Multicast for the Web

W3C Video Interest Group, 2021-04
Jake Holland, Akamai

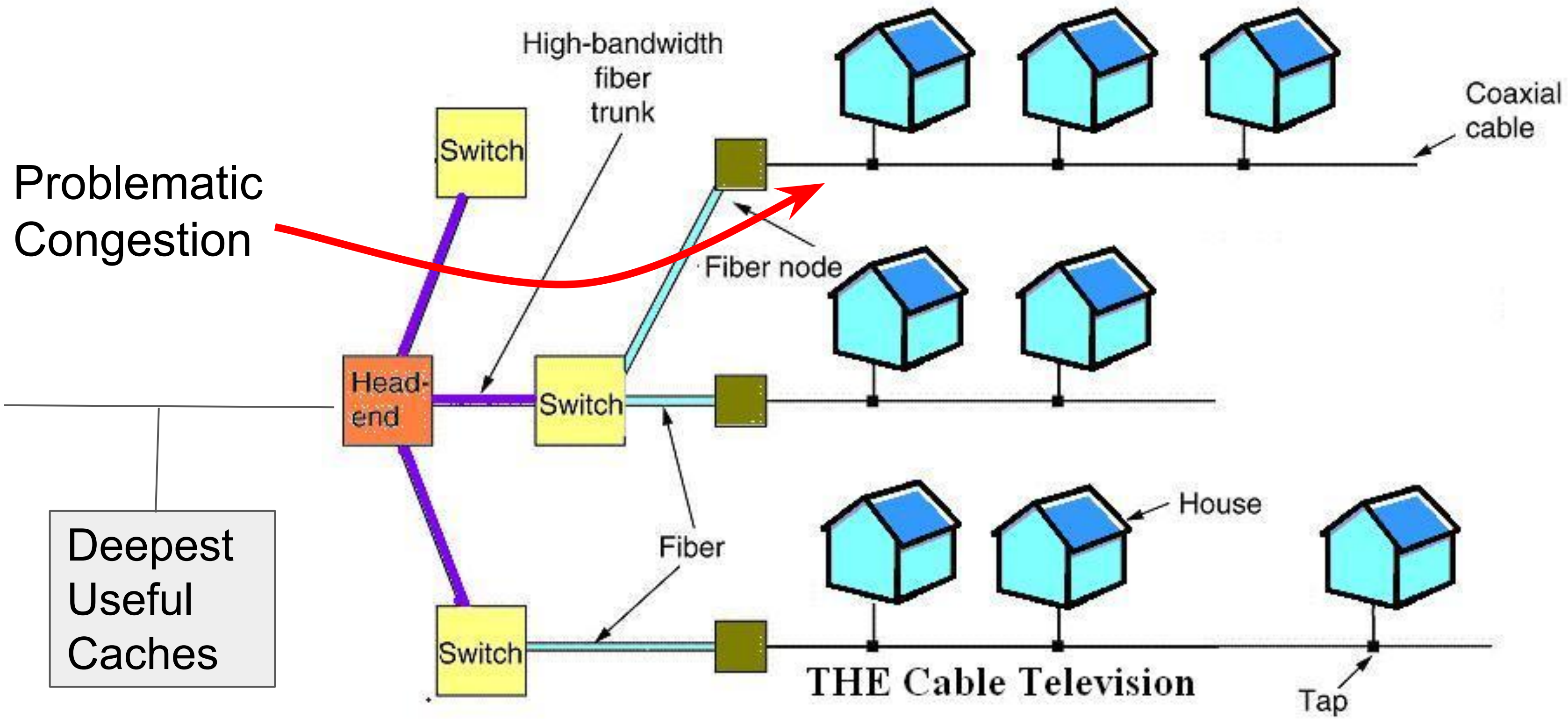
Outline

- What multicast means
- Why it’s useful
- Proposed Web API & Status
- Early Feedback & Next Steps
    - Segue to discussion

What Multicast Means

- Channels joined by IGMP or MLD from end user devices
- Individual IP packets delivered one-to-many
  - Replicated by network (or sent on broadcast link)
  - Identical payloads for all subscribers to same channel
  - No in-band 2-way communication
    - But: individualized out-of-band TLS to supplement is possible
      - E.g. for crypto anchors

