IAC-15- B3.9-YPVF.2

Techniques for the Use of Video over Delay Tolerant Networks as a Tool for Safety and Situational Awareness

Jeremy Pierce-Mayer
INSYEN AG, Germany, Jeremy.mayer@insyen.com

As human spaceflight is poised to head beyond low-earth orbit, the use of point-to-point communication (as was done for Apollo) or single-point data relay satellites (as is done for the International Space Station) will cease to remain cost-effective. In addition, there is an increase in the amount of data required for transmission.

Interplanetary Networking seeks to solve these issues. IPN envisions a mesh network which interconnects ground stations, rovers, relay satellites, and human spacecraft utilizing Delay-Tolerant Networking (DTN) technologies. This paper will outline the technical challenges and solutions which surround the use of delay-tolerant networking as a transfer medium for situational awareness video and particularly for Human Spaceflight. The problem of multimedia transmission is approached as a high-level issue suited for protocols running above a delay-tolerant network.

Initial testing of video-over-DTN was precluded by the lack of suitable tools for transmission and reception. As a solution, an UDP-to-Bundle Protocol gateway was developed. This gateway allows DTN applications to seamlessly interoperate with existing video systems, and has been tested with great success by DLR/GSOC and RosCosmos. Additionally, techniques for the transmission of 3D point cloud data will be shown. These techniques utilize octree data structures in order to perform spatial filtering and reduction of the data set, before encoding it for DTN transmission. One method removes the requirement for prediction frames, as such allowing an “intra-frame” 3D compression. Notably, the ability to define a “free-form” viewpoint of the image allows for an increase in situational awareness.

Finally, various combinations of these techniques will be shown. The combination of DTN and video transmission technologies allows for an increase in the video capabilities of future missions and consequently in situational awareness. A future rover may use a DTN relay network to transmit video to an orbiting crew via an international system of DTN-compatible relay satellites, allowing the crew and ground teams to rapidly evaluate a potential landing site. Once landed, the rover may seamlessly switch to utilizing the landers connectivity and continue to transmit video back to earth. Additionally, the lander may transmit a 3D map of the descent area generated from a LIDAR, providing a full image of the environment for the ground. Finally, while returning to an orbiting space station, the arriving spacecraft may transmit its docking camera (either directly or via a relay satellite), allowing the space station teams to confirm a successful docking profile

I. INTRODUCTION

Delay Tolerant Networking (DTN) is a protocol suite which has been designed from the ground-up to provide connectivity for sparsely connected network environments. By providing each individual node on the network with store-and-forward capabilities, DTN allows a guarantee of end-to-end communication, even if a direct communication path cannot be found. This technology is of great benefit to the space industry, as it allows robust mesh networks to be formed.

DTN is designed around the concept of bundles, which are arbitrary length data units, which are fragmented for transmission. In addition, DTN relies on the concept of convergence layer adapters (CLA), which offload bundle transmission to the link or network layer. For example, a CLA could be built upon TCP for use in standard IP networks, while a separate CLA could be built for space link protocols.

Provided that there is wide enough agency support, Delay-Tolerant Networking will allow for highly adaptive and robust interplanetary networking. Mission planners may utilize individual orbiting satellites as relay nodes which, in turn, communicate to other connected assets. This will allow space-to-ground assets such as the Deep Space Network to be run at capacity and for multiple concurrent missions.

DTN has been designed to provide basic quality of service, which has been extended by the Extended Class of Service (ECOS) system[1]. ECOS attaches a small header to DTN bundles which specifies (amongst other things) a highly-granular priority level between 0 and 255. This value specifies the relative transmission and forwarding priority, such that a bundle with a priority of 200 will be sent before one with a priority of 100. This capability is highly beneficial for overall situational awareness, as videos which are time-critical with a significant impact on the overall awareness for a given spacecraft may be prioritized over less critical data, such as e-mail.

For our testing, we utilized the ION DTN stack, developed by the Jet Propulsion Laboratory[2]. ION is a full-featured DTN implementation designed for spacecraft applications, and includes an implementation of the Licklider Transmission Protocol (LTP)[3]. LTP allows for the reliable transmission of data over inherently unreliable networks, such as UDP or space links.

