DISPATCH WG
ART Area
Virtual Meeting

IETF-111

Patrick McManus (co-chair)

Kirsty Paine (co-chair)

<dispatch-chairs@ietf.org>

Note Well

This is a reminder of IETF policies in effect on various topics such as patents or code of conduct. It is only meant to point you in the right direction. Exceptions may apply. The IETF's patent policy and the definition of an IETF "contribution" and "participation" are set forth in BCP 79; please read it carefully.

As a reminder:

- By participating in the IETF, you agree to follow IETF processes and policies.
- If you are aware that any IETF contribution is covered by patents or patent applications that are owned or controlled by you or your sponsor, you must disclose that fact, or not participate in the discussion.
- As a participant in or attendee to any IETF activity you acknowledge that written, audio, video, and photographic records of meetings may be made public.
- Personal information that you provide to IETF will be handled in accordance with the IETF Privacy Statement.
- As a participant or attendee, you agree to work respectfully with other participants; please contact the ombudsteam (https://www.ietf.org/contact/ombudsteam/) if you have questions or concerns about this.

Definitive information is in the documents listed below and other IETF BCPs. For advice, please talk to WG chairs or ADs:

- BCP 9 (Internet Standards Process)
- BCP 25 (Working Group processes)
- BCP 25 (Anti-Harassment Procedures)
- BCP 54 (Code of Conduct)
- BCP 78 (Copyright)
- BCP 79 (Patents, Participation)
- https://www.ietf.org/privacy-policy/ (Privacy Policy)

IETF 111 Virtual Meeting Participation

This session is being recorded

- **Participation Tips:**
  - **Please use a headset!**
  - **Please add yourselves to the Meetecho queue to speak. Please say your name before speaking**
  - **Side discussions in Jabber are fine, but please consider bringing important points to the queue.**
  - **Blue Sheets will be taken from the Meetecho roster**
- **Some useful links:**
  - **DISPATCH Session: https://gce.conf.meetecho.com/conference/?group=dispatch**
  - **Notes: https://codimd.ietf.org/notes-ietf-111-dispatch**
  - **Jabber (Mirrored in Meetecho): xmpp:dispatch@jabber.ietf.org?join**
  - **Meetecho Participant Guide: https://www.ietf.org/how/meetings/technology/meetecho-guide-participant/**
  - **Meetecho Documentation: https://www.ietf.org/media/documents/IETF111-Meetecho-Documentation.pdf**
  - **Meetecho Tutorial Video: https://youtu.be/0SPRhhHAg2I**

IETF 111 DISPATCH Virtual Meeting

26 July 2021

Agenda (Part 1 of 2)

- 1900-1905 Status and agenda bash (Chairs and ADs)

DISPATCH

- 1905-1925 -- **JWS Clear Text JSON Signature Option (JWS/CT) (Samuel Erdtman)**
- 1925-1935 -- **image/webp mime-type registration (James Zern)**
- 1935-1955 -- **NICER (usage profile of ICE) (Harald Alvestrand)**
- 1955-2005 -- **SDP Security Descriptions is NOT RECOMMENDED and Historic (John Mattsson)**
- 2005-2015 -- **The "large file in email" problem (Bron Gondwana)**

Agenda (Part 2 of 2)

ART Area

- 2015-2025 – **BoF, updates and other meetings of interest (ADs)**
- 2025-2030 -- **FFV1 v4 - new codec spec from CELLAR WG (Michael Richardson)**
- 2030-2040 -- **IPv6 Zone Identifiers in URIs (RFC6874bis) (Brian Carpenter)**
- 2040-2050 -- **Reliable (unreliable) streaming protocol (Kirill Pugin)**

AoB

A Reminder of Mailing lists

- General discussion of ART topics: art@ietf.org
- Discussion of work proposals that need to be dispatched: dispatch@ietf.org

DISPATCH Topics

JWS Clear Text JSON Signature Option

B. Jordan, Ed. - Broadcom
S. Erdtman - Spotify AB
A. Rundgren - Independent

Abstract

JWS Clear Text JSON Signature Option describes a method for extending JSON Web Signature (JWS) standard, called JWS/CT. By combining the detached mode of JWS with the JSON Canonicalization Scheme (JCS). Maintaining Signed JSON data in JSON format.

Why JSON Cleartext signatures

In many situations JSON data structures are already defined. To have the option to add a signature without packaging the data within the signature is important to improve backwards compatibility and adoption of signed data.

Large data structures in JSON format that are shared across and between organisations, and signed in later steps and reshared is not feasible with solutions that packages data within the signature, the nesting becomes unbearable.

Adding multiple signatures and nested data structures in different combinations is significantly easier to handle when the signature is packaged within the data and not the other way around.

If needing/wanting to transmit unsigned data it is easy to omit the signature instead of creating a JWS package and then use the “none” algorithm.

Maintaining JSON data as is while in transport is makes readability and debugging easier.

Constructs

JSON Canonicalization Schema and Detached JWS

RFC-8785 (JCS) Canonicalization

Human readable format

```
{
        "numbers": [333333333.33333329, 1E30, 4.50, 2e-3],
        "string": "\u20ac$\u000F\u000aA'\u0042\u0022\u005c\\\"\/",
        "literals": [null, true, false]
}
```

Canonical format (100% valid JSON):

```
{"numbers":[333333333.3333333,1e+30,4.5,0.002],"string":"
€$\u000f\nA'B\"\\\\\"/","literals":[null,true,false]}
```

Detach JWS Signature

Normal/Classic JWS Signature

eyJ0eXAiOiJKV1QiLA0KICJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJqb2UiLA0KICJleHAiOjEzMDA4MTkzODAsDQogImh0dHA6Ly9leGFtcGxlLmNvbS9pc19yb290Ijp0cnVlfQ.dBjftJeZ4CVP-mB92K27uhbUJU1p1r_wW1gFWFOEjXk

Detached JWS Signature

eyJ0eXAiOiJKV1QiLA0KICJhbGciOiJIUzI1NiJ9..dBjftJeZ4CVP-mB92K27uhbUJU1p1r_wW1gFWFOEjXk

JWS/CT Signing Process

JWS/CT Signing Process

Create the JSON Object to be Signed

Canonicalize the JSON Object to be Signed

Generate a JWS String

Assemble the Signed JSON Object

```
{
  "statement": "Hello signed world!",
  "otherProperties": [2000, true]
}
```

JWS/CT Signing Process

Create the JSON Object to be Signed

Canonicalize the JSON Object to be Signed

Generate a JWS String

Assemble the Signed JSON Object

```
{"otherProperties":[2000,true],"statement":"Hello signed world!"}
```

JWS/CT Signing Process

Create the JSON Object to be Signed

Canonicalize the JSON Object to be Signed

Generate a JWS String

Assemble the Signed JSON Object

eyJhbGciOiJIUzI1NiJ9..VHVItCBCb8Q5CI-49imarDtJeSxH2uLU0DhqQP5Zjw4

JWS/CT Signing Process

Create the JSON Object to be Signed

Canonicalize the JSON Object to be Signed

Generate a JWS String

Assemble the Signed JSON Object

```
{
  "statement": "Hello signed world!",
  "otherProperties": [2000, true],
  "signature": "eyJhbGciOiJIU..."
}
```

JWS/CT Verification Process

JWS/CT Verification Process

Parse the Signed JSON Object

Fetch the Signature Property String

Remove the Signature Property String

Canonicalize the Remaining JSON Object

Validate the JWS String

```
{
  "statement": "Hello signed world!",
  "otherProperties": [2000, true],
  "signature": "eyJhbGciOiJIU..."
}
```

JWS/CT Verification Process

Parse the Signed JSON Object

Fetch the Signature Property String

Remove the Signature Property String

Canonicalize the Remaining JSON Object

Validate the JWS String

eyJhbGciOiJIUzI1NiJ9..VHVItCBCb8Q5CI-49imarDtJeSxH2uLU0DhqQP5Zjw4

JWS/CT Verification Process

Parse the Signed JSON Object

Fetch the Signature Property String

Remove the Signature Property String

Canonicalize the Remaining JSON Object

Validate the JWS String

```
{
  "statement": "Hello signed world!",
  "otherProperties": [2000, true]
}
```

JWS/CT Verification Process

Parse the Signed JSON Object

Fetch the Signature Property String

Remove the Signature Property String

Canonicalize the Remaining JSON Object

Validate the JWS String

```
{"otherProperties":[2000,true],"statement":"Hello signed world!"}
```

JWS/CT Verification Process

Parse the Signed JSON Object

Fetch the Signature Property String

Remove the Signature Property String

Canonicalize the Remaining JSON Object

Validate the JWS String

```
eyJhbGciOiJIUzI1NiJ9.eyJvdGhlclB
yb3BlcnRpZXMiOlsyMDAwLHRydW
VdLCJzdGF0ZW1lbnQiOiJIZWxsby
BzaWduZWQgd29ybGQhIn0.VHVIt
CBCb8Q5CI-49imarDtJeSxH2uLU0
DhqQP5Zjw4
```

JWS/CT Application Notes

The document does not dictate signature attribute name or location. It is up to the application to choose a suitable name and location for the signature attribute in its context.

The document does not define counter signatures, arrays of signatures or detached signatures, but it exemplifies how it could be done.

Path forward

We suggest that this work is moved forward under ISE, because:

- Canonicalization (RFC-8785) is published as an Independent Submission.
- JOSE WG is not active and the list has previously expressed limited interest in this work.

Path forward

We suggest that this work is moved forward under ISE, because:

- Canonicalization (RFC-8785) is published as an Independent Submission.
- JOSE WG is not active and the list has previously expressed limited interest in this work.

However

- **If you are interested we would love to get reviews**

Thank you!

Questions?
Comments!

WebP Media Type Registration

image/webp

James Zern (Google)

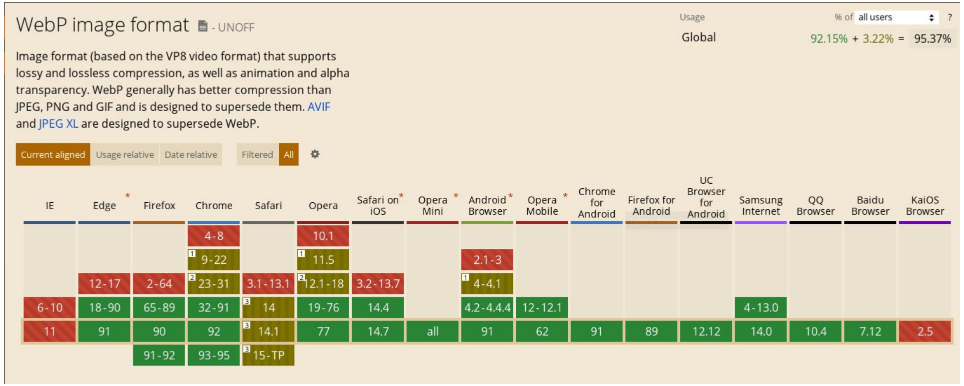
WebP Image Format

https://developers.google.com/speed/webp

- Lossy Compression (2010)
- Lossless Compression (2012)
- Alpha (Transparency) (2012)
- Animation (2013)

Availability

- All major browsers
- Image editing applications
  - ○ GIMP
  - ○ ImageMagick
  - ○ Photoshop (plugin)
  - ○ …
- [image/webp use (Debian)]



WebP image format - UNOFF

Usage % of all users ?

