Adding Quality of Service Support to Bundle Protocol Through an Extension Block

Teresa Algarra Ulierte, Koojana Kuladinithi, Andreas Timm-Giel
Institute of Communication Networks
Hamburg University of Technology
Hamburg, Germany
Email: { teresa.algarra.ulierte | koojana.kuladinithi | timm-giel }@tuhh.de

Felix Flentge
Directorate of Operations
European Space Agency
Darmstadt, Germany
Email: felix.flentge@esa.int

***Abstract*—The Bundle Protocol (BP) was designed to address the challenges inherent in space communications. While already in use in several projects led by various space agencies, including the European Space Agency (ESA) and the National Aeronautics and Space Administration (NASA), there is a need to expand BP's capabilities, including in Quality of Service (QoS) support, an area currently lacking standardization. This document proposes a dual QoS support block for BP which facilitates the definition of QoS requirements at the source in an immutable manner while allowing dynamic adjustments by networks or subnetworks. Furthermore, preliminary results are presented, analyzing the effects of the proposed traffic prioritization system and the weighted queue management. These results show improved end-to-end delay for time-sensitive information, and a higher rate of achieved QoS requirements for all priority classes, as well as a fairer approach to network scheduling.**

***Index Terms*—Delay- and Disruption-Tolerant Networks (DTN), Bundle Protocol (BP), Space Communications, Quality of Service (QoS).**

I. INTRODUCTION

Bundle Protocol (BP) was created along with Delay- and Disruption Tolerant Networks (DTN) with the goal of overcoming the specific challenges posed by space communications. Since its initial publication in 2007 (RFC 5050) [2], BP has gone through a series of modifications until reaching its current version, BP version 7 (RFC 9171) [3]. Thanks to the advantages it offers over other alternatives, it is already being used in experiments and projects carried out by several space agencies, including the METERON project from the European Space Agency (ESA) [4], the Korea Pathfinder Lunar Orbiter (KPLO) from the Korea Aerospace Research Institute (KARI) [5] or the BP-based communication between payloads at the International Space Station (ISS) and ground nodes as an effort between multiple agencies, including ESA and the National Aeronautics and Space Administration (NASA) [6].

Nevertheless, working groups across the Internet Engineering Task Force (IETF) and Consultative Committee for Space Data Systems (CCSDS) are currently working on expanding BP in order to make it the future standard of space communications. One of the areas in which there is a current lack of standardization is Quality of Service (QoS) support. Addressing this topic is a relevant stepping stone towards the development of BP, and it is essential for it to be able to support a wider range of traffic types along with their respective requirements, especially those involving future crewed missions such as ESA's Moon Village [7] or NASA's efforts towards putting humans on Mars [8].

This work proposes a QoS support block for BP which enables the definition of a set of QoS requirements for each bundle at the source, as well as the possibility of having an additional dynamic block added by an administrative domain or a subnetwork adjusting that information according to their local policies or resources.

The rest of the document is organized as follows:

- Section II provides a description of BP and DTN.
- Section III proposes an extension block to BP including QoS support. It includes a description of the general structure of the extension block (Section III-A), the definition of the QoS block added by the source (Section III-B), and the definition of the QoS block added by the networks and subnetworks (Section III-C).
- Section IV presents a set of preliminary results on one of the QoS parameters defined in the blocks, namely traffic prioritization. This includes a throughout depiction of the simulated scenario and the specific parameters used (Section IV-A), a set of results concerning the use of several classes for traffic prioritization as described in Section III-B (Section IV-B), and a set of results concerning the use of weighted queuing as the network scheduling algorithm (Section IV-C).
- Lastly, Section V concludes this document and presents the envisioned future work.

II. BP AND DTN

Space communications are considered a highly stressed communication environment due to the obstacles they present, some of which are:

- High propagation delays due to the long distances.
- Foreseeable intermittent connectivity between nodes.
- Frequent unforeseeable interruptions.

This activity was selected via the Open Space Innovation Platform (https://ideas.esa.int) as a Co-Sponsored Research Agreement and carried out under the Discovery program of, and funded by, the European Space Agency and the Hamburg University of Technology. The view expressed in this publication can in no way be taken to reflect the official opinion of the European Space Agency.

- Lack of an end-to-end path to the destination.
- Highly asymmetric data rates.
- High error rates.

These challenges are tackled by DTN through the use of a store-and-forward approach, meaning that the communication is carried out hop-by-hop through intermediate nodes which have persistent storage. This type of storage enables them to hold on to the information until the next visibility window is available [9].

DTN's approach can achieve transmission in scenarios with no end-to-end path, and a faster transmission (or equal in the worst case) than other approaches (see Figure 1).

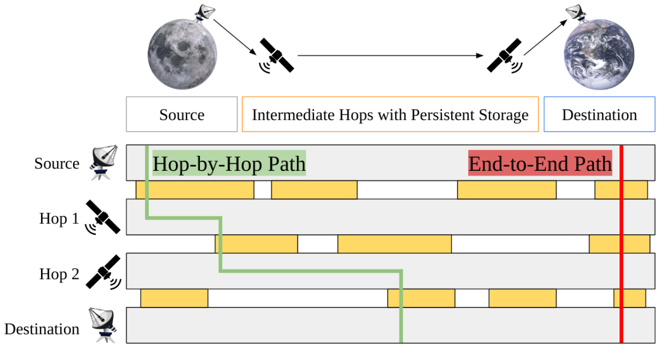


Fig. 1: DTN approach compared to an end-to-end approach.

BP acts as a sub-application layer protocol to support DTN (see Figure 2). It works by "bundling" the data in units called bundles, which include all the relevant information for them to traverse the network and be processed at their destination without the need for handshakes or further information exchanges [10]. This reduces the amount of back-and-forth communication between nodes, hence reducing costs and eliminating unnecessary waits due to the long propagation delays.

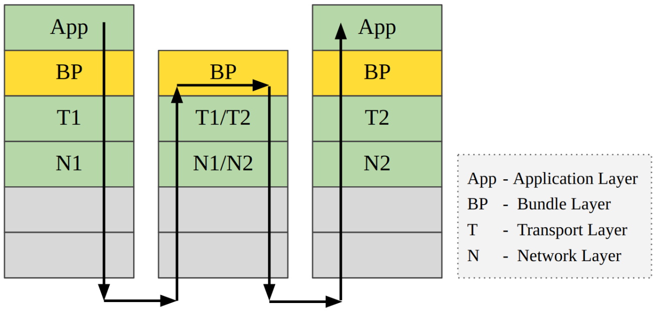


Fig. 2: Placement of BP as a sub-application layer.

