

Streaming Technologies

Pujan Srivastava
pujan@ait.asia

Outline

- ◆ Real-Time Vs Offline
- ◆ Basics
- ◆ Multicast
- ◆ Application Gateway
- ◆ Introduction to Videos
- ◆ Examples

Introduction

With the advancement of Computing Resources, now all personal computers have ability to produce videos and thus streaming.

Video processing requires several professional streaming steps mainly–

1. Studio Setup – Source
2. Video processing – Output
3. Streaming – Distribution

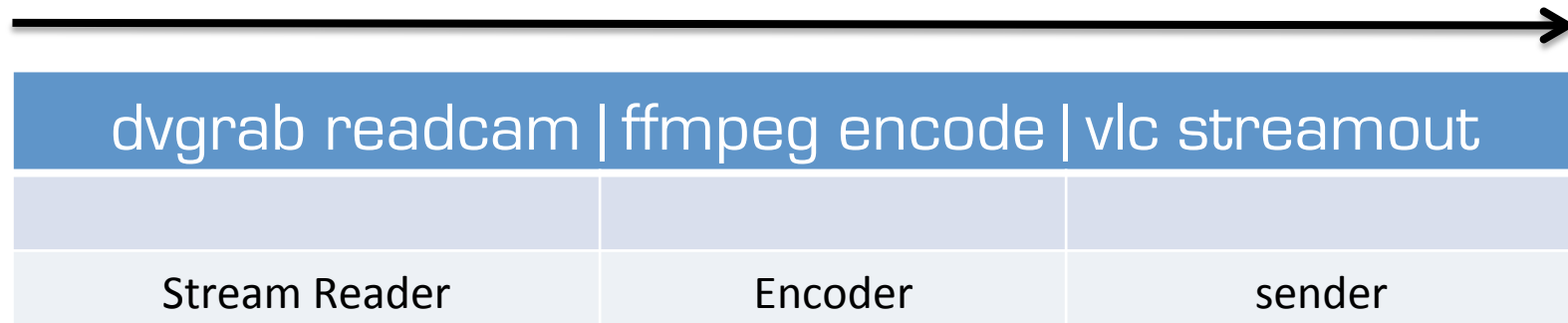
Taking advantage of Linux Piping

- If you want to redirect the output of one program into the input of another, just use a simple pipeline:

○ *program 1* arg arg | *program 2* arg arg | *program 3* arg arg

- *All programs in a pipeline run simultaneously.*
- Most programs typically use *blocking* I/O: if when they try to read their input and nothing is there, they *block*: that is, they stop, and the operating system de-schedules them to run until more input becomes available (to avoid eating up the CPU).
- If a program earlier in the pipeline is writing data faster than a later program can read it, eventually the pipe's buffer fills up and the writer blocks: the OS de-schedules it until the pipe's buffer gets emptied by the reader, and then it can continue writing again.

Taking Advantage of FFMPEG



```
dvgrab -format dv1 - | ffmpeg -f dv -i - -target ntsc-dvd -qscale 5 - |  
cvlc --sout "#duplicate{dst=std{dst=203.159.31.78:1234,mux=ts,access=udp}}"
```

"All programs in a pipeline run simultaneously".

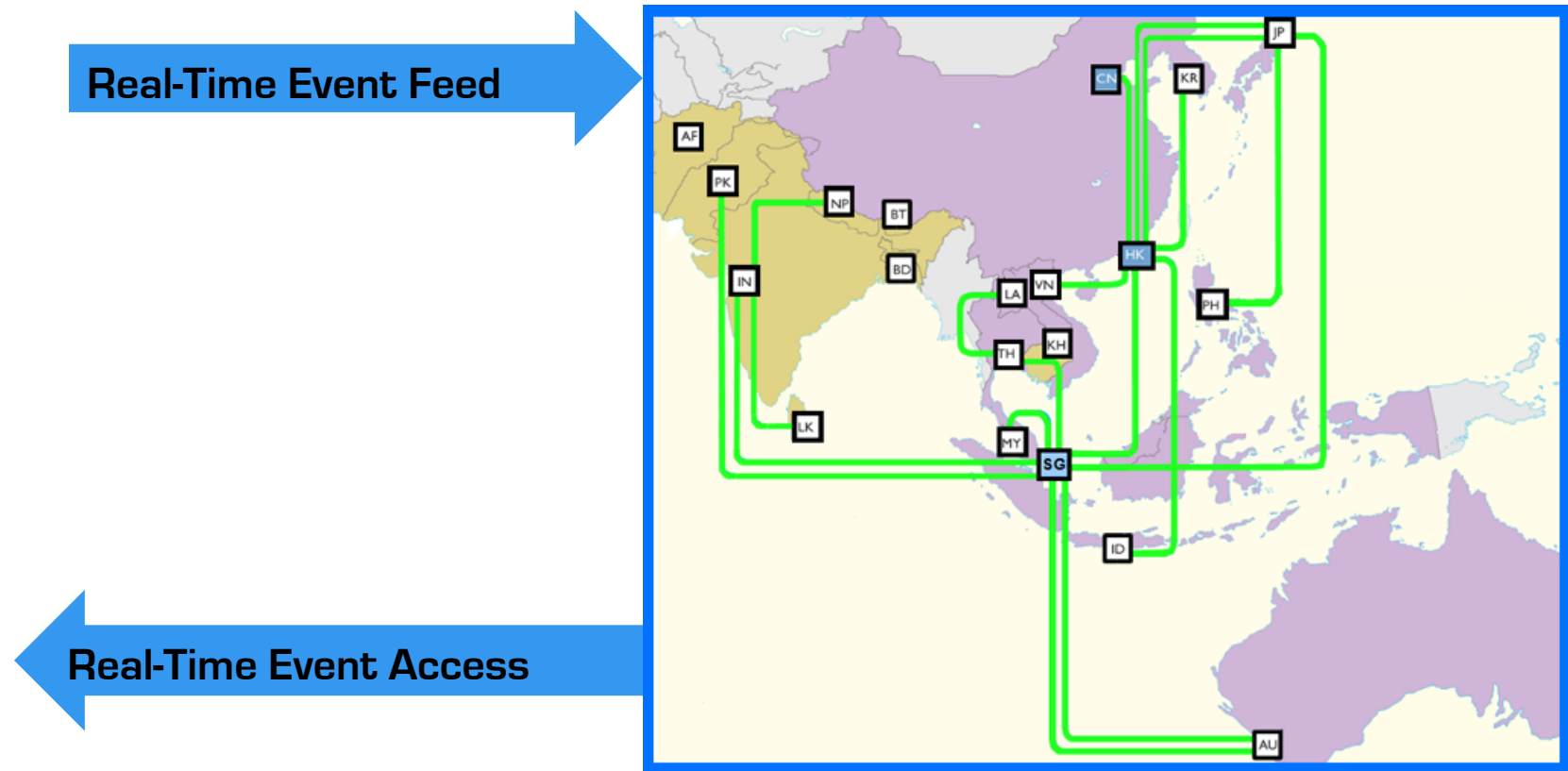
Here dvgrab, ffmpeg, cvlc are running simultaneously. Cvlc is non-gui version of VLC

Before the Streaming

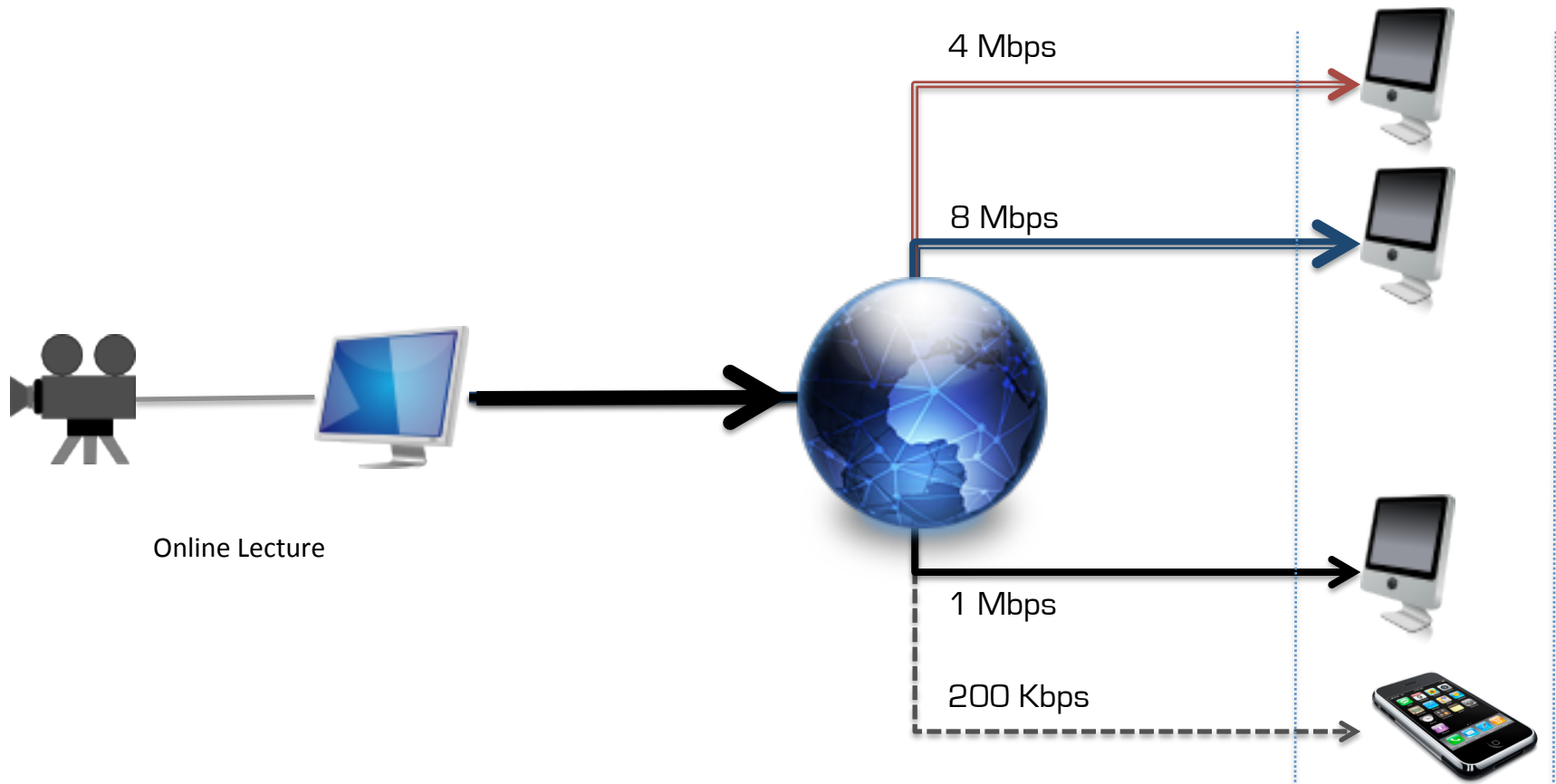
Know your audience and their characteristics –

1. Real-Time Events Vs Offline
2. Network Characteristics (Unicast/Multicast/HTTP/All)
3. Device Characteristics (Computer/Mobile)
4. Delay Sensitive/Insensitive (Type of event: Telemedicine?)
5. Interactive or Listener ?

