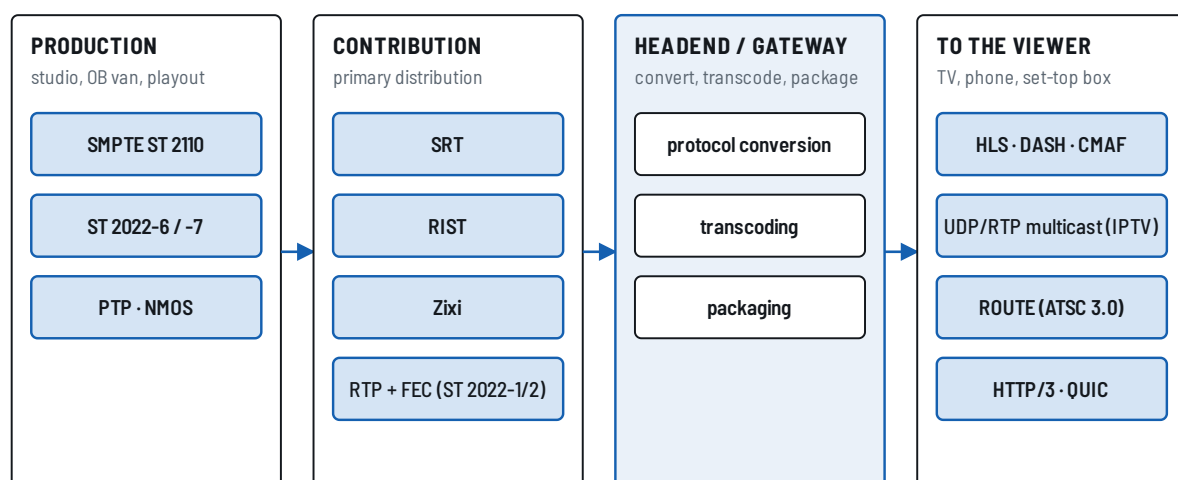


VIDEO OVER IP · STREAMING · ENGINEERING GUIDE

Video over IP Streaming Protocols

RTP, HLS, DASH, SRT, RIST, Zixi, SMPTE ST 2022/2110 and
ROUTE – how they work and where each one fits

IRENIS / BLANKOM · Technical guide · Ralf Riedel, Director Technical Sales & Engineering, IRENIS GmbH



RTMP survives mainly as an ingest protocol from encoders to streaming platforms.

Contents

PART 1 · FOUNDATIONS

1	Why video needs its own transport protocols	2
2	RTP and RTCP	5

PART 2 · DELIVERY TO VIEWERS

3	HTTP adaptive streaming	8
4	HLS, DASH and CMAF	11
5	Low-latency HTTP streaming	13
6	QUIC and HTTP/3	15
7	RTMP	17

PART 3 · CONTRIBUTION AND PRIMARY DISTRIBUTION

8	Error recovery on IP links	19
9	Zixi	22
10	SRT – Secure Reliable Transport	24
11	RIST – Reliable Internet Stream Transport	27
12	SRT, RIST and Zixi compared – and the gateway to the viewer	30

PART 4 · INSIDE THE BROADCAST FACILITY

13	From SDI to IP in the facility	32
14	SMPTE ST 2022	34
15	Seamless protection with ST 2022-7	36
16	SMPTE ST 2110, PTP and NMOS	38

PART 5 · BROADCAST, MULTICAST AND DECISIONS

17	ROUTE, FLUTE and ATSC 3.0	41
18	Choosing the right protocol	43

PART 6 · IMPLEMENTATION

19	Streaming in hardware – FPGA IP cores	45
----	---------------------------------------	----

APPENDIX

A	Glossary of abbreviations	48
B	Sources and references	51

Contact	53
---------	----

Video over IP Streaming Protocols · 1st edition 2026 · Version 1.0, September 2026

Author: Ralf Riedel, Director Technical Sales & Engineering, IRENIS GmbH

© 2022–2026 BLANKOM, IRENIS GmbH. All rights reserved. BLANKOM® and IRENIS® are registered brands of IRENIS GmbH. Other product and protocol names are trademarks of their owners.

Based on the 2022 handout “Video over IP protocols”, which Ralf Riedel compiled from third-party material: the ATEME white paper “Video over IP Protocols” and IP-core data sheets. This edition is a complete rewrite with new diagrams and updated standards; all sources are listed in Appendix B. Standards and specifications change; check the current version before you specify equipment. Technical details are subject to change without notice.

Online edition: <https://www.irenis-blankom.com/video-over-ip-streaming-protocols-guide/>

Why video needs its own transport protocols

The internet's classic transport protocols were designed long before anyone streamed live television. Every video protocol in this guide exists to close that gap for one particular job.

IRENIS / BLANKOM · Technical guide · Ralf Riedel, Director Technical Sales & Engineering, IRENIS GmbH

In this chapter [From broadcast to IP](#) [Why UDP and TCP are not enough](#) [Application layer framing](#)
[How this guide is organised](#)

KEY POINTS

- Video over IP now runs through the whole chain, from production to contribution to the viewer.
- UDP is fast but unprotected; TCP is reliable but its retransmissions add delay and it cannot multicast. Neither fits every video use case.
- Application layer framing (ALF) lets each protocol add exactly the error and rate control its use case needs. That is why there are so many protocols.

From broadcast to IP

Over the past fifteen to twenty years, the way people watch television has changed fundamentally. Viewers want to watch when they like, wherever they are, on whatever screen is at hand. Only the internet could satisfy this demand on a large scale: smart TVs, tablets, PCs, phones and game consoles all speak the core internet protocols IP, TCP and UDP, and they can connect from almost anywhere.

The same move to IP has since reached the professional side of television:

- **Production.** Cameras have IP interfaces, and IP switches and routers replace SDI routers and cabling.
- **Post-production** works on shared storage over IP networks.
- **Contribution and primary distribution** – the feeds from broadcasters to cable, satellite, IPTV and OTT operators – increasingly run over IP, including the public internet.

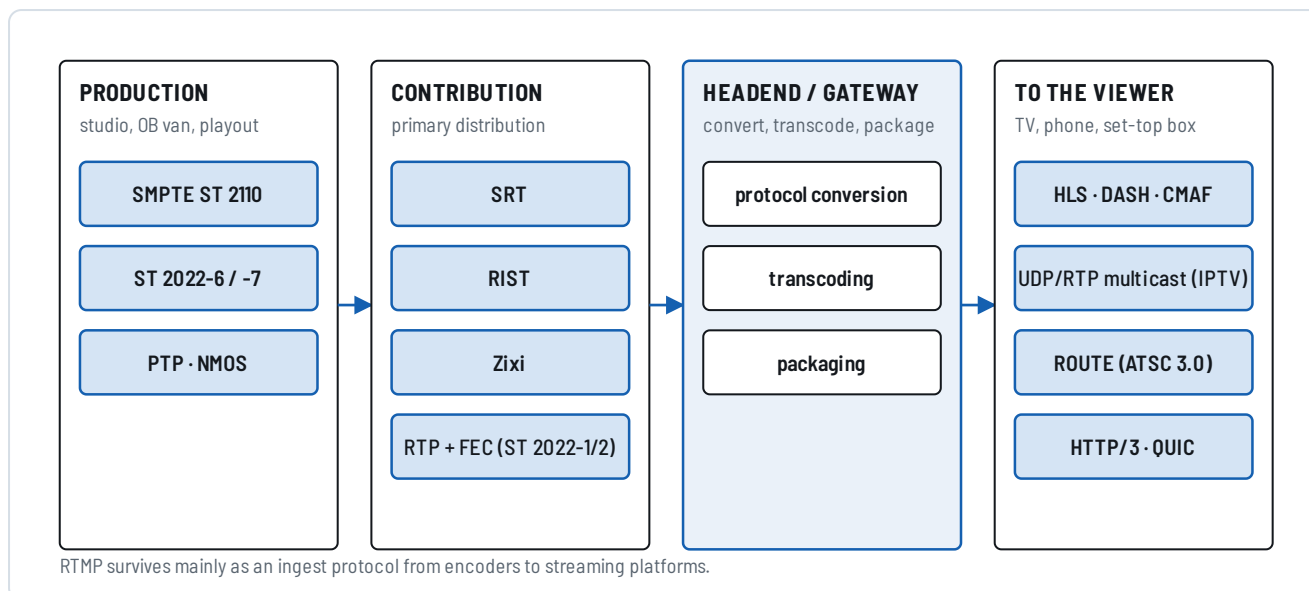


Figure 1.1 Where each protocol is used along the chain from production to the viewer.

Why UDP and TCP are not enough

UDP and TCP were designed long before video became the largest share of internet traffic, and neither is ideal for it.

	UDP	TCP
Delivery	Best effort: lost packets stay lost	Reliable: lost packets are retransmitted
Order	Packets may arrive out of order	Always in order
Delay	Low and constant	Grows with every retransmission and with distance
Multicast	Yes	No, point-to-point only
Rate control	None	Built-in congestion control, which can throttle a live stream

A live stream on plain UDP breaks up as soon as the network loses packets. On TCP it stays intact, but each lost packet stalls the stream for at least one round trip, and TCP's congestion control can lower the rate below what the video needs.

Application layer framing

The problem was recognised as early as 1990, when Clark and Tennenhouse proposed **application layer framing (ALF)**: the application itself takes over transport functions such as error and rate control and tailors them to its needs.

Almost every protocol in this guide follows this idea. Most run on UDP and add what the use case needs – sequence numbers, timestamps, retransmission, forward error correction, encryption. The first widely deployed example was RTP with its companion RTCP in the mid-1990s ([Chapter 2](#)). Many have followed: some proprietary, some open source, some fully standardised, others still drafts. The result is the alphabet soup of names that this guide sorts out.

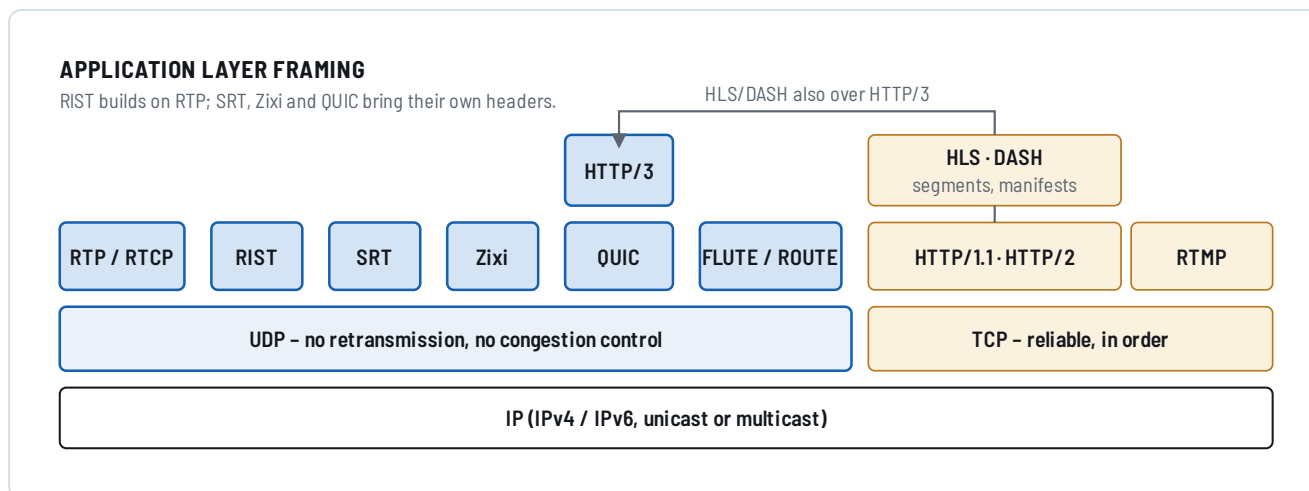


Figure 1.2 Which protocol runs on UDP and which on TCP.

How this guide is organised

Each chapter covers one protocol or one family, explains how it works and names the use cases where it brings the most value. The chapters follow the chain in the figure above:

- **Foundations:** RTP and RTCP ([Chapter 2](#)).
- **Delivery to viewers:** HTTP adaptive streaming ([Chapter 3](#)), HLS, DASH and CMAF ([Chapter 4](#)), low latency ([Chapter 5](#)), QUIC and HTTP/3 ([Chapter 6](#)) and RTMP ([Chapter 7](#)).
- **Contribution and primary distribution:** error-recovery basics ([Chapter 8](#)), Zixi ([Chapter 9](#)), SRT ([Chapter 10](#)), RIST ([Chapter 11](#)) and a comparison with the gateway to the viewer ([Chapter 12](#)).
- **Inside the broadcast facility:** why SDI moves to IP ([Chapter 13](#)), SMPTE ST 2022 ([Chapter 14](#)), seamless protection with ST 2022-7 ([Chapter 15](#)) and ST 2110 with PTP and NMOS ([Chapter 16](#)).
- **Broadcast and decisions:** ROUTE and ATSC 3.0 ([Chapter 17](#)) and the protocol choice by use case ([Chapter 18](#)).
- **Implementation:** how these functions are built in FPGA hardware ([Chapter 19](#)).

COMPANION GUIDE

The IPTV network inside a hotel, hospital or residential complex – multicast addressing, IGMP, headend design – is covered in the **IPTV Headend Planning Guide** of this series. This guide looks at the protocols that bring video to and from such networks.

RTP and RTCP

RTP was the first successful application-layer video protocol. It is still the basis of IPTV, of SMPTE ST 2022 and ST 2110, and of RIST.

IRENIS / BLANKOM · Technical guide · Ralf Riedel, Director Technical Sales & Engineering, IRENIS GmbH

In this chapter

[Technical overview](#)
[RTCP, the control channel](#)
[Deployment and typical applications](#)

KEY POINTS

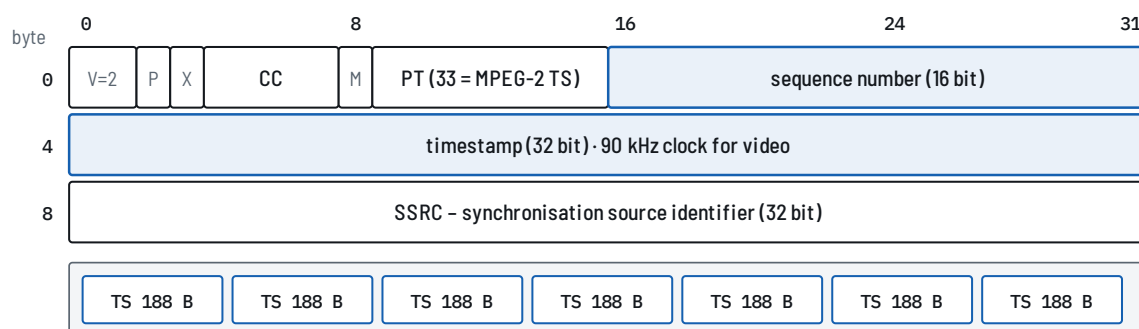
- RTP adds a 16-bit sequence number and a 32-bit timestamp to every packet. The receiver uses them to detect loss, restore the order and play out in time.
- RTCP carries reports on loss, jitter and delay on the next odd port. RTP itself does not recover lost packets.
- For viewers, HTTP streaming replaced RTP. RTP remains central in contribution, IPTV networks and broadcast facilities, usually with FEC.

Technical overview

The **Real-time Transport Protocol (RTP)** was specified in 1996 in RFC 1889 and revised in 2003 as RFC 3550. It normally runs on UDP and adds a small header to each packet. Two fields in it do most of the work:

- The **sequence number** (16 bit) increases by one with every packet. A gap shows that a packet was lost; a jump backwards shows that packets arrived out of order.
- The **timestamp** (32 bit) gives the sampling instant of the payload in units of the media clock – 90 kHz for video. The receiver uses it to synchronise audio and video and to play out each packet at the right time.

RTP FIXED HEADER · 12 BYTES (RFC 3550)



Payload: typically 7 transport stream packets = 1316 bytes (RFC 2250).

Loss detection: a gap in the sequence numbers. Reordering: sort by sequence number.

Figure 2.1 The 12-byte RTP header and a typical payload of seven TS packets.

