

NessieDVB - DVB network streaming server

User guide - hw version NessieDVB 2xS2

Version 1.0

©SmartImp s.r.o. 2010-2011

We would like to thank Mr. Akos Varga for the translation of this text.

We would also like to thank Mr. Martin Mikulasko for the webinterface design.

We would also like to thank our friends and betatesters, namely Mr. Jan Taimr and Mr. Jiří Skorunka, for helping us with debugging.

Contents

| | |
|--|----|
| NessieDVB - DVB network streaming server..... | 1 |
| Introduction..... | 4 |
| Functional description of the operating modes:..... | 4 |
| Virtual tuner mode (vtuner):..... | 4 |
| Software installation for the vtuner mode in the Linux PC..... | 5 |
| Multicasted streaming mode:..... | 5 |
| DLNA mode..... | 6 |
| Combined Vtuner or DLNA and multicasted streaming mode..... | 6 |
| Network infrastructure requirements for connecting NessieDVB to the network..... | 7 |
| NessieDVB setup:..... | 8 |
| Status:..... | 9 |
| Network set-up:..... | 10 |
| Network Configuration..... | 11 |
| Network Time..... | 14 |
| Tuners setup..... | 15 |
| Satellite selection..... | 16 |
| Update of satellite definition (ini) files..... | 17 |
| Satellite definition ,ini' file format..... | 18 |
| Tuner X LNB..... | 20 |
| Direct (no DiSEqC)..... | 21 |
| Simple DiSEqC..... | 22 |
| Advanced DiSEqC..... | 24 |
| Switch setup in 'uncommitted' mode..... | 24 |
| Setting up of cascaded 'uncommitted' a 'committed' switches..... | 26 |
| Setting the operating mode of the tuners..... | 28 |
| Disabled..... | 28 |
| VTuner (Virtual tuner)..... | 29 |
| DVBlasT - multicasting TV streamer (multicast server)..... | 29 |
| DLNA server..... | 37 |
| Support..... | 43 |
| Software installation on a Linux PC for vtuner mode..... | 49 |
| Driver installation..... | 49 |
| Driver loading using insmod..... | 49 |
| Driver loading using modprobe..... | 50 |
| Setting up the created devices' permissions..... | 50 |
| Specific requirements for particular distributions:..... | 52 |
| Vtunerc client..... | 53 |
| Compiling the client software..... | 53 |
| Running the client software..... | 54 |
| Using of the software for vtuner mode on the PC..... | 54 |
| Specification:..... | 55 |
| LED based device status signalization..... | 56 |
| Emergency firmware download..... | 56 |

List of images

| | |
|---|----|
| Drawing 1: Connection of NessieDVB in 2x Vtuner mode..... | 5 |
| Drawing 2: Connecting of NessieDVB in 2x multicasted stream..... | 6 |
| Drawing 3: Connecting NessieDVB in 1x Vtuner or DLNA and 1x multicasted streaming mode..... | 7 |
| Drawing 4: Typical structure of network infrastructure in home and SOHO circumstances..... | 8 |
| Drawing 5: Web Interface - Status..... | 10 |
| Drawing 6: Web Interface - Network Submenu..... | 11 |
| Drawing 7: Web Interface - Network Default Configuration..... | 12 |
| Drawing 8: Web Interface - Network Configuration - Fixed IP..... | 13 |
| Drawing 9: Web Interface - Network Configuration - saving of set values..... | 14 |
| Drawing 10: Web Interface - Network Time..... | 15 |
| Drawing 11: Web Interface - Tuners Submenu..... | 16 |
| Drawing 12: Web Interface - Satellite Selection..... | 17 |
| Drawing 13: Web Interface - Satellite Selection 2..... | 18 |
| Drawing 14: Web Interface - Tuner LNB Setting - Direct..... | 21 |
| Drawing 15: Direct (no DiSEqC)..... | 22 |
| Drawing 16: Simple LNB DiSEqC - two input DiSEqC switch (command „burst“ used)..... | 23 |
| Drawing 17: Web Interface - Tuner LNB Setting - 2 Inputs Switch..... | 23 |
| Drawing 18: Simple LNB DiSEqC - 4 Input DiSEqC switch (command „committed“ used)..... | 24 |
| Drawing 19: Web Interface - Tuner LNB Setting - 4 Inputs Switch..... | 24 |
| Drawing 20: Advanced DiSEqC - uncommitted switch..... | 25 |
| Drawing 21: Web Interface - Setup of uncommitted switch..... | 26 |
| Drawing 22: Advanced DiSEqC - cascade of uncommitted and committed switches..... | 27 |
| Drawing 23: Web Interface - Setting up of cascade of switches..... | 28 |
| Drawing 24: Web Interface - Setting up the operating mode of the tuners..... | 29 |
| Drawing 25: Web Interface - DVBlaster - Multicast TV Streamer..... | 30 |
| Drawing 26: Web Interface - DVBlaster - Multicast TV Streamer 2..... | 31 |
| Drawing 27: Web Interface - DVBlaster - Multicast TV Streamer 3 - transponder scan..... | 32 |
| Drawing 28: Web Interface - DVBlaster - Multicast TV Streamer 4 - services selection..... | 33 |
| Drawing 29: Reception of multicasted program in VLC..... | 34 |
| Drawing 30: Web Interface - Status after setup - both tuners in dvblast mode..... | 35 |
| Drawing 31: Web Interface - Status after setup - both tuners in dvblast mode - signal quality..... | 36 |
| Drawing 32: Web Interface - Status after setup - both tuners in vtuner mode..... | 37 |
| Drawing 33: Web interface - DLNA mode setting..... | 38 |
| Drawing 34: Web Interface - DLNA mode setting - adding of scanner services..... | 39 |
| Drawing 35: Web Interface - DLNA mode setting - selection from the global database..... | 40 |
| Drawing 36: Web Interface - DLNA mode setting - services approval..... | 41 |
| Drawing 37: DLNA TV client - main menu..... | 42 |
| Drawing 38: DLNA TV client - channel offer..... | 42 |
| Drawing 39: DLNA VLC client..... | 43 |
| Drawing 40: Web Interface - Firmware Update..... | 44 |
| Drawing 41: Web Interface - Firmware Update Step 1..... | 45 |
| Drawing 42: Web Interface - Firmware Update Step 2 - firmware file selection..... | 46 |
| Drawing 43: Web Interface - Firmware Update Step 3 - corrupted file test result..... | 47 |
| Drawing 44: Web Interface - Firmware Update Step 3 - after fw upload and successful integrity test..... | 48 |
| Drawing 45: Web Interface - Firmware Update Step 4 - flashing itself..... | 49 |

Introduction

This user guide is dedicated to NessieDVB 2xS2 device users. We'll refer to the device as NessieDVB further in this guide.

NessieDVB is a device covering the requirements of newly developing multimedia data sharing, generating and transmission trends; via network infrastructure. To commonly used devices such as NAS (Network-Attached Storage), streaming servers of video content (youtube.com, stream.cz) and devices for displaying of multimedia streams (television sets with direct internet connection, computers and multimedia receivers) we offer the possibility to add another source of multimedia content without changing the network infrastructure - DVB broadcasting.

Emphasis has been put on simplicity of control and friendliness of the user interface during the design of NessieDVB. Although it uses the most advanced technologies it does not require any deep knowledge of configuration tasks from the user, nor the skills of remote configuration via remote access (ssh, telnet, command line). At the same time it allows practically unlimited number of configuration of functional modes and DiSEqC configurations of the connected LNB.

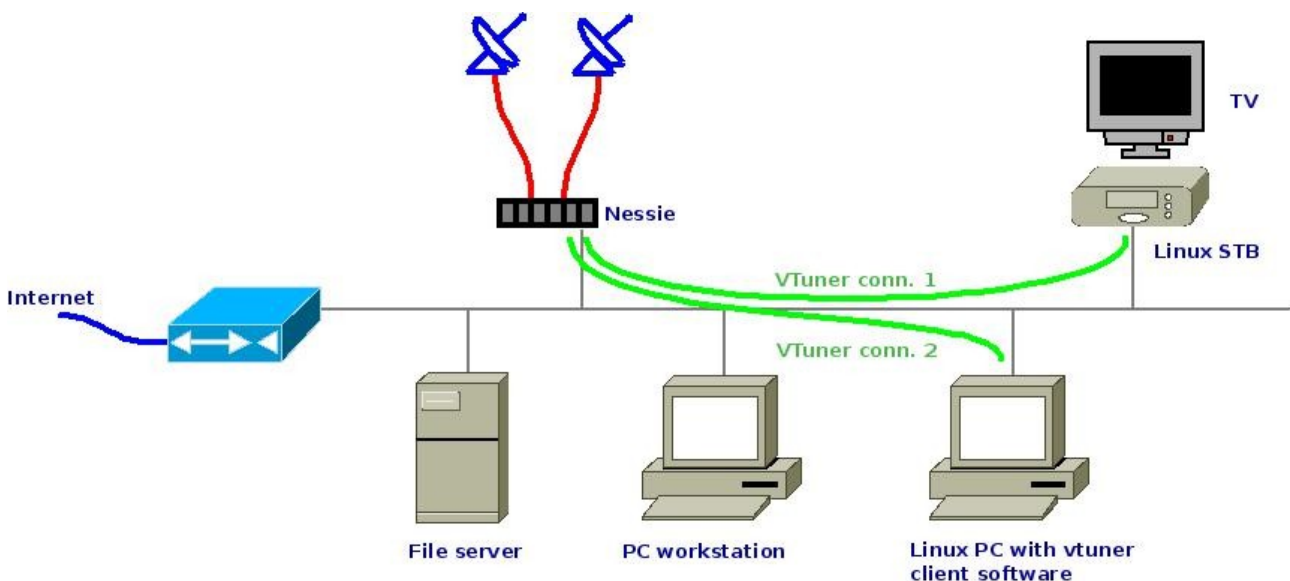
NessieDVB contains two tuners for satellite signal reception in the DVB-S and DVB-S2 standards. Each of the tuners can be configured separately to every supported mode. They will show up on the network as two separate devices behind the same network interface.

Functional description of the operating modes:

NessieDVB can be used in several modes. The functional usage varies on the end device type and mode respectively.

Virtual tuner mode (vtuner):

In this mode the NessieDVB tuner can be used as a virtual network tuner by a device that is either able to receive satellite DVB signal (e.g. Linux based receivers - set-top boxes) or with software programs that support this mode. In fact it is the implementation of the protocol and interface based on the interface implemented by Dream Multimedia in the DreamBox 800 a 8000 family of devices and that is (or will be) supported in other Linux based satellite receivers using Enigma II HD as the user interface.



Drawing 1: Connection of NessieDVB in 2x Vtuner mode

In this mode the configured tuner will behave against the end devices as if it was a local tuner on the device. It is thus possible to add a new tuner to the device (e.g. for recording of other

than the actually watched channel) or to leverage only the connected virtual tuner for instance because of the impossibility to interconnect the receiver with the satellite dish.

As the receiving end device also a Linux PC can be used with an experimental driver of virtual network tuner. The driver is available for the Linux operating system (kernel version $\geq 2.6.30$, lower versions haven't been tested, moreover it's essential to have DVB-API ≥ 5.0 , i.e. non-patched kernel lower than 2.6.28 will not work, it's necessary to patch) and during the operation a device emulating a local DVB receiver will show up (that is a complete structure of `/dev/dvb/adapter` will be created) and it's possible to utilize in connection with any software program dedicated for dvb devices (kaffeine, vdr, mythtv, xbmc etc.).

In this case the particular tuner is selected automatically from all the devices connected to the local network.

After tuner selection this particular tuner will be unavailable for all other end devices. On the dedicated device however it's possible to provide all the operations as if it was a local tuner. This is the main difference to all other configured modes (Multicast, DLNA).

Software installation for the vtuner mode in the Linux PC

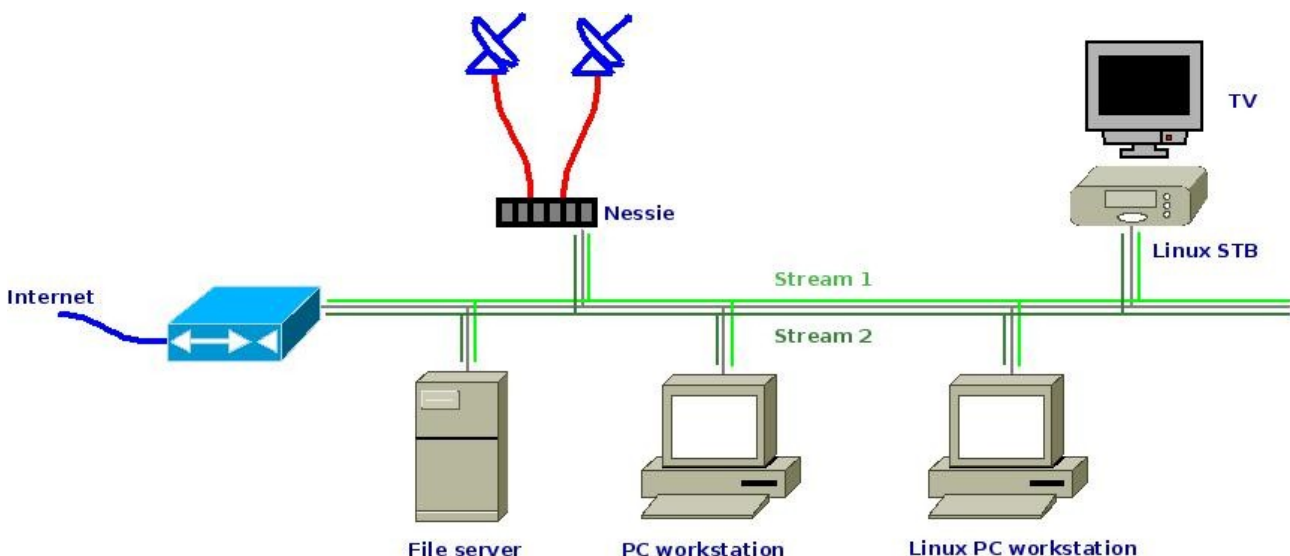
The required software for the vtuner mode consists of two parts. First is the driver part creating in `/dev/` the appropriate (required) number of dvb adapters and `vtunerX` devices (one for each created adapter) and the second part is the client software. The client software is a standard executable program that is necessary to launch for each required adapter separately.

After launching the client it connects to NessieDVB using the required mode (DVB-S/S2) and after opening the device `/dev/adapterX/frontend0` the client application launches a data and control network connection to Nessi. This connection is terminated after the termination of the client not after the closing of the dvb application, thus if the sharing of NessieDVB amongst more computers is needed, it's necessary to arrange it either manually or using a launching script.

For detail about software installation for vtuner mode refer to chapter "Software installation on a Linux PC for vtuner mode".

Multicasted streaming mode:

This mode is dedicated to broadcasting of one or more particular channels from a given transponder (i.e. broadcasting frequency) to the local network. On the local network this stream can be received by any number of network devices like a personal computer, tablet or Smartphone.



Drawing 2: Connecting of NessieDVB in 2x multicasted stream

As a receiving software for such a stream the typical vlc (<http://www.videolan.org/vlc/>) application can be used, that is available practically for any commonly used operation systems and also is part of the program equipment of some of the modern televisions. The stream is fetched to the network input of all devices connected to the local network and the receiving program can decide which stream to process (display, store to disk and so).

DLNA mode

DLNA is the abbreviation for Digital Living Network Alliance. It's a kind of standardized network protocol where devices configured as DLNA servers offer to the local network a list of multimedia streams that they are able to provide (video, audio, images etc.).

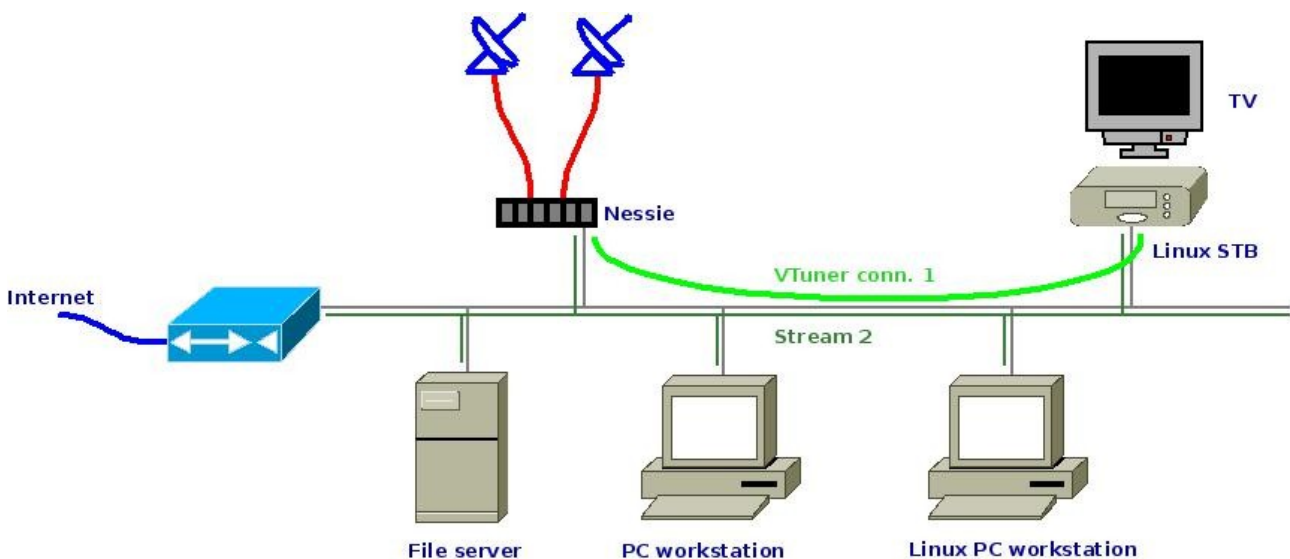
DLNA devices that are able to process this content (play, display) are in a mode of DLNA client. They listen to the local offerings of all DLNA servers and can display the offering of all multimedia content that are able to play/display.

In this mode NessieDVB will behave like a DLNA server with the difference to standard DLNA server, that it does not offer the play/display of static content but the preconfigured live DVB TV and Radio channels from the satellite.

As far as the network connection type is concerned it's equivalent to the vTuner mode - see Drawing 1: Connection of NessieDVB in 2x Vtuner mode.

Combined Vtuner or DLNA and multicasted streaming mode

This mode is in fact the combination of the previous modes when each of the tuners is configured to a different mode and we name it only as an example of the possibilities of setting up NessieDVB.



Drawing 3: Connecting NessieDVB in 1x Vtuner or DLNA and 1x multicasted streaming mode

Network infrastructure requirements for connecting NessieDVB to the network

NessieDVB has a 10/100 ethernet port with automatic polarity and parity switching for automatic configuration against the connected cable (MDIX) and connected port.

It's necessary to realize that ethernet is a bus system. This is partially eliminated by using switches but if there are multicasted packets in the system they are directed to every port of the switch. That means they're present on all segments of the local network and thus the access on the internal bus of the switch is controlled by means of CSMA/CD as if a HUB was used.

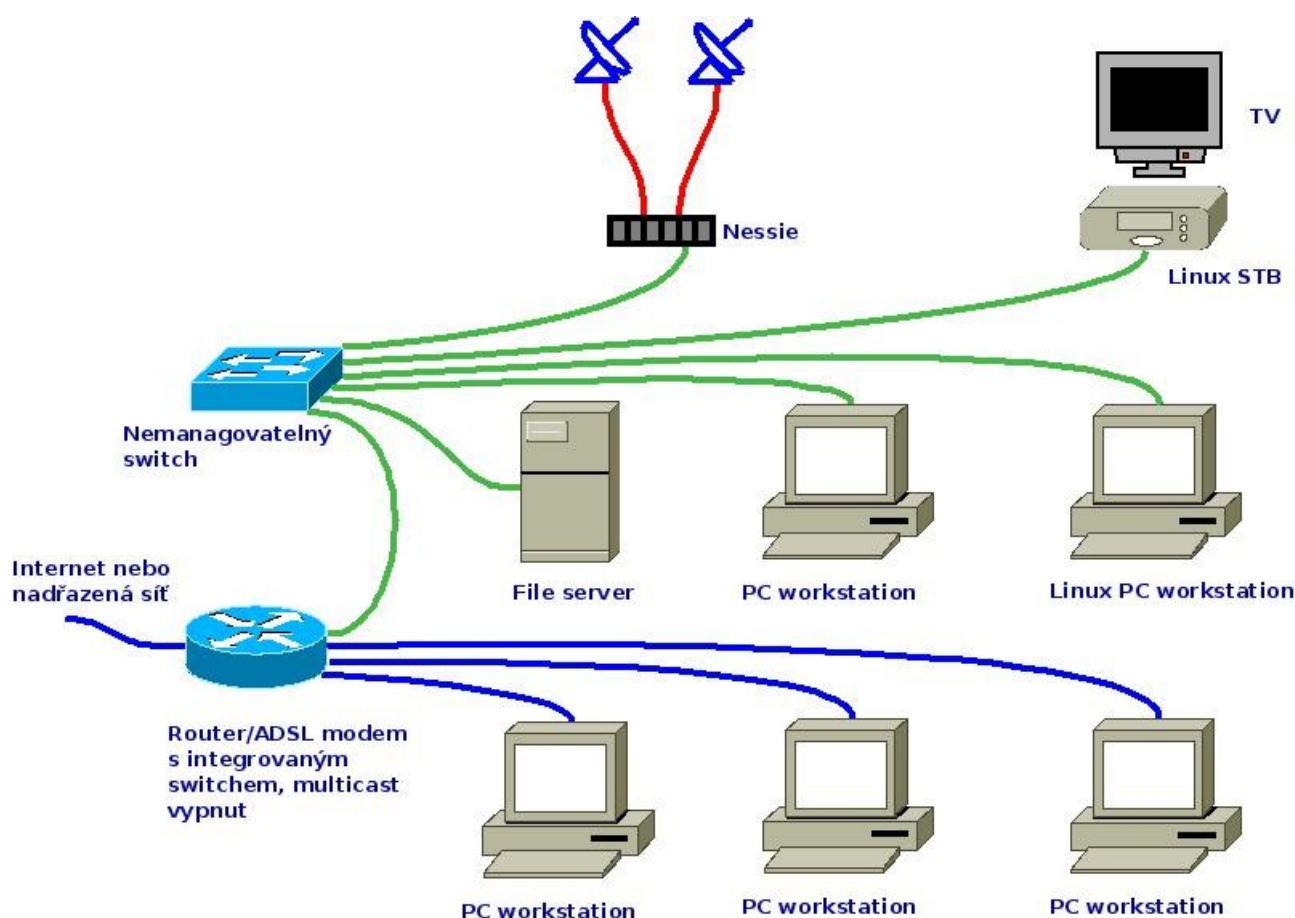
This must be considered when Nessie has to be used in multicasted streaming mode on higher bitrates. NessieDVB can namely supply to the network multicasted streaming up to 40Mbps and this can - on cheaper managed switches, routers and modems with integrated switches (e.g. Zyxel Prestige 334, Zyxel P661H a such) - cause their congestion.

