Network Working Group
Internet-Draft
Intended status: Experimental
Expires: October 30, 2010

D. Brown
Raytheon BBN Technologies
S. Farrell
Trinity College Dublin
S. Burleigh
Jet Propulsion Laboratory
April 28, 2010

# DTN Bundle Age Block for Expiration without UTC
## draft-irtf-dtnrg-bundle-age-block-00

## Abstract

As originally specified, [RFC5050] presumes that any DTN node will have access to accurate real world time. Experience has shown that there are devices and networks where accurate real world time is difficult or impossible to consistently obtain.

This draft proposes an extension block that contains the current age of a bundle in order to support bundle expiration for nodes and networks that have faulty, intermittent, or no notion of the real world time. Bundle age may be used to expire bundles by a Bundle Protocol Agent which does not have access to accurate real world time. The Age must be updated at each hop in order to maintain accuracy.

## Status of this Memo









## Copyright Notice

Table of Contents







## 1. Requirements Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119.

## 2. Other Terminology

This document distinguishes between devices which are only able to measure elapsed time and those which have access to global time. Access to global time will be referred to as Coordinated Universal Time (UTC) whether the node stores UTC directly or can infer it based on the local wall clock time and current time zone. Devices which do not have access to UTC will be referred to as having "node local" or just "local" time.

Accuracy refers to the ability of a node to maintain correct elapsed or UTC time since the last synchronization information received. Lack of accuracy is also referred to as clock drift.

Precision refers to the granularity of the time representation. For example, microseconds is higher precision than milliseconds.





## 3. Introduction

Experience has shown that clock drift in DTN nodes is sometimes unavoidable and has detrimental effects on the protocol. The detrimental effects are magnified for bundles sourced with short lifetimes.

Additionally, [RFC5050] compliance is not possible when devices do not have access to accurate UTC via either synchronization or an accurate, persistent battery-backed UTC clock. An [RFC5050]-compliant DTN implementation currently requires either an accurate UTC clock or a battery-backed RTC and the consistent availability of synchronization signals.

There is a variety of scenarios where neither of these requirements can be met. Many COTS devices such as cell phones, smartphones, and military radios contain no internal battery suitable for a persistent RTC, and so provide no time when powered on outside the reach of provider infrastructure.

In the case of smartphones, these devices are generally tamper-resistent and as such offer no reasonable means for changing an internal battery. Military devices tend to eschew internal consumer oriented batteries which may leak, preferring instead external hardened battery packs which may be disconnected frequently, making a persistent clock impractical.

## 4. Age Extension Block

This document proposes an Age Extension Block (AEB), which denotes the time since the bundle has been created, with microsecond precision.

The Age Extension Block format below includes the [RFC5050] required block header fields.

```
+----------------+----------------+-------------------+------------+
|  Block type    | Proc. Flags (*)|  Block length(*)  |   Age(*)   |
+----------------+----------------+-------------------+------------+
```

(*) Self-Delimiting Numeric Values (SDNVs). See RFC 5050 Sec. 4.1

Support for the AEB by BPA implementations is RECOMMENDED for interoperability but not required.

The Age field is defined to represent the approximate elapsed number of microseconds since the creation of the bundle.

The "Block must be replicated in every fragment" bit must be set for the AEB. This also dictates that the AEB must occur before the payload block per [RFC5050] Sec. 4.3.





## 5. Applicability

Tracking bundle age solely via the AEB is insufficient for applications where a bundle spends an indeterminate amount of time in suspension. When a bundle with a zero-valued CreationTimestamp is stored to persistent media, for example, and the time of its storage is unknown or inaccurate, its age cannot in general be determined with any reasonable accuracy upon later being accessed.

An example of this situation is when a bundle with a zero-valued CreationTime is stored on a USB mass storage device regardless of whether it is treated as a DTN link or node. Unless the time of storage is tracked separately or known to be accurately stored on the filesystem, then the Age is unknown upon access.

See also Section 6.6.

