

Quality of Service Extension for Delay- and Disruption-Tolerant Networking

Teresa Algarra Ulierte^{a*}, Felix Flentge^b, Andreas Timm-Giel^c

^a *Hamburg University of Technology, Germany, teresa.algarra.ulierte@tuhh.de*

^b *Directorate of Operations, European Space Agency, Germany, felix.flentge@esa.int*

^c *Hamburg University of Technology, Germany, timm-giel@tuhh.de*

* Corresponding Author

Abstract

As the number of future space missions grows, so does the need for robust and efficient communications for them. An efficient approach is Delay- and Disruption-Tolerant Networking (DTN) with the Bundle Protocol (BP), which is already in use by some space agencies and is the agreed standard for space internetworking. Nevertheless, the current BP version lacks support for Quality of Service (QoS), which is essential for ensuring reliable and timely data transfer in future missions. This improvement is especially relevant when looking at upcoming Moon and Mars missions which will have different types of data flows (housekeeping telecommand and telemetry, emergency traffic, navigation messages, science data, voice and video communication among others) with different requirements and priorities.

This paper presents a comprehensive QoS extension block for BPv7, addressing this critical gap in the current protocol specification. The proposed block enables specification of multiple QoS parameters at the bundle source, including traffic prioritization with fairness considerations, reliability options, latest-only delivery, and bundle storage classes. By providing a scalable framework that supports different priority classes, reliability requirements, and resource management strategies, this extension addresses the full spectrum of communication needs for future lunar and planetary missions. Another key contribution is the definition of a volume-based nested interleaving scheduling algorithm that balances prioritization needs with fairness, preventing data starvation for lower-priority communication flows while ensuring timely delivery of high-priority data. Simulation results demonstrate the effectiveness of this approach, showing that nested interleaving achieves a middle ground between strict prioritization and round-robin scheduling.

In summary, this novel approach to QoS in DTN networks addresses a critical gap in space communications technology and BP, paving the way for more effective mission operations where diverse data types can coexist within constrained network environments.

Keywords: Delay- and Disruption-Tolerant Networks (DTN), Bundle Protocol (BP), Space Communications, Quality of Service (QoS)

1. Introduction

Space exploration is on the cusp of a significant expansion, with numerous missions planned for the Moon, Mars, and beyond [1]. The European Space Agency (ESA), the Japan Aerospace Exploration Agency (JAXA), the National Aeronautics and Space Administration (NASA) and other space agencies worldwide are developing missions that will require robust communication systems capable of operating in highly stressed environments. These missions will generate diverse data types, from emergency communications and telemetry to scientific data and video streams, each with its own specific requirements for transmission reliability and timeliness.

Delay- and Disruption-Tolerant Networks (DTN) with Bundle Protocol (BP) has emerged as the standard for networked space communications. DTN addresses the challenges of space communications, including long propagation delays, intermittent connectivity, and heterogeneous links with varying and asymmetric data rates [2]. BP provides an efficient approach for transmitting information across DTN networks [RFC 5050], [4]. BP has already been successfully demonstrated in projects like METERON project [5], PACE [6] and deployed on the International Space Station (ISS) [7].

However, the current BP specification lacks a comprehensive Quality of Service (QoS) framework. This limits the protocol's ability to efficiently handle the aforementioned diverse communication requirements of future space missions. While previous work has demonstrated the benefits of certain QoS mechanisms for quasi-real-time communications [8], [9], [10], a more comprehensive solution is needed to address the full spectrum of QoS requirements. Therefore, this paper presents a complete QoS extension block for BP that enables the specification of different QoS parameters at the bundle source. The extension covers multiple aspects of QoS, providing a flexible framework that can adapt to the evolving needs of space communications.

The rest of this paper is organized as follows:

- **Section 2** presents DTN and BP in depth.
- **Section 3** discusses related work in the field of QoS for DTN and BP.
- **Section 4** presents the design of the QoS extension block, detailing the QoS parameters.
- Finally, **Section 5** concludes the paper and discusses future work.

2. DTN & BP

DTN resolves the space communications challenges through a store-and-forward mechanism where data moves on a hop-by-hop basis between nodes equipped with persistent storage (see Figure 1). These intermediate nodes retain the information until the next communication opportunity becomes available [11].



Figure 1: DTN Hop-by-Hop Approach

This approach enables successful data transfer even when there is no end-to-end path. Even if an end-to-end path exists, DTN can achieve either faster transmission times, or at minimum, equivalent performance (see Figure 2).

