

# Media Transport

M16614

David Singer, Apple

MMT Workshop, London

# Overview

---

- Use cases, needs, preferences
- Critique the past
- Look to the future



# ***USE-CASES, NEEDS, PREFERENCES***

# Use cases etc.

---

- Stored source vs. live
  - One-to-one vs. one-to-many
- Stored: users prefer download
- Live punch-in
- Content Rights Management

# Network Issues

---

- Packet-based, Internet (IP)
- IP Multicast is (very) rare
- Network-address translation, firewalls, caches, proxies, and content replication systems are very common

# Congestion

---

- Don't Blast-and-hope
- Be 'friendly'
- Network operators prefer download

# Data Issues

---

- Source from or...
  - Record/cache to...
  - An MP4 file
  - Carry and handle meta-data
-



# ***CRITIQUE***



# ***MP4 OVER HTTP***

# MP4 issues

---

- Great for download
  - Movie fragments can help
- Cannot be used for punch-in
- Making a fragment that can be played as it arrives takes buffering
- Time-based seeking relies on

---

sample tables



# ***RTP/UDP WITH RTSP/TCP***

# Design Assumptions

---

- Many-to-Many, use multicast
- De-multiplex by port
- Many clocks
- Blast and hope

# Network usage

---

- The IP multicast design issue
- RTP is Port-happy:
  - 2 per stream
  - plus 1 TCP for control
- Uses UDP, not naturally TCP-friendly

# Data issues

---

- Error concealment is codec sensitive
    - Custom payload formats
    - Re-packetization hard
  - Content-sensitivity in network hard
    - Buffer GOPs or handle timing
-

# Clock Recovery

---

- So many clocks:
  - source and destination per-media
  - source and destination reference/system



# ***MPEG-2 TRANSPORT ON UDP***

# Network

---

## ➤ Links

- Bit-serial
- ATM AAL1

## ➤ Framing

- start codes, not length fields

## ➤ Padding to CBR

---

# Data Management

---

- ▶ Streams-on-disc are the file format
  - random access, data layout are all obscure, take work



# ***NEEDS***

# Fit in!

---

- Source
- Distribution
- Delivery and consumption

# Source

---

- Fit into workflows and content prep.
    - Media Preparation
    - Posting and staging
  - Use automatable formats
    - Text for program, session descriptions etc.
-

# Distribution

---

- Co-exist w. NAT, Cache, Firewall
  - Leverage web servers and content distribution networks (CDNs)
  - Use TCP, even HTTP to be network-friendly
  - Prefer storage over bandwidth
-

# Delivery

---

- Use download for stored content, users prefer stalls to errors
- Provide context, leverage the web
- Handle accessibility and user preferences



# ***INDUSTRY ISSUES***

# Good enough or excellent?

---

- To be used, a new solution must be *much* better
- Existing investment in 'non-optimal' often makes it 'optimal'

# A Recent Solution

---

- MPEG-2 Transport streams
    - chunked at GOP boundaries, served over HTTP as separate files
    - Described by an updating textual playlist, served over HTTP
  
  - Easy fit for source, network, user
-

# Conclusion

---

- Fit in
- Get the right assumptions
- Make something so much better, it has traction