

Streaming networks with VLC

By Jean-Paul Saman, <jean-paul.saman@m2x.nl>

Introduction

The VideoLAN project started at L'Ecole Central des Paris in 1996. Its goal was to develop high quality streaming for the Campus network. In 2001 the project went Open Source providing a complete High Quality Streaming solution available under the GPL.

Today the VideoLAN project is known for its adherence to international streaming standards. The multimedia client and server known as VLC is used as test tool, by Universities, R&D departments, Mobile-, Cable modem-, Settopbox- and Streaming Server manufacturers. VLC is also used in commercial products (Freebox, Di.com).

A common misconception is that "VLC media player" is only a client, but it is also a multimedia streamer. Originally it was only a client, but since it gained multimedia streaming capabilities the difference between client and server functionality has vanished and the name was changed from VideoLAN Client to VLC media player. The naming contributes to the confusion that some new users experience.

The VideoLAN project provides a complete streaming solution that is ready to be deployed in an enterprise or home streaming system. It includes a streaming server, client and mini-SAP server for multiple platforms.

Streaming networks with VLC

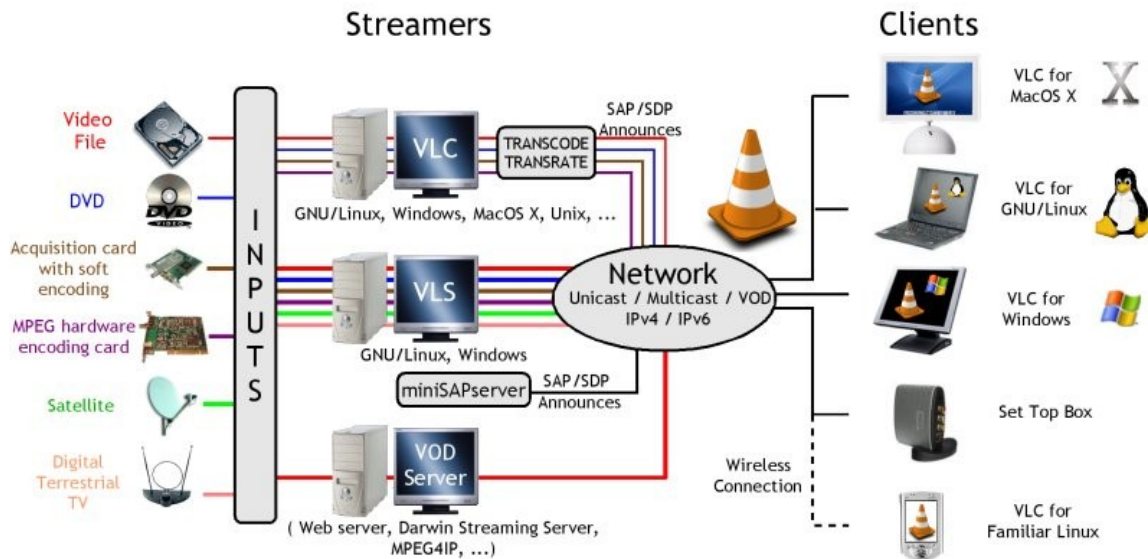


Figure 1: VideoLAN network architecture

Figure 1 shows that VLC can use different types of hardware as input. To name a few: DVD-, VCD-, SVCD drives, Acquisition-, Encoding cards (PVR 250/350), Satellite dish (DVB-S/C) and Terrestrial TV (DVB-T). The figure is not complete it omits: Webcams, DV camcorders, network cameras (AXIS), DirectShow/DirectX devices and more.

All those devices and inputs require a broad support of codecs: QuickTime, WMV 1/2, MPEG-1/2/3/4, Ogg, Vorbis, Theora and 3GPP. VLC media player can play almost all multimedia formats available today (as long as an Open Source codec implementation is available). A full list of features can be found online at <http://www.videolan.org/vlc/features.html>.

VLC's streaming capabilities have surpassed that of the VideoLAN server (which was the first approach to a streaming server). The VideoLAN server is no longer developed and considered deprecated, but the source code is still available online.

The VideoLAN solution has been ported to many operating systems, amongst which you will find: MacOS X, Win32/95/98/XP/2000/2003, Linux (RH, SuSE, Mandrake, Gentoo, Debian, Familiar) BeOS, QNX, PocketPC, *BSD (Free-, Net-, OpenBSD). It also runs on a wide range of devices: PC's, MacOS X, laptops, settopboxes, PDA and mobile phones.