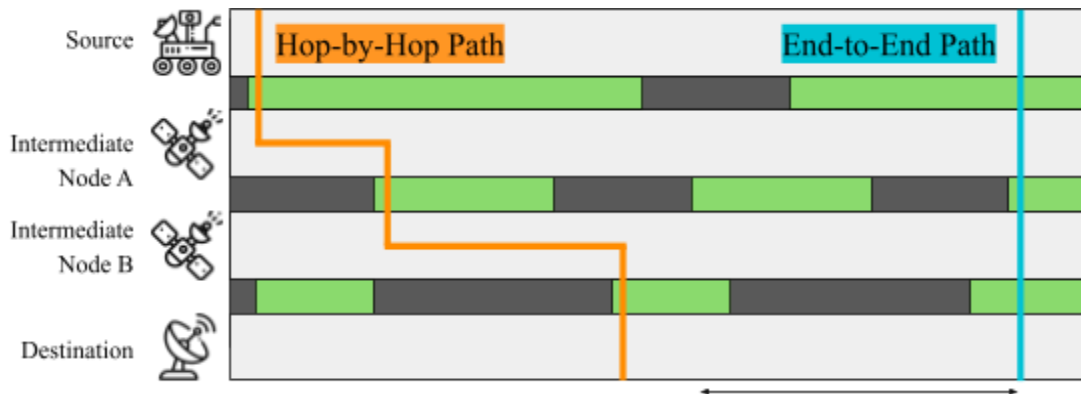


Figure 2: DTN vs. End-to-End Path Transmission Time

Within the DTN architecture, the Bundle Protocol serves as an intermediary protocol layer between the Application and the Transport layers (see Figure 3). Unlike traditional networking approaches, BP encapsulates information into self-contained "bundle" packages. Each bundle is equipped with comprehensive metadata and payload, functioning as autonomous units capable of traversing the network independently [12]. This self-sufficiency eliminates the requirement for handshakes. The results are particularly advantageous in high-latency scenarios, as it minimizes the overhead and reduces both the economic costs and long delays.

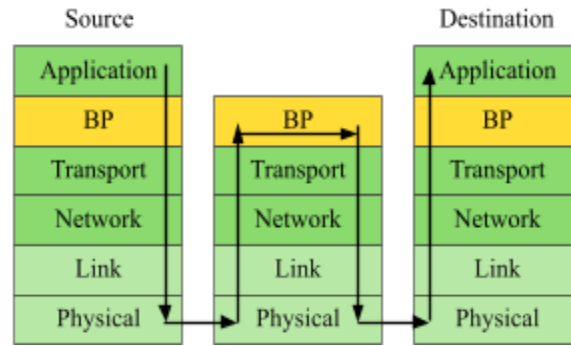


Figure 3: Bundle Protocol Layer

Bundles are made of three building blocks (see Figure 4):

- **Primary Bundle Block:** it contains all the essential information needed for the bundle to be transmitted, such as the source node ID, the destination node ID or the creation time. It cannot be modified along the transmission, and can be protected with BPsec [13].
- **Payload Block:** this block contains the Application Data Unit, i.e. the information to be delivered.
- **Extension Blocks:** these blocks are optional, and contain additional information about the bundle, provide security or processing information. They provide information such as a hop count, or Bundle Integrity Blocks protecting e.g. the primary block.

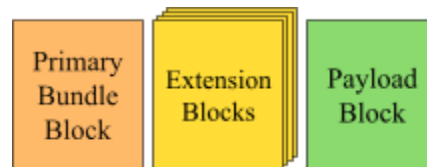


Figure 4: Bundle Structure

Although it is envisioned to be the standard for networked space communications in the near future, BP still lacks a comprehensive QoS framework to handle the diverse requirements of different types of space mission data. These limitations hinder BP's ability to efficiently handle future space missions, which will need to regularly transmit and receive data packets with different characteristics and requirements. Some examples of data types that have to be taken into account are science data, telemetry & telecommands, sensor data, video streaming or astronaut communications. Moreover, this data may be traveling through interconnected devices belonging to different networks and administrative domains.

3. Related Work

The development of QoS mechanisms for DTNs and BP has been an active area of research, with several approaches proposed to address the challenges of providing reliable and timely communications. It was addressed for the first time in RFC 4838, where two bits in the primary block defined three priority categories: expedited, normal, and bulk (referred to as "cardinal priorities") [2]. This classification system was incorporated into RFC 5050, and remained standard through BPv6, appearing in various implementations including NASA's Interplanetary Overlay Network (ION) [3].

ION subsequently incorporated the Extended Class Of Service (ECOS) extension [14], which introduced more granular priority levels and additional functionality flags to indicate critical bundles, streaming content, and reliability requirements. With the transition to BPv7 in RFC 9171, the primary block no longer contained cardinal priorities, as QoS elements were sent to a dedicated extension block [4]. This led to an updated ECOS proposal that incorporated CBOR encoding to align with BPv7 specifications [15]. These existing QoS frameworks have inherent limitations regarding scalability, both due to their structural design and restricted QoS parameter sets. Additionally, they lack mechanisms for maintaining QoS continuity when bundles traverse multiple administrative domains. Nevertheless, these implementations are used as the foundation for this and previous work.

In previous work [8], [9], the feasibility of using traffic prioritization over DTN BP was investigated, which demonstrated that traffic prioritization could significantly reduce delays for high-priority bundles. This work established the foundation for adding traffic prioritization to the QoS extension block presented in [10]. However, the simulations showed that fairness and data starvation was a problem derived from the use of traffic prioritization.

The work presented in this paper builds upon these previous efforts to provide a comprehensive QoS framework that addresses the specific requirements of future communication links, and addresses the issue of fairness in prioritized networks with limited resources.

4. QoS Extension Block for BP

This section describes the extension block design, as well as the QoS parameters included in its definition.

