

VIDEO STREAMING PROTOCOLS AND DELIVERY METHODS FOR PHYSICAL SECURITY

MEDIA CODING | SESSION CONTROL | TRANSPORT | DELIVERY | APPLICATION

**A reference guide for
network video systems**



VIDEO STREAMING PROTOCOLS AND DELIVERY METHODS FOR PHYSICAL SECURITY

Network video systems combine media coding, session control, transport, delivery and application services. This reference explains how those functions fit together and connects physical security requirements to candidate mechanisms. It covers live viewing and archive review. Selection is based on the complete media path, required interoperability and measured behavior.

How to use this paper. Start with Section 7 for the requirements-to-mechanisms table. Sections 1–6 explain the technologies; Sections 8–10 define deployment checks and measurement. Section 11 addresses specification status and implementation terms.

Contents

01	Scope and functional categories	3
02	Codec compatibility and developments	4
03	RTSP and WebRTC	6
04	HTTP delivery with HLS and DASH	7
05	SRT, RIST and QUIC	8
06	Comparison of the core technologies	9
07	Requirements mapped to candidate mechanisms	11
08	Latency, reliability and capacity	12
09	Integration and media protection	13
10	Evidence-based evaluation procedure	14
11	Specification status and implementation terms	15
12	Conclusion	15
	Appendix A Glossary	17
	References	18

Acknowledgment This guide was produced by SIA’s Video and Vision Subcommittee, which reviews, develops, promotes and advocates for the adoption of relevant industrywide standards relating to video. SIA thanks SIA Video and Vision Subcommittee Chair Jacob Hengel (chief technology officer and board member at YourSix), and the members of the subcommittee for contributing insights that shaped the development of this guide.

Informational notice. This paper provides technical education. It is not a normative standard, conformance test or product endorsement. No comparative product testing is reported.

01

Scope and functional categories

The core comparison covers RTSP with RTP, WebRTC, HLS including LL-HLS, DASH including low-latency operation, SRT, RIST and QUIC. These are technologies at different functional levels. The tables retain that distinction: QUIC is an underlying transport; RTSP is session control paired here with its RTP media path.

The paper also addresses codecs and selected adjacent technologies where they affect the video path: RTMP/RTMPS, JPEG video, WebCodecs, SFrame and media signing. Device interface specifications, conformance profiles and connection-establishment methods, such as tunneling and device-initiated connectivity, are referenced but not described; their definitions belong to the organizations that publish them. It is not an exhaustive catalogue of media formats or device APIs. No market-share, adoption or future-dominance claims are made.

Functions in a complete media path

Table 1 describes functional categories, rather than strict OSI layers. A technology may span several categories. Successful integration requires compatible interfaces at each handoff.

Table 1: Functions in an IP video system

Function	Examples	Compatibility question
Media coding	H.264, H.265, VP8, VP9, AV1, AV2	Can the decoder accept the stream’s coding tools, profile, level or tier limits, bit depth and chroma format? [36, 40, 52]
Media packaging	RTP payload formats, MPEG transport streams, fragmented MP4, CMAF	Can each receiver interpret the media framing and timing? [3, 18]
Session and service control	RTSP; ONVIF services; WHIP; application signaling	How are sources, sessions, recordings and requested operations selected? [1, 32]
Media delivery	WebRTC; HLS; DASH; SRT; RIST	Which timing, adaptation, recovery and distribution mechanisms are available? See Sections 3–5.
Network transport	UDP; TCP; QUIC	Are datagrams or ordered streams required, and which network paths are permitted? [10, 27, 29]
Communication protection	TLS; DTLS; SRTP; SFrame	Which endpoints hold keys, and which media or metadata is protected? [4, 9, 28, 46]

References are attached to specific technical claims. The use-case mapping and evaluation procedure are editorial synthesis intended to organize assessment; they are not new protocol requirements.

02

Codec compatibility and developments

H.264 and H.265 compatibility

H.264/AVC and H.265/HEVC are coding formats with separate RTP payload specifications. Compatibility requires support for the stream's coding features and operating limits. Profile and level labels need not be identical at both ends: the decoder must support the transmitted profile and sufficient level capacity, together with applicable tier, bit-depth, chroma and packetization constraints. A decoder's support for one profile does not establish support for every other profile. [40, 52]

WebRTC endpoint requirements

RFC 7742 requires both VP8 and H.264 Constrained Baseline for WebRTC browsers and for WebRTC non-browsers that transmit or receive video. Its separately defined category, "WebRTC-compatible endpoint," can interoperate without meeting every endpoint requirement and is not subject to that mandatory codec set. A camera or native application is not exempt merely because it lacks a browser interface. Procurement should distinguish a conforming endpoint claim from compatibility with a particular service. [13]

Additional codecs are negotiated by the participating endpoints. Support for an encoded file does not establish support for its WebRTC RTP path. Receiving and sending capabilities can differ; both the negotiated codec parameters and the required direction of operation must be checked. [6, 40, 53]

H.265 in browser WebRTC

H.265 is a relevant WebRTC option where the endpoints support it. Chrome 136 release notes document its addition to WebRTC in 2025; WebKit's Safari 18.0 notes document HEVC RTP packetization support in 2024. Chromium's implementation documentation ties support to platform and hardware capabilities and does not provide a software encoding fallback. Chromium's API owners stated they are open to disabling H.265 WebRTC support for a percentage of users, so capable hardware does not guarantee the codec will be offered; the codec is used only if both endpoints negotiate it. These are implementation developments, not a universal browser guarantee. [41, 42, 53]

RFC 7798 supplies the HEVC RTP payload format. The more specific H.265 Profile for WebRTC document is still Internet-Draft revision 09, dated 20 July 2026. In a surveillance path, forwarding H.265 without transcoding is a candidate only when the source, intermediary and receiving implementation agree on the required profile, level/tier, packetization and other negotiated constraints. [40, 43]