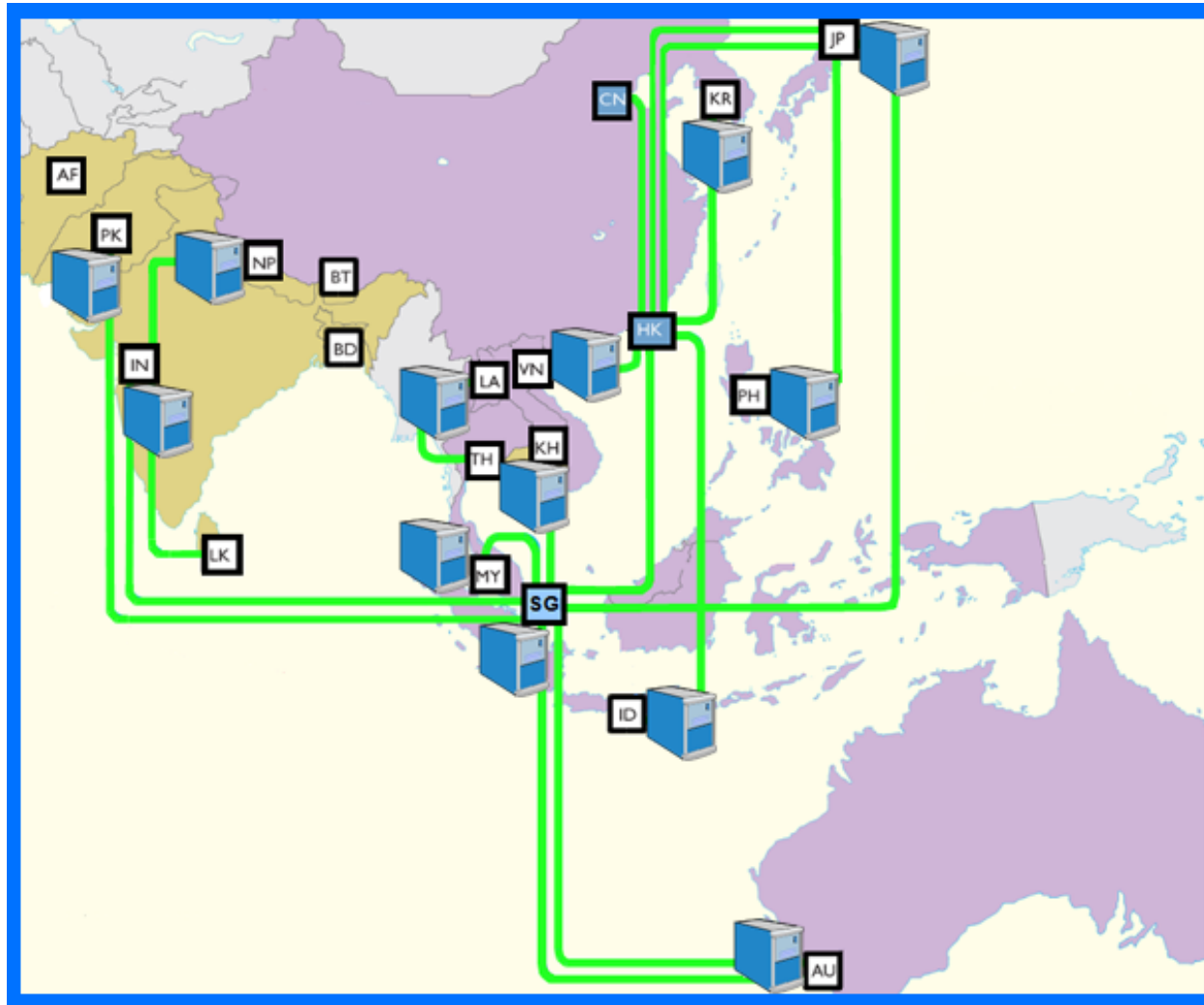
Purpose



Purpose - Heterogeneity



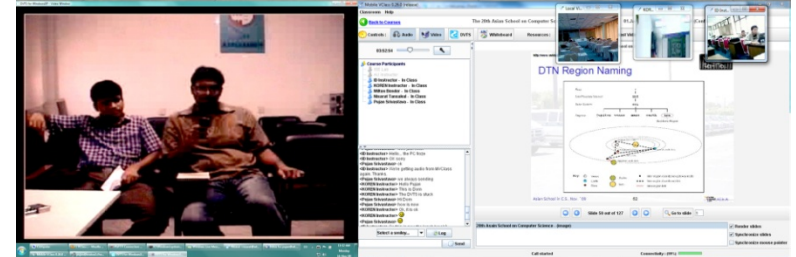
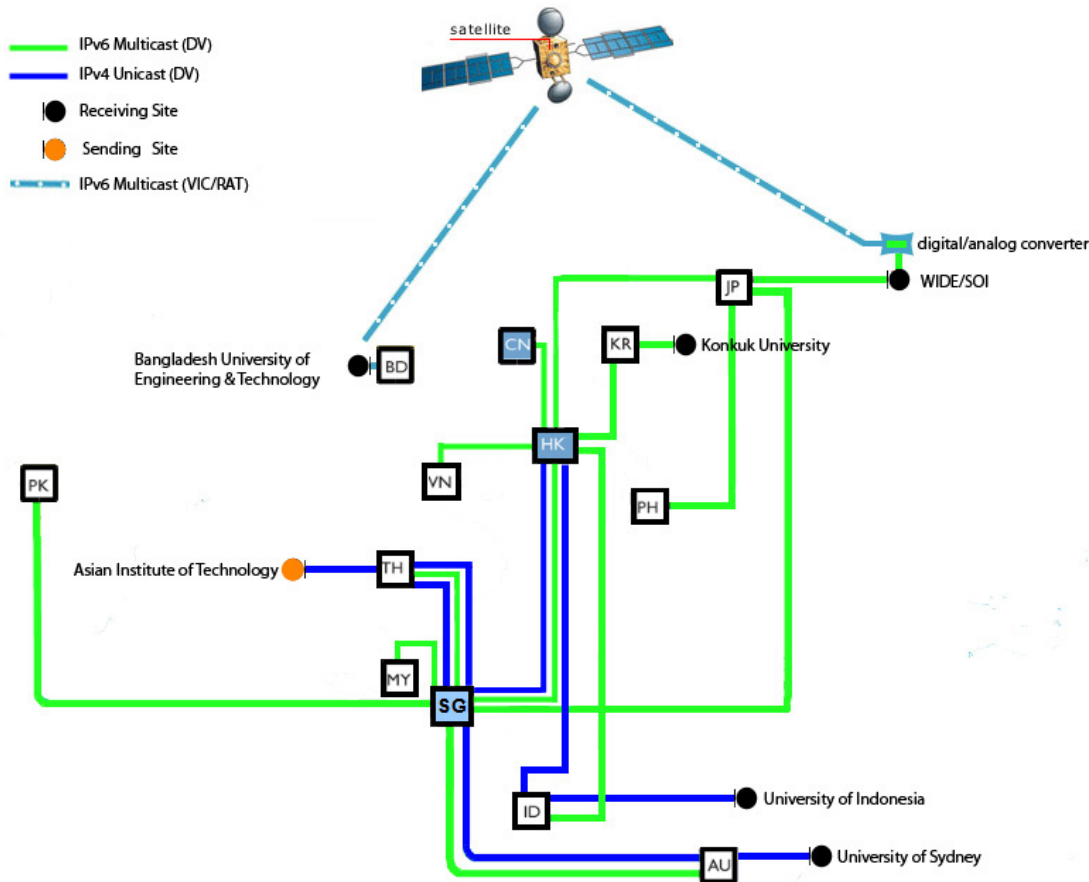
Purpose - Heterogeneity



20th Asian School on Computer Science

15-16 November 2009
AIT, Thailand

Remote: Konkuk University, Korea
University of Sydney, Australia
BUET, Bangladesh (through SOI Asia)
University of Indonesia



CanalAVIST Project

- Biweekly HD Streaming – IPv6 Multicast, HTTP, UDP
 - Real-time & Interactive Event Broadcasting – HD and Low Q
- www.canalavist.org

CanalAVIST coming together to share learning, experiences and resources to develop our people.

- Home
- CanalAVIST +
- Members +
- Events +
- CanalAVIST Map
- News
- Streams
- Calendar
- Sponsors

Gold

Silver

Bronze

CanalAVIST

In ASEAN countries, there is a wide diversity of availability of talks from experts and luminary speakers. From every other day in more developed countries, to once a blue moon in some remote universities in developing countries, the disparity amongst haves and have-nots is great enough to warrant an e-solution.

CanalAVIST is a part of the [ASEAN Virtual Institute of Science and Technology \(AVIST\)](#) and the ASEAN Science and Technology Research and Education Network Alliance (ASTRENA) under ASEAN Committee for Science and Technology to provide channels for seamless education, teaching, training, conferencing, lectures, talks through ASEAN countries for ASEAN researchers and students.

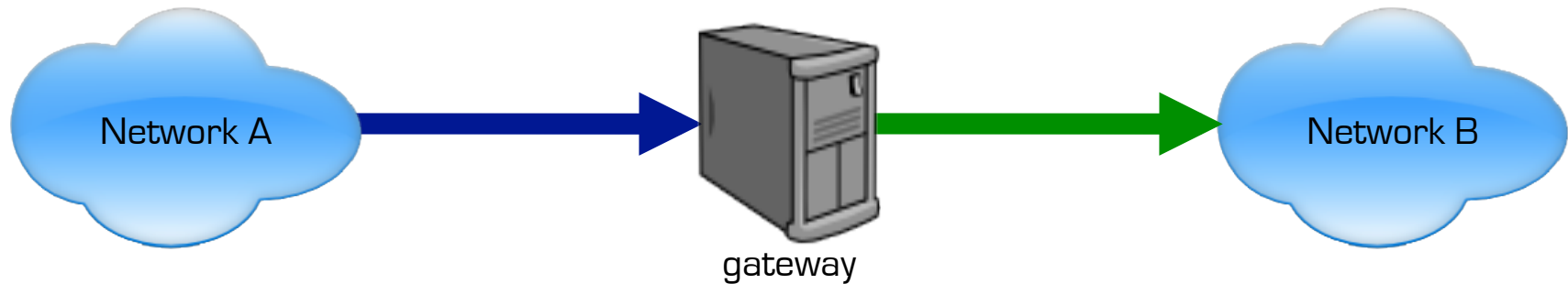
The ASEAN Virtual Institute of Science and Technology (AVIST) was set up by ministerial approval in 2005, following development by ASEAN Committee on Science and Technology (COST) as part of the strategy for continuing professional education, hosted in Asian Institute of Technology (AIT), Bangkok Thailand. Since then, AVIST has conducted various online e-learning courses to a large number of participants throughout ASEAN. The ASEAN Science and Technology Research and Education Network Alliance (ASTRENA) was mooted in 1998 and implemented in 2006. Since then, two ASTRENA meetings have been carried out, ASTRENA-1 in Brunei, March 2006 and ASTRENA-2 in Manila, January 2007. Other ASTRENA associated meetings have also been conducted.

