Uplink media delivery in 5G: Protocols & Encoding

This 5G-MAG report complements the publication Uplink media delivery in 5G: Architectures & Features [1] which identified, described and analysed relevant systems and features enabling the enhancement of uplink media delivery and traffic management in the content of media production and contribution.

This report expands on the following aspects:

- Transport protocols and features available for uplink media delivery.
- Identification of potential enhancements for the delivery of content over 3GPP systems.
- Identification of potential enhancements to video encoding.

Contents

Transport protocols and potential enhancements 2
SRT (Secure Reliable Transport) 3
SST (Safe Streams Transport) 4
RIST (Reliable Internet Stream Transport) 5
WebRTC (Web Real-Time Communication) 6
QUIC 7
Potential encoding improvements 11
Related documentation 13

Transport protocols and potential enhancements

A comparison matrix of features in commonly used transport protocols is presented below leveraging those already defined in 3GPP TR 26.805 [2].

Feature		SRT	SST	NDI	RTP profiles				QUIC	
					SMPTE ST2110	RIST	IPMX	WebRTC	RTP over QUIC	Media over QUIC (MoQ)
Public specification		Yes	No	No	Yes	Yes	Yes	Yes	Yes	Yes
Based on protocol		UDT/UDP	UDP	TCP or UDP	RTP/UDP	RTP/UDP	RTP/UDP	SRTP/UDP	RTP/QUIC/UDP	QUIC/UDP
Interoperability		Wide support	Proprietary	Proprietary	Wide support	Good	In progress	Wide support	In progress	In progress
Loss detection		Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Loss recovery/ Error correction	ARQ	Yes	Dynamic	Optional	No	Yes	No	Optional	Optional	Optional
	FEC	Optional	No	Optional	Optional	Optional	No	Optional	No	No
Encryption		Optional (AES)	Optional (AES)	Optional	No	Optional (DTLS)	Optional (AES)	Yes (SRTP)	Yes (TLS 1.3)	Yes (TLS 1.3)
Multicast		No	No	Yes	Yes	Yes	Yes	No	No	No
Link bonding	Higher throughput	No	Yes	Yes	Yes	Yes	Yes	No	Future	Future
	Redundancy	Yes	Yes	Yes	Yes	Yes	Yes	No	Future	Future
Codec support		Agnostic	Agnostic	H.264, H.265 or Proprietary	Specified as Payload Format for RTP			Minimum mandatory set	Agnostic	To be decided
Codec packaging		Typically MPEG-2 TS [3]	Typically MPEG-2 TS [3]	Discrete streams	Elementary streams	Typically MPEG-2 TS [3]	Elementary streams or MPEG-2 TS [3]	Elementary streams	Elementary streams or MPEG-2 TS [3]	To be decided
Timing/synchronization		Asynchronous	Asynchronous	Both	Synchronous	Asynchronous	Both	Asynchronous	Asynchronous	Asynchronous

SRT (Secure Reliable Transport)

SRT (Secure Reliable Transport) is an open-source transport protocol developed by Haivision [4][5] used to transport data and media over unreliable connections. A GitHub repository is also available in [6].

SRT is able to transport multiple different types of content and is agnostic to the encoding formats and resolution. Typically SRT transports multiplexed compressed audio and video data in an MPEG-2 Transport Stream to provide in-band audio/video synchronization. The compressed video streams employ H.265 or H.264 codecs with Constant Bit Rate (CBR) encoding. The video quality is automatically adjusted to match available network capacity. Password-based encryption of streams is available, employing 128-/256-bit AES stream encryption/decryption.

SRT employs a modified version of the UDP-based Data Transfer (UDT) protocol [7] that utilises send and receive buffers with a re-transmission mechanism for lost packets. The SRT packets employ UDP [8] with an SRT header field which includes a timestamp. The SRT header identifies control and data packet types. Control packets implement the fast re-transmit mechanism, indicating positive acknowledgement (ACK), negative acknowledgement (NAK) or a Fake ACK for data packets. The data packets carry the encapsulated media data.