For these purposes the cheaper unmanaged 1Gbps switches (e.g.: ZyXEL GS-105B, 108B and such) has shown during practical tests as the best; or professional switches..

If the local network is connected to the Internet or any other superior network using a cheaper router/modem with the possibility to control multicasted addresses than it's useful to switch off multicast on these devices and the network segment - where multicasted traffic will be used - connect using an unmanaged 1Gbps switch. This solution seems to be at the moment fully functionally sufficient and minimally demanding - by financial means.

The following drawing will illustrate the schematical connection of such a network structure. Multicasted network segments are drawn in green color, network segments without multicasted packets generated by Nessie are blue.

On all segments it's possible to use vtuner or DLNA mode without any limitation.



Drawing 4: Typical structure of network infrastructure in home and SOHO circumstances

NessieDVB setup:

NessieDVB is set up and configured using a web user interface. If during the configuration of NessieDVB by mistake an invalid configuration is set up and approved and it's not possible to connect to the device's configuration interface, by pressing the device's reset button (below a small hole on the front panel) for a period longer than 10s (use a pen, pencil or other slim tool); the factory default values are set up again.

Factory default set-up:

The factory default configuration of NessieDVB is DHCP mode. If there's no DHCP server present on the local network (or the network cable is not connected to NessieDVB) then after one minute the 192.168.1.11 address, with netmask 255.255.255.0 is set-up for static configuration with the given values.

Status:

After connecting to the correct IP address of NessieDVB the following web page will be shown:

The screenshot shows a Mozilla Firefox browser window with the address bar displaying `http://10.0.1.197/index.html`. The page title is `:: NETWORK | relativity ::`. The main content area has a yellow background with a floral pattern and the text **STATUS NETWORK TUNERS LOGS SUPPORT** at the top. Below this, there is a status bar showing `Up: 00:07 * IP address: 10.0.1.197 * MAC: 00:50:c2:09:d7:01`. The page is divided into three main sections: **SYSTEM**, **NETWORK**, and **TUNER 1**, followed by **TUNER 2**. Each section contains a table of device information and a checkbox for `Enable auto-refresh`. The **SYSTEM** section shows the model as `NessieDVB 2xS2`, serial number as `000000DEADBEEF01`, hardware firmware as `NESSIE version HW=1.1, FPGA=9.5`, and software version as `Pingu5.14-1 2011-03-10 (test beta)`. The **NETWORK** section shows the hostname as `nessie`, IP address as `10.0.1.197`, subnet mask as `255.255.255.0`, gateway as `10.0.1.1`, and traffic as `3.928 kb/s` with a progress bar at 1%. The **TUNER 1** and **TUNER 2** sections both show the type as `Satellite DVB-S2 NIM (Sharp BS2F7HZ0169)` and mode as `disabled`. The **NessieDVB** logo is visible in the bottom right corner of the page. The browser's status bar at the bottom shows `Done`.

| SYSTEM | |
|---------------|------------------------------------|
| Model | NessieDVB 2xS2 |
| Serial Number | 000000DEADBEEF01 |
| HW Firmware | NESSIE version HW=1.1, FPGA=9.5 |
| SW Version | Pingu5.14-1 2011-03-10 (test beta) |

| NETWORK | |
|-------------|---------------|
| Hostname | nessie |
| IP Address | 10.0.1.197 |
| Subnet Mask | 255.255.255.0 |
| Gateway | 10.0.1.1 |
| Traffic | 3.928 kb/s |

| TUNER 1 | |
|---------|--|
| Type | Satellite DVB-S2 NIM (Sharp BS2F7HZ0169) |
| Mode | disabled |

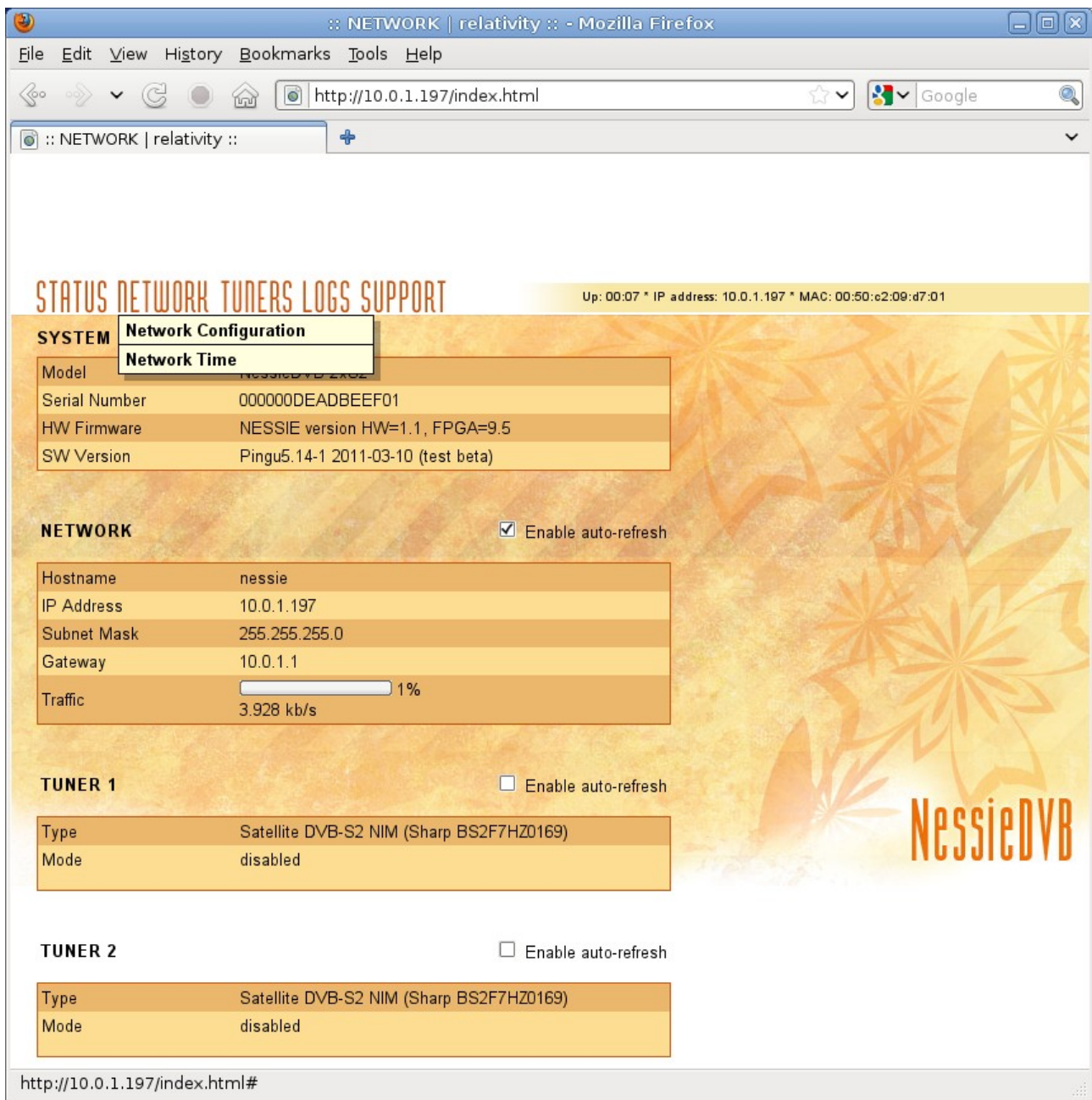
| TUNER 2 | |
|---------|--|
| Type | Satellite DVB-S2 NIM (Sharp BS2F7HZ0169) |
| Mode | disabled |

Drawing 5: Web Interface - Status

On this initial page the basic overview of the device set-up and device-information (version of hardware, firmware and software shipped) are available. Further the network adapter status information, current bit-rate speed of communication on the network interface is displayed. Also information about internal tuners and their set-up and mode is shown; and after enabling the refreshing on the adapter also the signal quality on the given adapter is displayed. The factory default mode for both tuners is "disabled".

Network set-up:

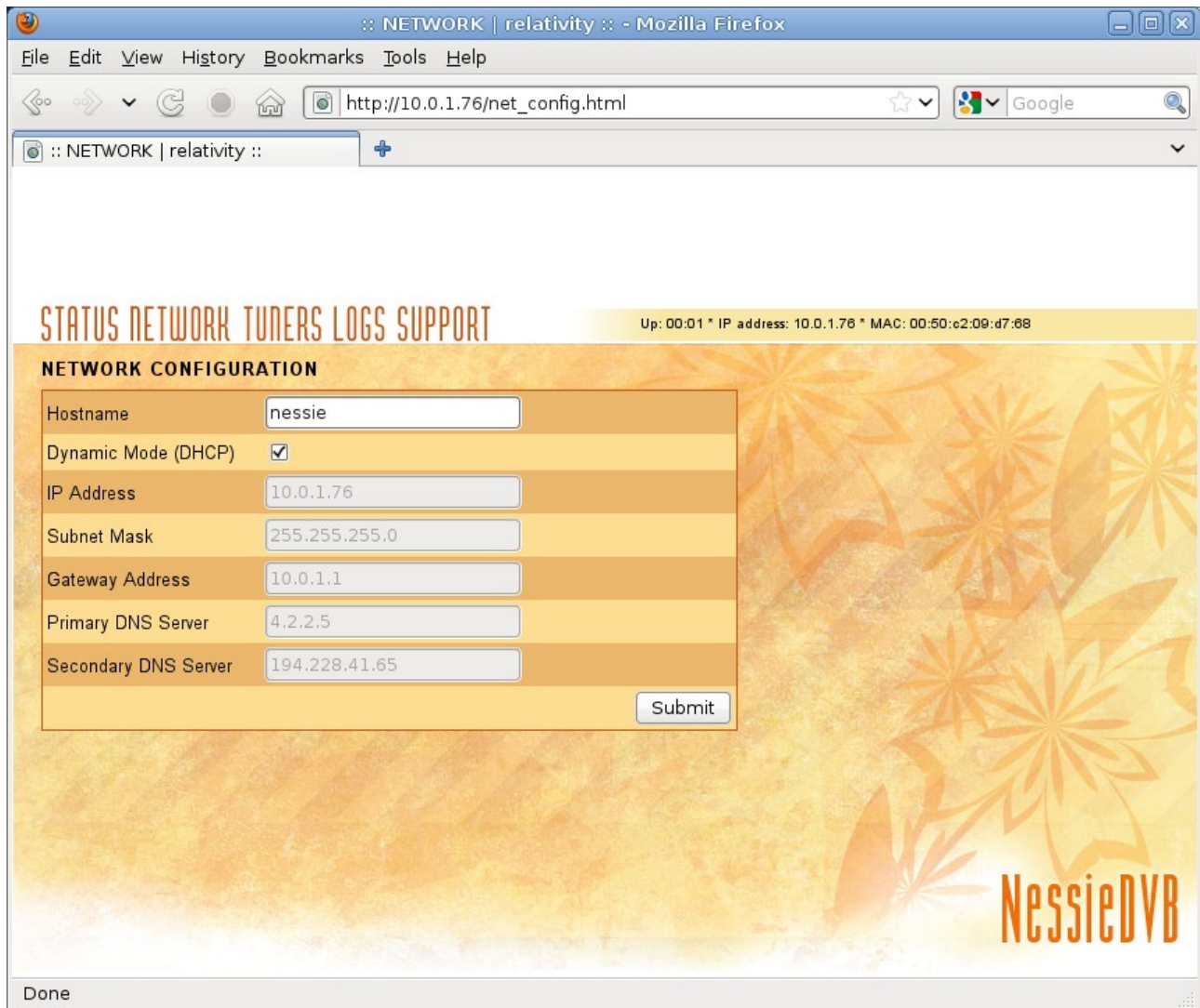
The NETWORK submenu is displayed by placing the mouse over the NETWORK menu point:



Drawing 6: Web Interface - Network Submenu

Network Configuration

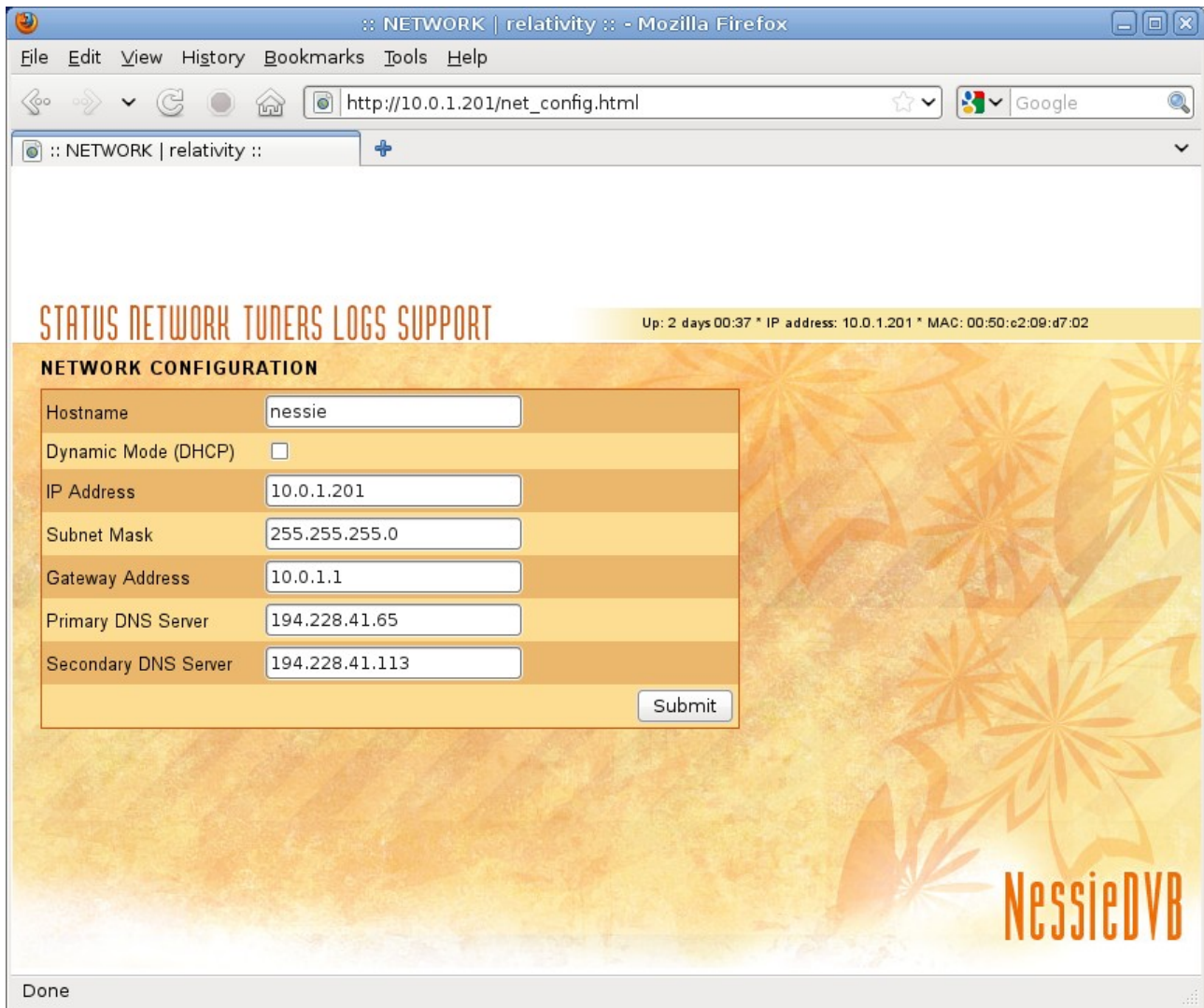
After choosing the "Network Configuration" item the complete network interface setup screen is displayed. The factory default network configuration mode of the device is DHCP, so the actual IP address of the device can be found on the DHCP server using the MAC address of NessieDVB (on the production label).



Drawing 7: Web Interface - Network Default Configuration

IF NessieDVB does not find a DHCP server it will set up the 192.168.1.11 IP address with, netmask 255.255.255.0. It can be used if the device's IP address cannot be found out from the DHCP server (e.g. because the device was booted without a connected ethernet cable and it's connected more after both LED diodes on the front panel turn green, or even later).

If we want to use NessieDVB with a static IP address here we can change it – as described in the following chapter – to suit our needs.



Drawing 8: Web Interface - Network Configuration – Fixed IP

If the DHCP mode is unchecked, the complete IP configuration can be set up manually. In this form the NessieDVB's network setup should be modified so that it would fit Your network's configuration; i.e. assign a free IP address to the device set up the netmask and the default gateway.

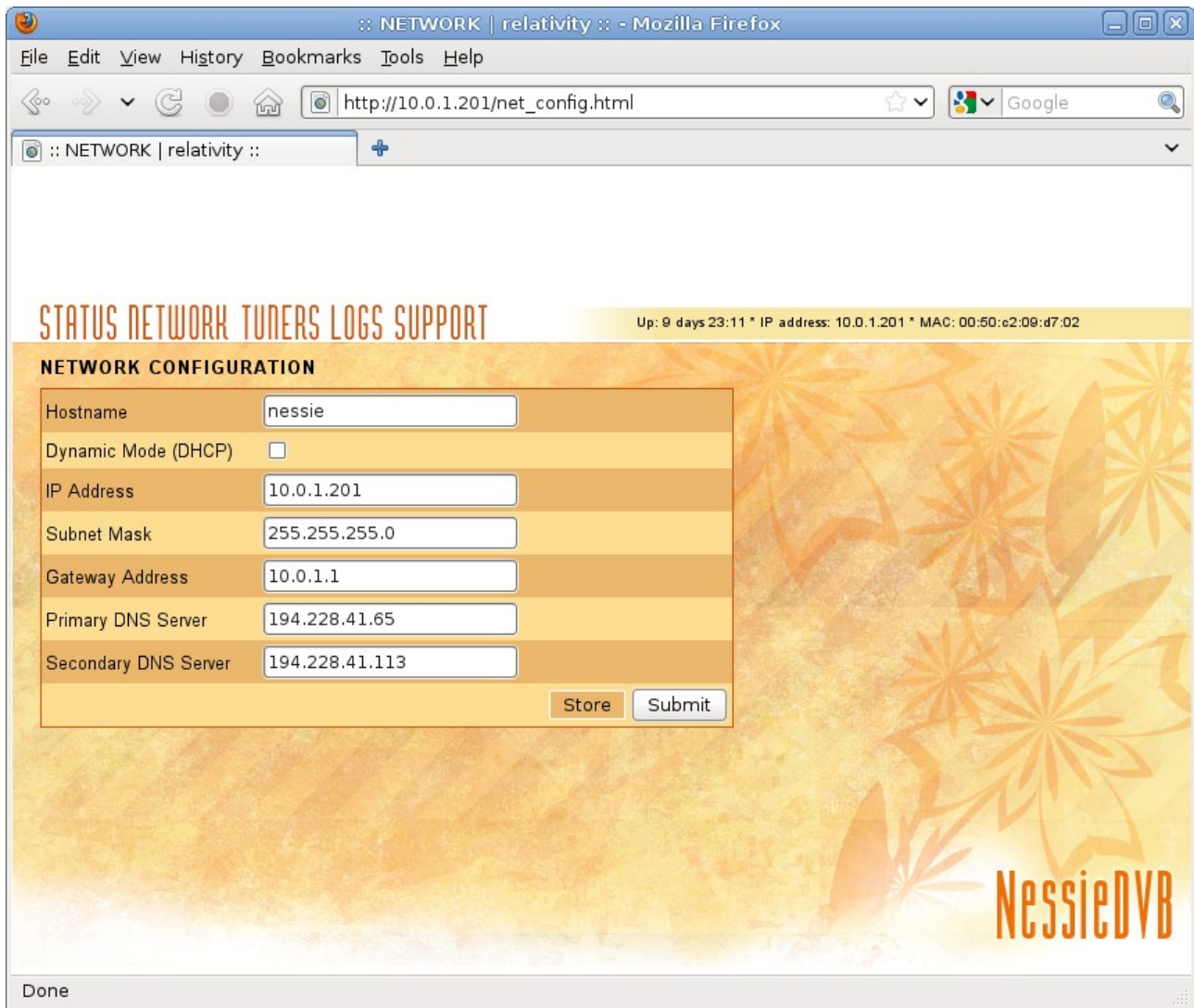
In this phase there can be a problem with the DNS server setup – if You're not sure about the IP address of Your local DNS server or the DNS server of our provider, we can use an address of the free public DNS servers, e.g. from: <http://theos.in/windows-xp/free-fast-public-dns-server-list/> - these DNS servers are in most cases even faster than the DNS server of the service providers. The local DNS server address can be found most often in the superior router setup, ADSL or cable modem setup. The DNS server address must be entered in the same octet format XX.XX.XX.XX , like other fields in this setup form, i.e. it's not possible to use symbolic names.

This setup is not mandatory for the device function itself, but if we intend to use the direct update of satellite positions, the setup is required.

Using the "Submit" button the parameters can be saved to the device and they will be used immediately, but they are not saved yet to the permanent memory.

In this state, if the new entered network configuration is incorrect, the device reboot restore last permanent configuration.

By this the network configuration can be tested without the permanent storage.



