

---

## **EuQoS MCAST Installation Manual**

*Editors* : *Marc Brogle, Dragan Milic (UBern)*

*Date* : *06.05.08*

*No. of pages* : *28*

*Synopsis* :

*Contributions* :

---

## Executive Summary

This document presents the steps to install and configure the EuQoS MCAST module.

## Revision History

Revision	Date	History
V1.0.1	06/02/2008	Updated it with info about version 4.0.4.
V1.0	12/12/2007	Added QCM configuration description and parts about Linux/Mac OS X configuration, rearranged the sections and subsections
V0.12	24/09/2007	Added Mac OS X install instructions
V0.11	15/08/2007	Updated for new version of mcast v4.0.1 (which fixed a DHT issue) and also added text about 3s delay of test client.
V0.1.beta	06/08/2007	First draft of the installation manual covering only the Windows platform.

## Table of Contents

<b>REVISION HISTORY .....</b>	<b>3</b>
<b>TABLE OF CONTENTS .....</b>	<b>4</b>
<b>1 INTRODUCTION.....</b>	<b>5</b>
<b>2 GENERAL HARDWARE AND SOFTWARE REQUIREMENTS .....</b>	<b>6</b>
<b>3 DEPLOYMENT PACKAGES .....</b>	<b>7</b>
<b>4 INSTALLATION AND CONFIGURATION OF TAP .....</b>	<b>8</b>
4.1 INSTALLATION OF TAP UNDER WINDOWS.....	8
4.2 CONFIGURING OF TAP UNDER WINDOWS.....	13
4.3 INSTALLATION OF TAP UNDER MAC OS X .....	14
4.4 CONFIGURING OF TAP UNDER MAC OS X .....	20
4.5 INSTALLATION OF TAP UNDER LINUX.....	20
4.6 CONFIGURING OF TAP UNDER LINUX.....	20
<b>5 INSTALLING AND CONFIGURING MCAST .....</b>	<b>21</b>
5.1 INSTALLING MCAST.....	21
5.2 CONFIGURING MCAST.....	21
<b>6 STARTING AND TESTING MCAST INSTALLATION .....</b>	<b>22</b>
6.1 STARTING MCAST INSTALLATION UNDER LINUX OR MAC OS X.....	22
6.2 STARTING MCAST INSTALLATION UNDER WINDOWS .....	23
6.3 TESTING MCAST INSTALLATION .....	24
<b>7 EXAMPLE CODE FOR USING QOS-ENABLED MCAST (DUMMYCLIENT) .....</b>	<b>25</b>
<b>8 TROUBLESHOOTING .....</b>	<b>27</b>
8.1 MCAST IS WORKING BUT THE APPLICATION DOES NOT RECEIVE ANY IP MULTICAST PACKETS.....	27
8.2 PROBLEMS WITH VLC RUNNING UNDER WINDOWS.....	27
8.3 PROBLEMS WHEN STARTING MCAST UNDER WINDOWS .....	28

---

## 1 Introduction

This document presents the set of steps required to install and configure the EuQoS MCAST Software module and the required helper tools / drivers.

This manual has been built and tested in the following way:

1. The EuQoS MCAST Software module system has been validated in UBern testbed.
2. EuQoS MCAST has been tested with the EuQoS P#4 system.
3. LAAS and TID used this installation manual and the MCAST packages available at the FTP server to successfully install the Software.

This manual shows the steps required to install and configure the EuQoS MCAST components under Windows, Linux and Mac OS X.

## 2 General Hardware and Software Requirements

### Basic Hardware requirements

There are no special Hardware requirements for MCAST, a standard PC with at least 512 MB of RAM should be sufficient to install and use MCAST.

### Basic Software requirements

- Sun JDK1.5
- Ant 1.5.x (only if you need to compile MCAST, which is not required for the installation)
- Operating System:
  - Windows 2000 or Windows XP but NOT yet Windows Vista!
  - Mac OS X Tiger (10.4.x) or Leopard (10.5.x)
  - Linux with a kernel 2.4 or newer with enabled TunTap functionality
- TAP interface (using OpenVPN installer for Windows, the tuntap driver for Mac OS X, or the since kernel 2.2+ in Linux included tuntap driver for Linux) installed and configured (described in this manual)
- An IP Multicast application (like “DummyClient”, VLC or mchat) for testing

The Sun JDK 1.5 has to be installed but has not to be made available in the \$PATH (%PATH% for Windows) environment variable. The path to the JDK 1.5 can be set explicitly just for starting the MCAST module. This allows having a JDK 1.4.2 installation being used as the default JVM for all other (and especially EuQoS-related) applications.

If problems arise during the installation of two different JDKs / JVMs, especially concerning the default JVM being used by the system, please contact Marc Brogle ([brogle@iam.unibe.ch](mailto:brogle@iam.unibe.ch)) and Dragan Milic ([milic@iam.unibe.ch](mailto:milic@iam.unibe.ch)) for further support.

The installation of the Sun JDK 1.5 and Ant 1.5.x are not described in this manual but instead their own specific installation manuals can be used directly.

### 3 Deployment packages

The MCAST deployment and additional utility / driver packages are available at the EuQoS FTP server. This manual is referred to the SW packages available in: **/home/euqos/prototype\_4/mcast/**

In this folder you can find the following files:

- mcast\_v4.0.4.zip → EuQoS MCAST package for Windows / Linux / Mac OS X using EuQoS prototype P#4
- EuQoS\_MCAST\_Installation\_Manual\_v1.0.1.pdf → this installation manual
- openvpn-2.0.9-install.exe → Windows OpenVPN installer (contains TAP interface for Windows)
- tuntap\_tiger\_20060513.tar.gz → Mac OS X 10.4 tuntap driver (universal binary)
- tuntap\_leopard\_20071114.tar.gz → Mac OS X 10.5 tuntap driver (universal binary)
- mchat.java → a simple java IP Multicast chat application
- jdk-1\_5\_0\_12-windows-i586-p.exe → JDK 1.5 for Windows installer
- vlc-0.8.6c-win32.exe → VLC (Video Lan Client) for Windows
- vlc-0.8.6c-mac.dmg → VLC (Video Lan Client) for Mac OS X (universal binary)
- vlc-0.8.6c-win32.exe → VLC (Video Lan Client) for Windows

## 4 Installation and configuration of TAP

### 4.1 Installation of TAP under Windows

In the following subsections we describe how to install and configure the TAP driver (which is required for MCAST) for the Windows platform.

Please, **follow the steps** described in this installation manual.

1. Launch the OpenVPN (openvpn-2.0.9-install.exe) installer downloaded from the EuQoS FTP server.



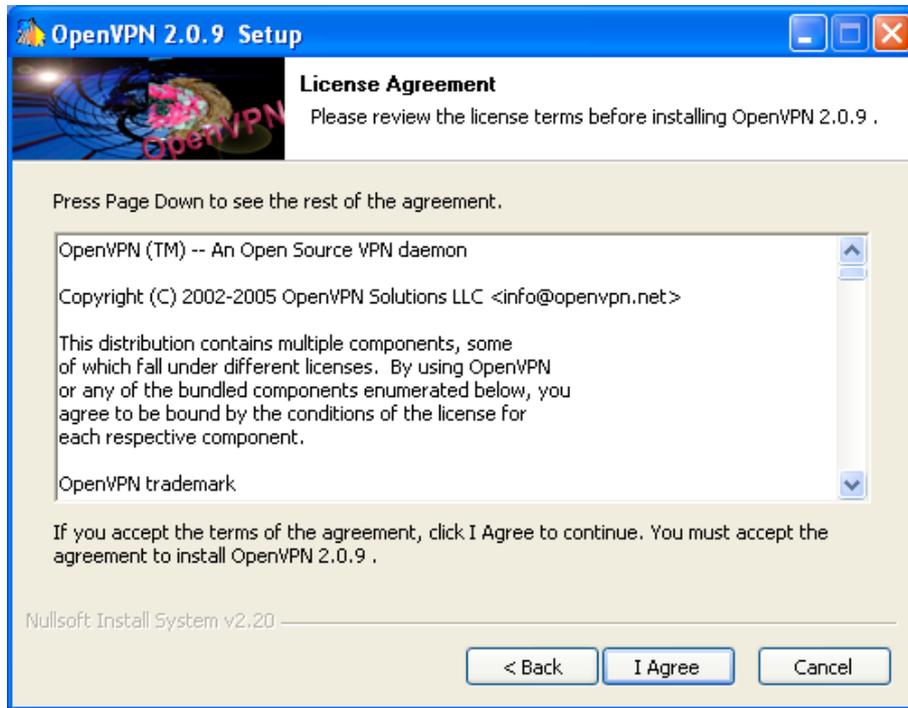
*Figure 4-1: OpenVPN 2.0.9 installation file*

2. In the welcome screen click “Next >” to continue the installation.



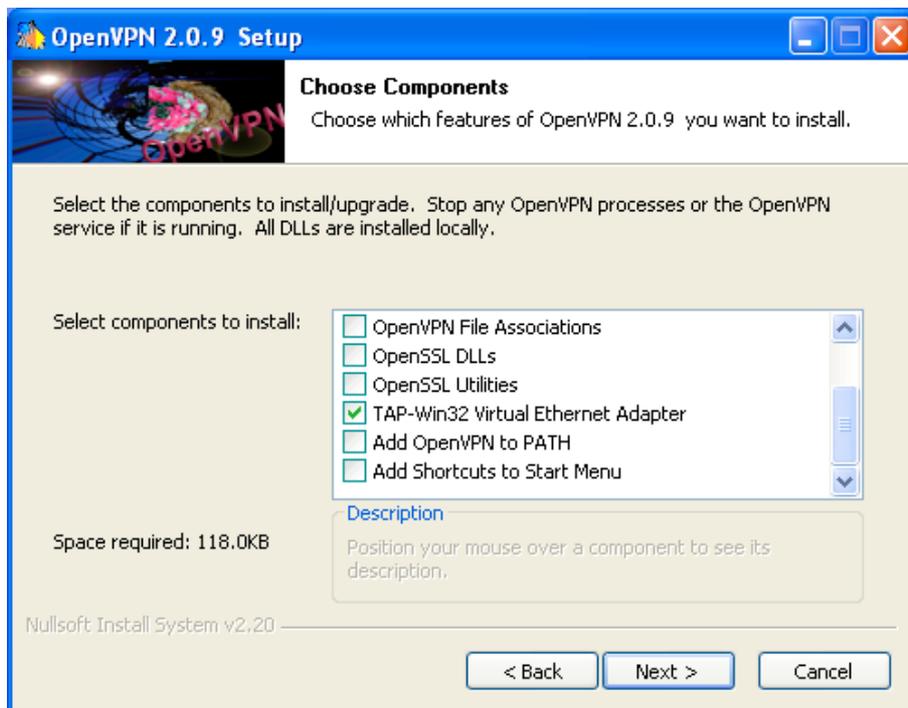
*Figure 4-2: OpenVPN installer welcome screen*

