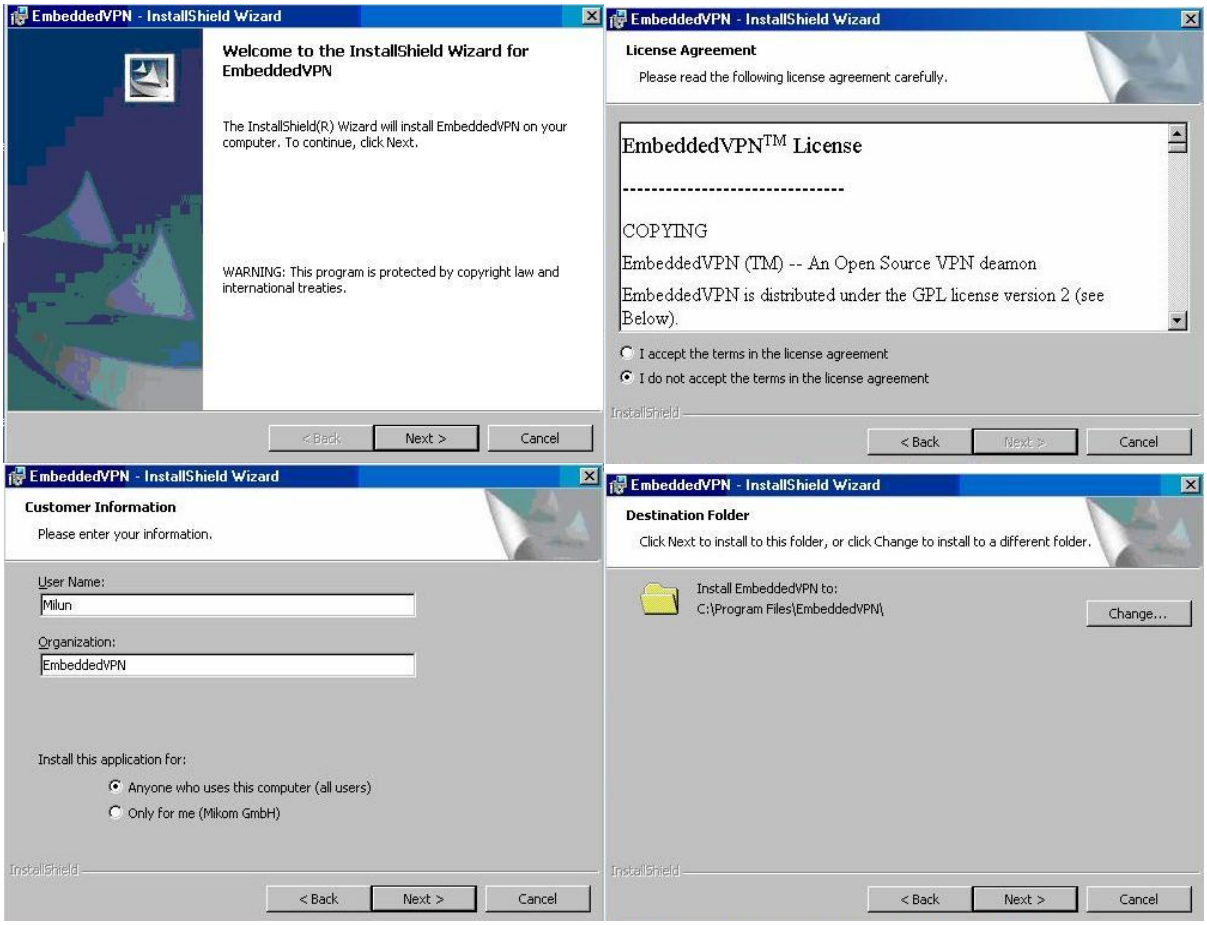
EmbeddedVPN installation procedures

EmbeddedVPN is application which is installed and tested on a different Microsoft Operating Systems. It runs on Windows 2000 Server, Windows 2000 Professional, XP Professional, Windows 2003 Server....
The installation process is very simple and passes through 6 standard installation steps. During the installation it is allowed to be changed destination folder. It is only configuration parameter which can be changed by installer. If installation can not run it is very likely that Windows Installer service has not started. In that case it is necessary to be started same service from: Control Panel -> Administrative Tools -> Services -> Windows Installer.



