



Datavetenskap

---

**Christofer Olsson**

**Carl Hellström**

# **Installation och konfiguration av en router**

---

Examensarbete, C-nivå

2002:23



# **Installation och konfiguration av en router**

**Christofer Olsson**

**Carl Hellström**



Denna rapport är skriven som en del av det arbete som krävs för att erhålla en kandidatexamen i datavetenskap. Allt material i denna rapport, vilket inte är vårt eget, har blivit tydligt identifierat och inget material är inkluderat som tidigare använts för erhållande av annan examen.

---

Christofer Olsson

---

Carl Hellström

Godkänd, 2002-06-06

---

Handledare: Niklas Nikitin

---

Examinator: Stefan Lindskog



## **Sammanfattning**

Detta är ett examensarbete som handlar om server-verksamhet för hemmabruk. Syftet var att utreda vilket operativsystem som lämpar sig bäst med avseende på, kostnad, säkerhet samt funktionalitet för ett mindre nätverk. Det finns även med en guide som beskriver hur man installerar och konfigurerar det operativsystem vi valde, och en beskrivning om varför vi valde just detta. Dessutom utförde vi ett antal tester som kan ge en uppfattning om systemets möjligheter och begränsningar. Operativsystemet vi valde att installera var OpenBSD 3.0. Vi testade operativsystemet på tre olika områden, som router, som webbserver och som ftp-server.

# **Installation and configuration of a router**

## **Abstract**

This is a bachelor's project that deals with server related issues for the home. The purpose was to investigate which operating system that was most qualified regarding price, security and functionality for a small network. The report also contains a guide on how to install the chosen operating system and how to configure it, and a discussion of why we considered that operating system to be the most appropriate for our purpose. We performed a number of tests that can give a picture of opportunities and limits of the system. The operating system we chose to install was OpenBSD 3.0. We tested the operating system in three different areas, as a gateway, as a webserver and as a ftp-server.



## **Tack**

Vi vill tacka alla som givit sina datorer och tillbehör till oss för att vi skulle kunna genomföra detta arbete. Våra vänner lånade ut allt från datorer, skärmar och tangentbord, till nätverksutrustning och nätverkskort. Ett särskilt tack vill vi ge Olsson själv som lånat ut sin övervåning som arbetsplats under dessa intensiva och lärofyllda månader. Karlstad Universitet har också bidragit med utrustning, en 200 Mhz Pentium-dator, och en skärm. På dataskroten har vi dessutom funnit tre skärmar. Dataenheten ska ha ett tack för ett extra IP-nummer (130.243.31.229) till lägenheten på Campus där vi mestadels har jobbat med arbetet. Niklas Nikitin ska ha ett stort tack som handledare för att givit oss teknisk rådgivning och kommentarer under hela arbetets gång.

# Innehållsförteckning

<b>1</b>	<b>Inledning .....</b>	<b>1</b>
1.1	Syfte .....	1
1.2	Vad är en router .....	2
1.3	Översikt.....	2
<b>2</b>	<b>Bakgrund .....</b>	<b>2</b>
2.1	Val av operativsystem?.....	3
2.1.1	Windows NT/2000/XP .....	4
2.1.2	Linux RedHat.....	5
2.1.3	Linux Mandrake.....	5
2.1.4	Linux Debian.....	5
2.1.5	Linux Slackware .....	5
2.1.6	Unix NetBSD .....	6
2.1.7	Unix FreeBSD.....	6
2.1.8	Unix OpenBSD .....	6
2.2	Slutsats förstudie.....	7
<b>3</b>	<b>Konstruktion.....</b>	<b>8</b>
3.1	Installation .....	10
3.2	Applikationer .....	10
3.2.1	Network Address Translation .....	10
3.2.2	Paketfilter .....	12
3.2.3	Apache Webbserver .....	13
3.2.4	Secure Shell SSH .....	13
3.2.5	File Transfer Portocoll FTP .....	13
3.2.6	Network File System NFS .....	14
3.2.7	Dynamic Host Configuration Protocol DHCP .....	14
3.2.8	Qadsl.....	14
<b>4</b>	<b>Test .....</b>	<b>15</b>
4.1	Test av Webbserver .....	15
4.1.1	Test av en 100 megabyte fil.....	16
4.1.2	Test av en 10 megabyte fil.....	16
4.1.3	Test av inläsning av hemsida.....	16
4.1.4	Slutsats från webbserver testet .....	17
4.2	Test av ftp-server .....	17
4.2.1	Test med 100 stycken små filer .....	17
4.2.2	Test av överföring av fil på 100 megabyte.....	18
4.2.3	Test av överföring av liten fil 10 MB.....	18
4.2.4	Slutsats från ftp-server testet .....	18

4.3	Överföringshastighets test med router .....	19
4.3.1	Test av överföring av fil på 600 megabyte .....	19
4.3.2	Test skicka 10 megabyte fil .....	20
4.3.3	Test av överföring 1000 filer .....	20
4.3.4	Slutsats av router testet .....	20
4.4	Applikationstest på routern .....	21
4.4.1	Test av NAT .....	21
4.4.2	Test av PF .....	22
4.4.3	Test av DHCP .....	22
4.5	SSH test .....	22
4.6	NFS test .....	23
4.7	Qadsl test .....	23
<b>5</b>	<b>Erfarenheter och rekommendationer .....</b>	<b>23</b>
<b>6</b>	<b>Slutsats .....</b>	<b>24</b>
	<b>Ordlista.....</b>	<b>25</b>
	<b>Referenser .....</b>	<b>26</b>
<b>A</b>	<b>Installation av OpenBSD .....</b>	<b>27</b>
A.1	Hårdvara .....	27
A.2	Startdiskett .....	27
A.3	Uppdelning av hårddisk till partitioner .....	28
A.4	Konfigurering av nätverket .....	29
A.5	Val av installationspaket .....	32
<b>B</b>	<b>Konfiguration .....</b>	<b>33</b>
B.1	Konfiguration av paketfiltret PF .....	33
B.2	Konfiguration av Network Address Translation (NAT) .....	37
B.3	Konfiguration av Apache .....	38
B.4	Konfiguration av FTP .....	40
B.5	Konfiguration av NFS .....	41
B.6	Konfiguration av DHCP .....	41
B.7	Konfiguration av Qadsl .....	43
<b>C</b>	<b>Tabeller .....</b>	<b>45</b>
<b>D</b>	<b>Diagram.....</b>	<b>52</b>

## Figurförteckning

Figur 3.1: Tänkt utseende på Campus-nätverket	9
Figur 3.2: Tänkt utseende på adsl-nätverket	9
Figur 3.3: Network Address Translation exempel	11
Figur 4.1: Webbserver test	16
Figur 4.2: Ftp-server test beskrivning	17
Figur 4.3: Sammakoppling med router	19
Figur 4.4: Sammankoppling med switch	19
Figur A.1: Partitioneringsfråga	28
Figur A.2: Partitionering av Campusroutern	29
Figur A.3: Nätverskkonfiguration för Campusroutern	30
Figur A.4: Nätverskkonfiguration av webbservern	30
Figur A.5: Nätverskkonfiguration av ftp-servern	31
Figur A.6: Nätverskkonfiguration av Telia-routern	31
Figur A.7: Nätverskkonfiguration av DNS	31
Figur B.8: Förklaring av pass-regeln i paketfiltret	33
Figur B.9: Förklaring av block-regeln i paketfiltret	34
Figur B.10: Förklaring av ordningsföljden i paketfiltret	34
Figur B.11: Förklaring av quick-regeln i paketfiltret	34
Figur B.12: Blockering av felaktiga IP-adresser	35
Figur B.13: Öppning av portar i paketfiltret	35
Figur B.14: Förklaring av log regeln i paketfiltret	35
Figur B.15: Förklaring av kommandot <code>tcpdump</code>	36
Figur B.16: Förklaring av protokolltyper i paketfiltret	36
Figur B.17: Förklaring av scrub kommandot	36
Figur B.18: Förklaring av namnsättning i paketfiltret	37
Figur B.19: Förklaring av nat-syntaxen	37
Figur B.20: Förklaring av rdr-syntaxen	38
Figur B.21: Personliga hemsidor	39
Figur B.22: Förklaring av ftp-flaggor	40
Figur B.23: Konfiguration av <code>dhcpcd.conf</code>	42
Figur B.24: Konfiguration av <code>hostname.conf</code> för DHCP	43
Figur B.25: Konfiguration av <code>dhclient.conf</code>	43
Figur B.26: <code>Qadsl.conf</code> beskrivning	44

## Tabellförteckning

Tabell 2.1: Specifikation över respektive operativsystem	4
Tabell 2.2: Överblick av vanliga tjänster till respektive operativsystem	7
Tabell 2.3: Överblick av vanliga nätverkstjänster till respektive operativsystem	7
Tabell A.1: Hårdvara som finns i våra datorer	27
Tabell A.2: Installationspaket till OpenBSD 3.0	32
Tabell B.3: Övriga filer för ftp-servern	40
Tabell C.1: Webbservertest	45
Tabell C.2: Ftp-servertest	47
Tabell C.3: Routerstest	49
Tabell C.4: Direktuppkopplingstest	51

## Diagramförteckning

Diagram D.1: Webbservertest .....	52
Diagram D.2: Ftp-servertest, uppladdning.....	53
Diagram D.3: Ftp-servertest, nedladdning .....	53
Diagram D.4: Routertest, nedladdning .....	54
Diagram D.5: Routertest, uppladdning .....	54
Diagram D.6: Test utan router, nedladdning .....	55
Diagram D.7: Test utan router, uppladdning .....	55
Diagram D.8: Jämförelse mellan router och switch, nedladdning .....	56
Diagram D.9: Jämförelse mellan router och switch, uppladdning.....	56

# 1 Inledning

Vi har själva valt en uppgift att implementera och testa en router för hemmabruk samt någon form av brandvägg till denna. I dagsläget har de flesta bredband hemma i något utförande. Ofta vill man koppla in flera datorer till samma anslutning, och dessutom är det idag nästan ett krav på att man använder sig av en brandvägg. Om internetleverantören inte kan erbjuda mer än en IP-adress är det omöjligt att ansluta mer än en dator till det externa nätverket. I detta fall är den enda möjligheten att använda sig av en router om man vill ansluta flera datorer. Det finns färdiga routrar att köpa men vi har valt att installera och konfigurera en vanlig persondator (PC) för detta syfte. Valet är baserat på att det i dagsläget ofta finns äldre datorer som inte används och att dessa datorer fungerar utmärkt för ett sådant ändamål. Vidare krävs att man använder sig av ett operativsystem på denna dator som ska tillhandahålla stöd för funktioner som krävs för att routern ska fungera på ett korrekt sätt. Vi har studerat några operativsystem till PC-datorer som har stöd för de krav som ställs. Dessa har vi analyserat för att slutligen välja ett för installation och test. Analysen bygger på de grundläggande kraven man kan ställa på en sådan server när det gäller funktionalitet, säkerhet och kostnad. Kostnadsfrågan är dock mycket avgörande för vårt val för att prisbilden inte ska bli hög. Anledningen till att priset spelar stor roll är att vår ekonomi inte tillåter höga kostnader då arbetet inte är ett projekt som universitetet eller något företag har beställt. Utredningsarbetet sträcker sig även till krav på hårdvaran i datorn samt i mycket liten utsträckning på användarvänligheten i operativsystemet.

## 1.1 Syfte

Anledningen till val av examensarbete bygger på vår nyfikenhet att förstå och göra det möjligt att skapa en router. Vi har också haft stort intresse för hur detta kan konstrueras på ett sätt som inte är kostsamt för en student med pressad ekonomi. Stor del ligger också i att hitta ett operativsystem som klarar de krav vi ställer. I stora drag gäller alltså vårt arbete att testa det valda systemet på ett sätt som avslöjar brister och svagheter som ska sammanställas till en slutgiltig bedömning av systemet. Dock är det inte alls säkert att vi finner några svagheter som vi har kunskap att utreda eller förstå. Vi vill också få en inblick i hur man i ett mindre nätverk för hemmabruk kan få en bra ordning med avseende på struktur, säkerhet och funktionalitet baserat på operativsystemet vi väljer. Dokumentationen ska även vara skriven

så att den ska kunna användas till viss del som manual för användare som vill installera vårt val av system. Vidare kommer vi att installera olika typer av applikationer som kan vara till nytta i ett system som riktar sig till hemmets användningsområden. Med detta menas applikationer som till exempel en webserver. De olika applikationstyperna vi har använt beskrivs också med avsnitt som visar hur dessa kan installeras och konfigureras.

## **1.2 Vad är en router**

Förenklat så kan man se en router som en dator som delar ut internet till andra datorer. I och med det blir datorerna, som tilldelas internet, klienter till routern. Datorn som är router kan tilldela internetuppkopplingar på två sätt. Antingen tilldelas klientdatorerna ett statsikt IP-nummer eller så tilldelas klientdatorerna ett dynamiskt IP-nummer. Routern tar hand om all inkommande och utgående trafik, och dirigerar trafiken till rätt klient. En router använder sig av två nätverkskort. Ett för att ta emot och skicka data internt, och ett för att ta emot och skicka data externt.

## **1.3 Översikt**

Vi kommer i denna uppsats att studera de olika moment som ingår när man installerar en router för bruk i hemmet eller i ett mindre nätverk. En rapport om hur installationen kan se ut kommer även att redovisas. Denna rapport bygger på slutsatser vi har funnit under installationen av operativsystemet samt tester som utförts på systemet.

## **2 Bakgrund**

När detta examensarbete blev ett faktum var våra erfarenheter av systemuppbyggnad med datanätverk mycket begränsade. Det första naturliga steget var att påbörja en omfattande inläsning på området. Detta innebar att undersöka olika operativsystem som kan användas för en router. Att hitta operativsystem som klarar av att användas som router gjordes mestadels med hjälp av informationssökningar på internet. Många av de operativsystemen vi hittade slopades direkt utan att vi kommer att namnge dessa, då de inte var lämpliga eller omöjliga att implementera på en PC-dator. Den stora anledningen till att vi valde denna uppgift är att vi saknar kontroll över våra egna nätverk i hemmet. Kapitlet kommer att utreda och presentera



olika varianter av de operativsystem vi fann lämpliga att använda till våra behov. Denna utredning kommer att ligga till grund för det val av operativsystem vi kommer att göra.

## **2.1 Val av operativsystem?**

Uppgiften var att bygga ett nätverk, med fokus på en router. Det första valet man då stöter på är vilket Operativsystem man ska välja. Det finns två huvudkategorier, Windows NT/2000/XP och Unix/Linux. För oss och många andra som ska installera en router för hemmabruk är det viktigt att komponenterna är billiga. Vi undersökte därför ett antal olika operativsystem för att avgöra vilket som var mest lämpat för våra krav. Vi hittade en mängd möjliga alternativ och började med att undersöka vad de olika hade att erbjuda och vilken prisbild de hade. Nedan följer en beskrivning av de olika alternativen som finns att välja mellan. Vi har även sammanställt en tabell, se tabell 2.1, över de olika operativsystemen med avseende på systemkrav och kostnad. Även användbarhet har vägts in, men då detta inte är den viktigaste uppgiften för en server är detta ingen avgörande egenskap. Att avgöra vilken säkerhet de kan erbjuda är svårt och vi kan inte avgöra det korrekt då vi inte har den kunskap som krävs för sådana utlåtanden. Dock undersökte vi problem som de har haft tidigare samt om olika uppdateringar har tillkommit för att säkerheten ska ökas och gjorde vår bedömning med avseende på säkerheten utifrån detta.

Operativsystem	Systemkrav enligt tillverkare	Kostnad
Windows NT	Pentium processor, 32	Server Enterprise : \$3,999 Server 4.0 (standard): \$1,129 Server 4.0 (academic): \$679
Windows 2000	Pentium II 233 eller snabbare med 64 MB Ram, 500 MB hårddisk	Advanced Server: \$3,999 Server: \$999 Professional: \$319
Windows XP	Pentium II 300 Mhz, 64 MB ram, 1,5 Gb hårddisk	Home edition: \$122 Professional: \$215
Linux Redhat	Rekommenderat: Pentium eller snabbare processor med 64 MB RAM och 1.2 GB Hårddisk Minimum:X86 Processor med 32 MB RAM och 650 MB Hårddisk	Free
Linux Mandrake	Pentium processor eller kompatibel Minst 32 MB RAM	Free
Linux Debian	386 eller högre, 16 MB ram, 65 MB hårddisk	Free
Linux Slackware	Intel 386, 16 MB ram, 500 MB hårddisk	Free
Unix OpenBSD	486 processor eller snabbare. Minst 8 MB Ram, ca 100 Mb hårddisk	Free
Unix NetBSD	486 processor eller snabbare. Minst 8 MB Ram, ca 150 Mb hårddisk	Free
Unix FreeBSD	486 processor eller snabbare. Minst 8 MB Ram, ca 150 Mb hårddisk	Free

*Tabell 2.1: Specifikation över respektive operativsystem*

### **2.1.1 Windows NT/2000/XP**

Valet vi gjorde påverkades av kostnad, säkerhet och användbarhet. Alla versioner av Microsoft Windows kostar pengar och då är det svårt att motivera varför dessa ska användas när det finns gratisalternativ. Dock har Microsoft Windows en bra support som många företag föredrar, vilket är orsaken till varför Windows är vanligast på företag. I många av de billigare distributionerna av Microsoft Windows finns det inte alla nödvändiga tjänster som behövs för att bygga upp en router, utan man måste installera och köpa tillägsprogram för detta. Man måste nog säga att Microsoft erbjuder grafiskt stiliga operativsystem och dessa är ganska lättarbetade och förståeliga. Detta har dock också ett pris, man behöver kraftigare hårdvara än i andra operativsystem. Säkerhetsuppdateringar är något som Microsoft ganska frekvent släpper till sina produkter vilket man kan ha i åtanke om man nu vill välja dessa system då detta kan betyda att det finns relativt många fel i deras system.