3. Accept the license agreement by clicking “I agree”.



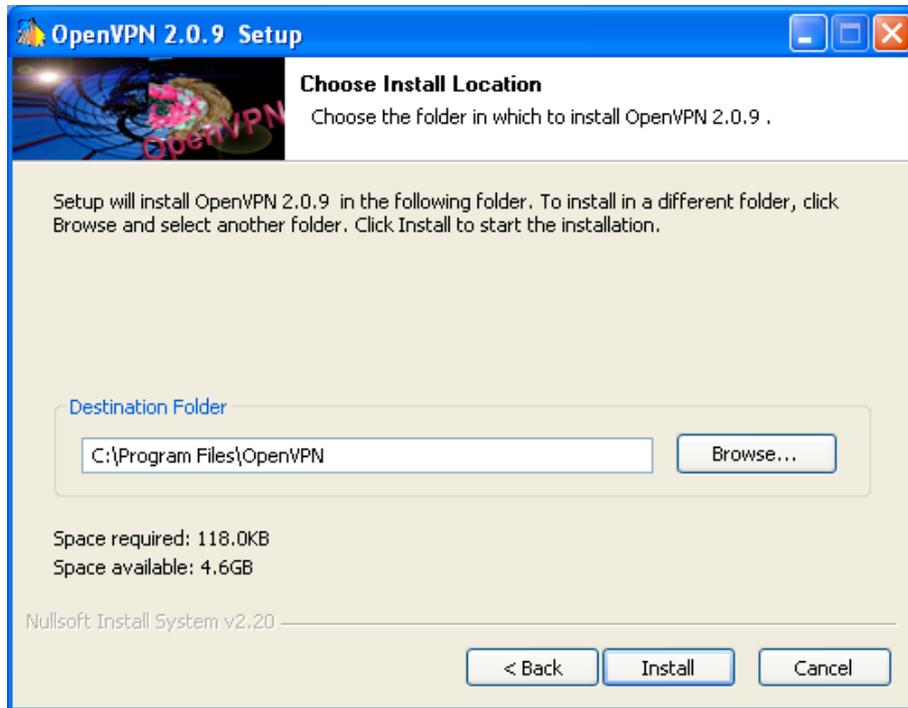
**Figure 4-3: License agreement of OpenVPN**

4. Select only “TAP-Win32 Virtual Ethernet Adapter” and unselect all other options.



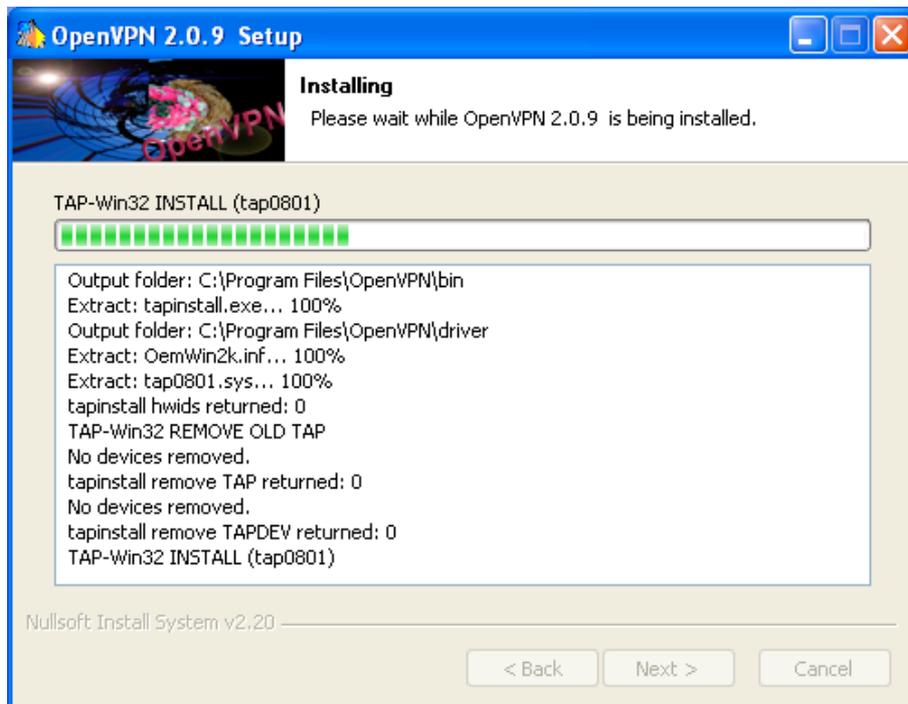
**Figure 4-4: Selection of installation components**

5. Select the installation path and click on “Install”.



*Figure 4-5: Selection of installation path*

6. The installation will start to copy the drivers.



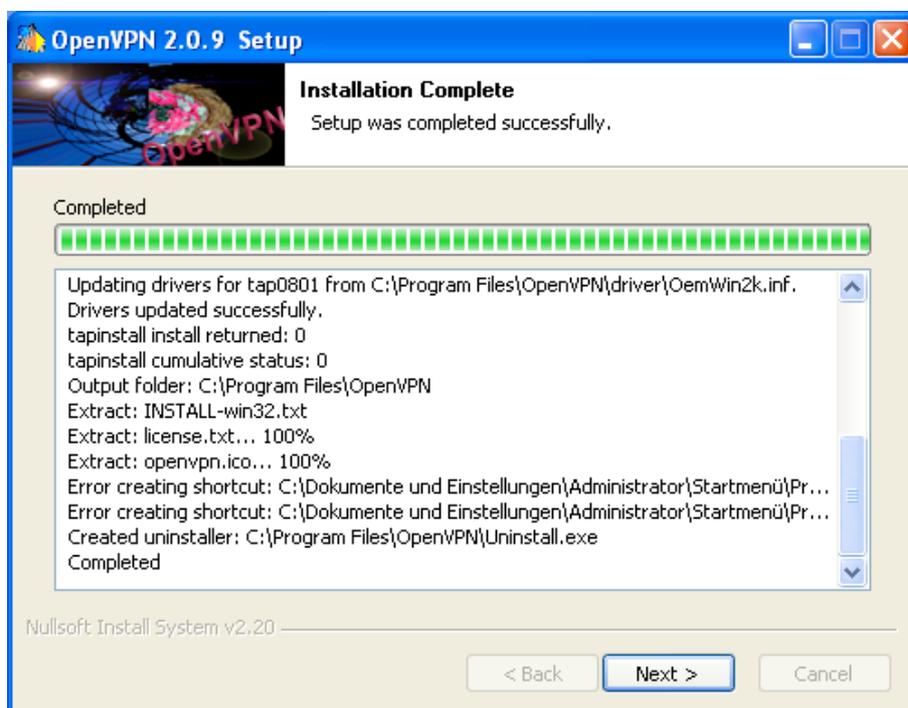
*Figure 4-6: Installation of driver files*

7. Confirm installation of the “not validated” driver by clicking on “Continue Installation”.



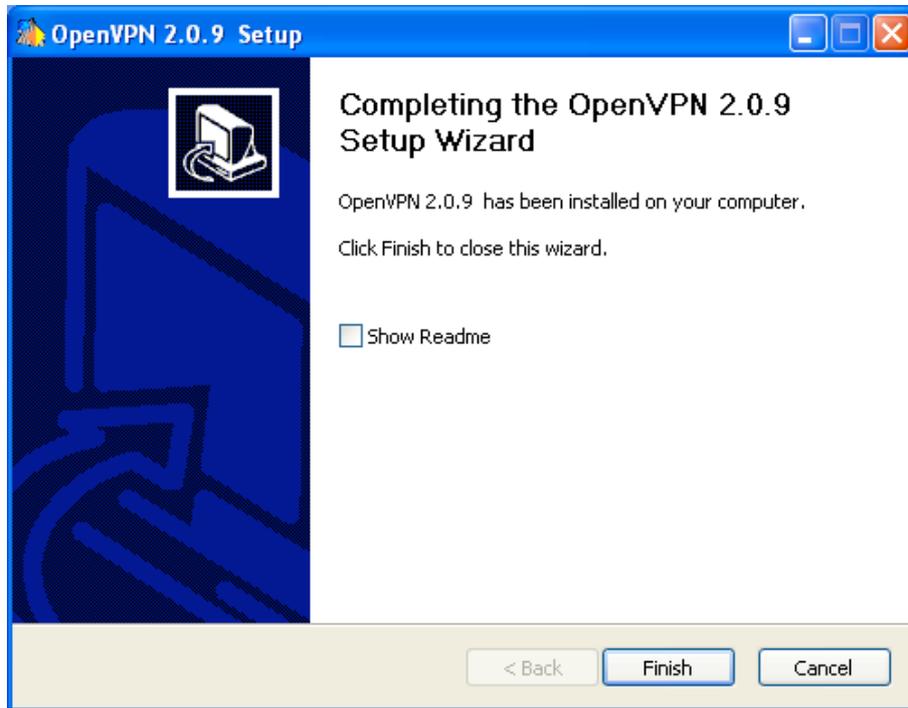
*Figure 4-7: Confirmation request for installing driver*

8. Wait until the installation is complete, then click “Next >”.



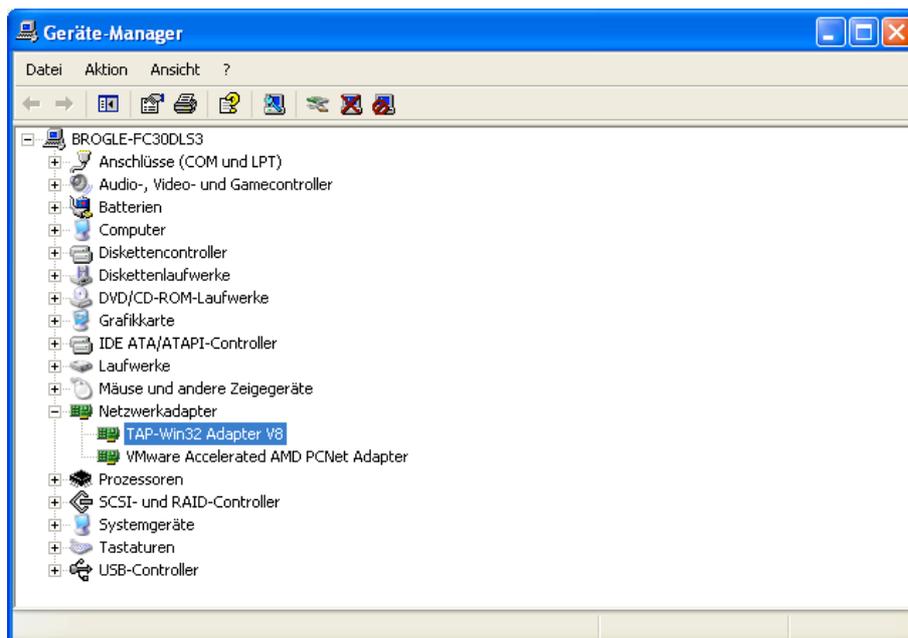
*Figure 4-8: Confirmation of copied files*

