Scalable media delivery on the Web with HTTP Server Push

Presentation to W3C Web5G Workshop

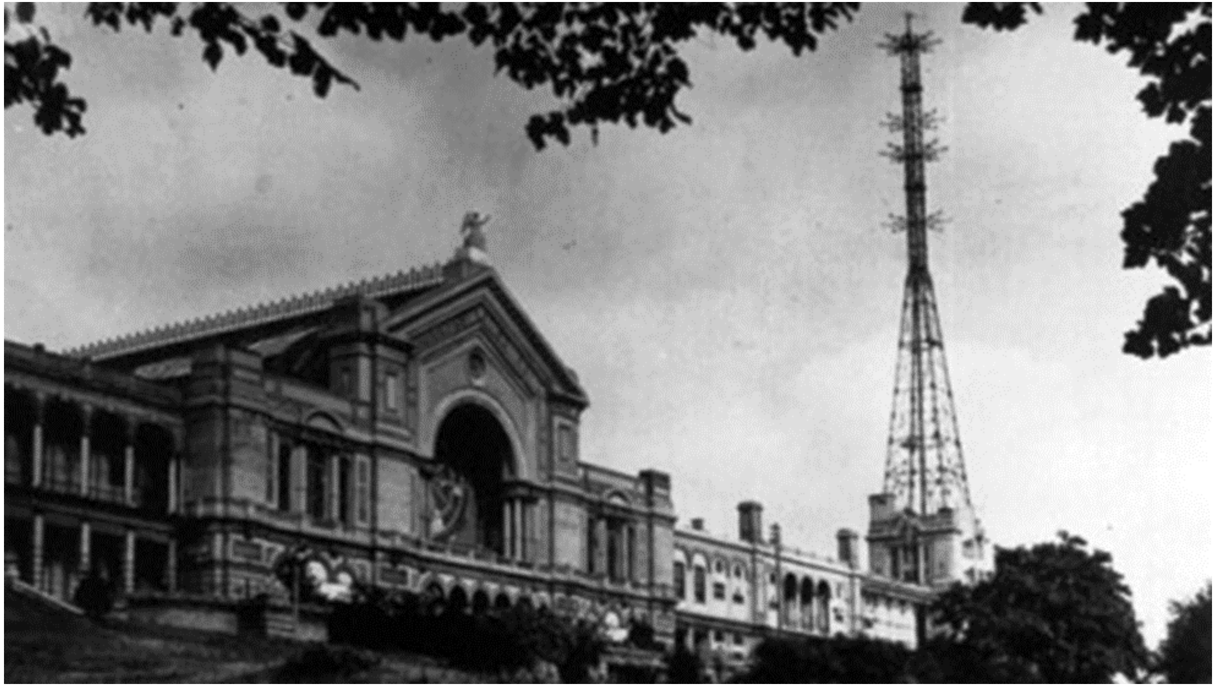
10th May 2018

Lucas Pardue

BBC | Research & Development

Motivation

Life for broadcasters used to be simple…

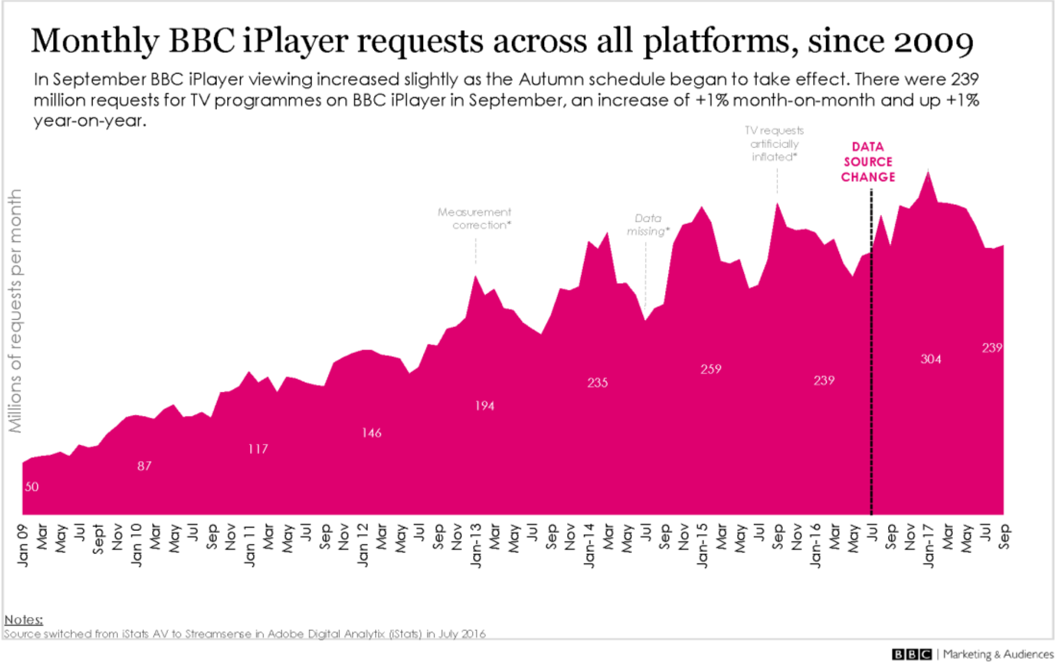


Source: BBC

- Broadcasting from a terrestrial transmitter has a fixed cost.
- The cost doesn't depend on how many people tune in.

The Internet doesn't work like this!

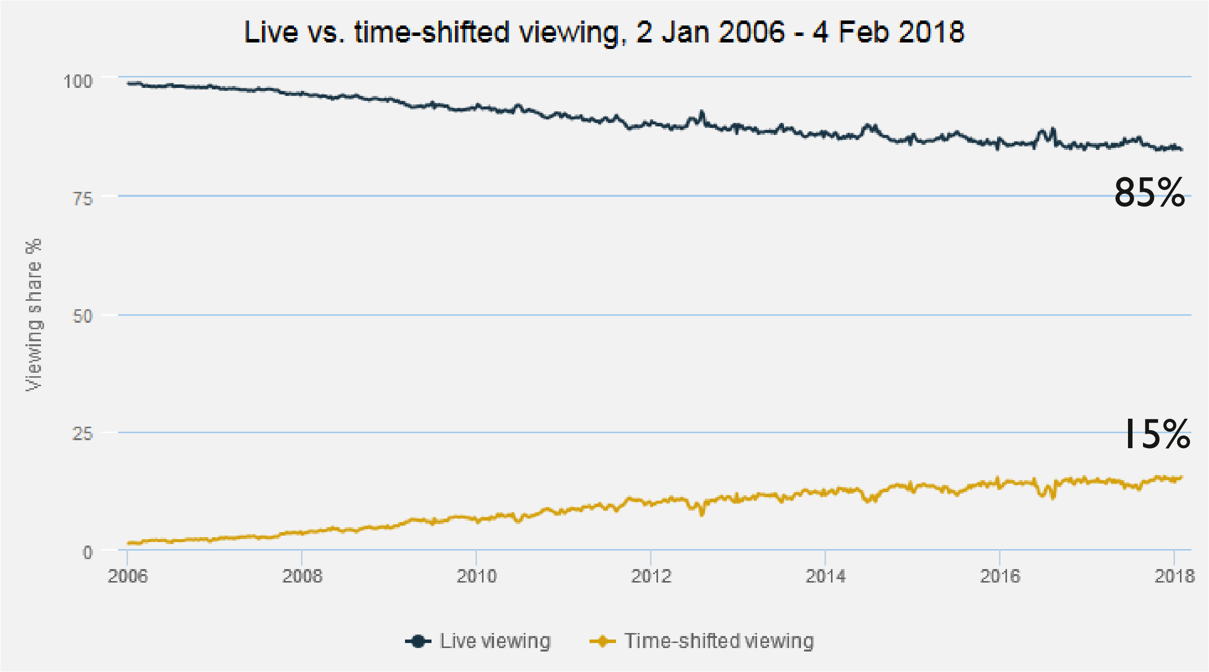
The iPlayer service keeps getting more popular



- Usage of the BBC iPlayer service is climbing steadily.
  - **272M** on-demand programmes requested per month in 2017.
  - First episode of Blue Planet II was requested **4.8M** times.
- CDNs charge per byte delivered by the edge cache.

The cost of providing the service rises in proportion to its popularity.

