

Draft Recommendation for  
Space Data System Standards

|  |
| --- |
| Real Time Protocol over Delay Tolerant Networking for Video Applications |

PROPOSED Draft Recommended Standard

CCSDS 000.0-W-0

White Book D2

February 2018

AUTHORITY

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
|  | Issue: | White Book, Issue 0 |  |
|  | Date: | February 2018 |  |
|  | Location: | Not Applicable |  |
|  | | | |

**(WHEN THIS RECOMMENDED STANDARD IS FINALIZED, IT WILL CONTAIN THE FOLLOWING STATEMENT OF AUTHORITY:)**

This document has been approved for publication by the Management Council of the Consultative Committee for Space Data Systems (CCSDS) and represents the consensus technical agreement of the participating CCSDS Member Agencies. The procedure for review and authorization of CCSDS documents is detailed in *Organization and Processes for the Consultative Committee for Space Data Systems* (CCSDS A02.1-Y-4), and the record of Agency participation in the authorization of this document can be obtained from the CCSDS Secretariat at the address below.

This document is published and maintained by:

CCSDS Secretariat

Space Communications and Navigation Office, 7L70

Space Operations Mission Directorate

NASA Headquarters

Washington, DC 20546-0001, USA

FOREWORD

[Foreword text specific to this document goes here. The text below is boilerplate.]

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. CCSDS shall not be held responsible for identifying any or all such patent rights.

Through the process of normal evolution, it is expected that expansion, deletion, or modification of this document may occur. This Recommended Standard is therefore subject to CCSDS document management and change control procedures, which are defined in *Organization and Processes for the Consultative Committee for Space Data Systems* (CCSDS A02.1-Y-4). Current versions of CCSDS documents are maintained at the CCSDS Web site:

http://www.ccsds.org/

Questions relating to the contents or status of this document should be addressed to the CCSDS Secretariat at the address indicated on page i.

At time of publication, the active Member and Observer Agencies of the CCSDS were:

Member Agencies

* Agenzia Spaziale Italiana (ASI)/Italy.
* Canadian Space Agency (CSA)/Canada.
* Centre National d’Etudes Spatiales (CNES)/France.
* China National Space Administration (CNSA)/People’s Republic of China.
* Deutsches Zentrum für Luft- und Raumfahrt (DLR)/Germany.
* European Space Agency (ESA)/Europe.
* Federal Space Agency (FSA)/Russian Federation.
* Instituto Nacional de Pesquisas Espaciais (INPE)/Brazil.
* Japan Aerospace Exploration Agency (JAXA)/Japan.
* National Aeronautics and Space Administration (NASA)/USA.
* UK Space Agency/United Kingdom.

Observer Agencies

* Austrian Space Agency (ASA)/Austria.
* Belgian Federal Science Policy Office (BFSPO)/Belgium.
* Central Research Institute of Machine Building (TsNIIMash)/Russian Federation.
* China Satellite Launch and Tracking Control General, Beijing Institute of Tracking and Telecommunications Technology (CLTC/BITTT)/China.
* Chinese Academy of Sciences (CAS)/China.
* Chinese Academy of Space Technology (CAST)/China.
* Commonwealth Scientific and Industrial Research Organization (CSIRO)/Australia.
* Danish National Space Center (DNSC)/Denmark.
* Departamento de Ciência e Tecnologia Aeroespacial (DCTA)/Brazil.
* European Organization for the Exploitation of Meteorological Satellites (EUMETSAT)/Europe.
* European Telecommunications Satellite Organization (EUTELSAT)/Europe.
* Geo-Informatics and Space Technology Development Agency (GISTDA)/Thailand.
* Hellenic National Space Committee (HNSC)/Greece.
* Indian Space Research Organization (ISRO)/India.
* Institute of Space Research (IKI)/Russian Federation.
* KFKI Research Institute for Particle & Nuclear Physics (KFKI)/Hungary.
* Korea Aerospace Research Institute (KARI)/Korea.
* Ministry of Communications (MOC)/Israel.
* National Institute of Information and Communications Technology (NICT)/Japan.
* National Oceanic and Atmospheric Administration (NOAA)/USA.
* National Space Agency of the Republic of Kazakhstan (NSARK)/Kazakhstan.
* National Space Organization (NSPO)/Chinese Taipei.
* Naval Center for Space Technology (NCST)/USA.
* Scientific and Technological Research Council of Turkey (TUBITAK)/Turkey.
* South African National Space Agency (SANSA)/Republic of South Africa.
* Space and Upper Atmosphere Research Commission (SUPARCO)/Pakistan.
* Swedish Space Corporation (SSC)/Sweden.
* Swiss Space Office (SSO)/Switzerland.
* United States Geological Survey (USGS)/USA.

PREFACE

This document is a draft CCSDS Recommended Standard. Its ‘White Book’ status indicates that its contents are not stable, and several iterations resulting in substantial technical changes are likely to occur before it is considered to be sufficiently mature to be released for review by the CCSDS Agencies.

Implementers are cautioned **not** to fabricate any final equipment in accordance with this document’s technical content.

Recipients of this draft are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

DOCUMENT CONTROL

|  |  |  |  |
| --- | --- | --- | --- |
| **Document** | **Title and Issue** | **Date** | **Status** |
| CCSDS 000.0-W-0 | [Document Title], Proposed Draft Recommended Standard, Issue 0 | November 2014 | Current draft |
|  |  |  |  |
|  |  |  |  |

CONTENTS

Section Page

[1 Introduction 1-1](#_Toc527638447)

[1.1 Purpose And Scope 1-1](#_Toc527638448)

[1.2 Applicability 1-1](#_Toc527638449)

[1.3 Rationale 1-1](#_Toc527638450)

[2 Overview 2-1](#_Toc527638451)

[2.1 Assumptions 2-1](#_Toc527638452)

[2.2 Flow diagram – RTP in larger picture 2-2](#_Toc527638453)

[2.3 Technical Overview 2-1](#_Toc527638454)

[2.4 RTP Over DTN 2-8](#_Toc527638455)

[2.5 Legacy Compatibility 2-10](#_Toc527638456)

[3 RTP in Space-Based Links (NORMATIVE) 3-11](#_Toc527638457)

[3.1 Overview 3-11](#_Toc527638458)

[3.2 RTP PDU Formatting 3-11](#_Toc527638459)

[3.3 RTP Concatenation 3-11](#_Toc527638467)

[3.4 Receiver-Side Fragmentation 3-12](#_Toc527638468)

[3.5 DTN Transmission of RTP Packets 3-13](#_Toc527638469)

[3.6 Signaling Data (SDP Description & RTCP) 3-13](#_Toc527638470)

[~~4~~ ~~Conceptual: Ground Systems & Display Systems~~ 4-17](#_Toc527638471)

[ANNEX A Implementation Conformance Statement (ICS) Proforma (normative) A-1](#_Toc527638472)

[ANNEX B Security, SANA, and Patent Considerations (Informative) B-1](#_Toc527638473)

Figure

[Figure 2.1 Flow Diagram - -On Board Assets 2-3](#_Toc527638474)

[Figure 2.2 Flow Diagram - Ground Segment 2-4](#_Toc527638475)

[Figure 2.3 RTP Packet Header 2-1](#_Toc527638476)

[Figure 2.4 RTP Relationships 2-2](#_Toc527638477)

[Figure 2.5 Spacecraft Network Design 2-3](#_Toc527638478)

[Figure 2.6 RTCP Overview 2-4](#_Toc527638479)

[Figure 2.7 RTCP Sender Report Message 2-5](#_Toc527638480)

[Figure 2.8 Example of SDP Data. 2-7](#_Toc527638481)

[Figure 2.9 SDP Translation Between DTN and IP Domains 2-7](#_Toc527638482)

[Figure 2.10 Theoretical IP-To-RTP System (Credit: DLR) 2-9](#_Toc527638483)

[Figure 3.1 RTP Concatenation - Overview 3-12](#_Toc527638484)

[Figure 3.2 RTCP & Video Synchronization 3-16](#_Toc527638485)

Table

No table of contents entries found.

