

DTN Research Group
Internet-Draft
Intended status: Experimental
Expires: May 27, 2010

S. Farrell
A. Mc Mahon
Trinity College Dublin
J. Ott
Helsinki University of Technology
November 23, 2009

Handling Issues with Real Time in the Bundle Protocol
draft-farrell-dtnrg-alt-time-00

Abstract

The Bundle Protocol ([RFC 5050](#)) requires the use of real time clocks to handle bundle expiry. This sometimes has some drawbacks; this specification explains some such situations, briefly considers generic ways to ameliorate those, and defines some alternative approaches for handling bundle expiry that do not require real time clocks.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on May 27, 2010.

Copyright Notice

Copyright (c) 2009 IETF Trust and the persons identified as the document authors. All rights reserved.

Internet-Draft

Handling BP RTC Issues

November 2009

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the BSD License.

Table of Contents

1.	Introduction	3
2.	Issues Addressed	5
3.	AltTime Mechanisms	6
4.	Security Considerations	9
5.	IANA Considerations	10
6.	Acknowledgements	11
7.	References	12
7.1.	Normative References	12
7.2.	Informative References	12
	Authors' Addresses	13

1. Introduction

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 [[RFC2119](#)].

[[Editorial and other comments to be resolved are noted in double square brackets thusly.]]

The Bundle Protocol (BP) specified in [RFC 5050](#) [[BP](#)] implements the delay tolerant networking (DTN) architecture specified in [RFC 4838](#) [[DTNARCH](#)]. The BP has a creation time field in its primary block. This field, together with a lifetime field is used to determine when a bundle should be expired, after which it is considered safe to delete the bundle. The creation time field contains the wall-clock time at which the bundle was created. Some experiments with the BP have shown that there are nodes for which maintaining an accurate real time clock (RTC) is either hard or impractical. Other experiments have shown that managing expiry using a time window is problematic since it can be hard to know how large a window to use. This specification briefly outlines those problems and defines some new mechanisms for handling bundle expiry that do not require an RTC.

It is worth noting that even though the above problems are real, they nonetheless have not prevented real experiments with the BP, which though using an RTC, is actually a very undemanding user.

The "model" DTN node assumed here is a device that:

- may or may not have a battery-backed RTC, in general we may assume not,

- operates within a reboot cycle, i.e. it reboots occasionally,

- can keep some notion of (strictly) monotonically increasing sense of time across reboots, i.e. has some form of non-volatile

storage, (it is, after all implementing [RFC 5050](#)), and,

so that the node's sense of time may bear some relation to real time even though the skew may be significant.

So we can speak of the device having a "clock," even if that is not an RTC.

In such a device if malfunctioning or malicious software overwrites the clock, then the node's bundles may be dropped by other nodes due to the presence of the bad value in the bundle creation time.

Clocks may also vary randomly due to users changing timezones or (rarely) manipulating their clocks for other reasons (or having some semi-broken wannabe smart software running that does it for them). If you try to rely on operator-supplied time on your mobile and you will sometimes find it jumping back and forth, e.g., by one hour as happened to one of the authors recently.

This specification defines extensions to the BP that can be used even in scenarios where a DTN node's clock is not sufficiently accurate so that other DTN nodes will forward its bundles.

[[Future versions of this I-D could specify:

- some general rules defining how a node is supposed to operate if it has no assured notion of time;

- maybe some text on how other nodes should deal with this, i.e., accept that they may talk to people who are out of sync;

- some procedures to assess if someone you are talking to is way off (or you are); we could have an evil-style bit in a handshake indicating that "I may not have a good clue of time"; but I am not so sure what one could imply if this bit was not set;

- an NTP-like scheme for setting your clock based on values seen in the BP.

Depending on the final scope of this draft, we may change the title to "Handling Bundle Expiry without Real Time Clocks" or something

else that reflects consensus within the DTNRG.

]]

[2.](#) Issues Addressed

The issues below have been noted in experiments using the BP, and are addressed by this specification. Other issues may well exist, but are not `[[yet]]` tackled here.

There are some DTN nodes that have no RTC. `[[need a citation if that's not just theory]]` In such a case, the node cannot determine a good value to place in the creation time field of the bundle primary block.

Similar to the above, but much more common, is the case where a node's RTC has never been properly set, (e.g. the first time the node is powered on), or where the RTC has a bad time, due to one form of error or another. This can happen due to software bugs (scribbling on a memory mapped RTC) or if the node's power is "dirty," fluctuating voltages can corrupt an RTC.

We expect that both issues above are much more common at the extreme periphery of networks, since nodes that are reasonably well connected, even for short periods, can easily synchronise their clocks to a level sufficient for the BP.

Separately from the above, the requirement that bundles expire a fixed number of seconds after their creation time has also occasionally proven problematic. In this case, the issue is how to select a sensible value for a particular bundle. In some DTNs that can be a difficult task, leading either to bundles that consume storage for too long (e.g. if multi-copy routing is in use), or bundles that expire for no good reason, sometimes just before they would otherwise have been successfully delivered.

3. AltTime Mechanisms

We define two new mechanisms for handling bundle expiry. The first uses a hop count and the second, called the deferred window scheme, uses a time window where the clock only starts ticking after the bundle has reached some node that does know the wall-clock time.