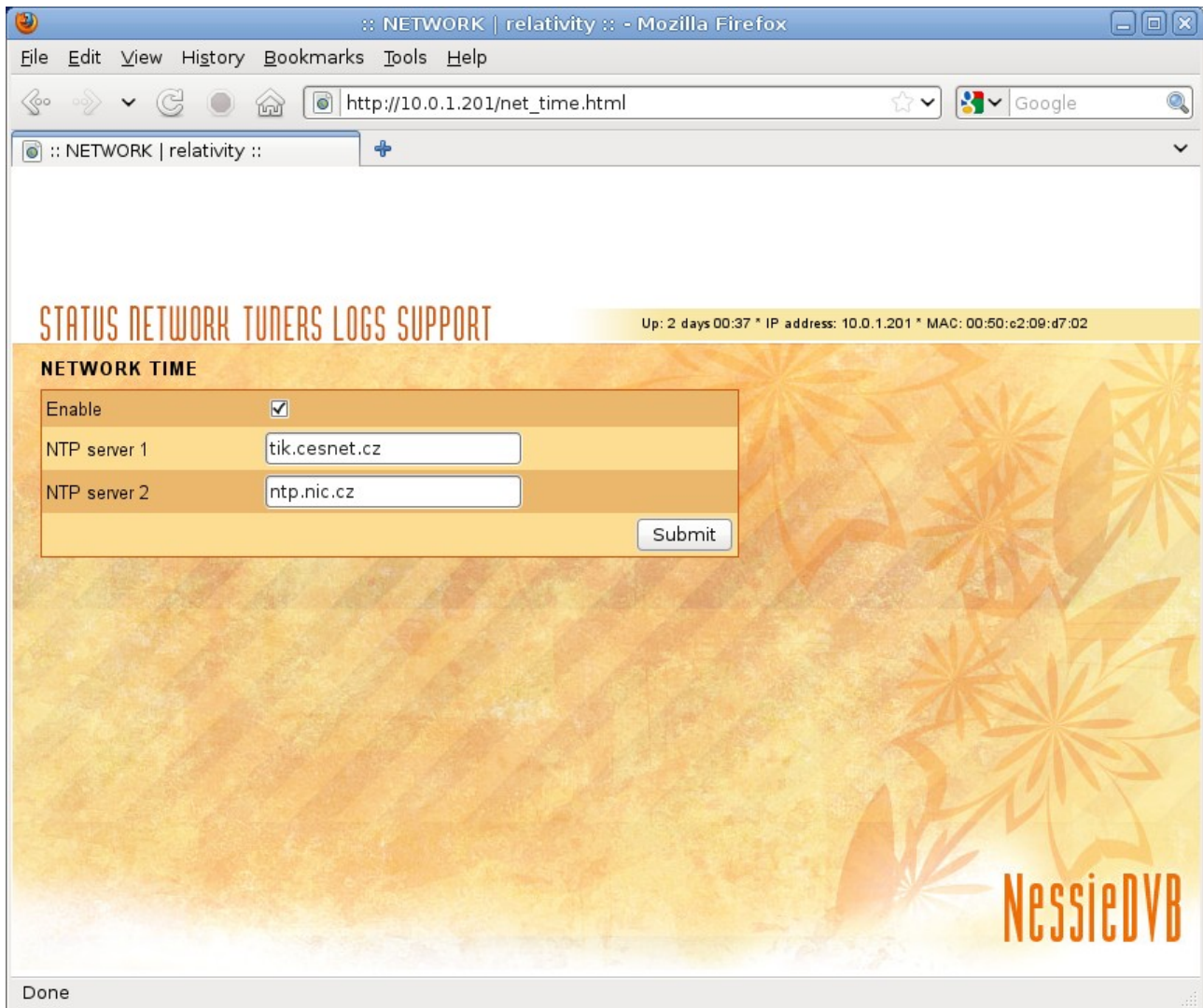
Drawing 9: Web Interface - Network Configuration – saving of set values

After submitting the configuration with the "Store" button the network setup is stored into the permanent memory and thus even after reboot will be used instead of the factory default settings or the previous configuration.

Network Time

The local time of NessieDVB can be set and synchronized using the ntp protocol from any of the internet time servers. This however is not mandatory for the correct functioning of the device.

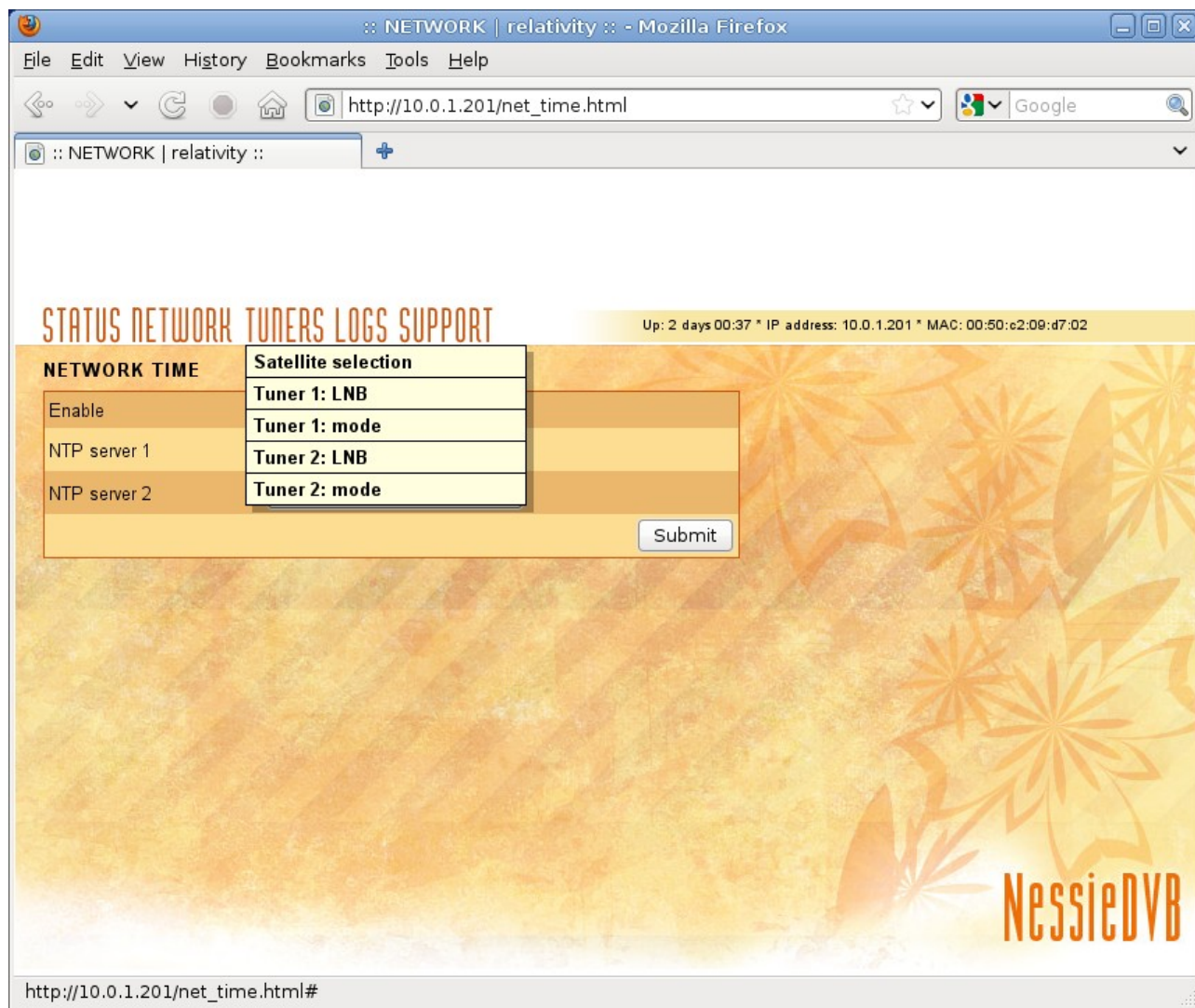
Without using NTP the system time will be counted from 1.1.2010 since reboot.



Drawing 10: Web Interface - Network Time

Tuners setup

After passing the mouse over the "TUNERS" menu a submenu with five items is displayed. Using these menu items can be configured the LNB (DiSEqC) of both tuners and their operating mode..

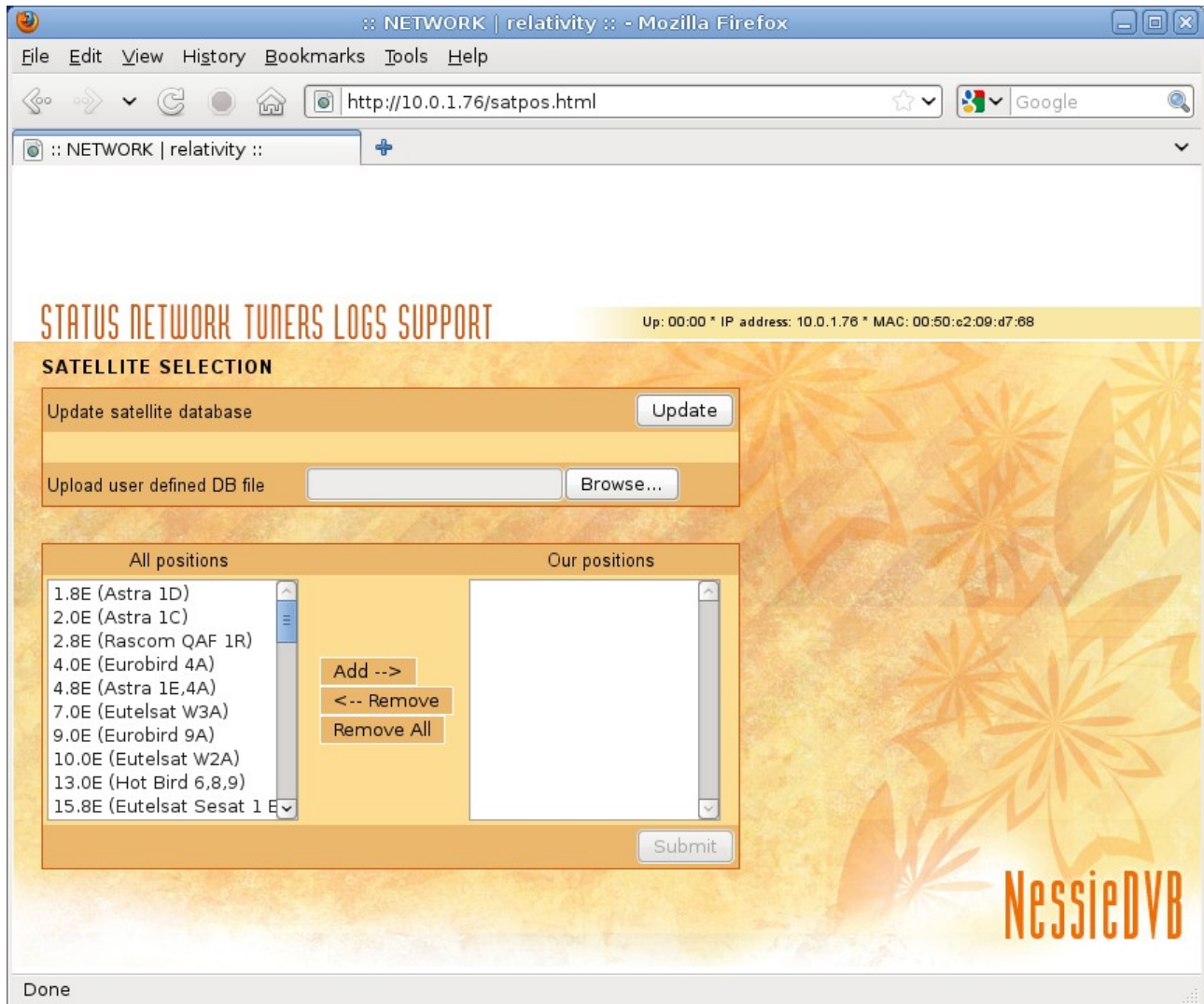


Drawing 11: Web Interface - Tuners Submenu

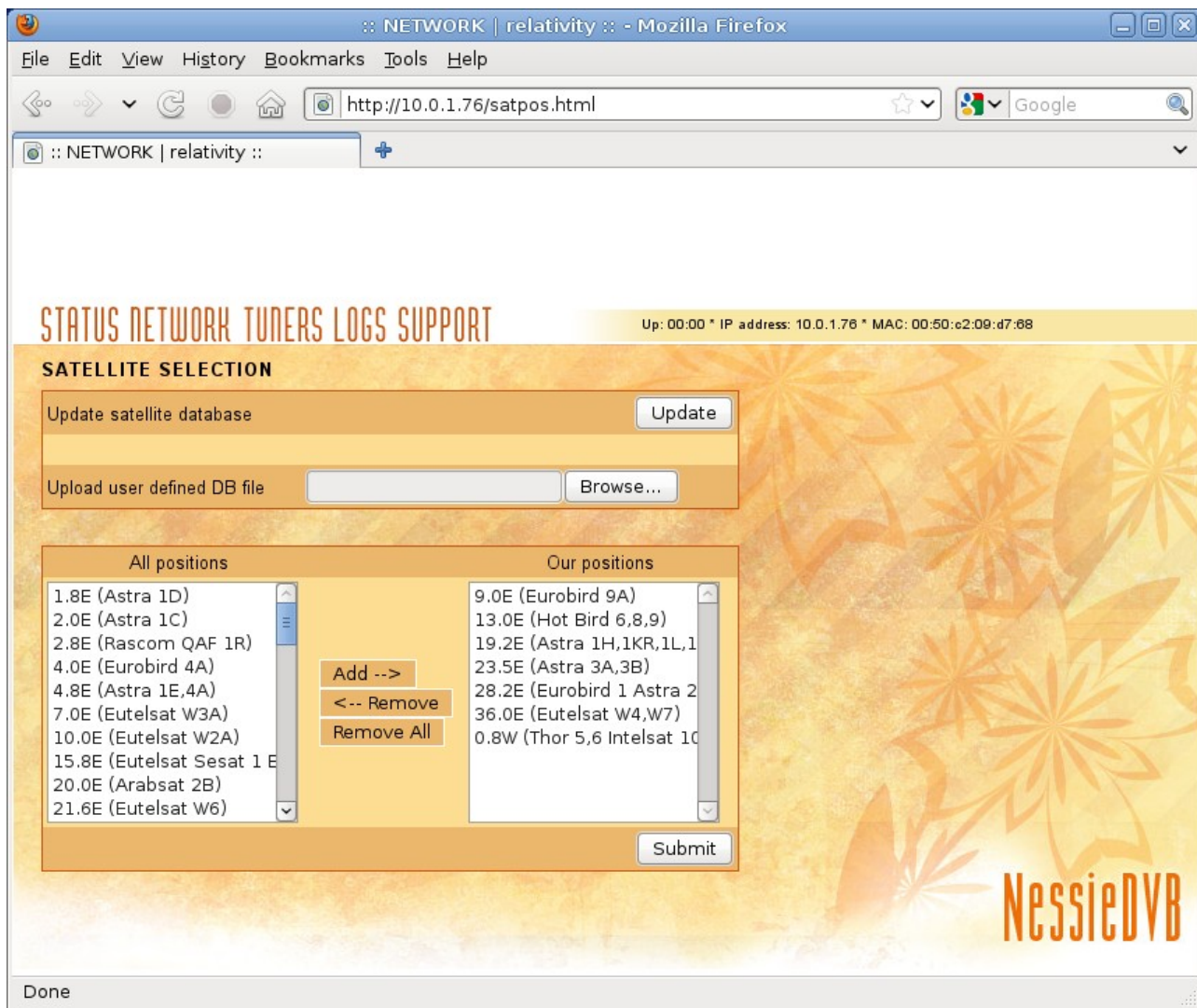
After setting up the tuners - according to the chosen combination of mode settings - a STATUS screen like on the following Drawing 30: Web Interface - Status after setup - both tuners in dvblast mode changes.

Satellite selection

This web interface screen is designated for selecting of satellite positions that we are able to receive. These selected satellite positions are then available in the following configurations of DiSEqC, so as the following screens are simpler and easier to read and configure.



Drawing 12: Web Interface - Satellite Selection



Drawing 13: Web Interface - Satellite Selection 2

The list of transponders and their parameters – from which the broadcasting takes place – are taken from standard definition file, so called ‘ini’ files.

Update of satellite definition (ini) files

The satellite definition files can be updated in two ways.

Update of all publicly available files

To complete this task the „Update“ button on the upper „Update satellite database“ field is used. Using this choice the definition file from the <http://www.fastsatfinder.com/bin/Transponders.zip> URL is downloaded and used. This file contains the definition ‘ini’ files for all DVB satellite positions (150 positions currently).

Update of single definition files

Using of this dialog it’s possible to transfer a single definition ‘ini’ file from the local computer to NessieDVB. By this it’s possible to update only some satellite position definitions, eventually alter or create some ‘ini’ files and use them in NessieDVB.

Using the dialog screen the satellites displayed in the further LNB configurations are selected.

Satellite definition ,ini' file format

The satellite definition ini file is a text file that can be edited eventually created by any standard text editor. It is in the pure text format so it can be modified using any text editor. The filename is uniquely defining the satellite position on the geostationary orbit, namely in **tenth of angular degree of eastern length**, eventually padded by zeros for four character positions. Examples for the particular quadrants:

| Satellite | Position | File |
|----------------|-------------------|----------|
| Eurobird 9A | 9.0°E | 0090.ini |
| Koreasat 3 | 116.0°E | 1160.ini |
| Galaxy 19 | 97.0°W (=263.0°E) | 2630.ini |
| Hispasat 1C,1D | 30.0°W (=330.0°E) | 3300.ini |

The satellite definition file consists of a number of sections, where the name of the section is specified in the beginning of the section in square brackets.

On the following example (0090.ini) the sections that are important for processing by Nessii are shown.:

```
[SATTYPE]
1=0090
2=Eurobird 9A

[DVB]
0=29
1=11727,V,27500,34
2=11747,H,27500,34
3=11785,H,27500,23,S2;8PSK
4=11804,V,27500,34
5=11823,H,27500,34
6=11843,V,27500,34
7=11861,H,27500,23,S2;8PSK
8=11881,V,27500,23,S2;8PSK
9=11900,H,27500,34
10=11919,V,27500,34
11=11938,H,27500,34
12=11958,V,27500,34
13=11977,H,27500,34
14=11996,V,27500,34
15=12015,H,27500,23,S2;8PSK
16=12034,V,27500,34,S2;8PSK
17=12054,H,27500,23,S2;8PSK
18=12074,V,27500,34,S2;8PSK
19=12092,H,27500,34
20=12130,H,27500,34
21=12207,H,27500,23,S2;8PSK
22=12245,H,27500,34,S2;8PSK
23=12284,H,27500,23,S2;8PSK
24=12303,V,26400,23,S2;8PSK
25=12322,H,27500,34
26=12360,H,27500,34
27=12380,V,26400,23,S2;8PSK
28=12399,H,27500,34
29=12437,H,27500,34
```

In the SATTYPE section element 1 the satellite position is defined (the same way as it was defined in the definition file name) and the satellite name is in the element 2.

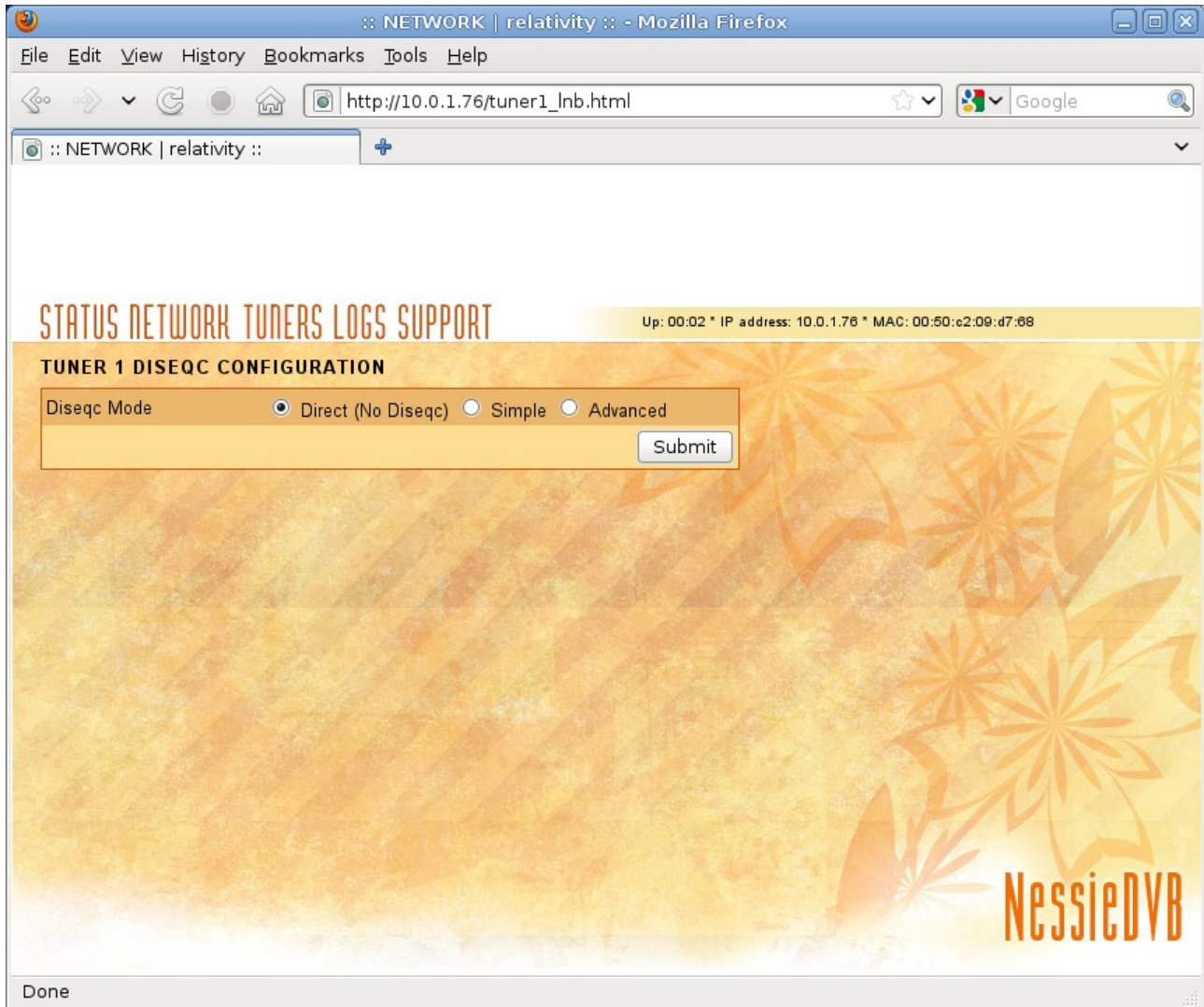
In the DVB section element 0 defines the number N of following transponders (in this case 29) and elements 1 to N define the particular transponders.

The transponder definition element consists of the following items, separated by comma:

- frequency (in MHz)
- polarization (vertical - V, horizontal -H, left rotation - L, right rotation - R)
- symbol rate (in KSps)
- FEC
- norm – optional item, default is DVB-S (can be defined also as S), if it is DVB-S2 it's defined as DVB-S2 or S2
- modulation (optional, default is QPSK, if it's different it must be specified – e.g. 8PSK, 16APSK). Can be divided from norm by semicolon

Tuner X LNB

In this menu the configuration of LNB connected to a particular tuner can be selected. It is required for Dvblast and DLNA modes but has no meaning for Vtuner mode.