### **2.1.2 Linux RedHat**

RedHat är en den största och mest använda distributionen av Linux. RedHat är helt gratis och har öppen källkod. Vill man ha Cd-skivor och manual får man dock betala en viss summa. Till skillnad från rena Unixsystem och vissa Linux-distributioner så ska RedHat vara minst lika lätt att installera som många av Microsoft Windows utgåvor. RedHat har också stöd för den mesta hårdvaran. I en standard installation som tar uppåt 500 MB så ingår många program för serververksamhet, till exempel webb-, ftp-, och fildelnings-server. Ett grafiskt gränssnitt är med som standard och även en webbläsare. RedHat är klart för serververksamhet och ska även kunna fungera som arbetsstation.

### **2.1.3 Linux Mandrake**

Linux Mandrakes idé är precis som för RedHat att saluföra sig som ett operativsystem som ska kunna användas för många olika syften. Den ska kombinera Linux flexibilitet med Grafisk design, och det finns många guider för att underlätta. Vid till exempel installation av Mandrake väljer man om den ska bli en nätverksserver eller om den ska bli en arbetsstation. En egenskap som utmärker Mandrake lite extra är att det finns en mini-installation på endast 65 MB. I en standard installation finns det en möjlighet att välja till flera nätverksverktyg, kontorsprogram samt multimedia-applikationer. Säkerhetsaspekten i både Mandrake- och RedHat-systemen är god och de innehåller oftast bra och lättanvända verktyg.

### **2.1.4 Linux Debian**

Linux Debian är en fri version som folk runt om i världen gemensamt utvecklat. Deras grundfilosofi är att koden ska vara öppen (Open Source) så att koden ska kunna förbättras av alla, och att ett bra system ska komma alla till godo. Fel i systemet ska inte döljas utan insyn ska kunna ges i alla felrapporter. Det ska också vara lätt att genom bara några få kommandon uppdatera hela systemet. Till Linux Debian finns det många paket att fritt ladda ned från deras hemsida som innehåller olika program man kan tänkas behöva.

### **2.1.5 Linux Slackware**

Linux Slackware är en av de mest avancerade distributionerna av Linux. Denna distribution riktar sig till de användare som vill ha fullständig kontroll över sitt system. Det som i andra Linux-distributioner kan konfigureras automatiskt, till exempel nätverkskort, måste man i Slackware konfigurera helt själv. Detta gör att användaren har större kontroll på sitt system och kan därmed få en bra överblick på de tjänster den vill erbjuda. Denna version är skapad för att fungera för alla typer av användningsområden. Allt från webb-, ftp- och mail-server till

brandvägg. Det ska även gå bra att använda operativsystemet som klientdator. Säkerhetsmässigt verkar det vara bra och det finns mycket hjälp att tillgå som till alla Linux-system. Linux Slackware är den av Linux-distributionerna som är mest lik Unix. Men den följer ändå Linux-standarden som till exempel Linux filsystem.

### **2.1.6 Unix NetBSD**

Unix NetBSD inriktar sig på att vara portabelt till många olika plattformar och kan dessutom emulera många andra olika system. NetBSD släpps i tre olika versioner. En version som är väl beprövad på det olika systemen tillsammans med olika program. Denna version kommer ut periodvis. En annan version som släpps är en daglig version som åtgärdar de senaste buggarna. Den sista versionen är en version som uppdateras varje natt, för dem som inte kräver väl beprövad säkerhet i alla lägen utan vill ha den senaste programvaran i stället.

### **2.1.7 Unix FreeBSD**

Unix FreeBSD bygger precis som OpenBSD och NetBSD på Unix BSDs kärna. FreeBSD inriktar sig på avancerad tillämpning för nätverk. FreeBSD har avancerad minneshantering och även ett avancerat diskhanteringssystem. För att få program mer effektiva utnyttjar FreeBSD en metod som möjliggör att operativsystemets kärna tar hand om applikationers funktionellitet. Det ska även va möjligt att köra program skrivna för Linux och övriga Unix distributioner under FreeBSD. Alla versioner av X-Windows ska fungera under FreeBSD.

### **2.1.8 Unix OpenBSD**

Alla BSD-distributioner är fria för nedladdning från internet och all dokumentation finns på respektive distributions hemsida. Support finns i form av forum och FAQ, och en viss del av online-hjälp finns att få. I en standard installation av dessa operativsystem så ingår alla grundläggande krav man kan ha för att använda datorn som en server. Men till skillnaden från Linux är Unix OpenBSD inte lika lättanvänt och lämpar sig ofta inte lika bra som arbetsstation. OpenBSD inriktar sig precis som FreeBSD på nätverkshantering och säkerhet, men ska också fungera på många plattformar med låga systemkrav. Användare och utvecklare ska kunna följa utvecklingsarbetet direkt på OpenBSD's hemsida [3]. OpenBSD inriktar sig på att vara nummer ett inom säkerhet, och fixa buggar och säkerhetsbrister innan de upptäcks. OpenBSD ska följa olika standards som ANSI, POSIX och X/Open. Kryptering ska vara integrerat i systemet.

## 2.2 Slutsats förstudie

Vi har sammanställt en tabell med vanliga tjänster, se tabell 2.2, och vanliga nätverkstjänster, se tabell 2.3, för att guida till val av operativsystem.

Operativsystem	Online manual	Support	Öppen källkod	Grafiskt gränssnitt
Windows NT/2000/XP	Ja/delvis	Ja	Nej	Ja
Linux Red Hat	Ja	Ja	Ja	Ja
Linux Mandrake	Ja	Ja	Ja	Ja
Linux Debian	Ja	Nej/FAQ + forum	Ja	Ja
Linux Slackware	Ja	Nej/FAQ + forum	Ja	Ja
Unix Open/Net/Free BSD	Ja	Nej/FAQ + forum	Ja	Nej/Nej/Ja

Tabell 2.2: Överblick av vanliga tjänster till respektive operativsystem

Operativsystem	Fil-server	ftp-server	Mail-server	Webb-server	Router
Windows NT/2000/XP	Ja	Nej	Nej	Nej/Nej/Ja	Delvis
Linux Red Hat	Ja	Ja	Ja	Ja	Ja
Linux Mandrake	Ja	Ja	Ja	Ja	Ja
Linux Debian	Ja	Ja	Ja	Ja	Ja
Linux Slackware	Ja	Ja	Ja	Ja	Ja
Unix Open/Net/Free BSD	Ja	Ja	Ja	Ja	Ja

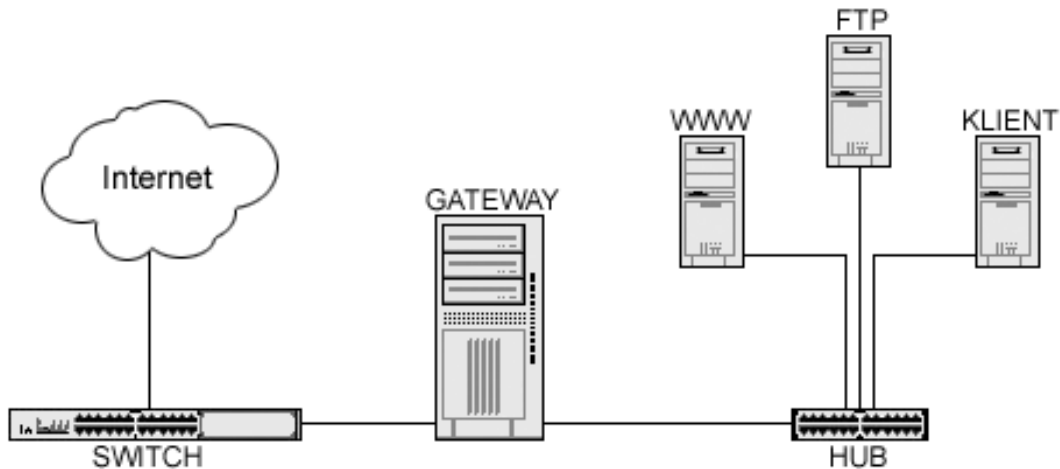
Tabell 2.3: Överblick av vanliga nätverkstjänster till respektive operativsystem

Vi valde att slopa Microsofts produkter då alla dessa kostar pengar, något som vi inte har i detta projekt. Microsoft erbjuder dock det absolut mest lättanvända operativsystemet vilket kan göra det till en vinnare om man har pengar och inte bryr sig om att de släpper många säkerhetsrelaterade uppdateringar till sina produkter. Microsoft har mycket bra support till alla sina operativsystem. Linux-distributionerna valdes bort på grund av att vi inte fann dem lika spännande som Unix BSD-distributionerna. Vi valde att titta lite mer på säkerheten i OpenBSD, då de andra Unix BSD-distributionerna inte hade fått lika mycket beröm med avseende på säkerhet och stabilitet som OpenBSD av kritiker på internet. Säkerhetsföreskriften till OpenBSD bygger på fullständig öppenhet när det gäller säkerhetsfrågor och problem. Alla eventuella säkerhetsrisker eller buggar kan rapporteras in

till en särskild säkerhetsgrupp som på heltid granskar och kommer med uppdateringar till systemet. Eftersom koden är öppen kan även utomstående granska och komma med förbättringar. Denna källkod granskas flera gånger av olika representanter från kända säkerhetsföretag. De senaste versionerna av OpenBSD har få eller inga ändringar då inga säkerhetsrisker eller buggar påträffats [2]. På alla punkter kom vi fram till att OpenBSD är ett bra val för att installera en router för ett litet nätverk. Användarvänligheten är något sämre än Microsofts produkter samt vissa versioner av Linux, men eftersom maskinen inte ska användas som en klientdator så har detta mindre betydelse. OpenBSD har också de bland de minsta systemkraven i sammanhanget då detta är ganska avgörande med tanke på att denna dator inte ska vara speciellt dyr.

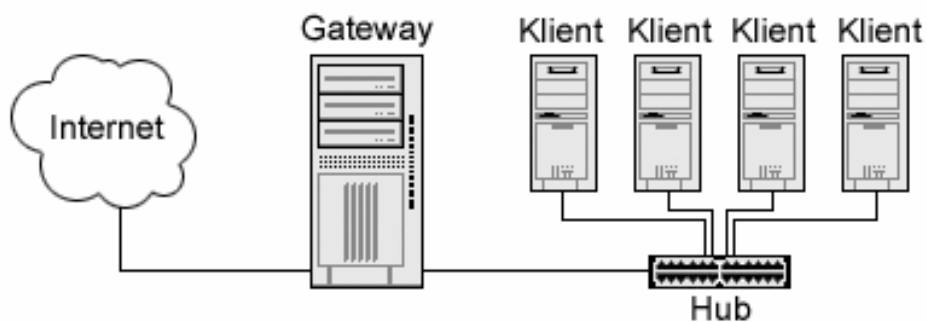
### **3 Konstruktion**

Steg ett i en process att bygga en router är att skaffa lämplig hårdvara. Vi fick låna en dator av skolan som vi installerade med Microsoft Windows 98 för att använda oss av vid installation av OpenBSD. Denna dator installerades med en ftp-server för att tillhandahålla installationspaketen från OpenBSD. De övriga datorerna som användes till de olika serverna har bekanta till oss lånat ut. Mycket av hårdvaran i datorerna är relativt gammal i datasammanhang, men detta är inte något problem då stödet [3] för äldre hårdvara är mycket bra. Vi har valt att testa OpenBSD på två olika typer av internetuppkopplingar. En router kommer att använda sig av en statisk IP-adress och den andra av en dynamisk IP-adress. Den router som använder sig av en statisk IP-adress har Sunet som internetleverantör och den routern som är kopplad till dynamisk IP-adress har Telias adsl som leverantör. Figur 3.1 ger en överblick i hur vi har tänkt bygga upp nätverket till routern med den statiska IP-adressen.



Figur 3.1: Tänkt utseende på Campus-nätverket

I detta nätverk kommer vikten att ligga på säkerhet mot routern. Med detta menar vi att kontrollen kommer att vara fokuserad på den externa delen (internet) och inte på interna säkerhetsrisker. Säkerhetsaspekten från det interna nätverket kommer vi inte att behandla i denna uppsats då tid till detta ej har blivit avsatt. Operativsystem som kommer att användas för de olika datorerna inom det interna nätverket är uteslutande OpenBSD 3.0 förutom en klient med Microsoft Windows 98. Klientdatorn kommer att användas som ftp-server för att installation av OpenBSD ska kunna vara möjligt. Den router som har adsl som internetuppkoppling är sammankopplat enligt figur 3.2.



Figur 3.2: Tänkt utseende på adsl-nätverket

I detta nätverk ligger det inga servrar bakom routern. Alltså i detta nätverk är det bara klienter som arbetar vilket gör att samma säkerhet inte kan förekomma som i nätverket på Campus.

Detta på grund av att klienter ofta har många olika program som körs mot det externa nätverket som i sin tur kräver att mer portar är öppna på routern. Att säkra ett system som har det här utseendet är mycket svårt då man inte vill blockera applikationer eller portar för användare. Det finns ytterligare ett problem i detta nätverk och det är att vi inte vet hur man konfigurerar Telias adsltjänst i OpenBSD. Dock finns det program att ladda ned från internet som enbart är till för att köra en adsltjänst från Telia. Detta var vi tvingade att göra och det är det enda program som vi har använt oss av som inte har funnits i en standardinstallation av OpenBSD.

### **3.1 Installation**

Att installera OpenBSD kräver att man antingen har köpt originalskivor från OpenBSD eller att man installerar över internet via en ftp-server eller en html-server. Vi har inte köpt originalskivorna utan installerar våra PC-datorer från en ftp-server. En förklaring över hur man installerar beskrivs i Bilaga A.2.

### **3.2 Applikationer**

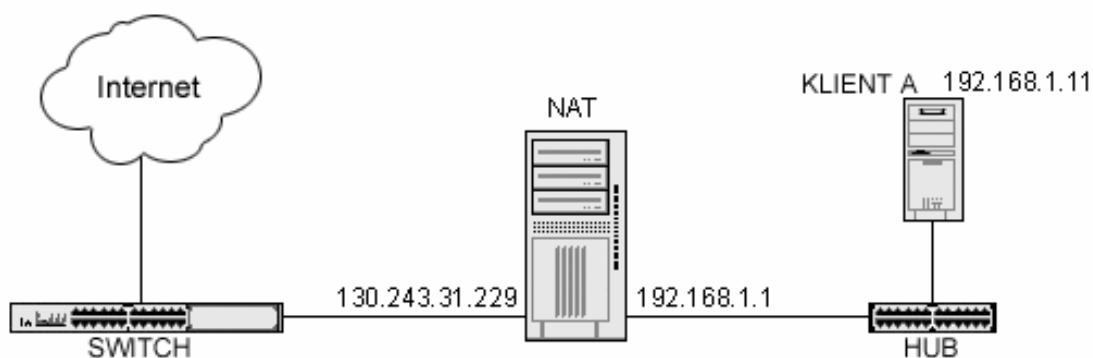
De applikationer som vi har använt oss av finns med i en standardinstallation av OpenBSD, dock finns det undantaget som beskrevs i kapitel 3 om programmet Qadsl som används för uppkopplingen mot Telias adsltjänst. Anledningen till att detta program var lämpligt att ladda ned var att vi inte visste hur vi skulle få igång denna tjänst enbart genom att konfigurera routern. Detta problem tar vi upp vidare i ett avsatt avsnitt som handlar om hur installationen av adsltjänsten mot OpenBSD går tillväga. För övrigt ingår alla andra applikationer i en standardinstallation av OpenBSD. Det var ett av kraven att programvaran till detta examensarbete inte skulle kosta pengar och alltså inte heller applikationerna.

#### **3.2.1 Network Address Translation**

Network Adress Translation (NAT) erbjuder ett enkelt sätt att länka nätverk till en adresserbar IP-adress som kan leda ut till det externa nätverket, till exempel internet. För att förklara lite mer ingående följer här en förklaring på vad NAT är och ett exempel på hur det fungerar. Ofta har man inte tillgång till mer än en riktig IP-adress, alltså en IP-adress från sin Internet Service Provider (ISP). Om man då vill använda sina klienter som finns på det interna



nätverket (LAN) mot det externa nätverket (internet) behöver man något som länkar samman dessa två olika nät. Problematiken ligger i att IP-adresserna i det interna nätverket inte kan användas på internet för dessa är inte riktiga IP-adresser. För att tilldela IP-adresser i det interna nätverket kan man använda de som finns tillgängliga i RFC 1918 [6]. I RFC 1918 finns det avsatta IP-adressegment som är för interna nätverk och andra tjänster. Det är rekommenderat att använda adressedgment i RFC 1918 då man tilldelar IP-adresser i det interna nätverket då dessa adresser aldrig kan finnas på det externa nätverket. Detta på grund av att en ISP inte kan använda sig av segmenten i RFC 1918. När väl klienterna tilldelats IP-adresser och routern har en officiellt riktig IP-adress kan man konfigurera sin NAT applikation. För att förklara hur förloppet för NAT ser ut ges här en förklaring med hänvisning till figur 3.3.



Figur 3.3: Network Address Translattion exempel

När en klient i det interna nätverket (klient A, IP:192.168.1.11) vill kommunicera med en kontakt på det externa nätverket (internet, IP:130.243.31.26), sänder klient A ut ett TCP-paket med en förfrågan om anslutning. I pakethuvudet på TCP-paketet finns IP-adressen till klient A och den begärda kontaktens IP-adress. Routern som är installerad med NAT tar emot detta TCP-paket eftersom klient A har konfigurerat att routern har IP-adress 192.168.1.1 och byter IP-adressen i källfället på TCP-paketet från klient A till sin egen IP-adress som är kopplad till det externa nätverket (IP:130.243.31.229). Den externa kontakten tror nu att paketet ursprungligen kom från routern, då den egentligen kom från klient A. Den externa kontakten kommer då att skicka tillbaka ett svar till routern som om det var den som ursprungligen upprättade kontakten. När routern får svaret översätter den snabbt destinationsfället i TCP-paketet från sitt eget till IP-adressen som klient A har och skickar paketet till klienten. Klienten i detta fall är helt ovetande om hur händelseförloppet ser ut för den externa trafiken. I bilaga B.2 har vi gjort en liten guide för hur man på ett enkelt sätt kan konfigurera NAT på

sin server. Vi beskriver vad som ska ändras och vad man bör tänka på.

