

DTN Research Group
Internet-Draft
Intended status: Experimental
Expires: February 13, 2010

S. Symington
R. Durst
K. Scott
The MITRE Corporation
August 12, 2009

Delay-Tolerant Networking Custodial Multicast Extensions
draft-symington-dtnrg-bundle-multicast-custodial-06

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/1id-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 February 13, 2010.

Copyright Notice

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

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents in effect on the date of publication of this document (<http://trustee.ietf.org/license-info>). Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

Abstract

This document defines optional extensions to the Bundle Protocol [[refs.DTNBP](#)] for supporting the custodial multicast delivery of bundles within a Delay-Tolerant Network (DTN). The protocol extensions described herein may be used to support custody transfer and custody-based reforwarding during the transmission of a bundle from a single source node to multiple destination nodes.

Table of Contents

1.	Introduction	5
1.1.	Related Documents	5
1.2.	Terminology	6
2.	The Basic Problems of Custodial Multicast	8
3.	Objectives, Assumptions, and Design Principles	9
3.1.	Receiver-driven versus Source-driven Multicast Models	9
3.2.	Custodial Multicast Objectives	10
3.3.	Assumptions and Design Principles for Custodial Multicast Extensions	11
4.	Multicast Routing Protocol Requirements and Desired Features	14
4.1.	Only CMC-nodes may be branching points	14
4.2.	Every bundle copy must be able to be individually accounted for	14
4.3.	If convergence layer multicast is used, the forwarding node must know the number of next-hop recipient nodes	16
4.4.	CMC nodes should keep track of de-registration requests in order to be able to distinguish delayed de-registrations from routing failures	16
5.	Overview of Mechanisms for Supporting Custodial Multicast	19
5.1.	Special Responsibilities of Custodial Nodes	21
5.2.	Special Responsibilities of (Non-Custodial) Branching Nodes	22
6.	Identifying Bundle Copies	24
7.	Proxy EID Extension Block	26
8.	Bundle Protocol Extensions to Support Custody Transfer and Retransmission for Multicast Bundles	28
8.1.	Definitions	28
8.2.	Bundle Processing Steps for Custodial Multicast Bundles	28
8.2.1.	Bundle Forwarding	28
8.2.2.	Forwarding Contraindicated	31
8.2.3.	Forwarding Failed	32
8.2.4.	Bundle Reception	32
8.2.5.	Local Bundle Delivery	33
8.2.6.	Custody Transfer	33
8.2.7.	Custody Acceptance	33
8.2.8.	Custody Release	34
8.2.9.	Custody Transfer Success	34
8.2.10.	Custody Transfer Failure	35
8.2.11.	Bundle Deletion	38
8.3.	Reception of Custody Signals	38
9.	Custodial Multicast and the Retransmission Block	39
10.	Performance Impact	40
11.	Security Considerations	41
12.	IANA Considerations	42
13.	References	43

[13.1](#). Normative References [43](#)
[13.2](#). Informative References [43](#)
Authors' Addresses [45](#)

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 [RFC 2119](#) [refs.[RFC2119](#)].

Multicasting as defined in the DTN context is the ability of a source bundle node to transmit a bundle to a destination multicast endpoint without necessarily having to originate a separate bundle for each bundle node that is registered in that endpoint. The Bundle Protocol [[refs.DTNBP](#)] is designed to support delivery of bundles to all types of endpoints (singleton, anycast, and multicast), but it only supports custodial delivery (custody transfer and custody-based re-forwarding) of bundles that are sent to singleton endpoints. Its custodial delivery mechanisms are not applicable to bundles sent to multicast endpoints.

This document defines optional extensions to the Bundle Protocol [[refs.DTNBP](#)] for supporting the custodial multicast delivery of bundles within a Delay-Tolerant Network (DTN). The protocol extensions described herein may be used to support custody transfer and custody-based reforwarding during the transmission of a bundle from a single source node to multiple destination nodes.

The Bundle Protocol extensions to support custodial multicast that are described in this document are OPTIONAL for deployment with the Bundle Protocol. Use of these extensions is also OPTIONAL. Bundle Protocol implementations that claim to be custodial-multicast-capable (CMC) MUST support all of the features defined herein, as well as all of the features defined in the Bundle Protocol [[refs.DTNBP](#)]. In addition, in any given network that purports to support custodial multicast delivery, a multicast routing protocol is also required to set up the multicast distribution path. The extensions defined in this document are multicast-routing-protocol-independent in the sense that they are designed to work with any multicast routing protocol, providing that the multicast routing protocol meets certain requirements, which are defined in [Section 4](#).

1.1. Related Documents

This document is best read and understood within the context of the following other DTN documents:

- The Delay-Tolerant Network Architecture [[refs.DTNArch](#)] defines the architecture for delay-tolerant networks, but only briefly discusses multicasting.

- The Bundle Protocol Specification [[refs.DTNBP](#)] defines the format and processing of the blocks used to implement the basic bundle protocol. This document does not explicitly discuss DTN multicast, but it does define two concepts that are important for supporting multicast:

the concept of endpoints that contain multiple nodes, all of which are intended recipients of a bundle, and

the concept that a node may both deliver and forward a bundle and, when forwarding a bundle, may forward it to multiple next-hop nodes.

