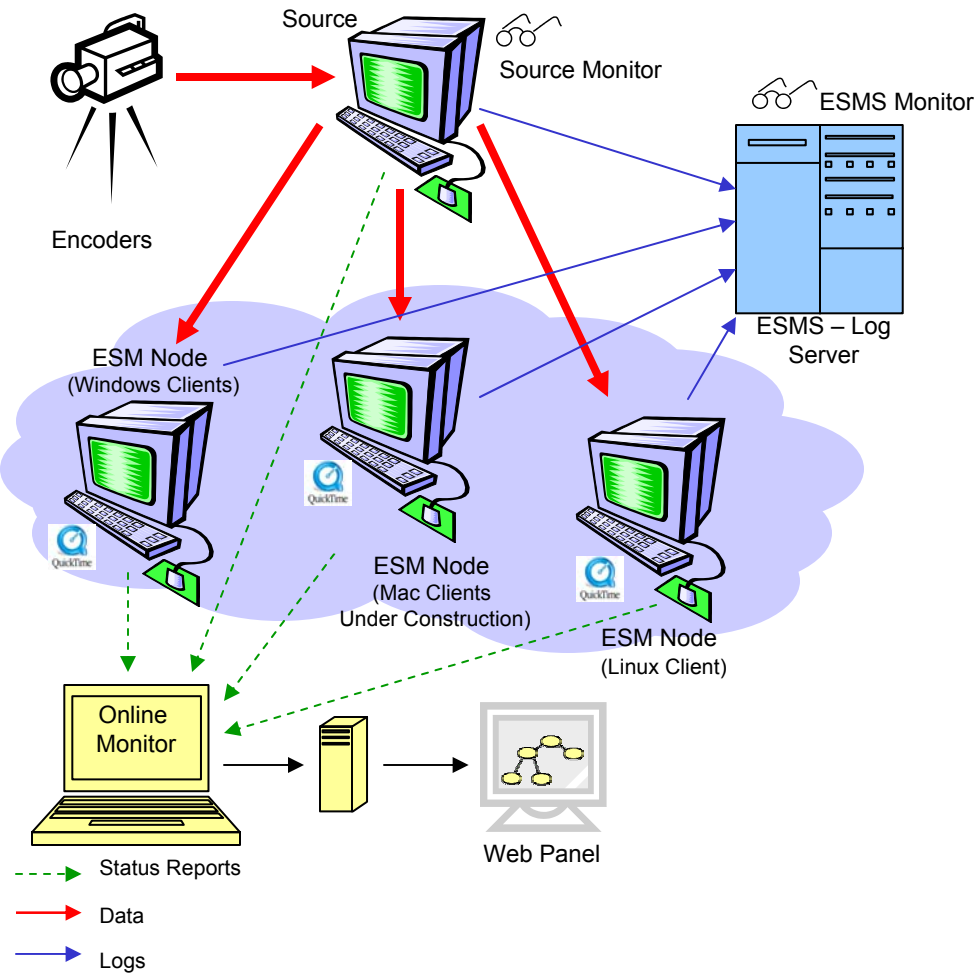
ESMS – Log Server

Brian Goodman

Overlay tree gets quite large (we hope). How to gather performance/error logs in a robust and scalable fashion?

Constraint: We would like to avoid recompiling the kernel.

- *Problem:* Linux has a default limit of 512 threads per process
  - *Solution:* Don't use threads!
- *Problem:* Select() can poll only 1024 file descriptors by default (we need 2 per connection!)
  - *Solution:* Multiple instances per machine– run each one on a different port
  - *Solution:* Multiple machines–give clients a list of machine/port pairs; they can pick one randomly for each connect
- *Problem:* On a dropped connection/failed ESMS, how much data was lost in transit? What should be re-sent?
  - *Solution:* Keep a buffer of recently sent data, at least as large as the kernel buffer
  - *Solution:* On reconnect, sent byte number to begin from, entire buffer– ESMS can simply seek() and begin appending!



Linux Porting

Chris Palow

ESM consists of three parts for the viewers: the overlay node, a QuickTime player, and the program that coordinates the interaction between the other two.

- *Problem*: Find a QuickTime player for Linux:
  - Has to support a video and an audio codec we can broadcast
  - Has to support receiving the broadcast in a RTP stream
- *Possible Solution*: Open source players
  - They support codecs we use
  - They don't support the encapsulation in RTP streams.
- *Solution*: Emulate the Windows Apple QuickTime Player.
  - Not "true" emulation.
  - ISA is the same just the system calls that are different
  - Wine wraps Windows system calls to their Linux equivalents.
- *Not a problem*: ESM's overlay code is UNIX native
  - it works in Windows by using the Windows analog of Wine: Cygwin.
- *Problem*: Program that coordinates all the individual components that make up our system was Windows specific
- *Solution*: Rewrite the code to be more portable

Online Monitoring System

Jiin Joo Ong

- *Problem*: How do we "view" the overlay network?
  - Ideally, want to know n^2 bandwidth, delay, etc.
  - Want an easy way to spot errors in the system, aid debugging.
- *Solution*: Dedicated machine that receives feedback from all components of the system
  - Each ESM Node sends UDP packets reporting their status.
  - Source Monitor, Web Server Monitor and Esms Monitor give detailed status report.
  - Information is parsed and presented in graphical format instantly. (~10 sec.)
  - Web access to information, ubiquitous monitoring of the system.

Robustness of Key Components

- Source Monitor and Esms Monitor serve as process restart agents.
  - Detects unusual activity. (e.g. source process / Esms process died)
  - *Auto-Recovery*! Restarts process, automatically adjust logging mechanism immediately (~5 sec.)