How can CDNs support non-HTTP traffic?
By: Jim Metzler

The Hyper Text Transport Protocol (HTTP), which has been in use for roughly twenty years, is a text-based, request-response protocol that reflects the traditional client-server computing model. In one common scenario in which HTTP is used, a Web browser acts as the client and an application running on a computer hosting a Web site acts as the *server*. The client submits an HTTP*request* message to the server. The server, which provides resources such as HTML files and other content, returns a response message to the client.

Because it is a text-based, request-response protocol, it is appropriate to use HTTP in the scenario described above: A user accessing HTML files. It is less appropriate, however, to use HTTP to support other traffic types, including bi-directional, real time communications such as voice and video. That is why in 2011 Google released an open source project for browser-based, real time communications known as WebRTC[1] [2]. In addition to Google's open source project, an initiative was established to standardize the relevant protocols in the IETF[3]. While the IETF has not mandated a particular signaling protocol, SIP[4] over Websockets[5] is often used.

As noted in the *2015 WebRTC State of the Market Report*[6], over half of businesses are already using WebRTC or plan to use it in 2015, and over three-quarters of respondents will use it next year and beyond. In addition to the large and growing interest in WebRTC, a number of real-time communications services which don't rely on HPPT have been introduced into the market. That includes Viper[7], ooVoo[8], and WhatsApp voice[9].

Many of the traditional Content Delivery Networks (CDNs) were designed to support remote users accessing centralized information and because of that design, these CDNs focus their acceleration at the HTTP or application layer. As a result these CDNs can't accelerate traffic that runs over WebRTC, SIP, Websockets or any other non-HTTP traffic. One of the unique characteristics of Teridion's CDN is that it works at the routing layer, not the application layer. As a result, Teridion's CDN accelerates any traffic, whether it is based on HTTP or some other protocol.

[1] http://www.webrtc.org/
[2] http://www.html5rocks.com/en/tutorials/webrtc/basics/
[3] http://tools.ietf.org/wg/rtcweb/
[4] https://en.wikipedia.org/wiki/Session_Initiation_Protocol
[5] https://www.websocket.org/
[6] http://www.webtorials.com/main/resource/papers/webtorials/2015-WebRTC/2015_WebRTC.pdf

[7] http://www.viber.com/en/
[8] http://www.oovoo.com/
[9] http://www.techtimes.com/articles/47891/20150423/whatsapp-voice-calling-for-ios-begins-to-roll-out.htm