Bundle in Bundle Encapsulation Overview

Scott C. Burleigh
IPN Group
McLean, USA

ORCID 0000-0003-3768-5413

Carlo Caini
DEI/ARCES
University of Bologna
Bologna, Italy
ORCID 0000-0002-6576-7320

Andrea Grano
ARCES
University of Bologna
Bologna, Italy
ORCID

Abstract—**The aim of this paper is to introduce Bundle in Bundle Encapsulation (BIBE). BIBE is an experimental extension of the Bundle Protocol (BP), the pillar at the foundation of the Delay-/Disruption- Tolerant Networking architecture, whose use is planned in next space mission to Moon, and, in longer terms, to build an Interplanetary network. Among BIBE features, the paper analyzes in detail the optional BIBE retransmission mechanism, which should offer all the advantages of the Custody Transfer option that was present in previous versions of BP, but without its problems. BIBE applications are potentially so numerous that they cannot all be examined in one paper, thus here the focus is on those that are non-security related, such as BIBE retransmission on both unidirectional and space links, transient QoS, and transient "critical" forwarding. All these cases have been exercised using the latest version of ION, the DTN suite maintained by NASA JPL, by means of an enhanced version of a pre-existing test program. The paper concludes with a detailed study of an example of BIBE traffic, based on the use of the Wireshark network analyzer, for which a specific BIBE dissector was designed by the authors.**

Keywords— Delay-/Disruption-Tolerant Networking, Inter-Planetary Networking, Bundle Protocol, ION, BIBE

I. Introduction

Delay-/Disruption-Tolerant Networking (DTN) architecture [1], [2] was devised as a means of enabling scalable, automatic communication over networks characterized by highly variable and/or lengthy lapses in connectivity whose duration cannot be anticipated by statistical analysis of traffic. The most familiar example of such networks is data interchange over interplanetary links among space exploration assets, but similar challenges are encountered in some terrestrial communication environments. DTN architecture differs from the Internet architecture in several profound ways, because communication within these challenging environments required the design and implementation of new "delay-tolerant" networking features.

At the core of the DTN architecture we have the insertion of the new Bundle Layer, between Application and Transport, with its associated Bundle Protocol (BP); this protocol had to be robust against long delays and link intermittency, as well as to address the same operational problems that continue to challenge the Internet itself, particularly with regard to security and quality of service, although in a delay-tolerant manner. BP was first formally defined in RFC 5050 (version 6, BPv6) [1], then adopted by the CCSDS (Consultative Committee for Space Data Systems) [4], and, more recently, with the new version 7 (BPv7), in IETF RFC 9171 [5].

As DTN deployment began to be contemplated for an ever-widening range of use cases, both in interplanetary space and on the surface of Earth, it became clear that very different configurations of BP would be needed not only in different parts of the DTN network, but also on different segments of a single path. As early as 2009 it became apparent that dynamic modification of the operational parameters of BP would entail modification of the BP data units – "bundles" – traversing those segments. To preserve the integrity of the bundles as originally issued, it was proposed to prepend new bundle "header" blocks to the original bundles, effectively encapsulating those original bundles in new bundles that were configured as required for the path segment that had to be traversed. This early conception of Bundle-In-Bundle Encapsulation (BIBE) was intended to support content-centric networking and defense against traffic analysis in DTN, among other innovations. The BIBE idea was then resurrected in 2013 but in a different form, i.e. as a "convergence layer" protocol as explained below.

One of the core architectural principles of DTN is the division of data forwarding responsibilities between BP, functioning as an overlay protocol that forwards data end-to-end over time-varying topology, and point-to-point transport protocols at an underlying "convergence layer." Convergence-layer protocols perform direct transmission between entities that are adjacent within that time-varying topology – that is, are the endpoints of individual data path segments. Since data transmission over path segments is necessarily delegated to convergence-layer protocols, a straightforward approach to segment conditioning would be to perform BIBE bundle encapsulation in a convergence-layer protocol – that is, to use BP itself at the convergence layer under BP when modification of path segment behavior is required. Virtual connectivity among BIBE-enabled nodes constitutes a further, smaller and simpler layer of overlay above the DTN network, such that a single segment of the BIBE overlay may subsume any number of segments of the underlying "real" delay-tolerant network. Current proposed specifications for BIBE implement this concept [6], but with the additional feature of optional BIBE retransmission, making BIBE a notionally "reliable" convergence-layer protocol.