Streaming VLC/DVTS Application Gateways

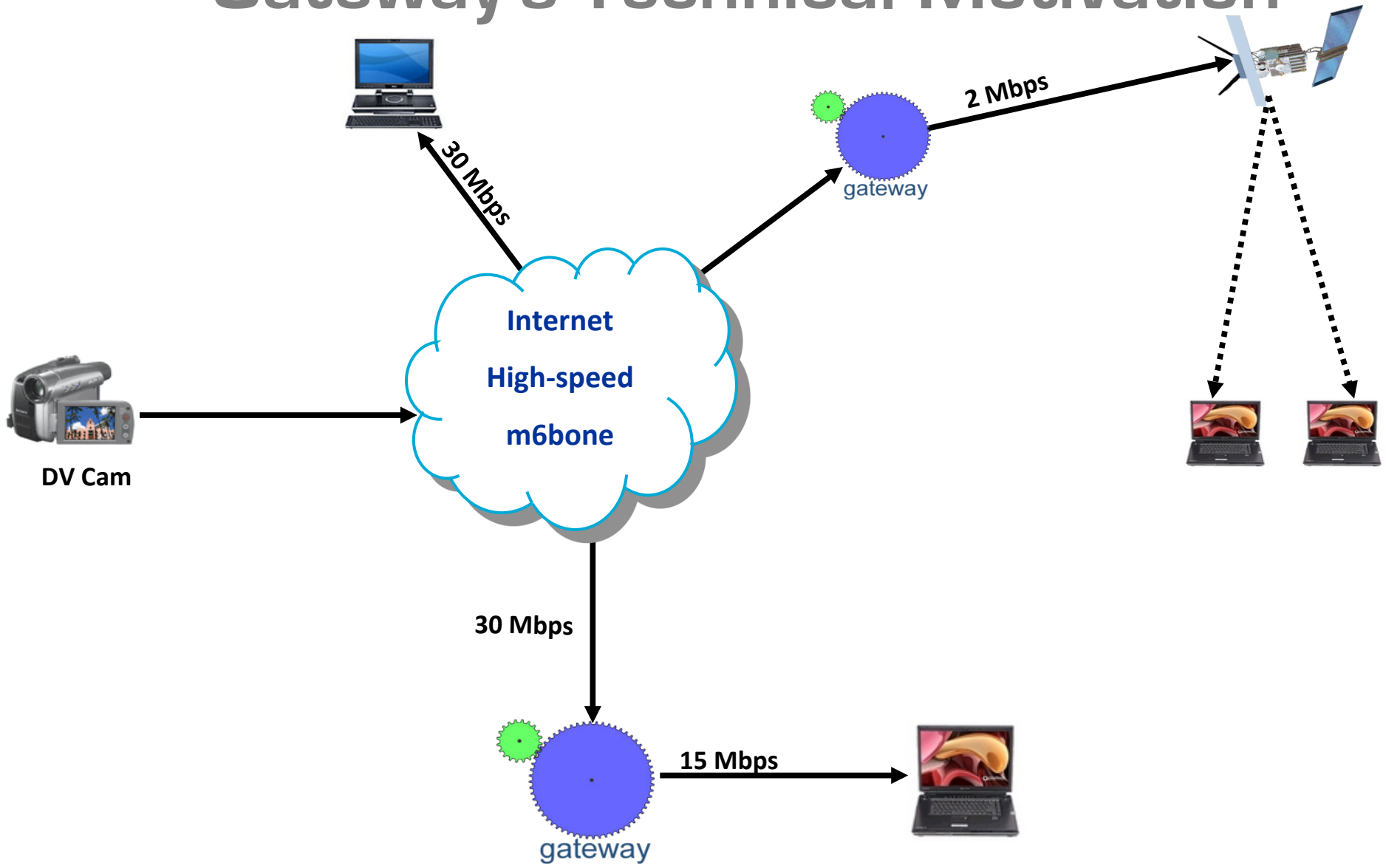
Application Gateway

Relays Video stream

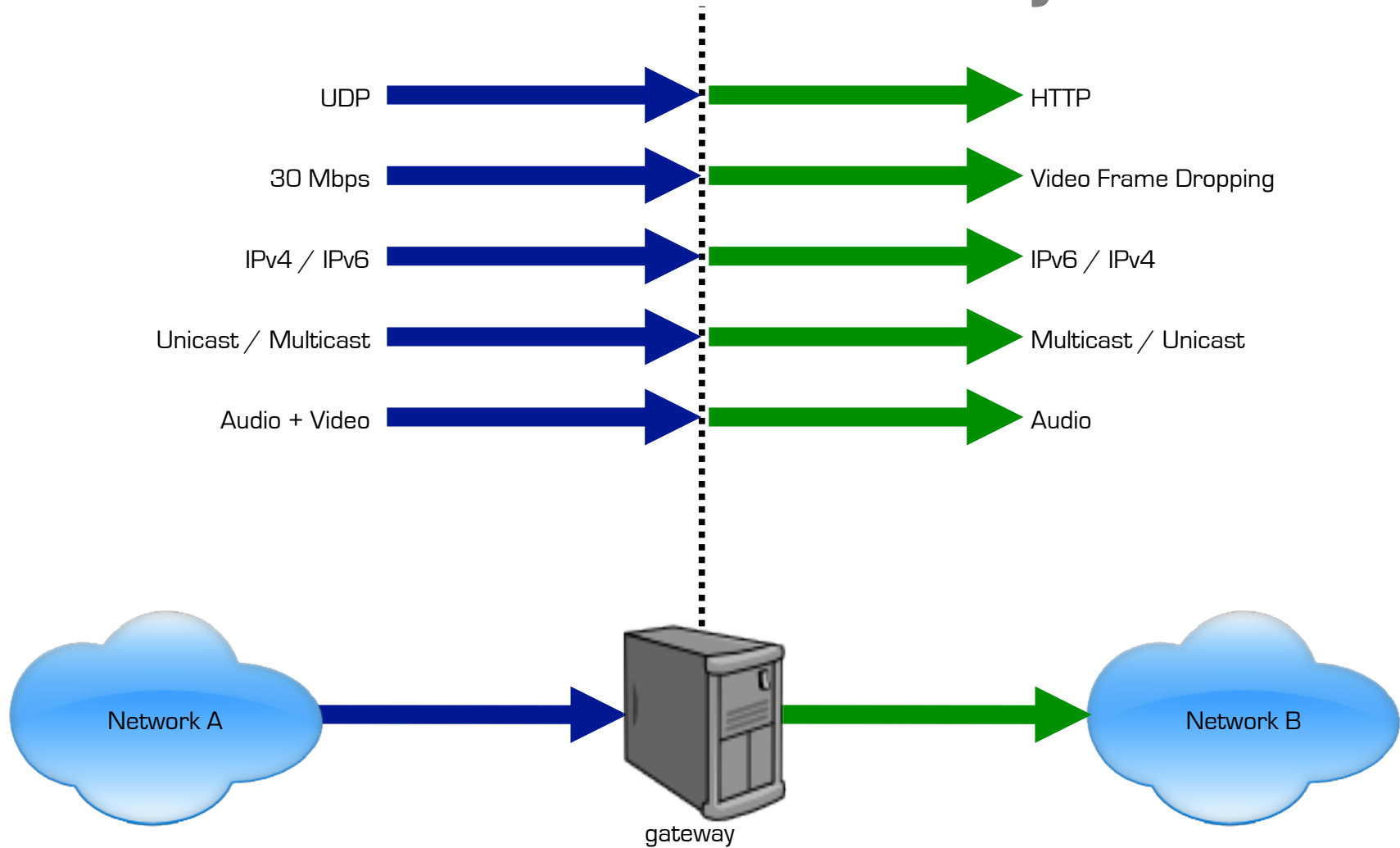
Linux/Mac OSX OSs



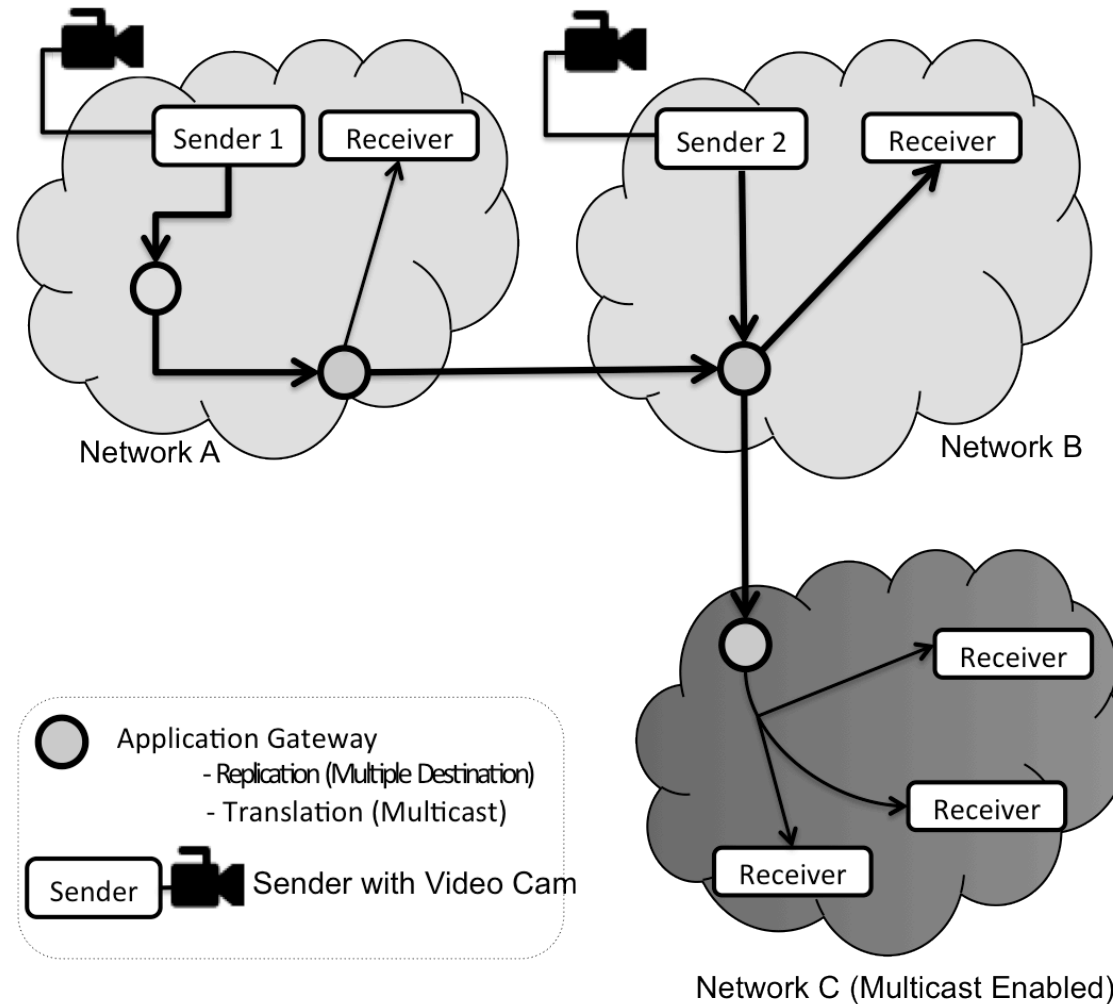
Gateway's Technical Motivation



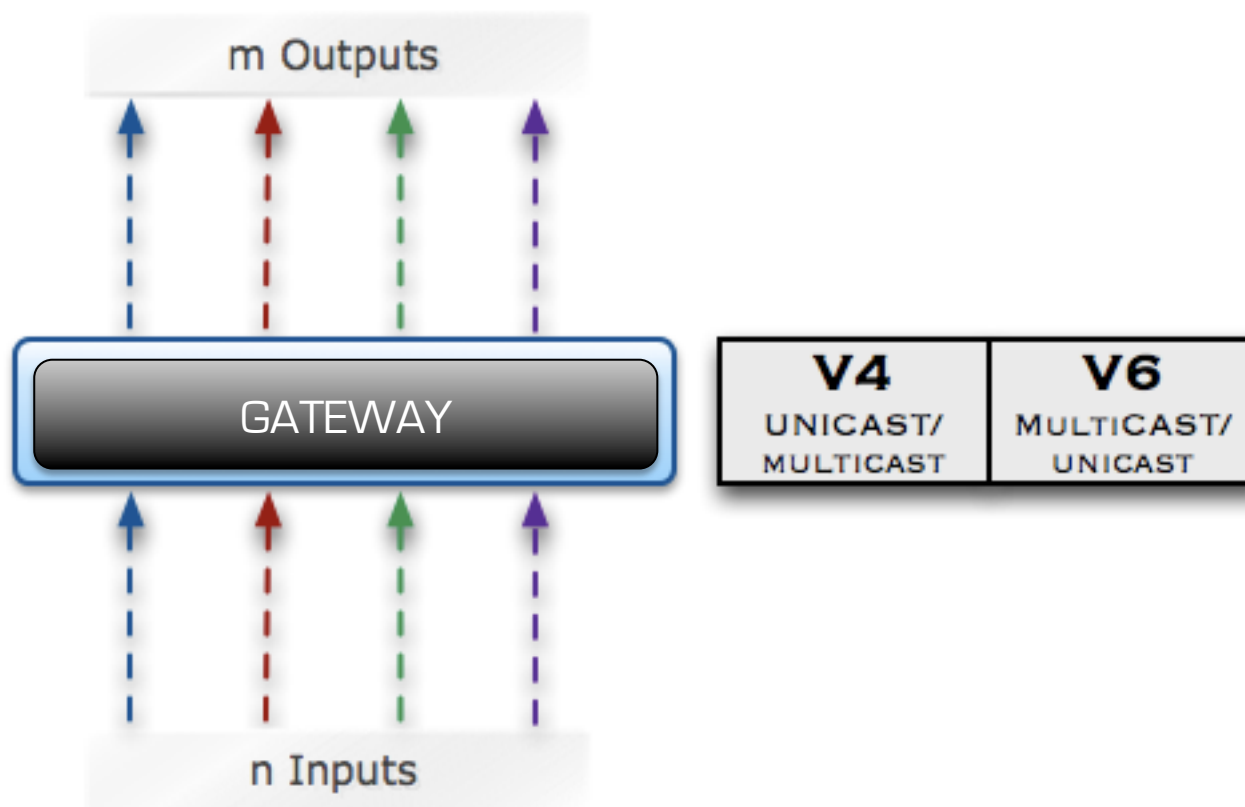
Features of Gateways



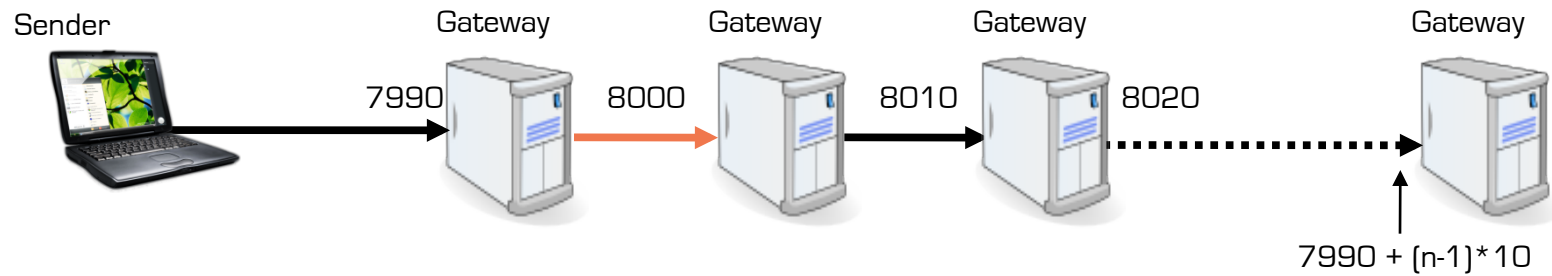
Application Gateways



Functions of App. Gateway

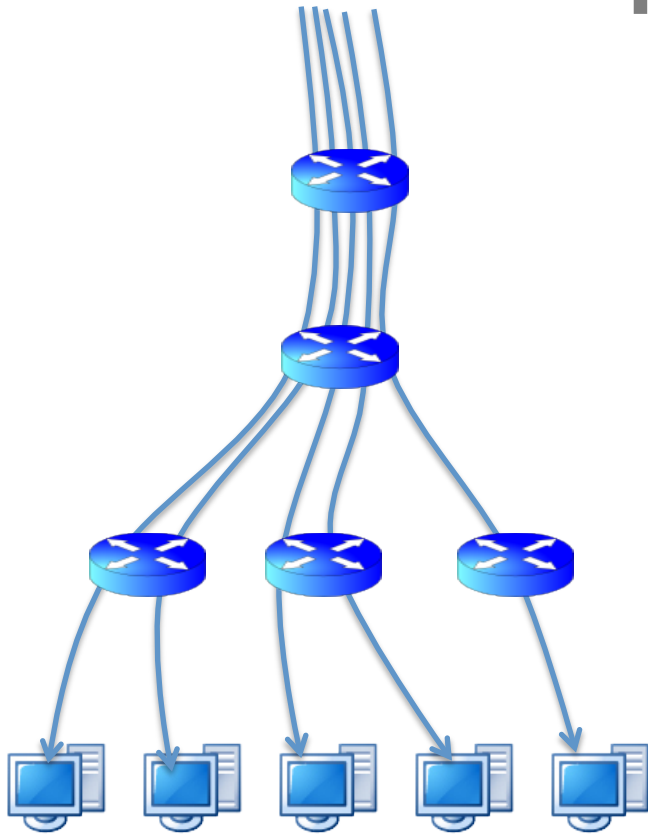


Cascaded Gateways

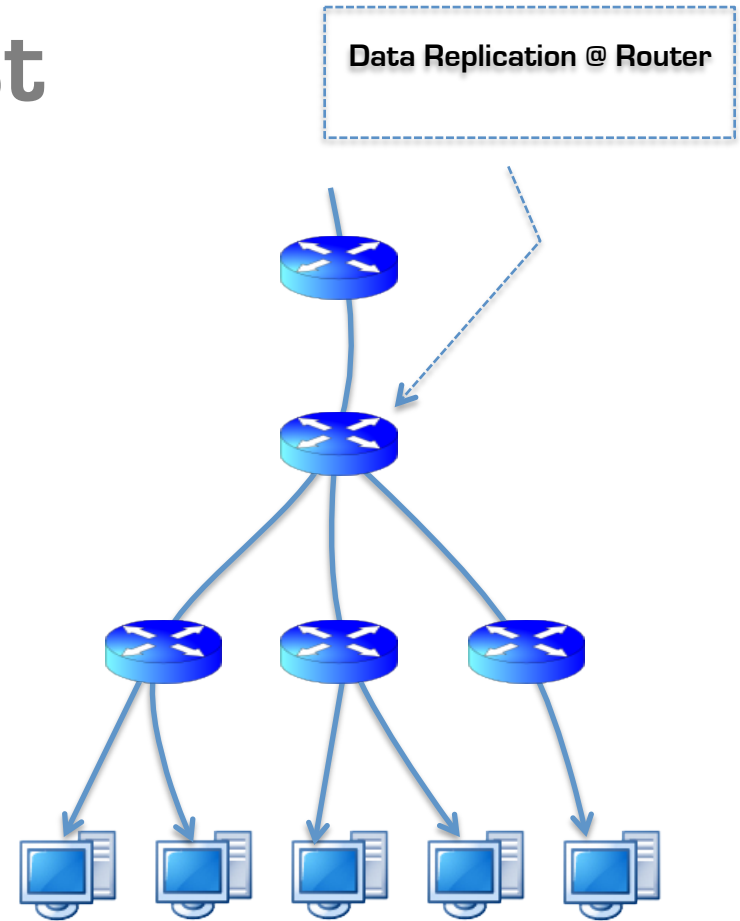


Multicast