The structure of a bundle (see Figure 3) contains three types of blocks:

- Primary Bundle Block: it contains all the basic information of the bundle, such as the source node ID, the destination node ID or the creation time, among others. It is the first block of the bundle, and it must be immutable, meaning that it cannot be modified along the transmission path [3]. This is protected through BP's security contexts, BPSec [11].
- Extension Blocks: these contain additional information about the bundle, and they are optional. Some examples are blocks that allow for hop count, or that give information about the bundle age. Some nodes may not be able to process some extension blocks, in which case the block processing control flags will indicate the expected behavior towards the extension block (see Section III-A).
- Payload Block: this block contains the actual information to be delivered to the destination.

This work focuses on the definition of a new type of extension block, as it will be described in Section III.

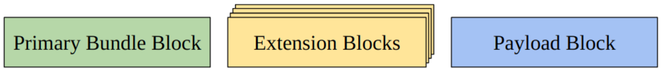


Fig. 3: Structure of a bundle.

QoS for DTN BP is a topic that has been previously addressed, the first time being in RFC 4838 [1]. It defined three priority levels (also known as "cardinal priorities") with two bits in the primary block: expedited, normal and bulk. This approach was taken on by RFC 5050 [2] and used up until BPv6, meaning that they were also part of the various DTN and BP implementations, such as ION [13]. Later on, ION also implemented another QoS draft, the Extended Class Of Service (ECOS) extension [14]. ECOS introduced a finer subdivision of priorities as well as bit flags indicating if the bundle is critical, if it is part of a streaming, and if reliable transmission is needed, among other features. When moving to BPv7 with the RFC 9171 [3], the cardinal priorities were removed from the primary block to consolidate the QoS elements going into a single extension block, for which the proposal was a second version of ECOS which included minor updates like CBOR encoding to match the new BPv7 requirements [15].

These QoS implementation drafts have a limited scalability due to their own structure and the limited amount of QoS parameters they are ready to assess, but also because they do not consider interoperability between different networks, leading to a lack of QoS assessment when a bundle traverses several administrative domains. They however served as a basis for the proposed extension block presented in the rest of this work, which focuses on adding adaptability and scalability, and on refining QoS management over DTNs.

III. EXTENSION BLOCK

The placement of the QoS support mechanism is done in an extension block, which allows for it to be flexible, while still having the chance of being authenticated with the primary bundle block through a bundle integrity block as specified in BPSec [11], protecting it from attacks and unauthorized modifications.

A. General Structure

When defining this extension, there is a need to outline two different blocks: User QoS Extension Block (UQEB) and the

Network QoS Extension Block (NQEB) (see Figure 4). This distinction allows for the source node to provide immutable QoS information through the UQEB, which is authenticated along with the primary bundle block through a bundle integrity block, while also giving the networks and subnetworks the freedom of adding supplementary dynamic information through the NQEB. This information might specify how this bundle should be handled, easing the decision-making progress of the intermediate nodes of the network, especially for resource constrained nodes.

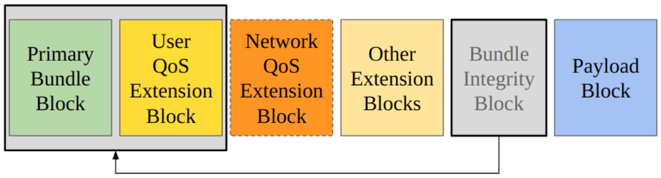


Fig. 4: Bundle structure including the UQEB and the NQEB.

Both UQEB and NQEB are built analogously following the canonical bundle block format, meaning it contains certain standard information, and the actual information that it conveys. Part of the standard information are the processing control flags, which indicate the expected node behavior when encountering an unknown or unprocessable block (see Table I).

TABLE I: Block processing control flags.

Bit Position	**Description**
0	Block must be replicated in every fragment
1	Transmit status report if block can't be processed
2	Delete bundle if block can't be processed
3	Reserved
4	Discard block if it can't be processed
5	Reserved
6	Reserved
7 – 63	Unassigned

For these blocks, the flags must be set to the following values:

- Block processing control flag 0 must be set to 1: the QoS block must be replicated in every fragment in case of bundle fragmentation.
- Block processing control flag 1 should be set to 0: if the bundle is received by an intermediate node which does not support this extension block, no status report should be sent (as they should be limited).
- Block processing control flag 2 must be set to 0: if the bundle is received by an intermediate node which does not support this extension block, the bundle must not be deleted.
- Block processing control flag 4 must be set to 0: if the bundle is received by an intermediate node which does not support this extension block, the block must not be discarded.

Setting block processing control flags 1, 2 and 4 to 0 allows the bundle to traverse the network transparently even if the QoS extension block is not supported by the intermediate nodes.

The entirety of the block is encoded in CBOR as per RFC 9171 specification [3]. Moreover, the QoS parameters are represented through a CBOR map consisting of the QoS parameters to be described (key) and its requirement (value). This ensures that a QoS parameter is always accompanied by a value and vice versa, hence eliminating one possible source of errors and making it faster for the decoder to process [12].

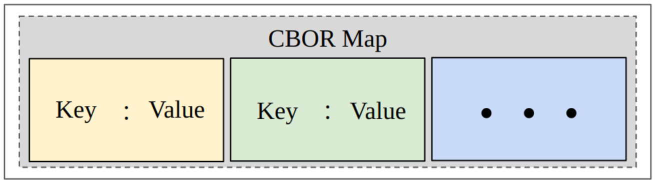


Fig. 5: CBOR map encoding.

When using a definite-length CBOR map, values from 0 to $2^{64}-1$ can be represented. However, keys and values are defined in the range of 0 to 23 in this work, making them a 1-Byte CBOR Tiny Field Encoding for compact representation. Therefore, values above 23 are currently unassigned.

The structure of the blocks is almost identical, with the only structural difference being the need to include the inserting node ID in the NQEB (see Figure 6). This is due to the fact that this block will be added, modified and removed by intermediate nodes along the bundle's path.

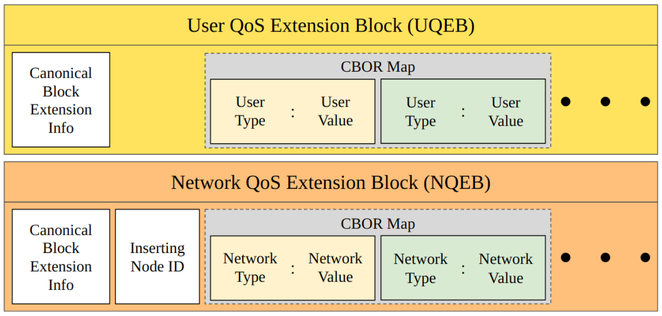


Fig. 6: Dual extension block structure.

