



# Streaming Ahead

With QuickTime

What are our options?

Overview of how this system works

Advantages of using this relaying technique

Streamer Server setup

Relay Server setup

UTstreamer Java applet

References and tips

AARON LAND

CENTER FOR INSTRUCTIONAL TECHNOLOGIES

THE UNIVERSITY OF TEXAS AT AUSTIN

[HTTP://WWW.UTEXAS.EDU/ACADEMIC/CIT](http://www.utexas.edu/academic/cit)

FEB 2003

## What are our options?

In the streaming video world there are only three significant competitors: Microsoft's Windows Media, RealNetwork's RealVideo, and Apple Computer's QuickTime. All three have their own formats and differing levels of compatibility in this multiplatform world. The University of Texas at Austin is looking to push the performance of these products beyond anything yet advertised. With its large student body and the anticipated growing demand for streaming video in the future, UT Austin will be pushing the limits of streaming technology to its fullest.

Using a single QuickTime, Real Video, or Windows Media Server works well for streaming high-quality content to about 100 viewers. When that number grows to over 500 however, the hardware and network can start to buckle under the increased load. To design a high-quality system to handle in excess of 5,000 streams and beyond, we must discard conventional approaches. (A high-quality stream means a larger image size at higher frame rates and near loss-less quality imagery than what we see on average on the Internet today.)

A distributed network would help farm out the anticipated expanding load of streaming video among many servers to reduce the per-server and network stress. Apple's QuickTime Streaming server can be configured to handle such a large task and provide scalability. This "server-farm" technique has been used for handling large computations that scientists demand, and has been used on UT Austin's campus for many years.

This paper will describe a configuration design that enables many servers to work together, and that provides the maximum in efficiency and compatibility to an extremely large volume of users for today's audience and audiences of the future.

## Overview of how this system works

How the overall system will look and its interconnections are outlined in Figure 1. This layout provides functional reliability, along with many options for expansion and upgrades. Another strength is in the streaming software, which allows the system to route different individual kinds of requests.

Broadcaster, Streamer Server, Relay Server, and Multicast Relay are names that have been applied to the duties of individual hardware and software components. These names will be used throughout this document. We start with a Broadcaster, which can be a computer or hardware box. Its only goal is to take a source, usually video and audio (but not limited to just these two), and send it to a Streamer Server. This Streamer Server is then used as a router to pass the incoming broadcast to a bank of Relay Servers for mass distribution.