9. Finish the installation by clicking on “Finish”.



*Figure 4-9: Finishing the installation*

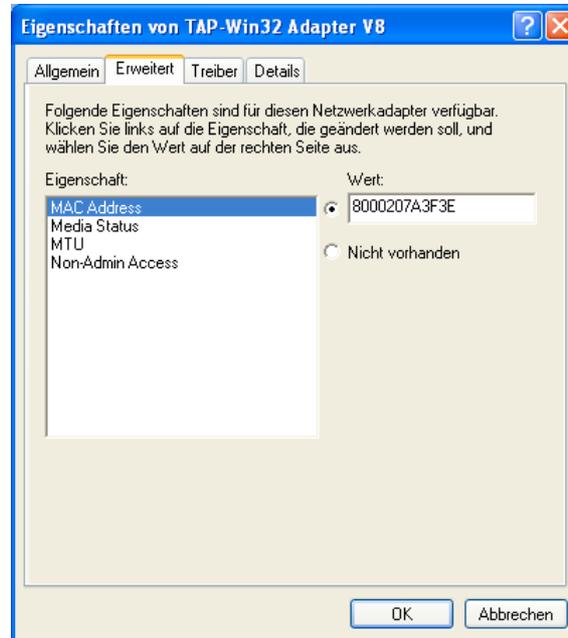
10. After the TAP driver has been successfully installed via the OpenVPN installer the new virtual network interface has to be configured (see Section 4.2).



*Figure 4-10: Successful installation of the TAP interface*

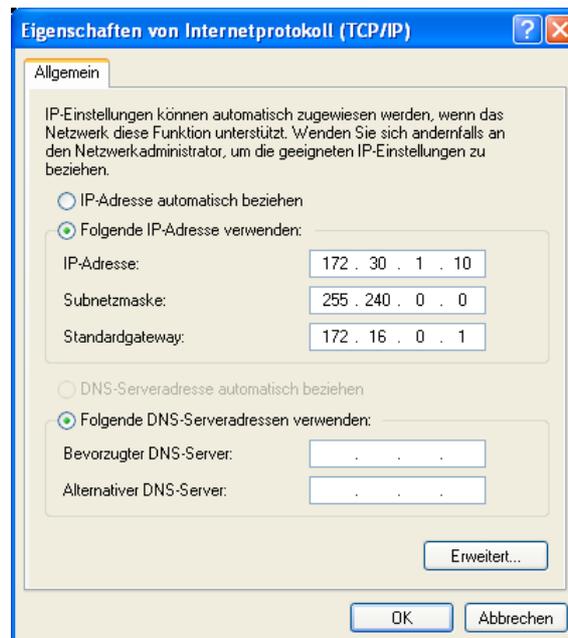
## 4.2 Configuring of TAP under Windows

1. Set a valid (not yet used) MAC address for the TAP interface.



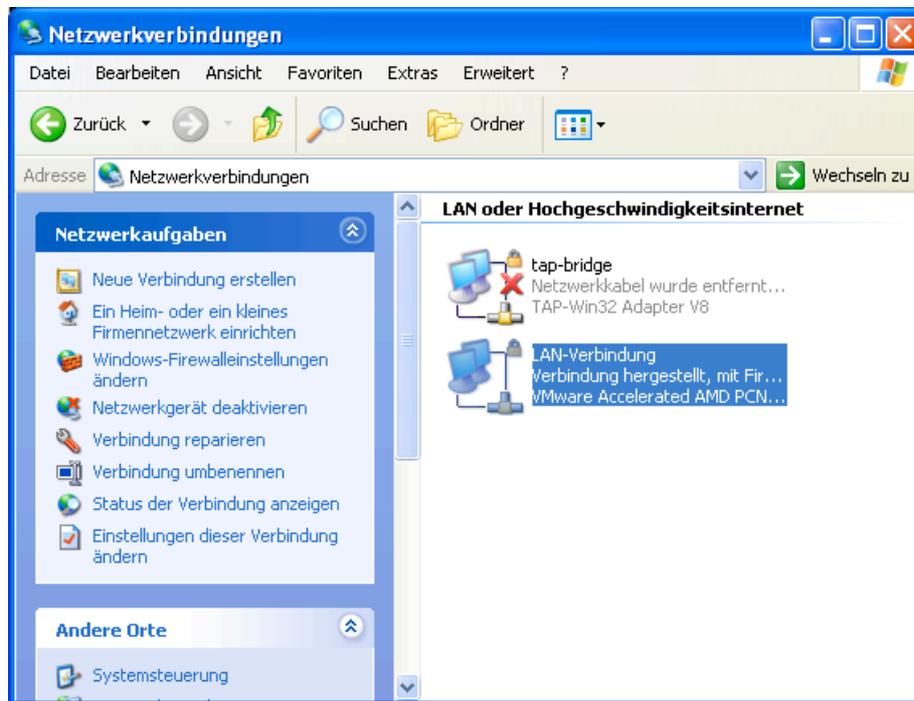
*Figure 4-11: Setting a MAC address*

2. Set also a (not yet used) valid IP address from the range: 172.16.0.0 – 172.31.255.255.



*Figure 4-12: Setting an IP address*

3. Rename the TAP Interface in the “network connections” settings to “tap-bridge”.



*Figure 4-13: Network connections*

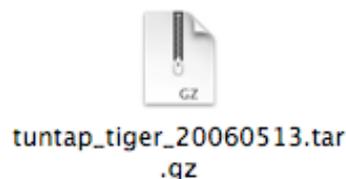
Windows will tell you, that there is no cable plugged in the “tap-bridge”. This is no problem, as long as MCAST is not running, the interface is not “opened” hence Windows will claim that “no network cable” is connected. After you started MCAST, this message should disappear and the “tap-bridge” should be connected as any other working network interface in Windows.

#### 4.3 Installation of TAP under Mac OS X

In the following subsections we describe how to install and configure the TAP driver (which is required for MCAST) for the Mac OS X platform.

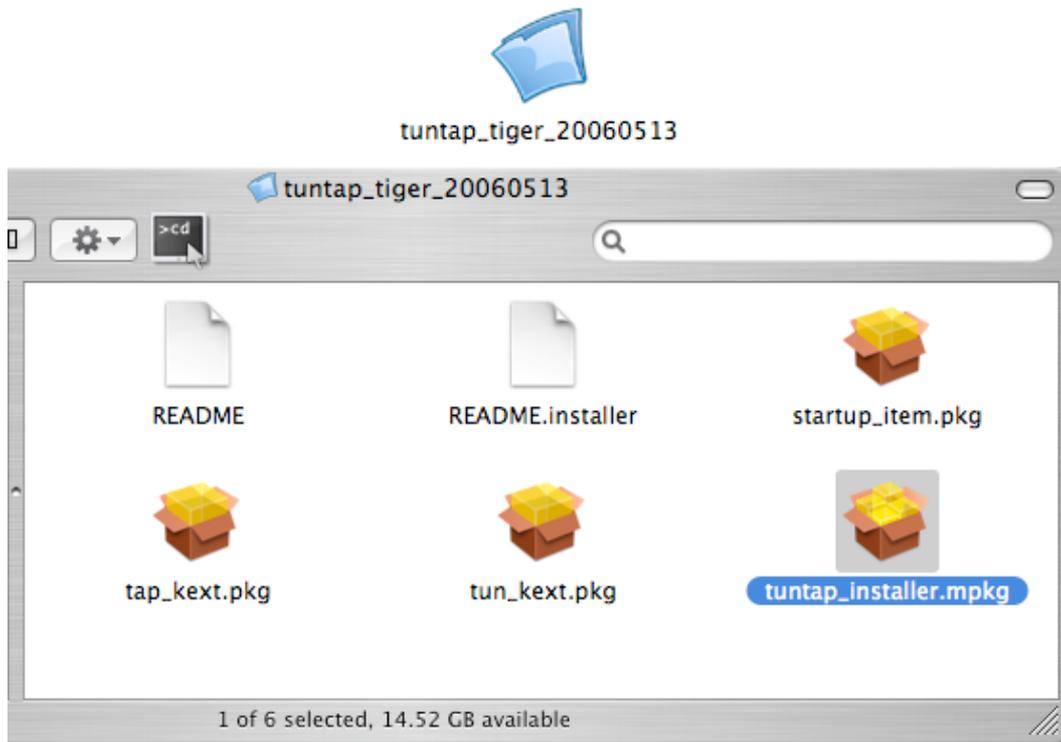
Please, **follow the steps** described in this installation manual.

1. Unpack the tuntap tiger driver archive (tuntap\_tiger\_20060513.tar.gz) downloaded from the EuQoS FTP server.



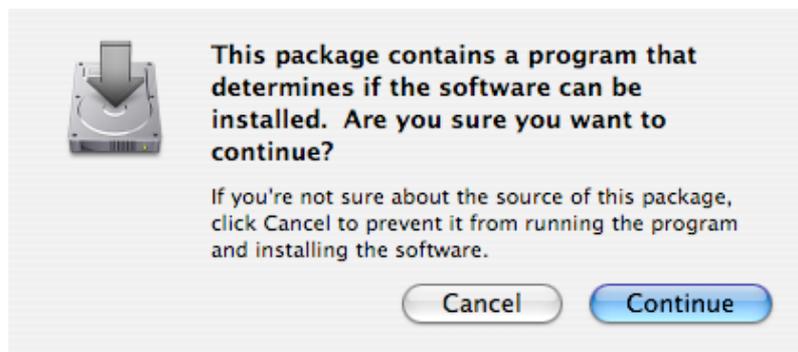
*Figure 4-14: Tuntap driver archive for Mac OS X*