METHODS FOR VIDEO TRANSMISSION:

Two systems for video transmission via DTN networks have been developed, and will be discussed in the following sections. 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 H.264 video encoder and decoder and designed to natively function with DTN. In the following sections, both will be discussed, as well as a comparison of their advantages and disadvantages.

Transparent Gateway:

The author has developed several methods by which the transmission of video over DTN links may be performed, each with individual strengths and weaknesses. The simplest of these is the "transparent UDP gateway", which 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, a standard originally developed for the transmission of digital television programs[4]. A single transport stream may contain multiple programs, each containing multiple video and audio streams and represented as program streams. As the transport stream specification was originally designed for the transmission over satellite, cable, and over-the-air which are unidirectional and error-prone, it is also well-suited to transmission via UDP connections. MPEG transport streams are composed of 188 byte packets, which, when transmitted via UDP, are packaged in groups, and transmitted as UDP packets. Typically, 7 TS packets are bundled, creating a 1316-byte payload which may be transmitted as a single UDP packet. By its design, MPEG TS provides an interleaving system, wherein a single 188-byte packet can contain a payload containing compressed audio, video, or various program tables defining what is available within the stream. In addition, timing information is sent, allowing the decoder to synchronize with the encoder for presentation to the end-user.

The gateway, written in C, 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 within the payload, and a sequence number is prefixed to the serialized data. The timestamp is intended to aid in providing a timed output at the receiver. While DTN provides a sequence count composed of a second-resolution transmission time as well as a count which is guaranteed to be unique for any packet sent in that second, the gateway adds an additional sequence number which is guaranteed to linearly increase. This simplified the implementation of lost & out of order packet recovery and counters, which was of great use in testing and troubleshooting. The addition of this metadata allows the gateway to avoid an understanding of the protocol being transmitted via the DTN link, while still being able to provide a properly timed transparent connection.

The gateway provides the major advantage that it can be utilized as a drop-in replacement for existing link-layer protocols, and interconnects a significant majority of the networkable video equipment which is currently available on the market. It should be noted that other multimedia protocols such as RTP have been successfully tested with the gateway. It was decided that MPEG-TS would be used for further testing, due to its wide market support, as well as being the recommended CCSDS video transport protocol[5] and possessing a relatively low overhead (4 bytes minimum).

Direct H.264 transmission:

In the process of testing the transparent gateway, it was quickly discovered that DTN provides a greater advantage 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. As a result, it was decided to implement a native DTN encoder/decoder solution in C++.

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 and are typically encoded in whichever transport protocol is in use. Frames are encoded in the packet-oriented H.264 Network Abstraction Layer (NAL) format[6] Likewise, the decoder attempts to avoid the creation of buffers, and simply decodes the data provided in the bundles and displays them.

Comparison:

Exhaustive testing between both systems has been performed in house, with the industry standard MPEG-2 and H.264 video codecs utilized for a majority of tests.

Some tests were performed using H.265 using x265 and mp4box[7]. It was noted that the CPU requirements for H.265 compression were extremely high, so it was decided to performing further testing at a later date when encoding is more efficient. When hardware encoders and decoders were required, Ericsson encoders (CE-XH 40) and decoders (RX-1290) were used. For other tests, as well as for the implementation of the native solution, FFMPEG[8] was used as an encoder and decoder, and VLC[9] was used as a player for the transparent gateway. Our native solution includes a decoder and viewer, so the use of VLC was unnecessary.

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 relatively complex.

A disadvantage of the gateway method is the relative difficultly in handling packet loss and periods without connectivity. As the MPEG-TS data is interleaved, the loss or out of order reception of a single bundle, if undetected, could potentially cause significant interruption to the outputted signal. 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 buffer size, a user can reduce the visual impact of out-of-order packets. As a result, the UDP gateway requires a series of internal buffers: The first is for reordering the received bundles, while the second extracts the individual packets and handles the output.

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. Any packet which contains a timestamp is less than the current “running” timestamp is assumed to have arrived out-of-order and is archived. The disadvantage of this system is its uniqueness. 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, although the author has not yet performed this work. In addition, the encoder must be based on a codec which supports frame-based output, such as H.264, motion JPEG2000, and H.265.

The gateway, running over LTP with a 2 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.