Streaming networks with VLC

The challenge of the VideoLAN team is to support all those environments and operating systems as good as possible. It is difficult to find enough package maintainers and developers that assist in bug-fixing, in implementing new features and in porting VLC to other platforms or devices.

Streaming networks

A streaming network can be organized in different ways. All have different properties, advantages and disadvantages. In this chapter 4 different forms will be discussed: broadcast, unicast, multicast and video on demand (VOD).

Broadcast

In a broadcast streaming network the streaming server sends the multimedia stream to the networks broadcast address. Clients like video players or settopboxes listen to the broadcast address or to a specific port. Each multimedia stream is sent on a different port number.

Definition:

Continuous streaming of multimedia content to a complete network segment (subnet, netmask eg: 172.16.10.255).

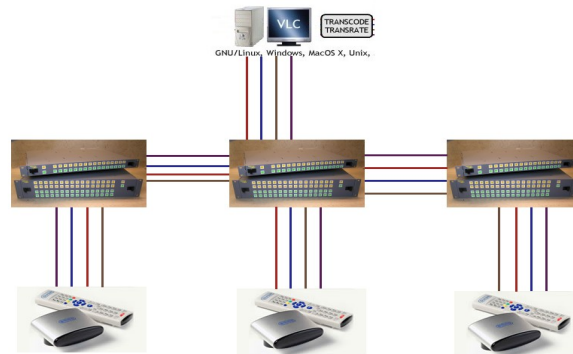


Figure 2: Broadcast network

Disadvantage:

- *All streams are sent to a complete subnet continuously.*

Streaming networks with VLC

- *Bandwidth of network is inefficiently used.*
- *All clients share the same stream.*
- *A client cannot control the stream with pause, play, stop, fastforward or rewind commands..*

Advantage:

- *All streams are available everywhere on the network.*

Here is a commandline example with VLC as broadcast streaming server and as client.

VLC as streaming server:

```
$> vlc terminator_240x192\@750.mpg --sout '#std{access=udp,mux=ts,dst=192.168.0.255}'
```

VLC as client:

```
$> vlc udp://
```

The server in the above example streams the multimedia content to port 1234. This is VLC default port. If a settopbox is used as client then configure it to listen on port 1234.

Unicast

In a unicast streaming network the streaming server sends the multimedia stream to a defined set of ip-addresses. Clients like video players or settopboxes listen each to their own address or to a specific port. Each multimedia stream is sent on a different port number and ip-address.

Definition:

Continuous streaming of multimedia content to a number of known ip-addresses. Every viewer has its own private stream.

Streaming networks with VLC

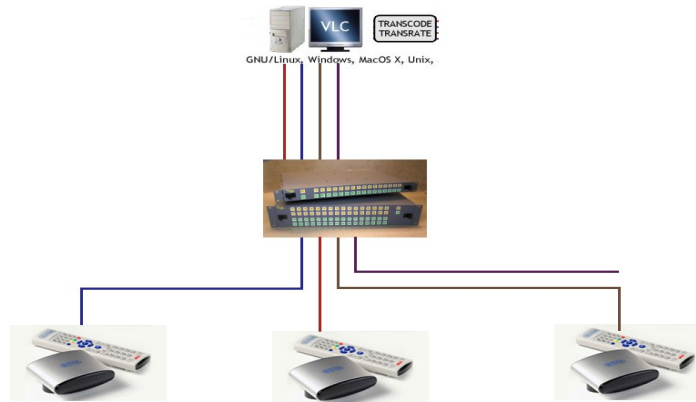


Figure 3: Unicast network

Disadvantage:

- Streams are sent to only a number of known clients (ip-addresses).
- Inflexible setup; adding new clients means starting a new stream.
- Stream is sent even when the client is not listening.
- A client cannot control the stream by using pause, play, stop, fastforward or rewind commands.

Advantage:

- Better usage of network bandwidth.
- Each client has a private stream.

Here is a commandline example with VLC as unicast streaming server and as client.