Sections III-B and III-C describe each block separately.

B. *User QoS Extension Block*

The UQEB contains the QoS management information specified by the source node. This information must not be modified throughout the bundle transmission, which can be enforced through a bundle integrity block as previously shown in Figure 4 [11].

The parameters that are currently envisioned to be included in the UQEB are traffic prioritization, reliability and latest-only delivery (see Table II). These will be described in detail in Sections III-B1, III-B2 and III-B3.

TABLE II: UQEB keys.

Key	Value
Traffic Prioritization	00
Reliability	01
Latest-Only Delivery	02
Reserved for Future Use	03 — 23
Unassigned	24 — $2^{64} - 1$

1) Traffic Prioritization: The first QoS parameter defined in the UQEB is traffic prioritization. It indicates with which urgency must a bundle be forwarded. The higher the priority, the shorter the waiting time should be until service. There are three main classes:

- Critical: this label is assigned to bundles whose content is crucial and must be served and forwarded with utmost priority. This type of bundles include information such as medical data of astronauts, solar weather alerts etc.
- Quasi-Real-Time (QRT): this label is assigned to bundles whose content is time-sensitive, but which are not categorized as critical. This type of bundles include telecommands, video streaming etc.
- Store-and-Forward (S&F): this label is assigned to bundles whose content is not time-constrained, so they can afford to have a larger delay. This type of bundles include science data, for example.

Both QRT and S&F are defined following the requirements specified for LunaNet [16], while Critical is derived from the needs of the upcoming crewed missions [17]. The effectivity of having three main classes is studied in Section IV-B. Moreover, this class-based mechanism is analogous to the Differentiate Services used by IP networks [18], hence opening the door to interoperability.

These classes should be forwarded following a strict priority approach, meaning that critical bundles must always be served first, QRT bundles shall only be served if there are no critical bundles in the queue, and S&F bundles shall only be served if there are no critical or QRT bundles in the queue.

These three classes are encoded as shown in Table III, with critical having one value assigned, while QRT and S&F have 12 and 11 assigned values respectively.

TABLE III: Priority values.

Priority	Encoding
Critical	00
Quasi-Real-Time	01 — 12
Store-and-Forward	13 — 23
Unassigned	24 — $2^{64} - 1$

This distribution ensures that there can be several values within QRT and S&F, allowing for a granulated description of the priority of the bundle through sub-priorities (see Figure 7). The three main priorities are managed in a strict manner to ensure that the QoS requirements of the critical and QRT bundles are met, but within the QRT and S&F sub-priorities, weighted queuing should be implemented to avoid data starvation of the lower sub-priorities. This approach will be further analyzed in Section IV-C.

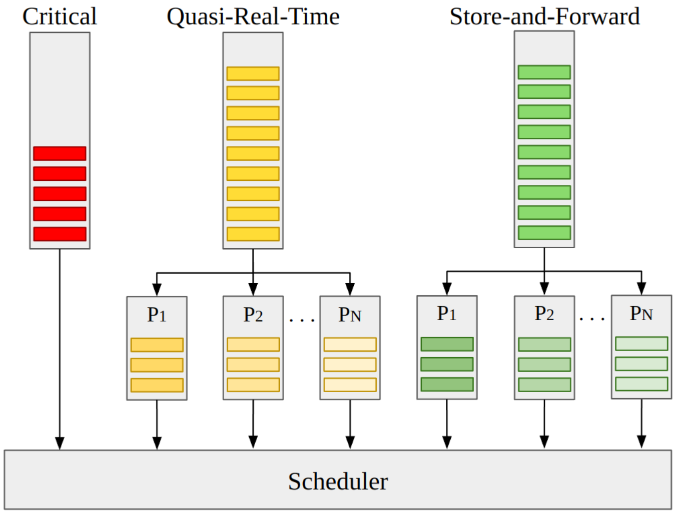


Fig. 7: Queue management for different priority levels.

2) Reliability: The second QoS parameter defined in the UQEB is reliability. It indicates if the system should ensure the reliable transmission of a bundle or not, hence having two main values: reliable or unreliable. These are encoded as shown in Table IV, and can be achieved through the following means:

- A reliable transmission can be achieved by using a reliable convergence layer which can re-transmit lost data such as Transmission Control Protocol Convergence Layer (TCPCL) or Licklider Transmission Protocol Convergence Layer (LTPCL) with red data, by using a very reliable communication channel or by adding more coding at the link layer.
- An unreliable transmission can be achieved by using a convergence layer which does not apply re-transmission such as User Datagram Protocol Convergence Layer (UDPCL), by using a communication channel that might be less reliable, or by reducing or eliminating the coding at the link layer, thus having less overhead as a trade-off.

TABLE IV: Reliability values.

Reliability	Encoding
Strictly Reliable	00
Reliable if possible	01
Reserved for Future Use	02 — 21
Unreliable if possible	22
Strictly Unreliable	23
Unassigned	24 — $2^{64} - 1$

The reliability options presented in Table IV have the following definitions:

- "Strictly Reliable" means that the bundle must be transmitted reliably (through the aforementioned means). If not possible, the node should wait for a chance to reliably transmit the bundle, or it should drop it.
- "Reliable if possible" means that the bundle should be transmitted reliably if it is possible. Else, it can be transmitted unreliably.
- "Unreliable if possible" means that the bundle should be transmitted unreliably if it is possible. Else, it can be transmitted reliably.
- "Strictly Unreliable" means that the bundle must be transmitted unreliably. If not possible, the node should wait for a chance to unreliably transmit the bundle, or it should drop it.

3) Latest-Only Delivery: The third QoS parameter defined in the UQEB is latest-only delivery. This parameter indicates to the node that, out of several bundles that might have been received having the same source node ID and destination endpoint ID, only the latest one should be kept and forwarded. This is useful for cases such as sensor networks that send information periodically, and for which only the last piece of information is relevant [19]. This parameter is encoded as a binary flag, as shown in Table V. However, future work might add granularity to this parameter, such as keeping a certain number of the latest bundles.

TABLE V: Latest-only delivery values.