RTP deliberately does not define how a particular medium is carried. Separate **payload format** specifications do that. For IPTV the important one is RFC 2250: an RTP packet carries a whole number of 188-byte MPEG-2 transport stream packets, typically seven (1316 bytes), which fits into one Ethernet frame. The payload type for MPEG-2 TS is 33.

RTCP, the control channel

The **RTP Control Protocol (RTCP)** is specified in the same RFC. It runs beside the media stream, conventionally on the next odd port (RTP on 5004, RTCP on 5005). Senders and receivers exchange reports:

- **Sender reports** relate the RTP timestamp to wall-clock time and count packets and bytes sent.
- **Receiver reports** tell the sender the loss rate, the jitter and the round-trip delay.

A sender can use this information, for example, to adapt its rate. RTCP is limited to a small share of the session bandwidth (typically 5 %).

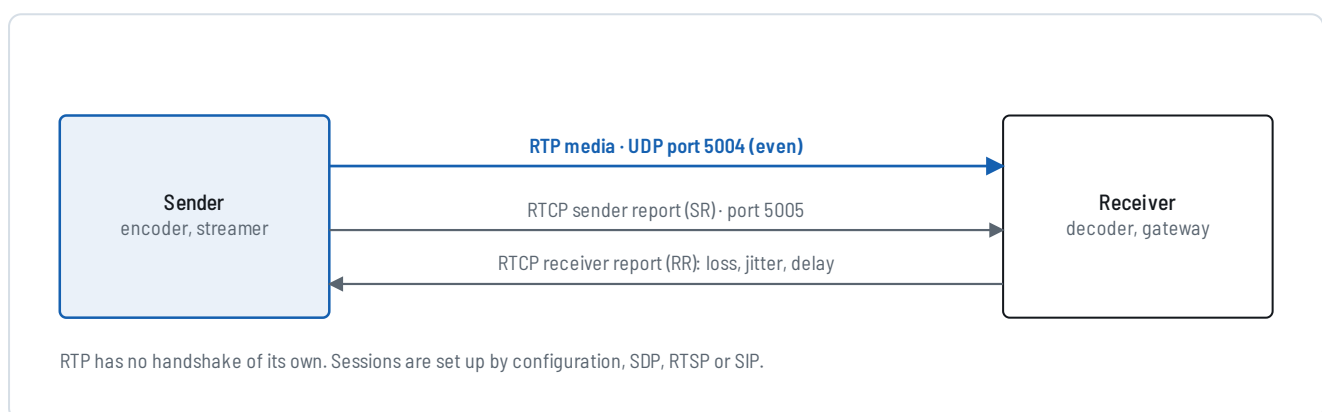


Figure 2.2 RTP carries the media; RTCP reports travel in both directions on the next port.

CORRECTION TO THE 2022 HANDOUT

RTP has no handshake or session set-up of its own. Sessions are configured statically (as in most IPTV headends) or signalled with SDP, RTSP or SIP. RTSP also provides the play, pause and teardown commands.

Deployment and typical applications

RTP was the first successful protocol for streaming to end users in the late 1990s and early 2000s. It had two weaknesses in that role:

1. Firewalls often blocked the UDP ports it used.
2. It was designed before content delivery networks (CDNs) and could not use their caches. It did not scale to millions of viewers watching the same live event.

When HTTP adaptive streaming arrived in the late 2000s ([Chapter 3](#)), it replaced RTP for delivery over the internet to viewers.

RTP remains the workhorse of professional video over IP:

- **Contribution and primary distribution** over managed networks, for example from a national centre to regional stations, usually protected by FEC according to SMPTE ST 2022-1 ([Chapter 8](#), [Chapter 14](#)).

- **IPTV networks** in hotels, hospitals and residential complexes, where UDP or RTP multicast carries each TV service.
- **Broadcast facilities**, where ST 2022-6 and ST 2110 are built on RTP ([Chapter 14](#), [Chapter 16](#)).
- **RIST**, which uses RTP and RTCP as its foundation ([Chapter 11](#)).

RTP has only basic support for encryption. Where content protection matters, use SRTP or a protocol with built-in encryption such as SRT or RIST Main Profile.

HTTP adaptive streaming

Today almost all video watched on phones, tablets and smart TVs arrives as a series of small files over HTTP. The player chooses, segment by segment, the best quality the connection allows.

IRENIS / BLANKOM · Technical guide · Ralf Riedel, Director Technical Sales & Engineering, IRENIS GmbH

In this chapter

[Why HTTP](#)[Multi-bitrate encoding](#)[Adaptive bit rate streaming](#)

KEY POINTS

- HTTP runs over TCP on ports 80/443, passes almost every firewall and uses the caches of content delivery networks.
- Content is encoded at several bit rates and cut into aligned segments. A manifest lists variants and segments.
- The player measures its throughput and switches between variants at segment boundaries. The intelligence sits in the client, which is why the scheme scales.

Why HTTP

The shortcomings of RTP ([Chapter 2](#)) were overcome in the late 2000s by adaptive streaming over HTTP.

HTTP was created in the mid-1990s (HTTP/1.0 in 1996, HTTP/1.1 in 1997) to move web pages from a server to a browser. As the web grew, an entire infrastructure grew with it: **content delivery networks (CDNs)**, hierarchies of servers and caches that bring content close to millions of users.

HTTP transactions are stateless: a client sends a GET request for a file – a web page or a piece of media – and the server, or an edge cache that already holds a copy, returns it. HTTP runs over TCP and inherits its error and flow control. Web access became so essential that firewalls let HTTP (TCP port 80) and HTTPS (port 443) through while blocking most other ports, including the UDP ports used by RTP.

Multi-bitrate encoding

Streaming pioneers in the 1990s faced a dilemma. Viewers connected with very different bandwidths – dial-up, ISDN, DSL, leased lines. Encode at a high bit rate and most of the audience cannot watch; encode for the lowest common denominator and the best-connected customers get poor pictures.

The first answer, introduced by RealNetworks in the late 1990s, was to encode the same content at several bit rates and let the user choose at the start. This helped, but it had two flaws:

1. Viewers had to know their connection speed, including the overheads of TCP, IP, Ethernet and the physical layer. Few could.
2. It assumed the bandwidth stays constant for the whole programme – a poor assumption on the shared, constantly changing internet.

Adaptive bit rate streaming

In 2007 Move Networks added two ideas that made the concept work:

- Each variant is encoded as a series of short, independently decodable **segments** instead of one file. A two-hour film, for example, becomes 720 segments of 10 seconds. Segment boundaries are aligned across all variants.
- A **manifest** lists the variants (the bit rates of the **ABR ladder**) and the location and duration of every segment.

At the start of a session the player loads the manifest and requests segments with HTTP GET, beginning with a default variant. While downloading it continuously estimates its throughput and moves up or down the ladder at segment boundaries. Its goal is the highest quality that does not exceed the estimated bandwidth, while following fluctuations.

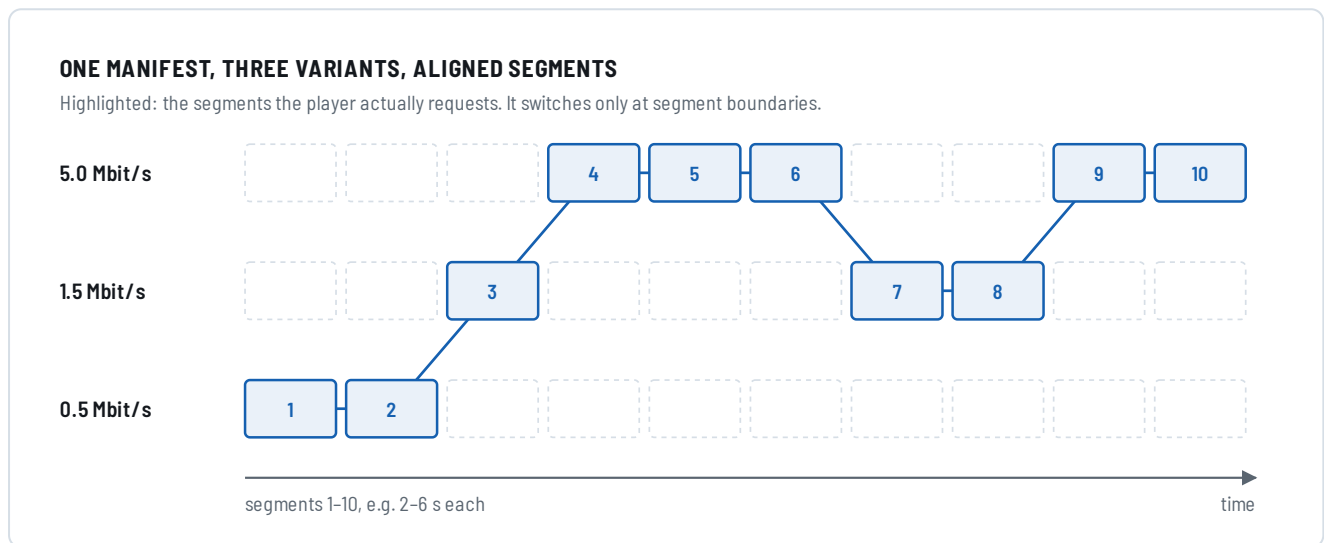


Figure 3.1 Adaptive bit rate: the player moves between variants at segment boundaries.

Strictly speaking this is not streaming at all but a series of file downloads. That is its strength:

- Segments are ordinary files, so **CDN caches** serve them and the scheme scales to very large audiences.
- HTTP(S) passes **firewalls**.
- The **bandwidth adaptation** is dynamic.
- The **intelligence sits in the client**, not in the server. Unlike RTP, there is no session state on the server and no RTCP exchange, although TCP below still performs error and flow control.

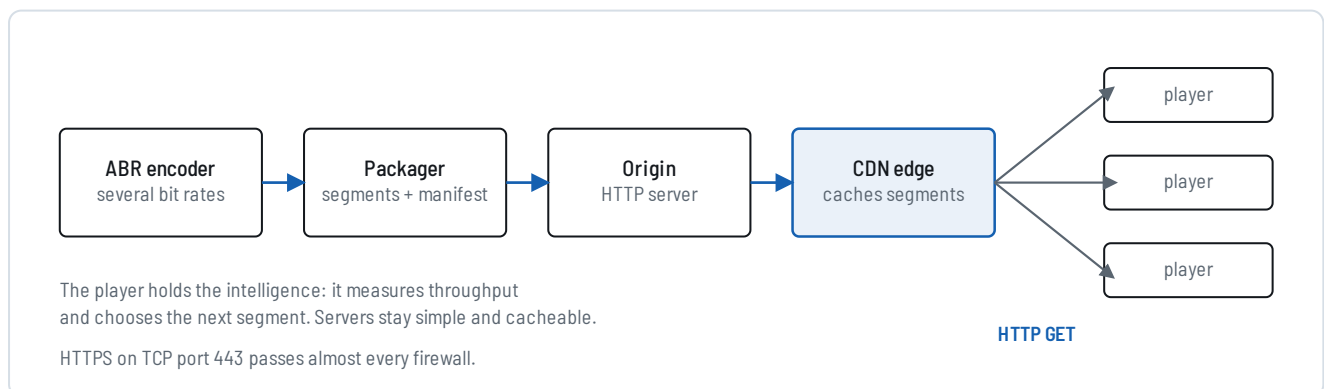


Figure 3.2 The HTTP streaming chain from encoder to player.

DESIGNING AN ABR LADDER

Choose the top rung for the best screen you serve, the lowest for the weakest mobile connection, and space the rungs roughly by a factor of 1.5 to 2 in bit rate. More rungs mean smoother adaptation, but also more encoding capacity, storage and CDN cache.

The next chapter shows how the competing implementations of this idea converged on HLS, DASH and CMAF ([Chapter 4](#)).

HLS, DASH and CMAF

Four vendors turned the same idea into four incompatible formats. Two survived, and a common container now lets them share the same files.

IRENIS / BLANKOM · Technical guide · Ralf Riedel, Director Technical Sales & Engineering, IRENIS GmbH

In this chapter

Fragmentation

MPEG-DASH

CMAF: one set of files for both

KEY POINTS

- Smooth Streaming (Microsoft), HLS (Apple) and HDS (Adobe) used the same principle but different containers and manifests.
- MPEG-DASH (2012) is the open standard. DASH and HLS dominate delivery to viewers today; Smooth Streaming and HDS are obsolete.
- CMAF stores the media once as fragmented MP4. An HLS playlist and a DASH manifest both point to the same segments, which halves storage and CDN cache.

Fragmentation

Move Networks' idea ([Chapter 3](#)) was quickly taken up by the large streaming vendors:

TECHNOLOGY	VENDOR	YEAR	MEDIA CONTAINER	MANIFEST
Smooth Streaming	Microsoft	2008	MP4 (fragmented)	XML (.ismc)
HTTP Live Streaming (HLS)	Apple	2009	MPEG-2 TS, fMP4 since 2016	playlist (.m3u8)
HTTP Dynamic Streaming (HDS)	Adobe	2010	F4V	XML (.f4m)
MPEG-DASH	ISO/IEC 23009-1	2012	fMP4 (also TS)	XML (.mpd)

All followed the same principle but used different containers and manifest syntaxes, and they did not interoperate. Content providers had to encode and package the same content several times, and CDNs had to store and cache every copy – wasting storage and bandwidth.

MPEG-DASH

To end the fragmentation, the Moving Picture Experts Group (MPEG) developed **Dynamic Adaptive Streaming over HTTP (DASH)**, ratified in 2011 and published as ISO/IEC 23009-1 in 2012. DASH uses the fragmented MP4 container and a standardised manifest, the Media Presentation Description (MPD). It supports features such as dynamic ad insertion, several camera angles, multiple audio languages, subtitles and captions, and mixing live and recorded content.

DASH has replaced Smooth Streaming and HDS. HLS remains in wide use because Apple devices – a large share of the market – support it natively. **DASH and HLS together dominate IP delivery to viewers today.**

CMAF: one set of files for both

The **Common Media Application Format (CMAF, ISO/IEC 23000-19, 2018)** defines a single container, based on fragmented MP4, for segments that are delivered by DASH as well as HLS. The packager produces the media once; an HLS playlist and a DASH MPD both describe the same segments.

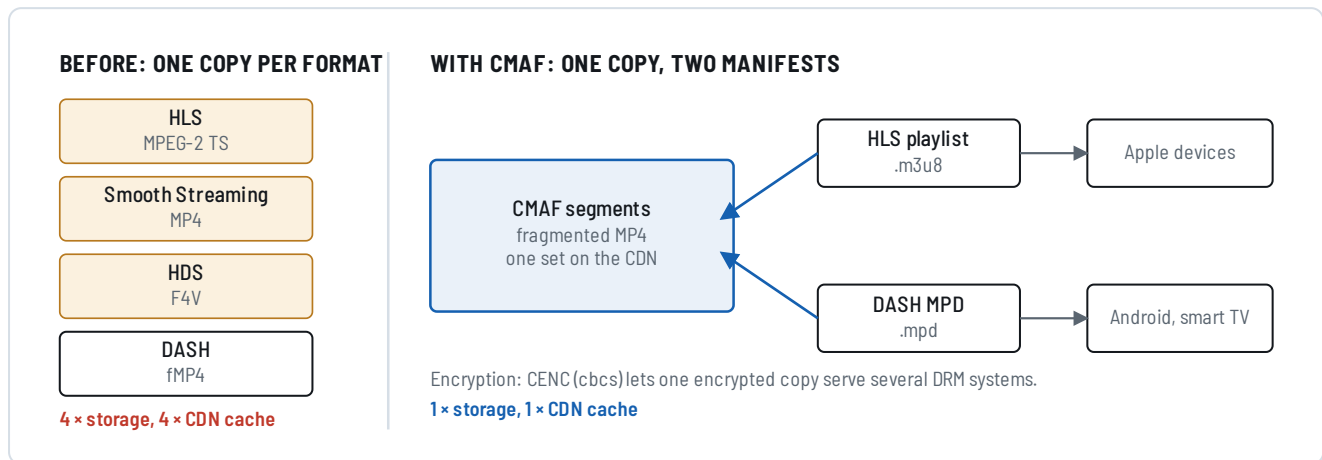


Figure 4.1 CMAF replaces four copies in four formats with one set of segments.

