	VAASAN YLIOPISTO **Seinäjoki**	

Reaaliaikaiset Internet-palvelut ja SIP - Retiisi

Palomuurien ja NATin tuomat haasteet neuvottelujärjestelmien käytössä

Kirjoittajat:	Lassi Ylikojola Vaasan yliopisto Seinäjoki
Kuvaus:	Palomuurien ja nat:in tuomat haasteet neuvottelujärjestelmien käytössä
Tiedostonimi:	retiisi_neuvottelujärjestelmät_ja_verkkotekniikka.pdf
Dokumentin tila:	Julkaistu
Versio:	1.0
Päiväys:	31.03.2006

Sisällysluettelo:

1.Johdanto

Tämä Retiisi –projektin osaraportti käsittelee erilaisten videoneuvottelujärjestelmien vaatimuksia ja ongelmia ip-pohjaisissa verkkoympäristöissä joissa käytetään palomuureja ja nat-osoitekäännöstä. Dokumentti on tarkoitettu ensisijaisesti ip-verkkojen operaattoreiden käyttöön tilanteessa, missä verkossa on käytössä rajoittavia tekijöitä, kuten palomuurit ja nat. Dokumentti jakautuu ongelman kuvaukseen, testijärjestelmien kuvaukseen ja vaatimuksiin, ja mahdollisiin ratkaisumahdollisuuksiin nykytilanteessa. SIP-pohjaisia järjestelmiä ja rtp-proxyn mahdollisuuksia käsitellään erillisessä Retiisi -projektin dokumentissa "Palomuurit ja NAT SIP - palvelun rakentamisessa".

Tämä dokumentti käsittelee neuvottelujärjestelmiä, jotka koostuvat neuvottelujärjestelmän päätelaiteyksiköistä ja niiden välisen monipisteneuvottelun mahdollistavista palvelimista. Oletuksena kaikissa tämän dokumentin neuvottelujärjestelmäkuvauksissa on, että pc-pohjaisissa client/server -tyyppisissä neuvottelujärjestelmissä palvelin sijaitsee julkisessa ip-osoitteessa ja päätelaite nat/palomuurin takana. Samoin h.323 videoneuvottelujärjestelmissä monipisteneuvottelun siltaava mcu-yksikkö sijaitsee julkisessa ip-osoittessa tai dmz-alueella ja päätelaitteet sijaitsevat nat/palomuurin takana. Neuvottelujärjestelmien yleinen käytäntö on, että järjestelmän palvelin/mcu-yksikkö sijaitsee julkisessa ip-avaruudessa.

2. Ongelman kuvaus

Kuinka turvata erilaisten neuvottelujärjestelmien turvallisuus julkisissa verkoissa ja toimivuus verkkoympäristöissä missä on käytössä palomuuri ja nat-osoitekäännöksiä? Neuvottelujärjestelmien moninaisuus(kymmeniä erilaisia) ja niiden eriävät vaatimukset palomuurien osalta ja toisaalta medioiden(video/audio) nat-läpäisykyky asettavat haasteen neuvottelujärjestelmien onnistuneen käytön kannalta. Erilaiset nat-osoitekäännöstyypit ja palomuurit on kuvattu tarkemmin erillisessä Retiisi projektin raportissa "Palomuurit ja nat SIP-palvelun rakentamisessa".

Kaikki neuvottelujärjestelmät eivät tällä hetkellä toimi signaloinnin (signaloinnilla perustetaan, puretaan ja muutetaan neuvottelujärjestelmien audio/video mediasessiot) ja medioiden (audio ja video) osalta nat-osoitekäännöksen läpi. Näitä ovat testijärjestelmistämme h.323-pohjaiset neuvottelujärjestelmät(symmetrinen nat**,** ks.*"Palomuurit ja nat SIP-palvelun rakentamisessa"*), ja Access Grid, minkä mediatyökalut videon ja audion lähettämiseen on alunpitäen kehitetty 1990-luvun puolella verkkoihin, missä on käytössä multicast-reititys. Multicast reititys (ks. Viite 11. Internet protocol multicast) on tapa reitittää samaa tietoa, tässä tapauksessa neuvottelujärjestelmille yhteisesti audion ja videon data verkkokapasiteettia säästävästi. Edellä mainituista neuvottelujärjestelmistä h.323:n osalta ITU tekee työtä h.460.18-19 (ks. 12. Traversal of H.323 signalling/media across Network Address Translators and Firewalls) päästäkseen eroon nat/palomuuriongelmista ja Access Grid osaltaan miettii ratkaisuja nat läpäisyyn tunneloinneista.