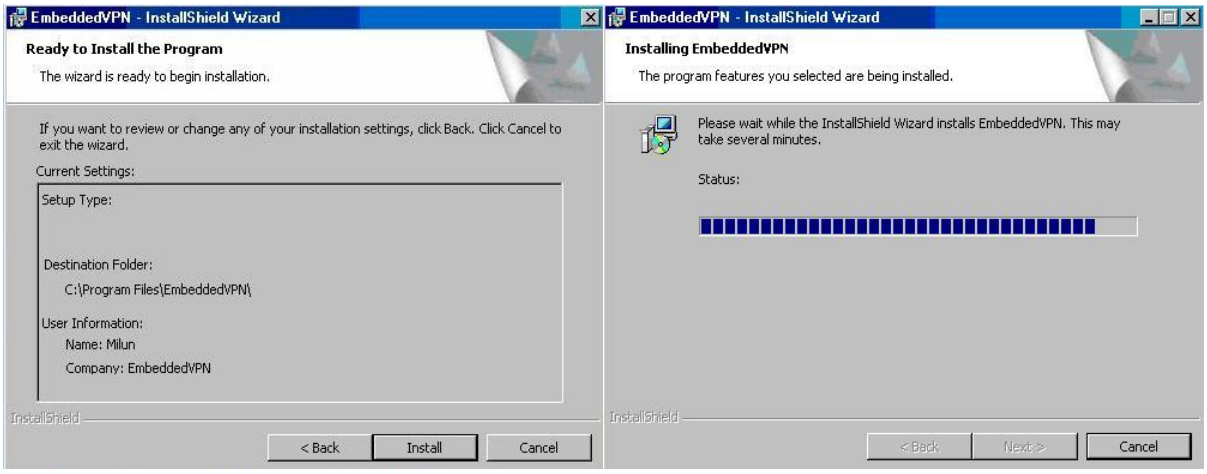


Figure 1: EmbeddedVPN application installation windows

After successfully installed EmbeddedVPN application there is one extra step which is necessary to be performed before application become ready for service: Adding a **TAP-Win32 virtual Ethernet adapter**. Virtual Ethernet adapter can be added using batch file pointed from EmbeddedVPN shortcuts section:
Start->Programs->EmbeddedVPN->Add a new TAP Win-32 virtual Ethernet adapter
Installation of TAP adapter takes about 10-15 seconds.

During installation of TAP adapter Windows OS can send message box with information that TAP-Win32 Adapter has not passed Windows Logo testing (Figure 2b). This message box should be ignored and process of installation should continue pressing "Continue" button. After successfully finished installation of Win32 TAP adapter a new Network Connection must appear in Control Panel-> Network Connections: **TAP-Win32 Adapter-V8**.

