On the Performance of Packet Layer Coding for Delay Tolerant Data Streaming in Deep Space

Xiaoyu Zhu(✉), Dongxu Hou, Kanglian Zhao, Wenfeng Li, and Yuan Fang

School of Electronic Science and Engineering, Nanjing University, Nanjing 210023, People's Republic of China
mf1723072@smail.nju.edu.cn, kaldon@163.com, {zhaokanglian, yfang}@nju.edu.cn, leewf_cn@hotmail.com

Abstract. This paper presents a research on the performance of packet layer coding for space streaming, which integrates Low Density Parity Check (LDPC) Code into space streaming service. By implementing LDPC code into higher layers, such as bundle layer or transport layer, the space streaming service can achieve stronger error correction capability, which improves the quality of experience of space streaming service. To evaluate the performance of packet layer coding for space streaming, emulation of data streaming in space are conducted. The results of the experiments show that compared with the original transmission scheme, space streaming service with packet layer coding has obvious improvements.

Keywords: Delay/disruption tolerant network · Packet layer coding · Low Density Parity Check Code · Space streaming · Streaming data delivery · Deep space

1 Introduction

With the development of aerospace technology, more and more manned deep space program will be conducted in the future. Compared with the traditional deep space communication service, data transmission of manned space program has higher requirements, especially in space streaming service. In manned exploration missions, the proportion of streaming data transmission will be much larger than that of traditional unmanned programs which require mostly reliable data transmission. However, because of the severe communication conditions in the deep space environment, the space streaming service faces many problems. The characteristics of deep space networks such as long delay and high bit error rate lengthen the delivery time, reduce the transmission success rate, and reduce the transmission efficiency, which greatly affects the quality of streaming service. Therefore, it is getting more and more attention of researchers on how to improve the performance of streaming data delivery in deep space.

Q. Wu et al. (Eds.): WiSATS 2020, LNICST 357, pp. 58–68, 2021.
https://doi.org/10.1007/978-3-030-69069-4_6

To solve this problem, the issue of data streaming in DTN [1–4] (Delay/disruptive tolerant networking) has been studied. In [5], erasure codes were implemented on video streaming system over lossy networks. Furthermore, in [6], the notion of 'substitutable content summary frames' has been introduced. An additional 'summary frame' is added to the original stream to improve the performance of video transmission even in bad condition. And in [7], another transport level mechanism based on coding scheme called Tetrys was proposed. An on-the-fly coding scheme is applied to provide the reliability. In 2013, a new space streaming transmission scheme called Bundle Streaming Service(BSS) was implemented in DTN. BSS can ensure the in-order delivery by a forwarding strategy and guarantee the integrity of the data by retransmitting the lost data. In addition, the Consultative Committee for Space Data Systems (CCSDS) suggested a new idea for video applications in deep space by implementing Real-Time Transport Protocol (RTP) over DTN [9]. However, because of the severe communication conditions, BSS and RTP over DTN both have shortcomings in deep service, such as long stream delivery time and high packet loss rate. Therefore, Forward Error Correction (FEC) codes, such as Reed-Solomon (RS) codes, has been combined with BSS in [10].

This paper conducts further research on the performance of packet layer coding for streaming transmission service in deep Space, by integrating LDPC Code [11] into BSS and RTP over DTN.

The rest of the paper is organized as follows. In Sect. 2, we will introduce the related work briefly. In Sect. 3, the design and implementation of packet layer coding for space streaming are shown in details. In Sect. 4, a typical experiment is conducted and we will analyze the experimental results. At last, a conclusion is given in Sect. 5.

2 Related Work

2.1 Overview of BSS

In order to achieve the goal of streaming data delivery in deep space, BSS was implemented in DTN, which can ensure the in-order delivery in real-time display, and provide the playback function. The stark of BSS is shown in Fig. 1.

The focus of BSS is to balance the relationship between unreliable and reliable transmission. BSS does not adopt a reliable transmission protocol for all data, but only uses the reliable transmission protocol to retransmit the lost packets, and most of the data is sent using an unreliable transmission protocol to guarantee the minimum transmission delay. BSS provides two kinds of transmission services, one of them is best effort service and another is reliable service. First of all, when the data is sent for the first time, it will be transmitted through the best effort service by UDP or green LTP to ensure that the data is sent to the receiving side as soon as possible. During the communication process, if some packets are lost, the reliable service will retransmit the lost packets by TCP or red LTP to ensure the reliability of transmission.