### 3.2.2 Paketfilter

Paketfiltret som följer med OpenBSD 3.0 anses vara mycket bra. Applikationen är framtagen av gruppen som utvecklar och konstruerar OpenBSD. Anledningen till att de har utvecklat ett eget paketfilter är att paketfiltret de använde sig av i föregående distribution inte var tillräckligt bra enligt gruppens krav. Paketfiltret anses vara mycket pålitligt om man konfigurerar det på ett korrekt sätt. Dock kan aldrig systemet bli fullständigt skyddat då detta skulle resultera i att systemet blir helt obrukbart, det finns alltid hål om man har ett öppet system. Skillnaden mellan ett paketfilter och en brandvägg är att ett paketfilter inte söker av paketen mer än i dess huvud. Paketets huvud innehåller en mängd information bland annat paketets destinations-IP-adress och käll-IP-adress. Brandväggen däremot söker igenom hela paketet, alltså även datafältet. Summan av detta blir att paketfiltret inte till exempel kan hitta korrupt eller smittad data i paketet. Fördelen med att använda ett paketfilter istället för en brandvägg är att paketfiltret är mycket snabbare på att förmedla paket då processen att kontrollera innehållet inte förekommer. Paketfiltret är alltså inte en komplett lösning med aspekt för säkerheten i systemet. Säkerheten ökar dock dramatiskt om man använder sig av ett paketfilter mot den nod som mynnar ut mot det externa nätverket. Paketfiltret kan givetvis också konfigureras mot det interna nätverket. För att få en ökad säkerhet på klienterna i det interna nätverket bör brandväggar implementeras på dessa. För större nätverk är detta dock inte ett bra alternativ, detta på grund av att underhållet blir alltför krävande för att bibehålla säkerheten. Paketfiltret har två huvudsakliga uppgifter, att ta hand om restriktioner och vidareförmedling av paket samt att länka samman värdar och subnät till de externa adresserna. För att skapa en så säker implementation som möjligt av sin paketfilterpolicy, ska man inte tillåta någonting som standard, det vill säga all trafik som inte är explicit deklarerad i policyn blir nekat. Om man håller sig till denna ganska lätta regel blir regeldefinitionerna mindre och man skyddar sig från att missa exponerade delar i systemet. I bilaga B.1 har vi gjort en liten guide för hur man kan konfigurera paketfilterreglerna på sin router. Vi beskriver vad som ska ändras och vad man bör tänka på.

### **3.2.3 Apache Webbserver**

Vi ville använda en dator installerad med OpenBSD som webbserver, och detta går alldeles utmärkt då det finns programvara för detta som standard i operativsystemet. Det webbserverprogram som finns med är Apache v1.3. Apache är marknadens största webbserverprogram, och har varit det sedan 1996. Enligt undersökning så används Apache på 54% av alla webbservrar på internet [7]. Målet med projektet Apache är att tillhandahålla en säker webbserver med öppen kod som har stöd för rådande standard för internetsidor. Den nyaste versionen av Apache webbserver är Apache 2.0.35. Den har till skillnad från version 1.3, stöd för flertrådar-processer och IPv6. Det som är nytt i Apache 2.0 kan läsas på Apaches hemsida [8]. Vi har dock valt att inte uppgradera eftersom vi inte har några stora krav på vårans webbserver och inte kommer använda de nya funktionerna som version 2.0 erbjuder. I bilaga B.3 har vi förklarat vad som behövde göras för att starta upp en Apache webbserver.

### **3.2.4 Secure Shell SSH**

När man har ett nätverk med en server så behöver man efter installation inte ha en bildskärm, mus, eller tangentbord till sin server. Man loggar helt enkelt in från en annan dator till servern och kan göra ändringar och inställningar från den andra datorn. Detta kan man göra såväl från en dator inom det lokala nätverket som utifrån internet. För att kunna göra detta krävs ett protokoll och en applikation. Telnet är en sådan applikation som kan användas för fjärrinloggning, men med OpenBSD och de flesta andra Unix och Linux-distributioner kan man använda sig av Secure Shell (SSH) som har bättre stöd för säkerhet vid fjärrinloggning. SSH ingår i alla OpenBSD-versioner efter 2.5. Det har stark autentisering och säker kommunikation över nätverk. Telnet skickar till exempel alla lösenord i klartext och då kan man avlyssna lösenord och data, medan SSH använder Rivest Shamir Adleman (RSA)-kryptering. Även data skickas med kryptering. I OpenBSD 2.9 och 3.0 finns det stöd för SSH2. Detta protokoll har stöd för fler tjänster och använder sig av en ännu bättre kryptering med hjälp av Diffie-Hellman (DH)- och Digital Signature Algorithm (DSA)-algoritmer. Det ingår även i SSH2 stöd för SFTP2, som är ett ftp-protokoll med säker överföring.

### **3.2.5 File Transfer Portocoll FTP**

Att ha en ftp-server aktiverad på sin OpenBSD-server kan vara väldigt användbart. Särskilt när man använder den som webbserver och utifrån vill uppdatera den med nya hemsidor och filer. Här kan man ge användare tillgång till gemensamma mappar och filer. En användare

kan även få en egen mapp som bara den har tillgång till. I bilaga B.4 beskriver vi hur man kan aktivera och konfigurera applikationen för OpenBSD.

### **3.2.6 Network File System NFS**

NFS är en standard för filsystem inom Unix, och används för att dela ut filer över det lokala nätverket. Fördelen med detta är att klienter kan använda gemensamma filer och applikationer. Användare kan även ha en personlig hemkatalog för att spara sina filer. Om man tillåter åtkomst av filer från det externa nätverket så kan denna trafik synas på det externa nätverket, vilket medför en säkerhetsrisk. Det är också viktigt att man har en lista på vilka klienter som ska ha åtkomstmöjligheter att skapa och radera filer på NFS-servern. I bilaga B.5 beskriver vi hur man kan aktivera och konfigurera applikationen för OpenBSD.

### **3.2.7 Dynamic Host Configuration Protocol DHCP**

Att använda sin router som en DHCP-server kan vara mycket smidigt då denna applikation gör att man slipper att konfigurera nätverket för sina klienter i nätverket. DHCP-servern tilldelar alltså information till klienterna så som IP-adress, DNS och standard router. Vi valde att installera vår router med denna funktion då vi tycker att det kan vara användbart även i mindre nätverk. DHCP-servern lagrar vilken information den har givit en klient genom att ta dess Media Access Control (MAC) adress. MAC-adressen är unik, då denna bara kan förekomma på ett specifikt nätverkskort. Varje gång en klient försöker kontakta DHCP-servern tittar denna om den har lagrat klientens MAC-adress förut. Om så är fallet försöker DHCP-servern ge klienten samma information som den gav förra gången den blev förfrågad från samma klient. Om så inte är fallet lagrar den klientens MAC-adress samt den information den ger ut för att vid en senare förfrågan kunna dela ut samma information. Hur installationen ser ut visas i bilaga B.6.

### **3.2.8 Qadsl**

För att det ska vara möjligt att använda sig av en tjänst som Telias adsl i OpenBSD, var vi tvingade att använda ett program som klarade av att ansluta sig mot denna. Det ska vara möjligt att ansluta sig manuellt mot Telia, men vi hade varken kunskap eller tid för undersöka problemet djupare. Lösningen till problemet var alltså ett program som heter `qadsl` som vi

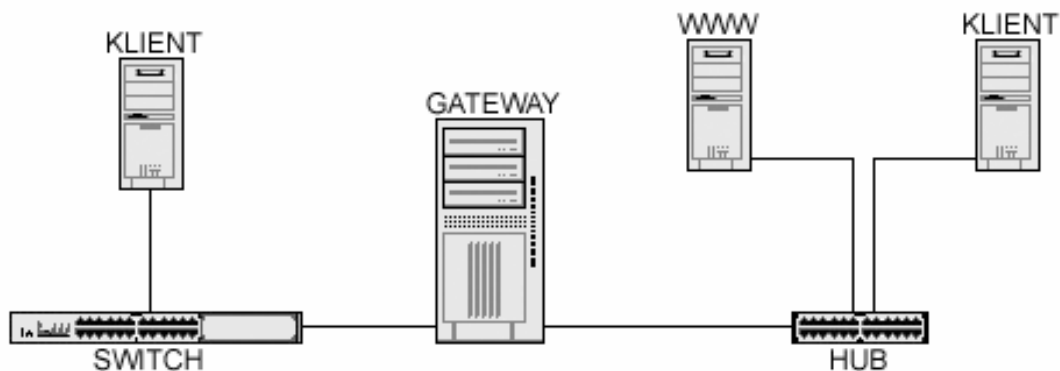
hittade på internet. Programmet kopplar upp sig mot Telia och ger den nödvändiga informationen som behövs till routern. Hur detta är installerat beskrivs i bilaga B.7.

## **4 Test**

I detta kapitel kommer våra tester mot applikationer och servrar att redovisas. Testerna är helt ovetenskapliga då testutförandet inte gjordes med ett program skapat enbart för uppmätning av nätverksprestanda. Vi har ingen jämförelse att göra mot andra operativsystem vilket gör att den prestanda vi redovisar bara kan ses med eller utan routern, då denna är den enda referens vi har. Detta innebär att ftp-server testet och webbserver testet kommer att redovisas med avseende på skillnaden i laddningstid från det interna nätverket respektive det externa nätverket. Vi kommer bara att visa medelvärden, tabeller och diagram i detta kapitel men de fullständiga testresultaten vi har fått under vår testperiod finns deklarerade i bilaga C och D. Klienterna vi använde oss av i detta test var installerade med Microsoft Windows XP. Anledningen till att vi valde att använda oss av Microsoft Windows XP var att denna distribution har fått goda rekommendationer [10] i avseende på nätverksprestanda.

### **4.1 Test av Webbserver**

Som webbserver har vi valt att använda oss av en Pentium 200 Mhz, med 32 megabyte i ramminne och en hårddisk på 540 megabyte. Detta kan tyckas vara en ganska undermålig dator för webbserver men för hemmabruk bör detta räcka mer än väl. Dock kom det att visa sig i testerna att maximal prestanda inte kunde uppnås. På webbservern använde vi Apache 1.3 som var med i standardinstallationen av OpenBSD 3.0. På klientdatorn som vi testade att ladda hem filer med samt att ladda in hemsidan använde vi en Pentium 4, 1500 Mhz, med operativsystem Windows XP och Internet Explorer 5.0 som vi lånade av en vän. Figur 4.1 visar hur vår koppling ser ut. Klienterna i figur 4.1 är alltså samma dator som flyttades runt under testerna för att få ett så korrekt värde som möjligt.



*Figur 4.1: Webbserver test*

#### **4.1.1 Test av en 100 megabyte fil**

Första testet gick ut på att ladda en stor fil på 100 megabyte från webbservern till först en dator i det externa nätverket, och sedan till en dator i det interna nätverket via http-protokollet. Detta gjordes tre gånger för få ett bättre medelvärde. Att ladda hem en fil över http som är så stor som 100 megabyte är inte särskilt vanligt, men man kan se det som att till exempel sända en film över en webbserver. För detaljerad beskrivning se bilaga C, tabell C.1.

#### **4.1.2 Test av en 10 megabyte fil**

Andra testet gick ut på att skicka en mindre fil på 10 megabyte. Precis som på det första testfallet gjordes testet tre gånger och testet inkluderade intern överföring och extern överföring. Detta skulle kunna jämföras med att ladda ner musik från en hemsida. För detaljerad beskrivning se bilaga C, tabell C.1.

#### **4.1.3 Test av inläsning av hemsida**

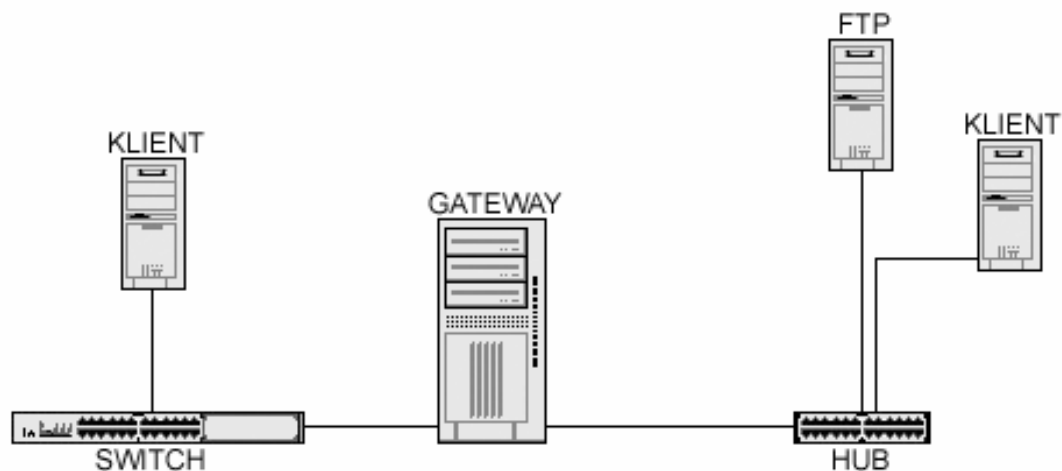
Det sista testet vi gjorde på webbservern var att ladda ned en hemsida. Hemsidan bestod av både länkar, bilder och text. Sidan som laddades ned, `index.html`, var på 1200 kilobyte inklusive bilder. Detta test gjordes både från en dator i det interna nätverket och från en dator i det externa nätverket. Testerna gjordes tre gånger vardera. Mellan varje test tömdes allt cache-minne på webbläsaren på den dator som utförde testet. För detaljerad beskrivning se bilaga C, tabell C.1.

#### 4.1.4 Slutsats från webserver testet

Det som testades var prestanda för tre olika fall. Först ladda ned en 100 megabyte fil, sedan 10 megabyte fil och sist ladda in `index.html`. Som man kan se enligt bilaga D, diagram D.1, så varierar hastigheten från 280 kilobyte/sekund till 325 kilobyte/sekund. Detta anser vi vara en ganska liten variation, och webbservern klarar alla testfallen bra med tanke på den maskinvara webbservern var utrustad med. Skillnaden mellan överföringen via en dator i det interna nätverket och en dator i det externa nätverket varierar mellan 3 % och 10 %, vilket vi anser vara litet. Sammantaget kan man säga att webbservern hanterar många mindre filer lika bra som den hanterar mellanstora och stora filer vilket kan ses tydligare i bilaga C, diagram D.1.

## 4.2 Test av ftp-server

Ftp-server testet gick ut på att testa prestandan för en ftp-server med Unix OpenBSD. För att få en uppskattning av maximal prestanda valde vi tre olika testfall. Dessa skulle spegla ett vanligt användande av en ftp-server. Målet var att se hur bra ftp-servern kunde hantera olika sorters ftp-trafik, samt att se eventuella fördröjningar via routern. Figur 4.2.1 visar hur vår sammankoppling ser ut.



Figur 4.2: Ftp-server test beskrivning

### 4.2.1 Test med 100 stycken små filer

Testet gick ut på att överföra 100 stycken små filer mellan ftp-servern och en klientdator. Filernas storlek varierade från den minsta på 1 kilobyte till den största på 564 kilobyte. Sammanlagt var filerna på 4816 kilobyte. Överföringen gick till så att en dator i det externa

nätverket laddade ned filerna från ftp-servern, och sen laddade upp filerna till ftp-servern. På samma sätt gjordes för en dator i det interna nätverket. Alla testerna gjordes tre gånger. För detaljerad beskrivning se bilaga C, tabell C.2.

#### **4.2.2 Test av överföring av fil på 100 megabyte**

Testet gick ut på att få en bild över vilken överföringshastighet det var för stora filer. Filen som skickades var på 100 megabyte. Denna fil laddades först upp på ftp-servern och sen ned från ftp-servern med en dator från det externa nätverket. Efter det gjordes samma sak med en dator i det interna nätverket. Alla test gjordes tre gånger. För detaljerad beskrivning se bilaga C, tabell C.2.

#### **4.2.3 Test av överföring av liten fil 10 MB**

Detta test utfördes på samma sätt som för filen på 100 megabyte. En 10 megabyte stor fil överfördes först från en klientdator i det externa nätverket till ftp-server, och sen från ftp-servern till klientdatorn i det interna nätverket. Efter det gjordes samma test med en intern klientdator. Alla test gjordes tre gånger. För detaljerad beskrivning se bilaga C, tabell C.2.