2. In the folder `tuntap_tiger_20060513` launch the installer called “`tuntap_installer.mpkg`”.



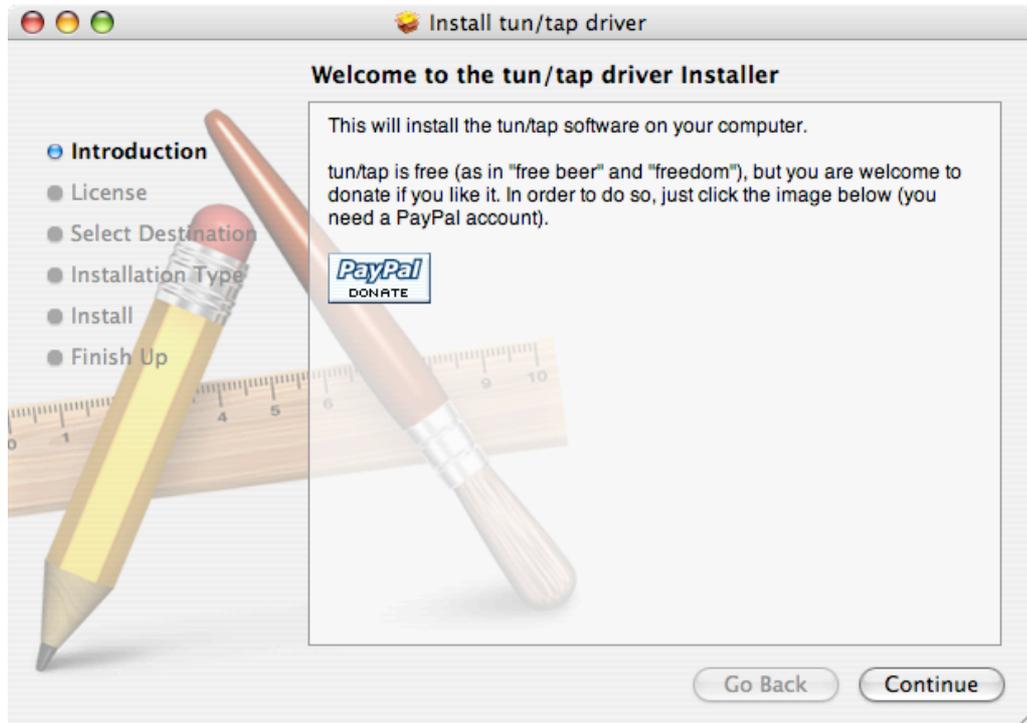
*Figure 4-15: Tuntap driver installer for Mac OS X*

3. Click on “Continue” on the installation confirmation requester.



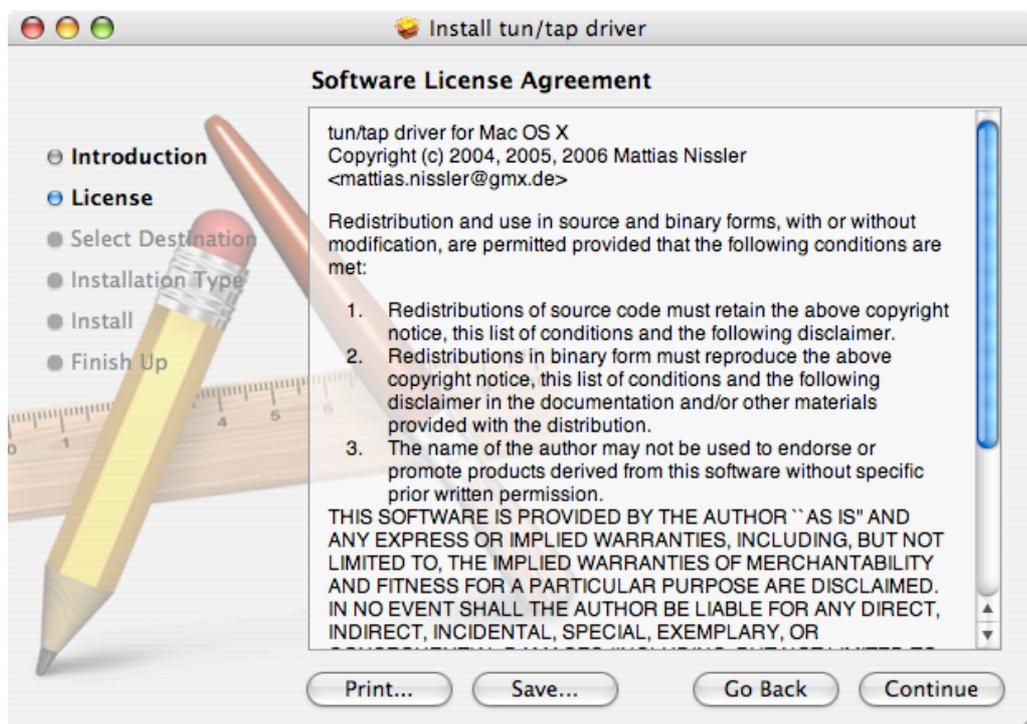
*Figure 4-16: Installation conformation requester*

- Click on “Continue” on the welcome screen.



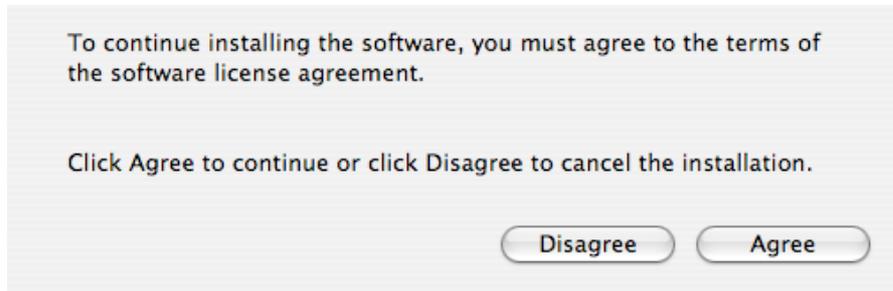
*Figure 4-17: Welcome screen of the installer*

- Click on “Continue” when you have read the Software License Agreement.



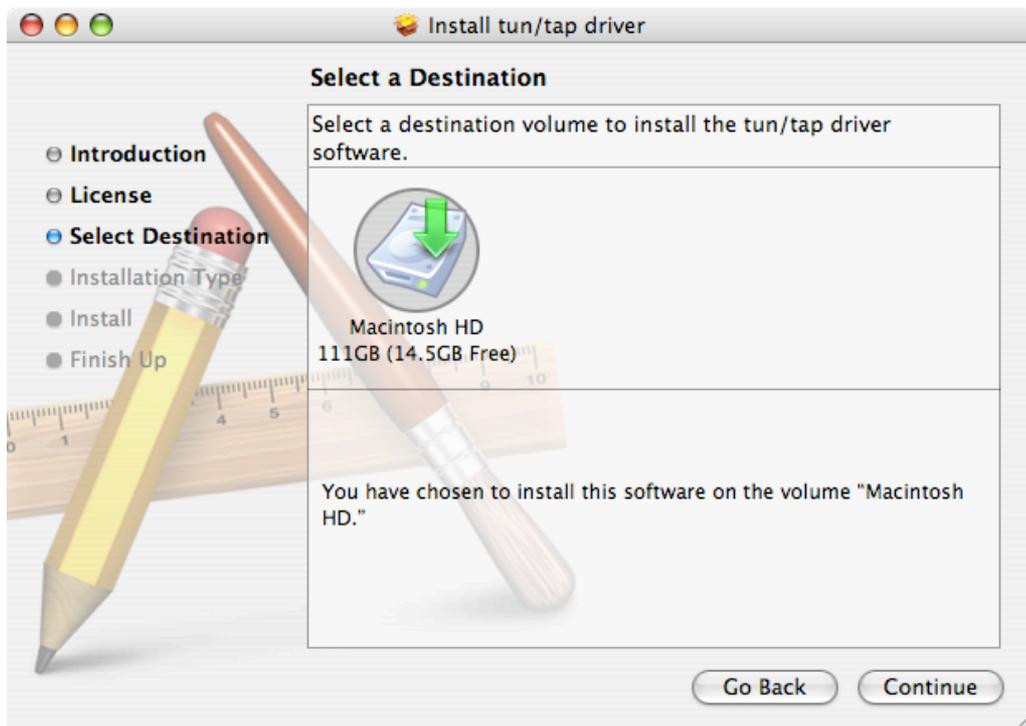
*Figure 4-18: Software License Agreement*

6. Click on “Agree” to accept the Software License Agreement.



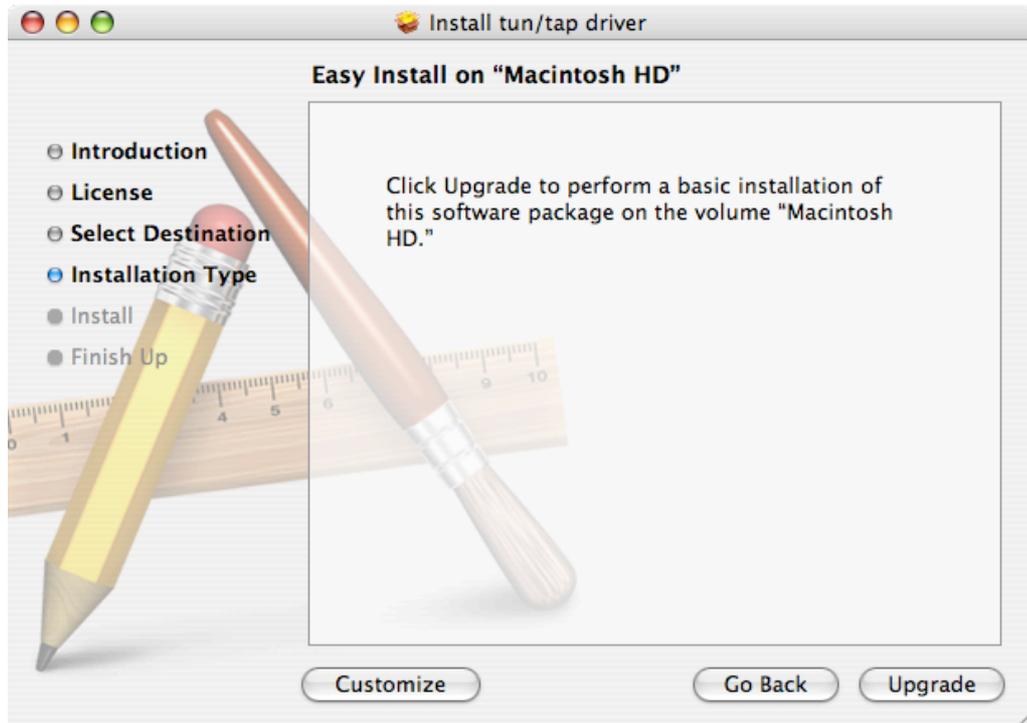
*Figure 4-19: Accepting the Software License Agreement*

7. Select a destination (the boot hard drive!) and click on “Continue”.



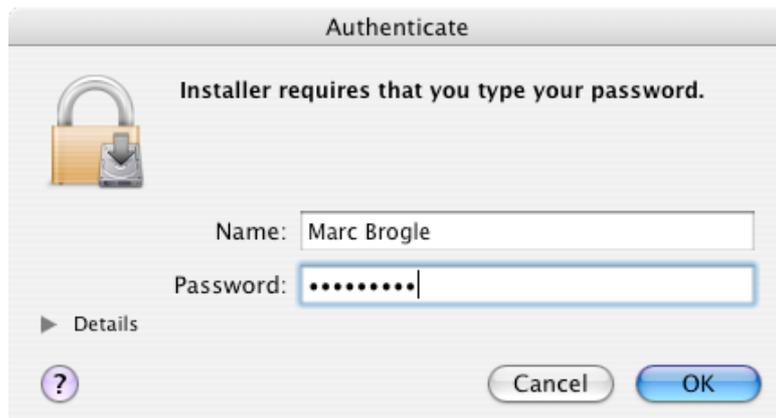
*Figure 4-20: Installation destination selection*

- Click on “Install” (fresh install) or “Upgrade” (update existing) to start the installation.