VLC as streaming server streaming using the default port 1234:

```
$> vlc terminator_240x192\@750.mpg -sout '#std{access=udp,mux=ts,dst=192.168.0.184}'  
$> vlc dvdsimple:///dev/dvd -sout '#std{access=udp,mux=ts,dst=192.168.0.20}'
```

VLC as client listening on the default port 1234:

```
$> vlc udp://
```

Multicast

In a multicast streaming network the streaming server sends the multimedia stream to a multicast group (in the 224.0.0.0 to 239.255.255.255 range). Clients like video players or settopboxes send a multicast JOIN message to the nearest router/switch. The router/switch then forwards the requested multicast channel towards the listening multimedia client. Each multimedia streams is sent on a different multicast ip-address.

Definition:

Continuous streaming of multimedia content to multiple viewers without unnecessary duplication of the source stream. Every viewer shares the same multimedia content stream.

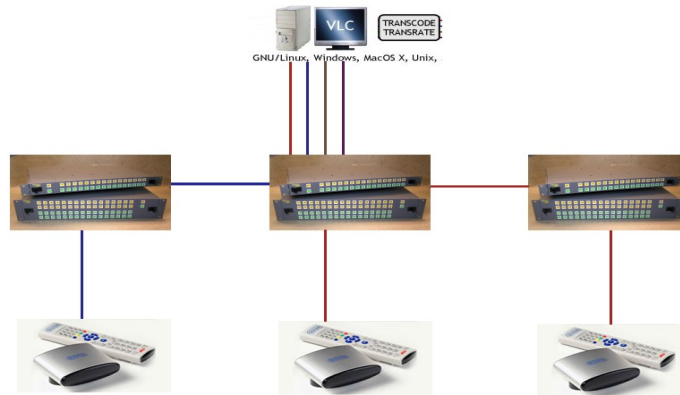


Figure 4: Multicast network

Disadvantage:

- Multiple clients share the same stream and cannot control it with pause, play, stop, fastforward or rewind commands.
- Server streams all streams to its nearest router/switch even when no client is watching.

Advantage:

- Best use of network, streams are duplicated at the network level closest to the client. All clients, servers, switches and routers must support multicast.

A multicast network is harder to setup than a broadcast or unicast network. All clients, servers, routers and switches must support it. Make sure your operating system (of client and server) has multicast enabled and a multicast route exist in its routing table. The following example adds a multicast route on a unix-

Streaming networks with VLC

based system:

```
$> /sbin/route add -net 224.0.0.0 netmask 240.0.0.0 dev eth0

$> /sbin/route

Kernel IP routing table
Destination      Gateway          Genmask         Flags Metric Ref    Use Iface
192.168.0.0      *                255.255.255.0   U        0      0      0 eth0
169.254.0.0      *                255.255.0.0     U        0      0      0 eth0
224.0.0.0        *                240.0.0.0       U        0      0      0 eth0
default          192.168.0.250   0.0.0.0         UG       0      0      0 eth0
```

Deploying multicast streaming for a local area network requires mapping of multimedia streams to ip-addresses. Which address of the multicast group, defined from 224.0.0.0 up till 239.255.255.255, can be used? Only a small portion of the entire range is available for use by multicast applications. The Internet Engineering Task Force (IETF) has an RFC, number 3171, that defines the address ranges that are available for multicast traffic (<http://www.ietf.org/rfc/rfc3171.txt?number=3171>). In the following table an overview of that definition is given:

<i>Address range</i>	<i>Netmask</i>	<i>Description</i>
224.0.0.0 – 224.0.0.255	224.0.0/24	LAN protocol traffic between routers/switches.
224.0.1.0 – 224.0.1.255	224.0.1/24	Internet protocol traffic between routers/switches.

Streaming networks with VLC

<i>Address range</i>	<i>Netmask</i>	<i>Description</i>
224.0.2.0 – 224.0.255.0		AD-HOC Block (globally routed addresses).
224.1.0.0 – 224.1.255.255	224.1/16	ST Multicast Groups
224.2.0.0 – 224.2.255.255	224.2/16	SDP/SAP Block is for applications that receive addresses through the SDP (RFC 2974) protocol.
224.252.0.0 – 224.255.255.255		DIS Transient Block
225.0.0.0 – 231.255.255.255		RESERVED
232.0.0.0 – 232.255.255.255.255	232/8	Source Specific Multicast Block (local to the host)
233.0.0.0 – 233.255.255.255	233/8	GLOP Block (globally scoped statically assigned group).
234.0.0.0 – 238.255.255.255		RESERVED
239.0.0.0 – 239.255.255.255	239/8	Administratively Scoped Block. Addresses for local within a domain.

