



MAX

Building P2P Multiplayer Games

or what I've learned during MAX Racer development

Tom Krcha | Platform Evangelist

FlashRealtime.com | @tomkrcha



WHAT, P2P?

NAT/Firewall Traversal

NetGroup

Native Multicast

NetGroupSendMode

Fusion

NetGroupSendResult

NetGroupReplicationStrategy

Encryption

RTMFP

IP Address Mobility

NetStream

Directed Routing

IT'S HUGE!

Posting

Application-Level-Multicast

Bootstrap

GroupSpecifier

Topology

NetStreamMulticastInfo

Object Replication

Partial Reliability

NetGroupInfo

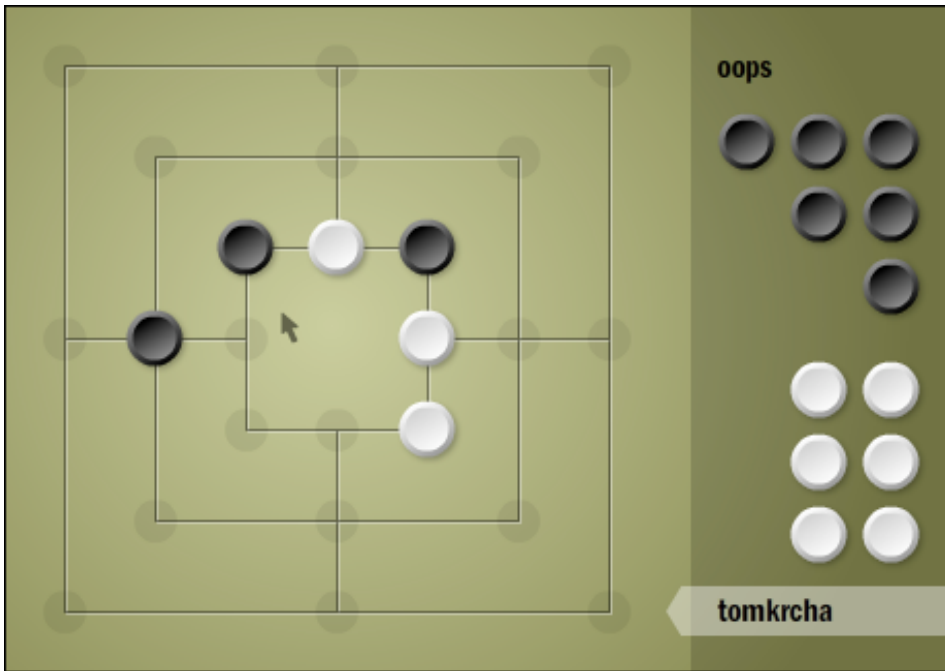
NetGroupReceiveMode

NetConnection

Congestion Control

The most simple P2P Game

Demo game - MILL



Author

Pavel Šimek, Geewa
Czech Republic

<http://nestor.cz/mill/>

RTMFP - the cornerstone of P2P in Flash

- ▶ Real Time Media Flow Protocol
- ▶ Introduced in Flash Player 10 and upgraded in FP 10.1
- ▶ Based on UDP (lossy, better latency)
- ▶ Encrypted 128-bit AES
- ▶ Need to accept every incoming connection

USE CASES

- ▶ Multiplayer games,
- ▶ VoIP
- ▶ Audio/Video
- ▶ Collaboration
- ▶ Chat

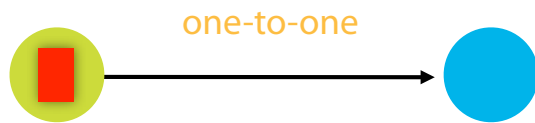
Firewall fallback to RTMP/T (Flash Media Server)

What is P2P?

- Point-to-point

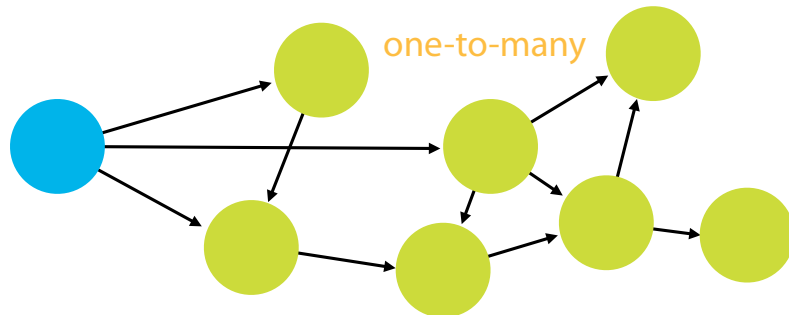
- Live streaming
- Document delivery

Direct NetStream



- Live Application-Level Multicast

- Broadcast (1 to many, some latency tolerable)
- Interactive (many to many, or 1 to many with feedback, low latency required)

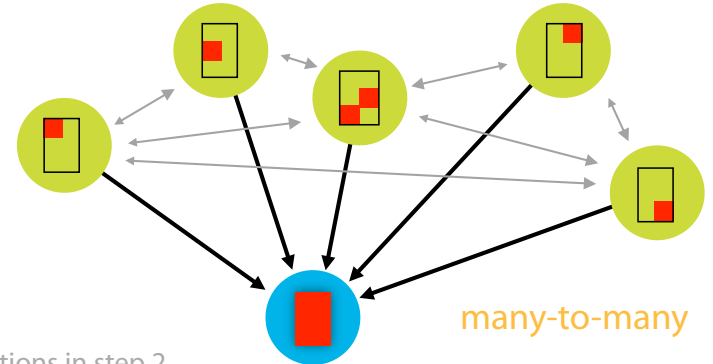


Multicast

- "Swarming"

- Large-file download (possibly progressive)

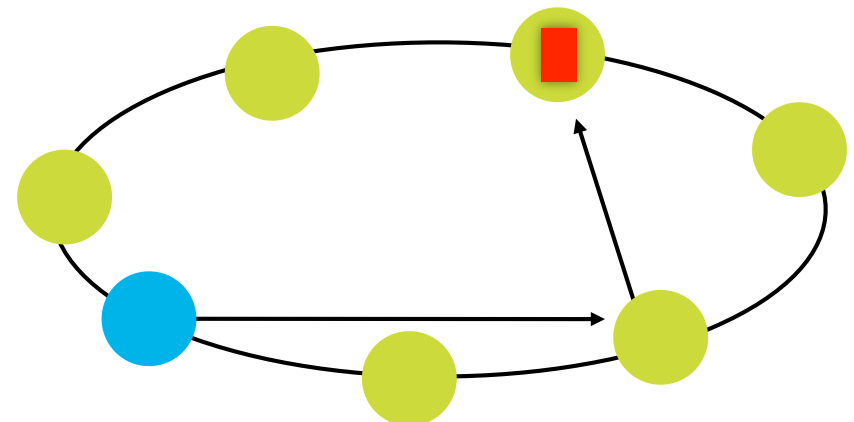
Object Replication



*Connections in step 2

- Distributed Data Storage (distributed hash table)

- DHT-like structures to form distributed database



Directed Routing

P2P API in Flash Player

- GroupSpecifier
- NetGroup
 - NetGroupReceiveMode
 - NetGroupReplicationStrategy
 - NetGroupSendMode
 - NetGroupSendResult
- NetStream (updated)
 - NetStreamMulticastInfo

Data messages in P2P Direct Connections
are always delivered

Posting, Multicast, Directed Routing are the best effort delivery.

