribution (BSD) och släpptes under en speciellt framtagen licens, BSD-licensen, som innebär att upphovsmannen behåller upphovsrätten men annars i stort sett var liktydig med att lämna ut källkoden till PD alltså fritt fram att använda för alla utan några andra krav på motprestation än att de som bidragit till koden skulle omnämnas i dokumentationen. Än extremare var Massachusetts Institute of Technology (MIT) som inte heller hade det kravet. För att inte behöva köpa en licens från AT&T för att köra BSD skrev man om hela systemet från grunden i Berkeley efter en rättstvist med AT&T.

2.4 Gnu och copyleft, friare än fritt?

> I don't have a problem with someone using their talents to become successful, I just don't think the highest calling is success. Things like freedom and the expansion of knowledge are beyond success, beyond the personal. Personal success is not wrong, but it is limited in importance, and once you have enough of it it is a shame to keep striving for that, instead of for truth, beauty, or justice.
>
> *Richard Stallman*

Figur 13: copyleft

Efter att RMS lämnat sin Lisp-tolk till Symbolics som PD ansåg han att det var naivt att tro på att den kod som släpptes fri till PD eller med licenser som BSD och hans eget universitet MIT använde skulle fortsätta att utvecklas och komma användarna till godo utan bara absorberas av kommersiella intressen. Han anammade därför copylefttanken som dittills förekommit i korta formuleringar i kommentarer för mindre program. Han tog hjälp av jurister att skriva en licens som skulle säkra copyleft i juridisk prövning, General Public License (GPL), och grundade Gnu's Not Unix (GNU)-projektet och Free Software Foundation (FSF) för att försäkra sig om ett operativsystem som skulle fortsätta att utvecklas fritt genom att alla som tog del av framstegen även skulle tvingas att lämna sina egna bidrag för det gemensammas bästa. Eftersom inte ens de låga kraven som ställdes i BSD-licensen kunde accepteras av GNU skrevs allt om från grunden på nytt.

Den ende som kan ställa några krav om hur ett verk ska användas är upphovsrättsinnehavaren, det är alltså centralt i copylefttanken att man behåller upphovsrätten. Om många har gjort vidareutvecklingar av ett verk under copyleft är det många som delar på upphovsrätten. Eftersom detta skulle kunna vara praktiskt besvärligt i en rättstvist är det vanligt att upphovsmännen skriver över sin upphovsrätt till en organisation de litar på (för många programvaror i GNU t.ex. FSF) som bevakar upphovsrätten och tar en eventuell rättstvist om den.

2.5 Licenser för annan information

GPL och BSD-licensen är skrivna för och används främst till datorprogram. För andra ändamål som bilder och texter finns andra motsvarande licenser för de specifika ändamålen som Lesser General Public License (LGPL), GNU Free Documentation License (GFDL) och Creative Commons.

2.6 Distributioner av operativsystem

I praktiken installerar ingen normal människa ett helt operativsystem genom att samla ihop alla de program som behövs utan installerar en distribution av ett operativ-

system. Distributioner av operativsystem görs av en organisation eller ett företag. De kan ha väldigt olika syn på licensfrågan. I ena änden av skalan finns rent kommersiella företag som levererar sina system i färdigkompilerad form med en End-User License Agreements (EULA) som är ett juridiskt avtal mellan användaren och distributören som ofta innehåller klausuler om att man inte får försöka ta reda på hur den är uppbyggd eller använda systemet för att göra en liknande kopia. Dessa måste då vara uppbyggda av kod skrivna av företaget som ger ut licensen, släppt till PD eller under BSD-liknande licenser. T.ex. Microsoft Windows.

Vissa distributörer baserar sina system på publicerar en del av sin källkod i förhoppningen att få hjälp av utomstående att bättra på den eller för att de är bundna av GPL-liknande licenser och anser andra delar vara företagshemligheter. Dessa är i praktiken baserade på antingen BSD eller GNU/Linux. T.ex. Apple OS X, Sun Solaris och Redhat Linux.

I den andra änden av skalan finner vi de som håller all källkod öppen antingen under BSD och liknande licenser eller GPL och liknande. T.ex. Debian GNU/Linux, Gentoo och OpenBSD.

Oavsett vilken typ av distribution man använder finns det inga restriktioner för att installera applikationsprogram som är fria eller slutna. Vanliga installationer är t.ex. Openoffice (GPL) på MS-windows (proprietärt) eller Matlab (proprietärt) på GNU/Linux-system.

2.7 Diskussionsfrågor

2.1 Slå upp hur din upphovsrätt behandlas om du laddar upp bilder till:

1. Wikimedia commons (http://commons.wikimedia.org).
2. Facebook (http://facebook.com).
3. Andra sociala media som ni använder.

Vilka bilder har du rätt att ladda upp till respektive databas (dina egna, andras enligt copyleft, andras i public domain) och hur kan de spridas vidare?

2.2 Se Valeji, *Att använda Linux och Gnu.* 2:a uppl. http://www.df.lth.se/~triad/gnulinux/gnulinux-2upl.pdf. Jag har tidigare använt delar av boken som kurslitteratur genom att ge länken till elever. Hade jag kunnat skriva ut den till dem istället? Läs om hur Linus resonerat och diskutera om hur ni skulle ha gjort i samma situation.

2.3 Se Collins-Sussman et al, *Version Control with Subversion* http://svnbook.red-bean.com/. Hur tror ni författarna respektive förlaget har resonerat kring licencieringen?

2.4 Se Raymond, *The Cathedral and the Bazaar* http://www.catb.org/~esr/writings/cathedral-bazaar/. Det är en bok som handlar om ideerna kring copyright, hur är själva boken licensierad? Vad har det fått för konsekvenser för spridningen?

2.5 Se Thelin, *Datorteknik 1A V2011 - Lärobok*. Boken innehåller en del bilder gemensamt med denna skrift. Hur behandlas de olika (ideella och ekonomiska) aspekterna av upphovsrätten? Varför tror ni Thelin inte skriver ut länkar till originalen?

2.6 Red Hat, Inc. är ett börsnoterat företag med 2800 anställda och en årsomsättning på 650 miljoner US-dollar. De står bakom distributionen Red Hat enterprise Linux. Hur ser affärsmodellen ut?

2.7 När Apple Inc. i slutet av 90-talet stod inför att ersätta sitt föråldrade operativsystem var MS Windows, Sun solaris, GNU, Linux och BSD redan etablerade på marknaden. Hur gjorde man sitt nya operativsystem (OS X) och hur tror du man resonerade för att komma fram till det?

2.8 Instuderingsfrågor

2.8 a) Personuppgiftslagen är till för att skydda den personliga integreteten. Vilken grundförutsättning gäller för när lagen ska tillämpas vid behandling av personuppifter?
b) Nämn någon annan lag som kan göra att PUL upphävs i ett visst fall.
c) Nämn något annat undantag då lagen inte tillämpas trots grundförutsättningen i a.
d) Vad ska man göra när man behöver upprätta ett register med uppgifter om personer som omfattas av PUL för att inte bryta mot lagen?

2.9 a) Om du sitter i din bänk och filmar en lärares föreläsning under en lektion, vem har då upphovsrätten till alstret?
b) Vad menas med verkshöjd?
c) Hur länge gäller upphovsrätten för verk som väl omfattats av den?
d) Vilka av följande sätt att använda ett upphovsrättsskyddat verk har upphovsrättsinnehavaren rätt att bestämma över?
A. Uppspelning som bakgrundsmusik på restaurang.
B. Uppvisade av verket på lektion i skolan.
C. Uppförande av en teaterpjäs för betalande publik.
D. Skyltning med verket i skyltfönster.
E. Kopiera innehållet på sina skivor till sin mp3-spelare.
F. Kopiera datorprogram från sin stationära dator till sin laptop.
G. Sälja vidare en köpt CD-skiva.
e) Utöver ekonomiska rättigheter har upphovsmannen ideella rättigheter. Nämn en av dessa två rättigheter.
f) Nämn två skillnader mellan upphovsrätten för en bok och ett datorprogram.

2.10 a) Beskriv med egna ord principen bakom en copyleft-licens. Jämför t.ex. Creative commons BSD-lika licens `http://tinyurl.com/335p7ho` med deras copyleftlicens `http://tinyurl.com/69n9w7a` för en enkel beskrivning.
b) Vad är det man vill komma åt med de juridiskt komplicerade copyleftlicenserna?
c) Den här skriften är släppt under en copyleftlicens. Är det frivilligt gjort?

2.11 Betrakta bilderna i det här avsnittet i kompendiet. Till vissa finns det en referens med länk till upphovsrättsinnehavaren men inte till andra. Varför? Behövs det inte till de andra? (Figur 10 är Från Wikimedia Commons. och figur 13 är Ill.: Wikimedia commons användare Zscout370 Från Wikimedia Commons.)

3 TCP/IP

Mål Efter dagens övningar är det meningen att ni ska

1. förstå den principiella uppdelningen i olika nivåer av kommunikationsprotokoll i TCP/IP-stacken och
2. vad som menas med
   (a) paket,
   (b) portar,
   (c) server,
   (d) nätverksprotokoll.

Läs
- http://www.lysator.liu.se/~kjell-e/tekla/linux/security/tcpip.html utom avsnitten nätmask och routing.
- Nedanstående

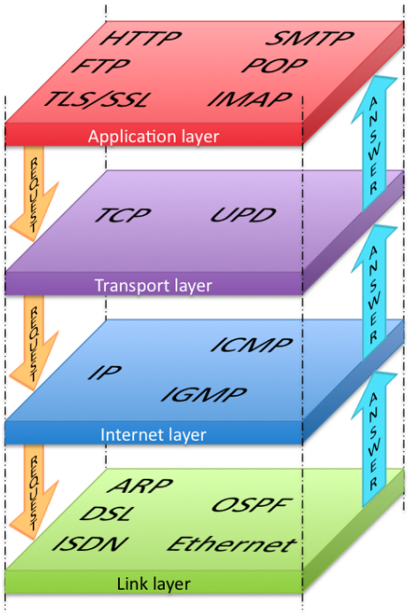
Besvara Instuderingsfrågorna 3.6.

Laboration Gör uppgifterna **??** till **??**. Utförandet ska beskrivas och resultaten skrivas ned i en individuell laborationsrapport.

3.1 TCP/IP-stacken

För att ordna trafiken i datornätverk måste den struktureras upp mycket. Det görs i några olika nivåer som jag ska försöka förklara här. Varje nivå har några olika protokoll, dvs överrenskomna standarder som de använder för att kommunicera. Det sätt att strukturera upp det här idag är nästan uteslutande den sk TCP/IP-stacken som använder fyra nivåer.

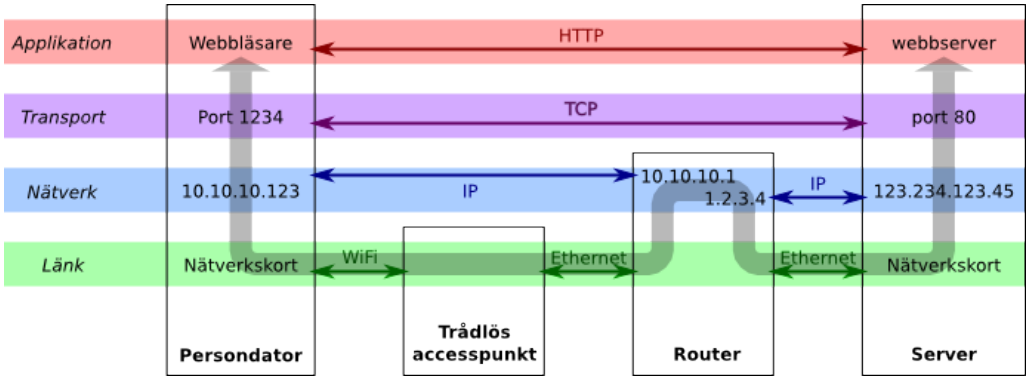
Ett protokoll behövs för att två parter ska kunna förstå varandra. Det enklaste sättet att förklara stacken är att det behövs ett lager för varje par av aktörer. Länklagret för varje fysisk burk i systemet, internet för varje delnät i nätverket, transportlagret för datorerna i respektive ände och applikationslagret för de två programmen som ska byta information. De två understa lagren behöver alltså ofta ändras för ett paket längs vägen.



Figur 14: TCP-IP stacken Ill.: Wikimedia commons användare Bughunter Från Wikimedia Commons.

3.1.1 Länk

Först och främst behövs den lägsta nivån för hur all trafik på som går genom en fysisk anslutning ska vara sturkturead, det kallas länkskiktet. Våra datorer är anslutna via Ethernet och era bärbara använder Wi-Fi här i skolan. För uppringda modem används PPP och det finns några till. Det här lagret ser bara till att två noder (datorer eller annan nätverksutrustning) som har en direkt förbindelse utan andra inblandade kan hålla kontakt med varandra. Vid en kommunikation mellan två datorer som inte sitter ihop direkt med varandra kan det därför användas flera olika länkprotokoll längs vägen.



Figur 15: Exempel på informationsutbyte. En person surfar på webben på laptopen och hämtar en webbsida från servern. Laptopen har trådlöst nätverk via WiFi och sitter på ett lokalt nätverk anslutet via routern till det nätverk där servern finns.

3.1.2 Nätverk (IP)

Det finns några olika protokoll i nätverksskiktet men det som totalt dominerar är Internet Protocol (IP).

Vid användning av IP överförs små paket med information och med uppgift om avsändar- och mottagaradress i form av IP-nummer. (Ett exempel på ett IP-nummer är 213.65.243.16.) Informationen kan överföras via flera alternativa vägar inom nätverket, varje enskilt paket söker sin väg. IP garanterar inte att informationen kommer fram till mottagaren, utan ansvaret för detta ligger på det aktuella protokollet i transportskiktet. Ett lokalt nätverk avgränsas av en nätverksmask som anger om transporten är lokal eller ej. Om mottagaren ligger utanför det lokala adressegmentet krävs att transporten sker med en Level-3 router. Vid koppling av ett nätverk till ett annat måste då routern "packa om" IP-paketen med ny adress.

Den version av IP vi använder heter IPv4 (Internet Protocol Version 4). Nästa version, som började att utvecklas 1991 av Internet Engineering Task Force (IETF), heter IPv6 (Internet Protocol Version 6). Specifikationen var klar 1997 och lanserades i juli 2004, då ICANN lade in IPv6-poster i sina DNS-datorer för .jp (Japan) och .kr (Korea).

I IPv4 består ett IP-nummer av 32 bitar, vilket är ekvivalent med 4 bytes och gör att antalet adresser är begränsade till drygt 2 miljarder. Vanligen anges IP-numret genom att skriva ut de 4 byten i decimal form med punkter emellan där varje siffergrupp motsvarar en byte. Många program kan hantera båda sätten att skriva inklusive ett tredje oktalt skrivsätt. Istället för tal anges Internetadresser ofta som en textsträng (domännamn), vilka sedan vid behov automatiskt slås upp i en stor katalog (en Domain Name Server (DNS)) till ett IP-nummer.

För att adresserna i IPv4 ska räcka krävs det att vissa tilldelas dynamiskt. Adresserna kan på det sättet användas av flera kunder, dock inte samtidigt. De s.k. fasta IP-numren tilldelas åt de datorer eller lokala nätverk som har fast uppkoppling som bredband eller egna anslutningar till företag o.s.v. De dynamiska IP-numren tilldelas oftast hemanvändare som får ett nytt nummer varje gång de kopplar upp sig mot Internet.

Vare sig statisk eller dynamisk så tilldelas adressen när datorn slås på (via Dynamic Host Configuration Protocol (DHCP)) eller när datorn ringer upp en modempool (via Point to Point Protocol (PPP).) IP-numret delas ut av internetoperatören såvida den inte används i ett lokalt nätverk. Det finns tre fördefinierade block av adresser som kan användas för lokala nätverk.

Felhantering IP-paket kan irra bort sig i nätverket om routrarna på vägen har dålig koll på var de ska skicka dem vidare eller om inte mottagaren är tillgänglig. Datorn kan ju

vara avslagen eller ha nätverksproblem. För att inte paketen ska irra runt i evighet mellan olika routrar och fylla upp internet med skräp finns det en nedräkningsprocess inbyggd. När man skickar ett paket anger man hur många routrar det maximalt får använda innan det är framme. Varje router som tar hand om paketet och skickar det vidare minskar den här siffran med ett. När det kommer ner till noll kastas paketet bort.

För att hålla reda på datapaketen i nätverket finns även ett särskilt protokoll för övervakning, Internet Control Message Protocol (ICMP). När t. ex. ett datapaket kastas bort för att det tagit för lång väg till sitt mål skickar den router som gjorde det ett paket tillbaks till avsändaren med information om att paketet försvann. ICMP paketen har inte lika hög prioritet som datapaketen och det skickas t. ex. ingen information om när de kastas bort av samma skäl.

Normalt finns ingen information om vilken väg paketen går mellan två punkter men om man vill reda ut det kan man använda tricket att skicka ett paket som man säger inte får passera någon router alls på vägen. Den första routern man kommer till kommer då att kasta bort det och skicka tillbaks ett felmeddelande. Så kan man skicka ett till paket mot samma mål och säga att det får passera en router på vägen och på så sätt få ett felmeddelande från nästa router på vägen. Om man tar tid på hur lång tid det tar innan felmeddelandet kommer tillbaks ser man hur lång tid de olika stegen tar. På så sätt kan man kartlägga vilken väg, eller få statistik på vilka vägar, om de ändras som paketen tar. Det kan vara intressant att se om man t. ex. tycker att man har dålig anslutning var flaskhalsen sitter. Ett program till linux som fungerar efter den här principen är traceroute, till windows finns tracert som gör samma sak.

3.1.3 Transport

Transportskiktet håller reda på hur de små paketen som skickas ska behandlas. Det talar bland annat om vilken port som paketet ska till på den mottagande datorn.

Det klart vanligaste som används för att föra över filer (t. ex. när man surfar på nätet som ju handlar om att ta hem en massa filer till datorn) är Transmission Control Protocol (TCP). Det ser till att de paket som kommer fram till mottagaren inte har ändrats på vägen genom en kontrollmekanism. Det ser också till att alla paketen har kommit fram och sorterar upp dem i ordning. En fil är oftast uppdelad på en mängd paket och eftersom de kanske har tagit olika vägar är det inte säkert att de kommer fram i samma ordning som de skickades. När de sorterats kan de sättas ihop till en fil igen. Om något paket i serien saknas skickar den mottagande datorn en fråga till den sändande att skicka det igen tills det finns en komplett samling.

För en del tidskritiska överföringar har man inte tid att samla ihop och sortera upp paketen, då används det mycket enklare User Datagram Protocol (UDP) som inte har dessa funktioner utan bara kastas ut på nätverket på vinst och förlust. Eftersom det används just för tidskritiska överföringar brukar de ofta ges högre prioritet i nätverket. Exempel på användingar är tidsangivelser från en tidsserver för att ställa klockan på datorn, det är uppenbarligen värdelös information att fråga om, men även i en del actionspel och röstöverföring med IP-telefoner får man hellre tappa bort en del paket och än att det ska bli uppehåll och hacka.

3.1.4 Tillämpningar

Den fjärde och högsta nivån är tillämpningslagret eller applikationslagret. Det är det här man normalt kommer i kontakt med som användare. Det är det protokoll som programmen man använder har för att prata med varandra. Vanliga protokoll är Hyper Text Transfer Protocol (HTTP) som används för World Wide Web (WWW), Internet Relay Chat (IRC) som änvänds för chatt, File Transfer Protocol (FTP) för filöverföringar, Bittorrent för fildelning och Secure SHell (SSH) för krypterade överföringar. Även bakom

kulisserna används en del tillämpningsprotokoll för att få nätverket att fungera t. ex. används DNS för att fråga efter vilket IP-nummer som motsvarar ett datornamn.

3.2 Datornamn

Varje enskild dator i ett nät har ett IP-nummer, det är ett unikt nummer i det nätet så att nätverksutrustningen kan skicka paketen till rätt dator. Eftersom det är svårt för människor att komma ihåg många långa nummer kan man i stället för IP-numren ofta ange ett datornamn. Det är då avgränsat med punkter så att den sista delen ger den toppdomän som datorn tillhör, på internet är de t. ex. se för sverige com för kommersiella företag och org för organisationer, till vänster kommer mer detaljerad information om underdomäner och före den första punkten kommer själva namnet på datorn. För att få tag på den dator som har namnet måste någon hålla reda på vilket namn som motsvarar vilket nummer, denne någon kallas Domain Name Server (DNS). En fördel med att använda namn istället för IP-nummer är att man enkelt kan byta namn på datorer i ett nätverk och att en dator kan ha flera olika alias (smeknamn) i DNSen. Om ni tänker på datornamn ni brukar se så är det väldigt ofta datorer heter www. I praktiken är det förmodligen väldigt få datorer som har det namnet, det är ett typiskt alias som man använder på den dator som man använder som webserver. Om man sen vill byta webserver så kopplar man in en ny dator på nätverket sätter upp en webserverdemon på den och när allt fungerar som man vill byter man bara ut aliaset www i DNSen så att det pekar på den nya datorn. På stora webbplatser som har mycket trafik, (sökmotorer, stora tidningar etc.) kan inte en dator ta hand om alla förfrågningar, då låter man DNSen ge olika IP-nummer varje gång till en massa datorer som allihop kör varsin webbserver med samma innehåll som får dela på bördan.

3.3 URL

En Uniform Resource Locator (URL) är adressen till en fil eller tjänst på nätet. Den består normalt av först en del som talar om vilket applikationsprotokoll som används följt av ett kolon och två snedstreck (`http://`, `ftp://` etc.). Det som kommer därefter beror på protokollet men för HTTP kommer namnet eller IP-numret på datorn som har tjänsten eller filen avgränsad av punkter (`www.fhsk.kiruna.se`, `192.168.87.90` etc.) och därefter en sökväg på datorn avgränsad av snedstreck (`/~johan/`, `/pub/os/linux` etc.). Det är alltid de här avgränsarna som gäller, idag har det blivit ganska vanligt med sk. nätfiske (phishing). Lurendrejare vill av någon anledning försöka ge sken av att vara någon de inte är på nätet t. ex. en känd bank. Då gäller det att ha en adress som ser trovärdig ut och det de ofta luras med är att någon av avgränsarna är utbytt, `http://www.bank.se.sv/login` är en dator i toppdomänen El Salvador medans `http://www.bank.se/sv/login` är en i toppdomänen Sverige!

3.4 Portar

Begreppet portar på datorer är lite tvetydigt. Från början när en dator var en isolerad burk som man ville ansluta något till gjorde man en kontakt där det gick att anluta detta något. Det gick då att sätta en sak där som skickade sina signaler via kontakten och den kalldes en port. Exempel så sådana är serieporten och parallellporten som t. ex. användes till skrivare.

När man började sätta ihop datorer i nätverk uppstod problemet att man ville använda samma kabel mellan datorer för flera olika program på de olika datorerna och helst dessutom samtidigt eller i alla fall blandade efter varandra. I transportprotokollen i IP-stacken finns det därför med i paketen en siffra som är en signal om vilket program på den mottagande datorn det är till. Tillämpningsprotokollen har särskilda standardvärden för dessa så man behöver sällan bekymra sig om dem. Det finns dock en del tillfällen då

man vill frångå standarden, speciellt om man kör flera program av samma sort måste man kunna skilja på vilket det är till, då säger man åt det ena av dem att lyssna på ett annat portnummer än det normala.

Eftersom olika tillämpningsprogram har olika stor säkerhet mot attacker och avlyssning används portnumret ofta för att bestämma vilka paket som ska få komma igenom en brandvägg.

3.5 Demoner

För att två datorer ska kunna prata med varandra krävs både att en pratar och att den andra hör när någon ropar på den. Program som skickar ut förfrågningar och alltså 'pratar' är vanliga program ni är vana att använda t. ex. webbläsare, de startar man när man vill använda dem. De program som lyssnar måste däremot göra det hela tiden, de kan ju inte veta när någon ska ropa. Att starta ett sånt program på sin dator kallas att sätta upp en server. (Alltså ett till ord som har flera betydelser, en server kan vara en fysisk dator som står i en datorhall och har till uppgift att svara på sådan anropstrafik men också en tjänst som en dator tillhandahåller.) Programmen som man kör när man har satt upp servern kallas demoner. De ligger i datorns minne och tar plats och ska därför vara så små som möjligt och efter de är startade behöver de inte anväda processorn förrens det kommer ett anrop på den port som demonen är satt att lyssna på. Det som då händer är oftast att demonen delar på sig i två processer, en svarar på det anropet som kom och håller kontakten med den som ropat, de sätter upp en egen länk mellan sig, och den andra går tillbaks till det föregående läget och fortsätter att ligga och lyssna på nya anrop.

3.6 Frågor

3.1 a) Förklara med egna ord vad varje nivå i TCP/IP stacken har för uppgift för att få en uppkoppling att fungera.
b) Vilket eller vilka protokoll i respektive lager används då du sitter med din laptop i klassrummet och öppnar en webbsida?

3.2 Vilken är den viktigaste orsaken att man utvecklat en IP version 6 för att ersätta version 4 som vi använder idag?

3.3 Vad gör en DNS?

3.4 Vad menas med en dynamisk IP-adress?

3.5 Hur undviker man att hela internet översvämmas av internetpaket som är adresserade till en mottagare som inte finns eller inte svarar?

3.6 Beskriv skillnaden i funktion för TCP och UDP och ge exempel på när man använder det ena eller andra.
a) Beskriv skillnaden i funktion för TCP och UDP.
b) Ge exempel på när man använder TCP.
c) Ge exempel på när man använder UDP.