Addresses in the range 239.0.0.0 – 239.255.255.255 can be safely used within a domain for multicast application traffic. Routers and switches use address from 224.0.0.0 to 224.0.0.255 and from 224.0.1.0 to 224.0.1.255 for communication. Using a route from these ranges is ill-advised.

Here is a commandline example with VLC as multicast streaming server and as client.

VLC as streaming server:

```
$> vlc terminator_240x192\@750.mpg -sout '#std{access=udp,mux=ts,dst=239.255.255.184}' -
ttl=3
$> vlc dvdsimple:///dev/dvd -sout '#std{access=udp,mux=ts,dst=239.255.255.22}' -ttl=3
```

VLC as client:

```
$> vlc udp://@239.255.255.184
```

In the above example -ttl=3 is added to the server commandline. A TTL is the “time to live” value that is in every IP packet. It tells how many routers/switches a IP-packet may cross before it is to be discarded. VLC has as default a ttl of 0, so all traffic stays local and doesn’t travel out of the local network. If there are multiple switches/routers in your network, then you must enlarge the number. It should be at least set to the number of switches/routers between server and client plus one.

Video On Demand

In a Video On Demand (VOD) streaming network the streaming server doesn't send multimedia content by default. Instead it waits for a request.

Definition:

Multimedia content is only streamed to viewers that request it explicitly ("on demand"). Each viewer has its own private stream.

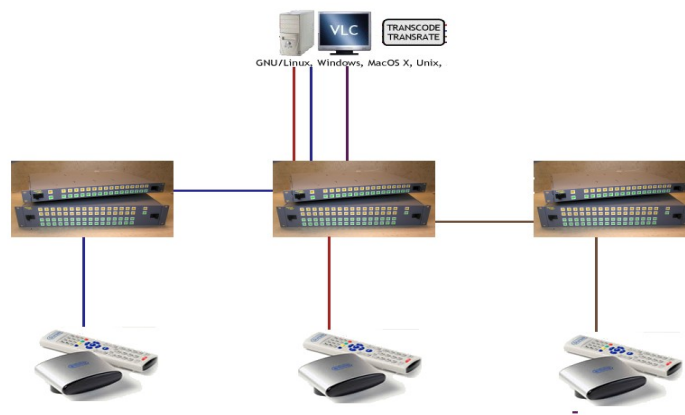


Figure 5: Video On Demand (VOD) network

Disadvantage:

- *Each client starts a new stream at the server.*

Advantage:

- *Each client has a private stream and controls it by using pause, play, stop, fastforward or rewind commands.*
- *Only streams that are viewed are streamed over the network.*
- *Better use of available bandwidth.*

VLC has a mode that allows it to work as a VOD server. This mode is called "VLM" (VideoLAN Media) and is controlled through the telnet interface. A VOD server uses the RTSP protocol (RFC 2326) to control the clients request. The default RTSP server port is 554, but only a root user may open any port below

Streaming networks with VLC

1024. Running VLC as root is ill-advised, it is better to use a higher port number (≥ 1024). In the example below 5554 is used as RTSP port number.

VLC as VOD streaming server:

```
$> vlc -I telnet --rtsp-host 0.0.0.0:5554
```

Use telnet to connect to the VLC telnet interface on port 4212:

```
$> telnet localhost 4212
Trying 127.0.0.1...
Connected to localhost.localdomain (127.0.0.1).
Escape character is '^]'.
Password: admin
Welcome, Master
> new terminator vod mux mp2t enabled
> setup terminator input /home/media/video/terminator/terminator_240x192\@750.mpg
> new dvd vod mux mp2t
> setup dvd input dvdsimple:///dev/dvd
> setup dvd enabled
```

VLC as client:

```
$> vlc rtsp://192.168.0.5:5554/terminator
or
$> vlc rtsp://192.168.0.5:5554/dvd
```

Firewalls

In the above paragraphs several setups for a streaming server have been discussed. Normally you would test the server setup first on the server system by running the client locally. The next step is then to run the client on another machine that has access to the server. At this point you'll find out that it works locally, but not on another machine. What is the problem, you wonder?

