# Home Brew IPTV head-end

Thomas Kernen

RMLL 2012 Release

# History & Objectives

- Build an IPTV head-end for my home that would fit my personal lab testing and viewing requirements
- In a “production” state but continues to be upgraded with new hardware and software
- Word of mouth has led to people being interested in knowing how I did it
- Presentation is designed to help those who wish to build themselves an IPTV head-end for their home, office, lab, school, campus, etc...
- Focus of this presentation is on the head-end requirements for streaming live radio and TV services and not Video on Demand services or receivers
- All the information in the presentation is provided on an “as is” basis without warranties of any kind, either express or implied

# Agenda

- Planing the Head-End
- Building the Head-End
- Media Players
- Support for IPv6
- What needs worked on (IMHO)
- References

# Planning the Head-End

# Introduction

- Evolution of TV viewing from 1 screen/home to multi-screen & multi-viewer environment
- Traditionally users have had a single physical source for their TV signal. Now there are multiple sources (satellite, cable, terrestrial, IPTV, etc)
- Receivers are evolving quickly and each generation introduces new and improved features. Hard to keep up and usually requires swapping out gear
- Delinearisation of content means most people want to record and then view it later
- Best off the shelf products tend to provide a subset of the features required to support the statements above, but then are usually locked into their own vendor scheme
- Home brew systems aren't perfect, your own requirements will determine if some or part of what follows is of any use to you

# Basics of reception of TV services

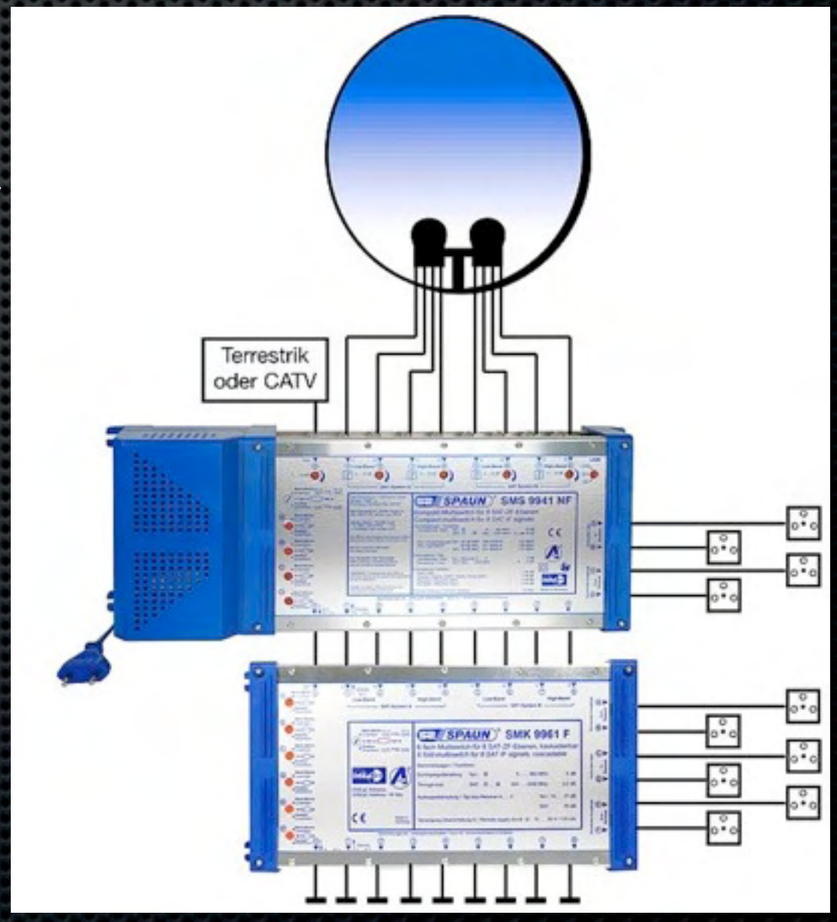
- In an analogue world, each frequency carries a single service (=TV channel)
- These days virtually all services are transmitted digitally
  - “analogue switch off” - ”digital switch over”
- Each frequency typically carries up to 12 or 13 different digital TV services
  - less for HD services
- Services are modulated on a carrier according to DVB specifications. T and T2 for Terrestrial, S and S2 for satellite, C for cable, ...
- Each modulation type requires a different front-end receiver
- All services on a given frequency are contained within an MPEG-2 Transport Stream
  - Multiple Program Transport Stream (MPTS) or Single Program Transport Stream (SPTS)
- And may be individually scrambled (= encrypted)

# Basics of Reception of TV services

- Terrestrial or cable source: full RF bandwidth and therefore services are available at the output of the wall socket. Multiple receivers can be attached without impacting one another
  - To receive a transport stream, the receiver must tune to a specific frequency and setup additional parameters (RF bandwidth, Symbol Rate, modulation scheme, Guard interval, Forward Error Correction, ...)
- With satellite sources, things are more complicated. Different satellites (orbital positions), polarisations (Horizontal/Vertical) and bands (low and high) come into play
  - To receive a transport stream, the receiver must tune to a specific satellite, frequency, polarisation, band and setup additional parameters (Symbol Rate, Forward Error Correction, modulation scheme, ...)
- In a classic receiver these additional parameters are normally hidden from the end user and may be auto detected or preprogrammed