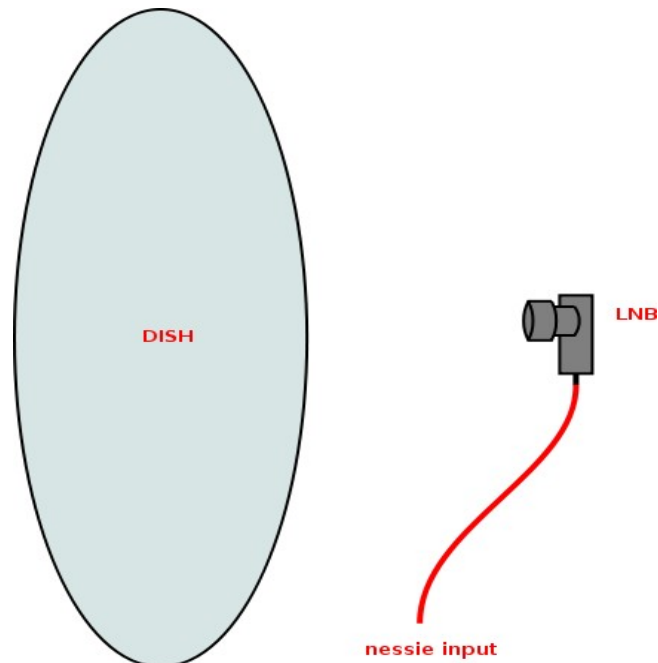


Drawing 14: Web Interface - Tuner LNB Setting – Direct

Three possibilities are available:

- Direct
- Simple
- Advanced

Direct (no DiSEqC)



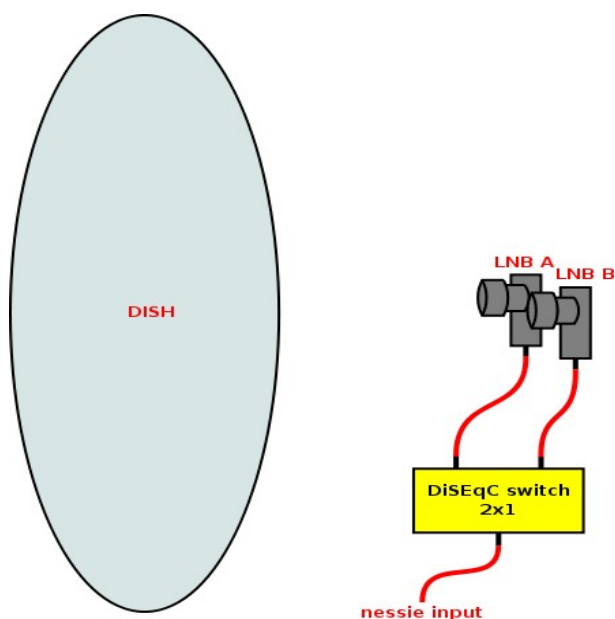
Drawing 15: Direct (no DiSEqC)

If we have got a simple LNB connected to the tuner with no DiSEqC switch at all we use "Direct".

The remaining two modes are for configuring the system with DiSEqC switches.

Simple DiSEqC

Using the simple mode two possibilities are available; with 2 or 4 inputs.

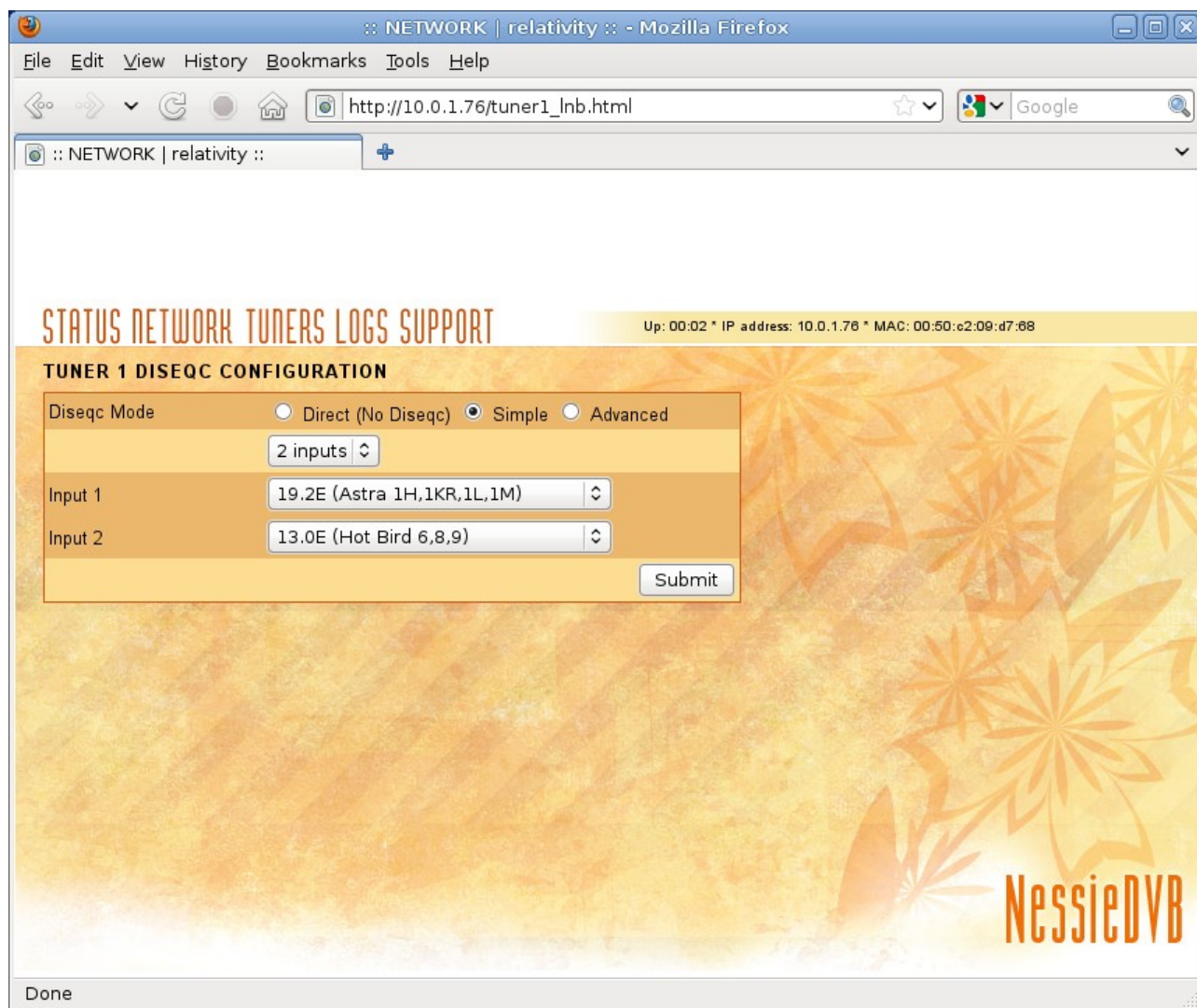


In case of two inputs the “burst” command is used for switching of the inputs (so called. minidisec).

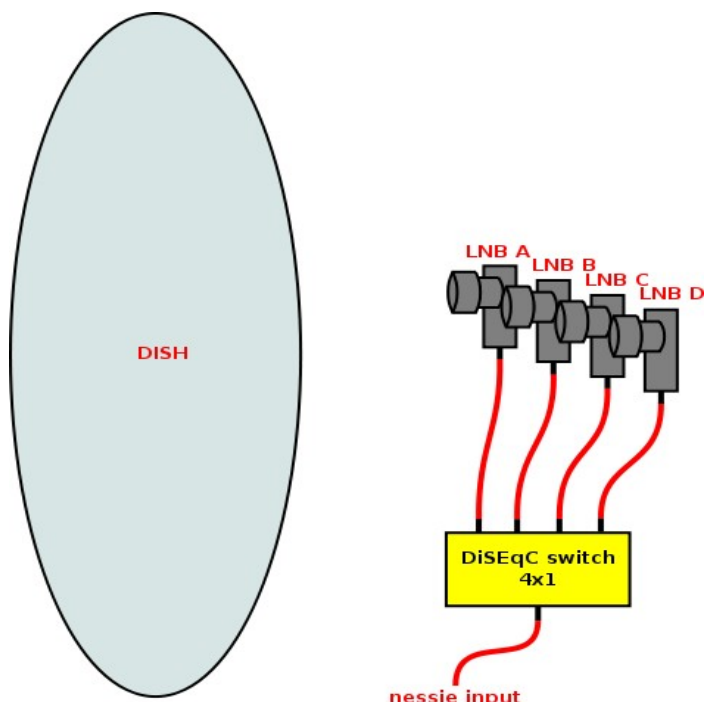
This mode is suitable for switching of DiSEqC switches integrated to double LNB converters (monoblocks) and for double input DiSEqC switches that support this mode.

The list at each input will contain the list of active satellite positions; from which the required satellite can be selected..

Drawing 16: Simple LNB DiSEqC – two input DiSEqC switch (command „burst“ used)



Drawing 17: Web Interface - Tuner LNB Setting – 2 Inputs Switch

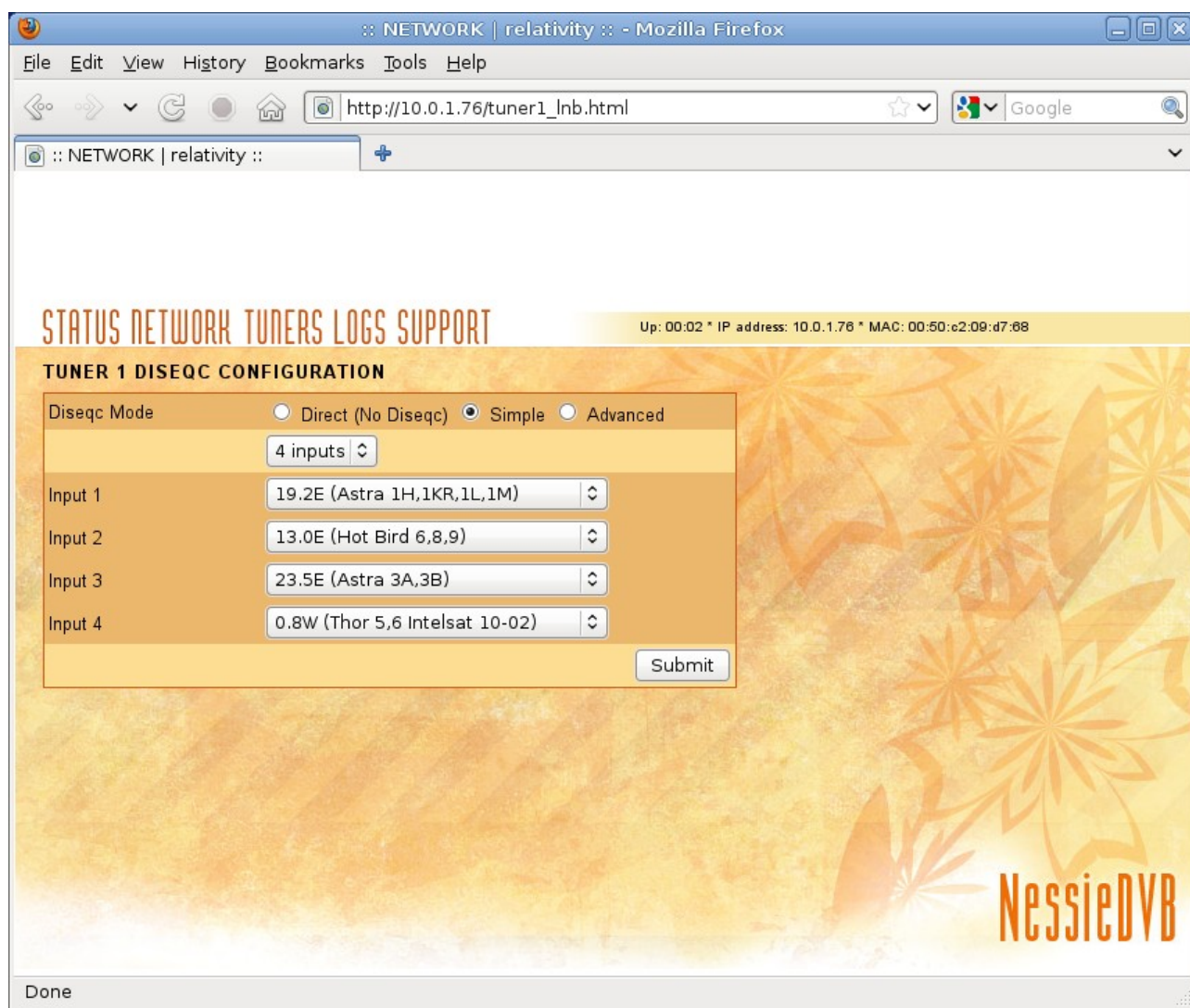


In case of selecting the 4 input mode the committed "position A - D" command (command - E0 10 38 Fx, that is transmitted 2 times) is used.

This mode is suitable for switching of four input DiSEqC switches that support this mode.

The listbox at the particular inputs will contain the list of satellites (selected in the previous screens) from which the required satellite can be selected for each input.

Drawing 18: Simple LNB DiSEqC – 4 Input DiSEqC switch (command „committed“ used)



Drawing 19: Web Interface - Tuner LNB Setting – 4 Inputs Switch

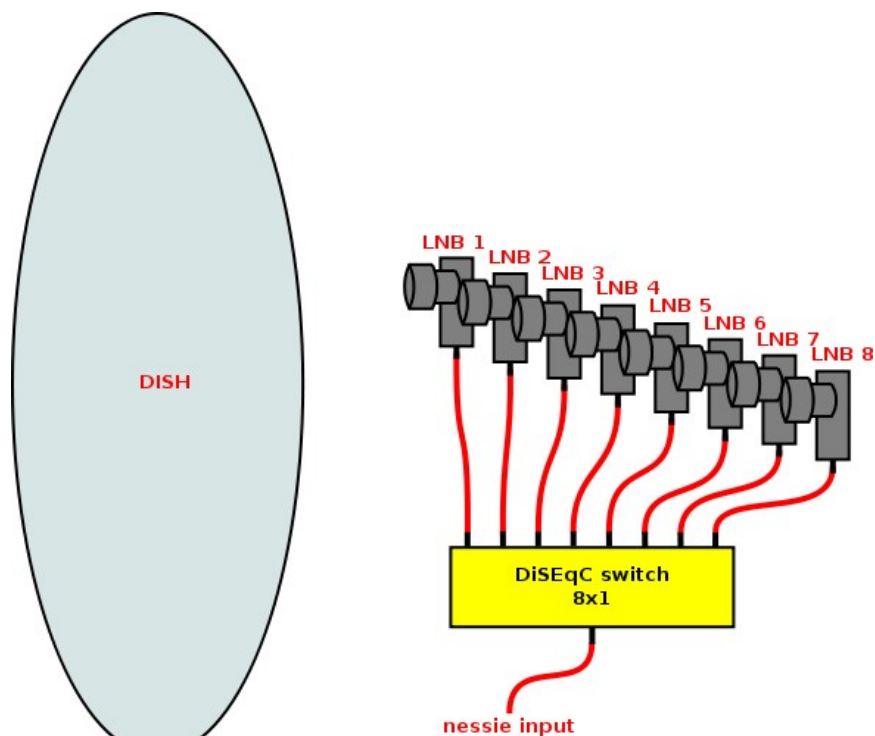
Advanced DiSEqC

This mode is for configuring of more sophisticated connection of LNBs and for the case of interconnecting of more converters and mode parabolic antennas eventually.

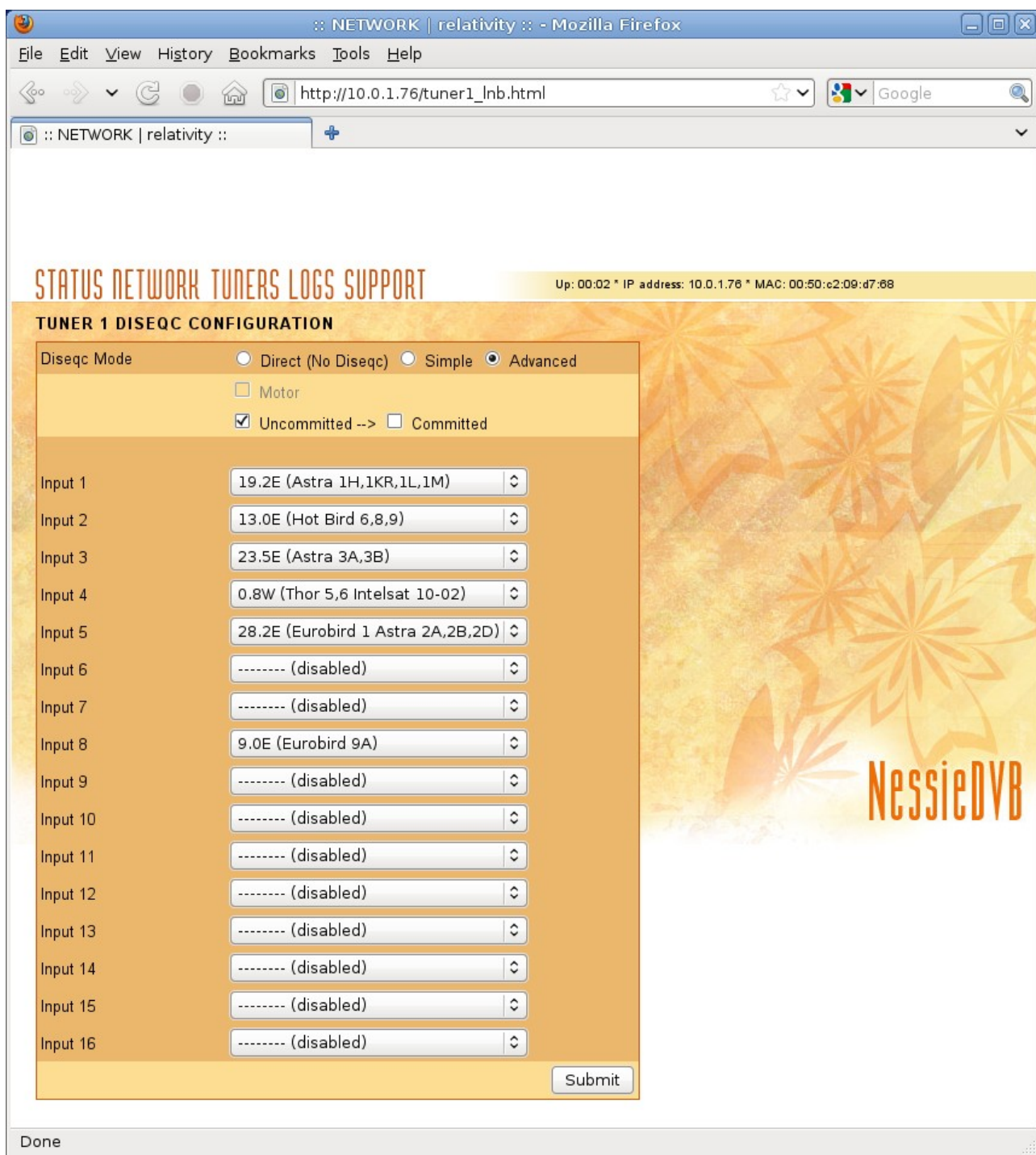
There are a number of options available. In this case the the particular menus does not block each other on the option “committed” is conditioned by a previous selection of “uncommitted” option. The following setup can thus be used:

- uncommitted – setting the switch to 'uncommitted' mode
- uncommitted with preceding 2 or 4 input 'committed' switch

Switch setup in 'uncommitted' mode



Drawing 20: Advanced DiSEqC – uncommitted switch

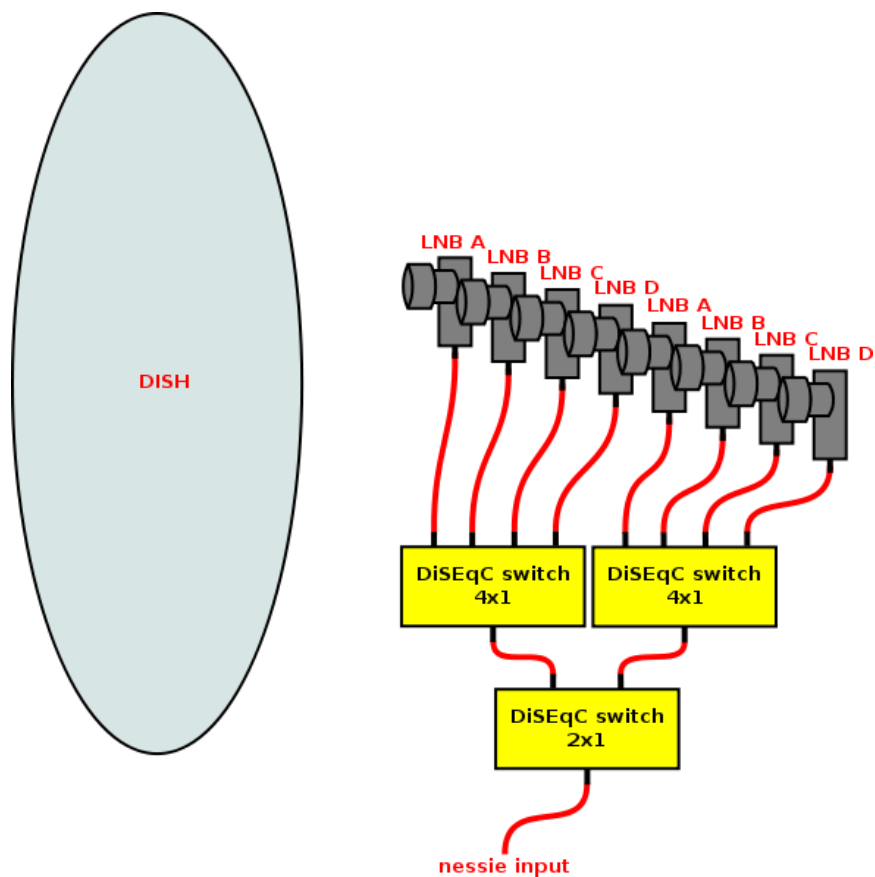


Drawing 21: Web Interface – Setup of uncommitted switch

In this mode uncommitted switches with 2, 4, 8 or 16 inputs can be set up. We assign an LNB tuned to an orbital position to each input.