# Introduction

Motion Imagery/Video Transmission over DTN is not simple, due to the nature of video over IP, which generally requires consistent data flow into a video decoder. Up to now, the use of MPEG Transport Streams (MPEG-TS) has been the default standard to encapsulate and format video transmission over IP. However, MPEG-TS does not work well in situations with frequent interruptions or excessive latency and is not very flexible with regard to packet size. DLR has been testing Real Time Protocol (RTP) over DTN for video and has found it is much more forgiving of interruptions and long latencies in the network. The fact that RTP packet size is arbitrary makes it much more flexible for video over DTN.

## Purpose And Scope

This document provides an overview and proposed methods for transmission of video over DTN using RTP. As more deep space missions will be utilizing DTN for data transmission and the use of video becomes even more prevalent, a standard method of formatting encoded video streams for transmission is required. Standardization makes it easier to design DTN nodes and networks. It also helps ensure higher reliability and quality of video transmission.

## Applicability

This standard is intended for all future missions which produce, consume, or distribute video via the bundle protocol. The details (formats, resolutions, bitrates, etc) of the video to be transmitted are largely omitted, in order to prevent immediate obsolescence.

## Rationale

This book is written with the assumption that future missions will rely upon a variety of cameras and imaging sensors. The encoding of these videos may be distributed throughout a spacecraft and/or mission, where the only common thread will be the usage of IP. Following the current trends in the media industry, the various encoders are foreseen to use RTP and/or a format which can be encapsulated into RTP.

### References

|  |  |
| --- | --- |
| [1] | IETF, "RTP: A Transport Protocol for Real-Time Applications," IETF, 2003. |
| [2] | IETF, "RFC 4566: SDP: Session Description Protocol," IETF, 2006. |
| [3] | IETF, RFC 2974: Session Announcement Protocol, IETF, 2000. |
| [4] | IETF, "SIP: Session Initiation Protocol," IETF, 2002. |
| [5] | CCSDS, "CCSDS Bundle Protocol Specification," CCSDS, 2015. |
| [6] | S. Burleigh, "draft-burleigh-dtnrg-imc-00: CBHE-Compatible Bundle Multicast," IETF. |

# Overview

In the past two decades, manned spaceflight has undergone many changes. What was once unthinkable is now typical, throughout various components of spacecraft programs. Within spacecraft video systems, the embrace of Commercial Off-the-Shelf (COTS) video equipment, such as action cameras and handheld devices, is wide spread. As a result, the once-centralized nature of video transmission where all cameras transmit to a single encoder & transmission system has been forsaken for a distributed system, where some cameras are encoded by a centralized system, while others have built-in encoders. Furthermore, the variety of ways which video can be transmitted has increased; cameras may use wired networks, uncompressed video mechanisms such as SDI, wireless networks, or point-to-point links. The Real-time Transport Protocol has been used as a single consistent factor for this variety of video transmission opportunities, as it is robust to jitter, well-defined within the IETF, easy to implement, and extendable. In the modern broadcast world, everything from uncompressed video (SDI) to highly-compressed video (H.265) may be transmitted via RTP, over IP & non-IP networks.

This paradigm shift is not unique to the video transmission world; spacecraft transmission has suffered the same fate. A single spacecraft may now have a variety of distinct up & down-link mechanisms, as well as a separate modicum of on-board connectivity options for payloads and visiting vehicles, such as wireless networks based upon 802.11, Proximity-1, etc. The Bundle Protocol has been designed to unify this conglomeration by providing an overlay network for heterogeneous networks, where the concepts of delay and disruption tolerance were integrated from conception.

It is assumed that future deep space networks will rely upon DTN & the Bundle Protocol for their communication, while using RTP for video transmission. Therefore, it is pertinent that best-practices for the union of the two is well-defined prior to any future missions. This standard addresses the following:

* Encapsulation of RTP data on space links.
* RTP performance tuning for BP-based space links.
* Addressing of RTP data, in order to facilitate multiple-sender-multiple-receiver scenarios.
* Interoperability of standards.

## Assumptions

List of relevant RFC’s, list of permitted video/audio formats, etc. Basic assumptions, such as usage of NTP, Restrictions (no fragmentation, that sort of thing)

[Normative specifications appear in sections 3 through *n*. See CCSDS A20.0-Y-4, *CCSDS Publications Manual* (Yellow Book, Issue 4, April 2014).

All sections and annexes should be separated by Word continuous section breaks.]

This section needs to be completed!

## Flow diagram – RTP in larger picture

RTP provides a robust and mature infrastructure for the distribution of video within complex missions, while allowing for a relatively simple interface with on-board and ground hardware. Furthermore, RTP can be interfaced with legacy MPEG Transport Stream based hardware & software without altering the payload data (*e.g.* Video data). The remainder of this section will provide a theoretical implementation of an on-board RTP-based video system for a partially-manned deep space mission, before progressing to its associated ground segment.

The remainder of this section will solely focus on real-time video; as archived video acquired via LOS would also be encapsulated in RTP, the exact details of reordering and replay are out of scope of this book.

### On-board Implementation

Given the operational experiences gained from the International Space Station, it is imperative that the video system of a next-generation spacecraft/mission is designed with extensive flexibility, allowing for new commercial cameras and video sources to be quickly added. Therefore, this section envisions a “partitioned” video system, which is, whenever possible, based entirely off of IP. The basic concept of a partitioned video system is that, while some core functionality (such as cameras and encoders) is inherent to the spacecraft and can be considered to be critical spacecraft data, other sources are transient and/or best-effort (scientific cameras, PR functionality, etc). Furthermore, each spacecraft and rover within the mission may have their own subset of critical video sources. If it is considered that multiple spacecraft or components thereof may dock or otherwise be attached to each other via high-bandwidth connections, it is logical to design a video system which may use the encoders of one spacecraft to encode or transmit video from another. Besides the core encoding functionality, it must be expected that payloads may contain specialized cameras or video sources. Depending upon the location of the payload, it may be optimal to encode the video in the camera, while in other cases, that video should be transmitted to the core encoders.