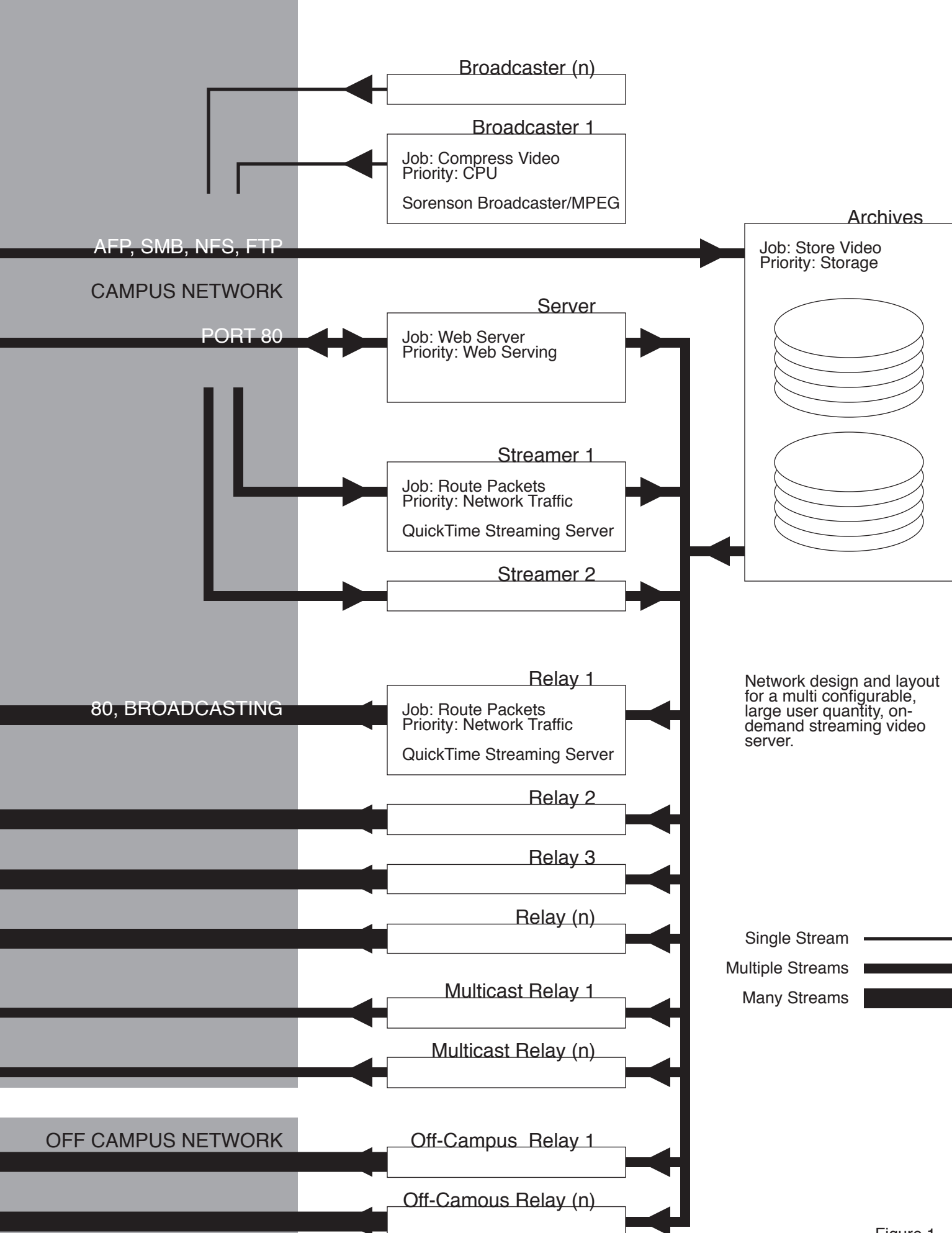


Figure 1

# Advantages of using a relayed server system:

## **Live broadcasts:**

One Streamer Server can accept about 500 high-quality live video broadcasts, distributing them down the pipe to all of the relays. Adding a second Streamer Server provides backup for the first as well as doubling the total number of broadcasts, tripling with a third, and so on. Each broadcast is then sent down a private network to the Relay Servers for distribution. Isolating each Relay Server to only this task helps streamline the overall system, especially in handling the load of the network.

## **Separate relay servers:**

Each Relay Server can simultaneously handle both live and prerecorded events. Live broadcasts would come directly from a Streamer Server, while prerecorded media would come from a central archive. A single Relay Server can handle about 500 high-quality streams and having a bank of Relay Servers enables delivery to an ever-expandable number of viewers. Each server works independently, using the same sources but with separate network connections. The more servers there are, the more reliable the overall system is. Further additional Relay Servers can be placed elsewhere in separate buildings or other schools to reduce network usage and increase reliability of the stream.

## **Multicast broadcasts:**

Any Relay Server could easily be configured to provide multicast streams. A multicast signal stream is similar to a television station: one stream is sent to the whole campus for anyone to watch. Some examples for using a multicast relay would be live events, such as a public lecture, commencement address, or football games.

## **Off-campus networks:**

Off-campus streams, for off-campus students on DSL/cable or high school classrooms will not have the speed of the on-campus network. To solve this problem Relay Servers could be placed at more immediate locations (as well as in individual buildings on campus) to provide more reliable access.

## **Compatible content:**

This system is not restricted to just QuickTime content. This system routes any kind of *rtsp* (real-time streaming protocol) packets, which is the Internet standard for streaming media. Real Video is compatible with *rtsp*, but Windows media uses a proprietary, non-standard protocol, so this system will not be able to stream Windows Media.