Two further tools help:

- **Byte-range addressing.** Instead of one file per segment, a player can request a range of bytes from a larger file. Both HLS and DASH support this, which reduces the number of files on the origin and CDN.
- **Common encryption (CENC).** With the *cbcs* scheme, one encrypted copy can be decrypted by the major DRM systems.

RECOMMENDATION FOR DELIVERY TO VIEWERS

Package once in CMAF, publish HLS and DASH manifests, and use byte-range addressing where your CDN supports it. For live streams use the low-latency variants ([Chapter 5](#)).

Low-latency HTTP streaming

A goal cheered by the neighbours half a minute before you see it spoils any live sports broadcast. Chunked encoding and transfer bring HTTP streaming close to broadcast latency.

IRENIS / BLANKOM · Technical guide · Ralf Riedel, Director Technical Sales & Engineering, IRENIS GmbH

In this chapter

[Where the delay comes from](#)

[Chunked encoding and transfer](#)

[The standard variants](#)

KEY POINTS

- Classic HTTP live streams run 20–45 seconds behind real time because players buffer several whole segments.
- Chunked encoding splits each segment into chunks of about 500 ms that travel through packager, origin, CDN and player as soon as they are ready.
- Chunks are not independently decodable. The player still switches bit rates only at segment boundaries.

Where the delay comes from

Compared with RTP, HTTP adaptive streaming has one weakness: latency. A typical live session runs 20 to 45 seconds behind the live event. The reasons add up along the chain:

- The encoder can release a segment only when it is complete – with 6-second segments, that alone is 6 seconds.
- The packager, origin and CDN each handle whole segments.
- Players keep a buffer of typically three segments before starting playback, to ride out throughput variations.

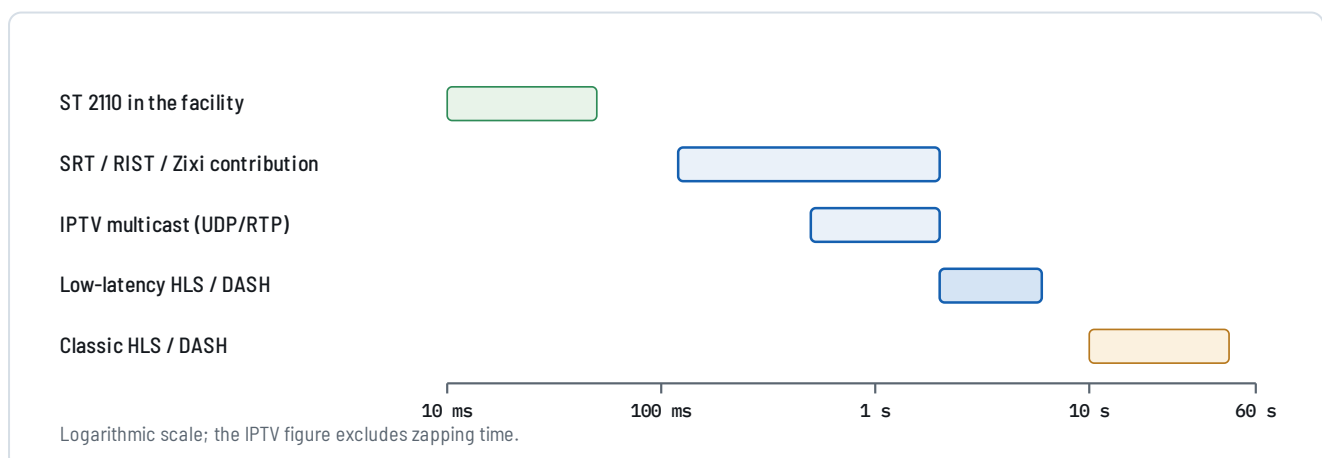


Figure 5.1 Typical end-to-end latency of the protocol families.

Chunked encoding and transfer

The key technique is to cut each segment (typically 2 to 4 seconds) into smaller **chunks** of as little as 500 ms. Each chunk leaves the encoder as soon as it is encoded and is forwarded – with HTTP chunked transfer encoding – through the packager, the origin server and the CDN to the player, without waiting for the whole segment.

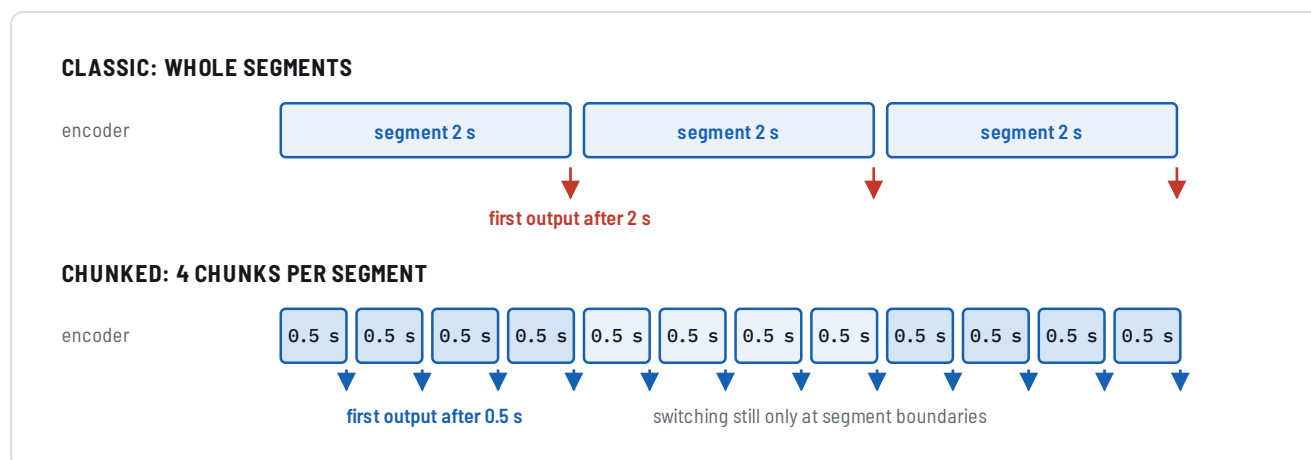


Figure 5.2 Chunked encoding releases data every 500 ms instead of every segment.

Chunks are not independently decodable; only a whole segment starts with a key frame. Bit-rate switching therefore still happens at segment boundaries. End-to-end latencies of a few seconds are reached in practice.

The standard variants

VARIANT	PRINCIPLE
Low-latency DASH (LL-DASH)	CMAF chunks with chunked transfer; guidelines by DASH-IF and DVB
Low-Latency HLS (LL-HLS, Apple, 2019/2020)	<i>Partial segments</i> announced in the playlist, blocking playlist reload and preload hints
CMAF chunks	Common basis: the same chunked CMAF media can serve LL-DASH and LL-HLS

The need for low latency has made CMAF the natural common format for both DASH and HLS ([Chapter 4](#)).

GETTING LOW LATENCY IN PRACTICE

- ☐ Encoder and packager must support CMAF chunks (or LL-HLS partial segments)
- ☐ Origin and CDN must pass chunked transfer through without buffering whole segments
- ☐ Short segments (2 s) with frequent key frames
- ☐ Player configured for a small live-edge buffer; test under real network conditions
- ☐ Accept a small loss in compression efficiency and more requests per second on the CDN

QUIC and HTTP/3

QUIC applies the idea of application layer framing to the web itself. HTTP/3 runs over UDP, and adaptive streams can benefit without any change to their formats.

IRENIS / BLANKOM · Technical guide · Ralf Riedel, Director Technical Sales & Engineering, IRENIS GmbH

In this chapter

[Background: head-of-line blocking](#)

[HTTP/3 over QUIC](#)

[Status](#)

KEY POINTS

- HTTP/2 multiplexes many streams over one TCP connection; one lost packet blocks them all. QUIC delivers each stream independently.
- QUIC includes TLS 1.3. Connection and encryption set-up take one round trip, and zero when a connection is resumed.
- Connections survive a change of network, for example from mobile data to Wi-Fi. QUIC is standardised (RFC 9000, 2021) and HTTP/3 (RFC 9114, 2022).

Background: head-of-line blocking

With HTTP/1.1 a browser typically opens half a dozen TCP connections to a server. HTTP/2 introduced **multiplexing**: many logical streams share one TCP connection. That reduces connection overhead and lets TCP's congestion control work more efficiently.

But multiplexing brings a new problem. TCP delivers bytes strictly in order. If one packet on the connection is lost, **all** logical streams wait until it has been retransmitted – even those whose data arrived complete. This is called *head-of-line blocking*.

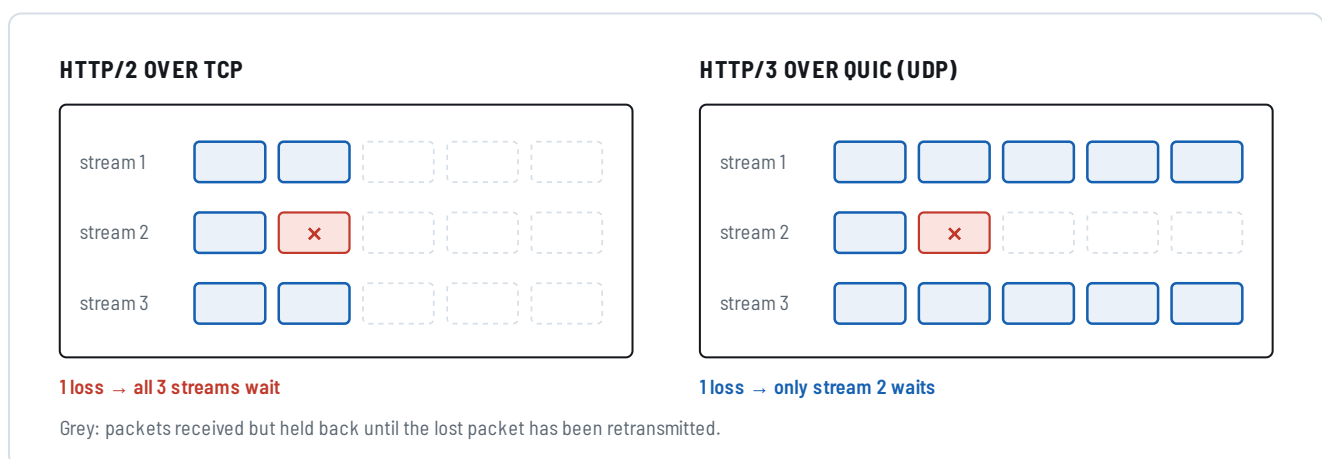


Figure 6.1 Head-of-line blocking with TCP and independent streams with QUIC.

HTTP/3 over QUIC

The solution is an application-layer transport in the spirit of ALF ([Chapter 1](#)): **QUIC** runs over UDP and provides TCP-like reliability with lower latency. It keeps the multiplexing of HTTP/2, but each logical stream is delivered independently, with error and congestion control performed at the application layer. HTTP over QUIC

is **HTTP/3**.

Important properties:

- **Built-in encryption.** QUIC always uses TLS 1.3. Key exchange is part of the connection handshake, which saves at least one round trip at start-up. A resumed connection can send data immediately (0-RTT).
- **Connection migration.** A QUIC connection is identified by a connection ID, not by IP addresses and ports. When a phone moves from the mobile network to home Wi-Fi – often to save mobile data – the connection continues. With TCP it would time out and have to be re-established, which viewers notice as a stall.
- **Flow control** per stream and per connection.
- **Error recovery** by retransmission (ARQ) with unique packet numbers.

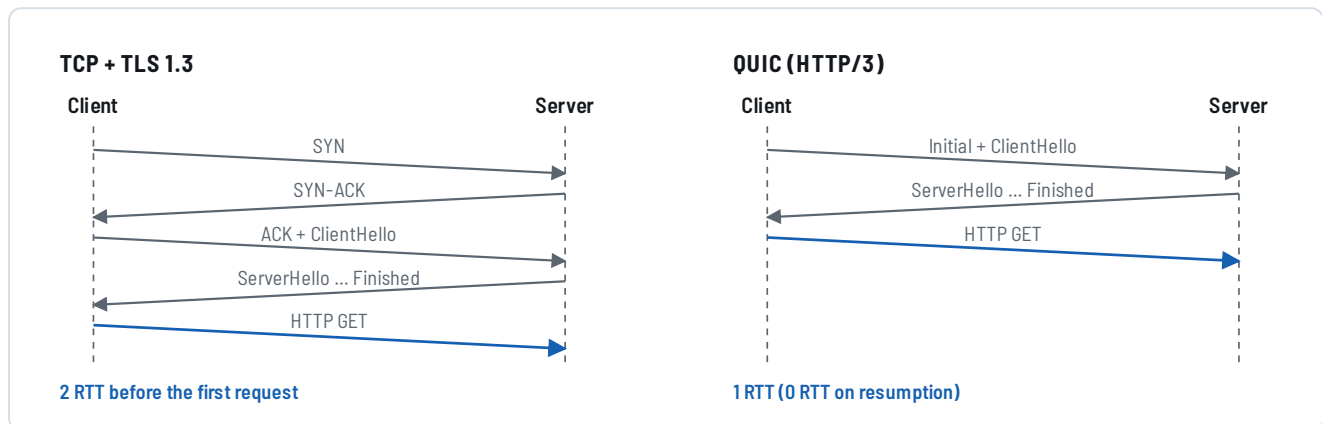


Figure 6.2 Round trips before the first request: TCP + TLS 1.3 versus QUIC.

Status

QUIC was first developed by Google and used in Chrome. The IETF standardised it in 2021 (RFC 9000 to 9002), and HTTP/3 followed in 2022 (RFC 9114). All major browsers, many CDNs and mobile apps support it. HLS and DASH segments are delivered over HTTP/3 without any change to their formats.

UPDATED SINCE THE 2022 HANDOUT

The handout described QUIC and HTTP/3 as drafts and mentioned a planned FEC extension. Both are now published standards; QUIC version 2 (RFC 9369, 2023) does not add FEC. The IETF is working on *Media over QUIC* (MoQ), a new live-media protocol on QUIC. Watch it, but it is not yet a basis for product decisions.

RTMP

RTMP was the protocol of the Flash player. The player is gone, but many encoders and platforms still use RTMP to hand over a live stream.

IRENIS / BLANKOM · Technical guide · Ralf Riedel, Director Technical Sales & Engineering, IRENIS GmbH

In this chapter

[How RTMP works](#)

[RTMP today](#)

KEY POINTS

- RTMP was developed by Macromedia (later Adobe) for Flash. It runs over TCP, by default on port 1935.
- Adobe ended Flash support at the end of 2020. RTMP is not used for delivery to viewers any more.
- It lives on as a simple ingest protocol from encoders to streaming platforms. For new contribution links, prefer SRT or RIST.

How RTMP works

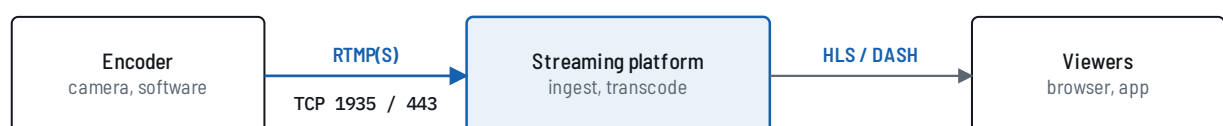
The **Real-Time Messaging Protocol (RTMP)** was developed by Macromedia, later part of Adobe, to stream multimedia between a Flash media server and a Flash client, usually a browser plug-in.

- RTMP runs over **TCP** (default port 1935) and relies on its error and flow control.
- Audio, video and control messages travel in separate virtual channels over one connection.
- The media streams are cut into **chunks** whose maximum size the two sides negotiate (the default is 128 bytes).
- Each chunk header carries timing information to keep audio and video synchronised.
- Content protection uses TLS (**RTMPS**) or the lightweight Adobe encryption RTMPE.

RTMP today

Flash was hugely popular for video in the late 1990s and 2000s. Adobe ended its support on 31 December 2020, and browsers removed the plug-in. RTMP is no longer used for delivery to viewers.

It survives as an **ingest protocol**: many encoders, cameras and software tools push live streams by RTMP(S) to social media and streaming platforms, which transcode them and deliver HLS or DASH.



