

**From:** Troy Batterberry  
**Sent:** Thursday, May 17, 2001 12:19 PM  
**To:** Will Poole; Mike Beckerman; Chadd Knowlton  
**Cc:** Amir Majidimehr; Tom Honeybone; Gary Schare; Bret O'Rourke  
**Subject:** RE: windowsmedia.com experience

See comments below:

-----Original Message-----  
**From:** Will Poole  
**Sent:** Tuesday, May 15, 2001 11:34 PM  
**To:** Troy Batterberry; Mike Beckerman; Chadd Knowlton  
**Cc:** Gary Schare; Amir Majidimehr; Tom Honeybone  
**Subject:** RE: windowsmedia.com experience

Basically you are saying a few things here:  
our current implementation of MBR video is effectively worthless. Not clear whether we'll get to a better solution in Eclipse.

**[Troy Batterberry]** I am cautiously optimistic we are on track to address the known shortcomings of MBR in eclipse. It will certainly be much better than the current implementation. firewalls / proxy will continue to be a problem for a long time

**[Troy Batterberry]** Yes. However, our strong push away from datagram fragmentation in Eclipse (by both the repackitization feature and also the changes to the Titan Encoder that creates small ASF packets by default when possible) should alleviate many known NAT/firewall issues. By having the encoder create smaller (<1500 byte) ASF packets, we eliminate many known problems with NATs/firewalls for all player versions and all server versions. We are currently investigating the impact of evangelizing the key sites to start producing content files with smaller ASF packets using the existing released encoder. (Packet size is an adjustable setting in the v7 encoder). Unfortunately, problems with fragmenting WMA audio packets across an ASF packet preclude this down-level solution from working in some specific cases. The server performance is also degraded by streaming smaller packets. I'll have more data soon.

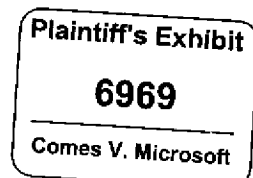
client feedback to user and to server is not great, at best

At the risk of proving my ignorance, please explain to me why we can't do the following, even for legacy servers, in an updated client and with easily re-authored content metafiles:

- 1) Automatically switch from UDP to HTTP (on a semi-permanent basis rather than every time)
    - a) detect when the stream over UDP is lousy
    - b) offer switch to HTTP (and include "do automatically next time, or do always unless I change it later" option)
    - c) detect when HTTP is lousy and put better msgs in the status area
- [Troy Batterberry]** Today, we roll over to HTTP based on a variety of faults/problems with MMSU/MMST. However, we do not roll over when our UDP reception is poor. It is worthwhile to investigate rolling over to MMST/HTTP (and infrequently changing back to MMSU with our existing latch logic) in such cases when streaming from a v4.x server. We really need to think through the logic though - TCP-based streaming will be far worse in many scenarios. Characteristics such as round-trip times, additional network headroom (to overcome occasional TCP-slow start issues following inevitable packet loss), and buffer size factor into the heuristics. Depending on the implementation, it could also greatly increase the complexity of the client code. Note the proposed solution of rolling over to HTTP midstream is problematic against the current Hercules server implementation and server-side playlists. With Hercules, it may be best to dynamically roll over from RTSP/UDP to RTSP/TCP when heuristics determine it appropriate.

- 2) Offer simple ASX-specified tag set that says either

MS-CC-Bu 00000058926  
HIGHLY CONFIDENTIAL



MS-PCAIA 5008916

(a) play this standard bitrate stream, if things don't go well, switch to this lower bitrate version of the same thing starting at about the same offset., or

(b) play the fixed rate stream rate that most closely matches the players most recently determined network throughput without going over...

**[Troy Batterberry]** Your proposed solution does provide some good benefits:

1. It is compatible with all released servers.
2. It theoretically could allow the content author to create a series of single bitrate files that are used by both streaming servers and download sites. (MBR files are obviously very unattractive for download scenarios).
3. It allows for very fine control by the content author.

However, adding MBR-like capabilities in the metafile (ASX/WVX/WAX) syntax has several problems.

