Using the Data Channel

Session: D1-5
Richard Tworek
Principal
WebRTC Strategies
rtworek@webrtcstrategies.com
Session Presenters

• Jan-Ivar Bruaroey
  – Sr. Software Engineer
  – Mozilla

• Dan Ristic
  – Developer Evangelist
  – PubNub
Jan-Ivar Bruaroey
Sr. Software Engineer
jib@mozilla.com
The Data Channel

What is it?
What does it provide?
Is it fast / reliable / secure?
When is it useful?
What does the API look like?
How do I use it?
What existing applications are there?
When will it be available?
The Data Channel: What is it?

It's a part of HTML5 and WebRTC
It lets web apps communicate directly with other instances over the internet
It takes web development beyond the server-client to client-client

(For those familiar with WebRTC:)

It takes WebRTC beyond VoIP to sending any form of data
Like everything else, still being ratified in the WebRTC Standard at IETF & W3C

http://dev.w3.org/2011/webrtc/editor/webrtc.html
The Data Channel: What does it provide?

- Provides the means for web apps to communicate directly without the data going through a server
- Uses the existing WebRTC PeerConnection mechanism (Offer/Answer/Ice)
- Provides a reliable bi-directional encrypted connection over UDP
- Open multiple DataChannels over the same connection (multiplexing)
- Provides a JavaScript API modeled on WebSockets (Talk to peers the same way you talk to servers today)
The Data Channel: Fast, Reliable, Secure

Peer-to-peer data transmission gives you low latency between clients (skipping server) - Throughput is limited only by upload speed

Persistent connection. Open and close new data channels on a whim

DataChannel is a thin protocol on-top of mature and secure standards:
Uses SCTP over DTLS (encrypted UDP), giving you Reliable (TCP-like), Unreliable (UDP-like) and partially-reliable transmission with congestion control *

Leverages existing browser security: JavaScript sandbox. Cross-origin restrictions. Same privacy footprint as existing tech (WebSockets, HTTP)

Later: Support for identity-providers (like Mozilla Persona)
The Data Channel: When is it useful?

When low latency between users of your app is critical:
- E.g. Realtime collaborative apps, games, calls, interactive shared experiences
- Client-initiated requests, client-driven information and file sharing.
  (Remember you can send data to server as well for storage with WebSockets)

You can use it to transmit metadata etc. during VoIP calls. For example:
SDP for renegotiation of calls can travel over a DataChannel within the call itself

Multi-user experiences of 10+ people can work using a mesh

Can offload bandwidth from servers having to funnel things from A to B (and C)

Distributed processing networks (improve resilience, reduce server load)
The Data Channel: The JavaScript API

// Creating a basic data channel on a PeerConnection

var channel = pc.createDataChannel("my udp", { ordered: false, maxRetransmits: 0 });

interface RTCDataChannel : EventTarget {
    attribute EventHandler onopen;
    attribute EventHandler onclose;
    attribute EventHandler onerror;
    attribute EventHandler onmessage;
    attribute DOMString binaryType;
    void close();
    void send(DOMString data);
    void send(Blob data);
    void send(ArrayBuffer data);
    void send(ArrayBufferView data);
};

dictionary RTCDataChannellInit {
    boolean ordered = true;
    unsigned short? maxRetransmitTime = null;
    unsigned short? maxRetransmits = null;
    DOMString protocol = "";
    boolean negotiated = true;
    unsigned short? id = null;
};
The Data Channel: How do I use it?

// Alice wants to IM with Bob

var pc = new RTCPeerConnection();
var channel = pc.createDataChannel(
"chatting", {protocol: "text/plain"});

channel.onopen = function () {
    channel.send("Hi Bob!");
};
channel.onmessage = function (event) {
    console.log("Bob says: " + event.data);
};

// Insert regular WebRTC call to Bob here ...

document.forms[0].onsubmit = function () {
    channel.send (document.forms[0][0].value);
}
The Data Channel: How do I use it?