## 6. Age Block Processing

### 6.1. At Nodes without AEB Support

Nodes which do not support the AEB must have access to UTC time and therefore can only expire bundles on the basis described in [RFC5050].

To improve interoperability with BPAs that implement the support for the AEB, whenever a BPA that does not support processing the AEB receives a bundle with creation time zero the BPA MAY use zero as 'the current time' for the purposes of section 5.5 of RFC5050 with respect to treatment of that bundle. When implemented, this mechanism prevents deletion of the bundle due to an incorrectly computed expiration time.

All further specification of AEB treatment applies only to nodes which support the AEB unless stated otherwise.

### 6.2. At nodes with AEB support

It is expected that implementations which support the AEB will have a means of tracking the elapsed time a bundle is resident at a node in order to appropriately update the AEB age field upon delivery to a local endpoint or forwarding to another node, or to determine the time a bundle should be expired.

### 6.3. Expiration

If the AEB is supported by a receiving node, the bundle MUST be treated as expired if Age > Lifetime.

### 6.4. Upon Bundle Creation

Since a zero-valued Creation Time field is used to signal that the sender does not have access to accurate UTC, then a BPA MUST NOT create a bundle with both a zero-valued Creation Time and no AEB.

For the sake of interoperability it is RECOMMENDED that an AEB be provided whenever it is not impractical to do so.

#### 6.4.1. At nodes with UTC

There may be DTNs where all nodes have accurate realtime clocks, and bundles are not expected to travel to other networks. In these cases, A BPA MAY add a bundle age extension block when creating a bundle. In all other cases, where it is possible that bundles may be received by nodes without accurate realtime clocks, the AEB SHOULD be





added at creation time.

If the BPA has access to UTC upon creation of a bundle, it SHOULD place the current UTC into the Creation Timestamp field when creating a bundle.

#### 6.4.2. At nodes without UTC

If a BPA does not have access to UTC or chooses not to set the Creation Timestamp on UTC, a BPA MUST create an AEB with value 0 and set the Lifetime field to the desired time to live for the bundle.

### 6.5. Upon BPA Enqueuing to CLA

#### 6.5.1. At nodes with UTC

Any time a bundle is enqueued at a CLA for transmission by a BPA with access to UTC, the BPA SHOULD first update the AEB age field as UTC - CreationTimestamp. This applies whether the bundle originated at the node or this node is forwarding a bundle originating at another node.

#### 6.5.2. At nodes without UTC

If UTC is unavailable, the AEB age field should be increased by the time which has elapsed since the age field was last updated or if the age field was not updated, by the elapsed time since the bundle was received. This applies whether the bundle originated at the node or this node is forwarding a bundle originating at another node.

### 6.6. Upon Retrieval from Persistent Storage

A bundle with a zero-valued CreationTime and with an indeterminate age SHOULD be treated as expired upon being read from persistent storage. This situation arises, for example, when a node without access to UTC accesses bundles from persistent storage after power cycling. Such a node cannot determine the elapsed time that a bundle has spent in persistent storage across power cycles.

Bundles with a non-zero CreationTime MAY be forwarded since it may be possible for some node with UTC to accurately update the AEB age field.

### 6.7. At CLA Transmission and Reception

In some networks a convergence layer and/or the CLA may impose non-negligible delays. In deep space networks, propagation delay can be significant. Other CLAs may impose other delays, for example CLAs which provide some notion of reliable delivery to multiple neighbors.

A CLA SHOULD convey additional delays imposed either by non-neglible propagation delay or non-negligible queuing delay at the CLA. The CLA implementation should make provisions for either the sender or receiver or some combination of sender and receiver to provide this information.

This representation SHOULD be made available to the receiving BPA as an elapsed value conveyed by the CLA to the BPA with the bundle.