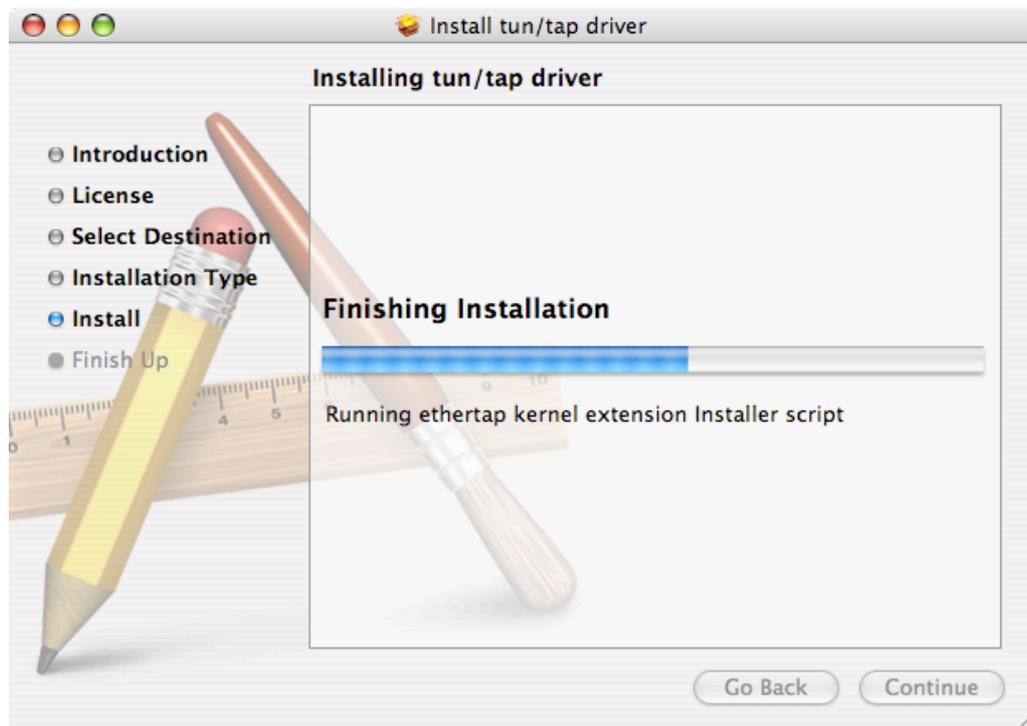
*Figure 4-21: Start the installation*

- Authenticate (root rights required) yourself and click on “OK”.



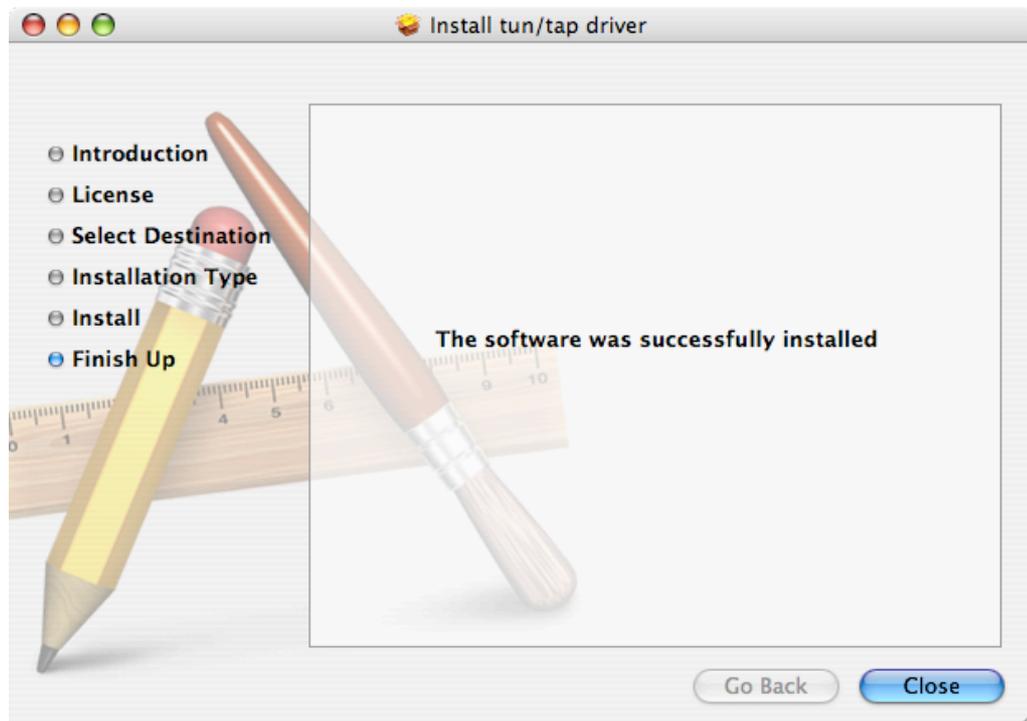
*Figure 4-22: Authentication requester*

10. Wait until all files are installed.



*Figure 4-23: Installation progress*

11. Click on “Close” to finish the installation.



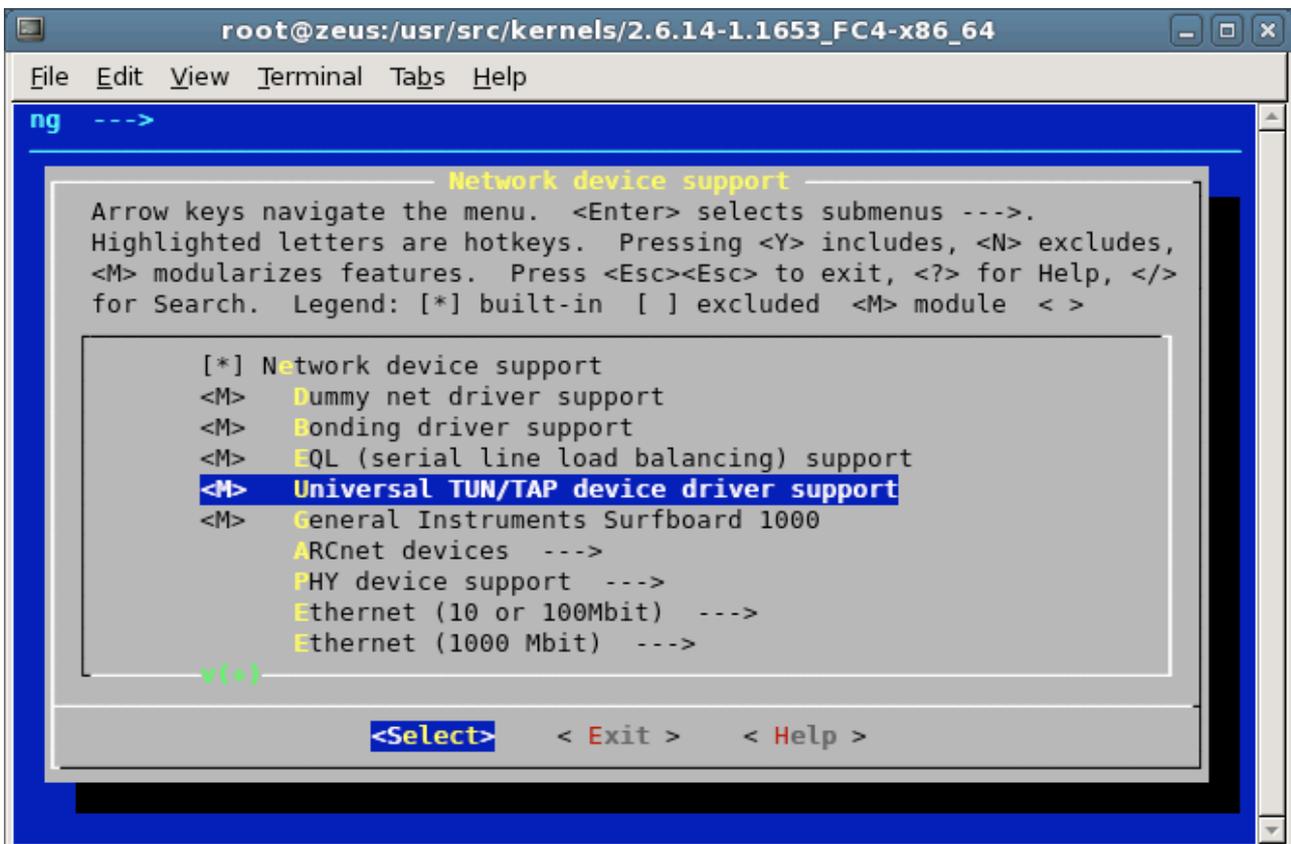
*Figure 4-24: Successful installation*

#### 4.4 Configuring of TAP under Mac OS X

No configuration is required, after you restarted the system you have successfully installed and enabled the tuntap driver for Mac OS X.

#### 4.5 Installation of TAP under Linux

Normally Linux comes with TAP support “out of the box”: Nevertheless, if you need to compile a new kernel version that supports TAP use `make menuconfig` and navigate to “Device Drivers → Network device support → Universal TUN/TAP device driver support” and enable the “Universal TUN/TAP device driver support” as shown in Figure 4-25.



*Figure 4-25: TAP support for the Linux kernel*

#### 4.6 Configuring of TAP under Linux

No configuration is required for TAP under Linux, restart the system if you recompiled the kernel.

## 5 Installing and configuring MCAST

The following subsections will describe the installation and configuration of the MCAST module.

Please note, that the MCAST version provided at the moment will take the QoS requests from applications, but will actually not perform the corresponding EuQoS reservations. This will ease the development of EuQoS MCAST aware applications using Multicast QoS-reservations without requiring a fully working EuQoS environment.