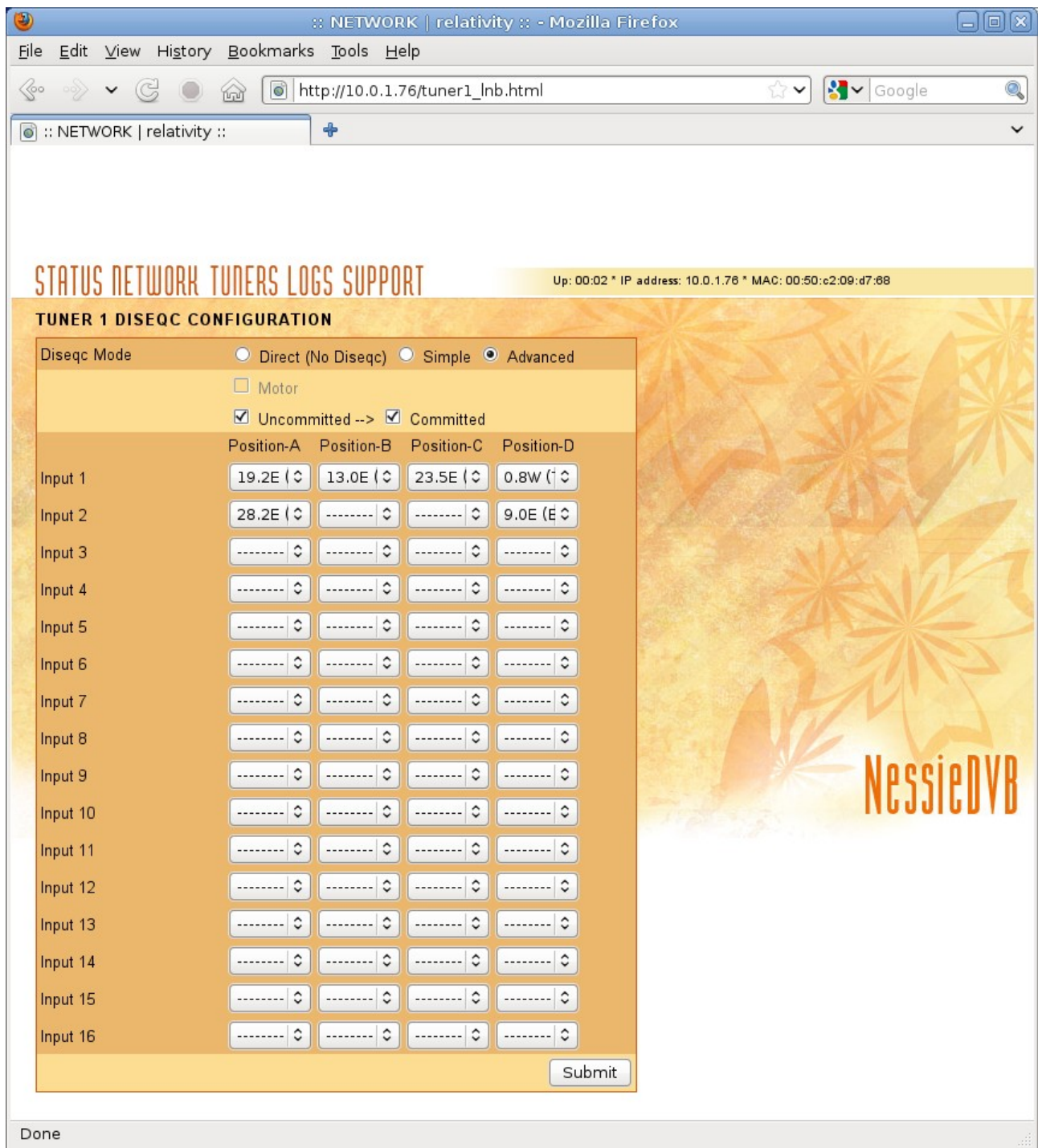
Switches of 'uncommitted' (option) type are used for controlling of a higher number of LNB converters.

Setting up of cascaded 'uncommitted' a 'committed' switches



Drawing 22: Advanced DiSEqC – cascade of uncommitted and committed switches

If to the 'uncommitted' option we add the option 'committed'; the setup form layout is changed so that to each input of an 'uncommitted' switch inputs of preceding 'committed' switches are assigned. We fill out the part of the table that we have connected, e.g. on the drawing above the configuration of 2-input 'uncommitted' switch with 2 preceding 4-input switches. This configuration is probably one of the most common configurations, since it's not only used as a connection of 3 switches but also internally in some types of DiSEqC switches (e.g. one of the modes of the P168 switch from EMP-Centauri).



Drawing 23: Web Interface – Setting up of cascade of switches

It is also possible to configure most sophisticated interconnection of LNBs – e.g. 8-input 'uncommitted' switch, that will have superposed 2-input switches in front of some inputs (some of them can be realized also in form of monoblocks), some inputs can have 4-input switches in front of them and some can be connected directly to converters.

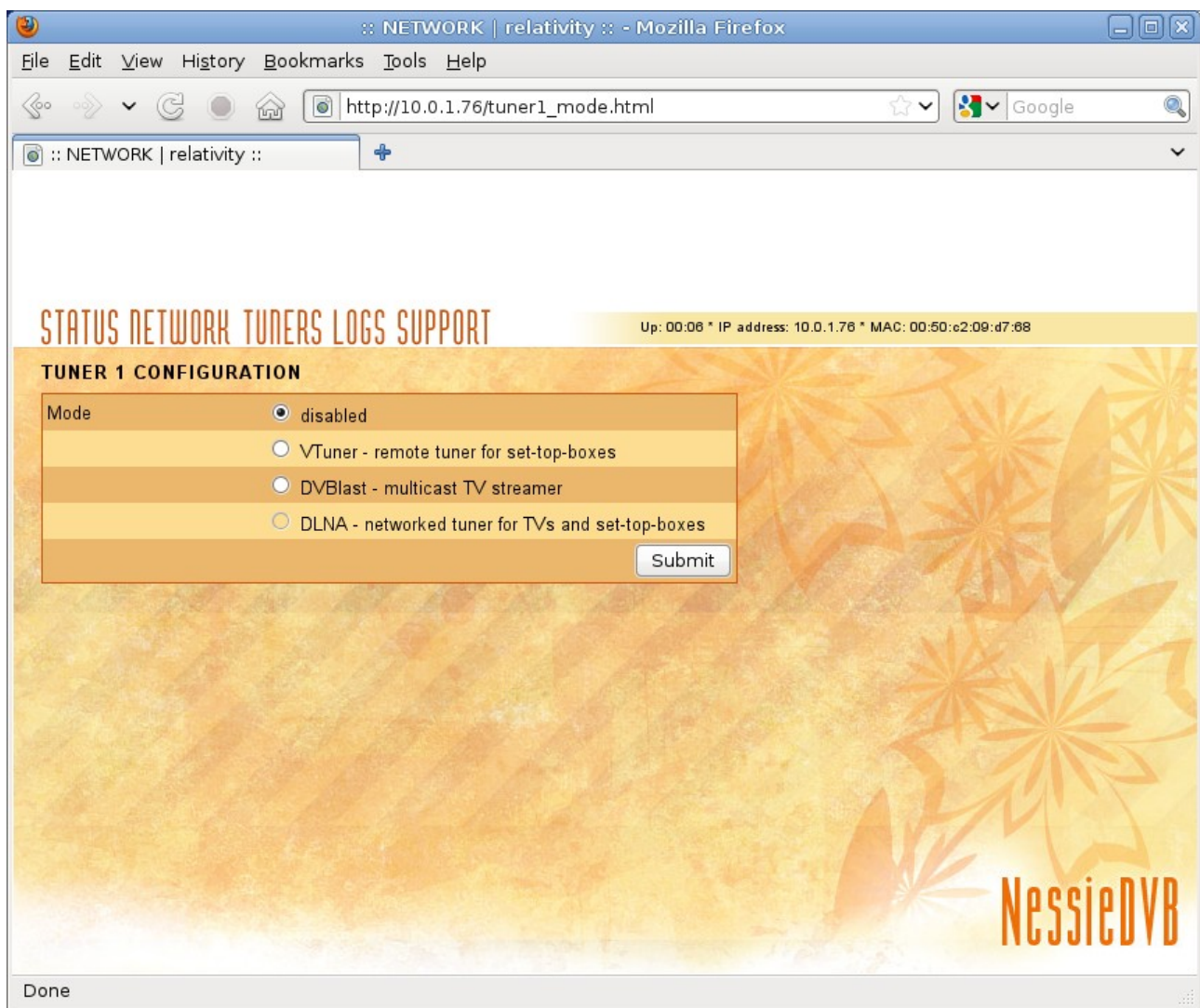
If the input is connected directly to the converter the corresponding position is selected in the first field.

Setting the operating mode of the tuners

In this screen the operating mode of particular tuners is set up. The following options are available:

- disabled
- vtuner server
- multicast server
- DLNA server

Every of the selected modes must be approved after selection by the „Submit“ button. In case of some modes the configuration continues on the further page.



Drawing 24: Web Interface – Setting up the operating mode of the tuners

Disabled

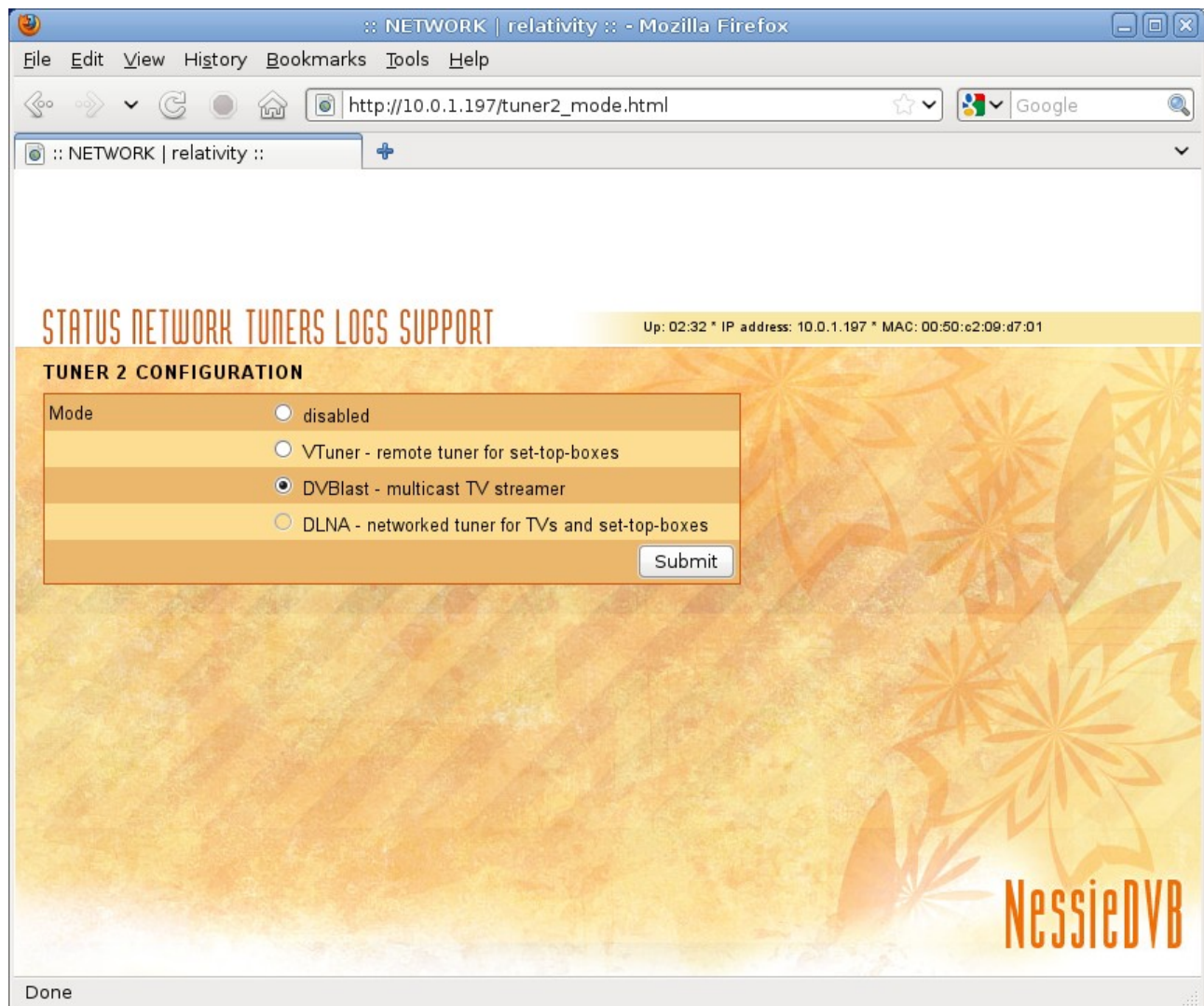
If the „disabled“ mode is selected the corresponding tuner will not be activated at all.

VTuner (Virtual tuner)

If we choose „vtuner“ than the selected tuner is switched to the mode of remote tuner for connecting to linux based STB or a computer with the „vtuner client“ software installed. In the current version for this mode there's nothing more to configure on NessieDVB because the basic version of vtuner daemon is used according to the publicly available source code published by Dream Multimedia (<http://code.google.com/p/dreamtuner/>).

In this mode the LNB setup as described previously (DiSEqC configuration) is not respected and full control over the tuner including the DiSEqC device is passed to the remote client (STB, PC).

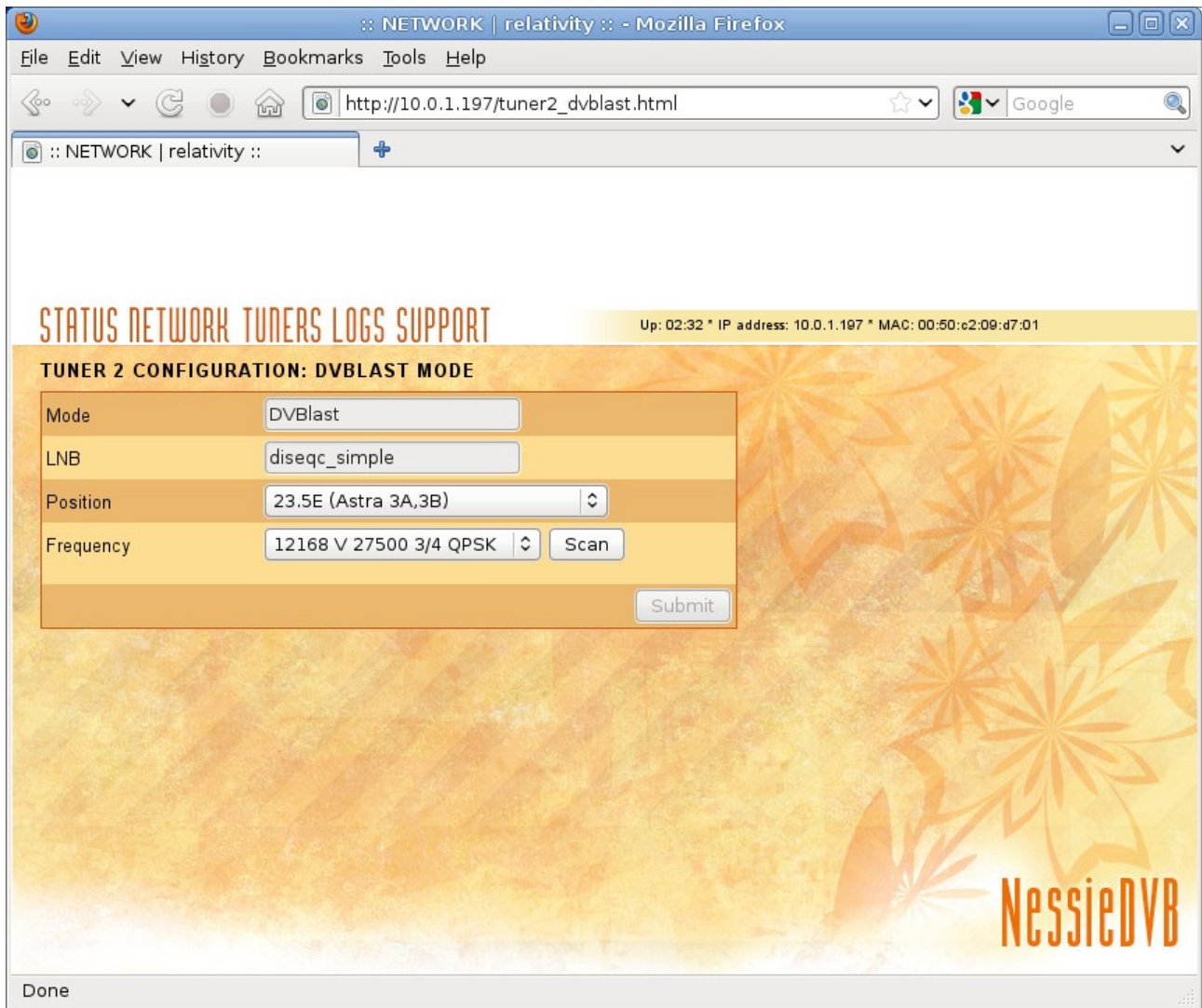
DVBlast - multicasting TV streamer (multicast server)



Drawing 25: Web Interface – DVBlast - Multicast TV Streamer

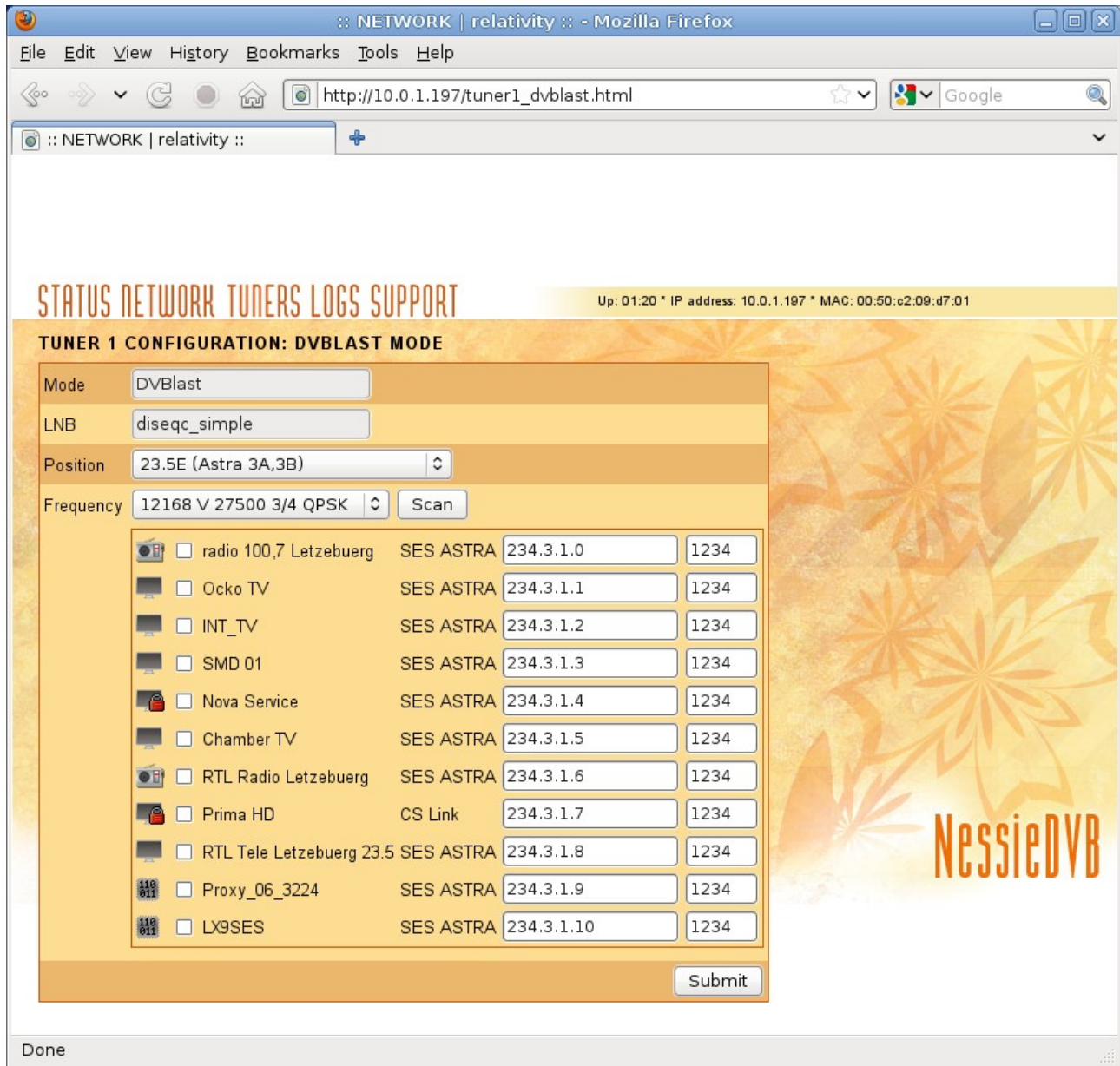
For streaming on multicasted addresses the DVBlast software is used that is available on the <http://www.videolan.org/projects/dvblast.html> .

After submitting of the selected mode the user is automatically redirected to the next page where the satellite and the transponder (broadcasting frequency) of the chosen program can be selected.



Drawing 26: Web Interface – DVBlast - Multicast TV Streamer 2

After submitting the selected satellite and transponder the “Scan” button is used to start channel search. NessieDVB then provides the survey of services broadcasted on the selected transponder and displays them in the next screen (this may take a while):



Drawing 27: Web Interface – DVblast - Multicast TV Streamer 3 - transponder scan

For each service a small icon is displayed stating if the service is a television broadcasting, radio station or data transmission. If the service is scrambled a red lock is displayed in the corner of the icon. Choose a service that you want to broadcast on the multicasted address, using the rtp protocol, to the local network and approve it by pressing the „Submit“ button. The broadcasting will start immediately after submit is pressed. It's necessary to realize however, that the broadcasted stream is not processed anyhow; so if the service on the satellite is scrambled (encrypted) it will be streamed to the local network in the same encrypted form too.