3.7 Hur är datornamnet avgränsat i en Uniform Resource Locator (URL)?

3.8 Portar är ett tvetydigt begrepp i datorsammanhang. Beskriv vad som menas i båda betydelserna.

3.9 Vad menas med en demon?

3.10 Hur används en port som bevakas av en demon respektive en port som inte gör det?

3.11 En server kan beskriva både hård och mjukvara, beskriv de två meningarna.

4 Webbserver

Vi börjar med att skriva dokument i Hypertext Markup Language (HTML) och installerar så webbservern Apache.

Mål Målet med dagens övningar är att alla ska kunna skriva en webbsida, och lägga ut den på nätet med en webbserver.

Läs Nedanstående

Besvara Instuderingsfrågorna i stycke 4.6.

Övningar Gör övningarna i stycket 4.4 i grupper om 2-3 personer. Redovisning av uppgiften är att ni visar upp att ni har en fungerande webbserver i gruppen innan ni stänger av datorn.

Laboration Gör uppgifterna i stycket 4.5. Utförandet ska beskrivas och resultaten skrivas ned i en individuell laborationsrapport.

4.1 Webb

WWW är ett system av sammanlänkade hypertextdokument som är åtkomliga för datorer anslutna till internet. Det uppfanns av Tim Berners-Lee vid Conseil Européen pour la Recherche Nucléaire (CERN) 1991. Webben har sedan utvecklats med en del och innefattar en mängd olika filformat. För att standardisera dessa har det bildats ett internationellt standardiseringsorgan World Wide Web Consortsium (W3C) som leds av Tim Berners-Lee, (se figur 16). Alla organisationer kan bli medlemmar i W3C och vara med och bestämma hur standarden ska se ut men medlemsavgiften är ganska hög så det är främst regeringar och stora företag och organisationer som är medlemmar.

Figur 16: Tim Berners-Lee kom på WWW och leder idag W3C Copyright: Flickr användare captsolo Från Flickr.

4.2 HTML

HTML är ett strukturerat språk för att presentera i första hand text och bilder. Formateringen av innehåller görs med sk taggar, t ex en huvudrubrik görs genom att skriva `<h1>Rubrik</h1>`. Det är `<h1>` och `</h1>` som är taggarna. De kommer oftast i par där sluttaggen är lika som öppningstaggen förutom att de börjar med ett snedstreck.

4.2.1 Hypertext

Med hypertext menas att man kan länka ihop dokument genom att göra vissa ord (eller bilder eller andra element) har funktionen att de innehåller en adress till ett annat dokument eller en annan plats i samma dokument. Det har man dels för att man länka in referenser till helt andra dokument som ligger på internet, dels använder man dem för att strukturera upp sin webbplats. Det typiska användandet är att man har en första presentationssida med en rubrik som visar var man kommit och en innehållsförteckning över vad som finns på webbplatsen i form av länkar.

4.2.2 Struktur vs. design

Det som är utmärkande för HTML är att den som skriver dokumentet kan indikera vad hon vill ha fram för effekt men det är webbläsaren som användaren använder för att visa det som bestämmer hur det presenteras på skärmen. Det skiljer jobbet att forma ett dokument för en webbsida från att göra ett dokument som ska skrivas ut på papper genom att på ett papper kan man bestämma precis hur det ska se ut medans en webbsida får man se till att den är väl strukturerad och lita på att webbläsaren visar det på ett bra sätt för läsaren. Ett vanligt misstag när man skriver webbsidor är att man försöker detaljstyra hur presentationen ska se ut med mängder av HTML-taggar och tittar på resultatet i en webbläsare och tycker att det blir snyggt och inte tänker på att resultatet kan bli helt annorlunda i en annan webbläsare. För att råda bot mot detta har man tagit fram ett strukturerat sätt att skriva önskemål om hur den grafiska presentationen ska se ut i det komplementerande språket Cascading Style Sheets (CSS). Att skriva sådana kommer vi att ta upp i avsnitt 6.

Det har på senare år med stora datorskärmar på stationära datorer, smartphones, surfplattor och netbooks blivit mycket mer viktigt att dela upp innehåll och designformatering eftersom de är så olika.

4.3 Webbserver

En normal webbsida i ren HTML går att titta på direkt genom att öppna filen i webbläsaren. För att någon på en annan dator ska kunna se sidan som ligger på din dator behövs en webbserver. Den enklaste webbservern är en demon som lyssnar efter anrop på port 80 och bara svarar med att skicka ut filer helt opåverkade.

Vi ska använda den absolut vanligaste webbservern – Apache. Den kan fås att göra mycket mer än att bara skicka ut filer. T. ex. kan den interpretera program som är skrivna i PHP Hypertext Preprocessor (PHP) och skicka ut resultatet i HTML.

Sådana extrafunktioner för webbservern brukar behandlas genom att man installerar olika hjälpprogram till den, i Apache heter de moduler, själva webbserverprogrammet ska ju helst vara så litet som möjligt eftersom det ligger och kör hela tiden och tar kraft från datorn. Till webservern finns alltså en mängd olika moduler varav man laddar in de som man vill använda.

Figur 17: Tim Berners-Lees NeXTcube, världens första webserver. Foto: Wikipedia en användare Coolcaesar Från Wikimedia Commons.

För att ge snabb respons startar Apache flera demonprocesser, när ni har fått igång apache kan ni se med `ps aux`. Eftersom demonen inte är en utan flera processer finns det ett särskilt skript för att starta och stänga av den. Det ligger tillsammans med en mängd andra liknande skript i katalogen `/etc/init.d/`. De skript som ligger där körs normalt när datorn startar upp. Om man behöver starta om webbservern gör man det med kommandot `/etc/init.d/apache2 restart`.

4.4 Övningar

- Ö1 Gå igenom http://www.w3.org/MarkUp/Guide/ och gör en HTML-sida som har en överskrift, innehåller en lista och/eller tabell, en länk till en annan webbsida och en bild. Sidan ska vara strukturerad med doctype head och body. Använd en

texteditor för att skriva koden och kontrollera hur den ser ut genom att öppna filen med webbläsaren.

*Ö2 Använd W3Cs valideringstjänst http://validator.w3.org/ för att se att sidan är korrekt skriven.

4.5 Laboration

Laborationen får gärna göras i grupper om två eller tre personer men var och en ska redovisa med en egenhändigt författad labrapport.

Till övningarna behöver du ett Linux-system. Eftersom vi ska sätta igång serverprocesser och ändra i konfigureringarna till servern är det tryggast att alla startar upp från ett externt medium, CD eller USB-sticka.

4.5.1 Redovisning

Redovisningen av laborationen ska du göra enskilt i form av en laborationsrapport vare sig du gjort laboration själv eller i laborationsgrupp. Rapporten ska skickas in via e-post till `johan.arvelius@rymdgymnasiet.com`. Den ska innehålla:

Försättsblad eller huvud med rubrik, namn, datum.

Material och metod Tala om vilket operativsystem du använt, vilka programvaror du använt och kortfattat hur de fungerar.

Utförande där du redogör för vad du gjort och tar med de saker om står upptagna under Uppgifter. Utförandet kan gärna innehålla men får inte enbart bestå av kommenterade utdata som genererats av program under uppgifterna.

Slutsatser Summera resultaten. Vad var det viktigaste i laborationen? Sammanfatta kort och kärnfullt. Även reflexioner över vad som hände och vad du lärt dig.

4.5.2 Uppgifter

•Ö3 Den här övningen går ut på att installera en webbserver och konfigurera den.

1. Starta upp en dator med ett linuxsystem som inte har en webbserver installerad från CD eller USB-sticka. Detta (och resterande uppgifter) kan du med fördel göra gemensamt med någon kamrat, så kan ni använda en dator som server och en för att googla med om det är några problem.

   Se till att ni kan komma åt nätverket och logga in med er vanliga inloggning genom att öppna webbläsaren. Om det är problem, börja med att klicka på nätverkssymbolen och välja rätt nätverk. Om inte nätverkskortet kommer igång alls, prova att ansluta datorn till ethernet med kabel.

2. Gör en katalog som heter `public_html` direkt i er hemkatalog (d. v. s. med det totala namnet `\home\user\public_html`). Det kan man göra genom att välja menyn *platser hemmamapp* så i det nya fönstret *Arkiv Skapa mapp* eller med kommandona

   ```
   $ cd
   $ mkdir public_html
   ```

3. Flytta filen ni gjorde i övning Ö1 till katalogen `public_html` som ni just gjorde och döp den till `index.html`.

4. Starta webbläsaren Firefox och skriv in adressen `http://localhost/~user` i webbläsarens adressfält och se om ni får upp något. (Det bör ni inte få.) Tecknet `~` kallas tilde och det brukar man kunna skriva genom att hålla ned tangenten Alt Gr och trycka på `^`, om inget tecken kommer kan man behöva trycka ner mellanslagstangenten efter.

5. Installera webbservern. Det gör ni genom att välja menyn *System/Administration/Pakethanteraren Synaptic*. Sök efter och välj att installera paketet `apache2`, då bör ni få en lista med andra paket som behövs för att installera dem (beroenden). Installera den listan ni får. Samma sak går att göra med kommandot

```
$ sudo apt-get install apache2
```

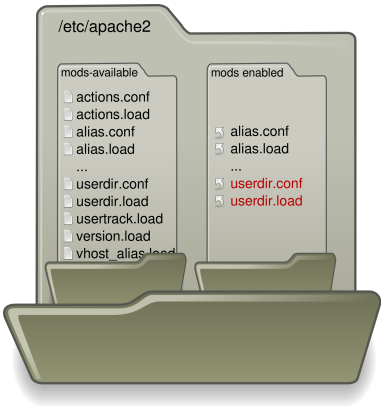
6. Beroende på vilken distribution du använder kan den ha en konfiguration där modulen för att visa användarnas kataloger laddas. Skriv in adressen `http://localhost/~user` i webbläsaren igen och se vad som händer nu. Om ni ser er sida är det klart annars måste man själv lägga till länkar till modulerna. Det görs genom att gå till katalogen `/etc/apache2/mods-enabled` där moduler som ska laddas ligger och göra än länk till modulen som ligger i `/etc/apache2/mods-avaliable/`, det är två filer som ska länkas, `userdir.conf` och `userdir.load`. Det görs med kommandona

```
$ cd /etc/apache2/mods-enabled
$ sudo ln -s ../mods-available/userdir.conf .
$ sudo ln -s ../mods-available/userdir.load .
```

   Observera mellanslagen och punkten på slutet.

   Slutligen ska webbservern startas om med kommandot

```
$ sudo /etc/init.d/apache2 restart
```



Figur 18: Alla moduler som finns installerade ligger i `mods-available` men apache använder de som ligger i `mods-enabled`. För att använda en modul får man lägga in länkar till den modul man vill använda.

7. Skriv in adressen `http://localhost/~user` i webbläsaren igen och se vad som händer nu. Den här gången borde ni om ni följt instruktionerna se sidan ni har skrivit. Tecknet ~ (tilde) i adressen betyder att användarnamn är en katalog i `/home/`. Webbservern är, eventuellt efter vår egen konfigurering, konfigurerad så att den letar efter och visar innehållet i den speciella katalogen `public_html`, (det vore ju inte helt lyckat om den la ut allt som låg i hemkatalogen på webben så att vem som helst får se). Om adressen slutar med namnet på en katalog och inte en fil letar den efter filnamnet `index.html`. Var noga med att visa upp för läraren att du har en fungerande server innan du stänger av datorn.

4.6 Frågor

4.1 a) Vilket program är det som bestämmer hur en webbsida presenteras för den som tittar på den, webbservern eller webbläsaren?
   b) Vad är vitsen med att kontrollera sin HTML-kod med en validerare istället för att helt enkelt kolla att den ser bra ut i webbläsaren?
   c) Standarder för webben slås fast av W3C. Vilken typ av medlemmar har de?
   d) Vad är webbserverns jobb för en webbplats med statiska HTML-sidor?
   e) Vad är webbserverns jobb för en webbplats med interaktiva sidor?

4.2 Vad är skillnaden på HTML och eXensible Hypertext Markup Language (XHTML)?

4.3 a) Vem uppfann HTML och därmed webben?
b) Vilken organisation har idag hand om att definiera standarder för de protokoll som används på webben?
c) Vem leder den organisationen?

5 SSH

Läs
- Nedanstående
- Dokumentation till SSH implementeringen enligt länk i texten nedan.

Övningar Gör övning Ö4

5.1 Teori

SSH är ett system för krypterad inloggning och tvåvägskommunikation mellan två datorer. Systemet bygger på att den som ska logga in på en dator med den kryperade anslutningen har ett användarkonto tillgängligt på den datorn. Den dator man ska logga in på måste ha en demon som lyssnar efter SSH-anrop, normalt på port 22. Den dator man använder för att göra anlutningen från behöver ha ett SSH-klientprogram.

Kommunikation sker med symmetrisk kryptering genom utbyte av symmetriska nycklar via asymmetrisk kryptering, se stycke 14.4. Ett asymmetriskt nyckelpar för varje dator skapas vid installation av programmen. När man loggar in första gången till en dator med SSH kan man, beroende på inställningarna i programmet bli tillfrågad om att godkänna den publika nyckeln som servern sänder ut. Man får då ett digitalt fingeravtryck på nyckeln som man kontrollera mot ett fingeravtryck man fått på ett annat sätt, oftast har man inte tillgång till denna information utan väljer att lita på nyckeln. Nyckeln lagras dock lokalt och ger en säkerhet i att SSH kommer att vägra ansluta till en dator med samma namn eller IP-adress som ger ut en annan nyckel. Lagringen av nyckeln sker automatiskt.

För att logga in kan SSH antingen skicka lösenordet över den krypterade anslutningen som sätts upp mellan datorerna. Ett säkrare alternativ är att man lagrar ett till asymmetriskt nyckelpar som är privat för användaren. Dessa kan man då använda för att verifiera identiteten på användaren. Den hemliga nyckeln kan man låsa med en passfras som måste anges för att den ska fungera, på så sätt hindrar man att obehöriga som gjort intrång på datorn kan använda nyckeln för att logga in på andra system.

5.2 Praktik

Implementeringar av SSH finns många. De vanligaste är för de vanliga operativsystemen är:

- OpenSSH (`http://www.openssh.com/`) för OpenBSD och många andra posixsystem (UNIX/Linux/BSD/OSX).
- PuTTY (`http://www.chiark.greenend.org.uk/~sgtatham/putty/`) för Windows.
- WinSCP (`http://winscp.net/`) för Windows baserad på PuTTY.
- Cygwin (`http://www.cygwin.com/`) ger en hel liten UNIX-miljö som ett program i Windows och kan användas för att installera OpenSSH (eller UNIX-program i största allmänhet) under Windows. Detta är en ganska stor installation som garanterat är betydligt besvärligare än att installera WinSCP eller PuTTY.

För att skapa egna asymmeteriska nyckelpar behöver man följa instruktionerna för resptektive program. I linuxmiljön som finns på servern är OpenSSH-demon installerad.

I ett system med OpenSSH ska det finnas i en fil `/home/`*`användarnamn`*`/.ssh/authorized_keys2` för att man ska kunna logg in till den. Man kan hämta den med filöverföring i den SSH-anslutning man redan har. Genom ett kommando som `scp` eller liknande, se dokumentationen för respektive program.

Ni borde inte behöva göra så mycket på servern men om man du behöver finns en kort sammanfattning av användbara kommandon i tabell 1.

Kommando	**Vanliga växlar**	**Förklaring**
ls	-l -a -F	lista filer (*list*)
cd		byt katalog (*change directory*)
cp	-r	kopiera fil (*copy*)
mv		flytta eller döp om fil (*move*)
rm	-R -f	ta bort fil (*remove*)
chown		ändra ägare till fil eller katalog (*change owner*)
man	-k	läs hjälp (*manual*)
su	-	byt användaridentitet (*switch user*)
sudo		kör kommando som root-användare (*switch user do*)
ps	aux -u	lista processer
kill	-9	stäng ner processer
dhclient		begär ny ipadress
scp	-r	kopiera fil till/från annan dator (*secure copy*)
ssh	-X	logga in till annan dator (*secure shell*)
more, less		skriv ut texten i en fil så mycket som får plats i fönstret
nano, pico		enkla texteditorer
exit		avsluta skalet (oftast logga ut)

Tabell 1: Vanliga kommandon i POSIX-system.

6 Stil till webben

Mål Efter dagens övningar ska du kunna skilja ut stilen för din webbsida från innehållet på den och förstå vitsen med att göra det.

Läs
- Stycke 4.2.2
- Nedanstående

Övningar Gör övningarna i stycket 6.2.

6.1 Att sätta sin stil på webbsidan

Som påpekats i stycke 4.2.2 är god stil på nätet att skilja ut designen av sidor på webben från innehållet. Det kan göras med CSS. Att samma text kan ges ganska olika uttryck syns lätt vid ett besök på t. ex. http://www.csszengarden.com/, sidan talar för sig själv. Att tänka på när man gör design för webben är att hålla sig till *relativa* mått på allt, alltså t. ex. sätta bredden på en spalt till 30% av fönstrets bredd istället för att definiera den till 400 pixlar bred. Det är webbläsaren som ska presentera innehållet på skärmen och en god design är den som fungerar på en liten och stor skärm i ett smalt eller brett fönster, på skärmar med dålig kontrast etc. Det är fullt möjligt att definiera hur bred en text ska vara på skärmen men det är fantastiskt enerverande att läsa om inte en hel rad får plats t. ex. i telefonen. Att formge text kallas typografi (se t. ex. http://www.youtube.com/watch?v=Ki6rcXvUWP0) och är del i formgivningen.

6.2 Övningar

Ö4 a) Installera ett program som kan hantera filöverföringar med SSH:
- OpenSSH (http://www.openssh.com/) om du har ett UNIX-liknande system (UNIX/Linux/BSD/OSX).
- WinSCP (http://winscp.net/) för Windows.

b) Anslut till `elev.rymdgymnasiet.com/` via SCP du ska använda dig av användarnamnet och lösenordet du fått per e-post. Välj filprotokollet SCP och portnummer 22.

c) Skapa en katalog `public_html` på ditt konto på servern

d) För över den HTML-fil du gjorde på förra laborationen till katalogen `public_html` på servern.

Ö5 Använd CSS för att ändra utseendet på sidan. HTML.net har en tutorial på http://www.html.net/tutorials/css/, action börjar på lesson 2. En svensk motsvarighet finns på http://kaxigt.com/10-grundlaggande-lektioner-i-css/ Följ lektionerna och ändra stilen på din sida.

- Ändra färger på text och bakgrund.
- Sätt olika färger för länkar som redan är besökta.

Pröva dig fram och se vad du kan göra

Ö6 Strukturera upp stilmallen och lägg den i en extern fil.

Ö7 Se till att du endast har stilformatering i CSS och har rensat bort alla HTML-taggar som har med utformningen att göra.

Ö8 För över färska versionerna av CSS- och HTML-filerna till servern när du är nöjd med resultatet.

6.3 Frågor

6.1 a) Vilka anledningar finns för att skilja ut stilen för en sida från koden som strukturerar upp informationen på den?
b) Vem utformar CSS-standarden?

7 E-post

Mål Med dagens moment är det meningen att ni ska

1. Förstå hur e-postsystemet fungerar i princip.
2. Veta hur systemet har utvecklats med nya standarder och varför.
3. Förstå hur spam sprids och något om vad som görs för att undvika det.

Läs

- Nedanstående

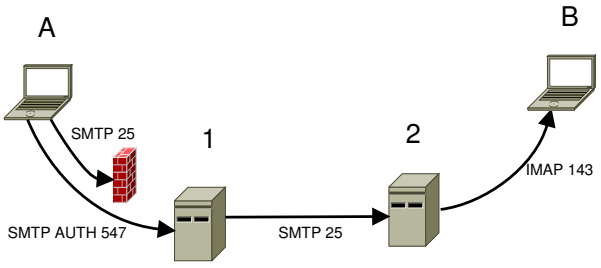
Besvara Instuderingsfrågorna 7.2.

E-post är en av de äldsta tjänsterna på internet och har funnits i en mängd olika varianter. Det mailsystem som vi använder idag med Simple Mail Transfer Protocol (SMTP) som standardprotokoll togs fram 1981 och var redan från början en kompromiss för att kunna kopplas ihop och kunna översätta meddelanden till och från de olika andra mailstandarder som fanns då. Detta märks i att det i delar är omodernt i uppbyggnaden och man har fått göra flera uppdateringar och efterhandslösningar för att motverka problemen. Från grunden är det byggt med stor säkerhet för att meddelanden når fram till adressaten medans spam och skydd mot förfalskade meddelanden och obehörig åtkomst inte alls beaktades från början.

Trots åldern är e-postleverans ett betydligt mer komplicerat system än t. ex. den mycket modernare webben och involverar ofta flera servar och applikationsprotokoll.

7.1 Ett e-brevs väg från A till B

E-post kan bara skickas mellan en avsändare och en mottagare som har varsit e-postkonto. Det ligger i sakens natur att den som skickar ett e-brev inte vet om mottagaren är uppkopplad för att ta emot sitt meddelande när det skickas och det måste då kunna lagras på en server som är i gång hela tiden tills det kommer fram. I själva verket är det en trestegsprocess som i alla fall klassiskt ser ut så här:



Figur 19: E-post från A till B.

1. Användaren A skickar e-brevet till en server där hen har sitt e-postkonto, utgående mailserver, Mailserver 1 i figur 19. Det säkerställer att användaren får iväg brevet från sin dator och kan koppla ur den från nätverket. E-brevet lagras nu som en fil på datorn där den utgående mailservern körs.

2. Nästa steg är att datorn där den utgående mailservern körs anropar den server där användare B har sin post lagrad. B:s inkommande mailserver, Mailserver 2 i figur 19. I det här kommunikationssteg är det allså Mailserver 2 som är server medans datorn som kör mailserver 1 är klient. Om inte Mailserver 2 svarar fortsätter 1 att skicka anrop till den. Om den inte fått svar inom rimlig till lägger den ett felmeddelande som ett epost till avsändaren. Om den får kontakt flyttas ebrevet vidare till Mailserver 2.

3. Nu ligger e-brevet och väntar på att B ska efterfråga det. Det sker när B startar sitt e-postprogram som tar kontakt med mailserver 2. Meddelandet kan då antingen överföras till dator B, vilket var det normala tidigare eller ligga kvar på den inkommande mailservern så användare B laddar över det tillfälligt när hen vill läsa det och sedan kastar det igen.

Överföringen från 1 till 2 i figuren görs med SMTP som är mailservrarnas standardprotokoll. Det är som sagt ett av de äldsta protokollen på internet och har vissa egenheter som ger vissa problem idag. Texten kodas enligt standarden i 7 bitars ASCII vilket bara har plats för de bokstäver som ingår i det engelska alfabetet.[1] Avsändaren anges genom att de skrivs in som metadata tillsammans med meddelandet utan någon kontroll av äktheten och meddelanden skickas i klartext utan kryptering. Serverporten för SMTP är 25.

7.1.1 Utgående mailserver

Överföringen från A till 1 kan också gå med SMTP. Ett problem med det är att SMTP inte kräver någon inloggning eller annan autentisering av avsändaren. Om användaren får en standardaansluting med SMTP till servern kan den därför utnyttjas av spammare som helt enkelt skriver in användare A:s adress som avsändaraddress i sina spam och skickar dem via Mailserver 1. Eftersom det var bredbandsleverantörerna som hölls ansvariga för att spamen som kom från deras nät stängdes port 25 från de flesta bredbandsleverantörer ned i slutet på 1990-talet. Istället används en variant av SMTP med autentisering SMTP Authentication (SMTP AUTH), där A måste kunna ge lösenord, på port 587. Alternativt kan man köra SMTP krypterat med Transport Layer Security (TLS) på port 465. A och 1 kan också efter handskakning komma överrens att kryptera överföringen med TLS även vid anslutning till någon av standardportarna 25 eller 587. *Kontentan för användaren A är att mailprogrammet ska sättas att använda port 25, 465 eller 587 med eller utan kryptering beroende på vad den utgående mailservern kräver och bredbandsleverantören tillåter.*

7.1.2 Inkommande mailserver

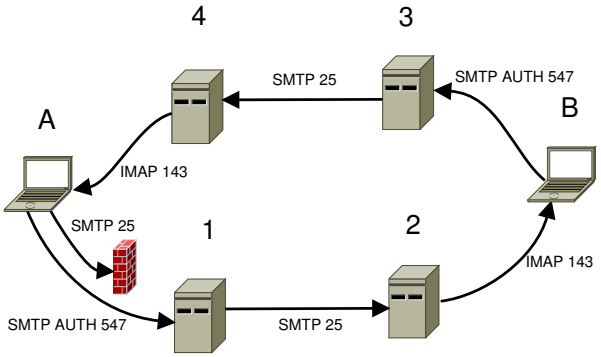
Hämtningen av post av användaren B från servern 2 kan också ske på olika sätt. Grunden i kommunikationen är att det är Bs epostprogram som är klient och anropar servern 2. Det finns i stort sätt två alternativa protokoll, det äldre Post Office Protocol (POP) och vidareutvecklingen Internet Message Access Protocol (IMAP). POPs normalkonfigurering är att föra över meddelandena till den egna datorn och radera dem på servern, medans IMAPs normalkonfigurering är att ta hem en kopia på det meddelande användaren B vill läsa från servern, låta den ligga kvar på servern 2 och kasta kopian från datorn när epostprogrammet stängs ner igen. Den senare varianten fungerar bättre i dagens samhälle där användare ofta vill läsa sin epost från flera olika datorer och ha tillgång till alla meddelanden från alla sina datorer. Man kan också välja att både låta meddelanden ligga kvar på servern och behålla dem på datorn, sk synkronisering. Då har man en egen backup på sin post om man inte litar på att den som driver ens ingående mailserver sköter backup korrekt.

