Lessons from the Real-Time Web

Bernard Aboba <bernarda@microsoft.com>
Principal Architect, Microsoft

Abstract

The real-time web is not new. For much of the last decade, developers have been building real-time web applications that are today being used by tens of millions of users on a daily basis. These services, including WebEx, Live Meeting, Adobe Connect, DimDim and others, have proven themselves valuable to users in enterprise, small business and educational markets. This paper describes some of the lessons that have been learned by these services -- lessons which in many cases contradict fundamental assumptions that have guided the development of the IETF's Realtime applications area.

Introduction

The real-time Web is not new. Services such as WebEx and Placeware began operation in the early 2000s and by now these services as well as others such as Adobe Connect and DimDim have matured, both in terms of the functionality offered as well as the number of users they serve. Today these services support audio and video conferencing as well as multi-user chat and screen sharing on a wide variety of platforms, including multiple operating systems and form factors, including PC desktops, notebooks, mobile handsets and tablets. Given the experience in providing this wide array of realtime services to a large number of customers, it is worth examining the implications for the future of the real-time web.

This paper examines several aspects of this experience, including:

- The real-time web "only just works"
- HTTP as transport
- Mobile platform support

The Real-Time Web Only Just Works

While web conferencing services are not yet as wildly successful as the most popular websites on the Internet, these services now cater to millions of users so they qualify as successful within the terminology defined in [RFC5218]. This is particularly impressive given the variety of services that are being offered.

One reason that Web Conferencing services work as well as they do is that the services utilize a client-server architecture complex enough to provide the required functionality, but no more. In the terminology coined by Mark Handley in [Just-Works], web-based conferencing systems “only just work”. As noted in [Just-Works], it is just these kind of bare-bones designs that often garner success on the Internet and prove surprisingly resistant to competition from over-engineered competitors. This not only has paid dividends in terms of reduced complexity and improved resilience, but also in terms of improved agility. An example of this is the work required to support new technologies such as IDNA and IPv6.

While improved support for internationalization was not one of the original motivations for the design of existing web conferencing architectures, it has turned out to be one of the fortuitous benefits of a web-based approach to realtime communications. As a result, the addition of support for technologies such as IDNA and IPv6 has proven much easier for web conferencing clients than for the backend infrastructure.

While all major browsers today support IDNA to at least some degree, the same cannot be said of SIP implementations. For example, while RFC 3261 theoretically supports SIP internationalization, in practice few implementations support IDNA. Few SIP service providers support routing of requests to email-style URIs at all, let alone requests destined to E.164 numbers with an internationalized domain name.

The situation is somewhat better with XMPP implementations, many of which appear to have put significant effort into ensuring support for latin and non-latin scripts and support of non-latin usernames and passwords. However, tests performed on XMPP clients with non-Latin scripts have disclosed quite a few bugs, and addressing these issues in a complete and comprehensive way will no doubt require significant additional effort.

The lesson here is that the web-based development model provides the developer with at least partial solutions to a number of hard problems. Internationalization is not the only example of this; some of the same characteristics can be seen with support for IPv6. Once again, support for IPv6 within browsers is substantially better than the situation with SIP implementations of IPv6, which can best be described as “primitive at best”.

HTTP as Transport

In many cases, real-time web services have been built on HTTP transport, not only for signaling, but also for media. While initially, this approach may have been a matter of convenience, over time some fundamental advantages to this approach have become apparent.

Foremost among these is the ability to traverse firewalls. While the widespread deployment of NAT has brought with it a set of challenges that have largely been addressed by the development of protocols such as STUN, TURN and ICE, in practice this has not proven sufficient to enable real-time connectivity in a number of practical uses cases. One such case is the deployment of a highly restrictive firewall, such as one which blocks UDP traffic in both directions, or only allows establishment of TCP connections to a few well-known destination ports such as port 80 or 443. In some cases, even connections established to well-known ports may face Deep Packet Inspection (DPI) that can mark traffic as “suspect” if it utilizes features which are not currently mainstream.

The deployment of such highly restrictive firewalls has now become so common that real-time web developers addressing the enterprise market cannot afford to ignore them. As a result, transport of both signaling and media over HTTP, once a convenience, has now become an indispensible feature which is supported by virtually every major web conferencing service. However, given the increasing deployment of DPI, we should not under-estimate the difficulties that could potentially face the adoption and deployment of the HTTP extensions now contemplated by the IETF.

HTTP for signaling

While some web conferencing clients include support for IP-based transport of protocols such as SIP and XMPP, virtually every service supports HTTP-based transport for signalling in order to address firewall traversal issues. However, beyond this basic commonality, existing web conferencing services appear to utilize a wide variety of approaches, including XMPP over BOSH, SIP encapsulation within HTTP, web-service and REST APIs, etc. Just as the development of web-based email services seems to have progressed without the need for additional standards beyond those that the web already provides, so too is there no apparent need for convergence on standards for HTTP-based realtime signaling beyond HTTP standards themselves and the bi-directional HTTP transport functionality provided by approaches such as BOSH and (more recently) websockets.