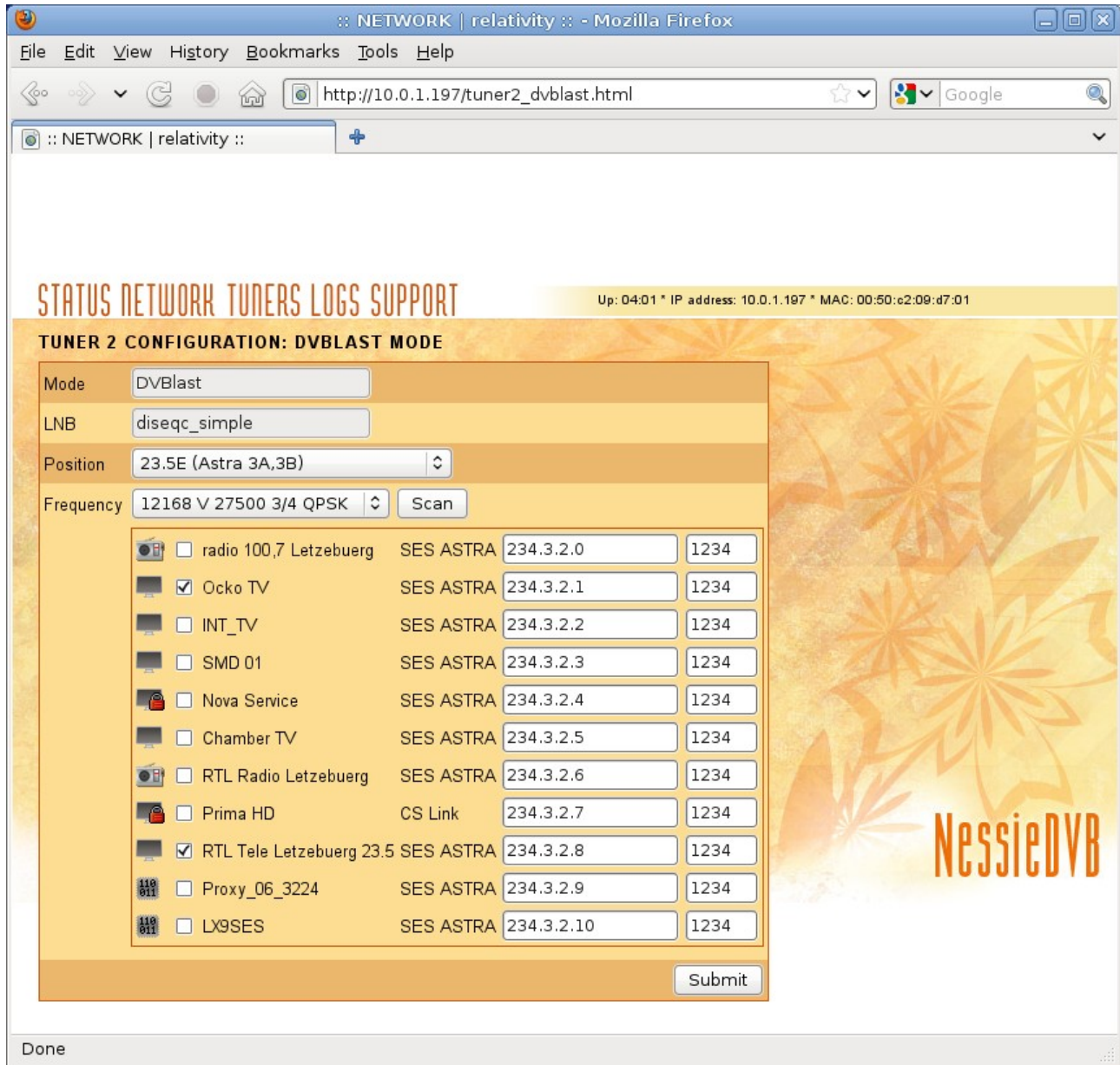
The default port and IP address where the stream is sent can be changed here. But the address must be from a range assigned for multicasted transmission, i.e. 224.0.0.0 to 239.255.255.255. The multicasted IP address is not related to real IP address of NessieDVB in any way.

If we want to stream more services from one transponder it's good to follow the next guidelines to not exhaust the system:

- If the bitrate of the service is higher than 8Mbps it's suitable to stream only one service

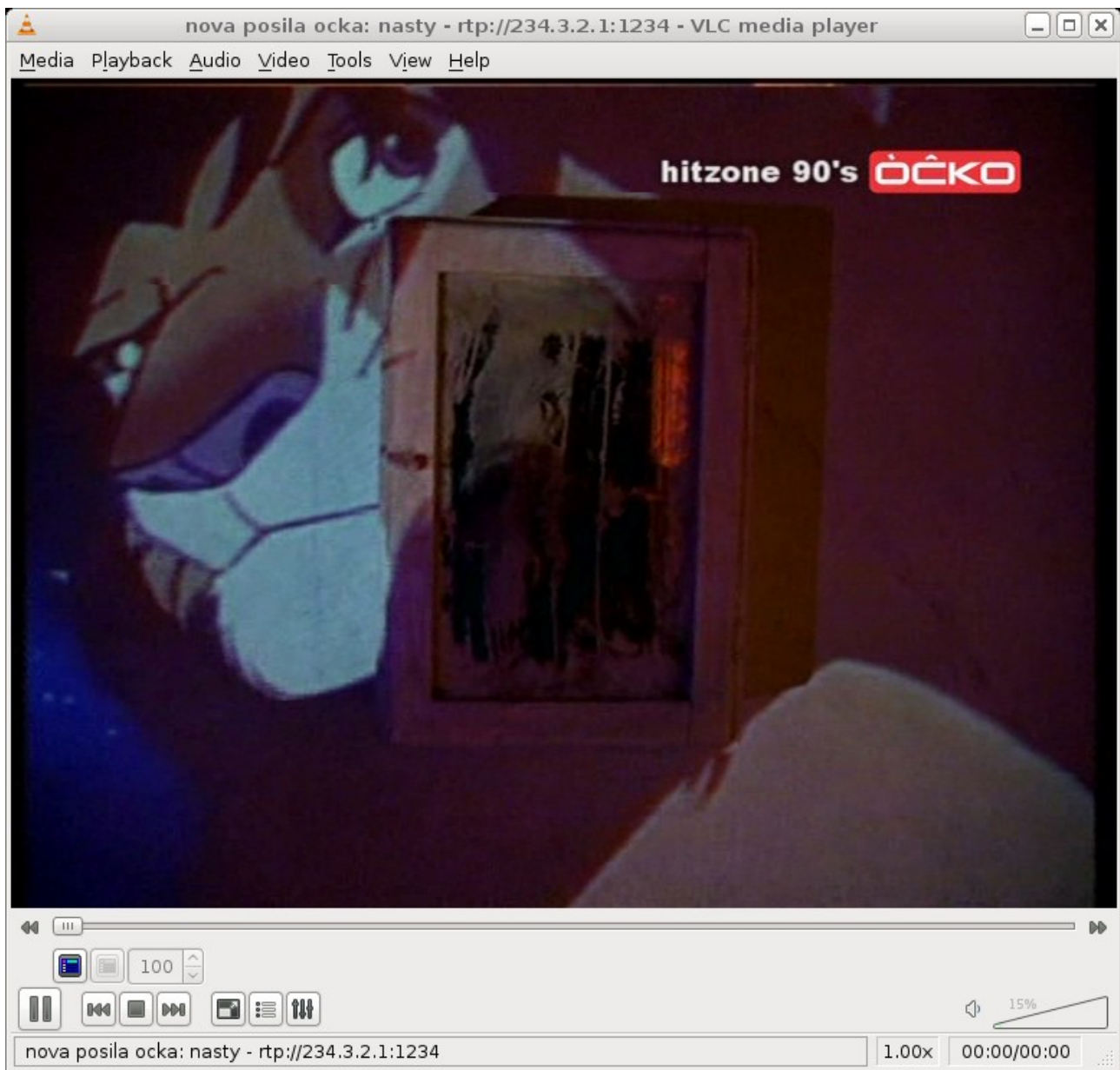
- If the bitrate of the service is between 3Mbps and 8Mbps it's suitable to stream only 2 services maximum
- If the bitrate of the service is lower than 3Mbps three services can be streamed

In case of exceeding of these values (especially strong exceeding) it's not possible to ensure that the selected services will be streamed with no errors, but for NessieDVB this does not represent any risk of damage.



Drawing 28: Web Interface – DVblast - Multicast TV Streamer 4 - services selection

For reception of such a streamed transmission in vlc, for instance, we open the stream with address `rtp://234.3.2.1:1234`, or we can use the link in the „STATUS“ screen eventually (it's necessary to have properly configured web browser and player for the RTP streaming):



Drawing 29: Reception of multicasted program in VLC

After setting up the device it's possible to get back to the initial page by selecting "STATUS" in the top menu. Here it can be seen which services (programs) are streamed on which addresses and from which tuner.

The screenshot shows a Mozilla Firefox browser window displaying the web interface of a NessieDVB device. The address bar shows `http://10.0.1.197/index.html`. The page has a yellow background with a floral pattern and the text "STATUS NETWORK TUNERS LOGS SUPPORT" at the top. A status bar at the top right indicates "Up: 03:52 * IP address: 10.0.1.197 * MAC: 00:50:c2:09:d7:01".

SYSTEM

| | |
|---------------|------------------------------------|
| Model | NessieDVB 2xS2 |
| Serial Number | 000000DEADBEEF01 |
| HW Firmware | NESSIE version HW=1.1, FPGA=9.5 |
| SW Version | Pingu5.14-1 2011-03-10 (test beta) |

NETWORK ☒ Enable auto-refresh

| | |
|-------------|---------------|
| Hostname | nessie |
| IP Address | 10.0.1.197 |
| Subnet Mask | 255.255.255.0 |
| Gateway | 10.0.1.1 |
| Traffic | 9.548 Mb/s |

TUNER 1 ☐ Enable auto-refresh

| | |
|--|--|
| Type | Satellite DVB-S2 NIM (Sharp BS2F7HZ0169) |
| Mode | dvblast |
| Einsfestival 28722 rtsp://234.3.1.1:1234 | |

TUNER 2 ☐ Enable auto-refresh

| | |
|---|--|
| Type | Satellite DVB-S2 NIM (Sharp BS2F7HZ0169) |
| Mode | dvblast |
| Ocko TV 5201 rtsp://234.3.2.1:1234 | |
| RTL Tele Letzebuerg 23.5 5240 rtsp://234.3.2.8:1234 | |

Done

Drawing 30: Web Interface – Status after setup – both tuners in dvblast mode

If we enable „auto-refresh“ at any of the tuners than we get information about the quality of the received signal:

The screenshot shows a Mozilla Firefox browser window displaying the NessieDVB web interface. The address bar shows `http://10.0.1.197/index.html`. The page has a yellow background with a floral pattern and the text "STATUS NETWORK TUNERS LOGS SUPPORT" at the top. Below this, there are three main sections: SYSTEM, NETWORK, and TUNER 1, followed by TUNER 2. The SYSTEM section shows details about the device (NessieDVB 2xS2) and its firmware. The NETWORK section shows network configuration (hostname: nessie, IP: 10.0.1.197) and a traffic bar at 10%. The TUNER 1 section shows it is in dvblast mode with a signal quality bar at 100%. The TUNER 2 section shows it is also in dvblast mode with a signal quality bar at 100%. The bottom of the browser window shows "Done".

SYSTEM

| | |
|---------------|------------------------------------|
| Model | NessieDVB 2xS2 |
| Serial Number | 000000DEADBEEF01 |
| HW Firmware | NESSIE version HW=1.1, FPGA=9.5 |
| SW Version | Pingu5.14-1 2011-03-10 (test beta) |

NETWORK ☒ Enable auto-refresh

| | |
|-------------|--|
| Hostname | nessie |
| IP Address | 10.0.1.197 |
| Subnet Mask | 255.255.255.0 |
| Gateway | 10.0.1.1 |
| Traffic | <div><div></div></div> 10% 9.538 Mb/s |

TUNER 1 ☐ Enable auto-refresh

| | |
|--|--|
| Type | Satellite DVB-S2 NIM (Sharp BS2F7HZ0169) |
| Mode | dvblast |
| Einsfestival 28722 rtsp://234.3.1.1:1234 | |

TUNER 2 ☒ Enable auto-refresh

| | |
|---|--|
| Type | Satellite DVB-S2 NIM (Sharp BS2F7HZ0169) |
| Status | LOCKED |
| Signal | <div><div></div></div> 100% BER: 0 |
| Mode | dvblast |
| Ocko TV 5201 rtsp://234.3.2.1:1234 RTL Tele Letzebuerg 23.5 5240 rtsp://234.3.2.8:1234 | |

Drawing 31: Web Interface – Status after setup – both tuners in dvblast mode – signal quality

If the device is set up so both tuners are in vtuner mode, the initial status screen will change as follows:

The screenshot shows a Mozilla Firefox browser window displaying the NessieDVB web interface. The address bar shows `http://10.0.1.201/index.html`. The page title is `:: NETWORK | relativity ::`. The interface has a yellow background with a floral pattern and the **NessieDVB** logo in the bottom right.

At the top right, a status bar indicates: `Up: 1 days 23:21 * IP address: 10.0.1.201 * MAC: 00:50:e2:09:d7:02`.

STATUS NETWORK TUNERS LOGS SUPPORT

SYSTEM

| | |
|---------------|------------------------------------|
| Model | NessieDVB 2xS2 |
| Serial Number | 000000DEADBEEF02 |
| HW Firmware | NESSIE version HW=1.1, FPGA=9.5 |
| SW Version | Pingu5.14-1 2011-03-10 (test beta) |

NETWORK

☒ Enable auto-refresh

| | |
|-------------|--|
| Hostname | nessie |
| IP Address | 10.0.1.201 |
| Subnet Mask | 255.255.255.0 |
| Gateway | 10.0.1.1 |
| Traffic | 28.380 Mb/s <div><div></div></div> 29% |

TUNER 1

☒ Enable auto-refresh

| | |
|--------|--|
| Type | Satellite DVB-S2 NIM (Sharp BS2F7HZ0169) |
| Status | LOCKED |
| Signal | <div><div></div></div> 79% BER: 0 |
| Mode | vtuner BUSY: 10.0.1.101:51451 (from 2011-03-16 08:21) |

TUNER 2

☒ Enable auto-refresh

| | |
|--------|--|
| Type | Satellite DVB-S2 NIM (Sharp BS2F7HZ0169) |
| Status | LOCKED |
| Signal | <div><div></div></div> 100% BER: 0 |
| Mode | vtuner BUSY: 10.0.1.101:51601 (from 2011-03-16 08:21) |

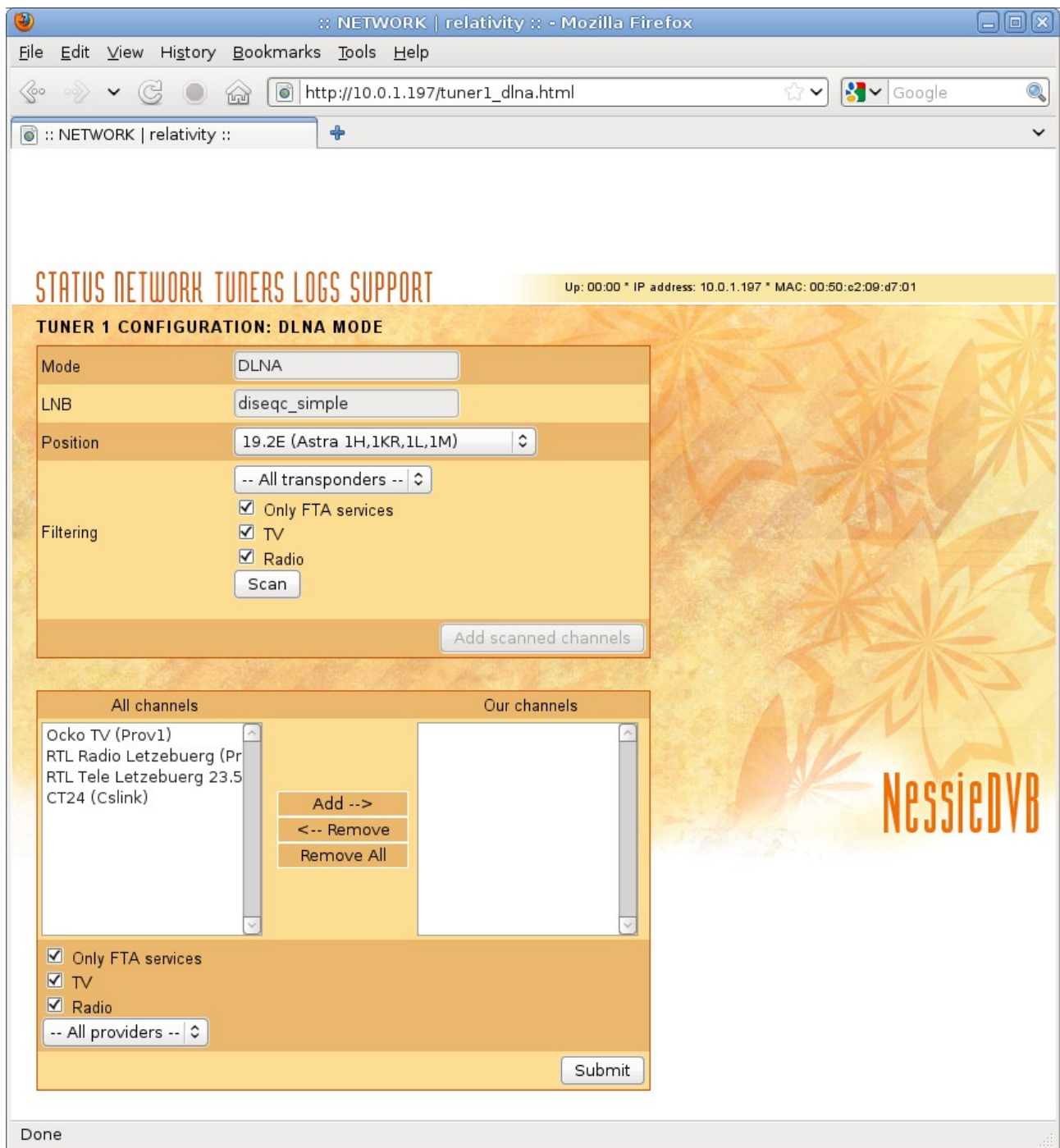
Done

Drawing 32: Web Interface – Status after setup – both tuners in vtuner mode

It's not recommended to leave the signal quality check permanently turned on since the permanently running monitor will unnecessarily overload the device.

DLNA server

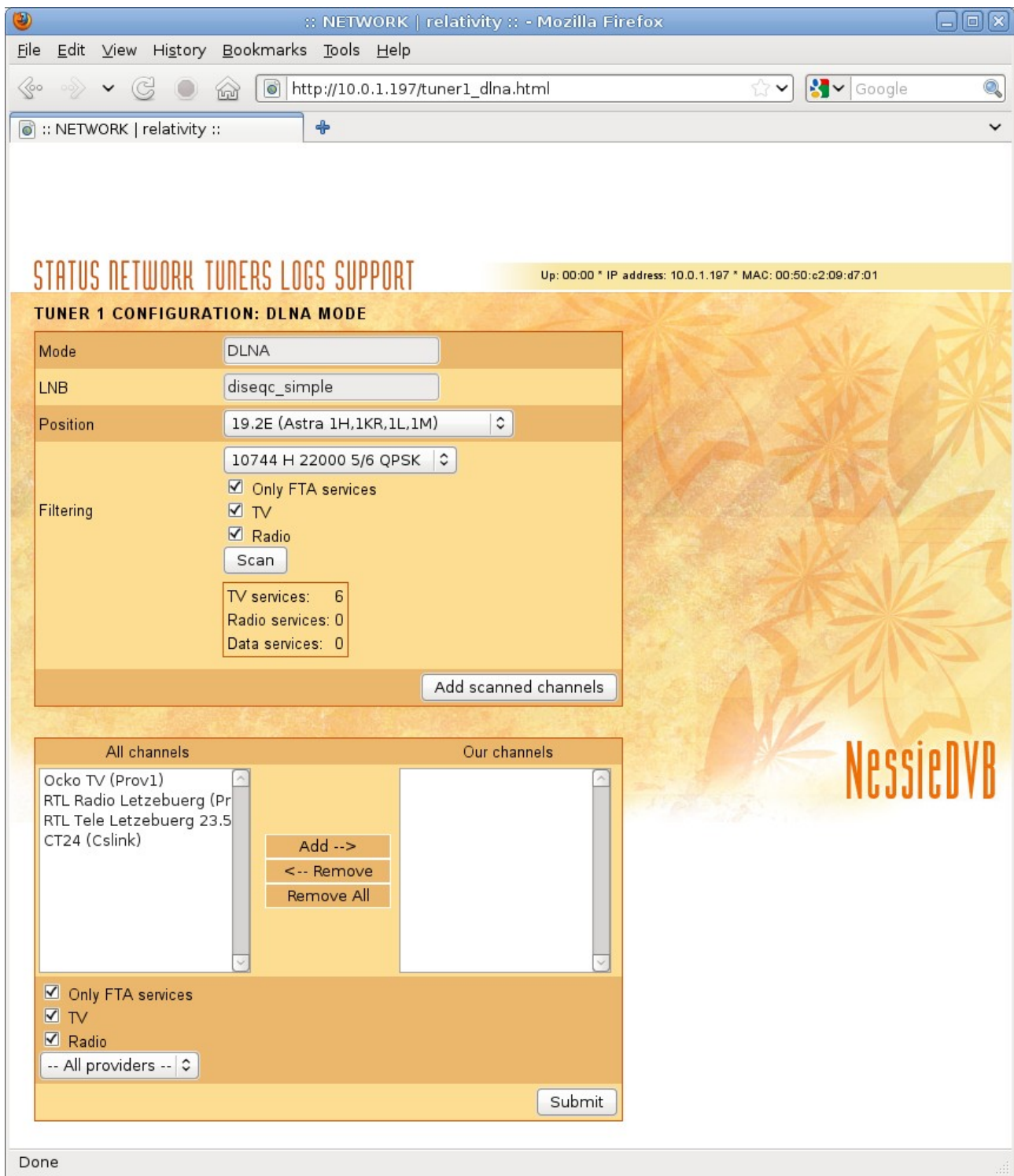
After selecting and approving of this mode the DLNA configuration screen is displayed.