Nat-osoitekäännöksen lisäksi neuvottelujärjestelmien ongelmaksi muodostuu palomuuriasetukset. Palomuureissa yläporttialueet (1024-49151,49152-65535) ovat lähes poikkeuksetta

suljettuja. Jokainen testattu järjestelmä käyttää useita yhtäaikaisia yhteyksiä ja Access Gridin tapauksessa oletuksena on multicast-reititys, joskaan se ei ole pakollinen.

Neuvottelujärjestelmät tarvitsevat huomattavasti kaistaa. Esimerkiksi seitsemän pisteen access grid neuvottelu vaatii kaistaa noin 10Mbps jokaiselta neuvottelupisteeltä. Mikäli kyseessä on unicast-reitiys on kaistan tarve palvelimen osalta 70Mbps. Tämän takia yhtenäisen multicast reitityksen levittäminen alueille, missä sitä ei ole saatavilla, vähentäisi kaistantarvetta lähestyttäessä reitityksessä neuvottelujärjestelmien palvelinta. Kaistantarve on kuitenkin sama päätepisteille olkoon kyseessä unicast tai multicast reititys (ks. Viite 11. Internet protocol multicast).

Testiympäristössämme oli Retiisi -projektin käytössä kolme erilaista neuvottelujärjestelmää, joiden ominaisuuksista ja porttivaatimuksista sekä verkkoteknisistä vaatimuksista on seuraavassa luvussa kuvaukset. Järjestelmistä ainoastaan kaupallinen Marratech on tällä hetkellä natläpäisevä.

3. Testijärjestelmien kuvaus, ominaisuudet ja porttivaatimukset

3.1. h.323

Tällä hetkellä käytetyin neuvottelujärjestelmä on h.323. Sen mukaiset videoneuvottelujärjestelmät ovat ainoita järjestelmiä, jotka perustuvat standardiin (ITU). Tällä hetkellä h.323 videoneuvottelujärjestelmät ovat ongelmallisia nat- ja palomuuriympäristöissä.

Yleisiä h.323 järjestelmien käyttämiä portteja:
Katso. viite 14. VCON: Traversing firewalls with video over ip: Issues and solutions:
http://www.vcon.com/pdfdoc/eng/wp/031113.eng.wp.Firewalls_and_Proxy_Servers.pdf

1719 Static UDP Gatekeeper RAS
1720 Static TCP Q.931 (Call Setup)
1024-65535 Dynamic TCP H.245 (Call Parameters)
1024-65535 Dynamic UDP (RTP) Video Data Streams
1024-65535 Dynamic UDP (RTP) Audio Data Streams
1024-65535 Dynamic UDP (RTCP) Control Information

Multicast:kyllä
Unicast:kyllä

Kehityssuuntia:
H.460.18 Traversal of H.323 signalling across Network Address Translators and Firewalls
H.460.19 Traversal of H.323 media across Network Address Translators and Firewalls

3.2. Access Grid

Access Grid on kokoelma ohjelmistotyökaluja, joita käytetään pääasiassa ryhmien väliseen työskentelyyn. Se ei ole standardi, vaan toteutus. Access Grid käyttää medioiden lähettämiseen multicastia, vaikkakin pystyy tekemään medioiden unicast-siltauksen verkkoihin, joihin multicast reititys on estynyt. Unicast siltauksella tarkoitetaan tilannetta missä rtp-media kierrätetään palvelimen kautta. Tällöin palvelimen verkkokaistan tarve kasvaa aina liitettäessä uusi piste neuvottelujärjestelmään, kun taas multicst reitityksessä palvelimen pään verkon kapasiteetin tarve ei muutu (ks. Viite 11. internet protocol multicast). Access Grid on hankalin jär-

jestelmä, kun käytetään palomuureja ja nat-osoitekäännöstä. Sen käyttämät ohjelmat videon ja audion lähettämiseen on suunniteltu multicast-verkkoihin eivätkä ne läpäise nat-osoitekäännöstä.
Yleiset/vaadittavat portit (ks. Viite 1.Alonso Javier Gomez , Access grid port usage):
Pääteohjelmisto VenueClient:
port 8000/TCP (Virtual Venue Server port) on the machine hosting the AG2 venue server. *)
port 8002/TCP (Event port) on the machine hosting the AG2 venue server. *)
port 8004/TCP (Text port) on the machine hosting the AG2 venue server.*)
port 8006/TCP (Data port) on the machine hosting the AG2 venue server.*)
*) portti alue voidaan määritellä itse. Usein portit ovat 9000,9002,9004,9006