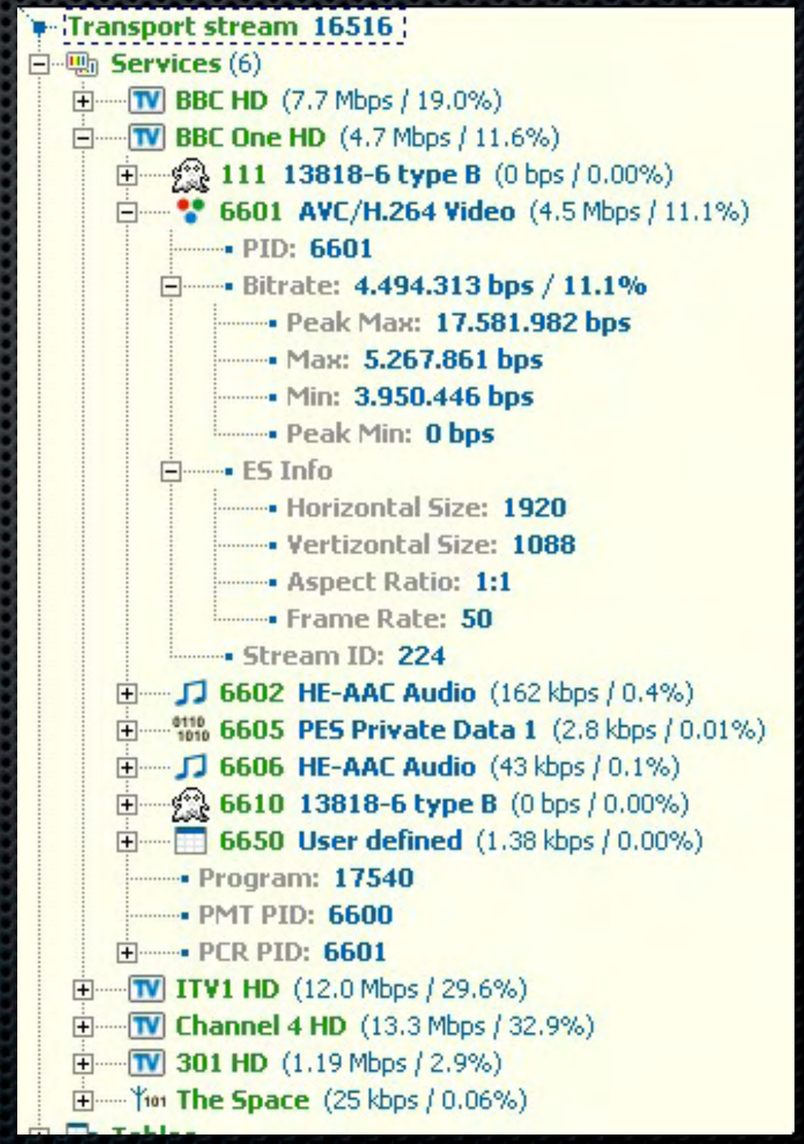
# Basics of Reception of TV services

- To receive multiple satellite feeds we need to include a few components into the reception system
- Each Low Noise Block (LNBs) needs 4 outputs to provide the 2 polarisations (horizontal/vertical) for each of the 2 bands (low/high). This called a quattro LNB
- If you need services from 3 different satellites you need 3 LNBs, either mounted on a universal dish that has a bracket for those 3 orbital positions, or on 2 or 3 separate dishes.
- All the outputs of the LNBs need to be wired to a DiSEqC switch. Different models exist depending on the number of inputs from the LNBs and outputs to the receivers are required. Each satellite will be linked to an input number (usually 1 to 16)
- Pro systems don't use DiSEqC since each input is wired to a L-Band Multichannel multiplexer



# Anatomy of a Transport Stream

- Example of the structure of an MPTS
- Service analysed here is BBC ONE HD
  - taken from a UK Freeview DVB-T2 multiplex
  - satellite and cable use the same overall format
- Transport Stream ID: 16516
- Video PID: 6601
- Audio PIDs: 6602, 6606
- Private Data PIDs: 6605



# Scrambled services

- There are many scrambling vendors on the market
  - Different content providers use different systems
- Scrambled services are setup to protect access to the content
- These are usually bound to distribution rights for a specific territory
- Some scrambling systems used by content providers may not provide the option of using a standalone subscription card but requires to be paired with a specific provided receiver
- Please check with your content provider(s) if their services can be used with a standard compatible receiver and what scrambling system is used

# Planning your channel line up

- Use one of the numerous databases to identify what services you are interested in
- Make a list of all of them: Frequencies, modulations, scrambling
- Classify by frequencies, transports and modulations = number of required receivers

| 28.2°E | Astra 1N | 10773.00 H 45 UK | DVB-S | QPSK | 22000 5/6 | 33.8 Mbps | NID:2 | TID:2045 |
|---|---|---|---|---|---|---|---|---|

| Zap | Name | Country | Category | Packages | Encryption | SID | VPID | Audio | PMT | PCR | TXT | Last updated |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ZAP | BBC One London 16:9 | United Kingdom | General | BBC Freesat Sky Digital | Clear | 6301 | 5000 | 5001 eng 5002 eng | 256 | 5000 | 5003 | 2012-02-25 + |
| ZAP | BBC Two England 16:9 | United Kingdom | General | BBC Freesat Sky Digital | Clear | 6302 | 5100 | 5101 eng 5102 eng | 258 | 5100 | 5103 | 2012-02-25 + |
| ZAP | CBBC 16:9 8h-20h | United Kingdom | Children | BBC Freesat Sky Digital | Clear | 6317 | 5200 | 5201 eng 5202 eng | 262 | 5200 | 5203 | 2012-02-25 + |
| ZAP | BBC Three 16:9 20h-5h | United Kingdom | General | BBC Freesat Sky Digital | Clear | 6319 | 5200 | 5201 eng 5202 eng | 266 | 5200 | 5203 | 2012-02-25 + |
| ZAP | BBC One West 16:9 | United Kingdom | General | BBC Freesat Sky Digital | Clear | 6341 | 5900 | 5901 eng | 268 | 5900 | 5903 | 2012-02-25 + |
| ZAP | BBC One Cambridge 16:9 | United Kingdom | General | BBC Freesat Sky Digital | Clear | 6351 | 5700 | 5701 eng | 269 | 5700 | 5703 | 2012-02-25 + |
| ZAP | BBC One Channel Islands 16:9 | United Kingdom | General | BBC Freesat Sky Digital | Clear | 6361 | 5800 | 5801 eng | 270 | 5800 | 5803 | 2012-02-25 + |



Source: King Of Sat

# Differences between receivers and HEs

- Consumer receiver tunes to **the** service the user is currently viewing. In a head-end **all** services that need to be received are permanently tuned to
- Consumer receiver is designed for descrambling **the** service currently viewed. In a head-end **all** services in the transport stream need to be descrambled. This requires special Conditional Access Modules that can descrambled multiple services at the same time.
  Prefer CAM vendors with professional series such as Aston and SMiT
- Consumer receiver with IP capabilities will typically stream **a** single channel. In a head-end we stream **all** the services
- To allow any end device in the network to receive any service at any point in time:
  - We need multiple receivers tuned to all the channels to be able to receive, descramble and stream around the clock

# Building the head-end

# What you need to build your own HE

- Hardware:
  - A server case and a motherboard with enough slots for all your receivers (PCI and PCIe)
  - Receivers for your different sources (Analogue, DVB-T/T2, DVB-S/S2, DVB-C/C2)
  - Conditional Interface (CI) daughterboards for descrambling content protected services
  - Conditional Access Modules (CAM) for the daughterboards
  - Subscription cards for the different services you wish to descramble
- Software:
  - A modern Linux distribution (Ubuntu for example) with a recent kernel
  - Video4Linux (V4L) DVB drivers and utilities from Linuxtv.org
  - DVB streaming applications such as DVBlast
  - A decent understanding of how Linux works and how to get applications running