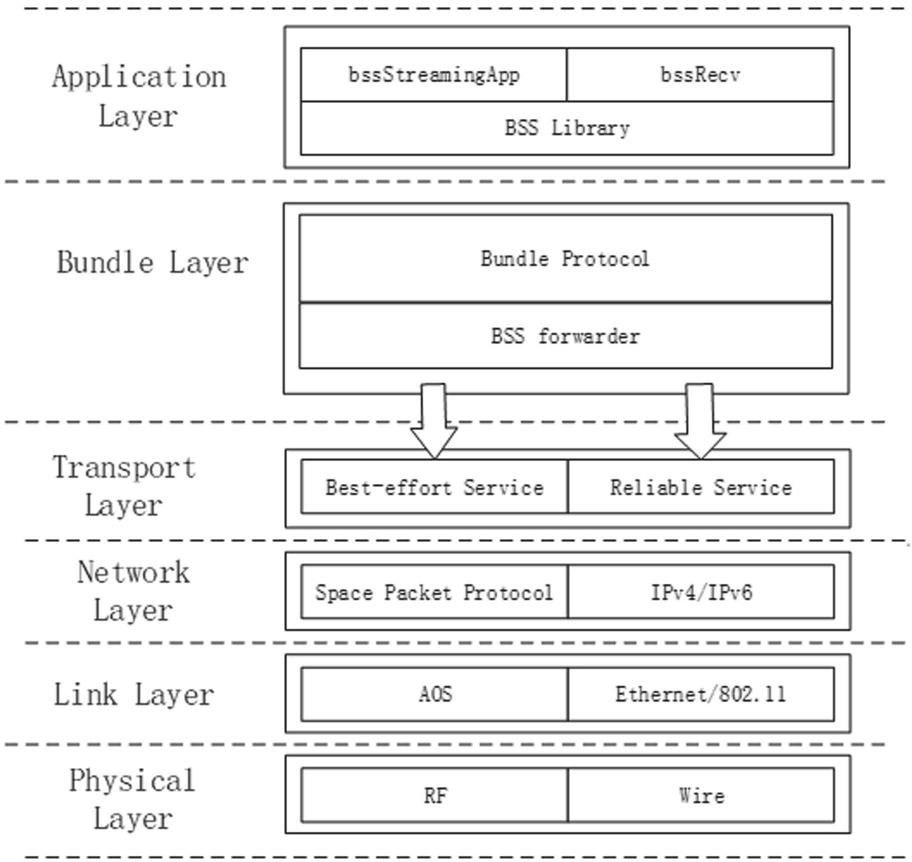


Fig 1. The stack of BSS

2.2 Overview of RTP Over DTN

Besides BSS, a new space streaming transmission scheme called RTP over DTN was proposed by CCSDS in 2018. RTP over DTN is a method of video over DTN using RTP, which can ensure higher quality of video transmission in deep space.

The Real-Time Transport Protocol (RTP) [12] provides a lightweight packet format for the transmission of media-related payloads over IP-based networks. RTP is designed around a fixed 12-byte header, along with a variable-length header extension field, as can be seen in Fig. 2. 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. 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.

V=2	P	X	CC	M	PT	Sequence Number
timestamp						
Synchronization source (SSRC) identifier						
Contributing source (CSRC) identifier						

Fig 2. The header of RTP [12].

In order to make it apply to deep space, RTP should be implemented over Bundle Protocol (BP) [13]. RTP may be encapsulated into DTN bundles with minimal modification, instead treating the entirety of the RTP packet as a singular bundle. Considering RTP packets have arbitrary sizes, so it can be concatenated, and the following rules should be followed:

- Concatenated packets must have the same marker value. If the marker value changes, a new bundle should be started.
- All RTP payloads in a concatenated bundle must belong to the same media stream & have the same timestamp.
- After the first RTP packet, all headers can be stripped.

Figure 3 shows an example of a concatenated RTP bundle. The state of the Marker bit is shown, while the sequence counters and timestamps are omitted.

