AMT Multicast Production Implementation

Scott Brown – GM, USA

IETF 85

11/04/2012

Presentation Outline

- Ecosystem Challenges
- Design Goals
- From Trial to Production
- Technology implementation background

What is the challenge?

- Today streaming video to 2MM simultaneous @ 2Mbps is difficult and expensive to achieve even across multiple content distribution network providers.

- Trying to achieve this today outside of the US, is simply not a reality for most broadcasters due to the fact that most of the CDN capital is deployed US centric.

- TV sized audiences are coming, and at relatively high quality 2-5 Mbps.

- Current IPTV deployments with provisioned QOS Multicast are looking to provide this content to Multiscreen over the broadband / unmanaged side of the network

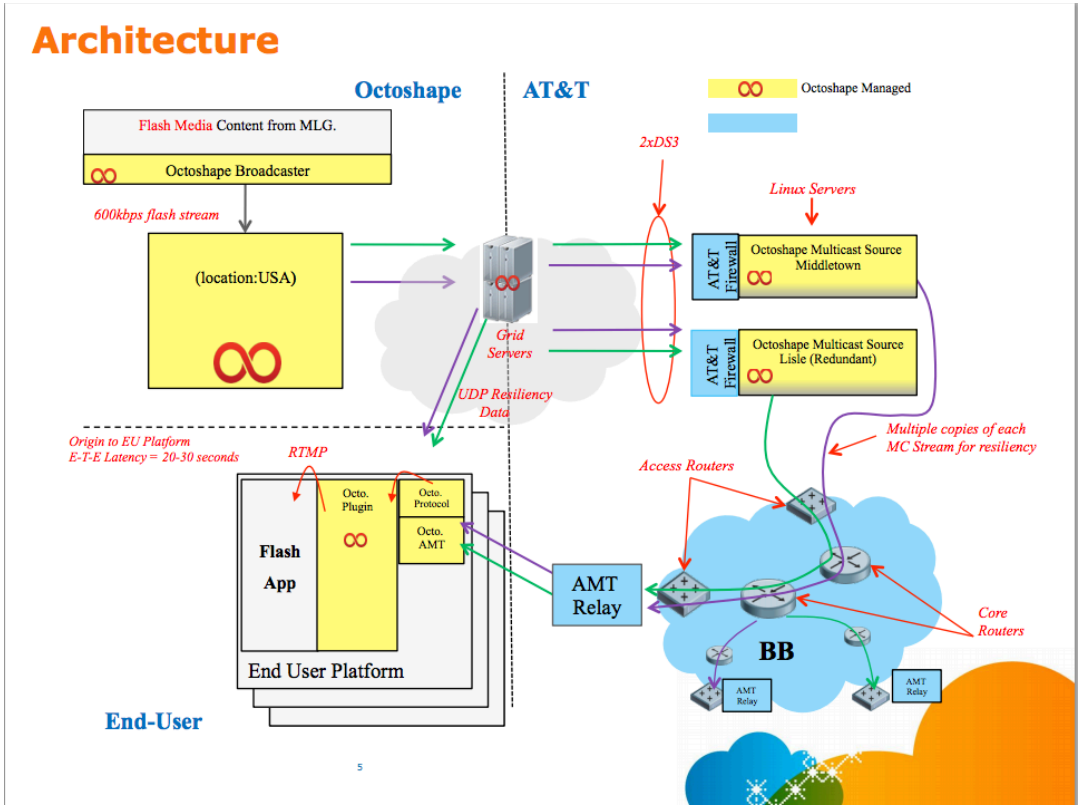
Multicast Design Goals

- Needs a transparent transition path to Multicast (video streaming must work even if multicast or AMT does not exist)
- Needs to support HTTP, but HTTP streaming is not well suited for multicast; it is a progressive download technology which simulates streaming
- Needs to work without telcos having to make complicated Multicast peering arrangements
- Needs to work over best effort networks without making the telco/operator do expensive provisioning for QOS
- The solution must also support adaptive bitrate to enable different consumer broadband, mobile and wifi networks
- Must not require the telco/operator to install proprietary software in their network