1. It doesn't really offer a down-level player solution...something that is arguably more important than a down-level server solution. Server adoption has historically been much easier/quicker than client adoption. (We may even want to consider mandating that all WindowsMedia.com references must point to a Hercules server at some point). Even the eclipse MBR solution offers down-level client compatibility and some basic functionality.
2. Metafile parsing is not done in the SDK. 3rd party players (built on the SDK) would be problematic and unable to utilize these new tags without releasing a new player. Note 3rd party players can dynamically select and stream MBR files today. A Zeus runtime upgrade will provide them with enhanced MBR functionality.
3. This metafile-style MBR functionality would largely duplicate the MBR capabilities already present in v7 and enhanced in eclipse. Given the very large cost of building and testing MBR functionality, creating and supporting yet another solution is not practical given the current engineering staff and schedule constraints.
4. The solution places the burden (and the propensity for error) on the content author to write the correct syntax in the metafile file. Currently, the MBR solution in v7 (and eclipse) is seamless to the content author - a single URL (and a single property encoded content file) is all that is needed. (The SDK code makes all the decisions based on the dynamic response of the network).
5. The proposed solution has serious design implications for Hercules server-side playlists.
6. Cross-platform players would need to be updated with new metafile parsing code in order to utilize the basic functionality.
7. Because of the extremely limited initial reach of a metafile solution, CDNs and content authors would likely be very reluctant to deploy it.

3) have the player figure out what rate it is really seeing before asking for the streams in (2), caching the rate with a time stamp and re-determining it when needed.

**[Troy Batterberry]** Storing/caching bandwidth measurements is already being done in most of our players, and a statistical mean value is calculated. The mean value is programmatically available (and writeable) via the v7 player control.

4) have the client notice when video is lousy/thinned (I see it all the time, but the stats almost always say quality is 100%!!) and pop up after a while with a few suggestions and a pointer to web site to diagnose/re-tune things.

**[Troy Batterberry]** Agreed this is very bad and it needs to be fixed. The statistics currently only measure packet loss as the main quality indicator. Since TCP transmission ensures we never lose a packet, we report 100% even when the player thins or re-buffers. There are multiple DCRs/bugs that have been tracking this topic. We simply need to get it fixed.

6) get a bandwidth detection widget built into IE n+1 that every site can use to detect what the user has and offer only the appropriate content.

**[Troy Batterberry]** Our b/w detection is built directly into our wire protocols and the streaming media server. Today, the b/w measurement occurs during each open request, and aside from

MS-CC-Bu 000000058927  
HIGHLY CONFIDENTIAL

MS-PCIA 5008917

reactive thinning, it is not really updated during the streaming session. The long-term goal is to make the b/w detection logic much more dynamic/continuous during a streaming session. This implies that the logic should probably remain in the streaming protocol and not be transitioned to IE at this point, but more investigation is needed.

The ability to dynamically generate URLs based on the user's known b/w is compelling while we wait for our eclipse MBR solution to be deployed. Some of this may be possible today with some very creative script code and the v7 player control. (I need to look into this further)  
Troubleshooting pages could point the user through steps to potentially rectify problems...and we could gather better statistics on just how many people are hitting the problem. Unfortunately, having content authors deploy a v7-specific web page will require some efforts on our part given the minority reach of that player. A dual page approach (one for PCs/devices that lack v7 or later) would have to be used.

I agree with Jim that we should be able to do more with both the user experience and the overall system prior to a world-wide deployment of Eclipse that will take years (but which will clearly help on many fronts as you describe below).

-----Original Message-----

From: Troy Battenberry  
Sent: Tuesday, May 15, 2001 6:30 PM  
To: Will Poole; Mike Beckerman; Chadd Knowlton  
Subject: RE: windowsmedia.com experience

### **I. Executive Summary.**

Unfortunately, there are many known problems in the existing v7 platform that result in a poor streaming experience. Many significant low-level improvements have been added to the Eclipse platform release to address streaming problems and limitations associated with MBR deployments and penetration of firewalls/NATs. We also need to improve our end-user experience for displaying network status information. Finally, we need to continue to refine our network bandwidth detection logic.

### **II. Detailed Information.**

The following are known problems that may be relevant to Jim's comments.