### 6.8. Upon Reception by BPA

In general, a DTN node should maintain an accurate representation of a bundle’s age so that the bundle can be accurately expired and the AEB field can be accurately maintained across transmissions. Each time the bundle is delivered to a local endpoint or forwarded to another node, the AEB should be made to reflect the age of the bundle as accurately as possible. This implies that nodes without UTC will need to store the UTC or node-local time associated with the reception of a bundle in order to later determine the elapsed resident time and accurately update the AEB age field upon transmission or delivery, or to determine the UTC or node-local time at which the bundle should expire. The age field is updated as Age = Age + ElapsedTime, where ElapsedTime = NodeLocalTime - RecordedNodeLocalTime or ElapsedTime = UTC - RecordedUTC.

The BPA SHOULD take into account elapsed time spent at a CLA if the CLA provides this information. The age field should be updated upon reception by the BPA in this case by Age = Age + ElapsedTimeAtCLA.

### 6.9. While Bundle Resident at BPA

A resident bundle whose age exceeds its lifetime while residing at a node should be expired. Note that age in this context needs to include the bundle’s AEB age field and any elapsed time while resident at the node which is not presently accounted for in the age field.





## 7. Interoperability

Interoperability can be achieved between nodes which support AEB or between nodes which have access to UTC. Since the AEB provides the necessary time information for a node without UTC to process the bundle, the only circumstance in which interoperability cannot be achieved is between an implementation which does not support the AEB (and which therefore must have access to UTC), and another node which does not have access to UTC.

If a bundle is sourced by a UTC node without an AEB, nodes without UTC cannot reasonably process the bundle. If a bundle is sourced by a node without UTC (and must therefore have an AEB), this bundle cannot be reasonably processed by a UTC node which has no AEB support (with the possible exception of being allowed to forward the bundle without delay, see Section 6.1).

This interoperability issue may be partly mitigated by the provision of a gateway node which adds AEB extension blocks to bundles which are sourced without one. This allows nodes without UTC to process bundles sourced by UTC nodes that do not support the AEB.

For the time being, interoperability can only be fully realized in networks which contain only nodes with UTC or in networks where all nodes implement the AEB. See Section 8.2.

### 7.1. Bundle Forwarding Examples

#### 7.1.1. UTC to non-UTC

A UTC node which supports the age extension block creates a bundle which has a UTC timestamp for the creation field, and presumably a small or zero-valued AEB age field. The bundle is forwarded to a non-UTC node. The non-UTC node examines the age field, compares Age to Lifetime and determines that the bundle is still valid. The node also associates the node-local time with the bundle as soon as it arrives. Upon retransmitting the bundle or delivering the bundle to an application, presuming it has not expired, the node calculates the AEB age field as: Age = Age + UTC - RecordedUTC.

#### 7.1.2. Non-UTC to UTC

A Non-UTC node can only process bundles which have an AEB and so we can presume that a bundle forwarded from a Non-UTC node has an AEB. We will also presume for this example that the bundle originated like it did in the previous example at a UTC node and therefore has a non-zero CreationTimestamp. In this case the bundle arrives at the receiving UTC node which, seeing the non-zero CreationTimestamp ignores the AEB and processes the bundle as described in RFC 5050. Upon forwarding the bundle to a next hop, the UTC node updates the Age field as: Age = UTC - CreationTimestamp.

If the bundle was instead sourced at a Non-UTC node, then the bundle has a zero-valued CreationTimestamp. Upon receiving this bundle, the UTC node records the bundle's UTC time of arrival. Upon transmitting or delivering this bundle, the node updates the AEB age field based on UTC - RecordedBundleUTC.

### 7.2. Interaction with Fragmentation

A BPA needs to fragment a bundle which is larger than the MTU imposed by the CLA over which the bundle will be forwarded. In that case, the BPA creates bundle fragments which are themselves bundles. These bundles may be forwarded at different times and therefore must carry different age values. Because of this, the "Block must be replicated in every fragment" bit must be set for the AEB, and each bundle fragment must have its AEB age field appropriately set according to the specifications contained here.

### 7.3. Security

When security is a concern and since the AEB age field can change at each hop, the AEB MAY be encrypted on a hop-by-hop basis via the Bundle Security Protocol provided by [I-D.irtf-dtnrg-bundle-security] Section 2.5. In that case, the Security-destination MUST be present and MUST specify the EID of the next forwarding hop.