Videotyökalu vic:
palvelimella määritelty multicast osoite UDP
palvelimella määritelty multicast osoite +1 UDP
Audiotyökalu rat:
palvelimella määritelty multicast osoite UDP
palvelimella määritelty multicast osoite +1 UDP
lisäksi aina 224.255.222.239 47000 UDP

Mikäli käytetään medioiden siltausta multicast -verkosta unicast -verkkoon:
video:
siltauskoneen osoite/porttinumero
Siltauskoneen osoite/porttinumero +1
Audio:
siltauskoneen osoite/porttinumero
Siltauskoneen osoite/porttinumero +1

vnc tcp 5900
+muita jaettuja ohjelmistoja

Multicast:kyllä
Unicast:kyllä

Kehityssuunta:
Multicast tunnelointi.

3.3. Marratech

Marratech on esimerkkinä suljetusta kaupallisesta tuotteesta. Testatuista järjestelmistä se on ainoa, jonka media (audio ja video) läpäisee nat-tyypit.

Oletuksena UDP portit 52000 to 52999. Porttialue on palvelimella määriteltävissä. Jokainen päätelaite käyttää 12 porttia valitulta alueelta audiolle, videolle ja jaetuille työkaluille. Jaettuja työkaluja ovat mm. Työpöydän jako(vnc), jaetut piirtotyökalu, jaettu dokumentti, jaettu powrpoint...)

Multicast:kyllä
Unicast:kyllä

4. Ratkaisumahdollisuuksista

Uudet tekniikat on aina sovitettava olemassa olevan tekniikan kanssa yhteen. Tämänhetkisessä tilanteessa, missä käytettävien neuvottelujärjestelmien mediat (audio,video) eivät pysty läpäisemään nat-osoitekäännöstä, on näiden järjestelmien osalta turvauduttava tunnelointitekniikoihin, millä ongelma nykytilanteessa voidaan kiertää (ks. Viite 8. Open VPN and the SSL revolution ja 10.IPSec NAT Transparency). Tunnelointitekniikoiden toisena tehtävänä on salata neuvottelujärjestelmissä mahdollisesti liikkuva suojaamaton tieto videon, audion ja jaettujen työkalujen osalta. Mahdollisesti suojaamattomasta neuvottelutilanteessa käytetystä työkalusta voidaan mainita esimerkkinä työpöydän jakoon käytetty vnc(virtual network computing, http://realvnc.com).

Toinen syy tunneloinnin käyttämiseen on palomuuriongelmien välttäminen. Useasti neuvottelujärjestelmien vaatimat portit ovat palomuurilla suljettuja ja on mahdollista, että tarkkaa porttivaatimusta on hyvin vaikea määrittää (porttiavaruus on eri palvelimilla erilainen ja porttiavaruus riippuu siitä, mitä järjestelmän osia käytetään).

Tämän dokumentin liitteenä olevat konfiguraatiotiedostot liittyvät esimerkkitunnelointiin missä nat-osoitekäännöksen takana oleva pääte sillataan tunneloinnin läpi niin, että se ikään kuin olisi osana verkkoa, missä neuvottelujärjestelmän palvelin sijaitsee. Sillattu päätelaite näkee itsensä osana verkkoa ja saa verkon kaikki toiminnot (broadcast, dhcp...) sillatusta verkosta. Lisäksi tunnelin kautta pusketaan myös multicast päätelaitteelle. Päätteen kaikki verkkoliikenne ohjataan tunnelin läpi. Palomuuriasetuksiin voidaan määritellä se liikenne, mikä päätelaitteelle sallitaan.