7.1.3 Serverdatorer och demoner

En server i strikt mening är en process som lyssnar efter anrop på en öppen port. Det betyder att "servern" 1 (där jag med "server" menar den vardagliga meningen en dator som har till uppgift att köra serverprocesser) i själva verket måste köra minst en server

[1] De allra flesta mailservrar klarar av andra kodningssystem med tillägg men om någon på vägen inte gör det måste meddelandena fortfarande kodas om till 7 bitars ASCII av den mailklient som vill föra dit meddelandet

(demon) för en eller flera av portarna 25, 465 eller 587 och ett klientprogram för att anropa andra mailservrar. Dessa två behöver egentligen inte köra på samma dator. "Servern" 2 behöver på samma sätt köra minst två olika serverprocesser, en SMTP-server för inkommande post på port 25 och en POP- eller IMAP-server för anrop från användare med på port 110 eller 143. Inte heller dessa behöver vara på samma dator.

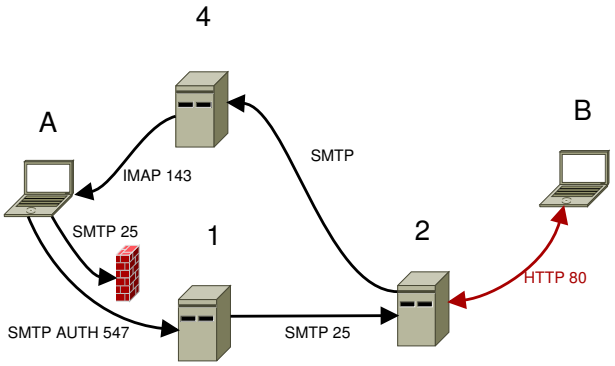


Figur 20: Motsatt riktning på försändelserna behöver inte gå samma väg.

Den motsatta vägen att skicka ett ebrev från B till A involverar inte en enda av de server eller klientprocesser som behandlats eftersom de alla är enkelriktade. Det är inte ens säkert att motsvarande servrar i motsatt rikning (3 och 4 i figur 20) har något som helst med 1 eller 2 att göra och inkommande och utgående mailserver måste oftast ställas in var för sig i e-postprogrammet.

7.1.4 Webmail

På senare år har det blivit mer populärt att låta "mailservern" alltså datorn som har hand om användarens epost köra en webbserver med ett webbgränssnitt som man kan ansluta till med vilken webbläsare som helst, så kallad webmail, se figur 21. Då har man ingen in- eller utgående mailserver utan kommunikationen till den gemensamma webmailservern sker med HTTP på port 80 eller hellre krypterat Hyper Text Transfer Protocol Secure (HTTPS) på port 443, se vidare stycke 14.



Figur 21: B använder webmailfunktionen på sin mailserver.

Observera att själva meddelandet aldrig förs över till dator B utan man använder ett webbgränssnitt för att titta på det medans det ligger på servern. Med webmail har man i regel inte heller någon möjlighet att spara ner meddelanden på datorn, men väl bilagda filer.

7.2 Frågor

7.1 Beskriv vägen för ett e-brev från en användare A till en användare B via A:s utgående mailserver och B:s inkommande mailserver. Det blir tre anslutningar. Ge exempel på vilka protokoll som kan användas vid dem. Vem är server respektive klient vid varje anslutning och vilka portar används.

7.2 Internetleverantörer spärrar normalt port 25 i brandväggen för alla sina kunder. Varför gör man det?

8 Programmering

Mål Vi ska lära oss att skriva interaktiva funktioner till våra hemsidor, och ta hand om html-forms.

Läs Introduction i http://se2.php.net/manual/en/.

Övningar Gör övningarna Ö9 till Ö14 i stycket 8.4.

Redovisning •Lägg upp de två filer ni har skapat på servern. Kontrollera att ni kan se dem i webbläsaren.

I och med veckans övningar är trestegsraketen där du ska ha kodat i HTML, CSS och PHP färdig. Resultatet är del i betygsunderlaget. Se till att dina filer minst uppfyller kravspecifikationen i uppgifterna. Sidorna som du skrivit och som genereras av skripten ska validera både HTML och CSS och inte minst viktigt är att alla filer är prydligt uppställda med god logisk struktur, väl kommenterade och indenterade.

8.1 Kod interpreterad av webbservern eller CGI-skript?

För att göra något mer än helt statiska sidor med HTML-kod på webbsidor, alltså något som kan ändras och interager med användaren på något vis måste det finnas något program som tolkar och bestämmer vad som ska hända. Det finns två typer av sådana program:

- Antingen är det ett program som laddas över till datorn hos den som besöker sidan, det kallas client side processing, det vanligaste språket för detta är javascript. Då bäddas programkoden in i HTML-koden och det fungerar bara om webläsaren som användaren har förstår programkoden och kan köra den.
- Alternativet är att datorn som kör webbservern också kör programmet. I det fallet måste programmet generera en komplett HTML-kod (eller CSS-kod, eller båda). Detta kallas server side processing och kan i sin tur delas upp på två olika typer:
  - Antingen laddar man in en interpretator i webbservern som en modul. Då tolkas programkoden i den modulen och översätts till HTML-kod. Det är ett ganska snabbt och smidigt sätt att programmera. I vissa fall som t. ex. PHP kan man skriva HTML-kod som vanligt och bädda in snuttar av PHP-kod mitt i där det behövs.
  - Alternativet är att webbservern anropar ett annat program som genrerar HTML-kod. Det kallas CGI-script Common Gateway Interface (CGI) är ett standardiserat sätt som webbläsaren skickar vidare information till programmen som körs. Det kallas CGI-script och är ofta skript i något interpreterande språk som python eller Perl men det kan också vara färdigkompilerad kod skriven i ett generellt språk som t. ex. C.

Vilken metod man än väljer så finns det viktiga säkerhetsaspekter att tänka på. Det är framför allt den maskin som kör programkoden som kan drabbas. Client side processing med javascript bör man alltså hålla sig lite sunt skeptisk till och kan om man är på dukla delar av webben vara program som letar efter säkerhetsluckor på ens dator för att göra intrång. När man sätter upp en server med server side processing har man istället att se till först och främst att webbservern inte tillåts att köra program hur som helst på dator utan har väldigt strikt reglerat vilka program som körs. Det görs på samma sätt som för HTML-sidorna genom att man definierar någon eller några speciella mappar där man lägger de program som ska få köras. En vanlig typ av attacker mot en webbserver är angriparen försöker gissa hur koden som ska tolka det som matas in ser ut och matar

in programkod i det språk som används som ska få programmet att göra något annat än vad som var tänkt.

8.2 Vanliga UNIX-kommandon

En server har ofta inte något grafiskt gränssnitt alls eller är inte lätt tillgänglig genom ett grafiskt gränssnitt. De normala sätten att hantera serverns inställningar är genom något webgränssnitt eller en textterminal. För att hantera filerna behöver man alltså kunna några grundläggande kommandon för UNIX-system. Se tabell 1.

8.3 Editera filer på servern

Ett av problemen med att programmera för webben är att när man programmerar vill man hela tiden testköra programmen för att se att det fungerar. Om man som i fallet med server side processing är beroende av att webbservern kör programmet finns det fyra strategier att välja på:

- Logga in till servern med en textterminal och editera koden direkt på servern med en texteditor som kan köras i terminalen.
- Montera filsystemet eller en katalog från servern på den lokala datorn så att man kan editera dem lokalt men spara på servern.
- Föra över filerna med t. ex. scp varje gång.
- Installera webbserver lokalt för att provköra.

8.3.1 Editera i textterminal

På vår server finns åtminstone texteditorerna nano, emacs och vi installerade som alla går att köra i textterminalen. Av dessa är nano enklast och fungerar ganska intuitivt men ger inte mycket hjälp, emacs är en mycket avancerad editor som är lite egensinnig i användandet och vi är en mycket egensinnig editor som är svår att lära sig men väldigt effektiv och de som tagit sig över tröskeln brukar inte överge den.

Observera att textterminaler normalt inte har stöd för mus så alla kommandon måste normalt göras med tangentbordet.

Nano Nano startas med kommandot `nano` *filnamn*. Man flyttar runt markören med piltangenterna, sparar med Ctrl-o (alltså trycker ner Control-tangenten och o samtidigt) och stänger programmet med Ctrl-x andra kommandon står listade längst ner i terminalen.

Emacs Emacs startas med kommandot `emacs` *filnamn*. Kommandon är generellt ganska långa i emacs. Man kan enkelt hoppa runt i texten med piltangetnerna. Spara en fil gör man med Ctrl-x Ctrl-s (alltså trycker först control och x samtidigt och sen control och s.) Öppna en till fil gör man med Ctrl-x Ctrl-f och filnamnet. Spara under ett nytt filnamn Ctrl-x Ctrl-w. Stänga programmet gör man mend Ctrl-x Ctrl-c. Om man börjar med ett kommando och ångrar sig kan man trycka Ctrl-g. Ångra gör man med Ctrl-_ (Ctrl-shift-- på svenskt tangentbord).

Det kan vara praktiskt att ha se två filer man har öppna samtidigt. Då kan man dela på fönstret med Ctrl-x 2 (alltså först control och k samtidigt och så släppa upp control innan man slår 2). Flytta markören mellan halvorna gör man med Ctrl-x o och man kommer tillbaks till helvisning av ett fönster med Ctrl-x 1. I avsaknad av mus kan man klippa ut text genom att ställa markören där man vill börja markeringen och trycka Ctrl-Space och sen klistra in den igen med Ctrl-y.

Man kan få mycket hjälp från emacs genom att den protesterar över vissa fel man skriver, om man har en terminal som stödjer färger visar den olika saker i olika färger så det är lättare att se strukturen i filen etc. Vad som visas beror på vilken slags kod man skriverman säger att editorn har olika moder för olika slags kod. För att det ska fungera måste den förstås veta vilken slags kod man skriver. Normalt gissar den det från filändelsen men om det blir fel kan man byta mod med `Meta`-`x` och sen namnet på den mode man vill ha t. ex. html-mode eller php-mode. På vanliga PC-tangentbord finns ingen tangent som det står meta på. Ofta är det windows-knappen eller `Alt` som fungerar som meta. Om man inte hittar den kan man trycka ner och släppa upp `Esc` så blir kommandot alltså `Esc` `x` *mode-namn*.

En rad i en textfil är allt som står mellan ett nyradstecken (newline) och nästa. Den kan vara hur lång som helst. Emacs visar normalt texten som den är med rader som är för långa för att få plats i fönstret fortsättandes på nästa rad. Det markeras med ett \ i slutet på raden i fönstret. Eftersom det är ganska jobbigt att läsa sån text kan man få emacs att sätta in nyradstecken (samma som man får genom att trycka på returtangenten) i raderna så att de blir lagom långa med kommandot `Meta`-`q` (eller `Esc` `q`). Beroende på vilken mode man har kan den fixa med indentering och en del specialformatering samtidigt.

Vi (VIsual editor) som är officiell del av POSIX-standarden är de verkliga nerdarnas favorit, fullständigt obegriplig för dödliga. Det finns bara en sak man behöver lära sig (man kan tyvärr råka ut för system där VI är den förvalda editorn och man plötsligt sitter med någon viktig systemfil öppnad i VI) och det är det avsluta: `Esc` `Esc` `:` `q` `!`. Om man börjar skriva på tangentbordet är stor risk att man har gjort massor med med ändringar och sparat dem till filen innan man vet ordet av.

8.4 Övningar

Ö9 Gör en googlesökning efter 'foo' och titta i adressraden i webbläsaren. Studera vad som kommer efter frågetecknet i addressraden. Det här är ett exempel på hur det du skrev in i sökformuläret skickas vidare till sökmotorn.

Ö10 Logga in till servern med SSH och använd en textbaserad editor enligt nedan och prova att göra små ändringar i filen direkt på servern.

Beroende på vad du gjort för installation för SCP i övning Ö4 är det olika program för att koppla upp en SSH-session. För installationen av WinSCP medföljer programmet PuTTY, kör det. I OpenSSH är det ett kommando `ssh` i terminalen.

Ö11 Gå till http://www.w3schools.com/php/php_syntax.asp, läs och stega dig fram till 'variables'. Ändra namnet på filen ni gjorde i övning Ö1 till `index.php`. Det kan göras genom att logga in på servern med ssh och ge kommandona:

```
$ cd ~/public_html
$ mv index.html index.php
```

eller anslut med scp och byt namn på filen.

Börja att titta på sidan och se att ni fortfarande ser samma sida. Gör ett kodstycke PHP i sidan där ni lagrar en textsträng i en variabel och sedan skriver ut den med kommandot `echo` i filen `index.php`.

Ö12 Stega er vidare och läs i lektionen på w3schools till 'If...Else'. Lägg till ett nytt PHP-kodstycke där ni gör en ny variabel `$foo` med ett numeriskt värde. Gör en if-sats som skriver ut "värdet är mindre än tio" om värdet på variabeln är mindre än 10. Prova

med att ändra värdet på variabeln i koden om de två fallen fungerar. Komplettera med en else-sats som skriver ut ”värdet är större än 10”. Nu blir det ju tokigt om ni ändrar till just 10 så skriv slutligen in en elseif-sats för specialfallet när variabeln har värdet 10.

Ö13 Ändra namnet på variabeln i if- och elseif-satserna från `$foo` till `$_GET['foo']`. När ni nu ska titta på filen i webbläsaren skriver ni till `?foo=10` efter filnamnet. Prova att ändra värdet i webbläsaren.

Ö14 Gå igenom 'A simple tutorial' på http://se2.php.net/manual/en//tutorial.php. Skriv två sidor där den ena är en ren HTLM-sida. Den ska ha ett formulär som innehåller två inmatningsrutor och länkar till den andra sidan. Den andra sidan ska vara en sida med php-skript i som skriver ut det som man matar in i formuläret.

Fortsätt sedan att experimentera och titta vad man kan göra. Städa upp och snygga till både HTML, CSS och PHP i filerna så det är strukturerat och validerar både HTML och CSS. Alla filer ska vara uppladdade på ditt konto på servern.

9 Versionskontroll

Läs Nedanstående

Övningar Ö15 Du ska ingående i en grupp om 3-4 personer skriva en essä enligt instruktioner i distansuppgiften. Den ska skrivas gemensamt i en fil som skall ligga under versionskontroll på elev.rymdgymnasiet.com. Repositorien finns att checka ut via svn från. Servern kommer att vara öppen för inloggning till fredag 26/10 för fullbordande av uppgiften.

9.1 Teori

Versionskontroll handlar om att man sparar alla ändringar som görs till en fil. Det görs genom att lagra filen en gång och sen alla ändringar som görs till den. Det ger ett system där man kan backa tillbaks och se vilka ändringar man själv eller någon annan har gjort till filen och ta fram gamla versioner om man gjort några fel i den. Det ställe där alla ändringar ligger är en slags databas över ändringar som kallas repositorie. Versionskontrollsystem är ett näst intill oundgängligt verktyg i alla samarbetsprojekt.

Det finns några olika upplägg på hur det fungerar. Först och främst är det skillnad på var repositorien finns. Det är två olika typer:

Centraliserade system Här håller man repositorien på en server och alla som ska jobba med data tar ut en kopia av en version (oftast den senaste) som finns i repositorien, det kallas att checka ut filen. När man har gjort ändringar kopierar man tillbaks dem till servern, det kallas att checka in ändringarna.

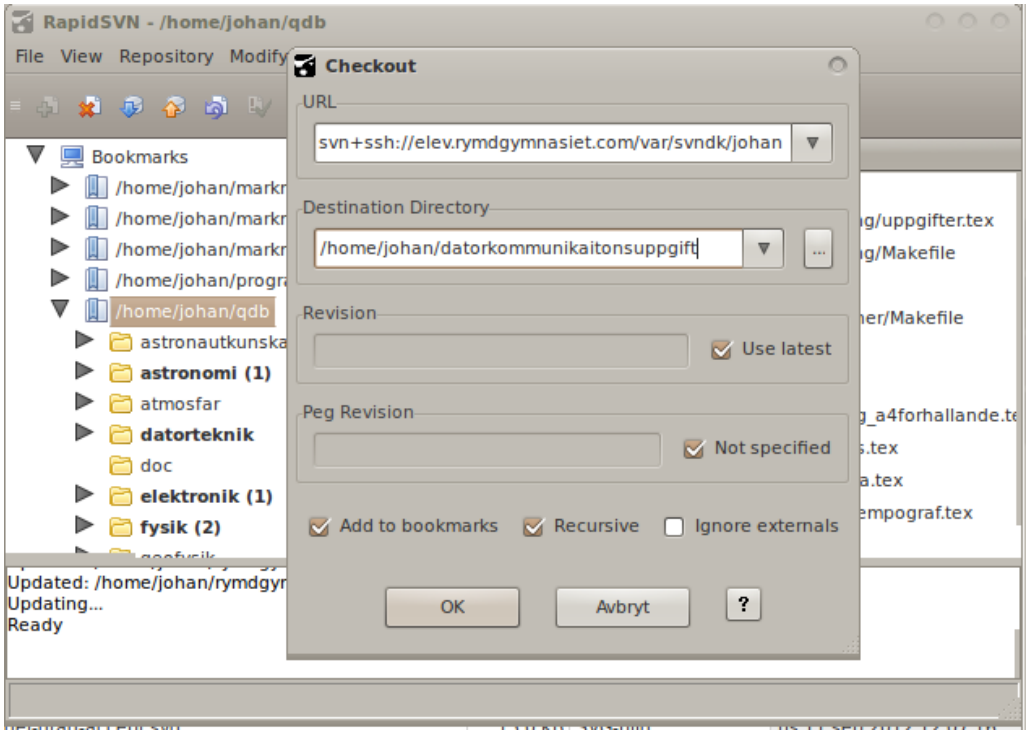
Till centraliserade system finns två underkategorier för hur man ska göra om användare vill göra ändringar samtidigt.

- Det enklaste och idag väldigt omoderna sättet är att låsa filen när man checkar ut den. Det betyder att ingen mer kan checka ut filen innan man har checkat in den igen.
- Det något mer komplicerade sättet är att man bakar ihop filerna om det finns olika ändringar i den, det kallas merge (sammanslagning). Exempel på sådana system är Concurrent Versions System (CVS) och Subversion (SVN). Det fungerar enkelt för system som bygger på textfiler t. ex. HTML eftersom det är lätt att hålla reda på vad som ändrats. För filer som inte är baserade på text t. ex. ordbehandlarformat är det svårare. Det kräver nästan att det finns någon funktion i programmet som hanterar det filformatet för att jämföra innehållet i filer.

Distribuerade system Här finns det komplett information om alla ändringar på alla datorer som använder det. Om man vill få ut filer för att jobba med dem kopierar man alltså hela repositorien. Det betyder att det inte behöver finnas någon speciell server utan man kopierar från varandra. Det ger hög säkerhet för att det finns många kopior och det gör att man kan jobba och skicka in ändringar till repositorien utan att ha kontakt med någon server. Filerna som behöver kopieras är större än de som man ska använda. Exempel på distribuerade versionskontrollsystem är det öppna Git och kommersiella Bitkeeper. I distribuerade system använder man alltid merge för att sammanföra ändringar.

9.2 Subversion

För övningen i den här kursen har jag valt att använda Subversion (SVN). Det är ett centraliserat system med en server. På servern ligger repositorien. Jag har valt att för enkelhets skull endast göra en repositorie. En repositorie kan innehålla ett i princip hur



Figur 22: Utcheckning av en ny arbetskopia med rapidsvn.

stort filsystem som helst med katalogstruktur och länkar. För uppgiften har jag gjort en katalog för varje grupp i repositorien. Det finns ingen teknisk begränsning om att ändra i varandras filer med det upplägget men en av de saker som systemet är till för att hålla reda på är vilken användare som gjort ändringarna så var och en får stå för sina handlingar. Utöver mina förklaringar här finns det en bra manual man kan slå i på nätet. Pilato et. al. http://svnbook.red-bean.com.

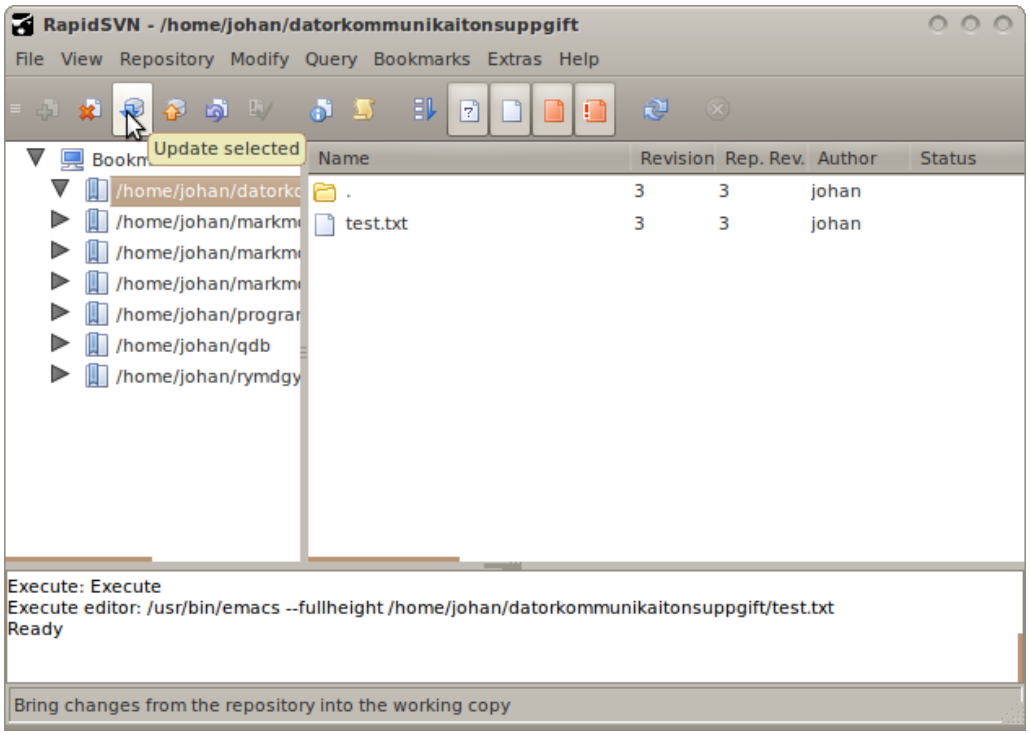
Det första man behöver göra för att redigera filer i ett versionskontrollsystem är att checka ut den aktuella versionen. Då får man lokal kopia, en så kallad arbetskopia. Arbetskopian är en eller flera helt vanliga filer. De filerna arbetar man med precis som vanligt i de program som kan hantera den typen av filer. Det är bara i en sån arbetskopia man kan göra ändringar och det ända sättet att få dem in till repositorien är att checka in ändringarna man har gjort i sin arbetskopia till repositorien.

En av idéerna med ett versionskontrollsystem är att säkra alla ändringar så att det inte ska försvinna någon information. Det är därför enkelt att ta bort en fil i den nuvarande versionen men den finns då kvar i alla sina tidigare versioner.

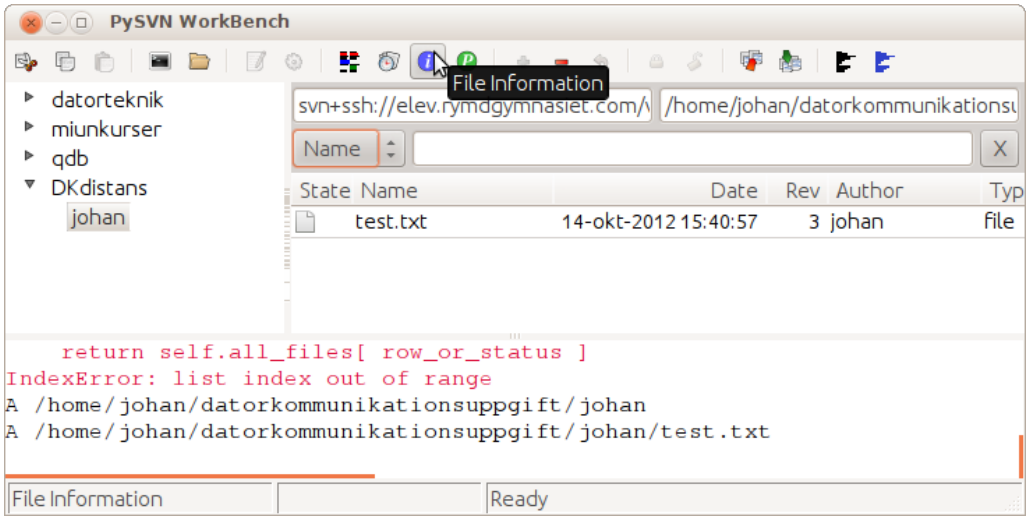
För att ta med en ny fil till sin arbetskopia måste man tala om för svn att den ska tas med i arbetskopian.

Hanteringen av filer kan göras med kommandon i terminalen eller med ett program med Graphical User Interface (GUI). Figur 22 visar hur utcheckning av en ny arbetskopia kan se ut i det grafiska svn-programmet RapidSVN. När man väl checkat ut sin arbetskopia ser det ut som i figur 23. PySVN är ett annat exempel på grafisk svnmiljö. Ett exempel på hur det kan se ut återfinnes i figur 24.

För Windows finns TortoiseSVN, http://tortoisesvn.net. Det är integrerat i windows shell så det finns ingen särskilt program att starta utan om man vill lägga till saker kan man högerklicka på filer eller kataloger i filhanteraren och om man lyckats med installationen ska det finnas en undermeny som heter TortoiseSVN med val för olika saker man vill göra med filerna.



Figur 23: Utcheckning av en ny arbetskopia med rapidsvn.