4.1 User QoS Extension Block (UQEB)

Building on previous work [10], the User QoS Extension Block (UQEB) allows a user to request specific QoS parameters when generating a bundle. The UQEB is added at the source node, and must not be altered during transit. To guarantee this immutability, the primary block and the UQEB should be protected through a BPSec Integrity Block (BIB) [13] (see Figure 5).

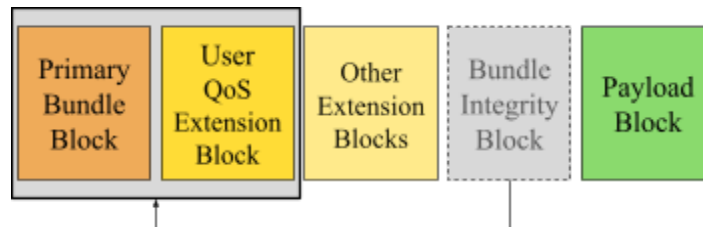


Figure 5: Placement of the UQEB in the Bundle Structure

The UQEB follows the canonical bundle block format as defined in RFC917 (see Figure 6) [4]. The block-type-specific data is a definite-length Concise Binary Object Representation (CBOR) map, where the keys represent QoS parameters and the values represent their corresponding values. Definite-length CBOR map allow for values from 0 to 2^{64-1} to be represented [16]. This work however considers the range from 0 to 23 only, making them a 1-Byte CBOR Tiny Field Encoding for compact representation. Values above 23 are hence referred to as “unassigned”.

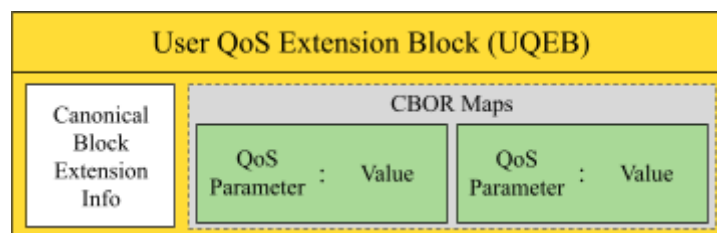


Figure 6: Structure of the UQEB

The canonical block extension information contains relevant information about the extension block, among which are the block processing control flags. These flags indicate the expected behaviour of a node towards the extension block, should this node not be able to process it. For the UQEB, the block processing control flag values are:

- The Block processing control flag Bit 0 shall be set to ‘1’ for the extension block to be replicated in every fragment in case of fragmentation.
- The Block processing control flags Bit 1, Bit 2 and Bit 4 should be set to ‘0’ to allow bundle nodes not supporting this extension block to pass it transparently without deleting or discarding it.

4.2 QoS Requirements

Based on the analysis of the communication needs and on findings from previous work, the following QoS requirements are tackled:

- **Priority:** different types of mission data have varying latency requirements, from critical emergency communications that require quasi-real-time delivery to scientific data that can tolerate longer delays. It is addressed in this work with traffic prioritization.
- **Fairness:** in scenarios with several missions, users and data sources [17], fairness in resource allocation is important to ensure that all missions can meet their communication needs to a certain extent. It is addressed in this work with a fair scheduling algorithm for the traffic prioritization system.
- **Reliability:** some mission data will require guaranteed delivery, such as commands and telemetry, whereas other data may tolerate occasional losses, such as periodic sensor readings. It is addressed in this work with a required reliability parameter.
- **Age of Information:** for certain applications, such as sensor monitoring and navigation, only the most recent data is relevant, and older data can be discarded if newer data is available. It is addressed in this work with a latest-only delivery binary flag.
- **Storage and Resource Utilization:** efficient utilization of limited persistent storage at intermediate nodes is crucial for situations in which contact windows are scarce or infrequent, and where data is at risk of being dropped due to full storage. It is addressed in this work with a dropping ordering system.

Each of these QoS parameters is assigned a “key” for the CBOR map, as can be seen in Table 1.

Table 1. QoS Parameter Keys

QoS Parameter	Key
Traffic Prioritization	0
Reliability	1
Latest-Only Delivery	2
Bundle Storage	3
Reserved for Future Use	3 - 23
Unassigned	24 - 2 ⁶⁴⁻¹

The following subsections describe each QoS key in detail.

4.2.1 Traffic Prioritization

Traffic prioritization determines the order of forwarding when a link to a next hop is available. Three main traffic priorities are defined:

- **Expedited:** bundles with this priority are considered of extreme urgency, and are granted immediate forwarding. This is done by placing them at the beginning of the queue.
- **Normal:** bundles with this priority should be delivered in a timely manner, but are not critical. They are forwarded after expedited bundles but before bulk bundles.
- **Bulk:** bundles with this priority are not time-sensitive, and can therefore allow for a higher end-to-end delay in their delivery. They are forwarded only in the absence of expedited and normal bundles.

The values assigned to each traffic type can be seen in Table 2. While expedited has only one assigned value, both normal and bulk have several assigned values each. This allows for a set of sub-priorities within each.

