Examensarbete

HÖGSKOLAN I SKÖVDE

ROUTERMJUKVAROR BASERADE PÅ ÖPPEN KÄLLKOD

Jämförelsestudie mellan open source routrar

Examensarbete inom huvudområdet Datalogi
Grundnivå 15 högskolepoäng
Vårtermin 2013

Pär Fahlesson

Handledare: Mikael Lebram
Examinator: Jonas Mellin

Sammanfattning

En router kan i dagsläget bestå av vanlig X86/PC-hårdvara vilket medför att hårdvaran inte är lika låst som den hos vanliga hårdvaruroutrar. Vidare kan dessa X86/PC-routrar köras med öppna programvaror. Många gånger är även dessa programvaror gratis och det gör att dessa mjukvaruroutrar är billiga att införskaffa. Detta arbete har studerat ett urval av tre mjukvaruroutrar för att se ifall det fanns några skillnader mellan dem. Det som undersöktes var prestanda, funktionalitet och support. Detta har gjorts genom att utföra en litteraturanalys och ett experiment, där experimentet testade prestandan. Noterbara skillnader som hittades var att de stödde olika funktioner och att supporten som dessa utvecklare försedde var olika.

Nyckelord: Open source, mjukvaruroutrar, jämförelsestudie

Innehållsförteckning

1 Introduktion

Routrar är centrala enheter i varje nätverk och de utför ett flertal olika funktioner. En router kan definieras som en enhet som sammankopplar två eller fler nätverk, vanligtvis mellan det lokala nätverket (LAN) och Internet (WAN). Deras primära funktion är att dirigera trafik mellan dessa olika nätverk. Trafiken består av paket som i sin tur består av två beståndsdelar, adresser och data. Dirigering görs via rutter som är de vägar som ett paket behöver ta för att nå sin destination. Routrarna stödjer olika metoder för att de ska hantera rutter samt hur de ska lära sig av andra routrar för vilka nätverk som finns i närheten. En metod är statiska rutter som konfigureras manuellt av administratören. En annan metod för att hantera rutter är att använda dynamiska protokoll. Dessa protokoll hanterar och propagerar rutter automatiskt.

En hårdvarurouter består av en specialutvecklad hård- och mjukvara som är optimerad för detta ändamål, vilket gör att de kan uppnå höga överföringskapaciteter (Cisco, 2010). Dessa enheter är i de flesta fallen proprietära och stängda vilket gör att de är i många fall svåra att modifiera. Dock har vissa utvecklare börjat tillhandahålla verktyg som ska göra det möjligt att förändra begränsade delar av mjukvaran (Juniper, 2011). Vissa routrar ger möjligheten till att ändra vissa delar av hårdvaran, så som portdensiteten. Det vill säga mängden nätverksportar som finns. Dock om någon större förändring ska ske så måste routern i många fall bytas ut helt. Det kan vara en dyr affär, speciellt i de fall där fler än en router skall bytas ut.

Ett alternativ till hårdvaruroutrar är mjukvaruroutrar byggda på öppen källkod, som oftast är gratis, tillsammans med standardiserad X86/PC-hårdvara. Dessa mjukvaruroutrar kan modifieras fritt efter behov. Det leder till att om mjukvaruroutern behöver nya funktioner eller tjänster så går det att lägga till det. Vidare om hårdvaran behöver förändras så kan det ske med enkla medel då hårdvaran är standardiserad och alla delar kan bytas ut.

Problem som kan uppstå är att det finns många mjukvaror att välja mellan och det kan vara en svår uppgift att välja vilken mjukvara som passar bäst för situationen. Denna studie presenterar och jämför ett urval av mjukvaror med avseende på funktionalitet, support och prestanda. Studien är uppdelad i två delar. En litteraturanalys som har undersökt vilka funktioner dessa mjukvaror stödjer samt vilken typ av support de erbjuder. Den andra delen består av ett experiment som har jämfört vilka prestandaskillnader som förekommer med avseende på överföringskapacitet och förlorade paket.

Litteraturanalysen har analyserat de litterära verk och den systemdokumentation som finns. Experimenten är gjort genom att skicka en stor mängd trafik mellan två nätverk. Det har testats med olika paketstorkar för att se hur väl de presterar under olika fårhållanden.

2 Bakgrund

Detta kapitel beskriver grundläggande nätverks- och routerbegrepp samt belyser vanligt förekommande routerfunktioner. Kapitlet kommer även att beskriva och ge insikt om öppen källkods- och proprietärmjukvara.

2.1 Nätverksbegrepp

I denna sektion tas relevanta nätverksbegrepp upp.

Ett nätverk i denna kontext är en sammankoppling av två eller fler enheter som kommunicerar med varandra. Det finns många typer av nätverk men de två som är relevanta och vanligt förekommande är:

- LAN (Local Area Network) som är den benämningen på det lokala nätverket. Det som kännetecknar ett LAN är att det har en kort utsträckning och kontrolleras oftast av en person eller organisation. Exempelvis ett hemmanätverk.
- WAN (Wide Area Network) är ett nätverk som sträcker sig över ett stort område. Ett exempel på detta är Internet.

Kommunikation mellan de enheter, det vill säga de klienter, servrar och annan utrustning som finns på nätverket möjliggörs tack vare att enheterna använder protokoll som beskriver hur denna kommunikation ska gå till. TCP/IP är en samling av nätverksprotokoll som möjliggör att enheterna förstår varandra och protokollen beskriver också hur enheterna skall adresseras. En enhet har två adresser, en IP-adress och en MAC-adress. IP-adressen används för att dirigera trafik mellan olika nätverk medan MAC-adressen används vid dirigering av trafik inom det lokala nätverkssegmentet.

Vid dirigering inom det lokala segmentet används oftast en switch som är en enhet som dirigerar trafik med hjälp av MAC-addresserna. För att dirigera trafik mellan olika nätverken så behövs det en router.

Data delas upp i mindre delar när den skickas över ett nätverk och dessa delar kallas för paket. Ett paket består av två delar, en header och en payload. Headern innehåller bland annat mottagaren och avsändarens IP-adress medan payloaden är den data som skickas.

2.2 Router

En router är en enhet som sammankopplar två eller fler nätverk och dirigerar datatrafik mellan de sammankopplade nätverken. En vanlig sammankoppling är mellan ett LAN och ett WAN. Det vill säga mellan det lokala nätverket och Internet.

Dirigeringen sker genom att routern analyserar inkommande datapaket och dess header. Denna information använder routern för att skicka vidare paketet till sin destination. För att veta vilka nätverk som finns har routern en tabell med dynamiska och statiska rutter samt direktanslutna nätverk, även kallat routingtabell. Routern letar igenom denna tabell för att se om det aktuella paketet skall till något av de kända nätverken. Om paketets mottagaradress finns med i tabellen, så dirigeras det paketet till det gränssnitt som har det nätverket som sin rutt (Parziale m.fl, 2006). Ett gränssnitt i denna kontext är en nätverksport. Om ingen rutt hittas i tabellen, dirigeras paketet till en standardrutt som kallas för standardgateway.

2.2.1 Statiska rutter

Statiska rutter konfigureras av en administratör och det utförs ingen kommunikation mellan routrarna om den nuvarande topologin. Det betyder alltså att router A delges ingen information om router B:s statiska rutt till router C. Det betyder att routrarna inte kan dynamiskt ändra sina rutter. Om router A hade haft en statisk inlagd rutt till ett nätverk på router C och det nätverket hade försvunnit så hade router A fortsatt att dirigera berörd trafik dit tills det att nätverket hade blivit aktivt igen eller manuellt tagits bort (Parziale m.fl, 2006). Statiska rutter ger dock administratören mer kontroll över nätverket samt att det ger en ökad säkerhet då alla rutter är förbestämda och manuellt inlagda av en administratör (Parziale m.fl, 2006).

2.2.2 Dynamiska rutter

Dynamiska rutter är de rutter som propagerats av dynamiska routingprotokoll. Routingtabellen uppdaterats automatiskt allt eftersom nätverket förändras. Det menas att om ett nytt nätverk kopplas in på router A och det har konfigurerats att propageras till de andra routrar så kommer router B att automatiskt uppdatera sin tabell med det nätverket och trafiken kan dirigeras dit (Parziale m.fl, 2006). Om två routrar använder sig av två olika routerprotokoll för samma länk så kommer de inte kunna kommunicera med varandra. Olika routingprotokoll hanterar propageringen och hanteringen av rutterna på olika vis. Protokollen kan klassificeras i fyra huvudkategorier:

- Distance vector-protokoll
- Link state-protokoll
- Path vector-protokoll
- Hybrid-protokoll

Kortfattat så bygger Distance vector-protokoll på att veta hur många routrar ett paket måste passera för nå sin destination. Link state-protokoll baserar sina beslut på hur bra länken är mellan de olika routrarna är. Path vector-protokoll försöker få en helhetsbild av hela rutten till destinationen för att motverka komplikationer. Hybrid-protokoll är som namnet låter, hybrider mellan de protokoll som finns.

2.2.3 Routerfunktioner

En router kan utföra mer än bara dirigering av trafik. De kan i allmänhet tillföra exempelvis dessa tjänster och funktioner:

- DHCP (Dynamic Host Configuration Protocol),
- NAT (Network Address Translation)
- Paketfiltrering
- VPN (Virtual Private Network)
- QoS (Quality of Service)

DHCP är en tjänst som tilldelar enheter på nätverket IP-adresser och annan information så att de inte behöver ange dessa manuellt.

NAT är funktion som översätter en IP-adress till en annan. Exempelvis privata adresser som inte kan vistas på Internet till publika IP-adresser som kan det. En annan NAT-typ är port forwarding som möjliggör att inkommande anslutningar på en specifik port till routern kan skickas vidare till en intern resurs.

Paketfiltrering är en typ av funktion som filtrerar trafik beroende på vilka regler som konfigurerats av administratören. Det är en säkerhetsmekanism som kan blockera eller tillåta de paket som skickas igenom, till eller från routern. De regler som konfigureras kan bland annat blockera eller tillåta trafik beroende på vilken port som används, vilken tid på dygnet som paketet inkommer till routern eller vilken IP-adress som finns i headern.

VPN används för att sätta upp säkra tunnlar mellan två enheter som finns på exempelvis två olika nätverk. Det gör att exempelvis en som arbetar hemifrån kan ansluta till företagets resurser på ett säkert vis. Det finns två typer av VPN, fjärråtkomst-VPN och plats-till-plats-VPN. Den första metoden liknar det exemplet som beskrivs tidigare. Plats-till-plats-VPN gör att ett företag kan ha ett gemensamt nätverk för flera kontor även om kontoren är geografiskt utspridda på olika platser i landet. En användare märker ingen skillnad om de utnyttjar en resurs på kontor A eller kontor B.

QoS är en teknik för att forma trafiken så att den inte överstiger en viss gräns eller för att tillförse viss trafik med en högre prioritet. Det vill säga att en viss trafik får företräde när den lämnar nätverket. QoS kan konfigureras på olika vis och med olika strategier för hur prioriteringen ska gå till.

Dessa är några av de grundläggande funktionerna som en router kan utföra och det är därför som dessa funktioner och tjänster är medtagna i denna studie. Dessa funktioner finns både på kommersiella hårdvaruroutrar och på öppna mjukvaruroutrar (Juniper,2013; Buechler & Pingle, 2009). Studien har undersökt till vilken grad dessa funktioner har implementerats samt om funktionerna skiljer sig mellan routrarna.