**Cost effectiveness:**

All the software for this system is free. One of the disadvantages of Real is the additional per-user license fees, which can grow to be very costly. Another is the use of a single large server. Although there are ways to make multiple server setups, they would each need their own separate setups and administration. Given the distributive properties of the proposed system, no upgrades are needed to make existing servers bigger or faster. Instead, similar servers are added without ever shutting down the overall system. This design becomes even more cost-efficient as the overall user load grows.

<b>~500 Users</b> (single servers)	<b>Software Cost</b>	<b>Hardware Cost</b>	<b>Total</b>
QuickTime	Free	\$3,000 (Apple Xserve)	\$3,000
Windows Media	(NT Ad Server included)	\$3,000 (Dell PowerEdge 1650)	\$3,000
Real Video	\$2,000	\$2,000 (Sun Fire V100)*	\$4,000

<b>~1000 Users</b>	<b>Software Cost</b>	<b>Hardware Cost</b>	<b>Total</b>
QuickTime	Free	\$4,000 x3 (Apple Xserve)	\$12,000
Windows Media	(NT Ad Server included)	\$26,000 (Dell PowerEdge 6650)	\$26,000
Real Video	\$5,000 - \$8,000	\$20,000 (Sun Enterprise 420R)*	\$25,000 - \$28,000

<b>~5000 Users</b>	<b>Software Cost</b>	<b>Hardware Cost</b>	<b>Total</b>
QuickTime	Free	\$4,000 x10 (Apple Xserve)	\$40,000
Windows Media	(NT Ad Server included)	\$60,000 (Compaq ProLiant DL760)	\$60,000
Real Video	\$10,000	\$80,000 (Sun Enterprise 3500)*	\$90,000

- **Note:** Real Video works on the same hardware as Windows Media, but is optimized for UNIX. Prices and hardware as of spring 2003.

## Streamer Server setup:

The Streamer Servers in this layout handle all the incoming streams and send them to the Relay servers. Below is a simple example of how to set up broadcast users, each of whom will be able to broadcast to the Streamer Servers. If there are multiple Streamer Servers, this setup must be copied or customized for each. In this example there are four Broadcasters (one being the Administrator); each will be granted individual access privileges to the server.

**Note:** These steps require knowledge of the UNIX terminal and super-user access. All files must be UNIX text files, not .txt, .rtf, etc., and should be edited through the UNIX terminal to ensure accuracy. Basic knowledge of QuickTime Streaming Server is also required.

### Creating broadcaster user accounts:

```
[streamer1: ~] Admin% sudo qtpasswd -p *password* admin
[streamer1: ~] Admin% sudo qtpasswd -p *password* bat7
[streamer1: ~] Admin% sudo qtpasswd -p *password* geo4
[streamer1: ~] Admin% sudo qtpasswd -p *password* com8
```

Four users have been created. This data is stored in the file /Library/QuickTimeStreaming/Config/qtuser. Here is an example of what you will find:

```
real m Streaming Server
admin: . 8Mf6q/EalR3. : f5a9c24140af4372629c8fab6de4ff61
bat7: Qx8wguS11U08s: e8a447c626c62ab083a92839b61d62cf
geo4: asdh65q4l k2j 346bn2k3j 46l 2j k346b2k3l : l sj vmzx9044
com8: 689j h2l j k3347j h457: k2hklj 7h257. j 2h58j h2j k4h85
```

These users can be stored as a “group” for additional customizations. This data is stored in the file /Library/QuickTimeStreaming/Config/qtgroups. Change that file to reflect the setup.

```
admins: admin
broadcasters: bat7 geo4 com8
```

### Setup directory access:

For security purposes, it is necessary to limit who has access to the system and what each authorized user can do. Each Streamer Server should be set up to allow incoming broadcasts from individual Broadcasters created like those above and to prohibit others from using the server to stream directly.

In order to control global access, we create a qtaccess file in /Movies with the information below. This will allow only the administrators to have access to the root directory and all subdirectories without their own qtaccess file. **Note:** all qtaccess files should be owned by root and allow only write-access by root and administrators for security.

```
<Limit WRITE>
require group admins
</Limit>
require group admins
```

Only users in the group admins can read and write.