Table 2. Traffic Priorities

Traffic Type	Values
Expedited	0
Normal	1 - 12
Bulk	13 - 23
Unassigned	24 - 2 ⁶⁴⁻¹

As described above, the three main priorities follow a strict order. For the sub-priorities on the other hand, a fair scheduling algorithm is used, **nested interleaving**. It is a volume-based scheduling algorithm that allocates bandwidth such that each sub-priority receives twice the bandwidth of the sub-priority below it. The fairness is therefore measured in volume transmitted, not in the number of bundles.

The nested interleaving algorithm divides time into slots, and assigns a subpriority to be transmitted during each:

- The first sub-priority (P1) is assigned every second slot.
- Out of the idle slots between P1 transmissions, the second sub-priority (P2) is assigned every second slot, meaning it appears every fourth slot.
- This pattern continues for all sub-priorities, resulting in an interleaved transmission schedule where P1 appears every 2nd time slot, P2 every 4th time slot, P3 every 8th time slot, and so on (see Figures 7 and 8).

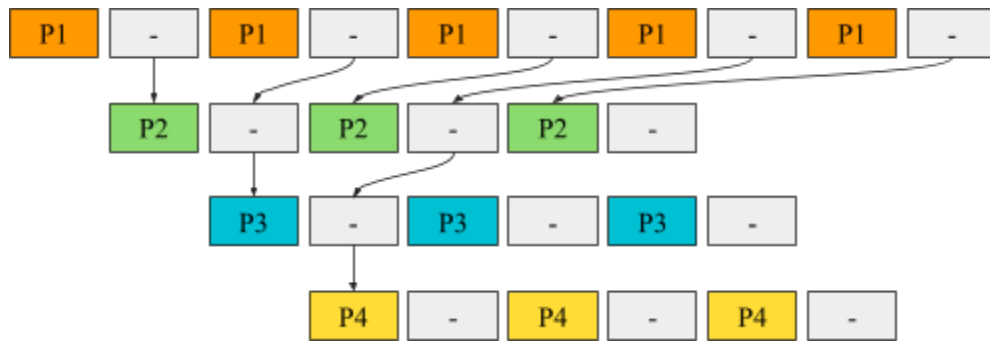


Figure 7: Nested Interleaving Slot Assignment



Figure 8: Nested Interleaving Slot Schedule

Each sub-priority then appears with a frequency of $\frac{1}{2^{sp}}$, with sp being the sub-priority number. Therefore, sub-priority 1 appears with a frequency of $\frac{1}{2^1}$, so it occupies 50% of the schedule. Sub-priority 2 appears with a frequency of $\frac{1}{2^2}$, so it occupies 25% of the schedule, sub-priority 3 appears with a frequency of $\frac{1}{2^3}$, so 12.5% of the schedule and so on. This can be seen in Figure 9, where subfigure (b) has a logarithmic vertical axis for better readability of the results.

The highest sub-priority appears in slot number 2^{n-1} , where n is the sub-priority number. In the case of normal, there are 12, which means that sub-priority number 12 appears for the first time in time slot number 2^{11} , so 2048. Bulk has 11 sub-priorities, which means that the sub-priority number 11 appears for the first time in time slot number 2^{10} , so 1024. From that point on, the schedule is periodic, meaning that it only needs to be calculated once from the first time slot until the time slot where the last sub-priority appears for the first time, and it can then be reused. This reduces the computational complexity of the algorithm. Moreover, the complexity grows linearly with the number of sub-priorities, not exponentially, as shown in Figure 10 for the case at hand with 12 sub-priorities.