ATT / MLG Pilot

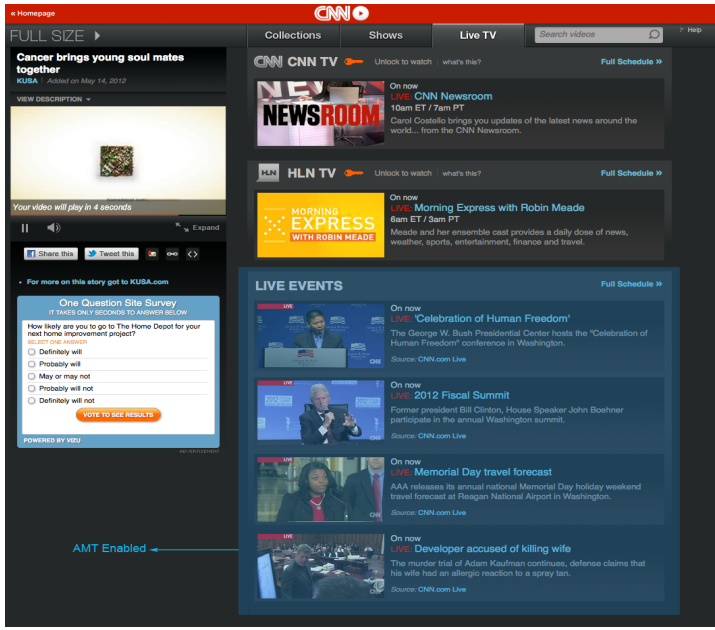
- In March 2011 AT&T and Octoshape presented an AMT Pilot architecture using a gaming customer streaming event over the public Internet

- In April 2012 Octoshape launched a production streaming platform on this AT&T / AMT network



Global Production AMT Examples

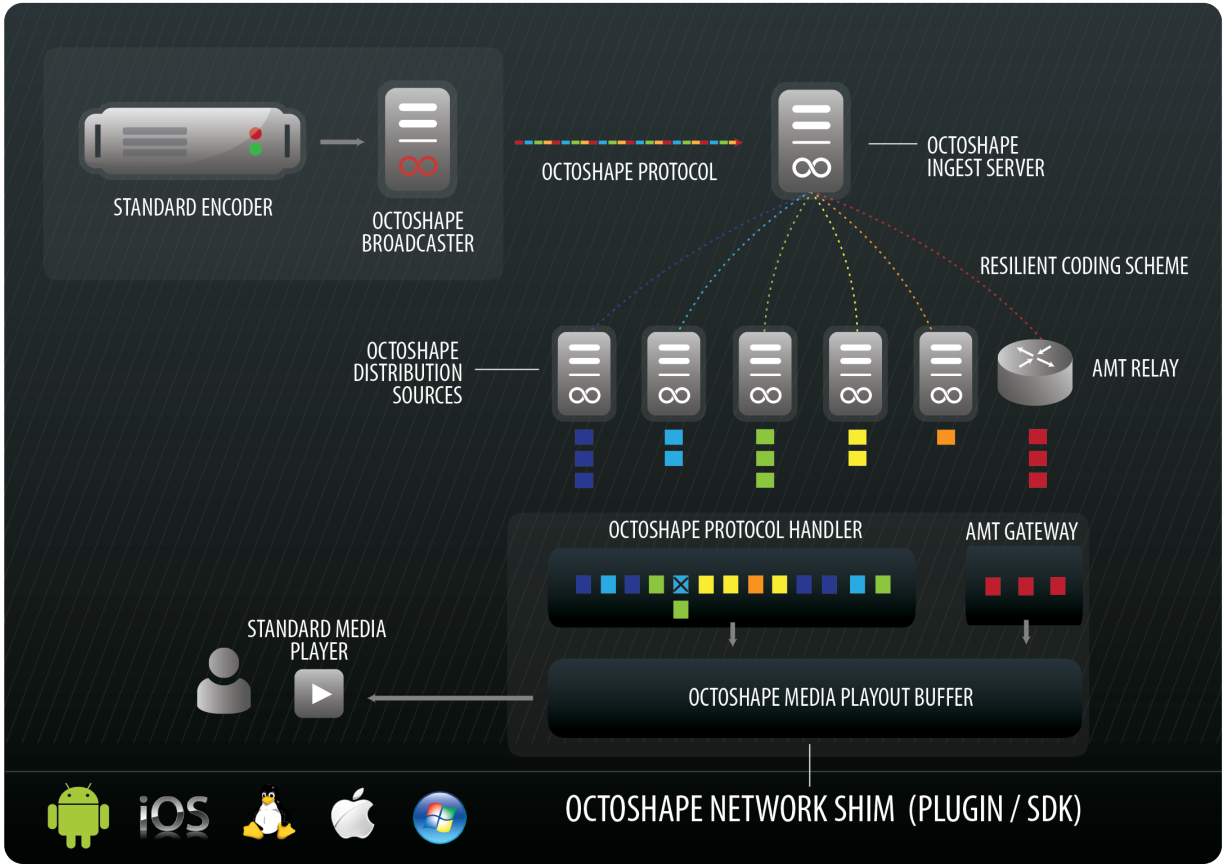
- CNN.Com Live - USA
- NBA League Pass - USA
- Eurovision Song Contest - EMEA
- Digiturk UEFA Football - Turkey
- TV Record Olympics - Brazil
- IPC / ATOS Paralympics – London