VP9

The WebM Project dates VP9 availability to 17 June 2013. RFC 9628, published on the IETF Standards Track in March 2025, specifies its RTP payload format, profile-id signaling and provisions for temporal and spatial scalability. The later RTP publication date is distinct from the codec's initial availability. Endpoint checks include the selected profile and any scalable configuration. [38, 39]

AV1

The Alliance for Open Media (AOMedia) publishes AV1 bitstream and decoding requirements and a separate RTP payload specification. RTP payload version 1.0.0 is identified as an AOM Final Deliverable dated Dec. 15, 2024. It defines packetization and Session Description Protocol (SDP) parameters including profile, level and tier, with offer/answer negotiation and declarative descriptions applicable to RTSP. [35, 36]

VIDEO STREAMING PROTOCOLS AND DELIVERY METHODS FOR PHYSICAL SECURITY

AV2

AOMedia lists AV2 specification and AOMedia Video Model reference software version 1.0.0 as released on May 28, 2026, superseding the January 2026 working draft. It describes support for screen content, AR/VR and split-screen delivery. A released coding specification establishes the format; implementation evaluation separately establishes encoder, transport, recording and decoder support. AV1 decoding capability does not imply AV2 capability. [37]

Codec evaluation

A reproducible comparison uses defined source content, encoder implementation and preset, resolution, frame rate, bitrate policy, random-access interval, delay constraint and quality criterion. Surveillance workloads can include motion, low-light noise and task-relevant detail. Report encoding delay, decoded quality, CPU/GPU use, energy and concurrent stream capacity together.

A measured coding-efficiency result applies to the tested content, operating points and implementations. This paper assigns no universal bandwidth or storage saving to H.265, VP9, AV1 or AV2. Procurement evidence should identify the actual workload and required image detail.

Browser decoding interfaces

Browser media support has several surfaces: a platform media player, Media Source Extensions (MSE), WebRTC and WebCodecs. MSE accepts supported media segments for playback. WebCodecs exposes encoding and decoding interfaces and mandates no particular codec. A WebCodecs application must supply the surrounding transport, media parsing, timing and rendering logic. WebCodecs is cited as the latest published W3C Working Draft. [34, 47]

For each required client, record the browser and operating-system versions, available codec configuration, hardware acceleration and sustained multistream behavior. Test the intended playback API and direction, rather than inferring support from a different API or from the codec name alone.

03

RTSP and WebRTC

RTSP control and RTP media

Real Time Streaming Protocol (RTSP) controls media sessions through operations such as DESCRIBE, SETUP, PLAY, PAUSE and TEARDOWN. The media commonly uses the Real-Time Transport Protocol (RTP), with RTCP reports supporting synchronization and feedback. RTP sequence numbers and timestamps support media processing; delivery and capacity guarantees require additional mechanisms. [1, 3]

RTSP can negotiate RTP over UDP, including multicast where supported, or media interleaved over TCP. RTSP 2.0 supersedes the RTSP 1.0 specification but is not generally wire compatible with it. ONVIF Streaming Specification 26.06 explicitly uses RTSP 1.0. Identify the version, extensions and selected media path. [2, 5]

Protection can include RTSP over TLS, SRTP for separately transported RTP, or a protected tunnel. TLS on the control connection does not protect an independent, unencrypted UDP media path. Authentication, certificate validation, media protection and termination points are separate checks. [2, 4]

WebRTC media and connectivity

WebRTC combines W3C browser APIs with IETF protocols. Audio and video use RTP with mandatory SRTP protection and DTLS-SRTP key establishment. The suite includes feedback and congestion-control requirements. Data channels use SCTP over DTLS; the application defines the meaning of data such as events and analytics metadata. [6, 8, 9, 12]

Interactive Connectivity Establishment (ICE) tests candidate paths. STUN assists connectivity discovery; TURN supplies a relay path. RFC 8835 Section 3.4 requires support for TURN over TCP and TURN over TLS over TCP between the WebRTC endpoint and TURN server for networks that block UDP. The same section also requires ICE-TCP candidate support, which can reach peers with public addresses across UDP-blocking firewalls without a TURN relay. These modes describe that leg, rather than requiring TCP on every relay-to-peer leg. A connection may be direct or relayed, subject to network policy and reachable infrastructure. [10, 11]

Signaling and recorded playback

The general WebRTC architecture leaves application signaling to the service. RFC 9725 defines WHIP for HTTP-based WebRTC ingestion. WHEP addresses viewer egress, but revision 04, dated 22 June 2026, remains an Internet-Draft. [7, 32, 45]

WebRTC can carry recorded media when the application supplies recording selection, seek operations and timestamp mapping. Neither WHIP nor a WHEP session supplies a complete surveillance recording catalogue or archive service. Codec requirements and optional H.265, VP9 and AV1 support are discussed in Section 2.