Administrators can then create access for individual users: Inside /Movies make a directory called /Live; inside /Live make a directory for each user. This exact layout is not necessary but is highly recommended, as it will keep all the *.sdp* files separate and allow for individual user access configurations.

```
/Movies/Live/admin/  
/Movies/Live/bat7/  
/Movies/Live/geo4/  
/Movies/Live/com8/
```

Create a *qtaccess* file inside /Movies/Live/bat7/ and enter the information based on the example below. This is how Broadcaster bat7 is allowed to broadcast to this server. Adding the Broadcaster again at the bottom will also allow them to stream **from** this server, but this should be used only for testing purposes. Repeat this step for each user in the appropriate folder.

```
<Limit WRITE>                Only user admin and bat7 can read and write.  
require user bat7  
require group admins  
</Limit>  
require user bat7  
require group admins
```

### Relay settings:

Using the QuickTime Streaming Server Admin, create individual relay streams to each Relay Server. Load <http://localhost:1220/> into a browser, log in and click Relay Settings.

**Relay Details**

Relay Name:

Status:  Enabled

---

**Source Settings**

Source Hostname or IP Address:

Mount Point:

Request incoming stream

User Name:

Password:

Wait for announced stream(s)

### Destination settings:

Create a destination setting for each of the Relay Servers. Hostnames should be the addresses on the private network. Make sure all Mount Point files are in the /Live folder and have the same name.

#### Destination Settings (1 of 2)

Hostname or IP Address:

Relay via TCP

Mount Point:

User Name:

Password:

Relay via UDP

Base Port:

Multicast TTL:

#### Destination Settings (2 of 2)

Hostname or IP Address:

Relay via TCP

Mount Point:

User Name:

Password:

Relay via UDP

Base Port:

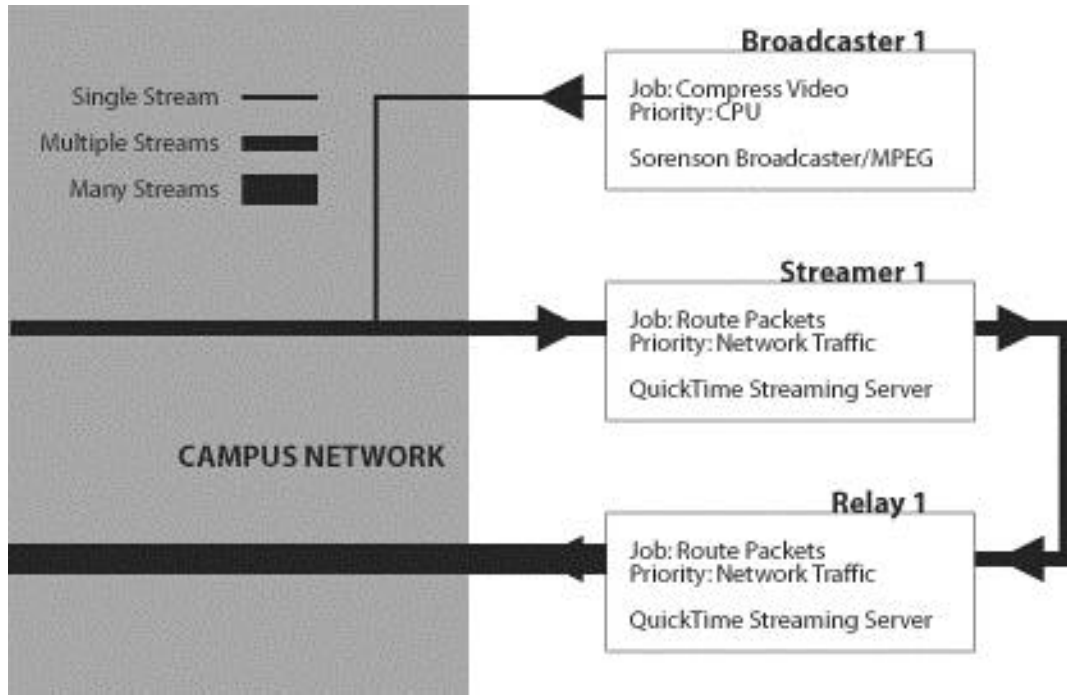
Multicast TTL:

### Final step:

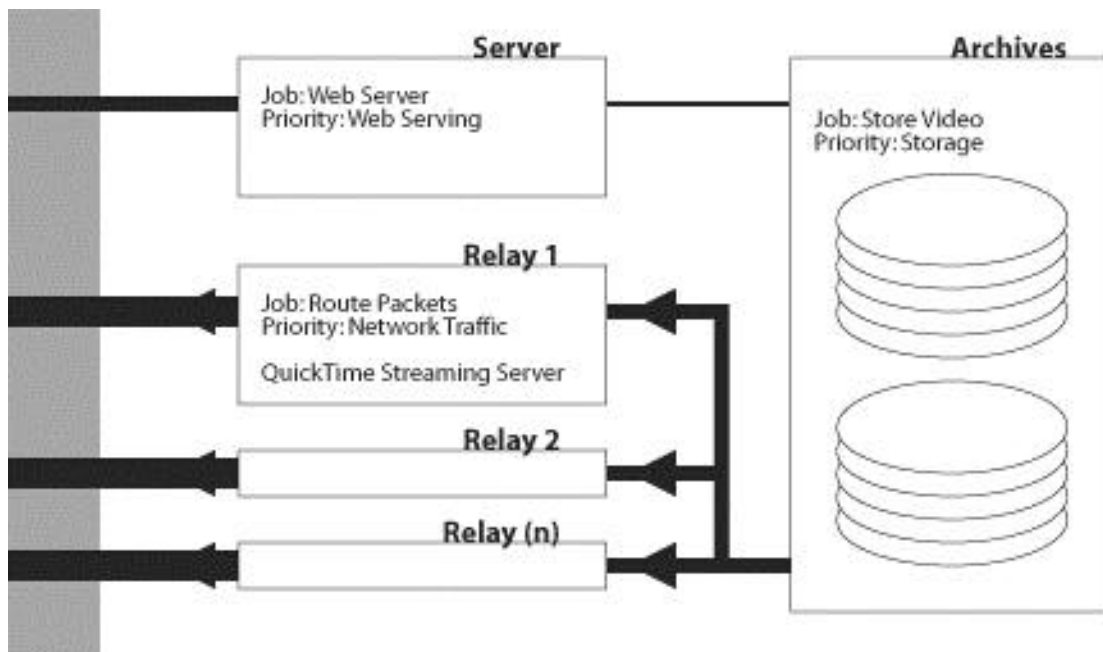
Repeat all these steps for each additional Streamer Server.

## Relay Server setup:

The Relay Server simply takes streams from a Streamer Server and relays them to the individual viewers. The more Relay servers added to the system, the greater the number of viewers who can simultaneously use the system.



Relay servers can take live and archived media simultaneously and stream them on demand.



## Create Relay user accounts:

As with the Streamer Server, we need to implement user access restrictions to prevent improper use of the network. Only one user needs access to this server: the user account used by the Streamer Server above.

```
[relay1: ~] Admin% sudo qtpasswd -p password admin
[relay1: ~] Admin% sudo qtpasswd -p password relay1
```

/Library/QuickTimeStreaming/Config/qtusers:

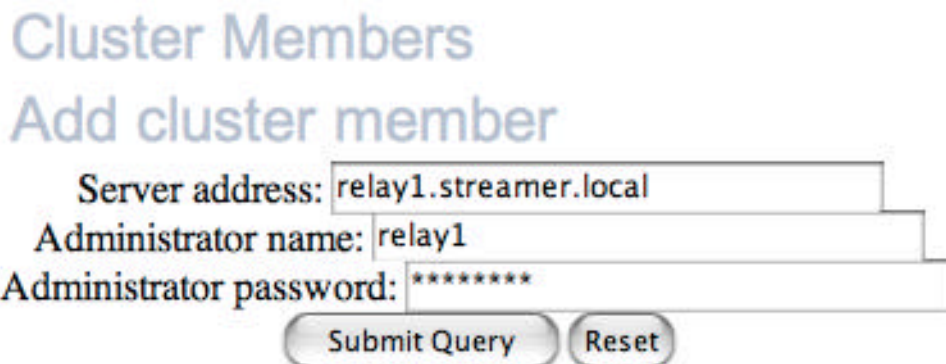
```
real m Streaming Server
admin: 234sdfh6. 234dfh2: 125812l hklj hkj 2hsdsdfhj 78d: ff
relay1: lj kh7j qery. k3h78qer: 347lj qhe7lj qebnqng3q344j
```