Flash players are gone since 2021; RTMP lives on as a simple, widely supported ingest protocol.

Figure 7.1 RTMP as the ingest protocol to a streaming platform.

RECOMMENDATION

Use RTMP only where a platform or legacy device requires it. For contribution over the internet, SRT ([Chapter 10](#)) and RIST ([Chapter 11](#)) offer lower latency, packet-loss recovery over UDP and stronger encryption. An IP-to-IP gateway can convert an incoming RTMP stream into multicast for an IPTV network ([Chapter 12](#)).

Error recovery on IP links

Every contribution protocol has to answer one question – what happens when the network loses a packet? There are only three basic answers, and each has its price.

IRENIS / BLANKOM · Technical guide · Ralf Riedel, Director Technical Sales & Engineering, IRENIS GmbH

In this chapter [Three strategies](#) [Forward error correction \(FEC\)](#) [Retransmission \(ARQ\)](#)
[Dual path \(seamless protection\)](#)

KEY POINTS

- FEC sends redundancy in advance. It needs no return channel and adds little delay, but costs a fixed overhead and fails on long loss bursts.
- ARQ retransmits only what is lost. It needs a return channel and a latency buffer of several round-trip times.
- Dual-path transmission (ST 2022-7) sends everything twice over separate routes. It costs double bandwidth but survives the loss of a whole path.

Three strategies

The protocols in the following chapters combine the same three building blocks in different ways. Understanding them makes it easy to compare SRT, RIST, Zixi and the SMPTE standards.

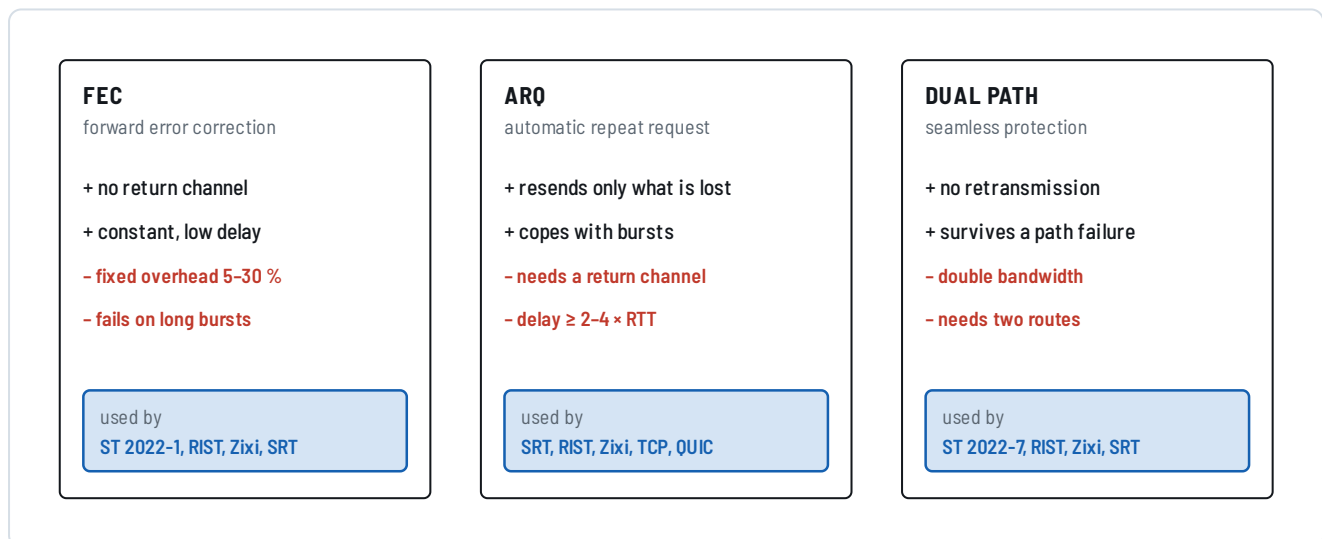


Figure 8.1 FEC, ARQ and dual-path transmission compared.

Forward error correction (FEC)

The sender adds **parity packets** calculated from the media packets. If a media packet is lost, the receiver rebuilds it from the others and the parity. No return channel is needed, so FEC also works on one-way links and for multicast.

The best-known scheme for video is the row and column FEC of **SMPTE ST 2022-1** (based on Pro-MPEG COP3). The media packets are arranged in a matrix of L columns and D rows; an XOR parity packet is calculated for each column and optionally for each row.

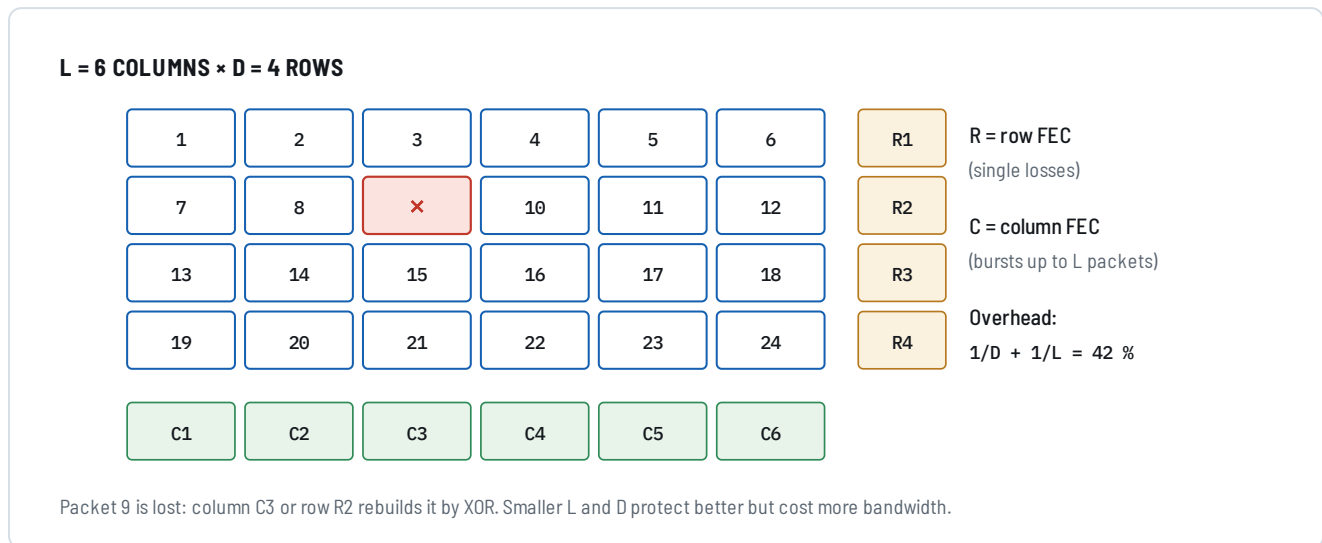


Figure 8.2 Row and column FEC: a lost packet is rebuilt from parity packets.

- Column FEC repairs a burst of up to L consecutive lost packets.
- Row FEC repairs single losses.
- The overhead is $1/D$ for column FEC plus $1/L$ for row FEC. The delay is about one matrix of packets.

FEC has a fixed cost, whether the network loses packets or not, and it cannot repair losses beyond its design, for example several packets in one column.

Retransmission (ARQ)

With **automatic repeat request (ARQ)** the receiver asks the sender to resend lost packets. Modern video protocols use **negative acknowledgements (NACK)**: the receiver reports only what is missing, and the sender retransmits only those packets (*selective retransmission*).

ARQ uses bandwidth only when packets are actually lost, and it copes with bursts. Its price is latency: the receiver must buffer long enough for a loss to be detected, reported and repaired, possibly more than once. Rules of thumb:

- The latency buffer should be **at least 3 to 4 times the round-trip time (RTT)**.
- The bandwidth must have headroom for retransmissions on top of the stream rate – typically 25 % or more on lossy links.
- ARQ needs a **return channel**; it does not work on one-way satellite links or for large multicast audiences.

Dual path (seamless protection)

The sender duplicates every packet and sends the copies over **two independent network paths**. The receiver takes each packet from whichever path delivers it first. This is standardised in **SMPTE ST 2022-7** ([Chapter 15](#)) and also offered by RIST, Zixi and SRT.

No retransmission, no acknowledgements and no failover software are needed; even the complete failure of one path is invisible at the output. The price is double bandwidth and two genuinely separate routes.

WHICH ONE WHEN?

- One-way link, multicast or very low latency: **FEC**.
- Internet link with a return channel and a latency budget of a few hundred milliseconds: **ARQ**, as in SRT and RIST.
- Mission-critical links and inside broadcast facilities: **dual path**, often combined with FEC or ARQ.

Zixi

Zixi was one of the first protocols to make broadcast-quality live video over the public internet practical. It is proprietary but widely deployed.

IRENIS / BLANKOM · Technical guide · Ralf Riedel, Director Technical Sales & Engineering, IRENIS GmbH

In this chapter

Technical overview

The Zixi ecosystem

Bonding and redundancy

Deployment

KEY POINTS

- Zixi runs over UDP and combines a content-aware FEC with ARQ. Latency and robustness can be traded against each other.
- Its ecosystem consists of feeders, a central broadcaster and receivers.
- Two network paths can be bonded for more throughput or used for hitless failover. Content is protected with AES and DTLS.

Technical overview

The **Zixi protocol**, developed by the company of the same name, delivers live video with low latency, reliability and content protection. It runs over UDP and recovers from errors with a combination of:

- **Z-FEC**, a dynamic, content-aware forward error correction, and
- **ARQ**, retransmission of lost packets ([Chapter 8](#)).

In low-latency operation (below about 1500 ms) Zixi uses this hybrid approach, with FEC adding up to 30 % overhead. The user can trade latency against robustness. The latency should be set to **at least three times the round-trip time**.

Zixi also adapts the transmission bit rate dynamically. Content is protected with **AES encryption** (up to 256-bit keys) and **DTLS**.

The Zixi ecosystem

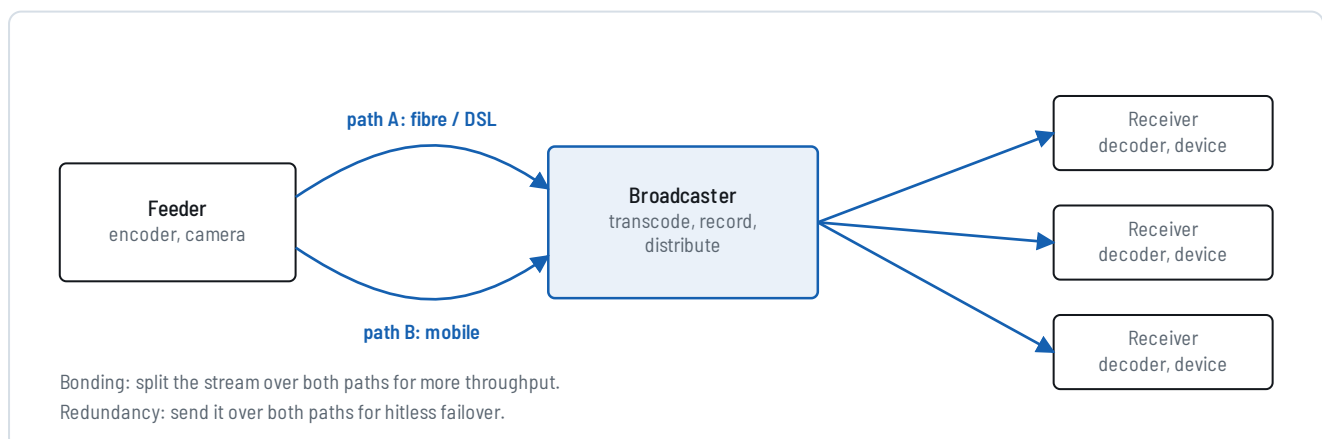


Figure 9.1 Feeder, broadcaster and receivers, with two bonded or redundant paths.

- The **feeder** is the source – an encoder or a camera.
- The **broadcaster** is the central component. It transcodes, transmuxes, records and distributes at several bit rates and in several protocols.
- The **receiver** is typically a decoder or an end device.

The Zixi protocol is used between feeder and broadcaster, between broadcaster and receiver, and between two broadcasters.

Bonding and redundancy

Zixi can use two transmission paths at the same time, for example a fixed line and a mobile connection:

- **Bonding:** the stream is split over both paths to reach a higher total throughput.
- **Redundancy:** the stream is sent over both paths for hitless failover.

Bonding requires the feeder output in *push* mode; this mode can also be used between two broadcasters.

Deployment

Despite being proprietary, Zixi is widely deployed, especially for primary distribution to remote facilities and over private, multicast-enabled networks. Many encoders, decoders and cloud services include Zixi under licence. Where interoperability between vendors is the main requirement, compare it with the open protocols SRT ([Chapter 10](#)) and RIST ([Chapter 11](#)).

SRT – Secure Reliable Transport

SRT is the most widely used open protocol for live contribution over the internet. It brings secure, reliable UDP transport to encoders and decoders of many vendors.

IRENIS / BLANKOM · Technical guide · Ralf Riedel, Director Technical Sales & Engineering, IRENIS GmbH

In this chapter Purpose Technical overview Loss recovery with ACK and NAK Encryption
FEC and bonding Deployment and typical applications

KEY POINTS

- SRT runs over UDP and recovers lost packets with NAK-based retransmission inside a fixed latency window, typically $3-4 \times \text{RTT}$.
- Payloads are encrypted with AES (128, 192 or 256 bit); the key is derived from a shared passphrase.
- SRT is open source and supported by hundreds of vendors. It is a de facto standard, not a formal one.

Purpose

Secure Reliable Transport (SRT) carries live contribution streams that need low latency, reliability and encryption. It uses UDP and was developed as an alternative to RTMP and HTTP streaming, whose reliance on TCP adds latency ([Chapter 5](#), [Chapter 7](#)). It was also an open alternative to Zixi ([Chapter 9](#)) that lets encoders and decoders of different vendors interoperate.

SRT was developed by Haivision, released as open source in 2017 and is promoted by the **SRT Alliance**, founded the same year, with several hundred member companies. The protocol is documented in an IETF Internet-Draft but is not an RFC; it is an industry standard by adoption.

Technical overview

SRT follows the ALF principle ([Chapter 1](#)) and descends from **UDT**, the UDP-based Data Transfer protocol. UDT was built for large file transfers; SRT adapts it for live streaming (its default mode) while still supporting file transfer.

- **Session-based.** Unlike stateless HTTP, SRT establishes a connection with a handshake, keeps it alive with keep-alive messages and closes it explicitly.
- **Sequence number and timestamp** in every packet, for loss detection, reordering and timing. The receiver plays out each packet at a fixed delay after it was sent (*timestamp-based packet delivery*), which removes network jitter.
- **Stream ID.** A caller can name the stream it wants, so one listener port can serve many streams.

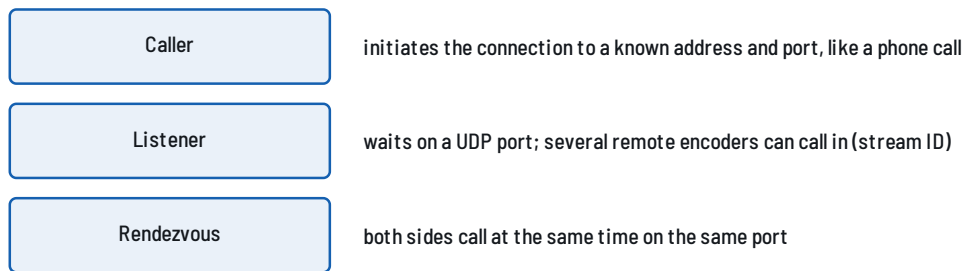


Figure 10.1 The three SRT connection modes.

Loss recovery with ACK and NAK

SRT recovers lost packets by ARQ (Chapter 8):

- The receiver sends a **NAK** as soon as it detects a gap. The sender keeps every packet in its buffer for the duration of the latency and retransmits lost packets with priority over new ones.
- The receiver waits for a missing packet until its playout time. If it has not arrived by then, it is dropped (*too-late packet drop*) and the stream continues.
- **Full ACKs** every 10 ms report round-trip time, link capacity and receive rate. The sender answers with an ACKACK so that the receiver can measure the RTT.