Key Problem Solved: Access Network Congestion



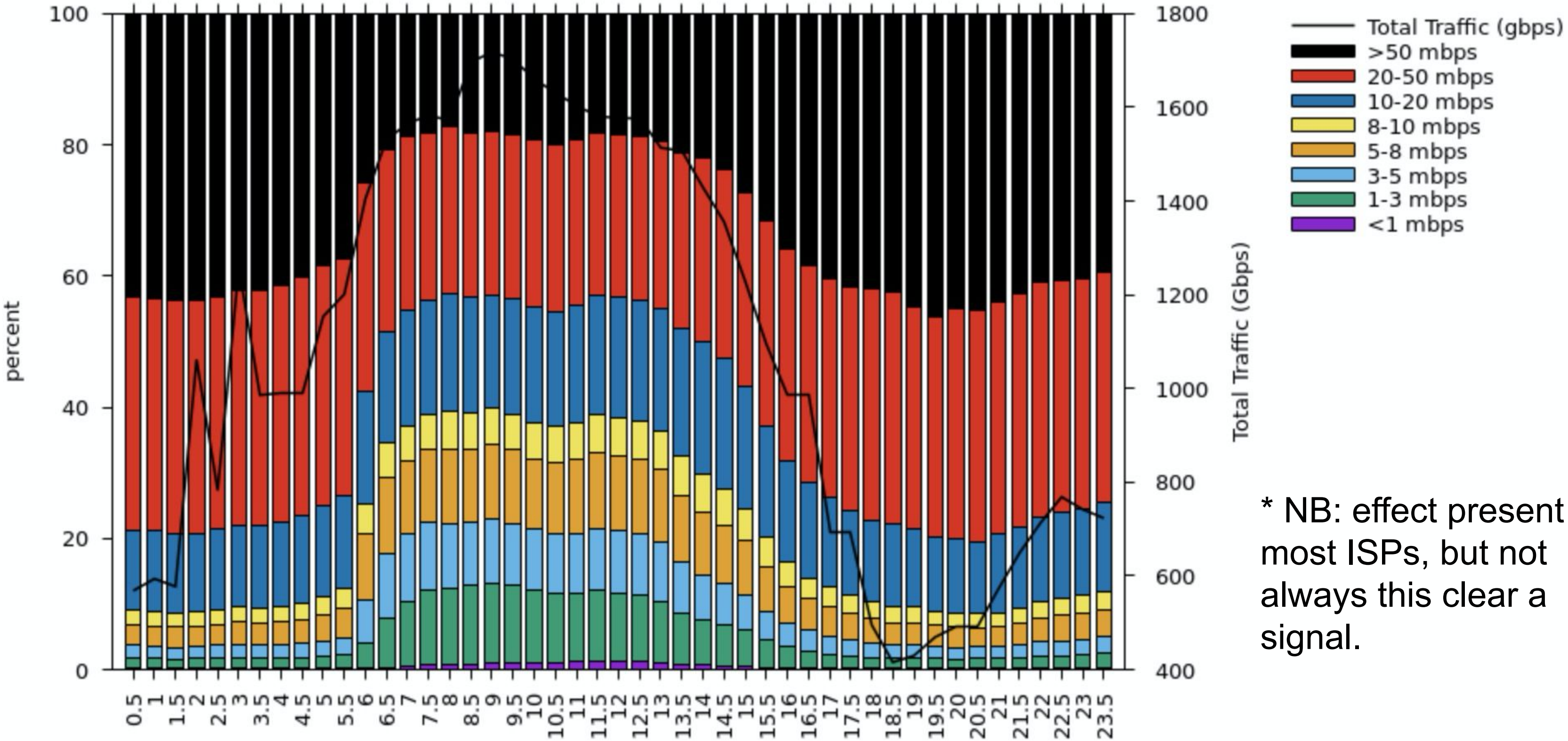
Problematic Congestion

Deepest Useful Caches

Cable Network Diagram By Saub09 at English Wikibooks, CC BY-SA 2.5, https://commons.wikimedia.org/w/index.php?curid=61793561

User Experience: Effects of Congestion

Observed goodput into large ISP* by Time of Day (high-traffic day, 100KB+ objects)



* NB: effect present in most ISPs, but not always this clear a signal.

Access Technologies: gain estimates at bottleneck links

Broadcast link capabilities can be leveraged by multicast? (up to?)

- Fiber (GPON, etc): yes (~**3k**/ONT)
- Cable: yes (~**2k**/service group)
- DSL: depends (~**1.5k**/chassis)
  - PPP-based deployments can’t use broadcast
  - Helps uplink bandwidth, but similar power usage
- Ethernet: usually (~**2k** in enterprise/university/apartment networks)
  - Needs L2 snooping & replication capability--usually there, not always
- 3G & 4G: sort-of (with eMBMS: ~**3k**/tower, special signaling)
- 5G: yes (with Xcast: ~**3k**/tower?, normal signaling?)
- ATSC: maybe one day (~**10-100k**/antenna, will need special signaling)