Despite this long evolutionary development, or better because the concept until recently was evolving, BIBE has never been presented in the scientific literature, but only partially discussed inside standardization bodies. Now the concept is mature enough to be presented in the public literature, which is the aim of the present paper. Here we will refer to the fully functional BIBE implementation that can be found in the latest version of ION (Interplanetary Overlay Network), the DTN protocol suite supported by NASA-JPL (Jet Propulsion Laboratory) [7]. This implementation is based on the Internet draft [6], with minor differences.

BIBE as currently defined is intended to address several concerns, some related to BP security, others to reliability or QoS enforcement. As potential applications of BIBE functionality are actually too numerous to describe in one paper, those related to security will be left to future work. Here we will focus, instead, on the new BIBE reliability mechanism and other non-security-oriented applications, such as transient QoS and transient critical forwarding. The analysis is completed with the study of an example of BIBE traffic, based on the use of the Wireshark network analyzer, for which a brand new BIBE dissector was designed by the authors.

II. DTN Architecture and Bundle Protocol

As shown in Fig. 1, the DTN architecture relies on the insertion of the new "Bundle layer", between Application and lower layers (typically Transport) [2]. Note that between a DTN hop and the next, i.e. inside a DTN hop, we can have non-DTN subnets (represented by networks A, B and C in the figure). For example, we can assume that the 3-hop scheme of Fig. 1 applies to an Earth to Mars scenario where the first node on the left is on Earth, the second node a NASA gateway to space, the third a Lander on Mars, and the last a Rover. All non-DTN nodes are transparent to BP, which works as an "overlay" on top of them.

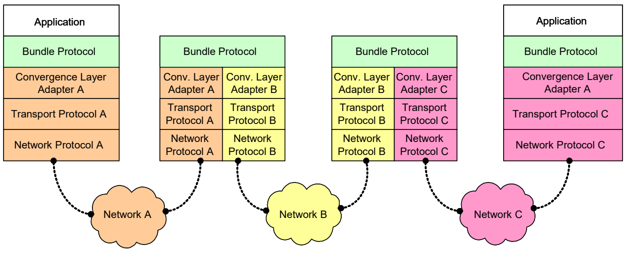

Fig. 1. DTN architecture and Bundle protocol stack

This first fundamental difference with respect to the TCP/IP architecture is the redefinition of the Transport layer, which is no more end-to-end but limited to one DTN hop, as evident from the figure. This redefinition is of the highest importance as it enables the adoption of different transport protocols on different hops, which is essential to counteract the different challenges posed by each hop. In our example, we could continue to use TCP on both Earth or Martian hops, where delay is low and link continuous, while the use of a specialized protocol such as LTP (Licklider Transmission Protocol) [9], [10] is in order in the central hop, between the terrestrial gateway and the lander, where the delay is vice versa huge and connectivity intermittent. The Convergence Layer Adapter (CLA) is a simple interface between the BP and the protocol below it, as shown in the figure.

A second fundamental difference with the Internet architecture is storage at intermediate nodes, which is necessary to cope with intermittent connectivity, or disruption. BP always stores bundles in local databases on each node, possibly even for long intervals (minutes or even hours) if the link to the next node is temporarily not available, e.g. because of lack of contacts.

Another important feature, crucial in the case of BIBE, is that bundles can optionally been retransmitted by a BP node. Bundle retransmission may happen either after a failure of a reliable convergence layer (e.g. if an LTP red session is cancelled), or when custodial retransmission is set. In the former case, RFC 5050 [1] and RFC 9171 [5] left the choice of retransmitting or not the bundle to BP implementations (most of them, including ION [7] and Unibo-BP [11] do it). In the latter case, custodial retransmission is performed in BPv6 by Custody Transfer, while in BPv7 is left to BIBE, as shown in the next session. In both cases, retransmission is optional.

III. From Custody Transfer to BIBE Retransmissions

Bundle retransmission is one of the most important features provided by BIBE, thus its significant differences from its predecessor deserve to be fully explained.

A. *Custody transfer in BPv6*

The original custodial transfer mechanism is described in general terms in RFC 4838 [2], with implementation details in RFC 5050 [1]. The mechanism is often misinterpreted, so it is worth summarizing the main points. The source, and only it, may raise the "custody transfer is requested" flag in the primary block. If raised, each BP node that receives the bundle, including the source and the destination nodes, is asked to become bundle "custodian". A key point is that a node is left free to accept the source request or not. In either case an administrative bundle, the "custody signal", is sent to the current custodian, if any (to this end, BPv6 bundles have the custodian endpoint identifier in the primary block).