04

HTTP delivery with HLS and DASH

HLS and low-latency HLS

HTTP Live Streaming (HLS) uses playlists identifying media segments and available renditions. Clients retrieve those resources and can select among renditions prepared by the service. HLS supports live and on-demand presentations, including recorded-video playback. RFC 8216 is an Informational RFC in the Independent Submission stream. [14]

Low-latency HLS (LL-HLS) adds partial segments, blocking playlist reload, preload hints and related mechanisms. The encoder, packager, origin, distribution infrastructure and player participate in making newly produced media available earlier. Ordinary HLS support does not establish support for these extensions. [15, 16]

The cited second-edition HLS draft is revision 22, dated 1 May 2026. Its Appendix B.1 low-latency profile specifies HTTP/2 or HTTP/3 delivery. Playback distance from the live edge depends in part on partial-segment duration. This constrains a configuration without setting a universal glass-to-glass delay. Its Independent Submission status and lack of IETF consensus are stated in Section 11. The profile also directs clients to verify server compliance before playing closer than two Target Durations from live, and to fall back to regular-latency playback otherwise. [16]

DASH and low-latency DASH

Dynamic Adaptive Streaming over HTTP (DASH) describes presentations through a Media Presentation Description (MPD) and segment formats. ISO/IEC 23009-1:2026 is the sixth edition, published in July 2026. Both ordinary and low-latency delivery belong in a system assessment; LL-DASH is not a separate underlying transport protocol. [17, 18]

A low-latency workflow can deliver and process Common Media Application Format (CMAF) chunks before the complete segment is available. Availability signaling, clock alignment, player buffering and catch-up behavior affect delay. DASH `availabilityTimeOffset` and `availabilityTimeComplete` describe availability conditions, rather than guaranteed presentation timing. [18, 19]

Distribution and client behavior

HLS and DASH can use HTTP origins and content delivery networks (CDNs). For low-latency operation, intermediaries need to preserve early media availability; buffering a response until completion can remove that benefit. Rendition alignment, cache behavior, request authorization and client buffering belong in the system design. [15, 18, 30]

HTTP/3 uses QUIC over UDP, so HTTP delivery is not confined to TCP. Negotiated HTTP version and fallback behavior are observable properties of a session. HTTPS protects a connection to its termination point; media-object encryption can protect a different boundary. [14, 20, 28]

Browser playback uses a compatible platform player or application player, for example through MSE. Check the actual codec, container, API and low-latency feature support on each required client. Archive usability also requires time indexing, seek behavior and recording access controls. [15, 19, 34]

05

SRT, RIST and QUIC

Secure Reliable Transport

Secure Reliable Transport (SRT) provides UDP-based delivery with Automatic Repeat reQuest (ARQ) recovery. The sender and receiver separately agree on payload format and codec. An SRT link can carry media from an encoder to a processing or recording service, followed by a different viewer-delivery path. [21, 23]

Live-mode operation uses configurable recovery buffering and late-packet handling. More recovery time permits additional retransmission opportunities while increasing delay. Assessment measures residual loss and recording completeness alongside camera-to-display delay. [23]

Encryption requires configuration. SRT_SECURE has an empty default in the reference library, disabling encryption. SRT_ENFORCEDENCRYPTION checks agreement between endpoint settings and can accept two empty passphrases. Cryptographic mode and build options also matter. [22]

Reliable Internet Stream Transport

Reliable Internet Stream Transport (RIST) is specified by the Video Services Forum TR-06 series. Simple Profile provides RTP/RTCP-based transport and retransmission over UDP. Recovery needs buffer time, available bandwidth and a return path for feedback. [24]

Main Profile adds tunneling and two security modes: DTLS 1.2 and pre-shared key (PSK) AES-CTR encryption. Section 7.3 states that the PSK encryption algorithm does not provide origin authentication and is therefore susceptible to replay and data injection, with the risk mitigated by GRE and RIST handling of duplicate and out-of-order packets. Section 7.5 and Annex D define an optional EAP SHA256-SRP6a protocol that adds mutual client and server authentication at session establishment and can derive the PSK passphrase. Advanced Profile has additional capabilities and security provisions. Record the profile, revision, encryption mode and whether EAP-SRP authentication is enabled. [25, 26]

Section 6.2 lists TLS_RSA_WITH_NULL_SHA256, a suite that authenticates without encrypting, among the mandatory DTLS cipher suites, and requires that users be able to disable individual suites. A DTLS-mode RIST link therefore needs its negotiated cipher suite verified. [25]

QUIC and media protocols above it

QUIC is an encrypted transport over UDP. Its reliable streams deliver ordered bytes within each stream; loss on one stream need not block unrelated streams. Shared congestion control and application dependencies can still affect them. The optional QUIC DATAGRAM extension carries datagrams without QUIC retransmission. [27, 29]

QUIC uses TLS 1.3 for its handshake and key establishment. HTTP/3 maps HTTP onto QUIC, including HLS or DASH delivered over HTTP/3. A media application still supplies coding, timing, control and playback behavior above the transport. [20, 27, 28]

Media over QUIC Transport (MOQT) defines publish/subscribe behavior over QUIC or WebTransport. The reviewed revision 21 is dated Sept. 8, 2026, and remains an Internet-Draft, separate from the published QUIC RFCs. Interoperability claims for this work need an identified draft version. [33]

06

Comparison of the core technologies

Tables 2 and 3 compare the same seven technology groups while retaining their different roles. HLS and DASH rows include ordinary delivery and their low-latency modes. Security capabilities require compatible implementations and configuration.

Table 2: Roles, transport protection and client access

Technology and role	Transport and protection	Client access and integration
RTSP + RTP Session control + media	UDP media or TCP interleaving; TLS, SRTP or tunnels can protect applicable legs. [1–4]	RTSP client or gateway/application integration. ONVIF specifies HTTP and WebSocket carriage; client media handling is still required. [5]
WebRTC Protocol suite + API	DTLS-SRTP media protection; ICE with direct or relay paths, including TURN over TCP and TURN over TLS over TCP. [8–11]	Browser API or native implementation, with agreed signaling, codecs and authorization. [6, 7, 13]
HLS/LL-HLS HTTP delivery	HTTPS connection and media-object encryption are distinct. The cited LL profile specifies HTTP/2 or HTTP/3. [14, 16]	Compatible native or application player. Check ordinary HLS, LL extensions and decoder support separately. [15, 34]
DASH/LL-DASH HTTP delivery	HTTP delivery, including HTTP/3 where supported. Connection and content protection require explicit configuration. [17, 20]	DASH player and decoder. Low-latency mode also needs aligned availability signaling, content and player behavior. [18, 19]
SRT Contribution transport	UDP, retransmission and configured recovery timing; optional payload encryption. [21–23]	Compatible endpoints or a gateway into a viewing format; payload and codec agreement required.
RIST Contribution transport	UDP-based recovery. Security depends on profile, mode, cipher suite and whether EAP-SRP authentication is enabled. [24–26]	Compatible profile and capabilities at endpoints; browser viewing needs a suitable integration or gateway.
QUIC Underlying transport	Encrypted UDP transport; reliable streams and optional unreliable datagrams. [27–29]	Higher-layer interfaces such as HTTP/3 enable browser use; a media mapping and player are still needed. [20]