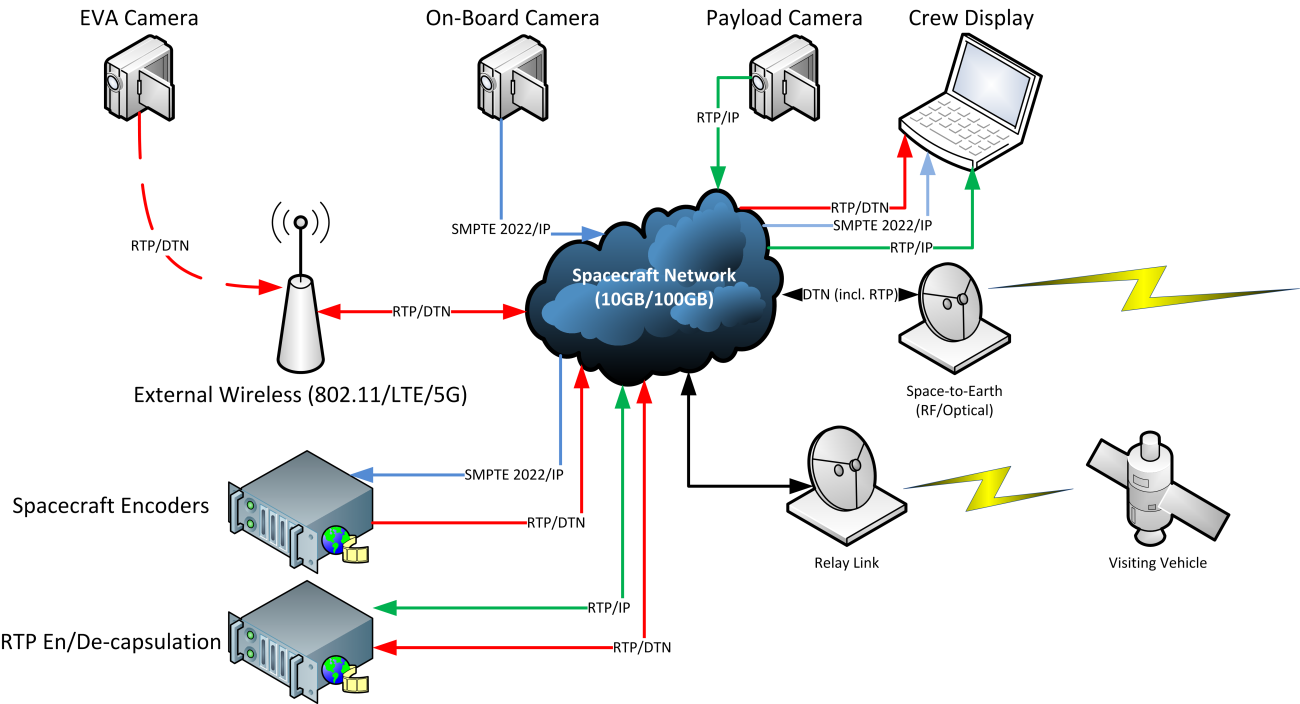


Figure 2.1 Flow Diagram - -On Board Assets

Such a system may be designed entirely around RTP as an encapsulation format; while the best-known uses of RTP are as conveyance for compressed video data, it is also possible to convey uncompressed (or lightly compressed) video data via RTP, using SMPTE 2022 (Needs Citation). Figure 2.1 shows an example of such a system, although redundant links have been omitted for clarity. This design allows for extreme flexibility and for increased redundancy, in that the failure of individual encoders does not affect the capability to encode video. If additional redundancy is required for individual cameras, multiple network links may be furnished for them. However, it must be noted that uncompressed (SMPTE-2022) video requires extremely high bandwidths, on the order of gigabits/second; therefore, the mission designer must carefully design their network to ensure adequate capacity.

As can be seen from the diagram, IP is the sole transport mechanism for inter-spacecraft video, some of which may be encapsulated within the Bundle Protocol. EVA cameras are provided with their own encoders, and output an compressed video stream via RTP. More interestingly, on-board system cameras do not contain their own encoder; instead, they output SDI (as permitted by SMPTE 2022) via RTP, which are encoded by the “spacecraft encoders”, which can be positioned in a less extreme environment, such as inside the spacecraft. These devices can also be largely commercial, except for the DTN output functionality. Some payload cameras may also be commercial devices with integrated compressors, which can output RTP over IP, for encapsulation within the avionics system. This diagram also showcases a visiting vehicle link, which may also contrain compressed RTP-formatted video, which may be viewed on the crew display. Finally, the space-to-ground link, based upon DTN, is utilized to transport RTP-encapsulated video.

### Ground Segment

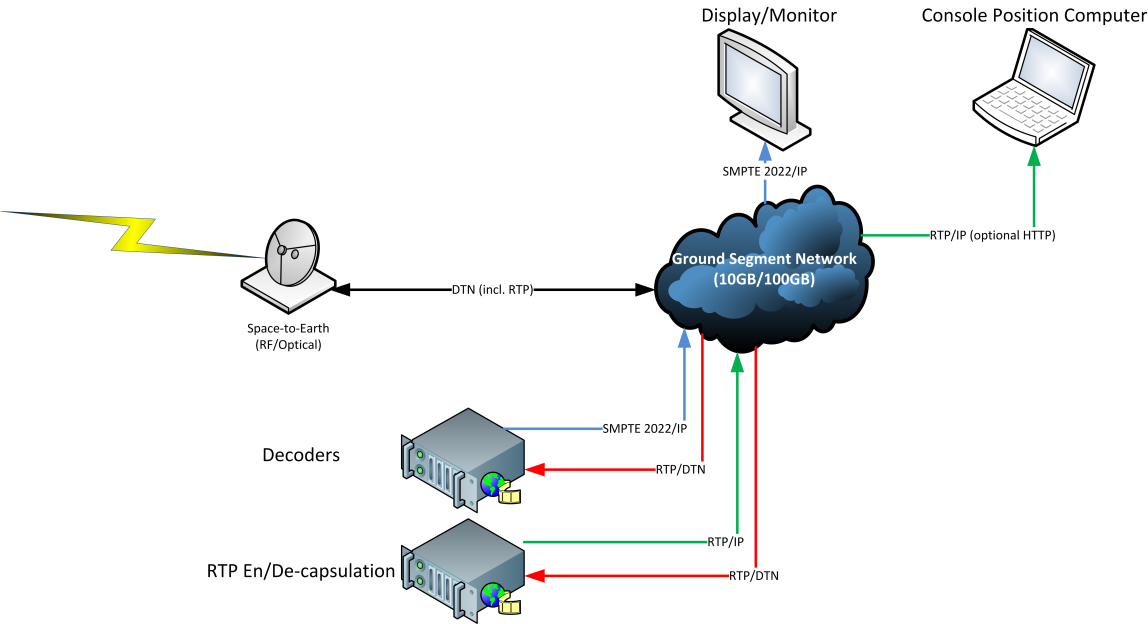


Figure 2.2 Flow Diagram - Ground Segment

Thanks to the homogeny provided by RTP, future ground segments may be simplified, thanks to both the usage of commercial software & hardware, as well as the reduced number of interfaces required for reception. An example ground segment may be found in Figure 2.2, where the acute reader may notice many similarities to the on-board implementation. This figure omits some components, such as recording and archival, although a proposed solution will be discussed in this section.

In the proposed ground segment, DTN-encapsulated RTP data containing compressed video is received from the earth terminal of the Space-to-ground link. This data is sent to the RTP en/decapsulation unit, which removes the bundle protocol headers and transmits the RTP data to the ground segment network. The RTP decapsulation is the only custom component of the system, all further data distribution may be acclomplished with COTS hardware and software.

Indvidual console positions may use a COTS IPTV package (such as VLC Media Player) to select from various channels for display on their individual machine. Decoders may optionally be used, allowing the RTP streams to be decoded to SMPTE-2022. Displays without on-board decoding may be provided with uncompressed data via SMPTE 2022, using COTS IP/SDI converters. External users may make a VPN tunnel to the IPTV package (which, in some cases, also provides a website for channel selection), allowing them to select individual channels for display.

Finally, recording and archival may be provided by the options from the same COTS IPTV packages, where they capture RTP data from the network and encapsulate it into files, allowing for rapid archival and subsequent exchange of video data.

## Technical Overview

The Real-Time Transport Protocol (RTP) [1] provides a lightweight packet format for the transmission of media-related payloads over IP-based networks, and is a fundamental component of many higher-level protocols, such as the Session Initiation Protocol (SIP) [2], used in IP telephony. The remainder of this section serves as a brief overview of RTP; It shall not be misconstrued as an attempt at a complete description of the protocol, which may be found in [1].