To accept the custody implies a double commitment:

- Not to delete the bundle before either the arrival of a custody acceptance from another node or bundle expiration, whichever is first.
- To retransmit the bundle after one RTO (Retransmission Time Out), in the absence of custody notifications (positive or negative) from other nodes.

The first point is the more difficult to understand, as the difference between this and normal bundle processing is subtle. To clarify we must stress that in any case: a) the bundle must be saved in a local database; b) the bundle can be deleted when its lifetime expires. The difference is, thus, in the conditions for cancellation. In normal processing a bundle can be cancelled once BP receives a positive notification from the CLA in charge of transmitting the bundle to the next node. This may happen either at bundle transmission, if the CLA is unreliable (LTP green, UDP), or at completion of bundle transfer, in case of a reliable CLA (LTP red or TCP). By contrast, with the custody is necessary to wait for arrival of a custody acceptance from a new node that agrees to act as new custodian (see Fig. 2).

In the case of an unreliable CLA the difference is so evident as not to require further comment, while for a reliable CLA it depends which node agrees to act as new custodian. If the node is the next, it coincides with CLA destination and the custody transfer does not generally add very much to the reliable CLA, as both work within one DTN hop. Otherwise, as in Fig. 2, the custody really does add a new level of reliability, as it covers multiple DTN hops at once, which is clearly impossible for a reliable CLA. In other words, the custody transfer may protect against node failures, provided that they are not custodian. An example is in order. Suppose that intermediate nodes are flying drones, and that a drone is shot down. The bundles still in custody on a previous node, i.e. for which the drone did not accept custody, are not definitively lost, although the drone itself has been destroyed, which clearly shows the power of this mechanism.

Summarizing, we can say that custody transfer is peculiar as it is neither an end-to-end nor a hop-by-hop reliability mechanism, although it can be both in specific cases: its usefulness is greatest when unreliable CLAs are selected, for example to transfer a bundle on a unidirectional link (see next sections), or when some nodes on the path to destination have less chance of survival than other nodes and thus they will refuse the custody to maintain bundle protected by the current custodian, a safer node.

Unfortunately, in spite of all these advantages, custody transfer is affected by two severe problems:

- As the current custodian does not know which of any node will accept custody, it is basically impossible to set an adequate RTO.
- If a bundle in custody is fragmented by another node, it becomes impossible to manage custody acceptance signals.

Eventually, in view of these problems, custody transfer was removed from BPv7 specifications. The decision was made easier when it was realized that the same advantages could be achieved with the new mechanism, BIBE.

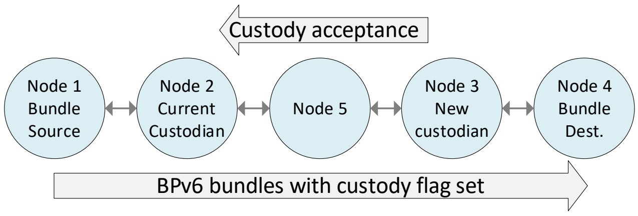

Fig. 2. An example of BPv6 custody transfer. Bundle in custody on node 2 must be acknowledged before the RTO expires by custody acceptance bundles sent back by the new custodian (node 3). However, as the current custodian does not know a priori which node will become the next custodian (i.e. that node 5 will refuse the custody while node 3 will accept it), it cannot properly set the RTO.

B. *Bundle retransmission in BIBE (BPv7)*

BIBE is an experimental extension of BPv7, still under development [6] but already available in ION and in a few other implementations. In brief, it tunnels bundles between a BIBE source and destination, thus creating a sort of virtual DTN hop between these two nodes. Formally, it appears as a CLA working on this virtual hop. BIBE retransmission is an optional feature of BIBE (as was custody transfer in BPv6), based on BRT (BIBE Reliable Transmission) signals sent back by BIBE destination to BIBE source (see Fig. 3).

The two main differences from the BPv6 custody transfer are the following:

- The BIBE virtual hop may encompass one or more physical DTN hops, but BIBE destination is always known to BIBE source; the RTO can therefore always be easily set, thus solving the first problem of BPv6 custody transfer.
- A BIBE bundle can be fragmented, but it is always defragmented before delivery to BIBE destination, which solves the second problem.

An additional feature is that multiple BRT signals can be aggregated in one bundle, which saves bandwidth on the return link. Other details and other features will be presented in the next sections.