Global 92.15% + 3.22% = 95.37%

Image format (based on the VP8 video format) that supports lossy and lossless compression, as well as animation and alpha transparency. WebP generally has better compression than JPEG, PNG and GIF and is designed to supersede them. AVIF and JPEG XL are designed to supersede WebP.

IE	Edge *	Firefox	Chrome	Safari	Opera	Safari on iOS *	Opera Mini *	Android Browser *	Opera Mobile *	Chrome for Android	Firefox for Android	UC Browser for Android	Samsung Internet	QQ Browser	Baidu Browser	KaiOS Browser
			4-8		10.1											
			[1] 9-22		[1] 11.5			2.1-3								
	12-17	2-64	[2] 23-31	3.1-13.1	[2] 12.1-18	3.2-13.7		[1] 4-4.1								
6-10	18-90	65-89	32-91	[3] 14	19-76	14.4		4.2-4.4.4	12-12.1				4-13.0			
11	91	90	92	[3] 14.1	77	14.7	all	91	62	91	89	12.12	14.0	10.4	7.12	2.5
		91-92	93-95	[3] 15-TP												

Working Group Discussion

WG Proposal: DISPATCH

IANA Considerations: IANA has updated the "Image Media Types" registry to include 'image/webp' as described in draft-zern-webp

NICER - Nicer ICE based on RTT

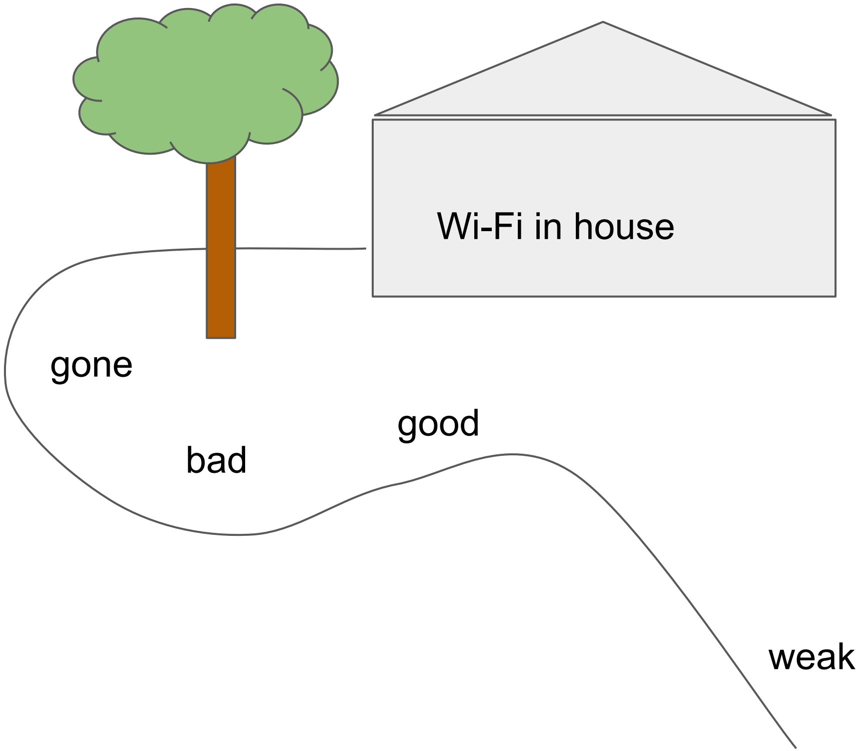
Jonas Oreland, Harald Alvestrand
Google

Problem Statement

ICE tries many paths.

It then selects one, and keeps it until ICE restart.

This is not always optimal.



Basic Idea of NICER

- Rather than throwing away non-best paths, keep them (or some of them)
- Keep probes running on them to give an idea of current quality
- When one path deteriorates, switch to another one
- If the bad path turns good again, switch back

Meshes well with Trickle-ICE: Making new candidates (and candidate pair sets) available whenever a new interface turns up.

Offers many opportunities for heuristic tuning and optimization

Requires a few changes in order to be interoperable

NICER: Required extensions to the ICE specification

- Permit the ICE controller to change the selected candidate pair
  - draft-thatcher-ice-renomination was one attempt at specifying this
- Be explicit that ICE candidate pairs with connectivity are not discarded
  - Those that never connect, or that are suspected of being duplicates, can be discarded
- Support Trickle ICE (RFC 8838)

We believe that with these extensions, any NICER ICE controller endpoint should be interoperable with any NICER ICE controlled endpoint.

NICER: Possible optimizations in the ICE protocol

There are a number of changes one could do in order to make NICER perform better.

NICER sends a large number of pings, so the size of the ping is important.

- Shorter checksums
- Omitting parameters whose value is unchanging after the first ping

NICER could also make good use of more information about the network on the ICE controlled side - however, this raises some privacy concerns.

NICER: ICE Controller behavior examples

NICER has to make two important decisions:

- When to ping
- When to switch

The ping decision may take into account phase of the call (ping a lot at first, more rarely later), stability of ping results (varying RTT may indicate a rapidly fluctuating network condition, requiring more pings for reliability), and other factors.

The switch decision may take into account RTT, ping failure rate, cost of interface (prefer free over $-per-gigabyte), and other factors

NICER: Tricks and lessons

NICER will send many pings. One thing we learned is that on mobile devices, activating a radio costs a lot of power budget. Thus - rather than running all pairs on individual timers, bunch up all the pings that use the same radio, so that they are sent at the same time, activating only once.

NICER will switch earlier than “switch-after-break” schemes based on ICE restart, but swtiching has a cost too - a certain hysteresis needs to be applied so that the connection does not oscillate.

NICER will not probe with a full load, so when the channel switches, congestion may occur. Bandwidth management needs to integrate with NICER.

NICER: Open issues

NICER works best when it has maximum information available. But it has to run in the ICE Controller, since that’s the one that initiates switches.

This works well for client to datacenter, since all the network complexity will be on the client side; it takes the controller role.

It works less well with peer-to-peer, unless the peers can exchange more information than is presently done.

NICER: Next steps

We seek the DISPATCH group's advice on one of the following:

- This is a bad idea because of <reason>. Do not speak of it again.
- This is something best done in private, but documented in public. Ask for an ind-sub Informational to tell others what you are doing.
- This is a small good idea. We should ask for an AD-sponsored PS.
- This is a big good idea. We should dispatch it to <Insert WG here>.
- This is a big good idea. We shoul reactivate or create a WG for it.

We do not have a preference at this time.

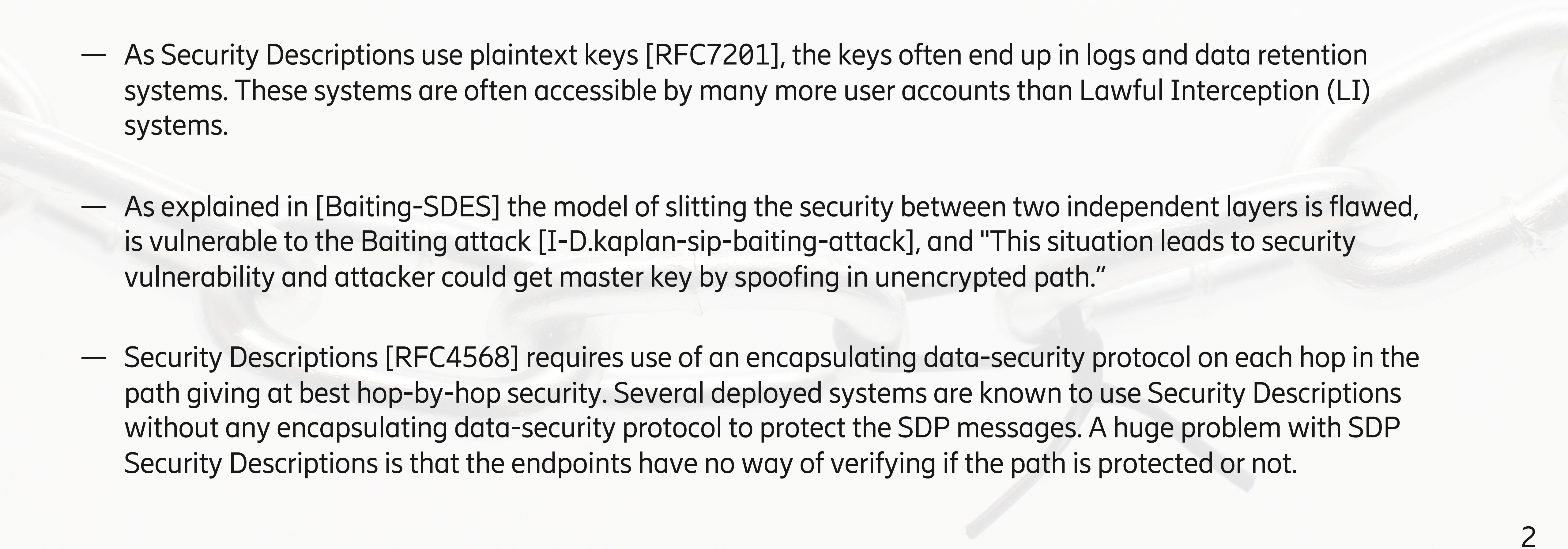
SDP Security Descriptions is NOT RECOMMENDED and Historic

draft-mattsson-dispatch-sdes-dont-dont-dont

IETF 111, DISPATCH, John Preuß Mattsson, Magnus Westerlund

Weakest Link by Yuri Samoilov https://yuri.samoilov.online/
Creative Commons Attribution 3.0 Unported (CC BY 3.0) https://creativecommons.org/licenses/by/3.0/

SDES has many known security weaknesses…



- Security Descriptions is vulnerable to SSRC collisions, which leads to so called "two-time pad" [RFC7201].
  - Worse than using a 32-bit MAC, as "two-time pad" may lead to loss of both confidentiality and integrity.
  - In addition to happening by itself with a non-negligible probability, the SSRC collision attack can also be triggered by an attacker in different ways. See e.g. [Replay-SDES].

- As Security Descriptions use plaintext keys [RFC7201], the keys often end up in logs and data retention systems. These systems are often accessible by many more user accounts than Lawful Interception (LI) systems.

- As explained in [Baiting-SDES] the model of slitting the security between two independent layers is flawed, is vulnerable to the Baiting attack [I-D.kaplan-sip-baiting-attack], and "This situation leads to security vulnerability and attacker could get master key by spoofing in unencrypted path."

- Security Descriptions [RFC4568] requires use of an encapsulating data-security protocol on each hop in the path giving at best hop-by-hop security. Several deployed systems are known to use Security Descriptions without any encapsulating data-security protocol to protect the SDP messages. A huge problem with SDP Security Descriptions is that the endpoints have no way of verifying if the path is protected or not.