2.3 Hårdvarurouter

Hårdvaruroutrar är routerenheter med dedikerad och specialutvecklad hård- och mjukvara som även är i regel stängda. De stödjer förbestämda funktioner, bland annat de som beskrivits i sektion 2.2.3 men också andra funktioner och kräver, om nya funktioner skall läggas till, att hårdvaran uppgraderas eller byts ut helt (Ye & MacGregor, 2008). Enligt Bianco, Finochietto, Galante, Mellia och Neri (2005), lider proprietära hårdvaruroutrar av kompabilitetsproblem, har mindre programmeringsmöjligheter och är inte så flexibla. Med kompabilitetsproblem så menas det att två eller fler enheter av olika fabrikat kan ha svårt att fungera tillsammans och om de är modulära så måste hårdvaran vara kompatibel med routern. Att en router är modulär betyder att viss hårdvara går att byta ut, så som nätverkskort. Dock måste modulen och routern vara utav samma fabrikat då exempelvis en Cisco-modul inte fungerar eller passar i en HP-router. Hårdvaruroutrarnas programmeringsmöjligheter är små då de bygger på proprietär källkod som inte är öppen för allmänheten. Dock har vissa tillverkare börjat att skapa APIs som gör det möjligt för tredjeparts utvecklare att ändra begränsade delar av routermjukvaran (Juniper, 2011). Bortsett från dessa nackdelar så kan dessa routrar nå en väldigt hög prestanda. Detta eftersom deras hårdvara är specialutvecklad och optimerad för att utföra routerfunktioner. Samt att mjukvaran som används är specialskriven och optimerad för den hårdvara som den jobbar med. Ett exempel på detta är en av Ciscos senaste "Carrier grade" router som har enligt Cisco (2010) en överföringskapacitet på 322 Terabits per sekund.

2.4 Mjukvarurouter

Ett alternativ till hårdvaruroutrar är routerenheter byggda på standardiserad X86/PC-hårdvara med mjukvaror som tillhandahåller routerfunktioner. Exempelvis så kan en vanlig PC göras om till en router. Det som krävs är att det finns två nätverkskort och en routermjukvara. Dessa enheter överkommer många av de nackdelar som de proprietära hårdvaruroutrarna lider av, så som att de kan modifieras fritt efter behov. Ny hårdvara för att öka prestandan eller förändra portdensiteten (mängden nätverksportar) kan enkelt läggas till eller bytas ut samt att enheterna kan utökas med andra tjänster så att de blir multifunktionella. En webbserver kan exempelvis hysas på samma maskin som routern. Detta är en fördel för att organisationer kan enkelt justera hur deras router ska se ut och prestera med billiga medel.

2.4.1 Licenser

En fördel är att de flesta mjukvarorna är öppna och gratis eller har en minimal kostnad i relation till de proprietära hårdvaruroutrarna. Två välanvända öppna källkodslicenser är Berkeley Software Distribution-licensen (BSD) och General Public License (GPL) (Open Source Initiative, 2013).

Dessa licenser gör det möjligt att förändra mjukvaruroutern efter behov samt att öppen källkod ger kontroll över mjukvaran som inte vore möjlig om det var en stängd proprietär mjukvara. Om nya funktioner skall läggas till är det fullt möjligt att göra det. Något som kan vara svårt på en proprietär router.

2.4.2 Utbud

På marknaden finns det ett flertal routermjukvaror som körs på olika operativsystem och som implementerar routerfunktionerna på olika vis. Enligt Wikipedia (2013) som har en sammanställd lista, finns det 33 stycken routerimplementationer. Dock så innehåller denna lista även routermjukvaror som är avsedda för inbäddade system (DD-WRT) och mjukvaror som är till för att användas vid forskning (AGH Live Routing). En annan källa som listar Linux- och BSD-distributioner är Distrowatch. Distrowatch är en hemsida dedikerad till att sammanställa och ge information om Linux- och BSD-distributioner. Vid sökning på ordet "router" på den hemsidan så listar den 16 stycken distributioner (Distrowatch, 2013). Varav nio av dem kan ses som routerdistributioner, resten är brandväggsdistributioner. Dock överlappar de på några av distributionerna.

Några av de populäraste distributionerna ifrån dessa två listor är Pfsense, ClearOS och Vyatta. För att avgöra vilka som är populära har Googles söktjänst använts. De med flest sökresultat har ansetts vara populärast. För en komplett bild över rangordningen se Appendix A - Popularitet.

2.4.3 Prestanda

Historiskt sett har prestandan hos mjukvaruroutrarna inte kunnat konkurrera med den hos de proprietära routrarna och de har inte kunnat nå en högre prestanda än 1-3 Gbps (Argyraki, Baset, Chun, Fall, Iannaccone, Knies, Kohler, Manesh, Nedevschi & Ratnasamy, 2008). Men på senare tid gjorts ett flertal forskningsarbeten angående detta och de har med diverse metoder kunnat nå en prestanda på 10 respektive 100 Gbps. (Gördén, Hagsand, Olsson, 2008; Han, Jang, Park, Moon, 2010).

Det är av vikt för beslutfattare att veta hur mjukvaruroutrar presterar då det är en viktig faktor som kan spela en roll i vilken mjukvarurouter som väljs. Då det kan skilja på prestandan har en del av de prestandamått som finns undersökts i denna studie.

2.5 Support

Support är en viktig aspekt då det är oundvikligt att inte stöta på något problem förr eller senare. Graden av hjälp som behövs kan variera men det är ändå viktigt att det finns någon form av hjälp då om problem uppstår så måste de oftast lösas fort. Support kan fås från flera medium och från olika håll. Denna studie har undersökt vilken support tillverkare av mjukvaruroutrar som är baserade på öppen källkod kan ge.

3 Problembeskrivning

Denna del tar upp varför denna studie är viktigt samt vilket syfte den har.

3.1 Syfte

Syftet med denna studie är att undersöka och jämföra ett urval av tre mjukvaruroutrar baserat på popularitet (Appendix A) . Studier har för avsikt att utröna vilka skillnader och likheter som existerar mellan mjukvaruroutrarna med avseende på prestanda, funktionalitet och support.

Denna studie ämnar sig åt att hjälpa beslutfattare att välja rätt mjukvarurouter då det finns många att välja mellan samt att det kan finnas markanta skillnader som avgör vilken som skall väljas. Det är även av vikt för beslutfattare att göra rätt val när de väljer sina mjukvaror då det kan leda till problem om de väljer fel. Exempel på problem som kan uppstå är:

- Mjukvaran blir inaktiv och inga nya uppdateringar görs.
- Funktioner som behövts finns inte eller att funktionerna som stöds har inte de fulla kapaciteter som hade behövts. Det kan handla om att exempelvis vissa dynamiska routingprotokoll inte stöds eller att QoS-funktionen inte stödjer de strategierna som hade behövts.
- Begränsad eller bristfällig support.
- Bristfällig kod som orsakar problem med drift och säkerhet.

Studien riktar sig mot användare som behöver kontroll över sin nätverksmiljö och som inte vill vara låst till en specifik hårdvara eller mjukvara.

3.2 Motivering

Motiveringen till denna studie ligger i att hårdvaruroutrar är svåra att förändra då mjukvaran är stängd och hårdvaran kan vara låst. Om nya funktioner eller ny hårdvara måste införskaffas så kan det innebära att enheten i sig får bytas ut och det kan vara en dyr affär.

Alternativet är att använda mjukvaruroutrar. Det som gör dem intressanta är att de kan installeras på, näst intill, vilken X86/PC-maskin som helst. Det går att omvandla en klient- eller servermaskin till en fullt fungerade router. Detta gör att de är billiga att införskaffa samt att de kan uppgraderas och förändras enkelt.

3.2.1 Delmål

Studien är uppdelad i tre delmål för att besvara frågan om vilka skillnader som existerar mellan urvalet av mjukvaruroutrar.

1. Det första delmålet har haft för syfte att besvara frågan om huruvida vissa centrala funktioner och tjänster stöds eller ej. Samt till vilken grad de stöds då funktionerna eller tjänsterna kan bete sig på olika sätt.
   De funktioner och tjänster som tagits med i denna lista
   De funktionerna och tjänsterna som har studerats är DHCP, NAT, Paketfiltrering, VPN, QoS och vilka dynamiska routingprotokoll som finns tillgängliga. Dessa funktioner och tjänster har tagits med då det är viktigt att veta hur och om:

- DHCP-tjänsten finns integrerad på routern då detta underlättar för organisationen då de inte behöver använda andra resurser för denna tjänst. Samt att de då kan tilldela sina enheter med IP-adresser och annan information på ett dynamiskt vis.
- NAT finns tillgängligt samt hur den går att konfigurera då det kan vara av vikt att veta om det stöds att konfigurera NAT:en på flera olika vis. Exempelvis om det stöds att ha både statiska och dynamiska översättningar.
- Det går att införa regler för paketfiltrering då detta är en säkerhetsmekanism som kan skydda nätverket mot vissa interna och externa hot.
- VPN stöds och till vilken grad. Detta för att VPN kan konfigureras med olika protokoll samt att det kan finnas begränsningar på hur många aktiva tunnlar som får vara igång samtidigt.
- QoS stöds och vilka strategier som går att införa. Det är av vikt att veta då viss trafik kan behövas att prioriteras.
- Dynamiska routingprotokoll stöds då dessa kan underlätta vid större nätverk då mindre manuell administration behövs.

Samt att de berör aktiviteter som:

- Underlättar administrationen och på ett dynamiskt vis kan utföra givna uppgifter. Exempelvis tilldela klienter med IP-adresser eller uppdatera routingtabellen.
- Säkrar upp nätverket.
- Gör det möjligt för administratören att forma trafiken efter behov.

2. Det andra delmålet har besvarat frågan om vilken sorts support som kunnats få ifrån utvecklarna. Det vill säga vilken typ av hjälp kunnats få samt från vilka medium. Detta för att det är en viktigt aspekt som bör övervägas när en ny mjukvara skall införskaffas. Detta och det ovannämna delmålet har undersökts med en litteraturstudie.

3. Det tredje delmålet har undersökt hur urvalet av mjukvaruroutrar presterar i relation till varandra. Prestandan som har mätts är överföringskapaciteten mellan två nätverk och antal tappade paket. Detta mål är viktigt att veta för att prestanda i allmänhet är en faktor som kan spela en roll när en mjukvarurouter skall väljas. För att undersöka detta mål har ett experiment utförts.

3.3 Avgränsningar

Nedan är de krav som bestämt vilka mjukvaror som har studerats:

- Mjukvaran måste ha öppen källkod och gå under en erkänd licens enligt OSI.
- Mjukvaran ska gå att installera på en X86/PC-baserad maskin.
- Mjukvaran måste vara aktiv och levande. Det vill säga att mjukvaran måste hållas uppdaterad.

En ytterligare avgränsning har gjorts för att begränsa urvalet till tre mjukvaror. Avgränsningen har gjorts genom att ta med de tre populäraste mjukvarorna och det enligt hur många sökresultat de presenterar vid en sökning på Googles sökmotor. Listan med resultat kan ses i appendix A - Popularitet. De tre mjukvaror som har behandlas i studien är:

- Pfsense
- ClearOS
- Vyatta

4 Metod

Detta kapitel kommer att presentera de metoder som har använts för att undersöka problemen.

4.1 Litteraturanalys

För att undersöka och ge svar på de två första delmålen samt för att hitta information om hela arbetet har en litteraturanalys genomförts. Enligt Berndtsson, Hansson, Olsson och Lundell (2002) så är det av vikt att a välja rätt källor, så som publicerad artiklar. Detta har dock varit problematiskt då information kring funktionalitet och support har främst hittats i de systemdokumentationer som tillverkarna själva har utgivit. Då dessa tar upp hur deras mjukvaror kan konfigureras och tillämpas. Vidare så diskuteras validiteten av de resultat som framtagits senare i denna studie.

För att undersöka vilken typ av support som kan erbjudas om dessa mjukvaruroutrar så har enbart tillverkarnas egna källor studerats. Med andra ord har det enbart undersökts till vilken grad som tillverkarna kan ge support då det skulle bli omöjligt att ta reda på exakt vilka externa parter som kan ge support. Detta för att det kan finnas väldigt många externa företag, organisationer eller enskilda personer som bedriver någon form av support för dessa routermjukvaror.

