Opportunities for Optimization through Parallelization of the Bundle Protocol

Consistent with the Recommendations in CCSDS 734.2-B-1 and the Definitions of RFC-5050

Gerard J. Holzmann*

ABSTRACT. — We consider potential parallelization strategies for the Bundle Protocol, and check if such efforts are consistent with (i.e., explicitly allow or disallow) the recommendations and definitions given in two main reference documents for this protocol. In this assessment, the text of the two guiding documents is taken as written and not subject to change. The effort is not to look for ambiguities in either document, but rather to see if they are consistent with (i.e., explicitly allow or disallow) specific optimization strategies based on parallelization.

I. Introduction

The Bundle Protocol (BP) is a protocol defined to operate above the Transport Layer in the traditional ISO/OSI model (cf. [1], Figure 1, p.4[1]), which means that it can be used over any type of lower layer protocol implementation at the Transport, Network, Data-link, and Physical layers. The stated target is its use over unreliable and scarce data links, to support functionality required for delay-tolerant network connections.

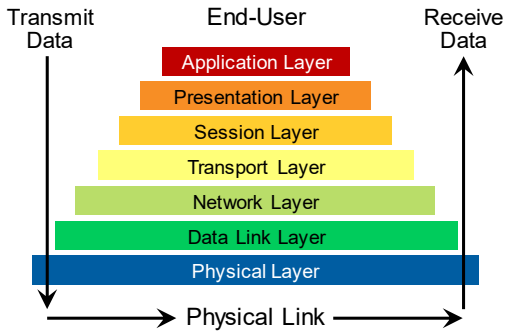
II. Optimization Approaches

There are two possible targets for optimization (e.g., through parallelization) of the bundle protocol: (a) to improve end-to-end throughput, e.g., to allow data to be sent faster, for instance by using parallel data streams (Section II.A), and (b) to send data at the same rate, but to use less time at the transmitter and/or at the receiver to process the data (Section II.B).

[1] Reference [2], p. 1-1, states instead that "BP provides Network Layer service," but this does not seem to refer to the standard OSI terminology.

* Software Assurance and Assurance Research Section.

The research described in this publication was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.



The Seven Layers of the ISO/OSI Reference Model

The reference in [1] to BP's place in the OSI hierarchy sets constraints on the types of optimization that can be considered for each of these two cases. First, if the lower protocol layers can only support single uni-directional transfers of data at low, and fixed, baud-rates (e.g., Voyager–Earth) then opportunities for increasing throughput are quite limited. If, however, the lower layers support the use of multiple simultaneous connections (Section II.A) may be realized by splitting the transfer of bundles across simultaneous BP sessions. This mode of transfer, which can be defined at a higher protocol layer, is independent of the definition and recommended use of the BP itself.

The second option for optimization (Section II.B) may be implemented at the Transport or Session Layers, and does not alter the behavior of either the BP protocol or the lower protocol layers. By parallelizing the preprocessing and postprocessing of bundles, the real-time requirements for sending and receiving bundle payloads can be reduced, and possibly could help match higher transmission rates that may be available at the lower layers (e.g., high-speed optical links).

Below I'll compare these two possible options against the guiding documents to see if there are any explicit statements or directions given there that may conflict with either approach.

A. Parallel BP Sessions at the Transport Layer

This approach to maintain multiple independent and simultaneous BP sessions for different parts of the bundle sequence, *provided* that the lower protocol layers can support this, is orthogonal to the definition or recommended use of the BP protocol itself and is therefore unlikely to conflict with its definitions. There is evidence in [2] that this use is indeed compliant with the definitions and recommendations.

In reference [2], the following relevant statements can be found.

In **Section 1.4.3** (and elsewhere), it is noted that *"[The] application may register multiple endpoints."* This capability is indeed useful if one were to use multiple independent BP sessions.

