

HTTP versus RTMP: Which Way to Go and Why?

What You Will Learn

Like picking the color of your next car, even rich-media delivery offers choices. Streaming has seen many changes over the past several years, and has continued recently with varied solutions that use different transport protocols. Each transport offers its own benefits and cost structure. How and why you integrate each will depend on your specific application and how each protocol applies to you. This paper explores HTTP and the Real Time Messaging Protocol (RTMP) and identifies the critical data points, considerations, and examples to help you make decisions regarding your enterprise.

HTTP vs. RTMP – Quick Reference		
	HTTP	RTMP
Server Components	Open	Specialized
Security/IP Protection	Basic	Enhanced
Caching/Scale	Native	Specialized
Player Interactivity	Basic	Extensive
Multicast Support	N/A	Supported
Plug-in Penetration	Nascent	Extensive
Firewall (port/protocol)	No restrictions	Some restrictions
Variable Bitrate Support	No impact	Susceptible to data spikes

What Is It All About?

New streaming technologies from Apple®, Adobe, and Microsoft have introduced new methods of media delivery that improve user experience and startup time—and add features and functions. To cover all of these offerings, we use the term adaptive bitrate (ABR) to denote that the technology provides for varying experiences, depending on the conditions on the user’s side. These primary changes in the present digital media delivery landscape borrow from the old adage: what is old is new again.

In the early days of web video delivery, users had to rely on progressive delivery (also referred to as progressive download [PDL]) of video, meaning that the bits of video were delivered to your player one packet at a time “in the blind,” with no communication between the server and player. When a reasonable percentage of the file was downloaded to disk, the player would begin playing the file. However too often the player caught up with the point at which the file was being delivered, and playback halted.

As a result streaming was created—a mode through which the video is passed to the player, with increased communication and monitoring in place, and it happens in real time between the player and server. If bandwidth degrades on the player side, it signals the server and “buffers” until it can obtain a suitable amount of packets of video to resume playback. Today, more video technologies are using HTTP again and reinventing your father’s PDL mechanism. By building in monitoring and quality-of-service (QoS) measurements, HTTP can provide vibrant video experiences, while at the same helping with scale and access concerns discussed later in this paper. The user with plentiful broadband is rewarded with the best video offering available, whereas the user experiencing fluctuations in bandwidth or low-grade connectivity will still receive the same video, albeit with a lower-quality experience.

The Acronyms

Hypertext Transfer Protocol (HTTP) refers to the protocol used to deliver webpages and images across the Internet worldwide. HTTP is an adopted, open standard—the most ubiquitous mode of delivery online. HTTP is a “stateless” protocol; think of it as an airline ticket to anywhere. HTTP can be delivered by a variety of web servers, both commercial and open source.

Real Time Streaming Protocol (RTSP) is a network control protocol used in entertainment and communications systems to control streaming media servers. RTSP is used to establish and control media sessions between two points, usually server and player client. Clients of media servers issue VCR-like commands, such as PLAY and PAUSE, to facilitate real-time control of playback of media files from the server. Microsoft’s Smooth Streaming is a hybrid delivery method that acts like traditional RTSP streaming but is based on HTTP PDL. Apple®’s HTTP Live Streaming (HLS) is another example of using HTTP to deliver video with new techniques.

Real-Time Messaging Protocol (RTMP) refers to the proprietary protocol developed by Adobe Systems for streaming audio, video, and data over the Internet between a Flash player and a Flash Media Server. Like RTSP, RTMP is an example of a traditional streaming protocol, though it is only one of many versions of streaming protocols for the web. RTMP is defined as a stateful protocol, meaning that from the first time a client connects until the time it disconnects, the streaming server keeps track of the client’s actions. The client communicates its actions, or “states,” to the server by issuing commands such as PLAY or PAUSE. When a session between the client and the server is established, the server begins sending the media as a steady stream of small information packets. This behavior continues and repeats until the server or player client closes the session.

In June 2009, Adobe Systems published the RTMP specification, meaning that content delivery networks (CDNs) and enterprises that use Flash technology can now author their own implementation—possibly resulting in Flash delivery, even the new Dynamic Streaming, enabled natively in HTTP.

Advantages and Disadvantages of HTTP, RTSP, and RTMP

Which technology and delivery mode will work best for you? The answer depends on many data points. Let’s explore a few. How large is your anticipated daily traffic or single event? Do you need to protect your content from being downloaded locally and republished by the user? (Think of a YouTube scenario where your favorite television show is available there, but probably should not be.) What is the length of your content: many files with short duration, or fewer files with very long length?

HTTP is less likely to be disallowed by routers, Network Address Translation (NAT), or firewall settings. No ports that are commonly closed by default need to be opened. Content is therefore more likely to get through to the client in more locations and without special settings. HTTP is also supported by more CDNs, a factor that can affect cost in large distribution models. In general, more available hardware and software works unmodified and as intended with HTTP than with RTSP or RTMP. Expertise in customizing HTTP content delivery using tools such as Hypertext Preprocessor (PHP) is also more widespread. Additionally, for large-scale events, HTTP natively and easily supports mirroring and edge caching, providing for massive-scale expansion when needed for the largest events. RTSP and RTMP can also be cached, but HTTP does so natively and without the need for proprietary or custom configurations.

