

Report Concerning Space Data System Standards

|  |
| --- |
| Streaming Services over Bundle Protocol REquirements |

Draft Informational Report

CCSDS 000.0-G-0

Draft Green Book

October 2014

AUTHORITY

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
|  | Issue: | Draft Green Book, Issue 0 |  |
|  | Date: | October 2014 |  |
|  | Location: | Not Applicable |  |
|  | | | |

**(WHEN THIS INFORMATIONAL REPORT 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 reflects the consensus of technical experts from 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).

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.]

Through the process of normal evolution, it is expected that expansion, deletion, or modification of this document may occur. This document 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.

DOCUMENT CONTROL

|  |  |  |  |
| --- | --- | --- | --- |
| **Document** | **Title and Issue** | **Date** | **Status** |
| CCSDS 000.0-G-0 | [Document Title], Draft Informational Report, Issue 0 | October 2014 | Current draft |
|  |  |  |  |
|  |  |  |  |

CONTENTS

Section Page

# Introduction

## Purpose and scope

The purpose of this document is to record requirements for streaming services over Bundle Protocol, with particular emphasis on streaming digital video over Bundle Protocol. Previous testing of video streams over Bundle Protocol will be documented. A common test configuration for continued testing and benchmarking of video (and other streaming data) will also be documented.

## References

The following documents are referenced in this Report. At the time of publication, the editions indicated were valid. All documents are subject to revision, and users of this Report are encouraged to investigate the possibility of applying the most recent editions of the documents indicated below. The CCSDS Secretariat maintains a register of currently valid CCSDS documents.

[A list of documents referenced in the report goes here. See CCSDS A20.0-Y-4, *CCSDS Publications Manual* (Yellow Book, Issue 4, April 2014) for reference list format.]

# OVERVIEW

Previous testing and real-life experience with streaming video over networks indicates that video streams are particularly susceptible to network jitter and lost packets. Video decoders typically buffer the incoming data stream to reconstitute the frames of video that were encoding using “group of pictures” algorithms that combine frames or disassemble video frames into blocks of pixels. If enough data is missing, even with buffering, or the data arrives jumbled or out of order beyond what the decoder’s buffering can handle, the decoder will either freeze the last good frame video and present it as live video output, or will simply default to a blank or colored screen.

It is likely that as humans endeavor to explore space beyond low Earth orbit, video will be included as important data transmitted back to Earth. Whether it is used for situational awareness, such as proximity of approaching spacecraft during docking and rendezvous, or monitoring an Extra Vehicular Activity, or for public use to allow the rest of us on Earth to “go along for the ride,” successful transmission and reception of video will become an important requirement for mission success. As these missions move beyond the Earth-Moon system, it is very likely the data communications will be over delay tolerant networks.

This Green Book will explore the requirements for video over bundle streaming protocols and document prototyping and testing of video over these protocols.

# Use Case Scenarios

## Low earth orbit

*Describe here real-time and file based work flows including LOS scenarios, such as large RED camera file transfer.*

## Cis lunar

*Describe here likely scenarios including proximity operations (lander or robot plus orbiting spacecraft, rendezvous, EVA) over variable links from S-band to Ku, Ka & X.*

## mars campaign

*Describe here multiple scenarios, expanding on CIS Lunar, accounting for extended light time, priority of content (example of low quality imagery for situational awareness), optical links to S-band, and extended proximity operations.*

# Requirements

*List here requirements for streaming capability over bundle protocols, keeping in mind the use case scenarios outlines in section 3.*

# Methods for transmission of video over bundle streaming protocols

## DLR – ENCODING AND ENCAPSULATION

DLR has developed two systems for video transmission via DTN networks. The first is the transparent gateway which aims to provide a simple transport for UDP-based media protocols which is agnostic of the protocol running through it. The second is a more advanced native encoder which integrates directly with a H.264 video encoder and decoder and designed to natively function with DTN.

### Transparent Gateway

The transparent gateway is a set of applications which encapsulate UDP data into DTN bundles while maintaining the important timing information. This technique is primarily used for MPEG Transport Streams. The gateway will ingest a user-configurable number of UDP packets directed towards it and add additional metadata, comprised of a size and a nanosecond-resolution timestamp, generated as a delta between UDP packet reception at the gateway. Once the given number of packets have been received, they are serialized. A header containing a count of packets and a sequence number is prefixed to the serialized data. The gateway can be utilized as a drop-in replacement for existing link-layer protocols. Other multimedia protocols such as RTP have been successfully tested with the gateway.