On-demand viewing is gaining ground…
…but linear viewing still dominates

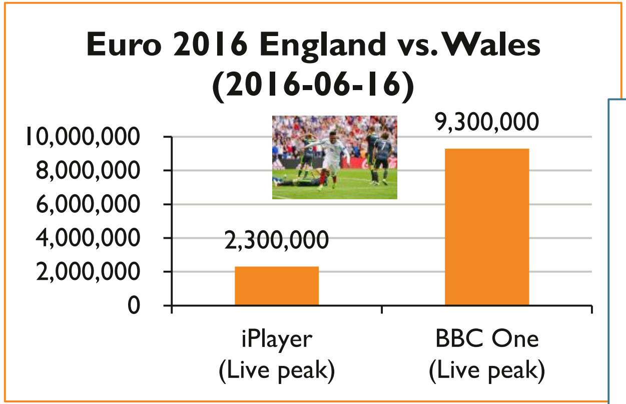


Source: [BARB]

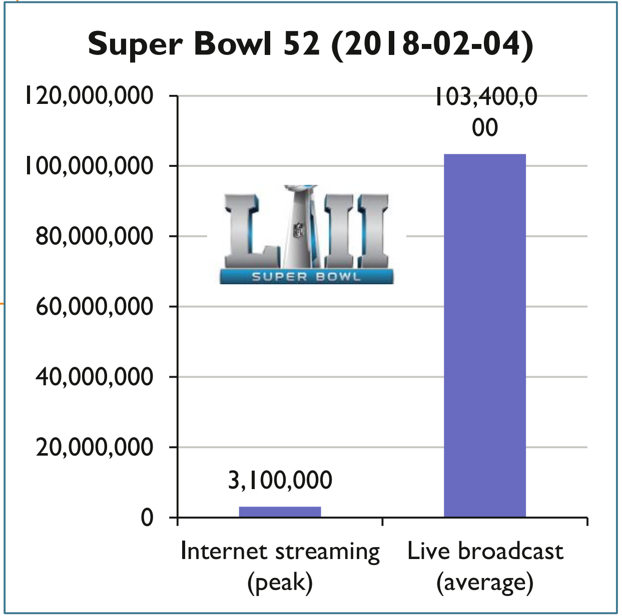
- For the main TV channels in the UK, live viewing represents only **85%** of consumption.
- The remaining **15%** is **DVR time-shifting**, **downloading** and **on-demand streaming**.
  - Time-shifting works best for genres like drama, comedy, entertainment and documentary.

But linear viewing still plays a major role for news, sport and big events.

Linear television is still popular for big events



Source: Radio Times



Source: AdWeek; USA Today

- The BBC's biggest streaming event to date was England vs. Wales in the Euro 2016 competition.
  - **2.3M** simultaneous users.
  - About 20% of total peak.
- Super Bowl 52 was watched by nearly fifty times as many viewers.
  - Streaming represented only 3% of total audience.

The potential audience for linear streaming is huge and scary.

There are more and more bits to shift!

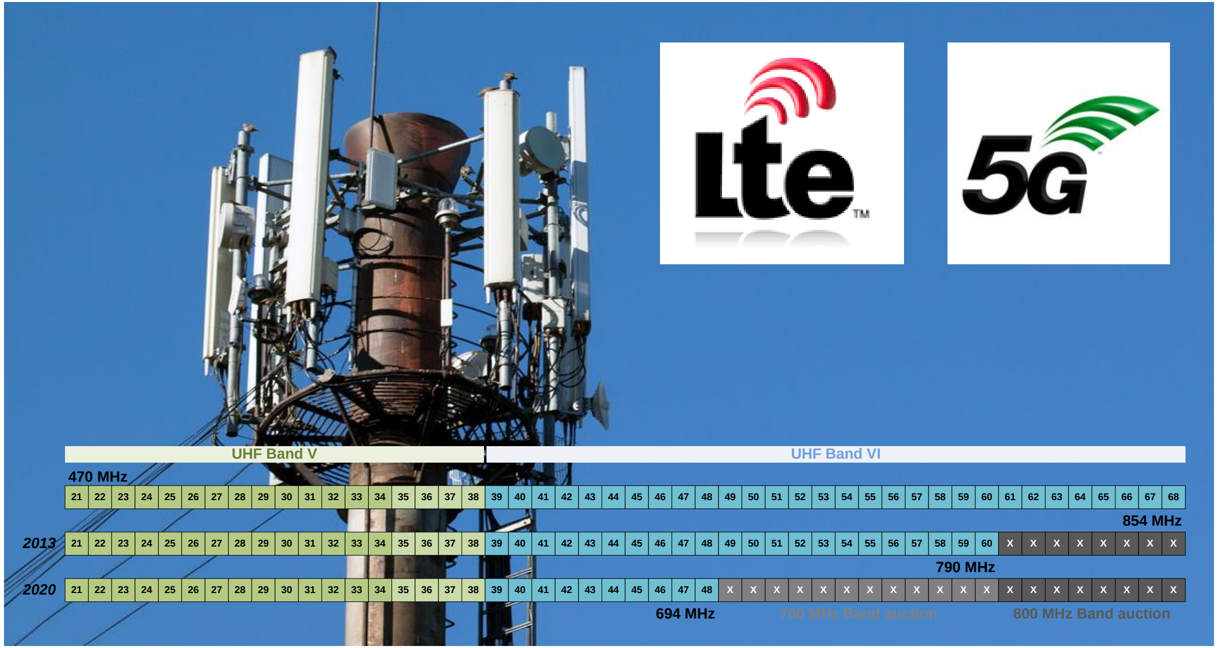
- Higher spatial resolution (SD, HD, UHD).
- Improved colour fidelity (High Dynamic Range).
- Better motion depiction (Higher Frame Rate).

Not to mention:

- New content experiences (3D, 360° Video, AR, VR).
- Next Generation Audio.

All this keeps driving up our CDN distribution costs.

Cellular is biting at our heels



- MNOs are hungry and have deep pockets.
  - 800 MHz band auctioned in 2013 for 4G.
  - 700 MHz band due to be auctioned soon for 5G.
- 3GPP has developed a technology stack called MBMS for media streaming over cellular radio networks.

Our ability to innovate in the broadcast space is now hampered by lack of available spectrum.

The challenge

10 × audience

5 × encoded bit rate

=

50 × load

- How to reach 98% of the UK population without any terrestrial spectrum.

To put that into perspective:

- The last Royal Wedding (April 2011) attracted a total UK live audience of more than 25M across all distribution modes.

Even if CDNs could deliver to that size of audience, we probably couldn't afford to pay for it.

What about IP multicast?

Media streaming over IP multicast

- Layer 3 packet replication reduces redundant transfer.
  - Particularly well suited for large scale audiences.
  - A great way to address the scalability challenge.
- IPTV is deployed around the world today and is supporting pay TV services.
  - Specified by DVB and published as ETSI **TS 102 034**.
  - Works well for managed networks / vertical deployments.
- 3GPP also makes use of IP multicast for its Multicast/Broadcast Multimedia Service (MBMS).

Media streaming over HTTP

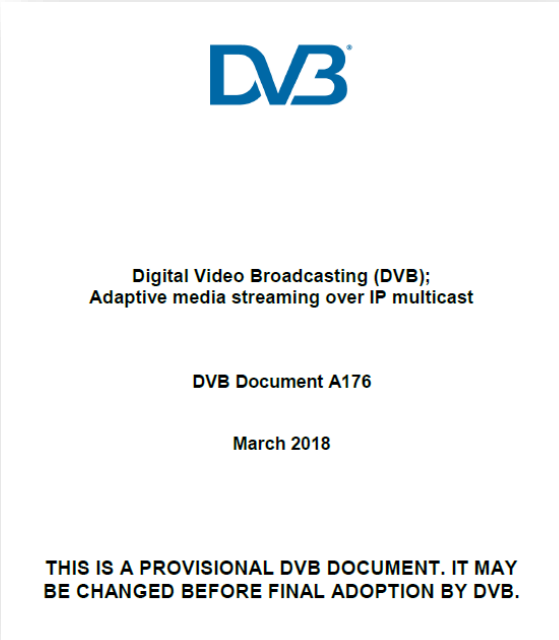
- Industry has embraced unicast HTTP-based streaming.
  - Enables delivery over many types of **access network** (managed and unmanaged), to many types of **end-user equipment**.
  - This is the basis of myriad over-the-top (OTT) Internet streaming services including the BBC iPlayer.
  - Success driven by use of **existing network technologies** and **Content Delivery Networks** (CDN).

Adaptive media streaming over unicast HTTP

- Divide media stream into short duration **segments.**
- Each segment is encoded at several bit rates.
- Client-side adaptation between these encodings, to suit local network conditions.
  - Graceful changes with respect to network degradation or improvement.
- Several proprietary streaming technologies exist:
  - Microsoft Smooth Streaming.
  - Adobe HDS.
  - Apple HLS.
- Industry aligning toward **MPEG-DASH**.

Standardisation of adaptive media streaming over IP multicast

- CableLabs specification for ABR Multicast on CATV networks.
- 3GPP MBMS specification for enhanced television (enTV) on cellular mobile networks.
- DVB specification on adaptive media streaming over IP multicast.
  - Commercial requirements driven by use cases.
  - Technical work undertaken by TM-IPI ABR Multicast task force led by Richard Bradbury, BBC R&D.
  - **DVB BlueBook A176**, released March 2018.