Käytetty tunnelointiohjelma on vapaan lähdekoodin openvpn. Ks. 8.Open VPN and the SSL revolution: http://www.sans.org/rr/whitepapers/vpns/1459.php Openvpn on ssl-pohjainen (ks. Viite16. SSL 3.0 specification) 'aito' tunnelointi -ratkaisu, siis ikään kuin Ipsec . Viite ... Se vaatii ohjelman asentamista kumpaankin tunnelin päähän. Se ei siis ole web-sovellus välityspalvelin, mitkä turvaavat web sovellusten liikennettä selaimelta palvelimelle. Openvpn tunneloinnilla voidaan tunneloida/sillata minkä tahansa aliverkko tai virtuaalinen ethernet adapterin käyttäen yhtä udp tai tcp porttia (defaul udp 1194 ks. viite 9. OpenVPN 2.0 howto).Se pystyy tunneloimaan nat-osoitekännöksen läpi ja käyttämään OpnSSL-kirjaston autentikointi- ja kryptausmenetelmiä liikenteen turvaamiseen.

Lopuksi vielä lyhyesti tunneloinnin eduista neuvottelujärjestelmien osalta nykyisessä tilanteessa:
1. Nat läpäisy: Kierrämme ongelman olemalla ikään kuin samassa verkossa.
2. Openvpn-tekniikalla yhden portin palomuuriratkaisu(default 1194): helpottaa palomuurin konfiguroinnissa.
3. Multicastin tunnelointi: Voimme saada yksittäisen päätelaitteen tai aliverkon multicastin piiriin, mikäli verkkojen välinen reititys ei sitä tee.
4. Autentikointi: tunnelointitekniikoiden tarjoamat sertifikaatti ja salasana autentikoinnit salaus: Sellaisten neuvottelujärjestelmien osien liikenteen salaus, mitkä evät itse sitä tee.

Haitoista keskeisin on tietoturvariski: liikenne turvaamattomasta verkosta yksityiseen verk-

koon (palomuurisäännöt).

Liitteenä olevat client.conf server.conf ja bridge.sh ovat openvpn tunnelloinnin parametrit. Bridge.sh luo siltauksen ja virtuaaliset ip-osoitteet verkkoon mihin pääteyhteys luodaan. Server.conf on openvpn-palvelimen käynnistyskonfiguraatio
client.conf on päätelaitteen käynnistyskonfiguraatio.

1. luodaan siltaus bridge. sh tiedostolla neljälle virtuaaliselle verkkoadapterille
2. käynnistetään openvpn-palvelin openvpn –config server.conf
3. käynnisttetään openvpn päätelaitteella openvpn –config client.conf

5. Viitteet:

1.Alonso Javier Gomez , Access grid port usage:
http://www.accessgrid.org/agdp/guide/ports/1.03/index.html
2.Marratech client firewall guide:
http://www.marratech.com/userman/client/app_firewall_guide.html
3.Marratech manager firewall guide:
http://www.marratech.com/userman/manager/app_firewalls.html
4.Traversing Firewalls with Video over IP: Issues and Solutions, VCON, August 2003
http://www.h323forum.org/papers/vcon_031113.eng.wp.Firewalls_and_Proxy_Servers.pdf
5.H. Schulzrinne:RFC 3550 - RTP: A Transport Protocol for Real-Time Applications
http://www.packetizer.com/rfc/rfc.cgi?num=3550
6.Baruch Sterman , Schwartz David: Nat traversal in SIP
http://corp.deltathree.com/technology/nattraversalinsip.pdf
7.J. Rosenberg:Interactive Connectivity Establishment (ICE): A Methodology for Network Address Translator (NAT) Traversal for the Session Initiation Protocol (SIP)
http://www.jdrosen.net/papers/draft-rosenberg-sipping-ice-00.html
8.Open VPN and the SSL revolution:
http://www.sans.org/rr/whitepapers/vpns/1459.php
9.OpenVPN 2.0 howto:
http://openvpn.net/howto.html
10.IPSec NAT Transparency:
http://www.cisco.com/en/US/products/ps6350/products_configuration_guide_chapter09186a0080455c72.html#wp1027186
11.Internet protocol multicast:
http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/ipmulti.htm#xtocid24
12. Traversal of H.323 signaling/media across Network Address Translators and Firewalls
http://www.packetizer.com/voip/h323/standards.html
13. h.323 firewall/nat traversal
http://www.h323forum.org/papers/301005_Firewall_NAT_Traversal_White_Paper.pdf
14. VCON: Traversing firewalls with video over ip: Issues and solutions:
http://www.vcon.com/pdfdoc/eng/wp/031113.eng.wp.Firewalls_and_Proxy_Servers.pdf
15. Ip ports and protocols used by h.323 devices:
http://www.teamsolutions.co.uk/tsfirewall.html
16. SSL 3.0 specification:

http://wp.netscape.com/eng/ssl3/

6. Lyhenteet

DNS	Domain Name System
ENUM	Enumeration
IETF	Internet Engineering Task Force
IP	Internet Protocol
NAPTR	Naming authority pointer
PSTN	Public Service Telephone Network
RADIUS	Remote Access Dial In User Service
RFC	Request For Comments
SCCP	Simple Client Control Protocol
SER	Sip Express Router
SIP	Session Initiation Protocol
SMS	Short Message System
SRV	Service resource
SCTP	Stream Control Transmission Protocol
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UDP	User Datagram Protocol
VOIP	Voice Over IP
VPN	Virtual private netwok
ITU	International telecommunication union