Latest-Only Delivery	**Encoding**
All valid	00
Latest bundle valid	01
Reserved for Future Use	02 — 23
Unassigned	$24 — 2^{64} - 1$

4) Workflow: An example of the usage of the UQEB can be seen in Figure 8. In this example, the source (S) aims to send three bundles (B1, B2 and B3) with a certain defined Traffic Prioritization (TP) and Reliability (R) to the destination (D) through an intermediate node (N). Bundle 1 has a priority of QRT and must be transmitted reliably (Re), bundle 2 has a priority of S&F and must be transmitted reliably, and bundle 3 has a priority of QRT as well, but must be transmitted unreliably (URe).

When transmitting these bundles, B1 and B3 have priority over B2 since QRT has a higher priority than S&F. If nothing else is specified, bundles of the same priority and sub-priority are served in a first in, first out (FIFO) manner. Therefore, B1 is served first, then B3, and lastly B2. When it comes to reliable and unreliable transmission, B1 and B2 will be sent reliably (whether by using TCPCL or LTPLC red as a convergence layer or by employing any other of the described mechanisms), while B3 will be sent in an unreliable manner.

C. Network QoS Extension Block

The NQEB is a response to the need of customization by networks and subnetworks under different administrative domains to support implementation of service level agreements. It can also be used by the networks and subnetworks to translate the UQEB according to their local policies and resources. In this way, not only can they qualify the stated QoS requirements, but also put them in a manner that is easy to process by all the intermediate nodes of the network or subnetwork, even the resource constrained ones. All in all, this block should complement (but not replace) the UQEB, and it shall be deleted once the bundle leaves the administrative domain in which it was added.

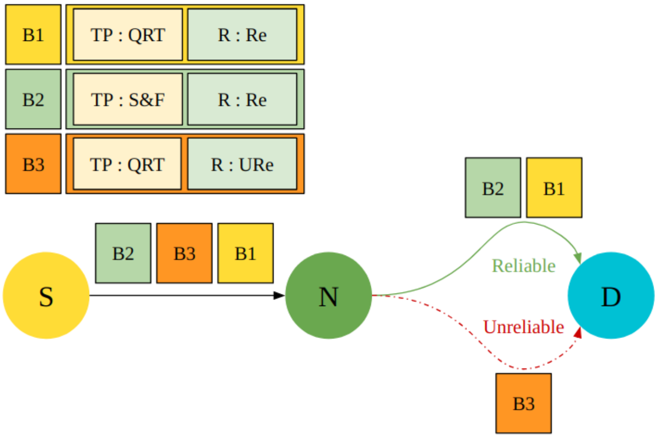


Fig. 8: Example of UQEB usage.

An example of the usage of the NQEB can be seen in Figure 9. In this case, there are three networks: N1, N2 and N3. The source is part of Network 1 (S_{N1}), and it aims to send a bundle with a certain defined Traffic Prioritization (TP) and Reliability (R) to the destination in Network 3 (S_{N3}) through Network 2. This path consists of an entry gateway (E_{N2}), and intermediate node (N_{N2}) and an exit gateway (X_{N2}).

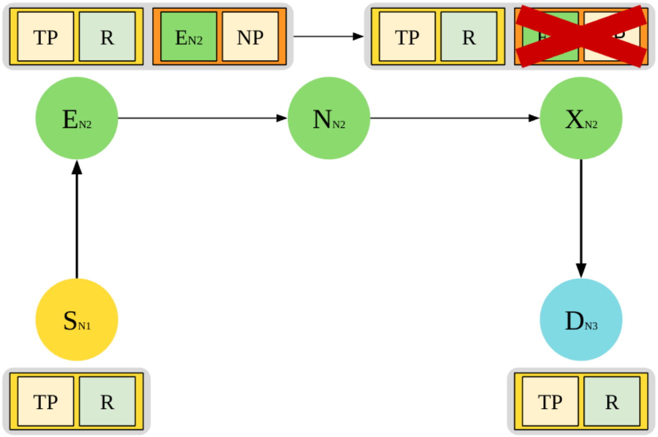


Fig. 9: Example of NQEB usage.

When the bundle leaves Network 1 and enters Network 2 through the entry gateway, the NQEB is appended to the bundle. It consists of the inserting node ID (E_{N2}) and relevant information for Network 2, in this example the Network Priority (NP). This NQEB is kept along the path within

Network 2 and, when reaching the exit gateway (X_{N2}), it is deleted. The bundle is now forwarded towards its destination (D_{N3}) with the UQEB intact and without the NQEB.

As a conclusion, the proposed dual structure provides an immutable block which contains standardized information about the QoS requirement of the bundle, and a dynamic block that can satisfy the needs of each network or subnetwork separately as well as giving the means to make it easier for every node to process the bundle according to its resources.

IV. PRELIMINARY RESULTS

In order to check the viability of the proposed structure, preliminary results are presented to highlight the impact on traffic prioritization together with weighted queue management.

A. Simulation

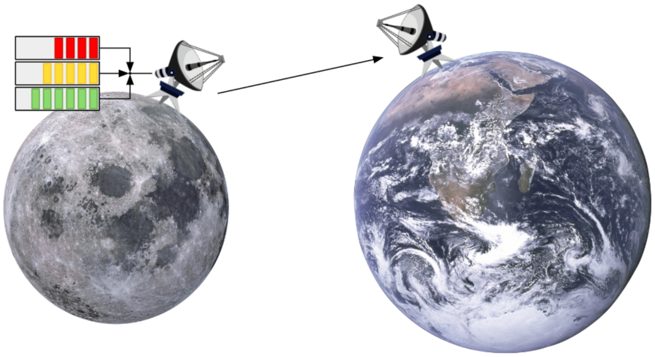


Fig. 10: Earth-to-Moon direct communication link.

The simulation is done following previous work, which depicts an Earth-to-Moon direct communication link (see Figure 10) [20]. This communication link is modeled through a Markov Chain (see Figure 11), containing three possible states:

- Success: the bundle transmission has been successful.
- Short-Term Loss: the bundle transmission has failed due to short-term losses, which can be caused by antenna-pointing errors, interferences or light atmospheric weather conditions.
- Long-Term Loss: the bundle transmission has failed due to the channel being blocked for a long period of time, which can be caused by space or Earth weather.