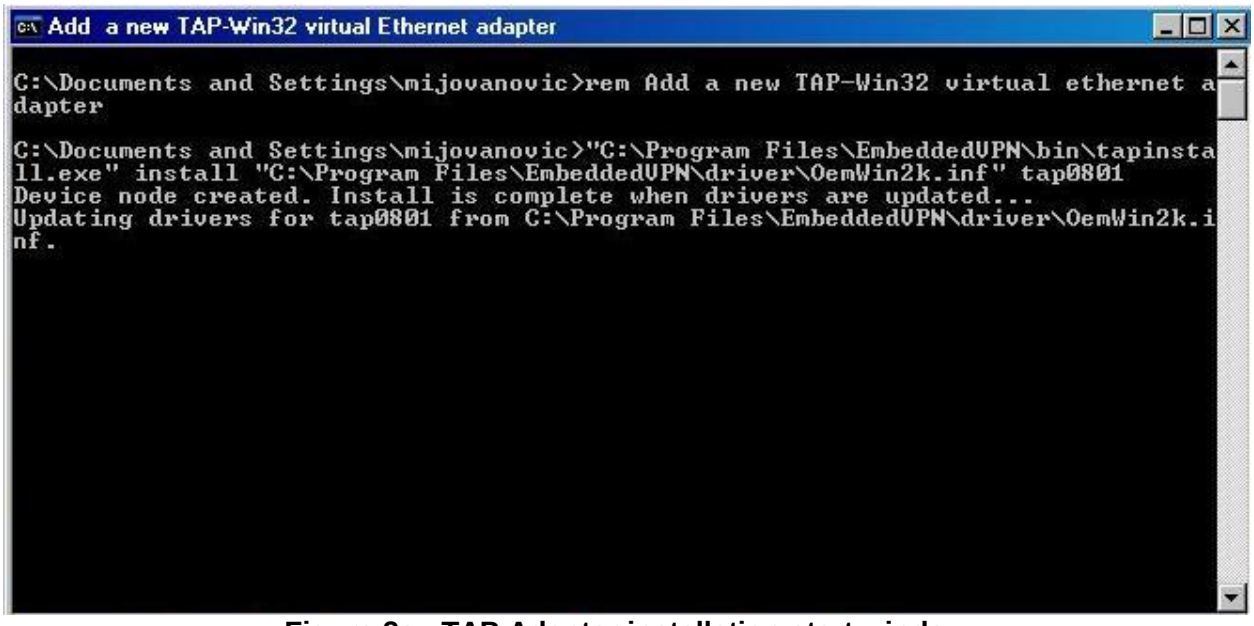


Figure 2a: TAP Adapter installation start window

Figure 2b: TAP Adapter installation warning message box

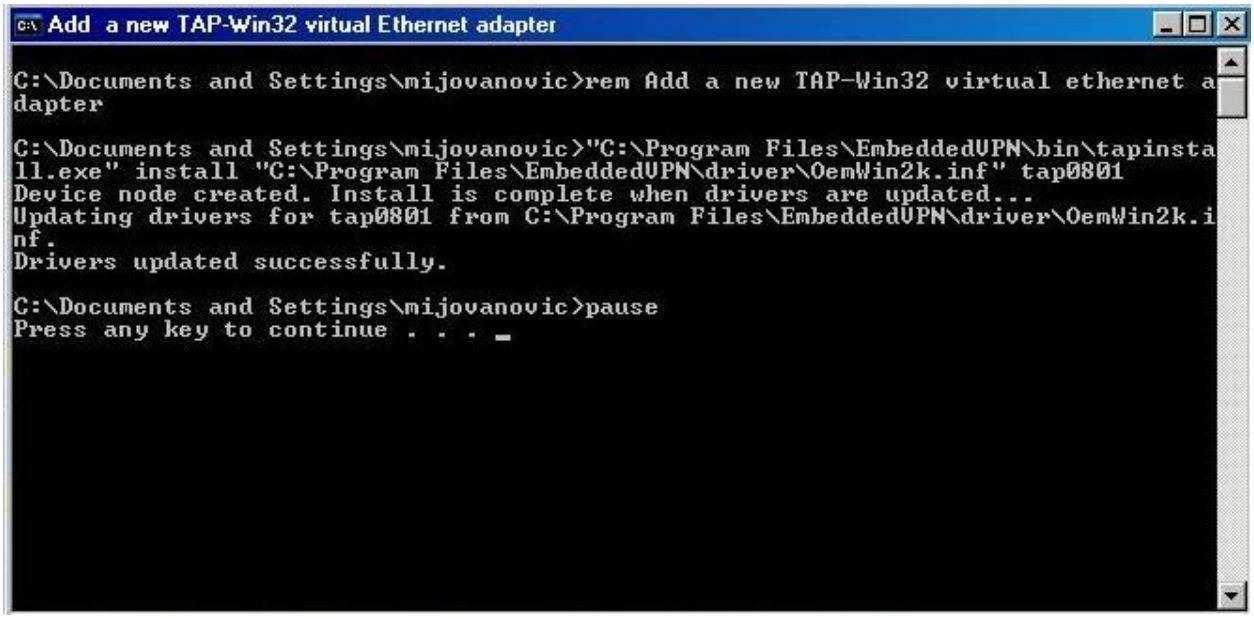


Figure 2c: TAP-Win32 Adapter V8 driver installation has succeeded

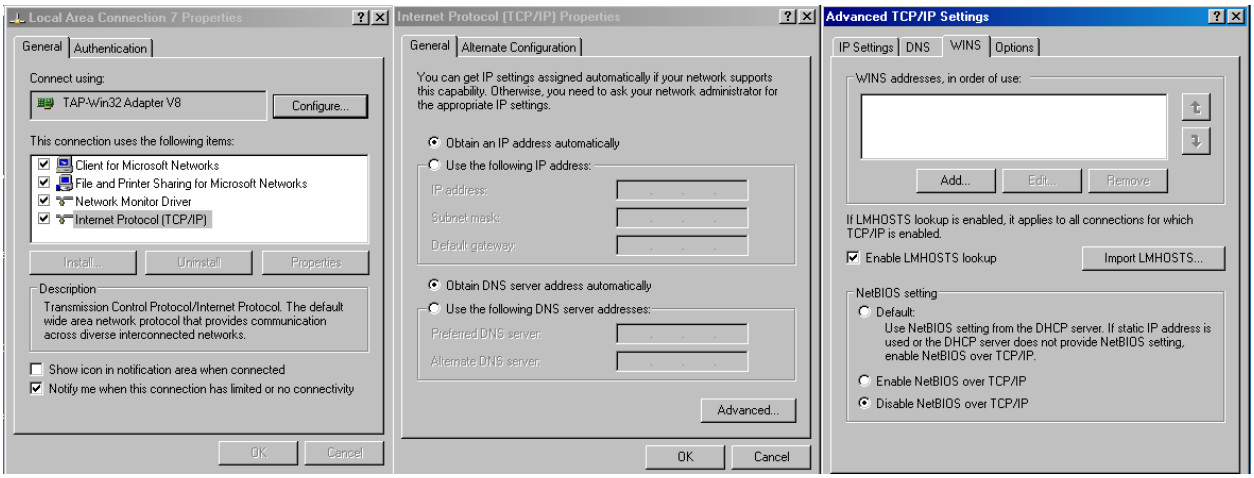


Figure 3: Disabling NetBIOS over TCP/IP on TAP Network Connection

Before starts EmbeddedVPN application it is strongly recommended that NetBIOS protocol should be disabled on a Local Area Connection dedicated to the Win32-TAP adapter. The main reason is that Windows OS starts broadcasting NetBIOS packets every half second to all VPN clients through VPN tunnel. The NetBIOS packets generated by Windows OS on a TAP network connection are not payload and they should be disabled (see Figure 3)

Embedded VPN Configuration Parameters

There are two configuration files with mandatory and optionally parameters used by EmbeddedVPN server. During installation of EmbeddedVPN application the file extension ***.evpn*** is registered in Windows registry as a configuration file extension used by EmbeddedVPN application. For running EmbeddedVPN application it is enough to have valid **.evpn** text file. Another text file with arbitrary extension can be used for assigning fix virtual IP addresses to VPN clients and is used by server. The pointer to this configuration file must exist in an .**evpn** file.