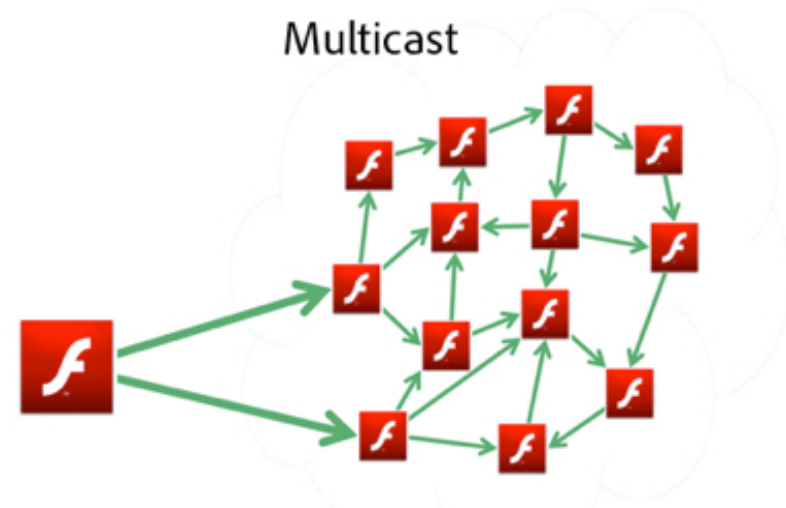
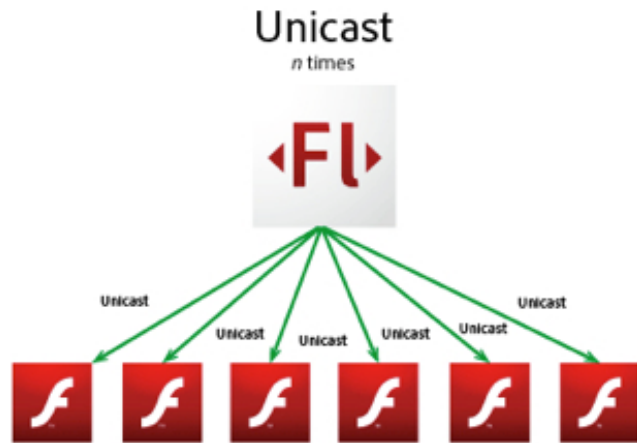
= Not fully reliable. Don't count on it if you need 100%.

Recap

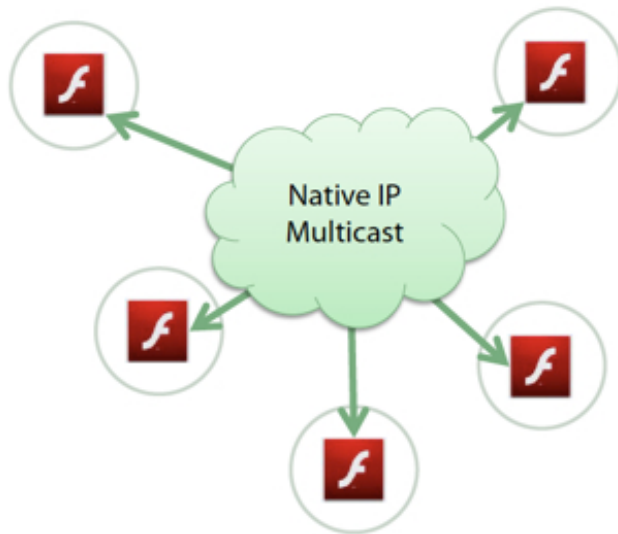
- Flash Player 10
 - Direct NetStream (DIRECT_CONNECTIONS)
 - fully reliable for data
 - best-effort for audio and video
- Flash Player 10.1
 - best-effort delivery:
 - Posting
 - Multicast
 - Directed Routing
 - fully reliable:
 - Object Replication (fully reliable)

So what to use for gaming?

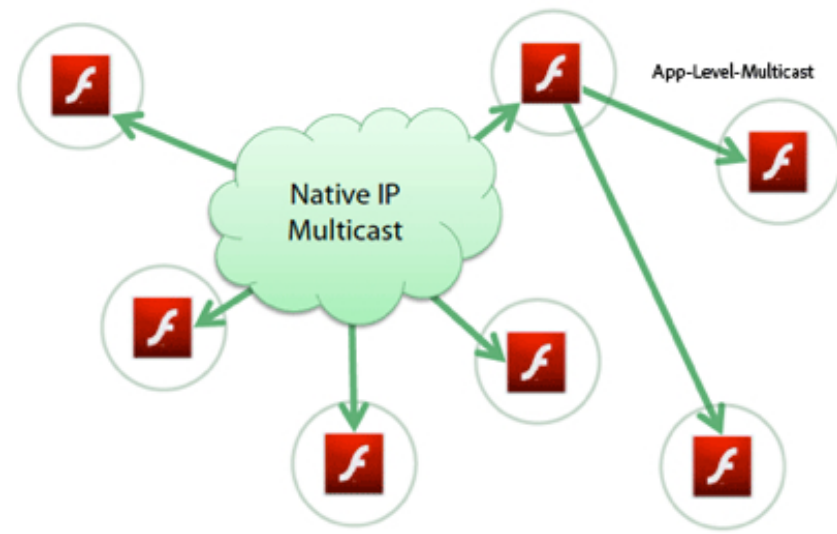
Multicast - What is it?