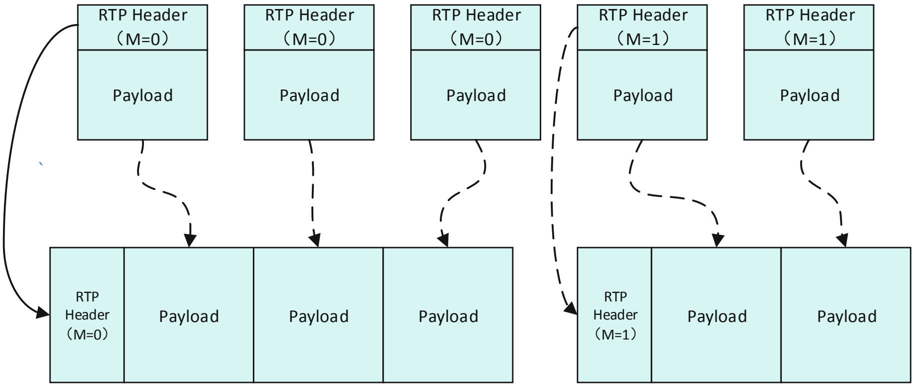


Fig 3. RTP concatenation

3 Packet Layer Coding for Space Streaming

As mentioned above, both of BSS and RTP over DTN have problem with the poor communication conditions in deep space, especially the high bit error rate, which may seriously damage the quality of space streaming service.

To solve this problem, packet layer coding is applied to space streaming in this paper. The design of packet layer coding for space streaming has been shown in Fig. 4. Note that LDPC code is performed at the bundle layer because we believe that implementing the FEC codes at the upper layer has stronger error correction capabilities by packet layer coding compared with the traditional error correction coding scheme at the data link layer. The logical process of the packet layer coding is shown in Fig. 5.

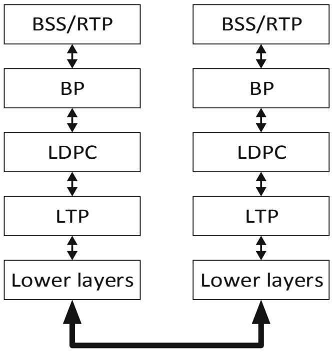


Fig 4. Design of packet layer coding for space streaming [14].

The Staircase LDPC codes, where the parity check matrix could be built by the decoder itself, on the spot, with a quick pseudo-random algorithm, is used in packet layer coding for space streaming. The library that uses these code is OpenFEC, developed by researchers of several French research institutes.

Firstly, several bundles converge into a data block. Then each data block would be cut into k LTP segments. The k LTP segments are sent to the LDPC encoding process, the encoding process includes the following steps:

a. Matrix filling: K LTP segments are passed to encoding process; each segment is an information symbol (Info packets in the figure); they are written as rows of an N-row matrix (the N-symbol codeword, or coding matrix).
b. Matrix encoding: M = N-K redundancy symbols (parity packets in the figure) are added in the last M row of the matrix. The code rate Rc = K/N represents the amount of information per codeword symbol. The lower the code rate, the higher the amount of redundancy introduced.
c. Matrix passing: the N rows are passed one-by-one to lower layers;

On the receiver side, the decoding process includes the following steps:

a. Matrix filling: Let L be the number of packet lost; N-L UDP datagrams arrive; their payload is read and written in an N row matrix (the N codeword at receiver side) leaving gaps (i.e. rows filled by zeros) in correspondence of missing symbols.