Liite 1. Tunnelointi / siltaus -esimerkki (openvpn)

Bridge.start:

```
#!/bin/bash

###################################
# Set up Ethernet bridge on Linux
# Requires: bridge-utils
###################################

# Define Bridge Interface
br="br0"

# Define list of TAP interfaces to be bridged,
# for example tap="tap0 tap1 tap2".
###Luodaan neljä virtuaalista adapteria siltauksen käyttöön
tap="tap0 tap1 tap2 tap3"

# Define physical ethernet interface to be bridged
```

```
# with TAP interface(s) above.
###Määritetään käytettävä fyysinen verkkokortti, sen ip maski ja verkon broadcast osoite
eth="eth0"
eth_ip="xxx.xxx.xxx.216"
eth_netmask="255.255.255.224"
eth_broadcast="xxx.xxx.xxx.223"

for t in $tap; do
   openvpn --mktun --dev $t
done

brctl addbr $br
brctl addif $br $eth

for t in $tap; do
   brctl addif $br $t
done

for t in $tap; do
   ifconfig $t 0.0.0.0 promisc up
done
ifconfig $eth 0.0.0.0 promisc up

ifconfig $br $eth_ip netmask $eth_netmask broadcast $eth_broadcast


# Flush and remove all chains.
iptables -F
iptables -X
iptables -t nat -F
iptables -t nat -X

# Set Default Policy ACCEPT
iptables -P FORWARD ACCEPT
iptables -P INPUT ACCEPT
iptables -P OUTPUT ACCEPT

### sallitaaan testivaiheessa kaikki liikenne virtuaalisille adaptereille
iptables -A INPUT -i tap0 -j ACCEPT
iptables -A INPUT -i tap1 -j ACCEPT
iptables -A INPUT -i tap2 -j ACCEPT
iptables -A INPUT -i tap3 -j ACCEPT
iptables -A INPUT -i br0 -j ACCEPT
iptables -A FORWARD -i br0 -j ACCEPT

echo 1 >/proc/sys/net/ipv4/ip_forward

###määritellään aliverkon gateway osoite
```

route add default gw xxx.xxx.xxx.193

Liite 2. Tunnelointi palvelin - esimerkki (openvpn)