Drawing 33: Web interface - DLNA mode setting

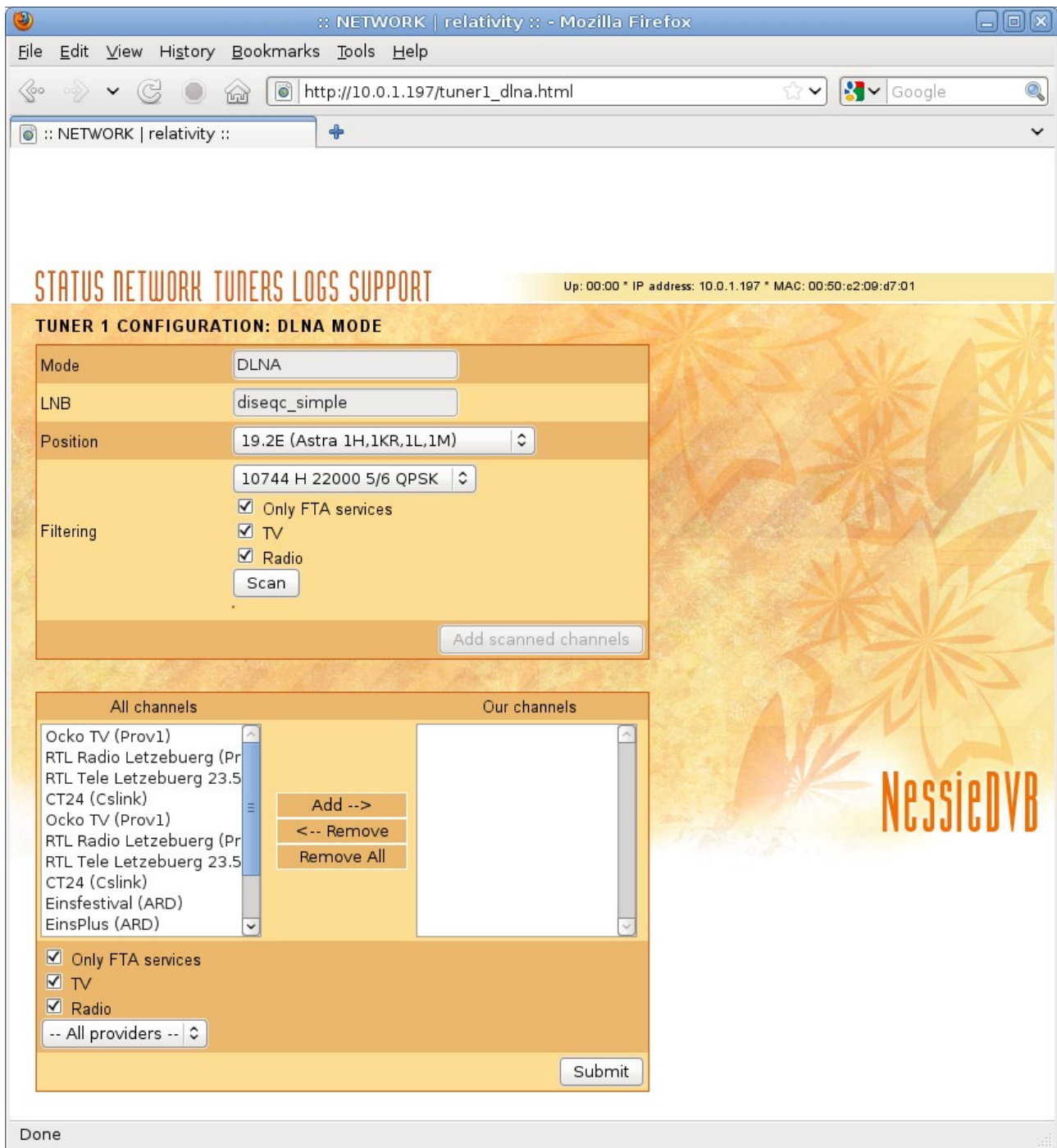
In this first step the satellite from where the services to be offered are selected. If only services offered by some of the transponders are to be used, after approval these transponders are scanned (similarly like in multicast) and sequentially are added to the global offering. For this scanning the filtering option is available – the possible conditions are „Only non-encrypted services“ (Only FTA services), TV and Radio. Also, it's possible to scan the whole satellite (all the transponders) – although this can take quite a long time; even hours.

After pressing Scan the selected transponders are scanned and by pressing „Add scanned channels“ the found services are added to the global offering database and displayed in the „All channels“ box.



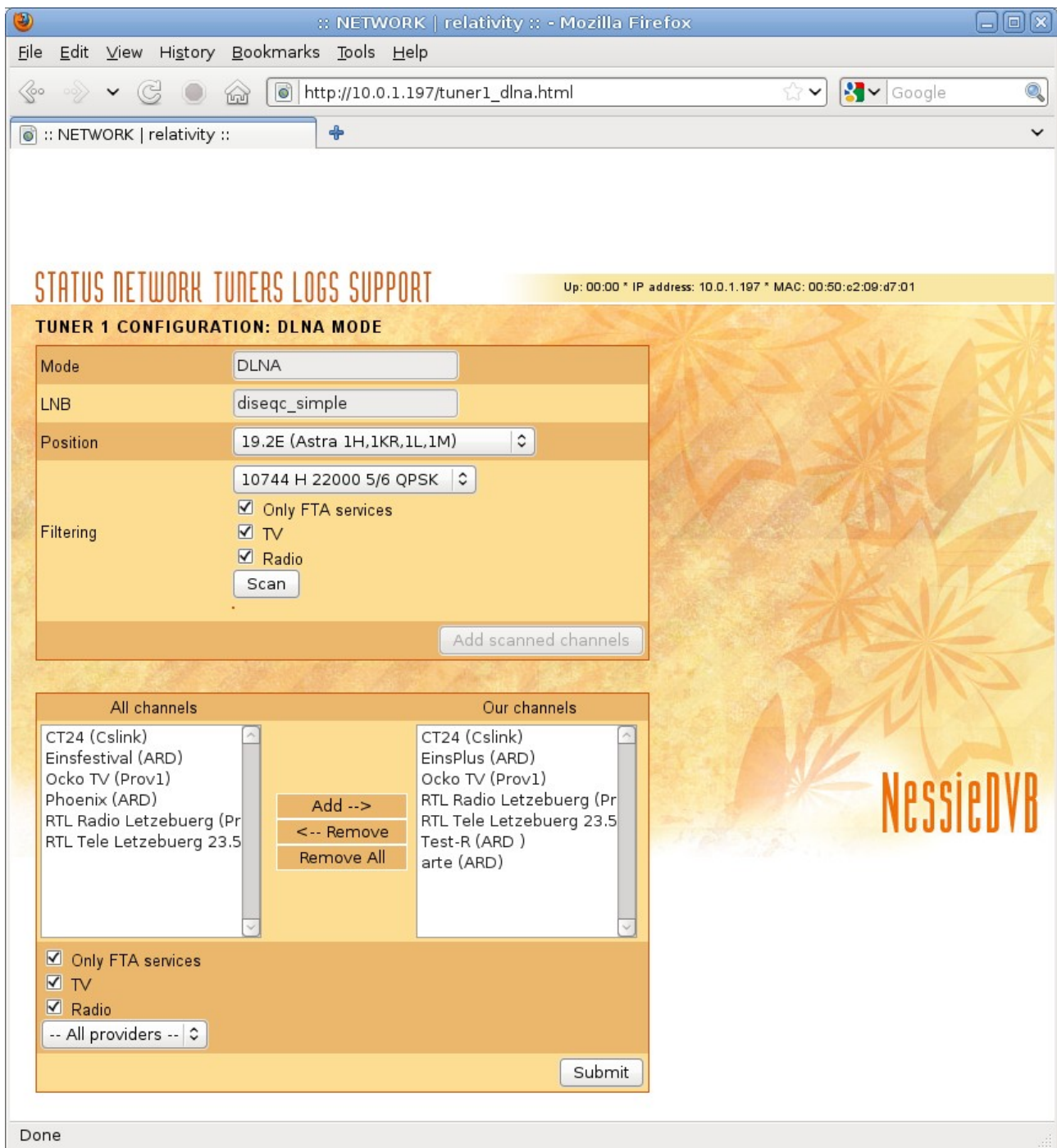
Drawing 34: Web Interface - DLNA mode setting – adding of scanner services

The services that we want to offer from a particular tuner's global database can be moved to the „Our channels“ list. To make the selection from the global database simpler we can utilize the filters to display the services – only unencrypted channels, TV, radio or the services by particular provider alternatively.



Drawing 35: Web Interface – DLNA mode setting – selection from the global database

The last step is to approve the selection by “Submit”. After approval the DLNA server is launched with the actual list of services to offer.



Drawing 36: Web Interface - DLNA mode setting – services approval

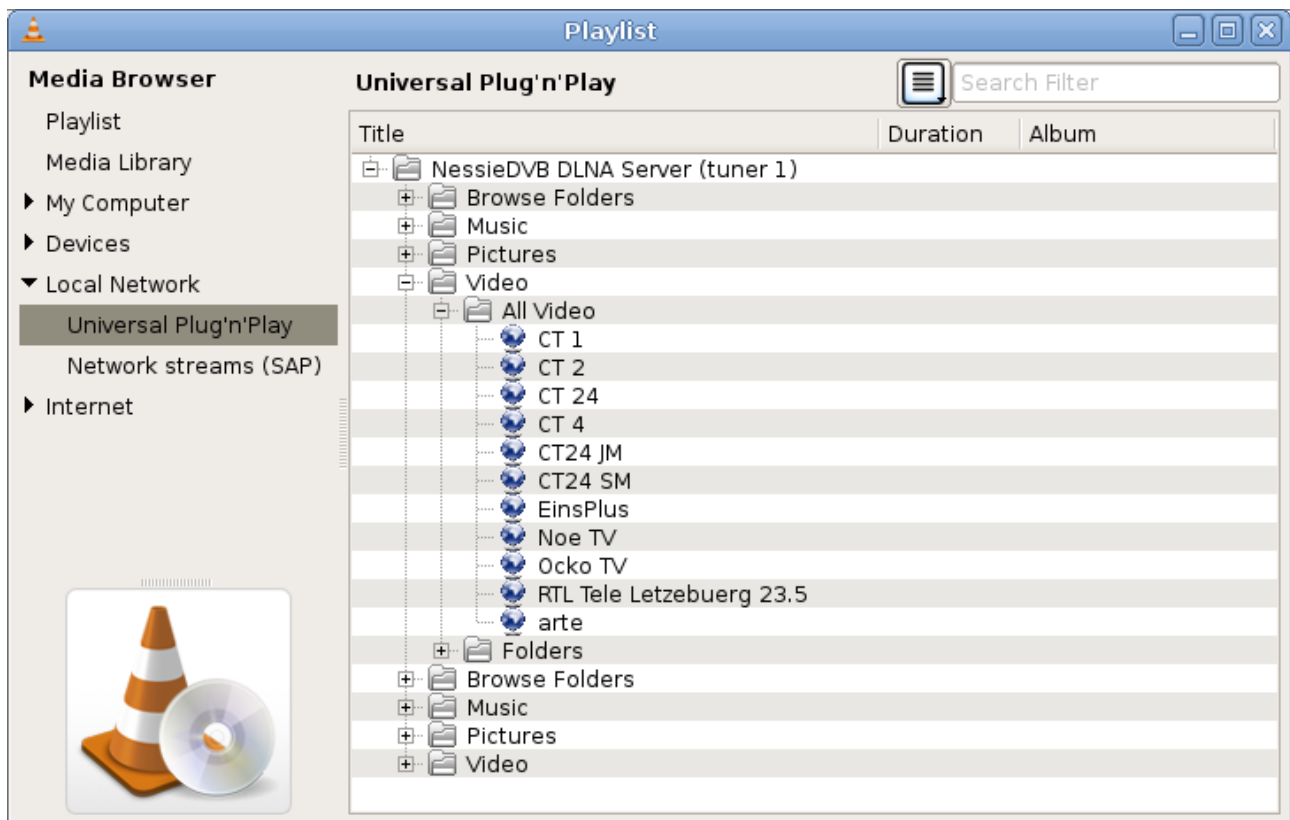
On the DLNA client it's possible then to see the offering from NessieDVB – amongst other DLNA sources (servers). The following screenshots will show an example of how this works.



Drawing 37: DLNA TV client - main menu



Drawing 38: DLNA TV client - channel offer



Drawing 39: DLNA VLC client

Please note that - like for multicasted transmission - if the channel is broadcasted encrypted, will be delivered to the client also in encrypted form, i.e. the client must take care of the decoding of the broadcast.

Support

This menu will allow us to upgrade internal software of NessieDVB. The assumption is that on the local computer - where we are setting up NessieDVB from - there's downloaded the software update file for the device.

STATUS NETWORK TUNERS LOGS SUPPORT Up: 00:58 * IP address: 10.0.1.197 * MAC: 00:50:c2:09:d7:01

SYSTEM

| | |
|---------------|---|
| Model | NessieDVB 2xS1 |
| Serial Number | 000000DEADBEEF01 |
| HW Firmware | NESSIE version HW=1.1, FPGA=9.5 |
| SW Version | Pingu5.14-7 2011-03-25 (test_intern beta) |

Firmware update
Reboot

NETWORK ☒ Enable auto-refresh

| | |
|-------------|---------------|
| Hostname | nessie |
| IP Address | 10.0.1.197 |
| Subnet Mask | 255.255.255.0 |
| Gateway | 10.0.1.1 |
| Traffic | 3.928 kb/s 1% |

TUNER 1 ☐ Enable auto-refresh

| | |
|------|--|
| Type | Satellite DVB-S2 NIM (Sharp BS2F7HZ0169) |
| Mode | dlna |

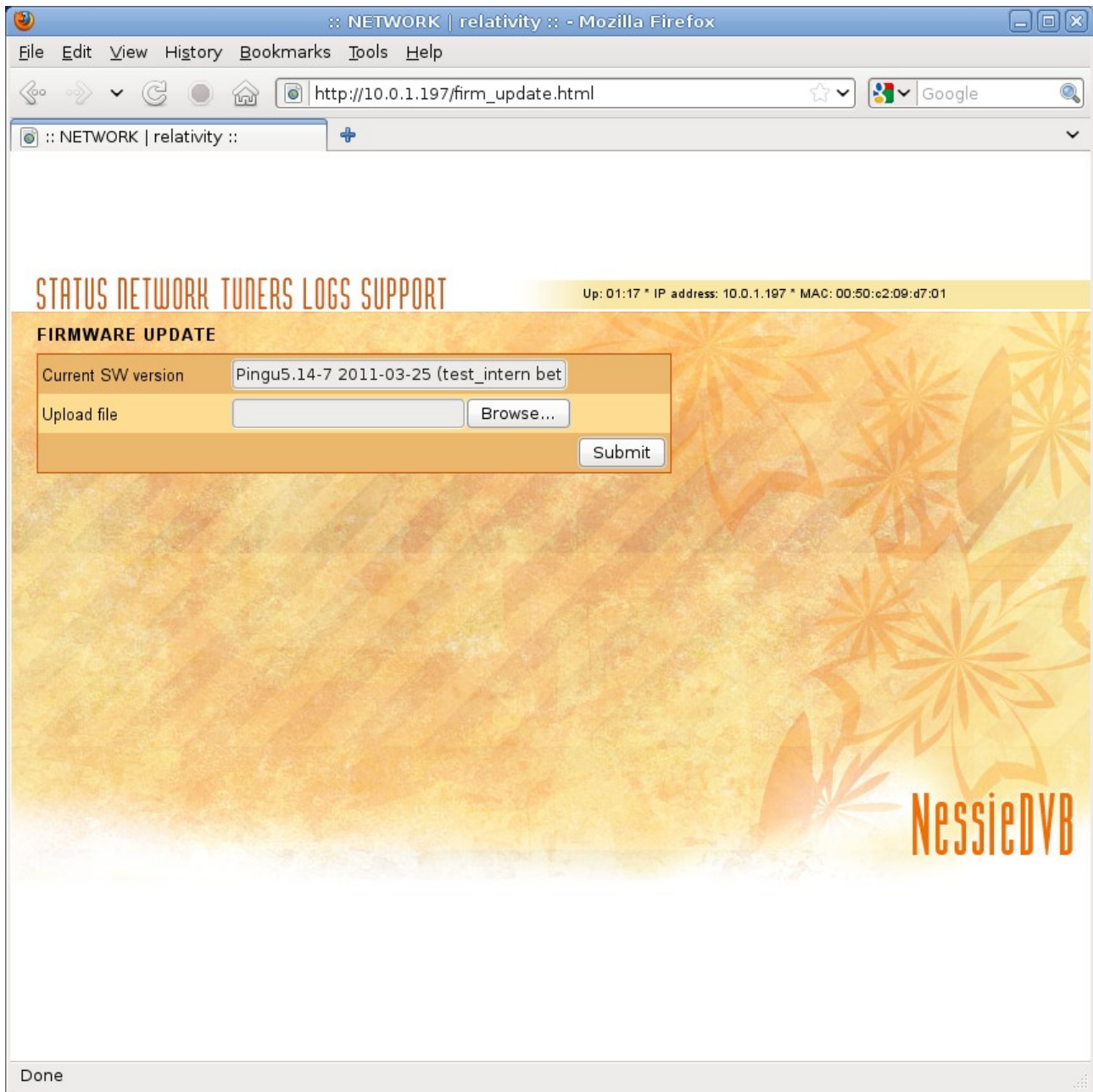
TUNER 2 ☐ Enable auto-refresh

| | |
|------|--|
| Type | Satellite DVB-S2 NIM (Sharp BS2F7HZ0169) |
| Mode | disabled |

http://10.0.1.197/#

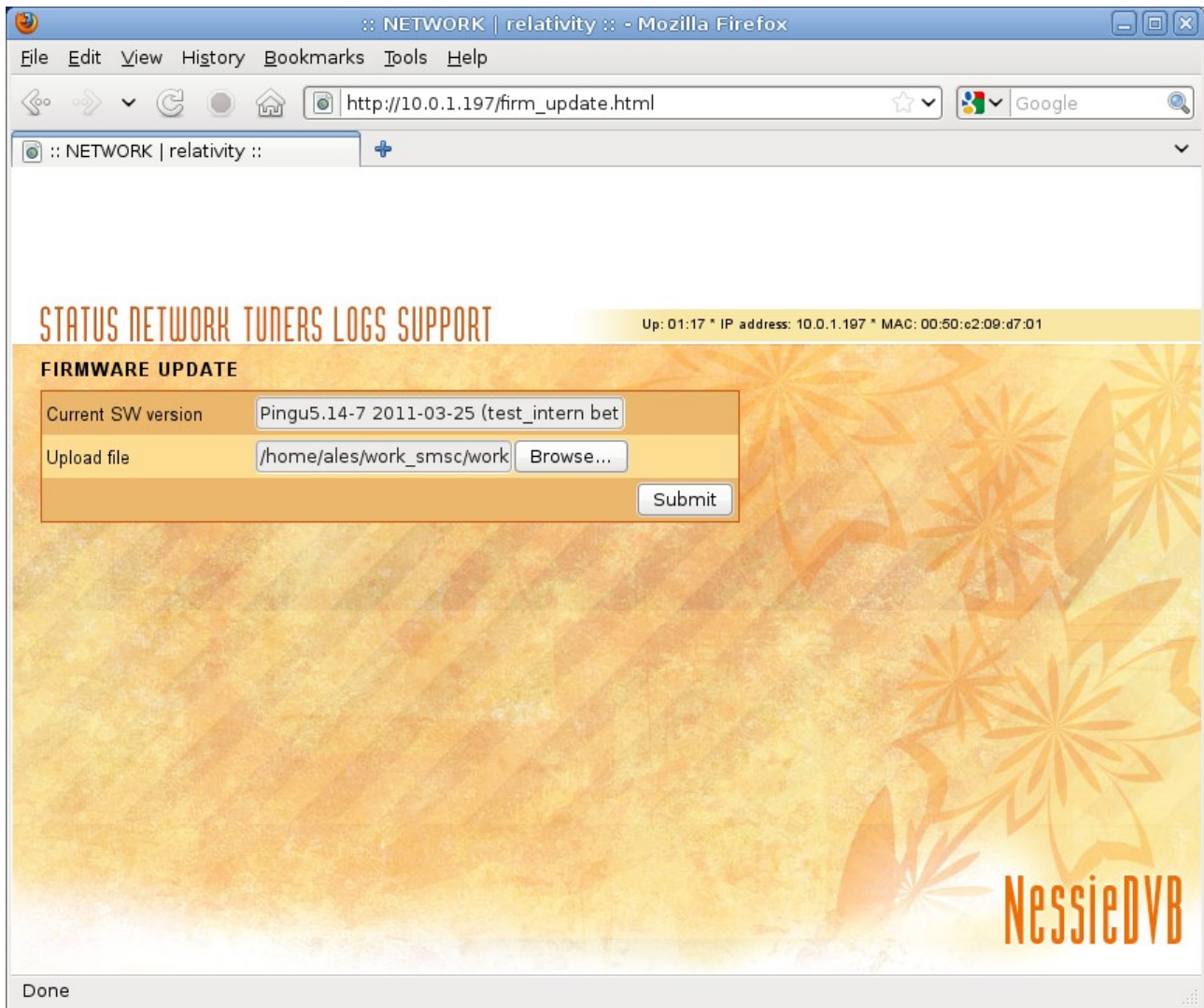
Drawing 40: Web Interface - Firmware Update

The software upgrade consists of the following basic steps:



Drawing 41: Web Interface - Firmware Update Step 1

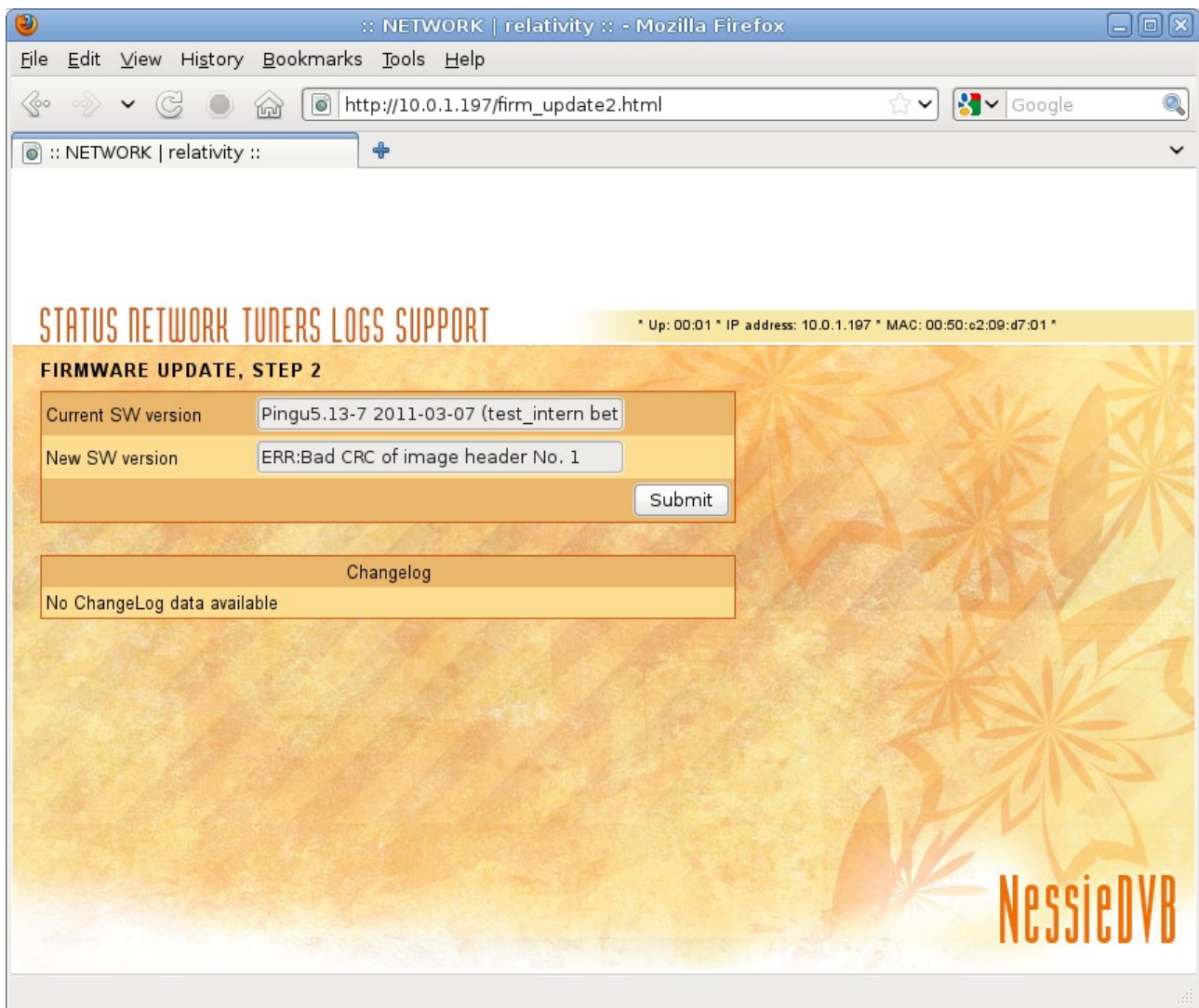
After choosing „Firmware update“ the current version of the software installed to NessieDVB is shown. Using the „Browse“ button and a standard file selection dialog we can choose the file containing the new software.