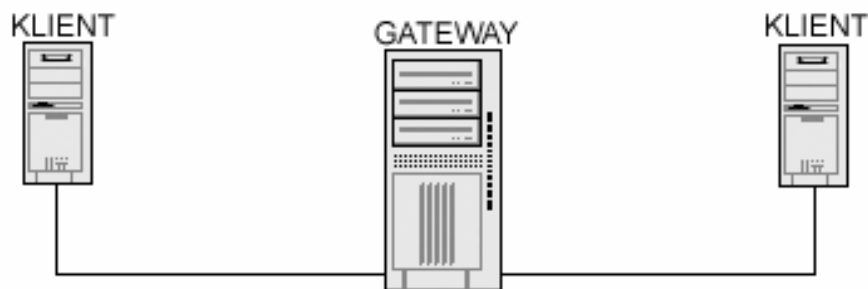
#### **4.2.4 Slutsats från ftp-server testet**

De olika testerna gick ut på att ta reda på hur bra den ftp-servern som följer med OpenBSD hanterar olika sorters trafik. Man kan tydligt se i bilaga D, diagram D.2, att överföringen av 100 små filer är betydligt långsammare än både överföringen av en 10 megabyte fil, och en 100 megabyte fil. Detta behöver inte betyda att ftp-servern är dålig på att hantera många små filer, utan helt enkelt att TCP-protokollets slowstart medför att ingen hög överföringshastighet hinner bli uppnådd innan filen är fullständigt överförd. Anmärkningsvärt är dock att den snabbaste hastigheten som uppnåddes var vid överföringen av en 10 megabyte stor fil, vilket troligen beror på att hela filen kan buffras i ftp-serverns internminne. Av någon anledning gick det betydligt långsammare att ladda upp många små filer, än att ladda ned många små filer från ftp-servern. Svaret på detta kan förklaras senare i testet av routern, se kapitel 4.3.4. Skillnaden mellan ftp-överföring via internt nätverk och externt nätverk varierar mellan 6 % och 9 %, med undantag för uppladdning av 100 små filer där det är en skillnad på 40 %. Överföringen av en 10 megabyte fil och en 100 megabyte fil, uppnådde hastigheter som lägst 750 kilobyte/sekund och som högst 947 kilobyte/sekund, vilket vi anser mycket bra. Dels för att det är nära den maximala hastigheten som kan uppnås i ett 10 megabit nätverk, och dels med tanke på den maskinvaran vi använde för ftp-servern.



### 4.3 Överföringshastighets test med router

Router testet gick ut på att ta reda på, om en router med operativsystemet OpenBSD försämrade höghastighetsöverföring av data mellan två datorer, och i så fall hur mycket. För att komma upp i en hög hastighet var alla datorer utrustade med nätverkskort som klarar en överföringshastighet på 100 megabit. I dessa test kan det komma in varierande resultat som inte beror på hur mycket som går att skicka i nätverket utan till exempel på hur snabbt en hårddisk klarar att ta emot data som kommer in från nätverkskortet. Därför fick en del test göras fler gånger än vad som visas i tabeller och diagram. En annan orsak som skulle kunna påverka testet är att den dator vi använde som router inte var snabb nog, och att det inte var operativsystemet som eventuellt påverkade mätresultaten. Figur 4.3 visar hur sammankopplingen såg ut när vi använde routern mellan klienterna. Figur 4.4 visar hur sammankopplingen såg ut när vi använde en switch mellan klienterna.



Figur 4.3: Sammakoppling med router



Figur 4.4: Sammakoppling med switch

#### 4.3.1 Test av överföring av fil på 600 megabyte

Testet gick ut på att via ftp-protokollet överföra en fil i båda riktningarna mellan två klientdatorer. Först testades överföringen med en router i mellan klientdatorerna och sen

testades överföringen direkt mellan klientdatorerna via en switch som klarar 100 megabit. Vi testade även om det var någon skillnad mellan att ladda upp en fil till den klientdator som körde ftp-server, eller ladda ned filen från ftp-servern till den andra klientdatorn. Alla test gjordes minst tre gånger för att få bättre värde. När denna stora fil på 600 megabyte överfördes använde vi hårddisken på ett sådant sätt att filen inte behövde fragmenteras. Detta gjordes för att inte hårddisken skulle på något sätt bromsa överföringen, vilket skulle kunna ge felaktiga resultat. Det fullständiga testresultatet finns i bilaga C, tabell C.3 och tabell C.4.

#### **4.3.2 Test skicka 10 megabyte fil**

Detta test liknade det testet vi gjorde med överföring av en fil på 600 megabyte. Vi skulle överföra en 10 megabyte stor fil mellan två klientdatorer i båda riktningarna, med och utan router. Utgångspunkten med testet var att ta reda på hur snabbt TCP-uppkopplingen kom till en maximal hastighet. Vi hade läst att det gick att ändra parametrarna för TCP-uppkopplingens slowstart, men vi behöll systemet som det var efter standardinstallationen. Det fullständiga testresultatet finns i bilaga C, tabell C.3 och tabell C.4.

#### **4.3.3 Test av överföring 1000 filer**

Sista testet gick ut på att överföra 1000 filer, på sammanlagt 16 megabyte. Alla filerna var väldigt små, med den minsta på 1 kb och den största på 30 kb. Samma mönster som för dom två första testen följdes. I detta test skulle det verkligen löna sig om man inte använde slowstart, men inte heller här ändrades på några inställningar för TCP-protokollet. Det fullständiga testresultatet finns i bilaga C, tabell C.3 och tabell C.4.

#### **4.3.4 Slutsats av router testet**

Router testet gjorde vi dels för att testa eventuella bromsklossar i nätverket och dels för att se hur bra en OpenBSD router skulle klara att hantera varierad trafik i mycket hög hastighet. Diagram i bilaga D, diagram D.4, diagram D.5, diagram D.6 och diagram D.7 visar hur dataflödet varierar mellan det interna och externa nätverket. Testet gjordes enbart för att upptäcka detta och är inte det mest relevanta i router-testet. Det viktiga var i stället att ta reda på skillnaden mellan att använda en router och att vara direktuppkopplad mellan två datorer. Testet utfördes genom att först skicka via routern, och sen via en 100 megabit switch. Eftersom överföring av 1000 små filer inte kunde uppnå hög hastighet på grund av slowstart begränsning, och överföring av en 10 megabyte fil och en 600 megabyte fil var likvärdiga, räcker det med att jämföra 600 megabyte överföring, med och utan router. Skillnaden i hastighet mellan trafiken som passerade genom routern, och trafiken som inte passerade

genom routern, varierar mellan 15 % och 40 %. En variation på så lite som 15 % i hastigheter på 6000 kilobyte/sek anser vi föredömligt bra. Det är svårt att uppskatta hur mycket denna variation beror på operativsystemets begränsningar eller på routerns hårdvarubegränsningar. Eftersom en viss variation på 10 % uppkommer i låga hastigheter under 1000 kb/sek, enligt ftp- och webb-testerna drar vi slutsatsen att operativsystemet till viss del bromsar överföringen. Då en uppskattning om operativsystemet bromsas proportionellt mot hastigheten kan vi bara spekulera i, och det skulle behövas många fler, och noggrannare tester för att få klarhet i detta. Det är dock så att man kan ändra en buffretstorlek och därmed öka prestandan genom att slowstarten skippas. Detta resulterar i att det går mycket snabbare att skicka många små filer. Detta har vi inte gjort då vi ville köra med en standard konfiguration. En viktig tanke är också hur mycket andra operativsystem bromsar överföringen jämfört med OpenBSD. Tester av den typen hade vi inte möjligheter att göra, och vårt mål var heller inte att utreda just vilket operativsystem som skulle vara allra effektivast. Målet var att utreda hur OpenBSD klarade av olika typer av överföring, och efter våra tester drar vi slutsatsen att operativsystemet klarar sig mycket bra i alla våra tester. Dock skulle viss optimering av TCP-protokollet kunna göras för att effektivare kunna överföra många små filer.

## **4.4 Applikationstest på routern**

Att testa hur bra OpenBSD är en omöjlig uppgift för vår del, då vi inte har något annat operativsystem att utföra en test med eller kunskap att göra ett sådant utlåtande. I detta kapitel har vi skrivit om våra tester av de applikationer vi har använt oss av samt vilka problem som fanns.

### **4.4.1 Test av NAT**

För att se om vår konfiguration av NAT fungerade på ett korrekt sätt tilldelades routern den kod som står i bilaga B.2. Testet gick ut på att de klienter som befann sig i det interna nätverket skulle få tillgång till det externa nätverket. Testets utfall blev lyckat och de datorer vi prövade blev länkade till det externa nätverket på ett korrekt sätt. Testet fortskred med test av omlänkningskommandot `rdr` som följer med NAT. Vi testade om vi från routern kunde länka om port 21 på det externa nätverk till en klient på det interna nätverket. Testets utfall blev lyckat och fungerade som utlovat.

#### **4.4.2 Test av PF**

För att pröva vår konfiguration av paketfiltret så att det fungerade på ett korrekt sätt utförde vi följande. Klienterna i det interna nätverket tilldelades tillgång mot det externa nätverket med hjälp av NAT. Testets syfte var att ta reda på om det var möjligt att blockera vissa portar, samt neka tillgång till vissa klienter i nätverket om man hade en IP-adress om inte var godkänd enligt regeldefinitionen. Testet utfördes med den kod som finns i bilaga B.1. Vi prövade först med att blockera en port för all inkommande trafik på routern och detta fungerade utmärkt. Även nästkommande test fungerade felfritt när vi prövade om routern kunde blockera en specifik IP-adress.

#### **4.4.3 Test av DHCP**

För att se om vår konfiguration av DHCP-servern fungerade på ett korrekt sätt tilldelade vi inte nätverksinformation till en klient som var installerad med OpenBSD. Testet utfördes med den kod som finns deklarerad i bilaga B.6. Testet fortskred så att vi tilldelade klientens nätverkskort att lyssna till DHCP och startade DHCP-rutinen på routern med `dhcpcd` kommandot. Klientdatorn startades om för att se om den blev till delad nätverksinformation från DHCP-servern. Utfallet blev lyckat och applikationen fungerade felfritt.

### **4.5 SSH test**

Våra tester mot SSH var inte direkt omfattande då denna funktion startas omgående i en standardinstallation av OpenBSD. Vi har inte gjort någonting själva för att få applikationen att fungera då en egen implementation inte var nödvändig. Under den tid vi arbetat med detta examensarbete använde vi oss mycket ofta av SSH för att inte behöva sitta fysiskt vid varje dator. Den klient-programvara vi använde oss av heter putty och är helt gratis för nedladdning från internet. Putty är utvecklat för Microsoft Windows plattformar. Vi är mycket nöjda med hur bra SSH fungerade. Dock fick vi problem med vår router på Campus då denna hade väldigt lång inloggningstid. Inloggningstiden på denna dator var mellan 60-65 sekunder och på de övriga hamnade den runt 1-4 sekunder. Vi vet inte vad detta berodde på och har inte hittat något hårdvarufel på datorn. För övrigt var vi nöjda med hur SSH fungerade och ser detta verktyg i sin helhet som mycket bra.

## 4.6 NFS test

På grund av tidsbrist valde vi att inte testa dela ut filer över nätverket med hjälp av NFS. Men vi tar ändå upp denna funktion som en del av arbetet och beskriver hur man går tillväga för att aktivera och konfigurera NFS-fildelning.

## 4.7 Qadsl test

Att sätta upp en router mot Telias adsl visade sig vara ganska lätt. Vi installerade en dator med OpenBSD och konfigurerade den till en router. Det externa nätverkskortet konfigurerades att lyssna till DHCP. Detta på grund av att Telias adsltjänst tilldelar sina kunder med en dynamisk IP-adress. Vi lyckades få all den vitala informationen från Telias DHCP-server. Nästa steg i processen var att starta Qadsl. Efter att vi konfigurerat de nödvändiga filerna var det bara att starta programmet. Utfallet blev lyckat och vi hade därmed lyckats att initiera en uppkoppling med OpenBSD mot Telias adsl.

## 5 Erfarenheter och rekommendationer

Detta arbete har givet oss en inblick i flera olika operativsystem. Med hjälp av den kunskapen kunde vi lättare utvärdera det operativsystem vi valde att testa och installera. När vi valde Unix OpenBSD visste vi att det skulle vara betydligt mer komplicerat att både installera och konfigurera än vi var vana med andra operativsystem. Nu i efterhand inser vi att det inte var särskilt mycket som behövdes ändras eller läggas till för att en standardinstallation ska fungera. Hade vi från början haft denna guide som vi själva skrivit skulle det varit betydligt mycket lättare att få systemet att köra de tjänster man vill använda sig av. Det hade dessutom gått mycket snabbare att ha all information samlad på ett ställe, än att leta runt på internet. Vi förstod också under arbetets gång att alla operativsystem med öppen källkod har utförlig dokumentation om hur man ska gå till väga för att installera och konfigurera deras operativsystem. Denna dokumentation finns på respektive operativsystem hemsida. Det finns även många forum där man kan ställa frågor om problem som har uppstått. Har man bara tid att leta igenom dessa hemsidor, finner man med allra största sannolikhet svaret på sina frågor. För de som vill använda sig av en server av något slag i ett mindre nätverk, antingen som router, brandvägg, fil-server, ftp-server eller webbserver så rekommenderar vi OpenBSD. För

nybörjare kan det vara svårt då viss förkunskap och förståelse för hur en dator fungerar och hur ett nätverk kan byggas upp krävs. Man bör även ha ett visst intresse för området. Vi använde inte något grafiskt gränssnitt för någon av våra servrar vi installerade och det var till en början något ovant. Detta kände vi inte var någon begränsning utan snarare en självklarhet då endast vissa textfiler behövde redigeras.

## 6 Slutsats

Vad har vi då kommit fram till i detta arbete? Vårt mål var att söka kunskap om olika operativsystem för serververksamhet, välja ett av dem, installera och testa operativsystemet och skriva en kortare guide för att visa hur man ska installera operativsystemet. Dessa mål uppnådde vi inom tidsramen för examensarbete. Vi fick dessutom en uppfattning om prestandan för operativsystemet OpenBSD med våra ftp-, webb-, och router –test. Två test som vi hade planerat att genomföra men som vi inte hade tid med var att testa mailfunktionen `sendmail`, och att testa fildelning via `NFS`.

En annan viktig aspekt som hade varit givande var att testa OpenBSD mot andra operativsystem. Då detta hade tagit betydligt mer tid, då ominstallationer av andra operativsystem hade varit nödvändiga. Dessutom är detta ett examensarbete på C-nivå för universitetet vilket inte gav oss tillräckligt med tid, dock kan vara ett område att utreda i en D-uppsats i framtiden.

Sammantaget skulle vi vilja säga att OpenBSD är ett bra operativsystem att använda för serververksamhet. Säkerheten är hög redan från grundinstallationen och operativsystemet är mycket stabilt. Det är relativt lätt att konfigurera och mycket hjälp finns att få på internet samt att det är gratis. Vi stötte inte heller på några större problem som vi inte lyckades lösa.

## Ordlista

<b>NIC</b>	Network Interface Communication
<b>Open Source</b>	Källkoden är öppen för alla
<b>RSA</b>	Rivest Shamir Adleman
<b>DH</b>	Diffie-Hellman
<b>DSA</b>	Digital Signature Algorithm
<b>Gb</b>	Gigabyte
<b>MB</b>	Megabyte
<b>Kb</b>	Kilobyte
<b>ISP</b>	Internet Service Provider
<b>NAT</b>	Network Address Translation
<b>DHCP</b>	Dynamic Host Configuration Protocol
<b>SSH</b>	Secure Shell
<b>PF</b>	Paketfilter
<b>NFS</b>	Network File System
<b>ANSI</b>	American National Standards Institute
<b>POSIX</b>	Portable Interface Unix
<b>IPv6</b>	Nästa generations IP-protokoll
<b>MAC</b>	Media Access Control

## Referenser

- [1] *OSdata, Server OS technical comparison*, <http://www.osdata.com/oses>, Maj 2002
- [2] *OpenBSD, OpenBSD:säkerhet*, <http://openbsd.appli.se/se/security.html>, Maj 2002
- [3] *OpenBSD, OpenBSD*, <http://www.openbsd.com/>, Maj 2002
- [4] *Real Quellet, Firewall using OpenBSD*, <http://www.realo.ca/bsdinstall.html>, Maj 2002
- [5] *K.Egevang, P.Francis, RFC 1631, The IP Network Address Translator (NAT)*, 1994
- [6] *Y.Rekhter, B.Moskowitz, D.Karrenberg med flera, RFC 1918, Address Allocation for Private Internets*, 1996
- [7] *The Apache Software Foundation, Apache project*, <http://httpd.apache.org/>, Maj 2002
- [8] *The Apache Software Foundation, Features with Apache 2.0*, [http://httpd.apache.org/docs-2.0/new\\_features\\_2\\_0.html.en](http://httpd.apache.org/docs-2.0/new_features_2_0.html.en), Maj 2002
- [9] *OpenBSD, System management* <http://www.openbsd.org/faq/faq10.html#10.6> Maj 2002
- [10] *Exsites Network, Exhardware* <http://www.exhardware.com/reviews.php?Id=51> Maj 2002
- [11] *QBA, Linuxsweden* <http://qba.linuxsweden.nu> MAJ 2002



## A Installation av OpenBSD

I detta avsnitt tittar vi över de olika delarna man måste stega igenom för att slutföra en installation av OpenBSD. Vi har dokumenterat de stora delarna under installationsfasen, dock tillkommer det små delar under installationen men vi har valt att inte redovisa dessa på grund av att de är mycket självförklarande.

### A.1 Hårdvara

Det som står först på tur innan man påbörjar en installation är att undersöka vilken hårdvara som man har, speciellt på nätverkskort är det viktigt att veta både fabrikat och bustyp. Vi har sammanställt en Tabell A.1 över hårdvaran som vi kommer att använda i detta projekt.

Namn	CPU	RAM	Hårddisk	Nätverkskort 1	Nätverkskort 2
Campusrouter	PII 233 MHz	64 MB	1700 MB	Realtek 8139	Realtek 8139
Webbserver	P 200 MHz	64 MB	850 MB	NE 2000 10MBit PCI	
Ftp-server	486 100 Mhz	16 MB	540 MB	NE 2000 ISA	
Teliarouter	PII 366 MHz	32 MB	540 MB	NE 2000 ISA	3com 3c509b ISA

*Tabell A.1: Hårdvara som finns i våra datorer*

### A.2 Startdiskett

OpenBSD-distributionerna finns i dagsläget inte att ladda ned som image-filer för CD-bränning. Om man inte köper originalskivorna från OpenBSD kan installationen ske genom en ftp- eller http-nedladdning från internet. För att kunna installera via ftp- eller http-protokollet måste man skapa en så kallad startdiskett för att konfigurera datorn innan själva installationen börjar. Detta görs genom att ta hem ett program som heter `rawrite` och en imagefil på startdisketten från hemsida till OpenBSD. Detta kräver dock att man har tillgång till en dator som kör något av Microsoft produkter. Om man skapar startdisketten från ett annat operativsystem så behöver man ett annat program än `rawrite`. På denna diskett finns det lagrat en mängd olika adresser till installationsfilerna, men vi valde att ta hem dessa filer först och sedan öppna en egen ftp-server på en av klienterna, detta för att installationen skulle

gå snabbare. När denna diskett är skapad är det bara att starta maskinen från denna och följa anvisningarna. Den nuvarande distributionen av OpenBSD är 3.0, Februari 2002.