IP-only Multicast



Multicast Fusion



Problem with multicast

low latency -> redistribution

Video: multicast realtime problem

Lower multicast latency/jitter/delay

NetStream.multicastWindowDuration

Problems with Posting

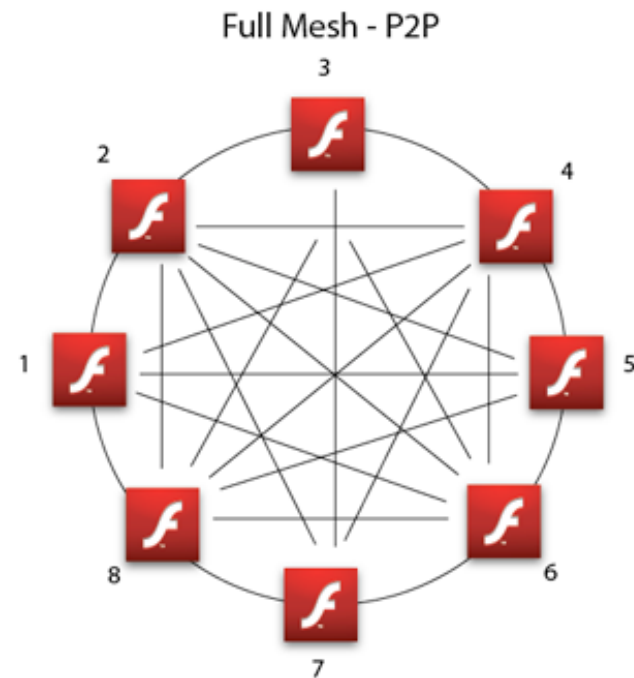
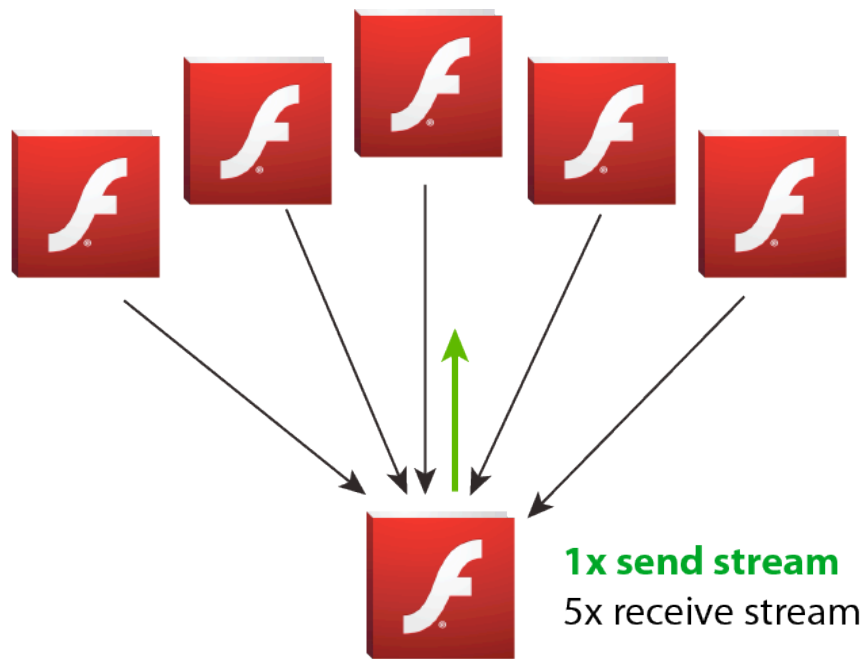
Low latency

Non-guaranteed order

A, B can be received in B, A order

P2P Games

- for absolute realtime you need to use DIRECT_CONNECTIONS
- why?



P2PGameLibrary

Tip: Compile and run in Flash Player

Tip: Run two or more windows

Tip: Have Snake.as and RemoteSnake.as
or Car.as and RemoteCar.as

Start with Roster

sync users before start

Sending direction changes? Not accurate!

1ms difference is already a problem

Demo snake directions only

Difference, when sending directions-only

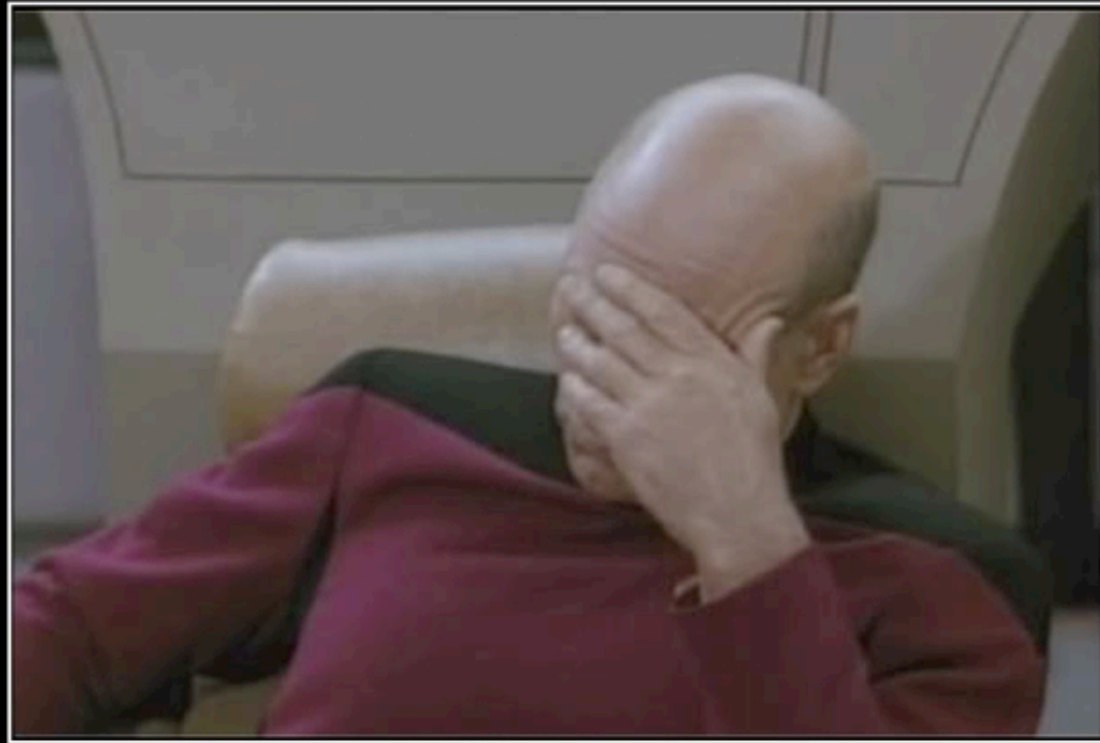
The screenshot shows a game window titled "/Users/tomkrcha/Documents/Adobe Assets/_DEV/SnakeGame/bin-debug/MainP2P.swf". The game area displays a snake's path in blue, which is highly convoluted and self-intersecting. A single brown line represents the snake's current direction. The console on the right shows a series of log messages, including "4649: new RealtimeChannel:" followed by several lines of hexadecimal data, and "4765: receiveStream: NetStream.Play.Reset". At the bottom right, there is a "Reset Game" button. The console also shows "user999 *user999" and "user900, user900" at the bottom left.

Sending POSITION changes?
Well, it's not fluent.
Choppy!

Depends on the accuracy.

Demo snake positions

Demo racing positions only



FACEPALM

Because expressing how dumb that was in words just doesn't work.

Solution?

Both?
Directions + Position (to fine tune)?
Better! But...

Demo racing positions+directions

Big plus = no need to send every position
change = less data



DOUBLE FACEPALM

FOR WHEN ONE FACEPALM DOESN'T CUT IT

DIY.DESPAIR.COM

Solution?

Forget precise movements.
Interpolations is a go!

SharedMovements - smoothing

Smoothing

```
protected function onEnterFrame(event:Event):void{  
  
    // EXACT  
    x = destX;  
    y = destY;  
  
    /// OR  
  
    // SMOOTHING  
    x -= (x-destX)*0.2;  
    y -= (y-destY)*0.2;  
}
```

Smoothing + Threshold

Demo Racing Interpolations

Demo MAX Racer: video 3 PCs

Start? Timing.