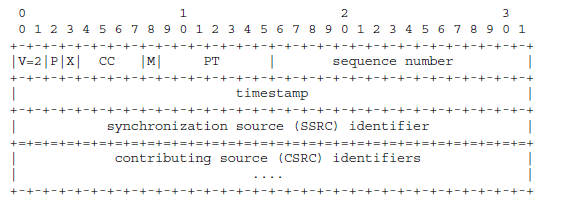


Figure 2.3 RTP Packet Header

RTP is designed around a fixed 12-byte header, along with a variable-length header extension field, as can be seen in Figure 2.3. Amongst other things, this header contains the Payload Type (PT) component, which specifies the type of data which is conveyed in this packet. Additionally, the header contains a timestamp and sequence counters, both of which will be discussed further. The last element to be aware of is the marker bit, which specifies that this packet contains “important data”, although the definition of importance is left to the payload type. Figure 2.4 provides an overview of the various IETF RFC’s which describe RTP.

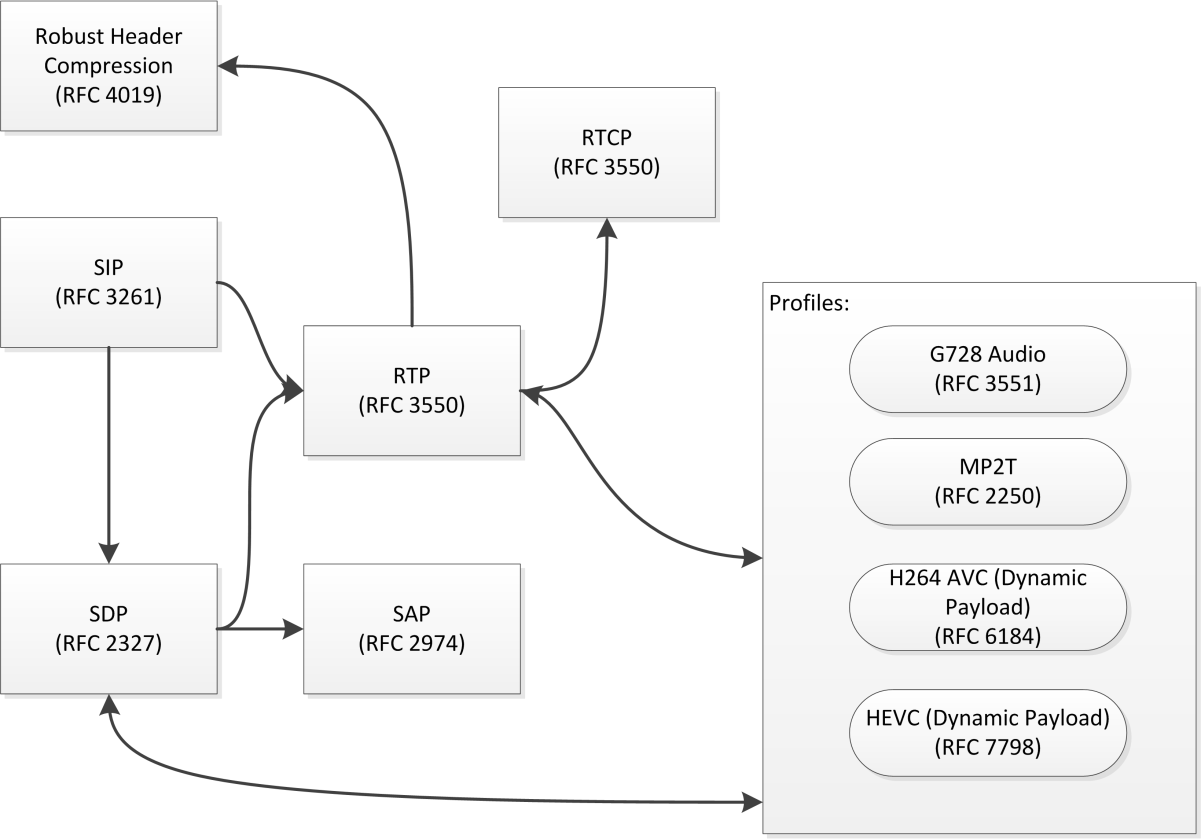


Figure 2.4 RTP Relationships

It is important to note that the RTP packet does not encode the size of the payload data, instead leaving that task to the underlying transport protocols. Furthermore, in order to prevent potential confusion, unless specifically mentioned, the term “payload” refers to the payload of the RTP packet, as opposed to the bundle payload.

### RTP Program structure

RTP profiles, with the exception of the RTP MPEG-TS profile, stipulate that each RTP stream contains a single type of data, such as audio or video. Therefore, a single RTP program may comprise multiple RTP streams, each of which exists on their own channel. In an IP/multicast-based network, it is the responsibility of the receiver to subscribe to all relevant multicasts. This is a flexible mechanism which allows for robust programming; for example, a single RTP-based video encoder could transmit 2 unique video streams at different bitrates along with 2 audio streams in different languages. Based upon the out-of-band signaling information, which is discussed further throughout section 3, the receiver may select the video stream which is best-suited to the available bandwidth as well as the preferred audio language.

This model is also applicable to space-based RTP networks, albeit with some caveats. It is foreseen that a spacecraft network may be heterogeneous, with portions built around standard IP networking, while other segments may be built around DTN, as shown in Figure 2.5. As with standard IP-based networks, data which is destined for a DTN network may be transmitted in one of two ways: unicast or multicast. If more than two DTN endpoints have been factored into the mission design, BP Multicast [5] may be used for transmission over space networks.



Figure 2.5 Spacecraft Network Design

#### RTP Concatenation Mechanisms

In order to facilitate variable Maximum Transmission Units (MTU) in a single transmission path, it is possible to concatenate RTP packets. This is not to be confused with the concatenation function provided by the H.264/265 Network Abstraction Layer (NAL). This concatenation function is applicable to data which is encapsulated within NAL transports, and is relatively complex. It is not recommended to do NAL concatenation within the encoded video data stream with additional concatenation being performed at the network layer.

The timestamp contained within the RTP header is critical for the decoding of data, but may be constant across multiple packets. The presence of a non-incrementing timestamp across multiple packets indicates that the data from all packets must be decoded at once. Therefore, the timestamp must be constant across all packets to be concatenated. Furthermore, when concatenating RTP data, care must be taken to avoid the loss of any data contained within the RTP header, as well as to prevent the interleaving of important and non-important data.

When concatenating RTP packets for DTN transport, a single RTP header is used followed by a variable-length blob of data. The transmitter is responsible for the concatenation, while the receiver must recreate the individual RTP packets, prudent to the maximum applicable MTU for the receiving IP network. This is a similar approach to that which is taken by Robust Header Compression (RFC 4019), but is not identical. RFC 4019 relies upon packet identifiers and/or sequence numbers, while DTN-based RTP concatenation relies upon the sequence numbers and timestamps from within the bundle protocol.

### RTP Synchronization Primitives

When compared to terrestrial RTP-based video systems, space-based networks are subject to unique difficulties with regards to the synchronization of video streams. Mission teams may require synchronization between received video and other sources of data, such as audio (either on camera or from a voice system) or telemetry. Therefore, some care must be taken in order to ensure reliable synchronization of audio and video.

RTP timestamps are based upon the sampling frequency of the encoded data and are intended to be independent to each other. For example, the sampling frequency of most video codecs is 90Khz, while the sampling frequency of telephone-quality audio is 8Khz. Therefore, even if these two programs are intended to be viewed simultaneously, their respective timestamps will increment by different quantities and at different frequencies. Furthermore, the timestamp may remain constant across multiple packets. RTP provides synchronization primitives which are applicable to streams sourced by a single encoder and/or multiple encoders, and provide sufficient fidelity to achieve “wall-clock” synchronization.