Multicast



Unicast: Unicast routing is concerned with where the packet is going to.



Multicast: Multicast routing is concerned with where the packet will be coming from.

Multicast

Replication @ Router ≈ Responsibility of Router

1. Check whether your router supports Multicast (check IOS ver).
[Cisco Router Check Tool](https://tools.cisco.com/ITDIT/CFN/) [tools.cisco.com/ITDIT/CFN/]
2. Configuration at router is required.

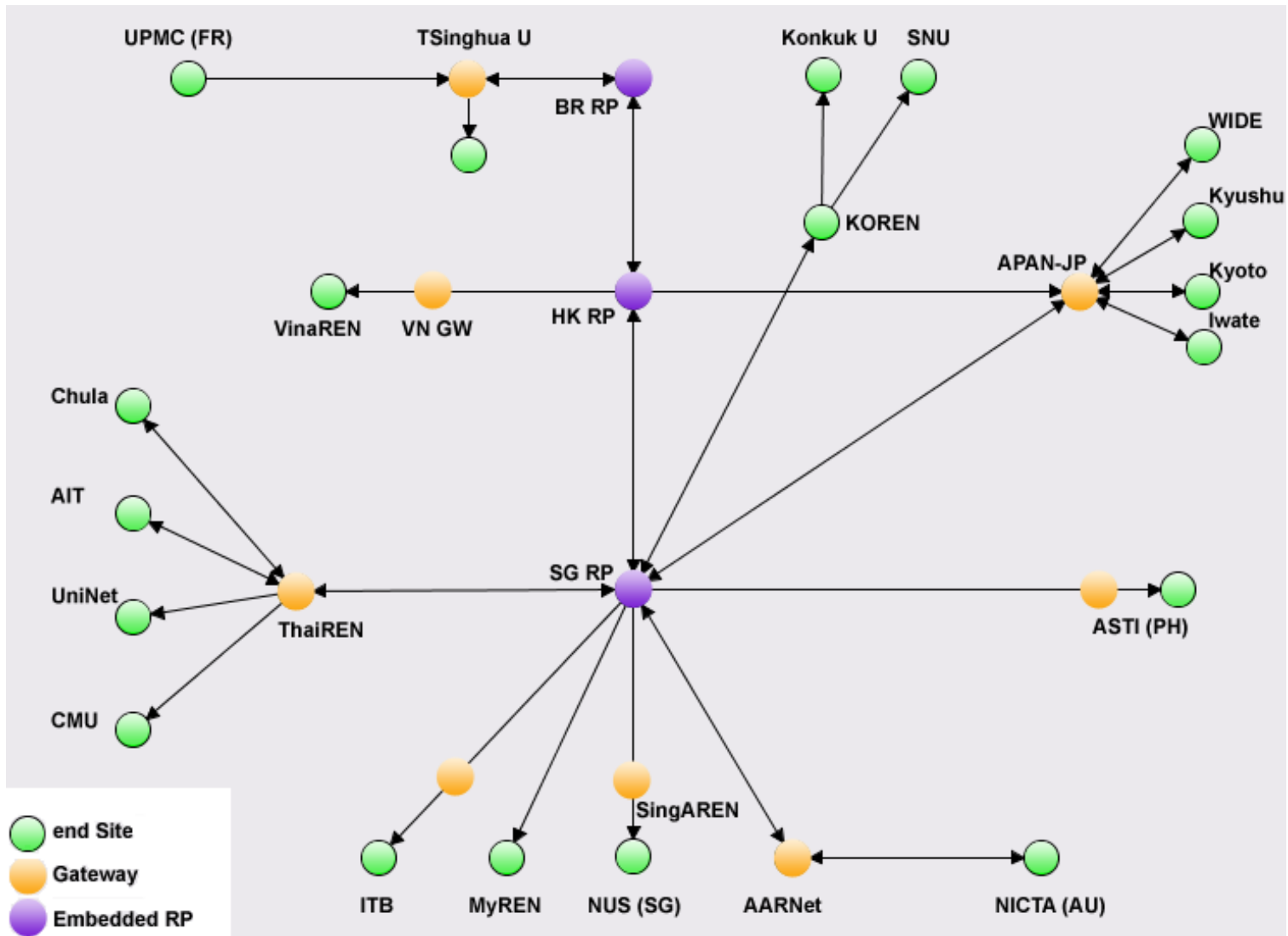
Receivers' capability of receiving Multicast Data!