DVB

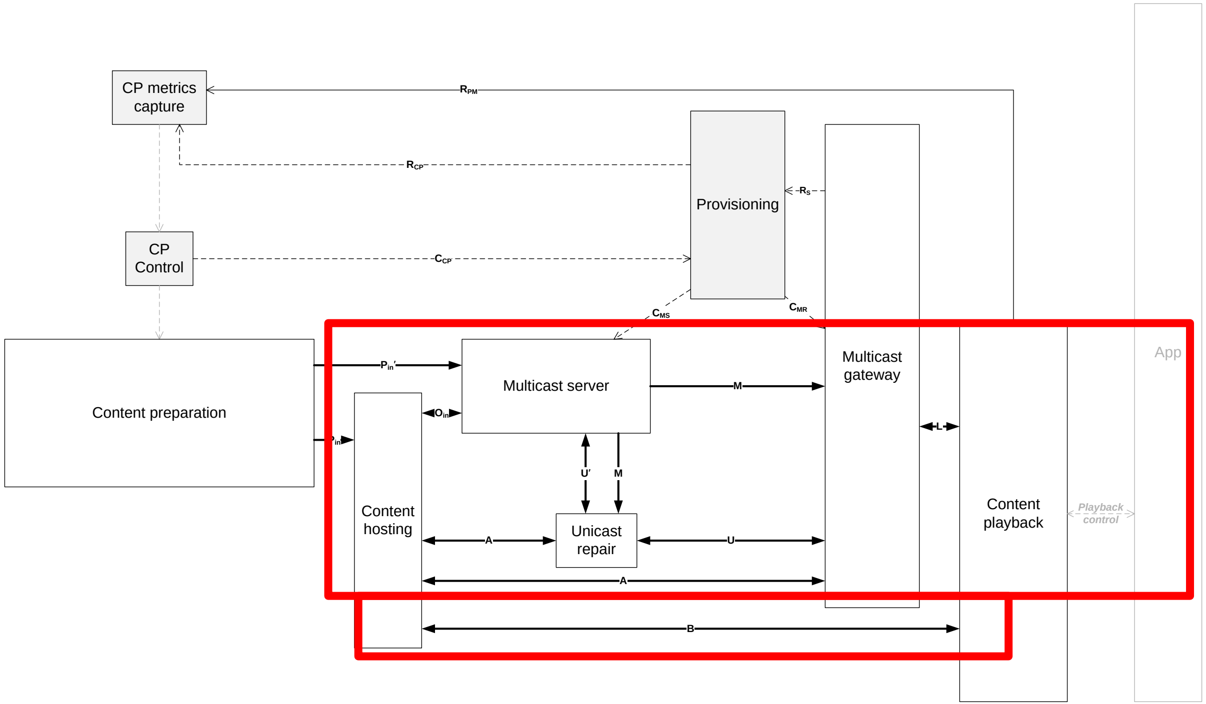
Digital Video Broadcasting (DVB);
Adaptive media streaming over IP multicast

DVB Document A176

March 2018

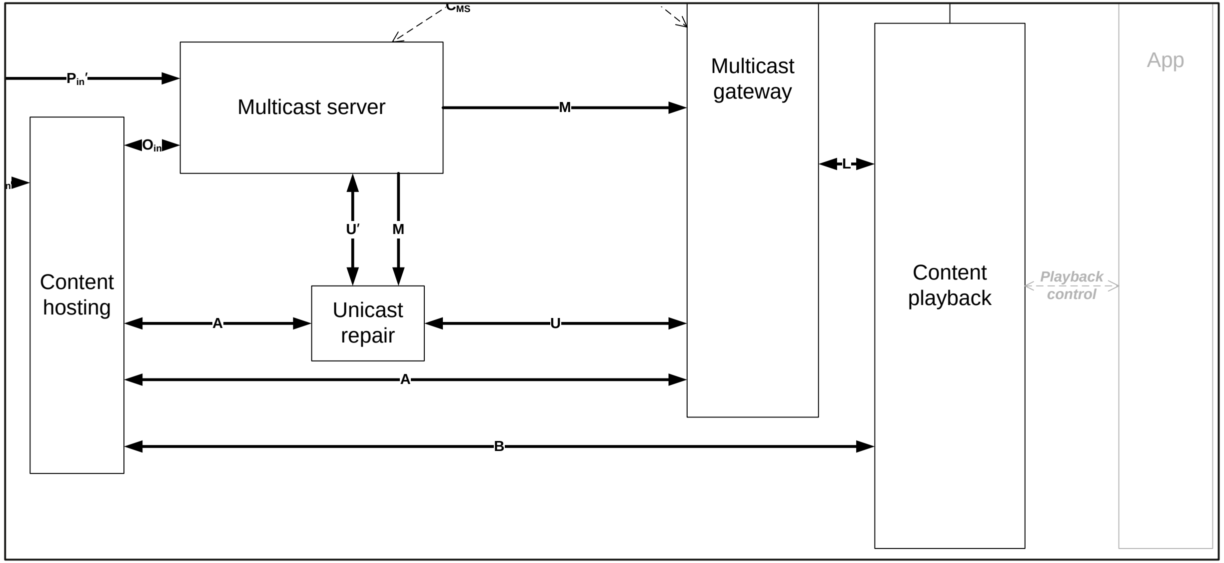
THIS IS A PROVISIONAL DVB DOCUMENT. IT MAY BE CHANGED BEFORE FINAL ADOPTION BY DVB.

DVB reference architecture for adaptive media streaming over IP multicast



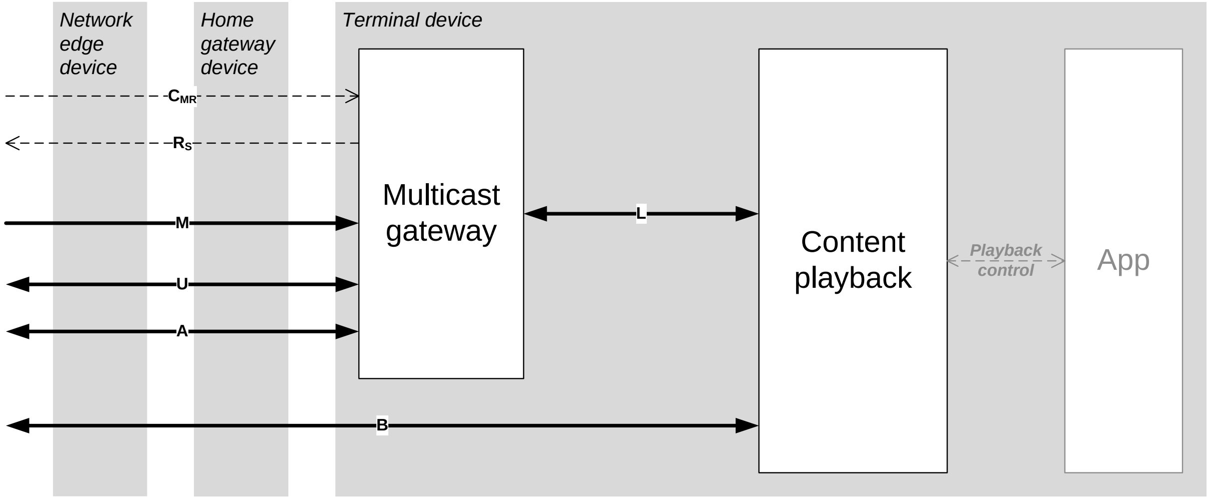
- Abstract reference, **functions** and **interfaces**.
  - Implementations may combine multiple functions in a single deployable unit.
- Conventional unicast case
- Unicast-to-multicast and back again
- Maintains technical compatibility with existing reception equipment.

Conversion between multicast and unicast in the data plane



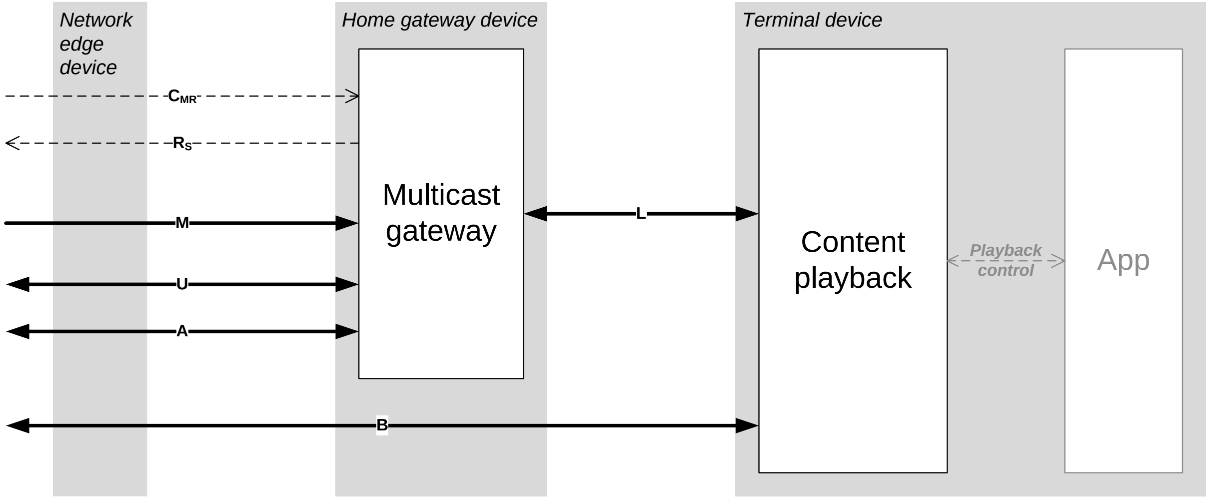
- The **Multicast server** function converts from unicast (**O**$_{in}$) to multicast (**M**) at the sending end.
- The **Multicast gateway** function performs the inverse conversion from multicast back to unicast.
- This is presented to the **Content playback** function via reference point **L**.
  - This is a conventional MPEG-DASH player.