The table describes communication paths. Device provisioning, user authorization, stored-recording protection and implementation security require their own controls and validation.

VIDEO STREAMING PROTOCOLS AND DELIVERY METHODS FOR PHYSICAL SECURITY

Media behavior and performance variables

Table 3: Media capabilities and operational dependencies

Technology	Media and control capabilities	Variables to measure or verify
RTSP + RTP	Session operations, RTP timing, RTCP feedback and multicast where supported. [1–3]	Encoder delay, chosen transport, recovery, client buffering and server replay extensions.
WebRTC	Real-time audio/video, congestion feedback and optional data channels. [8, 12]	Direct/relay route, jitter buffer, codec, repair policy and server topology; archive services if required.
HLS/LL-HLS	Live or on-demand playlists and rendition selection; LL mode adds partial segments and related mechanisms. [14–16]	Startup and archive seek, segment/part duration, live-edge distance, origin/CDN behavior and buffering.
DASH/LL-DASH	MPD-based live or on-demand presentation and adaptive representation selection; incremental delivery in suitable LL workflows. [17–19]	Startup and archive seek, chunk availability, clock alignment, player target and response buffering.
SRT	Payload-agnostic ARQ transport; FEC and bonding support depends on implementation and configuration. [21, 23]	Round-trip time, loss bursts, recovery buffer, spare bandwidth and late-packet policy.
RIST	RTP recovery and profile-dependent tunneling, multiplexing and path features. [24–26]	Profile agreement, return-path reachability, retransmission budget and available capacity.
QUIC	Multiplexed reliable streams and optional unreliable datagrams; application protocols organize media. [27, 29]	Media mapping, stream dependencies, congestion control, buffering, network policy and fallback.

Adaptive bitrate selection, congestion control and transcoding are separate operations. A sender can react to congestion without creating an HTTP rendition ladder. Repackaging can preserve encoded media when the destination accepts it; changing the coding format requires a supported conversion path. Record processing at every handoff.

Multichannel transport, multiple viewers and IP multicast also describe different properties. Multiple media flows do not establish multicast distribution. Scale testing should specify sources, renditions, viewers and concurrent decodes, rather than assign a fixed resource rating to a protocol name.

07

Requirements mapped to candidate mechanisms

Table 4 is a starting point for evaluation, not a ranking. The candidates follow from the functions described in Sections 2–6 and 9. Several may appear in one deployment; the final choice depends on measured requirements and supported interfaces.

Table 4: Physical security workflows and evaluation candidates

Workflow and requirement	Candidate mechanisms	Evidence needed before selection
Latency sensitive tasks	WebRTC for browser interaction; RTSP/RTP for capable native clients or a gateway. HLS/DASH, including LL modes, for compatible viewing workflows.	Test live delay and startup separately. Verify command/audio path, codec, network traversal, relay use and reconnect behavior.
Archive review	HLS or DASH on-demand delivery; RTSP replay with recording services; WebRTC with application-defined archive control.	Test requested versus actual timestamps, seek precision, discontinuities, playback rates, audio sync, permissions and export preservation.

Illustrative combined path. A site gateway can receive camera RTP under RTSP control, send a contribution feed to a recording service, and expose live WebRTC and recorded HLS/DASH to viewers. This is a design example derived from the functions above. It is not evidence of adoption or a recommendation for a particular product.

For a candidate path, first establish endpoint and codec compatibility. Then measure timing, continuity, security boundaries and capacity under the intended workload. A mechanism that satisfies one row's requirements may still need separate services for another row.

08

Latency, reliability and capacity

Measurement boundaries

Live viewing delay runs from a physical scene event to its presentation on the display. Capture, encoding, packaging, transit, recovery, buffering, decoding and display all contribute. Network round-trip time or a protocol buffer setting measures only part of that path. [30]

Table 5: Timing measurements for a surveillance system

Measurement	Start and end points	Reporting conditions
Live viewing delay	Scene event to corresponding displayed event.	Repeated observations, distribution, visual/clock reference and uncertainty.
Startup time	Viewer request to first usable decoded frame.	Cold/warm session, authentication, connection setup and random-access frame conditions.
Archive seek response	Seek request to first frame meeting the requested selection.	Requested/actual timestamps, precision, buffering and playback rate.
Interactive response	Command or talk request to observed effect; separately, return feedback.	Separate command-path delay from movement, processing and returned video/audio.
Recovery time	Defined failure onset or restoration point to usable media.	Trigger, outage length, loss pattern, reconnect policy and missing interval.