This Markov Chain can be described through Matrix 1:

$$\begin{pmatrix} 1 - P_{Loss} - P_{Block} & P_{Loss} & P_{Block} \\ 1 - P_{Loss} - P_{Block} & P_{Loss} & P_{Block} \\ 1 - P'_{Loss} - P_{C.Block} & P'_{Loss} & P_{C.Block} \end{pmatrix} \tag{1}$$

The three main probabilities described are the following:

- Loss Probability (P_{Loss}): probability of going or staying in a short-term loss state. It is taken as 4.3% [21]. Exceptionally, the probability of going to a short-term loss state from the long-term loss state is taken as $P'_{Loss} = P_{Loss} * (1 - P_{C.Block})$

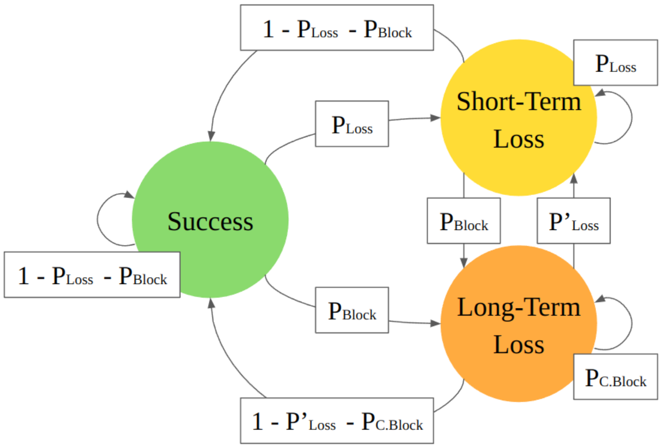


Fig. 11: Markov chain modeling the communication link.

- Blockage Probability (P_{Block}): probability of going into a long-term loss state. It is taken as the probability of a solar flare of classes C, M or X taking place, which is 0.003% [22].
- Continuous Blockage Probability ($P_{C.Block}$): probability of staying in a long-term loss state. It is derived from the length of a solar flare such as the ones described above, resulting in 99% [23].

The rest of the values of the Markov Chain can be derived from these as depicted in Figure 11, resulting in Matrix 2. These values are used as input in the simulation.

$$\begin{pmatrix} 95.697\% & 4.3\% & 0.003\% \\ 95.697\% & 4.3\% & 0.003\% \\ 0.957\% & 0.043\% & 99\% \end{pmatrix} \tag{2}$$

The implementation is done in Python using Simpy [24], and visualization of it can be seen in Figure 12, which shows the several steps that are followed. Firstly, the bundles are generated and received at the first node (step 1), where they are sent to different queues according to their priority type (step 2). These queues are then serviced in order, and the transmission of the bundles is attempted (step 3). If they are successfully transmitted or not depends on the state of the channel, which is decided according to the Markov Chain presented in Figure 11 (step 4). If the channel state is “Success”, the bundle is successfully transmitted, and it is counted as such; and if the channel state is either “Short-Term Loss” or “Long-Term Loss”, the transmission fails, and it is recorded as such (step 5).

Step 2 of Figure 12 can be seen in detail in Figure 13. After the bundles arrive at the queue, it is checked what their priority is. If it is critical, it is sent to the critical bundles queue; if it is QRT, it is sent to the QRT bundles queue; and if is S&F, it is sent to the S&F bundles queue.

Lastly, step 3 of Figure 12 can be seen in detail in Figure 14. The queues are checked, and if there are critical bundles in the queue, they are serviced using a FIFO approach. Should there be no critical queue, the QRT queue is checked. If there are

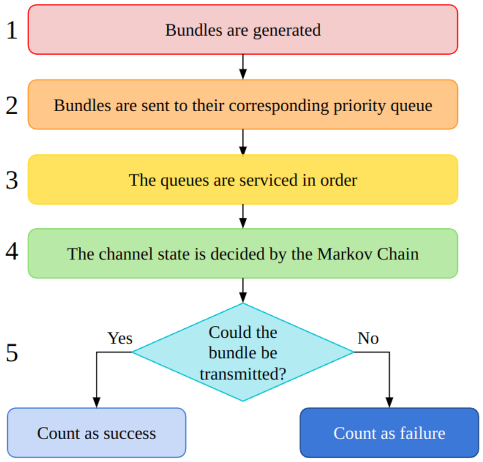


Fig. 12: Flowchart of the simulation.

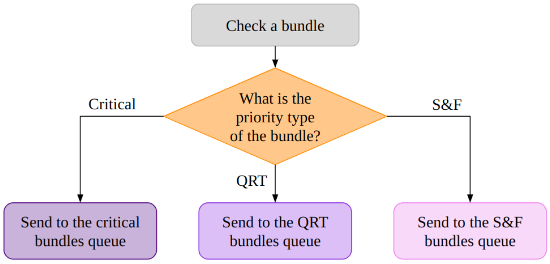


Fig. 13: Flowchart of the queuing distribution.

QRT bundles queuing, they are serviced following a weighted approach (see Figure 7). If the QRT queue is empty, the S&F queue is checked, and if there are bundles queuing, they are also serviced following a weighted approach.

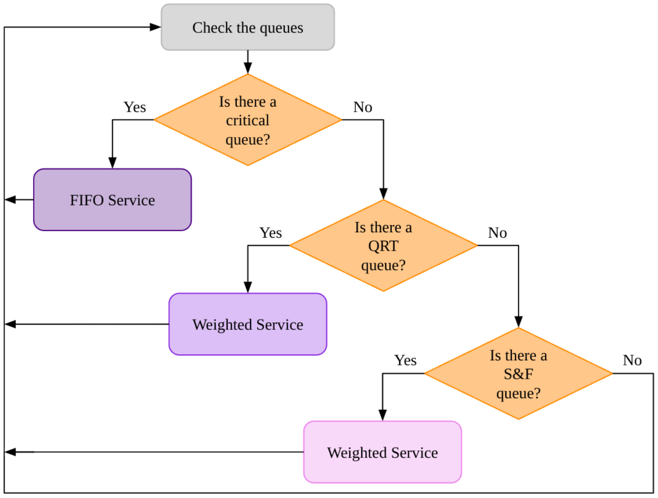


Fig. 14: Flowchart of the servicing process.

For the simulation, the longest distance between Earth and the Moon is assumed ($405\,500\,\mathrm{km}$), resulting in a propagation delay of $1.35\,\mathrm{s}$. The overall delay is taken as $1.5\,\mathrm{s}$ to account for any additional delays, including the $6.4\,\mathrm{ms}$ of transmission delay resulting of assuming a bundle size of $64\,\mathrm{kB}$ and a data rate of $10\,\mathrm{Mbit\,s^{-1}}$ [17]. Bundles are generated according to a uniform distribution while keeping the system stable, meaning that the combined bundle arrival time is slightly lower than the service time. The time mark for the communication to be considered QRT is set to $2.5\,\mathrm{s}$ following the requirements set by ESA [25]. Therefore, this work will measure the results for the $2.5\,\mathrm{s}$ mark and double the time for reference, the $5\,\mathrm{s}$ mark. These preliminary results are shown using the Cumulative Distribution Function (CDF) to measure the percentage of bundles (y-axis) which arrives within a certain time (x-axis), hence representing the end-to-end delay of the bundles in the simulation. Lastly, the experiments simulate the system running for 500 days.