# Hardware

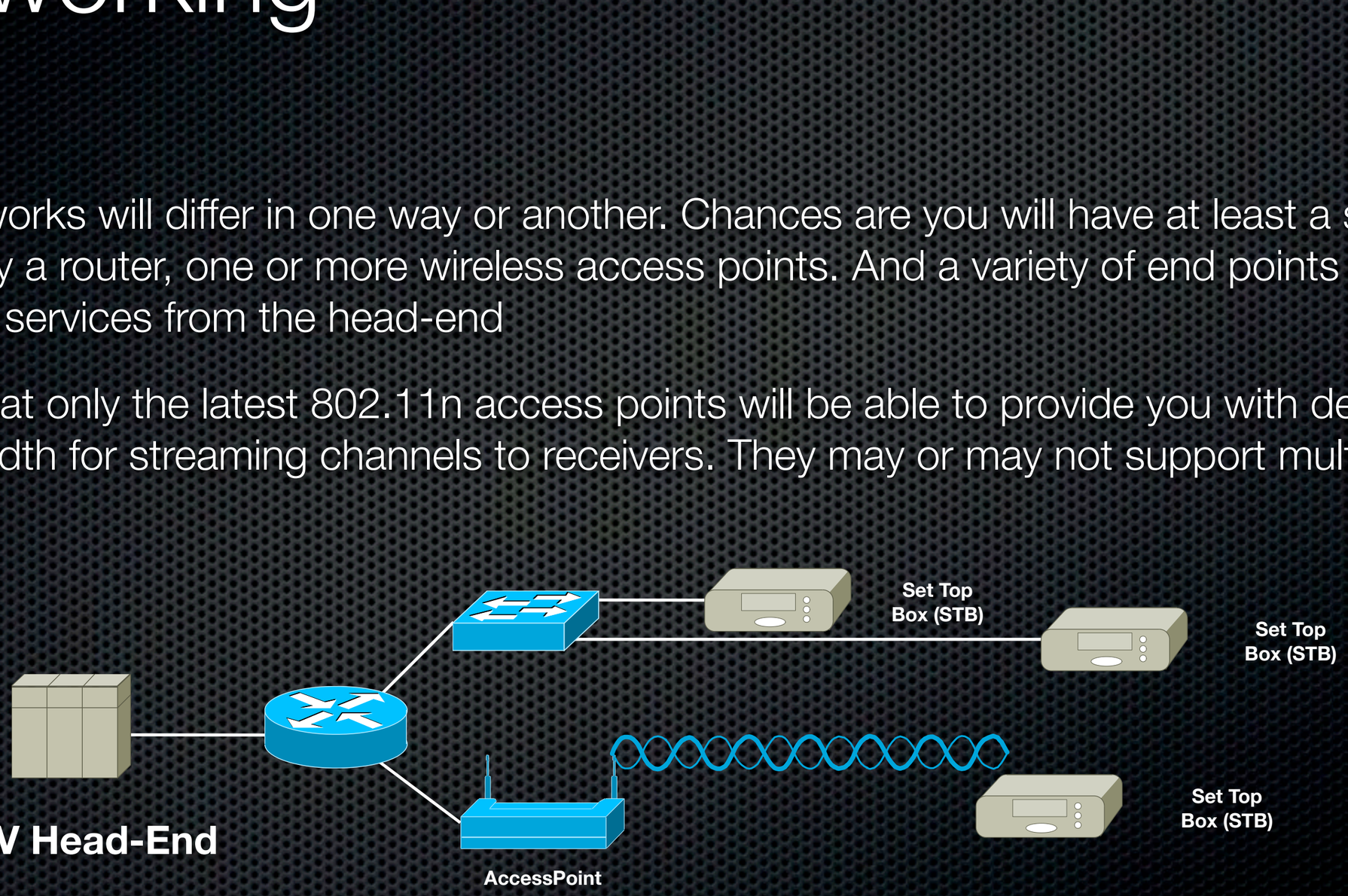
- Good server grade motherboard with enough PCI and PCIe slots for all your receivers
  - My initial design was a Supermicro C2SBC-Q (5x PCI, 2x PCIe)
  - Update: Use new generation motherboards with up to 7 PCIe slots on a single board for density
- Save on PCI/PCIe slots:
  - Use dual or quad tuner cards whenever possible
  - Use receivers with CI slots onboard/daughterboard or projects such as OSCAM
- Double check that your hardware supports the symbol rate and modulation for the services you want to receive. Check at a chip level and in the driver modules
- Read and re-read the LinuxTV.org Hardware Wiki to make sure you have the right hardware for your requirements

# Hardware

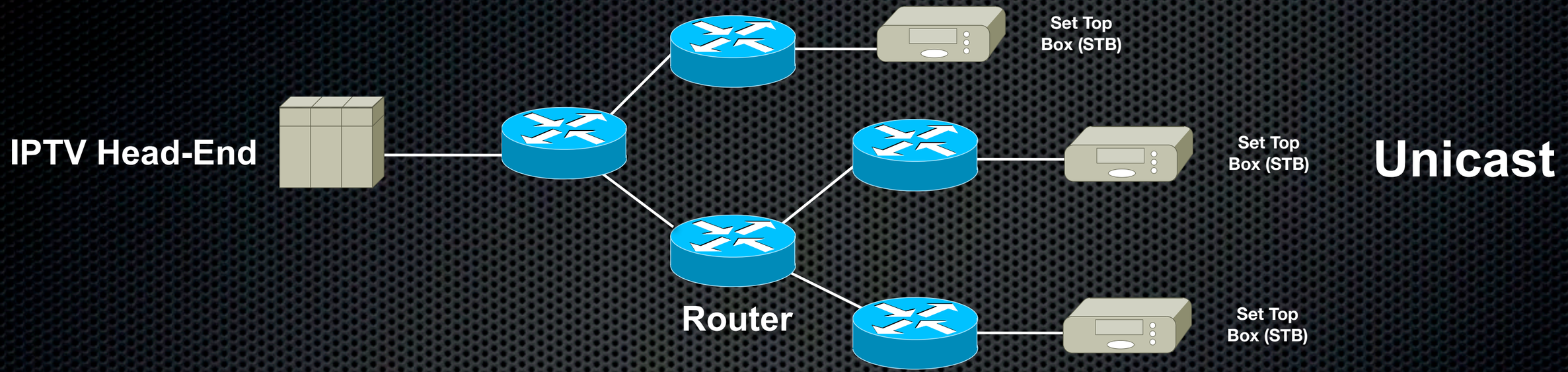
- Successfully used the following cards and CAMs (and many more), YMMV:
  - Technotrend S-1500 (PCI DVB-S) with CI daughterboard, T-1500 (PCI DVB-T), S-3200 (PCI DVB-S2)
  - TBS 6980 (PCIe DVB-S2) with dual tuner functionality
  - TBS 6925 (PCIe DVB-S2) with support for advanced DVB-S2 modes (ACM, VCM, MIS, 16/32APSK)
  - NetUp Dual DVB-S2-CI (PCIe DVB-S2) with dual tuner and dual CI slots onboard
  - DViCO FusionHDTV DVB-T Dual Digital 4 (PCI DVB-T) with dual tuner
  - Hauppauge WinTV Nova-T 500 (PCI DVB-T) with dual tuner
  - SMiT Viaccess, PowerCAM Pro, Neotion ACS 3.1
- Recommend using NetUp and TBS PCIe/PCI boards for decent hardware support

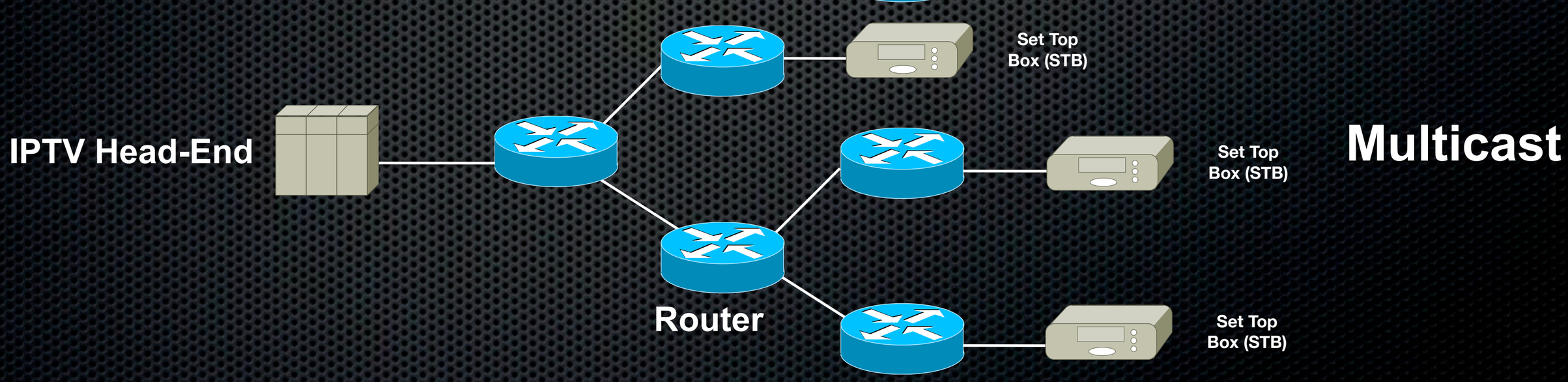
# Networking

- All networks will differ in one way or another. Chances are you will have at least a switch, possibly a router, one or more wireless access points. And a variety of end points that could receive services from the head-end
- Note that only the latest 802.11n access points will be able to provide you with decent bandwidth for streaming channels to receivers. They may or may not support multicast