An SRT connection is configured with a latency value, expressed in milliseconds, which determines a re-transmission window. This value determines the receive buffer size, while already-transmitted packets are stored in a send buffer of the same size. Transmitted packets missing from the receive buffer are flagged and re-transmitted. Packet loss can be monitored, and the programmable latency adjusted to compensate for network path delay variation. The send/receive buffer size can be configured to any value between 20 ms and 8000 ms. For media contribution, typical SRT programmable buffer values vary from approximately 200 ms to 1700 ms.

When using SRT over 3GPP systems, QoS handling and traffic filtering can be realized using a conventional 5-tuple association.

SST (Safe Streams Transport)

SST (Safe Streams Transport) [9] is one of several proprietary IP bonding technologies employed for media contribution over 3G/4G/5G cellular, LAN, Wi-Fi, satellite and the public Internet. SST simultaneously aggregates multiple network connections, dynamically adapts video bit rate according to network bandwidth fluctuations, protects stream content and supports the retransmission of lost data.

SST can also transmit bi-directional data, is able to transport multi-types of content and is agnostic to the encoding formats and resolution. Ancillary streams for file transfer, remote control, return video, talkback (intercom) and IP data are supported. Streams are transported using multiple simultaneous connections, whether 3G/4G/5G cellular, LAN, Wi-Fi or other, with packet re-ordering and ARQ. It is also the ability to prioritise the use of a specific connection.

SST is based on UDP and supports Automatic Repeat Request (ARQ) or Adaptive Forward Error Correction (A-FEC). If ARQ is enabled, lost packets are re-transmitted at cost of increased latency. If adaptive A-FEC is enabled, extra packets containing error correction data are transmitted, allowing lost packets to be recovered without re-transmission and the latency penalty. The error correction automatically adapts to match the quality of the connection. The error correction coding in the form of a two-dimensional array of packet data of variable size. For example, a 4×4 array of packets (16 packets of data) is encoded by row and column generating 8 packets of FEC data. At the most robust A-FEC setting, a loss of up to 20% of packets can be recovered.

For broadcast contribution, a target latency and bit rate are set. The encoding, ARQ and error correction are then configured for the target values, but adapt as the connection quality varies. Variable resolution can also be enabled. Multiple cameras can be synchronised, so that video streams are synchronised, reducing the buffering required for frame accurate switching.

When using SST over 3GPP systems, QoS handling and traffic filtering can be realized using a conventional 5-tuple association. It may be possible to handle QoS per independent connection within the network. However, even when multiple connections run on the same network it may not be possible to optimize overall traffic as the bonding mechanisms are realized between client and server and remain transparent to the network.

RIST (Reliable Internet Stream Transport)

RIST (Reliable Internet Stream Transport) is an open specification that seeks to provide reliable, high performance media transport, using RTP/UDP, over public IP networks. RIST was originally developed by the VSF (Video Services Forum) RIST Activity Group [10] but is now supported by the RIST Forum [11].

RIST is available as open-source code [12] and has been incorporated into a number of other projects, such as VLC, FFMPEG, GStreamer, OBS Studio and Wireshark. A key aim in the development of RIST is interoperability and certified RIST solutions are available from multiple vendors [13].

RIST specifies several profiles, each adding more capabilities. This makes it easier for vendors to add RIST interoperability with the simplest profile and then add functionality using the more advanced profiles.

The most basic interoperability profile, **RIST Simple Profile**, is described in TR-06-1 [14] covering:

- Interoperable Automatic Repeat reQuest (ARQ) based on RTCP with configurable behaviour for:
  - Recovery of packet loss, packet reordering and link failure using bit mask or range-based Negative Acknowledgement.
  - Removal of network introduced jitter.
- Transport of point-to-point SMPTE 2022-2 services.
- Bonding of several links using link aggregation.
- Seamless switching using SMTPE 2022-7.
- Optional Forward Error Correction using SMPTE 2022-1.
- Out-of-band transmission of protection data (retransmissions may use a separate link).
- RTT Echo Request / Response procedure to estimate the round-trip time.