Inter-program synchronization may be accomplished via several mechanisms, each of which are covered in one or more IETF RFC’s. The primary mechanism specified by the RTP RFC is via the usage of Real-Time Control Protocol (RTCP) messages. RTCP, described in Section 6 of [1] provides monitoring and control functionality for RTP streams. In an IP network, RTCP data for a given stream is provided on the next odd-numbered port. For example, a RTP stream which exists at 224.0.0.1:4220 would have an accompanying RTCP stream on port 4221. RTP and RTCP share similar packet structures, as shown in Figure 2.6. However, a RTCP packet must be aligned on a 32-bit boundary.

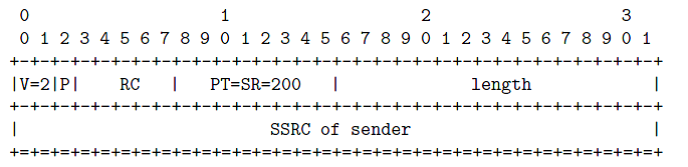


Figure 2.6 RTCP Overview

RTCP has several defined message types, each of which are intended to accomplish different tasks, and will not be described here. For synchronization, the RTCP Sender Report (SR) must be used. The Sender Report, described in section 6.4.1 of [1] and shown in Figure 2.7 is intended to be transmitted by all RTP senders or multiplexers, and contains all information required to synchronize RTP streams.

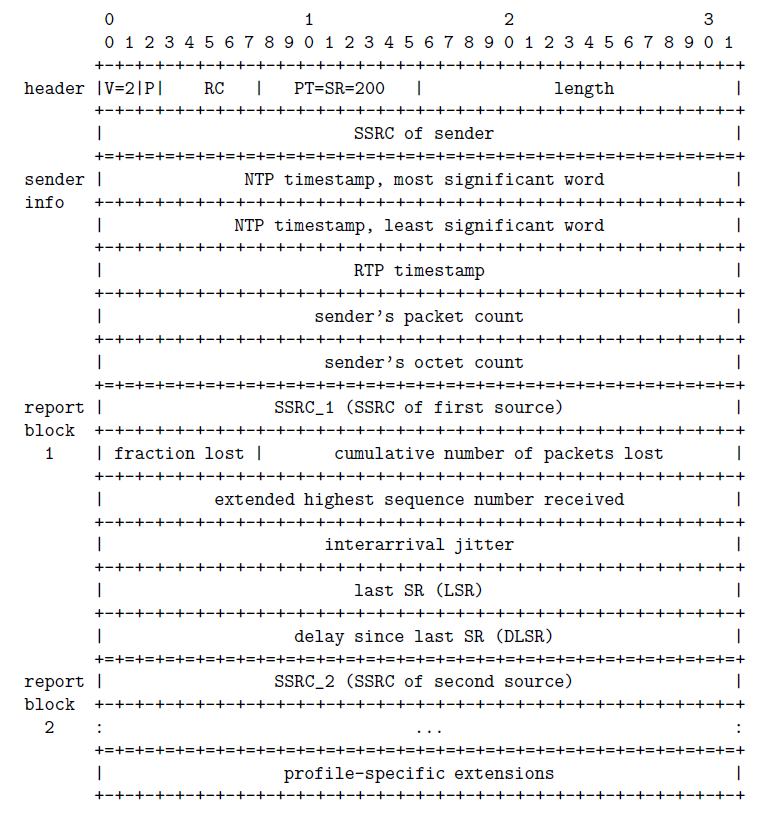


Figure 2.7 RTCP Sender Report Message

The Sender Report message is divided into one sender block and one or more reporting blocks. Encoders and processors which generate this block may omit the reporting blocks, and they may be stripped by re-multiplexers. The sender block contains the NTP timestamp at the time of generation, as well as the RTP timestamp for that media source, valid at the time of SR transmission. Receivers can use that information, along with the sampling frequency of the RTP stream to determine the offset from real-time. By receiving multiple sender reports for all components of the RTP stream, the streams may be coordinated.

### Stream Signalling

RTP focuses upon the transmission of media, and does not contain media description functionality as is present in the MPEG-TS standard. Therefore, external protocols must be used to ensure that all parties are aware of the available video streams.The remainder of this section serves to describe the primary media description protocols which are in use in the RTP ecosystem.

In a full-featured implementation of RTP-over-DTN, it is assumed that there may be several signaling channels which are used for synchronization and discovery of various RTP programs. In order to simplify the transmission of these streams, all signaling may follow the basic rules outlined in section 3.6, regardless of the content of the signaling channel, and outlined in the following sections.

#### SDP

The Session Description Protocol (SDP) [3] is a versatile text-based format for the description of media streams. It provides a mechanism for the exchange of information which is required by a user or decoder, such as the multicast port and addresses for various stream types, as well as the start and end times, names, etc. SDP messages are designed to be relatively small, allowing for their conveyance in a multicast packet, file, or other mechanism (such as DLNA or another service discovery mechanism).

An SDP description is formatted as a set of key=value tokens, delimited by newline characters. Each key must be one-letter, while each value may be an indeterminate length. Per the specification, the key and value must be separated with an equals sign (=), and without spaces. The full list of keys are available in [3], and will not be explained here. However, in order to uniquely describe the components of a single source, each component must be described by the connection information (c=) and the media name and transport address (m=). If the SDP file describes an IP-based stream, the connection parameter is typically a multicast address. The parameter is formatted as a 3-tuple of *(<nettype> <addrtype> <connection-address>*.

If the connection parameter describes a multicast program, then *nettype* must be “IN”, *addrtype* must be “IP4” or “IP6” (depending upon the usage of IPv6), and the connection address must describe the message. If describing a IPv4 multicast stream, the connection address must be formatted as *<base multicast>/<ttl>*. Optionally, the address may be formatted as *<base multicast>/[ttl]/<number of addresses>*. If the number of addresses is provided, it is assumed that the data is encoded using a layered or hierarchical mechanism, and the network will prune un-used addresses.

Within a DTN network, SDP occupies a similar position as in an IP-based network, acting as a mechanism to signal the presence, configuration, and location of program components. However, as the Bundle Protocol provides all necessary transport and validation mechanisms, SDP is transmitted as-is, with no additional protocol overhead (e.g. SAP). Figure 2.8 shows an example of properly-formatted SDP data, intended for transmission over the Bundle Protocol.

**c=DTN BP IPN:1**

m=video **2** RTP/AVP 96

a=rtpmap:96 H264/90000

a=fmtp:96 profile-level-id=640014;sprop-parameter-sets=Z2QAFKzZQ0R+f/zBfMMAQAAAAwBAAAAKI8UKZYA=,aOvssiw=

Figure 2.8 Example of SDP Data.

As the video and SDP data transits the DTN network, the SDP data must be created or altered by the transmitting node . Therefore, the following requirements apply to DTN-transmitted SDP data:

The details of SDP addressing as is used within a DTN network is further expanded in section 3.6.5.

##### Transport of SDP Messages

SDP may be transported over an IP-based network within (amongst others) Session Announcement Protocol (SAP), and the Real Time Streaming protocol (RTSP). It is up to the implementer to determine which transport mechanisms should be supported within the DTN transmitter and receiver.