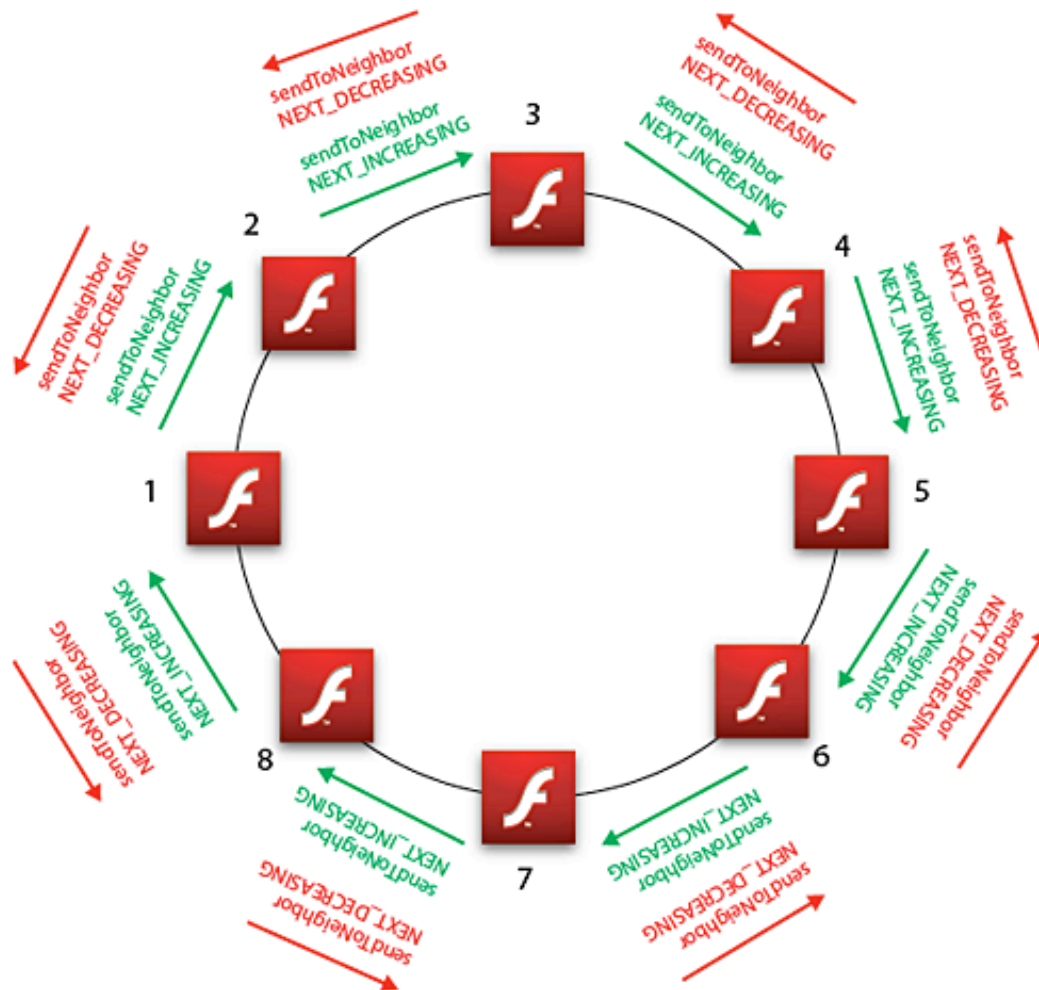
If your game works for 2 simultaneous people,
doesn't mean it works for 3, 10 or 1000.

Test in iterations.

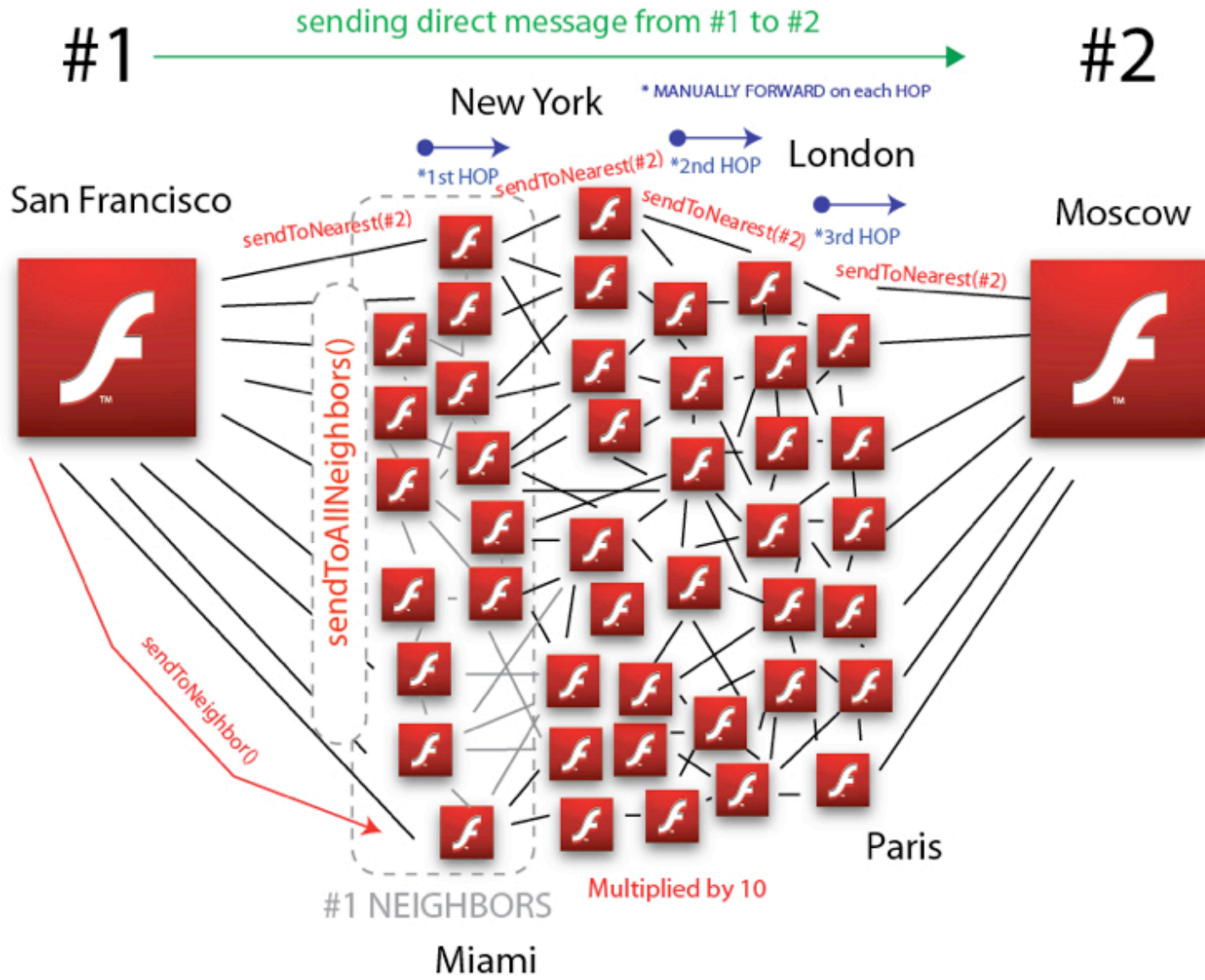
Maintaining users:
userAdded
userRemoved
userUpdated
userIdle