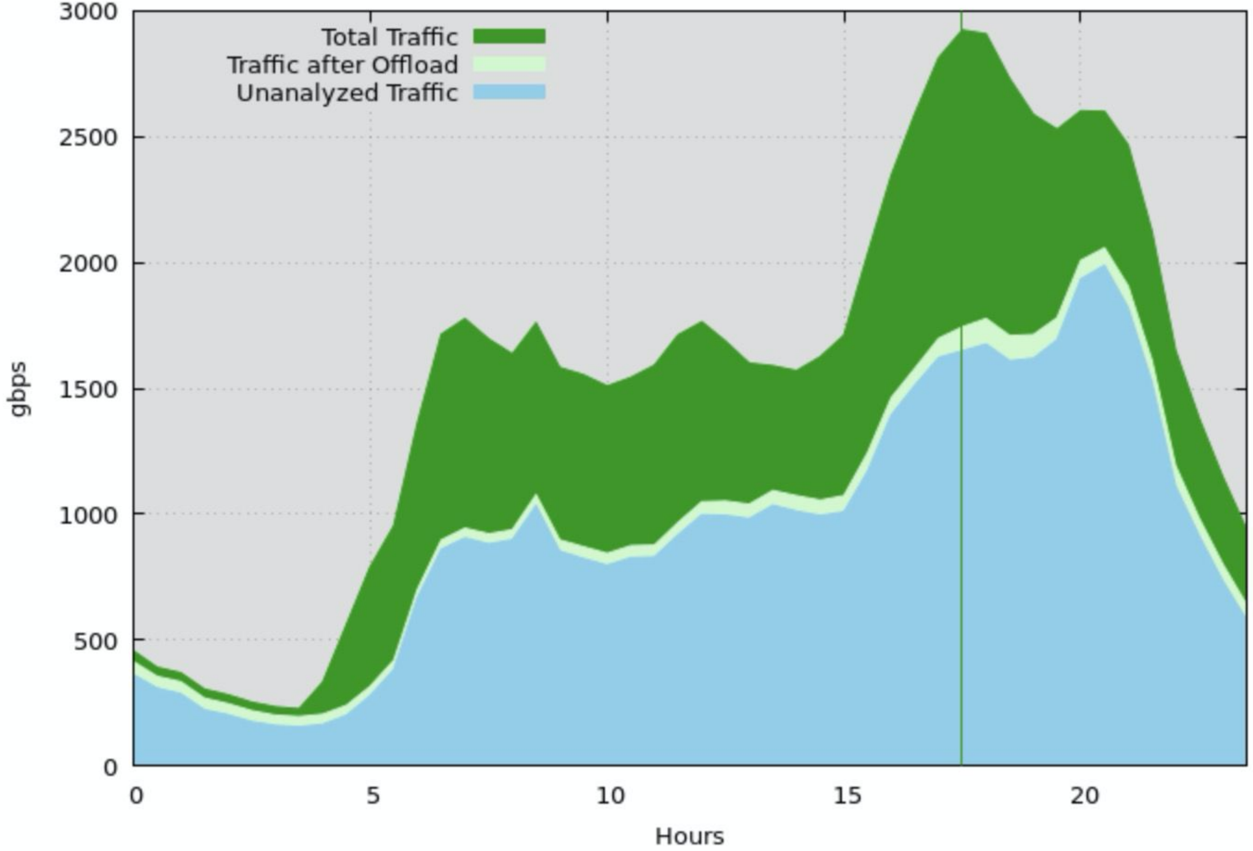
(* Wifi in homes may need updates--solutions exist, deployment spotty)

Other Effects

- Climate Impact
  - Internet=3.7% of carbon footprint globally (as much as air travel!)
- Cost of delivery & services
  - Network capital costs driven by peak load
  - Power needs/provider costs scale with traffic volume
  - Lower costs + competition => lower price for users