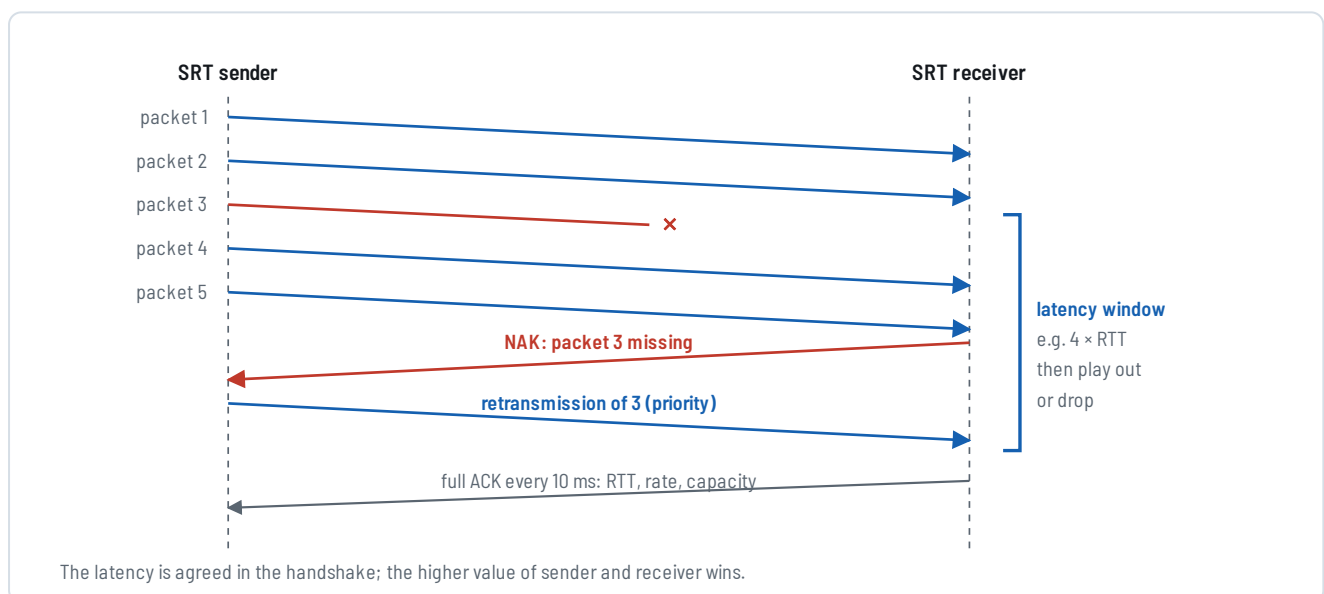


Figure 10.2 SRT recovers a lost packet within the agreed latency window.

The **latency** is set in milliseconds and agreed during the handshake; the higher value of both sides applies. A common rule is 3 to 4 times the RTT, with a minimum of about 120 ms, and more on lossy links. In live mode SRT does not throttle the stream; the encoder's rate control remains responsible for the bit rate.

Encryption

SRT encrypts the payload with **AES in counter mode (AES-CTR)** with 128, 192 or 256-bit keys; newer versions also offer AES-GCM. Counter mode allows decryption at any packet and tolerates loss. The key is derived from a passphrase configured on both sides and refreshed regularly. The header, including the sequence number, stays unencrypted.

FEC and bonding

The 2022 handout stated that SRT has no FEC mechanism and no bonding. Both have been added since:

- A built-in **packet filter** provides row and column FEC similar to ST 2022-1, which can be combined with ARQ.
- **Connection bonding** (socket groups, since SRT 1.5) sends a stream over several links, either in *broadcast* mode (every packet on every link, like ST 2022-7) or in *main/backup* mode.

Check the SRT version of your devices before relying on these features.

Deployment and typical applications

SRT is supported by a very wide range of encoders, decoders, gateways, cloud services and software tools. Typical uses:

- Contribution from studios, events and remote cameras over the public internet.
- Primary distribution to affiliates and operators.
- Feeding live channels into IPTV headends, where a gateway converts SRT into UDP multicast ([Chapter 12](#)).

SETTING UP AN SRT LINK

- | | |
|--------------------------|--|
| ☐ | Measure the RTT and packet loss of the link first |
| ☐ | Latency $\geq 4 \times \text{RTT}$ (more on lossy or mobile links); same or higher value on the receiver |
| ☐ | Bandwidth headroom of at least 25 % above the stream rate for retransmissions |
| ☐ | Passphrase on both ends (at least 10 characters) and key length agreed |
| ☐ | Decide who is caller and who is listener; open the listener's UDP port in the firewall |
| ☐ | Monitor retransmission rate and dropped packets, not just the picture |

RIST – Reliable Internet Stream Transport

RIST is the broadcast industry's answer to proprietary transport protocols – an open specification on top of RTP, written jointly by many vendors to guarantee interoperability.

IRENIS / BLANKOM · Technical guide · Ralf Riedel, Director Technical Sales & Engineering, IRENIS GmbH

In this chapter [Background](#) [Simple Profile \(TR-06-1, 2018\)](#) [Bonding](#) [Main Profile \(TR-06-2, 2020\)](#)
[Advanced Profile \(TR-06-3\)](#) [Market acceptance](#)

KEY POINTS

- RIST is specified by the Video Services Forum (VSF) in three profiles. Simple builds on RTP and RTCP; Main adds tunnelling, encryption and authentication.
- Loss recovery uses NACKs and selective retransmission; bonding over several links adds bandwidth or redundancy.
- Because only the protocol is standardised, performance on bad networks can still differ between vendors.

Background

The **Reliable Internet Stream Transport (RIST)** protocol was developed by the **Video Services Forum (VSF)** to provide an open standard for reliable streaming over the internet between products of different vendors. The VSF saw a "lack of compatibility between devices" because of proprietary solutions such as Zixi and, at the time, SRT. RIST is an ALF protocol that borrows heavily from RTP ([Chapter 2](#)).

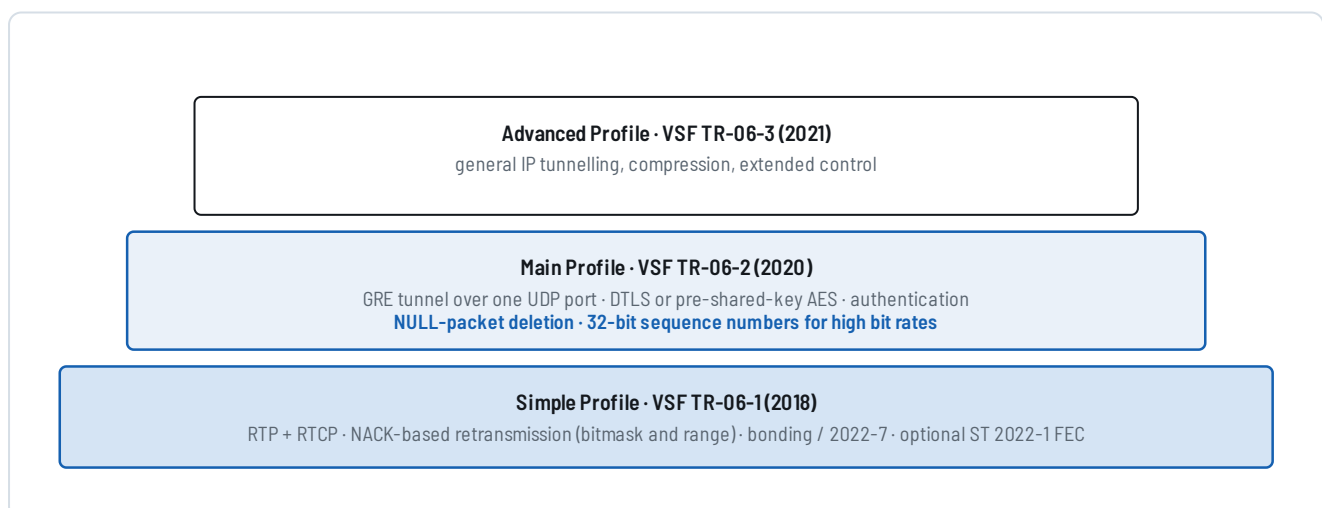


Figure 11.1 The three RIST profiles build on each other.

Simple Profile (TR-06-1, 2018)

The Simple Profile was published in October 2018. It uses **RTP** for the media and **RTCP** for control messages, mainly for error recovery.

- **NACK-based retransmission.** When the receiver detects a loss, it sends a negative acknowledgement; unlike TCP, there are no positive acknowledgements. The sender retransmits only the lost packets (*selective retransmission*) to save bandwidth.
- **Two NACK formats:** a bitmask for single packets and short bursts of up to 17 packets, and a range-based NACK for long bursts.
- The detection of loss and the timing of requests are left to the implementer.

As with every retransmission scheme, the receiver needs a buffer of one or more round-trip times plus reordering, and a packet may be requested several times (Chapter 8). RIST therefore suits links where ultra-low latency is not required or where the round-trip time is small.

Bonding

RIST supports **channel bonding** over several links such as fibre, Wi-Fi or LTE:

- **Load sharing:** a high-bit-rate RTP stream is divided between the links to combine their bandwidth.
- **Redundancy:** the stream is replicated on every link, as in ST 2022-7.

The receiver restores the order and removes duplicates.

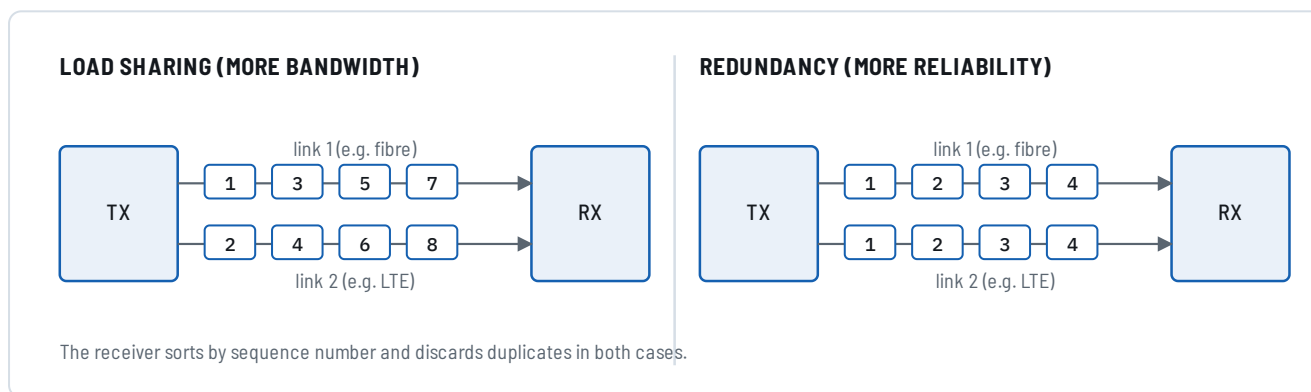


Figure 11.2 Load sharing and redundancy over two links.

Main Profile (TR-06-2, 2020)

The Main Profile, published in March 2020, adds:

- **Tunnelling and multiplexing.** All traffic between two RIST devices – several streams, RTCP and even other data – is carried through one GRE tunnel on a single UDP port. This simplifies firewalls and makes encryption of everything possible.
- **Encryption** with DTLS (RFC 6347) or with a pre-shared key (PSK) using AES.
- **Authentication** with certificates or, for simpler cases, passwords.
- **NULL-packet deletion.** MPEG-2 TS NULL packets are removed at the sender and their positions signalled in a bitmask so the receiver can re-insert them. This saves 3–5 % of bandwidth where NULL packets are present, for example after a statistical multiplexer.
- **High bit rate / high latency.** The sequence number is extended from 16 to 32 bits so that it does not wrap around within the buffer on fast, long links.
- Optional **SMPTE ST 2022-1 FEC** as a low-latency alternative to retransmission.

The Simple Profile has neither encryption nor authentication. Do not use it where these are required.

Advanced Profile (TR-06-3)

The third profile was announced in the handout as the *Enhanced Profile*, planned for 2021, with smart bandwidth optimisation, common session management, centralised *call home* control of many feeds, and hybrid internet/satellite operation. It was published by the VSF in 2021 as the **Advanced Profile** and widens RIST to general IP tunnelling and further functions. Check the current VSF documents for the exact feature set.

Market acceptance

When the handout was written, RIST devices were only starting to appear. RIST is now available in many encoders, decoders and gateways, and the open-source library *libRIST* makes implementation easy. With its broad vendor base, RIST is a good choice where multi-vendor interoperability, multicast and bonding matter. Implementations can still be improved by vendors, for example with bit-rate adaptation or failover logic.

SRT, RIST and Zixi compared – and the gateway to the viewer

No single protocol covers the whole chain. Contribution, distribution and delivery each use their own, and gateways translate between them.

IRENIS / BLANKOM · Technical guide · Ralf Riedel, Director Technical Sales & Engineering, IRENIS GmbH

In this chapter

[Side by side](#)

[Where they are used](#)

KEY POINTS

- SRT and RIST are open and interoperable; Zixi is proprietary with a strong ecosystem; RTP with FEC suits managed networks.
- All three internet protocols combine ARQ, optional FEC and encryption; their latency is set by the round-trip time.
- Gateways receive contribution streams and convert them into HLS/DASH for OTT viewers or into UDP/RTP multicast for IPTV networks.

Side by side

	RTP + FEC	SRT	RIST	ZIXI
Specification	IETF RFC 3550, SMPTE ST 2022-1/2	open source, IETF draft	VSF TR-06-1/-2/-3	proprietary
Transport	UDP	UDP	UDP (RTP)	UDP
Loss recovery	FEC	ARQ (NAK); FEC optional	ARQ (NACK); FEC optional	FEC + ARQ
Encryption	– (SRTP separately)	AES 128/192/256	DTLS or PSK AES (Main)	AES, DTLS
Bonding / dual path	ST 2022-7	socket groups (v1.5)	yes	yes
Multicast	yes	no (unicast)	yes	yes (private networks)
Typical network	managed	internet	internet and managed	internet and managed

Where they are used

RTP, Zixi, SRT and RIST are used mainly **in headends and for primary distribution** – from the studio or event to the operator. They are not delivered to phones and TVs. Before the viewer, a **gateway** converts them:

- to **HLS and DASH** for OTT delivery through CDNs ([Chapter 3](#), [Chapter 4](#)),

- to **UDP/RTP multicast** for IPTV networks in hotels, hospitals and residential complexes,
- to **SDI or HDMI** through IP decoders for monitoring and production.

The gateway can be hardware in a headend or software in a data centre or the cloud – a *virtualised headend*. Converting the transport protocol alone is fast; if the content must also change (codec, resolution, bit rate), the gateway transcodes as well.

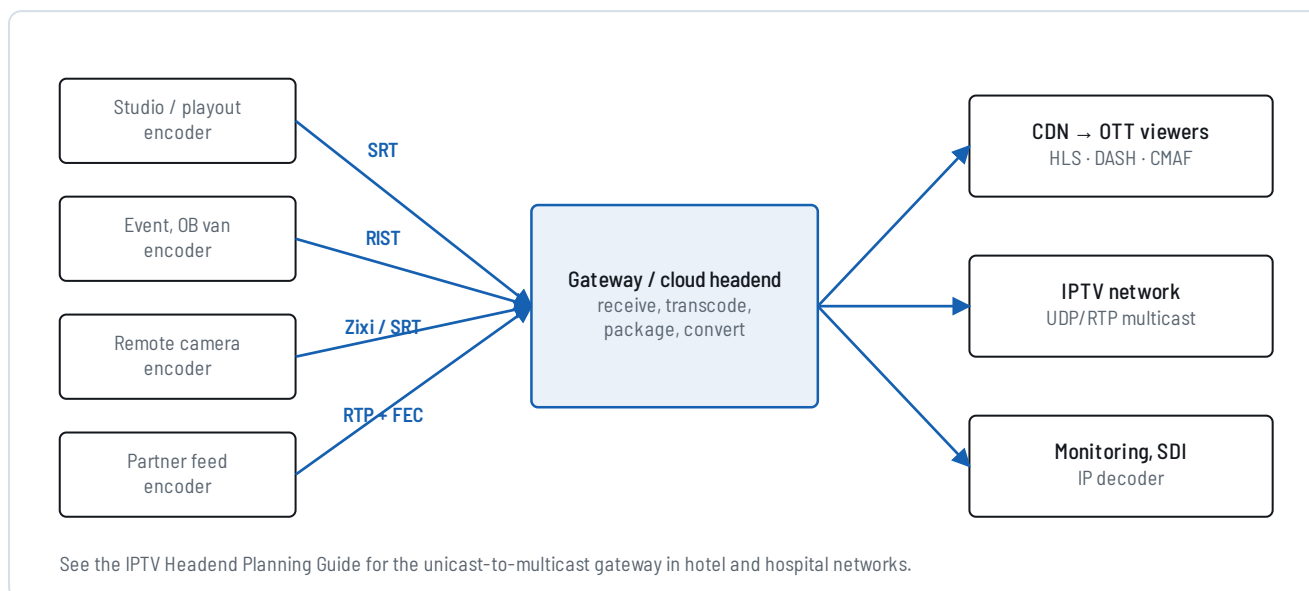


Figure 12.1 A gateway converts contribution streams for OTT viewers and IPTV networks.