##### SDP Translation Between Network Domains

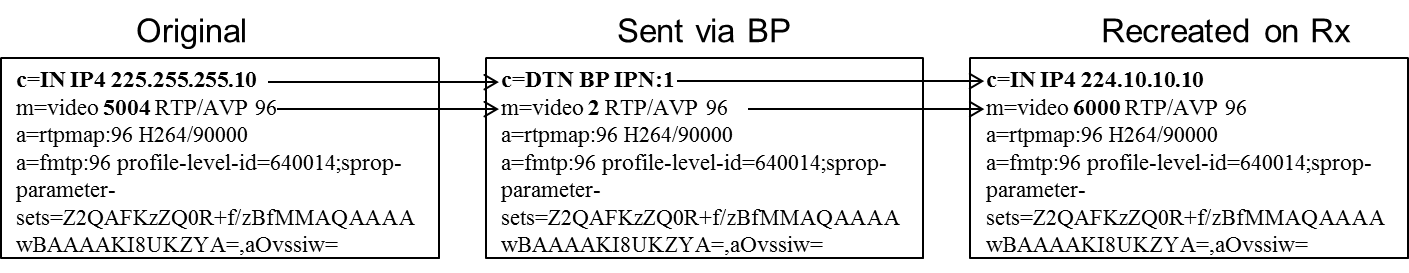


Figure 2.9 SDP Translation Between DTN and IP Domains

The SDP connection & media descriptor parameters are intended to identify the media source across networks. The addition of the DTN & BP parameter types extend SDP to identify media sources across RTP networks. Implementers may freely translate between IP and DTN environments by modifying the relevant portions of the SDP data, as shown in Figure 2.9. An IP-to-DTN RTP converter may map the DTN EID’s which contain media sources to multicast addresses and ports, allowing for a 1:1 modification of all connection information parameters to multicast addresses and media descriptors to ports within that particular address.

## RTP Over DTN

RTP may be encapsulated into DTN bundles with minimal modification, instead treating the entirety of the RTP packet as a singular bundle, as outlined in section 2.4.1. However, astute observaion of the RTP packet format may reveal redundancy between the RTP packet and the BP primary header; namely, both provide a timestamp and sequence number. Implementers of RTP over DTN must accept this, and not attempt to remove the RTP timestamp and sequence, as they are designed to accomplish differing goals, where the RTP timestamp is based upon the sample rate of the payload, while the BP timestamp is a wall-time timestamp, expressed in seconds. Furthermore, the timestamp for the RTP packet can be repeated for a media stream, in order to allow for fragmentation of RTP data. [1] states that all the payload of all packets for a single stream and with a single timestamp must be presented to the decoder at the same moment.

### 1:1 Encapsulation of RTP in DTN

The 1-to-1 encapsulation of RTP within the payload of a bundle may be accomplished without any modification of the RTP packet structure. This is the most basic possible implementation, but may not be efficient due to the fact that the average IP-based encoder uses a Maximum Transmission Unit (MTU) of 1400 bytes, which is acceptable for IP networks. However, given that the bundle protocol & convergence layer adapter will add their own headers, this may create unnecessary amounts of overhead. If possible, the MTU should be set to a value which represents the largest reasonable MTU for a user network before fragementation may occur. However, it is preferable that the RTP concatenation mechanism from section 3.3 is used.

### RTP-Over-DTN To IP

The normative section of this book specifies a robust set of guidelines which, taken in conjunction, allow for creation of an interoperable DTN-based video system. However, in many cases, the utility of such a system is limited by the lack of interfaces between standard IP and the DTN network. The remainder of this section provides a theoretical outline for an IP-to-DTN-to-IP-based video system, shown in Figure 2.10, which will remain in full complicance with section 3 of this book, as well as [1] and [2].

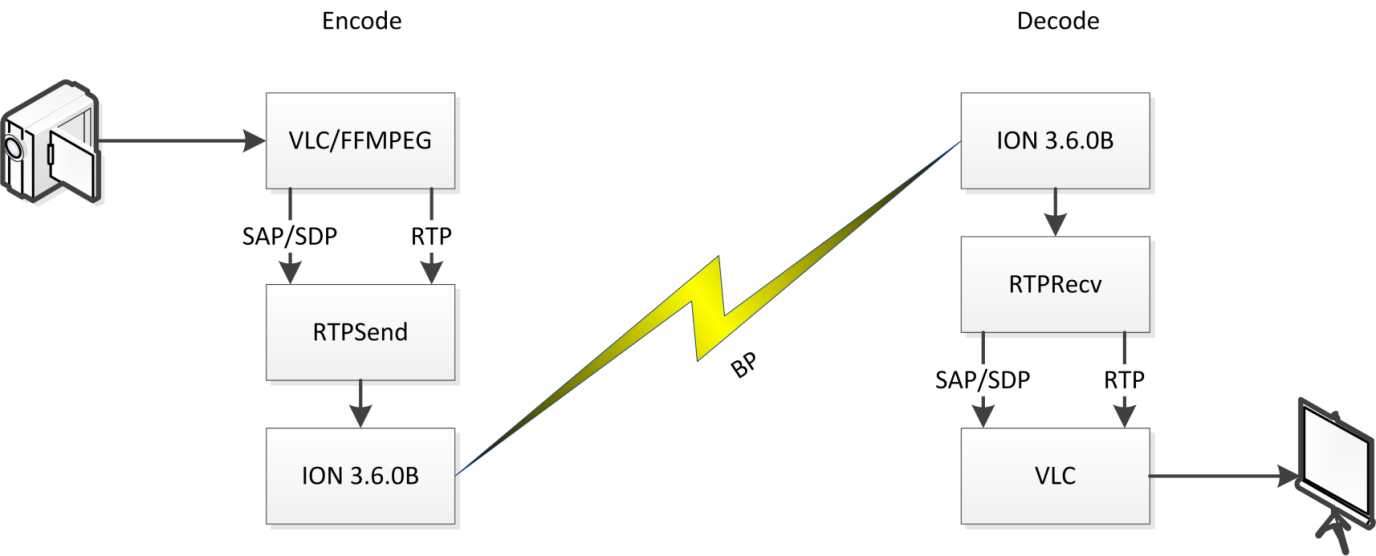


Figure 2.10 Theoretical IP-To-RTP System (Credit: DLR)

#### IP-To-RTP

In order to provide a easily-usable interface between IP and DTN, the Session Announcement Protocol from [3] is used to enumerate video sources within a multicast network. The media streams announced from SAP are described my multicast IP:port tuples, and are subscribed to by the application, which assigns each media source a unique DTN EID. The SDP data received in the SAP message is permutated, in order to remap the EIDs, as described in section 3.6.5.

The RTP concatenation process is run on every RTP stream and sent to the DTN network, as well as the SDP messages.

#### RTP-To-IP

On the DTN receiver, a process listens to the specific EID containing the SDP data. The process from section 3.6.5 is followed in reverse; the media sources described in the SDP data are mapped to a user-defined multicast IP, with incrementing ports, starting from a user-defined port.

The application listens to all the EIDs found within the SDP data and waits for the reception of RTP data on those endpoint IDs. Once RTP data is received, the fragmentation process from section 3.4 is performed, creating a range of RTP streams. Finally, the SDP data (with the new multicast IP addresses) is retransmitted as SAP messages.

#### Interoperability

As discussed in prior sections, the reception of a DTN-encapsulated RTP program may be envisioned as a set of concurrent operations, each of which acts upon different DTN EIDs:

1. Division and re-transmission of RTP packets.
2. Division and re-transmission of RTCP packets, if required.
3. Reception and retransmission of signaling data, if required.

In order to facilitate a flexible and interoperable structure for the receiving applications, it is recommendable that data structures can be made available for exchange between the various applications.

|  |  |  |
| --- | --- | --- |
| Name | From | Description |
| EID Map | Signaling | A list of all EID’s for this program, used for subscription to BP multicast groups. |
| SSRC->port Map | RTP Transmitter | A list of all SSRC’s, along with the multicast ports which are being used for transmission. |

## Legacy Compatibility

The progression from MPEG-TS to RTP has been a multi-decade process, supported by the vast resources of the broadcast industry. Therefore, well-developed methods exist for legacy MPEG-TS support via RTP-based transports.

If the MPEG-TS RTP profile is used, it is the responsibility of the mission developer to announce the stream via SDP over the DTN network.