Deployment model 1: Multicast gateway deployed in terminal device



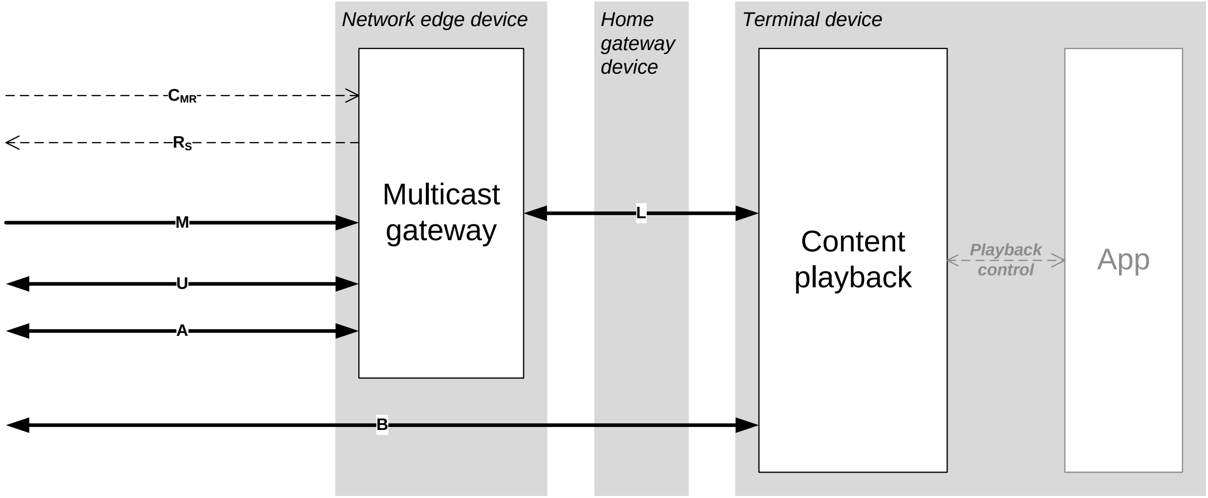
- Terminal device supports reception of IP multicast
  - e.g. **MBMS Client** in mobile phone.
- Multicast gateway and Content playback functions all in the terminal device.
- Relies on support for multicast in the core network.

Deployment model 2: Multicast gateway deployed in home gateway device



- Multicast gateway located on Customer Premises Equipment
  - e.g. home router supplied by the ISP.
- Multicast gateway can serve several users in the same home network.
- Suited more to fixed line deployments.

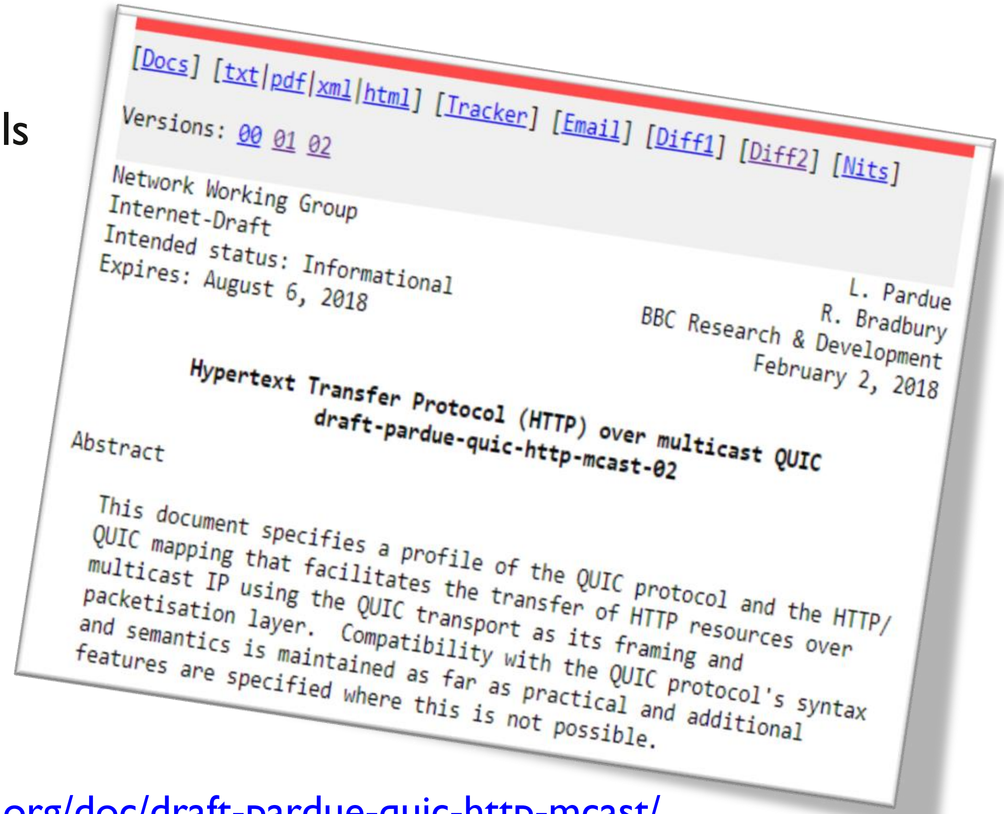
Deployment model 3: Multicast gateway deployed in network edge device



- Multicast gateway deployed in a network edge device.
  - e.g. MEC server node.
- Multicast gateway can serve several users in the same network service area.
  - e.g. all mobiles within a cluster of cell sites.
- Suitable when terminal device cannot receive IP multicast.
  - All traffic on access network is unicast.

HTTP over multicast QUIC

- An independent **Internet Draft** that fills some gaps between IP unicast and multicast.
  - Based on a new transport protocol called **QUIC**.
- Describes a means of service discovery using **HTTP Alternative Services** [RFC 7838].
- Bulk file delivery that intentionally supports a broad range of Use Cases.



[Docs] [txt|pdf|xml|html] [Tracker] [Email] [Diff1] [Diff2] [Nits]

Versions: 00 01 02

Network Working Group
Internet-Draft
Intended status: Informational
Expires: August 6, 2018

L. Pardue
R. Bradbury
BBC Research & Development
February 2, 2018

Hypertext Transfer Protocol (HTTP) over multicast QUIC
draft-pardue-quic-http-mcast-02

Abstract

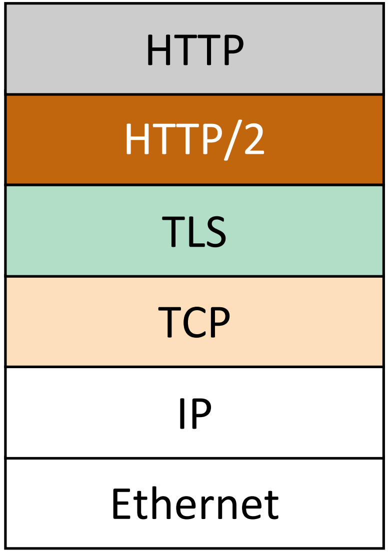
This document specifies a profile of the QUIC protocol and the HTTP/QUIC mapping that facilitates the transfer of HTTP resources over multicast IP using the QUIC transport as its framing and packetisation layer. Compatibility with the QUIC protocol's syntax and semantics is maintained as far as practical and additional features are specified where this is not possible.

https://datatracker.ietf.org/doc/draft-pardue-quic-http-mcast/

Server Push for unidirectional HTTP flows

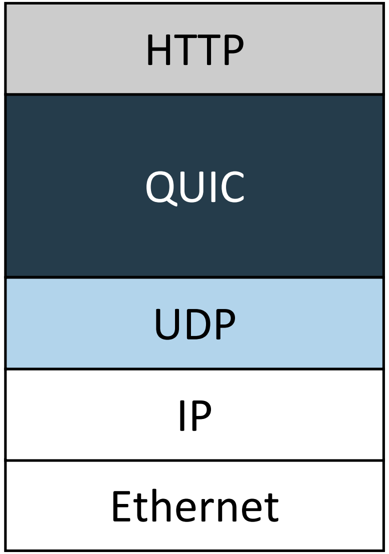
HTTP/2

- IETF RFC 7540 published in 2015.
- Maintains existing HTTP semantics…
  - Request/response.
  - Metadata (Header fields) and payload.
  - Caching.
  - Validation.
- …but switches from a text to a binary syntax.
  - Request and response messages are split into **Frames**.
    - **HEADERS** frames carry metadata.
    - **DATA** frames carry payload.
  - Frames are carried on logical **Streams**.
    - Allows multiplexing of requests and responses in a single TCP connection.
- Introduced **Server Push** protocol primitive.