Figur 24: Användarinterfacet i PySVN.

9.2.1 Arbetsgång

Det är endast första gången man som man ska arbeta med en viss repositorie som man behöver checka ut en arbetskopia. Det kan man göra genom att följa instruktionerna för en grafisk svn miljö om man har en sådan. Annars genom att gå till den mapp där man vill ha filerna och ge kommandot

```
$ svn checkout svn+ssh://elev.rymdgymnasiet.com/var/svndk/gruppnamn
```

Därefter är det en enkel ordning i tre steg:

1. Uppdatera arbetskopian.

   ```
   $ svn update
   ```

2. Gör dina ändringar

3. Checka in ändringarna.

   ```
   $ svn commit -m "kommentar"
   ```

9.2.2 Konflikter

Om man har gjort ändringar i samma ställe i en textfil som någon annan som har checkat in sina ändringar sedan man checkade ut sin arbetskopia kommer man att råka ut för en konflikt om man försöker checka in den. Det gör att subversion kommer att protestera och incheckningen avbryts. Istället får man tre versioner till av den fil där konflikten finns i sin arbetskopia. Det är den ursprungliga som man checkade ut, den som innehåller de ändringar man gjort och den som checkats in senast. Det man måste göra är att kontrollera hur de skiljer sig i från varandra och införa de ändringar som ska göras i den fil som har det ursprungliga filnamnet. När man är färdig med det måste man tala om för SVN att man har rett ut konflikten annars kommer man inte få checka in sin version. Det gör man med kommandot

```
$ svn resolve -accept working filnamn
```

En djupare förklaring till konfliktlösning finns i Pilato (http://svnbook.red-bean.com/en/1.7/svn-book.html#svn.tour.cycle.resolve).

10 Distansuppgift

Mål Distansuppgiften har två syften:

1. Att prova på att delta i ett sammarbetsprojekt över internet.
2. Att fördjupa sig i ett specialområde av datorkommunikation.

Uppgift Välj ett ämne att studera. Det skall vara ett specialområde för datorkommunikation, existerande, tilltänkt eller historiskt. Gärna med rymdanknytning. Se förslag på uppgifter nedan eller kom fram med egna.

Utförande Arbetet ska utföras i grupper om 3 till 4 personer *som inte befinner sig på samma ort* under distansveckan. Det ska mynna ut i en text som har utarbetats tillsammans till och med fredagen 26 oktober.

Arbetsmaterialet ska ligga på en server under versionskontroll där alla har var sin inloggning så det går enkelt att se vilka som har bidragit med uppdateringar.

För diskussioner kring arbetet med texten ska ni själva hitta en lämplig kommunikationskanal på internet. Det kan vara en Facebookgrupp, diskussionsgrupp ett chatrum eller liknande som passar er att jobba i. Läraren ska ges en inbjudan till denna diskussionskanal och kunna följa diskussionen.

Redovisning Betygsunderlag från uppgiften är aktivitet i diskussionsforumet om uppgiften, redigeringar på texten och innehållet i texten. Läraren kommer att checka ut den version som var gällande fredag 26 oktober kl. 23:59.

10.1 Förslag på rymdrelaterade uppgifter

Mars Telecommunications Orbiter Tänkt telekommunikationssatellit som skulle ta internet till Mars och hjälpa alla vetenskapliga marsfarkoster att sända hem data.

Luna Första Ryska satelliten som tog bilder på Månens baksida. Men hur kom de till Jorden?

Delay Tolerant Networking Teknik för nätverkstrafik när man inte kan få snabba svar och använda hanskakningsprocesser. Tänkt för rymdsonder och även testat för samer i Padjelanta.

Telekommunikationssatelliter Historisk tillbakablick eller beskrivning av hur det fungerar.

SETI Skulle vi kunna upptäcka signaler från andra tekniskt avancerade civilisationer?

Rover på Mars Hur kör man en radiostyrd bil med en timmes tidsfördröjning?

Radioreflektioner i jonosfären Radiosignaler kan i vissa fall studsa mellan jonosfären och jordytan ett helt varv runt jorden. När och hur?

GPS Hur fungerar de?

11 Kabelbundna anslutningar

Mål Efter veckans lektioner ska vi förstå skillnaden mellan olika kabeltyper, deras användning, känna igen vanliga kontakter och kunna kontaktera ethernetkabel.

Läs Nedanstående

Besvara
- Räkna talen i utdelat häfte om binära och hexadecimala tal.
- Instuderingsfrågorna i stycke 11.10.

Övning Gör övningarna i stycket 11.9. Endast en fungerande kabel behöver redovisas.

11.1 Bits och bytes

När man talar om överföringshastigheter och lagringskapacitet finns två begrepp som ofta förväxlas, bits och bytes.

En **bit**, beteckas med gement b, är den minsta delen information som lagras det är en logisk etta eller nolla, alltså en siffra i ett binärt tal. Den kan manifesteras på många olika sätt; spänning eller inte spänning på en ledare, en hög eller låg ton i frekvensmodulering, hög eller låg amplitud i amplitudmodulering, grop eller inte grop i en optisk skiva, magnetfält åt ena eller andra hållet på en magnetisk disk osv. Alla moderna datorer är uppbyggda kring att alla data reprenteras med två tillstånd och det är det som säger att våra datorer är binära.

Alfanumeriska tecken och även andra icke synliga styrtecken brukar lagras som åtta bitar, det är vad som kallas en **byte**, betecknas med versalt B, med moderna teckenkodningar kan man ibland använda flera byte för ett tecken. En byte kan också tolkas som ett tal $0–2^8 - 1$ (0–255) och tal som används för beräkningar i datorer lagras också som ett jämt antal bytes olika många beroende på hur stora tal som kan behöva skrivas eller om det är decimaltal, hur noggranna de behöver vara.

11.1.1 Prefix

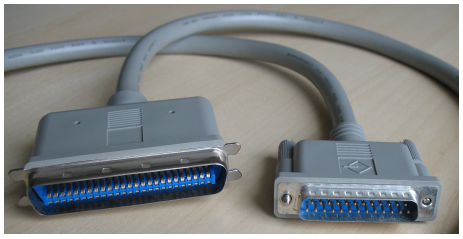
Eftersom de allra flesta tillämpningar av datorer naturligt handlar om multipler av två är det opraktisk att använda de vanliga prefixen kilo (k, 1000), mega (M 1000^2), giga (G, 1000^3) etc, så i datorsammanhang har man i stället ofta de binära prefixen kibi (Ki, $1024 = 2^{10}$), mebi (Mi, $1024^2 = 2^{20}$) osv, som är mer logiska. Tyvärr används dock både i tal och skrift ofta orden och beteckningarna för de digitala prefixen när det egentligen är de digitala motsvarigheterna som avses. Detta utnyttjas i sin tur ofta av minnestillverkare, bredbandsleverantörer etc för att många tolkar mega som att det ska betyda mebi men legalt är det förstås inte så. T. ex. en hårddisk på 1 TB man köper kan man nog vara ganska säker på att den är just 1 TB och inte 1 TiB som man skulle gissa eftersom en en terabyte är 10 % mindre än en tebibyte. På samma sätt skrivs ofta överföringskapacitet i avtal från bredbandsleverantörer i Mb/s istället för Mib/s för att det ger leverantören marginalen på sin sida.

11.2 Kabeltyper

Kommunikationskablar för datatrafik kan dels delas upp efter hur många signaledare de har.

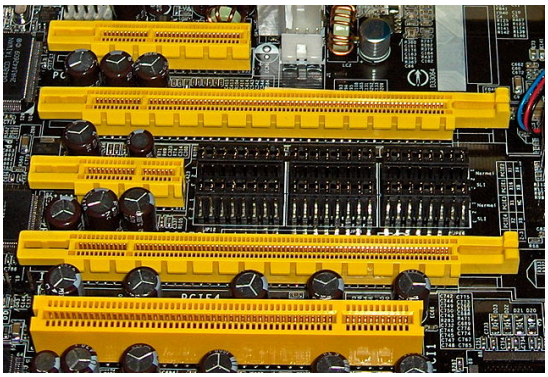
Seriell kabel har ett par ledare som överför signaler. Databitar sänds då i en ström efter varandra i samma ledare.

Parallell kabel har flera par ledare som samtidigt överför signaler ofta åtta par. Då kan en byte data sändas samtidigt.



Figur 25: Parallellkabel för SCSI med olika kontaktdon Foto: Wikipedia fr användare Quark67 Från Wikimedia Commons.

Oavsett typen har oftast kabeln fler ledare. De används för att styra överföringen på olika sätt eller för strömförsörjning. Det vanligaste är att det finns ett par ledare som håller takten för överföringen, alltså hoppar mellan två olika spänningar varje gång som det är dags att läsa av spänningen på den eller de kablar som har själva datatrafiken. Parallell kommunikation kan ge högre överföringskapacitet på korta avstånd. Innuti en dator används mycket parallella överföringar, mellan processorn och minneskretsarna är det normalt hundratals parallella kanaler, och till andra vitala chip och insticksskort går det också parallella anslutningar. I figur 26a syns slottar (kontakter) för insticksskort av typerna PCI och PCI express och i figur 26b ser man hur ett chip är anslutet med många ledningsbanor på moderkortet. Tidigare var det också vanligast att koppla till hårddiskar, CD-spelare, skrivare, band-



(a) PCI och PCI express-slotar. Foto: Wikipedia en användare snickerdo Från Wikimedia Commons.

(b) Nordbrygga Intel 810. Foto: Jonathan Zander Från Wikimedia Commons.

Figur 26: Parallellanslutningar i en dator.

stationer och liknande med parallella anslutningara som Parallel Advanced Technology Attachment (PATA) och Small Computer System Interface (SCSI) men i takt med att man ökar hastigheten på signalerna blir problemet att ledarna som ligger brevid varandra i kabeln stör varandra större så idag ersätts dessa med seriella anslutningar som Seriall Advanced Technology Attachment (SATA), Universal Serial Bus (USB) och firewire.

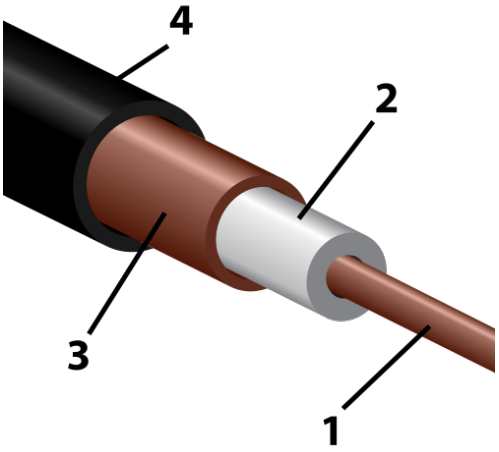
Beroende på hur ledarna i kabeln är ordnade är de också uppdelade på

Telefonkabel är den enklaste typen av kabel. Den har bara ett par ledare och de är inte skyddade mot störningar på något speciellt sätt. De är från början inte alls avsedda för datakommunikation men används mycket för att de redan finns dragna till varje hus. De är ofta den begränsande faktorn för hur snabb uppkopplingen blir till ett hem.

Partvinnad kabel är den idag vanligaste typen. Det billigaste och enklaste sättet att skydda kabeln mot störningar. Man tvinnar varje par av ledare kring varandra inuti kabeln. Störningar kommer genom att elektriska och magnetiska fält från omgivningen ger strömpulser i kablarna. Om ledarna är tvinnade kring varandra påverkas de ungefär lika av störningen.

Koaxial kabel tar detta ett steg längre. Den ena ledaren i ett par är en tråd och den andra läggs som en tub runt om den. De har då samma centrum och påverkas lika av

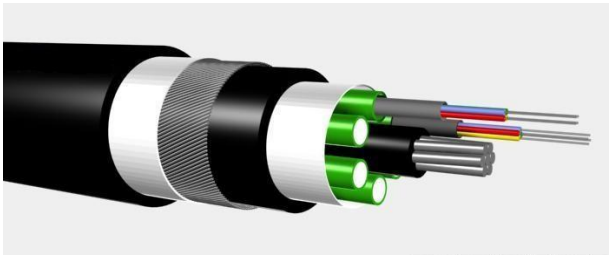
magnetfält och den yttre ledaren som omsluter den inre fungerar som en faradaybur för elektriska störningar. Koaxialkablar är dyrare att tillverka än partvinnade kablar och används allt mindre.



(a) Koaxialkabel. 1 signalledare, 2 isolering, 3 skärm, 4 isolering. Ill.: Wikimedia commons användare Tkgd2007 Från Wikimedia Commons.

(b) 10base2 en koaxial nätverkskabel för datornätverk. Foto: Wikipedia en användare Dflock Från Wikimedia Commons.

Figur 27: Koaxialkabel i princip och praktiken.



Figur 28: Optiska fiberkablar för stor bandbredd har många fibrer i samma kabel. Ill.: Wikimedia commons användare Srleffler Från Wikimedia Commons.

Fiberoptisk kabel bygger inte alls på elektriska signaler utan är långa glasfibrer som man sänder laserpulser genom. Det ger mycket högre överföringskapacietet men är dyrare att tillverka, kommunikationsutrustningen med lasrar i ändarna blir också dyrare. Kablarna kan inte heller böjas för kraftigt och är därför praktiskt svårare att dra inne i hus. Fiberoptisk kabel används främst för stomnätet, alltså mellan kopplingsstationer och serverhallar.

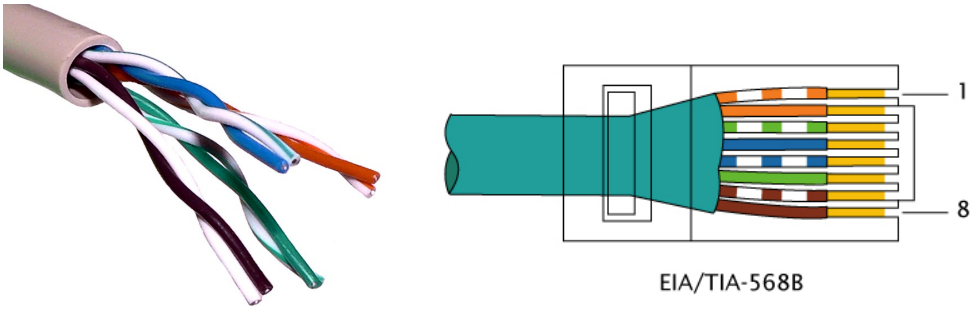
11.3 Partvinnad kabel

Eftersom det är avsevärt billigare att producera partvinnad kabel än koaxial eller fiberoptisk har man lagt ner mycket energi på att få fram sådana kablar som klarar mycket datatrafik. Grundiden med partvinnad kabel är att genom att tvinna de två ledarna som används för att skicka en signal kring varandra ska externa magnetfält som alltid inducerar strömmar mellan ledare ta ut varandra för att det blir lika stor area av loopar åt vardera hållet.

Detta allena hjälper mot externa störningar men om alla paren är likadant tvinnade kommer signalerna i de olika paren att interfera med varandra eftersom de ligger lika och signalerna är ju små strömmar som ger upphov till magnetfält. Sådan påverkan mellan ledarna i en kabel kallas överhörning (Cross Talk). För att undvika det tvinnas de olika paren olika mycket. I den vanligaste kabeln för Twisted pair ethernet med 8 ledare brukar den tätast tvinnade vara kring 1,5 cm per varv och den lösast tvinnade kring 2 cm per varv.

Kablarna och kontakterna finns i olika varianter för att klara olika mycket störningar och överföringskapacitet. Unshielded Twisted Pair (UTP) har ingen särskild skärming. Screened Twisted Pair (ScTP) eller Foil Twisted Pair (FTP) har en strumpa eller folie av ett ledande material runt hela bunten av ledare som jordas för att minska störningarna. Shielded Twisted Pair (STP) har en strumpa runt varje par av ledare som jordas det minskar störningarna både från omgivningen och mellan trådparen. Kombinationen av ScTP och STP alltså de som har skärmning både kring varje par och runt knippet kallas S-STP eller S/FTP

11.4 Twisted pair ethernet



(a) UTP-kabel Foto: Wikimedia commons användare Baran Ivo Från Wikimedia Commons.

(b) RJ45 koppling i ordning B Ill.: Wikipedia it användare Lp Från Wikimedia Commons.

Figur 29

Ethernet kopplas med partvinnad kabel. Det behövs två ledarpar för en anslutning men kablarna innehåller normalt 8 ledare så att man ska kunna använda den till två anslutningar. Ethernetkabel är den man oftast står inför att kontaktera (sätta kontakter i ändarna på) själv eftersom man ofta ska dra den i ett hus och köper kabeln på metervara. Det är viktigt att man kopplar paren rätt så att kabeln verkligen fungerar som en partvinnad kabel annars kommer den fungera dåligt. Det finns överenskomna färger på ledarna och en överrenskommen ordning för hur de ska kopplas också så att man ska kunna kontaktera om ena änden utan att behöva leta reda på den andra. Det finns två olika sätt att koppla ledarna, rakt eller korskopplat. Rak koppling är den vanligaste som man använder när man kopplar t. ex. en dator till en hub, switch eller router. Korskopplad kabel används när man kopplar ihop två likadana enheter, t. ex. direkt mellan två datorer.

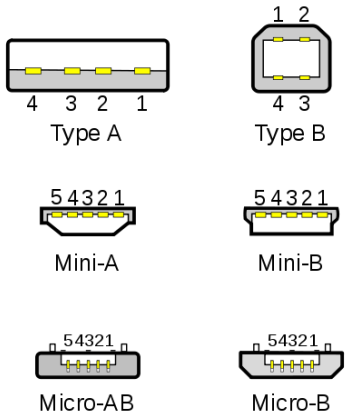
Kontakterna som används på kabeln kallas RJ45 och kläms på med en speciell tång som i ett handgrepp både gör anslutningen för alla ledarna och klämmer fast kabelhöljet i kontakten. Kablar man köper färdiga har kontakterna fastgjutna med gummi så de är tåligare.

Kablarna är klassificerade i kategorier beroende på hur stor överföringshastighet de klarar av (egentligen frekvensbandbredd). Den första vanliga kabeln var kategori 3 som ska klara 16 MHz bandbredd och var den vanligaste kabeln för att bygga upp Local

Area Network (LAN) på 90-talet och klarar 10BaseT alltså TP ethernet 10 MBit/s. Idag används mest kategori 5 (100 MHz) eller 6 (250 MHz) som klarar 1000BaseT, state of the art är kategori 7a (1 GHz) som klarar 10GBaseT (10 GBit/s). Värt att notera är att klassificeringen inte ställer krav på vilken typ av skärmning som används för att uppnå bandbredden. Kategori 6 kablar är fortfarande oftast UTP medans kategori 7 kablar oftast är S-STP.

11.5 USB

USB är den idag vanligaste standarden för att koppla kringutrustning för datorn. Det är en buss som måste styras av en värd som dirigerar trafiken i bussen men sen kan man använda sk USB-hubbar för att grena ut och sätta många enheter på samma buss. Eftersom det är en enhet som ska dirigera trafiken han man gjort kablar med anlutningarna olika i ändarna så att man bara ska kunna koppla in en värdenhet på bussen. Värden har en kontakt av typ A och alla andra enheter, om de kopplas in med kabel har uttag av typ B. Standarden har uppdaterats till högre hastighet vid ett par tillfällen men kontakterna har behållits, vissa äldre kablar kan dock ge problem på högre överföringshastigheter. Konktakterna har kommit i två mindre varianter Mini och Micro som är vanliga på utrustning som är fysiskt mindre, som kameror och telefoner. Mini USB har en A och en B variant precis som ursprungskopplingarna medans Micro har en B variant för utrustning som inte kan vara värd för bussen och en AB för de som kan det. Den som har AB-kontakt ska då själv kunna ta reda på om den kommunicerar med en annan värd eller ska agera värd själv.



Figur 30: USB-kontakter Ill.: Bruno Duyé Wikimedia commons användare Darx Från Wikimedia Commons.

11.6 Dataöverföring och strömförsörjning

En utveckning är att dataanlutningarna förses med en möjlighet att ge strömförsörjning över samma anslutning. Anledningen är förstås praktisk, att man ska få en sladd mindre i systemet. USB kan överföra lite drivström och fungerar bara med små saker men Power over ethernet är en teknik att driva anslutna större enheter det finns flera olika standarder med kraft upp till 25 W, det används framför allt till olika typer av nätverksutrustning som routrar, WiFi accesspunkter och IP-telefoner.

Också det motsatta arbetssättet är möjligt, alltså att skicka signaler över anslutningar som egentligen ar byggda för strömförsörjning. Homeplug är ett system där man kan utnytta elsystemet i ett hus för dataöverföring. Man ansluter då kopplingsdosor i eluttagen och kan få både nätsignal och ström ur uttaget. Det finns även Digital Subscriber Line (DSL)-lösningar som går över elnätet istället för över telefonnätet.

11.7 Video

Överföring av bild till från en dator, videospelare eller TV-mottagarbox sker med olika standarder. Traditionellt har man använt koaxialkabel till TV, senare bl. a. Scart som också överför styrsignaler mellan enheterna. För datorer har Video Graphics Array (VGA) varit förhärskande som bildskärmsanslutning i många år. Det har kommit fram några alternativformat som Digital Visual Interface (DVI). Eftersom problemet egentligen är det samma och eftersom gränsen mellan TV apparater och datormonitorer suddas ut används idag sammas standarder. Vanlig anslutning nu för både datorer och hemmabioutrustning är High-Definition Multimedia Interface (HDMI). Underhållningsbranchen driver hårt på

för att ta fram standarder för videoöverföring som är krypterade för att man inte ska kunna kringå kopieringsskydd, Digital Rights Management (DRM), genom att dumpa videoströmmen till skärmen. HDMI är en sådan som använder High bandwidth Digital Content Protection (HDCP), DVI finns i varianter som stöder HDCP.

Display Port (DP) är en standard som är tänkt att ersätta de andra och är den första bildskärmsanslutningen som bygger på paketteknik.

11.8 Kombinerade protokoll

Trenden nu går mot att kombinera överföringar av flera olika protokoll i samma anslutningar. T. ex. har Apple och Intel modifierat den tidigare videoporten DisplayPort till att både överföra videosignalen till skärmen och samtidigt ethernet och USB signaler. Alla nya Macar och Apples skärmar har denna nya standard Thunderbolt. Ett konkurerande system är HDBaseT som kan föra över 100BaseT ethernet, HD video och upp till 100 W strömförsörjning över kategori 5 eller 6 TP-kabel.

11.9 Övningar

•Ö16 Ni ska kontaktera en egen TP-kabel. Gör var sin ände på kabeln inom labbparet så båda får prova. Följ instruktionerna på http://www.kjell.com/fraga-kjell/hur-funkar-det/natverk/kabelinstallation/montering-av-rj45 noga. Testa kabeln med kabeltestare, märk upp den med namn och lämna till Johan.

Ö17 Leta reda på alla kabelanslutningar till din egen dator och ta reda på vad de är till.

11.10 Frågor

11.1 Beräkna hur mycket 1 GiB är uttryckt i

a) GB
b) MB
c) MiB
d) Gib
e) Gb

11.2 Beräkna hur mycket 2,3 GiB är uttryckt i

a) GB
b) MB
c) MiB
d) Gib
e) Gb

11.3 a) Vad skiljer en seriell och en parallell anslutning åt?
b) Vad är den mest avgörande faktorn för när man använder parallell eller seriell överföring?
c) Ge några exempel på parallella respektive seriella överföringsstandarder.
d) Gränsen mellan när man använder seriella respektive parallella anslutningar förskjuts hela tiden mot mer seriella anslutningar. Varför?

11.4 a) Beskriv varför partvinnad kabel är mindre känslig för störningar än kabel där ledarna ligger parallellt.
b) Hur förbättrar man partvinnad kabel för att klara ännu högre överföringskapacitet eller fungera i miljöer med mycket störningar?

11.5 Vad är det rent fysikaliskt som överförs vid signalöverföring i kabel med metalledare respektive fiberoptisk kabel?

11.6 USB-kontakter finns i en A och en B variant. Kablar har en av var sort i ändarna, varför har man det så?

11.7 Identifiera kontakterna i bilderna. Sök på nätet om du inte känner igen dem.

a) 

Foto: Wikimedia commons användare Macfan97 Från Wikimedia Commons.

b) 

Foto: Wikimedia commons användare Yoe Från Wikimedia Commons.

c) 

Foto: Wikimedia commons användare The High Fin Sperm Whale Från Wikimedia Commons.

d) 

Foto: Wikimedia commons användare Frankie1969 Från Wikimedia Commons.

12 Subnät

Mål Efter dagens övningar är det meningen att ni ska förstå vad ett LAN är och hur adresseringen av datorer inom LANet och mot omgivningen fungerar.

Läs

- http://www.lysator.liu.se/~kjell-e/tekla/linux/security/tcpip.html (http://tinyurl.com/arsrc5w) avsnitten "nätmask", "Paket och routing" inklusive exempel och "ICMP".
- http://www.lysator.liu.se/~kjell-e/tekla/linux/security/netservices.html (http://tinyurl.com/b3vf4yc) avsnitten "ICMP" inklusive "Ping", "Traceroute" och "Traceroute i Windows".
- Nedanstående

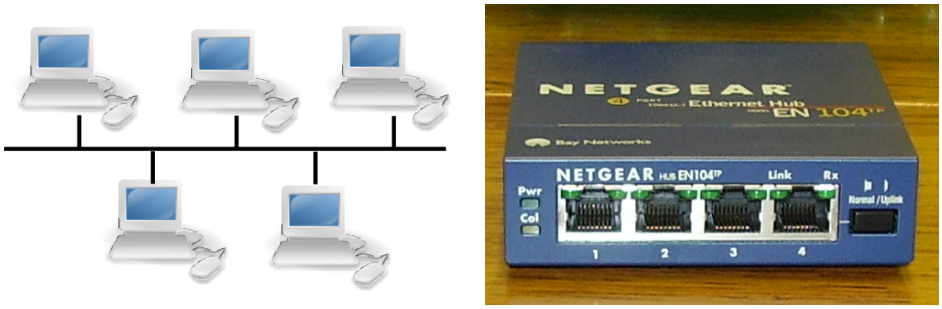
Övningar Gör övningarna i stycket 12.6.