The gateway implementation was complicated by the interleaving inherent in MPEG-TS data, as well as the 4-bit MPEG-TS sequence counter. The 4-bit counter overruns quickly, and will not typically (at higher bitrates) lend itself to the resequencing of data, even when that data is occurring within the same one-second DTN timestamp. The gateway receiver aims to prevent this by utilizing the sequence number to reorder packets into a buffer before outputting them based on the timestamp from the header. By tuning the input buffer size, a user can reduce the visual impact of out-of-order packets.

### Direct H.264 transmission

In the process of testing the transparent gateway, it was quickly discovered that DTN provides a greater advantage and requires less overhead when utilized with larger bundles, hence the addition of the multiple UDP packet encapsulation capability mentioned above. In addition, much of the robustness which MPEG-TS provides in terms of error-recovery and interleaving are inherent capabilities of a properly configured DTN link.

This encoder does not attempt to interleave data, instead relying on the underlying DTN stack to perform that task. Instead, the encoder outputs individual compressed frames as single bundles. Minimal metadata is added, comprised only of a width, height, and frame-rate, all of which are requirements for the initialization of the H.264 decoder. Frames are encoded in the packet-oriented H.264 Network Abstraction Layer (NAL) format. The decoder simply initializes a decoder, decodes the data provided in the bundles before finally displays them.

The native H.264 transmitter is extremely robust to interruption and packet loss. As LTP provides retransmission and fragmentation capability and will not present a bundle to the application layer before transmission has completed successfully, each bundle can be assumed to be intact. As such, each frame can be assumed to be intact as well. The order of packets is maintained via the timestamp from within the bundle protocol as well as a per-second count of frames. Any packet which contains a timestamp is less than the current “running” timestamp is assumed to have arrived out-of-order and is archived. Once the one-second frame count is equal to the framerate from the metadata, the video for that second is assumed to be 100% retrieved. The disadvantage of this system is the uniqueness of its implementation. The encoder and decoder are built using the FFMPEG libraries but are otherwise self-contained. It is technically possible to integrate it with other IP-based encoders and decoders by creating a new and functionally-identical MPEG TS output. It must be noted that the encoder must use a codec which supports frame-based output, such as H.264, motion JPEG2000, or H.265.

### Summary

Exhaustive in-house testing between both systems using MPEG-2, H.264, and H.265 has been performed. H.265 testing was ceased due to the high CPU requirements for software encoders. In general, it has been found that the native H.264 system provides higher video quality, although the integration between that system and the rest of a video pipeline is complex.

The gateway, running over LTP with a 2 second buffer have been shown to handle 8 mbps H.264 transport streams and allow for some packet loss with no visual degradation. Running with a smaller buffer demands a “perfect” connection, where even a small packet loss may cause a momentary disruption of audio or video.

The native transmission system running over LTP with a 25 frame buffer (one second at PAL rates), with an 8mbps encoding bitrate has been found to be resistant to a 10% bit error rate without visual degradation when running with a <1 second delay. If the delay is short enough, it is possible for any LTP retransmissions to occur before the next frame is due to be displayed. If the delay is longer than one second, then there may be some visual impact, but it will appear as dropped frames and eventually wind up in the archive. The time to archive can be shortened by using Bundle Streaming Service, though DLR has opted to not implement it.

## BUNDLE STREAMING SERVICE

Bundle Streaming Service (BSS) is a pair of complementary capabilities designed to provide satisfactory contemporaneous presentation of streamed data in transmission sequence, possibly with some omissions due to data loss in transit, while also supporting retrospective presentation of the same stream with all omissions automatically repaired by background retransmission.

BSS is not a video service per se: unlike the DLR technologies for video over DTN discussed above, it is not specifically tuned for video transmission. By the same token, it is not limited to video transmission: the general character of BSS data delivery can be applied to one-way voice transmission, to “real-time” telemetry, or to any other continuous data stream that can be transported by bundles. Good video display quality will always require application-layer data conditioning such as is performed by the DLT transparent gateway and direct H.264 systems. BSS, in contrast, focuses on transport resilience and buffer management.