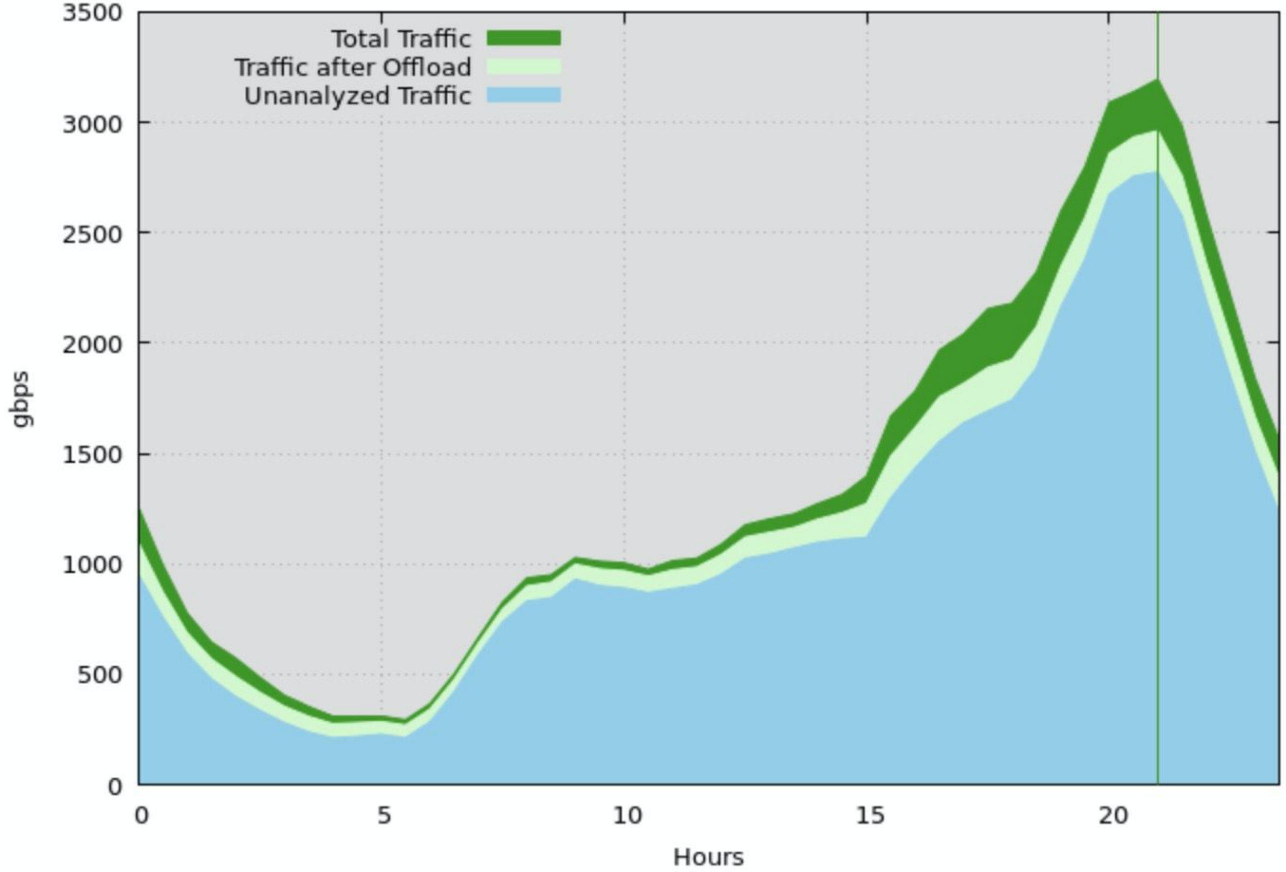
Avoidable Traffic (game/os downloads - new releases)

Under 100 streams: >40% reduction in peak load to ISP (high-traffic day)



Avoidable Traffic (game/os downloads - normal)

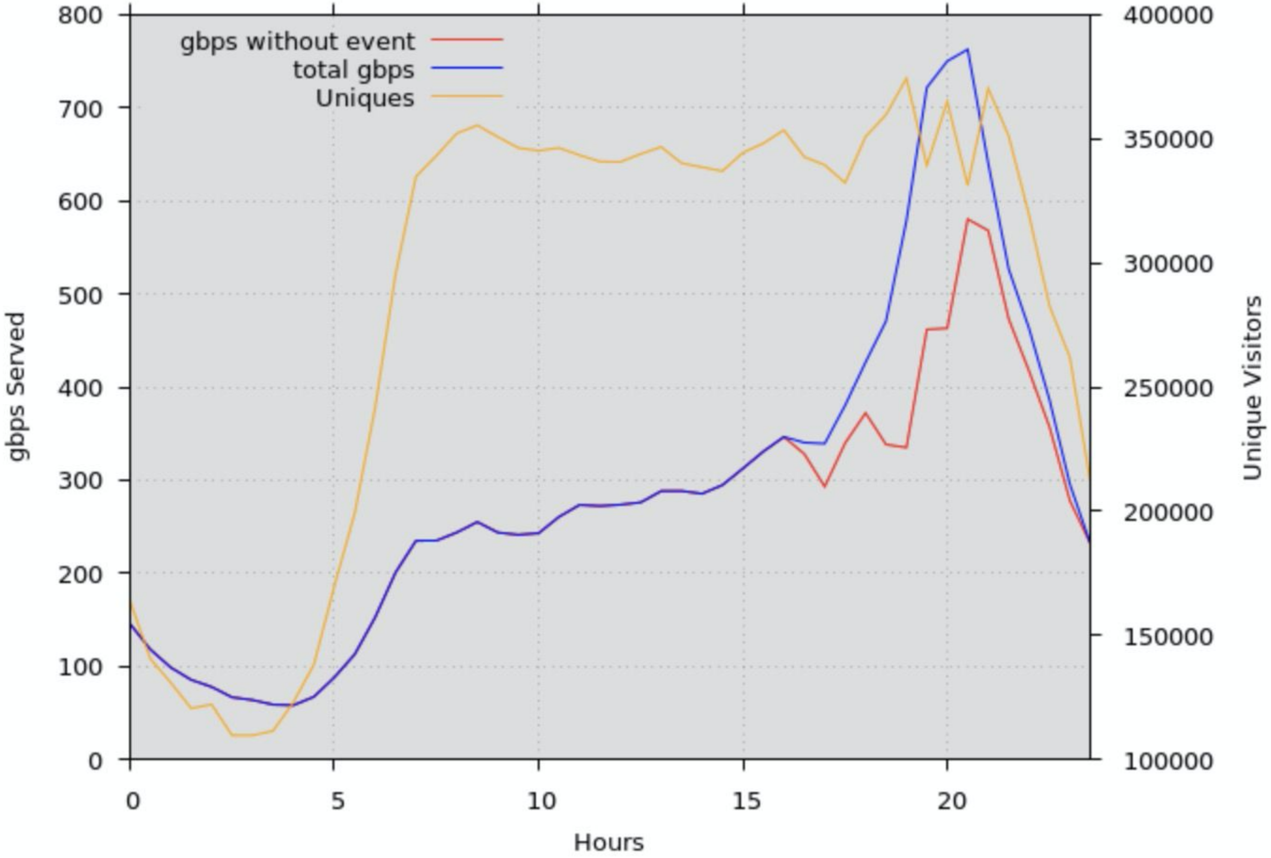
Under 100 streams: >8*% reduction overall traffic to ISP (normal day)