Streaming networks with VLC



Firewall blocking traffic



Figure 6: Firewall on server blocking traffic

Firewalls protect your machine from unauthorized access, but can also hinder you with allowed access. In the above scenario the firewall is most likely to block the traffic from the streaming server. A quick way of checking this is to disable the firewall on server and client system, temporarily. On Linux systems you can do as root on both client and server:

```
$> /etc/init.d/iptables stop
```

The firewall is now disabled and testing the streaming setup with the client on another computer can be done. If this works, then enable the firewall on the streaming server and test again.

Check with a network sniffer like `tcpdump` or `ethereal` if multimedia network traffic is leaving the server. VLC uses port 1234 by default for unicast. See below the output for a `tcpdump`-session in which no traffic is leaving the server on port 1234.

Streaming networks with VLC

```
$> /usr/sbin/tcpdump -i eth0 port 1234  
  
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode  
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes  
  
0 packets captured  
0 packets received by filter  
0 packets dropped by kernel
```

The following tcpdump output shows traffic that is leaving the streaming server on port 1234.

```
$> /usr/sbin/tcpdump -i eth0 port 1234  
  
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode  
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes  
  
10:53:55.205169 IP server.m2x.nl.32789 > 192.168.0.200.1234: UDP, length 1500  
10:54:19.418686 IP server.m2x.nl.32789 > 192.168.0.200.1234: UDP, length 1316  
10:54:19.454679 IP server.m2x.nl.32789 > 192.168.0.200.1234: UDP, length 1316  
10:54:19.490681 IP server.m2x.nl.32789 > 192.168.0.200.1234: UDP, length 1316
```

If there is still no traffic leaving the server, then a port on the server's firewall needs to be open. In our unicast example port 1234 needs to be allowed to leave the server.

```
$> iptables -A RH-Firewall-1-INPUT -p udp -m udp -s 192.168.0.0/24 --sport 1234 -j ACCEPT
```

If network traffic is leaving the server, then enable the firewall on the client side. Start up the video player (or use tcpdump, ethereal) on the client and see if video starts playing. If it doesn't, then the firewall on the client side needs to open a port. For instance (vlc's default port 1234):

```
$> iptables -A RH-Firewall-1-INPUT -p udp -m udp -s 192.168.0.0/24 --dport 1234 -j ACCEPT
```

Note: the difference between the above two firewall rules. On the client side you open the destination port (“--dport 1234”) on the server side you open the source port (“--sport 1234”).

In case of Video On Demand (RTSP) the client needs to send an SETUP request to the RTSP server. The firewall should accept traffic to the RTSP port (default port is 554). In the Chapter about “Video On Demand” the commandline example used port 5554 for RTSP traffic, because then VLC can run under

Streaming networks with VLC

normal user account. To open up the RTSP port on the server side for incoming requests the following rules should be put in the firewall (example for a Linux distribution):

```
$> iptables -A RH-Firewall-1-INPUT -p udp -m udp -s 192.168.0.0/24 --dport 5554 -j ACCEPT
```

Bandwidth

Initially the server systems sends only a few streams at the same time. After a while there is a need to add more streams. The question however is how many can be added? It turns out this question is not that easy to answer. The network infrastructure defines the maximum available bandwidth, while a multimedia stream consumes part of that bandwidth. I'll divide these in two paragraphs discussing what network bandwidth is available and how much bandwidth a multimedia stream consumes.

Network bandwidth

Today most systems have a 100/1000 Mbps NIC and are connected to a 100 Mbps Local Area Network (LAN). However this doesn't mean that your network or connection is configured to use the full 100 Mbps. It depends on the type of hubs, switches or routers that are used in the network infrastructure. In our example the 100/1000 Mbps NIC could be connected to a 10 Mbps hub. If this is the case then your streaming server can stream a theoretical maximum of 10 Mbps even when it capable of using the full 100 Mbps.

The tool `dmesg` (under Linux) shows the bootlog from the Linux kernel. It can tell you what network settings are used by the system. Is it on a 100 Mbps or a 10 Mbps network connection?