RIST Main Profile, described in TR-06-2 [15], adds the following:

- Transport of point-to-multipoint services
- Stream encryption for secure content
- VPN tunnelling for secure sender/receiver communication
- NAT traversal for improved interworking with consumer-style internet connectivity
- Null packet suppression for saving bandwidth
- Transport of high bandwidth streams (>100Mbps) for uncompressed or lightly compressed content
- Auto stream configuration for simpler operations

The **advanced or enhanced profile**, described in TR-06-3 [16], will add:

- Enhanced tunnelling capabilities
- Enhanced PSK Security
- Direct payload transport and Protocol Registry
- Flow Attributes

Vendor equipment is in general certified for the simple and main profiles. However, not all vendors are certified for all aspects of the main profile.

When using RIST over 3GPP systems, QoS handling and traffic filtering can be realized using a conventional 5-tuple association.

WebRTC (Web Real-Time Communication)

WebRTC [17] is a mechanism for allowing direct peer-to-peer communication primarily between standard web browsers. This can be used for file exchange, but also for exchanging live and interactive media sessions between two or more participants.

A series of JavaScript APIs is defined for web applications to access the WebRTC capabilities of the browser [18]. Therefore, a web page serves as the entry point for a conventional WebRTC session by containing the information needed to access the services described. However, it is possible for non-browser endpoints, such as a mobile app or dedicated device, to participate in a WebRTC session.

WebRTC media sessions make use of SDP offer/answer [19] to negotiate RTP sessions between endpoints over a WebRTC data channel, as defined in IETF RFC 8825 [18]. The 3GPP specifications introduce many features that are designed to improve the carriage of WebRTC media sessions over the 5G System.

When negotiating media sessions using SDP offer/answer, it is possible to negotiate maximum bit rates for each endpoint to use. Using the Network Support Function (NS-AF) of the RTC AF it is possible for the initiating RTC endpoint to request a bit rate recommendation that it can use in the SDP offer.

In public deployments of the 5G System (e.g. PLMN or PNI-NPN) peer-to-peer traffic between UEs is typically not permitted and traffic between UEs must be relayed via an Application Server. In the RTC System, the Selective Forwarding Unit functionality of the RTC AS Media Function provides this relay functionality when WebRTC sessions are carried in the 5G System. This restriction is less likely to be an issue in S-NPN deployments, because in this case the network operator has full control of the network configuration.

Once an RTC session has been negotiated, the RTC endpoint uses Interactive Connectivity Establishment (ICE) to discover the means to convey WebRTC traffic between itself and its peer via the ICE Function of the RTC AS. The ICE Function may provide a STUN server [20] that allows an endpoint to discover the NAT address mapping by the network and/or a TURN server [21] that relays WebRTC traffic directly.

WebRTC media sessions use Secure Real-time Transport Protocol (SRTP) [22] to carry media over UDP/IP. SRTP is a profile of RTP [23] which encrypts the RTP payload to provide confidentiality for the media transported between endpoints.

The RTP header fields are not encrypted and can still be seen by devices on the network such as the RTC AS and (using deep packet inspection) the UPF.

3GPP TS 26.522 [24] defines an RTP header extension to mark packets in an RTP session as belonging to a **PDU Set**. The UPF is able to inspect this labelling and expose it to gNodeB instances by copying the labels into GTP-U tunnel packet headers with the aim of preserving groups of packets belonging to the same media access unit intact when Radio Access Network capacity is constrained.

QUIC

QUIC [25] is a UDP-based, stream-multiplexing, loss-detecting and encrypted transport protocol developed by the IETF with the primary use case being HTTP/3 [26] but also being a general-purpose transport protocol suitable for other uses.

All data-carrying packets in a QUIC connection are acknowledged by the receiver. QUIC supports both reliable and unreliable transport of data:

- QUIC **streams** allow the multiplexing of independent data flows and guarantees reliable in-order delivery of data carried on each logical stream to the receiving application. A timeout-based Automatic Repeat Query (ARQ) mechanism corrects for packet loss on each stream without interfering with the delivery of data on other streams carried within the same QUIC transport session.
- Applications can also use the QUIC **datagram** extension [27] to send data unreliably between peers in a QUIC transport session. The application is informed of any packet losses detected by the transport layer and it can then choose whether to perform loss recovery itself at the application layer. Unlike streams, the datagram extension does not specify a means to multiplex flows within a QUIC connection.