Recovery and recording continuity

Retransmission is useful when missing data remains available and arrives before its deadline. Forward error correction (FEC) sends redundancy, whose recovery capability depends on the scheme and loss pattern. Lower playback delay leaves less time for variation and repair. Test freezes, visual damage and missing recorded intervals as well as packet loss. [8, 23, 24]

TCP and reliable QUIC streams provide ordered byte delivery within their stream boundaries, but presentation still has deadlines. Applications using UDP may implement recovery above it. Recording completeness and later backfill are separate from whether a live picture remains viewable. [23, 27]

Capacity at each processing node

A capacity report identifies source, rendition and viewer counts; simultaneous decodes; resolution, frame rate and bitrate; buffering; encryption; and whether each node forwards, relays or transcodes. Record CPU/GPU utilization, memory, network traffic and relevant energy measurements with hardware and software versions. An origin, TURN relay, media server and viewing device perform different work. [11, 30, 31]

09

Integration and media protection

ONVIF specifications

ONVIF publishes device and client interface specifications and conformance profiles, including profiles that address streaming, recording and metadata. This paper does not describe or interpret those profiles; readers should consult ONVIF for profile scope, requirements and conformance status. References to the ONVIF Streaming Specification in this paper concern only the RTSP and RTP behavior it specifies. [5, 57]

Methods that let a device initiate an outbound connection to a remote service, such as ONVIF Uplink and vendor cloud-connect services, change how a connection is established rather than how video is coded or delivered and are outside the scope of this paper. [44]

Additional ingest and image paths

RTSP and RTP can also be carried over HTTP, HTTPS or WebSocket where both endpoints support it; this is distinct from HLS or DASH segment delivery. [5]

RTMP is another possible source-to-service ingest path. RTMPS is RTMP carried over TLS. [54] Endpoint codec support and message-format extensions must agree: Veovera's Enhanced RTMP specification adds mappings for newer codecs, so support for legacy RTMP alone does not establish support for those extensions. [50]

Motion JPEG describes a sequence of independently JPEG-coded pictures, rather than one complete streaming protocol. RFC 2435 specifies JPEG over RTP. An HTTP multipart JPEG feed is a different encapsulation; its timing, authentication and client behavior must be checked as implemented. JPEG video is included here for integration context rather than as a performance-ranked alternative. [51, 55]

Encryption through media intermediaries

A TURN relay can forward protected WebRTC traffic without terminating SRTP. A media gateway or processing server may instead terminate one protected session and establish another. In that case, each leg is encrypted but the intermediary can have access to plaintext. [9, 11, 31]

SFrame, defined in RFC 9605, adds media-payload encryption that can preserve confidentiality through a selective forwarding unit (SFU) while exposing information needed for forwarding. The application must provide key management and required replay protection. This differs from enabling SRTP on each connection. A recorder or analytics processor that needs decoded content must be included appropriately in the authorized key and processing design. [46]

For every path, document which entities can decrypt media, which metadata remains visible, how recipients receive keys, and how access is revoked. For HTTP delivery, connection TLS and media-object encryption may terminate at different points. [14, 46]

Signed video and recording integrity

ONVIF Media Signing specifies how H.264 and H.265 video can be signed within the bitstream, with verification and export handling defined in ONVIF specifications; the associated Media Signing Add-on is a release candidate. [48, 49, 56] Signed-content verification, media confidentiality and an organization's evidence-handling process are separate requirements. For an export workflow, test whether signing information survives storage and export and whether a verifier can validate the exported recording. This paper makes no legal-admissibility determination.

10

Evidence-based evaluation procedure

This proposed procedure produces comparable engineering evidence. It is not a protocol conformance test or an SIA certification requirement. Acceptance limits come from the intended surveillance task and are declared before testing.

Define the task. Separate live observation, PTZ interaction, bidirectional audio, archive review, recording ingestion and multi-viewer distribution. Record timing, media quality, metadata, continuity and access-control requirements.

Identify the implementation. Record devices, servers, firmware, browser/OS/player versions, protocol revision or profile, codecs and enabled options. Capture the negotiated transport, direct or relay route, security mode and any repackaging or transcoding.

Control the media workload. Use the same source sequence where possible. Record scene motion, resolution, frame rate, bitrate policy, random-access interval, audio and concurrent streams. When adaptation changes quality, report the change with the timing result.

Exercise defined network conditions. Include a baseline and separately identified bandwidth limits, delay, variation, random and burst loss, reordering, interruptions and blocked UDP. State impairment location, direction, duration and method; preserve a repeatable configuration.

Measure outcomes together. Collect Table 5 timing measures, visual/audio defects, delivered quality, connection success, recovery, recording gaps and resource use. Report sample counts, duration, median and tail percentiles, plus measurement uncertainty or confidence intervals where justified.

Verify interoperability and protection. Check each required endpoint pair and optional feature. Observe encryption on each leg, certificate or key validation, authorization, revocation and reconnect behavior. Test required archive, metadata, signing and export functions. Report the specific configuration tested.

Assessment record

Retain a concise record of the use case, acceptance limits, endpoint versions, negotiated parameters, network conditions, media workload, observations and unresolved limitations. Attach configuration evidence and measurement results so a later firmware, browser or service change can be evaluated against the same requirements.

A result supports the tested path and workload. Broader claims require additional coverage of the relevant endpoint combinations, operating conditions and failure modes.

11

Specification status and implementation terms

A draft may be revised, replaced, expire or become an RFC after this review.