Server.conf:

```
#########################################################
# Sample OpenVPN 2.0 config file for            #
# multi-client server.                          #
#                                               #
# This file is for the server side              #
# of a many-clients <-> one-server              #
# OpenVPN configuration.                        #
#                                               #
# OpenVPN also supports                         #
# single-machine <-> single-machine             #
# configurations (See the Examples page         #
# on the web site for more info).               #
#                                               #
# This config should work on Windows            #
# or Linux/BSD systems.  Remember on            #
# Windows to quote pathnames and use            #
# double backslashes, e.g.:                     #
# "C:\\Program Files\\OpenVPN\\config\\foo.key" #
#                                               #
# Comments are preceded with '#' or ';'         #
#########################################################

# Which local IP address should OpenVPN
# listen on? (optional)
;local

# Which TCP/UDP port should OpenVPN listen on?
# If you want to run multiple OpenVPN instances
# on the same machine, use a different port
# number for each one.  You will need to
# open up this port on your firewall.
### määritellään palvelimen käyttämä portti
port 1194

# TCP or UDP server?
;proto tcp
###määritellään palvelimen käyttämä protokolla (UDP/TCP)
proto udp

# "dev tun" will create a routed IP tunnel,
# "dev tap" will create an ethernet tunnel.
```

```
# Use "dev tap" if you are ethernet bridging.
# If you want to control access policies
# over the VPN, you must create firewall
# rules for the the TUN/TAP interface.
# On non-Windows systems, you can give
# an explicit unit number, such as tun0.
# On Windows, use "dev-node" for this.
# On most systems, the VPN will not function
# unless you partially or fully disable
# the firewall for the TUN/TAP interface.
;dev tap
###Koska ethernet tunnelin, käytämme dev tap-määritystä
dev tap0

# Windows needs the TAP-Win32 adapter name
# from the Network Connections panel if you
# have more than one.  On XP SP2 or higher,
# you may need to selectively disable the
# Windows firewall for the TAP adapter.
# Non-Windows systems usually don't need this.
;dev-node lan4

# SSL/TLS root certificate (ca), certificate
# (cert), and private key (key).  Each client
# and the server must have their own cert and
# key file.  The server and all clients will
# use the same ca file.
#
# See the "easy-rsa" directory for a series
# of scripts for generating RSA certificates
# and private keys.  Remember to use
# a unique Common Name for the server
# and each of the client certificates.
#
# Any X509 key management system can be used.
# OpenVPN can also use a PKCS #12 formatted key file
# (see "pkcs12" directive in man page).
### Käytämme sertifikaaativarmennusta. Nämä on generoitava
ca /etc/openvpn/ca.crt
cert /etc/openvpn/server.crt
key /etc/openvpn/server.key  # This file should be kept secret

# Diffie hellman parameters.
# Generate your own with:
#   openssl dhparam -out dh1024.pem 1024
# Substitute 2048 for 1024 if you are using
# 2048 bit keys.
###DH sertifikaatti. Tämä on generoitava
```

```
dh /etc/openvpn/dh1024.pem

# Configure server mode and supply a VPN subnet
# for OpenVPN to draw client addresses from.
# The server will take 10.8.0.1 for itself,
# the rest will be made available to clients.
# Each client will be able to reach the server
# on 10.8.0.1. Comment this line out if you are
# ethernet bridging. See the man page for more info.
;server 10.8.0.0 255.255.255.0
;server-bridge 192.98.83.216 255.255.225.224 192.98.83.219 192.98.83.220

#push "dhcp-option DNS 192.98.80.1" # push DNS entries to openvpn client
### push käskyillä 'tyäönnamme vpn-clientille tietoja reitityksestä, DNS-plavelimesta ja
muusta
push "route-gateway xxx.xxx.xxx.216" # push default gateway
push "redirect-gateway def1"
#push "route-gateway xxx.xxx.xxx.193"
push "dhcp-option DNS xxx.yyy.yyy.1"


# Maintain a record of client <-> virtual IP address
# associations in this file.  If OpenVPN goes down or
# is restarted, reconnecting clients can be assigned
# the same virtual IP address from the pool that was
# previously assigned.
#ifconfig-pool-persist ipp.txt

# Configure server mode for ethernet bridging.
# You must first use your OS's bridging capability
# to bridge the TAP interface with the ethernet
# NIC interface.  Then you must manually set the
# IP/netmask on the bridge interface, here we
# assume 10.8.0.4/255.255.255.0.  Finally we
# must set aside an IP range in this subnet
# (start=10.8.0.50 end=10.8.0.100) to allocate
# to connecting clients.  Leave this line commented
# out unless you are ethernet bridging.
;server-bridge 10.8.0.4 255.255.255.0 10.8.0.50 10.8.0.100
###siltausmääritys. Vpn-palvelimen osoite, aliverkon peite, haluttu ip-alue aliverkosta
server-bridge xxx.xxx.xxx.216 255.255.225.224 xxx.xxx.xxx.217 xxx.xxx.xxx.219

# Push routes to the client to allow it
# to reach other private subnets behind
# the server.  Remember that these
# private subnets will also need
# to know to route the OpenVPN client
# address pool (10.8.0.0/255.255.255.0)
```

```
# back to the OpenVPN server.
### Haluamme päästä käsiksi vpn-palvelimen verkkoon:
push "route 192.98.83.192 255.255.255.224"