Besvara Instuderingsfrågorna 12.7.

12.1 Topologi

Ett LAN är ett lokalt nätverk av sammankopplade datorer. Idag i praktiken alltid sammankopplade med IP. I princip kan datorerna i LANet kommunicera direkt med varandra om de bara har varsin unik adress, vilket går att skriva in manuellt För att styra upp nätverket brukar man ha en DHCP-server. Det är den dator som ger ut adresser till de andra datorerna på nätverket.

12.1.1 Bussform

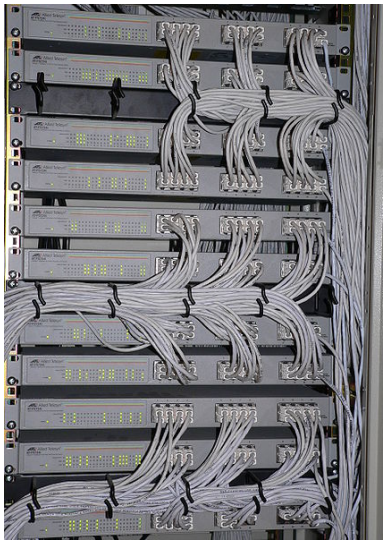


(a) Bussnät Ill.: Wikipedia gl användare Lmbuga Från Wikimedia Commons.

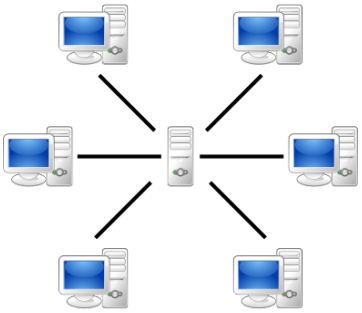
(b) Nätverkshubb. Foto: plugwash Från Wikimedia Commons.

Figur 31

Ett bussformigt nätverk betyder att alla datorer i nätet delar på en förbindelse som de alla måste samsas om. Den vanligaste bussen är ethernet där datorerna har var sitt nätverkskort och är kopplade till en hub där bussen sammankopplar alla ingångarna. Topologiskt ser det ut som figur 31a, men i praktiken eftersom många datorer kan vara anslutna till en hub som är inbyggd i en låda är det lätt att missta nätet för att vara av stjärnform. En **hub** kan ses som en grenkontakt för nätverkskablar, det är enhet med flera nätverkskontakter som i stort sett bara kopplar ihop dem så att alla kommer i kontakt med varandra och det bildas en buss. I tidigare implementationer av ethernet var det enklare att se att det var en buss eftersom datorerna kopplades samman med en koaxialkabel som gick vidare från dator till dator.



(a) Ett rack med switchar Foto: Wikimedia commons användare Parkis Från Wikimedia Commons.



(b) Stjärnformigt nät. Ill.: Wikimedia commons användare Mauro Bieg

Figur 32

12.1.2 Stjärnform

Om det är en **switch** som är anslutningspunkten har den en viss förmåga att dirigera trafiken så att paketen som kommer till switchen dirigeras endast till den fysiska anslutning där en dator med rätt adress var senast ansluten. Switchen är alltså intelligentare än en hub och läser adressen på paketen innan den skickar vidare. Det blir därför frågan om en buss på varje ingång till switchen. Då blir switchen istället centrum i ett stjärnformat nätverk. Eftersom den mesta trafiken går ut på bara en av de fysiska anslutningarna ger det mindre trafik i systemet och det blir effektivare. Det gäller särskilt nätet grenar upp sig många gånger så framför allt de kopplingspunkter som finns högre i hierarkin är nästan alltid switchar.

Också Trådlösa accesspunkter ger stjärnformade nät kring sig där samtliga datorer kommunicerar med accesspunkten men inte med varandra.

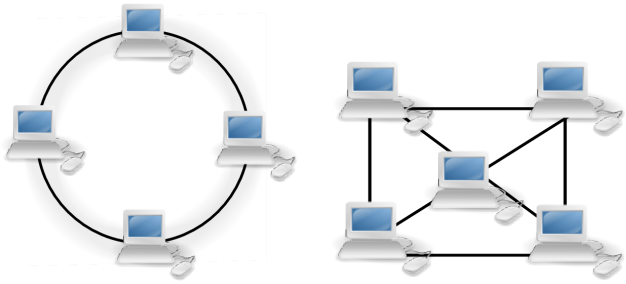
Stjärnformade nät är den äldsta typen för att de första anslutningarna till en dator var att man hade fjärrterminaler till en stor dator för att flera skulle kunna använda den samtidigt. När terminalerna byttes ut mot självständiga datorer var det ett defakto stjärnformat nätverk med direktanslutningar till huvuddatorn.

12.1.3 Ringform

Tidigare var ringformiga strukturer vanligt, där varje dator var kopplad till sina två närmsta grannar, se figur 33a systemet var känsligt för ett kabelbrott någonstans i ringen som gör att all kommunikation bryts.

12.1.4 Maskform

Det snabbaste sättet är förstås att varje dator i nätverket har direkt kontakt med alla andra. Det kallas maskformigt nätverk. På engelska Mesh, om alla noder är direkt sammankopplade med varandra kallas det True Mesh (figur 33b). Det är inte så vanligt. Superdatorer består idag i praktiken av en massa ordinära datorer som sitter sammankopplade med snabbt nätverk och delar på arbetet med en beräkning. De brukar då vara



(a) Ringformigt nät. Ill.: Wikipedia gl användare Lmbuga Från Wikimedia Commons.

(b) Maskformigt nät. Ill.: Wikipedia gl användare Lmbuga Från Wikimedia Commons.

Figur 33

kopplade i ett Mesh där datorerna står i en slags matris och varje nod är direkt kopplad till sina närmsta grannar alltså har 8 nätverkskort. Att bygga en superdator med True mesh blir oöverstigligt för att antalet noder är så stora.

För nätverk som eftersträvar större driftsäkerhet gör man en blandning av Mesh och stjärnstruktur med lite korskopplingar. Man kan också blanda Mesh och ringstruktur, t. ex Sunet är uppbyggt som en "snögubbestruktur" där huvudroutern på varje lärosäte är hopbunden med nästa router genom två kablar som går geografiskt olika rutter för att undvika driftstörningar om en kabel exempelvis grävs av. Det är en av de grundstenar som internet vilar på att det i större strukturer är maskformigt så att paket alltid har alternativa vägar att ta sig fram och att inte nätet ska vara beroende av en nod eller en anslutning för att fungera.

12.2 Routern

De små nätverken är sammankopplade till nästa större nätverk i hierarkin som alltså är ett nätverk av nätverk, ett sådant kallas för ett internet, med en router. Routerns uppgift är att dirigera trafiken mellan de två näten. Den avläser adressen i alla paket som kommer till den och avgör om det ska vidarebefordras till det andra nätverket eller inte. Routern kan vara en dator vilken som helst på nätverket som utrustats med anslutningar till de båda näten som ska kopplas ihop. I hemmiljö är routern ofta inbyggd i en switch om man har en separat switch för hemmabruk, i en trådlös accesspunkt eller i t. ex. i ett Asymmetric Digital Subscriber Line (ADSL)-modem. Routern får då två olika IP-nummer, ett i varje nätverk. Den punkt där de två nätverken kopplas ihop, alltså i praktiken vid routern kallas för gateway.

Figur 34: Router för hemmabruk sammanbyggd med en switch och en trådlös accesspunkt Foto: Wikimedia commons användare Evan-Amos Från Wikimedia Commons.

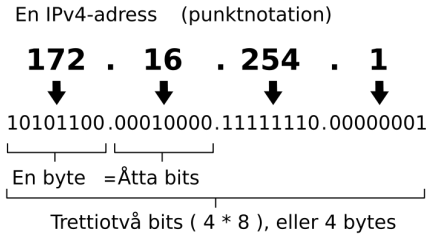
12.3 Adresser på Internet

Datorer anslutna till ett IP-nätverk har ett IP-nummer som är dess identitet på nätet. Det finns två versioner IP-nummer, det som i praktiken används i Sverige är version 4 som har 32 bitars längd men en övergång till version 6 med 128 bitars längd är dock att vänta inom de närmsta åren. IPv6 har börjat tas i bruk på vissa håll i världen. IPv4 adresser delas upp i adressrymder som är sjok av adresser som har ett visst antal bitar

av sitt nummer i binärrepresentation gemensamma. Den största adressrymden är den globala som används på Internet. Eftersom dessa adresser inte räcker till alla datorer som är anslutna har man ett system med lokala adressrymder för subnät.

12.4 Subnät

En del av ett nätverk som har ett eget adressegment, alltså en viss mängd adresser att dela ut till noderna i nätverket, kallas ett subnät. Det behöver dock inte vara en egen adressrymd skild från den globala. En del av ett nätverk som har ett eget adressegment och är kopplad till ett större nät med en eller flera routrar kallas ett subnät. Ett LAN kan bestå av ett eller flera subnät.



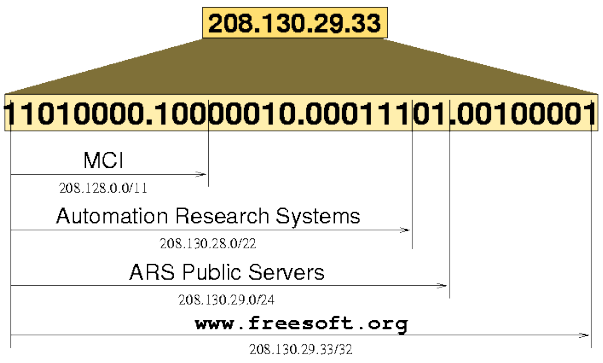
Figur 35: IP-adress för IPv4 Ill.: Wikipedia en användare Indeterminate Från Wikimedia Commons.

12.4.1 Nätmask

När en dator i ett nätverk ska skicka ett paket till en annan måste den veta om datorn finns på det lokala subnätet. Om så är fallet ska den skicka paketet direkt till den datorn annars ska den skicka det till routern för vidare befodran utanför subnätet.

Från början delades internet upp i delnät som kallades klass A, klass B eller klass C som använde 8, 16 respektive 24 bitar för nätverksnamnet och resten att numrera datorer. Det var ett väldigt ineffektivt utnyttjande av adressrymden eftersom varje nät då oftast fick mycket mer adresser än det hade användning för. Man har därför ersatt det med Classless Inter-Domain Routing (CIDR) som gör att man kan dela av ett subnät mellan vilka tal som helst i det binära talet som utgör en IP-adress. Det gör att subnätens storlek kan anpassas mycket effektivare.

Figur 36 visar ett exempel på hur en viss dator finns i ett subnät 208.130.29.0/24 som var en del av ett annat subnät 208.130.28.0/22 som i sin tur var del av ett subnät 208.128.0.0/11. Adresser till den högsta hierarkin av subnät som kan täcka en hel världsdel sköts av Internet Assigned Numbers Authority (IANA) som är en del av Internet Corporation for Assigned Names and Numbers (ICANN). Observera att subnätens nummer alltid börjar som IP-nummret för datorn när det skrivs ut binärt men att det inte är säkert när man skriver ut det på det gängse sättet med decimala siffror.



Figur 36: Exempel på uppdelning av subnät med CIDR. Ill.: Wikimedia commons användare Baccala@freesoft.org Från Wikimedia Commons.

För att datorn ska kunna avgöra det används en sk nätmask (netmask). Den fungerar så att datorn tar en IP-adress som bara är ett stort tal och lägger ihop det med

nätmasken i en binär and-operation. Detta gör den för både mottagaradressen och avsändaradressen och jämför sedan om dessa är lika. Lika adresser betyder att den är lokal olika betyder att den är extern. Nätmasken måste alltså innehålla ettor för ett antal av de mest signifikanta bitarna och nollor för resten det finns flera sätt att skriva ut detta t.ex. är /24, 255.255.255.0 och 11111111 11111111 11111111 00000000 tre sätt att skriva samma nätmask.

12.4.2 Lokal adressrymd

En av vitserna med ett subnät är att man kan göra en lokal adressrymd om man inte har fått globala adresser så det räcker till alla datorer på subnätet. Det fungerar så att DHCP-servern i subnätet delar ut IP-adresser i särskilda nummerserier som inte finns på det internet som man är kopplad till. Nummerserier som får användas för subnät kopplade till Internet är 192.168.X.X, 10.X.X.X och intervallet 172.16.0.0–172.31.255.255. På det viset undviker man att det blir tvetydigt vilken adress som avses. Internt inom subnätet hittar datorerna varandra på de adresser som DHCP-servern tilldelar dem. I ett subnät skickas paketen ut till hela nätverket och den dator som känner igen adressen tar hand om det medans de andra ignorerar det. Om den centrala enheten i stjärnnäverket är en hub pågår all trafik på det viset, medans en switch har ett internt minne av på vilken fysisk anslutning den dator som svarade på samma IP-nummer satt senast och börjar med att prova med att skicka bara på den anslutningen.

På internetet har hela nätverket det IP-nummer som routern har. Det är därför en lite svårare uppgift att kommunicera mellan datorer på internetet och subnätet. Eftersom lejonparten av kommunikation sker tvåvägs har routern en uppgift att fylla med att dirigera den trafiken rätt. Det är två ganska olika problem.

12.4.3 Klienter i lokal adressrymd

Enkla saker som att låta datorer i subnätet använda webben, alltså i princip anropa servrar på internetet är helt nödvändigt för att uppkopplingen ska ha något praktiskt värde. Sån trafik bygger på att det är datorn på subnätet som startar konversationen. Första paketet går alltså från adresserat till en adress på internetet från en adress på subnätet. I en handskakningsprocess som vi studerat tidigare[2] kommer datorerna överrens om ett par portar för den fortsatta konversationen. Detta får nu göras om av routern så att routern har en överrenskommelse till klienten på LANet och en externt till den dator som anropades på internetet. Den måste ju då ha sitt IP-nummer på internetet som avsändaradress för att paketen från servern ska hitta tillbaka och sen komma ihåg till vilken dator på LANet som svaret ska dirigeras.

Allt detta sker automatiskt och är förprogrammerat i routern.

12.4.4 Servrar i lokal adressrymd

Om däremot en dator på subnätet ska kunna svara på förfrågningar från en dator på internetet krävs däremot att det finns ett sätta veta vilken dator på subnätet som ska svara på förfrågan. Det gör man genom att sätta upp en överföringstabell i routern som kallas Network Allocation Table (NAT). Om det bara finns en webbserver på subnätet kan man helt enkelt säga åt routern att skicka alla anrop på port 80 (överrenskommen standard för HTTP) vidare till dess lokala IP-adress. På samma sätt för alla andra servrar som det bara finns en av i nätverket.

Om det finns flera datorer som har en webserver på subnätet måste man på något sätt skilja dem åt. Från internetet har ju båda datorerna samma IP-nummer, nämligen routerns. Det man gör då är att låta en av datorerna (eller båda) få anropen på en annan port än den standardiserade. Då kan man antingen ställa in servern på den datorn att

[2] http://www.lysator.liu.se/~kjell-e/tekla/linux/security/tcpip.html

lyssna på en annan port och sen ställa in routern på att skicka trafik på den porten till datorn i fråga eller så kan man ställa in servern på att svara på standardporten och ställa in routern att översätta både IP-nummer och portnummer när den dirigerar förfrågan.

12.5 Routingtabeller

Det är routerns arbete att tillsammans med de andra routrar den är ansluten till hålla reda på var olika adresser i nätverket finns och räkna ut den effektivaste vägen för ett paket att nå fram till sin destination. Det gör den genom att slå upp om en mottagaradress eller nätadressen till ett subnät där mottagaradressen finns är med en tabell den har lagrad som kallas routingtabell. I ett enkelt system som inte ändras ofta kan man skriva routingtabellen manuellt. I ett mer komplicerat system måste routern hela tiden uppdatera sin routingtabell. Det gör den genom att titta på både avsändar och mottagaradresserna i de paket den får och dra slutsatser om var olika datorer befinner sig. Routrar byter sedan informationen i sina routingtabeller med varandra så att en router har en tabell som innefattar även aktuella anslutningar för andra routrar i sin närhet och kan göra med välgrundade beslut.

12.6 Övningar

Ö18 http://www.ipnummer.se/ är en tjänst som visar vilket IP-nummer du har, vad visar den? Leta reda på i dina nätverksinställningar vad du har för ip-nummer. Stämmer de överrens? Förklara.

Ö19 Vi ska prova att reda på vilka routrar som används på vägen till olika datorer med traceroute. Vi kan t ex börja med att kolla vägen till google.

Övningen går att göra i de flesta operativsystem, det kan finnas program med grafiska gränssnitt också men jag tar kommandoradsversionen.

Ta hjälp av manualsidan för att tolka resultaten. Försök med några olika adresser. Hur många steg kommer du upp till? Är det några adresser där man inte får fram alla stegen?

a) Börja alltså med att ta upp en terminal, i windows ett dosfönster. I UNIX-lika system (Linux, OS X, Solaris m fl) ger man kommandot

```
$ traceroute google.se
```

och i Windows ger man istället

```
> tracert google.se
```

b) Om du gör övningen på skolans nätverk får du troligen bara stjärnor. Vad innebär det? Går det att få bättre resultat till dessa med andra portar och protokoll? Läs manualen och prova. Från svaret på den frågan kan man sluta sig till vilka paket som försvinner.

c) För att se hur paket går den motsatta vägen alltså till din dator måste man kunna köra traceroute från det andra hållet alltså från en dator ute på nätet till sin egen. Det finns en mängd datorer som har webbsidor med traceroute som du kan köra traceroute mot din dator. Prova med någon från listan på http://www.traceroute.org. Går det att hitta din dator eller nätverkets router?

d) Prova att göra traceroute till google från samma webbsida. Vilken dator hade kortast väg, google eller din dator?

e) Om du har tillgång till en smarttelefon kolla om den har någon traceroutefunktion. Om du har möjlighet att koppla upp telefonen som modem för din dator (via kabel, blåtand eller att göra telefonen till en trådlös accesspuntk) kan du sätta upp det och prova med traceroute från datorn. Observerara att du kan behöva ange vilket gränssnitt från datorn eller stänga wifianslutningen till skolnätet så den inte provar det igen.

f) Hemläxa för de som har uppkoppling hemma är att prova med traceroute där och se om det fungerar.

12.7 Frågor

12.1 a) Om man sätter upp ett eget LAN som är anslutet till internet via en router, spelar det då någon roll vilka IP-nummer man använder?
b) Vilka ska man i så fall använda?

12.2 Hur många nätverkskort behöver en router?

12.3 a) Vad menas med DoS-attaker?
b) Hur fungerar de?
c) Vilka datorer står oftast för själva attacken vid DoS-attacker?

12.4 a) Vad är en routers roll i ett nätverk?
b) Hur skiljer sig jobbet för en router som ansluter ett litet lokalt subnät som det på skolan till Internet jäntemot en av de routrar som den skickar vidare paket till? (Förutom det uppenbara att en större har mer trafik.) Nämn arbetsuppgifter som är viktigare för var och en av dem.

12.5 a) Vilka tre grundtyper av nätverkstopologier finns det?
b) Rita upp en skiss över hur noderna är kopplade i respektive typ.
c) Beskriv för och nackdelar med de tre grundläggande nätverkstopologierna.
d) Vilken topologi har en koppling med switch?
e) Vilken topologi har en koppling med hub?

12.6 a) Om en dator har i sina inställningar IP-adress 192.168.2.167 och nätmask 255.255.255.0 vilka av följande IP-adresser ligger då på samma subnät som datorn?
(1) 137.234.43.167
(2) 192.71.13.130
(3) 192.168.87.3
(4) 192.168.2.7
b) Vilka av de ovanstående adresserna hade varit på samma subnät som datorn om nätmasken istället varit 255.255.0.0?
c) Vad har subnätet i a) för adress?
d) Vilken är broadcastadressen för subnätet i a)?
e) Hur många bitar är masken i a)?

*12.7 En dator med ipadressen 123.240.123.234 har nätmask 255.192.0.0.

a) Skriv ut IP-adressen till datorn i binärform.
b) Skriv ut nätmasken i binärform.
c) Hur många bitar är masken.
d) Beräkna subnätets adress i binärform.
e) Skriv ut subnätets adress i decimalform.
f) Reflektera över varför talen i grupp två skiljer sig mellan datoradressen och subnätsadressen.

*12.8 Om du fått tilldelat adressegmentet 123.123.123.0–123.123.123.255 och vill utnyttja det till två subnät från en router, så stora som möjligt, hur ska du då konfigurera nätverken? Ange nätmask, nätverkets adress och broadcastadress för de två subnäten.

13 IP

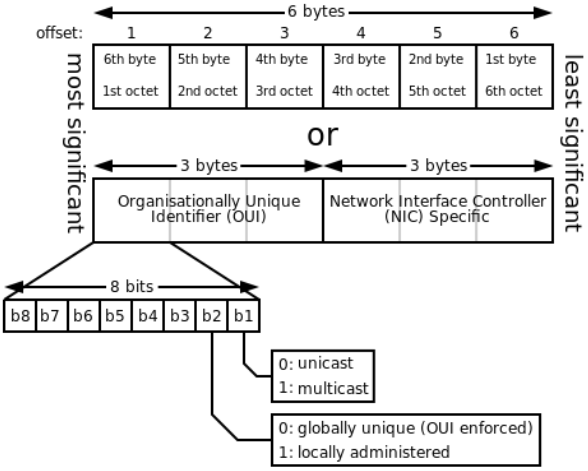
Mål Att förstå skillnaden i uppbyggnad mellan IPv4 och IPv6 och förstå hur övergången mellan dem kommer att ske.

Läs Nedanstående

Besvara Instuderingsfrågorna 13.6.

13.1 Hårdvaruadress

Varje elektronikkrets för nätverksanslutning till IP-nätverk får vid tillverkningen en hårdvaruadress. Det är ett 48 bitars heltal som är unikt för varje krets, alltså trådbundet eller trådlöst nätverkskort. Den kallas också MAC-adress. Den finns av praktiska skäl för att man ska kunna hålla isär olika enheter som kopplas till ett nätverk. Om man vill låtsas vara någon annan kan man oftast helt enkelt be operativsystemet att skicka ut en annan hårdvaruadress än den verkliga med chans för att det leder till problem om den verkliga också finns på nätverket. Hårdvaruadressen är den som används för att adressera paketen på länknivån i IP-stacken.



Figur 37: Uppbyggnaden av en MAC-48 adress. Ill.: Wikimedia commons användare Inductiveload Från Wikimedia Commons.

13.2 Adresshierarki

Fördelning av adresser fungerar ganska olika mellan IPv4 och IPV6. Adresser i IPv4 är ett 32 bitars heltal beskrivet i stycke 12.3. De användes till en början i en hierarki där man hade tre nivåer av subnät klass A, B ch C. Systemet var enkelt att administrera men ganska ineffektivt i fråga om adressrymd eftersom många adresser aldrig användes. Därför började man med nätmask som beskrivits i stycke 12.4.1, och lokal adressering med NAT.

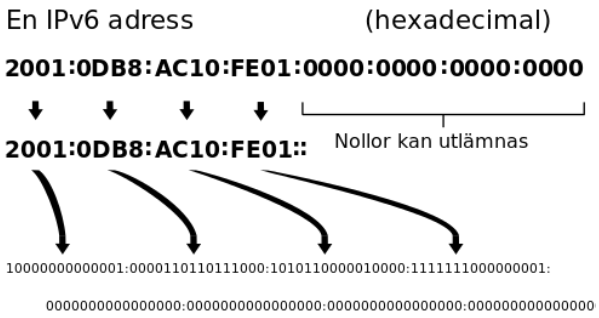
När nu ändå adresserna håller på att ta slut har man tagit fram en ersättare till IPv4 som heter IPv6 med längre adresser, se figur 38. I och med det bytet vill man förutom att utöka antalet tillgängliga adresser ändra strukturen på hur adresser tilldelas och fördelas så att många saker ska fungera enklare och effektivare. Det är alltså ganska mycket nytt tänk bakom.

IPv6 har en adresslängd på 128 bitar vilket ger $3 \cdot 10^{38}$ adresser vilket är ett enormt stort tal. Sett till antalet tillgängliga adresser skulle det ha räckt med mycket mindre utökning för att de ska räcka i all framtid om man fortsätter med ett liknande system som idag. Tanken med att ha en så stor utökning är att ta till vara den ekelhet som fanns i internets barndom och ha ett så stort överflöd av tillgängliga adresser att man inte behöver strula med nätmasker, i alla fall inte på användarnivå utan bara mellan routrar.

13.2.1 Utskrift av adresser

En IPv4-adress är ett 32 bitars heltal. I skrift brukar det skrivas uppdelat med punktnotation så att det uppdelat i 4 oktetter (bytes) skrivs ut med värdet i varje byte som decimaltal avgränsat av punkter, se fig 35. För nätadresser kan nätmaskens längd skrivas med antalet bitar i masken med ett snedstreck efter adressen t.ex. 192.168.0.0 /16.