/Library/QuickTimeStreaming/Config/qtgroups:

```
admins: admin
```

## Install DarwinCluster

The Relay Servers can be load-balanced using a plug-in called DarwinCluster. This tool allows one server to act as a master and distribute its load automatically among any number of additional servers. This tool is not required but can make the overall server process run more smoothly. Extract DarwinCluster on each Relay Server and run the installer as root (. /Install). Open a Web browser and go to <http://localhost:1220/clusterparse.cgi>. **Important: make sure Content management is DISABLED.** It is enabled by default; when it is enabled the servers will mirror the media among all the servers. Our media is coming from an archive and will not need this; therefore files (in /Local for example) must be updated manually. Make one server the Master and add each additional server as a slave.



Cluster Members

Add cluster member

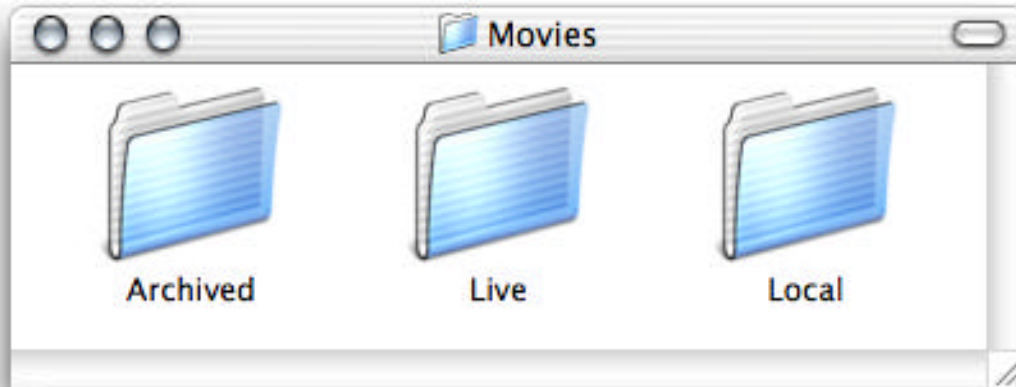
Server address:

Administrator name:

Administrator password:

## Set up directories:

Create a standard root directory /Movies as created in Streamer Server setup, and make three directories inside it: /Archived, /Live and /Local as shown below.



- /Archived - NFS mount to Archives
- /Live - Directory for all live broadcast files
- /Local - Directory used for commonly used movies  
(Contents must be mirrored across all Relay Servers)

As before, create a qtaccess file in the /Movies directory to restrict the default access to this server. Allow only the Streamer Server user to stream to this server, but anyone can watch the content on the server. **Note:** Further access restrictions are possible.

```
<Limit WRITE>                Only admins can read and write; anyone can read.  
require group admins  
</Limit>  
require any-user
```

Inside /Movies/Live, create this qtaccess file. User directories are not needed here because the relay will rename them appropriately.

```
<Limit WRITE>                Only admins and relay1 can write.  
require group admins  
require user relay1  
</Limit>  
require any-user                Anyone can read this stream.
```

## Final step:

Repeat all steps for each additional Relay Server.

# UTstreamer Java applet:

## Loading the applet:

UTstreamer is a Java applet that reads in an XML configuration file to display video, text, and picture content. The simplest way to load an applet is through a Web browser. Here is an example:

```
<html >
<head>
<title>UTstreamer Example</title>
</head>
<body>
<applet archive="/v1/UTstreamer.jar" code="UTstreamer" width="x" height="y">
  <param name="url" value="http://streaming.server/dir/xmlfile">
</applet>
</body>
</html >
```

The minimum requirements are: the location or the .jar file (this can be universal for all presentations); the code to run (should always be “UTstreamer”); the absolute width and height. Additional standard applet elements apply. The location of the XML file to be loaded must be added to a parameter tag with the name “url” and a value for the location.