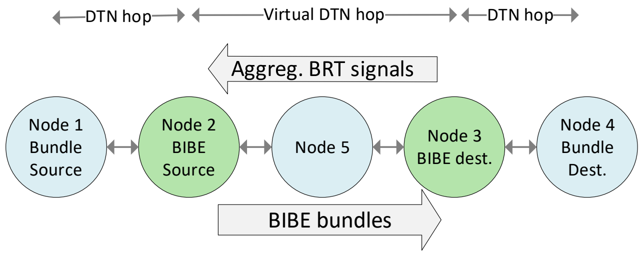

Fig. 3. BIBE creates a virtual DTN hop between BIBE source and destination and appears to the user as a CLA working on this hop. BIBE bundles must be acknowledged by BRT signals before the RTO expires. As BIBE source knows in advance the BIBE destination, the RTO setting is manageable.

IV. A Few BIBE Applications

Although applications are numerous, due to space constraints here we will limit the treatment to a few selected cases, not however involving bundle security.

A. *BIBE reliability on unidirectional links*

One of the most interesting features of custodial retransmission in BPv6 is the possibility of achieving BP reliability in the presence of unidirectional links, provided that an alternative return path is available at BP layer. This feature is preserved in BIBE.

Let us consider the topology shown in Fig. 4 as a possible application example. The BP source and destination, node 1 and 4, are connected by two intermediate nodes, 2 and 3. The link between 2 and 3 is at high speed, but unfortunately also unidirectional, hence reliability cannot be enforced at link layer (neither LTP red nor TCP can be used on it). An alternative path between 3 and 2 is however possible, via node 5, but the bandwidth available on this path is scarce. In such a situation, custodial retransmission solves the problem, by providing at BP layer what is impossible at link layer. The solution consists in applying BIBE retransmission on bundles forwarded by node 2 to node 3 on the direct unidirectional link. BRT signals, generated by 3, are sent back to node 2 via node 5. If a bundle is lost, BIBE will retransmit it after the RTO timer expires, and so on. The solution is elegant, effective and simple.

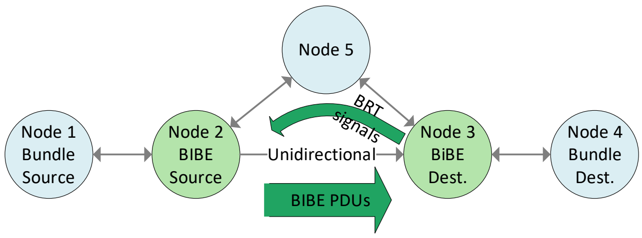

Fig. 4. Topology of the unidirectional link application. Reliability which cannot be provided at link layer on the unidirectional link 2-3, is provided at BP layer by BIBE: BRT signals generated at node 3 are possibly aggregated and forwarded to 2 via the low bandwidth alternative path.

B. *BIBE reliability on interplanetary links*

In specific cases, BIBE retransmission can be useful even in the presence of bidirectional links. As an example, consider the topology shown in Fig. 5. At first glance, it seems a simpler variant of the previous case, where this time the 2-3 link is bidirectional, which allowed us to drop node 5. Now reliability can be enforced simply by using a reliable CLA, such as LTP red, between 2 and 3, as usual. In such a situation, BIBE retransmissions, i.e. reliability at BP layer, makes sense when it is not convenient to use a reliable CLA. This could be the case of an Earth to Mars optical link, where the huge bandwidth delay product would impair the use of ARQ at link layer; instead, it could be advantageous to use Packet-Layer FEC, which relies on redundancy packets to cope with possible frame losses [12], [13], [14]. PL-FEC could thus be conveniently coupled with an unreliable CL, such as LTP green. One lasting problem is that the use of PL-FEC could reduce but not nullify the chance of a bundle loss, which would happen if frame losses exceeded the recovery capability of the code. BIBE retransmissions offer an effective solution to this, as the original bundle would be retransmitted after one RTO unless confirmed by the BIBE destination by means of a positive BRT signal.

In these very long delay links, an additional advantage of retransmission at BP layer, instead of at link layer, is that the route of the bundle is checked again before retransmission. This check can have three possible outcomes: a) a possible better route may be selected, b) the bundle is dropped as there are no more routes compatible with bundle lifetime, c) the bundle is sent again on the old path. While there is no advantage in the last case, the first two are advantageous, thus proving the usefulness of BIBE.