# RTP in Space-Based Links (NORMATIVE)

## Overview

While section 2 is intended to be a non-normative discussion of RTP, including the special tradeoffs that must be made in order to transport RTP over a DTN network, the remainder of this section provides an implementer with requirements for the implementation of RTP over DTN.

*Note: The remainder of this section assumes that the implementer is using an RTP implementation which is compliant with* [1] *and, if stream signaling is to be used,* [2]*.*

## RTP PDU Formatting

When transmitting RTP data over the Bundle Protocol, implementers must avoid the use of RTP fragmentation, instead relying upon BP fragmentation mechanisms.

If a RTP bundle is generated from an external source, the RTP Concatenation mechanism from Section 3.3 must be used.

If RTP packets must traverse from Bundle Protocol to non-Bundle Protocol based networks, those packets must be decapsulated from the bundle payload. If required, fragmentation must be implemented, as outlined in Section 3.4.

If Secure RTP is used, the padding bit of the RTP PDU header may be set; if set, concatenation must not be attempted.

Note: This does not include padding data which is created by the underlying video codec.

## RTP Concatenation

While some video formats may support aggregation, these mechanisms must not be used in place of the RTP concatenation process.

Note: H.264 & 265 support a form of packet aggregation. However, this mechanism involves the modification of data within the RTP PDU payload section. As a result, the RTP concatenator process would require a deep insight into the packet content. Additionally, it is assumed that such processes may be used by encoders and/or other non-DTN-aware parts of the pipeline, and such functionality should not be repeated.

All concatenated packets must have the same RTP timestamp. Therefore, concatenation shall be triggered based on a monotonic increase of the RTP timestamp; Optionally, the RTP sequence counter can be used in order to verify that there are no gaps prior to transmission of the concatenated data.

The Marker bit for all packets within a single concatenated bundle must be set to the same value. If the marker changes (regardless of all other packet simularities), the previously-concatenated bundle must be transmitted and concatenation of a new bundle shall start with that marked packet.

The state of the Extension (X) bit must remain constant for all concatenated packets in a single bundle. If the X bit is set, the extension data must remain constant across all packets. If the X bit is toggled or the extension data changes, then the bundle must be transmitted and concatenation of a new bundle shall start with the changed packet.

All RTP packets destined for concatenation in a single bundle must be part of the same media stream, as defined by the Payload type as well as the RTP SSRC and/or multicast port. This extends the best practices established in section 5.2 of [1].

The SSRC identifier must not change through the managed transmission path.

Subject to 3.2.4, if any RTP packets destined for concatenation have set the padding bit, the padding must be removed and the bit must be unset.

The sequence counter must increase by one for each outgoing concatenated RTP packet.

Bundles containing RTP packets must be transmitted as soon as concatenation is completed. Bitrate smoothing shall not be attempted.

Note: shows an example of a concatenated RTP bundle. The state of the Marker bit is shown, while the sequence counters and timestamps are omitted.

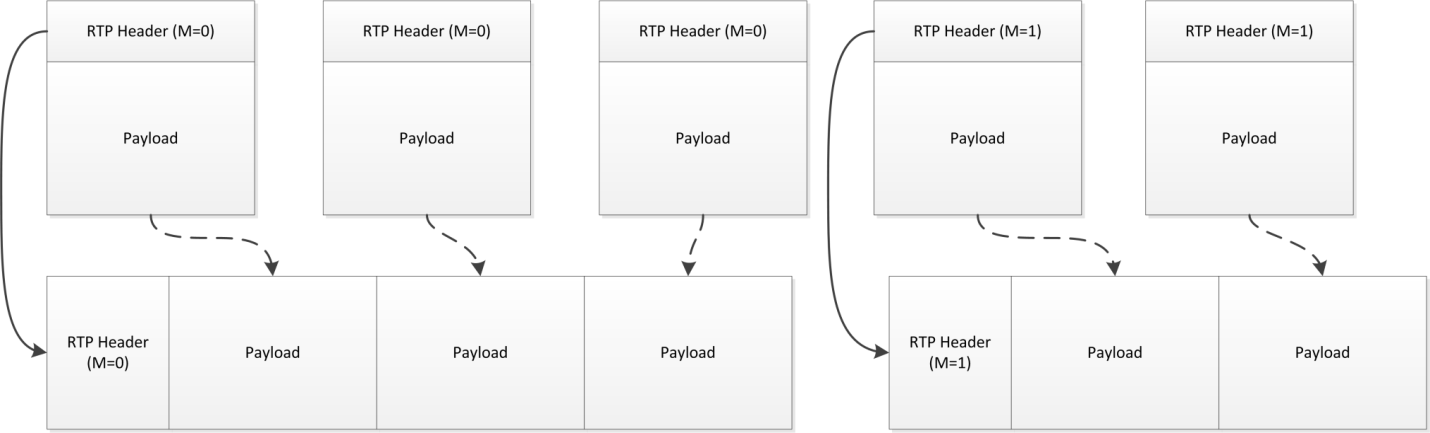


Figure 3.1 RTP Concatenation - Overview

## Receiver-Side Fragmentation

Note: This section primarily refers to the re-fragmentation of DTN-encapsulated RTP data. On the receiver, some basic rules shall be followed. These rules must be modified depending on the technology in use, and shall be expanded in following sections.

Any retransmission of the bundle or bundle segments is expected to occur at lower levels. Therefore, it is expected that fragmentation will only occur upon presentation of the complete bundle.

All incoming RTP payload data must be fragmented into acceptable segments, based upon the underlying output network MTU, which may be set via management or via Network Path MTU Discovery.

Each segment must be prepended with a RTP header and transmitted. With the exception of the sequence count, the RTP header of the outgoing packet must contain all data which was transmitted within the incoming RTP packet.

The sequence number of the outputted RTP packets shall be created independently from the input sequence number.

## DTN Transmission of RTP Packets

Each RTP source which is a component of one or more programs shall be transmitted on an individual EID.

If a bundle size limit it required, it shall be decided by the implementer.

## Signaling Data (SDP Description & RTCP)

All signaling messages shall be transmitted with custody transfer enabled.

All signaling messages shall be transmitted at least once every 30 seconds.

Each type of service message shall have its own unique EID.

If bundle multicast is used, multiple programs may transmit their service messages on the same EID.

### SDP

Unless the MPEG-TS RTP profile is used, the stream must be announced via SDP over the DTN network.

For all connection information parameters (c=), the *nettype* shall be set to “DTN”

For all connection information parameters (c=), the *addrtype* shall be set to “BP”

For all connection information parameters (c=), the address field shall be the RFC5050 URI representation of the CBHE node number where the stream or service may be found. The scheme name shall be included, while the service number and proceeding period (.) character must be omitted.

The address field of the connection parameter must specify a single DTN endpoint ID.

For each media descriptor (m=) in the SDP data, the *port* element must be changed to the service number of the BP EID which represents that media element.

Note: The receiving node must be able to unambiguously determine the EID of each media component by combining the address field of the connection information parameter with the port element of the media descriptor.

Other elements of the media description must be transmitted without alteration.

#### Discussion

A connection parameter received from an (multicast-based) IP network may be formatted as:

*c=IN IP4 224.1.0.1/255*

The equivalent for a IPN-based BP network would be:

*c=DTN BP ipn:1*

Likewise, a media descriptor for an a service transmitted on port 6000 of an IP multicast network would be formatted as:

*m=video* ***6000*** *RTP/AVP 96*

If CBHE service ID 2 is used for the outgoing media descriptor, this media descriptor would be formatted as:

*m=video* ***2*** *RTP/AVP 96*

Therefore, a connection parameter describing *ipn:1* as the root of the media service may be combined with the media descriptor describing service ID 2 in order to form the IPN name of *ipn:1.2*

### RTCP Concatenation

#### Transmission Concatenation & Receiver Re-fragmentation

Note: RTCP concatenation is slightly more complex than the RTP concatenation mechanism specified in section 3.3. RTCP packets are designed to be concatenated, and can be combined into a compound RTCP packet, as outlined in section 6.1 of [1]. The following rules must be observed by the implementer; these rules are normative, and override the suggestions provided within the RFC.