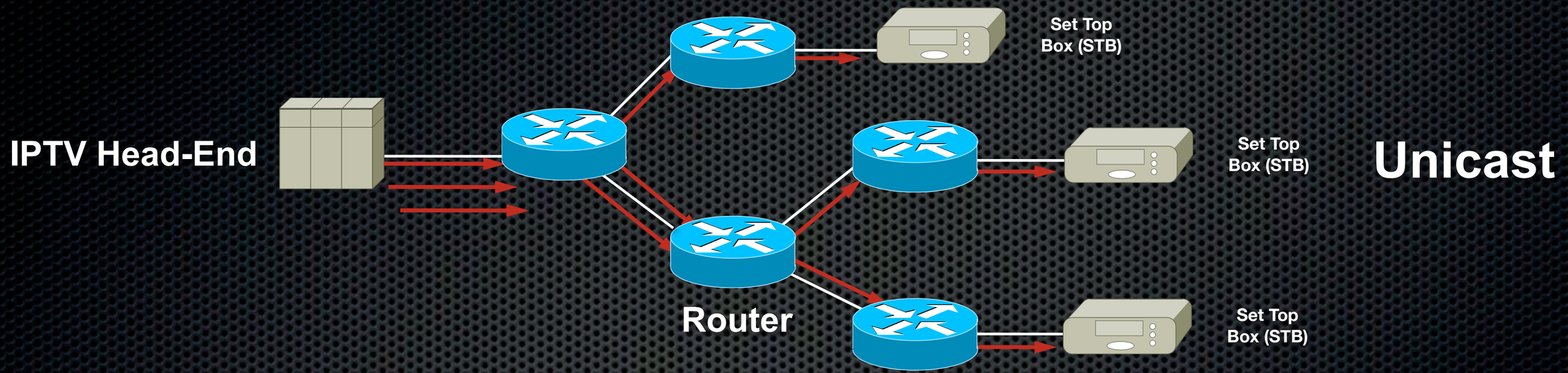


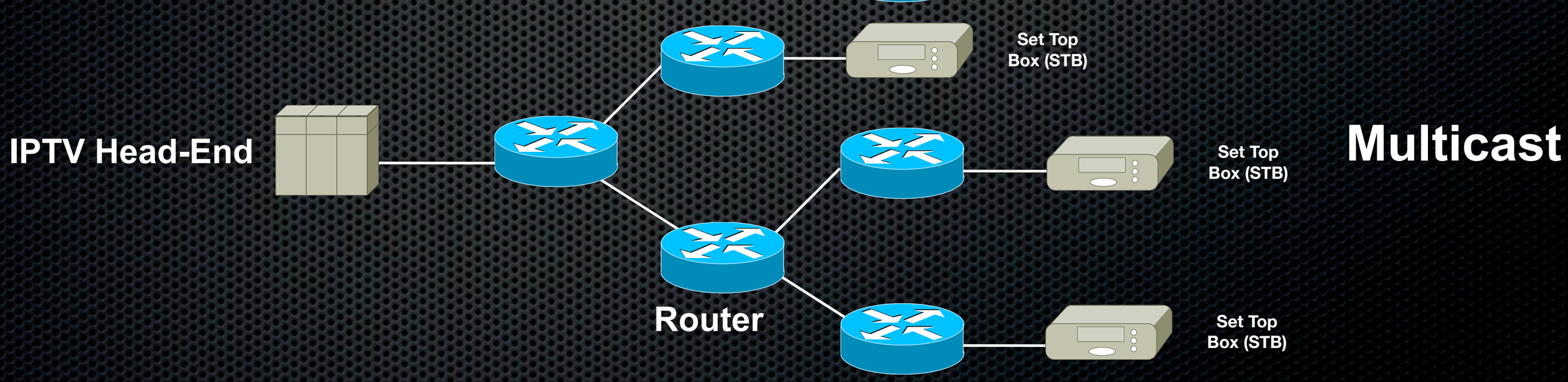
# Video streaming: Unicast vs. Multicast



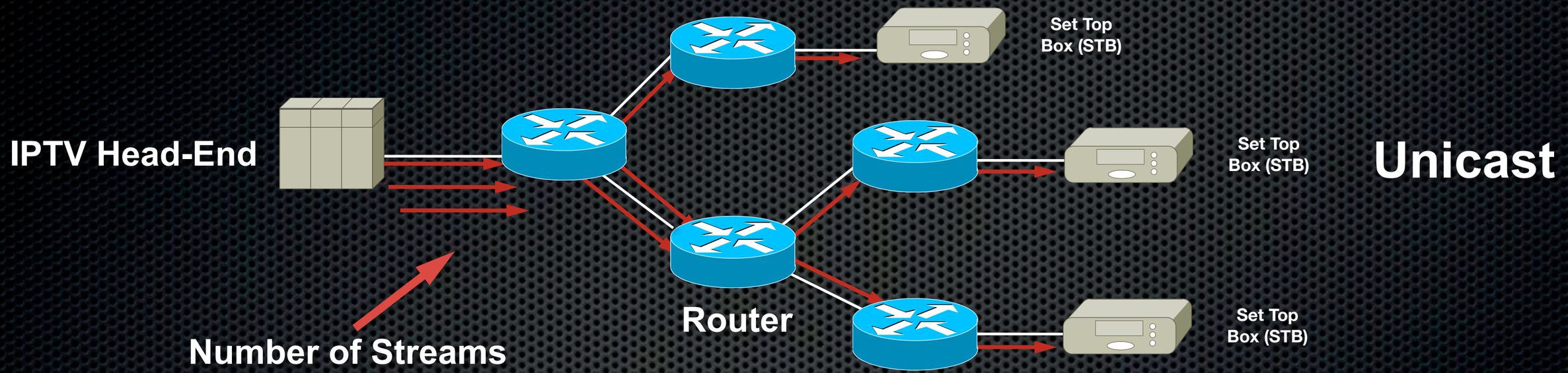



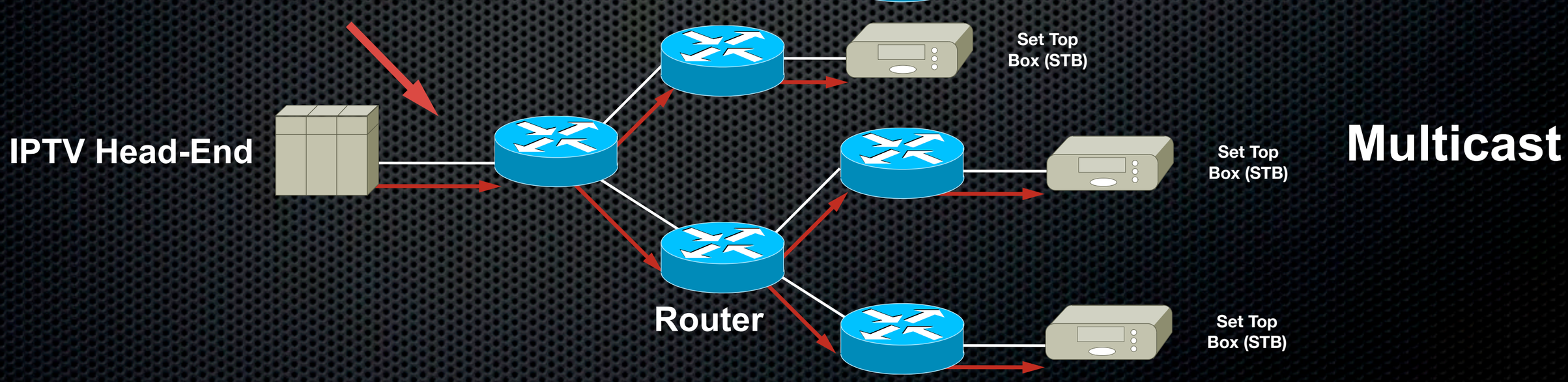
# Video streaming: Unicast vs. Multicast