För att minimera chansen att något har missats så har ett systematiskt tillvägagångssätt använts. Där källor har först hittats och sedan undersökt i en iterativ process. Om nya, relevanta, begrepp framkommit så har dessa använts för att söka efter ny information. Detta för att inte missa någon vital information men också för att försöka validera det som skrivits.

4.1.1 Alternativa metoder

Att undersöka vilka funktionella skillnader som existerar kan göras på mer än ett sätt. Det är fullt möjligt att undersöka skillnaderna rent praktiskt med ett experiment. Denna metod skulle även kunna validera huruvida om det som sägs i systemdokumentationen stämmer eller ej. Dock kan detta kräva mer resurser i form av tid då vissa funktioner och tjänster kan konfigureras på många olika sätt. Denna metod valdes inte just för denna orsak.

4.1.2 Motivering

Motiveringen är att denna metod lämpar sig bäst för besvara dessa delmål då andra metoder inte kan ge samma resultat eller så lämpar de sig inte till att besvara dessa två delmål. Ett exempel skulle vara att utföra en intervju med tillverkarna. Resultatet av denna intervju skulle kunna vara missvisande om exempelvis frågorna var felformulerade eller om den som intervjuas inte har all information.

4.2 Experiment

Ett experiment har genomförts för att besvara det tredje delmålet. Skiljer sig prestandan mellan urvalet av routermjukvarorna.

För att prestandatesta mjukvaruroutrarna har trafik genererats och skickats från båda sidor av mjukvaruroutern, för att sedan tas emot av andra klienter på vardera sida. Det vill säga att klient A på nätverk X har genererat trafik och skickat denne till klient B på nätverk Y. Samtidigt som klient C har genererat och skickat trafik från nätverk Y till klient D på nätverk

X. För en översiktlig bild av topologin se appendix B - Topologi. Detta ska testa hur väl mjukvaruroutrarna dirigerar trafik och värdena som mäts med denna metod är hur många paket per sekund (pps) som kan dirigeras samt hur många bits per sekund (bps) som skickas.

Olika storlekar på paket har testats för kunna ge en god analys av prestandan, då paketstorleken kan påverka prestandan. Storlekarna som har testats är 64 bytes, 500 bytes, 1000 bytes och 1500 bytes. Värdena valdes för att få en god spridning men också för att MTU (Maximum Transmission Unit) är vanligtvis satt till 1500 för ethernet anslutningar (IETF. 1984). Det betyder att paket över 1500 bytes fragmenteras (splittras upp i mindre delar).

Trafiken har skapats med hjälp utav Pktgen som är ett verktyg som kan generera stora mängder trafik. De valdes för att det kan enligt skaparen generera en stor mängd trafik med små medel (Olsson, 2005). Verktyget valdes framför andra verktyg för att det är integrerat med Linux-kärnan och därmed har en mindre overhead. Det tar mindre tid och använder inte lika mycket resurser som vissa andra verktyg gör för att generera trafik.

För att säkerhetsställa att all data är konsekvent har experimentet genomföras i en laborationsmiljö där varje klient har haft samma hårdvara samt att routrarna har installerats på samma maskin för varje test. Om det funnits funktioner på mjukvarorna som inte har med routing att göra, så som paketfiltrering och QoS så har de avaktiveras. Detta för att minska antalet variabler som kan ha en påverkan på resultatet. Alla mjukvaruroutrar har samma förutsättningar. Enligt Berndtsson m.fl (2002) så är det viktigt utföra mer än ett test då resultatet kan ha påverkats av någon yttre parameter. Det går inte att dra någon slutsats av ett test utan fler bör göras. I denna studie har det utförts minst tre tester per omgång. Det vill säga att minst tre tester har utförts per ändrad paketstorlek. Om alla tre tester visat liknande resultat så har det påvisat ett korrekt resultat enligt denna studie. Om värdena varit vitt skilda har ytterligare tester genomförts tills det att majoriteten av värdena har visat liknande resultat. Tre av de testerna har sedan valts ut. De utvalda värdena har sedan använts för att få fram medelvärden och detta för att få ut genomsnittliga värden. Detta räknades ut genom att addera resultaten och dividera dem med 3.

All data som denna metod har samlat in har sedan analyserats med Capinfos för att dra slutsatser om huruvida det finns några skillnader rent prestandamässigt. Alla resultat presenteras senare i denna studie.

4.2.1 Motivering

Motiveringen till varför ett experiment genomförs är för att andra metoder kan i vissa fall inte framställa lika trovärdiga resultat.

Simuleringar måste vara trovärdiga i den bemärkelsen att de måste vara modellerade efter den riktiga världen. Detta kan vara svårt att göra samt att just för att man utför processer inom en simulerad "värld" så kan resultaten man får ut inte vara lika trovärdiga. Resultaten kan vara felaktiga då man exempelvis inte tagit hänsyn till vissa variabler. Det kan även vara så att variablerna inte matchar den riktiga världen.

En litteraturanalys lämpar sig inte att utföra då de litteraturverk som finns kan i vissa fall inte ge den trovärdighet som ett experiment skulle kunnat ge. Dels för att informationen kan komma från tillverkarna själva. Men också för att informationen kan vara bristfällig i hur resultaten har framtagits. Om ingen förklaring ges för hur resultaten framtagits så kan den inte heller verifieras. Vid de fall att flera källor används så kan det vara svårt att sätta värdena

i relation till varandra då de kan ha utfört testerna på olika vis, med olika hårdvara och med olika konfigurationer samt med olika versioner av mjukvaran.

4.3 Validitet

Enligt Berndtsson m.fl (2002) är det av vikt att beakta validitet och att vara medveten om vilka hot det kan finnas mot validiteten. Gällande experimentet beaktas validiteten genom att repetera testerna och på det viset få ut trovärdiga resultat. Vidare konfigureras varje mjukvara på liknande sett för att minimera antalet variabler som kan påverka resultatet. Litteraturanalysen valideras genom att på ett systematiskt vis studera varje källa.

5 Genomförande

Detta kaptitel kommer ta upp hur studien har genomförts. Kapitlet är uppdelat i två delar, litteraturanalysen och experimentet.

5.1 Litteraturanalys

Litteraturanalysen har genomförts genom att studera de litterära verk som avser ämnet. Som beskrivits tidigare har främst tillverkarnas systemdokumentation använts för att besvara delmålen rörande funktionalitet och support då dessa källor innehar den information som krävs för att besvara målen. Dock har andra källor, vid den mån det funnits också analyserats för att besvara frågorna.

För att hitta information har sökmotorerna Google och Google Scholar[1] använts samt databaserna IEEE Xplore[2] ACM Digital Library[3].

Den information som framtagits under denna litteraturanalys har stått som grund till den analys utifrån det resultat som framkommit.

5.2 Experiment

Denna sektion avser att beskriva hur experimentet genomfördes. För en överskådlig vy över nätverkstopologin se Appendix B.

5.2.1 Installation

Mjukvaruroutrarna installerades på följande hårdvara genom att följa installationsguiden:

- Moderkort: HP 0A58h
- CPU: Intel Core 2 CPU E6400 @ 2.13GHz
- Minne: 2GB DDR2 533MHz
- 2x Nätverkskort: Intel Corporation 82541PI Gigabit Ethernet Controller.

De versioner som installerats är:

- Pfsense 2.0.3
- ClearOS 6.4.0
- Vyatta 6.5

Klienterna som skickade trafik installerades ej på någon hårdvara utan liveCDs användes för detta ändamål. En liveCD är en CD- eller DVD-skiva med ett komplett operativsystem som kan användas utan att behöva installera det. De klienter, en på varje nätverk, som tog emot alla paket installerades dock på hårddisk. Detta för att paketen skulle sparas till fil för vidare analys och för att liveCDs förlorar sin data vid omstart. Alla klienter hade ett Linux-operativsystem.

I resten av studien kommer klienterna att hänvisas till namnen LK1 och LK2 för liveCD klienterna och DK1 och DK2 för klienterna som tog emot paketen.

[1] http://www.google.se/ & http://scholar.google.se/
[2] http://ieeexplore.ieee.org/Xplore/home.jsp
[3] http://dl.acm.org/

5.2.2 Konfiguration - Pfsense

Två gränssnitt sattes upp med varsina IP-adress från två olika nätverk. 192.168.1.1/24 för gränssnitt em1 och 192.168.2.1/24 för em2. Detta gjordes lokalt på routern genom att följa de anvisningar som finns. Efter denna konfiguration kunde mjukvaruroutern konfigureras från dess webbgränssnitt. Åtkomst till webbgränssnittet fås genom att skriva in mjukvarurouterns LAN IP-adress i webbläsarens adressfält.

Genom webbgränssnittet stängdes brandväggen av. Detta gjordes under menyn System -> Advanced -> Firewall / NAT. Detta för att minimera antalet variabler som kan påverka experimentet. Inga ytterligare konfigurationer gjordes.

5.2.3 Konfiguration - ClearOS

ClearOS konfigurerades också först lokalt med två LAN, samma IP-adresser som ovan. Sedan kunde dess webbgränssnitt också användas.

Innan routern kunde konfigureras var en installations-/konfigurationsguide tvungen att genomföras. Denne krävde en Internetanslutning vilket medförde att det ena LAN:et kopplades bort och ersattes med en Internetanslutning.

Följande moment genomfördes i guiden:

Ett läge fick väljas, Gateway, privat server eller publik server. Här valdes privat server. Detta val är för de ClearOSmaskiner som inte ska vara uppkopplade mot Internet. Ingen brandvägg aktiveras. Nästa steg bad om information om gränssnittet. Då dessa skulle ändras i ett senare skede valdes det att inte göra något i detta steg. Det sista nätverksmomentet bad om vilken DNS-server som skulle användas. Då ingen DNS skulle behövas i detta experiment angavs 127.0.0.1.

Nästa moment i denna konfiguration bestod av att välja vilken version av ClearOS som skulle användas. Då det finns två versioner av ClearOS, en öppen och en betalversion. Den öppna valdes då denna studie handlar mjukvaruroutrar baserade på öppen källkod. Momentet efter det letade upp och installerade uppdateringar. Efter detta behövdes mjukvaran och maskinen registreras tillsammans med ett användarkonto.

Nästa steg bad om ett domännämn och hostnamn. Exjobb.local valdes som domän med router som hostnamn. Efter detta konfigurerades klockan.

De sista steget handlade om vilka applikationer som skulle laddas ner och installeras via ClearOS applikationscentral. Applikationer i ClearOS tillhandahåller tjänster och funktioner som kan integreras med ClearOS. Exempelvis VPN-applikationer som tillhandahåller VPN-funktioner. Det valdes att inte ladda ner eller installera några applikationer.

Under menyn Network -> IP-settings togs Internetanslutningen bort och ersattes med en LAN-anslutning.

5.2.4 Konfiguration - Vyatta

Vyatta konfigurerades på ett annat vis än vad de ovan gjordes då Vyattas öppna version inte tillhandahåller något webbgränssnitt. All konfiguration fick göras lokalt på maskinen via dess CLI (Command Line Interface). Följande konfigurationskommandon utfördes:

```
Configure
set interface ethernet eth1 address 192.168.1.1/24
```

```
set interface ethernet eth2 address 192.168.2.1/24
commit
```

5.2.5 Konfiguration - Klienter

LK1 och LK2 konfigurerade med dessa statiska IP-adresser, 192.168.1.10 och 192.168.2.10. DK1 och DK2 konfigurerades med IP-adresserna 192.168.1.15 och 192.168.2.15. Vidare installerades Tcpdump och Capinfos på DK1 och DK2. Tcpdump är ett verktyg för att samla in paket och Capinfos användes för att analysera de dumpfiler som Tcpdump skapade. Wireshark som också är ett insamlings- och analysverktyg skulle ha använts men det fungerade inte då filerna som skapades blev för stora för Wireshark att hantera. Samt att Capinfos kundes integreras med ett bash-script för automatisering av analys av dumpfiler, något som inte hade fungerat med Wireshark. På LK1 och LK2 hämtades det hem ett bash-script användes för att skicka filer tillsammans med Pktgen. Båda scripten kan ses i Appendix C och D.