#### **A). Problem: CDNs do not currently deploy our Multi-Bitrate (MBR) content feature.**

##### **Scenario:**

Jimall's request for a single URL that does all the automatic b/w detection/stream selection/thinning logic is technically possible today with the v7 platform. However, due to a variety of reasons/limitations, CDNs do not use the functionality. Instead, they encode content and author links with explicit "single-bitrate" streams.

##### **Cause:**

#### **1. No audio-MBR platform support.**

While a content author can create a file with multiple video streams, only one audio stream can be employed. Since the desired audio stream for a 28 kbit/s modem user will be considerably less than the desired audio stream for a 700 kbit/s DSL user, this limitation is considered quite severe. The limitation will be addressed in the eclipse release.

#### **2. No multiple video resolution platform support.**

The existing platform does not offer the ability to have multiple video resolutions in a single MBR file. For instance, a content author may wish to target a modem user with a 160x120 video resolution and a DSL user with a 320x240 resolution. This limitation will be addressed in the eclipse platform.

#### **3. Mux limitations in the encoder.**

MS-CC-Bu 000000058928  
HIGHLY CONFIDENTIAL

MS-PCAIA 5008918

Our current encoding mux has limitations that preclude the use of a wide range of stream bitrates in a single file. By abstracting ASF packet sizes from UDP datagram sizes via the use of dynamic repacketization in eclipse, this problem will be solved.

**4. Disk I/O performance degradation on the server when using MBR files.**

The server reads the entire MBR file to parse out the relevant stream(s) for a given client. Since our reads explicitly disable operating system caching, each individual client must read the entire file from disk. In the case where the MBR file has many individual streams, the perf hit to the disk i/o is very large. The perf team is investigating ways to enable some caching without causing disk paging. (Disk paging degrades the pseudo-realtime properties of the streaming scheduler in the server and needs to be avoided)

**Solution:**

1. Audio-MBR in Eclipse.

2. Multiple video MBR resolutions in Eclipse.

3. Increase evangelization of MBR.

Once we have a stronger MBR solution, our deployments team will increase the evangelization of the solution.

4. Dynamic repacketization support in eclipse.

**B). Problem: Streaming high-bitrate content via UDP through firewalls/NATs is problematic.**

**Scenario:**

Due to a variety of technical reasons, users streaming higher bitrate content (>100 kbits/s) through ISA/Winsock Proxy, Win98 ICS, Blackice defender, and others have observed severe problems. The problems are especially notable when using the MMSU (UDP) protocol and fragmented datagrams.

**Cause:**

1. Firewalls/NATs do not handle datagram fragmentation well.

When the datagram becomes fragmented, many firewalls mistakenly drop the trailing fragment. The end-user experience is terrible.

2. Incorrect MMSU protocol assumptions.

The MMSU (binary) protocol was not fully documented, and firewall/NAT vendors were provided with incomplete information on how to build the NAT/firewall. (In some cases, they attempted to reverse-engineer the protocol themselves. They made many mistakes). Protocol interop with these solutions remains a big problem.

**Solution:**

1. Post updated version of the MMSU protocol document for firewall vendors.

The document needed for firewall vendors to correctly build a product that supports marshalling of a MMSU stream was updated last month and will be posted to [www.microsoft.com](http://www.microsoft.com) shortly. Previously, it was not made widely available to ISVs. (Note this document contains the minimum information necessary to write a firewall for MMSU/MMST. It does not provide information for any cache/stream splitting proxy support).

2. Phase out MMSU/migrate users to RTSP.

RTSP offers a reasonably standard way of penetrating firewalls/NAT/etc with streaming media. This should reduce the amount of interop problems we are experiencing. (Note our HTTP streaming compatibility problems are extremely low/non-existent). We are in the process of filing several patents that extend the use of RTSP without breaking firewall interoperability and yet still protect our intellectual property.

MS-CC-Bu 00000058929  
HIGHLY CONFIDENTIAL

MS-PCAIA 5008919

3. Continue to work closely with ISA, Winsock Proxy, ICS, etc, and fix known interop bugs in their products.

For instance, we have been tracking a serious ISA bug that results in massive packet loss when using the Winsock proxy client in certain scenarios. They are committed to providing a fix soon. We have also worked with 3rd party vendors to address their interop problems.