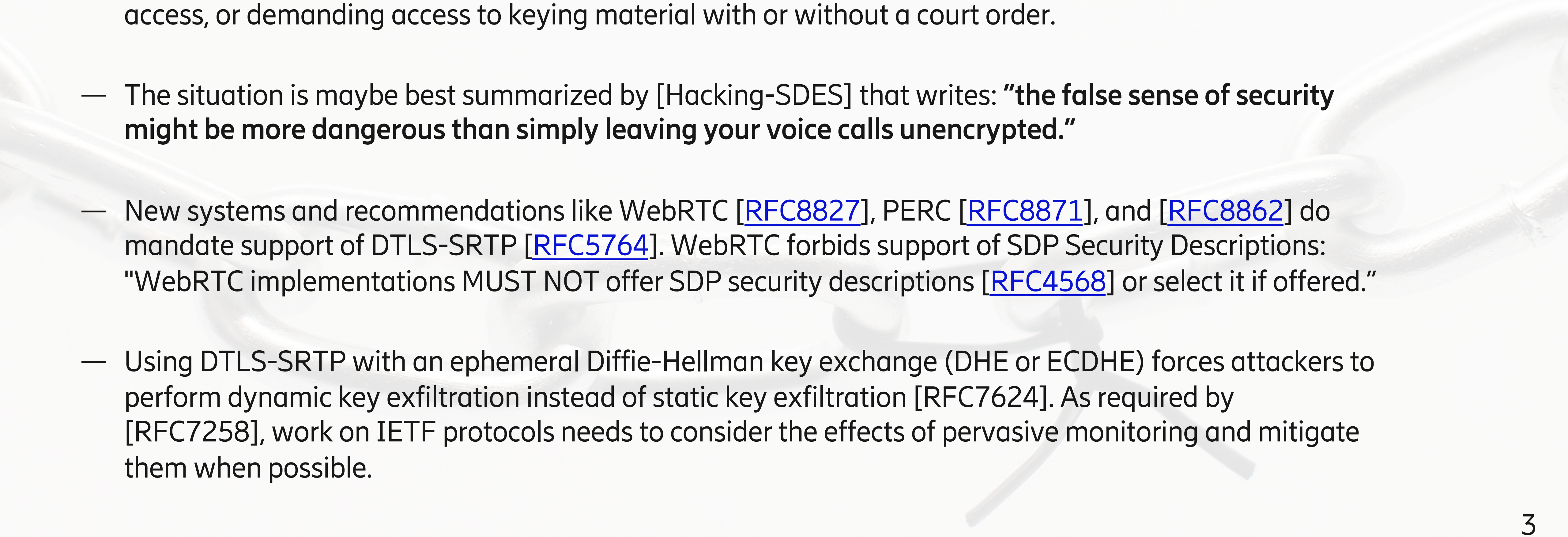
SDES has many known security weaknesses…

- If the encapsulating data-security protocol without Diffie-Hellman is used, access to long-term keys enables attackers to compromise past and future sessions. Entities can get access to (and have gotten access to) long-term key material in many different ways: physical attacks, hacking, software bugs, social engineering attacks, espionage, weaknesses injected in hardware, software, or standards, buying access, or demanding access to keying material with or without a court order.

- The situation is maybe best summarized by [Hacking-SDES] that writes: **”the false sense of security might be more dangerous than simply leaving your voice calls unencrypted.”**

- New systems and recommendations like WebRTC [RFC8827], PERC [RFC8871], and [RFC8862] do mandate support of DTLS-SRTP [RFC5764]. WebRTC forbids support of SDP Security Descriptions: "WebRTC implementations MUST NOT offer SDP security descriptions [RFC4568] or select it if offered.”

- Using DTLS-SRTP with an ephemeral Diffie-Hellman key exchange (DHE or ECDHE) forces attackers to perform dynamic key exfiltration instead of static key exfiltration [RFC7624]. As required by [RFC7258], work on IETF protocols needs to consider the effects of pervasive monitoring and mitigate them when possible.



The question is not if SDES should be phased out but how

- The security level SDP Security Description provide is not on the level expected by an IETF in force proposed standard and there exist alternatives.
- Many implementations, devices, and libraries support DTLS-SRTP.
- Current draft suggestion:

> This document reclassifies [RFC4568] (SDP Security Descriptions) to Historic Status and also obsoletes RFC 4568.
>
> This document updates [RFC7201] (Options for Securing RTP Sessions) to note that SDP Security Descriptions SHOULD NOT be used.
>
> This document specifies that use of the SDP Security Descriptions [RFC4568] is NOT RECOMMENDED. Existing deployments SHOULD mandate support of DTLS-SRTP [RFC5764] and long-term phase out use of SDP Security Descriptions. If it is known by out-of-band means that the other party supports DTLS-SRTP, then SDP Security Descriptions MUST NOT be offered or accepted. If it is not known if the other party supports DTLS-SRTP, both DTLS-SRTP and SDP Security Descriptions SHOULD be offered during a transition period. New deployments SHOULD forbid support of Security Descriptions [RFC4568].

- **Should this be handled in MMUSIC WG or somewhere else?**

The big file emailing problem

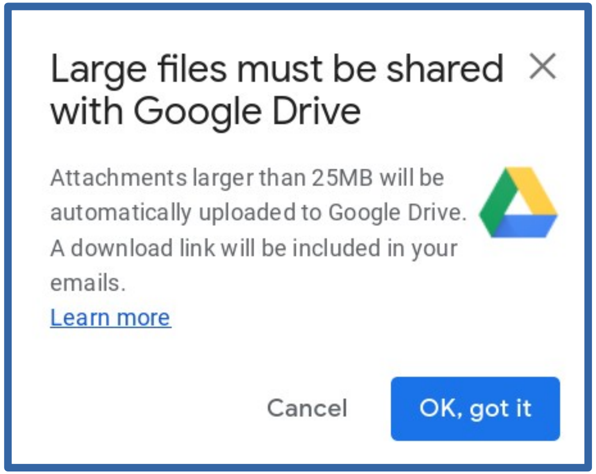
DISPATCH @IETF 111 / 26 Jul 2021
San Francciso / Virtual

Bron Gondwana <brong@fastmailteam.com>

Problem Statement

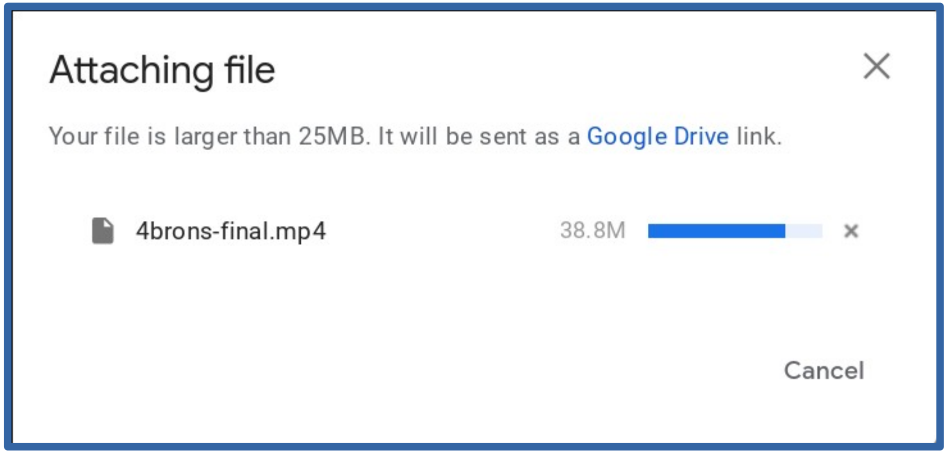
- End-users want to send files to each other
- Emails are limited to about 10Mb, as they have been for the past 20+ years
- The file sizes that end-users create keep getting bigger (4k video!)
- There is no standard solution

Solutions



Large files must be shared with Google Drive

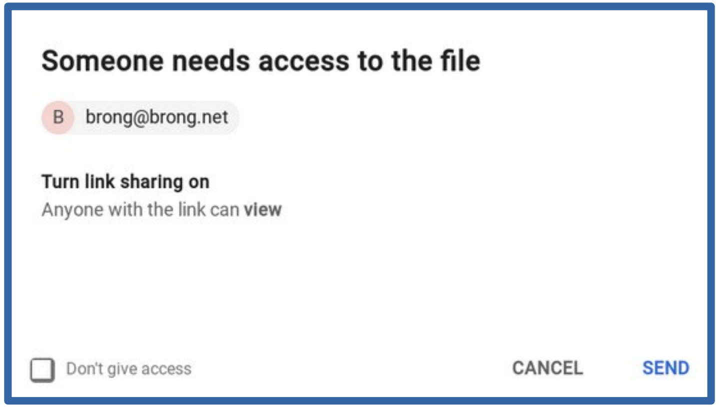
Attachments larger than 25MB will be automatically uploaded to Google Drive. A download link will be included in your emails.
Learn more



Attaching file

Your file is larger than 25MB. It will be sent as a Google Drive link.

4brons-final.mp4 38.8M

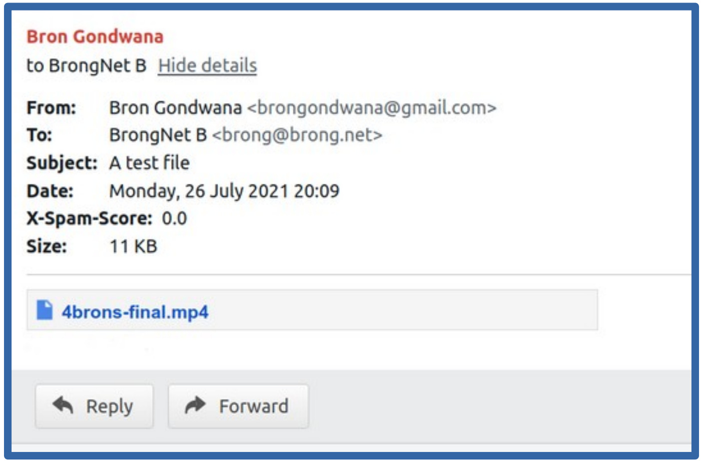


Someone needs access to the file

B brong@brong.net

Turn link sharing on
Anyone with the link can **view**

□ Don't give access

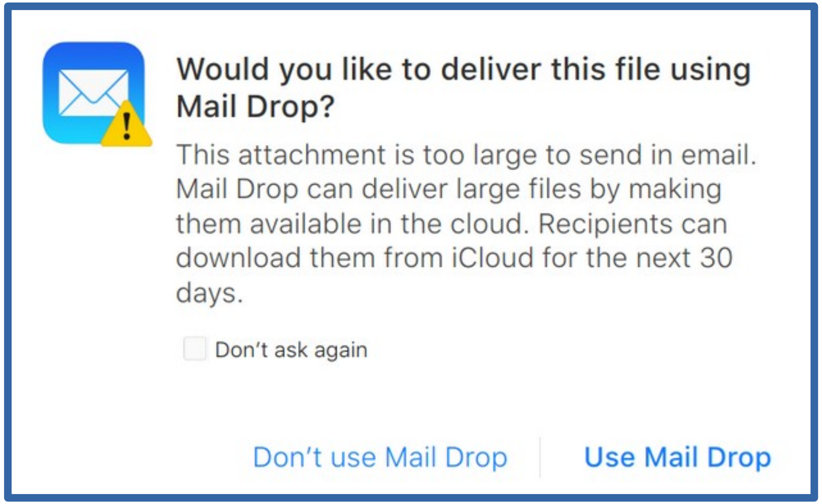


Bron Gondwana
to BrongNet B Hide details

From: Bron Gondwana <brongondwana@gmail.com>
To: BrongNet B <brong@brong.net>
Subject: A test file
Date: Monday, 26 July 2021 20:09
X-Spam-Score: 0.0
Size: 11 KB