5.2.6 Utförande

Experimentet utfördes genom att skicka trafik från båda sidor av mjukvaruroutern. Trafiken togs sedan emot av en klient på vardera sida för fortsatt analys.

Testerna genomfördes genom att först starta Tcpdump på DK1 och DK2 med följande kommando:

```
Tcpdump -nn -i eth1 udp port 9 -B 9000000 -w dumpfil.cap
```

Växeln "-nn" medför att inga namnuppslagningar görs. IP-adresser och portar översätts inte till några namn. Växeln användes för att minimera arbetsbördan samt att ingen DNS var tillgänglig. Växeln "-i eth1" talar om för tcpdump vilket gränssnitt som ska avlyssnas. Växlarna "udp" och "port 9" användes för att enbart samla in UDP paket som inkommer på port 9. Dessa värden valdes för att Pktgen skickar UDP paket till port 9. "-B" ökar bufferten som Tcpdump har till 9000000 bytes. Tcpdumps standardbuffert var för liten för de tester som utfördes. Tcpdump kastar de paket som inte får plats i bufferten och för att dessa tester ska prestandatesta hur en mjukvarurouter presterar är det av vikt att alla paket sparas. Sparning till fil görs med "-w dumpfil.cap" växeln.

Pktgen startas med hjälp av ett Bash-script och detta kan ses i Appendix C. Scriptet har inte skapats själv utan det kommer från Pktgen skaparen[4]. De ändringar som har gjorts i scriptet för att passa experimentet är:

- Gränssnittsnamnen har bytts ut till de som finns på LD1 och LD2.
- PKT_SIZE har ändrats för varje test (64, 500, 1000 och 1500).
- Count är satt till 15 000 000 (Antal paket som skickas).
- pgset dst är satt till DK1s eller DK2s IP-adress.
- pgset dst_mac är satt till mjukvarurouterns MAC-adress

Efter genomförd testning har den insamlade data analyserat med Capinfos tillsammans med ett Bash-script. Detta Script kan ses i Appendix D. Den data som fås ut efter att Capinfos har analyserat de dumpfiler som Tcpdump skapar är:

[4] För mer information besök:
http://www.linuxfoundation.org/collaborate/workgroups/networking/pktgen

- Antal insamlade paket.
- Den totala mängden data som mottagits.
- Tiden det tog från den första paketet att komma till det sista.
- Start och sluttid.
- Bits per sekund.
- Bytes per sekund.
- Medelstorleken på paketen.
- Medelvärdet av antal paket per sekund som inkommit.

6 Resultat - Litteraturanalys

Detta kapitel avser att presentera de resultat som framkommit under genomförandet av litteraturanalysen.

6.1 DHCP

DHCP är som beskrivits innan en tjänst som används för att tilldela klienter och annan utrustning med IP-adresser och annan relaterad information. IP-adresser kan tilldelas både dynamiskt och statiskt. En dynamisk tilldelning går ut på att klienter och annan utrustning tilldelas IP-adresser från en pool, det vill säga en samling av IP-adresser. En statisk tilldelning menas med att en MAC-adress kopplas till en specifik IP-adress. Det gör att exempelvis en klient alltid blir tilldelad samma IP-adress varje gång den ansluter till nätverket.

6.1.1 Likheter

De studerade mjukvaruroutrarna stöder alla denna tjänst.

Gemensamt kan de alla även tilldela vilken:

- Gateway och DNS-server som ska användas av enheten.
- NTP (Network Time Protocol)-server som enheten ska konfigureras med. Det vill säga vilken tidsserver skall enheten synkronisera sin klocka med.
- TFTP (Trivial File Transfer Protocol)- och "Network booting"-server som enheten ska konfigureras med.

6.1.2 Skillnader

Pfsense DHCP-tjänst kan konfigureras till att enbart tillåta de med statiska mappningar att kommunicera med routern. Om en enhet inte finns med i den statiska listan så kan den inte kommunicera med routern (Buechler & Pingle, 2009). Vidare är Pfsense DHCP-tjänst som standard aktiverad på de nätverksgränssnitt som vetter mot LAN:en. De pooler som uppkommer i denna standardkonfiguration använder sig poolerna av adresserna x.x.x.10-x.x.x.199 (Buechler & Pingle, 2009).

ClearOS DHCP-tjänst är inte installerad som standard utan den måste laddas ner och installeras ifrån dess applikationscentral (ClearCenter, 2012:*a*). Applikationscentralen är där ClerOS- och andre tredjepartsutvecklare lägger upp sina applikationer och moduler som kan användas på routern. När tjänsten installerats så är den förkonfigurerad till att använda poolen x.x.x.100-x.x.x.254 (ClearCenter, 2012:*a*).

Enligt Vyatta (2012:*a*) så kan deras DHCP-tjänst konfigureras till att även tilldela följande saker:

- Om enheten skall konfigureras för att kunna dirigera trafik.
- Vilken POP3 (Post Office Protocol3)- och/eller SMTP (Simple Mail Transfer Protocol)-server som finns tillgänglig.
- Statiska rutter.
- Proxy inställningar för webbläsaren.
- Domännamn som enheten ska konfigureras med.

6.2 NAT

NAT är som beskvits i bakgrunden en metod för att översätta IP-adresser. Antingen översätta privata IP-adresser till publika IP-adresser eller tvärtom. Vidare ingår även portforwarding i detta begrepp och det är att skicka vidare anslutningar som inkommer till en specifik port till en specifik enhet och port. Exempelvis anslutningar som inkommer på 10000 skall skickas vidare till enhet B och port 60. En port i denna kontext är ett nummer som talar om för enheten vilken typ av applikation som ska ha all data.

6.2.1 Likheter

Gemensamt för Pfsense och ClearOS är att NAT är aktiverat på de nätverksgränssnitt som vetter mot Internet. Det medför att all utgående trafik översätts till publika adresser. (Buechler & Pingle, 2009;Clearcenter, 2012:*b*).

Alla mjukvaror stöder 1:1 och 1:N-NAT. Det vill säga statiska översättningar mellan specifika IP-adresser och översättningar där enbart en publik IP-adress används till översättningar. För att särskilja alla anslutningar vid 1:N-NAT så används unika portnummer. Utöver detta kan även alla mjukvaror filtrera sin NAT-trafik beroende på portnummer, protokoll och IP-adress.

Mjukvarorna har även stöd för portforwarding.

6.2.2 Skillnader

Pfsense kan konfigureras med "NAT-Refelction" vilket möjliggör att interna enheter kan nå externa enheter som finns inom nätverket. Det vill säga om en det finns en webbserver med en publik och en privat IP-adress så kan interna enheter komma åt denne server genom att ansluta till dess publika IP-adress. Dock har dessa vissa begränsningar, portar över 500 kan inte nås och 1:1-NAT stöds ej samtidigt som denna metod används (Buechler & Pingle, 2009).

ClearOS 1:1-NAT är inte installerat som standard utan dess applikationscentral måste användas för att hämta hem denna funktion (Clearcenter, 2012:*b*).

Vyatta har inte NAT-konfiguret som standard (Vyatta, 2012*:b*). Dock kan Vyattas NAT konfigureras på några fler sätt än de andra mjukvarorna . Enligt Vyatta (2012:*b)* stöds även dessa typ av NAT:

- M:N. Flera interna IP-adresser översätts till flera publika IP-adresser.
- 1:N. Flera publika IP-adresser översätts till en intern IP-adress.
- Masquerade. Denna metod som används vid de tillfällen då WAN-gränssnittet har en dynamisk IP-adress. De interna IP-adresserna översätts till den IP-adress som WAN-gränssnittet har. Även här skiljs anslutningarna åt med hjälp utav unika portar. Metoden används då det inte går att specificera vilken publik IP-adress som ska användas vid översättningen då den inte är statisk.

6.3 Paketfiltrering

Paketfiltrering är som beskrivits i bakgrunden en metod för att filtrera trafik beroende en rad olika variabler. Filtrering kan hanteras på två sätt, tillståndskänslig (stateful) och icke tillståndskänslig (stateless). Den första metoden går ut på att enheten håller reda på alla

anslutningar och kan göra beslut per anslutning. Icke tillståndskänslig filtrering går ut på att varje paket inspekteras, oavsett om ett liknande paket godkänns sen innan.

Filtreringen bygger på att regler sätts upp och de läses av i en kronologisk ordning, från topp till botten. Vid en matchning så utförs en handling.

6.3.1 Likheter

De alla har stöd för paketfiltrering och enligt systemdokumentationen kan de filtrera trafik beroende på dessa variabler:

- IP-adress. Antingen källans adress eller destinationsadressen.
- Protokoll.
- Portnummer.
- Nätverksadress.
- MAC-adress (Vyatta och ClearOS).
- Klockslag.
- Anslutningar per sekund.

Som standard policy är paketfiltrerarna konfigurerade att blockera all trafik som inte godkänns, även kallat "implicit deny" (Buechler & Pingle, 2009;Vyatta, 2012*:c*)

6.3.2 Skillnader

Noterbara skillnader är att Vyatta och ClearOS använder sig av Iptables[5] som sin paketfiltrerare medan Pfsense använder sig utav Pf[6]. Vidare så är Pfsense paketfiltrerare tillståndskänslig som standard vilket de andra inte är (Buechler & Pingle, 2009; Clearcenter, 2012:*c;*Clearcenter, 2012:*d*;Vyatta, 2012*:c*).

Som standard är Pfsense och ClearOS paketfiltrerare aktiverad medan Vyattas paketfiltrerare får aktiveras manuellt. Dock är paketfiltrerarna enbart konfigurerade för att blockera alla inkommande anslutningar från Internet (Buechler & Pingle, 2009; Clearcenter, 2012:*l*). Om annan filtrering skall göras så får detta konfigureras manuellt. Dessutom kan ingen annan konfiguration göras på ClearOS webbgränssnitt utan mjukvaran kräver att andra brandväggsmoduler får installeras först (Clearcenter, 2012:*m*; Clearcenter, 2012:*n)*. Det vill säga om utgående trafik ska filtreras sp måste en modul laddas ner från applikationscentralen.

Pfsense har möjligheten att skriva regler baserat på vilket operativsystem som trafiken tillhör. Det har dessutom identifierats att Pfsense kan agera som en proxy för alla TCP-anslutningar (TCP utför en trevägs handskakning för att initiera anslutningar). Efter avklarad handskakning får den berörda enheten som trafiken var ämnad för tillbaka makten. Vidare kan även trafiken kontrolleras mot routingtabellen för att se ifall IP-adressen är "spoofad" eller ej. Det vill säga om någon angripare utgör sig att sitta på en enhet som den inte gör. Pfsense "Anti-spoofing" regel gör att trafiken kontrolleras mot dess routingtabell och om trafik kommer från WAN-gränssnittet (Internet) med en avsändaradress som finns på det lokala nätverket så kastas paketet. Om trafik initieras från det lokala nätverket med en avsändaradress som inte tillhör det lokala nätverket så kastas även de paketen (Buechler & Pingle, 2009). En annan funktion är att Pfsense blockerar trafik som inkommer på WAN-

[5] http://linux.die.net/man/8/iptables
[6] http://www.openbsd.org/faq/pf/

gränssnittet med en privat IP-adress som avsändaradress. Ytterligare en funktion som Pfsense kan erbjuda är att blockera "Bogon"-nätverk, vilket är nätverksadresser som inte skall finnas på Internet, så som reserverade och icke allokerade IP-adresser. Detta är dock IP-adresser som kan kommas att bli allokerade i framtiden och det betyder att Pfsense behöver hålla sig uppdaterad om dessa adresser. Detta sköts genom att en lista laddas ner från Pfsense hemsida som har dessa "bogus" nätverk. Uppdateringar sker kontinuerligt men listan laddas per standard bara ner en gång i månaden (Buechler & Pingle, 2009).