Of course, it may be objected that there is no point in retransmitting a full bundle instead of a part, even if this happens rarely with PL-FEC. Although this is true on ordinary terrestrial networks, where ARQ is the best solution, the above discussion should at least have made clear than when dealing with very high-speeds and long propagation delays, the situation is not the same, and alternative solutions, as that shown here, need to be investigated.

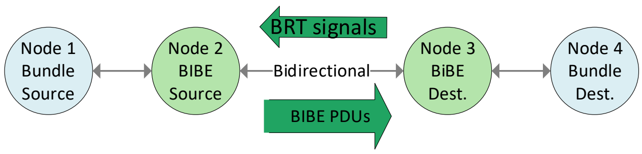

Fig. 5. Topology of the bidirectional interplanetary link application. Reliability is provided at BP layer by BIBE retransmissions (the CLA between 2 and 3 is unreliable): BRT signals generated at node 3 are directly forwarded to 2 as the link is bidirectional.

C. Transient quality of service

Retransmissions are an important feature offered by BIBE, but not the sole. Luckily for us (and the reader), the other advantages are more obvious, as they are somewhat similar to those provided by other tunneling schemes (e.g. for IP). A first possible application is enforcing a different QoS between two nodes of the network. Before examining this case, we must first remind the reader that QoS in BPv7 is totally in charge of extension blocks, such as ECOS (Extended Class of Service) [15]; this was first introduced for BPv6 when it was realized that the three BPv6 priority levels where not enough. As it was clearly impractical to introduce a new BP version just to accommodate more levels, the solution was to introduce a second level of priority for the "expedited" class, and to insert these new "ordinal" priorities in a bundle extension entirely devoted to QoS, that is ECOS. BPv7 took a further step by removing the standard priorities (now called "cardinal") from the primary block, with the eventual aim of adding them to a future QoS extension. In the meantime, cardinal priorities have been added to other QoS parameters in the new ECOS version for BPv7 [16], which although work in progress, it is at present available not only in ION, but also in DTNME [8] and Unibo-BP [11].

As always with QoS, the problem is that parameters are set by the source, which often has little interest in requiring anything less than the highest level of service. In BPv6, priority cannot be modified by intermediate nodes, as it is written in the primary block, but RFC 4838 already stated that priorities should not be interpreted as absolute, but relative to a source, thus offering implementers and network managers with an escape mechanism to enforce their desired priority rules, e.g. "the traffic originated by our nodes must always be treated preferentially compared to traffic generated by others, no matter the priority". Very recently, ESA researchers suggested the introduction of two distinct QoS extensions, one immutable, with values set by the source, and one modifiable by intermediate nodes, for example at the borders of a subnetwork. BIBE allows us to obtain the same result in an alternative way, because the encapsulating bundle naturally has its own QoS (at present its ECOS block), which is set by the BIBE source and that can differ from that set by the bundle source.

As an example, consider the topology shown in Fig. 6, where only essential nodes are represented. Nodes 1 and 4 are BP source and destination, while 2 and 3 are BIBE source and destination, as usual. The latter two are at the border of a DTN subnet managed by a different space entity. Node 2 can enforce any desired QoS on BIBE, which will apply until node 3 is reached. Once again, the BIBE solution is elegant, effective and simple at the same time.

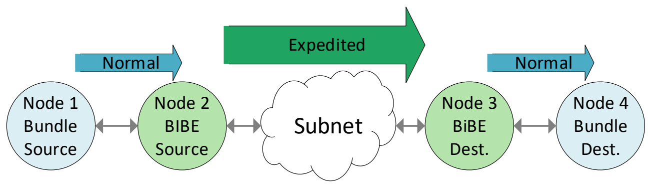

Fig. 6. Topology of the transient QoS application. Normal priority bundles are encapsulated in expedited priority BIBE bundles, thus enforcing a transient priority on the subnet between BIBE source and destination.

D. Transient "critical" forwarding of selected bundles

Ordinary bundles are forwarded in a single copy by the SABR (Scheduled-Aware Bundle Routing) algorithm, the CCSDS standard for routing in space networks [17], [18].