* lower bound. We think there’s much more but analysis is not complete.

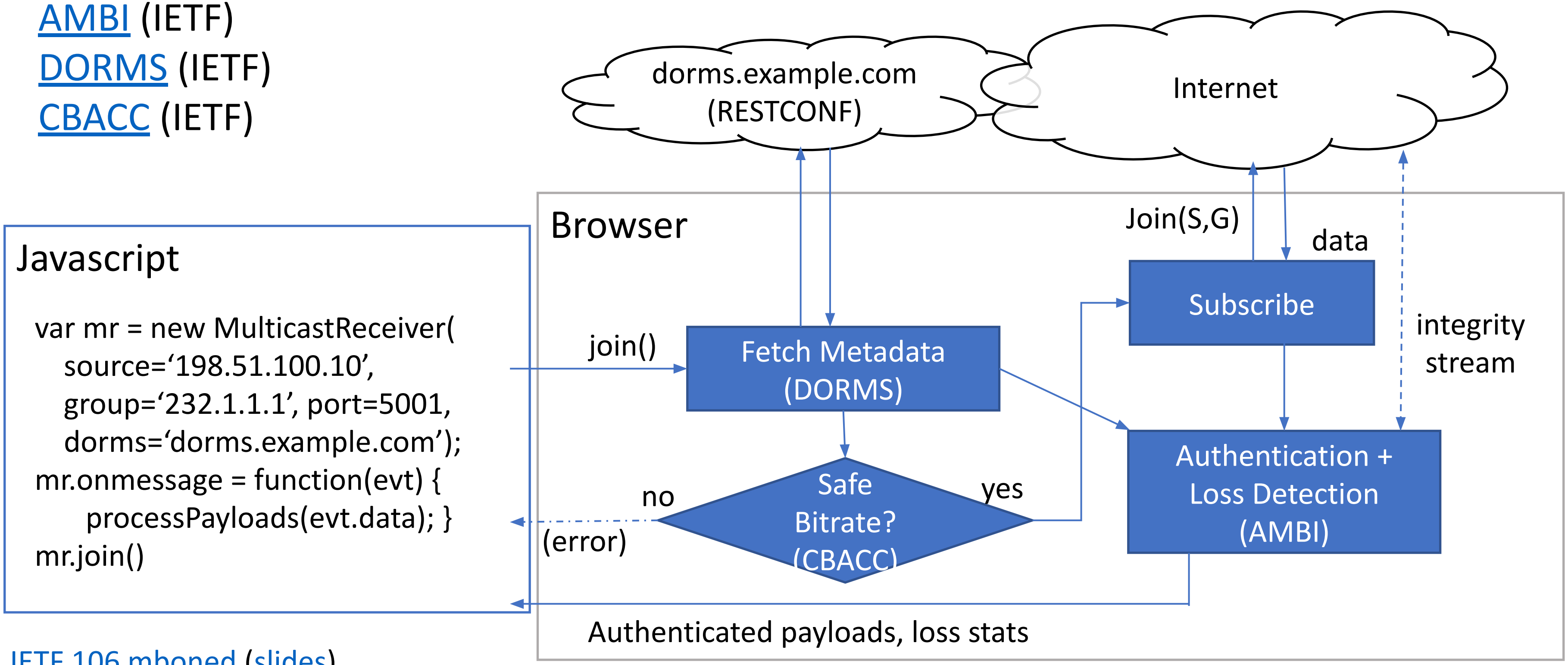
Avoidable Traffic (web video)