# To assign specific IP addresses to specific
# clients or if a connecting client has a private
# subnet behind it that should also have VPN access,
# use the subdirectory "ccd" for client-specific
# configuration files (see man page for more info).

# EXAMPLE: Suppose the client
# having the certificate common name "Thelonious"
# also has a small subnet behind his connecting
# machine, such as 192.168.40.128/255.255.255.248.
# First, uncomment out these lines:
;client-config-dir ccd
;route 192.168.40.128 255.255.255.248
# Then create a file ccd/Thelonious with this line:
#   iroute 192.168.40.128 255.255.255.248
# This will allow Thelonious' private subnet to
# access the VPN.  This example will only work
# if you are routing, not bridging, i.e. you are
# using "dev tun" and "server" directives.

# EXAMPLE: Suppose you want to give
# Thelonious a fixed VPN IP address of 10.9.0.1.
# First uncomment out these lines:
;client-config-dir ccd
;route 10.9.0.0 255.255.255.252
# Then add this line to ccd/Thelonious:
#   ifconfig-push 10.9.0.1 10.9.0.2

# Suppose that you want to enable different
# firewall access policies for different groups
# of clients.  There are two methods:
# (1) Run multiple OpenVPN daemons, one for each
#     group, and firewall the TUN/TAP interface
#     for each group/daemon appropriately.
# (2) (Advanced) Create a script to dynamically
#     modify the firewall in response to access
#     from different clients.  See man
#     page for more info on learn-address script.
;learn-address ./script

# If enabled, this directive will configure
# all clients to redirect their default
# network gateway through the VPN, causing
# all IP traffic such as web browsing and
```

```
# and DNS lookups to go through the VPN
# (The OpenVPN server machine may need to NAT
# the TUN/TAP interface to the internet in
# order for this to work properly).
# CAVEAT: May break client's network config if
# client's local DHCP server packets get routed
# through the tunnel.  Solution: make sure
# client's local DHCP server is reachable via
# a more specific route than the default route
# of 0.0.0.0/0.0.0.0.
;push "redirect-gateway"

# Certain Windows-specific network settings
# can be pushed to clients, such as DNS
# or WINS server addresses.  CAVEAT:
# http://openvpn.net/faq.html#dhcpcaveats
;push "dhcp-option DNS 10.8.0.1"
;push "dhcp-option WINS 10.8.0.1"

# Uncomment this directive to allow different
# clients to be able to "see" each other.
# By default, clients will only see the server.
# To force clients to only see the server, you
# will also need to appropriately firewall the
# server's TUN/TAP interface.
### päätteiden näkyvyys toisille:
client-to-client

# Uncomment this directive if multiple clients
# might connect with the same certificate/key
# files or common names.  This is recommended
# only for testing purposes.  For production use,
# each client should have its own certificate/key
# pair.
#
# IF YOU HAVE NOT GENERATED INDIVIDUAL
# CERTIFICATE/KEY PAIRS FOR EACH CLIENT,
# EACH HAVING ITS OWN UNIQUE "COMMON NAME",
# UNCOMMENT THIS LINE OUT.
;duplicate-cn

# The keepalive directive causes ping-like
# messages to be sent back and forth over
# the link so that each side knows when
# the other side has gone down.
# Ping every 10 seconds, assume that remote
# peer is down if no ping received during
# a 120 second time period.
```

```
### Generoi liikennettä
keepalive 10 120

# For extra security beyond that provided
# by SSL/TLS, create an "HMAC firewall"
# to help block DoS attacks and UDP port flooding.
#
# Generate with:
#   openvpn --genkey --secret ta.key
#
# The server and each client must have
# a copy of this key.
# The second parameter should be '0'
# on the server and '1' on the clients.
;tls-auth ta.key 0 # This file is secret

# Select a cryptographic cipher.
# This config item must be copied to
# the client config file as well.
;cipher BF-CBC        # Blowfish (default)
;cipher AES-128-CBC   # AES
;cipher DES-EDE3-CBC  # Triple-DES

# Enable compression on the VPN link.
# If you enable it here, you must also
# enable it in the client config file.
comp-lzo

# The maximum number of concurrently connected
# clients we want to allow.
;max-clients 100
# It's a good idea to reduce the OpenVPN
# daemon's privileges after initialization.
#
# You can uncomment this out on
# non-Windows systems.
;user nobody
;group nobody

# The persist options will try to avoid
# accessing certain resources on restart
# that may no longer be accessible because
# of the privilege downgrade.
persist-key
persist-tun
# Output a short status file showing
# current connections, truncated
# and rewritten every minute.
```

```
status openvpn-status.log

# By default, log messages will go to the syslog (or
# on Windows, if running as a service, they will go to
# the "\Program Files\OpenVPN\log" directory).
# Use log or log-append to override this default.
# "log" will truncate the log file on OpenVPN startup,
# while "log-append" will append to it.  Use one
# or the other (but not both).
;log         openvpn.log
;log-append  openvpn.log

# Set the appropriate level of log
# file verbosity.
#
# 0 is silent, except for fatal errors
# 4 is reasonable for general usage
# 5 and 6 can help to debug connection problems
# 9 is extremely verbose
verb 4

# Silence repeating messages.  At most 20
# sequential messages of the same message
# category will be output to the log.
;mute 20
```