Feature	Acknowledged	Recovery	Multiplexing
Streams	✔	✔	✔
Datagrams	✔	✘	✘

The IETF QUIC Working Group was originally chartered to include support for **Application Layer FEC (AL-FEC)** in the protocol, but this was descoped during development for reasons of time. Several individual attempts at AL-FEC have been published, including the most recent draft [28].

Specifications extending QUIC to **multicast** distribution have also been contributed to the QUIC Working Group. [29] adapts the QUIC framing format to a purely multicast usage and specifies the optional use of content digests and digital signatures to assert the integrity and authenticity of objects delivered over multicast. [30] associates a “multicast channel” with a conventional unicast QUIC transport connection, with acknowledgements and packet hashes exchanged over the unicast connection.

Congestion Control in QUIC

Because QUIC uses UDP, it tends to be implemented in user space instead of kernel space (as is the case with TCP and SCTP). This affords implementations with a lot more flexibility in selecting a congestion control algorithm to use for a QUIC transport session.

The QUIC specification does not mandate the use of any specific congestion control scheme, and it also does not preclude swapping the congestion control algorithm in use by an endpoint during the lifetime of a QUIC connection. Different congestion control schemes can even be used in each direction of a QUIC transport session, which may be useful if the uplink and downlink path of the connection have different characteristics. With these points in mind, it is recommended that implementations for media contribution and delivery use a congestion controller better suited for live media, such as Copa [31] or SCReAM [32].

Explicit Congestion Notification (ECN) as originally defined in IETF RFC 3168 [33] is a means to notify the recipient of an IP packet that congestion has been encountered on the network path using two bits of the IP packet header. IETF RFC 9331 [33] refines the classic ECN signalling protocol for use in Low-Latency, Low Loss, and Scalable throughput (L4S) services. In both cases, it is the responsibility of the receiving host to notify the sender that congestion has been experienced using a protocol above layer 3 (e.g. by using the ECN echo bit in the TCP packet header, or the RTCP XR ECN Summary Reports as defined in [35] in the case of an RTP or SRTP session), and the sender should react appropriately to prevent future packet loss resulting from the congestion.

As described in section 13.4 of [25], a QUIC endpoint reports the number of received packets that experienced congestion to its connection peer in an extended acknowledgement frame. Because this is carried inside the encrypted QUIC connection, the ECN status of a connection is not visible to middleboxes on the connection path in contrast to a TLS session carried over TCP, where the ECN echo bit is visible in the TCP packet header.

Considerations on use of QUIC in the User Plane

The QUIC transport protocol is deliberately designed to prevent pervasive monitoring of Internet traffic by nodes in the network path between the transport peers ("middleboxes"). Hence, all packets comprising a QUIC connection are encrypted, including transport-level interactions such as acknowledgements and ECN feedback.

This represents an obstruction to **Deep Packet Inspection** (DPI) techniques commonly in use today by the User Plane Function (UPF) of the 5G System to detect TCP and UDP traffic flows as part of its QoS Flow management and Lawful Intercept responsibilities.

Associations between a client and server pair of QUIC endpoints are called "paths" in the QUIC transport specification and are realised using a conventional 5-tuple association. In version 1 of the QUIC protocol, client endpoints may take advantage of a **Connection Migration** feature as specified in section 9 of [25]. This allows a client endpoint to migrate an established connection between network interfaces available to it without notice, such as migrating from Wi-Fi to a 5G connection when the UE moves away from a Wi-Fi access point.

The connection migration feature is not a mandatory feature and requires QUIC endpoints to negotiate their support for it at the time of transport connection establishment. If sudden migration of a connection is undesirable for a particular use case, a Media Client (5GMS Client or RTC Client) can simply disable the feature, therefore. However, this neglects a powerful feature of the QUIC transport protocol. A more robust solution would be for the Media Session Handler to update the service data flow information associated with dynamic QoS policies and/or Network Assistance sessions in response to connection migration events.