Standards and published specifications

RTSP 2.0, RTP, QUIC, WHIP and the cited WebRTC transport requirements are IETF specifications. The W3C WebRTC browser API is a Recommendation dated 13 March 2025. DASH is standardized by ISO/IEC; the sixth edition of ISO/IEC 23009-1 is published as 2026-07 in the ISO catalogue. [2, 3, 6, 10, 17, 27, 32]

HLS RFC 8216 is Informational and belongs to the Independent Submission stream. Its second-edition successor, draft-pantos-hls-rfc8216bis-22, also follows Independent Submission with intended Informational status. It is not endorsed by the IETF, has no formal standing in the IETF standards process as an individual draft, and explicitly does not represent IETF consensus. Hosting on Datatracker is not an IETF consensus endorsement. [14, 16] The IESG conflict review for draft-pantos-hls-rfc8216bis-22 concluded in April 2026 that the work does not prevent publication, and Apple has posted a later preliminary revision. The cited revision may therefore be superseded or published as an RFC. [16]

RIST is specified by VSF Technical Recommendations. SRT has public project documentation and an expired individual Internet-Draft; that draft is not an IETF standard. AOMedia publishes AV1 and AV2 specifications, including the AV1 RTP final deliverable. [21, 23–26, 36, 37] ONVIF profiles and service specifications have their own conformance framework. [57]

Work in progress

MOQT revision 21, WHEP revision 04 and H.265 Profile for WebRTC revision 09 are Internet-Drafts. The cited MOQT draft expires on 12 March 2027; its expiry is a document-management date, not an implementation-support deadline. WebCodecs is cited as the latest published W3C Working Draft. Media Source Extensions 2 is also a W3C Working Draft. The ONVIF Media Signing Add-on is a release candidate. [33, 34, 43, 45, 47, 49]

Implementation licensing

Specification status and implementation licensing are different matters. For example, the SRT reference library uses MPL 2.0, which does not establish the license of every SRT implementation. A technical label does not resolve applicable software licenses, codec patent terms, commercial contracts or support obligations. [21]

12

Conclusion

Physical security video can combine several complementary protocols and delivery methods. A useful selection process starts with the workflow, identifies compatible coding and interfaces, defines connectivity and protection boundaries, and measures the complete path. The requirements table and evaluation procedure provide a common basis for that assessment without ranking products or assigning universal performance claims.

APPENDIX & REFERENCES



VIDEO STREAMING PROTOCOLS AND DELIVERY METHODS FOR PHYSICAL SECURITY

Appendix A Glossary

Table 6: Selected terms and abbreviations

Term	Meaning in this paper
ARQ/FEC	Automatic Repeat reQuest uses retransmission; forward error correction uses added repair information.
AVC/HEVC	Advanced Video Coding (H.264) / High Efficiency Video Coding (H.265).
AV1/AV2	AOMedia video coding specifications, separate from their transport and playback integrations.
CDN	Content delivery network: distributed infrastructure for serving content.
CMAF	Common Media Application Format: media organization used in compatible segmented delivery workflows.
Codec/profile/level	Coding format; supported coding tools or features; operating limits. Compatibility requires the stream to fit decoder capabilities.
Contribution/ingest	Transport of source media into a processing, distribution or recording service.
DASH/MPD	Dynamic Adaptive Streaming over HTTP/Media Presentation Description.
DTLS/TLS	Datagram Transport Layer Security/Transport Layer Security.
HLS/LL-HLS	HTTP Live Streaming/its low-latency extensions.
ICE	Interactive Connectivity Establishment: candidate-path gathering and connectivity checks.
MSE/WebCodecs	Media Source Extensions accepts media segments; WebCodecs exposes codec interfaces.
NAT	Network address translation.
ONVIF profile	A conformance profile published by ONVIF; not a codec profile.
PTZ	Pan, tilt and zoom control.
QUIC/MOQT	Encrypted transport over UDP/Media over QUIC Transport, a higher-layer media protocol under development.
Rendition	An available encoding or presentation variant, for example at a different bitrate or resolution.
RIST/SRT	Reliable Internet Stream Transport/Secure Reliable Transport.
RTMP/RTMPS	Real-Time Messaging Protocol/RTMP carried over TLS.
RTP/RTCP/SRTP	Real-Time Transport Protocol/RTP Control Protocol/Secure RTP.
RTSP	Real-Time Streaming Protocol: session control for media delivery.
SDP	Session Description Protocol: descriptions of media and relevant session parameters.
SFU/SFrame	Selective forwarding unit/media-payload protection that can preserve confidentiality through an intermediary.
STUN/TURN	Session Traversal Utilities for NAT/Traversal Using Relays around NAT.
Transcoding	Decoding and re-encoding media; distinct from changing only its packaging.
WHIP/WHEP	WebRTC-HTTP Ingestion Protocol/WebRTC-HTTP Egress Protocol.

References

Citations identify protocol requirements, industry specifications or implementation documentation. The ISO catalogue supports edition, date and scope; detailed DASH behavior is supported by public DASH-IF guidance. Current-version URLs can change after review.