Our native transmission system running over LTP with a 25 frame buffer (one second), with an 8mbps encoding bitrate has been found to be resistant to a 10% bit error rate without visual degradation. This is well above the CCSDS recommendation for packet loss and jitter.

3D Point clouds:

It is foreseen by the author that many cameras of interest to astronauts and flight control teams may not strictly function in 2D. For example, structured light scanners or Time of Flight cameras output a 3D image containing a point cloud representation, of depth between an object and the lens, while an integrated camera processes the color imagery. Various algorithms will merge the color data onto the point data, providing a single cloud. The design of these algorithms is out of scope of this paper. A standard stereo camera may also be of interest, where the stereo imagery has to be reprocessed to 3D on-board for autonomous navigation systems. Since the 3D processing has already been performed, it can be directly transmitted to earth allowing the control teams a better understanding of how the robot "sees" the world around it.

This type of data can be easily represented as a 3D point cloud, which is an array of voxel elements, each containing additional parameters (such as color, normals, etc). While this format is easily serializable, it is also quite verbose and as a result, large. The problem of point-cloud compression has been well studied, and we have combined various aspects of the major compression schemes in order to provide a robust system for that type of compression.

The simplest point cloud compression is an outlier remover, where all points where are outside of a threshold are deleted. This method is not computationally intensive, though the data reduction can be minimal.

The final two methods utilize octree data structures[10]. Octrees provide a 3D environment for the subdivision of 3D clouds, where the entire scene is bounded in a single cube. That cube is then subdivided into smaller regions which contain voxels. Each of these regions are subdivided using the same methodology, until there is either a 1:1 relationship between regions and voxels or some threshold has been reached: either a threshold of regions has been reached, or that no region contains less than a given number of voxels. These trees may be used to create a new point cloud containing 1 voxel per region, therefore accomplishing a significant reduction in cloud density. Alternately, the octree itself may be serialized and transmitted with ancillary data indicating the color of said voxels[11].

Kammerl, et al have shown a method for the further compression of octrees by utilizing differential encoding[12]. Here, each octree is byte-encoded and the resulting tree is XORed against the last output. Finally, the output of this is sent to a range encoder. The resulting represents a highly compressed version of a temporal cloud, which can easily be reconstructed by the decoder. For greater spatial resolution, a set of coefficients may be encoded. These coefficients include a delta position for other voxels within the tree.

OPERATIONAL IMPLEMENTATION FOR MANNED SPACEFLIGHT:

In this section, several potential operational scenarios shall be investigated. The usage of DTN, as well as video-over-DTN will be explained for each, and the end-result will be provided.

ISS Operations:

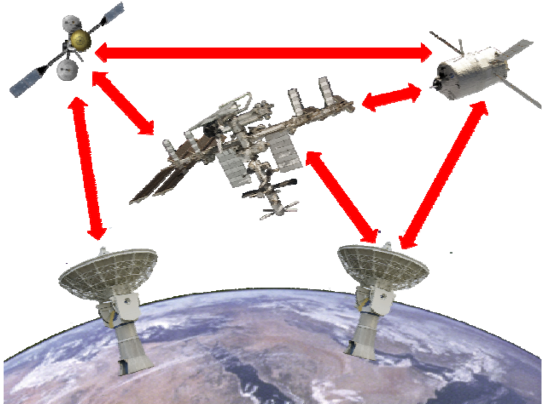

Fig. 1: Stylized overview of ISS connectivity options

The International Space Station (ISS) is connected to the earth via several downlinks, including the American Tracking and Data Relay (TDRS) satellite system, as well as various Russian systems, the Japanese Data Relay Test Satellite (DRTS), and eventually, the European Data Relay System (EDRS). These systems connect through a combination of VHF, S, Ku, and Ka bands, and each have their own protocols, maximum throughput, latency, AOS and LOS periods, and error rates, which may be different across different frequencies (*eg:* Ku and S bands). This provides a significant challenge to the mission planning teams who must decide which downlink should be available for a given activity.

Video is used on the space station for many applications, including visiting vehicle docking operations, scientific payloads, medical investigations, public affairs, and station management. This video may be generated from Russian, American, Japanese, or European assets. Some of the many encoders are commercial and off the shelf High Definition encoders while others are custom designed for the space station and designated for mission-critical activities. Finally, depending on the details of the activity, the video may be considered to be mission-critical and/or designated for medical or crew comfort uses.