En IPv6-adress är ett 128 bitars heltal. I skrift brukar det skrivas uppdelat i 8 block som vardera är 16 bitar (2 byte), blocken skrivs som hexadecimala tal med 1 till 4 siffror, se figur 38. Om det finns en eller flera grupper efter varandra som har värdet 0 kan de uteslutas genom att man skriver ett dubbelt kolontecken (::) istället. Adressen kan då tolkas och fyllas ut med nollor så den blir 128 bitar lång. Dubbelt kolon kan bara användas en gång per adress annars skulle man inte kunna veta hur många grupper med nollor som ska skrivas in på varje ställe.



Figur 38: Utskrift av en Ipv6-adress. Block med värdet 0 efter varandra kan uteslutas och skrivas med dubbelt kolon (::). I det här fallet skulle adressen kunnat avkortas lite till genom att den inledande nollan i grupp 2 också kunnat uteslutas.

13.2.2 Nätverksadress

I IPv4 avgränsas nätverk nätverket i adressen med hjälp av nätverksmasken och kan variera mycket. I IPv4 avgränsas alltid adressen i en nätverksdel som är 64 bitar och en individuell adress som är 64 bitar för en enskild dator. Det gör att man i operativsystem för datorer inte behöver bry sig om nätmasken. Man har också bytt namnet på den till routing prefix men det är precis detsamma som den gamla nätmasken. Det gör också routingen enklare för routrarna för att de inte ens behöver läsa in hela adressen utan bara första halvan. Av dessa delas normalt subnät med 48 bitars adress ut på högsta nivån, alltså med routingprefix /48. Det gör att Internet Service Provider (ISP) och den som köper uppkopplingen har 16 bitar för att dela upp nätverket i subnät och det är på denna nivå det finns någon egentlig användning för routing prefix.

13.3 Adressering av datorer

I IPv4 bestämms antalet adresser i ett nätverk av nätmasken, och tillåter normalt en adress per dator.

I IPv6 är den andra halvan av adressen är för enskilda datorer. Det innebär att på varje lokalt nätverk finns 2^{32} adresser, 4 miljarder gånger fler än alla möjliga IPv4 adresser. I ett sådant nätverk finns ingen anledning att spara på adressutrymme. Adresseringen av datorer skiljer sig också radikalt mellan IPv4 och IPv6.

13.3.1 IPv4

Sambandet mellan den logiska IP-adressen som används i internetlagret och den hårdvarumässiga MAC-adressen som används i länklagret i IP-stacken sköts genom utbyte av speciella Address Resolution Protocol (ARP)-paket och datorer och nätverksutrustning för därför ständigt en ARP-tabell där de håller reda på sambandet mellan dem för den utrustnig som datorn kan adressera på länknivån.

I IPv4 finns det två sätt att få en adress. Det ena är att man som användare av datorn skriver in ett IP-nummer manuellt. Om det är ett nummer som är en giltig adress som ligger i lokalt nätverk och ingen annan redan använder det fungerar det. Om man skriver in ett IPnummer som redan används blir det kollisioner och strul för båda datorerna som har samma adress. Det är alltså upp till den som vill ansluta en dator att höra med ansvarig för routern om en lämplig adress.

Det finns ett automatiskt sätt att lösa detta som heter DHCP där datorn går ut med en fråga på broadcast och får ett svar från DHCP-servern (ofta densamma som routern) med alla nödvändiga data för nätverket, en IP-adress för datorn, nätmask, och IP-adress för gateway. Detta svar adresseras till rätt dator med hjälp av hårdvaruadressen.

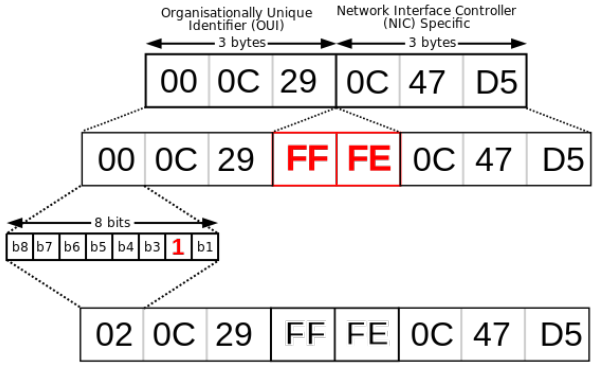
Förutom adresserna till de enskilda datorerna i nätverket finns även en broadcastadress som går till alla datorer i subnätet.

13.3.2 IPv6

I IPv6 med i praktiken obegränsat adressutrymme har man ersatt förfarandet att begära en adress från routern med ett nytt system där varje dator sätter upp flera egna IP-adresser. Till varje interface kan det finnas en mängd adresser. Systemet med broadcast från IPv4 är borttaget så varje dator svarar bara på paket som är adresserat till den. För att kunna adressera till flera datorer i ett nätverk ska datorn därför ha några adresser som inte är unika för den utan adresser som används av flera andra datorer också att skicka paket till en sådan adress kallas multicast.

En sådan adress är ff02::1 kallad all-nodes link-local multicast group som alla datorer sätter upp direkt när de märker att de har anslutning. Den har alltså samma funktion som den tidigare broadcastadressen. Routern ser till att inte släppa ut de paketen så att de håller sig i det lokala nätverket. Eftersom att skicka paket till alla datorer i ett nätverk ger mycket trafik försöker man undvika det så länge som möjligt, istället används andra grupper, alla datorer antas öppna en gruppadress som har ett särskilt prefix och som slutar med samma sista 24 bitar som varje adress den har som enskild adress. På det viset kan man skicka multicast till de datorer som har liknande adresser.

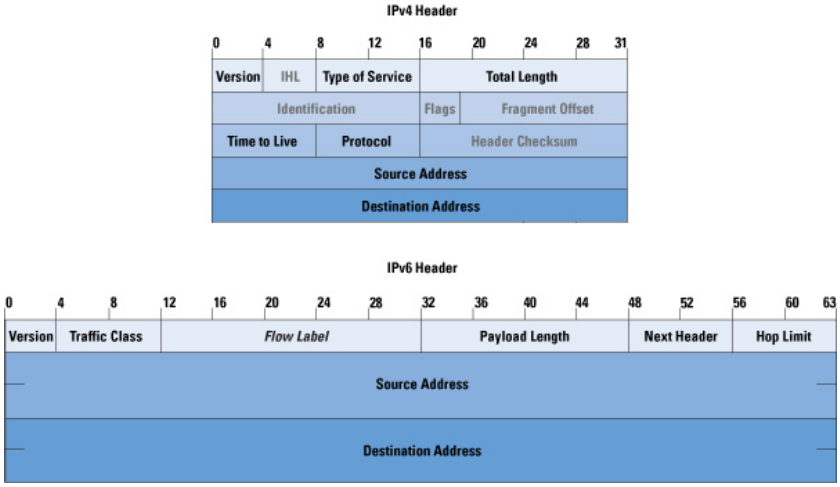
Först och främst för att kunna komma i kontakt med de andra datorerna på nätverket gör den en egen lokal adress för nätverket kallad Link-local address. För att vara säker på att den ska bli egen använder man hårdvaruadressen på nätverkskortet för 48 av de 64 bitarna, se figur 39. Eftersom hårdvaruadressen är unik ger det en adress som inte är upptagen. Eftersom å andra sidan man kan trixa och ändra så att datorn använder något annat än den riktiga hårdvaruadressen gör man ändå en kontroll genom att skicka ut ett paket till de gruppadress som alla datorer med ett visst möster i adressen också öppnar från den anonyma adressen ::/128. Om allt är normalt får man inget svar på den är adressen ledig och då ger datorn sig själv den adressen. Vid det här laget har datorn alltså en logisk adress i IP-lagret utan att ha behövt skicka några ARP-paket och det här systemet ersätter därför ARP för att hålla reda på hårdvaruadresserna för länknivån



Figur 39: En 64-bitars datoradress skapas genom Modified EUI-64 genom att 2 bytes $FFFE_{16}$ skjuts in mitt i MAC-adressen och sen inverterar man bit 7. Ill.: Wikimedia commons användare Grupp Från Wikimedia Commons.

inom nätverket. Först därefter skickar den ut en förfrågan till en speciell gruppadress för routrar och frågar efter om det finns någon router i nätverket.

13.4 Paket i IPv4 och IPv6



Figur 40: Jämförelse mellan header i IPv4 och IPv6. Headern för IPv6 blir totalt längre tack vare att adresserna är så långa men resten av headern är förenklad i förhållande till IPv4. Ill.: cisco Från Wikimedia Commons.

En annan förnyelse och effektivisering med IPv6 är att man tagit fasta på vilken information i headern på varje paket som används mest. Fig 40 visar hur headern ser ut i de olika paketen. Man har rationaliserat en del och lag till en möjlighet att ha något som kallas extended header med mer information efter den obligatoriska headern om man behöver den information som tidigare var med. Den största förändringen prestandamässigt är att header checksum har tagits bort. Det är en slags fingeravtryck av samma typ som vid kryptografisk signering (se vidare stycke 14.3.1) som räknas ut från den övriga informationen i headern. Eftersom räknaren "time to live" ändras vid varje router måste den också räknas om och ändras, vilket tar mycket längre tid än att ändra själva räknaren. Själva räknaren som tidigare kallades "time to live" kallas i IPv6 "hop limit".

13.5 Övergång till IPv6

Ett stort problem idag är hur övergången till IPv6 ska gå till. Båda systemen är idag igång men bildar två relativt isolerade system mellan vilka det är en del trixande att skicka information. Så länge det finns en övervikt av tjänster tillgängliga på servar med IPv4-adresser är incitamentet för de som har IPv4 adresser att byte ganska litet. Standarden för IPv6 stod mer eller mindre klar 1997, den började tas i bruk i domäerna .jp och .kr 2004, den första större användningen internationellt var vid OS i Peking 2008 eftersom det då gick åt väldigt mycket nya adresser och Kina då också hade behövt börja ta till IPv6. De tydligaste ansträngningarna för att försöka få IPv6 att ta luft hittils har varit World IPv6 Day 8 juni 2011 då man gjorde ett fullskaletest med att aktivera IPv6 parallellt med IPv4 för många av de stora webbtjänsterna under ett dygn samordnat av Internet Society. Testet föll ut utan stora problem och gjorde att de inblandade beslöt att öppna IPv6 permanent för dessa stora tjänster 6 juni 2012 kallat World IPv6 Launch Day. Det förväntas att övergången till IPv6 ska komma lite som en ketchupeffekt, att det har pågått under många år med att bara några få som tvingas till det skaffar IPv6-adresser men vid någon tidpunkt kommer det att finnas fler intressanta saker adresserade till IPv6 än till IPv4 så att det blir en ganska snabb överflyttning av de flesta, när och om det inträffar återstår att se.

Figur 41: Internet societies logga för World IPv6 launch Copyright: Internet Society Från http://www.worldipv6launch.org/downloads/.

13.5.1 Bytesstrategier

Målet att hela internet ska byta IP från version 4 till version 6 verkar de flesta vara överrens om. Under övergångsperioden då vissa har version 4 och andra har version 6 behövs övergångslösningar som ger möjlighet att komma åt även datorer som sitter på nätverk som använder det andra protokollet. Det har kommit en uppsjö av sådana. Problemet ligger inte framför allt på datorerna, det är ganska enkelt att få en dator att öppna både version 4 och 6 samtidigt på samma anslutning. Problemet ligger framför allt i nätverken där de som äger nätverk behöver göra stora insatser framför allt i form av arbetstid men kanske även uppdatera både hård- och mjukvara. Det handlar därför till viss del om att kunna sända IPv6-paket på IPv4 nätverk det kallas att tunnla IPv6 över IPv4. Det finns en mängd tekniska lösningar i drift för att göra det och för att adressera om mellan de två versionerna. De som tas upp nedan är inte heltäckande.

13.5.2 Dual stack

Den konceptuellt enklaste lösningen som också är den mest stabila och ger smidigast övergång är att helt enkelt öppna för båda protokollen samtidigt på samma fysiska nätverk, det kallas dual stack. Det är dock inte alltid det tekniskt enklaste och billigaste alternativet för att en del av nätverksutrustningen inte klarar av IPv6.

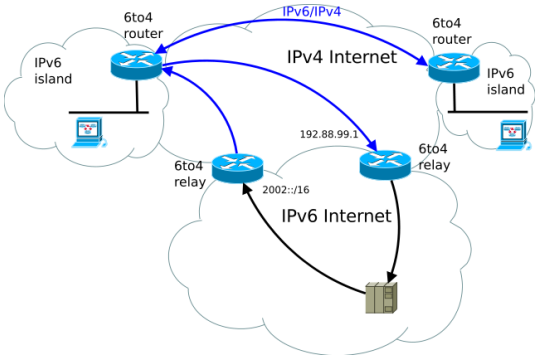
13.5.3 Tunnling

Tunnling betyder något förenklat att man manuellt sätter upp en länk från en router som finns i ett IPv6 nätverk till en annan i ett annat IPv6 nätverk. Man låter routern ta alla IPv6-paket som har adress någonstans i det andra nätverket och packar om IPv6-paketet som innehåll i ett IPv4-paket som skickas till routern i det andra nätverket. Den mottagande routern packar upp IPv4-paketet och betraktar det igen som ett IPv6-paket och skickar det vidare inom nätverket. Det kallas att tunnla IPv6 över IPv4. Det går att göra tvärs om och tunnla IPv4 över IPv6.

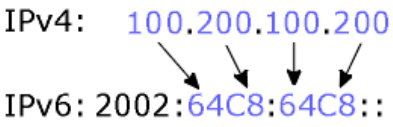
Konceptet är välkänt och används även på andra nivåer i IP-stacken. Vi har redan använt det när vi flyttat filer med SCP till webservern. SCP är i stort sett det gamla FTP-protokollet tunnlat över en SSH anslutning som står för krypteringen.

13.5.4 6to4

6to4 är en lösning där man bildar en låter datorerna få en IPv6-adress med ett speciellt prefix som sedan innehåller IPv4 adressen som ju bara är en fjärdedel så lång som IPv6-adressen som en del av routing prefix av denna. Adresser bildas genom att sätta de



(a) 6to4. Ill.: Wikimedia commons användare Mro Från Wikimedia Commons.



(b) Bildandet av en 6to4 IPv6 adress. Ill.: Wikipedia de användare Matthäus Wander Från Wikimedia Commons.

Figur 42

två första bitarna till 2002_{16} och sedan lägga in IPv4 adressen, se figur 42b. En IPv4 adress ger alltså ett helt /48 nät IPv6 adresser. För två isolerade nätverk som sitter som subnät på IPv4 internet går adressöversättningen direkt åt båda håll, se figur 42a. För att kommunicera med dem som har permanenta IPv6 adresser på internet måste dessa paket gå genom speciella routrar som kan adressera om dem.

13.6 Frågor

13.1 För var och en av följande adresser som alla utgör heltal: MAC-adress för nätverkskort, IPv4-adress, IPv6 adress:

a) Hur lång är adressen uttryckt i bitar?
b) Hur brukar de normalt skrivas ut i text för människor?
c) Var och en av adresserna delas i två delar med olika innebörd vilka?
d) Hur delas adressen i de två delarna?

13.2 Beskriv varför man använder så långa adresser som 128 bitar till IPv6.

13.3 Skriv följande IPv6 adresser på kortaste form.

a) 0000:0000:0000:0000:0000:0000:0000:0001 (loopbackadressen)
b) 2001:0db8:0000:0000:0000:ff00:0042:8329
c) 2001:0db8:0000:0000:ff00:0000:0042:8329

13.4 Beskriv skillnaden mellan hur det går till när en dator får en tilldelad adress i IPv4 mot självadressering i IPv6.

13.5 I IPv6 kan en och samma nätverksinterface ha flera IP-adresser. I själva verket har de allra minst 3, varför räcker det inte med en, som i IPv4?

13.6 Trots att headern för ett IPv6 är 40 B till skillnad från IPv4 headern som är 20 B kommer den i många fall att vara effektivare att hantera. Nämn två orsaker.

13.7 Vilken IPv6 nätadress skulle vi få på skolan om vi aktiverade 6to4 i routern som har publikt IP-nummer 83.150.146.103?

13.8 Ett annat sätt än de nämnda i texten som kan användas till att mappa adresser mellan IPv4 och IPv6 är att använda speciella IPv6 adresser i segmentet ::ffff:0:0/96 där man lägger IPv4 adressen som de sista 32 bitarna. Då kan man av bekvämlighet ibland skriva de sista två grupperna i punktnotation som fyra oktetter med decimala tal.

a) Vilken IPv4 adress mappas på detta sätt till ::ffff:123.45.123.45?
b) Vilken IPv4 adress mappas på detta sätt till ::ffff:c047:d37?
c) Vad blir IPv6 adressen som man får genom att mappa 173.194.44.119 på detta sett skriven med decimal notation för de sista fyra byten?

Vad blir IPv6 adressen som man får genom att mappa 173.194.69.139 på detta sett skriven helt i hexadecimal notation?

14 Kryptering

Mål Efter dagens lektion är det meningen att ni ska

1. förstå princierna bakom kryptering
2. förstå skillnaden mellan symmetrisk och asymmetrisk kryptering
3. känna till några användningsområden
4. kunna bedöma säkerhetsnivån på webbanslutningar.

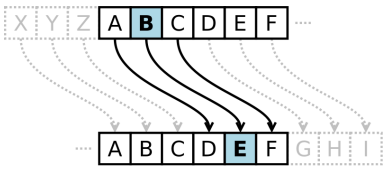
Läs
- Nedanstående

Övningar Gör övningarna i stycket 14.6.

Besvara Instuderingsfrågorna 14.7.

14.1 Introduktion

Kryptering har använts sedan Ceasars tid. Kryptering bygger på att man har ett överrenskommet sätt att beräka ett krypto av ett meddelande, en algoritm och en eller fler hemliga tal, kodnycklar, som används i algoritmen för att beräkna hur det krypterade meddelandet ska se ut. Du har säkert som barn provat att använda Caesars krypto. Man skriver ett meddelande med bokstäverna i alfabetet och förskjuter dem sedan ett antal steg i alfabetisk ordning, se figur 43. Det fungerar så att den kända algoritmen är att man ska förskjuta bokstäverna i alfabetet och den hemliga nyckeln är hur många steg man ska göra detta. Nu fungerar det inte särskilt bra om principen är känd eftersom man bara behöver prova max 29 gånger för förstå meddelandet även om man inte har nyckeln.

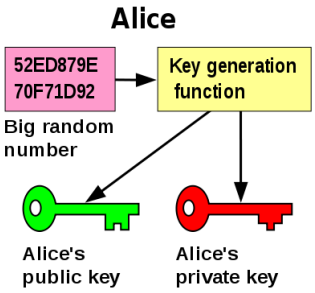


Figur 43: Ceasars chiffer är ett mycket enkelt krypto. Ill.: Wikimedia commons användare Cepheus Från Wikimedia Commons.

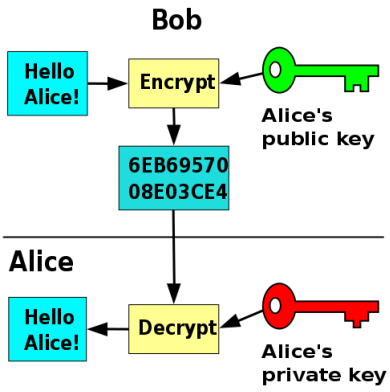
Idag skiljer man på två olika former av kryptering: Symmetrisk och asymmetrisk.

14.2 Symmetrisk kryptering

Ceasars chiffer är ett exempel på en symmetrisk kryptering, samma nyckel används för att kryptera meddelandet som för att dekryptera det. Det är ganska enkelt att göra ett sådant krypto. Man tar fram ett slumpvis heltal och så betraktar man hela det meddelande, fil eller lösenord man vill kryptera som ett heltal, det är ju för datorn ändå bara en rad med ettor och nollor. Sedan gör man någon överrenskommen matematisk operation med de två talen som är reversibel, alltså går att göra ogjord genom att göra beräkningen baklänges. I fallet Cesars krypto överför man bokstäverna till talen 1-29 och adderar nyckeln till varje tal och kodar om som bokstäver. Vid dekrypteringen gör man motsatsen dvs överför bokstäver till siffror, subtraherar nyckeln från varje tal och kodar tillbaks till bokstäver. Om kodnyckeln är stor och algoritmen skapar lite mer oordning i det kodade meddelandet än hos Caesars chiffer blir det snabbt mycket svårt att knäcka kryptot utan att känna kodnyckeln. Det stora problemet med metoden är att man på något hemligt vis måste använda samma nyckel.



(a) Bildande av nyckelpar. Ill.: David Göthberg Från Wikimedia Commons.



(b) Kryptering med Alice publika nyckel. Ill.: David Göthberg Från Wikimedia Commons.

Figur 44: Asymmetrisk kryptering. Ett par nycklar bildas, varav den ena hålls hemlig. Den publika nyckeln kan användas för att kryptera meddelanden som bara kan läsas av Alice som har den hemliga nyckeln.

14.3 Asymmetrisk kryptering

Boten mot den symmetriska krypteringens svaghet är att man använder en algoritm som inte är reversibel, dvs om man använder en viss nyckel för att kryptera meddelandet måste man använda en annan för att dekryptera det. De två nycklarna bildar ett par och de måste räknas ut tillsammans (figur 44a). Den nyckel som används för att kryptera ett meddelande med algoritmen, den publika, behöver man då inte hålla hemlig, den går ändå inte att använda för att läsa meddelandet. Den som ska ta emot det hemliga meddelandet har den andra, hemliga, nyckeln som bara kan användas för att dekryptera med (figur 44b). Problemet är att om algoritmen och den publika nyckeln är känd går det i princip att räkna ut den hemliga nyckeln. Det gäller alltså att göra en algoritm och ett nyckelpar där det är enkelt att räkna ut ett par nycklar tillsammans men svårt att räkna ut den hemliga nyckeln från den publika. Det som i praktiken används är att man tar två stora primtal ock multiplicerar ihop dem till ett annat mycket större tal. Det finns ingen känd matematisk algoritm för att primfaktorera stora tal på ett effektivt sätt men att multiplicera ihop två tal är enkelt. Man kan göra algoritmer där man använder det stora talet för kryptering men måste känna dess primfaktorer för att dekryptera det.

Den asymmetriska krypteringen fungerar bara enkelriktat. Vitsen med den är ju att den ena nyckeln inte behöver hållas hemlig och kan därför inte användas för kryptering. För att kunna ha en tvåvägskommunikation måste den andra parten i samtalet också generera ett nyckelpar där han behåller sin hemliga del och ger över den publika till den första. Meddelanden i de båda riktningarna går då med två helt olika krypteringar som inte har något med varandra att göra.

14.3.1 Digitala signaturer

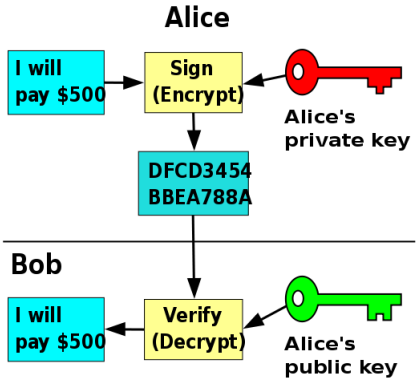
Man kan vända på algoritmerna om man har ett annat syfte. Om man har gett ut en publik nyckel som man inte anstränger sig att hålla hemlig kan ju avsändaren vara säker på att det han skickar inte kan läsas av någon annan en den tänkta mottagaren men mottagaren kan ju bara vara säker på att det han tar emot har kommit från någon som har hans publika nyckel. Det omvända problemet är alltså att ta reda på om avsändaren

är den han utger sig för att vara, dvs att sätta digitala signaturer på handlingar. Man kan då kryptera ett meddelande med sin hemliga nyckel, då kan vem som helst som har den publika nyckeln läsa meddelandet men vet då också att det kommer från den som har den hemliga nyckeln och att det inte är förvanskat på vägen.

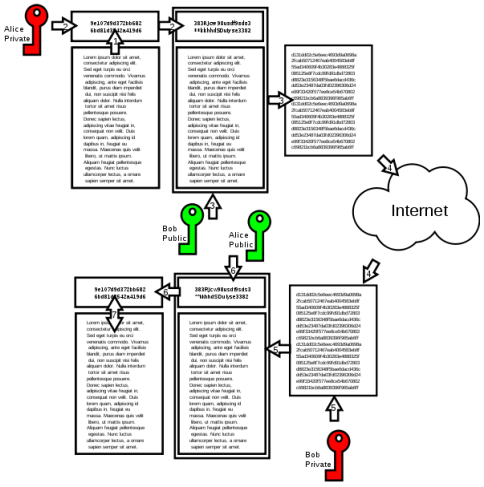
I praktiken behöver man inte skicka meddelandet krypterat eftersom det är enkelt att dekryptera. Man kan skicka meddelandet i klartext men ta ett digitalt fingeravtryck på den som man kryptererar för att visa originalets äkthet. Ett digitalt fingeravtryck är en algoritm som räknar ut ett mindre tal från det stora tal som hela meddelandet är och som är svår att göra baklänges så att det är svårt att göra ett annat meningsfullt meddelande som ger samma fingeravtryck.

Om man vill både signera och kryptera meddelandet blir gången som visas i figur45b.

1. En hash (ett fingeravtryck) tas på filen.
2. Alice privata nyckel används för att kryptera fingeravtrycket för att få en digital signatur. Signaturen läggs till datafilen.
3. Alice använder Bobs publika nyckel för att kryptera filen och signaturen.
4. Krypterat data skickas
5. Bob tar emot data och dekrypterar med sin privata nyckel.
6. Bob dekrypterar signaturen med Alice publika nyckel.
7. Bob jämför den dekrypterade signaturen med det fingeravtryck han själv tar på filen.



(a) Signering egen hemlig nyckel. Lägg märke till skillnaden mot figur 44a att meddelandet nu går från Alice till Bob. Ill.: David Göthberg Från Wikimedia Commons.