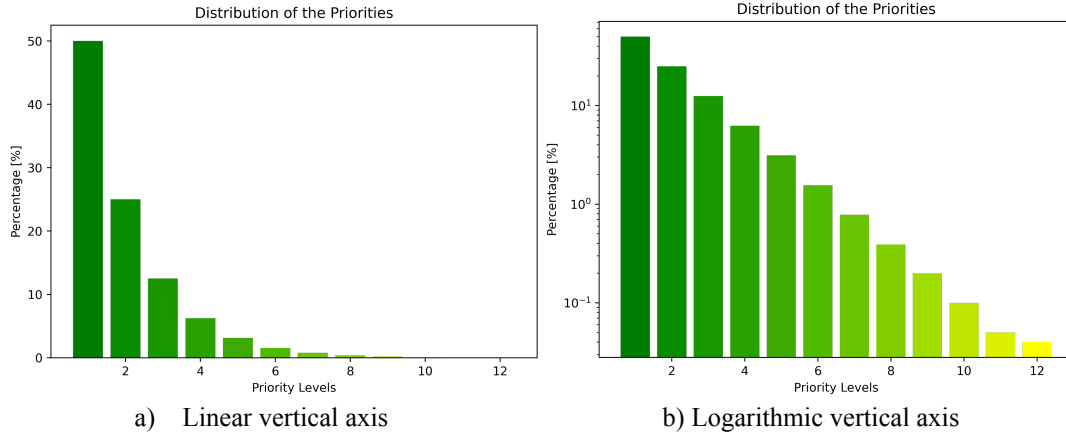


Figure 9: Nested Interleaving Schedule

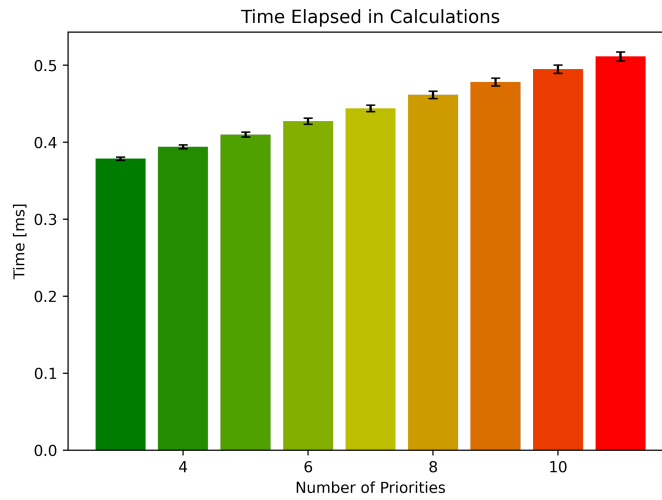


Figure 10: Computational Complexity of the Scheduling Algorithm for 12 Sub-Priorities

This scheduling algorithm ensures that higher sub-priorities get serviced twice as often as the following one, just like weighted fair queuing. Nevertheless, bundles can have varying sizes anywhere between a few kB and several MB. Therefore, the volume-based approach is implemented to guarantee that fairness is provided regardless of the bundle size. In order to do this, each assigned slot does not equate one transmitted bundle, but a certain volume of information (e.g. 256 kB, 512 kB etc.). Should a bundle be larger than the allocated volume for its transmission, it can still be serviced but it will be on “borrowed volume”. The following slot allocations for that subpriority will be skipped until the “debt” that the larger bundle has caused is settled (see Figure 11). Only complete bundles are sent, as no fragmentation is considered.



Figure 11: Volume-based Approach

In order to assess the benefits of the nested interleaving schedule, results are shown. The simulations resemble those performed in previous work, which assume a link between the Moon and Earth with both correlated and uncorrelated errors happening [8], [9], [10]. The scenario presents 4 traffic sources with different sub-priorities (P1, P2, P3 and P4) generating equally-sized bundles at the same rate and attempting to transmit through one node, meaning that a bottleneck is created. The results are shown in the form of a Cumulative Distribution Function (CDF), which depicts what percentage of bundles (y-axis) have arrived by a certain time (x-axis). The analysis of results

focuses on two performance benchmarks: first, the 2.5s end-to-end delay target established by ESA for the Moonlight project [17], and second, a 5s end-to-end delay, which is double the original latency goal and serves as an additional reference point.

Figure 12 shows the results of using **strict prioritization** for the servicing of the four sub-priorities. It can be seen that the CDF curves have a bigger gap between them. P1 achieves 73% delivered bundles by the 2.5s mark, versus 63% of P2 bundles, 53% of P3 bundles, and just 40% of P4 bundles. When looking at the 5s mark, P1 achieves 97% of delivered bundles, versus 90% of P2 bundles, 88% of P3 bundles and only 73% of P4 bundles.

The majority of P1 bundles encounter short delays, while P4 bundles take twice as much time as P1 to achieve the same percentage of delivered bundles (73% of P1 at 2.5s vs. 73% of P4 at 5s). This method implements a prioritization system without any fairness taken into consideration, which can result in increased delays for lower sub-priority bundles, and even lead to data starvation in situations with a higher amount of bundles and a more acute bottleneck.

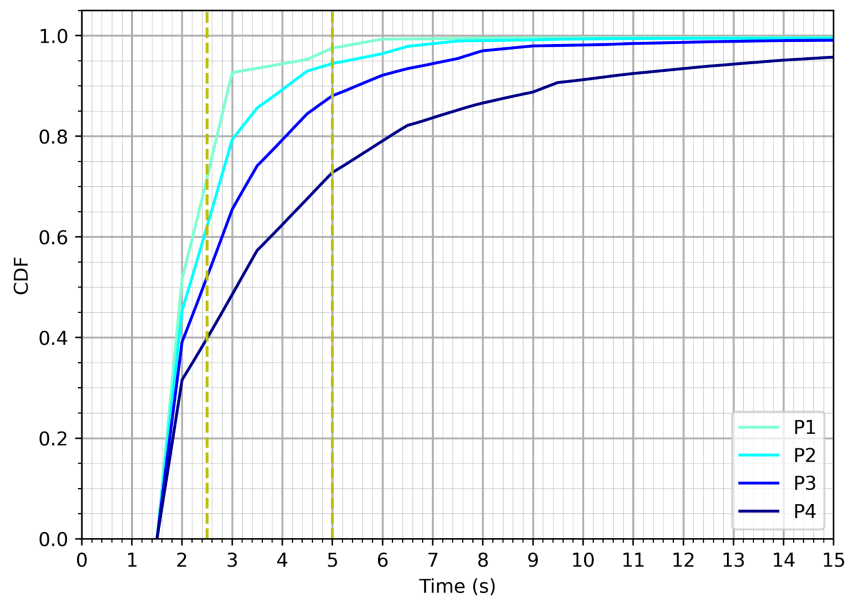


Figure 12: End-to-End Delay using Strict Prioritization as Scheduling Algorithm

Figure 13 shows the results of using **nested interleaving** as a scheduling algorithm. Since it has four sub-priorities, the scheduling algorithm follows the sequence: *P1, P2, P1, P3, P1, P2, P1, P4*. P3 and P4 appear the same amount of times since the last sub-priority appears as much as the second-to-last in order for them to add up to 100% of the slots assigned. Therefore, both P3 and P4 curves overlap, and are from now on referred to as one.