## Writing the XML file:

The XML file has a standard format that must be followed for the applet to read properly in a presentation. The XML file starts out with a standard header and a <stream> element. Within it a <viewer> is added to show where the movie display area is. Within that is a <video> element showing where the movie is located. Here is an example:

```
<?xml version="1.0" encoding="UTF-8"?>
<stream>
  <viewer>
    <video src="rtsp://web.server/location/of/movie.mov">
  </viewer>
</stream>
```

This is just a skeleton. Additional information needs to be added to complete a useable file. The next few pages will show a complete list of possible values to use to create a presentation.

## XML file layout and defaults:

Within this basic shell, much can be added to configure a presentation. By adding elements within <stream>, different media can be added to create a more complete and customized presentation. Each element has a set of properties that can be changed. If no value is entered then a predefined default value is used. Here is a list of possible elements and their default values.

```
<stream>
  <info
    title =      "Untitled"      Presentation title
    year  =      "0"             Year as number
    month =      "0"             Month as number
    day   =      "0"             Day as number
    hour  =      "0"             Hour as number
    minute =     "0"             Minute as number
    width =      "320"           Width of applet
    height =     "240"           Height of applet
  />
  <viewer
    posx =      "0"             Position in X dimension from top left of window
    posy =      "0"             Position in Y dimension from top left of window
    width =     "320"           Width in X dimension
    height =    "240"           Width in Y dimension
    scale =     "30"           Frames per second (This affects all "time" events.)
  >
  <video
    src =       Will fail if empty
    posx =     "0"             Position in X dimension from top left of viewer
    posy =     "0"             Position in Y dimension from top left of viewer
    width =    "320"           Width in X dimension
    height =   "240"           Width in Y dimension
    cropx =    "0"             Crop from left
    cropy =    "0"             Crop from top
    cropwidth = "0"           Crop from right
    cropheight = "0"         Crop from bottom
    time =     "0"             Event time in viewer timeline
    starttime = "0"           Crop time from beginning
    stoptime = "-1"          Crop time from end (default play to duration)
    volume =   "100"          Volume from 0 to 100 scale
  />
  <audio
    src =       Will fail if empty
    time =     "0"             Event time in viewer timeline
    starttime = "0"           Crop time from beginning
    stoptime = "-1"          Crop time from end
    volume =   "100"          Volume from 0 to 100 scale
  />
```

```

<carousel    Holds slides; multiple carousels are allowed
  posx =      "0"          Position in X dimension
  posy =      "0"          Position in Y dimension
  width =     "320"        Width in X dimension
  height =    "240"        Width in Y dimension
>
  <slide
    src =      Will fail if empty
    posx =     "0"          Position in X dimension
    posy =     "0"          Position in Y dimension
    width =    "320"        Width in X dimension
    height =   "240"        Width in Y dimension
    cropx =    "0"          Crop from left
    cropy =    "0"          Crop from top
    cropwidth = "0"        Crop from right
    cropheight = "0"       Crop from bottom
    time =     "0"          Event time in viewer timeline
  />
</carousel>
</viewer>
<chapters
  posx =      "0"          Position in X dimension
  posy =      "0"          Position in Y dimension
  width =     "320"        Width in X dimension
  height =    "240"        Width in Y dimension
>
  <chapter
    name =     "New Chapter" Chapter name
    time =     "0"          Event time in viewer timeline
  />
</chapters>
<controller
  posx =      "0"          Position in X dimension
  posy =      "0"          Position in Y dimension
  width =     "320"        Width in X dimension
  height =    "240"        Width in Y dimension
/>
<chat
  posx =      "0"          Position in X dimension
  posy =      "0"          Position in Y dimension
  width =     "320"        Width in X dimension
  height =    "240"        Width in Y dimension
>
  <text
    msg =      "String"    Message
    time =     "0"          Event time in viewer timeline
  />
</chat>

```

```
<caption
  posx =      "0"           Position in X dimension
  posy =      "0"           Position in Y dimension
  width =     "320"        Width in X dimension
  height =    "240"        Width in Y dimension
>
  <text
    txt =     "String"     Chapter name
    time =    "0"          Event time in viewer timeline
  />
</caption>

</stream>
```

## Layout example:

Here is an example showing how to use a single carousel slide presentation, with live video, chapter list, and controller. **Note:** The viewer in yellow includes the carousel and video rectangles.