Check if receiver supports IGMP/MLD

Multicast Notes

- ♪ Bandwidth saving over links.
- ♪ Enhanced efficiency, reduces server loads.
- ☀ Multicast is UDP!! (Do not expect reliable delivery, jitter may occur)

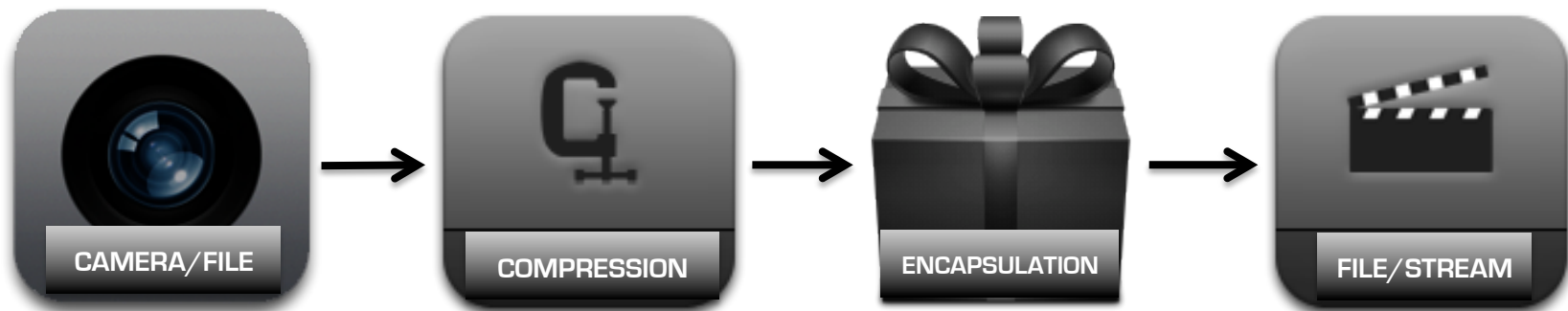
Multicast Tree in TEIN



Streaming Basics

Video

- Compression is central to the video production workflow.
- Video and audio streams are compressed, contained and then send to output as files or streams.
- Compression reduces the amount of data – Redundant information is removed to reduce the storage and bandwidth.
- Quality is reduced as the file gets more compressed.



Same Concept Applies to VLC

1. Get source
2. Compression
3. Encapsulation
4. Send Out

Real-time Transport Protocol (RTP)

- It defines packet format for delivering media over IP.
- RTP is used extensively in communication systems(Telephony, VoIP, Videoconference etc.).
- RTP Packets are transmitted using UDP-IP.
- RTP is usually implemented in the Application.
- RTP was originally designed for multicast, but also used for unicast.
- RTP is consist of RTP itself and RTCP.
- RTCP is used to monitor QoS and to convey info about the participants. (BW of RTCP is typically 5% of RTP)

Real-time Transport Protocol (RTP)

Services provided by RTP.

- Payload type identification
- Source Identification
- Time stamping
- Media Sequencing

RTP does not prevent out of order delivery

RTCP (RTP Control Protocol)

- ◆ It works with RTP.
- ◆ During RTP session, RTCP sends feedback on Quality back to sender.
- ◆ RTP and RTCP flows typically use separate UDP ports.
- ◆ Purpose: Reports the quality of the connection b/w sender & receivers

Types of RTCP Messages

1. Source Description – useful information about the source data (useful ASCII text strings [user & host name of participant, e-mail address, notes, ...])
2. Sender Report – report current time and amount of data sent so far” (cumulative frame & byte counts » wall clock/timestamp values)
3. Receiver Report – feedback to sender “what has been received so far” (frame loss/frame delivery rate)
4. Bye – source is disconnecting
5. Application - specific

RTSP (Real-Time Streaming Protocol)

RTSP is more of a framework than a Protocol.

RTSP uses delivery mechanism with the help of RTP.

RTSP allows the user to start, stop and replay content on demand.

RTSP acts as a “Remote Control”.

Example:

RTSP URL *rtsp://201.4.2.1/media/audiotrack*

rtsp:// is the identifier for TCP rtsp scheme (*rtspu://* for UDP)

media = name of the presentation

audiotrack = name of certain stream (this is optional)

Codec & Containers

Codec - Coder / Decoder

- reduces the data space (storage/network transfer).
- It uses specific compression algorithm.
- Example Codecs: MPEG-1, MPEG-2, MPEG-4, Vorbis, DivX, H264, H263, MPEG Audio, MP3, WAV, AAC, WMA

Container - is stream format or type.

- Example Container: AVI, Ogg, MOV, ASF, MP4, MP3, FLV
- Normally same as file format: file.avi.

Streaming Prerequisite

1. Output technology (HTTP, RTP, UDP in Unicast/Multicast)
2. Encapsulation (PS, TS, Raw, MPEG-1, MOV, WAV, FLV)
3. Video Formats (MPEG-1, MPEG-2, MPEG-4, DivX etc)
4. Audio Formats (MPEG, MP3, AAC, u-law, a-law etc)

Selection of Frame Rate

| Video | fps |
|-------|----------|
| NTSC | 29.97 |
| PAL | 25 |
| Film | 24 |
| HD | 24/30/60 |

Select lower frame rate if you need to reduce bandwidth and CPU processing.

Otherwise leave frame rate by-default!

For HD over network select 24

Selection of Video Data Rate

| Scenario | Frame Rate | Data Rate |
|---------------------|------------|------------|
| Mobile Content | 10-15 fps | 50-60 Kbps |
| Standard Definition | 24 fps | 1-2 Mbps |
| HD | 24 fps (p) | 4-6 Mbps |
| Full HD | 24 fps (p) | 6-8 Mbps |

Comparative Resolution

| Term | Resolution | Display Mode | Frame Rate | Example |
|---------|------------|--------------|------------|------------------------------|
| 720p60 | 1280x720 | progressive | 60 fps | HDTV, Blu-ray |
| 1080i60 | 1920x1080 | interlaced | 60 fps | HDTV, Blu-ray, HD camcorders |
| 1080p30 | 1920x1080 | progressive | 30 fps | HD camcorders |
| 1080p24 | 1920x1080 | progressive | 24 fps | Blu-ray, HD camcorders |
| 1080p60 | 1920x1080 | progressive | 60 fps | HD camcorders |

Streaming Via Gateway

HD - VideoLAN

1. Prepare
2. Download
3. Compile
4. Install

DV - DVRelay

1. Prepare
2. Download
3. Compile
4. Install

FFMPEG, FFPLAY

```
tein3@host:/$ sudo su
root@host:# wget tein3.canalavist.org/file.dv
root@host:# wget tein3.canalavist.org/pix.avi
root@host:# wget tein3.canalavist.org/hd.mov
```

```
root@host:# file -i pix.avi
root@host:# ffmpeg -i pix.avi
```

```
Duration: 00:04:36.72, start: 0.000000, bitrate: 1239 kb/s
  Stream #0.0: Video: msmpeg4, yuv420p, 640x352, 25 tbr, 25 tbn,
  25 tbc
  Stream #0.1: Audio: mp3, 48000 Hz, stereo, s16, 128 kb/s
```

```
root@host:# file file.dv
root@host:# ffplay file.dv
```

DV Tools & Applications

1. gcanbus
2. kino
3. Dvgrab
4. Testlibraw
5. DVTS
6. DVRelay

GScanbus

- Gscanbus is a little bus scanning, testing and topology visualizing tool for the Linux IEEE1394 cameras.
- A debugging tool for IEEE1394 development, but can also be used to simply check your IEEE1394 setup on Linux.
- <http://gscanbus.berlios.de/>

Connect Your DV Camera and Issue the following command

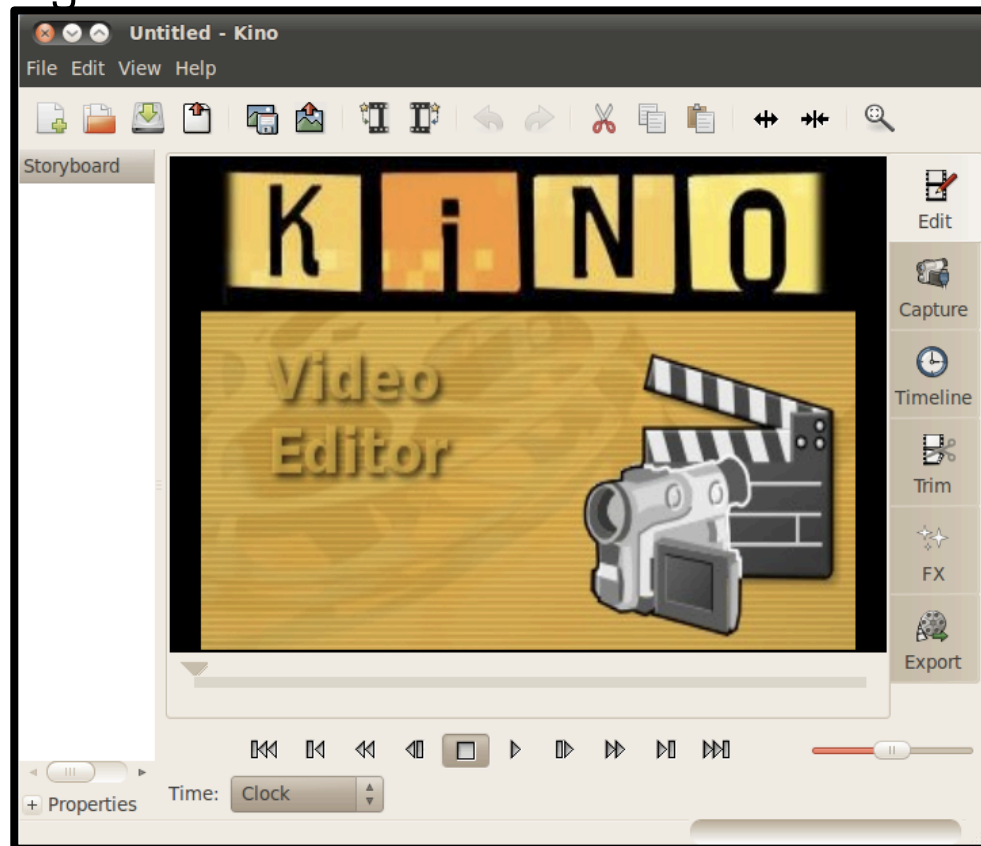
```
root@host:~# gscanbus
```

Kino

Kino is an DV editor

Kino can record/create/edit/play movies recorded with DVCams.