### A.3 Uppdelning av hårddisk till partitioner

Att dela upp hårddisken till partitioner i OpenBSD är ganska viktig då det kommer att lagras en hel del loggfiler på datorn. Om man väljer att inte skapa vissa partitioner kommer dator att fyllas upp av dessa loggfiler och tömma minnet i datorn vilket kan göra att datorn får ett systemavbrott. Till exempel, för vår router som ska köras 24 timmar om dygnet är det bra om man skapar en `/tmp` katalog där temporära filer kan lagras under körning, och om det blir för mycket filer i den katalogen fylls inte det övriga minnet upp på hårddisken utan bara `/tmp` partitionen blir full. Dock är den första frågan som OpenBSD kommer att ställa om man vill använda hela hårddisken till OpenBSD. Detta enligt figur A.1

```
Do you want to use the *entire* disk for OpenBSD? [no]
```

*Figur A.1: Partitioneringsfråga*

Svaret på den frågan är `no`. Detta på grund av att om svaret är `yes` så kommer hela `fdisk` processen att skippas och man får inte möjligheten att ändra cylinder/head/sector (CHS) på hårddisken. Anledningen till att man ofta vill ändra dessa värden är att på vissa system måste de vara konfigurerade till `C=0, H=0, S=1` för att kunna starta OpenBSD från hårddisken. Som standard väljer OpenBSD `C=0, H=1, S=1` och det är alltså när H blir satt som 1 som felet blir. Så när installationsprocessen frågar om man vill redigera CHS väljer man `yes` och får därmed möjlighet att ställa in rätt parametrar. Detta steg är som sagt inte nödvändigt för alla system. Vi märkte att äldre uppsättningar av hårdvara krävde att man redigerade CHS-inställningarna. Vidare tar programmet `disklabel` över när man kört `fdisk` för att organisera partitionerna med de tilldelade namnen. Vår partitionering av Campusroutern visas i figur A.2. Vi visar inte de övriga datorerna då dessa är väldigt lika Campusroutern.

## Campusrouter

#	size	offset	fstype	[fsize	bsize	cpgr]
a:	164241	63	4.2BSD	1024	8192	16 # /
b:	614880	164304	swap			#
c:	3335472	0	unused	0	0	#
d:	164304	779184	4.2BSD	1024	8192	16 # /tmp
e:	164304	943488	4.2BSD	1024	8192	16 # /var
g:	164304	1107792	4.2BSD	1024	8192	16 # /usr
h:	2062368	1272096	4.2BSD	1024	8192	16 # /home

*Figur A.2: Partitionering av Campusroutern*

## A.4 Konfigurering av nätverket

Att konfigurera nätverket på ett korrekt sätt är mycket viktigt om man inte har tillgång till originalskivorna då nätverket måste fungera korrekt om man ska lyckas med en ftp-, http-installation. OpenBSD stödjer de flesta nätverkskort, även äldre mindre kända nätverkskort med ISA-bus vilket inte är vanligt. Detta fina stöd för hårdvara gör det möjligt att använda äldre maskiner såsom till exempel en dator som är konstruerad av Intels 486-arkitektur med ett ISA-nätverkskort. I vårt fall har vi två stycken Realtek-nätverkskort med PCI-bus i routern och äldre ISA-nätverkskort i webbservern och ftp-servern. Det ena kortet i routern ska användas mot det externa nätverket och det andra mot det interna. Under installtionsprocessen kommer installtionsprogrammet att fråga om man vill konfigurera nätverket och detta bör man göra. I vårt fall fanns två nätverkskort i routern att konfigurera, `r10` och `r11`. Nätverkskonfigurationen för routern följer enligt figur A.3.

Namn: r10  
Korttyp: Realtek 10/100MB  
IP-adress: 130.243.31.229  
Nätmask: 255.255.254.0  
Media 100baseTX mediaopt full-duplex  
Detta kort är kopplat till det externa nätverket.

Namn: r11  
Korttyp: Realtek 10/100MB  
IP-adress: 192.168.1.1  
Nätmask: 255.255.255.0  
Media 100baseTX mediaopt full-duplex  
Detta kort är kopplat till det interna nätverket.

*Figur A.3: Nätverskkonfiguration för Campusroutern*

Vi konfigurerade även de två datorerna på det interna nätverket som ska användas som webbserver, se figur A.4, respektive ftp-server, se figur A.5, och en dator som vi ska använda mot Telias adsltjänst. På datorn som ska installeras för att använda Telias adsl, är man tvungen att konfigurera det nätverkskort som ska ligga mot det externa nätverket att lyssna till Dynamic Host Configuration Protocol (DHCP) för att registrera IP-adressen. Telia tilldelar dynamiska IP-adresser till sina kunder och det är alltså anledningen till att nätverkskortet måste lyssna till DHCP. Detta följer enligt figur A.6.

Namn: ep0  
Korttyp: 3Com Etherlink XL COMBO 10Mbps  
IP-adress: 192.168.1.11  
Nätmask: 255.255.255.0  
Media 10baseT

*Figur A.4: Nätverskkonfiguration av webbservern*

Namn: ne1  
Korttyp: NE2000 10Mbit PCI Ethernet  
IP-adress: 192.168.1.10  
Nätmask: 255.255.255.0  
Media 10baseT

*Figur A.5: Nätverskkonfiguration av ftp-servern*

Namn: ep0  
Korttyp: 3com 3c509b-TP 10Mbit ISA Ethernet  
IP-adress: DHCP  
Nätmask: DHCP  
Media 10baseT

Namn: ne1  
Korttyp: NE2000 10Mbit PCI Ethernet  
IP-adress: 192.168.1.1  
Nätmask: 255.255.255.0  
Media 10baseT

*Figur A.6: Nätverskkonfiguration av Telia-routern*

Man måste även konfigurera vilken Domain Name System (DNS) server man har samt vad som ska vara den externa vägen ut och vilken namnserver man ska använda. I vårt fall ser det ut enligt figur A.7 för Campusroutern. Datorerna i det interna nätverket har routerns IP-adress på alla tre posterna.

```
Enter DNS domain name: [] 193.10.220.34
Enter IP address of default route: [none] 130.243.31.1
Enter IP address of primary nameserver: [none] 130.243.31.1
```

*Figur A.7: Nätverkskonfiguration av DNS*

För adsl-servern får man inte upp de frågor som figur A.7 visar då denna initieras via DHCP och får sina adresser från Telias server.

## A.5 Val av installationspaket

Det som återstår av installationen är att välja vilka paket man vill ha med i installationen samt att specificera var ftp-servern finns i nätverket. För att installera från en ftp-server använde vi oss av en Microsoft-klient och ett ftp-program, BulletProof FTP v2.1.5. Microsoft-klienten befann sig inom det interna nätverket. Det går också att installera från någon av de fördefinierade internetserverna men om möjligheten finns att installera en egen ftp-server är det att rekommendera. I tabell A.2 beskrivs de olika paketen som finns till OpenBSD 3.0. Förklaringen till våra val av paket grundas av att vi inte har valt att testa det grafiska gränssnittet X11 som finns till OpenBSD.

### Tabellförklaring

Val : De paket som vi har valt att installera på våra datorer är markerade.

Namn: Paketets namn.

Innehåll: Datamaterialet som paketet innehåller.

Status: Indikerar vilka paketet är nödvändigt för en lyckad installation.

Val	Namn	Innehåll	Status
[X]	<b>Base30.tgz</b>	Innehåller basen för OpenBSD systemet	Nödvändig
[X]	<b>etc30.tgz</b>	Innehåller filerna i /etc	Nödvändig
[X]	<b>Comp30.tgz</b>	Innehåller kompilator och verktyg för denna	Rekommenderad
[X]	<b>man30.tgz</b>	Innehåller manual sidorna	Nödvändig
[ ]	<b>Misc30.tgz</b>	Innehåller misc information och installationsdokument	Valfri
[ ]	<b>Game30.tgz</b>	Innehåller spel till OpenBSD	Valfri
[ ]	<b>Xbase30.tgz</b>	Innehåller basen för installation av X11	Valfri
[ ]	<b>Xfont30.tgz</b>	Innehåller X11 tillbehör	Valfri
[ ]	<b>Xserv30.tgz</b>	Innehåller X server för X11	Valfri
[ ]	<b>Xshare30.tgz</b>	Innehåller manual sidor, inställningar, etc för X11.	Valfri
[X]	<b>Bsd</b>	Detta är kärnan för OpenBSD	Nödvändig

Tabell A.2: Installationspaket till OpenBSD 3.0

## B Konfiguration

I detta avsnittet redovisas vår konfiguration samt en beskrivning av de vanligaste funktionerna som finns i de applikationer vi har använt oss av. Detta är absolut inte en fullständig redovisning av respektive applikation, då detta skulle vara omöjligt att visa med tanke på att uppsatsen ligger på C-nivå.

### B.1 Konfiguration av paketfiltret PF

Paketfiltrets uppgift är att sortera vilka paket som ska få tillträde till routern. För att aktivera det inbyggda paketfiltret i OpenBSD krävs att man redigerar `rc.conf` som finns i mappen `/etc`. I denna fil finns en post där det står `pf=NO`. Posten ska ändras till `pf=YES`. OpenBSD är inte förkonfigurerat för att köra paketfiltret från en standardinstallation. När operativsystemet kör paketfiltret behöver man skriva sina regler för kontrollen. Detta görs i filen `pf.conf` som finns i `/etc` mappen. Som standard står där ingenting och då kommer heller inga paket igenom routern. Det är lämpligt att först exekvera hela systemet öppet för att se att allting fungerar som det ska. För att göra detta skrivs följande till `pf.conf` enligt figur B.8

```
Pass out all
Pass in all
```

*Figur B.8: Förklaring av pass-regeln i paketfiltret*

Detta filtrerar inga paket utan släpper in och ut alla paket som trafikerar routern. Om vi nu vill implementera lite mer kontroll över vad vi vill släppa in behöver vi veta vad det är vi ska skydda oss mot. Vi kommer här nedan att ta upp olika problem och beskriva hur man kan skydda sig mot dessa genom att konfigurera paketfilter-reglerna. Det är av största vikt att veta hur syntaxen fungerar för paketfiltret då detta har stor betydelse för hur paketen kommer att filtreras. Till exempel om vi lägger till en blockering på vår port 22 (ssh) på routern och skriver enligt figur B.9.

```
Pass out all
Pass in all
Block in on r10 inet proto tcp from any to any port 22
```

*Figur B.9: Förklaring av block-regeln i paketfiltret*

Detta kommer inte resultera i den önskade effekten att blockera all trafik som kommer in på port 22 med TCP-protokollet då det har betydelse i vilken ordning reglerna är skrivna. Anledningen till att paketen kommer igenom beror på att det finns en regel som säger `pass in all` före vår blockering. När ett paket kommer in tittar paketfiltret i våra regler och matchar pakethuvudets innehåll mot dessa regler. Den börjar med att titta på det första som står i regellistan och om inte det blev godkänt stegar den neråt i listan för att försöka få igenom en jämförelse med nästa regel. Det är den sista regel som blir matchad som kommer att gälla. Det räcker alltså inte för oss att flytta upp `block` regeln ett steg enligt figur B.10.

```
Pass out all
block in on r10 inet proto tcp from any to any port 22
Pass in all
```

*Figur B.10: Förklaring av ordningsföljden i paketfiltret*

Detta resulterar i att matchningen kommer att fortsätta även om det står en blockering. För att korrigera vår regellista behöver vi alltså skriva något som gör att matchningen bryter när en regel gäller. För detta använder vi oss av `quick` kommandot. Våra regler ser nu ut enligt figur B.11.

```
Pass out all
block in quick on r10 inet proto tcp from any to any port 22
pass in all
```

*Figur B.11: Förklaring av quick-regeln i paketfiltret*

Nu kommer alla paket som har sin destination till port 22 att blockeras och alla andra paket passerar filtret. Det finns ibland problem med IP-adresser som utlåter sig för att vara IP-adresser som finns med i RFC 1816 [6]. Dessa IP-adresser ska inte förekomma på internet



men om dessa uppkommer innebär det oftast att något är fel. För att blockera dessa IP-adresser skriver vi in följande enligt figur B.12 i vår policy.

```
block in quick on r10 inet from { 127.0.0.0/8, 192.168.0.0/16, \
172.16.0.0/12, 10.0.0.0/8 } to any
block out quick on r10 inet from any to { 127.0.0.0/8,
192.168.0.0/16, \
172.16.0.0/12, 10.0.0.0/8 }
pass out all
pass in all
```

*Figur B.12: Blockering av felaktiga IP-adresser*

Att använda sig av `pass in all` och `pass out all` är inte speciellt lyckat då vi vill skapa en relativt skyddad miljö för klienterna i det interna nätverket. För att öppna upp systemet så det går att använda behöver vi veta vilka portar som klientprogrammen använder sig av och sedan implementera regler i paketfiltret för dessa. Till exempel om man vill öppna port 21 för att exekvera en ftp-server på en av klienterna och låta alla klienter skicka ut på alla portar skriver man enligt figur B.13.

```
pass in quick on r10 inet proto tcp from any to any port 21
block in quick on r10 from any to any
pass out on r10 from any to any
```

*Figur B.13: Öppning av portar i paketfiltret*

Det är ofta önskvärt att registrera uppkopplingsförsök på de portar man har valt att blockera för att se om någon eller något försöker att göra ett intrång i systemet. Detta kan göras väldigt lätt med hjälp av kommandot `log` i paketfilterreglerna. Om vi nu skulle vilja se alla uppkopplingsförsök på port 21 som är blockerad så följer ett exempel enligt figur B.14.

```
block in log quick on r10 inet proto tcp from any to any port 21
```

*Figur B.14: Förklaring av log regeln i paketfiltret*

De paket som registreras sparas binärt i mappen `/var/log/pflog`. Om man nu vill undersöka vilka uppkopplings försök som har förekommit kan man skriva in ett kommando i kommandoprompten enligt figur B.15. Dock ser man inte det som händer i realtid.

```
# tcpdump -n -e -ttt -r /var/log/pflog
```

*Figur B.15: Förklaring av kommandot tcpdump*

Det är ofta så att någon form av multimedia används av klienterna och när det rör den här typen av program skickas data både över TCP- och UDP-protokollet. För att slippa skriva två regler kan man skriva protokolltyperna mellan två ”måsvingar”. Till exempel om man har en spelservr som exekverar på port 27015 och denna använder sig både av TCP och UDP kan man skriva enligt figur B.16.

```
pass in quick on rl0 inet proto { tcp, udp } from any to any port  
27015
```

*Figur B.16: Förklaring av protokolltyper i paketfiltret*

Om paketen som anländer eller skickas från routern är fragmenterade kan man använda sig av en regel som ordnar paketen innan leverans. Detta görs med kommandot `scrub`, se figur B.17. Anledningen till att det kan vara bra att ordna paketen är att vissa operativsystem kan få svårigheter med oordnade paket. Dock är detta ett ganska systemkrävande kommando och behöver inte användas om implementationen av TCP/IP protokollet är välgjort.

```
Scrub in all
```

*Figur B.17: Förklaring av scrub kommandot*

Om man byter ut ett nätverkskort eller ändrar sin IP-adress kan det vara en mycket bra idé att skriva om sina regler så att en sådan ändring inte innebär att man behöver redigera alla sina regler utan bara på en post. Exemplet enligt figur B.18 visar hur detta kan skrivas.

```

DoNotRoute="{ 127.0.0.0/8, 192.168.0.0/16, 172.16.0.0/12,
10.0.0.0/8 }"
ExtCard="r10"
block in quick on $ExtCard from $DoNotRoute to any

```

*Figur B.18: Förklaring av namnsättning i paketfiltret*

Med hjälp av den information som har redovisats i detta kapitel är det möjligt att skapa en relativt säker konfiguration av sitt paketfilter.

## **B.2 Konfiguration av Network Address Translation (NAT)**

Den fil i OpenBSD som konfigurerar NAT finns i `/etc` och heter `nat.conf`. I denna fil skriver man vilka interna IP-adresser som ska länkas till det externa nätverket. Syntaxen ger en bra överskådlighet över länknigen. Implementationen av NAT sker på routern och här sker också länkning av vidarebefodring till specifika IP-adresser med `rdx` kommandot. Kommandot `rdx` står för `redirect`, översatt till svenska ”omlänking”. För att få routern att vidareförmedla paket till sina klienter måste man deklarerera `net.inet.ip.forwarding=1` i filen `/etc/sysctl.conf`. Paketfiltret måste vara aktiverat för att det ska vara möjligt att köra NAT. I bilaga A.2.1 finns beskrivning hur detta aktiveras. Vi börjar med att visa NAT-syntaxen. Figur B.19 visar ett utdrag från `nat.conf` med förklaring över `nat` kommandot.

```

nat on r10 from 192.168.1.0/24 to any -> 130.243.31.229

```

`nat` : Indikerar att kommandot man ger är en NAT regel  
`r10` : Nätverks gränssnittet mot det externa nätverket  
`192.168.1.0/24` : IP adressen och nätmasken i Classless Internet Domain Routing (CIDR) format.  
`130.243.31.229` : Den externa IP adressen.