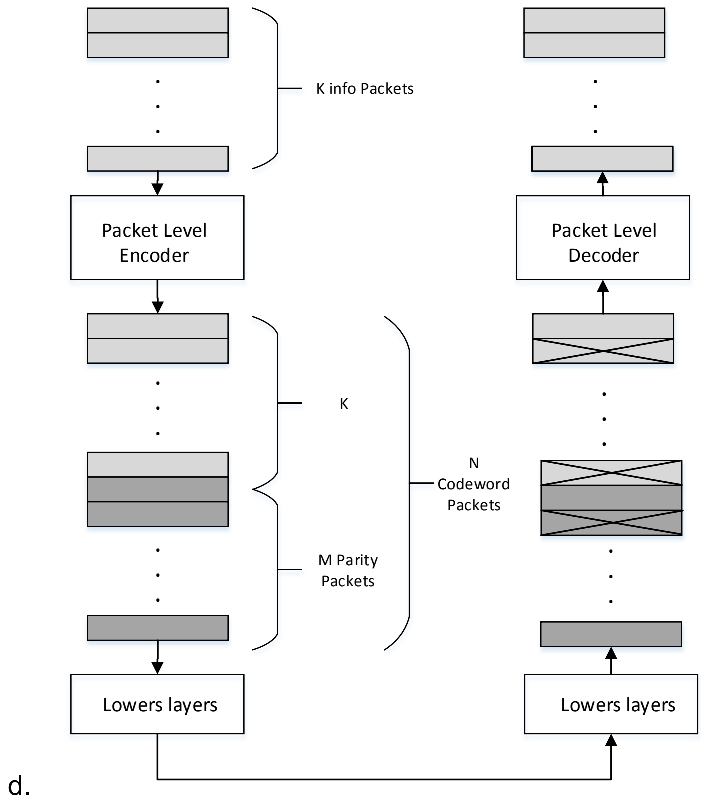


Fig 5. Logical process of the packet layer coding

b. Matrix decoding: The aim of the decoder is to extract the K information symbols (i.e. basically our K LTP segments) from the N-L received symbols. If the decoding is successful, all the first K rows are eventually filled. Put in other words, as the code is systematic, the missing info symbols (two in the figure) are recovered by exploiting the redundancy symbols. Thus, to have a success it is necessary to receive at least K of the N symbols transmitted, whatever they are (information or redundancy);
c. Matrix passing: the first K rows of the matrix, i.e. the info symbols, are read and passed one-by-one to upper layer.

4 Experiment Results

In this experiment, in order to simulate the real deep space communication environment, a two-node experimental scene is constructed, the experimental scene is shown in the Fig. 6.

The Host1 and Host2, represent the sender and receiver respectively. Both of them are equipped with the ION v3.6.1b [15]. And the last computer played the role of channel emulator, which can change the propagation delays and channel error rates. More details of the experimental factors and configuration are shown in Table 1.

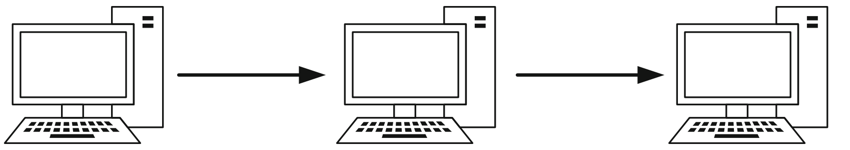


Fig 6. Emulated data streaming communication experiment

Table 1. Experimental configuration

Experimental factors	Setting/Values
DTN implementation	ION3.6.1b
Linux version	Ubuntu 16.0.4
Each bundle size(bytes)	11,844
Code rate	0.7
Propagation delay(s)	1–15
BER	1×10^{-7}, 5×10^{-7}, 1×10^{-6}, 5×10^{-6}
Streaming data size	20 s Stream data
Sample size	10 Repetitive runs

4.1 Performance on Packet Layer Coding for BSS

The following two metrics [8] were introduced to evaluate the performance on packet layer coding for BSS: the stream delivery time (SDT) and the end user's display efficiency (EDE). SDT represents the time of successful trans-mission of all the data to the receiver, and EDE represents the proportion of data successfully transferred for the first time.

In the first part of the experiment, the SDT is used to evaluate the transmission speed of streaming data delivery.

Figure 7 shows the effect of propagation delay on SDT at the same channel bit error rates. According to the results, BSS with LDPC performs better than original BSS in different propagation delay, which means no matter how long the delay is, BSS with LDPC can deliver all the data in a shorter time.

Figure 8 shows the results at different channel bit error rates. When the delay is 1 s, the BSS with LDPC can delivery all the data in 60 s even in bad channel conditions (BER = 5 × 10 – 6), while original BSS takes more time.

In the EDE evaluation, the result in Fig. 9 shows BSS with LDPC can display more streaming data in the receiver than original BSS.