B. *Traffic Prioritization*

In order to assess the effect of dividing the traffic into three priority classes, the generated traffic described in Section IV-A is divided into three types: critical, QRT and S&F.

The first simulation depicts a scenario in which all three classes are generated equally, meaning that one third of the generated traffic is critical, one third is QRT and one third is S&F (see Figure 15).

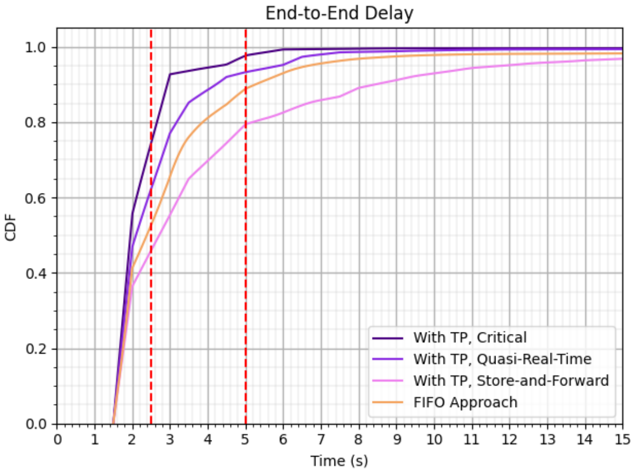


Fig. 15: End-to-end delay for the first simulation.

The critical class has the lowest end-to-end delay in the sense that it has the highest percentage of bundles arriving within the $2.5\,\mathrm{s}$ and $5\,\mathrm{s}$ marks (73% and 97.5% respectively). The QRT class has the second-highest percentage of bundles arriving within the set time marks (62% and 93% respectively), and the S&F class has the lowest amount of bundles arriving in those time limits (46% and 80% respectively). These results can be compared with the scenario without traffic prioritization, in which the same traffic is generated, but it is served following a FIFO approach. For this case, the percentage of received bundles is 57% for the $2.5\,\mathrm{s}$ mark and 90% for the $5\,\mathrm{s}$ mark. These results lead to the conclusion that dividing the incoming traffic into classes according to their priorities allows for the higher priority bundles to arrive with a lower end-to-end delay at the expense of lower priority bundles.

The second simulation depicts a more realistic scenario, in which critical bundles represent 10% of the generated traffic, and the remaining 90% of the generated traffic is equally distributed between the QRT and S&F classes, meaning 45% of the traffic each (see Figure 16).

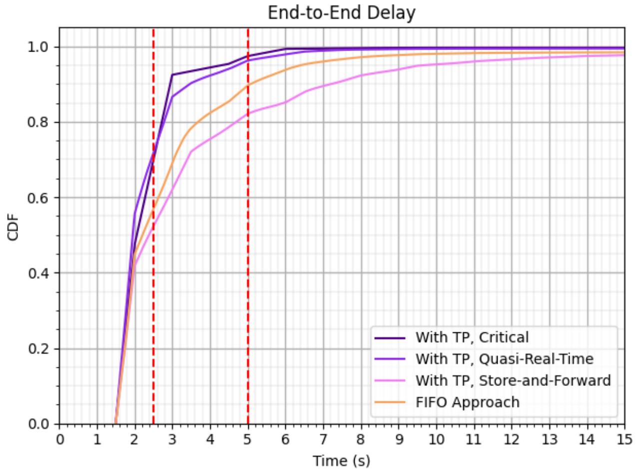


Fig. 16: End-to-end delay for the second simulation.

Once again, the critical class has the highest percentage of bundles arriving within the 2.5 s and 5 s marks (73% and 97.5% respectively). The QRT class the second-highest percentage of bundles arriving within the set time marks (71% and 96% respectively) and, the S&F has a lowest amount of bundles arriving in those time limits (47% and 83% respectively). For the FIFO scenario, the percentage of arriving bundles remains unchanged from the previous simulation since the total amount of traffic is the same: 57% for the 2.5 s mark and 90% for the 5 s mark. This simulation shows that, since there is a lower amount of critical traffic, the difference in performance between this and the QRT traffic is reduced, yielding a better arrival time for the second class. Since the scheduling is managed with a strict priorities approach, none of these improvements are reflected on the third class, S&F. Nevertheless, this is not seen as a drawback since, as mentioned in Section III, this traffic type does not need to be delivered within brief time limits for it to be valid.

In conclusion, these preliminary results show that traffic prioritization allows for a certain percentage of the traffic to arrive within the desired time limits at the expense of traffic that is not time-sensitive, such as S&F, which can be delivered with a higher delay, making it a desirable trade-off. Should one class have a lower amount of traffic, its performance advantage will be leaked onto the following class but not onto the others as a result of the strict prioritization. This is desirable for the case of traffic being either critical, QRT or S&F. Nevertheless, for bundles from the same class but with different sub-priorities this might not be an advantage but a drawback. Therefore, weighted queuing also needs to be studied.

C. Weighted Queuing

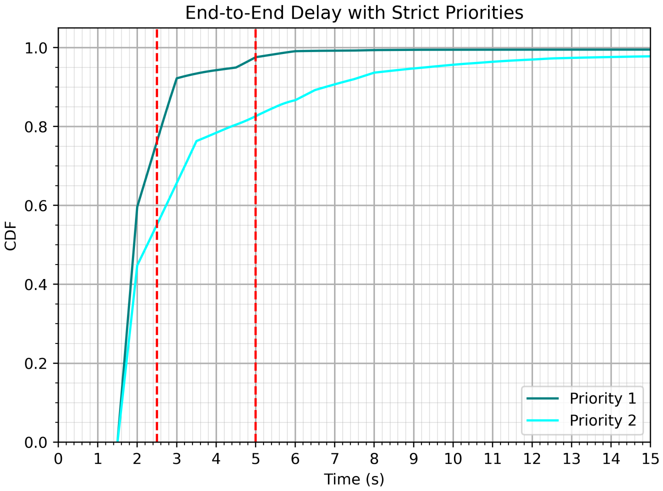
When scheduling the different sub-priorities withing a priority class, three main approaches can be followed, for which an analysis can be seen in Figure 17.