# Video streaming: Unicast vs. Multicast



# To Multicast or not to Multicast...

- Multicast is the preferred manner to distribute a single source to multiple end devices, but:
  - Many low end routers and switches may not handle multicast properly (CPU vs. HW replication)
  - If no routing in your network, turn IGMP snooping on in your switch (with IGMP querier)
  - Source Specific Multicast (SSM) is strongly recommended vs. Any Source Multicast (ASM)
  - Best practice range for SSM groups is 239.232.x.x/16
  - All modern operating systems support IGMPv3 and SSM
  - Tip: make sure hosts have a route for your multicast traffic for 224.0.0.0/4:
    - “route add -net 224.0.0.0 netmask 240.0.0.0 dev ethX”
- Unicast creates one stream per receiver, even if 2 receivers are viewing the same service. Whilst this might not be an issue with a couple of receivers, this doesn’t scale in an office, school, campus, etc...

# Multicast cheat sheet for your router(s)

- Very condensed version of what you need to do on your (Cisco) router(s) to allow IP multicast traffic to flow. May or may not apply to your specific setup and hardware
- Enable multicast globally using the “ip multicast-routing”
- Enable PIM sparse-mode on each interface that sends/receives mcast traffic using “ip pim sparse-mode”
- For IGMPv3/SSM configuration:
  - Add support for IGMPv3 on each interface using “ip igmp version 3”
  - Change SSM range from default 232/8 (Internet sourced SSM range) to 239.232.x.x (Scoped)
    - “ip pim ssm range 90” and “access-list 90 permit 239.232.0.0 0.0.255.255”
- For IGMPv2/ASM configuration:
  - Add Rendez-vous Point (RP) using “ip pim rp-address x.x.x.x” where x.x.x.x is the address of the loopback interface of your device. Make sure you add it to all your devices and that it is reachable

# Software compilation environment

- In order to be able to download and install the different software packages discussed in the next section you will be the following environment.
- All the syntax used here is specific to Ubuntu, others will need to adapt the syntax.
- **Make** “sudo apt-get install make”
- **GCC** “sudo apt-get install gcc”
- **Kernel-Headers** “sudo apt-get install linux-headers-$(uname -r) build-essential”
- **Python** is needed for Mercurial “sudo apt-get install python”
- **Git** “sudo apt-get install git-core”
- **Mercurial** “sudo apt-get mercurial libncurses-dev”

# V4L-DVB

- Grab latest V4L-DVB tree
  - V4L-DVB in distribution release rapidly out of sync with new modules and bug fixes
  - Retrieve v4l-dvb source tree: “hg clone http://linuxtv.org/hg/v4l-dvb”
  - Update sources later on: “cd v4l-dvb” and then “hg pull -u http://linuxtv.org/hg/v4l-dvb”
- Build v4l-dvb kernel modules:
  - Change to v4l-dvb directory “cd v4l-dvb”, build the modules “make”, install the modules “make install”
  - If more than 8 adapters, change DVB_MAX_ADAPTERS from 8 to a higher number in the menuconfig
  - Tip: Include only modules you really need for your hardware

# V4L-DVB modules in the kernel

- [ 7.870037] saa7146: register extension 'budget_ci dvb'.
- [ 7.870109] budget_ci dvb 0000:06:01.0: PCI INT A -> GSI 74 (level, low) -> IRQ 64
- [ 7.870236] saa7146: found saa7146 @ mem f82c4000 (revision 1, irq 64) (0x13c2,0x1012).
- [ 7.870247] saa7146 (0): dma buffer size 192512
- [ 7.870251] DVB: registering new adapter (TT-Budget-T-CI PCI)
- [ 7.931683] adapter has MAC addr = 00:d0:5c:04:43:06
- [ 7.932235] input: Budget-CI dvb ir receiver saa7146 (0) as /devices/pci0000:00/0000:00:06.0/0000:05:00.0/0000:06:01.0/input/input4
- [ 8.034327] DVB: registering adapter 0 frontend 0 (Philips TDA10046H DVB-T)...
- [ 8.012786] budget_ci dvb 0000:06:02.0: PCI INT A -> GSI 78 (level, low) -> IRQ 65
- [ 8.012881] saa7146: found saa7146 @ mem f8054000 (revision 1, irq 65) (0x13c2,0x1019).
- [ 8.012892] saa7146 (1): dma buffer size 192512
- [ 8.012896] DVB: registering new adapter (TT-Budget S2-3200 PCI)
- [ 8.070942] adapter has MAC addr = 00:d0:5c:0b:5b:6d
- [ 8.071348] input: Budget-CI dvb ir receiver saa7146 (1) as /devices/pci0000:00/0000:00:06.0/0000:05:00.0/0000:06:02.0/input/input5
- [ 8.414241] stb0899_attach: Attaching STB0899
- [ 8.443895] stb6100_attach: Attaching STB6100
- [ 8.454305] LNBx2x attached on addr=8
- [ 8.454310] DVB: registering adapter 1 frontend 0 (STB0899 Multistandard)...

# DVB-APPS

- dvb-apps is a set of tools to allow you to scan for available channels and then tune to a given service. Useful for preliminary debugging when you can't tune to a given service
  - To retrieve the dvb-apps source tree: "hg clone http://linuxtv.org/hg/dvb-apps"
  - To update the sources later on: "cd dvb-apps" and then "hg pull -u http://linuxtv.org/hg/dvb-apps"
  - Change into the directory "cd dvb-apps", build the modules "make", install "sudo make install"
- Details about this package are available here: http://www.linuxtv.org/wiki/index.php/LinuxTV_dvb-apps

# Checking your devices

- DVBsnoop can allow you to check if your DVB devices have been detected by the OS
- DVBsnoop will query the frontend (RF receiver) of your DVB device for card details such as frequency range, capabilities and current parameters
- This doesn't guarantee the card is in a working state but provides feedback on the detected capabilites and if it is recognised by the OS.
- Syntax is "dvbsnoop -s feinfo -frontend /dev/dvb/adapterX/frontendX" where adapterX is the dvb device number (starting with 0, 1, 2, ...) and the frontend of that device (0 or 1 for the second tuner of a dual tuner device)

```
dvbsnoop -s feinfo -frontend /dev/dvb/adapter0/frontend0
dvbsnoop V1.4.00 -- http://dvbsnoop.sourceforge.net/
------------------------------------------------------
FrontEnd Info...
------------------------------------------------------
Device: /dev/dvb/adapter0/frontend0
Basic capabilities:
    Name: "Philips TDA10046H DVB-T"
    Frontend-type:        OFDM (DVB-T)
    Frequency (min):      51000.000 kHz
    Frequency (max):      858000.000 kHz
    Frequency stepsiz:    166.667 kHz
    Frequency tolerance: 0
    Symbol rate (min):      0.000000 MSym/s
    Symbol rate (max):      0.000000 MSym/s
    Symbol rate tolerance: 0 ppm
    Notifier delay: 0 ms
    Frontend capabilities:
      auto inversion
      FEC 1/2
      FEC 2/3
      FEC 3/4
      FEC 5/6
      FEC 7/8
      FEC AUTO
      QPSK
      QAM 16
      QAM 64
      QAM AUTO
      auto transmission mode
      auto guard interval
Current parameters:
    Frequency:  0.000 kHz
    Inversion:  OFF
    Bandwidth:  8 MHz
    Stream code rate (hi prio):  FEC 1/2
    Stream code rate (lo prio):  FEC 1/2
    Modulation:  QPSK
    Transmission mode:  2k mode
    Guard interval:  1/32
    Hierarchy:  none
```

