

Building a Scalable Interactive Video Solution using Peer Assisted Networking in Flash Builder

Michael Thornburgh

#adobemax104

#adobemax82

#adobemax375

Kevin Towes



Instructors

Michael Thornburgh

Kevin Towes

TAs

Jody Bleyle

Matthew Kaufman

Jens Loeffler

Sarge Sargent

Kevin Streeter

- Uses NetConnection class like RTMP
- UDP vs TCP
- Partial Reliability
- Strong Encryption
- Peer-to-Peer in addition to Client-Server

Exercise 1

Make an RTMFP NetConnection to a local RTMFP server

- Adobe's hosted RTMFP rendezvous service
- Looks like FMS, but...
 - No client-server streaming
 - No recording
 - No server-side scripts or application logic
 - Requires (free) developer key to connect

- Provides P2P introduction service
 - Translates Peer IDs to IP address/port number
 - Performs UDP hole-punching for P2P NAT/Firewall traversal
 - Provides P2P Group Bootstrap function

- To get a developer key
 - Go to <http://labs.adobe.com/technologies/stratus/>
 - Sign up for a Stratus beta developer key
 - Sign in with your Adobe ID
 - Read and agree to the terms of service
 - Your developer key is displayed!

- No Internet access in the lab
 - Developer key for lab: **MAX2009-04893b3a8a06**
 - Case sensitive!
 - Key only works in this lab, won't work with public Stratus service

Exercise 2

Using a developer key to successfully connect

- Self-organizing peer-to-peer mesh
 - One or more RTMFP nodes
 - Direct or transitive connectivity between all members
 - Each member has $O(\log n)$ direct connections (neighbors)
 - Actually about $2 \log n + 13$ 😊
 - New members join (bootstrap)
 - Connect to at least one other already-joined member
 - Stratus can do this automatically if requested

- Described by a Group Specification (“groupspec”)
 - Names the Group
 - Enables capabilities
 - Multicast
 - Posting
 - Directed routing
 - Object replication
 - Server channel
 - LAN peer discovery using IP multicast
- Every member agrees on the name and capabilities of the group
 - If you add or subtract capabilities, that’s OK, it’s just a different group!

- Group Specifications are long opaque strings
 - Start with “G:” followed by hex digits
 - Hard to construct manually
- GroupSpecifier class constructs groupspecs symbolically
 - Name the group
 - Properties to enable/disable capabilities of the group
 - `multicastEnabled: Boolean` property
 - `postingEnabled: Boolean` property
 - `serverChannelEnabled: Boolean` property
 - When connected to Stratus, enabling this will make Stratus do auto-bootstrapping for you
 - Generate the Group Specification
 - `groupspecWithAuthorizations(): String` method

Exercise 3

Use the GroupSpecifier class to specify a Group

- Flash Player 10.0 introduced RTMFP end-to-end API
 - Second parameter of NetStream constructor is “peerID”
- Flash Player 10.1 extends this API for RTMFP P2P Groups
 - Accepts a Group Specification string
 - Group NetStreams used for multicast (P2P and/or IP multicast)
- New in Flash Player 10.1: NetGroup class for other P2P modes
 - Posting, directed routing, object replication
- New NetStatusEvent codes for groups

Exercise 4

Construct a NetGroup and Group NetStream

Permission dialog if a NetStream or NetGroup might use a computer's upload bandwidth in a P2P group

- Similar to Camera/Microphone permission dialog
- Pops up when new NetStream or NetGroup is constructed
- Must be accepted before NetStream/NetGroup will function
- Allow
 - `NetStream.Connect.Success`
 - `NetGroup.Connect.Success`

- **play()** or **publish()** on a Group NetStream
 - Just like client-server
 - After **NetStream.Connect.Success**
- Streams flow through P2P mesh instead of from/to server
- Multiple streams per group
- Multiple groups per NetConnection

Exercise 5

Subscribe to a video broadcast

- Similar to Multicast
 - For short, isolated messages instead of streaming media
 - Many-to-many case
 - vs Multicast for “one- or a-few-to-many”
- **`NetGroup.post(message:Object):String`**
 - Returns messageId, a hash of the message
- **`NetGroup.Posting.Notify` NetStatusEvent**
 - **`event.info.message:Object`**
 - **`event.info.messageID:String`**

Exercise 6

Using Posting to build a Chat

Like `NetStream.play()`, `NetStream.publish()` works in a P2P Group `NetStream`.

The “publisher” application in Flash Builder is already complete for publishing.

For reference, the lab handout lists the differences between the viewer and publisher applications.

- Randomize group name
- Attach camera and microphone
- Publish
- Detach camera on shutdown
- Minor modifications to UI

Q&A



CONNECT. DISCOVER. INSPIRE.