Multipath QUIC [36] is under active development by the IETF QUIC Working Group and allows a QUIC transport connection to utilise multiple paths for packet delivery. It is inherently difficult to discern between single- and multi-path QUIC traffic as it moves through the network, with only the endpoints involved aware of the multipath mapping.

ATSSS, as specified in TS 23.501 [37] Release 18, is a technology for sharing traffic between a single 3GPP access network and a single non-3GPP access network (such as Wi-Fi). This restriction precludes the use of ATSSS to bond multiple 3GPP access networks together on a device with multiple SIM cards.

Multipath QUIC with HTTP/3 and support for HTTP datagrams [38] and the extended CONNECT method [39] can be used as the transport session for an ATSSS session, as specified in clause 5.32.6.2.2 of TS 23.501 [37] and clause 6.1.4.1.1 of TS 24.193 [40]. However, ATSSS is designed as a tunnelling mechanism for PDU sessions that are not aware of the complexities of the underlying transport system, and as such details of how to apply QoS rules may not be appropriate for multipath QUIC flows which continue beyond the boundaries of the 5G System.

The feasibility study on the use of the DSCP/ToS Traffic Class field (see clause 5.3.4.3 of TR 26.804 [41]) should be revisited to address the interaction with QUIC transport connections.

Media transport over QUIC

As previously mentioned, HTTP/3 uses QUIC as its underlying transport protocol. Therefore, any media application that relies on HTTP can be made to work over QUIC transport through the use of HTTP/3, including the HTTP-based protocols to be specified in TS 26.512 [42] for the contribution of content by the 5GMS Client to the 5GMSu AS at reference point M4u, and for its subsequent egest from the 5GMSu AS to the 5GMSu Application Provider at reference point M2u.

Two separate approaches to delivering packet-based media streams over QUIC have been adopted by different working groups within the IETF:

- **RTP-over-QUIC** [43] defines a way of carrying existing RTP-based payloads on either reliable QUIC streams or unreliable datagrams.
- **Media-over-QUIC (MoQ) transport** [44] is a work in progress seeking to develop media handling application protocols from scratch, either over QUIC transport directly or using WebTransport over HTTP/3 [45].

These approaches are described in more detail below. Either or both may find their way into 3GPP specifications in future releases.

RTP-over-QUIC

RTP-over-QUIC does not change the syntax of the RTP packets which are carried over it. It can support reliable delivery of RTP frames in QUIC streams, or it can carry them (unreliably) as QUIC datagrams.

TS 26.506 [46] describes how to perform real-time media contribution in the 5G System, adopting the WebRTC framework to allow endpoints to exchange media using RTP streams alongside application data sent over WebRTC data channels. To support these real-time flows, some components of the system rely on being able to read RTP header extensions as they pass through. However, encapsulation of the RTP packets in encrypted QUIC packets prevents the UPF from seeing the information conveyed in these header extensions for downlink traffic.

The UE modem may be able to directly apply PDU Set labelling to uplink GTP-U tunnel packets when an uplink RTC session uses RTP-over-QUIC. This

ensures sympathetic handling by the RAN uplink of packets belonging to the same PDU Set. In the case of RTC, the QUIC transport connection terminates in an Application Server (the RTC AS), and this can directly gain access to the RTP header extensions after the QUIC layer decrypts its payloads. The RTC AS then proxies the data onto the intended recipient UE over a downlink RTP-over-QUIC transport connection as described by the Transport Relay topology described in section 3.2.1.1 of RFC 7667; or forward the RTP packets over a different transport protocol as described by the Transport Translator topology described in section 3.2.1.2 of RFC 7667 [47].

However, there is no mechanism defined at present to enable an Application Server to pass the content of RTP header extensions on to the UPF, for example to drive PDU Set handling.

Media-over-QUIC (MoQ)

One of the key features of MoQT is the concept of Relays. A Relay is a node within the delivery network for a given MoQ stream and can perform routing and fan-out of streams that utilise it. For contribution streams that span Wide Area Networks, the intention is to use shorter “hops” between relays instead of a single long round-trip connection between the client and server. Shorter hops mean lower latency repair for media objects transported on reliable streams in case any single hop in the network path between client and server experiences packet loss.