### BSS database library

At the receiver of the streamed transmission, the BSS database library is integrated into a user-defined Bundle Protocol application that acquires bundle payloads – application data units (ADUs) – destined for a designated BP endpoint. The acquired ADUs can be in any format that is meaningful to the application, as their content is opaque to the BSS library. The sender of those ADUs can be any application.

The receiving application delegates to a BSS library function the job of receiving these ADUs upon delivery from the bundle protocol agent (BPA). The BSS library function inspects the bundle creation times of the bundles that transported the delivered ADUs and dispatches the application data in one of two ways:

* If the bundle creation time of the ADU’s carrier bundle is greater than that of any previously received ADU from the same sender, then the content of the ADU is deemed “in order” and is passed to a “real-time” presentation function that must be provided by the application. The ADU content is also written to a database designed for very high-speed access, for future replay.
* Otherwise, the ADU content is deemed to have been delayed in transmission, possibly because it had to be retransmitted. Since it has arrived out of order, it must not be passed to the application’s real-time presentation function: if the data were video frames, for example, to do so would scramble the video display. Instead, the ADU content is only written to the database. ADU content in the database is ordered by transmission time, so over the course of the transmission the in-order and out-of-order data are merged in time sequence into a single uninterrupted stream, so that a higher-quality display of previously presented data can be viewed in replay.

### BSS Protocol

The other component of Bundle Streaming Service is Bundle Streaming Service Protocol (BSSP), a BP “convergence layer” protocol. Like all convergence-layer protocols, BSSP manages the transmission of bundles directly from one BP node to some other, topologically adjacent BP node. To do so, it operates two concurrent transmission channels, one unreliable, the other reliable. The implementations of these channels are opaque to BSSP and are established by node configuration: one BSSP engine might use UDP/IP for the unreliable channel and TCP/IP for the reliable channel, while another might use LTP “green” transmission for the unreliable channel and LTP “red” transmission for the reliable channel.

When a bundle is presented to BSSP for transmission, the protocol inspects the bundle’s creation time and dispatches the application data in one of two ways:

* If the bundle creation time is greater than that of any previously presented bundle from the same sender, with the same destination, then the bundle is transmitted using the unreliable channel. That is, data presented in order are forwarded in order over the unreliable channel, to minimize end-to-end delivery latency.
* Otherwise, the bundle is transmitted over the reliable channel where it is subject to automatic retransmission upon detection of data loss. It will arrive somewhat later than the in-order data, but its eventual end-to-end delivery is virtually assured.

Upon reception of a bundle sent on the reliable channel, the receiving BSSP engine simply passes the bundle up to the BPA for delivery or further forwarding.

Upon reception of a bundle sent on the unreliable channel, the receiving BSSP engine passes the bundle up to the BPA in the same way, but it also sends an acknowledgment back to the sending BSSP engine.

When the sending BSSP engine receives a BSSP acknowledgment for some forwarded bundle its transmission of that bundle is deemed complete. But if no such acknowledgment is received prior to expiration of a per-bundle timer that was set at the moment of transmission on the unreliable channel, then transmission on the unreliable channel is deemed to have failed. At that point the bundle is re-dispatched on the reliable channel exactly as if its creation time had been out of order when originally presented.

### Some notes on BSS

The two components of BSS (database library and protocol) are complementary, but neither is reliant on the other; each can be used by itself if that is desirable in a given deployment configuration.

A key advantage of the BSSP design is that, because it operates at the convergence layer underneath BP, it can support bundle multicast. Bundle multicast functions by sending copies of a given bundle to multiple topological neighbors; each such copy is conveyed separately by the applicable convergence-layer protocol, and any retransmission that is required in the course of that conveyance is managed privately by that convergence-layer adapter without any impact on transmission to any other neighbor. BSSP enables streaming application data presented to BP to be forwarded to an unlimited number of final destination applications with minimal end-to-end latency in a virtually error-free manner.

## common test scenarios for future study

*Here we outline common testing configurations for future trail blazers to allow them to add content to this book in the future*

1. [ANNEX TITLE]

[Annexes contain ancillary information. See CCSDS A20.0-Y-4, *CCSDS Publications Manual* (Yellow Book, Issue 4, April 2014) for discussion of the kinds of material contained in annexes.]