# Testing your Scrambling modules

- As part of the dvb-apps package, you have a tool called gnutv. This command line application allows you to check your rights for your descrambling cards

```
sudo gnutv -adapter 1 -cammenu
CAM Application type: 01
CAM Application manufacturer: 02ca
CAM Manufacturer code: 3000
CAM Menu string: PowerCam_HD V2.0.4
CAM supports the following ca system ids:
  0x0500
------------------------------
PowerCam_HD V2.0.4
Select a language
0. Quit menu
1. English
2. French
3. Spanish
4. German
5. Arabic A
6. Arabic B
Select one and press 'OK' to continue
Enter option: 1
------------------------------
PowerCam_HD V2.0.4
Main Menu
0. Quit menu
1. SmartCard & PIN
2. CAS
3. VP: XXXXX
4. Download Status
5. CSP - Disabled
Select one and press 'OK' to continue
Enter option: 1
```

```
------------------------------
PowerCam_HD V2.0.4
Valid card
0. Quit menu
1. Consultation
2. Authorization
Press 'OK' to continue
Enter option: 1
Please enter your PIN code: XXXX
------------------------------
PowerCam_HD V2.0.4
Card N?:  XXXX
0. Quit menu
1. BIS
2. BIS-PREPAID
3. ViaEutelsat1
4. ViaEutelsat2
5. ViaEutelsat3
6. BIS-EXTRA
Press 'OK' to continue
Enter option: 1
------------------------------
PowerCam_HD V2.0.4
BIS
0. Quit menu
1. Subscription
2. Pre-booking
Press 'OK' to continue
Enter option: 1
```

```
------------------------------
PowerCam_HD V2.0.4
0. Quit menu
1. Processing...
Please wait ...
Enter option:
 ------------------------------
PowerCam_HD V2.0.4
Subscription BIS
0. Quit menu
1. Subscription per class :
2. 11/22/2009 - 12/31/2009 Class:
3. 100,50
4. 09/22/2009 - 10/31/2009 Class:
5. 100,50
6. 09/13/2009 - 09/22/2009 Class:
7. 210,150,103,102,101,100
8. 50,2
9. 08/22/2009 - 09/30/2009 Class:
10. 100,50
Press 'OK' to continue
Enter option:
Main Menu
0. Quit menu
1. SmartCard & PIN
2. CAS
3. VP: 88730
4. Download Status
5. CSP - Disabled
Select one and press 'OK' to continue
Enter option: 1
```

# MuMuDVB

- MuMuDVB provides RTP/UDP multicast and HTTP unicast streaming capabilities. It takes all services from a transport stream and maps each service to a unique multicast group
- By default all services are streamed. Options allow to specify specifically which services are required (multiple audios, subtitles, Teletext, data, ...)
- Session Announcement Protocol (SAP) for receivers that can listen & build a services listing
- Download from source: “git clone http://mumudvbgit.braice.net/mumudvb”, “cd mumudvb”, “autoreconf -i -f”, “./configure”, “make”, “sudo make install”
- To launch the application for initial testing (debug) use the following:
  - “mumudvb -d -vvv -s -c <config_file>”
  - Once you have confirmation the setup is working you can skip the -d value

# MuMuDVB

- New builds now offer:
  - New DVB parser
  - Software based transcoding feature (needs some updating)
  - Supports Software descrambling (SASC-NG)
  - Building play lists for VLC Player and MythTV formats
  - Support for OpenWRT platform for mobile setups
  - RTP Timestamps
  - Override/specify adapter to use in the Command Line (vs. config file)
  - IPv6 support including SAP announcements
  - RF Signal level and error rate (if tuner/module supports it)
  - DVB-T2 support

# MuMuDVB output

MuMuDVB Version 1.6.1b_20100614
 --- Build information ---
Built with CAM support.
Built without transcoding support.
Built with ATSC support.
Built with ATSC long channel names support.
Built with support for DVB API Version 5 (DVB-S2).
Built with pthread support (used for periodic signal strength display, cam support, transcoding, and threaded read).
---

(snip)

Info: Multicast: You decided to send the RTP header (multicast only).
Info: SAP: Sap announces will be sent
Info: You have enabled the PAT Rewriting
Info: Main: Full autoconfiguration, we activate SDT rewritting. if you want to desactivate it see the README.
Info: Main: Full autoconfiguration, we activate sorting of the EIT PID. if you want to desactivate it see the README.
Info: Autoconf: The autoconfiguration auto update is enabled. If you want to disable it put "autoconf_pid_update=0" in your config file.
Info: Main: Streaming. Freq 10773000
Info: Tune: Using DVB card "ST STV0299 DVB-S"
Info: Tune: Tuning DVB-S to Freq: 1023000 kHz, LO frequency 9750000 kHz Pol:H Srate=22000000, LNB number: 3
Info: Tune: LNB voltage 18V
Info: Tune: DISEQC SETTING SUCCEDED
Info: Tune: FE_STATUS:
Info: Tune: Strength:  58880
Info: Tune: SNR:  59415
Info: Tune: FE_STATUS:
Info: Tune:  FE_HAS_SIGNAL : found something above the noise level
Info: Tune:  FE_HAS_CARRIER : found a DVB signal
Info: Tune:  FE_HAS_VITERBI : FEC is stable
Info: Tune:  FE_HAS_SYNC : found sync bytes
Info: Tune:  FE_HAS_LOCK : everything's working...
Info: Tune: Strength:  58907
Info: Tune: SNR:  59565
Info: Tune: Event:  Frequency: 10772919 (or 8727081)
Info: Tune:  SymbolRate: 22000000
Info: Tune:  FEC_inner:  5
Info: Tune: Bit error rate: 0
Info: Tune: Signal strength: 58891
Info: Tune: SNR: 59565
Info: Tune: FE_STATUS:
Info: Tune:  FE_HAS_SIGNAL : found something above the noise level
Info: Tune:  FE_HAS_CARRIER : found a DVB signal
Info: Tune:  FE_HAS_VITERBI : FEC is stable
Info: Tune:  FE_HAS_SYNC : found sync bytes
Info: Tune:  FE_HAS_LOCK : everything's working...
Info: Main: Card 0 tuned
Info: Main: The traffic will be shown every 10 seconds
Info: Autoconf: Autoconfiguration Start
Info: DVB: Bit error rate:  0 Signal strength:  59198 SNR:  59565
Info: Autoconf: Autoconfiguration done
Info: Autoconf: Diffusion 8 channels
Info: Autoconf: Channel number :  0, name : "BBC 1 London"  service id 6301
Info: Autoconf:  Multicast ip : 239.100.1.0:1234
Info: Autoconf: Channel number :  1, name : "BBC 2 England"  service id 6302
Info: Autoconf:  Multicast ip : 239.100.1.1:1234
Info: Autoconf: Channel number :  2, name : "ETV"  service id 6306
Info: Autoconf:  Multicast ip : 239.100.1.2:1234
Info: Autoconf: Channel number :  3, name : "BBC TES 3"  service id 6315
Info: Autoconf:  Multicast ip : 239.100.1.3:1234
Info: Autoconf: Channel number :  4, name : "BBC FOUR"  service id 6316
Info: Autoconf:  Multicast ip : 239.100.1.4:1234
Info: Autoconf: Channel number :  5, name : "BBC THREE"  service id 6319
Info: Autoconf:  Multicast ip : 239.100.1.5:1234
Info: Autoconf: Channel number :  6, name : "BBC 1 NI"  service id 6331
Info: Autoconf:  Multicast ip : 239.100.1.6:1234
Info: Autoconf: Channel number :  7, name : "FIVE"  service id 6335
Info: Autoconf:  Multicast ip : 239.100.1.7:1234
Info: DVB: Bit error rate:  0 Signal strength:  58891 SNR:  59574