Today we see regularly 10k to 100k simultaneous user events over this infrastructure

What is Octoshape

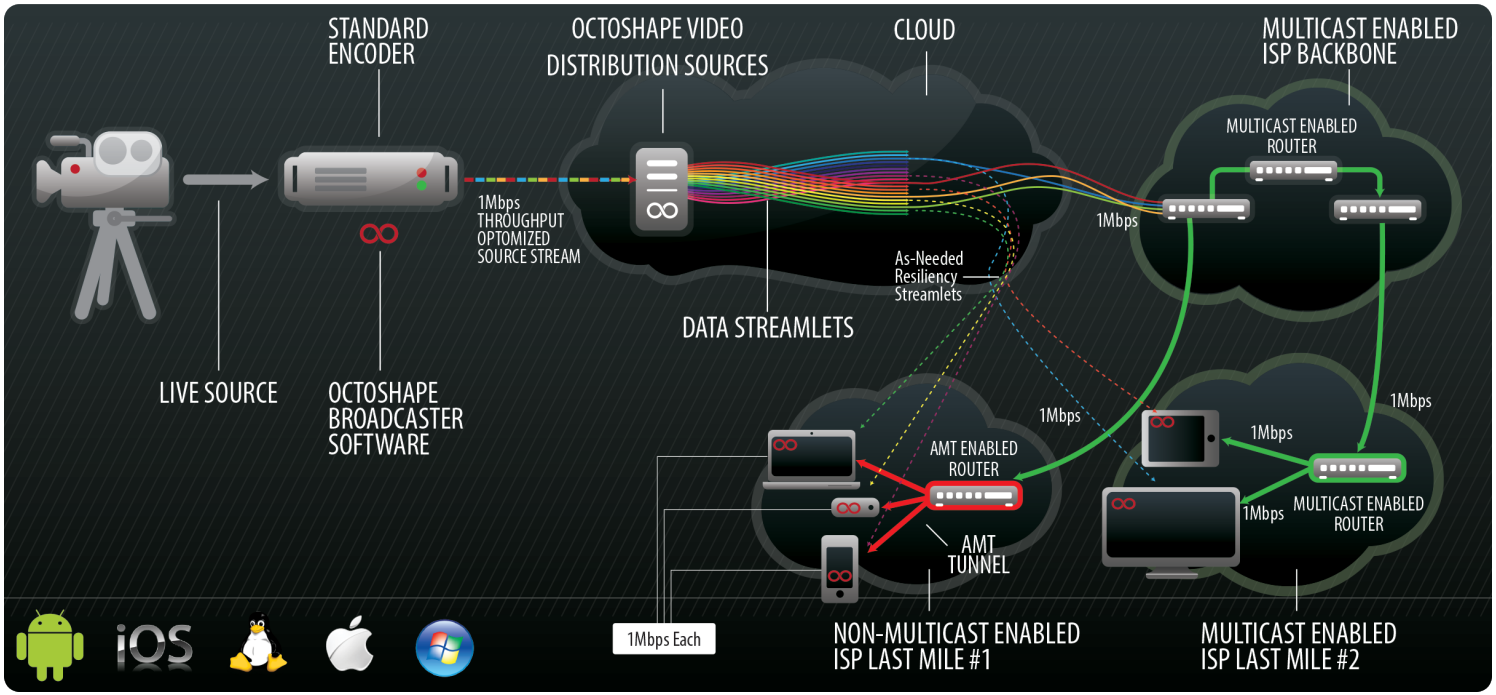
- Streaming media transport layer technology
- Core transport utilizes a UDP + resilient coding scheme like similar to a FEC scheme but without overhead to the client.
- This core transport enables video delivery over best effort networks to perform more like video delivery over provisioned networks