UNICAST IN, MULTICAST OUT

For hospitality and institutional IPTV, the typical task is to receive a few SRT, RTMP or HLS streams and put them into the multicast network beside the satellite services. The **IPTV Headend Planning Guide** of this series describes this in its chapter on bringing in internet streams, with BLANKOM's BIG-1050X and BIG-1100X IP-to-IP gateways.

CHOOSING A CONTRIBUTION PROTOCOL

- ☐ Must devices from several vendors interoperate? Prefer SRT or RIST
- ☐ Is there a return channel? If not, use FEC (RTP + ST 2022-1, or RIST/SRT with FEC only)
- ☐ Latency budget versus round-trip time: ARQ needs $3-4 \times \text{RTT}$
- ☐ Encryption and authentication required? Use SRT or RIST Main Profile, not RIST Simple or plain RTP
- ☐ Two independent links available? Use bonding or ST 2022-7 for hitless protection
- ☐ What do the existing encoders, decoders and cloud services already support?

From SDI to IP in the facility

Production facilities were the last stronghold of dedicated video cabling. Rising resolutions, remote production and the cloud are moving them to IP as well.

IRENIS / BLANKOM · Technical guide · Ralf Riedel, Director Technical Sales & Engineering, IRENIS GmbH

In this chapter

[SDI, the traditional backbone](#)[Why IP now](#)[The building blocks](#)

KEY POINTS

- SDI carries uncompressed video on coax point to point. It is predictable and reliable, and much capital is invested in it.
- UHD and higher frame rates push SDI to 12 Gbit/s and beyond, and SDI cannot cross long distances or reach the cloud.
- IP must match SDI's timing and reliability. SMPTE ST 2022, ST 2110, PTP and ST 2022-7 provide this.

SDI, the traditional backbone

Video has been distributed to viewers over IP since the mid-1990s, but IP took much longer to arrive in live production. There the **Serial Digital Interface (SDI)** on coaxial cable – SD-SDI, HD-SDI, 3G-SDI, 12G-SDI – has carried digital video, audio and metadata. SDI is point to point, has fixed latency and is extremely predictable. Broadcasters invested heavily in SDI infrastructure, including expensive routers, and doubted that IP could guarantee the same reliability.

Why IP now

Three developments changed the picture:

1. **Bandwidth.** SDI moves raw, uncompressed video. With UHD and 8K resolutions and frame rates of 50/60 and even 100/120 fps, a single signal needs 12, 24 or 48 Gbit/s.
2. **Distance.** SDI is not a practical way to move content over long distances, for example from a venue to the studio or from a broadcaster to operators.
3. **The cloud.** Production and playout in data centres and clouds can only be reached over IP.

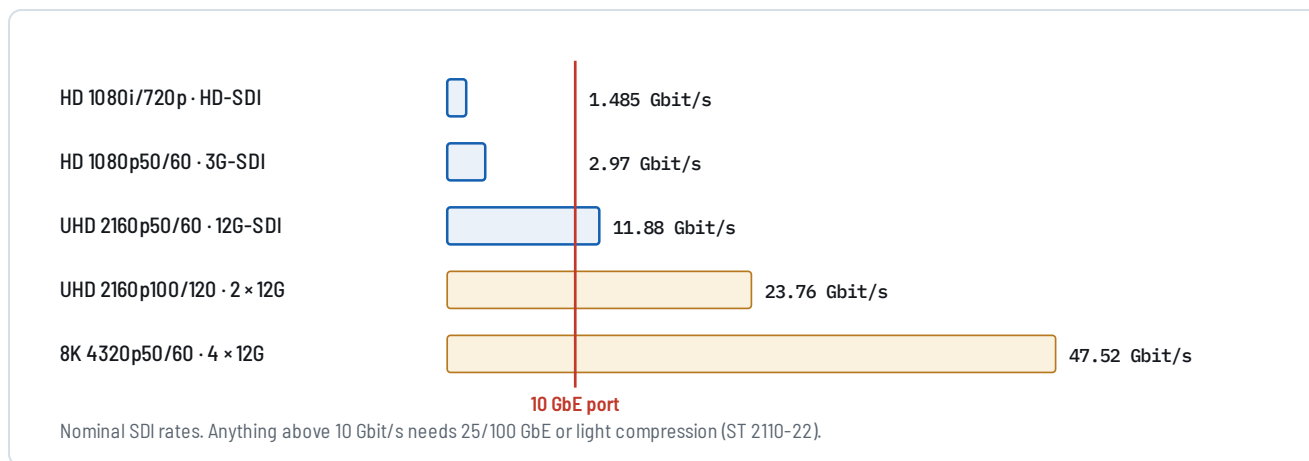


Figure 13.1 Uncompressed video bit rates grow beyond a 10 GbE port.

Ethernet switches with 25, 100 and 400 Gbit/s ports now offer the capacity, and commodity IP hardware replaces dedicated SDI routers. What IP has to add is SDI's discipline: precise timing, constant latency and protection against packet loss.

The building blocks

NEED	STANDARD	CHAPTER
Compressed streams between sites	SMPTE ST 2022-1/2	Chapter 14
SDI signals over IP as one bundle	SMPTE ST 2022-6	Chapter 14
Hitless protection	SMPTE ST 2022-7	Chapter 15
Separate video, audio and data streams	SMPTE ST 2110	Chapter 16
Common timing	PTP (IEEE 1588, SMPTE ST 2059)	Chapter 16
Discovery and connection management	AMWA NMOS	Chapter 16

CORRECTION TO THE 2022 HANDOUT

The handout linked the move to IP to CPUs coping with "SDI's data rate (400 Gbps)". SDI rates run from 1.5 to 12 Gbit/s per link; 400 Gbit/s is a modern Ethernet port speed. The decisive factor is network switching capacity, not CPU power.

SMPTE ST 2022

ST 2022 was the first step from dedicated video links to IP. Its first parts are the standard for compressed contribution over managed networks; part 6 put SDI itself on IP.

IRENIS / BLANKOM · Technical guide · Ralf Riedel, Director Technical Sales & Engineering, IRENIS GmbH

In this chapter Compressed streams: ST 2022-1 to -4 Why production needs uncompressed video
SDI over IP: ST 2022-6 and -5

KEY POINTS

- ST 2022-1 and -2 (2007) carry constant-bit-rate MPEG-2 transport streams over RTP with optional row/column FEC. They are widely deployed in primary distribution.
- ST 2022-6 (2012) carries a complete SDI signal – video, audio and data as one bundle – over RTP, with ST 2022-5 FEC.
- Uncompressed video needs frame-accurate switching and graphics insertion in production; compressed streams cannot provide that.

Compressed streams: ST 2022-1 to -4

The **Society of Motion Picture and Television Engineers (SMPTE)** published the first parts of ST 2022 in 2007:

- **ST 2022-2** carries constant-bit-rate (CBR) **MPEG-2 transport streams over IP**, encapsulated in RTP.
- **ST 2022-1** adds optional **FEC** with the row and column matrix described in [Chapter 8](#).
- **ST 2022-3** and **-4** extend this to variable-bit-rate streams.

These parts suit **primary distribution** from broadcasters to cable, satellite, IPTV and OTT operators. The video is moderately compressed – with JPEG 2000, MPEG-2, H.264/AVC or HEVC – at a few to several tens of Mbit/s. ST 2022-1/2 were commercially very successful and are widely deployed.

COMPRESSED · ST 2022-1/2/3/4

primary distribution, a few to tens of Mbit/s

IP · UDP · RTP

+ FEC stream(s) per ST 2022-1



payload: 1-7 TS packets of 188 bytes

- 2022-1** FEC for compressed streams
- 2022-2** CBR MPEG-2 TS over IP
- 2022-3/-4** VBR transport streams

UNCOMPRESSED · ST 2022-5/6/7

inside the facility, 1.5–12 Gbit/s

IP · UDP · RTP · HBRMT

1376 bytes of SDI data per packet

video + audio + ancillary as one bundle

- 2022-5** FEC for high bit rates
- 2022-6** SDI over IP
- 2022-7** seamless dual-path protection

Figure 14.1 The two families of SMPTE ST 2022.

Why production needs uncompressed video

Compressed streams are not practical for live production. Frame-accurate switching, inserting interstitials, logos and graphics – all of this is done on uncompressed pictures. Every decode and re-encode would also cost quality and latency.

SDI over IP: ST 2022-6 and -5

ST 2022-6 (2012) defines the carriage of **SDI signals over IP**: uncompressed video, audio and ancillary data such as captions and timing, as one bundle – the SDI signal is simply packetised.

- It uses RTP and a *high bit rate media transport* (HBRMT) header.
- Each packet carries **1376 bytes** of SDI data, so that it fits into one standard Ethernet frame together with the RTP, UDP and IP headers.
- ST 2022-6 made it possible to use IP routers and switches in production facilities.

Two protection mechanisms were defined:

- **ST 2022-5**: FEC for high-bit-rate media.
- **ST 2022-7**: seamless protection over two paths ([Chapter 15](#)).

WHERE ST 2022-6 STANDS TODAY

Because ST 2022-6 keeps video, audio and data bundled, it has largely been overtaken by ST 2110 ([Chapter 16](#)) in new facilities. Use it where existing equipment requires it or as a bridge from SDI.

Seamless protection with ST 2022-7

The simplest way to make an IP video link survive packet loss and even a network failure is to send it twice. ST 2022-7 standardises exactly that.

IRENIS / BLANKOM · Technical guide · Ralf Riedel, Director Technical Sales & Engineering, IRENIS GmbH

In this chapter

Principle

The receiver buffer

Strengths and costs

Where it is used

KEY POINTS

- The sender transmits identical RTP streams over two independent networks. The receiver takes each packet from whichever path delivers it.
- No retransmission, no acknowledgements and no failover software – switching is automatic and hitless.
- The price is double bandwidth and a receiver buffer that covers the delay difference between the paths.

Principle

SMPTE ST 2022-7 (*Seamless protection switching of RTP datagrams*) duplicates the stream at the sender and sends the copies over **two different paths**, often called the red and blue networks. If packets are lost on one path, the receiver reconstructs the stream from the packets of the other. Because both copies carry the same RTP sequence numbers, the receiver can merge them packet by packet.

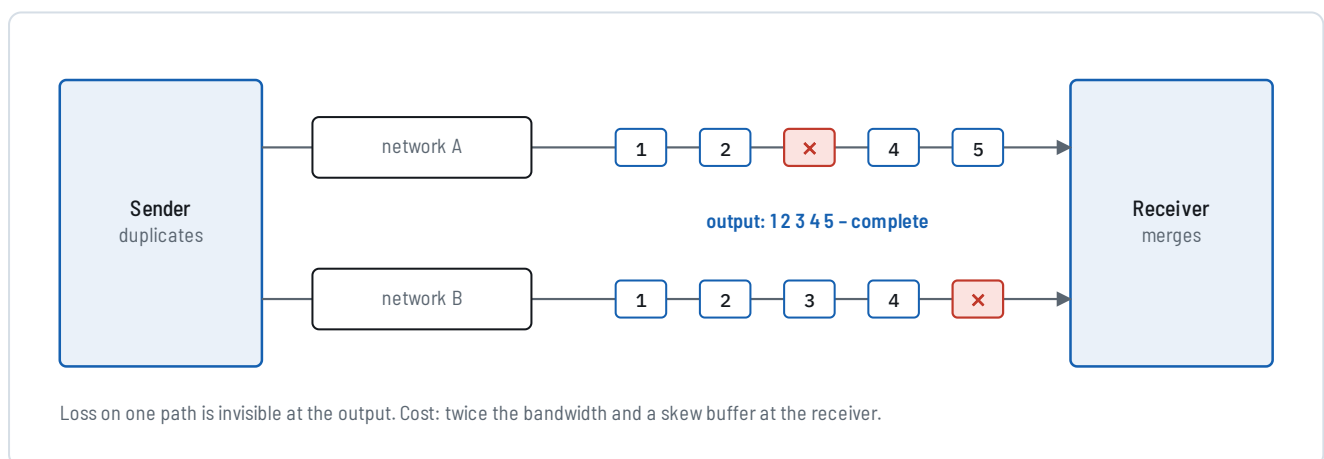


Figure 15.1 ST 2022-7: identical packets over two networks, merged at the receiver.

The receiver buffer

The two paths never have exactly the same delay, and each has its own jitter. The receiver therefore needs a buffer that covers the **path differential** (skew). The standard defines receiver classes by the skew they tolerate. The buffer size is a trade-off between latency and resilience: a larger buffer tolerates more different routes, but adds delay.

Strengths and costs

- **Strengths.** No retransmissions, no acknowledgements, no complex FEC. Failover is automatic, autonomous and seamless – there is no supervision software that has to detect a fault and act. Even the complete loss of one network is invisible at the output.
- **Costs.** Double bandwidth, two genuinely independent networks (separate switches, fibres and power), and more receiver ports.

Where it is used

ST 2022-7 protects ST 2022-6 and ST 2110 streams inside broadcast facilities and between facilities, where bandwidth is available. It is also used for compressed contribution over two managed networks, and similar dual-path modes exist in RIST, Zixi and SRT ([Chapter 12](#)).

PLANNING A 2022-7 SYSTEM

- ☐ Two physically separate networks, including separate switches and power feeds
- ☐ Every sender and receiver with two network interfaces
- ☐ Skew between the paths measured; receiver class and buffer chosen accordingly
- ☐ Both paths monitored – a silent failure of one path removes the protection

SMPTE ST 2110, PTP and NMOS

ST 2110 unbundles the SDI signal. Each essence travels as its own stream, and a common clock keeps them together. It is the standard for new IP production facilities.

IRENIS / BLANKOM · Technical guide · Ralf Riedel, Director Technical Sales & Engineering, IRENIS GmbH

In this chapter

From bundle to essences

The parts of ST 2110

Timing with PTP

Control with NMOS

A typical facility

KEY POINTS

- Video, audio and ancillary data are separate RTP streams (ST 2110-20/-22, -30/-31, -40), so each can be routed and processed on its own.
- All devices share one clock via the Precision Time Protocol (ST 2110-10, SMPTE ST 2059). Every switch must support PTP.
- AMWA NMOS lets devices register, be discovered and be connected automatically. ST 2022-7 provides protection.

From bundle to essences

ST 2022-6 carries audio, video and ancillary data as one bundle ([Chapter 14](#)). Many workflows need them separately – for example to replace the audio and captions of one language with another, without touching the video.

In 2015 the Video Services Forum published **TR-03**, which treated each medium as a separate stream, or *essence*, carried in RTP. This laid the groundwork for **SMPTE ST 2110**, whose first parts were published in 2017.