Media components delivered over MoQT carry routing information alongside additional information to aid with understanding the type of content being delivered, but the actual media content itself remains opaque and encrypted to prevent tampering or snooping of sensitive content on the network path.

Relays may cache content to perform loss recovery but also to allow other subscribers to access the same content both live and timeshifted. The relay concept may be realised in an Application Server such as the generalised Media AS.

MoQT, when standardised, is a good candidate for specification alongside the HTTP-based contribution and egest protocols already specified in clause 8 of TS 26.512 [42] Release 18 and the RTP-based media transport interface specified in clause 9 of TS 26.113 [48] Release 18.

Potential encoding improvements

Independently of the precise scenario, most of the uplink application may benefit from advanced coding tools, to lower the required bandwidth, increase the QoE and reduce end-to-end latency. The following sub-sections describe a set of improvements that may need to be supported.

Sub-picture latency codecs

For more aggressive scenarios, it is required to achieve sub-picture latency. While this can be done in a non-normative manner by starting to send encoded data before the end of the slice/picture encoding, some ways of doing exist normatively. This includes dependant slicing in HEVC or VVC for instance, but also low-latency image coding such as JPEG-XS.

Classification into two categories:

- Lossy codecs
- Visually lossless (mezzanine)

Gradual Decoder Refresh (GDR)

GDR is widely use in uplink contribution scenarios, to smooth the bit rate compared to classical low-latency structure (e.g. IPPPP…). While it was achieved in a “non-normative” manner in previous standards such as AVC or HEVC, it is normative in VVC. System aspects related to GDR should be supported in uplink streaming profiles and architecture. QoS would be impacted since bit rate would be more predictable using such a technique

Encoder synchronization

The multiple cameras in a venue are synchronized together to enable smooth program production. When the number of cameras goes high in a NPN production setup, the GoP structure used in the cameras’ encoders may overlap, resulting in multiple intra frames to be sent at the same time, which may lead to congestion. While GDR could address this issue at some extent, a mechanism to synchronize the rate-control decisions of multiple encoders running in parallel would avoid temporary network congestion and drops in the QoE.

Variable resolution

In a media production setup involving multiple cameras, only one camera output is used at a time (except in the mosaic case). It means multiple cameras are delivering high throughput at the same time, with only one being really used. To enable throughput optimization in such case, a solution may consist in lowering the resolution of the cameras and to switch the resolution up when the production decides to use a given camera. The recent video codec VVC supports variable resolution switching inside a

given video stream. Mechanisms should be introduced to support such communication and resolution switching.

Multi-layer encoding

Another solution to the problem described in the previous section would consist in leveraging a multi-layer codec with multiple resolution. A selected camera would send low-resolution base-layer all the time, plus the high-resolution enhancement-layer only when the camera is selected by the production. There could mechanism to prioritize the different layers according to production needs.

Summary of coding enhancements

The various codecs enhancements previously defined the network to support the following features:

- Support for sub-frame delivery in media streaming architecture.
- Support for GDR and multi-layer type of transmission in media streaming architecture.
- Support for variable and dynamic QoS switching between multiple profiles, based on layer, resolution, codec-level information of the transported media.

.