QUIC

- A UDP-based multiplexed and secure transport.
- Initial experiments by Google.
- Adopted by IETF for standardisation.
  - Specification happening in QUIC Working Group.
  - Family of specifications due for delivery by end of 2018.
- Concepts of Streams and Frames pulled into the transport layer.
- Google defined only HTTP/2 over QUIC.
  - IETF allows multiple application protocols to be mapped.
  - HTTP over QUIC mapping is a separate IETF specification and differs from HTTP/2.
- QUIC also has Server Push.



HTTP Server Push

- Server can speculatively send to a client data that it anticipates the client will need.
  - Effectively multiple responses to a single request.
  - Intended to improve web page performance.

- After receiving a request, a server can:
  1. Send a `PUSH_PROMISE` frame.
     - A pseudo-request with metadata (payload not permitted).
  2. Send a `HEADERS` frame.
     - Response metadata with HTTP header fields.
  3. Send `DATA` frame(s).
     - Response payload.

- Server Push is bound to a parent request.
- "Unbound Server Push" draft-pardue-quic-http-unbound-server-push.
- Proposed extension that removes binding and uses QUIC unidirectional streams.

Server Push for unidirectional flows

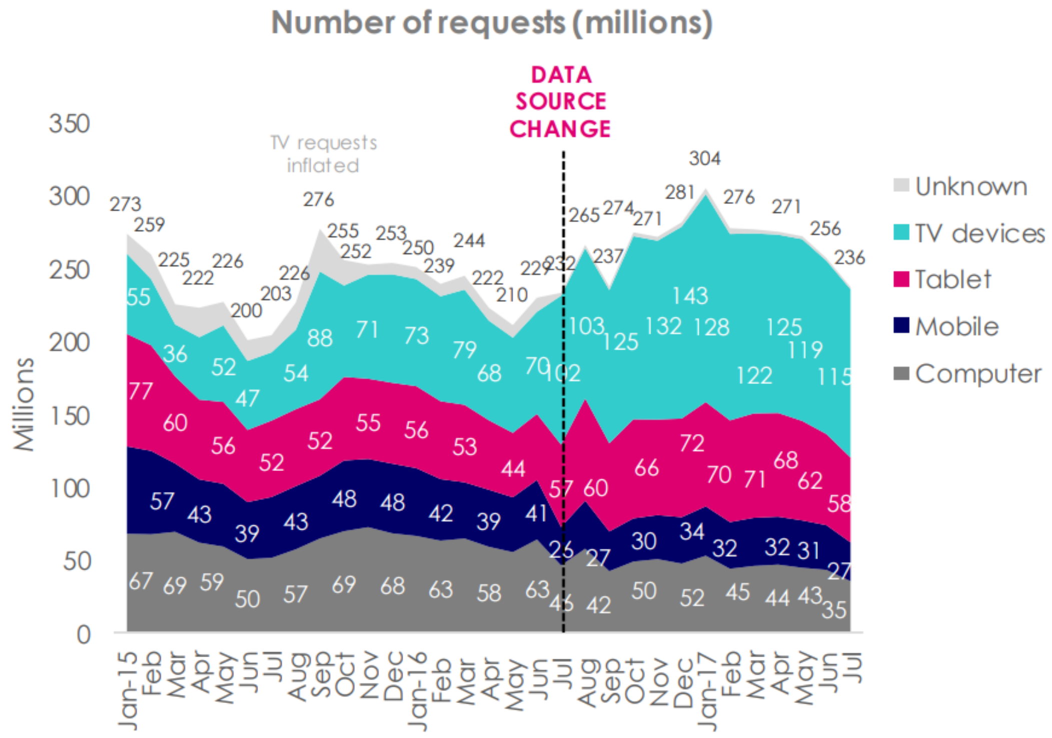
- Server Push is a HTTP primitive.
- Resources identified by web-friendly **URLs.**
  - Metadata is carried in a standard, well understood format.
- Well adapted to unidirectional HTTP flows carrying any form of content.
  - Our experimentation on embedded clients has proven this.
- We think Server Push is a strong candidate for Web application new use cases
  - It bears similarity to existing APIs, so what is different?

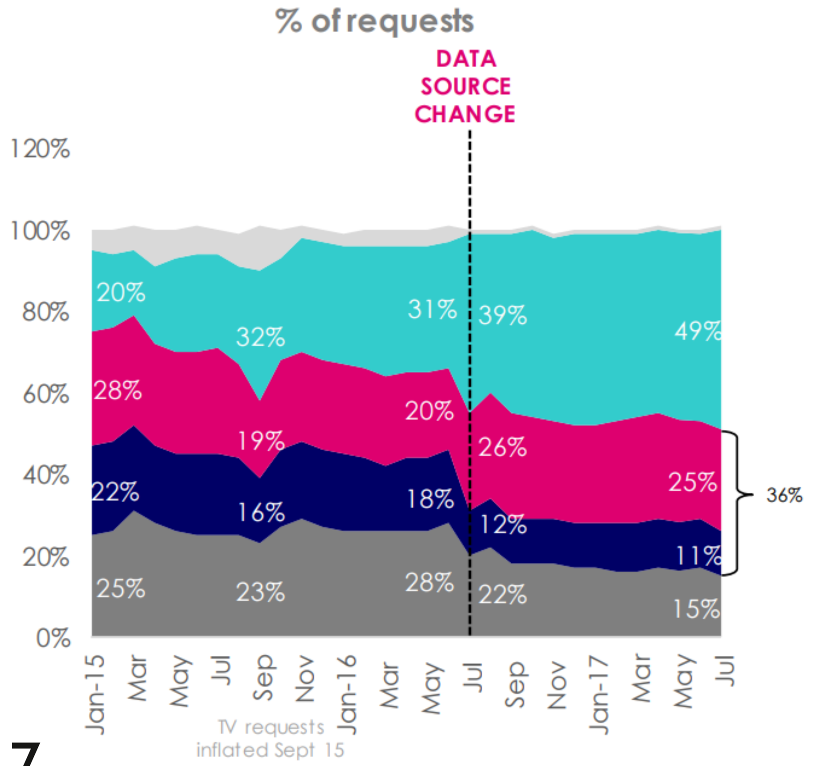
Server Push in comparison to similar APIs

<table>
<tr><th></th><th>URL visibility</th><th>Directionality</th><th>Distribution model</th><th>Supported payload types</th></tr>
<tr><td>Server Push</td><td>Native</td><td rowspan="3">Unidirectional</td><td rowspan="4">Centralised</td><td>Any permitted type</td></tr>
<tr><td>Server Sent Events</td><td rowspan="4">Mapped/ encapsulated</td><td>Text only</td></tr>
<tr><td>Push API</td><td rowspan="3">Any permitted type</td></tr>
<tr><td>WebSockets</td><td rowspan="2">Bidirectional</td></tr>
<tr><td>RTCDataChannel</td><td>Peer-to-peer</td></tr>
</table>

