Jupiter Video Protocol

Ron Frederick

Xerox PARC

Architecture Overview

Clients connect to a centralized server via TCP

- Connection is used for normal text output & out-of-band control messages
- Control messages manage audio, video, and windows
- Low bandwidth traffic only: media data is sent peer to peer

Server plays a coordinating role

- Key management and multicast address management
- Shared window state (whiteboard, etc.)

Long term plan is to do a distributed server implementation

- Avoid the single point of failure & performance bottleneck
- Better cope with user communities spread across administrative domains

Jupiter Client & Media Agents

Server could be eliminated entirely

- Use local session managers & agreement protocol

Jupiter client has no inherent UI

- Server downloads window layout into client
- Widget events are sent back to server for action

Videosend Protocol (Server->Client)

#$#videosend-list-sources

Requests available video sources from this client

#$#videosend-set-ttl ttl: NUM

Sets the multicast TTL for all video packets sent by this client

#$#videosend-start source: SNAME dest: IPADDR/PORT

Starts a video stream from a source to the specified destination

#$#videosend-stop source: SNAME [dest: IPADDR/PORT]

Stops a video stream, if it exists, from a source to the specified destination
If destination is omitted, all video streams from that source are stopped

#$#videosend-set-attrs source: SNAME ...

Sets video attributes for specified source

Videosend Protocol (Client->Server)

#$#videosend-sources list: STR

- Reply to $#$videosend-list-sources
- Returns triples of the form “SNAME STYPE SID”
- Source type affects which attributes are valid
- SID is a network identifier (for RTPv2, just the SSRC)

#$#videosend-frame-size source: SNAME size: “W H”

- Sends the size of the current video image from this source to the server
- Server assumes 160x120, 320x240, or 640x480 based on “size” attribute
- This message is sent to override that assumption

Videosend Attributes

Universal:

- color: BOOL (default: true)
  - Send color video data or not
- size: {small, medium, large} (default: medium)
  - Transmitted video size (roughly 160x120, 320x240, and 640x480)
- max-bandwidth: NUM (default: 128 kbps)
  - Transmitted bandwidth in kbps
- min-frame-spacing: NUM (default: 0 msec)
  - Minimum spacing between frames in msec
- key: HEXSTR (default: none)
  - Encryption key to use, specified in hex

Videosend Attributes (cont.)

VideoPix specific:

- port: {Composite1, Composite2, SVideo}
  - Input port to get video from

Screen specific:

- follow: {pointer, region, window}
  - Area of screen to transmit
  - Pointer follows mouse pointer
  - Region prompts user to drag out a region
  - Window prompts user to select a specific window

VideoPane Widget

Incoming video is displayed in a VideoPane widget

- Each videopane displays a single incoming stream
- Streams are identified by a ‘vid’ (SID/DESTIP/DESTPORT)
- Natural size is source dependent - scaling is done if necessary

#$#win-set-vid id: STR widget: STR address: VID

- Reset the ‘vid’ for this widget to the specified value

#$#win-set-vattrs id: STR widget: STR ...

- Set the specified attributes of this video widget:
- brightness: NUM
- contrast: NUM
  - Image brightness & contrast
- color: BOOL
  - Display color or not