Day To Day Operations:

Wide adoption of DTN on the station could help alleviate some of the above issues. A contact plan for the well-known AOS and LOS periods could be uploaded to the station DTN gateways, which would then decide the best method for transfer to earth. In the case that none of the currently available downlinks satisfy the requirements for packet transfer (due to available throughput, latency, etc), the data may be buffered on the gateway until such time that a suitable connection is available. Additionally, opportunistic connectivity may be utilized, where the gateway will attempt to connect via a link even if the exact schedule is unknown within the contact plan.

Video generated by the integrated encoders could be encoded using the CCSDS standards (8 MBPS H.264), and framed as NAL, before being transferred via DTN. Meanwhile, the COTS encoders outputting MPEG Transport Streams could be encapsulated utilizing the transparent gateway and transferred via DTN. On the ground, the streams could be decapsulated and decoded in real-time, while being simultaneously archived for later replay. Any data generated during LOS periods would eventually be downlinked and recombined in the archive. As such, once all LOS data has been transmitted, the archived copy would be a full copy of all video generated, including LOS and AOS.

Visiting Vehicles:

The ISS provides multiple systems in order to communicate directly with visiting spacecraft in order to provide monitoring and control capabilities. In addition, individual spacecraft likely possess additional links using the systems mentioned above or others. These links are used for telemetry and telecommand as well as video originating from the vehicle. For safety reasons, much of this video is considered mission critical and should be made available to the ISS crew as well as the visiting vehicle control center,

DTN provides several advantages to this type of situation. The first is the multicast capability provided by ION[13]. By utilizing this capability, video originating from the visiting vehicle can be transmitted to the ISS as well as the various control centers, all of whom have subscribed to the correct multicast endpoints. Docking is considered to be a mission-critical activity, so the video priority can be adjusted accordingly. Finally, as with general ISS operations, the vehicle can determine the best available connection in order to dispatch the required video and telemetry.

Some spacecraft include some form of depth camera, such as the DragonEye by SpaceX or LIRIS, an experiment flown on ESA's ATV-5 mission[14]. Such systems provide valuable navigational input for use in the on-board guidance and docking system, and are typically composed of an infrared camera as well as some type of distance sensor which generates distance data, either as a point cloud or a disparity map. This sensor can be based on a mechanical scan head, ToF, or a stereo camera. From the transmission standpoint, the only requirement is that the sensor can generate a 3D point cloud from this data, which can then be transmitted. As DTN allows for each bundle to have a distinctive priority level, the operational teams can dynamically reprioritize the point cloud data relative to the camera data. An additional operational constraint may potentially be the capabilities of the display computer system on the ISS, which may not have the computing power to display the point-cloud. In such a case, the point-cloud data may be sent directly from the vehicle to the ground, while the camera data is sent from the vehicle to ISS, before finally reaching the ground. This example displays the importance of proper synchronization of the video streams, which is a topic for further work.

Deep-Space Applications:

While the ISS largely maintains a regular connection schedule which an ample amount of bandwidth, this is rarely the case with deep-space missions. DTN is of an even more significant advantage to such missions.

Pre-landing Investigation:

Just as the Apollo missions relied on the Ranger and Surveyor programs in order to provide an initial insight in regards to the lunar surface, future mars missions will require knowledge of the Martian surface. This work has already begun, utilizing a wide array of satellites and landers which are currently (or have been) working around mars. It is obviously cost-effective to utilize these satellites as a relay point for communication between the manned spacecraft and the various ground control teams. In addition, the imaging instruments on rovers and landers may be transmitted directly via this relay network to the manned spacecraft, allowing the astronauts to maintain an understanding of the environment which awaits them. It is important to note that the various scientific and operational teams involved will have extremely varied requirements, so it is important to provide a balance between the requirements for all teams. While this is partially a task for the mission planners, we believe that a well-designed DTN network can simplify those constraints. For this example, we shall say that the lander planning teams require a 3D point cloud of the landing environment, while the lander flight team would like a standard video of the landing from the rover, as well as the lander. The astronauts in the lander would also like the point cloud. Using the existing DSN network, this is difficult, as it would use a disproportionate amount of uplink bandwidth.