4brons-final.mp4

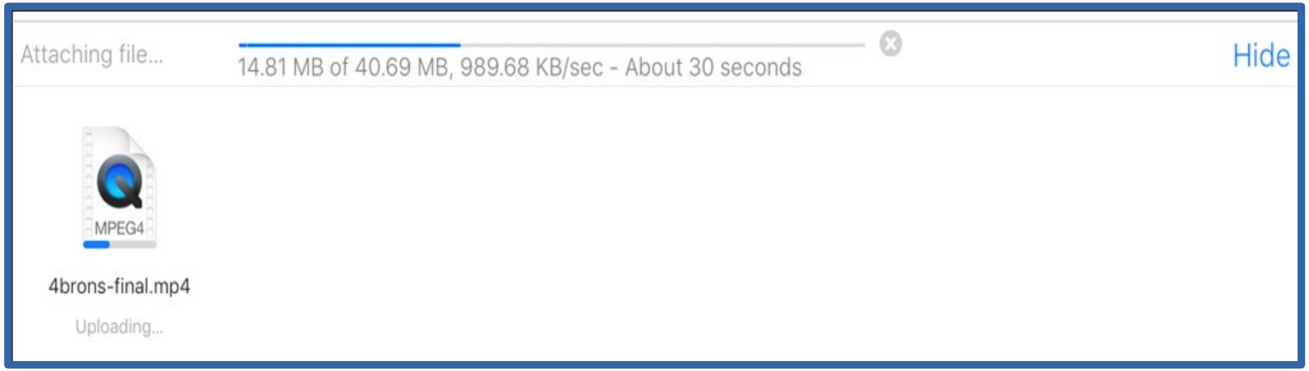
Solutions



Would you like to deliver this file using Mail Drop?

This attachment is too large to send in email. Mail Drop can deliver large files by making them available in the cloud. Recipients can download them from iCloud for the next 30 days.

□ Don't ask again

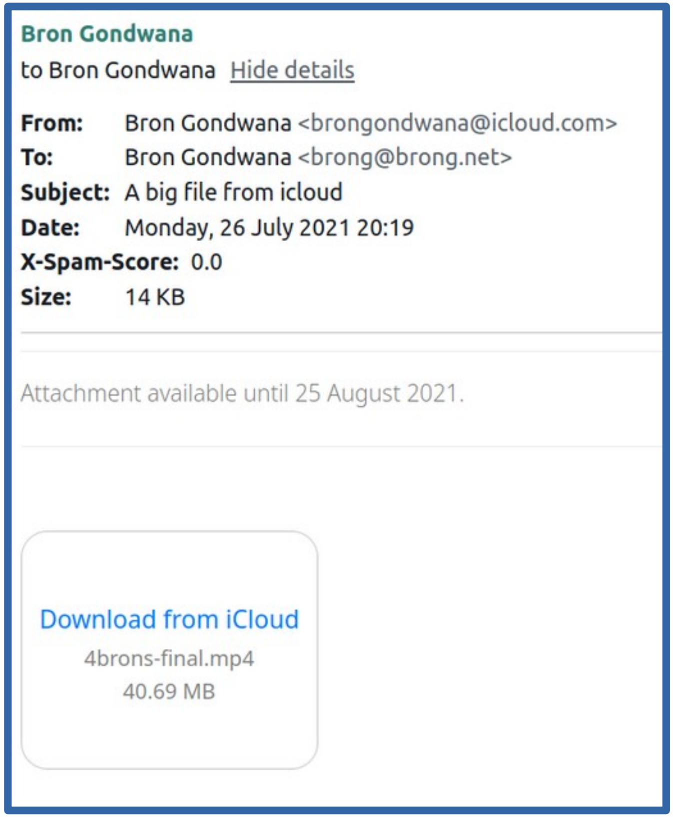


Attaching file...

14.81 MB of 40.69 MB, 989.68 KB/sec - About 30 seconds

4brons-final.mp4

Uploading...



Bron Gondwana

to Bron Gondwana Hide details

From: Bron Gondwana <brongondwana@icloud.com>
To: Bron Gondwana <brong@brong.net>
Subject: A big file from icloud
Date: Monday, 26 July 2021 20:19
X-Spam-Score: 0.0
Size: 14 KB

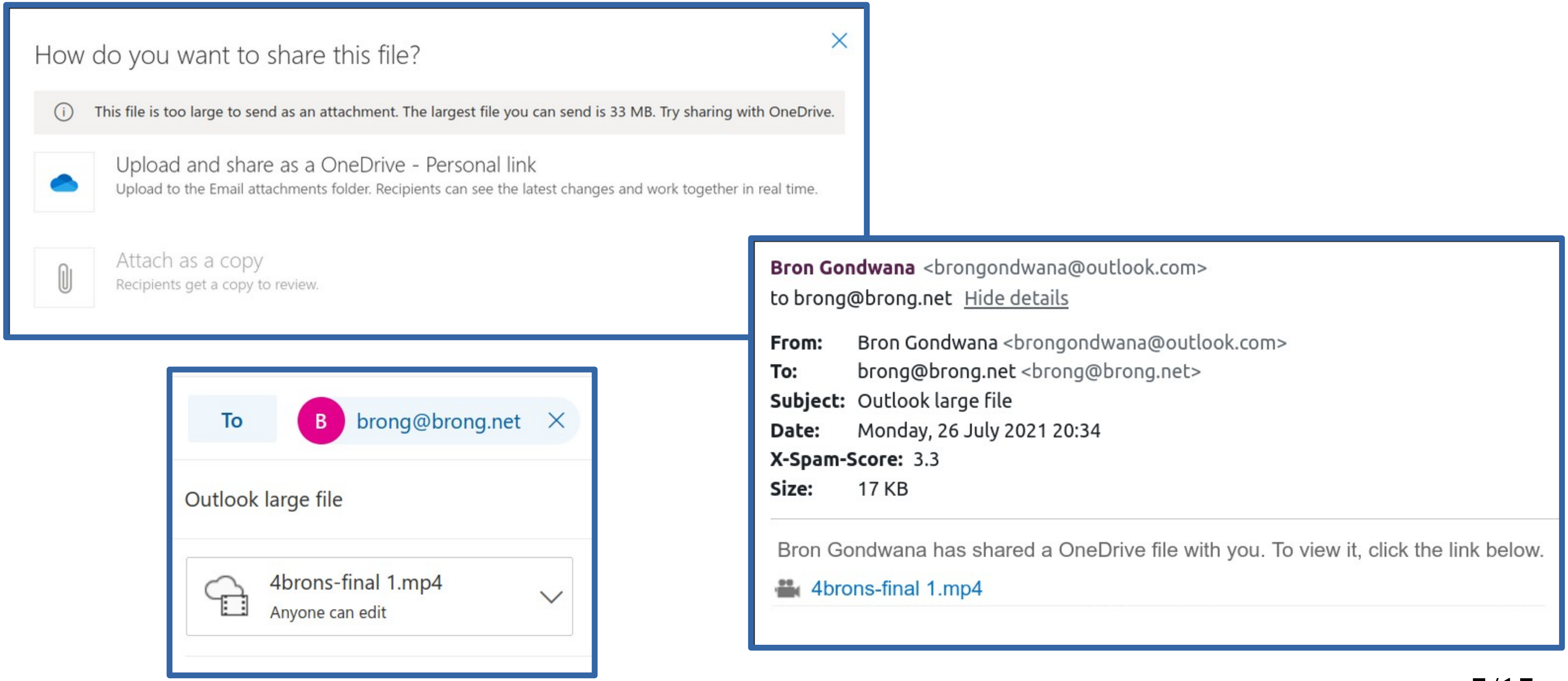
Attachment available until 25 August 2021.

Download from iCloud

4brons-final.mp4

40.69 MB

Solutions



How do you want to share this file?

ⓘ This file is too large to send as an attachment. The largest file you can send is 33 MB. Try sharing with OneDrive.

Upload and share as a OneDrive - Personal link
Upload to the Email attachments folder. Recipients can see the latest changes and work together in real time.

Attach as a copy
Recipients get a copy to review.

To: B brong@brong.net ✕

Outlook large file

4brons-final 1.mp4
Anyone can edit

Bron Gondwana <brongondwana@outlook.com>
to brong@brong.net Hide details

From: Bron Gondwana <brongondwana@outlook.com>
To: brong@brong.net <brong@brong.net>
Subject: Outlook large file
Date: Monday, 26 July 2021 20:34
X-Spam-Score: 3.3
Size: 17 KB

Bron Gondwana has shared a OneDrive file with you. To view it, click the link below.

4brons-final 1.mp4

Problems to solve

1. Upload of large files (restart)
2. Ownership/Responsibility of data
3. Discoverability/Embedability
   - Disposition: attachment vs inline
   - The virus-check / link follow problem
   - Search: “emails with attachments”

1. Upload of large files

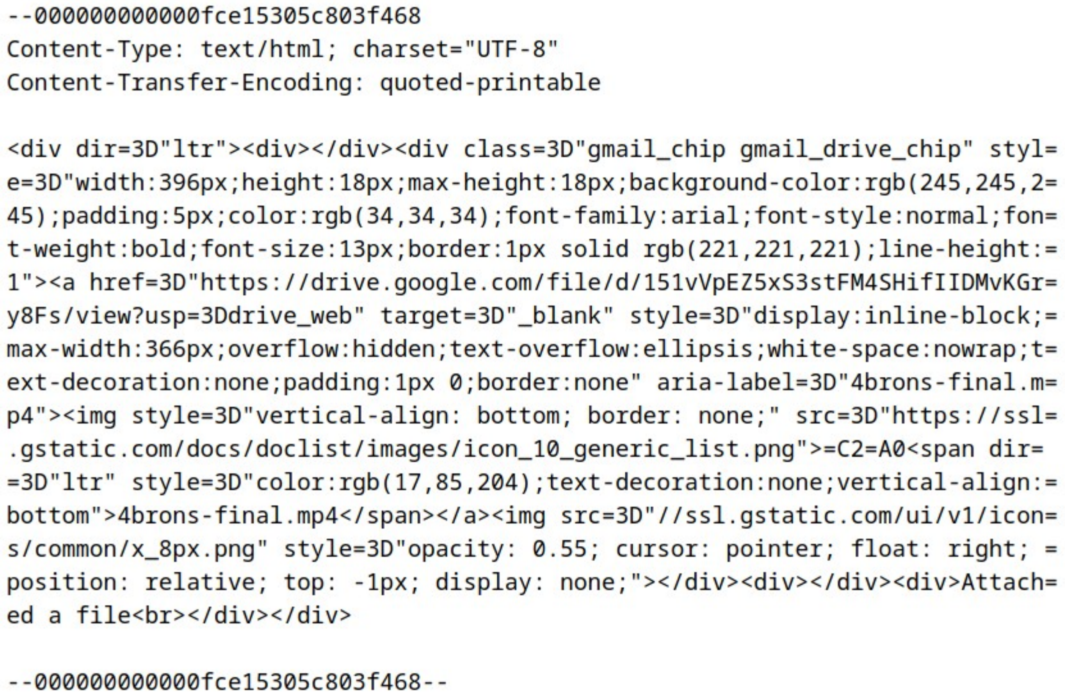
- This is a general problem
  - → httpbis?
  - → httpapi?
- IMAP / SMTP doesn’t support it
- We’d love this for JMAP
  - – My blob draft allows server-side concatenation so you can duct tape it by uploading chunks and joining them server side, but it’s horrible.