// Alice wants to IM with Bob

var pc = new RTCPeerConnection();
var channel = pc.createDataChannel(  
  "chatting", {protocol: "text/plain"});

channel.onopen = function () {  
  channel.send("Hi Bob!");
};
channel.onmessage = function (event) {  
  console.log("Bob says: " + event.data);
};

// Insert regular WebRTC call to Bob here ...

document.forms[0].onsubmit = function() {
  channel.send(document.forms[0][0].value);
}

// Bob answers

var pc = new RTCPeerConnection();

pc.ondatachannel = function (event) {
  var channel = event.channel;
  channel.onopen = function () {
    channel.send("Hi Alice!");
  };
  channel.onmessage = function (e) {
    console.log("Alice says: " + e.data);
  };

  // Insert regular WebRTC answer to Alice

  doc.forms[0].onsubmit = function() {
    channel.send(doc.forms[0][0].value);
  }
}
The Data Channel: Existing applications

Live example (chat): webrtc-experiment.appspot.com/DataChannel

Games:

BananaBread (3D multiplayer first-person shooter)  
developer.mozilla.org/en-US/demos/detail/bananabread

Cube-Slam (3D Video pong)  www.cubeslam.com

Drag'n'drop file-sharing: www.sharefest.me

Distributed content distribution network: peercdn.com
The Data Channel: When will it be available?

Very near future:

Firefox 22 (June 2013)
Chrome 29 (August? 2013)
Firefox for Android (Fall 2013)
Firefox OS (Fall 2013)

Now (developers):

Firefox Beta/Aurora/Nightly
Chrome Canary (older DataChannel API - in process of converting)
PubNub

Dan Ristic (Twitter: @dristic101)
Developer Evangelist
PubNub
danr@pubnub.com
PubNub Real-Time Network

PubNub **Galaxy** *(Multicast)*

PubNub **Pulse** *(Unicast)*

- Presence
- Analytics
- Storage/Playback
- Security
- Mobile Push
- Multiplexing
Why it Matters

• P2P is a big part of real-time networks
• It helps our users build and scale real-time applications
What is it?

High performance and low latency communication of arbitrary data

```javascript
var peerConnection = new RTCPeerConnection();
var dataChannel = peerConnection.createDataChannel("mylabel");
dataChannel.onmessage = function(event) {
    console.log("I got a message", event.data);
}
dataChannel.send("Hello World!");
```
What do I Expect?

CC Image by Tallapragada on Flickr
What do I Get?

• Well defined standards
• Generalized solution targeted for all use cases
• One of the highest speed connections you can get in the browser
• Still highly configurable

“My recommendation then is not only that you use UDP, but that you only use UDP.”

Glenn Fiedler on UDP vs. TCP
http://gafferongames.com/networking-for-game-programmers/udp-vs-tcp/
What are the Issues?

• Still in development
  – Only supported by Firefox and Chrome
  – Limits to API since it is still being worked on
• Lots of race conditions
  – Syncing ICE connections / SDP packets
  – Lots of places to fail and for different reasons
• Not the Magic Kingdom
  – Only one piece of a larger puzzle
  – Still need broadcast, presence, logging, analytics, and more to deliver a full experience
• Security concerns
  – What data do you trust users with?
Changing the Future

From This...

Server

Client

...To This

Server

Client

Client

Client
Great Examples

PeerCDN
https://peercdn.com/

Meet PeerCDN.
A new kind of CDN.

PeerCDN is a peer-to-peer distributed CDN that will make the web faster, more reliable, and help sites to reduce bandwidth costs.

Get on the mailing list.
you@example.com Subscribe
The latest PeerCDN updates and nothing else.

1 Reduce server and bandwidth costs.
PeerCDN automatically serves a site’s static resources (images, videos, and file downloads) over a peer-to-peer network made up of its visitors currently on the

Sharefest
http://www.sharefest.me/

It’s still really early days for WebRTC, but we hope to have PeerCDN serving the above screencast in a week or two. Check back soon!

Cobb Galleria, Atlanta Georgia
June 25-27, 2013
@webrtcexpo/#webrtcexpo
Questions

1. What are the most important things to know about the data channel?
2. What can you do with the data channel that you cannot do today?
3. We have heard about server less computing, does the data channel really enable this?
4. What are the security issues the data channel opens?
5. How fast is the data channel?
6. The data channel is UDP, how do we guarantee delivery?