It can be seen that the CDF curves have a smaller gap between them than when using strict prioritization. P1 achieves 70% delivered bundles by the 2.5s mark, versus 60% of P2 bundles, and 45% of P3 and of P4 bundles. When looking at the 5s mark, P1 achieves 95% of delivered bundles, versus 93% of P2 bundles, and 80% of P3 and of P4 bundles.

While P1 bundles still encounter shorter delays than the lower sub-priorities, the difference between them is smaller. This is especially apparent when looking at the performance of sub-priority P4, which has improved between 5% and 7% when compared to the strict prioritization scheduling simulation. This comes at the cost of a 2-3% setback for P1 and P2. This method implements a prioritization system which takes fairness into consideration, which results in slightly increased delays for higher sub-priority bundles in exchange for lower delays for lower sub-priority bundles.

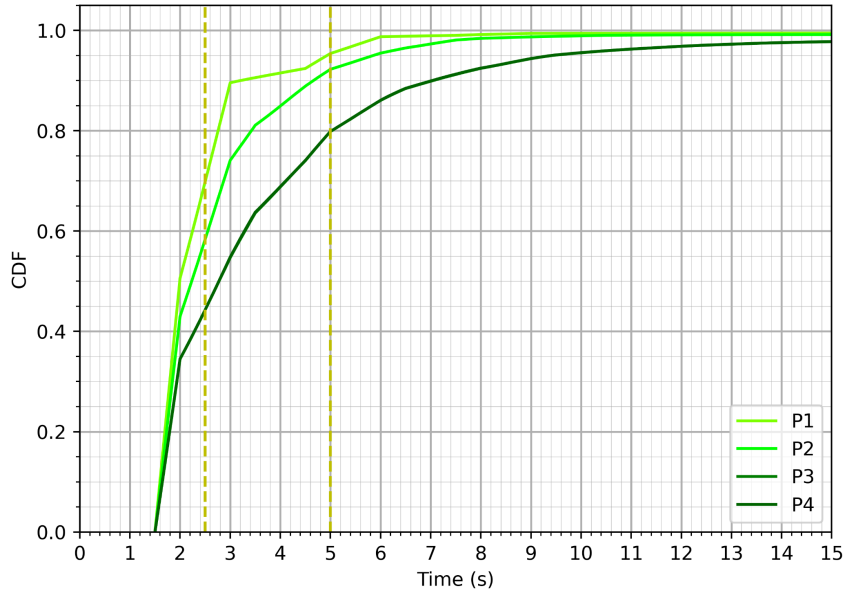


Figure 13: End-to-End Delay using Nested Interleaving as Scheduling Algorithm

Lastly, Figure 14 shows the results of using **round-robin** as a scheduling algorithm. All four sub-priorities have the same weight in the schedule and are served equally: *P1, P2, P3, P4, P1, P2, P3, P4*. Therefore, all CDF curves overlap, showing total fairness among the sub-priorities. P1 through P4 achieve just over 50% delivered bundles by the 2.5s mark, and 86% delivered bundles by the 5s mark. This method does not implement a prioritization system and is simply based on fairness, which results in shorter delays for lower sub-priority bundles at the expense of worse performance for higher sub-priority bundles.