Notes:

- *Deployed against AT&T AMT Relay*
- *Currently testing Juniper AMT Relay*
- *Overall Octoshape Gateway dev/integration took roughly one man month*

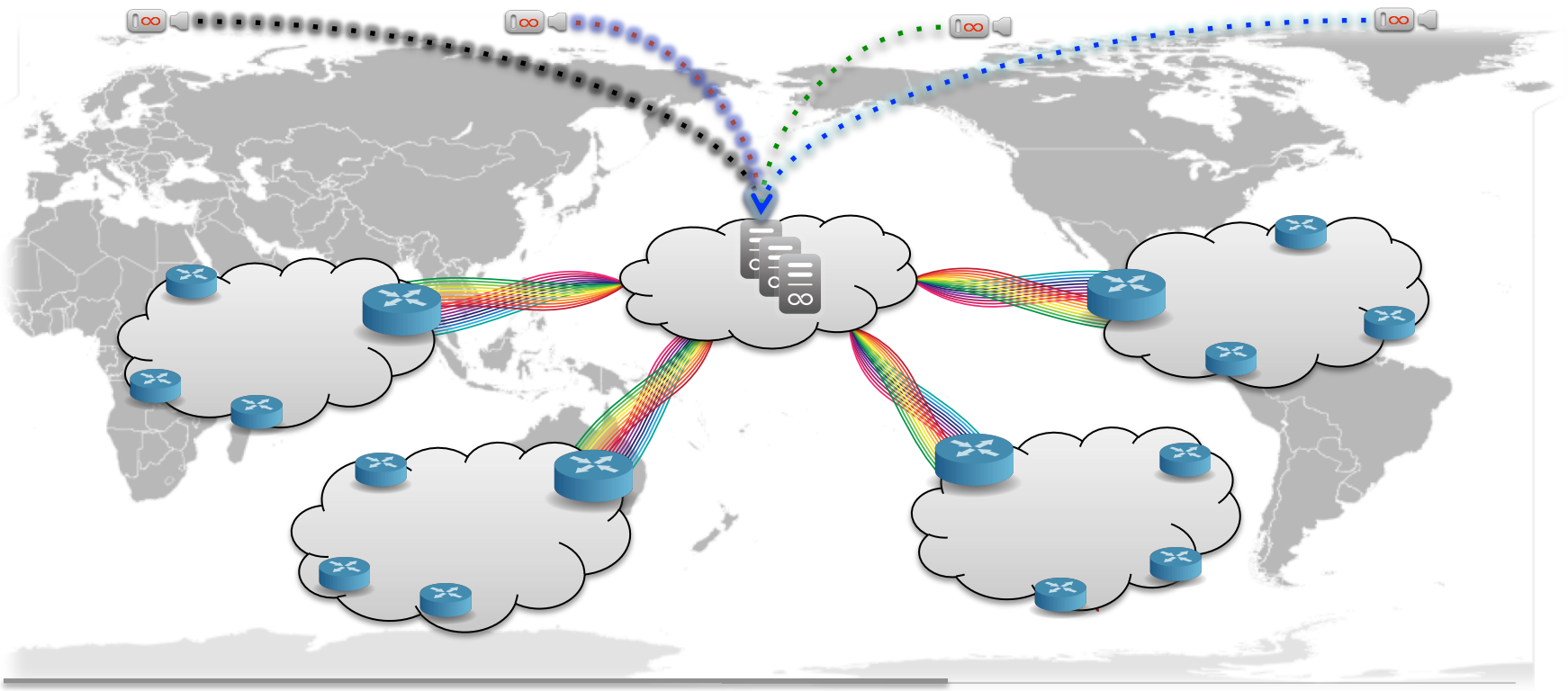
Automatic Multicast Tunneling



The Octoshape client has a AMT Gateway built in across platforms.

- If the client fails to join a Native S,G, it will send an anycast request out to find the closest AMT Relay
- If a valid AMT is found, Octoshape will begin to let go of video sources from the server infrastructure
- If any packets are dropped, the video experience survives due to the Octoshape resilient coding scheme and resilient flows coming from the server infrastructure

Global Multicast Network



- Collect broadcast video from around the world
- Stream globally regardless of multicast or AMT availability
- If a particular ISP (telco, cable operator, satellite or wireless broadband) would like to deploy native or AMT multicast, we build a tunnel and provide multicast reachability to our sources
- If a particular ISP wants to use their own private AMT relay address we will likely use a DNS reverse lookup technique to allow the local domain to overwrite a lookup

AMT Discovery Technique

We have not finalized our implementation here yet, but high level initial thoughts are below:

Step 1: Octoshape client learns of a Native Multicast address with a source address owned by Octoshape

Step 2: IGMP join fails with no Native Multicast reachability

Step 3: Octoshape client does a reverse DNS lookup for amt.”octoshapemulticastsourceip”.in.addr.arpa which will return the AMT relay address that the Octoshape AMT Gateway will attempt a tunnel