4. Reduce/eliminate the need to fragment UDP datagrams.

a). Dynamic repacketization.

The repacketization feature in the eclipse platform allows the server to stream UDP datagrams that generally won't become fragmented and cause problems with external applications. ASF packets are now abstracted from the wire packets (i.e. RTP packets). This solution requires both a v9 player and a v9 client to provide any benefit.

b). Set the default ASF packet size in the eclipse encoder to 1452 bytes.

By reducing the default ASF packet size in the encoder for internet bitrates (i.e. < 350 kbits/s), legacy clients will not receive fragmented datagrams. This will reduce the problems associated with UDP streaming.

5. Progressive Streaming.

This new streaming metaphor allows the v9 client to "stream ahead" by using extra network b/w available and caching on the disk. The feature should greatly improve the end-user experience on "bursty" networks such as cable modems and corporate firewalls where available b/w constantly fluctuates.

C). Problem: The media player does a poor job of warning users that it has initially "thinned" to audio only when insufficient b/w is available.

Scenario:

The user clicks on a audio/video URL, but they only receive audio. No error or status message is displayed.

Cause:

The b/w detection logic has determined that insufficient b/w exists to stream both the audio and the video portion of the file. In some cases, there truly is not sufficient b/w to completely stream the URL selected. In other cases, the b/w detection logic is erroneous.

Solution:

This is simply a poor platform design and has been present since the v6.4 player. We have plans to tell the user of the problem perhaps through a status message in Rosetta. More usability work is needed. We also need to continue to refine our b/w detection logic for diverse network topologies (i.e. PPTP, wireless, NAT/Firewalls, etc) to further improve the accuracy. B/w detection in general is a very difficult problem.

-Troy

-----Original Message-----

From: Mike Beckerman  
Sent: Tuesday, May 15, 2001 11:38 AM  
To: Will Poole  
Cc: Chadd Knowlton; Troy Batterberry  
Subject: RE: windowsmedia.com experience

Chadd, Troy, can you give Will some background on this?

-----Original Message-----

From: Will Poole  
Sent: Tuesday, May 15, 2001 9:10 AM

MS-CC-Bu 00000058930  
HIGHLY CONFIDENTIAL

MS-PCAIA 5008920

To: Mike Beckerman  
Subject: FW: windowsmedia.com experience

-----Original Message-----

From: Jim Allchin  
Sent: Tuesday, May 15, 2001 9:07 AM  
To: Will Poole  
Subject: RE: windowsmedia.com experience

It is a terrible experience. If you want video/audio to take off, we need to fix.

Jim

-----Original Message-----

From: Will Poole  
Sent: Tuesday, May 15, 2001 9:05 AM  
To: Jim Allchin  
Subject: RE: windowsmedia.com experience

agree about the experience; is a very hard problem. we are doing work in this area. I will get an update on where we are with it.

-----Original Message-----

From: Jim Allchin  
Sent: Tuesday, May 15, 2001 8:53 AM  
To: Will Poole  
Subject: RE: windowsmedia.com experience

On another topic....

The fact that you have to pick a speed is terrible. If you pick DSL or whatever on campus for example, the video is unusable. You hear the audio, but no video. We need to do better than this. It needs to be automatic. Our software should be able to sense whether we are dropping frames or not consistently and then fix the speed.

This is a huge usability problem and it could be a competitive advantage.

Jim

-----Original Message-----

From: Will Poole  
Sent: Tuesday, May 15, 2001 12:09 AM  
To: Bill Gates  
Cc: Eric Rudder; Jim Allchin  
Subject: windowsmedia.com experience

You clearly have some ideas on things we can improve on the new site. I got the one about personalizing to age. (We don't have personalized content generator based on profile/prefs yet, but will look and see what we can find). We have greatly improved the stream quality, which I know bothered you before. We do this via more stream monitoring tools and aggressively kicking off partner content that does not meet the bar.

Are there other things we should be looking at, especially as we figure out how to get the experience more integrated with MSN?

Thanks

MS-CC-Bu 000000058931  
HIGHLY CONFIDENTIAL

MS-PCAIA 5008921