*Figur B.19: Förklaring av nat-syntaxen*

För att länka portar till interna IP-adresser med NAT används som tidigare deklarerats `rdx` kommandot. Detta kommando ger oss alltså möjligheten att länka en dator i det interna

nätverket till en port på routern så att externa IP-adresser kan komma åt datorerna på det interna nätverket. Figur B.20 visar utdrag från `nat.conf` med förklaring över `rdr` kommandot.

```
rdr on r10 from any to any port 80 -> 192.168.1.11 port 80
```

`rdr` : Indikerar att kommandot man ger är en `rdr` regel

`r10` : Nätverks gränssnittet mot det externa nätverket

`any to any` : Beskriver vilka adresser som är inblandade, i exemplet är det alltså alla.

`port 80` : vilken port i datorn som är inblandad.

`192.168.1.11 port 80` : den datorn samt port som routern ska vidarebefordra paketen till.

*Figur B.20: Förklaring av `rdr`-syntaxen*

Med hjälp av information som redovisats i detta kapitel är det möjligt att utföra de länkningsman har tänkt sig.

### **B.3 Konfiguration av Apache**

Eftersom Apache webbserver finns med i grundinstallationen av OpenBSD så är det enda man behöver göra att sätta igång den och se till att den startar upp varje gång datorn startar. Det gör mn genom att gå in i filen `/etc/rc.conf.local` och ändra följande rad `httpd_flags=NO` till `httpd_flags=""`. Man kan även starta Apache webbserver manuellt genom att gå in i följande katalog `/usr/sbin/` och skriva `apachectl start`. Då startar webbservern upp direkt utan att man behöver starta om datorn. Om detta lyckades ska man kunna surfa in på webbservern och man får då upp en fördefinierad hemsida. Hemsidan ligger i en katalog som heter `/var/www/htdocs`. Om man har en ftp-server aktiverad på webbservern kan man lätt byta ut och uppdatera hemsidan från en annan dator, genom att helt enkelt ladda upp en ny och ersätta den tidigare.

I Apache webbserver är det enkelt att låta varje användare få sin personliga hemsida. För att aktivera det går man in i filen `/var/www/conf/httpd.conf` och redigerar raden `UserDir disabled` enligt figur B.21.

```

UserDir public_html

<Directory /home/*/public_html>
AllowOverride FileInfo AuthConfig Limit
Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec
<Limit GET POST OPTIONS PROPFIND>
Order allow,deny
Allow from all
</Limit>

<Limit PUT DELETE PATCH PROPPATCH MKCOL COPY MOVE LOCK UNLOCK>
Order deny,allow
Deny from all
</Limit>
</Directory>

```

*Figur B.21: Personliga hemsidor*

Man måste här skapa en mapp i varje konto där webbsidorna ska ligga, om det inte redan finns en. Den ska heta `public_html`. Om exempelvis användare `olsson` ska ha en personlig hemsida bli hans mapp `/home/olsson/public_html`. Den adress som tilldelas användaren `olsson` för sin personliga hemsida blir då i vårt fall `http://130.243.31.229/~olsson`. Man kan testa sin hemsida direkt via webbservern om man inte har någon annan dator att testa med. Lynx är en textbaserad webb-läsare som kan öppna mindre avancerade hemsidor.

Om man vill få sin webbserver säkrare så finns det några åtgärder man kan göra. Då man har personliga hemsidor för varje användare kan man se till att rootkontot inte har något webbutrymme. Detta görs genom att skriva följande rad `UserDir disabled root`. Även trafiken kan bli mycket säkrare genom att aktivera SSL, då detta ger en krypterad kommunikation. Det gör man genom att ändra samma rad i `rc.conf` som man startade upp webbservern med till `httpd_flags="-DSSL"` och sedan starta om webbservern. Det går också att göra det manuellt genom att skriva `apachectl stop` och sedan `apachectl startssl` i kommandotolken. På apaches hemsida [9] så kan man läsa en mer detaljerad beskrivning av hur man går tillväga och även läsa om andra funktioner så som att skapa personliga webbsidor och webbstatistik.

## B.4 Konfiguration av FTP

För att starta upp en ftp-server krävs att man ändrar i `rc.conf`. På den raden där det står `ftpd_flags=NO` ändrar man `ftpd_flags="D11USA"`. Nu kommer en ftp-server att starta varje gång servern startas upp. Det går givetvis att starta ftp applikationen manuellt genom att skriva kommandot `/usr/libexec/ftpd -D11USA` från kommandotolken. I figur B.22 ges en förklaring kring de flaggor vi har hittat.

D : Accepterar uppkoppling på ftp-prorten.

l : Loggar alla korrekta och felaktiga ftp-sektioner i syslog.

Med dubbel l (-ll) kommer också inkludera: retrieve (get), store (put), append, delete, create directory, remove directory, rename operations.

U : Loggar varje uppkopplingssektion och sparar dem i `/var/run/utmp`, och gör dem synliga för användare så att till exempel kommando som `who(1)` fungerar.

S : Loggar alla anonyma nedladdningar om filen `/var/log/ftpd` existerar.

A : Tillåter bara anonyma uppkopplingar eller konton listade i `/etc/ftpchroot`. Andra uppkopplingar tillåts ej.

*Figur B.22: Förklaring av ftp-flaggor*

De filer som styr applikationens utseende mot klienter och innehåll förklaras i tabell B.3.

Filnamn	Innehåll
<code>/etc/nologin</code>	Denna fil kan användas för att stänga av ftp-access.
<code>/etc/ftpwelcome</code>	Välkomstmeddelande. I denna fil kan man skriva in hur välkomstmeddelandet ska se ut som kommer upp när man kopplat upp sig på ftp-servern.
<code>/etc/motd</code>	Välkomstmeddelande efter login. Innehåller meddelande som kommer upp när man loggat in på ftp-servern.
<code>.message</code>	Denna fil kan man placera i alla mappar. Det kommer visas varje gång användaren öppnar mappen.
<code>/var/run/utmp</code>	En lista på alla användare i systemet.
<code>/var/log/ftpd</code>	Logg fil för nedladdning.

*Tabell B.3: Övriga filer för ftp-servern*

## B.5 Konfiguration av NFS

Starta upp en NFS filserver görs genom att ändra i filen `/etc/rc.conf`. I filen finns en post där det står `nfs_server=NO`. Denna post ska ändras till `nfs_server=YES`. För att tilldela åtkomsträttigheter till en katalog ändrar man i `/etc/exports`. Här är ett exempel på hur man lägger till en katalog och bestämmer vilka som ska kunna komma åt katalogen. `/work -alldirs -ro -network 10.0.0 -mask 255.255.255.0`. Detta innebär att katalogen `work` kommer bli tillgänglig via NFS filserver. Med `"-alldirs -ro"` indikeras att man kan komma åt allt inom `work` med endast läsrättigheter. För att specificera vilka som ska komma åt katalogen beskriver man med IP-nummer `10.0.0.0` och med en nätmask som är `255.255.255.0`. I filen `/etc/exports` beskriver man vad som ska kunna accessas och vilka som har rätt att göra det.

## B.6 Konfiguration av DHCP

I detta avsnitt kommer vi att stega igenom de nödvändigaste filerna som behöver redigeras för att kunna använda sig av DHCP-tjänsten. Detta inkluderar både server-sidan och klient-sidan. OpenBSD har ett mycket prominent stöd för denna funktion och applikationerna är lättarbetade. Dessa är `dhcpcd` och `dhclient`. Servern använder sig av `dhcpcd` och klienten använder sig av `dhclient`. För att inleda en konfiguration börjar vi konfigurera `dhcpcd.conf` som finns i mappen `/etc`. Filen har vi redigerat enligt figur B.23. En förklaring över koden finns i figuren.

```

# Network:                192.168.1.0/255.255.255.0
# Domain name:            campus.test.nu
# Name servers:           193.10.220.34
# Default router:         192.168.1.1
# Addresses:              192.168.1.32 - 192.168.1.127
#
shared-network LOCAL-NET {
    option domain-name "campus.test.nu";
    option domain-name-servers 193.10.220.34;

    subnet 192.168.1.0 netmask 255.255.255.0{
        option routers 192.168.1.1;

        range 192.168.1.32 192.168.1.127;
    }
}

```

*Figur B.23: Konfiguration av dhcpd.conf*

När väl man har redigerat `dhcpd.conf` till önskat innehåll, behöver man skapa en fil som ska heta `dhcpd.leases` i mappen `/var/db/`. Denna fil behöver inte innehålla någon kod utan den ska bara finnas i systemet. I `dhcpd.leases` kommer de datorer som får information från DHCP-servern att lagras med tillhörande information. Detta för att DHCP-servern ska kunna dela ut samma information som den vid ett tidigare tillfälle gjorde till just den klienten. Filen `dhcpd.interfaces` finns i `/etc` är det sista man behöver konfigurera för att starta en DHCP-server. I denna ska det enbart stå nätverkskortnamnet på det nätverkskort som ska agera som DHCP-server. I vårt fall står det bara `r11` i denna. För att sedan starta DHCP-server skrivs bara `dhcpd` i kommandoprompten.

För att konfigurera klientdatorn börjar vi med att redigera nätverkskortet att lyssna efter en DHCP-server. Detta görs genom att redigera filen `hostname.***`. Denna fil ska vara konfigurerad enligt figur B.24.



```
dhcp NONE NONE NONE
```

*Figur B.24: Konfiguration av hostname.conf för DHCP*

Nästa fil att redigera heter `dhclient.conf` och är lokerad i `/etc`. Det är denna fil som styr vad som ska lagras från DHCP-server. Vi har konfigurerat vår enligt figur B.25

```
initial-interval 1;
send host-name "slasken";
request subnet-mask, broadcast-address, routers,
        domain-name, domain-name-servers, host-name;
```

*Figur B.25: Konfiguration av dhclient.conf*

Nu är klientkonfigurationen klar och det som återstår är att starta applikationen på klienten. En förutsättning för att detta ska fungera är att man har startat `dhcpd` på servern. För att starta klienten skrivs kommandot `dhclient` följt av nätverkskort namnet, alltså till exempel för vårt system skrivs `dhclient ep0` in i kommandotolken. Nu börjar systemet leta efter en DHCP-server och förhoppningsvis kommer ett svar från denna och man har då fått en IP-adress, nätmask, DNS och standard router. För att undersöka om man fått information kan följande kommando skrivas in i kommandotolken: `ifconfig ep0`. Detta ger en översikt om vilken information som är kopplat till nätverkskortet.

## **B.7 Konfiguration av Qadsl**

Konfigurationen utgick från en dator installerad med OpenBSD. För att få datorn att agera som en router följde vi kapitel 3.2 ,3.3 och 3.4. Programvaran laddades ned från distributörens hemsida[11] och extraherades på en Microsoft Windows klient. De filer som framkom efter extraheringnen fördes över till Telia-routerns ftp-server. För att sedan installera programvaran som vi placerade i mappen `/home/ftp` använde vi oss av först `make bsd` följt av `make install` från kommandotolken. Kommandona skapar de nödvändiga filerna och placerar ut

dem i systemet där de ska vara. Ett krav är att man har installerat datorn med paketet `comp30.tgz` för att kompileringen ska vara möjlig. Den enda filen man behöver redigera är `qadsl.conf` som finns i mappen `/etc`. I denna fil ska man skriva in sitt login namn och sitt lösenord enligt utdraget i figur B.26. För att sedan starta applikationen skriver man följande i kommandotolten: `/usr/local/sbin/qadsl -l`.

```
# Username
USER=<Ditt loginnamn>
# Password
PASS=<Ditt lösenord>
```

*Figur B.26: Qadsl.conf beskrivning*

## C Tabeller

### WEBBSERVER

Test av stor fil 100 megabyte	storlek (kb)	tid (s)	kb/sek
Tid för nedladdning från externt nät, test 1	102918	348,01	295,73
Tid för nedladdning från externt nät, test 2	102918	353,13	291,45
Tid för nedladdning från externt nät, test 3	102918	353,9	290,81
Test av stor fil 100 megabyte			<b>292,66</b>
Tid för nedladdning från internt nät, test 1	102918	320,83	320,79
Tid för nedladdning från internt nät, test 2	102918	312,27	329,58
Tid för nedladdning från internt nät, test 3	102918	315,51	326,20
			<b>325,52</b>
Test av mindre fil 10 megabyte	storlek (kb)	tid (s)	kb/sek
Tid för nedladdning från externt nät, test 1	9908,6	31,98	309,84
Tid för nedladdning från externt nät, test 2	9908,6	31,02	319,43
Tid för nedladdning från externt nät, test 3	9908,6	37,95	261,10
Test av mindre fil 10 megabyte			<b>296,79</b>
Tid för nedladdning från internt nät, test 1	9908,6	30,46	325,30
Tid för nedladdning från internt nät, test 2	9908,6	37,57	263,74
Tid för nedladdning från internt nät, test 3	9908,6	30,12	328,97
			<b>306,00</b>
Inläsning av index.html	storlek (kb)	tid (s)	kb/sek
Tid för nedladdning från externt nät, test 1	1200,3	4,4	272,80
Tid för nedladdning från externt nät, test 2	1200,3	4,2	285,79
Tid för nedladdning från externt nät, test 3	1200,3	4,24	283,09
			<b>280,56</b>
Inläsning av index.html	storlek (kb)	tid (s)	kb/sek
Tid för nedladdning från internt nät, test 1	1200,3	4,05	296,37
Tid för nedladdning från internt nät, test 2	1200,3	4,33	277,21
Tid för nedladdning från internt nät, test 3	1200,3	4,15	289,23
			<b>287,60</b>

Tabell C.1: Webbservertest

## FTP-server

	Storlek	tid (s)	kb/sek
Test av många små filer: 100 st totalstorlek 4 816 89 byte(kb)			
Tid för nedladdning från externt nät, test 1	4704	25,45	184,83
Tid för nedladdning från externt nät, test 2	4704	32,47	144,87
Tid för nedladdning från externt nät, test 3	4704	31,32	150,19
			<b>159,97</b>

	Storlek	tid (s)	kb/sek
Test av många små filer: 100 st totalstorlek 4 816 89 byte(kb)			
Tid för uppladdning från externt nät, test 1	4704	164,86	28,53
Tid för uppladdning från externt nät, test 2	4704	165,45	28,43
Tid för uppladdning från externt nät, test 3	4704	150,57	31,24
			<b>29,40</b>

	Storlek	tid (s)	kb/sek
Test av många små filer: 100 st totalstorlek 4 816 89 byte(kb)			
Tid för nedladdning från internt nät, test 1	4704	32,63	144,16
Tid för nedladdning från internt nät, test 2	4704	22,91	205,33
Tid för nedladdning från internt nät, test 3	4704	26,48	177,64
			<b>175,71</b>

	Storlek	tid (s)	kb/sek
Test av många små filer: 100 st totalstorlek 4 816 89 byte(kb)			
Tid för uppladdning från internt nät, test 1	4704	95,59	49,21
Tid för uppladdning från internt nät, test 2	4704	106,1	44,34
Tid för uppladdning från internt nät, test 3	4704	93,33	50,40
			<b>47,98</b>

	Storlek	tid (s)	kb/sek
Test av en mindre fil: filstorlek 10 146 442 byte	(kb)		
Tid för nedladdning från externt nät, test 1	9909	12,09	819,60
Tid för nedladdning från externt nät, test 2	9909	12,41	798,47
Tid för nedladdning från externt nät, test 3	9909	12,05	822,32
			<b>813,47</b>

	Storlek	tid (s)	kb/sek
Test av en mindre fil: filstorlek 10 146 442 byte	(kb)		
Tid för uppladdning från externt nät, test 1	9909	12,35	802,35
Tid för uppladdning från externt nät, test 2	9909	13,44	737,28
Tid för uppladdning från externt nät, test 3	9909	14,22	696,84
			<b>745,49</b>

	Storlek		
	(kb)	tid (s)	kb/sek
Test av en mindre fil: filstorlek 10 146 442 byte			
Tid för nedladdning från internt nät, test 1	9909	10,37	955,54
Tid för nedladdning från internt nät, test 2	9909	10,50	943,71
Tid för nedladdning från internt nät, test 3	9909	10,52	941,92
			<b>947,06</b>
	Storlek		
	(kb)	tid (s)	kb/sek
Test av en mindre fil: filstorlek 10 146 442 byte			
Tid för uppladdning från internt nät, test 1	9909	11,00	900,82
Tid för uppladdning från internt nät, test 2	9909	11,53	859,41
Tid för uppladdning från internt nät, test 3	9909	12,57	788,31
			<b>849,51</b>
	Storlek		
	(kb)	tid (s)	kb/sek
Test av stor fil: filstorlek 105 388 032 bytes			
Tid för nedladdning från externt nät, test 1	102918	126,93	810,82
Tid för nedladdning från externt nät, test 2	102918	128,28	802,29
Tid för nedladdning från externt nät, test 3	102918	127,56	806,82
			<b>806,65</b>
	Storlek		
	(kb)	tid (s)	kb/sek
Test av stor fil: filstorlek 105 388 032 bytes			
Tid för uppladdning från externt nät, test 1	102918	129,08	797,32
Tid för uppladdning från externt nät, test 2	102918	130,08	791,19
Tid för uppladdning från externt nät, test 3	102918	130,05	791,37
			<b>793,29</b>
	Storlek		
	(kb)	tid (s)	kb/sek
Test av stor fil: filstorlek 105 388 032 bytes			
Tid för nedladdning från internt nät, test 1	102918	118,93	865,37
Tid för nedladdning från internt nät, test 2	102918	120,3	855,51
Tid för nedladdning från internt nät, test 3	102918	122,05	843,24
			<b>854,71</b>
	Storlek		
	(kb)	tid (s)	kb/sek
Test av stor fil: filstorlek 105 388 032 bytes			
Tid för uppladdning från internt nät, test 1	102918	124,08	829,45
Tid för uppladdning från internt nät, test 2	102918	126,12	816,03
Tid för uppladdning från internt nät, test 3	102918	125,6	819,41
			<b>821,63</b>