2. Ownership / Responsibility

- With “regular” emails, entire content is sent.
- Sender may keep a copy, but it’s not necessary for the protocol.
- iCloud gives an expiry date, others don’t even do that.
- Need a general way for recipient system to take a copy and tie lifetime to email lifetime.
- Centralised systems can upload and refcount, but email is decentralised.

3. Discoverability

Gmail:



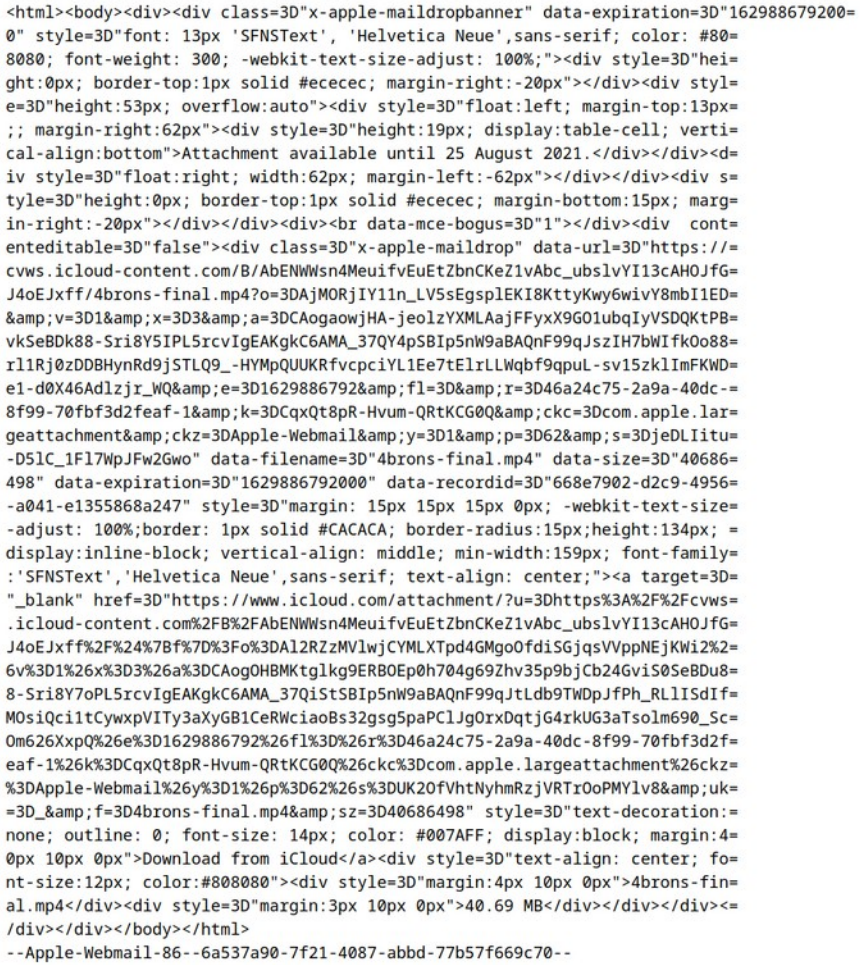
```
--000000000000fce15305c803f468
Content-Type: text/html; charset="UTF-8"
Content-Transfer-Encoding: quoted-printable

<div dir=3D"ltr"><div></div><div class=3D"gmail_chip gmail_drive_chip" styl=
e=3D"width:396px;height:18px;max-height:18px;background-color:rgb(245,245,2=
45);padding:5px;color:rgb(34,34,34);font-family:arial;font-style:normal;fon=
t-weight:bold;font-size:13px;border:1px solid rgb(221,221,221);line-height:=
1"><a href=3D"https://drive.google.com/file/d/151vVpEZ5xS3stFM4SHifIIDMvKGr=
y8Fs/view?usp=3Ddrive_web" target=3D"_blank" style=3D"display:inline-block;=
max-width:366px;overflow:hidden;text-overflow:ellipsis;white-space:nowrap;t=
ext-decoration:none;padding:1px 0;border:none" aria-label=3D"4brons-final.m=
p4"><img style=3D"vertical-align: bottom; border: none;" src=3D"https://ssl=
.gstatic.com/docs/doclist/images/icon_10_generic_list.png">=C2=A0<span dir=
=3D"ltr" style=3D"color:rgb(17,85,204);text-decoration:none;vertical-align:=
bottom">4brons-final.mp4</span></a><img src=3D"//ssl.gstatic.com/ui/v1/icon=
s/common/x_8px.png" style=3D"opacity: 0.55; cursor: pointer; float: right; =
position: relative; top: -1px; display: none;"></div><div></div><div>Attach=
ed a file<br></div></div>

--000000000000fce15305c803f468--
```

3. Discoverability

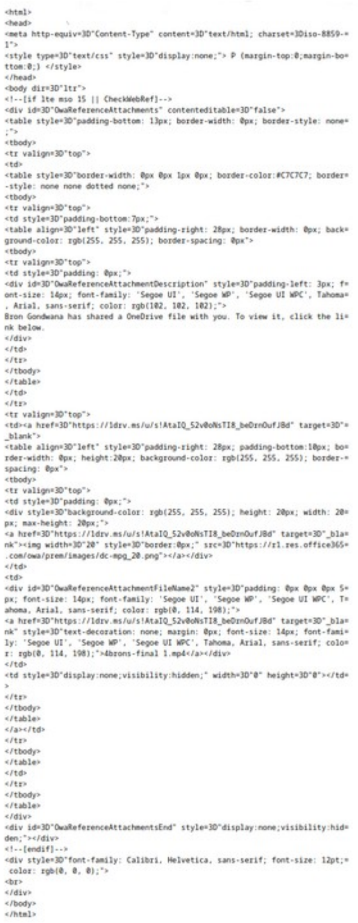
iCloud:



```
<html><body><div><div class=3D"x-apple-maildropbanner" data-expiration=3D"162988679200=
0" style=3D"font: 13px 'SFNSText', 'Helvetica Neue',sans-serif; color: #80=
8080; font-weight: 300; -webkit-text-size-adjust: 100%;"><div style=3D"hei=
ght:0px; border-top:1px solid #ececec; margin-right:-20px"></div><div styl=
e=3D"height:53px; overflow:auto"><div style=3D"float:left; margin-top:13px=
;; margin-right:62px"><div style=3D"height:19px; display:table-cell; verti=
cal-align:bottom">Attachment available until 25 August 2021.</div></div><d=
iv style=3D"float:right; width:62px; margin-left:-62px"></div></div><div s=
tyle=3D"height:0px; border-top:1px solid #ececec; margin-bottom:15px; marg=
in-right:-20px"></div></div><div><br data-mce-bogus=3D"1"></div><div  cont=
enteditable=3D"false"><div class=3D"x-apple-maildrop" data-url=3D"https://=
cvws.icloud-content.com/B/AbENWWsn4MeuifvEuEtZbnCKeZ1vAbc_ubslvYI13cAHOJfG=
J4oEJxff/4brons-final.mp4?o=3DAjMORjIY11n_LV5sEgsplEKI8KttyKwy6wivY8mbI1ED=
&amp;v=3D1&amp;x=3D3&amp;a=3DCAogaowjHA-jeolzYXMLAajFFyxX9GO1ubqIyVSDQKtPB=
vkSeBDk88-Sri8Y5IPL5rcvIgEAKgkC6AMA_37QY4pSBIp5nW9aBAQnF99qJszIH7bWIfkOo88=
rl1Rj0zDDBHynRd9jSTLQ9_-HYMpQUUKRfvcpciYL1Ee7tElrLLWqbf9qpuL-sv15zklImFKWD=
e1-d0X46Adlzjr_WQ&amp;e=3D1629886792&amp;fl=3D&amp;r=3D46a24c75-2a9a-40dc-=
8f99-70fbf3d2feaf-1&amp;k=3DCqxQt8pR-Hvum-QRtKCG0Q&amp;ckc=3Dcom.apple.lar=
geattachment&amp;ckz=3DApple-Webmail&amp;y=3D1&amp;p=3D62&amp;s=3DjeDLIitu=
-D5lC_1Fl7WpJFw2Gwo" data-filename=3D"4brons-final.mp4" data-size=3D"40686=
498" data-expiration=3D"1629886792000" data-recordid=3D"668e7902-d2c9-4956=
-a041-e1355868a247" style=3D"margin: 15px 15px 15px 0px; -webkit-text-size=
-adjust: 100%;border: 1px solid #CACACA; border-radius:15px;height:134px; =
display:inline-block; vertical-align: middle; min-width:159px; font-family=
:'SFNSText','Helvetica Neue',sans-serif; text-align: center;"><a target=3D=
"_blank" href=3D"https://www.icloud.com/attachment/?u=3Dhttps%3A%2F%2Fcvws=
.icloud-content.com%2FB%2FAbENWWsn4MeuifvEuEtZbnCKeZ1vAbc_ubslvYI13cAHOJfG=
J4oEJxff%2F%24%7Bf%7D%3Fo%3DAl2RZzMVlwjCYMLXTpd4GMgoOfdiSGjqsVVppNEjKWi2%2=
6v%3D1%26x%3D3%26a%3DCAogOHBMKtglkg9ERBOEp0h704g69Zhv35p9bjCb24GviS0SeBDu8=
8-Sri8Y7oPL5rcvIgEAKgkC6AMA_37QiStSBIp5nW9aBAQnF99qJtLdb9TWDpJfPh_RLlISdIf=
MOsiQci1tCywxpVITy3aXyGB1CeRWciaoBs32gsg5paPClJgOrxDqtjG4rkUG3aTsolm690_Sc=
Om626XxpQ%26e%3D1629886792%26fl%3D%26r%3D46a24c75-2a9a-40dc-8f99-70fbf3d2f=
eaf-1%26k%3DCqxQt8pR-Hvum-QRtKCG0Q%26ckc%3Dcom.apple.largeattachment%26ckz=
%3DApple-Webmail%26y%3D1%26p%3D62%26s%3DUK2OfVhtNyhmRzjVRTrOoPMYlv8&amp;uk=
=3D_&amp;f=3D4brons-final.mp4&amp;sz=3D40686498" style=3D"text-decoration:=
none; outline: 0; font-size: 14px; color: #007AFF; display:block; margin:4=
0px 10px 0px">Download from iCloud</a><div style=3D"text-align: center; fo=
nt-size:12px; color:#808080"><div style=3D"margin:4px 10px 0px">4brons-fin=
al.mp4</div><div style=3D"margin:3px 10px 0px">40.69 MB</div></div></div><=
/div></div></body></html>
--Apple-Webmail-86--6a537a90-7f21-4087-abbd-77b57f669c70--
```