Starting from the knowledge of contacts, SABR tries to find a set of candidate routes (sequences of contacts) to get to the bundle destination in time (i.e. before bundle lifetime expires), then it selects the fastest route and forwards the bundle to the entry node of this route. For safety, the route is recomputed at each node, as SABR cannot ensure actual availability of the selected path, either because of random variability of contacts, or because of the impact of traffic inserted by other nodes. To avoid this kind of problem, the original bundle QoS was extended by introducing the concept of "critical" bundles (bundles flagged as critical in ECOS). They are treated preferentially by SABR, which adopts for them controlled flooding instead of single-copy forwarding. A critical bundle is thus forwarded in multiple copies to all (but also sole) neighbors which have a chance of delivering the bundle on time.

To control the amount of flooding, BIBE allows the network manager to set the critical bundle in a transient way, i.e. only to traverse an “unstable” portion of the network, where SABR may have difficulties in providing acceptable performance for the reason explained above. In this way the flooding induced by the critical flag is confined to the sub-set of the network where it is actually necessary.

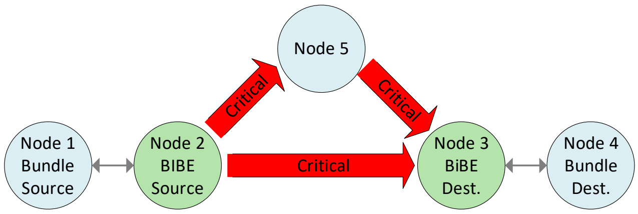

Fig. 7. Topology of a network with transient “critical” forwarding of selected bundles. Network instability between node 2 and 3 can be counteracted by setting the “critical” ECOS flag on BIBE bundles. As a result, BIBE bundles flagged as critical are forwarded in copy to all neighbors (nodes 3 and 5, instead of either 3 or 5 in the figure).

V. Tests

An interesting feature of ION is that it can be used to instantiate multiple DTN nodes on the same machine. This possibility is used by the tests contained in the “test” directory of ION, which are designed to check the correct implementation of the most significant features of the bundle protocol. These tests only require the user to launch a script file, called “dotest” and check if the test was successful or not at its end. Building on the pre-existing “bibect” test in ION-4.1.3, we designed a new version, called “bibe_advanced”, able to reproduce, with minimal changes, all the applications shown in the previous sections. This new test will be possibly added to the original one in future ION versions. In the meantime, it can be downloaded from [19].

A. The bibe_advanced layout and its configuration files

In practice, we added one node, namely 5, to the original four-node configuration of “bibect”, thus obtaining a layout similar to that represented in Fig. 4, which contains all the other as particular cases. For better clarity in configuration files, we decided to use LTP on all links except that from 2 to 3, where UDP was maintained to make use on this link of the “owltsim” channel emulator. This is an executable of ION in charge of introducing losses, which works only on UDP datagrams. Configuration files do not change when multiple nodes are launched from the same machine instead of distinct machines, with one important exception. As all nodes send and receive on the loopback IP address, as they are on the same physical machine, port numbers must be modified so that to serve not only as multiplexers, but also as addresses. In practice, it is enough to append the node number to the standard port; for example, instead of port 1113 for an LTP “induct”, we used 11131 for node 1, 11132 for node 2, etc. We also modified the contact plan so that it is possible to enable/disable the use of intermediate node 5 as desired, by just commenting a line.

B. Tests performed

By using a single Linux machine with ION 4.1.3, and starting from the bibe_advanced layout, we have successfully reproduced all cases described in the previous section. For brevity, we will describe in full details only the test related to custodial reliability on a unidirectional link, which is the most complex; all the others will be only briefly summarized, being trivial variants. The study is based on the results provided by the Wireshark traffic analyzer, to which we have added the BIBE and the ECOS dissectors that we developed on purpose (the extensions can be downloaded from [20]).

1) BIBE reliability on unidirectionl links

The layout is that of Fig. 4. The test consists in sending 3 small bundles of normal priority from node 1 to node 4; as the “owltism” emulator is configured to deterministically drop one bundle over every three, we expect the third bundle to be retransmitted by BIBE.

What happens can be better described with the help of the Wireshark screenshot shown in Fig. 8 (the use of the loopback interface brings the serendipitous advantage of having all traffic in one file). The first bundle is sent from node 1 to node 2 (frame no.1); it is then encapsulated in a larger BIBE bundle sent to the owltim simulator (no.4) and then forwarded to node 3 (no.5). These two lines are identical, but the two cases can be differentiated by inspecting port numbers, as we did. After being decapsulated, the original bundle is sent to node 4 (no. 6). The BRT signal generated on 3 is sent to 5 (no. 9) and from here forwarded to 2 (no.10), thus confirming the arrival of the original data bundle. These two lines can be differentiated by inspecting port numbers, as before. Note that the BIBE ACK is much smaller than the original data bundle. The same repeats after 1s for the second bundle (frames 11-20), as expected.