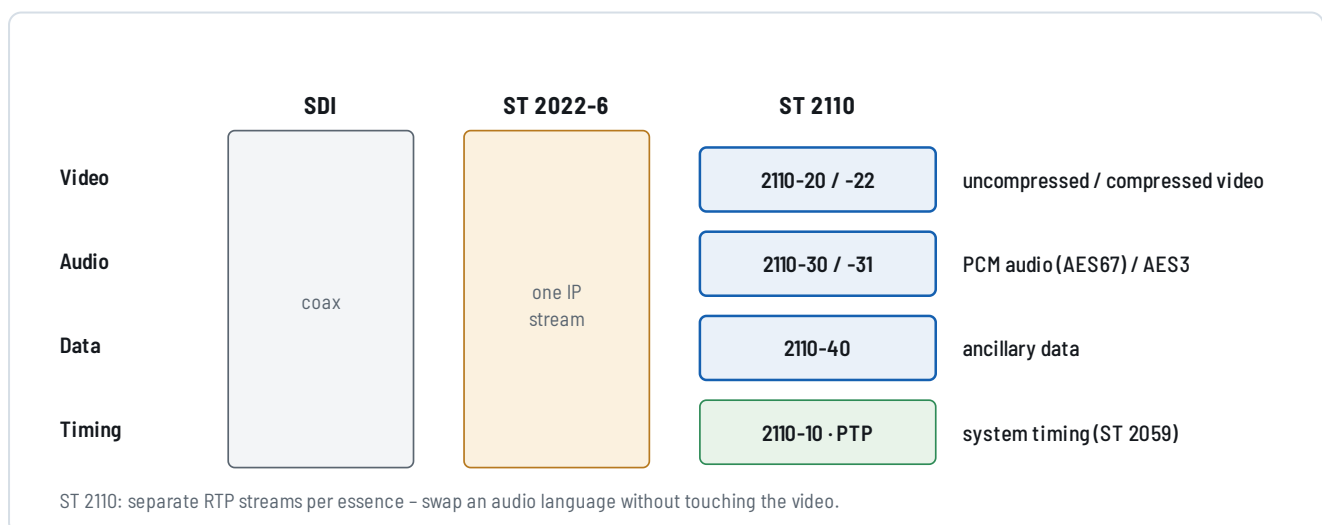


Figure 16.1 SDI and ST 2022-6 bundle the essences; ST 2110 separates them.

At a high level, the key difference between SDI and ST 2022-6 is the physical connection; the difference between ST 2022-6 and ST 2110 is the unbundling of the media.

The parts of ST 2110

PART	CONTENT
ST 2110-10	System timing and definitions (PTP)
ST 2110-20	Uncompressed active video
ST 2110-21	Traffic shaping and delivery timing of video
ST 2110-22	Constant-bit-rate compressed video, e.g. JPEG XS (2019)
ST 2110-30	PCM audio (based on AES67)
ST 2110-31	AES3 transparent transport
ST 2110-40	Ancillary data (captions, timecode and more)

ST 2110-22 carries lightly compressed video with codecs such as JPEG XS or TICO. JPEG XS typically compresses around 10:1 with a latency well below one millisecond, compared with tens of milliseconds for JPEG 2000. It brings UHD within reach of 10 and 25 GbE links.

For resilience, ST 2110 relies on **ST 2022-7** ([Chapter 15](#)). ST 2110 and ST 2022-7 are used inside broadcast facilities and between them, where bandwidth is not scarce.

Timing with PTP

Separate streams must stay in sync to the sample. ST 2110-10 uses the **Precision Time Protocol (PTP, IEEE 1588)** in the broadcast profile SMPTE ST 2059. A **grandmaster clock**, usually locked to GNSS, distributes time to every sender and receiver. The media switches must take part – as boundary clocks or transparent clocks – so that the network does not degrade the timing.

Control with NMOS

The **Networked Media Open Specifications (NMOS)** of the Advanced Media Workflow Association (AMWA) define how devices are managed:

- **IS-04** – registration and discovery: devices announce themselves and their streams.
- **IS-05** – connection management: a controller connects senders to receivers.

Further specifications cover events, audio channel mapping, security and more. NMOS greatly reduces manual set-up.

A typical facility

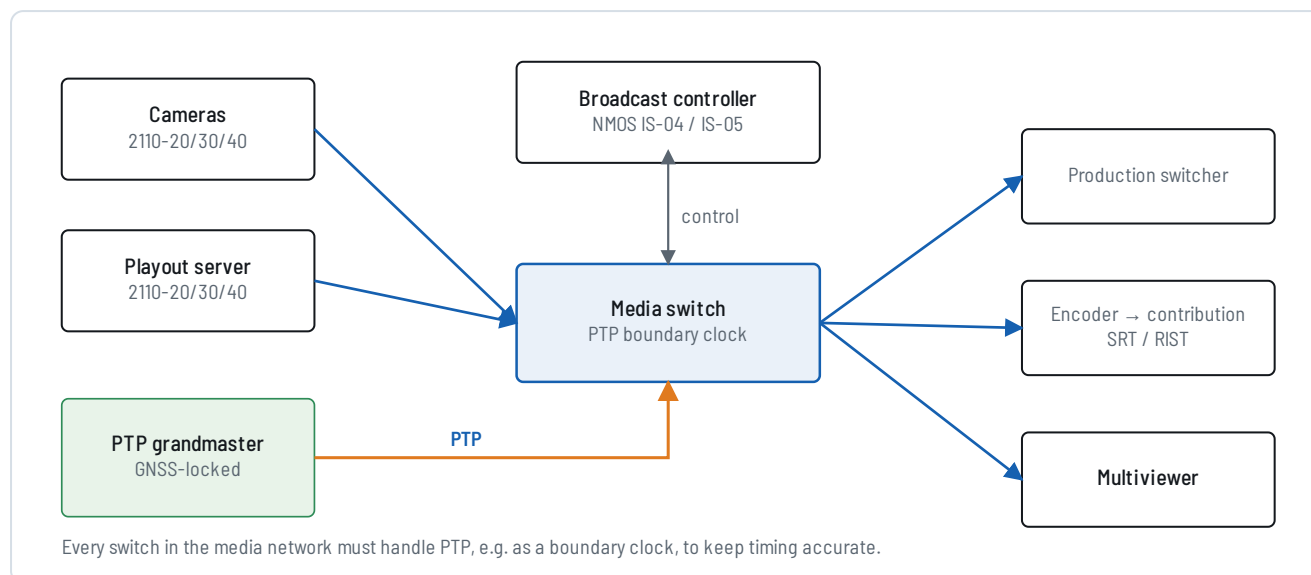


Figure 16.2 A typical ST 2110 facility with PTP timing and NMOS control.

A master clock supplies common timing to several sources – cameras, a playout server – each sending separate video, audio and data streams. A broadcast controller manages the sources and receivers via NMOS. The playout server can feed recorded content into the live production. Even the network switches must support PTP and re-clock it, to keep timing accurate across all nodes.

DEPLOYMENT

In 2022 the handout called ST 2110 "only recently developed". It is now the established standard for new IP production facilities and outside-broadcast vehicles, with broad multi-vendor interoperability tested at regular JT-NM events.

ROUTE, FLUTE and ATSC 3.0

Broadcast and multicast have no return channel, yet viewers' devices expect HTTP streaming. ROUTE bridges the two.

IRENIS / BLANKOM · Technical guide · Ralf Riedel, Director Technical Sales & Engineering, IRENIS GmbH

In this chapter [FLUTE](#) [ROUTE](#) [Deployment](#)

KEY POINTS

- ROUTE delivers application objects – files, DASH or HLS segments – over a unidirectional link, with FEC for reliability and support for low latency.
- It builds on FLUTE, the protocol for reliable file delivery over multicast.
- The receiver caches the objects and serves them to the player as a local HTTP server. ROUTE is part of ATSC 3.0 and of DVB-mABR.

FLUTE

File Delivery over Unidirectional Transport (FLUTE), RFC 6726) delivers files reliably over a multicast network to potentially huge numbers of receivers – for example a large software update to many devices at once. It uses IP multicast, runs over UDP, and adds reliability through **FEC** and congestion control through several multicast groups with different rates.

ROUTE

Real-time Object delivery over Unidirectional Transport (ROUTE) extends FLUTE for real-time media. It is specified in ATSC A/331 and was published by the IETF as **RFC 9223** in 2022 (the handout still referred to the 2020 draft).

- ROUTE delivers *application objects* – files, or DASH and HLS segments – including objects with real-time constraints.
- It adds **chunked delivery**, so CMAF chunks of low-latency DASH or HLS streams can be sent as they are produced ([Chapter 5](#)).
- A media-aware packetisation lets the sender optimise how objects are delivered.
- Security relies on underlying protocols such as IPsec.

On the receiver side, a ROUTE client caches the objects. The player retrieves them with ordinary HTTP requests: **the receiver acts as an HTTP server** for the application. Players built for OTT streaming therefore work unchanged.

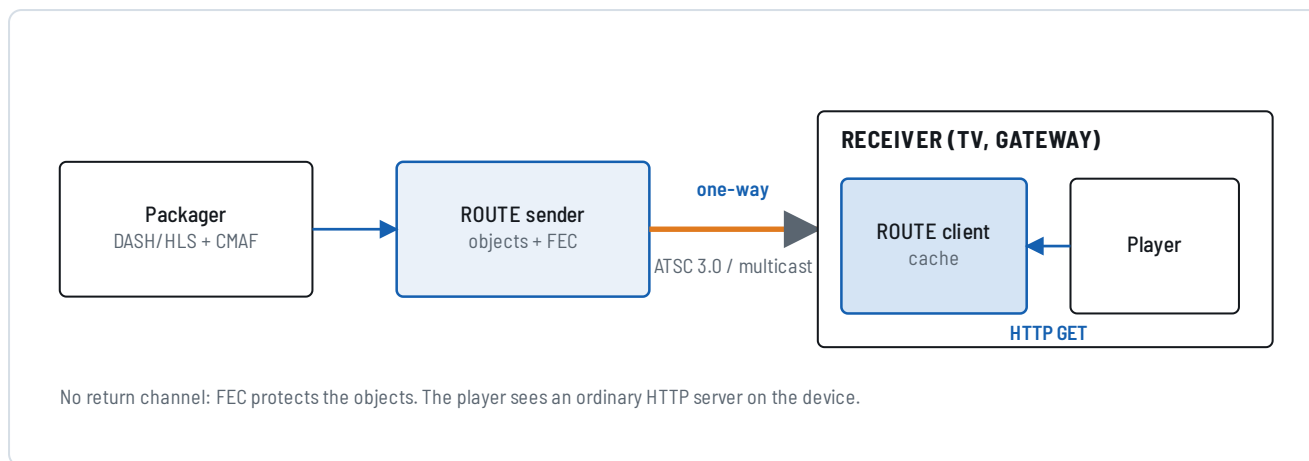


Figure 17.1 ROUTE: one-way object delivery with a local HTTP server in the receiver.

Deployment

- **ATSC 3.0**, the next-generation terrestrial broadcast standard in North America and South Korea, uses a profile of ROUTE. As ATSC 3.0 rolls out, ROUTE-capable TVs and gateways become common.
- **DVB-mABR** (*adaptive media streaming over IP multicast*) specifies ROUTE as one of its delivery protocols, for multicast ABR in operator networks.
- Its multicast- and broadcast-friendly nature makes ROUTE attractive for over-the-air broadcast and for multicast over managed networks.

AND IN HOSPITALITY IPTV?

Hotel and hospital IPTV networks carry each TV service as a UDP/RTP multicast transport stream, which every IPTV set-top box and hospitality TV understands. ROUTE and mABR matter where content is delivered to OTT-style players over multicast, but they are not needed for classic hospitality IPTV.

Choosing the right protocol

Requirements vary widely from one use case to the next, and so does the best protocol. This chapter sums up the recommendations of the guide.

IRENIS / BLANKOM · Technical guide · Ralf Riedel, Director Technical Sales & Engineering, IRENIS GmbH

In this chapter

Recommendations by use case

End-to-end systems need gateways

KEY POINTS

- Viewers – DASH and HLS with CMAF, low-latency variants for live. Headend and studio – ST 2110. Primary distribution – SRT or RIST, or RTP with FEC on managed networks.
- Hospitality and institutional IPTV stays with UDP/RTP multicast; gateways bring in SRT, RTMP or HLS sources.
- End-to-end systems always combine several protocols, so they need gateways that convert – with or without transcoding.

Recommendations by use case

USE CASE	FIRST CHOICE	NOTES
Inside the facility	SMPTE ST 2110 + ST 2022-7	ST 2022-6 only for existing equipment
Contribution over the internet	SRT · RIST	Zixi where the ecosystem is already in use
Managed private network	RTP + FEC (ST 2022-1/2)	add ST 2022-7 for two paths
Hotel / hospital IPTV	UDP/RTP multicast	gateway converts SRT/HLS input
OTT to phones and TVs	HLS + DASH with CMAF	low-latency variants for live
Broadcast / mABR	ROUTE	ATSC 3.0, DVB-mABR

Figure 18.1 First-choice protocols by use case.

USE CASE	RECOMMENDATION	NOTES
Distribution to viewers (OTT)	DASH and HLS with the CMAF container and byte-range addressing, to minimise storage and network load (Chapter 4)	For live streams use the low-latency variants with chunked encoding and transfer (Chapter 5). HTTP/3 over QUIC improves delivery on mobile networks (Chapter 6).
Broadcast over the air, ATSC 3.0	ROUTE (Chapter 17)	Required by ATSC 3.0; also used in DVB-mABR
Carriage inside the facility	SMPTE ST 2110 with ST 2022-7 (Chapter 16 , Chapter 15)	If you still use SDI, go straight to ST 2110. Use ST 2022-6 only for existing equipment.
Primary distribution to affiliates and operators	RIST or SRT (Chapter 10 , Chapter 11)	Zixi where its ecosystem is in place (Chapter 9). RTP is an option on managed networks, protected with FEC (ST 2022-1) and encryption.
Hospitality and institutional IPTV	UDP/RTP multicast	Satellite and terrestrial services from the headend; internet sources via a unicast-to-multicast gateway (Chapter 12)
Ingest to streaming platforms	RTMP(S) where required; SRT where supported (Chapter 7)	

UPDATED SINCE 2022

The handout recommended RIST "although support is limited because it is so new" and SRT or Zixi for immediate deployment. Today both SRT and RIST are widely available; choose by the equipment on both ends and the features you need ([Chapter 12](#)).

End-to-end systems need gateways

A complete system – headend, primary distribution and delivery to viewers – always uses several protocols. Somewhere the transport protocol must be converted, with or without transcoding the content. That is the job of **gateway** products, in hardware or software.

BEFORE YOU DECIDE

- ☐ Where does the signal come from, and in which format (SDI, SRT, RTMP, HLS, satellite)?
- ☐ Where must it go: studio, operator, OTT viewers, an IPTV network?
- ☐ Latency budget per stage, and the round-trip time of each link
- ☐ Managed network or public internet? Return channel available?
- ☐ Encryption, authentication and DRM requirements
- ☐ Redundancy: second path, second encoder, ST 2022-7
- ☐ Which protocols do the existing encoders, decoders, gateways and players support?

Before you make a final decision, send us your requirements. Our engineers analyse your use case and reply in writing with a proposal.

Streaming in hardware – FPGA IP cores

Headend encoders, gateways and receivers process dozens of streams in real time. Much of that work is done by dedicated logic in FPGAs, built from reusable IP cores.

IRENIS / BLANKOM · Technical guide · Ralf Riedel, Director Technical Sales & Engineering, IRENIS GmbH

In this chapter Why hardware Network layer Streaming layer Stream announcements: SAP/SDP
IP over DVB: ULE (RFC 4326)

KEY POINTS

- An FPGA streaming design consists of a UDP/IP stack with its own Ethernet MAC, an MDIO interface for the PHY, and RTP transmit and receive cores.
- The receive side must de-jitter bursty UDP/RTP streams and restore PCR accuracy to TR 101 290 (± 500 ns) at its DVB-SPI output.
- SAP/SDP announces multicast streams on the LAN; ULE (RFC 4326) carries IP packets inside DVB transport streams.

Why hardware

A software streamer on a general-purpose CPU is flexible, but a headend device that sends or receives dozens of transport streams with exact timing benefits from dedicated logic. FPGA and SoC vendors and third-party suppliers offer **IP cores** – ready-made function blocks – for the network and streaming functions. The 2022 handout included an overview of such cores from a third-party supplier; this chapter summarises the typical building blocks and their features.