VPN Server Mandatory Parameters:

- "udp" or "tcp" a carrier IP protocol
- "tap" or "tun" virtual tunnel type
- "USER" Identity & Security: List of VPN Clients Usernames and Passwords
- "server "IP range of VPN subnet (start IP address with net mask)

VPN Server Optional Parameter:

- Configuration file for assigning fix virtual IP addresses to EmbeddedVPN clients. If is not in use this configuration file, VPN client's IP will be chosen randomly from IP range. Please take a look on" **ifconfig-pool-persist "**parameter of .evpn config file .
- "client to client" allows clients to "see" each other
- "log" used for saving in the text file messages from DOS console

- "verb" debug level used for displaying/saving messages

VPN Client Mandatory Parameters:

- "client"
- "udp" or "tcp" a carrier IP protocol
- "tap" or "tun" virtual tunnel type
- "lport" local TCP/IP client's port
- "remote" VPN server IP address and port
- "ping " used for keeping open socket in firewall and for monitoring presence of server
- "encryption" used for defining authenthication and encryption options

VPN Clients Optional Parameter:

- "log" used for saving in the text file messages from DOS console
- "verb" debug level used for displaying/saving messages

Embedded VPN Server Sample Config Files:

This .evpn configuration file describes how to setup a configuration to accept a VPN connection from EmbeddedVPN clients. See the parameter section in the lines down.

```
# Protocol:
# TCP or UDP server?
# UDP is recommended, TCP only if higher applications
# use protocols based on UDP
# The client protocol must match server protocol
# Protocol:
proto udp
;proto tcp

# Port:
# Which TCP/UDP port should EmbeddedVPN server
# listens on?
# If you want to run multiple EmbeddedVPN instances
# on the same machine, use a different port
# number for each one.  You will need to
# open up this port on your firewall.
#Port:
port 11195

# IP-Range:
# Configure server mode and supply a VPN subnet
# for EmbeddedVPN to draw client addresses from.
# The server will take first possible address for
# itself, the rest will be made available to clients.
#IP-Range:
server 10.9.10.0 255.255.255.0

# persist_IP
# Maintain a record of client <-> virtual IP address
# associations in this file.  If EmbeddedVPN goes down or
# is restarted, reconnecting clients can be assigned
# the same virtual IP address from the pool that was
# previously assigned.
#persist_IP
ifconfig-pool-persist "C:\\Program Files\\EmbeddedVPN\\config\\IPConfig.txt" 30
```

```
# Log-File:
# Copy application output messages from DOS console
# to the text file. The messages will be displayed
# on console and saved in the file
#Log-File:
log EmbeddedVPNServer.log

client-to-client

# Log-Level:
# 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
#Log-Level:
verb 3


# Dev-Type:
# "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.
# With TUN IP address of clients will incremented by 2
# TAP prefered
#Dev-Type:
dev tap
#dev tun


# List of unique Usernames and Passwords assigned to
# EmbeddedVPN clients
#e.g.
USER: Nikola Tesla
USER: Amadeus Mozart
USER: Leonardo Davinci
USER: Charles Darwin
```

In the installation folder “C:\Program Files\EmbeddedVPN\config” exists second configuration file: “**IPConfig.txt**”. This file is used for assigning fix virtual IP address to VPN clients e.g.:

```
Nikola,10.9.10.2
Leonardo,10.9.10.18
Amadeus,10.9.10.123
Charles,10.9.10.211
```

It is mandatory that first VPN client in the config file gets first free IP address from IP range assigned for VPN subnet: e.g. `Nikola`, 10.9.10.2