- Strict Priorities: this means following the same approach for sub-priorities as it is already followed by the main three classes. In this case, no bundles with priority P_N are to be transmitted until all bundles with priority P_{N-1} have been served. This approach might be detrimental to the lower sub-priorities since, even though they might be labelled as QRT, they might suffer from resource starvation and not reach their QoS requirements (see Figure 17a).
- FIFO: this approach is fairer for the lower sub-priority bundles, but higher delays than desired might incur for bundles which, according to their sub-priority, should be served more urgently. It therefore defeats the point of having sub-priority classes and eliminates the granularity of this approach (see Figure 17b).
- Weighted Queuing: this approach is proposed as a middle ground between both extremes. The queue is to be served respecting the sub-priorities but not in a strict manner, meaning that higher sub-priorities will be serviced more often, while avoiding data starvation for lower sub-priority bundles. This is measured with a weight, which dictates the ratio with which the higher sub-priority queue will be serviced with regard to the lower sub-priority queue. For example, a weight of two will result in a 2:1 ratio, that is, in twice and many higher sub-priority bundles being sent as lower sub-priority (see Figure 18). This approach presents a trade-off between a lower delay for the higher sub-priority bundles and a higher delay for the lower sub-priority bundles while keeping their advantages (see Figure 17c).

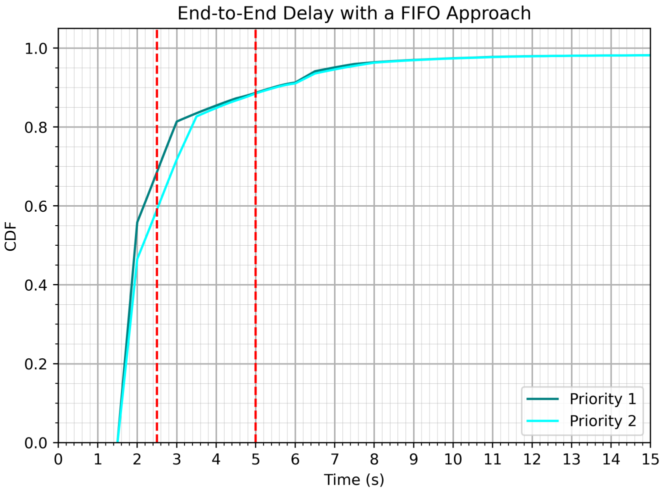
In these simulations, two sub-priorities are presented: priority 1 and priority 2. Both generate the same amount of traffic in a uniform manner, and the simulation is done analogously to the previously presented results.

When looking at the priority 1 bundles, the strict priorities approach has the best results, with a bundle transmission of 77% for the 2.5 s mark and of 97.5% for the 5 s mark. The FIFO approach has the worst performance, with a 70% and 88% respectively. Weighted queuing with a weight of 2 stays in the middle, with a 73% and 95% respectively. For the priority 2 bundles, the performance is the opposite: it is worst at strict priorities, with a bundle transmission of 55% for the 2.5 s mark and of 83% for the 5 s mark, and best at FIFO, with a bundle transmission of 60% for the 2.5 s mark and of 88% for the 5 s mark. Weighted queuing marks the middle ground, with a bundle transmission of 57% for the 2.5 s mark and of 85% for the 5 s mark.

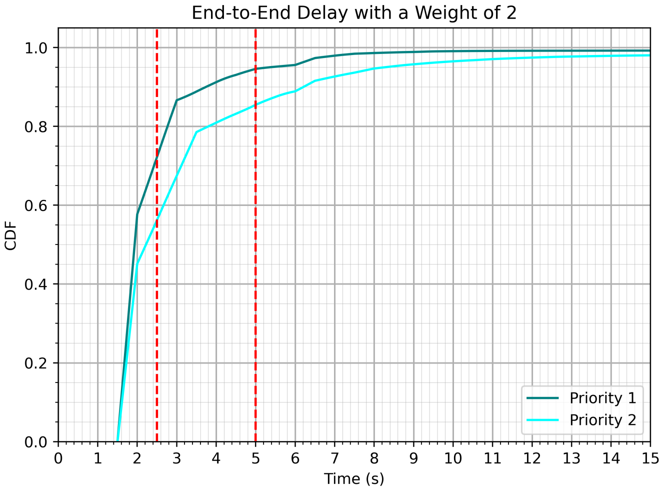
As a summary, both the strict priorities and FIFO approaches are not suitable for sub-priorities in which a certain



(a) Strict priorities approach.



(b) FIFO approach.



(c) Weighted approach with a weight of 2.

Fig. 17: End-to-end delay for the different approaches.

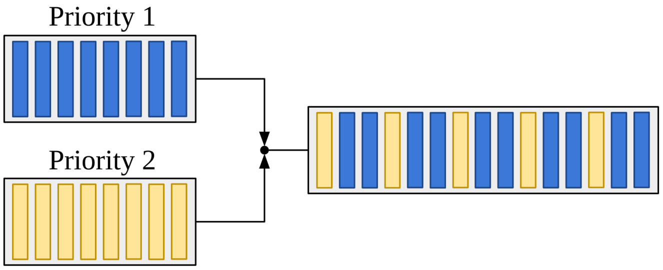


Fig. 18: Weighted queuing example with a weight of 2.

order must be maintained, but in which resource starvation needs to be avoided. Therefore, weighted queuing should be implemented to provide an approach that fulfills all the QoS requirements for these sub-priorities. Moreover, the chosen weight of 2 shows to work for the specific scenario hereby presented, being the fairest of all while respecting sub-priorities.

V. CONCLUSION

In conclusion, the specification of a QoS support system for BP is one of the stepping stones needed for it to be the future standard of space communications. This work proposes a dual extension block structure containing QoS requirements specified by the source in an immutable way, and a qualification of these requirements by the networks and subnetworks according to their local policies or resources. Moreover, preliminary results on the use of these blocks is presented, more specifically on the implementation of traffic prioritization.

Firstly, it is shown that using three main priority classes allows for critical and QRT bundles to have a higher chance of arriving within the set time marks than in the scenario with no traffic prioritization. As a trade-off, S&F bundles have a higher end-to-end delay, which is a desirable outcome since, by definition, they are not time-sensitive.

Secondly, three manners of treating sub-priorities are presented: strict priorities, FIFO and weighted queuing. It is shown that weighted queuing provides a fairer solution and allows for all sub-priority bundles to arrive within a range of the desired end-to-end delay while still maintaining a priority system. This is the desired outcome for situations in which strict priorities are not needed, since the three main priority classes already cover that area, but for which there is still a need for maintaining different sub-priorities among bundles.