www.kinodv.org

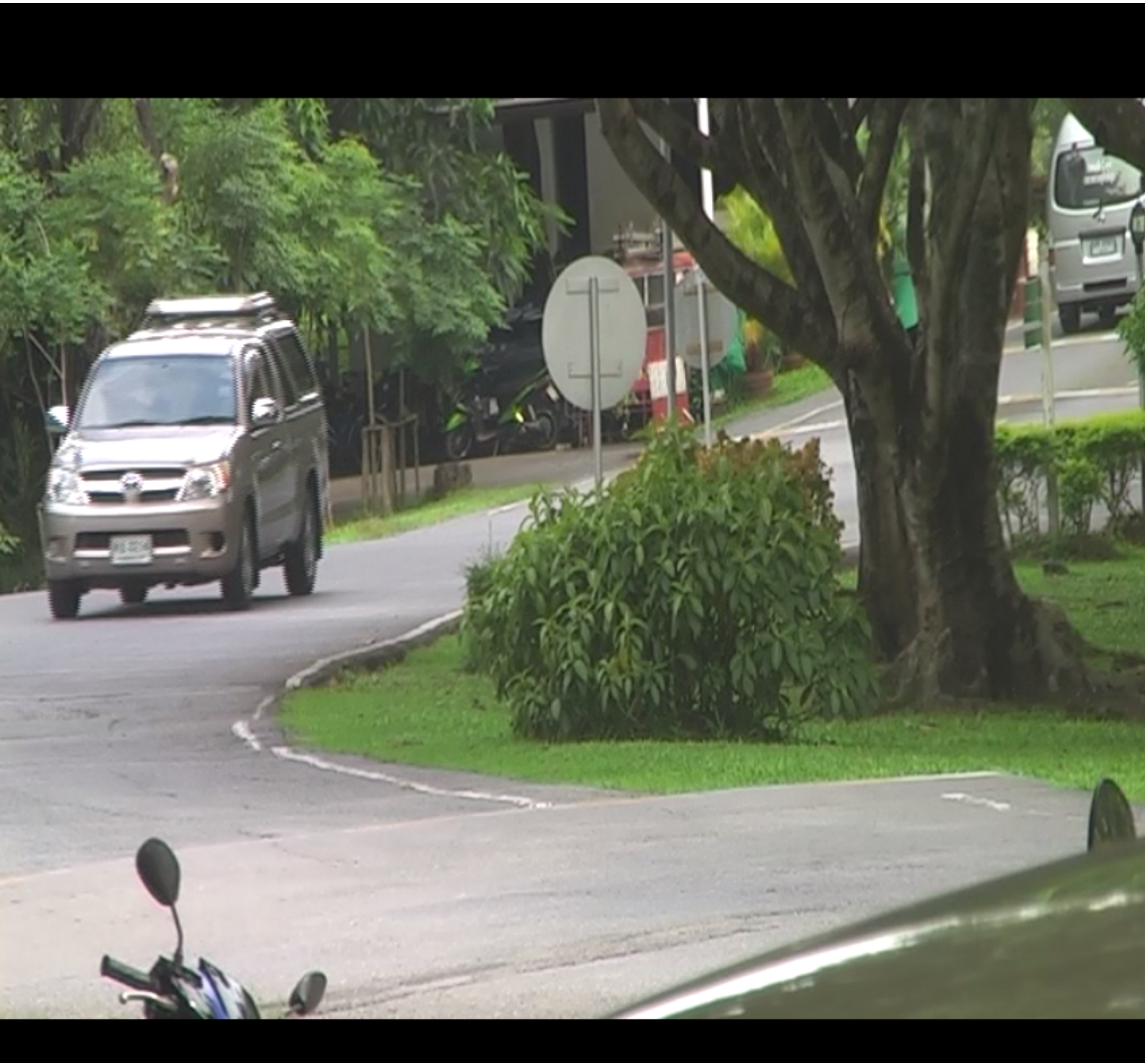


DVGrab

- Dvgrab is a program which saves the audio and video data from a DVCam into file. These AVI files can then be processed further by other video editing programs available for Windows or Linux.'
- It can also write raw DV to stdout (or pipe command). This is especially nice with recent versions of Xine and mplayer as they can display DV with deinterlacing and adjusting for the non-square pixel aspect ratio of DV.
- `$ dvgrab - | ffplay -`
- `$ dvgrab -format dv1 - | vlc -vv -`

DVGrab

```
mpplay version 0
built on Mar 22
configuration:
e-libschrödinger
-cpudetect --enab
WARNING: libra
libavutil con
bdirac --enable-
-enable-libvpx -
grab --enable-li
/usr/lib/i686/cm
libavcodec con
bdirac --enable-
-enable-libvpx -
grab --enable-li
/usr/lib/i686/cm
libavformat con
bdirac --enable-
-enable-libvpx -
grab --enable-li
/usr/lib/i686/cm
libavdevice con
bdirac --enable-
-enable-libvpx -
grab --enable-li
/usr/lib/i686/cm
libavfilter con
bgsm --enable-li
able-runtime-cpu
i686 --enable-sh
libswscale con
bgsm --enable-li
able-runtime-cpu
i686 --enable-sh
libpostproc con
bgsm --enable-li
able-runtime-cpu
i686 --enable-sh
libavutil
libavcodec
libavformat
libavdevice
libavfilter
libswscale
libpostproc
[dv @ 0x90d9080]
Input #0: dv, fr
```



```
able-bzlib --enable-libgsm --enabl
disable-stripping --enable-runtime
ed --disable-static

e-udpau --enable-bzlib --enable-li
--enable-threads --enable-zlib -
proc --enable-swscale --enable-x11
avid --enable-libdc1394 --shlibdir=

e-udpau --enable-bzlib --enable-li
--enable-threads --enable-zlib -
proc --enable-swscale --enable-x11
avid --enable-libdc1394 --shlibdir=

e-udpau --enable-bzlib --enable-li
--enable-threads --enable-zlib -
proc --enable-swscale --enable-x11
avid --enable-libdc1394 --shlibdir=

e-udpau --enable-bzlib --enable-li
--enable-threads --enable-zlib -
proc --enable-swscale --enable-x11
avid --enable-libdc1394 --shlibdir=

e-udpau --enable-bzlib --enable-li
ble-libvpx --disable-stripping --en
shlibdir=/usr/lib/i686/cmov --cpu=

e-udpau --enable-bzlib --enable-li
ble-libvpx --disable-stripping --en
shlibdir=/usr/lib/i686/cmov --cpu=

e-udpau --enable-bzlib --enable-li
ble-libvpx --disable-stripping --en
shlibdir=/usr/lib/i686/cmov --cpu=
```

Testlibraw

Tests basic functionality on libraw1394

```
root@localhost:/ # testlibraw
```

```
successfully got handle
```

```
current generation number: 1
```

```
1 card found
```

```
card 0, name: ohci1394
```

```
2 nodes on bus, local ID is 1, IRM is 1
```

- doing transactions with custom tag handler

 - read from node 0... completed with value 0x08ce0404

 - read from node 1... completed with value 0x51f80404

Coriander

Coriander is the Linux GUI for controlling a Digital Camera through the IEEE1394 bus (aka FireWire, or iLink).

Coriander will also let you

- record video,
- send images to an FTP site,
- convert the video to a V4L stream,
- live display is provided.

Coriander will work with any camera that is compatible with the IIDC specifications (also known as DCAM specs). This includes most 1394 cams and a majority of industrial or scientific cameras too.

But any cameras in which you can insert a video tape (camcorders) will NOT work with coriander.

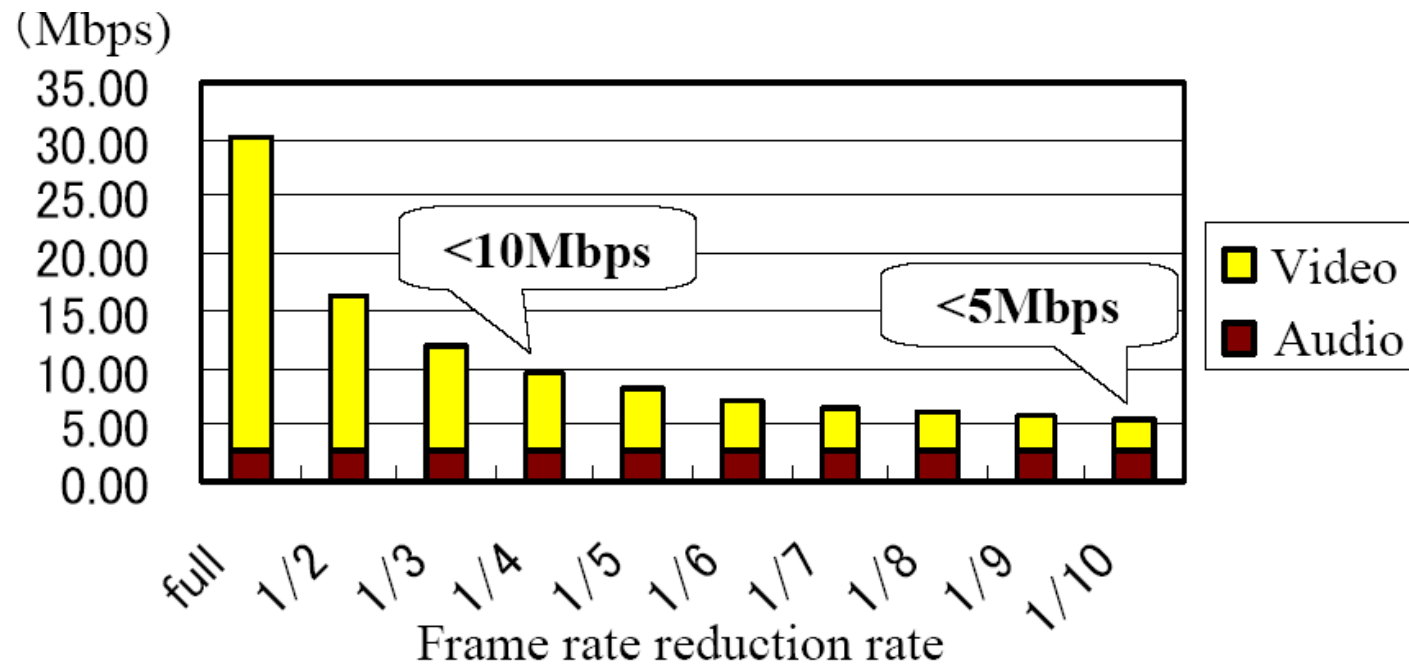
Firecontrol

Simple console oriented tool based on libraw1394 to issue asynchronous read, write and lock commands to some nodes on the bus as well as to force bus resets.

The main advantage of this tool over other available tools (like gscanbus) is (besides not requiring a graphical interface), that it reports the exact acknowledge and response codes and therefore is very useful for debugging purposes.