Access is another consideration. Can everyone view your content after you make it available? From a site visitor’s point of view, one advantage of using HTTP is access. Many corporate networks use firewalls to block specific content. Popular methods of blocking are protocol and port restrictions. Some firewall rules allow only HTTP content served over port 80. However, this luxury is not always shared with RTMP-delivered Flash Video. The

default port for RTMP connections is 1935—a port that may not be allowed on tight firewalls. If the first attempt of the Flash Player to play video over port 1935 fails, it tries to reconnect using a few different methods. To summarize, if you do not want to deal with firewalls and proxies, you should consider HTTP.

One benefit with RTMP worth mentioning here is its ability to provide multicast support. If you run an enterprise and want to take one stream inside your corporate network and deliver it to many users without initiating a new connection for each user, RTMP is the best technology. HTTP does not provide this function, nor do CDNs.

Security is critical with a lot of content, and if you are a distributor or aggregator, you know the term digital rights management (DRM) all too well. Can or will you allow your content to be saved locally on a user's machine? RTMP has broad DRM support, and rights holders are very familiar with the ecosystem.

For example, one version of RTMP is RTMP Encryption (RTMPE), a new method for real-time cryptographic protection of content offered by Adobe. Some HTTP approaches such as Smooth Streaming are rapidly moving toward deployment of solutions for DRM as well. However, securing your content may be a concern. Some ABR technologies provide for secure transmission through Secure HTTP (HTTPS), which represents the secure way of transmitting HTTP video. HTTPS, however, is not a commonly accepted form of DRM, and it may provide a contractual and legal hurdle with rights holders who stipulate that DRM be used to protect their content. As a mode of transport, HTTP can be captured and saved locally. Chunked delivery can make this local capture more challenging, but the ability still exists. Conversely, RTMP is much less susceptible to local data capture, but in rare instances streams have been circumvented and saved locally using third-party applications.

The availability of plug-ins may be a concern to you as you consider these technologies. Flash is the most broadly deployed plug-in, but it has caveats. Dynamic Streaming requires Flash Player 10, which is growing in adoption (and pegged at nearly 80 percent as of July 2009), but has not reached full market penetration yet. Microsoft's Silverlight has broad support on new PC builds; it is being used for some large-scale sports events (March Madness, Wimbledon, and 2010 Winter Olympics), but has not yet achieved the reach that Flash currently enjoys. Additionally, organizations such as Major League Baseball (MLB) and Indy Racing are using technologies from third-party providers that use HTTP, such as Swarmcast and Conviva. To summarize, if your video is compelling and users are required to install a plug-in, most will.

Tracking your content and how many people view it is a "must have" for most, but how granular must your details be? Due to the nature of RTMP discussed previously—where the player and server are in constant communication—reporting within RTMP is much more detailed and granular when compared to HTTP. The logs derived from RTMP servers provide specific data about what precisely a user did within a video. How long did the user watch? At what point in the video did the user disconnect? Did the user grab the "scroll bar" and advance through the video to a specific point? All these questions can be answered with an RTMP-based solution today. Expect HTTP-based technologies to improve on this deficiency in the future, but reporting is more robust with RTMP now.

Finally, and perhaps most importantly, what will it cost you to provide this content to your users? What will a CDN charge you to provide your content in one mode versus another? In the past, many CDNs charged a premium for RTMP streaming transit, because of the specialized skills needed to set up and manage those servers and the complexity in ensuring the content could get to the user. Today, the "Flash tax" is gone and the total cost to deliver content through RTMP has decreased. However, HTTP still represents most web traffic. Consider that pushing video through those same protocols and methods enables CDNs and those who own their infrastructure to realize economies of scale and the cost savings that accompany them. As a result, your total cost of ownership (TCO) may be less with HTTP than with RTMP.

Summary

So picking a delivery method may not be as easy as picking the color of that new car. All three major ABR technologies have their benefits and risks. Adobe's Flash is the ubiquitous choice for video-on-demand (VoD) and has tremendous penetration on the desktop. Microsoft's Smooth Streaming has had some early live event successes, and it helped provide video for the 2010 Winter Olympics. Apple's HTTP Live Streaming is available on 25 million handsets, but it has limited desktop reach and longer latency and startup because of the data network it operates on.

Whether you choose HTTP, RTMP, or both, the Cisco® Digital Media Processor Family of products supports both technologies now; the processors are ready to use, with ease of setup and use, and we have a rich history of success with our customers. We are ready to help you assemble your next digital media solution.

For more information, please visit [Cisco Media Processor Family page](#) or call your local Cisco account representative.



Americas Headquarters
Cisco Systems, Inc.
San Jose, CA

Asia Pacific Headquarters
Cisco Systems (USA) Pte. Ltd.
Singapore

Europe Headquarters
Cisco Systems International BV Amsterdam,
The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at www.cisco.com/go/offices.

Cisco and the Cisco Logo are trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and other countries. A listing of Cisco's trademarks can be found at www.cisco.com/go/trademarks. Third party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1005R)