Lastly, while this work presents a detailed proposal and preliminary results, a more in-depth analysis is needed in the areas of:

- Weighted Queuing: these preliminary results depict two sub-priorities aiming to transmit at the same time with equal traffic. Nevertheless, more complex scenarios with a higher number of sub-priorities and with differently distributed weights between them must be studied. Furthermore, this scenario considers all bundles to be the same size for simplicity, but weighted queuing should be implemented per byte number and not per bundle number to avoid larger bundles congesting the channel, which adds complexity to the system.

- Reliability: more in-depth study on the manners in which different levels of reliability can be achieved must be done. Moreover, future implementations of this extension blocks might differentiate between manners of achieving reliability, specifying if the bundle requires retransmission of lost data, specific channel coding etc.
- Latest-only delivery: as of now, this key is presented as a binary flag. However, further research should determine if a higher granularity of outcomes is required.
- Implementation: the proposed extension blocks shall be implemented using established BP implementations such as ESA's BPI [26] or NASA's ION [13] to allow further analysis of its impact, as well as more complex results.

REFERENCES

[1] Cerf, V., Burleigh, S., Hooke, A., Torgerson, L., Durst, R., Scott, K., Fall, K. and Weiss, H. "RFC 4838: Delay-Tolerant Networking Architecture". (2007).

[2] Scott, Keith, and Scott Burleigh. "RFC 5050: Bundle protocol specification". (2007).

[3] Burleigh, Scott, Kevin Fall, and Edward J. Birrane. "Bundle Protocol Version 7." Request for Comments. Internet Engineering Task Force, January 2022. https://doi.org/10.17487/RFC9171.

[4] ESA, "METERON Project". Online, last accessed: 13.06.2024. Available: https://www.esa.int/Enabling_Support/Space_Engineering_Technology/Automation_and_Robotics/METERON_Project.

[5] NSSDC Archive, "Korea Pathfinder Lunar Orbiter". Online, last accessed: 13.06.2024. Available: https://nssdc.gsfc.nasa.gov/nmc/spacecraft/display.action?id=2022-094A.

[6] CCSDS, Tech. Rep., "CCSDS White Paper on the Use of BP Version 7," 2020. Available: https://ntrs.nasa.gov/citations/20205008473.

[7] ESA, "Moon Village". Online, last accessed: 13.06.2024. Available: https://www.esa.int/About_Us/Ministerial_Council_2016/Moon_Village.

[8] NASA, "Humans to Mars". Online, last accessed: 13.06.2024. Available: https://www.nasa.gov/humans-in-space/humans-to-mars/.

[9] Forrest Warthman et al. "Delay-and disruption-tolerant networks (DTNs)". A Tutorial. V. 0, Interplanetary Internet Special Interest Group (2012): 5–9.

[10] CCSDS, "CCSDS Bundle Protocol Specification", 2015. https://public.ccsds.org/Pubs/734x2b1.pdf.

[11] Birrane, Edward J., Sarah Heiner, and Ken McKeever."The BPSec Security Mechanism". (2023): 93–114.

[12] Bormann, C., and Hoffman, P. "Concise Binary Object Representation" RFC8949. Internet Engineering Task Force, December 2020. https://www.rfc-editor.org/rfc/rfc8949.html.

[13] NASA, "Interplanetary Overlay Network". Online, last accessed: 13.06.2024. Available: https://www.nasa.gov/directorates/somd/space-communications-navigation-program/interplanetary-overlay-network/.

[14] Burleigh, Scott, "Bundle Protocol Extended Class Of Service (ECOS)", 2014. Online, last accessed: 13.06.2024. Available: https://datatracker.ietf.org/doc/draft-irtf-dtnrg-ecos/.

[15] Burleigh, Scott and Templin, Fred, "Bundle Protocol Extended Class Of Service (ECOS)", 2021. Online, last accessed: 13.06.2024. Available: https://datatracker.ietf.org/doc/draft-burleigh-dtn-ecos/.

[16] Israel, David J., Kendall D. Mauldin, Christopher J. Roberts, Jason W. Mitchell, Antti A. Pulkkinen, La Vida D. Cooper, Michael A. Johnson, Steven D. Christe, and Cheryl J. Gramling. "LunaNet: A Flexible and Extensible Lunar Exploration Communications and Navigation Infrastructure." In 2020 IEEE Aerospace Conference, 1–14, 2020. https://doi.org/10.1109/AERO47225.2020.9172509.

[17] Lunar Communications Architecture Working Group, "The Future Lunar Communications Architecture". Interagency Operations Advisory Group, 2022.

[18] Carpenter, Brian E., and Kathleen Nichols. "Differentiated services in the Internet". Proceedings of the IEEE 90.9 (2002): 1479-1494.

[19] Baumgärtner, Lars. "Dtnmqtt: A Resilient Drop-In Solution for MQTT in Challenging Network Conditions", 2024. [Unpublished]

[20] Algarra Ulierte, Teresa, et al. "Lunar Communication Services: Feasibility Study on Traffic Prioritization of Quasi-Real Time Communications over DTNs". 2023 IEEE International Conference on Wireless for Space and Extreme Environments (WiSEE). IEEE, 2023.

[21] ESA, "Optimised CCSDS Protocol Stack for High Data Rate (Erasure Vectors)". https://esastar-publication-ext.sso.esa.int/ESATenderActions/details/8509.

[22] Naoto Nishizuka, Yuki Kubo, Komei Sugiura et al. "Reliable probability forecast of solar flares: deep flare net-reliable (DeFN-R)". In: The Astrophysical Journal 899.2 (2020), p. 150.

[23] Yangfan Guo, Bo Liang, Song Feng et al. "Feature Identification and Statistical Characteristics of Quasi-periodic Pulsation in Solar Flares using the Markov-Chain-Monte-Carlo Approach". In: The Astrophysical Journal 944.1 (2023), p. 16.

[24] N. Matloff, "Introduction to Discrete-event Simulation and the Simpy Language" Davis, CA. Dept of Computer Science. University of California at Davis. Retrieved on August, vol. 2, no. 2009, pp. 1-–33, 2008.

[25] Project Moonlight, https://www.esa.int/Applications/Telecommunications_Integrated_Applications/Lunar_satellites. Last accessed 4 Apr. 2023.

[26] ESA, "Extending the Internet into Space". Online, last accessed: 13.06.2024. Available: https://esoc.esa.int/extending-internet-space.