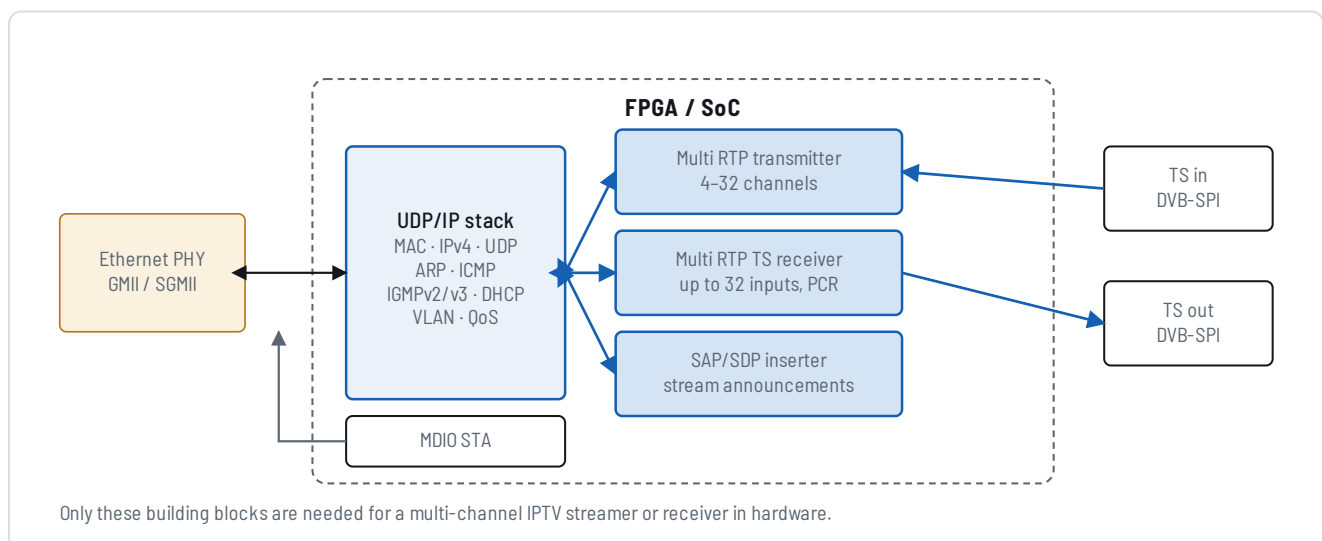


Figure 19.1 An IPTV streamer or receiver built from FPGA IP cores.

Network layer

UDP/IP stack

- Sends and receives UDP packets on an Ethernet network, with its own Ethernet MAC (Ethernet II frames).
- PHY interfaces GMII, RGMII or SGMII; 10/100/1000 Mbit/s.
- Protocols: IPv4, UDP, ARP, ICMP, IGMPv2/v3, DHCP client, VLAN tagging (IEEE 802.1Q).
- Reception of up to 32 multicast groups; point-to-point UDP transmit channels.
- Jumbo frames up to 9 kB and raw frames.
- Per transmit channel: DiffServ (QoS) marking and a programmable IP time-to-live.

MDIO management interface (STA)

- Companion core to the UDP/IP stack for controlling the Ethernet PHY.
- Reads and writes the PHY registers according to IEEE 802.3 Clause 22; up to 32 devices with 32 registers each.
- Works with any MDIO device compliant with Clause 22.

Streaming layer

CORE	FUNCTION
RTP transmitter	Sends RTP packets (RTPv2) over a routed network, e.g. IPTV-compliant streams. Option: 2D FEC to SMPTE ST 2022-1 for MPEG-TS (payload type 33).
Multi RTP transmitter	Sends 4 to 32 independent MPEG-TS RTP channels, with 1 to 7 TS packets per UDP/RTP packet.
RTP receiver	Receives an RTP stream with MPEG-TS payload; compensates the 90 kHz timestamp clock (± 100 ppm) and reorders packets after loss or disorder. Also accepts plain UDP.
Multi RTP TS receiver	Up to 32 UDP or RTP inputs, type detected automatically; any packing of 1 to 7 TS packets. Produces DVB-SPI outputs from bursty or jittery streams (see below).

The **Multi RTP TS receiver** shows what de-jittering involves:

- For CBR inputs it restores the **PCR accuracy required by ETSI TR 101 290 (± 500 ns)** with a programmable latency.
- A bypass mode handles VBR streams.
- Bit-rate variations of up to 32 kbit/s are followed without a discontinuity at the output; above that, the PCR calculation restarts.

Combined, these cores form complete **IPTV transmitters** (UDP/IP stack + MDIO + multi RTP transmitter) that send MPEG-TS over Ethernet, and **IPTV receivers** (UDP/IP stack + MDIO + multi RTP TS receiver) that output one or several CBR transport streams on DVB-SPI.

Stream announcements: SAP/SDP

A **SAP/SDP inserter** periodically sends Session Announcement Protocol (SAP, RFC 2974) packets with Session Description Protocol (SDP) content, for example to multicast address 239.255.255.255 on UDP port 9875. Receivers that support SAP/SDP learn from them everything they need to join the multicast streams on the network. The core generates the SAP header, encapsulates SDP descriptions written by the CPU, and handles up to 128 channels.

IP over DVB: ULE (RFC 4326)

Unidirectional Lightweight Encapsulation (ULE) carries IP packets inside MPEG-2 transport streams, for data over DVB-S, DVB-C or DVB-T links.

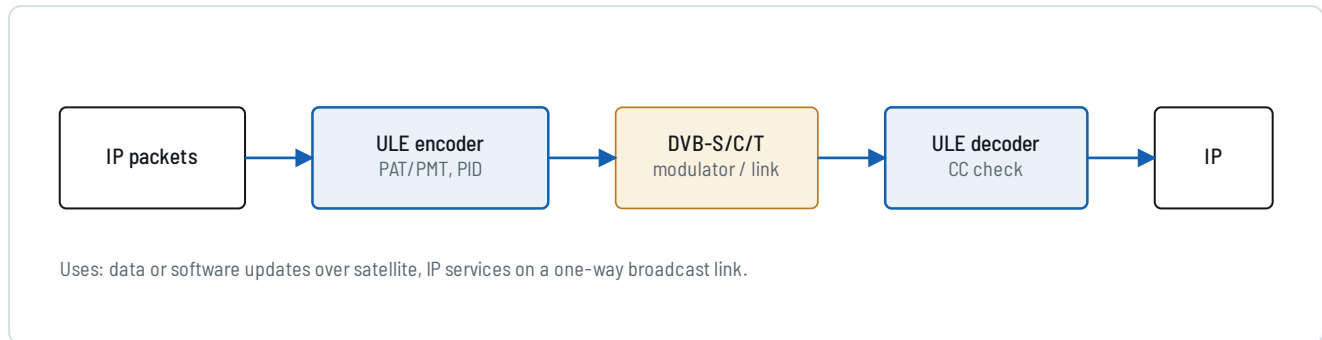


Figure 19.2 IP over DVB with Unidirectional Lightweight Encapsulation.

- The **encoder** inserts IP packets into a user transport stream, with or without padding completion, outputs 188 or 204-byte CBR packets (with RS 204 coding), generates PAT and PMT with configurable PIDs and PSI repetition, and has a configurable output rate.
- The **decoder** extracts IP packets, analyses PAT and PMT automatically to find the payload, and raises alarms for continuity-counter errors and length faults.

SOURCE

The feature lists in this chapter summarise data sheets of third-party IP cores reproduced in the 2022 handout. They describe typical capabilities, not BLANKOM products.

Glossary of abbreviations

Streaming joins broadcast, IT and web vocabulary. These are the abbreviations used in this guide.

IRENIS / BLANKOM · Technical guide · Ralf Riedel, Director Technical Sales & Engineering, IRENIS GmbH

KEY POINTS

- Chapter references show where each term is explained.
- Protocol names are often acronyms; QUIC is not any more – it is simply the name.
- The IPTV Headend Planning Guide has a glossary for satellite, DVB and multicast terms.

TERM	MEANING
ABR	Adaptive bit rate (Chapter 3)
AES	Advanced Encryption Standard
ALF	Application layer framing (Chapter 1)
AMWA	Advanced Media Workflow Association (Chapter 16)
ARQ	Automatic repeat request (Chapter 8)
ATSC	Advanced Television Systems Committee (Chapter 17)
B2B	Business to business
CBR / VBR	Constant / variable bit rate
CDN	Content delivery network (Chapter 3)
CENC	Common encryption (Chapter 4)
CMAF	Common Media Application Format (Chapter 4)
DASH	Dynamic Adaptive Streaming over HTTP (Chapter 4)
DRM	Digital rights management
DTLS	Datagram Transport Layer Security
DVB-SPI	DVB synchronous parallel interface (Chapter 19)
FEC	Forward error correction (Chapter 8)
FLUTE	File Delivery over Unidirectional Transport (Chapter 17)
fMP4	Fragmented MP4
FPGA	Field-programmable gate array (Chapter 19)
GNSS	Global navigation satellite system
GRE	Generic Routing Encapsulation (Chapter 11)
HBRMT	High bit rate media transport (Chapter 14)
HDS	HTTP Dynamic Streaming (Adobe) (Chapter 4)
HLS / LL-HLS	HTTP Live Streaming / Low-Latency HLS (Chapter 4 , Chapter 5)
HTTP	Hypertext Transfer Protocol
IGMP	Internet Group Management Protocol
IP	Internet Protocol
MDIO	Management data input/output (Chapter 19)
mABR	Multicast adaptive bit rate (Chapter 17)
MMT	MPEG Media Transport
MPD	Media Presentation Description (DASH manifest)
MVPD	Multichannel video programming distributor
NACK / NAK	Negative acknowledgement (Chapter 8)

TERM	MEANING
NMOS	Networked Media Open Specifications (Chapter 16)
OTT	Over the top: delivery over the open internet
PCR	Program clock reference (Chapter 19)
PSK	Pre-shared key (Chapter 11)
PTP	Precision Time Protocol, IEEE 1588 (Chapter 16)
QUIC	Name of the protocol, no longer an acronym (Chapter 6)
RIST	Reliable Internet Stream Transport (Chapter 11)
ROUTE	Real-time Object delivery over Unidirectional Transport (Chapter 17)
RTCP	RTP Control Protocol (Chapter 2)
RTMP(S)	Real-Time Messaging Protocol (over TLS) (Chapter 7)
RTP	Real-time Transport Protocol (Chapter 2)
RTSP	Real Time Streaming Protocol
RTT	Round-trip time
SAP / SDP	Session Announcement / Description Protocol (Chapter 19)
SDI	Serial Digital Interface (Chapter 13)
SMPTE	Society of Motion Picture and Television Engineers
SRT	Secure Reliable Transport (Chapter 10)
TCP	Transmission Control Protocol
TLS	Transport Layer Security
TS	MPEG-2 transport stream
UDP	User Datagram Protocol
UDT	UDP-based Data Transfer protocol (Chapter 10)
ULE	Unidirectional Lightweight Encapsulation, RFC 4326 (Chapter 19)
VSF	Video Services Forum (Chapter 11)

Sources and references

This guide rewrites and updates the 2022 handout. These are its sources and the current versions of the standards it describes.

IRENIS / BLANKOM · Technical guide · Ralf Riedel, Director Technical Sales & Engineering, IRENIS GmbH

In this chapter Sources of the 2022 handout Foundations HTTP streaming and QUIC
Contribution protocols Broadcast facility Broadcast and multicast delivery Implementation

KEY POINTS

- The protocol chapters are based on the ATEME white paper "Video over IP Protocols", rewritten in our own words and updated to 2026.
- Standards are listed in their current version; several were drafts when the handout was written.
- Always check the latest edition of a standard before you specify equipment.

Sources of the 2022 handout

- ATEME: *Video over IP Protocols*, white paper, about 2021. Basis of the protocol chapters 1 to 18; their structure and recommendations are summarised and updated here.
- IP-core data sheets of a third-party FPGA IP supplier (UDP/IP stack, MDIO, RTP transmitter and receiver, SAP/SDP, ULE). Basis of [Chapter 19](#).

Foundations

- [ALF] Clark, D. D.; Tennenhouse, D. L.: *Architectural considerations for a new generation of protocols*. ACM SIGCOMM Computer Communication Review 20 (4), September 1990.
- RFC 768 – User Datagram Protocol, 1980.
- RFC 3550 – RTP: A Transport Protocol for Real-Time Applications, 2003 (obsoletes RFC 1889, 1996).
- RFC 2250 – RTP Payload Format for MPEG1/MPEG2 Video, 1998.
- RFC 7826 – Real-Time Streaming Protocol 2.0, 2016.

HTTP streaming and QUIC

- RFC 9110 – HTTP Semantics, 2022 (obsoletes RFC 2616 and RFC 2068).
- RFC 8216 – HTTP Live Streaming, 2017; HLS second edition with Low-Latency HLS (Apple).
- ISO/IEC 23009-1 – Dynamic adaptive streaming over HTTP (DASH), Part 1, first edition 2012, current edition.
- ISO/IEC 23000-19 – Common Media Application Format (CMAF), 2018.
- RFC 9000 – QUIC: A UDP-Based Multiplexed and Secure Transport, 2021; RFC 9001 (TLS for QUIC), RFC 9002 (loss detection).
- RFC 9114 – HTTP/3, 2022. RFC 9369 – QUIC Version 2, 2023.

Contribution protocols

- The SRT Protocol, IETF Internet-Draft draft-sharabayko-srt (first published 2020); SRT open-source project and SRT Alliance.
- VSF TR-06-1 – RIST Simple Profile, October 2018.
- VSF TR-06-2 – RIST Main Profile, March 2020.
- VSF TR-06-3 – RIST Advanced Profile, 2021.
- RFC 6347 – Datagram Transport Layer Security 1.2, 2012; RFC 9147 – DTLS 1.3, 2022.

Broadcast facility

- SMPTE ST 2022-1 – Forward error correction for real-time video/audio transport over IP networks, 2007.
- SMPTE ST 2022-2 – Unidirectional transport of constant bit rate MPEG-2 transport streams on IP networks, 2007.
- SMPTE ST 2022-5 – FEC for high bit rate media signals over IP networks (HBRMT), 2012.
- SMPTE ST 2022-6 – Transport of high bit rate media signals over IP networks (HBRMT), 2012.
- SMPTE ST 2022-7 – Seamless protection switching of RTP datagrams, current edition 2019.
- VSF TR-03 – Transport of uncompressed elementary stream media over IP, November 2015.
- SMPTE ST 2110 suite (-10, -20, -21, -22, -30, -31, -40), from 2017.
- SMPTE ST 2059-1/-2 – PTP profile for broadcast; IEEE 1588 – Precision Time Protocol.
- AMWA NMOS IS-04 and IS-05.

Broadcast and multicast delivery

- RFC 6726 – FLUTE, 2012 (obsoletes RFC 3926).
- RFC 9223 – ROUTE, 2022; ATSC A/331 – Signaling, Delivery, Synchronization and Error Protection.
- DVB A176 – Adaptive media streaming over IP multicast (DVB-mABR).

Implementation

- RFC 4326 – Unidirectional Lightweight Encapsulation (ULE) for transmission of IP datagrams over an MPEG-2 transport stream, 2005.
- RFC 2974 – Session Announcement Protocol; RFC 8866 – SDP, 2021.
- IEEE 802.3, Clause 22 – MDIO.
- ETSI TR 101 290 – Measurement guidelines for DVB systems.

More questions?

Please contact our international team at IRENIS – BLANKOM. With customers in every time zone, we prefer written enquiries: tell us the products, quantities and application, and our engineers will reply with a technical assessment and quotation.

Request a quote

[irenis-blankom.com/request-a-quote/](https://www.irenis-blankom.com/request-a-quote/)

Write to us

info@irenis-blankom.com

contact@blankom.de

www.irenis-blankom.com

www.blankom.de

IRENIS GmbH

Hauptstrasse 29

31171 Nordstemmen

Germany

E-mail: info@irenis-blankom.com · contact@blankom.de · Web: www.irenis-blankom.com · www.blankom.de

© 2022–2026 BLANKOM, IRENIS GmbH. All rights reserved.