Typically, IP-based transport of protocols such as such as SIP and XMPP is more common within the backend infrastructure. Particularly given the need for connectivity with the PSTN as well as federation with consumer XMPP-based services, this support seems likely to remain a requirement.

HTTP for Media

Due to the firewall traversal issues referred to earlier, virtually every web conferencing service also supports HTTP-based media transport. While this approach was viewed as highly unconventional in the early years given the long history of UDP-based transport of RTP, the HTTP-based media transport works very well in many cases.

Not only can HTTP-based transport deliver media in situations where it would not otherwise be possible, but tests such as [VOIP-SSL][RTP-Measure] and simulations such as [VOIP-Eval] demonstrate that in situations with low packet loss and congestion, reliable transport can deliver a high quality user experience. As noted in studies such as [Paxson], low packet loss is very common on the Internet. As a result, HTTP transport of media should not be thought of as a “problem” that needs to be “fixed”, but rather an example of the “only just works” philosophy that has brought the real-time web this far.

In situations with substantial packet loss, high queuing delays or both, both reliable and unreliable transport can experience difficulties in delivering a high-quality user experience. In these situations, aspects of the codec such as sensitivity to packet loss, use of FEC and bandwidth adaptation become critical, and as a result, the performance of media transports should not be evaluated in isolation, but rather in combination with a particular codec. ITU-

T codecs have little to commend themselves in this regard. As noted in [RFC3714], non-adaptive codecs such as G.711 don't perform well when treated as "best efforts" traffic within a congested environment, and compressive codecs such as G.729 are highly sensitive to packet loss.

Mobile Platform Support

Web conferencing services have encountered a number of issues in providing support for mobile platforms including dependence on plugins, shortage of screen real estate and lack of multi-tasking support. As a result, in most cases web conferencing services have chosen to develop native applications focused on particular usage scenarios, rather than attempting to provide a full web conferencing experience on mobile platforms with limited screen real estate.

Existing web conferencing services typically rely on plugins that may not be available on some mobile platforms. Plugins such as Flash are not only are used to provide support for basic capabilities targeted by HTML 5 such as devices (webcams and microphones), media streaming and display, etc. but they are also used to provide support for more advanced services such as screen sharing. As a result, it may be a while before all the features of existing web conferencing services can be provided based on HTML 5 without any reliance on plugins.

Shortage of screen real estate has also been an issue. While web conferencing services are inherently multi-modal, a web UI supporting multiple modes of communication such as audio and video, screen sharing and chat requires more screen real-estate that is available on mobile handsets. While this problem would be less acute on a tablet form factor, it is difficult to overcome on a mobile handset. As a result, web conferencing developers have frequently customized their mobile implementations to target particular usage scenarios, rather than providing an "all in one" user experience. For example, the intial WebEx native application for iPhone did not attempt to provide all the functionality of the Web implementation, but instead focused on enabling the user to view the presentation and connect to the audio session via an incoming call (non-VOIP).

Yet another problem has been the lack of multi-tasking support on some mobile platforms. Given that support for asynchronous notifications were only possible within native applications on some platforms, web applications were at an inherent disadvantage, and typically a web-based approach to application delivery was only taken when no other avenue was available.

This is not to say that the technologies of the real-time web are not useful on mobile platforms. Protocol support such as HTTP-based transport of signaling and media are as useful for native applications as they are for pure web-based approaches. However, there are a number of situations in which it may be useful to implement particular protocols without having to pull in other aspects of the real-time web stack, such as HTML 5 or Javascript. As a result, it is quite important that a clean separation exist between core web protocols such as HTTP and websockets and higher layer technologies.

References

[Just-Works] Mark Handley, "Why the Internet Only Just Works", http://www.cs.ucl.ac.uk/staff/

m.handley/papers/only-just-works.pdf

[Paxson] Vern Paxson Ph.D. Thesis “Measurements and Analysis of End-to-End Internet Dynamics”, ftp://ftp.ee.lbl.gov/papers/vp-**thesis**/dis.ps.gz

[RFC3714] IAB, “IAB Concerns Regarding Congestion Control for Voice Traffic in the Internet”, RFC 3714, http://tools.ietf.org/html/rfc3714

[RFC5218] IAB, “What Makes a Successful Protocol?”, RFC 5218, http://tools.ietf.org/html/rfc5218

[RTP-Measure] Hwan Hwang, Xia Yin, Zhiliang Wang, Huiming Wang, “The Internet Measurement of VoIP on different Transport Layer Protocols”, Tsinghua Universit, Beijing, China

[VOIP-Eval] Panagiotis Papadimitriou and Vassilis Tsaossidis, “Evaluation of Transport Services for VoIP”, 2006.

[VOIP-SSL] “Test Shows VOIP call quality can improve with SSL VPN links”, Network World, http://www.networkworld.com/reviews/2006/022006-ssl-voip-test.html?ts