In **Section 1.4.3.2**, in a note to the definition of a bundle protocol agent, BPA [emphasis added]: *"BPA functionality can be coded into individual nodes, as a shared library that is shared by any number of bundle nodes on a single computer, as a daemon whose services are invoked via*

*inter-process or network communication by **one or more bundle nodes** on **one or more computers**, or in hardware."*

Further, in the definition of application agents [AA, emphasis added]: *"AAs may perform arbitrarily complex application functions, perhaps even offering **multiplexed DTN** [delay tolerant network] **communication services** to a number of other applications. As with the BPA, the way AA performs its functions is wholly an implementation matter [...]"*

In **Section 2.4** [emphasis added]: *The Bundle Protocol as specified in this document does **not** provide the following services: (a) in-order delivery of bundles, (b) complete delivery of sequences of bundles.*

In **Section 3.3**, potentially supporting the identification of a reassembly requirement for separate bundle streams that are transmitted over independent connections with Extended Class of Service tags to identify *"special handling of bundles."*

And quite explicitly in **Section 4.1.2** [emphasis added]: *The BP node shall be implemented such that virtually **any number of transactions** may be conducted **concurrently** in various stages of transmission or reception at a single BP node.*

> *NOTE – To clarify: the implementation needs to be able to accept a primitive, and thereupon initiate a new transaction prior to the completion of previously initiated transactions. The requirement for concurrent transaction support therefore does not necessarily imply that the implementation needs to be able to begin initial transmission of data for one transaction while initial transmission of file data for one or more other transactions is still in progress. (But neither is support for this functional model precluded.)*

In RFC-5050 [1], we find the following relevant observations.

Section 3.1 (p. 5), it is noted that *"Multiple instances of the same bundle [...] might exist concurrently in different parts of a network."* This reference, though, more likely refers to a single sequential bundle sequence where component parts have reached different points along the chain towards the destination.

A more explicit statement that the use of parallel BP sessions is foreseen and compliant with the intent of the BP definition occurs in **Section 3.1** (p. 9) [emphasis added]: *"A transmission is a sustained effort by a node's bundle protocol agent to cause a bundle to be sent to all nodes in the minimum reception group of some endpoint (which may be the bundle's destination or may be some intermediate forwarding endpoint) in response to a transmission request issued by the node's application agent. **Any number of transmissions may be concurrently undertaken by the bundle protocol agent of a given node**."*

A potential counter-point appears in **Section 8** (p. 45) which states: *"The bundle protocol has been designed with the notion that it will be run over networks with scarce resources. [...] to send bundles over such constrained environments [...]."* This implies that the availability of high-bandwidth connections that could support parallel data transfer sessions, though still with high latency, is not anticipated, but of course also not prohibited or excluded from consideration.

Another counter-point, also in **Section 3.2** (pp. 9–10), is that the underlying architecture and implementation of bundle nodes are *not required* to support parallel sessions. Therefore, the option for parallel bundle transfer sessions would only exist between nodes along a forwarding chain that could support this capability. Such capabilities are not required for conformance to the BP protocol and therefore cannot be assumed.

Finally, still in [1], **Section 5.4**, Step 5 (p. 28) notes: *"To keep from possibly invalidating bundle security, the sequencing of the blocks in a forwarded bundle must not be changed as it transits a node; received blocks must be transmitted in the same relative order as that in which they were received."* This requirement, though, is likely intended to maintain the integrity of block transfers within individual sessions, and does not necessarily apply to the order of block transfers across parallel sessions.

B. Optimization of Preprocessing and Postprocessing of Bundles at the Network Layer

This second point where parallelization strategies can be used is in the preparation and/or the postprocessing of the bundle format at the Session Layer. Input, at the sender's side of the protocol, is the payload to be transmitted, which needs to be partitioned in blocks and formatted into bundle blocks. An example of the data format is given in Section 4.5 of [1] on pp. 17–18. This means setting the required flags and where necessary computing and filling in all required fields. The preprocessing can also include an encoding step for the payload itself.