In both cases, since we do not want to modify the primary block (to retain interoperability), we define some specific time values that act as indicators that one or other of the new schemes is intended to be used. Note however, that a BP agent that implements [RFC 5050](#) strictly might drop all such bundles, if it considers that the time window defined in the primary block is too large, so the use of these new schemes requires some co-ordination.

Both of the new schemes require that some bytes in the bundle are changed en-route. In order to allow for data-integrity we therefore use two new extension blocks, one that is fixed when the bundle is created, (the AltTime block) and one that may be modified en-route, (the CurAltTime block).

The AltTime block specifies which scheme(s) are to be used to expire the bundle and contains associated parameters, i.e. the max hop count and/or the window size. The CurAltTime block can contain the hops-remaining value and/or the window start time value (and an EID for the source of time used).

[[This is just a sketch to be filled in after the RG discusses things for a bit. (Not a lot...a bit:-)]]

An AltTime block has a domination rule, specifying which expiry schemes the sender wishes be applied to this bundle. There are three schemes (the one from [RFC 5050](#) and the two defined here). This field is an SDNV containing a bit mask with a one value meaning that the scheme applies. In the case where more than one scheme applies, then the bundle MAY be expired (and hence possibly deleted) when any of the schemes indicates that the bundle has expired.

[[The domination rule could be more complex, e.g. specifying other combinatorics, or could be simpler, e.g. only allowing one scheme to apply. We're not sure which is better.]]

If the domination rule indicates that the [RFC 5050](#) expiry scheme does not apply then the primary block MUST contain special values [[TBD]] indicating that the bundle creation timestamp and lifetime fields do not contain real values.

[[Are there any values that'd be easier or harder to handle in

existing [RFC 5050](#) implementations? We suspect not, in which case just some creation time sufficiently far in the past and some sufficiently long lifetime should be ok.]]

If the domination rule indicates that the hop count scheme applies then the AltTime extension block MUST contain a hop count value which is a non-zero SDNV.

[[A fixed width field here could be ok. Zero could mean infinity, but would we want that?]]

If the domination rule indicates that the deferred window scheme applies, then the AltTime block MUST contain a window size field that contains the number of seconds (as an SDNV) for which the bundle should be considered non-expired, after some RTC-capable node has added a window start time to the CurAltTime block.

[[Is seconds the right granularity?]]

Nodes that implement this specification MUST ensure that the CurAltTime block is present (it may have to be added by the first such node) and contains values that reflect the correct operation of the scheme(s) that apply.

For the hop count scheme this means ensuring that the CurAltTime block contains the hops-remaining count, which is the number of hops that remain before this bundle will be considered to be expired.

When the hop-count scheme is being enforced and the bundle has one hop remaining (hops-remaining in CurAltTime has the value one) then the bundle MAY be considered expired if it is not delivered directly to the destination EID.

[[Is the above correct?]]

For the deferred window scheme, the first node implementing this specification that considers that it has a good RTC MUST add the current value at the time of arrival of the bundle to the window start time field.

[[Any issue here with when to start the window? Could be the time of arrival, time of departure or something in between.]]

Once a window start time value is present in the CurAltTime block, then it MUST NOT be changed by subsequent nodes.

When the deferred window scheme is in operation, once the current time is later than the sum of the window start time from the

bundle MAY be considered to be expired.

4. Security Considerations

These schemes could allow a bad actor attempting a DoS attack to more easily consume resources in a DTN, for example, if all paths in a particular network under attack were less than N hops, then setting a hop count of N+1 would maximise the damage done by DoS bundles in an efficient manner (for the bad actor).

Since DoS-mitigation in DTNs is still a research area, one can only recommend requiring authentication of bundles [[BPsec](#)] as a way of at least making the bad actor accountable.

The CurAltTime block MAY be authenticated, and not authenticating that allows any node to easily cause another node to incorrectly expire a bundle.

[[Do we have a suitable ESB ciphersuite in the BSP for this?]]

If a node adds a window start time with a bad value (e.g. with the time significantly in the past), then that might act as an inadvertent DoS.

[[More TBD]]

[5.](#) IANA Considerations

For now, there are none. If an IANA registry is established for BP block types then entries in that registry would be required.

[6.](#) Acknowledgements

[[Whoever discusses this. Kevin Fall already dislikes the fact that this conflates the schemes:-)]]

[7.](#) References

[7.1.](#) Normative References

- [BP] Scott, K. and S. Burleigh, "Bundle Protocol Specification", [RFC 5050](#) , April 2007.
- [BPsec] Symington, S., Farrell, S., Weiss, H., and P. Lovell, "Bundle Security Protocol Specification", [draft-irtf-dtnrg-bundle-security-12.txt](#), work-in-progress, November 2009.
- [RFC2119] Bradner, S. and J. Reynolds, "Key words for use in RFCs to Indicate Requirement Levels", [RFC 2119](#) , October 1997.

[7.2.](#) Informative References

- [DTNARCH] 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.

Authors' Addresses

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

Alex Mc Mahon
Trinity College Dublin
Distributed Systems Group
Department of Computer Science

Trinity College
Dublin 2
Ireland

Phone: +353-1-896-2354
Email: alex.mcmahon@cs.tcd.ie

Joerg Ott
Helsinki University of Technology
Otakaari 5 A
Espoo FIN 02150
Finland

Email: jo@netlab.hut.fi