1 stream, >15% reduction in peak load to ISP (popular sport event day)



Browser API Proposal

Multicast Receive API (WICG)
AMBI (IETF)
DORMS (IETF)
CBACC (IETF)

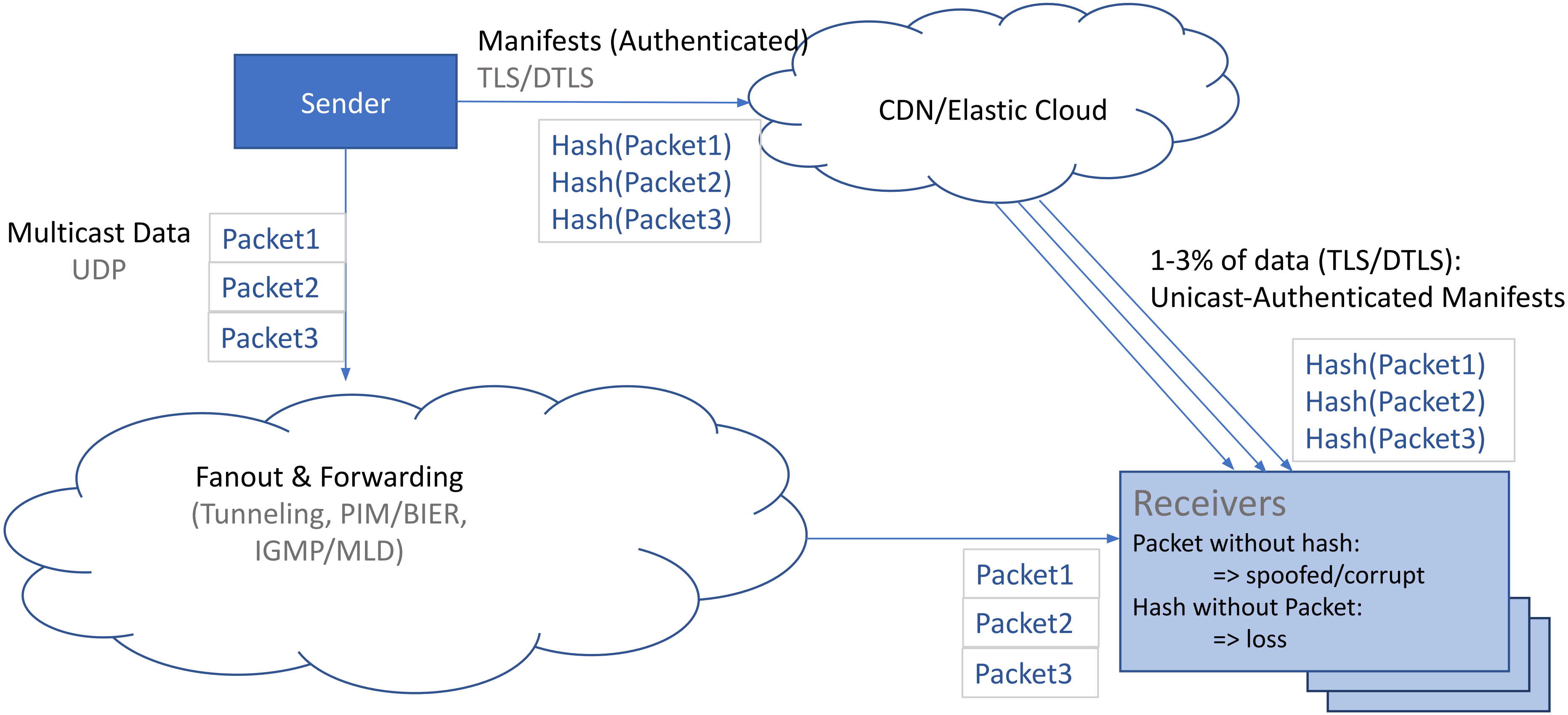


Javascript

```
var mr = new MulticastReceiver(
    source='198.51.100.10',
    group='232.1.1.1', port=5001,
    dorms='dorms.example.com');
mr.onmessage = function(evt) {
      processPayloads(evt.data); }
mr.join()
```

IETF 106 mboned (slides)

AMBI (Asymmetric Manifest-Based Integrity)