3. Discoverability

Outlook:



```
<html>
<head>
<meta http-equiv=3D"Content-Type" content=3D"text/html; charset=3Diso-8859-=
1">
<style type=3D"text/css" style=3D"display:none;"> P {margin-top:0;margin-bo=
ttom:0;} </style>
</head>
<body dir=3D"ltr">
<!--[if lte mso 15 || CheckWebRef]-->
<div id=3D"OwaReferenceAttachments" contenteditable=3D"false">
<table style=3D"padding-bottom: 13px; border-width: 0px; border-style: none=
;">
<tbody>
<tr valign=3D"top">
<td>
<table style=3D"border-width: 0px 0px 1px 0px; border-color:#C7C7C7; border=
-style: none none dotted none;">
<tbody>
<tr valign=3D"top">
<td style=3D"padding-bottom:7px;">
<table align=3D"left" style=3D"padding-right: 28px; border-width: 0px; back=
ground-color: rgb(255, 255, 255); border-spacing: 0px">
<tbody>
<tr valign=3D"top">
<td style=3D"padding: 0px;">
<div id=3D"OwaReferenceAttachmentDescription" style=3D"padding-left: 3px; f=
ont-size: 14px; font-family: 'Segoe UI', 'Segoe WP', 'Segoe UI WPC', Tahoma=
, Arial, sans-serif; color: rgb(102, 102, 102);">
Bron Gondwana has shared a OneDrive file with you. To view it, click the li=
nk below.
</div>
</td>
</tr>
</tbody>
</table>
</td>
</tr>
<tr valign=3D"top">
<td><a href=3D"https://1drv.ms/u/s!AtaIQ_52v0oNsTI8_beDrnOufJBd" target=3D"=
_blank">
<table align=3D"left" style=3D"padding-right: 28px; padding-bottom:10px; bo=
rder-width: 0px; height:20px; background-color: rgb(255, 255, 255); border-=
spacing: 0px">
<tbody>
<tr valign=3D"top">
<td style=3D"padding: 0px;">
<div style=3D"background-color: rgb(255, 255, 255); height: 20px; width: 20=
px; max-height: 20px;">
<a href=3D"https://1drv.ms/u/s!AtaIQ_52v0oNsTI8_beDrnOufJBd" target=3D"_bla=
nk"><img width=3D"20" style=3D"border:0px;" src=3D"https://r1.res.office365=
.com/owa/prem/images/dc-mpg_20.png"></a></div>
</td>
<td>
<div id=3D"OwaReferenceAttachmentFileName2" style=3D"padding: 0px 0px 0px 5=
px; font-size: 14px; font-family: 'Segoe UI', 'Segoe WP', 'Segoe UI WPC', T=
ahoma, Arial, sans-serif; color: rgb(0, 114, 198);">
<a href=3D"https://1drv.ms/u/s!AtaIQ_52v0oNsTI8_beDrnOufJBd" target=3D"_bla=
nk" style=3D"text-decoration: none; margin: 0px; font-size: 14px; font-fami=
ly: 'Segoe UI', 'Segoe WP', 'Segoe UI WPC', Tahoma, Arial, sans-serif; colo=
r: rgb(0, 114, 198);">4brons-final 1.mp4</a></div>
</td>
<td style=3D"display:none;visibility:hidden;" width=3D"0" height=3D"0"></td=
>
</tr>
</tbody>
</table>
</a></td>
</tr>
</tbody>
</table>
</td>
</tr>
</tbody>
</table>
</div>
<div id=3D"OwaReferenceAttachmentsEnd" style=3D"display:none;visibility:hid=
den;"></div>
<!--[endif]-->
<div style=3D"font-family: Calibri, Helvetica, sans-serif; font-size: 12pt;=
 color: rgb(0, 0, 0);">
<br>
</div>
</body>
</html>
```

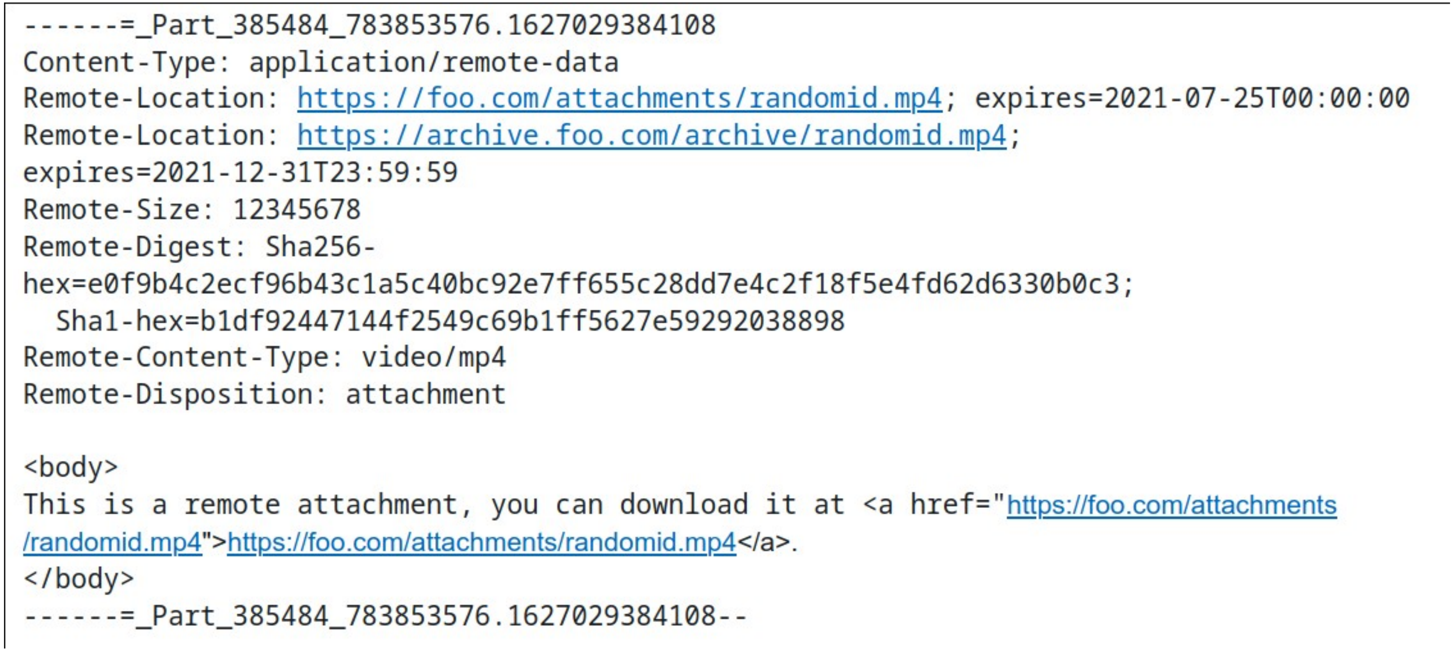
3. Discoverability

- Arbitrarily following links from emails is dangerous (GET side effects, content can change, timing/privacy leaks)
- Hard to know which links are “attached files”
- Lifetime uncertain
- Content can change after delivery (bypass virus and spam checks)

Proposal

- Upload work – to one of the http groups
- Ownership/Discovery: new MIME type, something like: application/remote-content
- Digest for content integrity protection and immutability
- Expiry timestamp tied to link
- UUID for deduplication and data lifetime
- Content-ID linkability within the message
- Recipient server SHOULD download a copy to virus check and then retain it for the lifetime of the email

e.g.



```
------=_Part_385484_783853576.1627029384108
Content-Type: application/remote-data
Remote-Location: https://foo.com/attachments/randomid.mp4; expires=2021-07-25T00:00:00
Remote-Location: https://archive.foo.com/archive/randomid.mp4;
expires=2021-12-31T23:59:59
Remote-Size: 12345678
Remote-Digest: Sha256-
hex=e0f9b4c2ecf96b43c1a5c40bc92e7ff655c28dd7e4c2f18f5e4fd62d6330b0c3;
  Sha1-hex=b1df92447144f2549c69b1ff5627e59292038898
Remote-Content-Type: video/mp4
Remote-Disposition: attachment

<body>
This is a remote attachment, you can download it at <a href="https://foo.com/attachments
/randomid.mp4">https://foo.com/attachments/randomid.mp4</a>.
</body>
------=_Part_385484_783853576.1627029384108--
```

Conclusion

- Has this already been solved somewhere and I just don’t know where to look?
- Am I barking up entirely the wrong tree?
  - I assume SMTP isn’t going to allow gigabyte files
  - I assume users are going to keep wanting to email larger and larger files.
  - I assume that email recipients want the lifetime of the files they have been sent to be identical to the emails themselves (so if they don’t delete the email, they don’t lose the file)
  - We need this for calendar event attachments too… and presumably every other messaging system has the same problem.
- Is there interest in doing work in this space?
  - What is a sensible group for it if so?
  - Who wants to get involved?

CELLAR

Codec **E**ncoding for **L**oss**L**ess **A**rchiving and **R**ealtime transmission

Michael Richardson
<mcr+ietf @ sandelman.ca>
Spencer Dawkins
<spencerdawkins.ietf@gmail.com>
Co-chairs

Running Code

- Matroska and EBML go back to 2002
  - Google WebM is fork
- Matroska is a container for multimedia, widely supported, but like a lot of open source, has some issues
- -> need for Rough Consensus

The story so far

- Following in the success of OPUS, a WG was formed in 2015.
- Published RFC7894 (EBML) in 2020
- FFv1 (v0,1,3) in RFC-editor Queue (RFC9043)
- FFv1 v4 being worked on
- Matroska document pretty firm, probably finish in 2021.
- Almost all CELLAR WG meetings have been virtual interims, fourth Tuesday of the month.
  - Open source authors, mostly working after hours
  - But, also includes archivists for whom this is part of their day job

What next?

- FFV1 v4 goals include
  - Better Compression and or Speed
  - BAYER support to efficiently store RAW color CCD images
  - Better error resilience
  - Arbitrary color spaces
  - Limit worst case size after compression
- Arbitrary color spaces: Allow storage of any 2D plane of samples not limited to red, green, blue, alpha or transforms of these.
- Examples: Infrared, Radar, surface vectors, height, age, temperature, charge, elasticity, velocity, acceleration
- Also: attestation.
  - What format should autonomous vehicles make their records in, such that they are compatible with courts world wide?