Liite 3. Tunnelointi pääte -esimerkki (openvpn)

Client.conf:

```
#################################################
# Sample client-side OpenVPN 2.0 config file #
# for connecting to multi-client server.     #
#                                            #
# This configuration can be used by multiple #
# clients, however each client should have   #
# its own cert and key files.                #
#                                            #
# On Windows, you might want to rename this  #
# file so it has a .ovpn extension           #
#################################################

# Specify that we are a client and that we
# will be pulling certain config file directives
# from the server.
client
```

```
# Use the same setting as you are using on
# the server.
# On most systems, the VPN will not function
# unless you partially or fully disable
# the firewall for the TUN/TAP interface.
###Tap koska kysymyksessä siltaus
dev tap

# Windows needs the TAP-Win32 adapter name
# from the Network Connections panel
# if you have more than one.  On XP SP2,
# you may need to disable the firewall
# for the TAP adapter.
;dev-node MyTap

# Are we connecting to a TCP or
# UDP server?  Use the same setting as
# on the server.
###Käytetty protokolla
proto udp

# The hostname/IP and port of the server.
# You can have multiple remote entries
# to load balance between the servers.
###Palvelimen osoite ja porttimääritys
Remote xxx.xxx.xxx.216 1194
# Choose a random host from the remote
# list for load-balancing.  Otherwise
# try hosts in the order specified.
;remote-random

# Keep trying indefinitely to resolve the
# host name of the OpenVPN server.  Very useful
# on machines which are not permanently connected
# to the internet such as laptops.
resolv-retry infinite

# Most clients don't need to bind to
# a specific local port number.
nobind

# Downgrade privileges after initialization (non-Windows only)
;user nobody
;group nobody

# Try to preserve some state across restarts.
persist-key
persist-tun
```

```
# If you are connecting through an
# HTTP proxy to reach the actual OpenVPN
# server, put the proxy server/IP and
# port number here.  See the man page
# if your proxy server requires
# authentication.
;http-proxy-retry # retry on connection failures
;http-proxy [proxy server] [proxy port #]

# Wireless networks often produce a lot
# of duplicate packets.  Set this flag
# to silence duplicate packet warnings.
;mute-replay-warnings

# SSL/TLS parms.
# See the server config file for more
# description.  It's best to use
# a separate .crt/.key file pair
# for each client.  A single ca
# file can be used for all clients.
###Serytifikaativarmennus. Nämä on generoitava
ca ca.crt
cert client4.crt
key client4.key

# Verify server certificate by checking
# that the certicate has the nsCertType
# field set to "server".  This is an
# important precaution to protect against
# a potential attack discussed here:
#  http://openvpn.net/howto.html#mitm
#
# To use this feature, you will need to generate
# your server certificates with the nsCertType
# field set to "server".  The build-key-server
# script in the easy-rsa folder will do this.
;ns-cert-type server

# If a tls-auth key is used on the server
# then every client must also have the key.
;tls-auth ta.key 1

# Select a cryptographic cipher.
# If the cipher option is used on the server
# then you must also specify it here.
;cipher x
```

```
# Enable compression on the VPN link.
# Don't enable this unless it is also
# enabled in the server config file.
comp-lzo

# Set log file verbosity.
verb 3
```

```
# Silence repeating messages
;mute 20
```

Liite 4. Testilaitteet / ohjelmistot

h.323:
Neljän pisteen ip-silta(mcu) tandberg 880 classic, h.323 ohjelmistopohjainen pääte gnome-meeting-1.2.2-1.FC4 (fedora core 4)
Marratech:
Marratech server 3.2.1 build 293(linux opensuse 10.0), Marratech client 5.1 build 861(Windows XP, linux)
Access grid:
Palvelin VenueServer 2.4, VenueClient 2.4 (palvelin ja päätepisteet linux-pohjaisia fedora core 4)
Tunnelointi:
Openvpn 2.0.5-3.fc4 (siltaus, palvelin ja pääte fedora core 4)
Nat:
iptables-1.3.0-2 (Fedora core 4)
smc barricade smc7004AABR