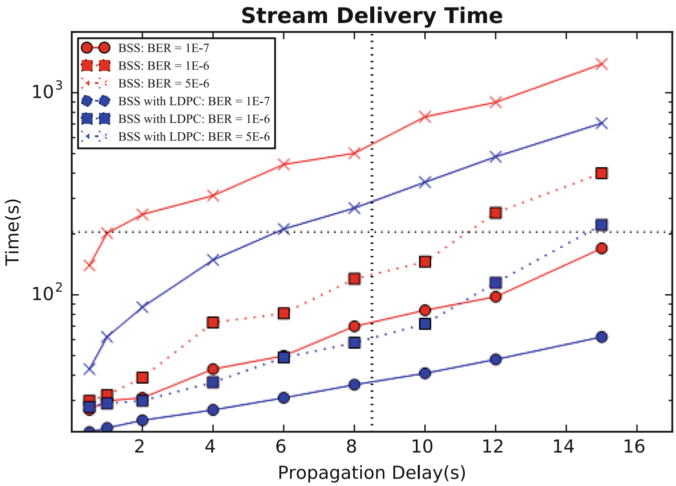


Fig. 7. Experimental result of LDPC-BSS based on SDT respect to delay.

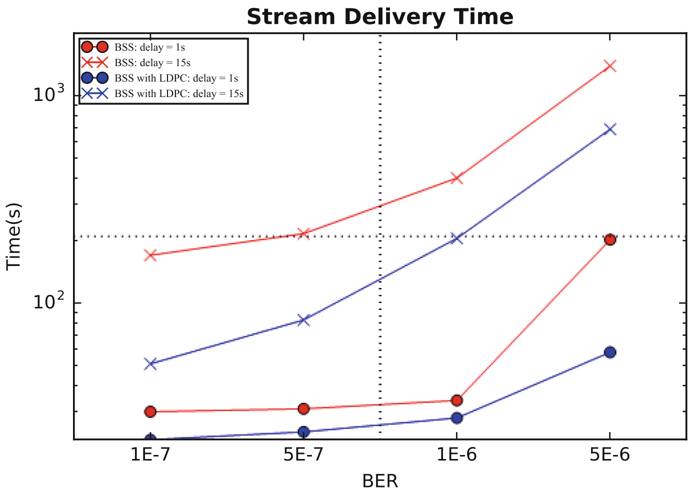
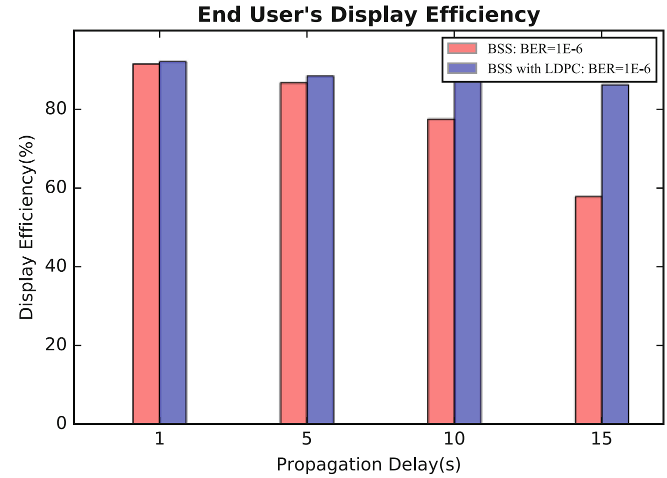


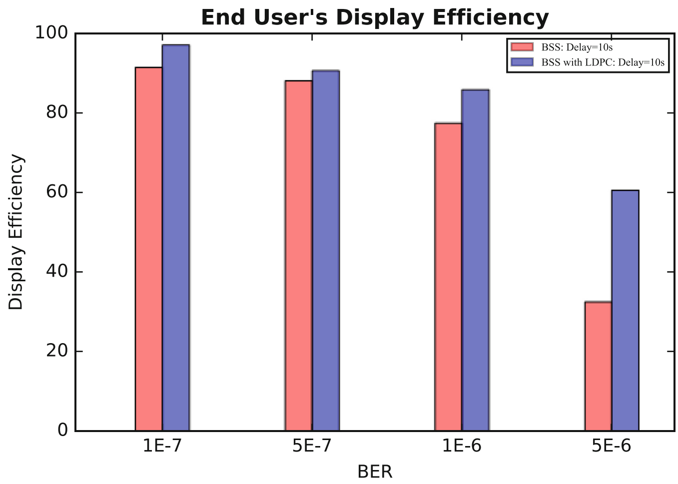
Fig 8. Experimental result of LDPC-BSS based on SDT respect to BER.