Vyattas regler kan konfigureras att gälla per nätverksgränssnitt eller för en zon, där en zon är ett eller fler nätverksgränssittgränssnitt (Vyatta, 2012*:c*). En av dessa metoder kan enbart vara aktiverat per nätverksgränssnitt. Zonmetoden gäller både för ingående och utgående trafik men då specificeras det från och till vilken zon instansen (samling regler) ska gälla för. Trafik menad för enheter inom samma zon kringgår paketfiltreringen (Vyatta, 2012*:c*).

6.4 VPN

Som beskrivits tidigare är VPN en metod för att sätta upp tunnlar mellan två enheter. Detta för att ge tillgång till resurser till exempelvis de som arbetar hemifrån.

6.4.1 Likheter

De alla ska enligt deras systemdokumentation ha tillgång till IPsec, OpenVPN, PPTP (Point-to-Point Tunneling Protocol) och L2TP (Layer 2 Tunneling Protocol) (Buechler & Pingle, 2009;Clearcenter, 2012:*e*; Vyatta, 2012*:d)*. Med undantaget för ClearOS som inte stöder L2TP.

Det har identifierats att alla mjukvaror har stöd för följande autentiseringsmetoder: användarnamn och lösenord, delade krypteringsnycklar och certifikat.

6.4.2 Skillnader

Om någon VPN-tjänst ska införskaffas på en ClearOS-maskin så måste de först installeras via deras applikationscentral.

Pfsense och Vyatta IPsec implementation har enbart stöd för IKEv1 (Internet Key Exchange) medan ClearOS har stöd för IkEv2 (Buechler & Pingle, 2009;Clearcenter, 2012*:g*;Vyatta, 2012*:d*). IKE är det protokoll som förhandlar om vilka säkerhetsparametrar som ska gälla. Dock så kan enbart Pfsense och ClearOS IPsec tunnlar konfigureras med "aggressive mode" (Buechler & Pingle, 2009;Vyatta, 2012*:d*). Detta kortar ner antalet meddelanden som behöver skickas mellan enheterna för att etablera vilka säkerhetsparametrar som skall användas. Alla mjukvarors IPsec implementation har stöd för ESP (Encapsulating Security Payload). Det vill säga stöd för kryptering av payload.

En annan noterbar skillnad gällande ClearOS IPsec-tunnlar är att det går att konfigurera dessa tunnlar att fungera med dynamiska IP-adresser. Något som inte identifierats hos de andra mjukvarorna. Detta ska enligt ClearCenter (2012:*f*) fungera på så sätt att de tar hand om denna administration på ett dynamiskt vis, via deras "ClearSDN" (Service Delivery Network) Dock kan denna metod enbart sätta upp tunnlar mellan ClearOS-maskiner (Clearcenter, 2012:*f*). Detta kostar dock 100USD/år (Clearcenter, 2012:*e*). Statiska tunnlar är fortfarande gratis.

En sista noterbar skillnad är också att Vyatta har tillgång VTI (Virtual Tunnel Interface) vilket är ett virtuellt tunnel nätverksgränssnitt. Detta virtuella nätverksgränssnitt agerar som ett vanligt nätverksgränssnitt och kan därför konfigureras med bland annat QoS och utsättas för paketfiltrering (Vyatta, 2012*:d)*.

6.5 QoS

Quality Of Service är en metod för att forma eller begränsa nätverkstrafiken efter behov. Detta kan göras genom att exempelvis klassificera eller prioritera viss trafik och kategorisera de olika typerna av trafik till olika köer. Dessa köer formar eller begränsar trafiken efter de kösystem som satts upp, så som först in först ut.

6.5.1 Likheter

Alla tre mjukvaror stöder QoS, dock till varierande grad.

6.5.2 Skillnader

Pfsense formar trafiken med hjälp av ALTQ (Alternate Queueing) som är ett ramverk för BSD operativsystem (Buechler & Pingle, 2009; Cho, 2004). Det ramverk som de andra mjukvarorna använder har inte identifierats.

Konfigurationen skiljer sig också mellan mjukvarorna då Pfsense QoS kan konfigureras på två sätt, manuellt eller med en guide medans de andra mjukvarornas QoS måste konfigureras manuellt.

Det har observerats att ClearOS har enbart stöd för några få konfigurationsmöjligheter i relation till de andra mjukvarorna. ClearOS QoS kan konfigureras till att begränsa hur många kilobit per sekund som ett nätverksgränssnitt eller tjänst får använda (Clearcenter 2012:*k*). Samt om QoS konfigurationen skall vara girig eller ej. Det vill säga om den ska försöka använda sig utav icke använd kapacitet. Desto girigare desto mer av den icke använda kapaciteten kommer den försöka att använda (Clearcenter 2012:*k*).

Pfsense och Vyattas QoS kan konfigureras till en större grad än ClearOS med fler typer av köer. Noterbart är att Vyattas QoS är aktiverat som standard och den prioriterar trafik beroende på vilken typ av trafik det är. Typen baseras på TOS- (Type Of Service) flaggan i varje paket (Vyatta, 2012*:e*). Vyatta har tre fördefinierade köer: Högst, näst högst och lägst prioritet. Paket i den högst prioriterade kön skickas först och de i den lägst prioriterade kön skickas sist. Paketen i köerna skickas ut i den ordning de inkommer, enligt FIFO (First In First First Out)-metoden (Vyatta, 2012*:e*).

Andra skillnader är också att Pfsense stöder dessa typer av köer (Buechler & Pingle, 2009):

- PRIQ (Priority Queueing) medför att den trafik som prioriterats högts kommer alltid att lämna routern först.
- CBQ (Class Based Queueing) som möjliggör att olika typer av trafik kan grupperas i olika grupper. Dessa grupper kan sedan prioriteras.
- HFSC (Hierarchical Fair Service Curve) som bygger på att köerna är uppbyggda på ett hierarkiskt vis med en "förälder" kö och flera "barn"- och "sub-barn"-köer. Dessa köer kan anpassas efter behov beroende på vilken typ av trafik som passerar genom Pfsense.

Samt att Pfsense kan konfigurera med dessa alternativ för varje kö (Buechler & Pingle, 2009):

- Kön är standard kö för all trafik som inte matchar någon kö.
- ACK/low-delay kö. Viktiga ACK-paket prioriteras.
- RED (Random Early Detection) som är en metod för att undvika överbelastning inom kön. Vid överbelastning kastas slumpmässiga paket.
- RIO (Random Early Detection In and Out) som är som RED fast den upprätthåller två köer, en för inkommande och en för utgående trafik.
- ECN (Explicit Congestion Notification) som medför att mängden trafik kan sänkas för en länk. Routern talar om för enheten som skickar data att länken är överbelastad och om enheten förstår detta så sänker den sin hastighet.
- Kön är en förälder kö för andra köer. "Barnköer" ärver föräldrakös konfiguration.

En annan skillnad som observerats är att Pfsensemjukvaran har möjligheten att kunna finjustera QoS-köerna till en hur många procent av den totala kapaciteten som de får använda.

Enligt Vyattas (2012:*e*) systemdokumentation stöds dessa QoS konfigurationsmöjligheter för utgående trafik:

- Drop-Tail. FIFO teknik som kastar de sist inkomna paketen om kön blir full.
- Fair Queue. Flöden av paket delas in i köer.
- Round-Robin. Trafik delas in i klasser och varje har likvärdigt mycket kapacitet.
- Traffic Shaper. Likt Round-Robin fast icke allokerad kapacitet kan användas av köerna. Kortfattat om en kö har en begränsning på 100 kb/s och det finns 100 kb/s icke använd kapacitet så kan kön vid de tillfällen trafiken överskrider 100 kb/s använda hela eller delar av den icke allokerade kapaciteten.
- Rate Control. Begränsar kapaciteten till ett visst värde.
- Random Detect. Innehåller både RED och WRED (Weighted RED). RED kastar slumpvis valda paket vid överbelastning medan WRED kan konfigureras till kasta mer paket från specifika köer.
- Priority Queue. Upp till sju köer kan konfigureras. Varje kö har en prioritet och den med högs skickar ut sina paket först. Den med lägst skickar ut sina paket sist.

Enligt Vyatta (2012*:e)* kan enbart en typ av QoS appliceras på inkommande trafik:

- Traffic Limiter. Begränsar trafik till en viss kapacitet. De paket som överskrider kapaciteten kastas.

6.6 Dynamiska routingprotokoll

Som beskrivits innan i bakgrunden kan routingtabellen fyllas på två sätt, statiskt eller dynamiskt. Det vill säga att den fylls av en administratör eller av ett dynamiskt routingprotokoll.

Den enda mjukvaran som för nuläget inte har stöd för dynamiska routingprotokoll är ClearOS.

6.6.1 Likheter
Pfsense och Vyatta har stöd för dessa dynamiska routing protokoll:

- OSPF (Open Shortest Path First)
- RIP (Routing Information Protocol)
- BGP (Border Gateway Protocol)

6.6.2 Skillnader
Skillnaderna som existera är att Vyatta använder sig utav Quagga medan Pfsense använder sig utav OpenBSD-protokoll som standard. Dock kan Quagga-protokoll hämtas hem och användas på Pfsenseroutern (Pfsense, 2013:*a*).

Vidare har Vyatta stöd för RIPing vilket är RIP för IPv6 (Guillen, Sossa, & Estupñán, 2012). Detta har inte Pfsense stöd för men det har stöd för OLSR (Open Link State Routing Protocol). OLSR är ett protokoll som är optimerat för mobila ad-hoc nätverk (IETF, 2003).

6.7 Support
Support innebär att någon får hjälp med att lösa en given uppgift.

6.7.1 Likheter
Det har identifierats att alla utvecklare till dessa mjukvaror kan erbjuda support via två medium, forum och den onlinedokumentation som existerar.

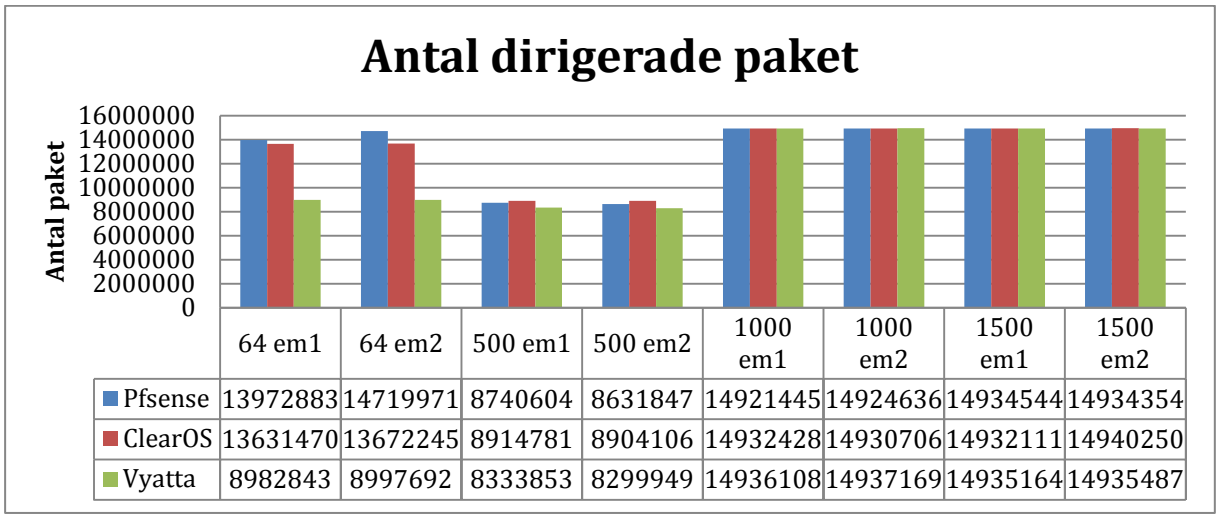
6.7.2 Skillnader
Skillnaden är att Pfsense kan erbjuda support via andra medium också. Dock kostar denna typen av support 600 USD per år för fem timmars support. Noterbart är att ytterligare timmar kan köpas till om behovet skulle uppstå. Enligt utvecklarna kan de erbjuda hjälp med konfiguration, nätverksdesign och migrering från andra produkter (Pfsense,2013:*b*).

