OpenTok™: A Free Open Source API for Video Conferencing in Distance Education

Jerald D. Cole

University of Bridgeport
Department of Instructional Technology

Abstract

OpenTok™ is a free open source software application that supports synchronous online instructional delivery via video/audio conferencing. When used in conjunction with a remote screen sharing solution, it is possible to realize a virtual experience commensurate with (if not superior to) that of a traditional “ground-based” classroom.

This paper covers the advantages of synchronous delivery over asynchronous approaches and the presentation layer of OpenTok™. The functionality of several API calls are then deconstructed in the context of a standard HTML file exemplar that instructors can appropriate for use in their own courses. Finally, we demonstrate and evaluate the application in the context of an actual semester course in instructional technology.

Synchronous versus Asynchronous Delivery

Online education offers students the convenience of instruction at a distance, the modalities of distance being time, pace and place. Unfortunately, the majority of online courses are conducted via asynchronous delivery, using learning management systems (LMSs) such as Blackboard™ or Moodle that lack the gestalt of a live, face-to-face classroom experience. The bandwidth requirements for asynchronous text-based communications needed to support LMSs are certainly lower—the opportunity cost being a loss of the sense of teacher presence. When it comes to presence, the proverb “out-of-sight; out-of-mind” seems apropos.

Three issues have limited the adoption of synchronous conferencing solutions in education:

(1) Many instructors are indoctrinated into prevailing models of asynchronous instruction, such as threaded discussion and email. They are not acculturated to and/or aware of the alternative.¹
(2) Channel capacity has proven to be a deterrent given the high cost of upgrading aging university network infrastructures to levels capable of supporting the traffic imposed by real-time dynamic media.²
(3) The majority of synchronous conferencing solutions are proprietary, with high subscription fees.³

Ironically, the current generation of students are digital natives and have grown-up acclimated to synchronous communications.⁴ They have high expectations as to the availability of these channels and use in instruction. Surprisingly, a growing number of college freshman have never sent an email, and consider it superfluous to instant messaging.⁵ Thus, it is faculty who need training to overcome the limitations given in issue (1), above. They are caught in the infamous DEC Gap Brown⁶ warns us of—locked into a snapshot of teaching technologies current at the time of their training, but anachronistic to the students of today.

By Christensen’s conservative estimate, over half of instruction at all levels (K-16) will be augmented by or conducted entirely online by 2019.⁷ The advent of ubiquitous consumer broadband access has largely eliminated constraint (2) by shifting the network burden onto the population of “connected” home-based students and teachers.

With the release of the OpenTok™ Application Programmer Interface (API), a multipoint video conferencing solution, the tables were turned on issue (3).
What is OpenTok™?

OpenTok™ is an open source Application Programmer Interface (API) released by TokBox, a proprietary online multipoint video conferencing provider. In January 2011 TokBox announced their intent to transition from a subscription conference hosting provider to a business model centered on the promotion and support of OpenTok™.

The complete set of documentation, tutorials, downloadable client APIs and server daemons are available at http://www.tokbox.com/opentok.

OpenTok™ supports up to 20 video/audio streams, allowing both the teacher and learner to see and hear each other in one concurrent session using consumer-grade broadband connections, such as school or home or wherever they may be situated.

Being free of charge, open source, cross-platform, server-sided with sophisticated load balancing, and presenting with a browser-based user interface (no downloading and installation of software needed), OpenTok™ video conferencing and its derivatives promise to greatly increase the use of synchronous conferencing in education.

The 20 simultaneous video/audio sessions limit currently imposed by OpenTok™ is twice the capacity of Skype (10 concurrent sessions at $8.50/month per participant), and industry-leading WebEx and GoToMeeting™—both of which support up to 6 feeds at $49.00/month. Six simultaneous feeds are not enough to involve an entire class; twenty is adequate, given that the average enrollment in project-based courses is 15 students. OpenTok™ is clearly the industry standard-bearer with zero licensing costs and superior performance.

OpenTok™ Interface

The OpenTok™ interface, depicted in Figure 1, is intuitive. It consists of a video presentation area accommodating up to 20 simultaneous sessions with full-duplex audio per session, a control area, and a text chat area.

Figure 1: OpenTok™ Interface.

When more than 20 participants join a session, they are relegated to “Voice only” frames, as shown in Figure 2. All participants see the same session screen and can participate in multipoint audio and text chats.

Figure 2: Concurrent sessions with more than twenty participants.
Menu options provide for session invites, sharing content on social networking sites, configuring I/O devices, and toggling full screen view as shown in Figure 3. Participants may be invited into a session via a contact list, email, Chat link, or posts on Facebook, Twitter and MySpace. It is also possible for participants to share-out a YouTube video or picture galleries on Flickr or Picasa in a session frame.