- [1] IETF. RFC 2326. Real Time Streaming Protocol (RTSP). April 1998. RTSP 1.0; obsoleted by RFC 7826.
<https://www.rfc-editor.org/rfc/rfc2326.html>
- [2] IETF. RFC 7826. Real-Time Streaming Protocol Version 2.0. December 2016. Sections 1, 4, 14, 19 and 21; Appendix C.
<https://www.rfc-editor.org/rfc/rfc7826.html>
- [3] IETF. RFC 3550. RTP: A Transport Protocol for Real-Time Applications. July 2003. Sections 1, 5 and 6.
<https://www.rfc-editor.org/rfc/rfc3550.html>
- [4] IETF. RFC 3711. The Secure Real-time Transport Protocol (SRTP). March 2004. Sections 1, 3 and 9.
<https://www.rfc-editor.org/rfc/rfc3711.html>
- [5] ONVIF. Streaming Specification. Version 26.06, June 2026. Sections 5–8. Current-version URL.
<https://www.onvif.org/specs/stream/ONVIF-Streaming-Spec.pdf>
- [6] W3C. WebRTC: Real-Time Communication in Browsers. Recommendation, March 13, 2025.
<https://www.w3.org/TR/2025/REC-webrtc-20250313/>
- [7] IETF. RFC 8825. Overview: Real-Time Protocols for Browser-Based Applications. January 2021. Sections 2 and 4.
<https://www.rfc-editor.org/rfc/rfc8825.html>
- [8] IETF. RFC 8834. Media Transport and Use of RTP in WebRTC. January 2021. Sections 4, 6, 7 and 9.
<https://www.rfc-editor.org/rfc/rfc8834.html>
- [9] IETF. RFC 8827. WebRTC Security Architecture. January 2021. Sections 4 and 6.
<https://www.rfc-editor.org/rfc/rfc8827.html>
- [10] IETF. RFC 8835. Transports for WebRTC. January 2021. Section 3.4: ICE, TURN, TURN over TCP, and TURN over TLS over TCP; Section 3.5: transport protocols.
<https://www.rfc-editor.org/rfc/rfc8835.html>
- [11] IETF. RFC 8656. Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN). February 2020. Sections 2 and 3.
<https://www.rfc-editor.org/rfc/rfc8656.html>
- [12] IETF. RFC 8831. WebRTC Data Channels. January 2021. Sections 4 and 6.
<https://www.rfc-editor.org/rfc/rfc8831.html>
- [13] IETF. RFC 7742. WebRTC Video Processing and Codec Requirements. March 2016. Sections 2, 5 and 6.2.
<https://www.rfc-editor.org/rfc/rfc7742.html>
- [14] RFC Editor. RFC 8216. HTTP Live Streaming. August 2017. Informational RFC. Sections 3, 4 and 6.
<https://www.rfc-editor.org/rfc/rfc8216.html>
- [15] Apple. Enabling Low-Latency HTTP Live Streaming (HLS). Developer documentation.
<https://developer.apple.com/documentation/http-live-streaming/enabling-low-latency-http-live-streaming-hls>

VIDEO STREAMING PROTOCOLS AND DELIVERY METHODS FOR PHYSICAL SECURITY

- [16] Pantos, R., editor. HTTP Live Streaming 2nd Edition. draft-pantos-hls-rfc8216bis-22, 1 May 2026. Independent Submission; intended Informational. Sections 4 and 6; Appendix B.1. Not an IETF consensus standard.
<https://datatracker.ietf.org/doc/html/draft-pantos-hls-rfc8216bis-22>
- [17] ISO/IEC. ISO/IEC 23009-1:2026. Information technology: Dynamic adaptive streaming over HTTP (DASH): Part 1: Media presentation description and segment formats. Edition 6, July 2026. Catalogue record used for edition, date and scope.
<https://www.iso.org/standard/89027.html>
- [18] DASH Industry Forum. Low Latency Modes for DASH. CR-Low-Latency-Live-r8, 2020. Prepublished interoperability guidance linked from IOP version 5 Part 4; not an ISO standard.
<https://dashif.org/docs/CR-Low-Latency-Live-r8.pdf>
- [19] DASH Industry Forum. dash.js Low Latency Streaming. Implementation documentation on CMAF chunks, availability signaling and playback configuration.
<https://dashif.org/dash.js/pages/usage/low-latency.html>
- [20] IETF. RFC 9114. HTTP/3. June 2022. Sections 2, 3, 4 and 6.
<https://www.rfc-editor.org/rfc/rfc9114.html>
- [21] SRT project. Secure Reliable Transport (SRT). Official repository, hosted by Haivision. Project features and MPL 2.0 reference-library license.
<https://github.com/Haivision/srt>
- [22] SRT project. SRT API Socket Options. SRTO_PASSPHRASE, SRTO_ENFORCEDENCRYPTION, SRTO_CRYPTOMODE and latency options. Current documentation.
<https://github.com/Haivision/srt/blob/master/docs/API/API-socket-options.md>
- [23] Sharabayko et al. The SRT Protocol. draft-sharabayko-srt-01, September 2021. Expired individual Internet-Draft, not an IETF standard. Sections 4.4 through 4.6.
<https://datatracker.ietf.org/doc/html/draft-sharabayko-srt-01>
- [24] Video Services Forum. VSF TR-06-1:2020. Reliable Internet Stream Transport (RIST) Simple Profile. 25 June 2020.
https://static.vsf.tv/download/technical_recommendations/VSF_TR-06-1_2020_06_25.pdf
- [25] Video Services Forum. VSF TR-06-2:2024. Reliable Internet Stream Transport (RIST) Main Profile. 12 June 2024. Sections 5 to 7, including the Section 7.3 note on origin authentication, Section 7.5 and Annex D (EAP SHA256-SRP6a).
https://static.vsf.tv/download/technical_recommendations/VSF_TR-06-2_2024_06_12.pdf
- [26] Video Services Forum. VSF TR-06-3:2024. RIST Protocol Specification: Advanced Profile. 12 June 2024. Sections 7 and 8.
https://static.vsf.tv/download/technical_recommendations/VSF_TR-06-3_2024_06_12.pdf
- [27] IETF. RFC 9000. QUIC: A UDP-Based Multiplexed and Secure Transport. May 2021. Sections 2, 9 and 13.
<https://www.rfc-editor.org/rfc/rfc9000.html>
- [28] IETF. RFC 9001. Using TLS to Secure QUIC. May 2021. Sections 4 and 5.
<https://www.rfc-editor.org/rfc/rfc9001.html>
- [29] IETF. RFC 9221. An Unreliable Datagram Extension to QUIC. March 2022. Sections 3 and 5.
<https://www.rfc-editor.org/rfc/rfc9221.html>