# DVBlast

- DVBlast is a simple and powerful MPEG-2 TS demux and streaming application
  - Designed to open DVB device, tune, place PIDs filters, config CAM and demux to output
  - No processing of the streams, transcoding, PID remapping or remultiplexing
  - Focus is on stability and minimal tampering with the stream source
- Supports V4L-DVB devices (e.g.: S/S2, T/T2, ...), DVB-ASI and UDP/RTP IP inputs
- CAM menus (MMI) and front-end status support via external application
- Download latest released code here: http://downloads.videolan.org/pub/videolan/dvblast/2.2/dvblast-2.2.tar.bz2
  - No configuration options, “make & make install” and your done

# biTStream

- Required by DVBlast since r154
  - Replaces unique previous DVBlast external dependency on libdvbpsi
- biTStream is a set of C headers allowing a simpler access to binary structures such as specified by MPEG, DVB, IETF, etc
- Download via “git clone git://git.videolan.org/bitstream.git”
- Install using “make install”, no autoconf for now
- Always update before you install new release of DVBlast

# DVBlast

- Command line for tuning parameters, config file for channels to be streamed
- “dvblast -u -f 10773000 -a 0 -n 0 -s 22000000 -v 18 -t 255 -S 1 -e -c <config.cfg>”
  - -u budget mode (no hardware filtering)
  - -f = frequency, a = adapter number, n = frontend, s = symbol rate
  - -v 13|18 voltage to be applied to trigger Vertical|Horizontal polarisation
  - -t = Multicast TTL, -S satellite number (0, 1-4)
  - -e enable EPG pass through (EIT data)
  - -c configuration file containing the specific services you want to multicast
- Options:
  - For DVB-S2 you must indicate the modulation (QPSK or PSK_8)
  - For DVB-T the bandwidth of the multiplex needs to be added (usually 8MHz)

# DVBlast (config file)

- Format is the following: <IP>[:<port>][/udp]  <always on>  <SID>  [<PID>,]*
  - There are three ways of configuring the PIDs to stream :
  - 1. SID-based
  - 239.232.0.1:1234  1  10750
  - DVBlast will stream all known PIDs from service 10750 (video, audio, and subtitles). The resulting stream is fully MPEG-compliant, with PAT and PMT.
  - 2. SID and PIDs
  - 239.232.0.1:1234  1  10750  1234,1235
  - DVBlast will stream SID 10750, but only PID 1234 and 1235 will be output. Other known PIDs will be discarded and removed from the PMT. The list of PIDs in the config file does not include the PAT and PMT, but it must include the PCR PID if it is different from the video or audio PID, otherwise the stream won't be compliant.
  - 3. PIDs only
  - 239.232.0.1:1234  1  0  0,128,1234,1235
  - DVBlast will only stream the PIDs passed. No PAT and PMT will be generated, so if they are not included the stream won't be compliant. Also the included PAT and PMT may contain ghost programs or ESes.

# DVBlast output

```
sudo dvblast -f 10773000 -n 0 -e -t 255 -v 18 -s 22000000 -S 3 -C -c BBC.cfg -r /tmp/dvblast.sock -a 0
DVBlast 2.2 (git-2.2-2-g38ef649)
warning: restarting
debug: using linux-dvb API version 5.4
debug: Frontend "STB0899 Multistandard" type "QPSK (DVB-S/S2)" supports:
debug:  frequency min: 950000, max: 2150000, stepsize: 0, tolerance: 0
debug:  symbolrate min: 5000000, max: 45000000, tolerance: 0
<snip>
debug: frequency 10773000 is in Ku-band (lower)
debug: configuring LNB to v=18 p=0 satnum=3
debug: tuning QPSK frontend to f=10773000 srate=22000000 inversion=-1 fec=999 rolloff=35 modulation=legacy pilot=-1
<snip>
debug: conf: 239.232.232.21:3000 config=0x61 sid=6301 pids[0]
<snip>
debug: frontend has acquired carrier
debug: frontend has acquired stable FEC
debug: frontend has acquired sync
info: frontend has acquired lock
frontend has acquired lock
debug: - Bit error rate: 0
debug: - Signal strength: 211
debug: - SNR: 117
<snip>
debug: new PAT tsid=2045 version=11
debug:   * program number=6301 pid=256
<snip>
```

BBC.cfg Configuration file

```
:BBC 1 London
239.232.232.21:3000          1     6301
:BBC 2 England
239.232.232.22:3000          1     6302
:BBC THREE
239.232.232.23:3000          1     6319
:BBC FOUR
239.232.232.24:3000          1     6316
:FIVE
239.232.232.25:3000          1     6335
```

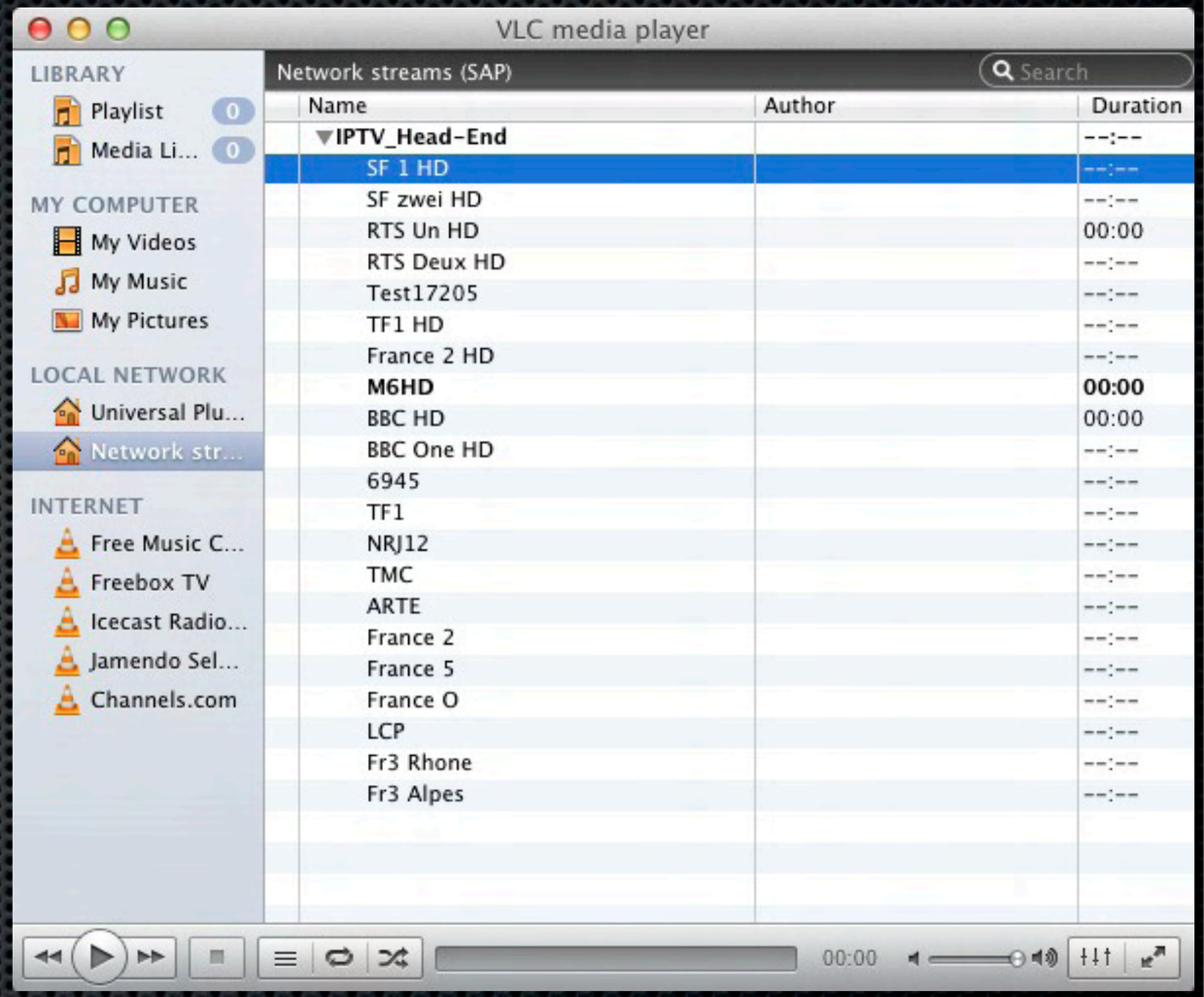
# DVBlast updates...