![Menu controls for session invites, sharing, with two right-most speed buttons for I/O configuration & toggling full screen.](image)

In addition to the standard set of widgets in the control area allowing for audio/video device configuration, end-users can mouse-over their video frame and access pop-up controls to mute audio and toggle video to On, Off orPaused states as shown in Figure 4. Double-clicking on a frame enlarges it, and video frames can be rearranged via drag-and-drop. Frames fractionally and uniformly resize as users join a given session to proportionally fill the available screen real estate.

![Configuration dialogs for camera and microphone settings.](image)

Both ad hoc conferencing and scheduled sessions are supported. An ad hoc session is accessible to participants via a URL that is set-up by the organizer and can be embedded in a webpage or sent via email. This permits for recurring meetings via a simple click of the mouse. Scheduled sessions offer the option for authenticated access via a login screen, effectively keeping them closed to tuition paying customers such as students or trainees.

A backend calendaring utility allows for scheduling sessions via a landing page that can be branded with design aspects of the home institution, date and time stamped, and synchronized with Outlook or Google Calendar.

**Architecture**

The OpenTok™ API compendium consists of

1. A client-side JavaScript library that integrates audio/video streams in the context of a webpage. (An ActionScript 3 (AS3) implementation is also available.)
2. A server-side library, available in Java, PHP, Python, and Ruby, that provides daemons for session authentication, token generation, and management.
The interplay of these components is given in Figure 5. OpenTok™ uses locally installed Flash video components that interface with webcams, microphones, and audio-out channels (such as speakers or headphones) resident on the client’s workstation. These are instantiated and destroyed dynamically by the API and transparent to the end-user. The requisite Flash plug-in can be freely obtained from the Adobe site at http://www.adobe.com if it is not already installed in the client’s browser instance.

The TokBox hosted cloud server answers calls from a developer’s page, which resides on the client’s dedicated host. It is possible to install both cloud services and the developer’s page on an independent host. The client-side instantiates virtual machines to manage the local session, handle unique tokens for subscribers (participants), and pushes streams mediated by the cloud server to the browser instances of all participants. Calls to the JavaScript library and embedded JavaScripts in the page can also be used to invoke functions locally.

OpenTok™ currently offers a cloud host to serve as a free sandbox environment that developers can use to build and test applications. Access to the sandbox requires registering with a name and email address, and agreeing to their terms of service in order to obtain an API Key. TokBox Inc will also host your application on their production server when you are ready to go public, also (currently) at no cost.

The complete set of client and server-side libraries can also be downloaded at no charge and run on the developer’s dedicated host. Documentation on the installation is available at http://tokbox.com.

**Figure 5: Underlying client/server workflows of the API.**

**Sample Embedded Video**

The container code given in Figure 6 provides exemplars for the set of calls required to initiate a basic OpenTok™ session in a stand-alone page, such as demo.html. (the reader can use any valid file name as long as it includes an html extension). The statements and parameters within the <object> container in lines 10—17 define the session and can be embedded as a component in any web document.

In order to test this example using the OpenTok™ sandbox, the developer would need substitute their API Key where indicated by the yellow hi-lites. The same key can be used should the developer wish to transition their application to the OpenTok™ production server. Alternatively, the key generated by a standalone host would be used.

For institutional use, the scheduler would need to be deployed to manage separate session keys for each instructor or course in order to prevent students from showing-up concurrently in unassigned sections.
Figure 6: Server-side API calls for an embedded video & text chat.

A deconstruction of the above container code follows.

<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.</td>
<td>&lt;object type=&quot;application/x-shockwave-flash&quot;</td>
<td>Creates an envelope on the webpage for a flash component.</td>
</tr>
<tr>
<td>11.</td>
<td>data=&quot;<a href="http://me.tokbox.com/vc/Your_API_Key_Here/10">http://me.tokbox.com/vc/Your_API_Key_Here/10</a>&quot; width=&quot;1024&quot; height=&quot;768&quot;&gt;</td>
<td>Initiates a session on the TokBox cloud hosting server using an authorization code (API_Key) and limits the session to 10 participants.</td>
</tr>
<tr>
<td>12.</td>
<td>&lt;param name=&quot;movie&quot; value=&quot;http://me.tokbox.com/vc/Your_API_Key_Here/10&quot; /&gt;</td>
<td>Instantiates a local environment variable with the same name and constraints as the assigned server session.</td>
</tr>
<tr>
<td>13.</td>
<td>&lt;param name=&quot;allowFullScreen&quot; value=&quot;true&quot; /&gt;</td>
<td>Allows the session container to be maximized to full screen.</td>
</tr>
<tr>
<td>14.</td>
<td>&lt;param name=&quot;allowScriptAccess&quot; value=&quot;always&quot; /&gt;</td>
<td>Allows access the local JavaScript library and virtual machines.</td>
</tr>
<tr>
<td>15.</td>
<td>&lt;param name=&quot;flashvars&quot; value=&quot;&amp;textChat=true&quot; /&gt;</td>
<td>Enables a text chat area below the video session frame.</td>
</tr>
<tr>
<td>16.</td>
<td>&lt;/object&gt;</td>
<td>Delimits the envelope.</td>
</tr>
</tbody>
</table>