In this case, the rover would transmit two DTN streams, one of the 3D point-cloud and sent at an extremely high priority and with multicast. The second stream contains the video and is sent at a lower priority. This data would be sent to an orbiting relay satellite, which would bundle it for further transmission to the earth. In addition, as the astronauts in the lander have subscribed for the multicast stream from the point cloud, the relay satellite would transmit it to them. The manned lander would transmit a single stream, containing the landing video generated on-board. In the case of an interruption of connectivity between the relay and earth, all data would be buffered for further transmission. More importantly, if there is a reduction in the available throughput between the relay and earth, the DTN stack contained within the relay satellite will intelligently prioritize the data. In this case, the point cloud would be prioritized, and if required, the bundles which contain frames from the other streams would be dropped or transmitted later.

To the users on the earth, this behavior would be transparent other than the occasional dropped frame.

CONCLUSION

In this paper, several techniques of great benefit to manned spaceflight have been shown, and while each has their own strengths, all of them present advantages to the current method of space-ground communication. As shown in the Operation Implementation section, a combination of these techniques increases their benefits while minimizing the potential disadvantages. As future space networks increase in intelligence, new and previously inconceivable capabilities will become available, with new degrees of flexibility available.

ACKNOWLEDGEMENTS

The author wishes to acknowledge Dr. Osvaldo Peinado of DLR, who was of great benefit in supporting the DTN project. In addition, a special thanks goes to Austin Gosling, Ilinca Ioanid, and Patrick Brun of INSYEN, who were the first people to hear of many of these ideas. An additional thanks goes out to Ivan Antonov, Alexey Glazkov, and Anton Belov of RosCosmos who provided excellent testing support for the transparent gateway. Finally, a final thanks goes out to CCSDS and all of the ISS video teams and their collective knowledge, without which we would still be arguing over the benefits of video in general.

---

[1] Burleigh, S., “Bundle Protocol Extended Class Of Service”, internet-draft, [online] https://tools.ietf.org/html/draft-irtf-dtnrg-ecos-05

[2] ION DTN Download [online] http://sourceforge.net/projects/ion-dtn/

[3] Licklider Transmission Protocol (LTP) for CCSDS. Issue 1. Recommendation for Space Data System Standards (Blue Book), CCSDS 734.1-B-1. Washington, D.C.: CCSDS, May 2015

[4] Generic coding of moving pictures and associated audio information – Part 1: Systems, ISO Standard, ISO/IEC 13818-1, Geneva: ISO, July 2015

[5] Section 3.6 in CCSDS Digital Motion Imagery Blue Book (pg 3-11) , Recommended Standard, CCSDS 766.1-B-1, Washington DC.:CCSDS, June 2015

[6] Information technology – Coding of audio-visual objects – Part 10:Advanced Video Coding, International Standard, ISO/IEC 14496-10:2014.Geneva: ISO, September 2014

[7] Mp4box Download [online] https://gpac.wp.mines-telecom.fr/mp4box/

[8] FFmpeg Download [online] https://www.ffmpeg.org/

[9] VLC Download [online] http://www.videolan.org/vlc/index.html

[10] Meagher, D., “Octree Encoding: A New Technique for the Representation, Manipulation and Display of Arbitrary 3-D Objects by Computer”, Technical Report, Rensselaer Polytechnic Institute, report number: IPL-TR-80-111

[11] Schnabel, R.; Klein, R.: “Octree-based Point-Cloud Compression”, in proceedings of *Eurographics Symposium On Point-based Graphics,* July 2006

[12] Kammerl, J.; Blodow, N.; Rusu, R.B.; Gedikli, S.; Beetz, M.; Steinbach, E., "Real-time compression of point cloud streams," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on* , vol., no., pp.778-785, 14-18 May 2012

[13] Burleigh, S., “CBHE-Compatible Bundle Multicast”, internet-draft, [online] https://tools.ietf.org/html/draft-burleigh-dtnrg-imc-00

[14] Kolb, F et al, “The LIRIS-2 3D imaging LIDAR on ATV-5”, in ASTRA 2015 proceedings