The third bundle has a different story: it is sent to node 2 (no.21) and then encapsulated in a BIBE bundle to owltsim (no.24) as usual, but the BIBE bundle this time is silently discarded by the simulator. As a result, nothing happens until, after about 4s, the BIBE RTO expires on node 2, causing the retransmission of the original bundle in a new BIBE bundle (no.25). From here on the test continues as usual (no. 26-31).

No.	Time	Source	Destination	Protocol	Length
1	0.000…	ipn:1.1	ipn:4.1	BPv7	139
4	0.004…	ipn:2.0	ipn:3.0	BPv7 BIBE PDU	216
5	0.089…	ipn:2.0	ipn:3.0	BPv7 BIBE PDU	216
6	0.093…	ipn:1.1	ipn:4.1	BPv7	140
9	0.961…	ipn:3.0	ipn:2.0	BPv7 BRT signal	132
10	0.965…	ipn:3.0	ipn:2.0	BPv7 BRT signal	132
11	1.006…	ipn:1.1	ipn:4.1	BPv7	139
14	1.007…	ipn:2.0	ipn:3.0	BPv7 BIBE PDU	216
15	1.090…	ipn:2.0	ipn:3.0	BPv7 BIBE PDU	216
16	1.092…	ipn:1.1	ipn:4.1	BPv7	140
19	1.960…	ipn:3.0	ipn:2.0	BPv7 BRT signal	132
20	1.961…	ipn:3.0	ipn:2.0	BPv7 BRT signal	132
21	2.013…	ipn:1.1	ipn:4.1	BPv7	139
24	2.014…	ipn:2.0	ipn:3.0	BPv7 BIBE PDU	216
25	5.952…	ipn:2.0	ipn:3.0	BPv7 BIBE PDU	218
26	5.997…	ipn:2.0	ipn:3.0	BPv7 BIBE PDU	218
27	5.998…	ipn:1.1	ipn:4.1	BPv7	141
30	6.961…	ipn:3.0	ipn:2.0	BPv7 BRT signal	132
31	6.962…	ipn:3.0	ipn:2.0	BPv7 BRT signal	132

Fig. 8. Wireshark screenshot of the reliability on unidirectional link application. BIBE dissector has been added to the standard package.

By comparing the length of the first two frames, we can deduce that the overhead introduced by the BIBE encapsulation is of less than 80 bytes, which is a negligible value in most cases, as data bundles are usually much larger than IP packets. This is not the case in the experiment, where, for greater clarity, we wanted to have one bundle contained into one frame.

2) Other tests

To verify BIBE reliability on a bidirectional link, we repeated the previous test, but without node 5 (it was enough to comment the last line of the contact plan and relaunch the "dotest" file); results are not reported here for brevity. To enforce a different priority on BIBE bundles, it was sufficient to set the priority field in the only instruction present in the "amroc.biberc" file of node 2, to a priority different from the original one. Actually, we already did it in the BIBE reliability test just examined, where the priority of BIBE bundles was set to expedited, while the original priority was normal, thus checking both transient QoS (Fig. 6) and BIBE reliability (Fig. 4) at the same time. Analogously, for all other QoS flags, including critical forwarding.

C. Tips on Wirshark configuration and other remarks

If on the one hand the use of the sole loopback interface facilitates the collection of frames, on the other hand it impairs the analysis for two reasons. First, Wireshark relies on standard UDP or TCP ports to identify the right protocol dissector for the encapsulated protocol, e.g. the UDP port 1113 is associated to LTP. By using different ports, such as 11131, 11132, etc. all frames appear on the main Wireshark window as UDP frames, which makes the analysis impossible. To overcome this problem, it is however possible to manually inform Wireshark of the new matches. These settings are saved in a file (./config/wireshark) and can be easily moved from one machine to the other. For the user's convenience, the file used for bibe_advanced, "decode_as_entries", and the original capture "bibe.dmp" are both included in [19].