VIDEO STREAMING PROTOCOLS AND DELIVERY METHODS FOR PHYSICAL SECURITY

- [30] IETF. RFC 9317. Operational Considerations for Streaming Media. October 2022. Informational RFC. Sections 4 and 5.
<https://www.rfc-editor.org/rfc/rfc9317.html>
- [31] IETF. RFC 7667. RTP Topologies. November 2015. Informational RFC. Media intermediaries and security considerations.
<https://www.rfc-editor.org/rfc/rfc7667.html>
- [32] IETF. RFC 9725. WebRTC-HTTP Ingestion Protocol (WHIP). March 2025. Sections 3 and 4.
<https://www.rfc-editor.org/rfc/rfc9725.html>
- [33] IETF MoQ Working Group. Media over QUIC Transport. draft-ietf-moq-transport-21, 8 September 2026. Internet-Draft; intended Standards Track. Expires 12 March 2027.
<https://datatracker.ietf.org/doc/html/draft-ietf-moq-transport-21>
- [34] W3C. Media Source Extensions. Media Source Extensions 2 is a W3C Working Draft. Used for the application-fed browser playback model; draft additions are not treated as universal implementation support.
<https://www.w3.org/TR/media-source-2/>
- [35] Alliance for Open Media. AV1 Video Codec. Specification overview and links to the AV1 bitstream, decoding and implementation resources.
<https://aomedia.org/specifications/av1/>
- [36] Alliance for Open Media. RTP Payload Format For AV1. Version 1.0.0, Final Deliverable, 15 December 2024. Sections 4, 5 and 7.
<https://aomediacodec.github.io/av1-rtp-spec/v1.0.0.html>
- [37] Alliance for Open Media. AV2 Specification. Release record for specification and reference software version 1.0.0, 28 May 2026; supersedes the 5 January 2026 working draft.
<https://av2.aomedia.org/>
- [38] IETF. RFC 9628. RTP Payload Format for VP9 Video. March 2025. Standards Track. Sections 3, 4 and 6.
<https://www.rfc-editor.org/rfc/rfc9628.html>
- [39] The WebM Project. VP9 Video Codec Summary. Records initial availability on 17 June 2013.
<https://www.webmproject.org/vp9/>
- [40] IETF. RFC 7798. RTP Payload Format for High Efficiency Video Coding (HEVC). March 2016. Sections 7 and 8.
<https://www.rfc-editor.org/rfc/rfc7798.html>
- [41] Chrome for Developers. Chrome 136 release notes. H265 (HEVC) codec support in WebRTC. 2025.
<https://developer.chrome.com/release-notes/136>
- [42] WebKit. WebKit Features in Safari 18.0. 16 September 2024. WebRTC section. The article misprints the HEVC RTP RFC number as 7789; the correct reference is RFC 7798 [40].
<https://webkit.org/blog/15865/webkit-features-in-safari-18-0/>
- [43] IETF AVTCORE Working Group. H.265 Profile for WebRTC. draft-ietf-avtcore-hevc-webrtc-09, 20 July 2026. Internet-Draft; intended Standards Track.
<https://datatracker.ietf.org/doc/html/draft-ietf-avtcore-hevc-webrtc-09>
- [44] ONVIF. Uplink Specification. Current-version URL.
<https://www.onvif.org/specs/srv/uplink/ONVIF-Uplink-Spec.pdf>

VIDEO STREAMING PROTOCOLS AND DELIVERY METHODS FOR PHYSICAL SECURITY

- [45] IETF WISH Working Group. WebRTC-HTTP Egress Protocol (WHEP). draft-ietf-wish-whep-04, 22 June 2026. Internet-Draft; intended Standards Track.
<https://datatracker.ietf.org/doc/html/draft-ietf-wish-whep-04>
- [46] IETF. RFC 9605. Secure Frame (SFrame): Lightweight Authenticated Encryption for Real-Time Media. August 2024. Sections 1, 5, 7 and 9.
<https://www.rfc-editor.org/rfc/rfc9605.html>
- [47] W3C. WebCodecs. Latest published Working Draft. Abstract and codec interfaces; no required codec set.
<https://www.w3.org/TR/webcodecs/>
- [48] ONVIF. Media Signing Specification. Version 26.06, June 2026. Sections 1, 4 and 5; Annex B. Current-version URL.
<https://www.onvif.org/specs/stream/ONVIF-MediaSigning-Spec.pdf>
- [49] ONVIF. Media Signing Add-on (Release Candidate). Public overview.
<https://www.onvif.org/add-on/media-signing-add-on/>
- [50] Veovera Software Organization. Enhanced RTMP v2. Project specification and protocol extensions. Current document.
<https://veovera.org/docs/enhanced/enhanced-rtmp-v2>
- [51] IETF. RFC 2435. RTP Payload Format for JPEG-compressed Video. October 1998.
<https://www.rfc-editor.org/rfc/rfc2435.html>
- [52] IETF. RFC 6184. RTP Payload Format for H.264 Video. May 2011. Sections 8.1 and 8.2: profile, level and packetization signaling.
<https://www.rfc-editor.org/rfc/rfc6184.html>
- [53] Chromium project. Intent to Ship: H265 (HEVC) codec support in WebRTC. 3 March 2025. Platform and hardware dependencies; negotiation.
<https://groups.google.com/a/chromium.org/g/blink-dev/c/3h8IL8a377c>
- [54] Google for Developers. Delivering Live YouTube Content via RTMPS. Used for the RTMP-over-TLS definition, not service comparison.
<https://developers.google.com/youtube/v3/live/guides/rtmps-ingestion>
- [55] WHATWG. HTML Standard. Section 4.8.4.3: image processing, including multipart/x-mixed-replace resources. Living Standard.
<https://html.spec.whatwg.org/multipage/images.html>
- [56] ONVIF. Export File Format Specification. Version 26.06, June 2026.
<https://www.onvif.org/specs/2606/ONVIF-ExportFileFormat-Spec-v2606.pdf>
- [57] ONVIF. Profiles. Overview of ONVIF conformance profiles.
<https://www.onvif.org/profiles/>