Directed Routing RING

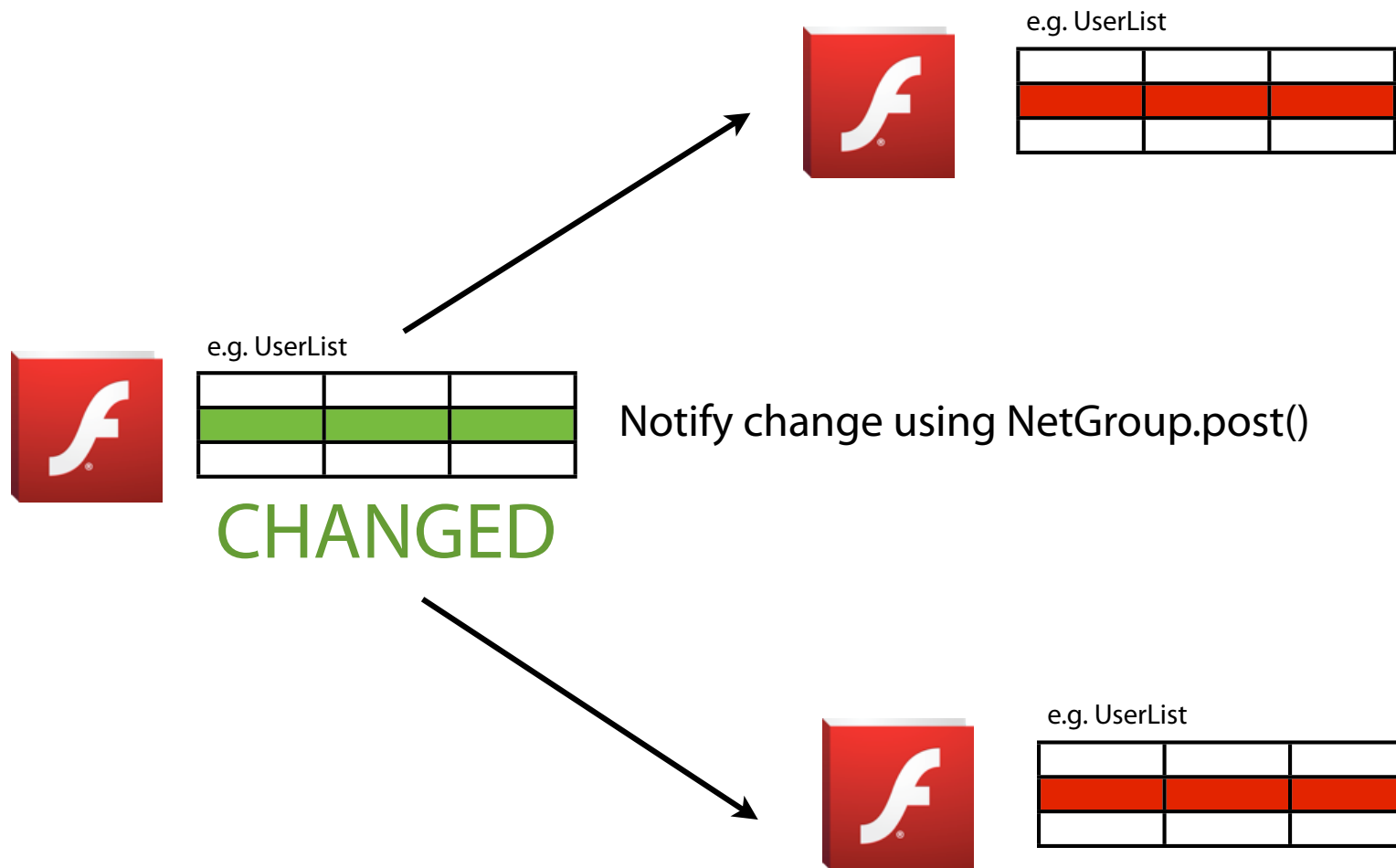
Ring topology with sendToNeighbor



Directed Routing overview



A prior to Distributed HashTables

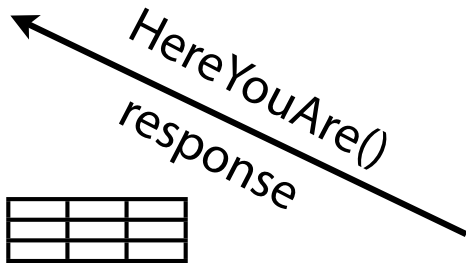


A prior to Distributed HashTables

New
comer

Nearest
Peer

Another
Peer



e.g. UserList

e.g. UserList


```
NetGroup.sendToNeighbor(ADDRESS);
```

Directed Routing send

- *Sending message*

```
var message:Object = new Object();  
message.destination = netGroup.convertPeerIDToGroupAddress(peer2ID);  
message.value = "Hello I am message from #1";
```

```
netGroup.sendToNearest(msg, msg.dest);
```

- *Receiving message*

```
netGroup.addEventListener(NetStatusEvent.NET_STATUS, netStatus);
```

```
function netStatus(event:NetStatusEvent):void{  
    switch(e.info.code){  
        case "NetGroup.SendTo.Notify":  
            // e.info.message contains our message Object  
            trace("Received Message: "+e.info.message.value);  
            break;  
    }  
}
```