Vidare har utvecklarna av Pfsense släppt boken "Pfsense: The Definitive Guide" som tar upp allt som berör mjukvaran.

7 Resultat - Experimentet

Som beskrivits innan så har det skickats 15mlj paket till de båda nätverksgränssnitten, det vill säga en total mängd av 30mlj paket har skickats per test för att prestandatesta mjukvaruroutrarna.

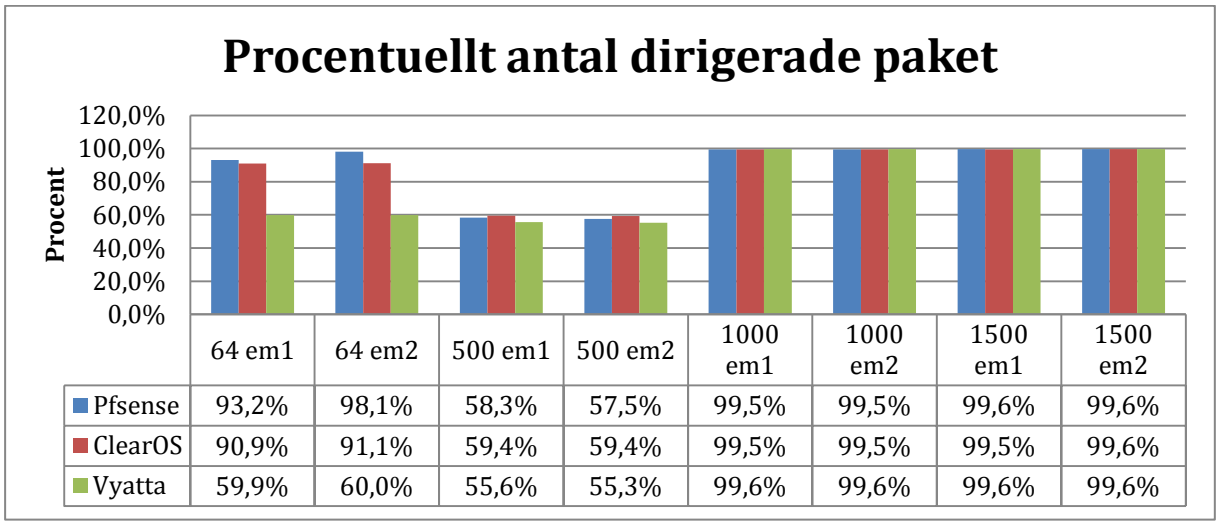
Antal framkomna paket till DK1 och DK2 kan ses i figur 1. Kolumnerna med samma namn representerar varsitt test. 64 em1 och 64 em2 representerar i detta fallet det test där paketstorleken var satt till 64 bytes.



	64 em1	64 em2	500 em1	500 em2	1000 em1	1000 em2	1500 em1	1500 em2
Pfsense	13972883	14719971	8740604	8631847	14921445	14924636	14934544	14934354
ClearOS	13631470	13672245	8914781	8904106	14932428	14930706	14932111	14940250
Vyatta	8982843	8997692	8333853	8299949	14936108	14937169	14935164	14935487

Figur 1: Visar hur många paket som mottagits på vardera sida. Em1 och Em2 är de båda nätverksgränssnitten på mjukvaruroutern.

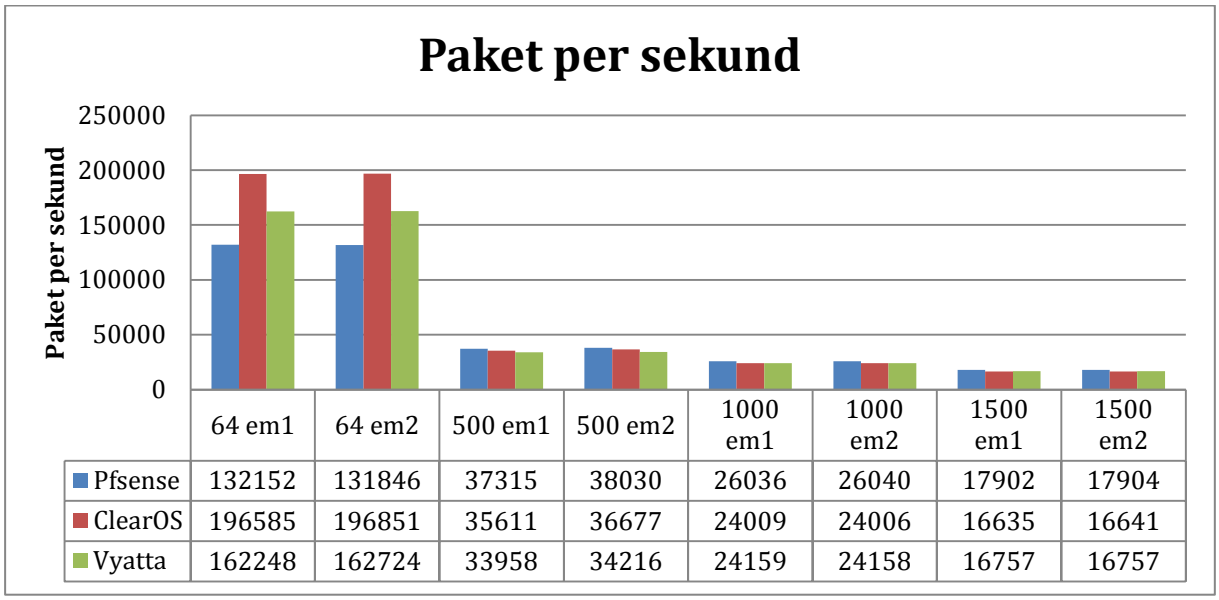
Figur 2 visar samma data som ovan fast i procent. Testerna representeras likadant som i den förra figuren. 100% motsvarar 15mlj paket.



	64 em1	64 em2	500 em1	500 em2	1000 em1	1000 em2	1500 em1	1500 em2
Pfsense	93,2%	98,1%	58,3%	57,5%	99,5%	99,5%	99,6%	99,6%
ClearOS	90,9%	91,1%	59,4%	59,4%	99,5%	99,5%	99,5%	99,6%
Vyatta	59,9%	60,0%	55,6%	55,3%	99,6%	99,6%	99,6%	99,6%

Figur 2: Visar procentuellt hur många paket som mottagits på vardera sida.

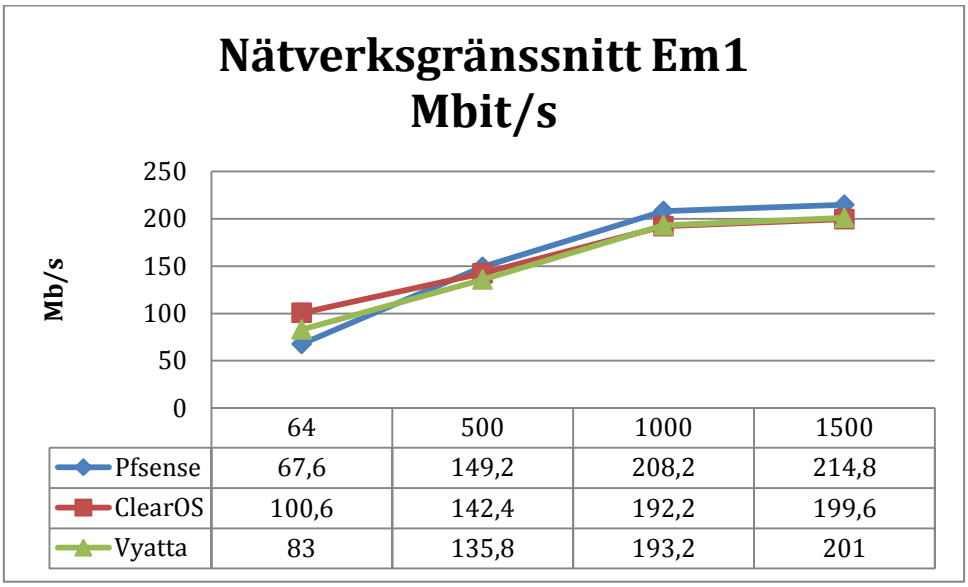
Figur 3 visar det genomsnittliga antalet dirigerade paket per sekund. Testerna representeras likadant som i den förra figuren.



	64 em1	64 em2	500 em1	500 em2	1000 em1	1000 em2	1500 em1	1500 em2
Pfsense	132152	131846	37315	38030	26036	26040	17902	17904
ClearOS	196585	196851	35611	36677	24009	24006	16635	16641
Vyatta	162248	162724	33958	34216	24159	24158	16757	16757

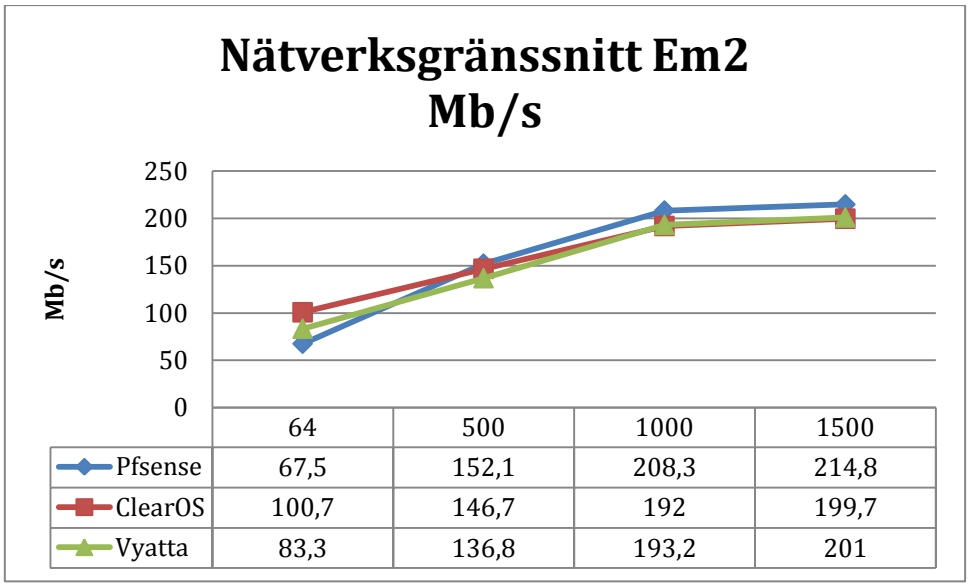
Figur 3: Paket per sekund som dirigerats.

Nästkommande två grafer, graf 1 och graf 2 visar hur många megabits per sekund som överförts per nätverksgränssnitt. Y-axeln visar megabits per sekund och X-axeln visar paketstorlek och mjukvarurouter.



	64	500	1000	1500
Pfsense	67,6	149,2	208,2	214,8
ClearOS	100,6	142,4	192,2	199,6
Vyatta	83	135,8	193,2	201

Graf 1: Visar överföringskapaciteten för nätverksgränssnittet Em1.