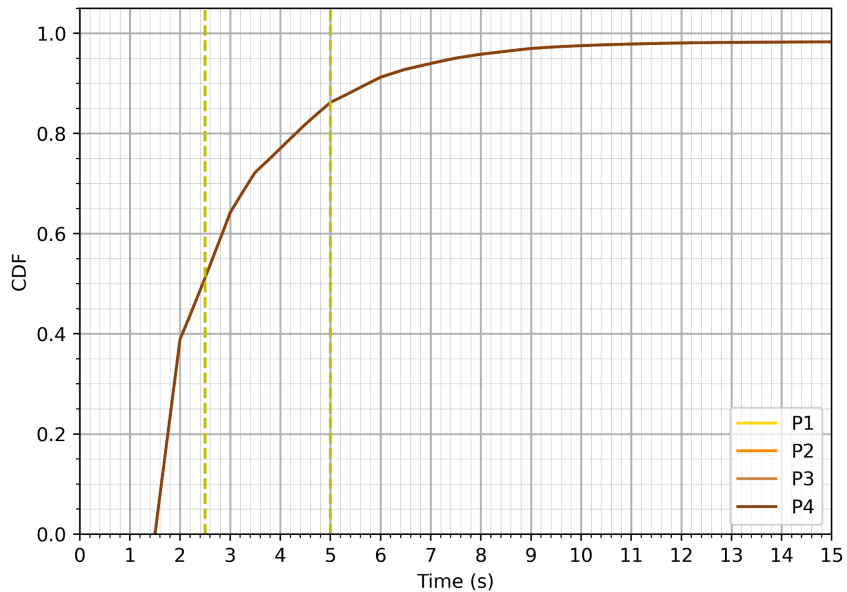


Figure 14: End-to-End Delay using Round-Robin as Scheduling Algorithm

In conclusion, these figures show the range between strict prioritization and total fairness in scheduling strategies. On the one hand, strict prioritization favors high sub-priorities, resulting in high delays and possibly data starvation for low sub-priorities. On the other hand, round-robin ensures fair treatment at the expense of higher sub-priority bundles not being able to comply with their possible delay requirements. Nested interleaving attempts to mediate between both extremes, offering a system that provides higher sub-priorities with twice as much volume as the following level, while still avoiding data starvation and extremely high delays for lower sub-priorities. Moreover, its

interleaving mechanism ensures that all sub-priorities are able to transmit their information in a timely manner, rather than waiting for higher sub-priorities to transmit for longer periods of time and risking losing contact to the next node by the time their slot comes. Since the three main priorities (expedited, normal and bulk) follow the strict priorities scheduling, it is considered that nested interleaving provides a more fair approach to sub-priorities while still maintaining the hierarchical system.

4.2.2 Reliability

Reliability indicates whether the system should ensure the reliable transmission of a bundle. This is done through the selection or configuration of a Convergence Layer Adapter (CLA) with respect to Automatic Repeat reQuest (ARQ) capabilities. Two main reliability options are defined:

- **Reliable:** the bundle should be transmitted with a convergence layer configuration that detects and re-transmits lost data. This may be achieved through the use of Transmission Control Protocol Convergence Layer (TCPCL) or Licklider Transmission Protocol (LTP) in red mode.
 - If the reliability is set to “ARQ” (value 1), the bundle must be sent in this manner. Otherwise, it should not be dispatched.
 - If the reliability is set to “ARQ if possible” (value 2), it should be sent in this manner. Nevertheless, it should still be dispatched unreliably if no reliable option is possible or available.
- **Unreliable:** the bundle should be transmitted with a convergence layer configuration that does not retransmit lost data, e.g. in cases where low latencies are preferred over reliability. This may be achieved through Encapsulation Packet Protocol Convergence Layer (EPPCL) or Space Packet Protocol Convergence Layer (SPPCL), User Datagram Protocol Convergence Layer (UDPCL), or LTP in green mode.
 - If the reliability is set to “no ARQ” (value -1), it must be sent in this manner. Otherwise, it should not be dispatched.
 - If the reliability is set to “no ARQ if possible” (value -2), it should be sent in this manner. Nevertheless, it should still be dispatched reliably if no unreliable option is possible or available.

Table 3. Traffic Priorities

Required Reliability		Values
Unreliable	Reserved for future use	-23 - -3
	No ARQ if possible	-2
	No ARQ	-1
Reliable	ARQ	1
	ARQ if possible	2
	Reserved for future use	3 - 23
Unassigned		$-2^{64-1} - -24, 24 - 2^{64-1}$

Both “ARQ if possible” and “no ARQ is possible” represent cases in which a CLA might not be available, a contact with which it can be used might not be available (such as only unidirectional links available instead of bidirectional) etc. These options provide flexibility for a dynamic and sometimes unpredictable environment.

The use of negative and positive numbers to describe “Unreliable” and “Reliable” respectively provides a wider range of values reserved for future use, since a finer granularity in this description is desired.

4.2.3 Latest-Only Delivery

Latest-only delivery prevents the transmission of aged bundles that may not be relevant anymore. It is only applicable to “batches” of bundles, meaning bundles with the same source and destination endpoint IDs. The following options are defined:

- All valid (value 0) indicates that bundles shall not be not discarded based on their age.

- Latest-only (value 1) indicates that only the newest bundle from a batch is kept and forwarded; the rest are discarded.

Table 4. Latest-Only Delivery

Latest-Only Delivery Type	Values
All Valid	0
Latest-Only	1
Reserved for future use	2 - 23
Unassigned	24 - 2 ⁶⁴⁻¹

This parameter is particularly useful for applications like sensor networks that transmit information periodically, where only the most recent data is relevant.

4.2.4 Bundle Storage

Bundle storage determines the bundle dropping order if a node's storage is full. It defines 24 classes which work in descending order:

- Bundles marked with class 24 are the first to be dropped when needed.
- Once no class 24 bundles are left, class 23 bundles are dropped.
- Class 1 bundles are the last to be dropped in case of storage scarcity.
- If several bundles of the same class need to be discarded, they should be discarded according to their remaining time to live, with the shortest time to live first.

Table 5. Bundle Storage

Storage Class	Values
Class 1	0
Class 2	1
Class 3	2
...	...
Class 24	23
Unassigned	24 - 2 ⁶⁴⁻¹

If a bundle does not have a bundle storage class assigned, it is treated as class 12 by default.

5. Conclusions

This paper has presented a comprehensive QoS extension block for Bundle Protocol that addresses the diverse communication requirements of future space missions. The extension block provides a flexible framework for specifying QoS parameters at the bundle source, including traffic prioritization, reliability options, latest-only delivery, and bundle storage classes. It is intended to publish the QoS extension block specification as a CCSDS Orange Book this year.