### 5.1 Installing MCAST

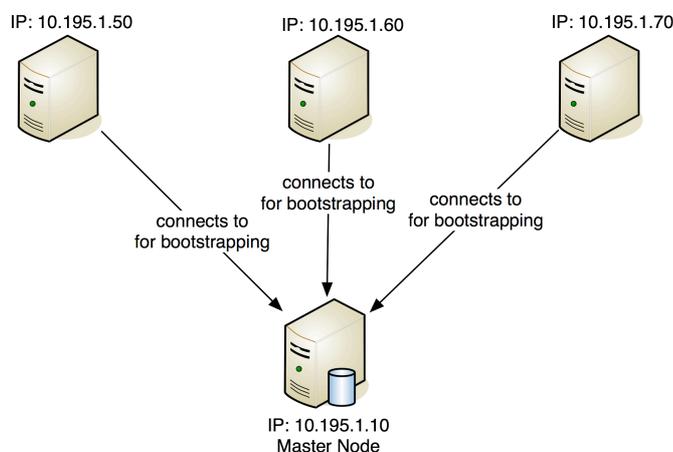
To install MCAST unzip the archive `mcast_v4.0.4.zip` downloaded from the EuQoS FTP server to a directory of your choice, e.g. `C:\euqos\p4\mcast`.

### 5.2 Configuring MCAST

Edit the `qcm.properties` in the `conf/` (resp. `conf\`) subfolder of the MCAST installation to point to **your** AQSSN (e.g. for UBern): `aqssn.location = http://10.95.0.4:8080/aqssn/services/AQSSN`

To configure MCAST you have to modify the start file called `start-mcast.bat` (Windows) or `start-mcast.sh` (Mac OS X & Linux) in the directory where you installed MCAST as follows:

1. Set the `$JAVA_HOME` (for Mac OS X and Linux) or `%JAVA_HOME%` (for Windows) variable to point to the SUN JDK 1.5 installation directory.
2. One host has to be the master node to which an instance of MCAST has to connect to for bootstrapping (to find the P2P network for a given multicast group). On that host set the `$MASTER_NODE` resp. `%MASTER_NODE%` to its own public IP address as follows: `IP_ADDRESS/2222` (e.g. `10.195.1.10/2222`). For all other instances of MCAST set this variable accordingly to point to the same master node defined before (see also Figure 5-1).



For all hosts (including the master node itself), the variable `$MASTER_NODE` is set to `10.195.1.10/2222`

*Figure 5-1: Connections resulting from setting `$MASTER_NODE`*

## 6 Starting and testing MCAST installation

### 6.1 Starting MCAST installation under Linux or Mac OS X

To start MCAST under Linux or Mac OS X run the file `start-mcast.sh` as **ROOT** user in the MCAST installation directory after you have modified it as described in Section 5.2.

Please note:

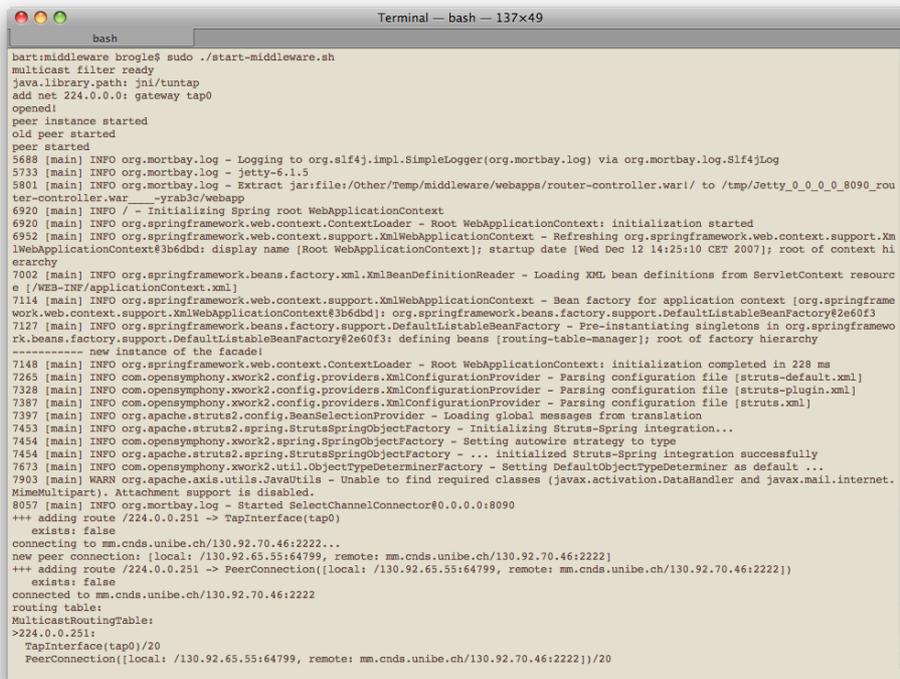
- If you start the master node, it will wait for connections from other instances of MCAST.
- If you start a non-master node instance of MCAST, it will try to connect to the master node as it has been defined during configuration (as described in Section 5.2).

The file `start-mcast.sh` looks e.g. as follows:

```
export FORCE_ROOT=${FORCE_ROOT:=false}
export MASTER_NODE=${MASTER_NODE:=10.195.0.6/2222}
export TEST_CODE=false
export USE_EUQOS=true
export SHOW_QCM_CONSOLE=false
export QCM_USERNAME=${QCM_USERNAME:=dummyauth90@euqos.org}
export QCM_PASSWORD=${QCM_PASSWORD:=password}

java -Djava.library.path=jni/tuntap -jar multicast-middleware.jar
```

A successful start of MCAST (connecting to remote master node) is shown in Figure 6-1.



```
bash
bart@middleware brogle$ sudo ./start-middleware.sh
multicast filter ready
java.library.path: jni/tuntap
add net 224.0.0.0: gateway tap0
opened!
peer instance started
old peer started
peer started
5688 [main] INFO org.mortbay.log - Logging to org.slf4j.impl.SimpleLogger(org.mortbay.log) via org.mortbay.log.Slf4jLog
5733 [main] INFO org.mortbay.log - jetty-6.1.5
5801 [main] INFO org.mortbay.log - Extract jar:file:/Other/Temp/middleware/webapps/router-controller.war!/ to /tmp/Jetty_0_0_0_8090_router-controller.war__yrb3c/webapp
6920 [main] INFO / - Initializing Spring root WebApplicationContext
6920 [main] INFO org.springframework.web.context.ContextLoader - Root WebApplicationContext: initialization started
6952 [main] INFO org.springframework.web.context.support.XmlWebApplicationContext - Refreshing org.springframework.web.context.support.XmlWebApplicationContext#3b6dbd: display name [Root WebApplicationContext]; startup date [Wed Dec 12 14:25:10 CET 2007]; root of context hierarchy
7002 [main] INFO org.springframework.beans.factory.xml.XmlBeanDefinitionReader - Loading XML bean definitions from ServletContext resource [/WEB-INF/applicationContext.xml]
7114 [main] INFO org.springframework.web.context.support.XmlWebApplicationContext - Bean factory for application context [org.springframework.web.context.support.XmlWebApplicationContext#3b6dbd]: org.springframework.beans.factory.support.DefaultListableBeanFactory#2e60f3
7127 [main] INFO org.springframework.beans.factory.support.DefaultListableBeanFactory - Pre-instantiating singletons in org.springframework.beans.factory.support.DefaultListableBeanFactory#2e60f3: defining beans [routing-table-manager]; root of factory hierarchy
----- new instance of the facade!
7148 [main] INFO org.springframework.web.context.ContextLoader - Root WebApplicationContext: initialization completed in 228 ms
7265 [main] INFO com.opensymphony.xwork2.config.providers.XmlConfigurationProvider - Parsing configuration file [struts-default.xml]
7328 [main] INFO com.opensymphony.xwork2.config.providers.XmlConfigurationProvider - Parsing configuration file [struts-plugin.xml]
7387 [main] INFO org.opensymphony.xwork2.config.providers.XmlConfigurationProvider - Parsing configuration file [struts.xml]
7397 [main] INFO org.apache.struts2.config.BeanSelectionProvider - Loading global messages from translation
7453 [main] INFO org.apache.struts2.spring.StrutsSpringObjectFactory - Initializing Struts-Spring integration...
7454 [main] INFO com.opensymphony.xwork2.spring.SpringObjectFactory - Setting autowire strategy to type
7454 [main] INFO org.apache.struts2.spring.StrutsSpringObjectFactory - ... initialized Struts-Spring integration successfully
7673 [main] INFO com.opensymphony.xwork2.util.ObjectTypeDeterminerFactory - Setting DefaultObjectTypeDeterminer as default ...
7903 [main] WARN org.apache.axis.utils.JavaUtils - Unable to find required classes [javax.activation.DataHandler and javax.mail.internet.MimeMultipart]. Attachment support is disabled.
8057 [main] INFO org.mortbay.log - Started SelectChannelConnector@0.0.0.0:8090
+++ adding route /224.0.0.251 -> TapInterface(tap0)
exists: false
connecting to mm.cnds.unibe.ch/130.92.70.46:2222...
new peer connection: [local: /130.92.65.55:64799, remote: mm.cnds.unibe.ch/130.92.70.46:2222]
+++ adding route /224.0.0.251 -> PeerConnection([local: /130.92.65.55:64799, remote: mm.cnds.unibe.ch/130.92.70.46:2222])
exists: false
connected to mm.cnds.unibe.ch/130.92.70.46:2222
routing table:
MulticastRoutingTable:
>224.0.0.251:
TapInterface(tap0)/20
PeerConnection([local: /130.92.65.55:64799, remote: mm.cnds.unibe.ch/130.92.70.46:2222])/20
```

**Figure 6-1: Successful launch of MCAST under Mac OS X**

## 6.2 Starting MCAST installation under Windows

To start MCAST under Windows run the file `start-mcast.bat` in the MCAST installation directory after you have modified it as described in Section 5.2.

Please note:

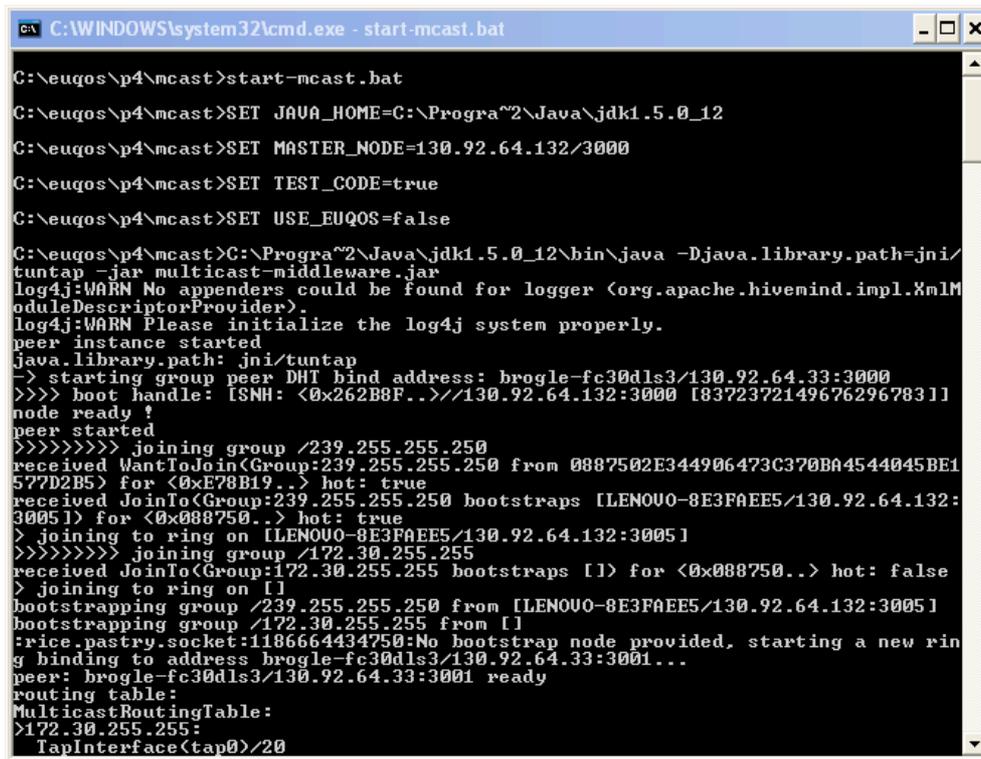
- If you start the master node, it will wait for connections from other instances of MCAST.
- If you start a non-master node instance of MCAST, it will try to connect to the master node as it has been defined during configuration (as described in Section 5.2).