Drawing 42: Web Interface - Firmware Update Step 2 – firmware file selection

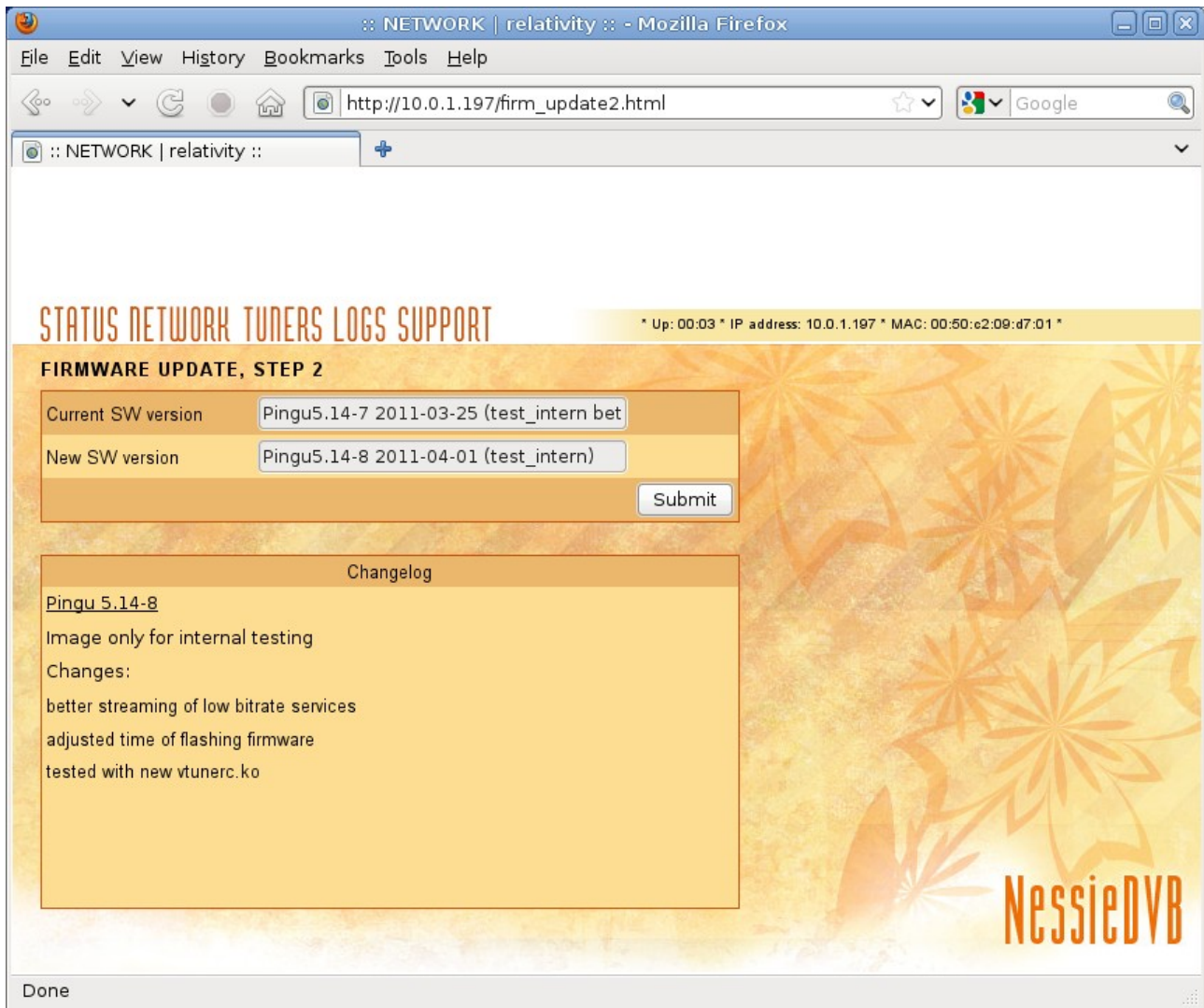
After approving with the „Submit“ button the file is uploaded to the device. Next the file integrity check is provided and the test if the actual software version is supported by the HW of NessieDVB

If a corrupted file is uploaded than the „New SW Version“ field will display the error type and the eventual submit of the software update will be ignored.



Drawing 43: Web Interface - Firmware Update Step 3 – corrupted file test result

After a successful test the new software version is displayed together with the changelog.

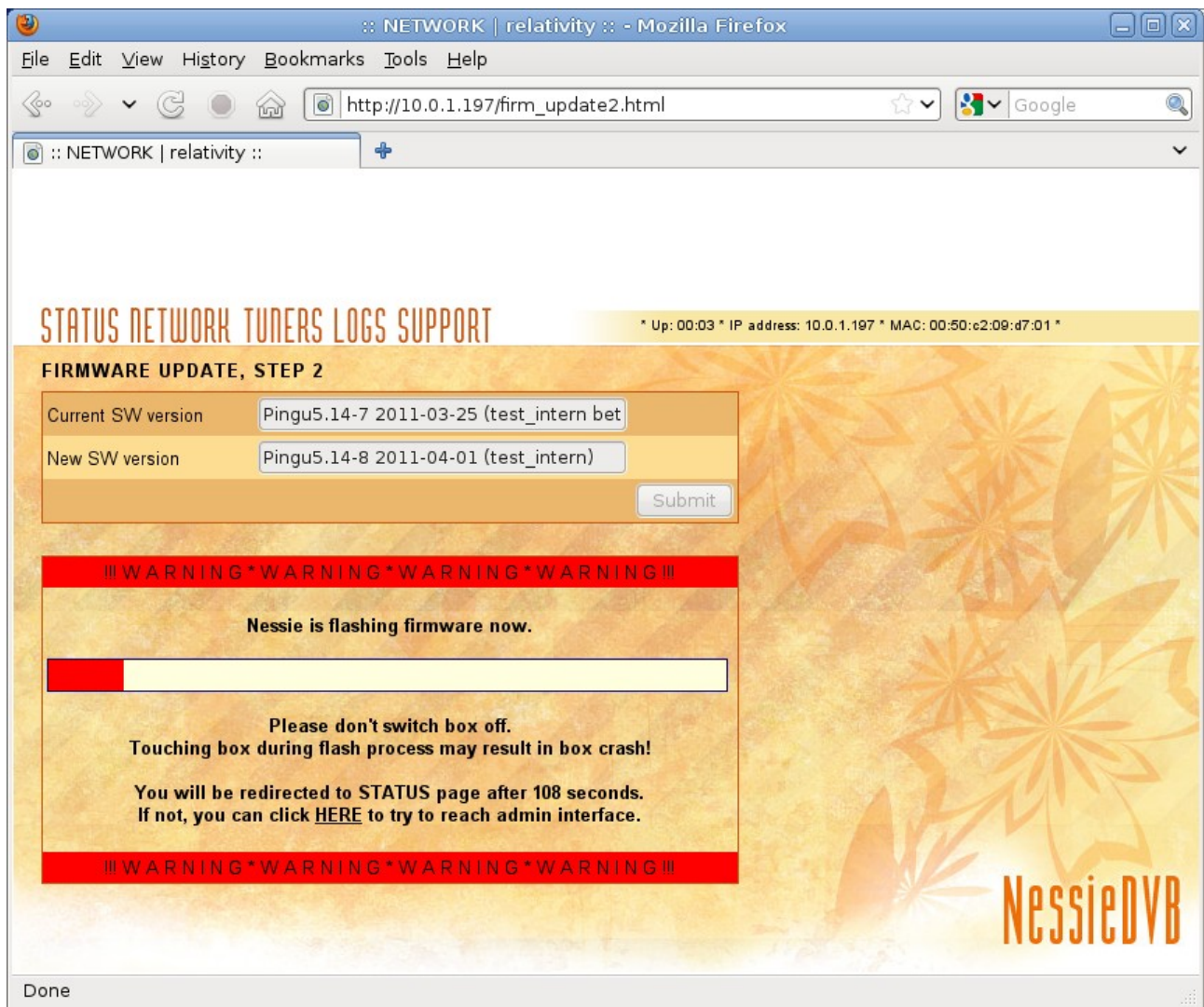


Drawing 44: Web Interface - Firmware Update Step 3 – after fw upload and successful integrity test

After pressing the „Submit“ button the device’s new software is saved to the device.

Please note that this is a fairly sensitive operation. It's necessary to ensure, that during the operation there's no power loss on the device (if possible use a UPS). Similar warning is also displayed on the web interface.

The steps to correct an eventual damage during the update are described at the end of this manual.



Drawing 45: Web Interface - Firmware Update Step 4 – flashing itself

According to the size of the part of the software, that is updated in NessieDVB, the process will take something between 1.5-3 minutes. The time estimate to complete the process is displayed on the screen so as the user will know how long to wait. The update task is displayed using a progress bar.

After a successful update of software, NessieDVB is automatically rebooted. The user settings are retained. The estimated update time is including the time for reboot.

Software installation on a Linux PC for vtuner mode

For vtuner mode the following is necessary on the PC side:

OS Linux with complete source codes of programs critical to successfully compile the following kernel module and vtuner client. No particular distribution is required; the following information should be sufficient for anything recent with the condition that the particular steps should be modified according to the actual requirements. The client (vtunerc) should be portable between particular PCs and distributions (with the assumption of equivalent glibc version and same processor architecture), while the kernel module (vtunerc.ko) should be created each time for the actually used kernel. For the same reasons it's advisable in case of upgrading to a new kernel to recreate the kernel module while the vtunerc client can remain the same.

Necessary program equipment:

DVB-API version 5.0 or higher (is specified in linux/include/linux/dvb/version.h),

installed C (gcc) compiler, make and binutils. Recent versions are recommended. The compilation is straightforward with make 3.80 and higher, gcc 4.3.3 and higher (including version 4.5.x) and binutils 2.18 to 2.21). Lower versions may (but doesn't have to) be usable.

Driver installation

After unpacking of the vtunerc.tar.gz archive the compilation is done by:

```
make KDIR=/path/to/kernel/sources
```

the directory with the driver source codes, /path/to/kernel/sources is necessary replace with the path to the source code (headers). If the source tree of the kernel is in the standard path, i.e. /usr/src/linux (it can be a symbolic link to the actual kernel tree), then only 'make' without parameters is enough.

The compiled kernel can be installed by using one of the following methods:

Driver loading using insmod

The driver is installed using the command

```
insmod vtunerc.ko
```

If the driver cannot be loaded it's necessary to ascertain that the kernel includes the dvb-core driver.

Dvb-core can be compiled into the kernel or can be present as a module. If dvb-core is in a form of module than either the sequential loading of the two modules is required:

```
insmod dvb-core.ko (in the directory where dvb-core.ko resides or with full path)  
insmod vtunerc.ko
```

or to use the modprobe command like described below.

If this module cannot be loaded than there is probable mismatch between the kernel and the module (e.g. the change of the gcc version between the kernel compilation and the vtunerc.ko module compilation, or a configuration change and kernel recompilation without installing of the new modules or the used kernel version does not fit with the version where the module was compiled and so on).

If there's the need to start more virtual adapters than we can use

```
insmod vtunerc.ko adapters=N
```

where N is 2 to 4.

Driver loading using modprobe

Another and perhaps better way of loading the module into the kernel is to copy or move vtunerc.ko to other kernel modules (typically in /lib/modules/KERNEL_VERSION/) under some directory (e.g. /lib/modules/KERNEL_VERSION/misc) and issue the command

```
depmod -a
```

The module itself is then loaded by the command

```
modprobe vtunerc adapters=N
```

(Unlike for the insmod command here we cannot use the full filename vtunerc.ko, but only the short name, i.e. vtunerc. This is a common error giving us a somewhat misleading error message:

```
"FATAL: Module vtunerc.ko not found".
```

where N has the same meaning as in the previous case. In this case the modprobe command will take care of the loading of other related modules (dvb-core) automatically by itself. If the loading of the module does not succeed probably the same type of conflicts are happening as it was described in case of the insmod command.

Setting up the created devices' permissions

After loading the drivers the corresponding number of adapters are created in /dev/dvb/ and

the same number of vtunercN devices in /dev/. In all /dev/dvb/adapterN/directories there are created the demux0 and dvr0 devices.

It's necessary to ensure that all the /dev/vtunercN devices were available for reading and writing by a non privileged user. This can be done from the command line by

```
sudo chmod 666 /dev/vtunerc*
```

Or automatically by the system, using the udev rules. If there's a file in the system, called /etc/udev/rules.d/10-local.rules then we add the following line to it:

```
KERNEL=="vtunerc*", MODE="0666"
```

Or it is possible to create a corresponding file with the above name and content.

The filename is not critical it needs to be however in the given directory and have the '.rules' extension. If in your distribution the udev rules files are placed anywhere else than /etc/udev/rules.d, add the new rule to that directory.

For udev to process the new rules it's needed to inform it by the command:

```
udevtrigger
```

or for newer version of udevd

```
udevadm trigger
```

Specific requirements for particular distributions:

Gentoo:

No specific process is necessary only it's recommended to have the kernel compiled with the „DVB For Linux“ switch set to YES (if we use modprobe it can be also available like a module) – see point 4 at the Debian section.

Mandriva, RedHat and clones(Centos, RHEL), Suse:

All distributions from 2011 year or later has required kernel with correct DVB API already.

Debian, Ubuntu, Kubuntu:

Because these distributions do not contain the files necessary to install the vtunerc driver, it's necessary to:

- if we have a kernel of version 2.6.30 and higher, install the kernel source code (they're installed as compressed file into /usr/src/), uncompress it and create a link called 'linux' to the created directory structure., e.g.:

```
cd /usr/src/  
sudo tar -xjf linux-source-2.6.30.tar.bz2  
sudo ln -s linux-source-2.6.30 linux
```

- if our kernel version is lower than 2.6.30 or we have a problem loading the dvb-core.ko kernel module than we need to install an own kernel from the source code. The following process can be used for that, for instance:

1. We install the packages, needed for the further work:

```
apt-get install kernel-package libncurses5-dev fakeroot wget bzip2 build-essential
```

2. We install the version 2.6.30 (or higher) kernel source code

```
cd /usr/src  
wget http://www.kernel.org/pub/linux/kernel/v2.6/linux-2.6.30.tar.bz2  
tar xjf linux-2.6.30.tar.bz2  
ln -s linux-2.6.30 linux
```

3. We configure the installed code for the further work

```
cd linux-2.6.30  
make mrproper  
cp /boot/config-`uname -r` ./config  
make menuconfig #Load an Alternate Configuration File" ---> .config
```

4. Change the „DVB For Linux“ to YES

5. We save the configuration and end make menuconfig

6. We adjust the environment for the optimal compilation of our kernel for our processor, we set N to the number of cores of the processor (this step is not mandatory but without

it the compilation will take longer).

```
export CONCURRENTLY_LEVEL=N
```

7. We create the installation packages for simpler installation and automatic configuration of grub

```
make-kpkg clean
fakeroot make-kpkg --initrd --append-to-version=-custom kernel_image
kernel_headers
```

8. Make sure that the packages are created

```
cd ..
ls *.deb
```

9. Install the new kernel (the names of the packages can differ if we have a different processor architecture or we've chosen a different differentiator string eventually)

```
su
dpkg -i ./linux-image-2.6.30-custom_2.6.30-custom-10.00.Custom_i386.deb
dpkg -i ./linux-headers-2.6.30-custom_2.6.30-custom-10.00.Custom_i386.deb
```

10. And boot to the new kernel. If we use grub as the bootloader, the new kernel should have been added to the kernel list automatically as the default selection.

```
reboot
```

Vtunerc client

The missing device /dev/dvb/adapaterN/frontend0 is created after client application component is started. Before launching it's necessary to have the vtunerc.ko driver loaded to the kernel.

Compiling the client software

We unpack the dreamtuner-read-only-DATE.tar.bz2 package downloaded from www.nessiedvb.com to /usr/local/src/. Based on the architecture of our computer we switch to the build/ARCH subdirectory

(For 32 bit OS on an Intel and AMD processor ARCH is equal to i686, for 64 bit OS on these processors it's then x86_64.) In this directory we launch :

```
make client
```

an executable file for our architecture is created under dist/ARCH. Further using

```
sudo make install
```

we copy this file to /usr/local/bin as vtunerc. If this directory does not exists than the above command will create it. This file can be run as any other installed executable program.

Because make install only copies the required binary file to the target directory, it's also

enough to copy the file to any directory that's included in the path of executable programs.

Running the client software

The launching of the client software is possible either by only

```
vtunerc -S
```

or if more frontends are required for more adapters then by:

```
vtunerc -S -a N
```

where N is the adapter index. In this case it's required to launch the client separately for each adapter; in a separate console or – better – to redirect the standard output of the client to /dev/null or to a log file and run it in the background using a command like

```
vtunerc -S -a N 2>&1 > /dev/null &
```

or

```
vtunerc -S -a N 2>&1 >> LOGFILE &
```

or to create a script containing the given commands or their variants alternatively. Advanced and recommended way of running of the clients is the usage of the svscan or monit programs.

Using of the software for vtuner mode on the PC

If the vtunerc.ko driver is loaded and the required number – equaling to the number of emulated DVB-S2 adapters - of clients are run it's possible to use any software for processing of DVB signals that is available for OS Linux, like VDR, myth-TV, kaffeine, MeTV and so on.

Warning: because of the chain of cooperation functions of the particular parts it's necessary to start the whole system always in the proper order:

Loading of vtunerc.ko → start vtunerc client → start the program for the stream reception

And also to stop in the proper order:

stop the program for the stream reception → stop vtunerc client → rmmod vtunerc.co

If this requirements are not met, there is the chance to end up with problems that are possible to solve only by rebooting the PC (client in the form of non-killable zombie, vtunerc.ko, that cannot be loaded or unloaded etc..).

Specification:

Power 12V/2A stabilized

Load (typical) 4-12VA, depending on the connected devices in operating mode
 (together with the supply adapter included)

Mech. Size 164x136x39mm

Operating conditions 5-45°C, environment with non-condensing humidity

Input/Output 2xLNB in
 2xLNB out (loop-through)
 Supply connector
 Network connector (RJ-45, 100/10Mbps, FD/HD, MDIX)

Generated data flow up to 60Mbps

The device is manufactured in the Czech Republic. All the sources used inside are switched with at least 90% of efficiency so the operating temperature is kept rather low and the device and thus the device's reliability is extended greatly. The heat coming from inside the device is – due to the device constructions – drained to the metal body and this because of the black color efficiently emits it to the environment. By this a relatively low operating temperature is achieved inside the device even in case of full load and only using of passive cooling.

LED based device status signalization

Normal boot and run:

| LED1 | LED2 | Status |
|-------------|-------------|--|
| Red | Red | Test fw |
| Orange | Orange | Booting |
| Green | Orange | Tuner 1 initialization OK |
| Green | Green | Tuner 2 initialization OK, normal run |

If any of the LED diods remains in Orange color after boot it means that the initialization of the given tuner (or other part in the given receiving channel) was not completed correctly. Switch off the device wait for few tens of seconds and start again.

Emergency firmware download

If during the firmware update there's for instance a power loss then NessieDVB will remain without a working software. This is indicated by a status LED after switching on and booting by red status of diods lasting longer than 5 seconds. If this status is detected the automatic run of the emergency firmware download process is initiated. Because there's no web interface and configuration, the emergency update is provided in a fixed IP address configuration and the progress indication and error states indication is done by the LED diodes on the front panel.

To the emergency download of the image the following is necessary:

- a tftp server on the same network segment, like NessieDVB with a fixed IP address of 10.0.1.101 (Nessie in this moment hes the IP address equal to 10.0.1.229, netmask 255.255.255.0).
- in the tftp server's root directory have the firmware image available and named as Nessie.img
- switch off and on the charging of NessieDVB
- wait until both diodes began to blink Red and Green on NessieDVB (alternately) with a frequency of cca 2,5Hz
- switch off and on the charging of NessieDVB

| LED1 | LED2 | Status |
|---------------------|---------------------|--|
| Off | Blinking Red | FTP server found on address 10.0.1.101 |
| Blinking Red | Off | No Nessie.img image found on the tftp server |
| Blinking Orange/Red | Red | Deletion of the flash |
| Red | Blinking Orange/Red | Writing of the flash |
| Blinking Red/Green | Blinking Red/Green | The image download completed, Nessie is waiting for switching off and on |