Streaming networks with VLC

```
$> dmesg | less
ip_tables: (C) 2000-2006 Netfilter Core Team
Netfilter messages via NETLINK v0.30.
ip_conntrack version 2.4 (8189 buckets, 65512 max) - 232 bytes per conntrack
ADDRCONF(NETDEV_UP): eth0: link is not ready
eth0: network connection up using port A
  speed:      100
  autonegotiation: yes
  duplex mode: full
  flowctrl:   none
  irq moderation: disabled
  scatter-gather: disabled
  tx-checksum: disabled
  rx-checksum: disabled
ADDRCONF(NETDEV_CHANGE): eth0: link becomes ready
```

In

the above output there are three settings that are important to control: speed, autonegotiation and duplex mode. Fix these three settings to the maximum that your network equipment is capable of. To get a reliable 100 Mbps network disable the autonegotiation, because it doesn't always use the best setting that is available.

In our example above full-duplex mode is used, which means the 100 Mbps is in both directions at the same time. If it says half duplex, then the network connection will have to take turns in using the cable. This will result in extra delays for the network connections.

All network equipment: hubs, routers and switches between streaming server and client must support the same settings. In our example they all need to support 100 Mbps, no autonegotiation and full duplex in their network connections. Once all systems (server and client) NIC's, hubs, routers and switches are configured correctly, then 70-80% of their capacity can be used for multimedia streaming.

Multimedia Stream bandwidth

The bandwidth of the multimedia stream depends on the source of the stream. Is it read from satellite (DVB-C/S/T), DVD, Acquisition card, Webcam or a file. Some typical numbers are:

Source	Bandwidth
DVB-C/S/T	Average: 3.5 Mbps Maximum: 5 Mbps
DVD	Average: 5 Mbps Maximum: 10 Mbps

Streaming networks with VLC

<i>Source</i>	<i>Bandwidth</i>
Acquisition Card	Varies but most are 1 Mbps
WebCam	Varies but most are <1 Mbps
Files	Varies but most are <1 Mbps

The table already indicates there is no such thing as the "bandwidth" for a multimedia stream. When using files as input it is even more difficult to predict what the bandwidth will be, because of the many different ways it can be encoded (MPEG-2, MPEG-4, DivX, H264, Theora, etc.).

The bandwidth of a multimedia stream is not constant. It varies over time, this is called **Variable bitrate (VBR)**. In the following figure this is illustrated with a fictive example. For clarity the average, real and maximum bandwidth have been plotted.

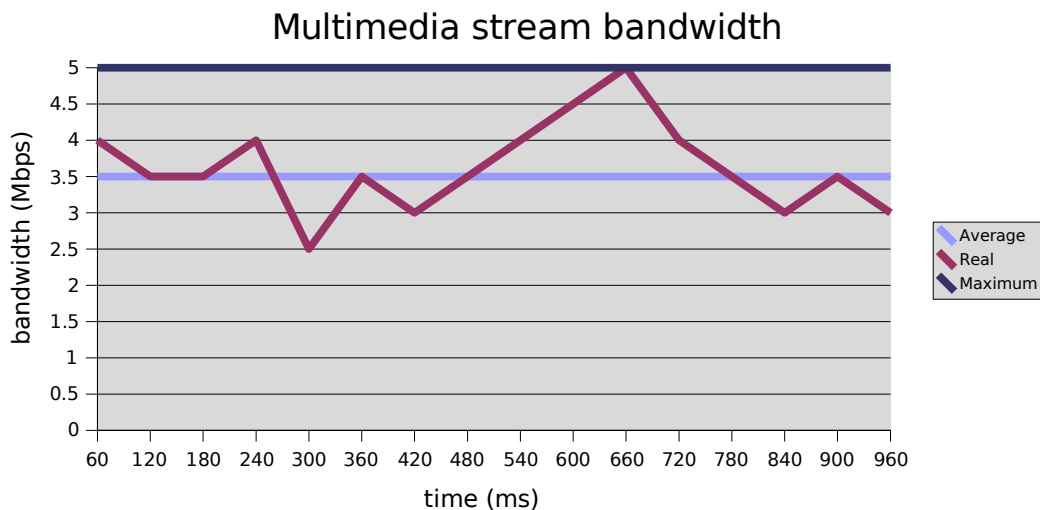


Figure 7: Multimedia stream bandwidth example

The bitrate is variable because Audio/Video encoders partition the video in different type of frames. These frames have different sizes. They are called Intra-coded (I-frame), Predictive-coded (P-frame) and Bidirectionally predictive-coded (B-frames). The I-frame is the largest frame, followed by the P- and B-frame.