The file `start-mcast.bat` looks e.g. as follows:

```
set JAVA_HOME= C:\Progra~1\Java\jdk1.5.0_12
set FORCE_ROOT=false
set MASTER_NODE=10.195.0.6/2222
set TEST_CODE=false
set USE_EUQOS=true
set SHOW_QCM_CONSOLE=false
set QCM_USERNAME=dummyauth90@euqos.org
set QCM_PASSWORD=password

%JAVA_HOME%\bin\java -Djava.library.path=jni/tuntap -jar multicast-middleware.jar
```

A successful start of MCAST (connecting to remote master node) is shown in Figure 6-2.



```
C:\WINDOWS\system32\cmd.exe - start-mcast.bat
C:\euqos\p4\mcast>start-mcast.bat
C:\euqos\p4\mcast>SET JAVA_HOME=C:\Progra~2\Java\jdk1.5.0_12
C:\euqos\p4\mcast>SET MASTER_NODE=130.92.64.132/3000
C:\euqos\p4\mcast>SET TEST_CODE=true
C:\euqos\p4\mcast>SET USE_EUQOS=false
C:\euqos\p4\mcast>C:\Progra~2\Java\jdk1.5.0_12\bin\java -Djava.library.path=jni/
tuntap -jar multicast-middleware.jar
log4j:WARN No appenders could be found for logger <org.apache.hivemind.impl.XmlM
oduleDescriptorProvider>.
log4j:WARN Please initialize the log4j system properly.
peer instance started
java.library.path: jni/tuntap
-> starting group peer DHT bind address: brogle-fc30dls3/130.92.64.33:3000
>>>> boot handle: [SNH: <0x262B8F..>]/130.92.64.132:3000 I8372372149676296783 11
node ready ?
peer started
>>>>>>> joining group /239.255.255.250
received WantToJoin(Group:239.255.255.250 from 0887502E344906473C370BA4544045BE1
577D2B5) for <0xE78B19..> hot: true
received JoinTo(Group:239.255.255.250 bootstraps [LENOU0-8E3FAEE5/130.92.64.132:
3005]) for <0x088750..> hot: true
> joining to ring on [LENOU0-8E3FAEE5/130.92.64.132:3005]
>>>>>>> joining group /172.30.255.255
received JoinTo(Group:172.30.255.255 bootstraps [1]) for <0x088750..> hot: false
> joining to ring on []
bootstrapping group /239.255.255.250 from [LENOU0-8E3FAEE5/130.92.64.132:3005]
bootstrapping group /172.30.255.255 from []
:rice.pastry.socket:1186664434750:No bootstrap node provided, starting a new rin
g binding to address brogle-fc30dls3/130.92.64.33:3001...
peer: brogle-fc30dls3/130.92.64.33:3001 ready
routing table:
MulticastRoutingTable:
>172.30.255.255:
TapInterface<tap0>/20
```

*Figure 6-2: Successful launch of MCAST under Windows*

### 6.3 Testing MCAST installation

To test MCAST you need at least two running and connected instances of MCAST, where one is defined as the master node and the other(s) is/are connecting to this master node (for details please see Section 5.2).

When all the instances of MCAST are running you can start the test application located in the client/ (resp. client\)) subfolder of the mcast installation directory. This test application will join to a multicast group, wait 1 sec. and then send the message “is alive” prefixed with its name & IP address. All already active test applications will show this message as they receive it through MCAST.

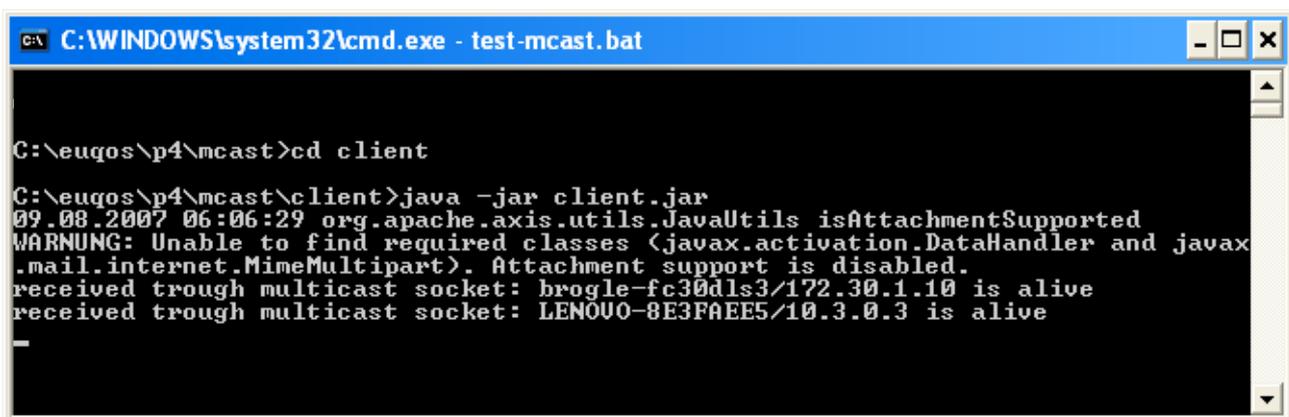
You can start the test application from within the MCAST installation folder as follows:

```
java -Djava.net.preferIPv4Stack=true -jar client/client.jar
```

Example procedure for testing two or more MCAST installations:

- Start the test application as described above on the master node. The following message will appear 1 second after the application has been started: “received through multicast socket: hostname/x.x.x.x is alive” (where hostname will be the master node’s host name and x.x.x.x will be the master nodes IP address).
- Start the test application as described above on another host running MCAST, the message “received through multicast socket: hostname/x.x.x.x is alive” will show on the test application that has been just started as well as on the instance of the test application is running on the master node (where hostname will be the host’s name and x.x.x.x will be the IP address of the host where the second application has been launched).
- If you run the application on additional machines (where MCAST is already running and connected to the master node), the message “ ... is alive” will show on all previously started and active test applications running on other hosts (including the master node).

A successful test of MCAST with two instances would look e.g. as follows:



```
C:\WINDOWS\system32\cmd.exe - test-mcast.bat

C:\euqos\p4\mcast>cd client

C:\euqos\p4\mcast\client>java -jar client.jar
09.08.2007 06:06:29 org.apache.axis.utils.JavaUtils isAttachmentSupported
WARNING: Unable to find required classes (javax.activation.DataHandler and javax
.mail.internet.MimeMultipart). Attachment support is disabled.
received through multicast socket: brogle-fc30dls3/172.30.1.10 is alive
received through multicast socket: LENOVO-8E3FAEE5/10.3.0.3 is alive
```

*Figure 6-3: Successful test of MCAST with two instances*

## 7 Example code for using QoS-enabled MCAST (DummyClient)

In this Section we present the source code of the test application used in Section 0.

To write your own client program using MCAST you can just include the library client.jar, which is located in the “client” directory of your MCAST installation. Do not move the client.jar to another location, since it uses relative paths to include other libraries from the MCAST installation.

The following parts will be marked with certain colors for better understanding:

- The Windows specific part to find the TAP interface is marked in **orange**.
- The steps required for opening a multicast socket and joining a group are marked in **blue**.
- The commands for performing the QoS reservation are marked in **red**.
- Data reception parts are marked in **green**.

```
import ch.unibe.iam.mcast.client;
import java.io.IOException;
import java.net.DatagramPacket;
import java.net.InetAddress;
import java.net.InetAddress;
import java.net.InetSocketAddress;
import java.net.MulticastSocket;
import java.net.NetworkInterface;
import java.net.SocketException;
import java.util.Enumeration;

public class DummyClient {

    private static NetworkInterface determineInterface() throws SocketException {
        NetworkInterface ret=null;
        for (Enumeration ne=NetworkInterface.getNetworkInterfaces(); ne.hasMoreElements();) {
            NetworkInterface ni=(NetworkInterface) ne.nextElement();
            // System.out.println("iface: "+ni.getName()+" display name: "+ni.getDisplayName());
            if (ni.getDisplayName().startsWith("TAP-Win32")) {
                ret=ni;
            }
        }
        return ret;
    }

    public static void main(String argv[]) throws IOException, ReservationSystemFailureException {
        InetSocketAddress groupSocketAddress=new InetSocketAddress(
            InetAddress.getByAddress(new byte[]{(byte) 239,0,0,9}),1234);
        MulticastSocket ms=new MulticastSocket(groupSocketAddress.getPort());

        NetworkInterface ni=determineInterface();
        if (ni != null) {
            ms.setNetworkInterface(ni);
        }

        ms.joinGroup(groupSocketAddress.getAddress());
    }
}
```

```
MulticastReservation mr=ReservationManager.createSession(groupSocketAddress, 0);  
mr.reserveResources(2*1024*1024, 300, 20);
```

```
byte[] packetBuffer=new byte[65536];  
DatagramPacket dp=new DatagramPacket(packetBuffer, packetBuffer.length);
```

```
boolean terminate=false;
```

```
byte toSend[]=(InetAddress.getLocalHost().toString()+" is alive").getBytes("UTF-8");  
System.arraycopy(toSend, 0, packetBuffer, 0, toSend.length);
```

```
dp.setData(packetBuffer);  
dp.setLength(toSend.length);  
dp.setSocketAddress(groupSocketAddress);  
ms.setTimeToLive(255);  
ms.send(dp);  
dp.setLength(packetBuffer.length);
```

```
do {  
    ms.receive(dp);  
    //byte data[]=dp.getData();  
    int len=dp.getLength();  
    String received=new String(packetBuffer,0,len,"UTF-8");  
    System.out.println("received trough multicast socket: "+received);  
    // TODO set terminate to true to end the loop  
} while (!terminate);
```

```
mr.release();  
ms.close();
```

```
}
```

```
}
```

The libraries (jar files) required to compile and run the DummyClient or your own application using MCAST are (relative to the “client” subdirectory in the MCAST installation directory):

- ../lib/euqos-p3/axis-1.4.jar
- ../lib/euqos-p3/axis-jaxrpc-1.4.jar
- ../lib/euqos-p3/axis-schema-1.3.jar
- ../lib/euqos-p3/axis-wsdl4j-1.5.1.jar
- ../lib/euqos-p3/commons-discovery-0.2.jar
- ../lib/euqos-p3/commons-interfaces-3.0.jar
- ../lib/euqos-p3/commons-logging-1.0.4.jar
- ../lib/euqos-p3/saaj-api-1.2.jar"

If you need further assistance in developing a skeleton and assemble the required libraries please contact Dragan Milic ([milic@iam.unibe.ch](mailto:milic@iam.unibe.ch)) for further support.