(b) Kryptering och signering av dokument. Ill.: Wikimedia commons användare Pluke Från Wikimedia Commons.

Figur 45: Signering.

14.3.2 Vem kan man lita på?

Problemet med asymmeterisk kryptering och digitala signaturer är förstås att veta att den publika nyckel man fått verkligen hör ihop med den person man tror att det är. I de mest strikta sammanhangen gäller det bara att fysiskt träffa personen i fråga och få en utskrift av nyckeln på papper som man kan arkivera för att kontrollera att man använder rätt nyckel när man kontrollerar den digitala signaturen. I större sammanhang med många inblandade samlar man alla publika nycklarna på en server och så får en användare sätta sin digitala signatur på en annan användares signatur som han har kontrollerat är äkta. Då kan man välja att lita på de personer för vilka man själv har

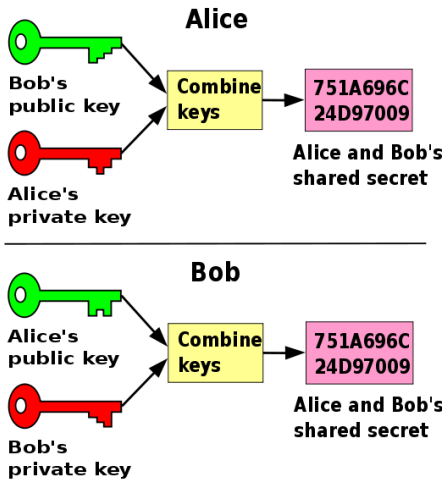
kontrollerat att nycklarna är äkta och säga att man litar på alla nycklar dessa personer har kontrollerat, eller vidare ett eller flera steg till, lita på nycklar som signerats av de personer vars nycklar har signerats av de personer jag kontrollerat osv. På det sättet får man ett nätverk av tillitsfulla personer (web of trust). Det är avgörande t.ex. för att stora projekt med öppen källkod ska fungera där bara redan kända eller personer som redan involverade personer litar på ska kunna delta.

För krypterad trafik på webben använder man asymmetrisk kryptering (för utbyte av symmetriska nycklar se nedan) men för en webbplats är det oftast inte praktiskt att alla besökare kommer till innehavaren och får en utskrift av kodnyckeln. Istället används certifikat som innebär signering av nyckeln på samma sätt som i ett web of trust, men där man har en utfärdare som man kan lita på vars uppdrag är att ta reda på att den som vill certifiera sin nyckel är den han utger sig för att vara. Det finns några tiotal sådana Certificate Authoritys (CAs) som har så gott renommé att de vanliga webbläsarna har listat dem som tillitsfulla. För de webbplatser som har sina nycklar certifierade av någon av dessa CAs som webbläsaren man använder listar som tillitsfulla får man inte upp någon fråga om man ska lita på krypteringsnyckeln.

14.4 Utbyte av symmetriska nycklar

Ett av de stora problemen med asymmetrisk kryptering är att de i princip är möjliga att knäcka. För att de inte ska vara praktiskt möjliga att knäcka, (det ska inte vara möjligt att räkna fram den hemliga nyckeln inom rimlig tid) måste man använda ganska stora nycklar och själva krypteringen och dekrypteringen blir då matematiskt jobbig att utföra och tar mycket processorkraft. Det kan ju gå an för små överföringar som ett ebrev, men inte för stora filöverföringar. Ett vanligt sätt är därför att man börjar en filöverföring med att generera en slumpmässig och mycket kortare symmetrisk krypteringsnyckel och så krypterar man den med sin hemliga asymmetriska nyckel. Därefter kan man använda symmetrisk kryptering för med den ännu hemliga nyckeln för den stora filen.

Whitfield Diffie and Martin Hellman visade 1976 att man inte ens behöver skicka över den symmetriska nyckeln. Med speciella algoritmer går det att konstruera nyckelsystemet så att de två parterna kan konstruera en likadan symmetrisk nyckel från sin egen hemliga nyckel och motpartens publika nyckel på ett sätt som inte går att rekonstruera utan att man känner någon av de hemliga nycklarna.



Figur 46: Utbyte av symmetrisk nyckel enligt Diffie-Hellman. Ill.: David Göthberg Från Wikimedia Commons.

14.5 Implementeringar

På trådlösa nätverk används kryptering för uppkopplingen på nätverksnivån. Det finns två standarder Wired Equivalent Privacy (WEP), som är knäckt för 10 år sedan och idag i praktiken inte ger något skydd alls, och dess ersättare Wi-Fi Protected Access (WPA). Det är en kryptering bara för den luftburna signalen mellan datorn och den trådlösa accesspunkten. Det används främst för att kontrollera vem som ska kunna logga in på ett nätverk och ger begränsad säkerhetsvinst för användaren eftersom krypteringen tar slut vid accesspunkten och kan avlyssnas nästan lika enkelt på det trådburna nätverket. Det man i praktiken behöver göra är att välja WPA och knappa in ett lösenord för att komma ut på nätverket.

14.5.1 SSL/TLS

Secure Sockets Layer (SSL) och dess efterträdare TLS är den kryptering som är vanligast på webben. Den infördes av Netscape men är nu en öppen standard. Det är en utväxling av symmetriska nycklar via asymmetrisk kryptering. Det är en kryptering som kan användas på många olika protokoll i applikationslagret. Webbtrafik som normalt är HTTP på port 80 blir med krypteringen HTTPS som sker på port 443. Krypteringen är implemnterad i webbläsaren och webbservern. I praktiken ser man att man använder krypteringen på att URLen börjar på `https://` istället för `http://`. Den publika nyckeln som webservern ger ut för den asymmeteriska krypteringen måste certifieras. Det kan göras av en CA vilket man ska kräva av alla kommersiella bolag för att lita på dem på nätet. Webserverns ägare kan också certifiera sina nycklar själv. Detta certifikat betyder då inget ur säkerhetssynpunkt för den som surfar, man får då upp en varningsruta i webbläsaren med en fråga om man ska välja att lita på nyckeln. Detta är det vanligaste för mindre webbservrar eftersom det är ganska dyrt att certifiera nycklar från en CA med gott renommé. Kontrollen CA gör kan vara olika rigorös, den noggrannaste kontrollen är Extended Validation (EV) och brukar visas genom att företagets namn visas i grönt framför URLen i webbläsarfönstret. Det används främst av banker och liknande.

14.5.2 SSH

För inloggning till datorer är SSH idag den vanligaste metoden. Det sker med symmetrisk kryptering genom utbyte av symmetriska nycklar via asymmetrisk kryptering eller via bildande av symmetrisk nyckel med Diffie-Hellmans metod. Servern är en dedikerad server som lyssnar på port 22 och är betrodd att logga in användre av operativsystemet. Vid installation av SSH på en dator bildas det ett nyckelpar för datorn. Dessa nycklar används sedan vid utbytet av symmetriska nycklar. Första gången en uppkoppling sker måste man godkänna att den datorns publika nyckel läggs till i en lista över kända datorer.

Verifieringen av identiteten på den som loggar in kan ske genom att användarens lösenord skickas via den krypterade anslutningen eller genom att användaren skapar par av asymmetriska nycklar där de publika nycklarna växlas mellan de två datorerna. Vid användning av den hemliga nyckeln på den lokala datorn bakas den ihop med en lösenfras som man väljer när nyckeln skapas så att den är obrukbar utan denna lösenfras. På så sätt undviker man att någon kan stjäla nycklar för att logga in på användarens konton på andra datorer.

14.5.3 PGP/GPG

För både kryptering och digital signering av filer och framför allt epost används Pretty Good Privacy (PGP) eller gnus motsvarighet GNU Privacy Guard (GPG). Det bygger på asymmetrisk kryptering enligt vad som beskrivits ovan och trovärdigheten bygger på web of trust.

14.5.4 DRM

Kryptering används med blandad framgång för olika typer av kopieringsskydd, DRM. Man krypterar då innehållet i den musik, film eller vad man vill skydda och dekrypterar den när man ska använda den. Metoden är inte så framgångsrik, det är lätt at förstå eftersom man måste ge användaren nyckeln till kryptot för att man ska kunna höra, se eller vad man ska göra med verket, men förbjuda honom att använda den till något annat.

Det försöker man göra antingen genom att bygga in krypteringsnyckeln i apparater som i fallet med DVD. DVD har en kryptering, Content Scrambling System (CSS), och alla DVD-spelare för anslutning till TV har krypteringsnycklarna inbyggda medans DVD-spelare i datorer inte har det. Kryptot knäcktes av en norsk förstaårsgymnasist känd

Figur 47: Jon Lech Johansen Copyright: Jon Lech Johansen Från http://nanocr.eu/picture.php.

som DVD-Jon, (figur 47) och är nu allmänt tillgängligt på nätet så man kan använda ett program i datorn för att göra dekrypteringen istället.

I andra fall försöker man att bygga in dekrypteringen i programvara som släpps specifikt för att spela upp filer som är krypterade i ett visst format. De flesta eller alla sådana försök är knäckta men vållar stora problem för användarna som blir låsta till att köpa film och musik i ett visst format för att passa till den mjukvara man har. I vissa undersökningar har det framkommit att den främsta anledningen illegala fildelare angett för att ladda ner filer illegalt är praktiska snarare än ekonomiska för att det är så svårt att hantera de krypterade formaten.

Ett led i detta är att de flesta nya datorer idag har ett ett chip på moderkortet eller en krets i processorn som innehåller en individuell hemlig nyckel för datorn, som bara tillverkaren av kretsen känner till och som inte datorägaren kan ta reda på. Detta mycket omstridda koncept kallas av industrin Trusted Computing medans kritikerna myntat begreppet Treacherous Computing. Ett av kraven för Windows 8 är att detta ska finnas i datorn så att det går låsa datorn till att bara kunna starta upp kärnan i det operativsystem som den levereras med. I tidare förslag på kraven hade de att detta även skulle vara aktiverat i datorerna så att det inte skulle vara möjligt att installera något annat operativsystem alls men det har tagits bort.

14.6 Övningar

Ö20 Besök några olika sidor som använder HTTPS med olika höga säkerhetskrav. T. ex. https://secure.wikimedia.org/wikipedia/sv/wiki/Portal:Huvudsida, https://elev.rymdgymnasiet.com/, https://internetbanken.privat.nordea.se/nsp/engine. Blir du tillfrågad om du ska lita på certifikatet? Vad kan du se för information om krypteringen i din webbläsare? Kontrollera sidinformation, (en hänglåsikon i nedre högra hörnet i Firefox). Vem har utfärdat certifikatet för sidan? Markerar webbläsaren på något sätt i adressfältet som ger vägledning?

Ö21 Undersök de sidor du normalt använder och har inloggningsuppgifter till (webmailkonton, sociala nätverk, internetbutiker, din bank etc). Vilka använder kryptering? Vilka erbjuder kryptering som alternativ?

Ö22 Gå till en webbhandelsplats, gärna en du handlat på tidigare och har ett konto, börja göra ett inköp och kontrollera vad som händer. Annars kan du skapa ett dumkonto någonstans (http://www.clasohlson.se). Är någon del av kedjan, (titta i webbutiken, lägga i kundvagnen, gå till kassan, betala) krypterad? Verkar behandlingen bra? Avbryt köpet innan du behöver betala.

14.7 Frågor

14.1 a) Nämn något användningsområde för symmetrisk respektive assymmetrisk kryptering.
b) Varför är asymmetriska krypteringsnycklar längre än symmetriska för samma säkerhet?
c) Mycket datortid har använts för att beräkna stora primtal. Varav intresset för det?
d) Hur många krypteringsnycklar går det åt för att få upp en tvåvägskommunikation med symmetrisk kryptering?
e) Hur många krypteringsnycklar går det åt för att få upp en tvåvägskommunikation med asymmetrisk kryptering?

14.2 a) Vilka metoder finns för att veta att den krypterade fil man får verkligen är från den avsändare som man antar.
b) Vilken av dessa används av banker?
c) Vilken av dessa används av fria mjukvaruprojekt?
d) Vilken av dessa används av https://elev.rymdgymnasiet.com/?

14.3 Uppdrag granskning i SVT gjorde ett reportage om hur enkelt det var att hacka WEP och avlyssna trafiken i hemanvändares trådlösa nätverk. Där sa hackern som tog sig in i nätverken hos folk ungefär: "Om du nu skulle skicka ditt kreditkortsnummer skulle jag se det direkt här på min dator." Är det så? Om han har rätt vad har då användaren för säkerhetsproblem?

*14.4 Vid skapandet av ett par SSH-nycklar anger användaren en lokal passfras som används till den lokala hemliga nyckeln. Inget lösenord används sedan över nätverket vid inloggning. Vid en SSL-inloggning till en webbsida kopplas den krypterade anslutningen upp och lösenordet för inloggning skickas till webbservern.
a) Vilket är mest tillförlitligt för användaren respektive serverägaren?
b) Varför?

14.5 Vilka saker bör man alltid titta efter när man loggar in till sin bank via en webbanslutning?

14.6 Ange för var och en av följande krypteringsinplementeringar var kryptering respektive dekryptering sker:
a) Secure Sockets Layer (SSL)
b) Secure SHell (SSH)
c) Wi-Fi Protected Access (WPA)
d) Pretty Good Privacy (PGP)
e) Content Scrambling System (CSS)
f) Filmdistribution via Trusted Computing.

14.7 a) Det talas ofta om att man ska se till att välja olika lösenord för alla olika konton man har på nätet. Är det viktigt? Även om man bara använder krypterade anslutningar och håller dem hemliga? Motivera.
b) Vissa webbplatser med inloggningskonton har en funktion där man kan få sitt lösenord skickat till den webbadress man angav när man skapade kontot. Är det säker hantering? Vad säger det om hur informationen hanteras på webbservern?

15 Mobil och radioteknik

Mål Efter dagens lektion är det meningen att ni ska

1. förstå modulering av radiosignaler
2. förstå cellsystemet för mobilnät
3. förstå hur uppkopplingar till telefoner kan separeras

Läs
- Nedanstående

Besvara Instuderingsfrågorna 15.5.

15.1 Utveckling

Mobiltelefoner är i grunden en radiosändare och -mottagare. De första mobiltelefonerna var radioapparater som kommunicerade med ett system med en enda basstation. Samtalen kopplades upp av en telefonist. I det första svenska systemet, Mobiltelefonissystem A, fanns en basstation i Stockholm en i Göteborg och en i Malmö.

Det stora steget för mobiltelefonin kom med cellsystemet som togs fram av AT&T. Med celler menas att det finns flera basstationer och uppkopplingen sker automatiskt till den basstation som har den bästa signalen och byter när man rör sig och kommer till en annan cell.

15.1.1 1G (NMT)

Det första mobilsystemet som fick spridning i Sverige var NMT. Det räknas som första gererationens mobilsystem. Det fanns två olika system: NMT 450 som använde frekvensbandet kring 450 MHz, det var överlägset dagens mobilsystem i räckvidd tack vare att det hade så låg frekvens (lång våglängd) och hög sändareffekt både på basstationerna och telefonerna. Telefonerna var stora och drog mycket ström med den höga effekten. Och NMT 900 “ficktelefonsystemet” med dubbla frekvensen och mindre telefoner. Det var ett helt analogt system och kunde alltså bara överföra ljud.

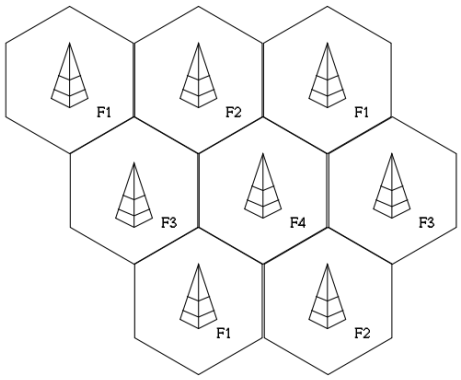
Figur 48: Nokia Talkman, en tidig NMT 450 telefon. Foto: Wikimedia commons användare krystof.k Från Wikimedia Commons.

15.1.2 2G (GSM)

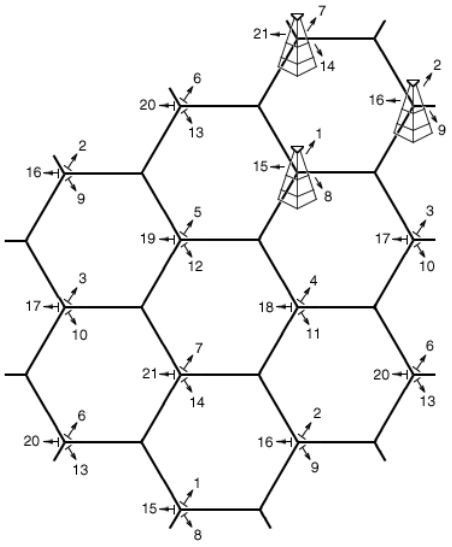
Den analoga tekniken ersattes så småningom med digitala system, vilket betecknar andra generationen. Det fanns ett par olika system varav GSM var det som användes i Sverige. Det digitala systemet gjorde att man kunde skicka med andra digitala signaler än ljud. Snabbt blev SMS mycket populärt. Digitala tjänster för koppling till internet infördes successivt.

15.1.3 3G (UMTS)

Tredje generationens mobiltelefonsystem är det som är vanligast i Sverige idag. Här används UMTS. Det har framför allt mycket större bandbredd än GSM. Under tredje generationen började man använda IP för vissa tjänster. Smartphonen slår igenom.



(a) Cellstruktur med rundstrålande basstationer. Ill.: Wikimedia commons användare Andrew pmk Från Wikimedia Commons.



(b) Cellstruktur med basstationer med 120° sektorer

15.1.4 4G

Fjärde generationens mobiltelefoni kännetecknas av att all trafik, även all ljudöverföring som telefonsamtal sker med IP så telefonsamtalen är i praktiken IP telefoni.

15.2 Celler

Mobilsystem kallas cellphone på engelska för att landskapet är uppdelat i celler där det är en eller flera sändare som har hand om kommunikationen i varje cell. Den enklaste typen är att sätta en rundstrålande antenn i mitten på cellen och låta anslutningen gå till den närmaste anslutningen enligt figur 49a. I praktiken idag så använder man istället antenner som strålar ut i en 120° vinkel och sätter tre sådana sändare på en mast i hörnet mellan tre celler enligt figur 49b, med den konfigurationen samverkar tre sändare om att täcka in en cell. Observera att det går åt precis lika många master i båda konfigurationerna.

Storleken på en cell kan variera. Det finns överföringstekniska gränser för hur långt det kan vara mellan en telefon och en basstation som beror på frekvensen på bärvågen, sändareffekt och miljön som mängden störningar och topografin som kan ge radioskugga etc. Det begränsar hur stora cellerna kan bli i områden med väldigt få användare.

I områden med fler användare är det snarare frekvensutrymmet som begränsar. Varje cell har en begränsad bandbredd, den är i stort sett den samma oavsett hur stort område cellen täcker. De användare som finns i cellen måste däremot dela på den bandbredd som finns så i där det finns många användare gör man mindre celler med sändare som håller lägre effekt för att kunna återanvända samma frekvensområde i en annat område så nära som möjligt. Bilderna med geometriskt homogena hexagonstrukturer är därför ganska förenklad.

15.3 Signalteori

En radiovåg kan användas för att sända över en signal på två i grunden olika sätt. Varje radioöverföring bärs fram med hjälp av en bärvåg. Det är en elektromagnetisk våg av en

(a) En rundstrålande inomhusantenn. Foto: Wikimedia commons användare Karlo Från Wikimedia Commons.

(b) 120° mobilantenner i ett cellhörn. Foto: Manfred Sauke Från Wikimedia Commons.

bestämd våglängd, och därmed bestämd frekvens. Den frekvensen ärd den man normalt ställer in på en radio när man väljer kanal. Sen är det avvikelser från den vågen som är själva signalen. Antingen kan man ändra amplituden på sigalen till omväxlande större och mindre, det kallas amplitudmodulerad (AM) signal se figur 49a. Alternativet är att ändra frekvensen lite grand till en liten aning högre och lägre än bärvågens normalfrekvens, det kallas frekvensmodulering (FM) se figur 49b. Oavsett vilken metod man väljer så måste det finnas tillräckligt med "plats" för signalen att ändras, d v s två olika signaler kan inte ha sina bärvågor på frekvenser så nära varandra att när man ändrar frekvensen så tas den upp av mottagaren som tar upp den andra signalen. Det kallas att signalen tar upp en viss bandbredd. Ju mer information man vill överföra i signalen ju mer måste man ändra den från bärvågen alltså ju större bandbredd behöver man ta upp.

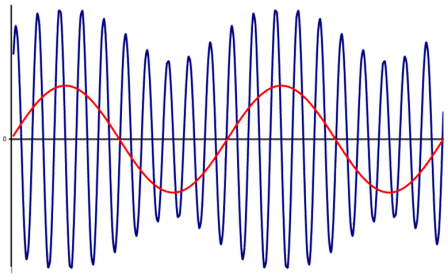
För digitala signaler som ska signalera logiska signaler som är på eller av finns ett tredje alternativ, nämligen att kasta om fasen på bärvågen, alltså göra 180° hopp i signalen eller om man så vill göra den spegelvänd. Det kallas fasmodulering.

Alla tre metoderna för digital signalöverföring av en signal visas i figur

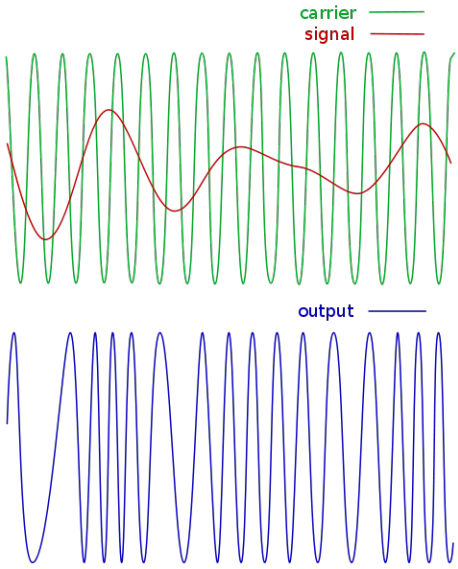
Den svenska matematikern Harry Nyquist visade 1927 att det teoretiskt inte går att överföra mer diskreta signaler än dubbla bandbredden. Om alltså bandbredden är 1 kHz kan man maximalt överföra 2000 bitar/sekund. I praktiken är det svår att komma ens nära den överföringskapaciteten och en mängd olika metoder som är varianter på de tre ovanstående har utvecklats.

15.4 Separering av signaler

För att celler ska kunna fungera brevid varandra så måste signalerna skilja sig på något vis. För både telefonsamtal och andra uppkopplingar inom en cell gäller också att den tillgängliga bandbredden måste fördelas mellan alla telefoner som är i den cell de befinner sig i. För att samtidigt klara av dessa två uppgifter behöver man alltså kunna dela upp signalerna på två sätt. Det finns i praktiken tre olika sätt att göra detta. FDMA, TDMA och CDMA.



(a) Amplitudmodulerad signal. Röd kurva markerar den signal man vill ha ut och blå visar den modulerade utsignalen. Ill.: Wikipedia cs användare Serych Från Wikimedia Commons.



(b) Frekvensmodulerad signal. Röd kurva markerar den signal man vill ha ut, grön bärvågen och blå visar den modulerade utsignalen. Ill.: Wikimedia commons användare Gvf och Gregors Från Wikimedia Commons.

15.4.1 FDMA

Det tekniskt enklaste sättet att dela signaler är att låta två signaler som ska skiljas från varandra använda olika frekvenser. Det kallas Frequency Division Multiple Access (FDMA) Varje signal har en bandbredd och bärfrekvenserna kan inte läggas så tätt att dessa överlappar varandra.

15.4.2 TDMA

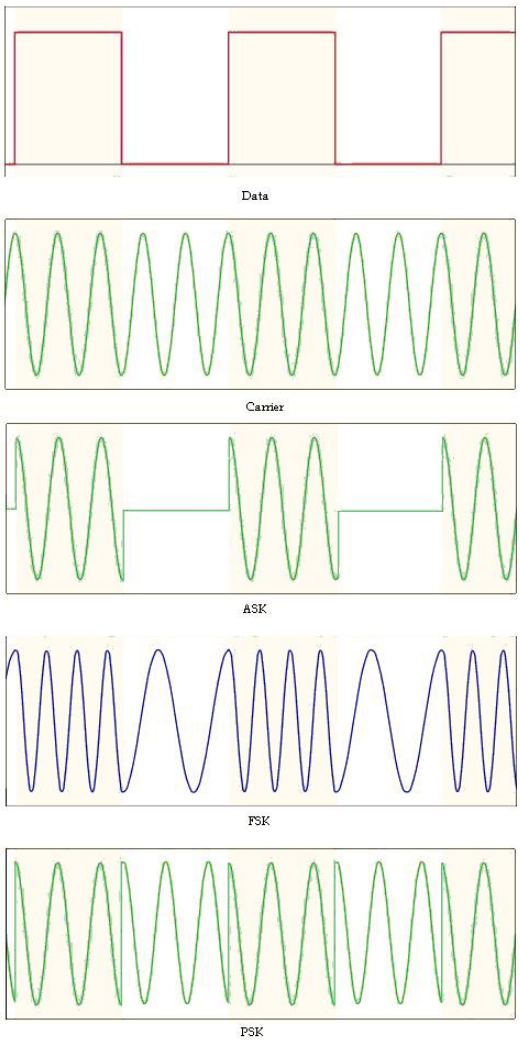
Ett annat sätt att dela signaler åt är att dela dem i tid, alltså låta en signal få ett litet kort tidsintervall på sig att sända, och sedan är det någon annans tur. Det kallas Time Division Multiple Access (TDMA) För talöverföring gäller det att tidsintervallen är korta eftersom man måste förskjuta signalen så lång tid som det är i luckorna.