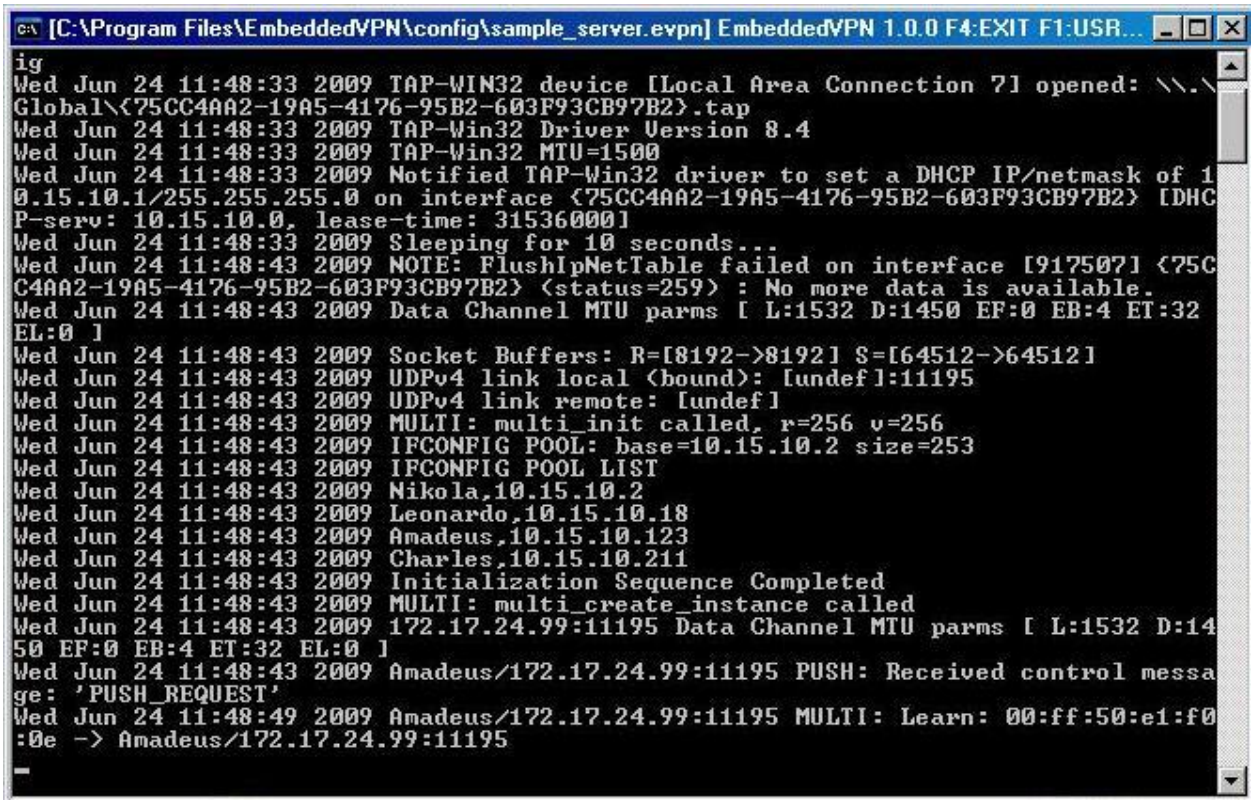


Figure 4: VPN server console window (established VPN tunnel with user "Amadeus")

On a Figure 4 we can observe following events that have occurred:

- VPN server has started and successfully initialized TAP driver
- VPN server is listening on a port 11195 and server uses IPv4 UDP socket
- Start IP address for VPN clients is 10.9.10.2 and 253 IP addresses are allocated for clients
- "Amadeus" VPN client has been authenticated and accepted by VPN server

Embedded VPN Client Sample Config Files:

This .evpn configuration file describes how to setup a configuration to create a VPN connection to EmbeddedVPN server. See the parameter section in the lines down.

```
client

# Protocol:
# TCP or UDP Client?
# UDP is recommended, TCP only if higher protocols are UDP based
# The client protocol must match server protocol
#Protocol:
proto udp
;proto tcp

# Port:
# Which TCP/UDP port should EmbeddedVPN client open
# and use for connection to the remote server
#Port:
```

```
lport 11196


# Remote EmbeddedVPN server IP address and port:
# remote xyz.xyz.xyz.xyz Port
remote 127.0.0.1 11195


# Log-File:
# Copy application output messages from DOS console
# to the text file. The messages will be displayed
# on console and saved in the file
#Log-File:
log EmbeddedVPNClient.log


# Log-Level:
# Set the appropriate level of log
# file verbosity.
# 0 is silent, except for fatal errors
# 3 or 4 is reasonable for general usage
# 5 and 6 can help to debug connection problems
# 9 is extremely verbose
#Log-Level:
verb 3

ping 30

# Dev-Type:
# "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.
# With TUN IP address of client will incremented by 2
# TAP prefered
#Dev-Type:
dev tap
#dev tun

# Unique Username and Password for this client
USER: Nikola Tesla

#Encryption is a client dependant feature and
#server accepts all requested encryptions.
#There are four available encryption levels
#which use 2 encryptions: AES and Blowfish.
#The VPN Tunnel can be active even in a nonencrypted mode

#encryption aes_128
#encryption blowfish_64
#encryption blowfish_32
encryption none
```