	64	500	1000	1500
Pfsense	67,5	152,1	208,3	214,8
ClearOS	100,7	146,7	192	199,7
Vyatta	83,3	136,8	193,2	201

Graf 2: Visar överföringskapaciteten för nätverksgränssnittet Em2.

8 Analys

I detta kapitel har resultaten från litteraturanalysen och prestandatesterna analyserats.

Mjukvarorna stöder de funktioner och tjänster som undersökts på olika vis. Pfsense och Vyatta har alla undersökta funktioner och tjänster integrerat i sin mjukvara medan ClearOS förlitar sig på sin applikationscentral. Vidare så stöder inte ClearOS bland annat dynamiska routingprotokoll och dess QoS är inte lika välgjord. Pfsense och Vyatta har mest gemensensamt gällande vilka funktioner och tjänster som stöds samt hur de kan konfigureras.

En annan noterbar skillnad som uppkom under litteraturanalysen och experimentet är att den öppna versionen av Vyattamjukvaran inte hade stöd för konfiguration via ett webbgränssnitt. Detta kunde enbart fås om mjukvaran uppgraderades till den proprietära mjukvaran.

Gällande Pfsense kan utvecklarna av denna mjukvara ge en mer väl etablerad typ av support än de andra då de kan ge tillgång till e-mail och telefonsupport. Dock kostar denna pengar och är begränsad till ett minimum av fem timmar. De alla stöder dock support via forum och den onlinedokumentation som finns.

Prestandatestet visade inga noterbara skillnader mer än vid paket på 64 bytes så kunde ClearOS prestera bäst, både gällande för pps och mb/s. Dock märktes det att vid 500 bytes så var antalet dirigerade paket lågt i relation till de andra testerna. Hur detta kom sig har inte kunnat identifierats. Då det är gemensamt för alla mjukvaror kan det röra sig om hårdvaran.

9 Slutsats

De slutsatser som kan dras är att det finns noterbara skillnader mellan mjukvarorna. Det som skiljer sig mest är vilka funktioner och tjänster som stöds samt vilken typ av support som kan fås av utvecklarna. Den mjukvara som stuckit ut mest är ClearOS då denne mjukvara inte har så många funktioner och tjänster installerat som standard. En ytterligare detalj är att ClearOS kräver att ett flertal konfigurationsmoment genomförs efter installationen innan den kan användas. Prestandatesterna visade enbart en skillnad och det var att ClearOS överföringskapacitet vid 64 bytes paket. Kapaciteten var större än de andras vid samma test.

10 Diskussion och framtida arbete

Detta kapitel har för avsikt att ge en diskussion om arbetet och de resultat som framkommit samt att ge förslag till framtida arbeten.

10.1 Diskussion

Resultaten gällande de två första delmålen har främst kommit från den systemdokumentation som finns. Detta medför att resultaten inte har validerats mot det verkliga livet. De vill säga det har inte undersökt om det som stått skrivet verkligen stämmer. Gällande prestandatestet så har det inte gett någon insikt i vilka skillnader som existerar, mer än ClearOS resultat vid paket på 64 bytes.

Noterbart var att de alla presterade så pass dåligt vid 500 bytes. Då alla presterar likvärdigt så tordes det vara hårdvaran men ingen definitiv förklaring har hittats. Dock varför de inte kunde prestera mer än 200mb/s har inte heller hittats men ett arbete från 2005 av Bianco m.fl och ett arbete från 2008 av Ye & Macgregor studerade bland annat vilka flaskhalsar hårdvaran har på en mjukvarurouter. De angav CPU, PCI bussen och nätverkskorten som potentiella flaskhalsar. Då detta experiment har använts sig utav lite äldre hårdvara så är det mycket troligt att CPU och PCI-bussen begränsat mjukvarurouterns kapacitet.

Gällande de tjänster och protokoll som denna studie har analyserat så är de enbart ett subset av all de tjänster och protokoll som en router kan stödja. Att analysera alla tjänster och protokoll hade gett en större förståelse om alla de likheter och skillnader som existerar. Dock var detta inte rimligt i denna studie då arbetsbördan hade blivit för stor. Den avgränsning som gjordes var att enbart studera vissa centrala tjänster och protokoll.

Nyttan med detta arbete är att det ger intressenter information om vilka funktioner och tjänster som stöds samt vilken typ av support som kan fås av utvecklarna. Denna information kan hjälpa dem att välja vilken mjukvara som passar de bäst. Som beskrivits innan så kan dock inte prestandatesten användas för att urskilja vilken mjukvara som presterar bäst eller sämst. Denna information får uppsökas på egen hand.

10.2 Framtida arbeten

Ett framtida arbete skulle kunna vara att validera ett urval av funktioner och tjänster för att se ifall det som skrivs i dokumentationen stämmer. Det behöver även utföras fler jämförelsestudier då det finns en avsaknad av dessa. Ett framtida arbete skulle kunna vara att undersöka vilka prestandaskillnader som existerar mellan en hårdvarurouter och en mjukvarurouter. Vidare bör det undersökas varför prestandan sjönk när det skickades paket som var 500 bytes stora.

Referenser

Argyraki, K., Baset, S., Chun, B. G., Fall, K., Iannaccone, G., Knies, A., Kohler, E., Manesh, M., Nedevschi, S. & Ratnasamy, S. (2008*) Can software routers scale?*. Proceedings of the ACM Workshop on Programmable Routers for Extensible Services of Tomorrow. Seattle, Washington, USA, 22 augusti.

Bianco, A., Finochietto, J., Galante, G., Mellia, M. & Neri, F. (2005) Open-source PC-based software routers: A viable approach to high-performance packet switchin. I: Marsan, M. A., Bianchi, G., Lisani, M. & Meo, M (red:er), *Quality of Service in Multiservice IP Networks* (s. 353-366). Proceedings of the Third International Workshop on Quality of Service in Multiservice IP Networks, QoS-IP 2005, 2-4 februari, 2005, Catania, Italien.

Buechler, C, M & Pingle, J. (2009) *Pfsense : The Definitive Guide: The Definitive Guide to the Pfsense Open Source Firewall and Router Distribution*. Reed Media Services.

Cisco. (2010) *Cisco Introduces Foundation for Next-Generation Internet: The Cisco CRS-3 Carrier Routing System*. Cisco Systems Inc. Kalifornien, USA. Tillgänglig på Internet: http://newsroom.cisco.com/dlls/2010/prod_030910.html [Hämtad 2013.02.24].

Cho, K. (2004) *ALTQ: Alternate Queueing for BSD UNIX* Tillgänglig på Internet: http://www.sonycsl.co.jp/~kjc/ [Hämtad 2013.04.15]

Distrowatch. (2013) *Search Distributions*. Unsigned Integer Limited. Hong Kongm Kina. Tillgänglig på Internet: http://distrowatch.com/search.php?category=Firewall#distrosearch [Hämtad 2013.02.24]

Guillen, E., Sossa, A.M. & Estupñán, E.P. (2012) Performance Analysis over Software Router vs. Hardware Router: A Practical Approach. I: Ao, S. I., Douglas, C., Grundfest, W. S & Burgstone, J (red:er), *Proceedings of the World Congress on Engineering and Computer Science 2012* (s. 973-987). International Conference on Communications Systems and Technologies, 24-26 oktober, 2012, San Fransisco, USA.

Gördén, B., Hagsand, O. & Olsson, R. (2008) *Towards 10Gbps open-source routing*. KTH Tillgänglig på Internet: http://www.iis.se/docs/10G-OS-router_2_.pdf [Hämtad 2013-02-18]

Han, S., Jang, K., Park, K. & Moon, S. (2010) Building a single-box 100 gbps software router. I: IEEE (red:er), *IEEE LANMAN 2010* (s.). The 17th IEEE International Workshop on Local and Metropolitan Area Networks, 5-7 maj, 2010, Long Beach, New Jersey, USA.

Internet Engineering Task Force. (1981) *Internet protocol*. Request for Comments 791. Tillgänglig på Internet: http://tools.ietf.org/html/rfc791 [Hämtad 2013-04-03]

Internet Engineering Task Force . (1984) *A Standard for the Transmission of IP Datagrams over Ethernet Networks*. Request for Comments 894. Tillgänglig på Internet: http://tools.ietf.org/html/rfc894 [Hämtad 2013-04-09]

Internet Engineering Task Force . (2003) *Optimized Link State Routing Protocol (OLSR)*. Request for Comments 3626. Tillgänglig på Internet: http://tools.ietf.org/html/rfc3626 [Hämtad 2013.05.19]

Juniper. (2011) *Juniper Networks. Networks and Security Manager*. Juniper Networks, Inc. Kalifornien, USA. Tillgänglig på Internet: http://www.juniper.net/techpubs/software/management/security-manager/nsm2010_1/nsm-api-guide.pdf [Hämtad 2013.02.23].

Juniper. (2013) *Juniper Networks. ACX Serial Universal Access Router Configuration Guide*. Juniper Networks, Inc. Kalifornien, USA. Tillgänglig på Internet: http://www.juniper.net/techpubs/en_US/junos12.3/information-products/pathway-pages/acx-series/acx-series.pdf [Hämtad 2013.06.10].

Olsson, R. (2005) *Pktgen the linux packet generator*. In proceedings of the Linux Symposium. Ottawa, Ontario Canada, 20-23 juli

Open Source Initiative. (2013) *Open Source Licenses*. Open Source Initiative. Kalifornien, USA. Tillgänglig på Internet http://opensource.org/licenses [Hämtad 2013.02.24]

Parziale, L., Britt, T. D., Davis, C., Forrester, J., Lie, W., Matthews, C. & Rosslot, N. (2006) *TCP/IP Tutorial and Technical Overview* (8:e upplagan). International Technical Support Organization. Tillgänglig på Internet: http://www.redbooks.ibm.com/redbooks/pdfs/gg243376.pdf [Hämtad 2013-02-08]

Pfsense (Version: 2.0.3) (2013*:a*) [Datorprogram] Austin TX: BSD Perimeter, LLC. Tillgänglig på Internet: http://www.pfsense.org [Hämtad 2013.04.12]

Pfsense. (2013*:b*) *Support Subscription*. Pfsense.org. Tillgänglig på Internet: https://portal.pfsense.org/index.php/support-subscription [Hämtad 2013-05-19]

Wikipedia. (2013) *List of router or firewall distributions*. Wikipedia Foundation, Inc. Kalifornien, USA. Tillgänglig på Internet: http://en.wikipedia.org/wiki/List_of_router_or_firewall_distributions [hämtad 2013.02.24]

Ye, Q. & MacGregor, M. H. (2008) Hardware bottleneck evaluation and analysis of a software PC-based router. I: Obaidat, S. M., Marzo, L. J., Szczerbicka, H. & Vila, P (red:er), *Performance Evaluation of Computerand Telecommunication Systems* (s.480-487). Proceedings of the 2008 International Symposium on Performance Evaluation of Computer and Telecommunication Systems, juni 16-18, 2008, Edinburgh, Storbritannien.