Context of Use and Evaluation

The University of Bridgeport offers an online graduate program in Instructional Technology that utilizes OpenTok™ for multipoint video conferencing, augmented by GoToMeeting™ (http://gotomeeting.com) for application screen sharing. GoToMeeting™ was selected from a pool of vendors after extensive testing and found to work best in our heterogeneous networking environment. In our evaluations Flash-based implementations of the screen sharing function are not robust and tend to bog-down networks. Note also that the 6 simultaneous video streams supported by GoToMeeting™ are deactivated as this function is handled better by OpenTok™.

Hardware Requirements

Program participants are required to have wired access to a persistent broadband connection, and provision a laptop that is less than 3 years old with a webcam and noise-cancelling headset. We also strongly recommend that they extend their monitor across two screens, as shown in Figure 7; Display 1 to maintain a persistent view of participants in the TokBox session, and park the GoToMeeting™ control panel. Display 2 is for “monkey-see-monkey-
do” presentations, and student work. All software is cross-platform (Macintosh, Windows and Linux friendly), and to as great an extent possible preferences are given to the use of open source solutions.

Figure 7: Typical class with OpenTok™ session for presence and GoToMeeting™ panel on Display 1 & shared screen on Display 2.

Feedback and Evaluation

Incidental anecdotal feedback from students regarding online courses using OpenTok™ has been enthusiastic. In keeping with national trends, most have the expectation of computer ownership from their undergraduate experience, where the majority have taken at least one course via asynchronous distance education. The synchronous online environment has all of the qualities of a face-to-face classroom experience, with none of the inconveniences (such as needing to hire baby sitters and/or driving to and from school—particularly under inclement weather conditions after a long day at work).

Formal end-of-semester course evaluations have also been positive. The playing field is leveled when all students are given equal time in the limelight and there are no wall-flowers. There is more time for interactions before and after class, and all report a more relaxed atmosphere where they can eat and drink in the comfort of their own homes, and even stand up and stretch if necessary during lecture.

From the instructor’s perspective, synchronous delivery lowers the burden of teaching online, because verbal communication is extemporaneous, while asynchronous delivery requires the advanced preparation and aggregation of all curriculum materials online. Having to read textual submissions and respond to them in written form is also far more labor intensive. It is more natural to conduct eye-to-eye conversations, where in short order the sense of geographical distance evaporates, and the screen is in effect just a piece of glass between participants.

The consensus is that via the modality of synchronous online, they are able to cover 10-15% more material per class. Instructors also concur that the level of student technology savviness has increased notably each year since the advent of the program in Fall 2004. The environment lends itself well to the constructivist pedagogical framework used in the program, which is project-based. Our approach is to rotate among participants, bringing each to the fore during video sessions while using GoToMeeting™ screen sharing for remote demonstrations and keyboard/mouse control.

Future

Factors driving students to pursue distance education include reduced income, lack of job security, and less time for traditional residential academic experiences. With the promise of increased bandwidth (including “fiber to the door”), commodification of Internet access, the triple-play of ubiquity, functionality and portability of inexpensive networked devices (such as tablets and cell phones), and the driving need for ongoing education and retraining, it is little surprise that online education now predominates.¹¹
With TokBox promoting the OpenTok™ standard and its wide adoption, its longevity, continued maintenance, and enhancement are insured. Initiatives are already underway to move OpenTok™ away from its reliance on Flash to the newer HTML 5 standard. In the near future, advancements in graphics processor technology and connectivity speeds will further improve the end-user experience, increasing both the resolution and number of concurrent participants.

1 Comparative Analysis of Webconferencing Tools; http://www.webconferencingsoftware.org/webconferencing-system.php.

2 Øystein Arisland, Kjell, Is bandwidth really a factor limiting real time Web interactivity, or are Web experts the main limitation?, Position paper for the Workshop Real Time Multimedia and the Web, http://www.w3.org/AudioVideo/9610_Workshop/paper08/paper08.html.


7 Christensen, Ibid, p 69.