4.2 Performance on Packet Layer Coding for RTP Over DTN

The performance on packet layer coding for RTP over DTN is focused on the number of lost packets respect to propagation delay and BER. Table 2, Table 3 show the number of lost packets within the transmission by RTP over DTN with LDPC and RTP over DTN without LDPC.



(a)



(b)

Fig. 9 Experimental result of LDPC-BSS based on EDE respect to (a) delay (b) BER

It is apparent that packet layer coding can improve the performance of space streaming, including BSS and RTP over DTN. There are some reasons. Compared with original BSS, packet layer coding provides stronger error correction capability, which can avoid retransmission and get stable EDE. When it comes to RTP over DTN, packet layer coding can provide higher quality service by reducing the number of lost packets.

Table 2 Number of lost packets respect to propagation delay (BER = 1E-6)

Delay /s	Lost packets without LDPC	Lost packets with LDPC
1	30	29
6	47	41
10	80	45
15	150	52

Table 3 Number of lost packets respect to BER (Delay = 10 s)

BER/10^{-7}	Lost packets without LDPC	Lost packets with LDPC
1	30	9
5	42	28
10	80	45
50	240	130

5 Conclusion

Based on the emulations in this work, we evaluated the contribution of packet layer coding to streaming data delivery for future deep space applications. And the results of the experiments strongly prove that space streaming with packet layer coding has great advantages for the original space streaming service, no matter in terms of BSS or RTP over DTN. Benefited from stronger error correction capability, packet layer coding ensures the reliability of streaming data delivery and improve the quality of streaming service.

References

1. Cerf, V., Burleigh, S., et al.: Delay-Tolerant Networking Architecture, RFC 4838, IETF (2007)
2. Farrell, S., Cahill, V., Geraghty, D., et al.: When TCP breaks: delay-and disruption-tolerant networking. IEEE Internet Comput. **10**(4), 72–78 (2006)
3. Fall, K.: A delay-tolerant network architecture for challenged internets. Sigcomm03 **33**(4), 27–34 (2003)
4. Burleigh, S., Hooke, A., Torgerson, L., et al.: Delay-tolerant networking: an approach to interplanetary Internet. IEEE Commun. Mag. **41**(6), 128–136 (2003)
5. Zhang, X.J., Peng, X.H.: A testbed of erasure coding on video streaming system over lossy networks. In: International Symposium on Communications and Information Technologies, pp. 535–540. IEEE (2007)
6. Liu, T., Nelakuditi, S.: Disruption-tolerant content-aware video streaming. In: MULTIMEDIA 04: Proceedings of the 12th Annual ACM International Conference on Multimedia, pp. 420–423. New York (2004)

7. Tournoux, P.U., Lochin, E., Leguay, J., et al.: Robust streaming in delay tolerant networks. In: IEEE International Conference on Communications, pp.1–5. IEEE (2010)
8. Sotirios-Angelos, L., Scott, C.B., Vassilis, T.: Bundle streaming service: design, implemetation and performance evaluation. Trans. Emerging Tel. Tech. **26**, 905–917 (2015)
9. Draft Recommendation for Space Data System Standards, Real Time Protocol over Delay Tolerant Networking For Video Applications, White Book, February 2018, www.ccsds.org
10. Qian, Q., Zhu, X., Zhao, K., Li, W.: Integration of reed–solomon codes to bundle streaming service (BSS) for deep space Communications. In: Liang, Q., Liu, X., Na, Z., Wang, W., Jiasong, M., Zhang, B. (eds.) Communications, Signal Processing, and Systems: Proceedings of the 2018 CSPS Volume I: Communications, pp. 207–215. Springer , Singapore (2019). https://doi.org/10.1007/978-981-13-6264-4_25
11. Gallager, R.G.: Low-Density Parity-Check Codes . MIT Press, Cambridge (1963)
12. Schulzrinne, H., Casner, S., Frederick, R., et al.: RTP: a transport protocol for real-time applications. Rfc **2**(2), 459–482 (1995)
13. Burleigh, S.C.: Interplanetary Overlay Network Bundle Protocol Implementation (2011)
14. ECLSA enhancements to support the OpenFEC codec library and to take advantage of it characteristic features. https://amslaurea.unibo.it/
15. Burleigh, S.: Interplanetary Overlay Network: an Implementation of the DTN Bundle Protocol. In: Consumer Communications and Networking Conference, 2007, CCNC 2007, 4th IEEE. IEEE (2007).