Testing and Basic Troubleshooting

On the server side main debugging information are printed on server console window. On Figure 5 we can observe that VPN user “Nikola” has two times successfully connected to the VPN server and VPN server has deleted previous record of VPN client. We can observe that requests for VPN connection came from two different IP addresses (90.187.21.156 and 90.186.43.216) but the same virtual IP address has been assigned to predefined VPN client with username “Nikola” (10.9.10.2).

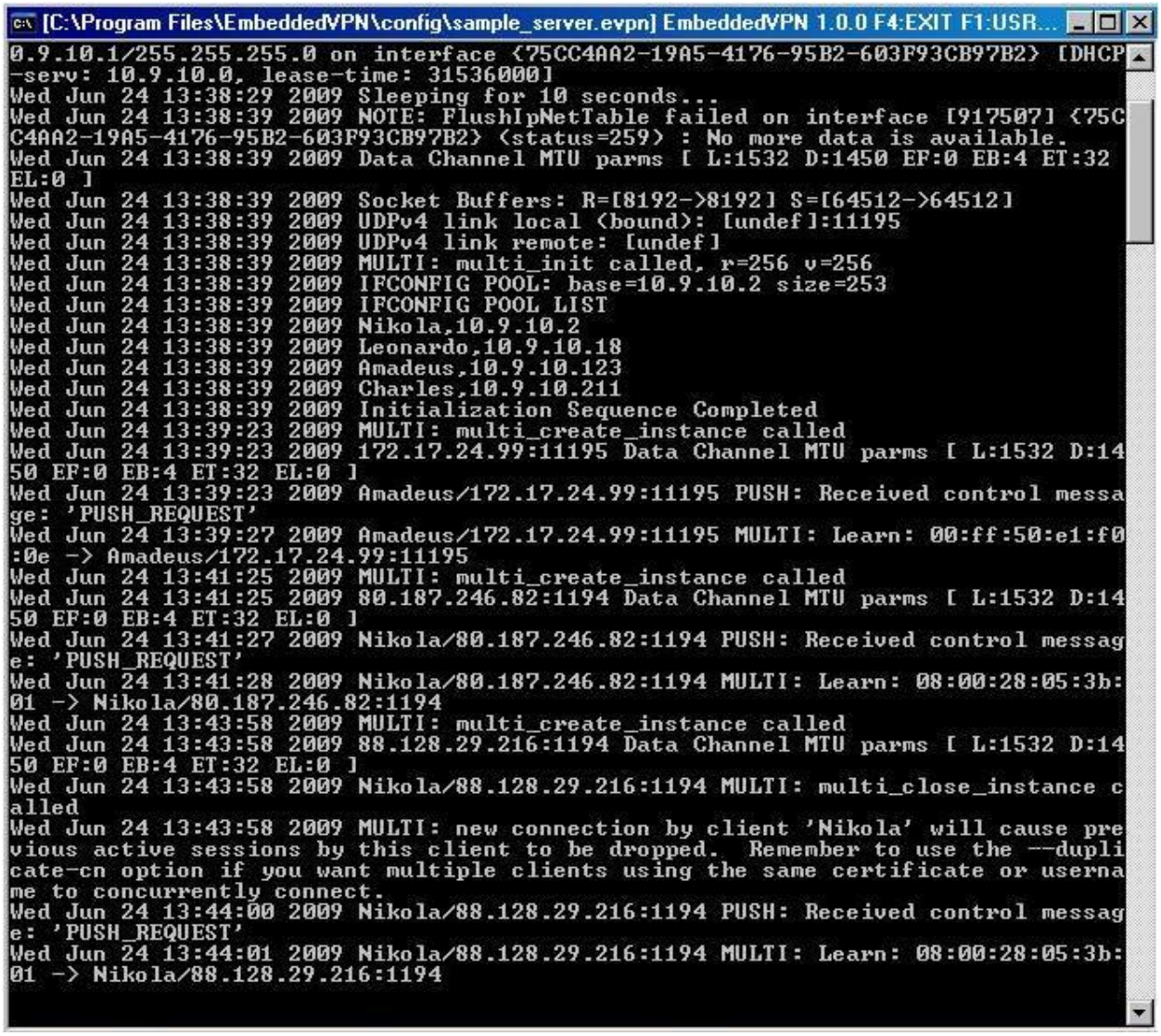


Figure 5: VPN Server console window

VPN clients connected to the server can be "pinged" using standard DOS ping command .It is very likely that real (LAN) IP address of client is inaccessible but virtual IP address is accessible.