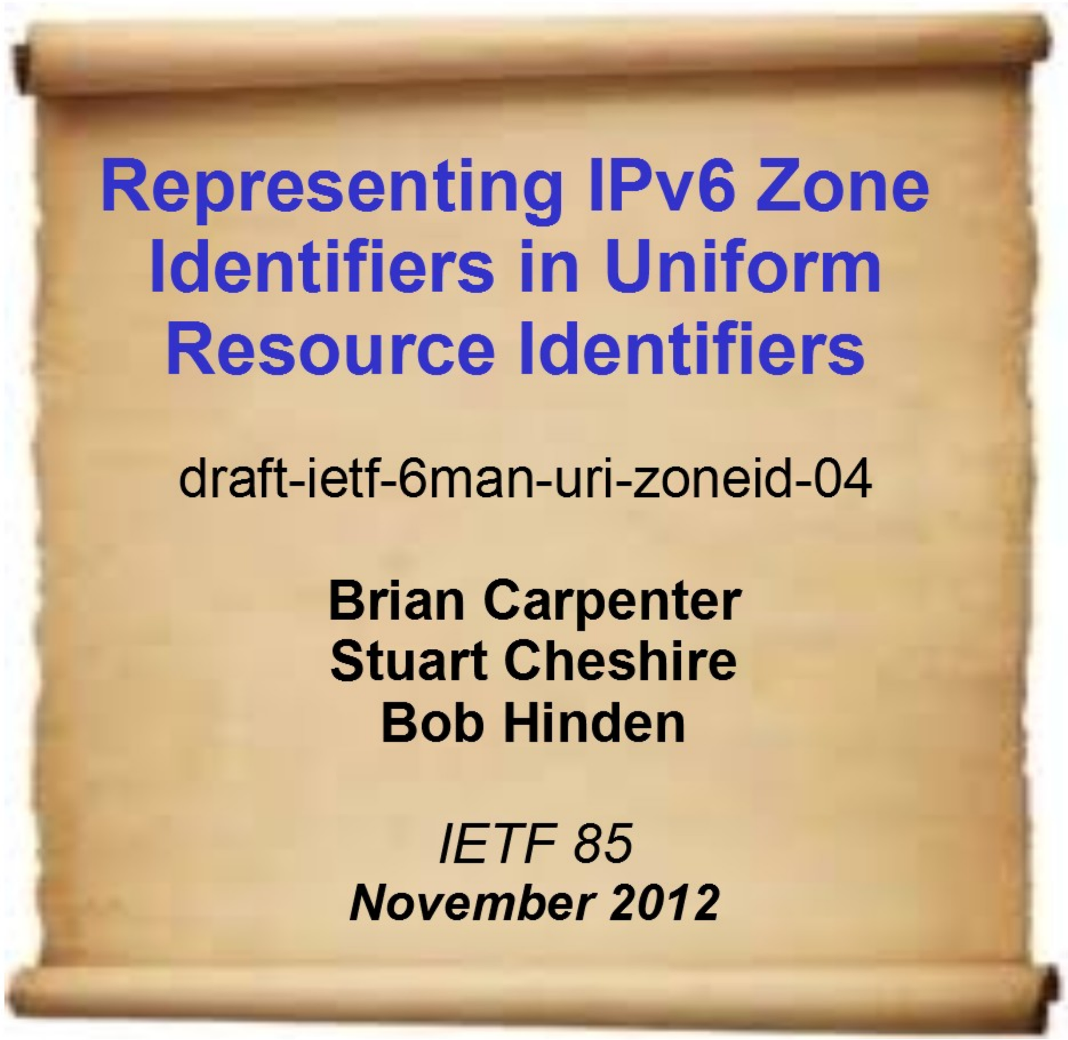
Representing IPv6 Zone Identifiers in Uniform Resource Identifiers

draft-carpenter-6man-rfc6874bis

Brian Carpenter
Bob Hinden

IETF 111
July 2021

We discussed this just recently...



Representing IPv6 Zone Identifiers in Uniform Resource Identifiers

draft-ietf-6man-uri-zoneid-04

Brian Carpenter
Stuart Cheshire
Bob Hinden

IETF 85
November 2012

... which became RFC 6874

Motivation

- Literal addresses in URIs are mainly intended for operational and diagnostic use.
- Sometimes, there is a need to make tests that relate to IPv6 link local addresses via a specific interface on the host.
  - A web browser may be the handiest tool for this
  - It may be the only tool for reconfiguring misconfigured devices
- At least one application (CUPS printing) requires HTTP usage of link local addresses via a specific interface.

Fail

- For link-local addresses, RFC 4007 defines a text representation of the Zone Identifier (in practice equal to an interface name):

  ```
  fe80::abcd%eth0
  ```

- Widely supported and used in IPv6-land
- RFC 6874 defined a mapping for the Zone ID in URI syntax.
  - No known current browsers support it.
  - The browser community (WHATWG) decided this explicitly.

Problems with RFC6874 (1)

- Modifies the IP-literal branch of the ABNF for URIs (RFC 3986)
  - **`http://[fe80::abcd%25eth0]`** becomes legal
  - using %25 as separator, i.e. RFC 4007 notation with URI escaping
- This prevents cut and paste (e.g. from ping to URL)
  - Arguably, required because % is always an escape character in URIs
  - Arguably, unnecessary if parsers follow ABNF rigidly
- <u>Proposal</u>: no change

Problems with RFC6874 (2)

- Requires hosts to delete the Zone ID from outgoing URIs (in the HTTP *Host* header)
  - Violates the normal behaviour of HTTP/1.1 (RFC 7230)
  - At the least, awkward to code
  - Breaks CUPS

- <u>Proposal</u>: delete this requirement

Problems with RFC6874 (3)

- Suggests URL parsers should support the %25 encoding but heuristically accept (for example)

  ```
  fe80::abcd%eth0
  ```

  instead of

  ```
  fe80::abcd%25eth0
  ```

  - Very tricky to code
  - Confusing to users

- Proposal: delete this suggestion

Feedback requested

- IPv6 community, including operators
- ART Area
- W3C
- Browser implementers, including WHATWG

RUSH
Reliable (Unreliable) Streaming protocol

Kirill Pugin, Facebook

Motivation

- Applications have different latency requirements
  - Live streaming of soccer match may be ok with 10-30 seconds latency
  - Interactive live streams like Gaming would benefit from lower latency (< 5 seconds)
- Extensibility
  - New audio/video codec support
  - New client-server interactions
  - Multi-strack support, including captions

Motivation

- Reliability
    - Disconnects can happen due to network change, errors or server maintenance
- Quality
    - Better signals from network to adjust audio/video bitrate to network conditions (Adaptive Bitrate selection)

Motivation (continued)

- RTC – focused on P2P and low latency upload, doesn’t give a choice between latency and quality.
- RTMP – old, not flexible – no new codec support, some implementations don’t support reconnect.
- DASH – doesn’t allow per-frame level control - hard to control latency.

RUSH

RUSH is a bidirectional application level protocol designed for live video ingestion that runs on top of QUIC.

RUSH was built as a replacement for RTMP (Real-Time Messaging Protocol) with the goal to provide support for new audio and video codecs, extensibility in the form of new message types, and multi-track support.

In addition, RUSH gives applications option to control data delivery guarantees by utilizing QUIC streams.

Wire format

Client and server exchanges information using frames. Frames can be different types and data passed within a frame depends on its type.

Generic frame format:

```
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+-- ...
| 0 : 1 : 2 : 3 : 4 : 5 : 6 : 7 | 8 : 9 : 10: 11: 12: 13: 14: 15| 16|
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+-- ...
|            LENGTH             |              ID               | T |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+-- ...
```

Wire format

- **LENGTH(64):** Each frame starts with length field, 64 bit size that tells size of the frame in bytes.

- **ID(64):** 64 bit frame sequence number, every new frame MUST have a sequence ID greater than that of the previous frame within the same track.

- **TYPE(8):** 1 byte representing type of the frame.

Frames

RUSH defines 7 frames types:

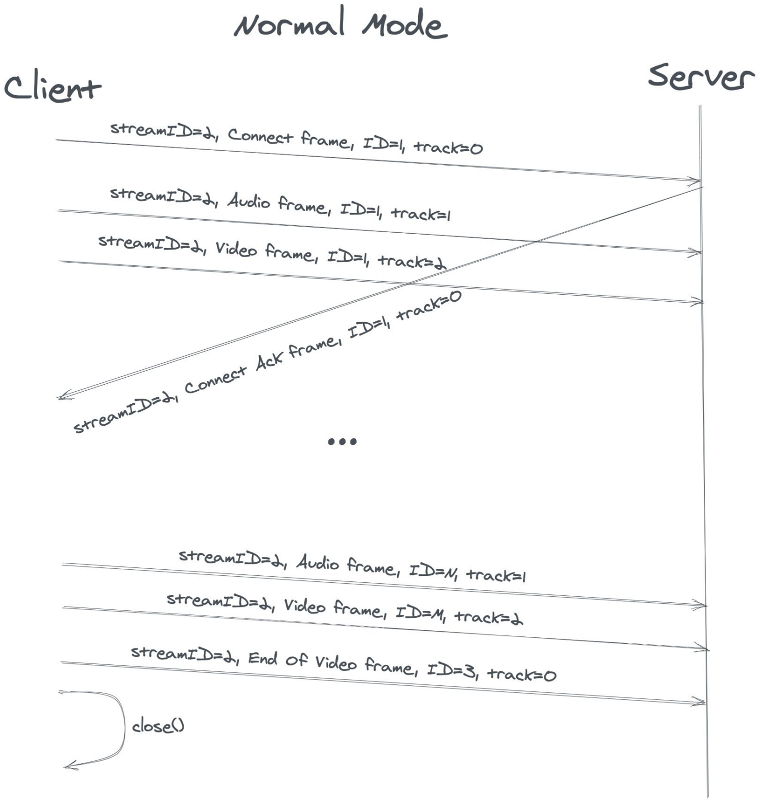
- Connect frame
- Connect Ack frame
- End Of Video frame
- Error frame
- Audio frame
- Video frame
- GoAway frame

How it works

There are two modes defined:

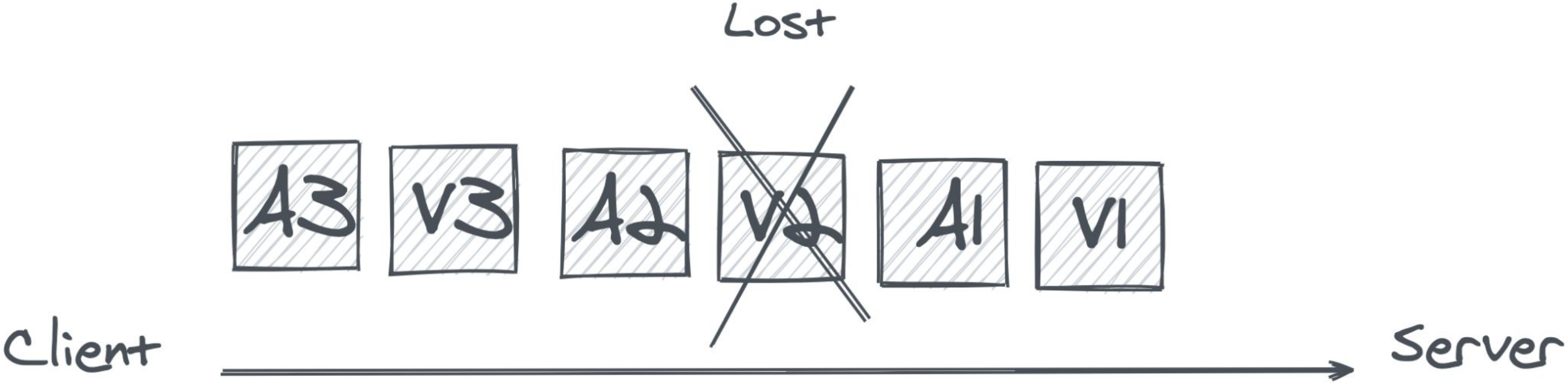
- **Normal mode**: RUSH uses one bidirectional QUIC stream to send data and receive data. Using one stream guarantees reliable, in-order delivery - applications can rely on QUIC transport layer to retransmit lost packets. The performance characteristics of this mode are “similar” to RTMP over TCP.