- Introduced a buffering scheme to smooth packet output
- Added support for per output options
- Added ATSC A/52 stream types
- Added option to build MRTG graph files
- Moved from SVN to GIT some time ago
  - “git clone git://git.videolan.org/dvblast.git”
- Better support for descrambling by:
  - supporting asynchronous I/O CAM operations
  - removing limitations on number of CA systems IDs and programs

# Media Players

# VLC Media Player

- A well known Open Source media player, supports most OSes, a large number of file formats, audio and video codecs
- In the options enable support for SAP announcement to pickup the services via the playlist menu (that's if sessions are announced and being received)
- If no SAP service, you can load a file containing all the services settings or enter manually
- GPU decode acceleration introduced in VLC 1.1 for Windows Vista/7 and Linux (Nvidia and AMD/ATI) and planned for OS X in 2.1 release



# VLC Media Player - Services list format

- Using the Extended M3U Playlist format, you can create a service list for VLC
- Load the file directly into VLC locally or remotely over your network
- This will load the services list into the VLC playlist and allow you to select the services you wish to view
- Double click on the service and your up and running

```
#EXTM3U
#EXTINF:0,1 - Channel Name
rtp://@239.192.1.1:1234
#EXTINF:0,2 - Channel Name
rtp://@239.192.1.2:1234
#EXTINF:0,3 - Channel Name
rtp://@239.192.1.3:1234
#EXTINF:0,4 - Channel Name
rtp://@239.192.1.4:1234
#EXTINF:0,5 - Channel Name
rtp://@239.192.1.5:1234
```

```
#EXTM3U
#EXTINF:0,1 - Channel Name
rtp://192.168.1.100@239.232.1.1:1234
#EXTINF:0,2 - Channel Name
rtp://192.168.1.100@239.232.1.2:1234
#EXTINF:0,3 - Channel Name
rtp://192.168.1.100@239.232.1.3:1234
#EXTINF:0,4 - Channel Name
rtp://192.168.1.100@239.232.1.4:1234
#EXTINF:0,5 - Channel Name
rtp://192.168.1.100@239.232.1.5:1234
```

# MythTV

- Open Source software digital video recorder (DVR). Has been under development since 2002 and contains most features a modern DVR would provide
- Split in 2 main components:
  - MythBackend: Server applications dealing with scheduling, recording, keeping TV listings up to date
  - MythFrontend: Provides end-user interface and communicates with one or more backends
- Since version 0.21 and with some extra work you can integrate IPTV multicast sources as a valid input for MythTV
- MythTV wiki provides further details on how to do this for some IPTV service provides, so does a link in the reference section. This will require some tweaking to get it running

# Android client

- After much trial and error:
  - Google Galaxy Nexus phone running Android 4.0.4
  - Wifi network with multicast support enabled
  - RTP encapsulated streams
- 2 possible clients:
  - VLC for Android using post v0.1 nightly builds (still under development)
  - Daroon Player

# Support for IPv6

# Network support for IPv6 (quick intro)

- Multicast is not always well (or at all) supported on some networking devices, this applies even more to IPv6 multicast
- First select the scope you wish to use for the distribution of your content
  - Important to contain your multicast traffic within specific boundaries
- Enable IPv6 multicast globally using the “ipv6 multicast-routing”
- IPv6 needs to be configured and enabled on all the networked interfaces that will be in use
- PIM Sparse Mode and MLDv2 enabled by default in Cisco IOS (when features are supported)
- For MLDv2/SSM configuration (Supported in Windows Vista/7 and Linux 3.0 kernels)
- For IGMPv2/ASM configuration (Rendez-Vous Point)
  - Add Rendez-vous Point (RP) using “ipv6 pim rp-address x.x.x.x”. Same instructions as IPv4 syntax

# Head-end & receiver support for IPv6

- Good support for IPv6 in the Linux 3.0 kernel branch used in current distributions
- DVBlast supports IPv6 since r93 commit of the code. Available in current releases
- No change on DVBlast command line syntax, only config file syntax for the multicast address. Note that the IPv6 address must be in between brackets
  - Ex: [FF15::ABCD]:1234 1 10750
- VLC Media Player supports IPv6 out of the box
- MythTV supports IPv6 with a bit of extra work
- Some STB vendors also support IPv6 multicast, still patchy

# What needs worked on (IMHO)

# Areas of improvement

- Web based EPG interface independent of receiver application/player
  - Grab DVB-SI data from multicast sources, create XMLTV output
  - Publish to web interface for users and feeding home media systems
  - Merge with external EPG provider data sources
    - Dynamic channel lineup based on user preferences
    - Output syntax in multiple formats for various players
- Better IPv6 and/or multicast support
  - In wider set of receivers (STB, applications, smartphones, tablets)
  - Entry level networking equipment

# Summary

- Basics related to building a video head-end
- Advantages and constraints related to such a setup
- Keep in mind your own requirements: services, network, receivers
- What needs worked on to expand ecosystem (IMHO)
- Have fun with your own project, please document and share with others....

# References

# References

- DVB-T: http://en.wikipedia.org/wiki/DVB-T & http://en.wikipedia.org/wiki/DVB-T2
- DVB-S2: http://en.wikipedia.org/wiki/DVB-S2
- DVB-C: http://en.wikipedia.org/wiki/DVB-C
- DVB-CI: http://en.wikipedia.org/wiki/DVB-CI
- CAM: http://en.wikipedia.org/wiki/Conditional_Access_Module
- King of Sat: http://en.kingofsat.net/
- Aston Professional: http://www.aston-france.com/uk-modules-professionnels.html
- SMiT http://www.smit.com.cn/index.php?lang=en

# References

- V4L-DVB: http://www.linuxtv.org/
- Linux-media ML archives: http://www.mail-archive.com/linux-media@vger.kernel.org/
- MuMuDVB: http://mumudvb.braice.net/
- DVBlast:  http://www.videolan.org/projects/dvblast.html
- TSDecrypt: http://georgi.unixsol.org/programs/tsdecrypt/
- VLC: http://www.videolan.org/
- MythTV: http://www.mythtv.org/
- IPTV with MythTV: http://www.avenard.org/iptv/MythTV.html
- TVheadend: http://www.lonelycoder.com/hts/tvheadend_overview.html
- Wireshark: http://www.wireshark.org/