## 8 Troubleshooting

If you have problems with MCAST please have a look at the following subsections for a possible solution. If you do not find an answer there, please contact Marc Brogle ([brogle@iam.unibe.ch](mailto:brogle@iam.unibe.ch)) and Dragan Milic ([milic@iam.unibe.ch](mailto:milic@iam.unibe.ch)) for assistance.

### 8.1 MCAST is working but the application does not receive any IP multicast packets

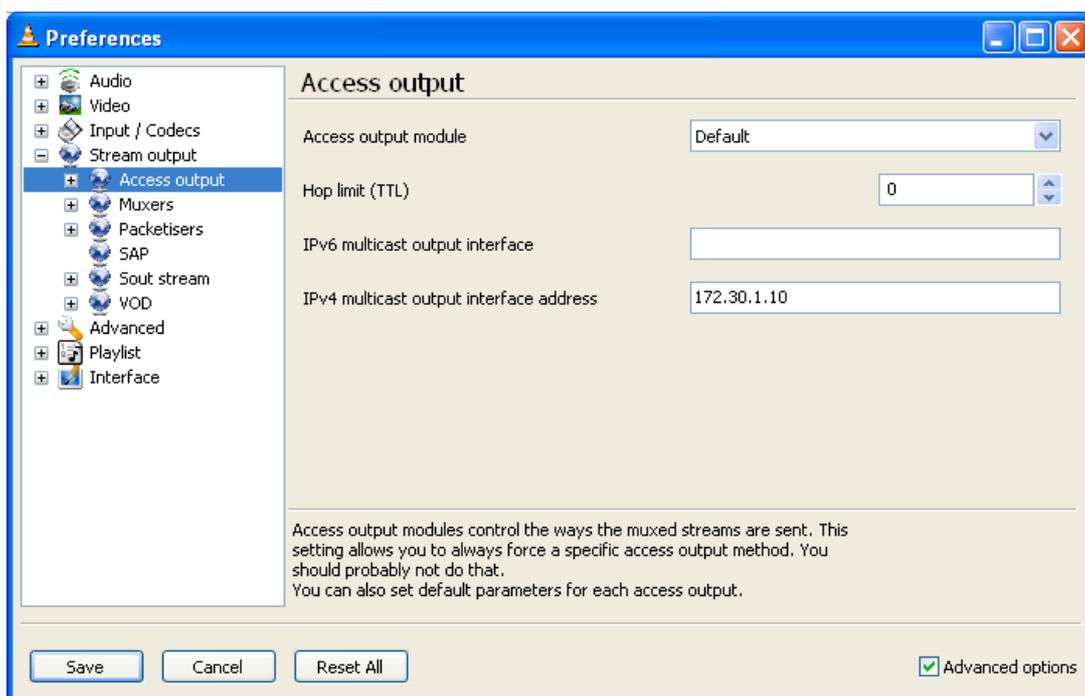
**Problem:** MCAST is configured properly; all hosts connect to the master peer and in the console of MCAST I can see that the connections are working and the multicast group joins are performed successfully. Still my application does not receive the IP Multicast packets on the machine.

**Solution:** If you are running Windows, please make sure, that the TAP interface has a unique IP address, which is not used by any other host running MCAST!

### 8.2 Problems with VLC running under Windows

**Problem:** When I want to send/receive IP Multicast data through MCAST under windows, the UDP packets are not routed through the TAP interface.

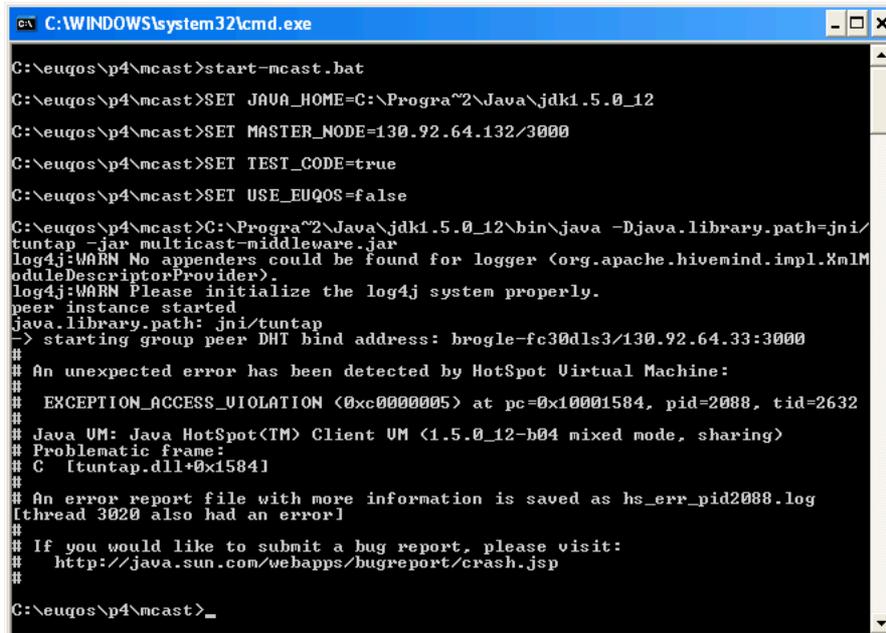
**Solution:** You have to set the IP address that has been assigned to the TAP interface in the “IPv4 multicast output interface address” option of VLC! To change these settings open VLC and hit CTRL-S to open the preferences window, enable “Advanced options” in the lower right and then select on the left in the section “Stream output” the sub-section “Access output” and enter the TAP IP address for the field “IPv4 multicast output interface address” (see also Figure 8-1).



*Figure 8-1: VLC option for setting the IPv4 multicast output interface address*

### 8.3 Problems when starting MCAST under Windows

**Problem:** When I run MCAST under Windows a message error message saying something like “... EXCEPTION\_ACCESS\_VIOLATION ...” appears (see also Figure 8-2).



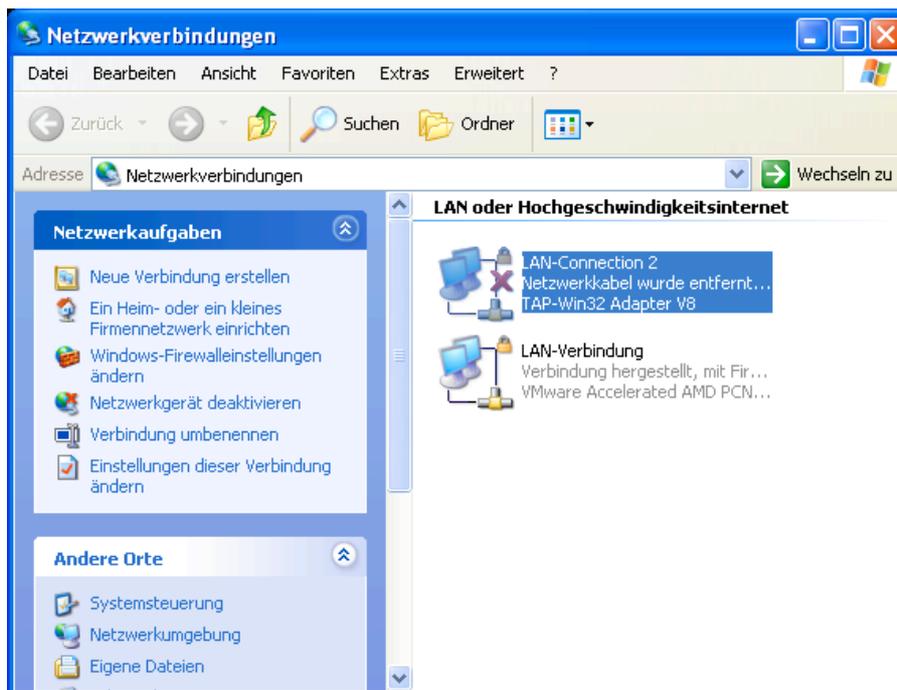
```

C:\WINDOWS\system32\cmd.exe
C:\Neuqos\p4\mcast>start-mcast.bat
C:\Neuqos\p4\mcast>SET JAVA_HOME=C:\Progra~2\Java\jdk1.5.0_12
C:\Neuqos\p4\mcast>SET MASTER_NODE=130.92.64.132/3000
C:\Neuqos\p4\mcast>SET TEST_CODE=true
C:\Neuqos\p4\mcast>SET USE_EUQOS=false
C:\Neuqos\p4\mcast>C:\Progra~2\Java\jdk1.5.0_12\bin\java -Djava.library.path=jni/
tuntap -jar multicast-middleware.jar
log4j:WARN No appenders could be found for logger (org.apache.hivemind.impl.XmlM
oduleDescriptorProvider).
log4j:WARN Please initialize the log4j system properly.
peer instance started
java.library.path: jni/tuntap
-> starting group peer DHT bind address: brogle-fc30dls3/130.92.64.33:3000
##
## An unexpected error has been detected by HotSpot Virtual Machine:
##
## EXCEPTION_ACCESS_VIOLATION (0xc0000005) at pc=0x10001584, pid=2088, tid=2632
##
## Java VM: Java HotSpot(TM) Client VM (1.5.0_12-b04 mixed mode, sharing)
## Problematic frame:
## C [tuntap.dll+0x1584]
##
## An error report file with more information is saved as hs_err_pid2088.log
[thread 3020 also had an error]
##
## If you would like to submit a bug report, please visit:
## http://java.sun.com/webapps/bugreport/crash.jsp
##
C:\Neuqos\p4\mcast>_

```

*Figure 8-2: Error message with wrong TAP interface name*

**Solution:** Be sure that the TAP interface in the network connection settings has been renamed to “tap-bridge” and does not have another name like e.g. “LAN Connection 2” (see also Figure 8-3).



*Figure 8-3: Wrongly named TAP interface*