<http://firecontrol.sourceforge.net>

DV Bandwidth



Video discard (audio never)

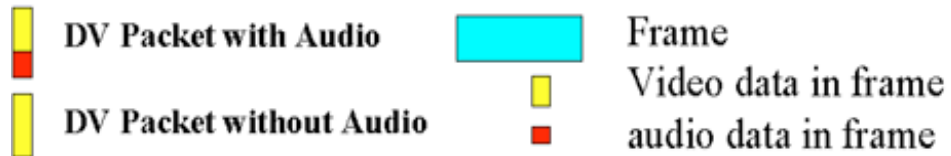
Full rate digital video stream



Half rate digital video stream



1/3 rate digital video stream



VideoLAN

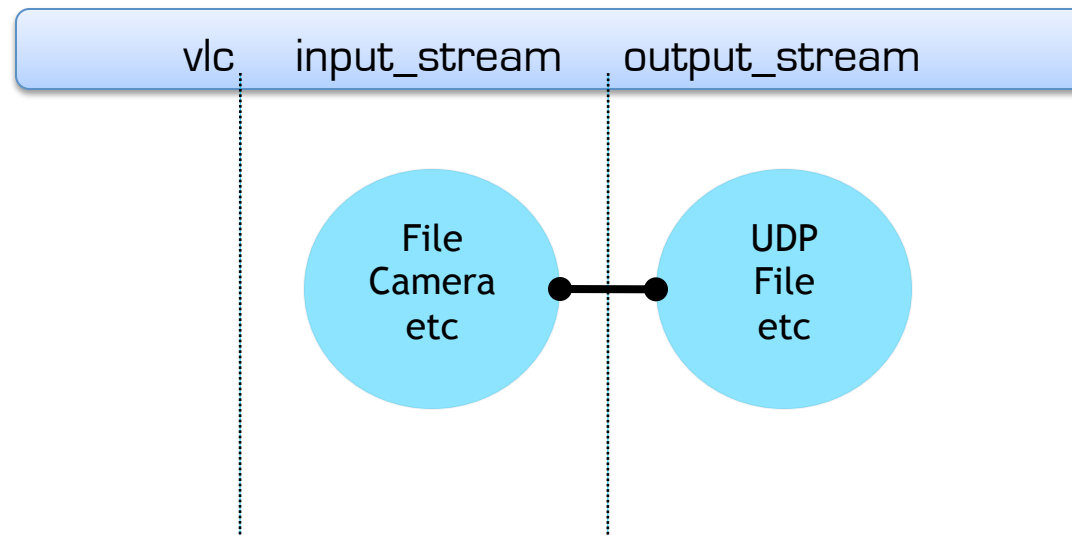
VideoLAN is a project that develops software for playing video and other media formats across a local area network (LAN). It originally developed two programs for media streaming, VideoLAN Client (VLC) and VideoLAN Server (VLS), but most of the features of VLS have been incorporated into VLC, with the result renamed **VLC media player**. [www.videolan.org]

The project began as a student endeavor at École Centrale Paris (France), but after releasing the software under the free software/open source GNU General Public License.

VLC's Binaries are available for: Windows / Mac OSX / Linux

But in order to support DV Format, VLC need to compile from source code on Linux.

Streaming from VLC



Important Parameters

| | |
|----------|--|
| | |
| vb | video bitrate |
| scale | allows to resize the size of the video. |
| fps | frame per second: fps=30 |
| height | height of the transcoded video, in pixels: height: 720 |
| width | width of the transcoded video in pixels: width=1280 |
| mux | encapsulation method of resulting stream after mux{ts/ps/asf} |
| dst | destination: dst=display, dst=transcode{}, dst=rtp{}, dst=std{},dst=ip |
| ttl | set ttl values: -ttl 16 |
| loop | set looping in case if you are sending file: -loop |
| channels | mono mode (= 1) or stereo (= 2) |
| ab | audio bitrates |
| vcodec | video codec |
| acodec | audio codec |

Example (HD Streaming 3Mbps)

```
#!/usr/bin/env bash
```

```
#
```

```
#
```

```
VCODEC="div2"
```

```
VB="3096"
```

```
ACODEC="mp3"
```

```
AB="96"
```

```
DST="203.159.13.2:1234"
```

```
#
```

```
#
```

```
dvgrab -format dv1 -| vlc -vv - :demux=rawdv -l dummy -sout "#transcode  
{vcodec=$VCODEC,vb=$VB,width=1280,height=720,scale=0.5,acodec=  
$ACODEC,ab=$AB};duplicate{dst=std{access=udp,mux=ts,dst=$DST}}"
```



Output & Encapsulation

For transmission of stream, Encapsulation= TS (MPEG-TS)

Output Technology = UDP, RTP, HTTP

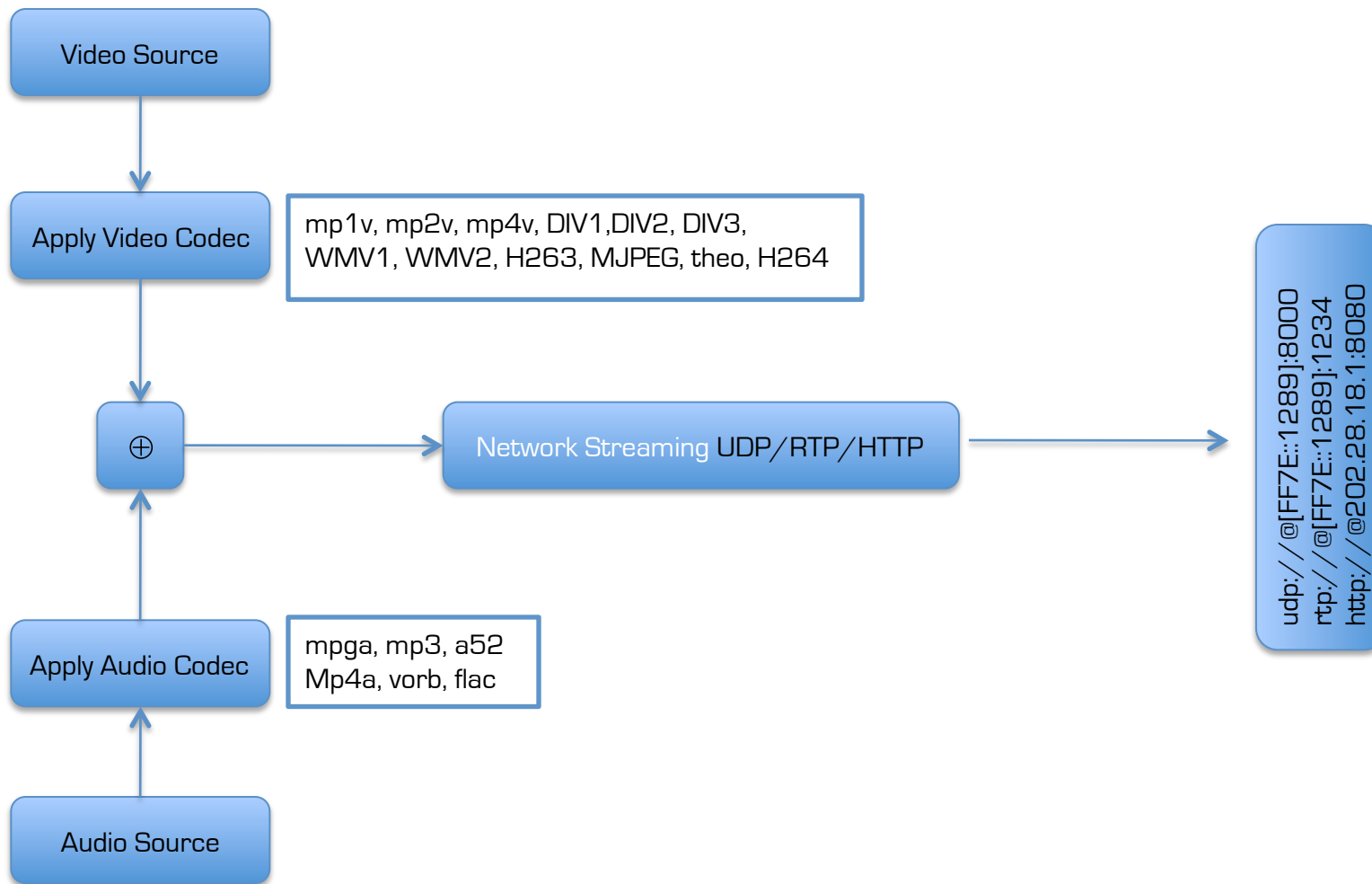
Codec = Video: mp4v, div2/3, wmv1/2

Audio: aac, mp3, mp4a

Bit-rate = Audio Bit-rate = 128 Kbps (good enough for mp3)

Video Bit-rate = ? (I will tell you later)

How to Stream



Skipped Error

If HD Resolution 1280x720 and transmission videorate=2M (for ex) then many frames will be skipped at sender's PC.

[0xa0af74c] stream_out_transcode stream out debug: late picture skipped (sequence no)

Video file's videobitrate > vb (streaming video bitrate)



ffmpeg -i file.avi

vlc's vb

Input Streams

Demonstrate 3 types of input streams:

1. File
2. Web camera
3. DV Camera

Stream from File

```
$ vlc -vw file:///home/user/pix.avi
```



Stream from webcam

1. Check which video devices are present in the kernel

```
root@wunca:/ # ls -la /dev/video*  
crw-rw—+ 1 root video 81, 0 2011-07-22 15:13 /dev/video0  
crw-rw—+ 1 root video 81, 1 2011-07-22 15:13 /dev/video1
```

2. Run

```
root@wunca:/ # vlc -vv v4l2:///dev/video0
```


Stream from DVCam

1. Check to see if your Firewire devices are available to your computer

```
root@wunca:/# ls -la /dev/raw1394 or /dev/fw*  
crw-rw—+ 1 root video 251, 0 2011-07-22 15:13 /dev/fw0  
crw-rw—+ 1 root video 251, 0 2011-07-22 15:13 /dev/fw1
```