- *Forwarding message*

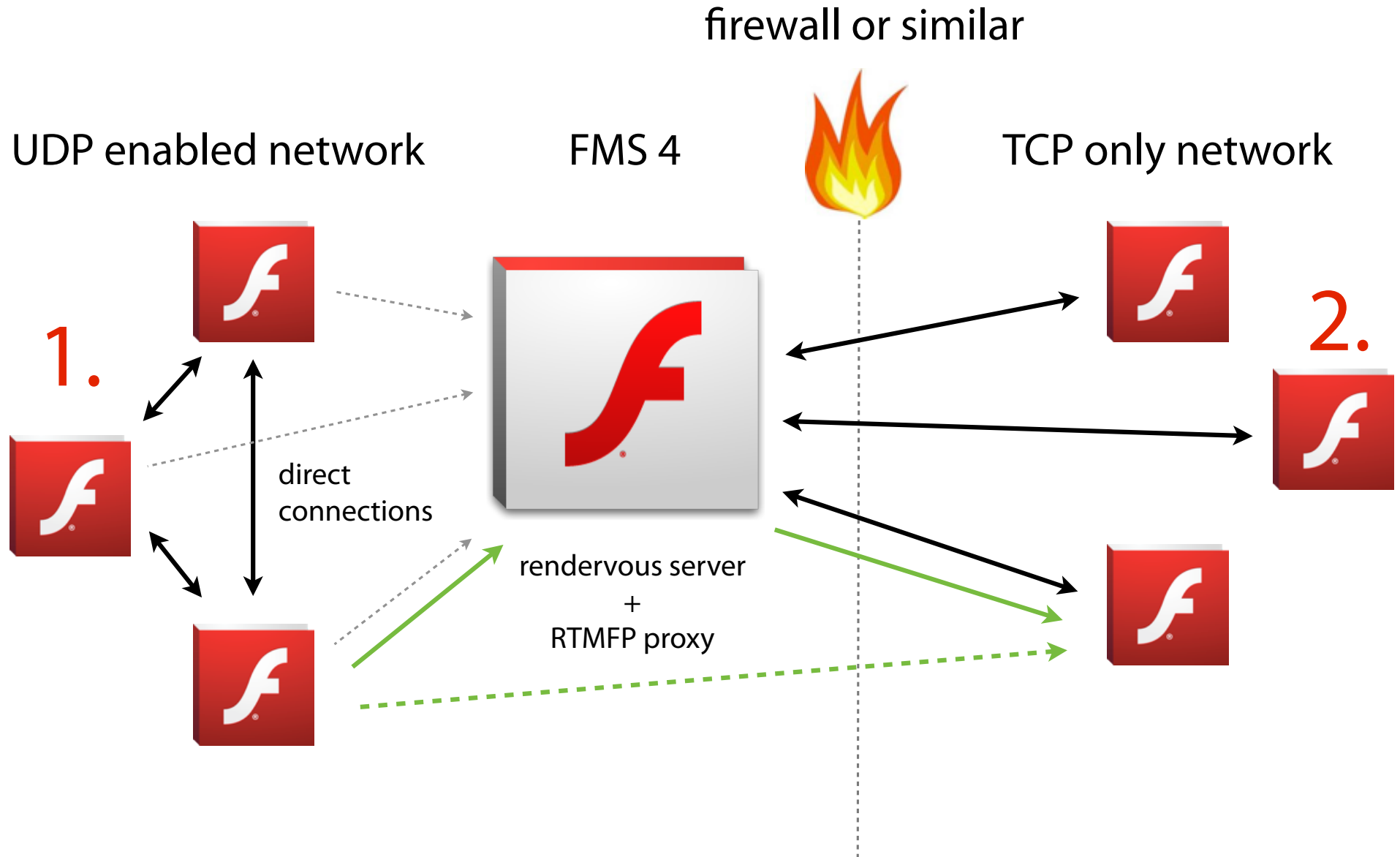
```
netGroup.addEventListener(NetStatusEvent.NET_STATUS, netStatus);
```

```
function netStatus(event:NetStatusEvent):void{  
    switch(e.info.code){  
        case "NetGroup.SendTo.Notify":  
            if(e.info.fromLocal == true){  
                // We have reached final destination  
                trace("Received Message: "+e.info.message.value);  
            }else{  
                // Forwarding  
                netGroup.sendToNearest(e.info.message, e.info.message.destination);  
            }  
            break;  
    }  
}
```