Multicast HTTP/QUIC to the Browser

Requests by device type

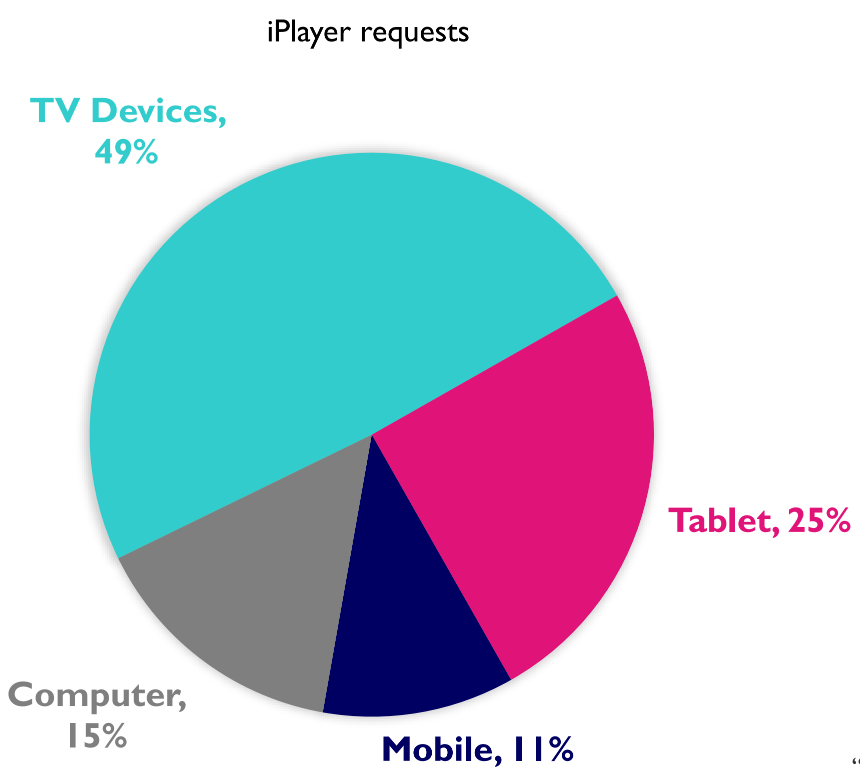


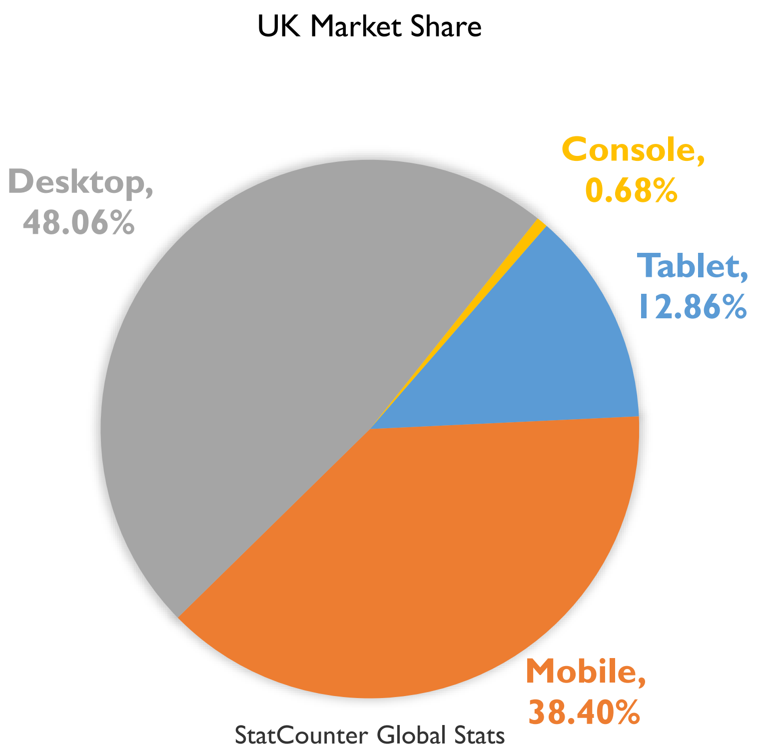


July 2017

iPlayer Performance Pack,

Apples to oranges



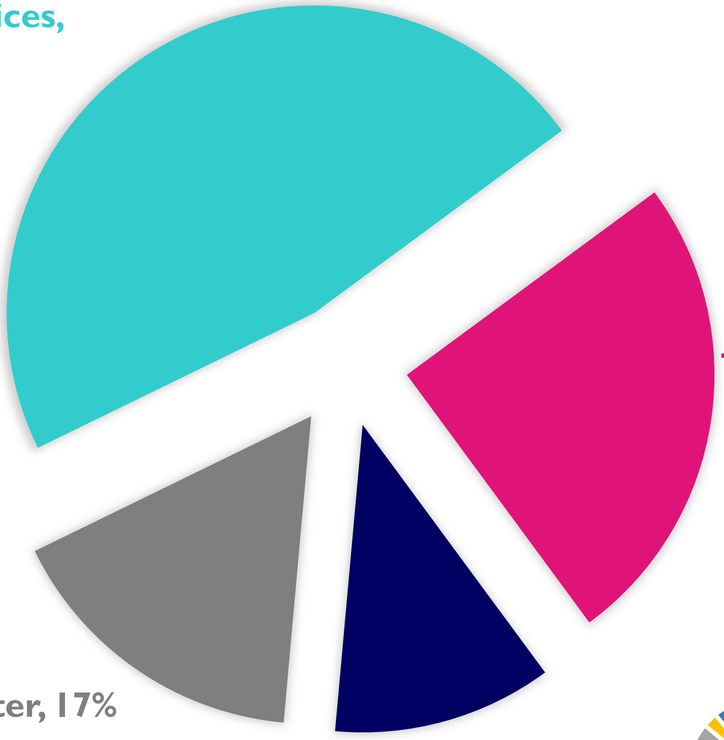


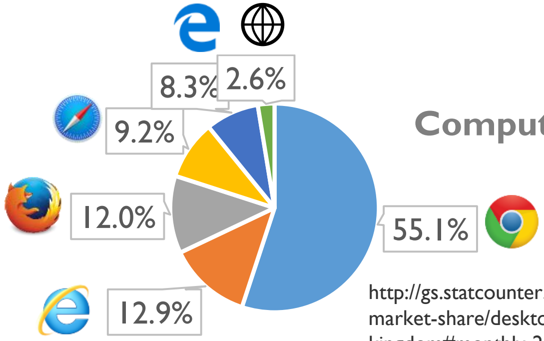
StatCounter Global Stats

"stats are based on over 15 billion page views per month recorded across more than 2.5 million websites"

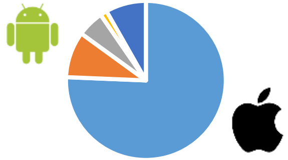
http://gs.statcounter.com/platform-market-share/all/united-kingdom/#monthly-201707-201707-bar

Requests by device type

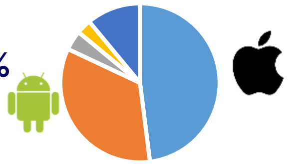




http://gs.statcounter.com/browser-market-share/desktop/united-kingdom#monthly-201706-201707

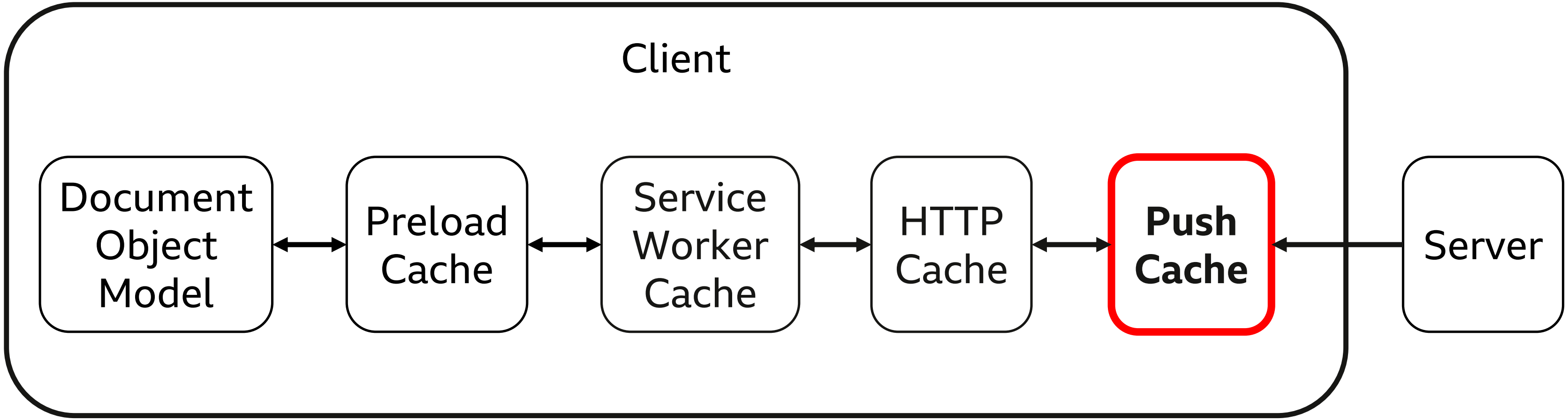


https://www.statista.com/statistics/487815/market-share-tablet-device-vendors-uk/



https://www.statista.com/statistics/387227/market-share-of-smartphone-manufacturers-in-the-uk/

Opaque Server Push



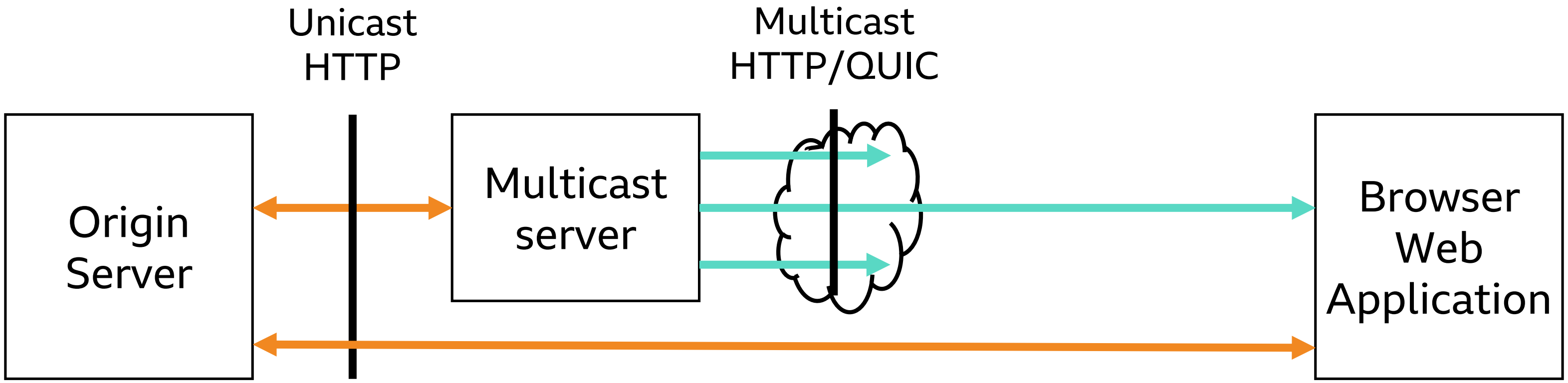
Server Push API proposal

- Alignment with the W3C Fetch API.
- Usable in many contexts.