I P B B P B B B P B B P

A video stream is constructed of a combination of I-, P- and B-frames. The highest peak in the bandwidth is caused by an I-frame. P- and B- frames cause smaller peaks, since they carry less information than an I-frame. A P-frame needs information from an I-frame, and a B-frame needs both P- and other B-frames. **Constant Bitrate (CBR)** streams are actually variable bitrate streams padded with zeros.

Figure 7 shows that the average bandwidth and real bandwidth never exceed the maximum multimedia

Streaming networks with VLC

bandwidth. The maximum bandwidth is determined from the highest peak of the real (observed) multimedia bandwidth.

$$\text{Avg}_{(\text{bandwidth})} < \text{Max}_{(\text{bandwidth})} \text{ and } \text{Real}_{(\text{bandwidth})} \leq \text{Max}_{(\text{bandwidth})}$$

Network planning

When planning a multimedia network the maximum bandwidth of the stream is taken into account. Not the average bandwidth, because that will result in network packet loss if several of the peaks occur at the same time. Then there is simply not enough bandwidth available. Dropped packets manifest itself in loss of picture and loss of audio quality.

In above example UDP network transport is assumed. UDP is a unreliable network protocol since it doesn't correct lost packets. When an UDP packet is dropped in the network then it is lost forever. You might be tempted to use TCP based protocols instead to counter the problem. Think again. UDP traffic is only dropped at the network level when there is not enough bandwidth, but a TCP packet is retransmitted when dropped. The retransmission happens on an already loaded network!

When a TCP packet is dropped on the network, the server will notice this and retransmits the lost packet. If the network is loaded then this will result in even more packet loss. The retransmission makes the load even higher. This is what I call the **TCP cascade effect**. In a streaming network a TCP based network will fail sooner than an UDP network. With multimedia streaming there is only a small window in time (40ms-200ms) that a lost packet is still useful. If it arrives later, then it will be too late to decode and display and is thus dropped by the multimedia player.

The following table show the number of multimedia streams a network can handle using a satellite (DVB-S) as source. Remember to load your network only for 80%.

<i>Network</i>	<i>Max_{load}</i>	<i># Streams</i>
10 Mbps	8 Mbps	1
100 Mbps	80 Mbps	16

Streaming multimedia in the same quality as the source (think of DVD's or DVB's streams) consumes most of the bandwidth available. In the above table only one DVB-stream can be sent on a 10Mbps network, while 16 streams can be sent on a 100 Mbps network. To overcome this limitation the quality of the stream can be lowered.

Lowering the quality of the source stream is done using a technique called transcoding. Transcoding involves the complete decoding of the source stream and encoding it with new settings. This process will reduce the quality of the source stream. In the example below a DVD (typical bandwidth up to 10 Mbps) is transcoded to a stream of 1,5 Mbps on average.

Streaming networks with VLC

```
$> vlc dvdsimple:///dev/dvd -sout '#transcode{vcodec=mp2v,vb=1024, \
acodec=mpga,ab=192,channels=2}:std{access=udp,mux=ts,dst=192.168.0.20}' \
-sout-ffmpeg-strict-rc
```

Translating this to our DVB example will give us a better utilization of the network bandwidth. A 10 Mbps network can now transmit DVB 5 streams, while a 100 Mbps will transmit 53 DVB streams. This is 3-5 times as much number of channels that can be sent.

<i>Network</i>	<i>Max_{load}</i>	<i># Streams</i>
10 Mbps	8 Mbps	5
100 Mbps	80 Mbps	53

Summary

The VLC media player is both a streaming server and a multimedia player capable of playing almost all multimedia files. As long as there is an Open Source compatible implementation of the underlying codecs. VLC media player is well-known for its compliance to streaming standards and used by industry players in the broadcast, consumers and R&D field. It is easy to setup using the GUI wizard or one of the other interfaces that VLC offers.

The VideoLAN website hosts HOWTO documents describing the most used VLC options and offers various support channels (IRC, mailinglists, Forums).

Setting up a streaming server and network using VLC media player requires knowledge of networking infrastructure and multimedia characteristics. In case of troubles there is always the VideoLAN support channels where the developers hang out. The best advise to debug a problem is to start simple and scale up from there.