If any incoming RTCP packets contain padding, it must be stripped during the concatenation process. The padding bit of all RTCP packets within the concatenated packet must be set to 0.

All non Sender Report RTCP packets must be ignored during the concatenation process.

The SSRC’s within the RTCP packets shall not be changed.

The payload of each bundle conveying concatenated RTCP data must contain the latest RTCP sender report from each SSRC which was received since the last elapse of the transmission interval; no other RTCP packets may be contained within a single bundle.

Note: Unlike RTP concatenation, RTCP packets from multiple SSRC’s may be multiplexed.

A RTCP Sender Report, as identified by a tuple of SSRC & Timestamp must not be repeated in multiple bundles.

#### RTCP Decapsulation

If the encapsulated RTCP packets are ultimately destined for an IP-based network, the following additional constraints must be applied:

The receiver must de-capsulate and forward every RTCP packet from a received and concatenated bundle.

Concatenated RTCP sender reports must be split into their component reports and transmitted to an odd-numbered port. Subject to the reccomendations in section 11 of [1], this port must be one above that which is used for RTP transmission of a single media source.

Note: since RTCP packets from multiple media sources may be concatenated, it is imperative that the implementer is aware that a single RTCP bundle may contain reports which will ultimately be destined for multiple ports on the IP network. Furthermore, as RTCP sender reports do not contain IP information, the only way to correlate the media sources from the RTCP packet with media sources on the network is via the SSRC of those component sources.

#### Packet Transmission Frequency:

The packet transmission algorithm outlined in section 6.3.1 of [1] shall be ignored and replaced with a fixed transmission interval of less than or equal to 15 seconds.

Note: As shown in Figure 3.2, the RTCP transmission interval determines the maximum period before real-time synchronization of RTP-based media streams with other sources (such as Telemetry) may be achieved, and must be considered carefully.

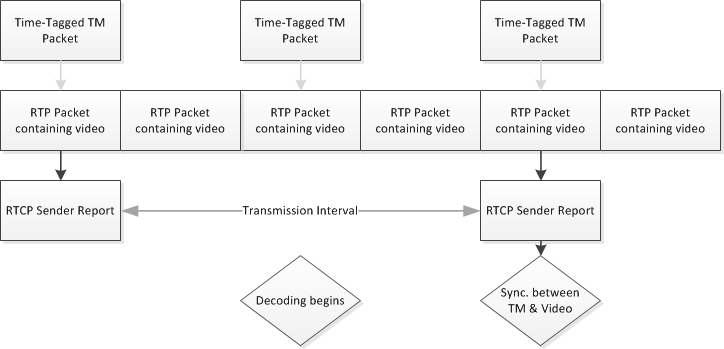


Figure 3.2 RTCP & Video Synchronization

After the transmission interval has elapsed, the concatenated bundle shall be transmitted only if non-empty.

Note: As a result of these rules, it is strongly encouraged that the receiver maintains additional state-related information, such as a table of SSRC’s and respective multicast addresses. Therefore, as stated in [1], the receiver will retransmit the RTCP packet on a multicast port which is one above the associated RTP stream. However, If the RTP and RTCP demultiplexers are different applications, the SSRC-port mapping from section 2.4.2.3 would be used for enquiry.

#### In-Band SPS/PPS Transmission of H264 Data

If in-band transmission of H.264/H.265 SPS/PPS data is to be used for video source, SDP data must only be transmitted if multiple media sources exist within a single program AND/OR RTCP is used.

# ~~Conceptual: Ground Systems & Display Systems~~

~~How this will be presented to ground systems & display devices. LOS handling. Division and replay of “real-time” data vs. that which was archived during LOS.~~

1. Implementation Conformance   
   Statement (ICS) Proforma  
     
   (normative)
   1. INTRODUCTION
      1. OVERVIEW

This annex provides the Implementation Conformance Statement (ICS) Requirements List (RL) for an implementation of [Specification]. The ICS for an implementation is generated by completing the RL in accordance with the instructions below. An implementation claiming conformance must satisfy the mandatory requirements referenced in the RL.

* + 1. ABBREVIATIONS AND CONVENTIONS

The RL consists of information in tabular form. The status of features is indicated using the abbreviations and conventions described below.

Item Column

The item column contains sequential numbers for items in the table.

Feature Column

The feature column contains a brief descriptive name for a feature. It implicitly means “Is this feature supported by the implementation?”

Status Column

The status column uses the following notations:

* M mandatory;
* O optional;
* C conditional;
* X prohibited;
* I out of scope;
* N/A not applicable.

Support Column Symbols

The support column is to be used by the implementer to state whether a feature is supported by entering Y, N, or N/A, indicating:

Y Yes, supported by the implementation.

N No, not supported by the implementation.

N/A Not applicable.

The support column should also be used, when appropriate, to enter values supported for a given capability.

* + 1. INSTRUCTIONS FOR COMPLETING THE RL

An implementer shows the extent of compliance to the Recommended Standard by completing the RL; that is, the state of compliance with all mandatory requirements and the options supported are shown. The resulting completed RL is called an ICS. The implementer shall complete the RL by entering appropriate responses in the support or values supported column, using the notation described in A1.2. If a conditional requirement is inapplicable, N/A should be used. If a mandatory requirement is not satisfied, exception information must be supplied by entering a reference X*i*, where *i* is a unique identifier, to an accompanying rationale for the noncompliance.

* 1. ICS PROFORMA FOR [SPECIFICATION]
     1. GENERAL INFORMATION
        1. Identification of ICS

|  |  |
| --- | --- |
| Date of Statement (DD/MM/YYYY) |  |
| ICS serial number |  |
| System Conformance statement cross-reference |  |

* + - 1. Identification of Implementation Under Test

|  |  |
| --- | --- |
| Implementation Name |  |
| Implementation Version |  |
| Special Configuration |  |
| Other Information |  |

* + - 1. Identification of Supplier

|  |  |
| --- | --- |
| Supplier |  |
| Contact Point for Queries |  |
| Implementation Name(s) and Versions |  |
| Other information necessary for full identification, e.g., name(s) and version(s) for machines and/or operating systems;  System Name(s) |  |

* + - 1. Identification of Specification

|  |  |
| --- | --- |
| [CCSDS Document Number] | |
| Have any exceptions been required?  NOTE – A YES answer means that the implementation does not conform to the Recommended Standard. Non-supported mandatory capabilities are to be identified in the ICS, with an explanation of why the implementation is non-conforming. | Yes [  ]      No [  ] |

* + 1. REQUIREMENTS LIST

[See CCSDS A20.1-Y-1, *CCSDS Implementation Conformance Statements* (Yellow Book, Issue 1, April 2014).]

1. Security, SANA, and Patent Considerations  
     
   (Informative)
   1. Security Considerations
      1. security concerns with respect to the CCSDS document
         1. Data Privacy
         2. Data Integrity
         3. Authentication of Communicating Entities
         4. Control of Access to Resources
         5. Availability of Resources
         6. Auditing of Resource Usage
      2. Potential threats and attack scenarios
      3. Consequences of not applying security to the technology
   2. SANA Considerations

[See CCSDS 313.0-Y-1, *Space Assigned Numbers Authority (SANA)—Role, Responsibilities, Policies, and Procedures* (Yellow Book, Issue 1, July 2011).]

* 1. Patent Considerations

[See CCSDS A20.0-Y-4, *CCSDS Publications Manual* (Yellow Book, Issue 4, April 2014).]