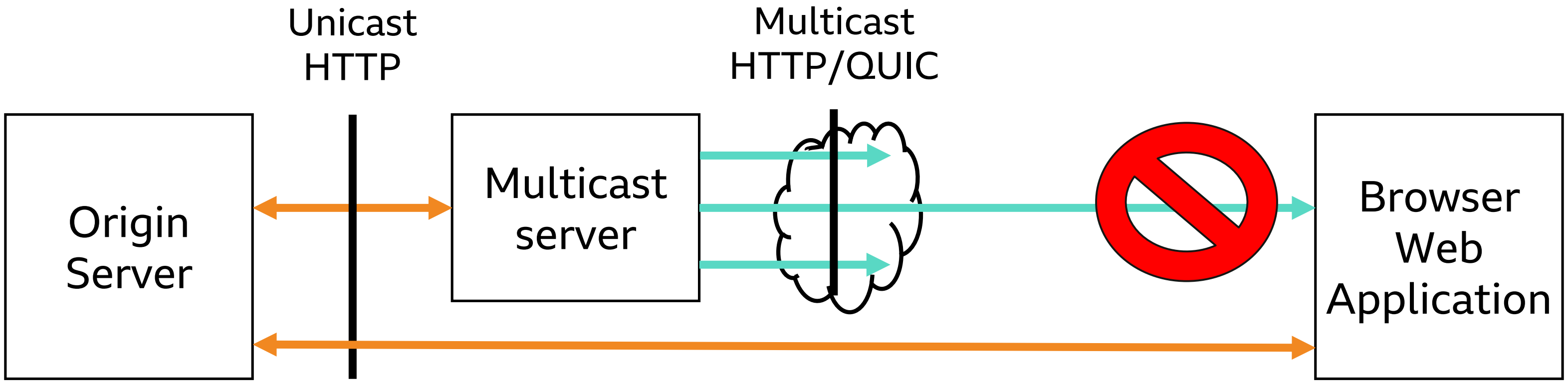
```
interface ResourcePushEvent : Event {
  readonly attribute Request request;
  readonly attribute Promise<Response> response;
}

dictionary ResourcePushEventInit : EventInit {
  required Request request;
  required Promise<Response> response;
}
```

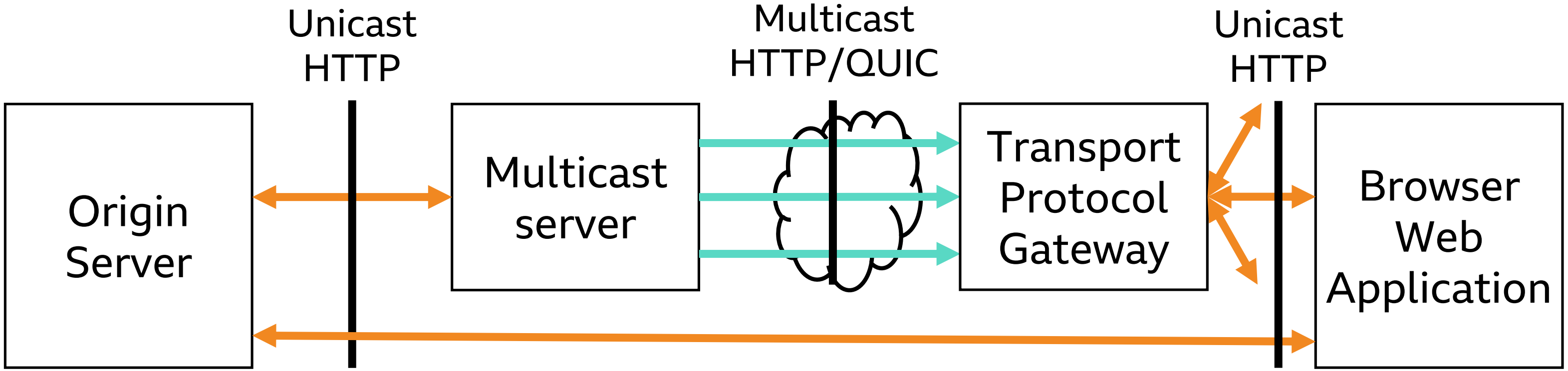
Browser web application



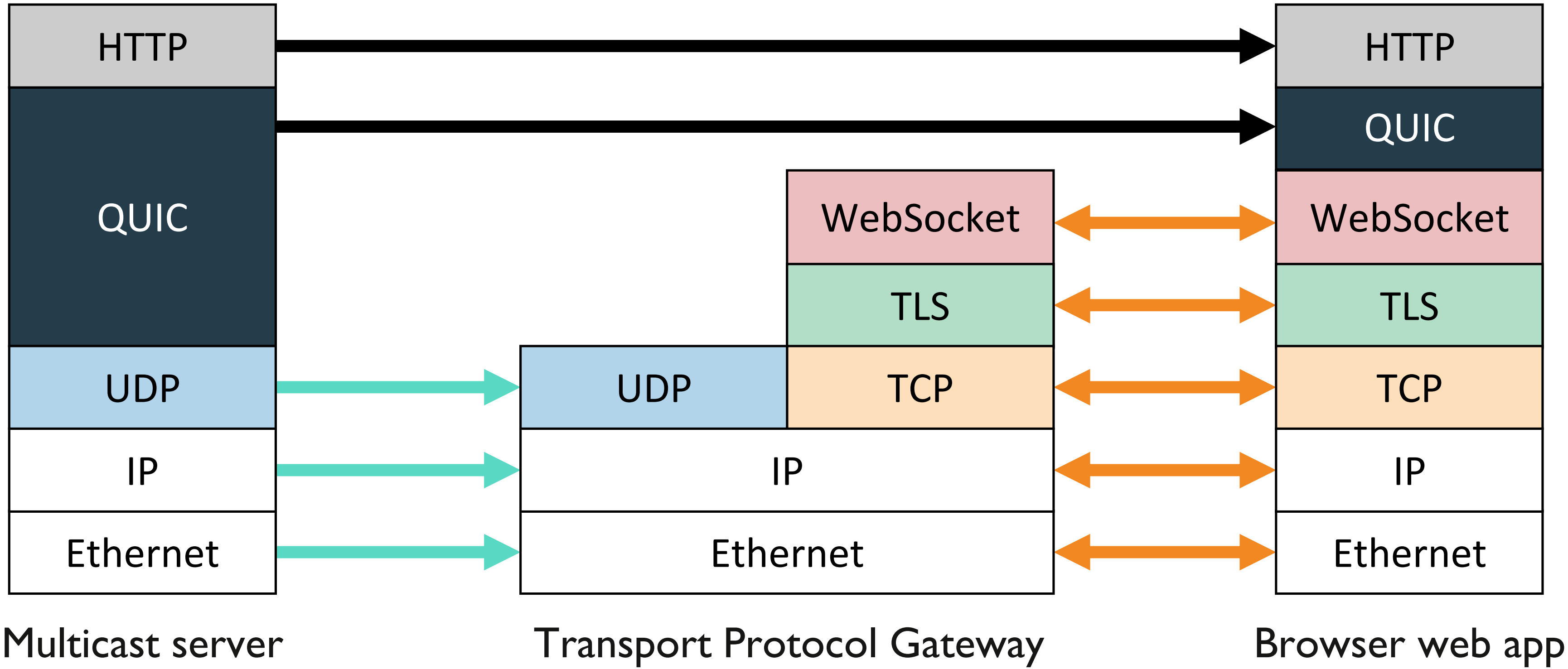
Browser web application



Transport Protocol Gateway (TPG)