**Note:** The ↵ are line breaks to aid reading and are not in the actual XML file.

```
<?xml version="1.0" encoding="UTF-8"?>
<stream>
  <info title="UTstreamer Test" year="2002" month="12" day="25" hour="0" ↵
minute="0" width="740" height="395"/>
  <viewer posx="100" posy="0" width="640" height="360" scale="15">
    <video src="rtsp://web.server/location/of/movie.mov" posx="480" ↵
posy="0" width="160" height="120" starttime="165" stoptime="43320"/>
    <carousel posx="0" posy="0" width="480" height="360">
      <slide src="http://web.server/location/Slide1.jpg" time="0"/>
      <slide src="http://web.server/location/Slide2.jpg" time="6135"/>
      <slide src="http://web.server/location/Slide3.jpg" time="8565"/>
      <slide src="http://web.server/location/Slide4.jpg" time="13440"/>
      <slide src="http://web.server/location/Slide5.jpg" time="14640"/>
      <slide src="http://web.server/location/Slide6.jpg" time="24630"/>
      <slide src="http://web.server/location/Slide7.jpg" time="35100"/>
      <slide src="http://web.server/location/Slide8.jpg" time="36495"/>
      <slide src="http://web.server/location/Slide9.jpg" time="42840"/>
    </carousel>
  </viewer>
  <controller posx="0" posy="360" width="740" height="35"/>
  <chapters posx="0" posy="0" width="100" height="360">
    <chapter name="Start" time="0"/>
    <chapter name="Chapter 1" time="6135"/>
    <chapter name="Chapter 2" time="8565"/>
    <chapter name="Chapter 3" time="35100"/>
  </chapters>
</stream>
```

## References and tips:

### Server ports:

Setting up security is always important. As much as possible, restrict access to the outside network with a firewall. Administration should be permitted only within the private network. A separate administrative computer could be added on the private network for making changes. The user accounts set within the QuickTime Streaming server are not system accounts and handle only streaming authentication.

Wide Area Network: Allow only outgoing streams:

RTSP/UDP	554	Streaming Media Port
TCP	554 or 80, 7070	Client Communication
UDP	6970-6999 (or 80 TCP)	Sending Media
UDP	10000-65635	Receive Broadcasts
TCP	8000	MP3 broadcasts (if used)

Local Area Network: Open more ports for administration and file sharing:

Same as above.

TCP	20-21	FTP
TCP	22	ssh (secure shell)
TCP/UDP	53	DNS
UDP	68	DHCP (client)
TCP	137-139	Windows File Sharing
UDP	137-138	Windows File Sharing
AFP/TCP	548	Apple File Server
TCP	1220	QTSS Admin
UDP	2049	NFS

To limit the ability to control the QTSS Admin Web site to only the LAN, edit the file `/Library/QuickTimeStreaming/Config/streamingadminserver.conf` and add to the end `bind=1. 2. 3. 4` where 1.2.3.4 is the IP address of the LAN network adapter.

### User Folders:

If there is a separate server with UNIX user accounts and you want them to be able to stream movies from their home folder, do the following: Create a `/Users` folder inside the current `/Movies` folder, then make a `/Movies` folders in the user's home folder and create an alias from the QTSS `/Movies/Users` folder to the user's `Movie` folder. Situations may vary.

```
cd /Movies/Users
```

```
ln -s /Users/user1/Movies user1
```

or

```
ln -s /Users/user1/Movies "~user1"           To act like an Apache user folder.
```

To access the files use this link.

```
rtsp://qtss.server.com/user1/movie.mov
```

or

```
rtsp://qtss.server.com/~user1/movie.mov     To act like an Apache user folder.
```

**More Documentation and links:**

For more information about QuickTime Streaming Server, visit.  
<http://www.apple.com/quicktime/products/qtss/>

For more information about Xserve, visit.  
<http://www.apple.com/xserve/>