Normal mode


Normal Mode
Client
Server
streamID=2, Connect frame, ID=1, track=0
streamID=2, Audio frame, ID=1, track=1
streamID=2, Video frame, ID=1, track=2
streamID=2, Connect Ack frame, ID=1, track=0
...
streamID=2, Audio frame, ID=N, track=1
streamID=2, Video frame, ID=M, track=2
streamID=2, End of Video frame, ID=3, track=0
close()

- Client sends **Connect** frame on bidirectional QUIC stream
- Client sends audio and video data on the same QUIC stream
- Only one video can be send on the same Connection
- Server replies with **Connect Ack** frame on the same QUIC stream
- Frames arrive **in order**
- Client sends **End Of Video** frame to indicate that video is done

Normal mode (what can go wrong?)

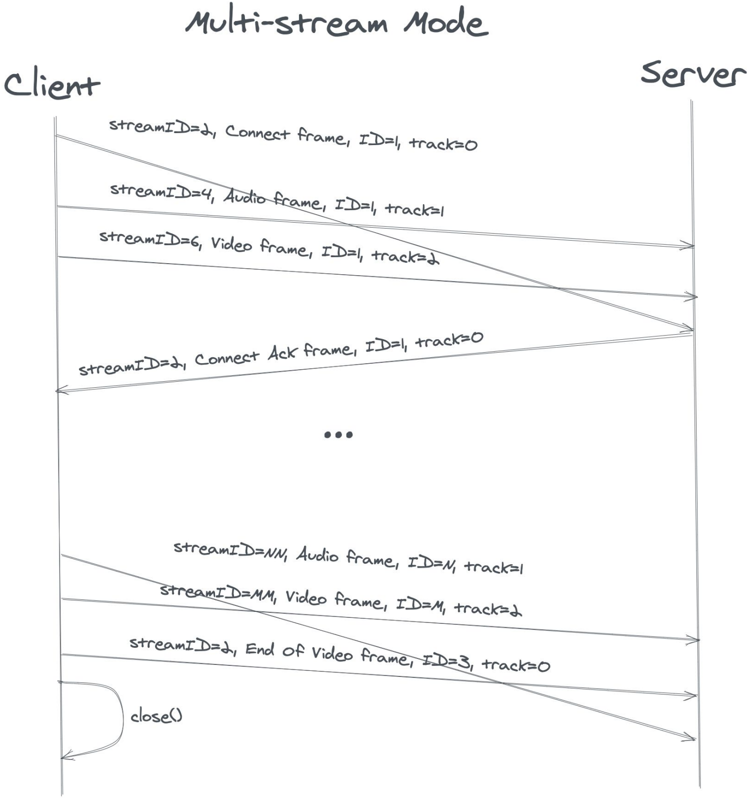


If **V2** frame is lost all frames sent after it will be not available to the server, until **V2** retransmitted - this is variation of head of line blocking and can affect latency and introduce jitter.

How it works

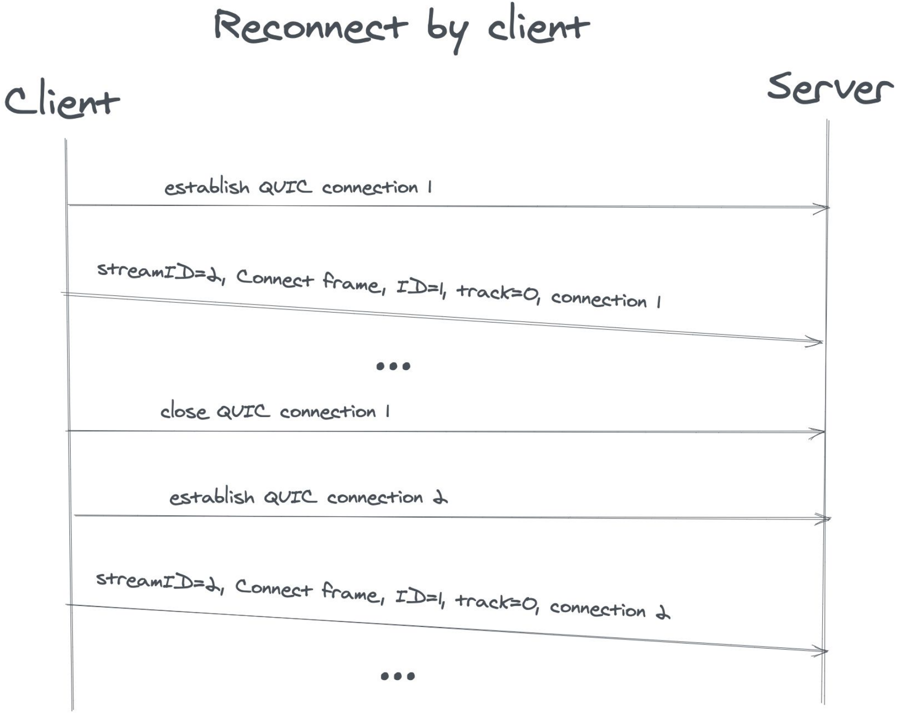
- **Multi-stream mode:** To address head of line blocking and also to give more control to application over delivery guarantees, in multi-stream mode, every new frame is sent on new QUIC bidirectional stream. Since QUIC streams are independent of each other this allows server receive data as it arrives and not wait for retransmissions of lost packets.

Multi-stream mode


Multi-stream Mode
Client
Server
streamID=2, Connect frame, ID=1, track=0
streamID=4, Audio frame, ID=1, track=1
streamID=6, Video frame, ID=1, track=2
streamID=2, Connect Ack frame, ID=1, track=0
...
streamID=NN, Audio frame, ID=N, track=1
streamID=MM, Video frame, ID=M, track=2
streamID=2, End of Video frame, ID=3, track=0
close()

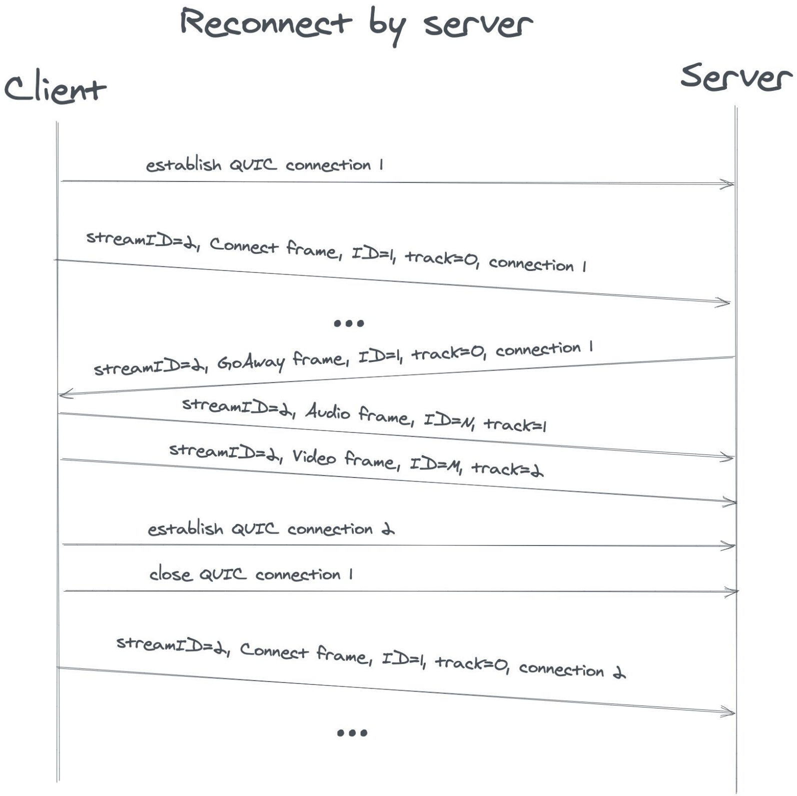
- Frames arrive out of order
- Server uses frame IDs within a track to detect missing frames, it’s up to server to “restore” order
- Client can stop retransmission by resetting the corresponding QUIC stream

How it works (reconnect)


Reconnect by client
Client
Server
establish QUIC connection 1
streamID=2, Connect frame, ID=1, track=0, connection 1
...
close QUIC connection 1
establish QUIC connection 2
streamID=2, Connect frame, ID=1, track=0, connection 2
...

- Client opens new QUIC connection
- Client closes currently used QUIC connection.
- Client follows normal connect flow and continue sending data on new QUIC connection

How it works (reconnect)


Reconnect by server
Client
Server
establish QUIC connection 1
streamID=2, Connect frame, ID=1, track=0, connection 1
...
streamID=2, GoAway frame, ID=1, track=0, connection 1
streamID=2, Audio frame, ID=N, track=1
streamID=2, Video frame, ID=M, track=2
establish QUIC connection 2
close QUIC connection 1
streamID=2, Connect frame, ID=1, track=0, connection 2
...

- Server sends GoAway method
- Client may send frames one current connection
- Client establishes new connection
- Client follows normal connect flow and continue sending data on new QUIC connection
- **NOTE:** server may close connection after sending GoAway, but before client finished sending frames on that connection - this may result in data loss

Questions?

Meetings of Interest

- BoFs:
  - Tues 1900 UTC **DANE AutheNtication for Iot Service Hardening (DANISH)**
  - Tues 2300 UTC **Oblivious HTTP (ohttp)**
  - Weds 1900 UTC **MAC Address Device Identification for Network and Application Services (MADINAS)**
  - Thurs 2030 UTC **SCIM Industry Next Steps (SINS)**
  - Fri 1900 UTC **Application-aware Networking (APN)**
- New ART WGs:
  - Tues 1900 UTC **Serialising Extended Data About Times and Events (SEDATE)**

Meetings of Interest

- Other Interesting Meetings (all times UTC):
  - Mon 2300 Security Dispatch (secdispatch)
  - Weds 1900 General Area Dispatch (gendispatch)
  - Weds 2130 Stay Home Meet Only Online (shmoo)
  - Thurs 2030 RFC Editor Future Development Program (rfcefdp)
  - Thurs 2330 IAB Open Meeting (iabopen)