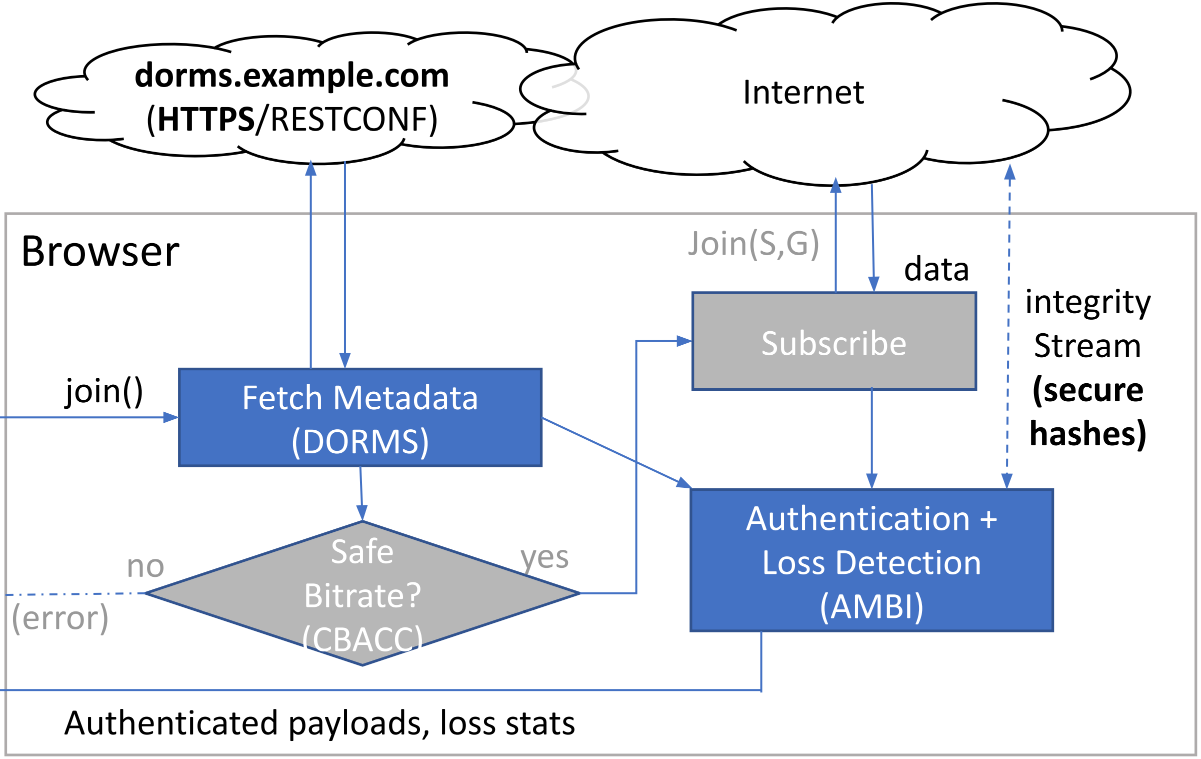
Packet without hash:
=> spoofed/corrupt
Hash without Packet:
=> loss

AMBI Chain of Trust

1. **Explicit** DORMS hostname from secure context (implicit ok iff DNSSEC--mostly for network)
2. CORS request to **DORMS** server (if not same origin)
3. **DORMS** has **AMBI** data with:
   a. integrity url
   b. Hash algorithm/params
4. Integrity stream over TLS/DTLS

Javascript

```
var mr = new MulticastReceiver(
    source=‘198.51.100.10’,
    group=‘232.1.1.1’, port=5001,
    dorms=‘dorms.example.com’);
mr.onmessage = function(evt) {
      processPayloads(evt.data); }
mr.join()
```



IETF 106 mboned (slides)

Early Feedback

- Security:
  - MUST require encryption for a new web API
    - Not visible to those without keys (in spite of one-to-many keys)
    - Makes on-path observation an active attack instead of passive
- Privacy:
  - Next-hop join exposure is fundamentally different from TLS/unicast
    - Addressable by other means? (e.g. random mac?)
  - Upstream benefits to privacy--indistinguishably shared destination IP
- Suitability:
  - Mixed-content experiments not welcome
  - Needs wider consensus & review (after adding encryption) before possibility to deem this non-mixed, due to fundamental differences with unicast/TLS