Tabell C.2: Ftp-servertest

## ROUTER

Skicka en stor fil : filstorlek 592 588 752 bytes	Storlek (kb)	tid (s)	kb/sek
Tid för uppladdning från internt till externt test 1	578700	105,54	5483,23
Tid för uppladdning från internt till externt test 2	578700	106,20	5449,15
Tid för uppladdning från internt till externt test 3	578700	105,23	5499,38
			<b>5477,25</b>
Skicka en stor fil : filstorlek 592 588 752 bytes	Storlek (kb)	tid (s)	kb/sek
Tid för nedladdning från internt till externt test 1	578700	164,14	3525,65
Tid för nedladdning från internt till externt test 2	578700	163,48	3539,88
Tid för nedladdning från internt till externt test 3	578700	165,03	3506,64
			<b>3524,06</b>
Skicka en stor fil : filstorlek 592 588 752 bytes	storlek (kb)	tid (s)	kb/sek
Tid för uppladdning från externt till internt test 1	578700	106,86	5415,50
Tid för uppladdning från externt till internt test 2	578700	105,90	5464,59
Tid för uppladdning från externt till internt test 3	578700	105,53	5483,75
			<b>5454,61</b>
Skicka en stor fil : filstorlek 592 588 752 bytes	storlek (kb)	tid (s)	kb/sek
Tid för nedladdning från externt till internt test 1	578700	137,85	4198,04
Tid för nedladdning från externt till internt test 2	578700	136,48	4240,18
Tid för nedladdning från externt till internt test 3	578700	138,50	4178,34
			<b>4205,52</b>
Skicka en mindre fil: 10 146 442 byte	storlek (kb)	tid (s)	kb/sek
Tid för uppladdning från internt till externt test 1	9908,6	2,76	3590,072
Tid för uppladdning från internt till externt test 2	9908,6	2,22	4463,333
Tid för uppladdning från internt till externt test 3	9908,6	2,46	4027,886
			<b>4027,097</b>
Skicka en mindre fil: 10 146 442 byte	storlek (kb)	tid (s)	kb/sek
Tid för nedladdning från internt till externt test 1	9908,6	2,87	3452,474
Tid för nedladdning från internt till externt test 2	9908,6	2,8	3538,786
Tid för nedladdning från internt till externt test 3	9908,6	2,95	3358,847
			<b>3450,036</b>
Skicka en mindre fil: 10 146 442 byte	storlek (kb)	tid (s)	kb/sek
Tid för uppladdning från externt till internt test 1	9908,6	2,2	4503,909
Tid för uppladdning från externt till internt test 2	9908,6	2,19	4524,475
Tid för uppladdning från externt till internt test 3	9908,6	2,14	4630,187
			<b>4552,857</b>

Skicka en mindre fil: 10 146 442 byte	storlek (kb)	tid (s)	kb/sek
Tid för nedladdning från externt till internt test 1	9908,6	2,51	3947,649
Tid för nedladdning från externt till internt test 2	9908,6	2,47	4011,579
Tid för nedladdning från externt till internt test 3	9908,6	2,48	3995,403
			<b>3984,877</b>
Skicka många filer över ftp: 1000 filer Totalstorlek 16 384 000 byte: Minsta fil 1 kb: Största fil 30 kb			
	storlek (kb)	tid (s)	kb/sek
Tid för uppladdning från internt till externt test 1	16000	217,42	73,59
Tid för uppladdning från internt till externt test 2	16000	215,34	74,30
Tid för uppladdning från internt till externt test 3	16000	218,35	73,28
			<b>73,72</b>
Skicka många filer över ftp: 1000 filer			
	storlek (kb)	tid (s)	kb/sek
Tid för nedladdning från externt till internt test 1	16000	223,14	71,70
Tid för nedladdning från externt till internt test 2	16000	224,87	71,15
Tid för nedladdning från externt till internt test 3	16000	223,32	71,65
			<b>71,50</b>
Skicka många filer över ftp: 1000 filer			
	storlek (kb)	tid (s)	kb/sek
Tid för uppladdning från externt till internt test 1	16000	232,31	68,87
Tid för uppladdning från externt till internt test 2	16000	234,81	68,14
Tid för uppladdning från externt till internt test 3	16000	231,43	69,14
			<b>68,72</b>
Skicka många filer över ftp: 1000 filer			
	storlek (kb)	tid (s)	kb/sek
Tid för nedladdning från internt till externt test 1	16000	202,87	78,87
Tid för nedladdning från internt till externt test 2	16000	200,90	79,64
Tid för nedladdning från internt till externt test 3	16000	200,52	79,79
			<b>79,43</b>

*Tabell C.3: Routertest*

## DIREKTUPPKOPPLING

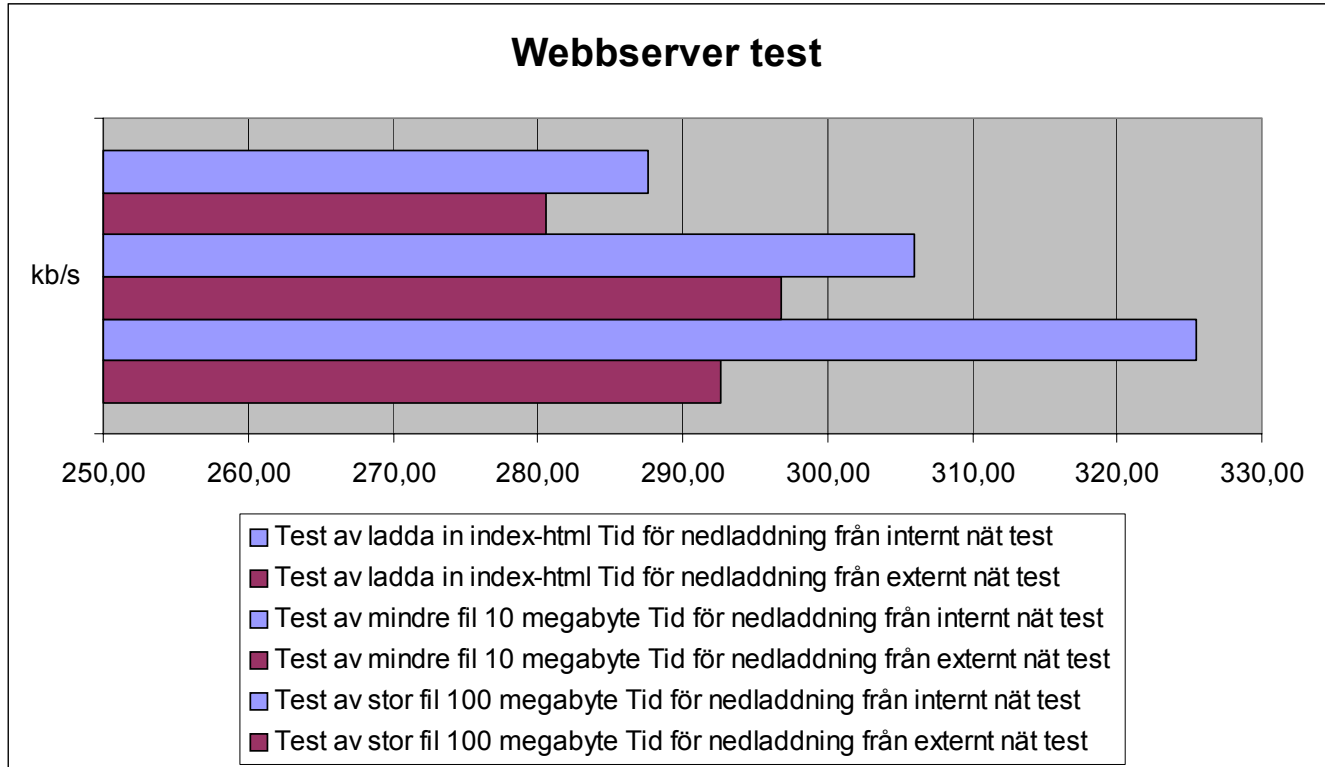
Skicka en stor fil : filstorlek 592 588 752 bytes	Storlek (kb)	tid (s)	kb/sek
Tid för uppladdning från internt till externt test 1	578700	89,85	6440,73
Tid för uppladdning från internt till externt test 2	578700	89,70	6451,51
Tid för uppladdning från internt till externt test 3	578700	89,30	6480,40
			<b>6457,55</b>
Skicka en stor fil : filstorlek 592 588 752 bytes	Storlek (kb)	tid (s)	kb/sek
Tid för nedladdning från internt till externt test 1	578700	97,74	5920,81
Tid för nedladdning från internt till externt test 2	578700	98,37	5882,89
Tid för nedladdning från internt till externt test 3	578700	97,90	5911,13
			<b>5904,95</b>
Skicka en stor fil : filstorlek 592 588 752 bytes	Storlek (kb)	tid (s)	kb/sek
Tid för uppladdning från externt till internt test 1	578700	83,64	6918,938
Tid för uppladdning från externt till internt test 2	578700	81,76	7078,033
Tid för uppladdning från externt till internt test 3	578700	83,84	6902,433
			<b>6966,468</b>
Skicka en stor fil : filstorlek 592 588 752 bytes	Storlek (kb)	tid (s)	kb/sek
Tid för nedladdning från externt till internt test 1	578700	67,74	8542,958
Tid för nedladdning från externt till internt test 2	578700	65,93	8777,491
Tid för nedladdning från externt till internt test 3	578700	66,1	8754,917
			<b>8691,789</b>
Skicka en mindre fil: 10 146 442 byte	Storlek (kb)	tid (s)	kb/sek
Tid för uppladdning från internt till externt test 1	9908,6	1,95	5081,33
Tid för uppladdning från internt till externt test 2	9908,6	1,54	6434,16
Tid för uppladdning från internt till externt test 3	9908,6	1,64	6041,83
			<b>5852,44</b>
Skicka en mindre fil: 10 146 442 byte	Storlek (kb)	tid (s)	kb/sek
Tid för nedladdning från internt till externt test 1	9908,6	1,62	6116,42
Tid för nedladdning från internt till externt test 2	9908,6	1,73	5727,51
Tid för nedladdning från internt till externt test 3	9908,6	1,64	6041,83
			<b>5961,92</b>
Skicka en mindre fil: 10 146 442 byte	Storlek (kb)	tid (s)	kb/sek
Tid för uppladdning från externt till internt test 1	9908,6	1,53	6476,21
Tid för uppladdning från externt till internt test 2	9908,6	1,67	5933,29
Tid för uppladdning från externt till internt test 3	9908,6	1,71	5794,50
			<b>6068,00</b>



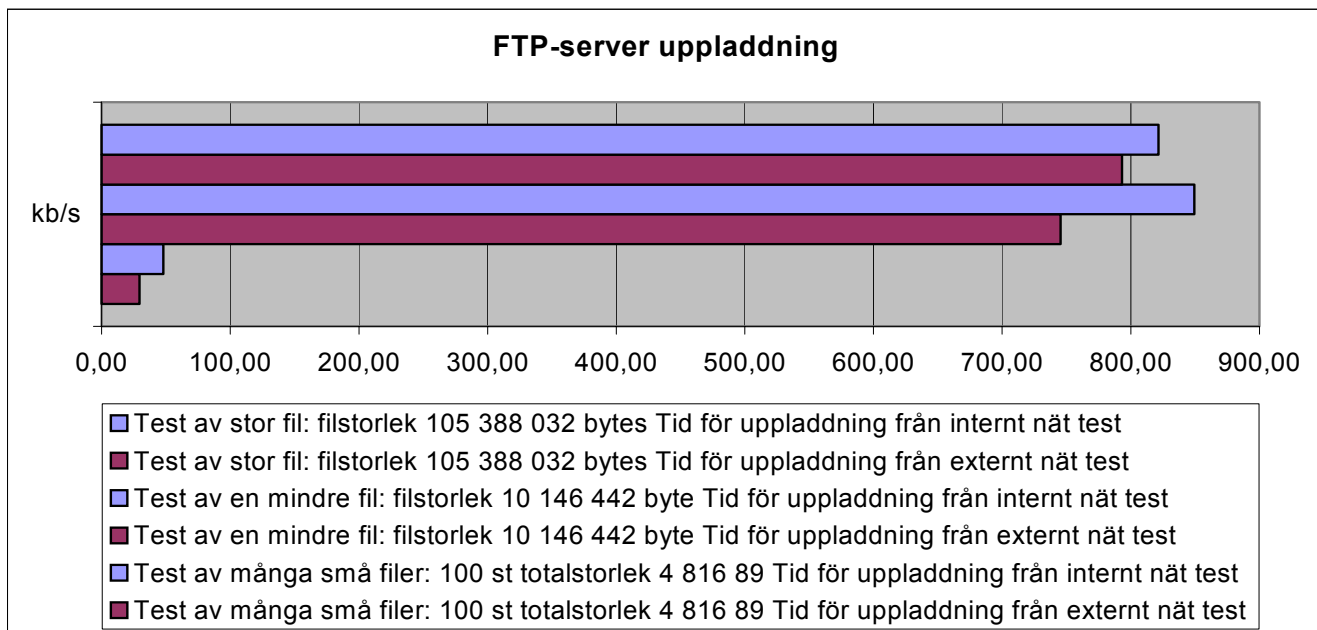
Skicka en mindre fil: 10 146 442 byte	Storlek (kb)	tid (s)	kb/sek
Tid för nedladdning från externt till internt test 1	9908,6	1,41	7027,38
Tid för nedladdning från externt till internt test 2	9908,6	1,5	6605,73
Tid för nedladdning från externt till internt test 3	9908,6	1,59	6231,82
			<b>6621,64</b>
Skicka många filer över ftp: 1000 filer Totalstorlek 16 384 000 byte : Minsta fil 1 kb : Största fil 2 kb			
Skicka många filer över ftp: 1000 filer	Storlek (kb)	tid (s)	kb/sek
Tid för uppladdning från internt till externt test 1	16000	201,78	79,29
Tid för uppladdning från internt till externt test 2	16000	203,13	78,77
Tid för uppladdning från internt till externt test 3	16000	202,5	79,01
			<b>79,02</b>
Skicka många filer över ftp: 1000 filer	Storlek (kb)	tid (s)	kb/sek
Tid för nedladdning från internt till externt test 1	16000	201,5	79,40
Tid för nedladdning från internt till externt test 2	16000	202,03	79,20
Tid för nedladdning från internt till externt test 3	16000	201,7	79,33
			<b>79,31</b>
Skicka många filer över ftp: 1000 filer	Storlek (kb)	tid (s)	kb/sek
Tid för uppladdning från externt till internt test	16000	201,32	<b>79,48</b>
Skicka många filer över ftp: 1000 filer	Storlek (kb)	tid (s)	kb/sek
Tid för nedladdning från externt till internt test	16000	200,78	<b>79,69</b>