VI. CONCLUSIONS

This paper has presented BIBE, an experimental extension of BP. Particular attention has been devoted to the analysis of its bundle retransmission mechanism, to show that it can offer the same advantages as the old "custody transfer" system supported in BPv6, without the problems that eventually led to the omission of custody transfer from BPv7. As examples of the utility of BIBE retransmission the paper has considered the cases of unidirectional links and interplanetary high-speed links. Other BIBE applications, such as transient QoS and transient "critical" forwarding have been discussed as well. All these application examples have been tested using the ION implementation of BIBE, by extending a pre-existing test. Results have been analyzed in detail by means of the Wireshark traffic analyzer, augmented by a BIBE dissector specifically developed by the authors. The hope is that this effort will contribute to widespread adoption of BIBE, including its inclusion in DTN standards.

REFERENCES

[1] Burleigh S, Hooke A, Torgerson L, et al. Delay-tolerant networking: an approach to interplanetary Internet. *IEEE Commun Mag*. 2003; 41(6): 128-136. https://doi.org/10.1109/mcom.2003.1204759

[2] V. Cerf , A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, H. Weiss "Delay-Tolerant Networking Architecture," Internet RFC 4838, Apr. 2007.

[3] K. Scott, S. Burleigh, "Bundle Protocol Specification", Internet RFC 5050, Nov. 2007.

[4] CCSDS 734.2-B-1 "CCSDS Bundle Protocol Specification", CCSDS Blue Book, Issue 1, Sept. 2015.

[5] S. Burleigh, K. Fall, E. Birrane, "Bundle Protocol Version 7", Internet RFC 9171, Jan. 2022.

[6] S. Burleigh, "Bundle-in-Bundle Encapsulation" , IETF Internet Draft, Jul. 2024, work in progress, https://datatracker.ietf.org/doc/draft-ietf-dtn-bibect/

[7] S. Burleigh, "Interplanetary Overlay Network: An Implementation of the DTN Bundle Protocol,"in the Proc. of 4th IEEE Consumer Commun. and Networking Conference, 2007, pp. 222-226, doi: 10.1109/CCNC.2007.51. Code: https://github.com/nasa-jpl/ION-DTN

[8] DTNME (DTN Marshall Enterprise) code: https://github.com/nasa/DTNME

[9] Ramadas M, Burleigh S, Farrell S. Licklider Transmission Protocol—motivation, Internet RFC 5326, 2008.

[10] Ramadas M, Burleigh S, Farrell S. Licklider Transmission Protocol—specification, Internet RFC 5327, 2008.

[11] C. Caini and L. Persampieri, "Design and Features of Unibo-BP, the Unibo Implementation of the DTN Bundle Protocol," in *IEEE Journal of Radio Frequency Identification*, vol. 8, pp. 458-467, 2024, doi: 10.1109/JRFID.2024.3358012 (Open Source)

[12] T. de Cola and M. Marchese, "Reliable data delivery over deep space networks: Benefits of long erasure codes over ARQ strategies", *IEEE Wireless Commun.*, vol. 17, no. 2, pp. 57-65, Apr. 2010.

[13] N. Alessi, C. Caini, T. de Cola and M. Raminella, "Packet Layer Erasure Coding in Interplanetary Links: The LTP Erasure Coding Link Service Adapter," in *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, no. 1, pp. 403-414, Feb. 2020, doi: 10.1109/TAES.2019.2916271.

[14] N. Alessi, C. Caini, A. Ciliberti and T. de Cola, "HSLTP—An LTP Variant for High-Speed Links and Memory Constrained Nodes," in *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, no. 4, pp. 2922-2933, Aug. 2020, DOI: 10.1109/TAES.2019.2958190.

[15] S. Burleigh, "Bundle Protocol Extended Class Of Service (ECOS)" IRTF draft (expired), July 2013, https://datatracker.ietf.org/doc/html/draft-irtf-dtnrg-ecos-05.

[16] S. Burleigh, F. Templin, "Bundle Protocol Extended Class Of Service (ECOS)", Internet draft (expired), May 2021. https://datatracker.ietf.org/doc/ draft-burleigh-dtn-ecos/

[17] CCSDS 734.3-B-1. Scheduled-Aware Bundle Routing, recommended standard, Blue Book, 2019. https://public.ccsds.org/Pubs/734x3b1.pdf

[18] C. Caini, G. M. De Cola, L. Persampieri, "Schedule-Aware Bundle Routing: Analysis and Enhancements", *International Journal of Satellite Communications and Networking*, vol. 39, no.3, pp. 237-243, May/June 2021. DOI: 10.1002/sat.1384

[19] The bibe_advanced test: https://gitlab.com/ccaini/bibe_advanced

[20] BIBE and ECOS dissectors for Wireshark: https://gitlab.com/ccaini/unibo-wireshark