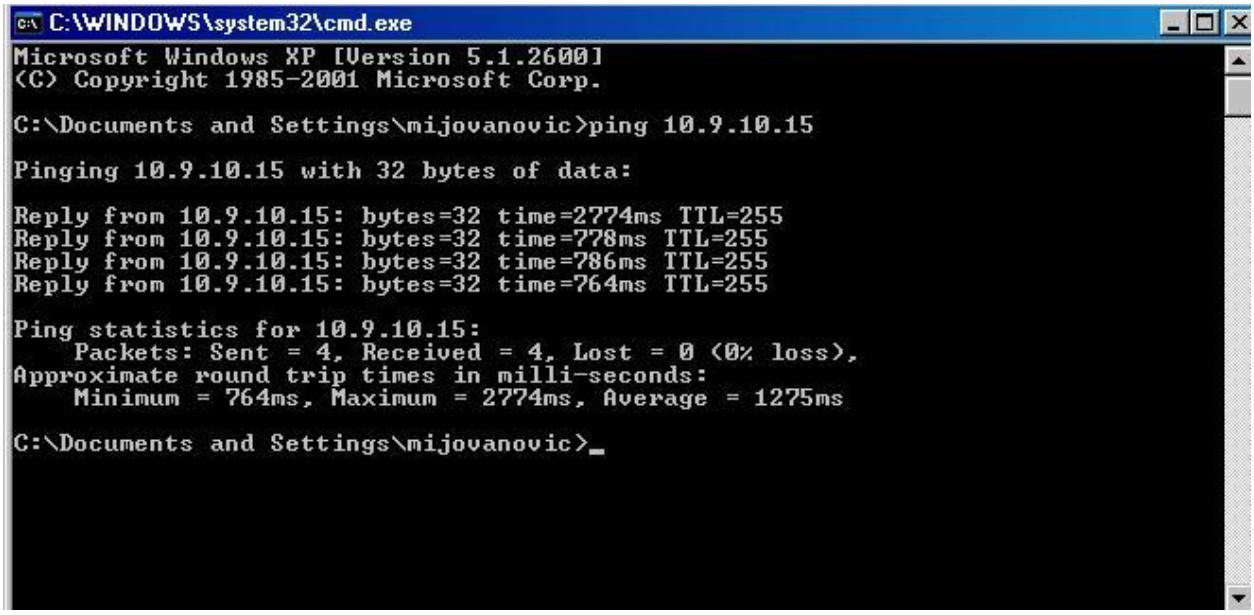


Figure 6 : DOS console used for pinging VPN clients

System requirements for Embedded VPN application

The Embedded VPN application is tested on an plenty Microsoft Operating Systems.
The hardware requirements for a server computer are dependant from number of VPN clients that will be deployed.

For a small VPN network with 2-3 clients it is enough to be used PIII 800Mhz processor with 256 MB of RAM memory. For a VPN network with lot of clients it is recommendation to be used fast computers e.g. P IV with 1 GB of RAM memory. The requirement for available hard disk space for successful VPN server installation is 5 MB.