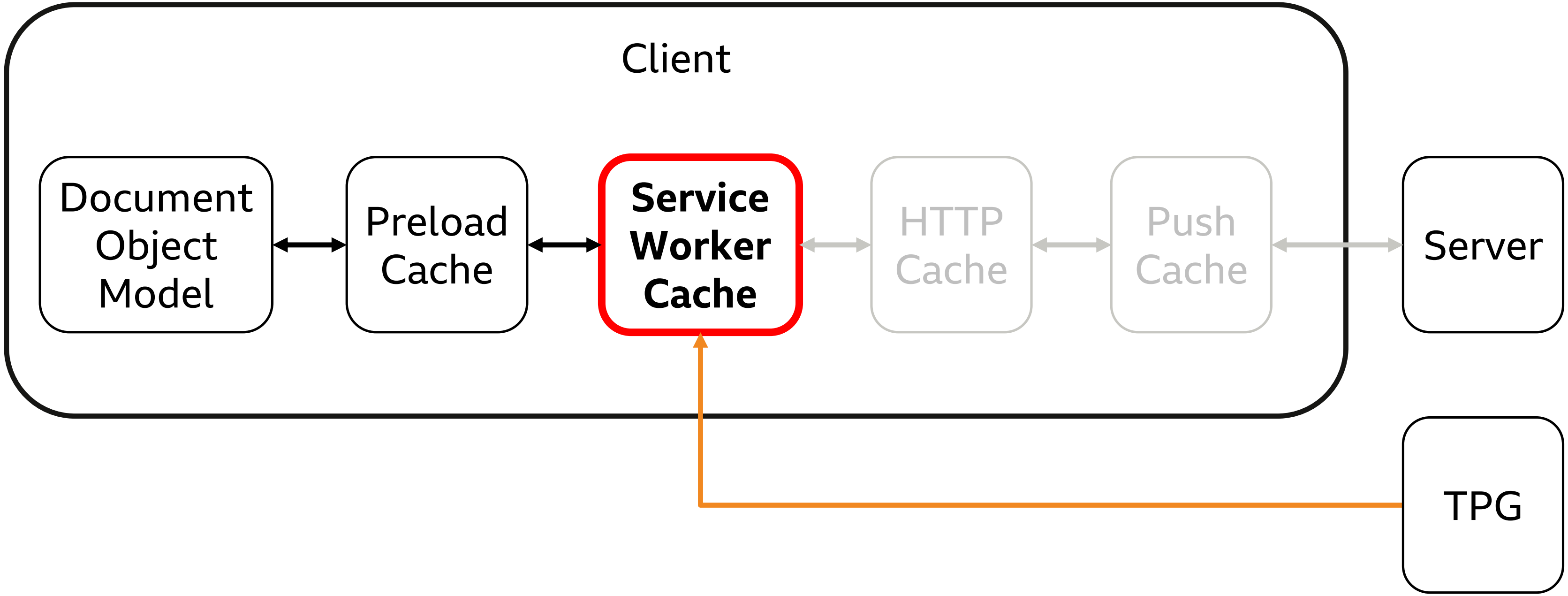
TPG layering



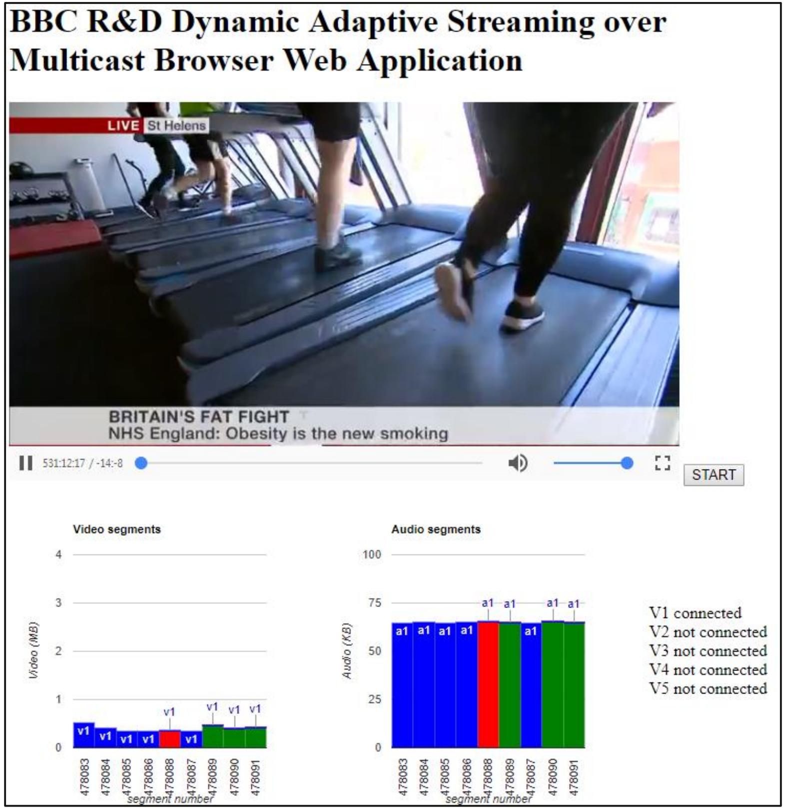
HTTP/QUIC deserialisation

- Delivering QUIC packets into the Web Application.
- We need to deserialise them back into HTTP messages.
  - No support from the browser core to do this.
- Port our existing C/C++ implementation to the browser environment using **emscripten.**
  - Generates performant code in the form of **WebAssembly.**
- JavaScript "glue" layer to allow interaction with web applications.
  - Server Push API (`ResourcePushEvent`).
  - Expose an `enqueuePacket()` function that lets us inject QUIC packets.

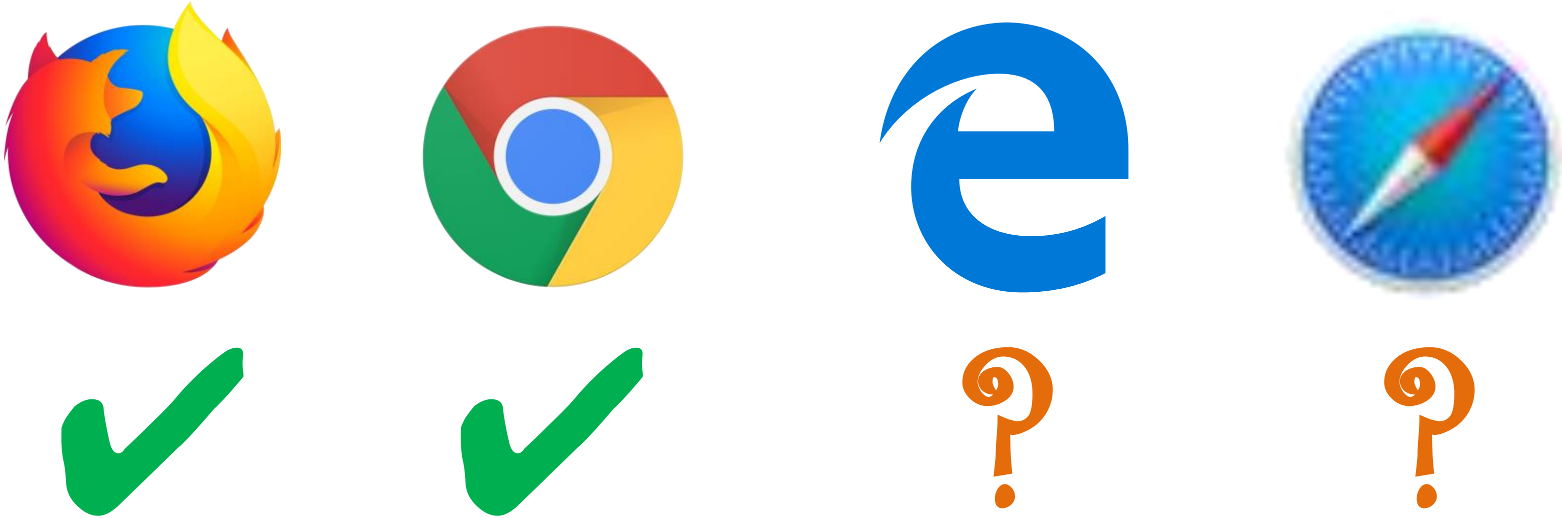
Service Worker cache



Browser web application in action



Browser support



Scalable media delivery on the Web with HTTP Server Push

- Public White Paper on these topics
- https://www.bbc.co.uk/rd/publications/whitepaper336

Scalable media delivery on the Web with HTTP Server Push

Lucas Pardue

2018 Apr

White Paper WHP 336 Apr 2018

Download

Abstract

Version 2 of the Hypertext Transfer Protocol (HTTP/2), published in 2015, introduced a Server Push protocol primitive, offering the possibility of improving user-perceived web page performance through the unsolicited delivery of additional dependent resources by a web server in response to an initial request from a client. Analysis of deployments in the wild has shown actual performance gains to vary greatly, and even reduced performance in some cases.

This paper explores Server Push from the perspective of Web Application logic implemented using JavaScript and the Web Platform. Current web browsers hold pushed resources in a so-called “Push Cache”, where they exist in effective purgatory until explicitly requested. Server Push is, in essence, hidden from the Web Application.

We contend that the failure to expose Server Push events to the Web Application layer has impeded the realisation of promised performance improvements. Furthermore, hiding Server Push restricts a new set of use cases that would benefit from a reactive approach to web-oriented HTTP delivery of resources, in particular, unidirectional flows such as long-lived bulk data delivery and low-latency delivery.

Conclusion

- We think that the future of linear television distribution is adaptive media streaming over the Internet using a mixture of unicast CDNs and IP multicast.
  - The two modes need to work hand-in-hand with each other to give a seamless user experience.
  - HTTP provides a common Layer 7 to achieve this seamlessness.
  - QUIC provides a common Layer 4 packet syntax.
  - Alt-Svc supports discovery of multicast from unicast.
- We have working end-to-end prototypes that demonstrate these principles.
- The Web Platform faces challenges in supporting new content delivery mechanisms.
  - Particularly when considering secure context requirements.

Thank you

bbc.co.uk/rd

Email:
lucas.pardue@bbc.co.uk

Twitter
@SimmerVigor

BBC | Research & Development