Moreover, a volume-based nested interleaving algorithm is proposed. It provides the traffic prioritization system with a fair manner of servicing sub-priority classes based on their delay requirements, but also avoiding extremely long delays and data starvation for lower sub-priorities. These claims are shown with the simulation results comparing nested interleaving with the benchmarks on both extremes: strict prioritization and round-robin.

The proposed QoS extensions for BP as well as the scheduling algorithm represent a significant advancement in achieving robust and efficient space communications. This enhancement addresses the communication needs of

upcoming lunar missions and supports future explorations deeper into our solar system. By overcoming the current limitations of BP, this research contributes to the development of a robust communication infrastructure capable of overcoming the obstacles tied to the highly stressed communication environment that space represents.

While the presented UQEB is standardised, network operators may use additional proprietary Network QoS Extension Blocks (NQEB) to implement Service Level Agreement (SLA) for their users. Such a block can be added and modified while traversing a network in order to add the QoS management information specified by the network. A preliminary definition of this concept has been presented in previous work [10], but a deeper study of this concept should be done in future work.

Other future work includes the analysis of the behaviour of nested interleaving's debt system when encountering varying bundle sizes, since the presented results assume equally-sized bundles in order to showcase the benefits of the system against non-volume based systems. Moreover, the QoS extension block should be validated in more representative scenarios for more realistic results [17] and, as a last step, it should be validated in real-world space missions and operations to further refine and improve its capabilities.

These future directions will continue to enhance the capabilities of BP and DTN for space communications, contributing to the success of future space exploration missions.

References

- [1] Lunar Communications Architecture Working Group, "The Future Lunar Communications Architecture". Interagency Operations Advisory Group, 2022.
- [2] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss. "RFC 4838: Delay-Tolerant Networking Architecture", 2007.
- [3] K. Scott, and S. Burleigh. "RFC 5050: Bundle protocol specification.", 2007.
- [4] S. Burleigh, K. Fall, and E. J. Birrane. "RFC 9171: Bundle Protocol Version 7.", 2022.
- [5] ESA, METERON Project. Online, last accessed: 25 March 2025. Available: https://www.esa.int/Enabling_Support/Space_Engineering_Technology/Automation_and_Robotics/METERON_Project
- [6] NASA, PACE Project. Online, last accessed: 25 March 2025. Available: <https://www.nasa.gov/technology/space-comms/nasas-near-space-network-enables-pace-climate-mission-to-phone-home/>
- [7] NASA, DTN Frequently Asked Questions Project. Online, last accessed: 25 March 2025. Available: <https://www.nasa.gov/technology/space-comms/delay-disruption-tolerant-networking-faq/>
- [8] T. Algarra Ulierte, K. Kuladinithi, A. Timm-Giel, and F. Flentge, "Lunar Communication Services: Feasibility Study on Traffic Prioritization of Quasi-Real Time Communications over DTNs," in IEEE Wireless for Space and Extreme Environments (WiSEE), 2023.
- [9] T. Algarra Ulierte, K. Kuladinithi, A. Timm-Giel, and F. Flentge, "Enabling Traffic Prioritization for Space Communications Over DTNs," IEEE Journal of Radio Frequency Identification, 2024.
- [10] T. Algarra Ulierte, K. Kuladinithi, A. Timm-Giel, and F. Flentge, "Adding Quality of Service Support to Bundle Protocol Through an Extension Block," in IEEE 10th International Conference on Space Mission Challenges for Information Technology (SMC-IT), pp. 115-124, July 2024.
- [11] F. Warthman "Delay-and disruption-tolerant networks (DTNs). A Tutorial" V. 0, Interplanetary Internet Special Interest Group, 2012.
- [12] CCSDS, "Bundle Protocol Specification," Blue Book, Issue 2, 2015.
- [13] K. McKeever, E. J. Birrane, "RFC 9172: Bundle Protocol Security (BPsec)", 2022.
- [14] S. Burleigh, "Bundle Protocol Extended Class Of Service (ECOS)", 2014. Online, last accessed: 25 March 2025. Available: <https://datatracker.ietf.org/doc/draft-irtf-dtnrg-ecos/>
- [15] S. Burleigh, and F. Templin, "Bundle Protocol Extended Class Of Service (ECOS)", 2021. Online, last accessed: 25 March 2025. Available: <https://datatracker.ietf.org/doc/html/draft-burleigh-dtn-ecos-00>
- [16] C. Bormann, "Concise Binary Object Representation (CBOR)," IETF, Tech. Rep., 2013. Online, last accessed: 25 March 2025. Available: <https://www.rfc-editor.org/rfc/rfc7049.html>
- [17] ESA, "Project Moonlight,." Online, last accessed: 25 March 2025. Available: https://www.esa.int/Applications/Telecommunications_Integrated_Applications/Lunar_satellites
- [18] C. Malnati, and F. Flentge, "Reference Scenarios for Evaluation of Disruption Tolerant Network Technologies", SpaceOps 2025 (forthcoming).