Explain P2PMessengerLib

Solving Failover Problems

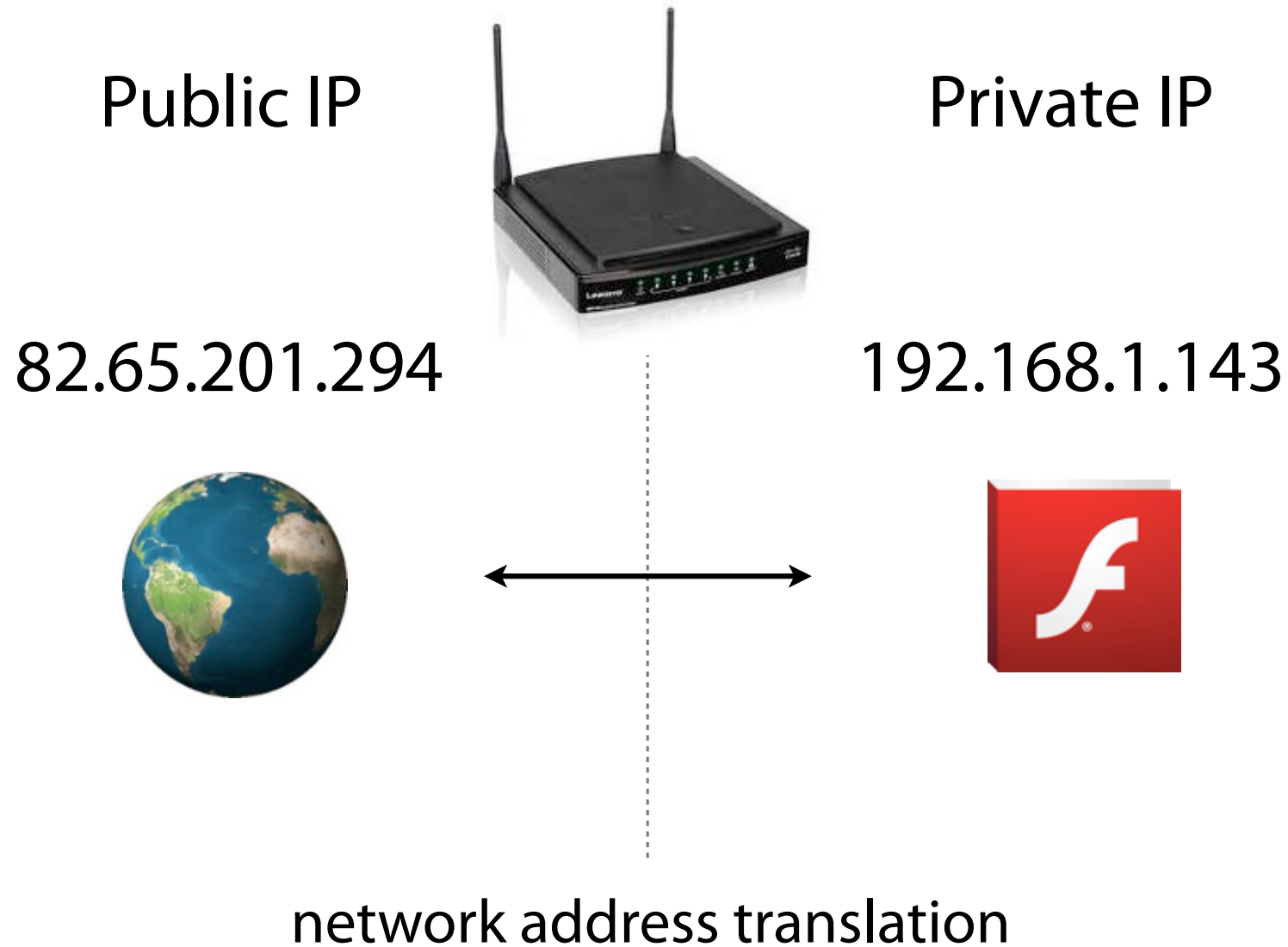
Flash Media Server 4 failover



FMS 4 P2P Server-Side JavaScript API

- NetGroup.addMemberHint()
- NetGroup.addNeighbor()
- NetGroup.close()
- NetGroup.convertPeerIDToGroupAddress()
- NetGroup.estimatedMemberCount
- NetGroup.info
- NetGroup.localCoverageTo
- NetGroup.localCoverageFrom
- NetGroup.onStatus()
- NetGroup.post()
- NetGroup.receiveMode
- NetGroup.sendToAllNeighbors()
- NetGroup.sendToNearest()
- NetGroup.sendToNeighbor()
and so on
- GroupSpecifier.addBootstrapPeer()
- GroupSpecifier.addIPMulticastAddress()
- GroupSpecifier.authorizations()
- GroupSpecifier.encodeBootstrapPeerIDSpec()
- GroupSpecifier.encodeIPMulticastAddressSpec()
- GroupSpecifier.encodePostingAuthorization()
- GroupSpecifier.encodePublishAuthorization()
- GroupSpecifier.groupSpecWithAuthorizations()
- GroupSpecifier.groupSpecWithoutAuthorizations()
- GroupSpecifier.ipMulticastMemberUpdatesEnabled
- GroupSpecifier.makeUnique()
- GroupSpecifier.multicastEnabled
- GroupSpecifier.objectReplicationEnabled
- GroupSpecifier.peerToPeerDisabled
- GroupSpecifier.postingEnabled
- GroupSpecifier.routingEnabled
- GroupSpecifier.serverChannelEnabled
- GroupSpecifier.setPostingPassword()
- GroupSpecifier.setPublishPassword()
- GroupSpecifier.toString()

NAT Traversal - it works!



Check your connection!



cc.rtmfp.net

3D game in Flash

OTHER P2P TIPS, TRICKS and FAQ



P2P works across devices
Win, Mac, Linux, Android, iOS

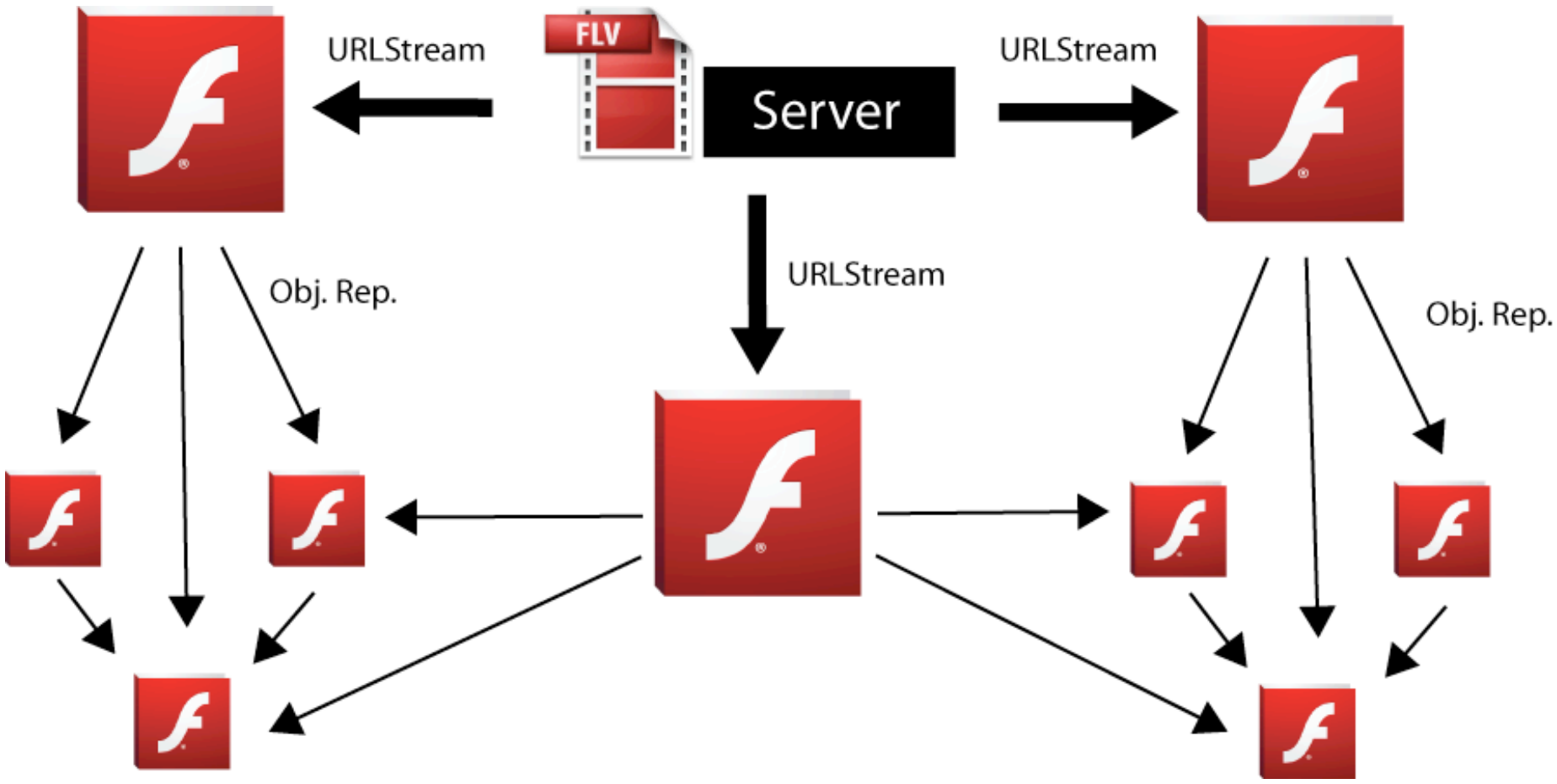


All P2P group operations work in LAN, except
DIRECT_CONNECTIONS via NetStream

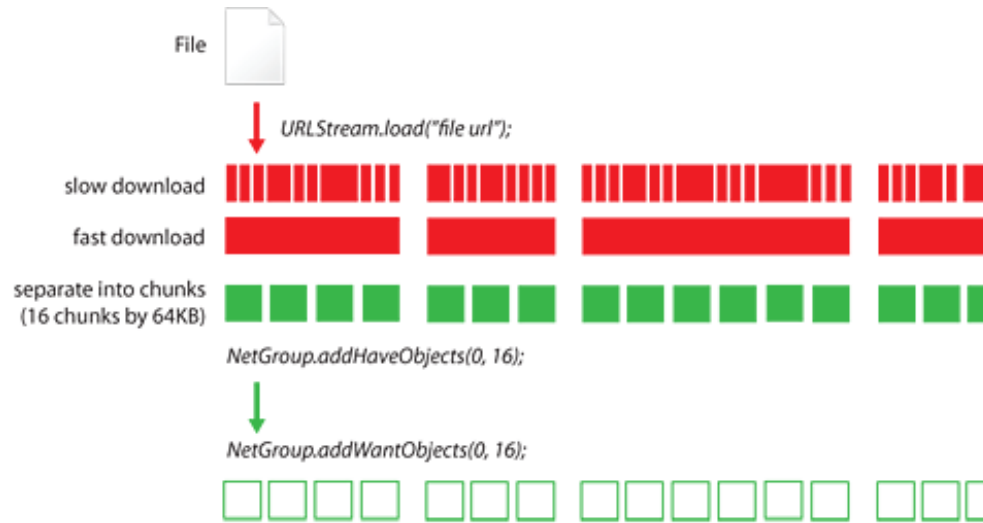
Demo Local P2P Chat

Demo: File Sharing

Video-on-Demand with Object Replication



Video-on-Demand with Object Replication



- Steps:

- download
- separate into chunks (64 KB)
- distribute

- Challenges:


- Download buffer for viewing (`NetStream.appendBytes`)
- Postponed buffer for P2P (mixed order of chunks)
- Deciding who is provider (choosing "machos")

Video-on-Demand with Object Replication

Sender

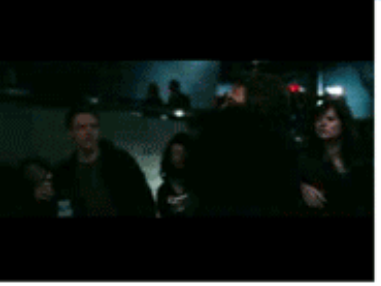
browse *.flv

up: 74
up: 75
up: 76
up: 77
up: 78
up: 79
up: 80
up: 81
up: 82
up: 83
up: 84







Receiver

down: 77
down: 78
down: 79
down: 80
down: 81
down: 82
down: 83
down: 84
fileShareComplete
onMetaData.duration: 149.149
stream: NetStream.Buffer.Full



Video-on-Demand with Object Replication

Start download (URL) or	or Start receiving (P2P)	or Start receiving (P2P)	or Start receiving (P2P)
<pre>up: 29 up: 30 up: 31 up: 32 up: 28 up: 29 up: 30 up: 31 stream: NetStream.Buffer.Empty</pre>	<pre>up: 24 up: 25 up: 26 up: 27 up: 28 up: 29 up: 30 up: 31 up: 32</pre>	<pre>down: 30 apnd: 30 down: 31 apnd: 31 down: 32 apnd: 32 lastChunkAppended - bufferTimer.stop() onMetaData.duration: 24.958 stream: NetStream.Buffer.Full</pre>	<pre>down: 31 onMetaData.duration: 24.958 stream: NetStream.Buffer.Full apnd: 28 apnd: 29 apnd: 30 apnd: 31 apnd: 32 lastChunkAppended - bufferTimer.stop()</pre>
 <p>Johnny Ensom and the Bonkato "Used Photoshop Today" Short and Snappy Wild Reg-Sized Crawford Mack</p>			

Demo: VoD over P2P

Libs

- AS3-P2P-LIB
- P2P Messenger Lib
- P2P Game Engine

Bonus: Hooking up game controllers

STEP 1

List HID devices available



STEP 2

Choose device



STEP 3

listen to update

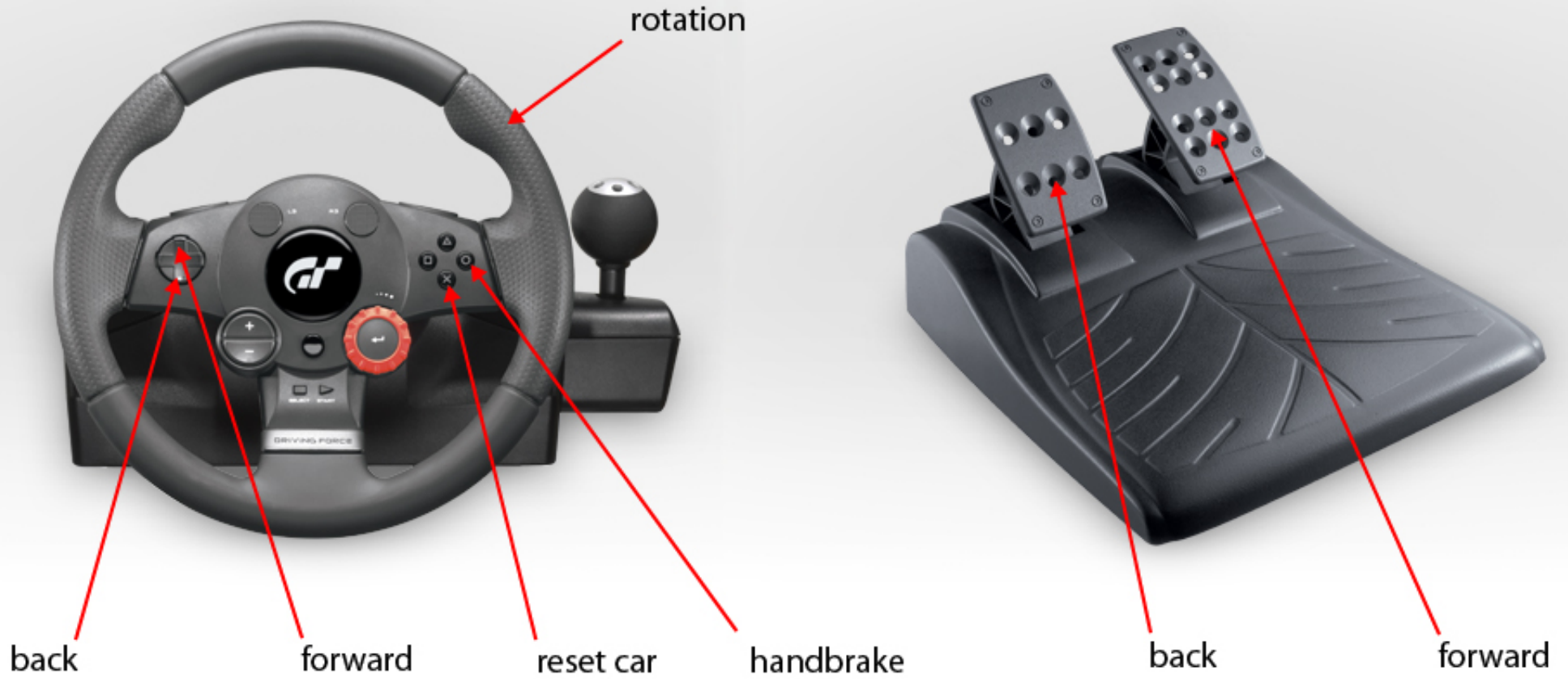
wheel rotation

handbrake

reset car

and so on...

Driving controls for MAX Racer



- R - reset
- Tab - stats
- Arrow keys - control the car
- Enter - enter free view mode
- W,S,A,D - control free view mode



Survey

MAX

CONNECT. DISCOVER. INSPIRE.

Thanks! Q/A

FlashRealtime.com

[@tomkrcha](https://twitter.com/tomkrcha)



MAX

CONNECT. DISCOVER. INSPIRE.

Q/A