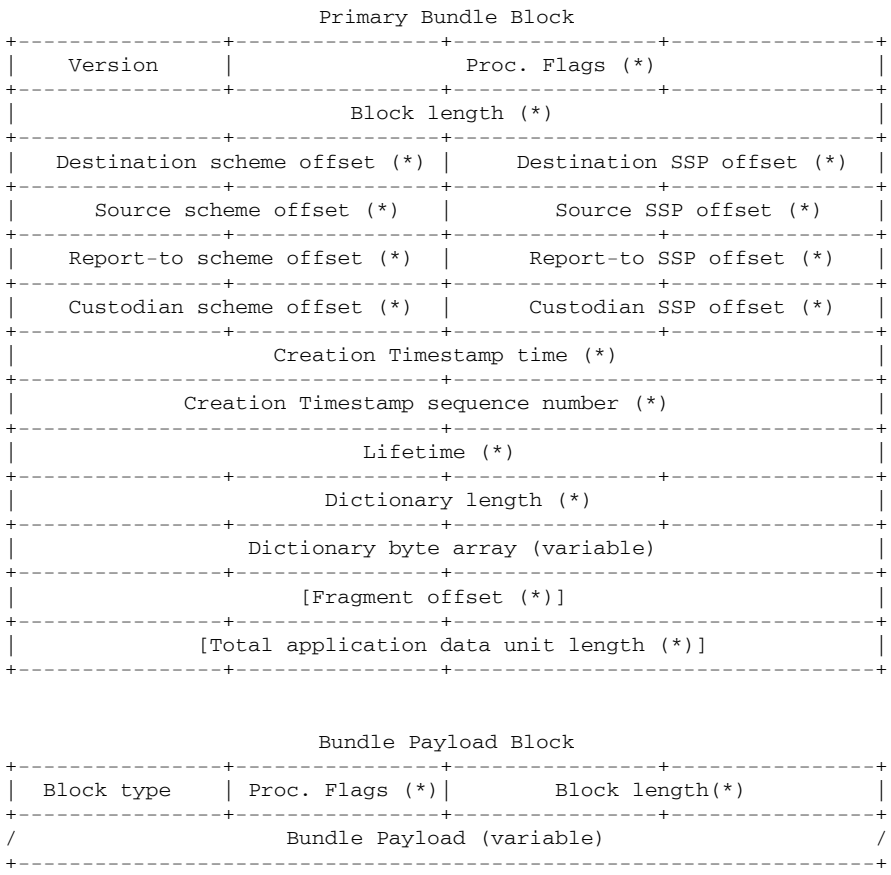
Typically, in communication protocols, the preprocessing phase is rarely a bottleneck, compared to postprocessing at the receiver. Nonetheless, it seems clear that each of these steps can be parallelized in a way that is *invisible* to both the upper and the lower protocol layers. For instance, while one bundle is in the process of being transmitted, the next bundle(s) can be prepared in parallel.

Similarly, while one bundle is being received, bundles received earlier can (continue to) be postprocessed in parallel. If, for example, postprocessing takes *N times* as long as basic data reception, the postprocessing can either be deferred until the data transmission session is completed, or it can be postprocessed and delivered in real-time by using *N* or more parallel streams to accomplish this.

Since the BP protocol does not require in sequence delivery, or even delivery of all parts of a payload, there is no additional requirement on how the payloads for bundle sequences are prepared, postprocessed, partitioned, or reassembled.

Both preprocessing and postprocessing can be separated from the raw machinery for transmission (which is the delivery to and retrieval from the lower protocol layers) and encoded in either software (using parallel threads of execution) or in hardware (e.g., using FPGAs or ASICs).

It would be more difficult, and likely not fruitful, to attempt to parallelize the preparation of a single bundle block for transmission, or the processing of a single block after reception, since many of the steps required are either inherently sequential, or require only minimal computation that could benefit from parallelization, e.g., filling in length fields, cf. Figure 5 in RFC-5050 [1], reproduced on the next page.



Reproduced from RFC-5050 [1], Section 4.5, p. 18

III. Conclusions

We conclude that parallelization of bundle protocol sessions is not inconsistent with either RFC-5050 [1] or CCSDS-734.2-B-1 [2], but that it does rely on matching capabilities in nodes along the forwarding chain, which cannot be presumed.

We also conclude that parallelization of the preparation of payload data at the sender, or its postprocessing at the receiver is independent of the BP protocol itself, and invisible to the protocol layers above and below the BP protocol. This approach, therefore, is not inconsistent with the recommendations of CCSDS 734.2-B-1 [2] or the definitions in RFC-5050 [1].

Acknowledgements

This study was performed by Gerard Holzmann, a JPL and Caltech affiliate with deep expertise in the field of software reliability and analysis. The study was initiated by Peter Shames, Manager of JPL Data System Standards Program, and reviewed for accuracy by Scott Burleigh, the lead editor for the creation of the DTN standard in CCSDS. The study was carried out for the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. It was funded by the NASA Space Communications and Navigation (SCaN) Standards Office as part of the JPL Data System Standards Program.

References

[1] RFC 5050, *Bundle Protocol Specification,* Request for Comments, K. Scott and S. Burleigh, Network Working Group, November 2007.

[2] Recommended Standard CCSDS 734.2-B-1, *Bundle Protocol Specification,* Blue Book, The Consultative Committee for Space Data Systems (CCSDS), September 2015.