Systemdokumentation

Clearcenter. (2012:*a*) *DHCP Server*. Clearcenter, Corp. Tillgänglig på Internet: http://www.clearcenter.com/support/documentation/clearos_enterprise_5.1/user_guide/dhcp_server [Hämtad 2013-04-22]

Clearcenter. (2012:*b*) *NAT Firewall*. Clearcenter, Corp. Tillgänglig på Internet: http://www.clearcenter.com/support/documentation/user_guide/nat_firewall [Hämtad 2013-04-23]

Clearcenter. (2012:*c*) *ClearOS 6.x*. Clearcenter, Corp. Tillgänglig på Internet: http://www.clearfoundation.com/docs/developer/packaging/clearos_6.x/start [Hämtad 2013-04-23]

Clearcenter. (2012:*d*) *IPTABLES*. Clearcenter, Corp. Tillgänglig på Internet: http://www.clearfoundation.com/docs/man/index.php?s=8&n=iptables [Hämtad 2013-04-23]

Clearcenter. (2012:*e*) *Marketplace*. Clearcenter, Corp. Tillgänglig på Internet: http://www.clearcenter.com/marketplace [Hämtad 2013-04-25]

Clearcenter. (2012:*f*) *Dynamic VPN*. Clearcenter, Corp. Tillgänglig på Internet: http://www.clearcenter.com/Services/clearsdn-dynamic-vpn-6.html [Hämtad 2013-04-25]

Clearcenter. (2012:*g*) *Static IPsec VPN*. Clearcenter, Corp. Tillgänglig på Internet: http://www.clearcenter.com/support/documentation/user_guide/static_ipsec_vpn [Hämtad 2013-04-26]

Clearcenter. (2012:*h*) *PPTP VPN*. Clearcenter, Corp. Tillgänglig på Internet: http://www.clearcenter.com/support/documentation/user_guide/pptp_server [Hämtad 2013-04-26]

Clearcenter. (2012:*i*) *OPEN VPN*. Clearcenter, Corp. Tillgänglig på Internet http://www.clearcenter.com/support/documentation/user_guide/openvpn [Hämtad 2013-04-26]

Clearcenter. (2012:*k*) *Bandwidth Manager*. Clearcenter, Corp. Tillgänglig på Internet http://www.clearcenter.com/support/documentation/user_guide/bandwidth_manager [Hämtad 2013-04-26]

Clearcenter. (2012:*l*) *Firewall Incomming*. Clearcenter, Corp. Tillgänglig på Internet: http://www.clearcenter.com/support/documentation/clearos_enterprise_5.1/user_guide/firewall_incoming [Hämtad 2013-05-15]

Clearcenter. (2012:*m*) *Custom Firewall*. Clearcenter, Corp. Tillgänglig på Internet: http://www.clearcenter.com/support/documentation/user_guide/custom_firewall [Hämtad 2013-05-15]

Clearcenter. (2012:*n*) *Egress Firewall*. Clearcenter, Corp. Tillgänglig på Internet: http://www.clearcenter.com/support/documentation/user_guide/egress_firewall [Hämtad 2013-05-15]

Vyatta. (2012*:a*) *Refernce Guide: Services*. Vyatta, Inc. Belmont, Kalifornien. Tillgänglig på Internet: http://www.vyatta.com/downloads/documentation/VC6.5/Vyatta-Services_6.5R1_v01.pdf [Hämtad 2013-04-27]

Vyatta. (2012*:b*) *Refernce Guide: NAT*. Vyatta, Inc. Belmont, Kalifornien. Tillgänglig på Internet: http://www.vyatta.com/downloads/documentation/VC6.5/Vyatta-NAT_6.5R1_v01.pdf [Hämtad 2013-04-27]

Vyatta. (2012*:c*) *Refernce Guide: Firewall*. Vyatta, Inc. Belmont, Kalifornien. Tillgänglig på Internet: http://www.vyatta.com/downloads/documentation/VC6.5/Vyatta-Firewall_6.5R1_v01.pdf [Hämtad 2013-04-27]

Vyatta. (2012*:d*) *Refernce Guide: VPN*. Vyatta, Inc. Belmont, Kalifornien. Tillgänglig på Internet: http://www.vyatta.com/downloads/documentation/VC6.5/Vyatta-VPN_6.5R1_v01.pdf [Hämtad 2013-04-27]

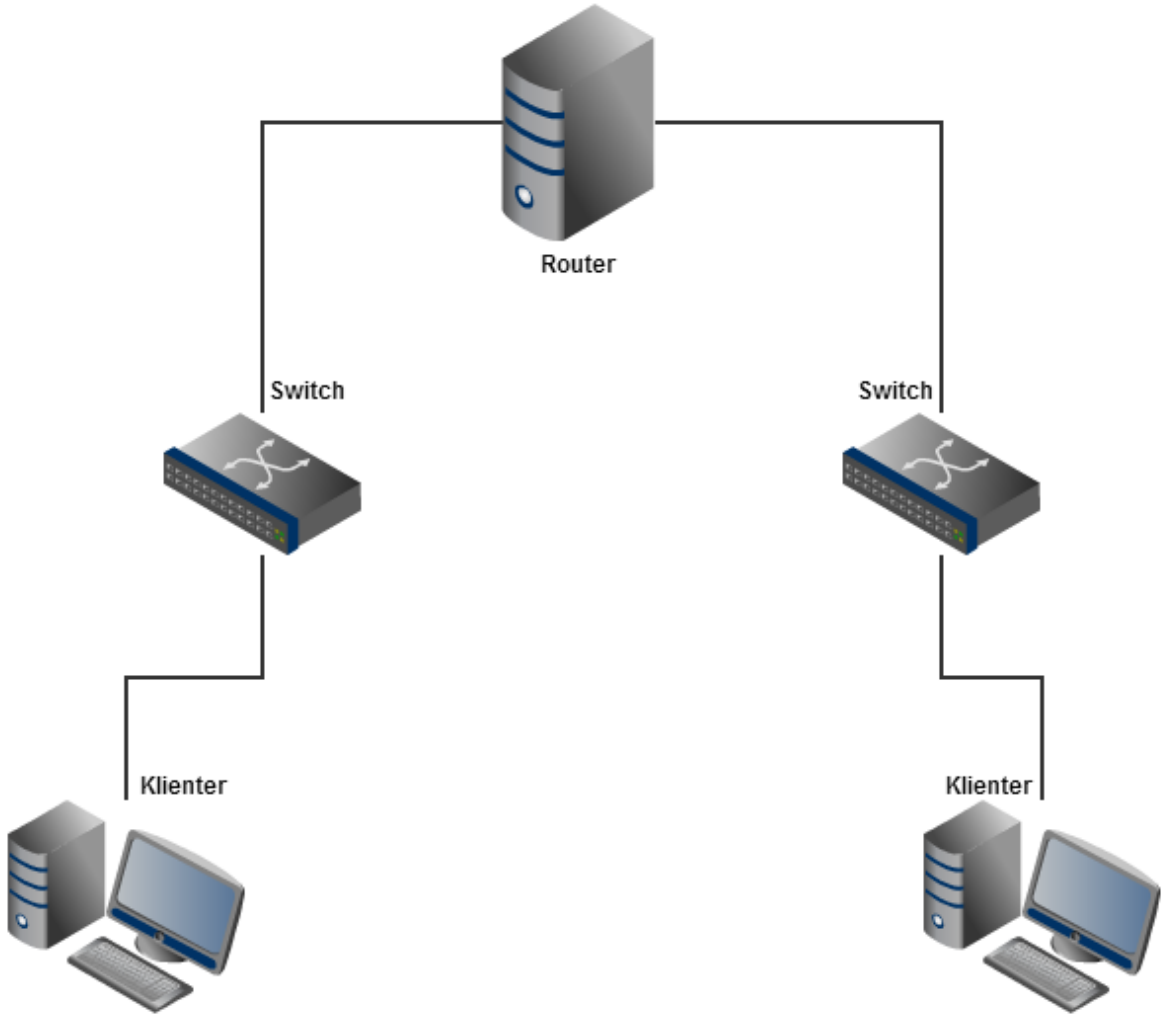
Vyatta. (2012*:e*) *Refernce Guide: QoS*. Vyatta, Inc. Belmont, Kalifornien. Tillgänglig på Internet: http://www.vyatta.com/downloads/documentation/VC6.5/Vyatta-QoS_6.5R1_v01.pdf [Hämtad 203-04-27]

Appendix A - Popularitet

Nedan tabell innehåller distributioner och mjukvaror från Wikipedias och Distrowatchs listor. De som inte togs med är de var på annat språk än engelska eller svenska eller var rena brandväggar. Sökningen gjordes 2013-03-03.

Distribution/ Mjukvara	Typ	Aktivitet	Sökresultat	Källa
Pfsense	Router/Brandvägg	2012-12-21	1 720 000	Wikipedia & Distrowatch
ClearOS	Router/Brandvägg/ Server	2013-02-22	1 010 000	Wikipedia & Distrowatch
Mikrotik RouterOS	Router	2013-02-22	892 000	Wikipedia
Vyatta	Router/Brandvägg	2012-11-20	791 000	Distrowatch & Wikipedia
Zentyal	Router/Brandvägg/ server	2013-03-01	584 000	Wikipedia
ClarkConnect	Router/Brandvägg	Inaktiv 2009	423 000	Wikipedia
M0n0wall	Router/Brandvägg	2012-11-12	411 000	Wikipedia & Distrowatch
Zeroshell	Router	2012-09-21	190 000	Wikipedia & Distrowatch
Freesco	Router	2012-02-05	158 000	Wikipedia
Fli4i	Router	2012-09-16	150 000	Wikipedia
BSD Router Project	Router	2013-01-14	136 000	Wikipedia
Devils-Linux	Router/Brandvägg	2012-12-31	63 900	Distrowatch & Wikipedia
Linux Embedded Applicance Framework	Router/Brandvägg/ Gateway/AP	2013-01-29	48 700	Wikipedia
AGH LIve Router	Router (forskningssyfte)	2013-11-02	45 700	Wikipedia
Trustix Secure Linux	Router/Brandvägg	Inaktiv 2007	42 900	Wikipedia
Untangle	Router/Brandvägg	2013	39 500	Wikipedia & Distrowatch
Floppyfw	Router/Brandvägg	2012-04-10	36 800	Wikipedia
Alpine Linux		2013-03-01	32 200	Wikipedia & Distrowatch
Engarde Secure Linux	Router/Brandvägg	Inaktiv 2008	32 300	Wikipedia
Linux Router Project	Router	Inaktiv 2003	28 300	Wikipedia
Bifrost Network Project	Router	2012-04-27	2 510	Wikipedia
Threenix	Router/Brandvägg	2013-02-25	771	Wikipedia

Appendix B - Topologi



Appendix C - Pktgen

```
#! /bin/sh

modprobe pktgen

function pgset() {
    local result

    echo $1 > $PGDEV

    result=`cat $PGDEV | fgrep "Result: OK:"`
    if [ "$result" = "" ]; then
         cat $PGDEV | fgrep Result:
    fi
}

function pg() {
    echo inject > $PGDEV
    cat $PGDEV
}

# Config Start Here -----------------------------------------------

# thread config
#  Each CPU has own thread. Two CPU exammple. We add eth1, eth2
respectivly.

PGDEV=/proc/net/pktgen/kpktgend_0
  echo "Removing all devices"
 pgset "rem_device_all"
  echo "Adding eth1"
 pgset "add_device eth1"
  echo "Setting max_before_softirq 10000"
 #pgset "max_before_softirq 10000"

# We need to remove old config since we dont use this thread. We can
only
# one NIC on one CPU due to affinity reasons.

PGDEV=/proc/net/pktgen/kpktgend_1
  echo "Removing all devices"
 pgset "rem_device_all"

# device config
CLONE_SKB="clone_skb 1000000"

PKT_SIZE="pkt_size 1500"

# COUNT 0 means forever
#COUNT="count 0"
COUNT="count 15000000"

PGDEV=/proc/net/pktgen/eth1
  echo "Configuring $PGDEV"
```

```
 pgset "$COUNT"
 pgset "$CLONE_SKB"
 pgset "$PKT_SIZE"
 pgset "dst 192.168.2.199"
 pgset "dst_mac  00:1b:21:68:40:61"
# Time to run
PGDEV=/proc/net/pktgen/pgctrl

 echo "Running... ctrl^C to stop"
 pgset "start"
 echo "Done"

#     Result     can     be     vieved     in     /proc/net/pktgen/eth1
```

Appendix D - Capinfos

```
#!/bin/bash
touch 1500_co.txt
capinfos 1500_co.cap >> 1500_co.txt
echo "1"
```

Samma rad kopierades för hur många filer som skulle analyseras. Namn och nummer byttes för varje fil.