This would allow a local ISP to intercept and overwrite requests for amt.*.in-addr.arpa to their own private AMT addressing

Step 3 Alternative: The Octoshape client would look up a standard Octoshape DNS name “amt.octoshape.net” to return the AMT address. This would allow ISP’s to rewrite this address to their own private AMT address

Summary

Octoshape allows content providers to deliver high quality, consistent bit rate video experiences to consumers. We abstract the server layer from the client so it can transparently pull data from servers, native or AMT multicast pumps.

This creates a path to allow a Native or AMT Multicast deployment in good time, at the pace in the network providers control. Octoshape will take advantage of the AMT and Native Multicast as deployed. As it is deployed the more efficiency gained and more deterministic the flows become, all transparent to the consumer.

What Octoshape brings to Multicast:

Inherent Resiliency: Opening up for Multicast over Best Effort Networks

Multi-Bitrate: The Octoshape Multi-bitrate technologies work inherently with Multicast

Multi-Platform: Octoshape is demonstrable on (Windows, Linux, Osx, iOS, Android)

Multicast Domain Traversal: Octoshape can push resilient stream flows over the Internet and inject them into Native Multicast domains globally

Reporting: Octoshape has client side reporting, facilitating byte delivered accounting

Thank You!

White Paper:
http://bit.ly/octoshapesolution