15.4.3 CDMA

Det tredje sättet som är klart svårast att förstå är Code Division Multiple Access (CDMA). Det går ut på att man kodar signalerna så sändaren och mottagaren är överrens om en kod. Man överlagrar då både koden och signalen till bärvågen. Det gör att bara den mottagare som blandar signalen som kommer med en viss kod kommer att få ut den signal som var blandad med den koden, medans en annan mottagare blandar samma signal med en annan kod och får ut den signal som blandades med den koden. Koderna måste vara mycket snabbare än signalerna som ska lagras på dem och därför tar signalen upp väldigt stor bandbredd.

15.4.4 OFDMA

En annan modern metod både för snabba radiobaserade överföringar som 4G systemen och för snabb överföring av signaler i trådbundna anslutningar som ADSL är Orthogonal



Figur 49: Modulering av en digitalsignal med amplitud, frekvens och fas. Ill.: Wikimedia commons användare B vatovski Från Wikimedia Commons.

Frequency Division Muliple Access (OFDMA) som är en variant på FDMA där man delar upp signalen i många bärvågor, många fler än antalet användare. Varje frekvens i intervallet måste då ha liten bandbredd för att inte störa ut de andra. Kanalerna kan då användas som en parallellöverföring, där antalet kanaler kan fördelas mellan mottagarna. Det visar sig även vara effektivt att göra så här även om det bara finns en mottagare som får alla kanalerna för att störningar ofta drabbar bara en smalt band av frekvenser och i fallet med OFDMA ger det ett ganska litet frånfall att man måste stänga några av kanalerna. Det här är en modern teknik som bygger på att man genom snabba beräkningar kan skilja signalerna.

15.4.5 Fördelning av uppkopplingar

Problemet är alltså att särskilja två saker 1) geografiskt närliggande celler och 2) enskilda telefoner i samma cell, och det finns tre metoder att skilja signalerna åt. Olika telefonsystem har använt olika kombinationer man kan t. ex. skilja cellerna genom att ha olika frekvensband och telefonerna i varje cell med olika koder.

I och med att trafiken blivit IP-baserade i G4 behöver man inte längre skilja ut olika telefoner i cellen från varandra, det finns ju adress i paketen så adresseringen i det systemet fungerar lika som på resten av internet, därmed behöver man bara en avskiljning, att skilja ut signalen från olika basstationer.

15.5 Frågor

.

15.1 Vad finns det för tre tekniker för att tillåta flera sändare att använda ett bestämt frekvensband i samma område?

15.2 Beskriv hur metoderna fungerar.

a) Frekvensmodulering
b) Amplitudmodulering
c) FDMA
d) TDMA

15.3 Hur skiljer sig storleken på celler i ett mobilnät mellan stad och landsbygd och varför är det så?

15.4 a) Vad är bärvågens uppgift för en vanlig analog frekvens- eller amplitudmodulerad radiosignal?
b) Vad kan man säga om bärvågens frekvens i förhållande till signalens, är den högre eller lägre? Förklara varför det måste vara så?
c) Hur förhåller sig signalens frekvens till bandbredden?

15.5 Rita upp hur cellstrukturen ser ut i ett mobilnät med hexagonala celler och sändare som sitter i hörnen mellan tre celler.

15.6 a) Hur stor digital signal går det maximalt teoretiskt att överföra i en signal med 100 Hz bandbredd?
b) Vad hette den svenska matematiker som visade på sambandet?

15.7 Varför vill man gärna ha olika separationsmetoder för att separera olika telefoner från varandra inom en cell respektive celler från varandra i telefonnätet?

16 Instuderingsfrågor 2

Skadlig kod 16.1 Definitionerna för vad som är olika typer av skadlig kod flyter ihop en del.

a) Beskriv skillnaden i spridningssätt mellan maskar och virus.
b) Vad kännetecknar trojaner?
c) Vad gör en sniffer eller keylogger?

16.2 Några sätt att skydda sin egen dator är att använda antivirusprogram, brandvägg och köra virtuella maskiner på datorn.

a) Beskriv vad ett antivirusprogram gör.
b) Beskriv vad en brandvägg gör.
c) På vilket sätt kan det skydda en att man använder en virtuell maskin om man ändå blir angripen av skadlig kod?

16.3 Vilka är de viktigaste sätten nätverksadministratören på en skola eller ett företag kan använda för att försvåra spridningen av skadlig kod?

A Kursmål enligt kursplan

Eleven skall	Behandlas under vecka
• kunna upprätta, underhålla och felsöka seriell och parallell kommunikation mellan persondatorer och deras kringutrustning	48
• kunna upprätta och konfigurera förbindelser med olika typer av modem	
• kunna ansluta, konfigurera och använda digitala tjänster	41–42
• känna till det publika telenätets anslutningsmöjligheter, prestanda, begränsningar och tjänster samt egenskaper för de vanligaste modemstandarderna	37, 45
• ha kunskap om principer och prestanda för olika typer av överförings- transport- och transmissionsprotokoll	35, 37, 38, 45, 46, 48, 49
• känna till begreppen ledningsegenskaper och kapacitet samt deras praktiska innebörd vid datakommunikation med olika överföringsmedier	45
• känna till funktion och prestanda hos olika digitala förbindelser	35, 37, 45, 48, 49
• ha kunskap om gällande datalagstiftning och förstå datasäkerhetens betydelse	35–37, 41–42, 47, 48
• ha kunskap om referensmodeller som beskriver struktur och funktion för datorkommunikation	37, 46
• ha kunskap om de vanligaste protokollen och programmen på internet samt hur de används	37–39, 46
• kunna använda programvaror för enklare diagnostisering, spårning och felsökning på internet	37
• ha kunskap om internets historik, uppbyggnad och kommunikationsteknik	35, 37–39
• ha kunskap om olika sätt att skydda information vid dataöverföringar.	47, 48

B Facit

1.1 a) Ett sätt att koda tecken som används i text till binära siffor.
b) Till fjärrskrivare (telegrafer).

1.2 a) Polska matematiker knäckte den och gav info till brittisk underrättelsetjänst.
b) Svenska matematikern Arne Beurling löste chiffret och Sverige hjälpte Finland under vinterkriget men höll i övrigt denna information hemlig.

1.3 a) Projektilbanor
b) Väderprognoser
c) Kodknäckning
d) T. ex. Zuse (D), von Neuman (USA) och Turing (GB).
e) Zuse Z3 (och Z1 och Z4)(D), ENIAC (USA), Colossus (GB).

1.4 a) USA, sent 60-tal
b) Direkt modemuppkoppling på telefontråd
c) Arpanet
d) Till Internet.

1.5 Av Tim Berners-Lee för forskare på CERN

1.6 a) 1 radiorör, 2 transistor, 3 integrerade kretsar, 4 VLSI
b) 1 hålremsor (stans och skrivare), 2 terminaler, 3, 4 skärm och tangentbord.
c) 1 maskinkod och assembler, 2 Fortran, COBOL, PL1, 3 C, sh, 4 BASIC, Java, C++
d) 1 stordatorn, 2 minidatorn, 3 arbetsstationen, 4 hemdatorn, persondatorn.

1.7 a) Churchill ansåg att ubåtarna var det största hotet under kriget. Båtar är tvugna att kommunicera via radio vilket är lätta att avlyssna. Britterna lyckades komma över både krypteringsnycklar och enigmamaskiner från bekämpade fartyg.
b) Flottan fick en modifierad Enigma med fler rotorer och därmed längre krypteringsnycklar.
c) Man tvingade tyskarna att använda radio istället vilket är mycket lättare att avlyssna. Med maskinerna i Bletchley park kunde britterna snabbt avkoda trafiken och följa tyskarnas respons.

1.8 a) Att avlyssna Tyskarnas trafik.
b) Siemens Halske T 52 Geheimfernschreiber (G-skrivaren).
c) Arne Beurling

2.8 a) Gäller generellt för automatisk behandling. Alt behandlar en levand fysisk person.
b) Offentlighetspricipen, yttrandefriheten.
c) Rent privat behandling av personuppgifter, konstnärlig, journalistisk eller litterär verksamhet, vid myndighetsutövning, för att kunna fullgöra en rättslig skyldighet, för att fullgöra ett avtal med den registrerade, för att en arbetsuppgift av allmänt intresse skall utföras (t.ex. arkivering, forskning och statistik) eller när intresset av behandlingen väger över integritetsrisken.
d) Ta in personernas uttryckliga medgivande.

2.9 a) läraren
b) Att ett verk har tillräcklig egensinnighet för att omfattas av upphovsrätt.
c) 70 år efter upphovsmannens död.
d) A,C,D,F
e) Rätt att anges och rätt till respekt.
f) För datorprogram gäller: Ej kopior för privat bruk, ej uthyrning eller utlåning till allmänheten, skyddad i allmän handling, tillhör normalt arbetsgivaren.

2.10 a) Den som sprider vidare ett verk under copyleft måste fortsätta använda samma licens (även på sina ev modifieringar).
b) Försäkra sig om att man får ta del av vidareutvecklade versioner.
c) Nej, den använder bilder av andra upphovsmän än författaren så den är tvingad till copyleft.

2.11 Bilderna på datorn och RMS omfattas av copyleft CC-BY-SA vilken säger att man måste ackreditera upphovsmannen och tillhandahålla källan till det

verk man använder den i till den som önskar. (Efterfråga den inte är du snäll för det blir en massa jobb för mig att plocka ihop alla filer som behövs.) Till bilden på Bill Gates är upphovsmannen okänd men bilden släppt till public domain av en myndighet. Till brevet uppges upphovsmannen tydligt (ideell rätt) men det är inte omfattat av copyleft så källan behöver inte anges (ekonomisk rätt). Copyleftsymbolen är i public domain för att den inte anses uppnå verkhöjd och därmed behöver inte heller upphovsmannen anges.

3.2 Antalet adresser är mycket större.

3.3 Översätter datoradresser i textform till IP-nummer.

3.4 Datorn som vill ansluta till nätet begär och får ett tillfälligt IP-nummer.

3.5 Det finns en räknare i paketet som räknar passerade routrar, till slut kastas de bort. ICMP tar hand om felhanteringen.

3.6 a) TCP har en invecklad kontrollmekanism som ser till att alla paket i en överföring verkligen kommer fram, eller att avsändaren får felmeddelanden om det inte går (tänk rekommenderat brev på posten). UDP skickar paket med destinationsadress men gör ingen kontroll av överföringen (som vykort).
b) Vanliga filöverföringar, HTML.
c) Tidssignaler andra tidskritiska anslutningar.

3.7 Efter första kolon före första punkten.

3.8 Antingen en numerisk port i ett IP nätverk eller så är det en fysisk serie eller parallellport.

3.9 En process som lyssnar efter anrop på en port.

3.10 En port som bevakas av en demon används av en server som kör på datorn för att ta han om anrop till servern. Andra portar öppnas av program som skickar ut förfrågningar.

3.11 En server kan vara en process som kör på en dator och bevakar anrop på en av datorns portar. Server kan också beteckna en dator som i första hand används för att köra en eller flera serverprocesser.

4.1 a) Webbläsaren
b) Webbläsare är ganska feltåliga och andra webbläsare kan reagera på fel inte syntes i din. Det finns också speciella former av webbläsare som uppläsande läsare för blinda som är mycket mer beroende av att standarden följs.
c) Stora företag och organisationer i IT-brancher och stater.
d) Att sända de filer som besökarna efterfrågar.
e) Antingen exekvera andra program på datorn som servern går på och skicka vidare input till dem eller interpretera programkod för ett språk den har stöd för beroende på vilken teknik som används. Det finns även interaktiv kod som körs av webbläsaren istället t. ex. javascript då blir det ingen skillnad mot statiska sidor för webbservern.

4.2 HTML är en delmängd av SGML medans XHTML är en delmängd av XML. XML är i sin tur en delmängd av SGML. XHTML är alltså striktare än HTML.

4.3 a) Tim Berners Lee
b) W3C, World Wide Web Consortium
c) Tim Berners Lee

6.1 a) Det finns flera. Webbläsare som inte förstår eller inte berörs av alla stilkommandon kan ignorera dem utan risk att förstöra informationsinnehållet. Man kan enklare använda olika stilmallar för olika presentationer. Man kan skriva innehållet i en logisk ordning som fungerar bättre i sammanhang då inte stilen kan användas.
b) W3C

7.1 1. A till utg. mailserver t. ex. SMTP AUTH, A:s dator klient, servern agerar server. Hög port till port 547.

2. Server till server SMTP A:s server agerar klient B:s server, hög port till port 25.
3. B:s inkommande mailserver till B t. ex. IMAP. B agerar klient servern server, hög port hos B port 143 hos servern.

7.2 Den utnyttjas av spammare, kunderna tvingas använda autentiserade protokoll i stället.

11.2 a) 1,07 GB
b) 1070 MB
c) 1024 MiB
d) 8,00 Gib
e) 8,56 Gb

11.2 a) 2,47 GB
b) 2469 MB
c) 2355 MiB
d) 18,40 Gib
e) 19,76 Gb

11.3 a) I parallell går flera bitar samtidigt på olika ledare, på seriell endast en.
b) Längden på anslutningen.
c) Parallell: interna databussar, PATA, SCSI, PCMCIA
Seriell: USB, Firewire, Ethernet
d) Seriella blir snabbare och om kapaciteten räcker är kablarna billigare och smidigare.

11.4 a) Magnetfälten i de öppna loopar som bildas mellan ledarna cancellerar.
b) Olika tvinning mellan paren, skärmning av hela kabeln och skärmning av de individuella paren.

11.5 Metalledare: spänningsvariationer, Optisk: fotoner.

11.6 Det måste vara bestämt vilken av enheterna som ska vara värd (host) och olika kontakter förhindrar att man kopplar ihop två värdar eller två periferienheter.

11.7 a) Ethernet, Firewire 800, Thunderbolt, USB, USB
b) USB 3, VGA, DisplayPort, USB 2
c) 3 × audio, Firewire 400, VGA, 2 × USB, 2 × PS/2, Serieport (DE-9), parallellport (DB-25), Ethernet
d) Mini Displayport, Displayport, USB.

12.1 a) Ja
b) Särskilda sparade nummerserier som 10.10.10.* och 192.168.*.*

12.2 Minst 2, ett till varje nätverk som ska sammankopplas.

12.3 a) Denial of service, överbelastning av en server.
b) Många datorer skickar förfrågningar till samma server samtidigt.
c) Botar, alltså crackade datorer, det kan vara din. ;-)

12.4 a) Att koppla ihop det med andra nätverk.
b) Lilla routern har omadressering med NAT. För stora routern är routingtabellen viktigare.

12.5 a) Bussform, stjärnform, ringform
b) se figurer
c) Buss är relativt enkelt men ger mycket onödig trafik, stjärn är effektivt men behöver, och beror av en central enhet, ringformigt nät är mycket enkelt och billigt men sårbart (beroende av alla ingående noder).
d) stjärn
e) buss

12.6 a) 4
b) 3, 4
c) 192.168.2.0
d) 192.168.2.255
e) 24

12.7 a) 01111011 10100000 01111011 11101010
b) 11111111 11000000 00000000 00000000
c) 10
d) 01111011 10000000 00000000 00000000
e) 123.128.0.0 /10

12.8 Nät 1: nätmask /25, adress 123.123.123.128, broadcast 123.123.123.255
Nät 2: nätmask /25, adress 123.123.123.0, broadcast 123.123.123.127

13.1 a) MAC 48 bitar, IPv4 32 bitar, IPv6 128 bitar.

b) MAC oktetter (Bytes) i hexkod avdelade med -
IPv4 oktetter i decimalform avdelade med .
IPv6 16 bitars block avdelade med :
c) MAC tillverkarnr och kretsnummer
IP nätverksadress och datoradress
d) MAC och IPv6 delas på mitten, IPv4 av nätmasken.

13.2 Stort överflöd av adresser gör nätverksuppdelning och adresstilldelning enklare.

13.3 a) ::1
b) 2001:db8::ff00:42:8329
c) i 2001:db8::ff00:0:42:8329

13.5 Broadcast i IPv4 har ersatts med multicast till grupper i IPv6 och gruppadresserna måste öppnas.

13.6 T. ex. checksum är borttagen, nätmask behövs oftast inte.

13.7 2002:53:96:92:67::

13.8 a) 123.45.123.45
b) 192.71.13.55
c) ::ffff:173.194.44.119
d) ::ffff:adc2:458b

14.1 a) Symmetrisk för alla typer av bulköverföring, assymetrisk för enskilda mindre objekt och framför allt för utbyte eller skapande av symmetriska nycklar.
b) Man kan i princip räkna ut de hemliga nycklarna från den offentliga.
c) Assymmetriska krypteringsnycklar bygger på stora primtal.
d) 1
e) 4

14.2 a) Certifiering av CA, web of trust.
b) CA
c) Web of trust
d) ingen

14.3 Trafiken till banken var helt okrypterad över internet.

14.4 a) SSH för användaren, tveksamt för serverägaren.
b) Användaren har ingen kontroll på hur servern hanteras hans lösenord vid SSL-uppkoppling. Serverägaren är i vilket fall beroende av att användaren hanterar sitt lösenord diskret, båda metoderna kan dessutom lagra lösenorden lokalt. Det gäller för serverägaren att se till att ingen har mer rättigheter än nödvändigt på servern efter inloggning.

14.5 Att webbadressen är riktig, att det går över en krypterad anslutning och att den är certifierad av en CA.

14.6 a) Webläsaren och webservern.
b) SSH-klientprogrammet och SSH-demonen.
c) Nätverkskortet och accesspunkten.
d) Av speciella program, ofta inbakat i mailprogram.
e) Före tillverkning av DVDn och i DVD-spelaren.
f) Filmbolaget och i skärmen eller projektorn.

14.7 a) För webbinloggningar används SSL där lösenordet skickas över till serversidan. Som användare har man ingen kontroll hur lösenorden hanteras på serversidan.
b) Att de i praktiken lagrat lösenordet i klartext.

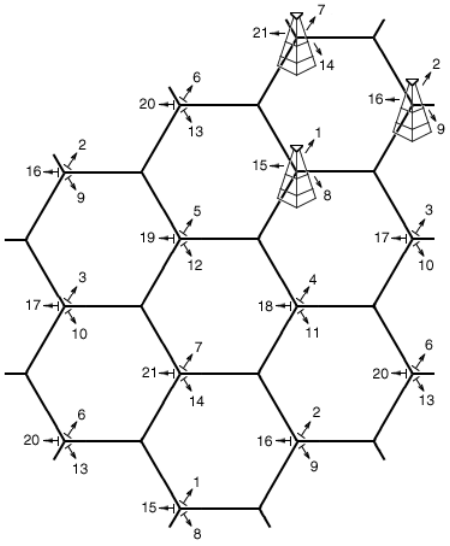
15.1 FDMA, TDMA, CDMA

15.2 a) Signalen får ändra frekvensen på bärvågen
b) Signalen får ändra amplituden på bärvågen
c) Frekvensbandet uppdelas så att varje användare får en smalare delband.
d) Användarnas signaler delas upp i tidsfönster efter varandra.

15.3 Cellerna är geografiskt mindre i staden för att antalet användare är begränsat i en cell.

15.4 a) Att ge en stabil våg som sändaren och mottagaren kan stämmas av för att komma i självsvängning kring.
b) Bärvågens måste vara betydligt högre. Den överlagrade signalen måste domineras av en klar grundfrekvens.

c) Ju högre frekvens signalen har ju större blir bandbredden.



15.5

15.6 a) max 200 bps
b) Harry Nyqvist

15.7 För att kunna återanvända mönstret i närliggande celler.

16.1 a) Maskar sprider sig aktivt genom att söka svagheter hos andra datorer i nätverket. Virus sprids genom infekterade filer.
b) Trojaner sprids genom att de är inbäddade i ett annat program som användaren luras att installera.
c) Lagrar alla tangenttryckningar som görs på datorn.

16.2 a) Söker igenom filer, antingen inkommande till datorn eller innehållet på hårddisken, efter kända instanser av skadlig kod. Kräver en uppdaterad lista.
b) Brandväggens grundprincip är att reglera vilka portar som är öppna på datorn.
c) Det är mycket enkelt att backa upp den virtuella maskinen genom att den är en enda stor fil på disken. Man kan alltså spara en backup på sin virtuella maskin innan man startar den och starta om från den versionen om man märker att det händer något mystiskt. Man kan också starta en ny virtuell maskin när man ska göra något som man tror innebär större risker än normalt.

16.3 Dela upp nätet i avgränsade subnät med brandväggar. Skanna epost med antivirus.

Akronymer

ADSL Asymmetric Digital Subscriber Line

AGP Accelerated Graphics Port

AMD American Micro Devices

AM amplitudmodulering

APT Advanced Package Tool

API Application Programming Interface

ARP Address Resolution Protocol

AT&T American Telephone & Telegraph

BIOS Basic Input/Output System

BARK Binär Aritmetisk ReläKalkylator

BASH Bourne Again SHell

BESK Binär Elektronisk SekvensKalkylator

BSD Berkeley Software Distribution

CA Certificate Authority

CD Compact Disc

CDMA Code Division Multiple Access

CERN Conseil Européen pour la Recherche Nucléaire

CGI Common Gateway Interface

CIDR Classless Inter-Domain Routing

CLI Command Line Interface

CPU Central Processing Unit

CSS Cascading Style Sheets

CSS Content Scrambling System

CVS Concurrent Versions System

DHCP Dynamic Host Configuration Protocol

DLL Dynamic Link Library

DNS Domain Name Server

DOS Disc Operating System

DoS Denial of Service

DP Display Port

DRM Digital Rights Management

DSL Digital Subscriber Line

DVD Digital Video Disc

DVI Digital Visual Interface

EMACS Editor MACroS

ENIAC Electronic Numerical Integrator And Calculator

ERA Engineering Research Associates

EULA End-User License Agreements

EV Extended Validation

FDMA Frequency Division Multiple Access

FM frekvensmodulering

FSB Front Side Bus

FSF Free Software Foundation

FTP File Transfer Protocol

FTP Foil Twisted Pair

GB Gigabyte

GFDL GNU Free Documentation License

GNU Gnu's Not Unix

GPG GNU Privacy Guard

GPL General Public License

GPRS General Packet Radio Service

GRUB GRand Unified Bootloader

GSM Groupe Spécial Mobile

GUI Graphical User Interface

HDCP High bandwidth Digital Content Protection

HDMI High-Definition Multimedia Interface

HTML Hypertext Markup Language

HTTP Hyper Text Transfer Protocol

HTTPS Hyper Text Transfer Protocol Secure

KDE Kool Desktop Environment

IANA Internet Assigned Numbers Authority

IBM International Buisness Machines

ICANN Internet Corporation for Assigned Names and Numbers

ICMP Internet Control Message Protocol

ICPL Initial Control Program Load

IEEE Institute of Electrical and Electronics Engineers

IETF Internet Engineering Task Force

IMAP Internet Message Access Protocol

IP Internet Protocol

IRC Internet Relay Chat

ISA Industry Standard Architecture

ISP Internet Service Provider

LAMP Linux Apache MySQL PHP/Perl/Python

LAN Local Area Network

LGPL Lesser General Public License

MCA Micro Channel Architecture

MIME Multipurpose Internet Mail Extensions

MIT Massachusetts Institute of Technology

MS Microsoft

NAT Network Allocation Table

NFS Network File System

NMT Nordiskt mobiltelefonisystem

NTP Network Time Protocol

OFDMA Orthogonal Frequency Division Muliple Access

OS Operativsystem

PATA Parallel Advanced Technology Attachment

PCI Peripheral Component Interconnect

PCMCIA Personal Computer Memory Card International Association

PD Public Domain

PDF Portable Document Format

PGA Pin Grid Array

PGP Pretty Good Privacy

PHP PHP Hypertext Preprocessor

POP Post Office Protocol

POSIX Portable Operating System Interface for UNIX

PPP Point to Point Protocol

PS Postscript

PUL personuppgiftslagen

RAID Redundant Array of Indepenent Discs

RAMAC Random Access Method of Accounting and Control

RMS Richard Stallman

SAAB Svenska Aeroplan AktieBolaget

SATA Seriall Advanced Technology Attachment

SCP Secure Copy

SCSI Small Computer System Interface

ScTP Screened Twisted Pair

SGML Standard Generalized Markup Language

SMS Short Message System

SMTP Simple Mail Transfer Protocol

SMTP AUTH SMTP Authentication

SOPA Stop Online Piracy Act

SQL Structured Query Language

SSD Solid State Drive

SSH Secure SHell

SSL Secure Sockets Layer

STP Shielded Twisted Pair

SVN Subversion

TCP Transmission Control Protocol

TDMA Time Division Multiple Access

TLS Transport Layer Security

TP Twisted Pair

UDP User Datagram Protocol

UMTS Universal Mobile Telecommunications System

URL Uniform Resource Locator

USB Universal Serial Bus

UTP Unshielded Twisted Pair

VGA Video Graphics Array

VI VIsual editor

VLSI Very Large Scale Integration

VPN Virtual Private Network

W3C World Wide Web Consortsium

WAN Wide Area Network

WEP Wired Equivalent Privacy

WPA Wi-Fi Protected Access

WWW World Wide Web

WYSIWYG What You See Is What You Get

XHTML eXensible Hypertext Markup Language

XML eXensible Markup Language

Referenser

Prestidge-King, C., The battle of crete: A re-evaluation, *Cross-sections: The Bruce Hall Academic Journal*, *III*, 117–131, 2007.

Rojas, R., How to make zuse's z3 a universal computer, *Annals of the History of Computing, IEEE*, *20*(3), 51–54, doi:10.1109/85.707574, 1998.