## 8. Future Considerations

### 8.1. IANA Considerations

An IANA block type registration for the AEB will need eventually be created.

### 8.2. Incorporation of Age into Bundle Primary Block

It is strongly recommended that specification of Age at bundle inception and the processing of Age values become mandated by moving the Age value in some form into the Bundle primary block at some future time. This will improve interoperability and precision of bundle expiration without detrimental effect on expiration semantics for current [RFC5050] implementations.

### 8.3. Margin of Error for Time Values

As previously shown, the AEB's age may contain some error. Propagation delay that is difficult or impossible to account for is one potential source of error. This type of error may accumulate at each hop. Another potential source of error is an inaccurate RTC. Nodes which have a somewhat synchronized but potentially inaccurate clock require some means for expressing the potential inaccuracy of Creation timestamps for sourced bundles.

In the former case, a Margin Of Error (MOE) field associated with the Age value seems like a reasonable mechanism for extending bundle lifetime in the face of accumulated Age error. The MOE field represents plus-or-minus uncertainty. For example, a 5 second MOE indicates that the Age is expected to be accurate to within +/- 5 seconds.

A bundle SHOULD NOT be considered expired unless Age - AgeMOE - CreationMOE > Lifetime.

In the latter case, a node with a somewhat synchronized RTC might create bundles with a non-zero Creation timestamp. In this case, the Age value can be considered a more accurate representation of the bundle's age than CurrentTime - CreationTime. However, without being able to represent this state of affairs, a node with an accurate RTC may incorrectly adjust the Age value since it may only presume that the CreationTime is accurate.

Considering MOE values for Age, Creation, RTC, the bundle SHOULD be expired if and only if Age - CreationMOE - AgeMOE > Lifetime or RTC - RTCMOE > Lifetime.

Here is a graphical depiction of MOE for Age, Creation time and RTC:

```
    ================== Lifetime ====================
                          |
                      ____|
                          |\
                + RTCMOE  | \
                      ----|  } <-- RTC
                - RTCMOE  | /
                      ____|/
                          |
                          |____
                         /|
                        / |   + AgeMOE
               Age --> {  |----
                        \ |   - AgeMOE
                         \|____
                          |
                          |
                          |____
                         /|
                        / |   + CreationMOE
          Creation --> {  |--
                        \ |   - CreationMOE
                         \|____
                          |

                              Margin of Error
```

This would seem to argue for an eventual specification of margin of error for some or all time fields specified in the bundle. Since these considerations involve additional complexity and potential changes to [RFC5050] itself, they are only noted in this document as future considerations and not treated normatively for the protocol.





## 9. References

[I-D.irtf-dtnrg-bundle-security]
Symington, S., Farrell, S., Weiss, H., and P. Lovell, "Bundle Security Protocol Specification", draft-irtf-dtnrg-bundle-security-15 (work in progress), February 2010.

[RFC4838] Cerf, V., Burleigh, S., Hooke, A., Torgerson, L., Durst, R., Scott, K., Fall, K., and H. Weiss, "Delay-Tolerant Networking Architecture", RFC 4838, April 2007.

[RFC5050] Scott, K. and S. Burleigh, "Bundle Protocol Specification", RFC 5050, November 2007.

Authors' Addresses

Daniel W. Brown
Raytheon BBN Technologies
10 Moulton St.
Cambridge, MA 02138
US

Email: dbrown@bbn.com

Stephen Farrell
Trinity College Dublin
Distributed Systems Group
Department of Computer Science
Trinity College
Dublin 2
Ireland

Phone: +353-1-896-2354
Email: stephen.farrell@cs.tcd.ie

Scott Burleigh
Jet Propulsion Laboratory
4800 Oak Grove Drive, m/s 301-490
Pasadena, California 91109
USA

Phone: +1-818-393-3353
Email: scott.c.burleigh@jpl.nasa.gov