Related documentation

[1] 5G-MAG Report: "Uplink media delivery: Architectures & Features"
[2] 3GPP TR 26.805: "Study on Media Production over 5G NPN Systems" (Release 17)
[3] ISO/IEC 13818-1:2023: "Information technology — Generic coding of moving pictures and associated audio information"
[4] Secure Reliable Transport Protocol – Technical Overview, Haivision 2018
[5] IEFT draft-sharabayko-srt-01: The SRT Protocol, 2021
[6] Secure Reliable Transport (SRT) Protocol - GitHub Repository, Haivision 2024
[7] IETF draft-gg-udt-03: UDT UDP-based Data Transfer Protocol, 2010
[8] IETF RFC 768: "User Datagram Protocol"
[9] How Haivision SafeStreams Transport (SST) Transmits Live Video over Cellular and Internet, Vidovation 2023
[10] Reliable Internet Stream Transport "RIST" Activity Group
[11] RIST Forum
[12] LibRIST repository
[13] RIST Tested Products
[14] VSF TR-06-1: RIST Protocol Specification – Simple Profile
[15] VSF TR-06-2: RIST Protocol Specification – Main Profile
[16] VSF TR-06-3: RIST Protocol Specification – Advanced Profile
[17] W3C Recommendation: "WebRTC: Real-Time Communication in Browsers", 06 March 2023
[18] IETF RFC 8825: "Overview: Real-Time Protocols for Browser-Based Applications"
[19] IETF RFC 3264: "An Offer/Answer Model with the Session Description Protocol (SDP)"
[20] IETF RFC 8489: "Session Traversal Utilities for NAT (STUN)"
[21] IETF RFC 8656: "Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN)"
[22] IETF RFC 3711: "The Secure Real-time Transport Protocol (SRTP)"
[23] IETF RFC 3550: "RTP: A Transport Protocol for Real-Time Applications"
[24] 3GPP TS 26.522: "5G Real-time Media Transport Protocol Configurations" (Release 18)
[25] IETF RFC 9000: "QUIC: A UDP-Based Multiplexed and Secure Transport"
[26] IETF RFC 9114: "HTTP/3"
[27] IETF RFC 9221: "An Unreliable Datagram Extension to QUIC"
[28] IETF draft-michel-quic-fec-01: "Forward Erasure Correction for QUIC loss recovery"
[29] IETF draft-pardue-quic-http-mcast-11: "Hypertext Transfer Protocol (HTTP) over multicast QUIC"
[30] IETF draft-jholland-quic-multicast-04: "Multicast Extension for QUIC"
[31] USENIX NSDI Technical Sessions 2018: "Copa: Practical Delay-Based Congestion Control for the Internet"
[32] IETF RFC 8298: "Self-Clocked Rate Adaptation for Multimedia"
[33] IETF RFC 3168: "The Addition of Explicit Congestion Notification (ECN) to IP"
[34] IETF RFC 9331: "The Explicit Congestion Notification (ECN) Protocol for Low Latency, Low Loss, and Scalable Throughput (L4S)"
[35] IETF RFC 6679: "Explicit Congestion Notification (ECN) for RTP over UDP"

[36] IETF draft-ietf-quic-multipath-06: “Multipath Extension for QUIC”
[37] 3GPP TS 23.501 “System architecture for the 5G System” (Release 18)
[38] IETF RFC 9297: “HTTP Datagrams and the Capsule Protocol”
[39] IETF RFC 9220: “Bootstrapping WebSockets with HTTP/3”
[40] 3GPP TS 24.193: “Access Traffic Steering, Switching and Splitting (ATSSS)” (Release 18)
[41] 3GPP TR 26.804: “Study on 5G media streaming extensions” (Release 18)
[42] 3GPP TS 26.512: “5G Media Streaming (5GMS); Protocols” (Release 18)
[43] IETF draft-ietf-avtcore-rtp-over-quic-08: “RTP over QUIC (RoQ)”
[44] IETF draft-ietf-moq-transport-02: “Media over QUIC Transport”
[45] IETF draft-ietf-webtrans-http3-08: “WebTransport over HTTP/3”
[46] 3GPP TS 26.506: “5G Real-time Media Communication Architecture”
[47] IETF RFC 7667: “RTP Topologies”, 2015
[48] 3GPP TS.26113: “Real-Time Media Communication; Protocols and APIs” (Release 18)

www.5g-mag.com

This is a report produced by the 5G-MAG Workgroup CP (Content Production – Standards and Architectures).

Version of the report: v.1.0

Date of publication: 27th January 2025

This 5G-MAG Report can be downloaded from www.5g-mag.com/reports

Feedback from the industry is welcome through https://github.com/5G-MAG/Requests-for-Feedback

Published by 5G-MAG | January 2025

5G Media Action Group (5G-MAG) Association
17A L'Ancienne-Route
1218 Grand-Saconnex (Switzerland)

info@5g-mag.com • www.5g-mag.com