*Tabell C.4: Direktuppkopplingstest*

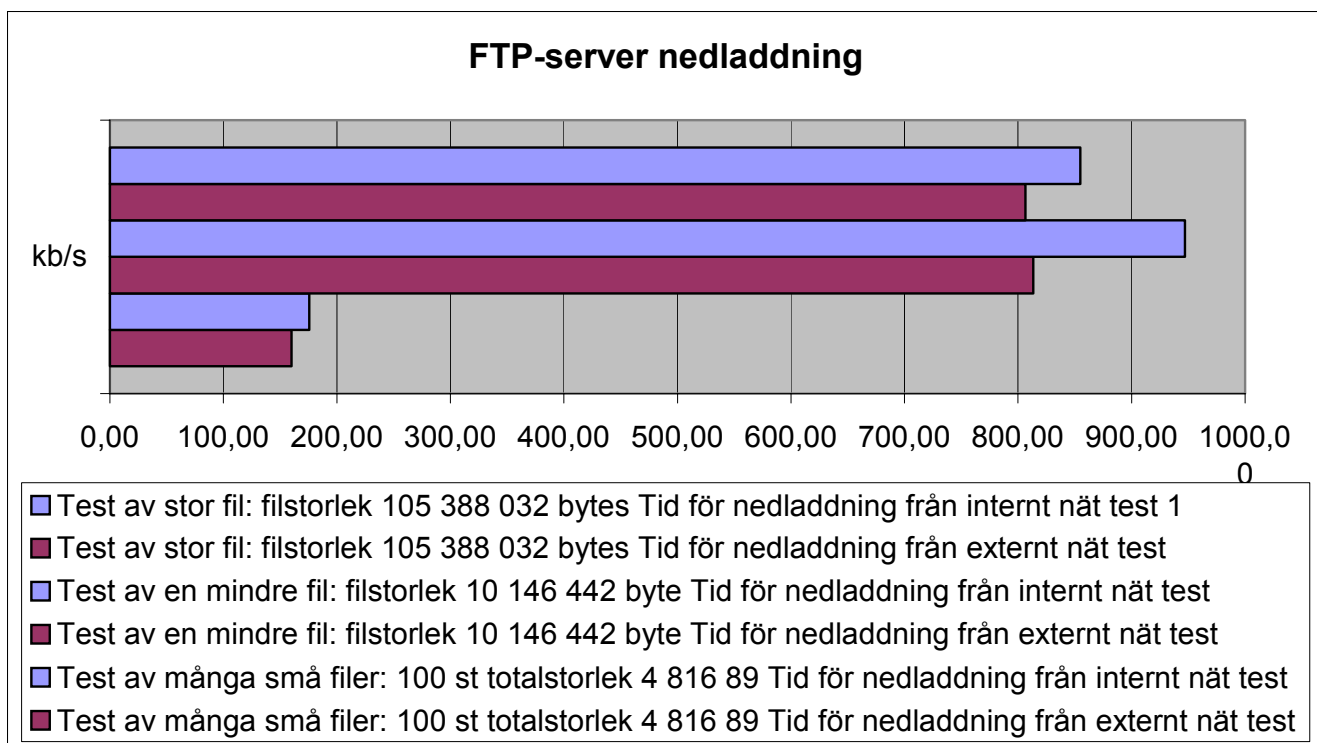
## D Diagram



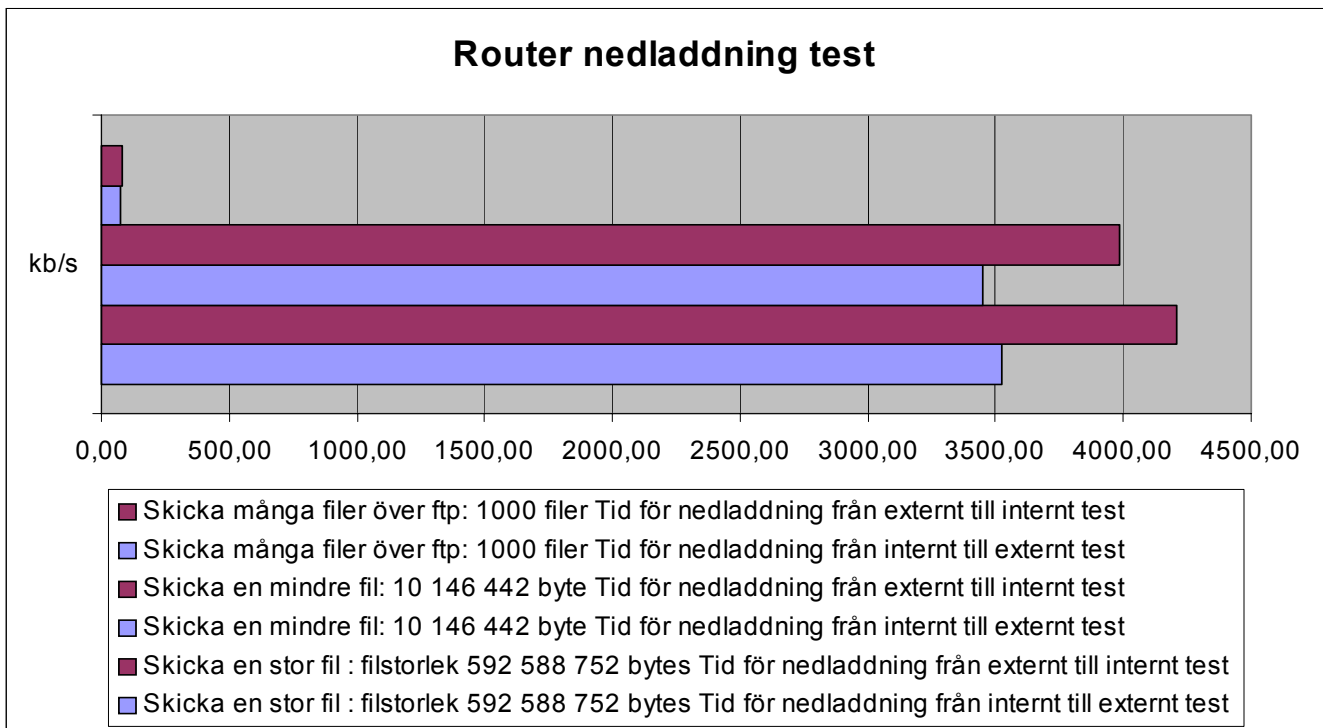
*Diagram D.1: Webbservertest*



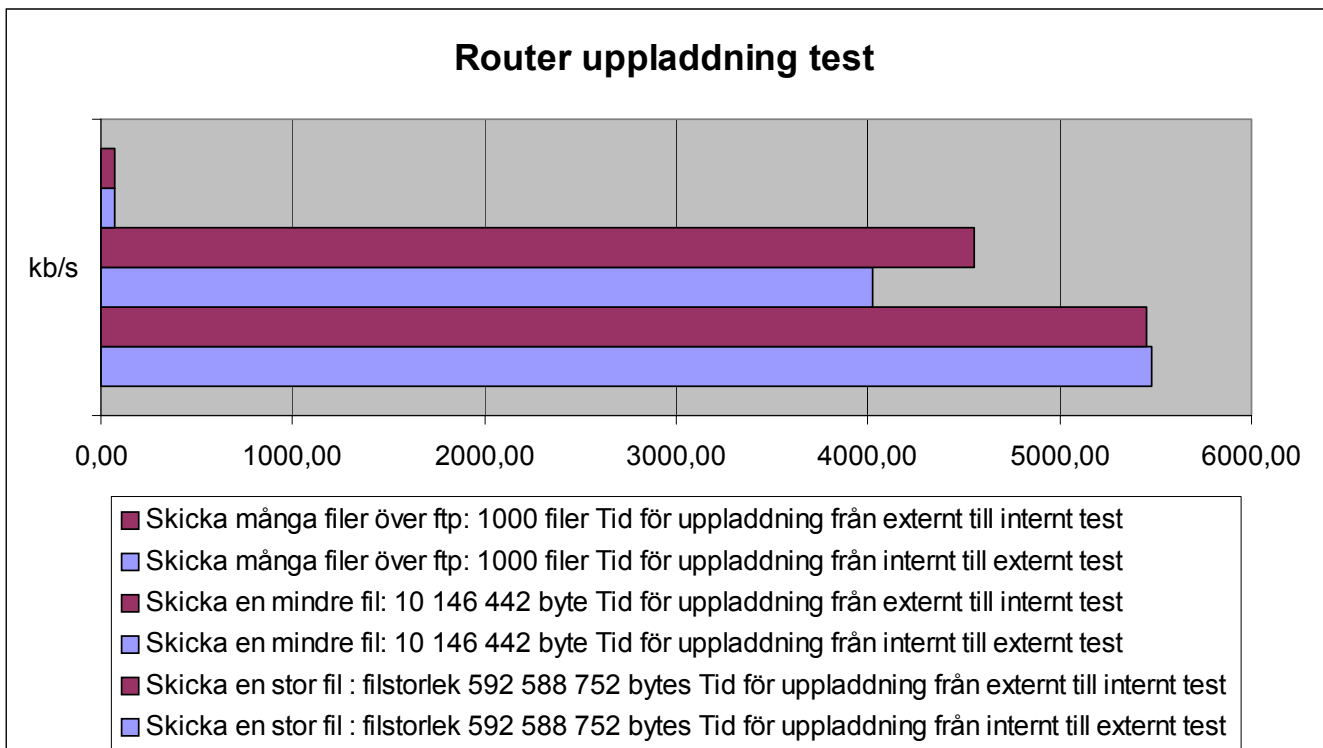
*Diagram D.2: Ftp-servertest, uppladdning*



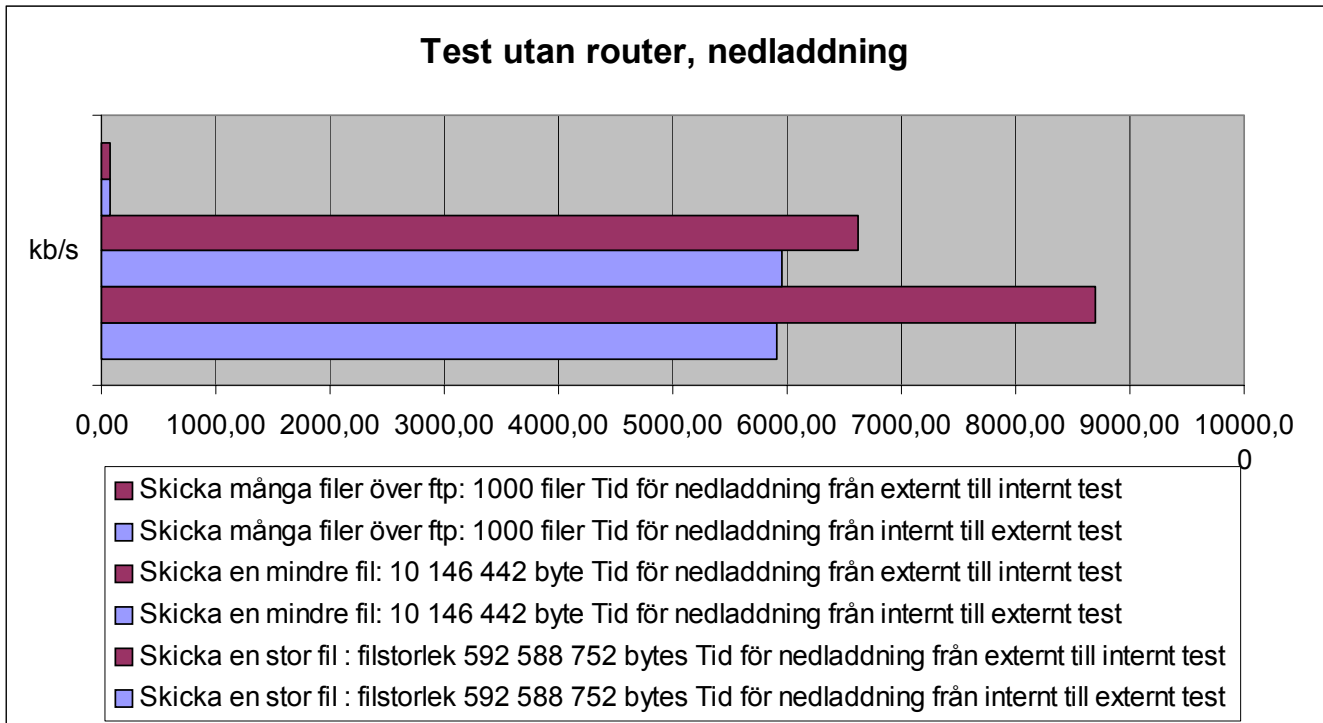
*Diagram D.3: Ftp-servertest, nedladdning*



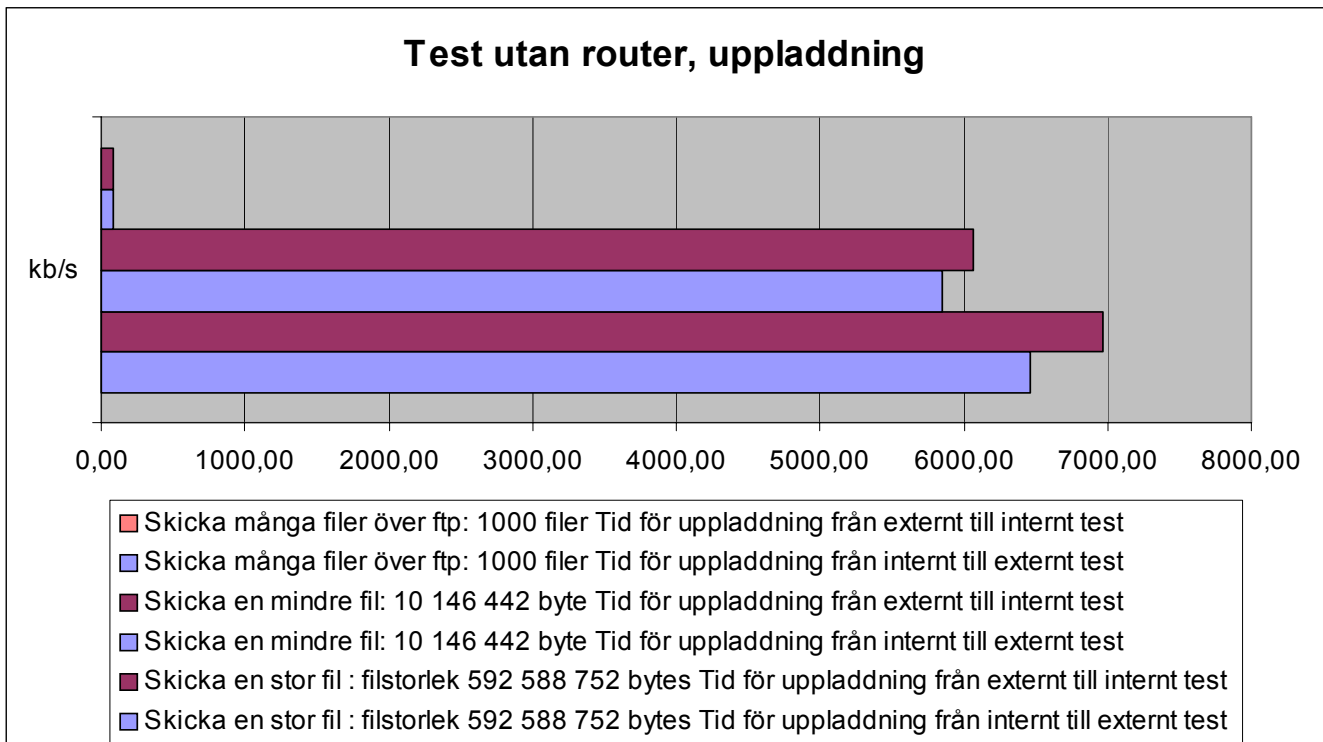
*Diagram D.4: Routertest, nedladdning*



*Diagram D.5: Routertest, uppladdning*



*Diagram D.6: Test utan router, nedladdning*



*Diagram D.7: Test utan router, uppladdning*

### Jämförelse router/100 Mbit Switch, 600 Mb fil nedladdning

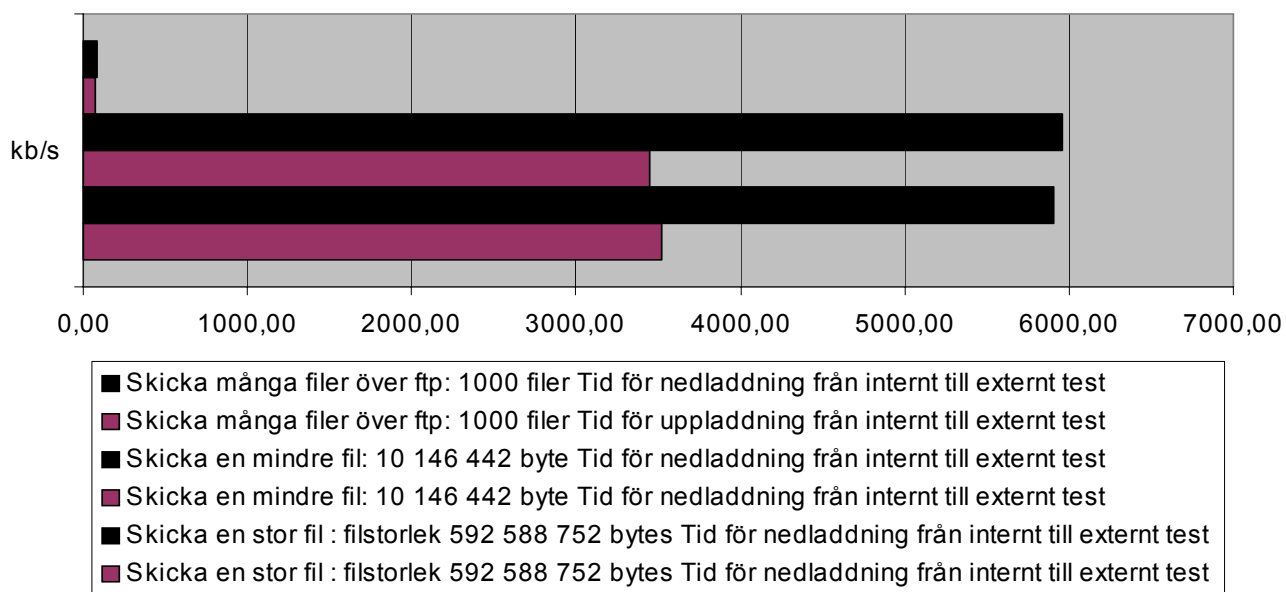


Diagram D.8: Jämförelse mellan router och switch, nedladdning

### Jämförelse router/100 Mbit Switch, 600 Mb fil uppladdning

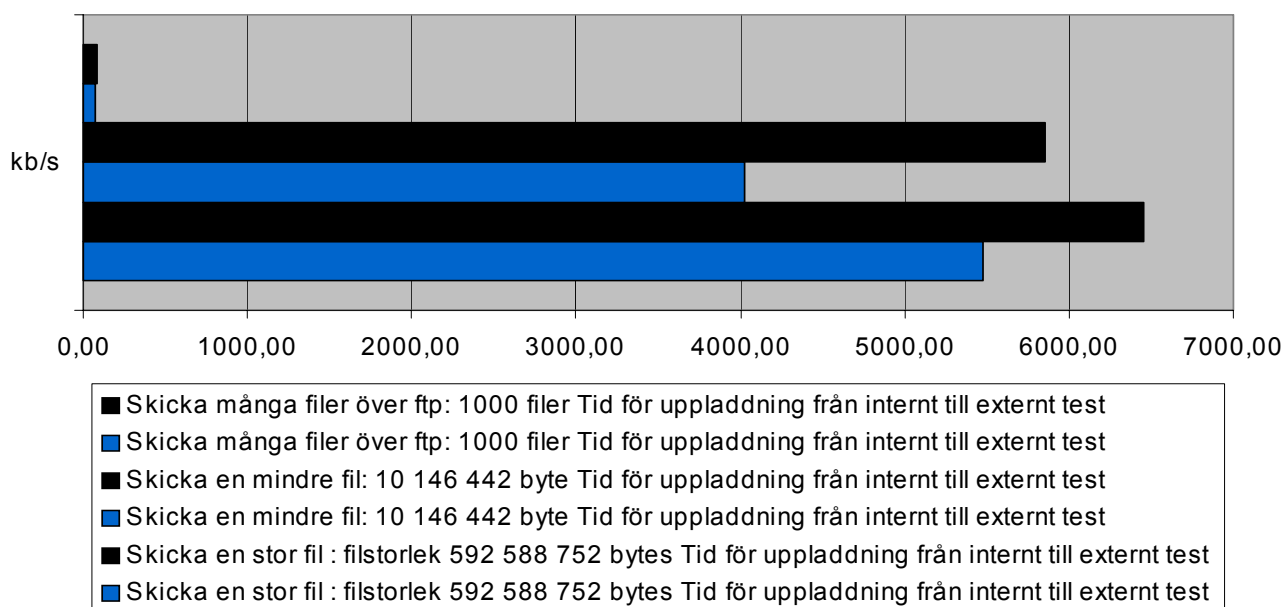


Diagram D.9: Jämförelse mellan router och switch, uppladdning