Thanks Ryan Sleevi, Tomasz Jamroszczak, Chris Palmer for Chromium net-dev thread

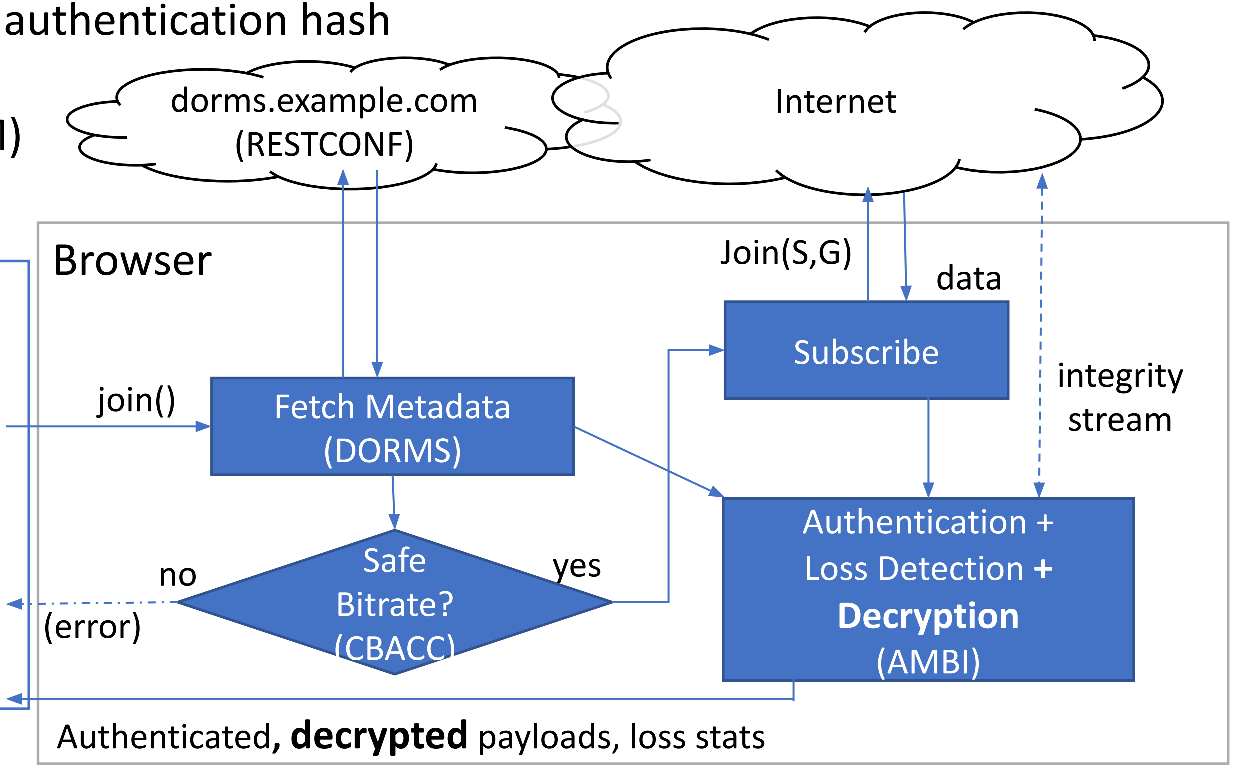
Option #1: add encryption to AMBI

Add key url+symmetric algorithm to AMBI metadata

- Encrypt payloads before authentication hash
- Identity controls for encryption key (like DRM)

Javascript

```
var mr = new MulticastReceiver(
   source='198.51.100.10',
   group='232.1.1.1', port=5001,
   dorms='dorms.example.com');
mr.onmessage = function(evt) {
     processPayloads(evt.data); }
mr.join()
```



Authenticated**, decrypted** payloads, loss stats

Option #2 (feedback suggestion): narrower APIs

- Separate multicast-capable APIs per use-case:
  - WebRTC extension to support multicast RTP
  - Segmented media delivery API (Maybe DVB’s protocols?)
  - Background downloader API (extend html5 download attribute?)
  - Pub/sub API? Others?
- Same challenges?
  - Needs AMBI-like integrity/authenticity & one-to-many encryption
  - Same fundamentals at network layer (doesn’t fix privacy concerns?)
- Maybe leverage DRM system for decryption & key control?
  - Can AMBI do this per-packet in option #1?
- We want this eventually for performance, regardless
  - But: Hard to pick the protocols to use ahead of experimenting

Side notes on DVB-MABR

Disambiguating multiple deployment options:

- Walled-garden, ISP to set top box (ETSI TS 103 769 V1.1.1)
  - Transparent to browser. Just HLS/DASH from STB.
  - Requires special hardware for user, deployed in home
  - Uncertain feasibility for non-ISP services
    - TLS anchor for local STB referral is tricky, but maybe plex-style is feasible? Needs local discovery and/or federation?
- Multicast delivery to end user devices (work in progress just began)
  - Looks feasible (see recent presentation to DVB for discussion)
  - Works for either option
    - Option 1: DVB wasm implementation using generic API
    - Option 2: DVB browser-embedded implementation

Next Step Considerations

Option 1 (generic multicast API)

Pros:

- See Extensible Web Manifesto
- Early-phase POC running
- Useful for existing vendors

Cons:

- CPU use in renderer
- Payload transport to renderer
- Security considerations?

Option 2 (narrow use-case APIs)

Pros:

- Performance w/same protocol
  - We'll want these anyway
- Less scope for trouble

Cons:

- More APIs
- Harder to experiment
- Best approaches not known