2. Check if display is working

```
root@wunca:/# dvgrab -f dv1 -|vlc -
```

```
root@wunca:/# dvgrab -f dv2 -|vlc -
```

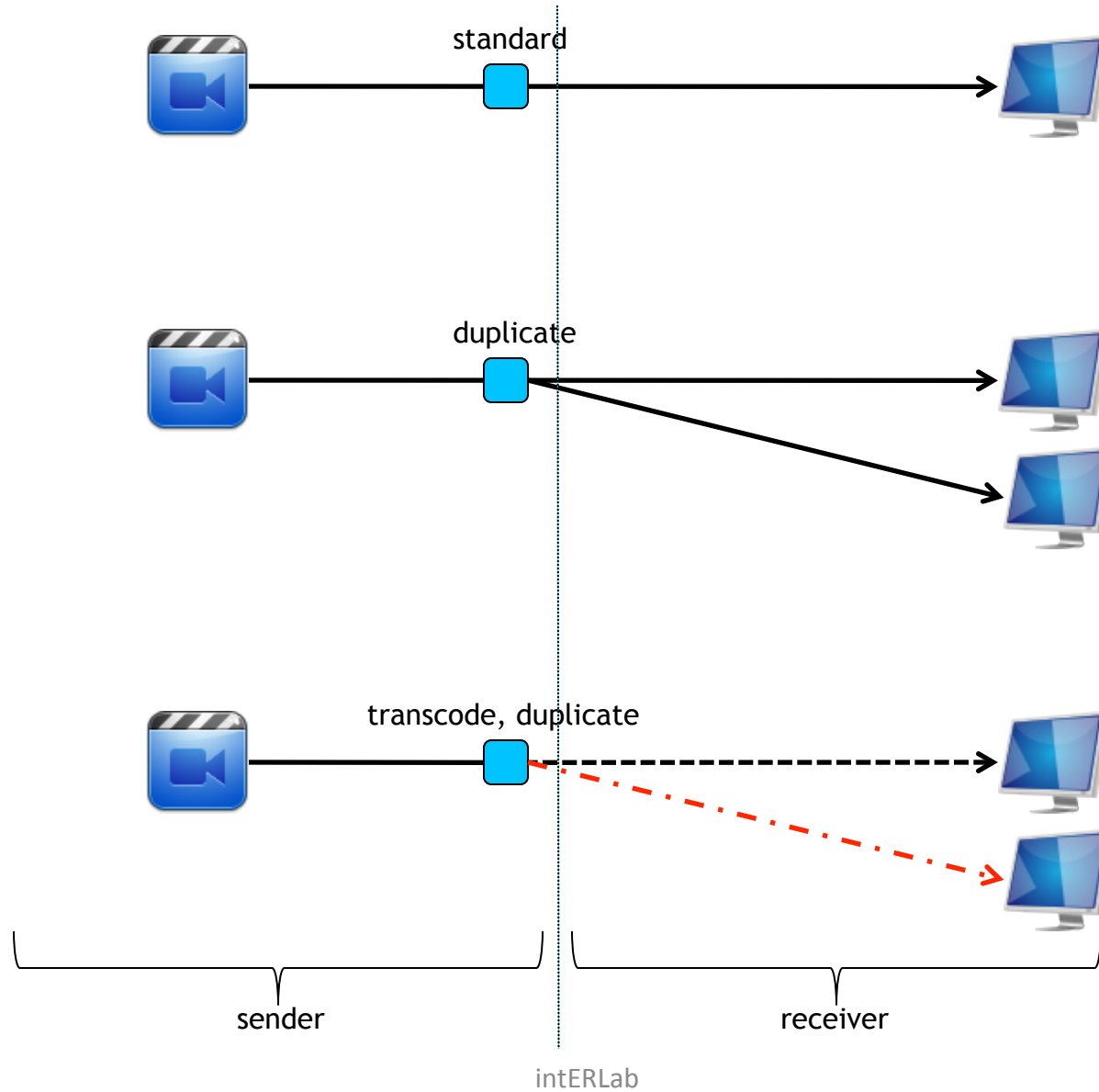
Output Stream

1. UDP
2. File
3. RTP
4. HTTP

Output Stream

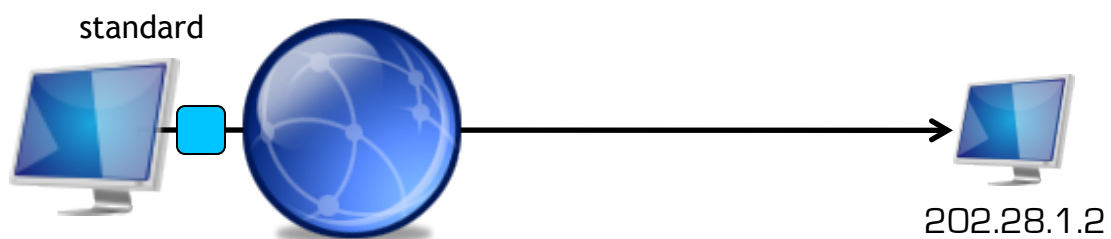
1. Use “**standard**” or “**std**” keyword when no transcoded stream out.
2. Use “**duplicate**” keyword when more than one streams are going out.
3. Use “**transcode**” keyword if you want to apply codec to stream.

Virtual Illustrations



Standard: UDP Unicast

```
vlc -vv pix.avi -sout "#standard{dst=202.28.1.2:1234,mux=ts,access=udp}"
```

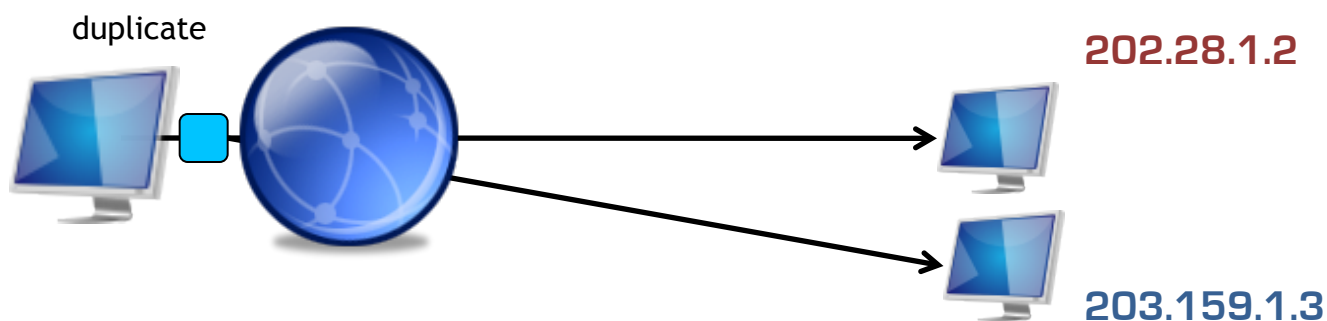


```
vlc udp://@202.28.1.2:1234
```

```
vlc udp://@[2001:.....:11]:1234
```

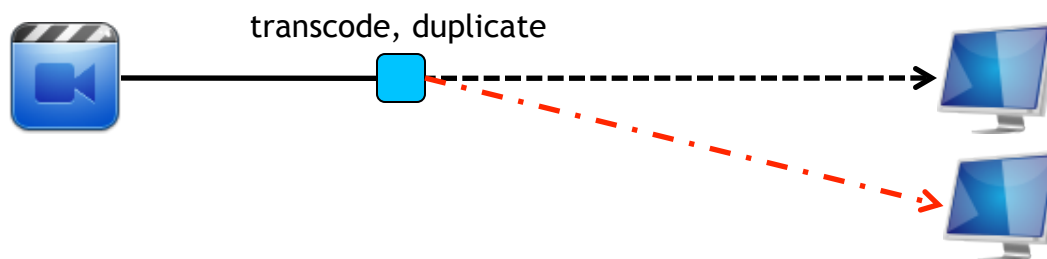
Duplicate: UDP Unicast 2 Receivers

```
vlc -vw pix.avi -sout "#duplicate{dst=std  
{access=udp,mux=ts,dst=202.28.1.2:1234},dst=std  
{access=udp,mux=ts,dst=203.159.1.3:1234}}"
```



DVCam: UDP Unicast 2 Receivers

```
dvgrab -format dv1 - | vlc -vv - :demux=rawdv -l dummy -sout '#transcode  
{vcodec=DIV2,vb=8000,scale=1,acodec=mp3,ab=128}:duplicate{dst=std  
{access=udp,mux=ts,dst=202.28.1.2:1234}, dst=std  
{access=udp,mux=ts,dst=203.159.1.3:1234}}'"
```



Optional

```
"#duplicate{dst=udp{dst=235.1.1.1:1234},dst=display}"
```

display on sender too.

List of all codecs available: vlc -list

Multicast

Server Side

1. IPv4 Multicast Sending

```
$ vlc -vvv pix.avi -sout "#standard{dst=235.1.1.1:1234,mux=ts,access=udp}"
```

2. IPv6 ASM Multicast Sending

```
$ vlc pix.avi -sout "#standard{dst=[FF7E:340:2001::1234]:1234, mux=ts,access=udp}"
```

Client Side

1. IPv4 Multicast Receiving

```
vlc udp://@235.1.1.1:1234
```

2. IPv6 Multicast Receiving

```
vlc udp://@[FF7E:340:2001::1234]:1234
```

```
vlc udp://[2001:3C8:E100:1019::48]@[FF3E:340:2001::1234]:1234
```

ASM

SSM

RTP Streaming

Server Side

```
dvgrab -f dv1 - | vlc -vv - :demux=rawdv -l dummy -sout '#transcode  
{vcodec=mp4v,vb=4096,height=720,width=1280,scale=0.5,acodec=mp3,ab=128  
}:rtp{dst=203.159.31.22,port=1234,sdp=rtsp://202.28.1.2/video.sdp}'
```

Client Side:

```
vlc rtsp://202.28.1.2/video.sdp
```

Source = Camera
Video = 4Mbps
Audio = 128 Kbps
Resolution = 1280 X 720
Video Codec = MP4
Audio Codec = MP3

VLC Compilation

By default DV is not enabled in VLC. You need to compile VLC from source file. DVGrab requires ROOT

Step (1) Install dependencies

```
$ apt-get install libhal-dev libdbus-qt-1-dev libavc1394-0 libavc1394-dev libdvdread-dev libmad0-dev libavcodec-dev libavformat-dev libavutil-dev libswscale-dev libpostproc-dev libfaad-dev libtwolame-dev liba52-dev libdca-dev libfribidi-dev libqt4-dev libraw1394-dev libdvdread-dev libavcodec-extra-52 libmpeg2-4-dev libdvbpsi5-dev libv4l-dev libtag1-dev liblua5.1-0-dev libshout3-dev libvcdinfo-dev libcddb2-dev libpulse-dev libflac-dev libfaac-dev libgnomevfs2-dev libxcb-shm0-dev libxcb-xv0-dev libosso-dev libportaudio-dev libmtp-dev libalsaplayer-dev libjack-dev libupnp-dev libprojectm-dev libass-dev libsqlite0-dev libsdl1.2-dev libschrödinger-dev libfluidsynth-dev librsvg2-dev libzvbi-dev libdirac-dev libmodplug-dev libnotify-dev libudev-dev lib64asound2-dev libasound2-dev libportaudio2 libcaca-dev libx11-xcb-dev libsqlite0-dev libsdl-image1.2-dev lua5.1 liblivemedia-dev livemedia-utils
```

Step (2) Download VLC Source from www.videolan.org, extract and run the following command:

```
$ ./configure --prefix=/usr/local --enable-switcher --enable-shout --disable-dc1394 --enable-dv --enable-dvdread --enable-v4l --enable-pvr --enable-gnomevfs --enable-vcdx --enable-faad --enable-twolame --enable-real --enable-realrtsp --enable-flac --enable-tremor --enable-theora --enable-ogg --enable-vorbis --enable-a52 --enable-gnomevfs --enable-dca --enable-run-as-root --enable-v4l2 --enable-live555 --disable-glx
```

Step (3)

```
$ make && make install
```

DVGrab & FFMPEG

```
$ apt-get install dvgrab
```

```
$ apt-get install ffmpeg
```

Conclusion

- Inexpensive computer with camera.
- Linux based operating system for high quality or DV streaming.
- Exploiting the use of the dvgrab | ffmpeg | vlc
- Utilize the multicast.
- Utilize the application gateway.

Thank You

Contact Us

Internet Education and Research Laboratory (intERLab)
Asian Institute of Technology
P.O.Box 4, Klong Luang
Pathumthani 12120,
Thailand

Tel:+66-2524-6611/6613 Fax:+66-2524-5375/6618

Queries/Info: info @ canalavist.org

Technical: tech @ canalavist.org

<http://www.facebook.com/ipv6mcast>

<http://stream.canalavist.org/>

<http://www.interlab.ait.asia/tein3/ipv6mcast/>