The Bundle Protocol also defines custody transfer procedures that apply to bundles that are destined for singleton endpoints; these custody transfer procedures, however, are explicitly stated to be applicable only to bundles that are destined for singleton endpoints and, therefore, by implication, not to be applicable to bundles that are destined for multicast endpoints.

- Delay-Tolerant Networking Bundle-in-Bundle Encapsulation [[refs.Encaps](#)] defines an encapsulation-specific application agent capability and a bundle payload format for placing one or more bundles inside of the payload field of an encapsulating bundle's Bundle Payload Block. This capability is useful, for example, for encapsulating one or more multicast bundles inside of a singleton bundle for custodial retransmission to a specific downstream DTN node.

[1.2.](#) Terminology

Multicast Endpoint - A bundle endpoint that is permitted to contain multiple nodes and that has as its minimum reception group (see [[refs.DTNBP](#)]) all of the nodes registered in that endpoint.

Branching Point - A bundle's delivery path has a branching point at a particular node if in the course of delivering the bundle to its minimum reception set that node either:

-delivers the bundle at that node and forwards it to at least one next hop node, or

forwards the bundle to more than one next hop node.

A node that is a branching point for a bundle is said to "branch" that bundle.

Singleton Bundle - a bundle that has a singleton endpoint ID as its

destination.

Multicast Bundle - a bundle that has a multicast endpoint ID as its destination.

Custodial Multicast Bundle - a Bundle that has a multicast endpoint ID as its destination and that has its custody transfer requested flag (in the bundle processing flags field) set to 1.

Custodial-multicast-capable (CMC) node - a node that implements all of the features defined in this document, as well as all of the features defined in the Bundle Protocol [[refs.DTNBP](#)].

Non-Custodial-multicast-capable (non-CMC) node - a node that does not implement all of the features defined in this document and all of the features defined in the Bundle Protocol [[refs.DTNBP](#)].

2. The Basic Problems of Custodial Multicast

Custody transfer and retransmission, which are defined for singleton bundles in the Bundle Protocol [[refs.DTNBP](#)], are fundamentally complicated when applied to multicast bundles. Upon receipt of a multicast bundle, a bundle node may deliver the bundle at that node, forward the bundle to one or more next-hop nodes, or both. If a node delivers a bundle and forwards it to at least one next hop node or forwards the bundle to more than one next hop node, then the bundle's path is said to have a branching point at that node. Branching points cause additional copies of a bundle to be created. A node that takes custody of a multicast bundle and then delivers and/or forwards it becomes responsible for all copies of that bundle that it delivers or forwards. In addition, nodes may be branching points for bundles without being custodians for those bundles, so a node that takes custody of a bundle is not only responsible for all copies of that bundle that it delivers or forwards; it also becomes responsible for all copies of the bundle that may be created as a result of downstream branching at other, non-custodial, branching nodes, until the copy is deleted, custody of the copy is transferred to another node, or the copy is delivered. Fulfilling this responsibility for all downstream copies is complicated by the fact that the custodian does not necessarily have any way of knowing whether or how many times a bundle that it forwards will be branched downstream.

Therefore, to support custodial delivery of multicast bundles, this document defines mechanisms to enable a custodian of a multicast bundle to determine when all downstream copies of a bundle have either been delivered or have been taken custody of by a downstream node, and to determine when a specific downstream undelivered bundle may need to be retransmitted. In addition, for purposes of conserving bandwidth, it defines a mechanism to enable bundle copies to be reforwarded selectively, to only those downstream branches of the delivery path that have not yet received them, rather than being forwarded indiscriminately to all downstream nodes.

3. Objectives, Assumptions, and Design Principles

3.1. Receiver-driven versus Source-driven Multicast Models

The protocol extensions described herein have been designed from the perspective of a receiver-driven multicast model. We believe that they are also applicable to a source-driven multicast model, though they might not be optimal for use in a source-driven model. We will explore the applicability of these extensions to source-driven multicast in future revisions of this I-D.

A receiver-driven multicast model differs from a source-driven multicast model insofar as in a receiver-driven multicast model, the source node does not necessarily need any knowledge of the number or the identities of the nodes that are members of the multicast endpoint. In a source-driven multicast model, on the other hand, the source node needs to know the identity of each node in the multicast endpoint. In DTN terminology, this means that in a source-driven multicast model, the source node needs to know the EIDs of each of the singleton endpoints in which each of the nodes that are members of the multicast destination endpoint are also members. Because the source in a source-driven multicast transmission knows the identities of the nodes in the multicast destination endpoint, source-driven multicast can include a list of the destination node singleton EIDs in the bundle and use unicast routing information to route the bundle to these destination nodes. Receiver-driven multicast, on the other hand, requires the formation of a multicast tree along which the bundle will be routed from the source to the nodes that are members of the destination multicast endpoint.

In terms of the source node being able to have certainty regarding the delivery of bundles delivered via custodial multicast, the source in a source-driven multicast model, because it knows the singleton EIDs of each intended destination node, can be notified as to whether or not each intended destination node actually received a copy of a given multicast bundle. The source in a source-driven multicast model, therefore, is able to determine whether or not all intended recipients of the multicast bundle actually received it. The source node in a receiver-driven multicast model, however, because it is unaware of the number of intended destination nodes and of the singleton EIDs of which they are members, is not able to determine whether all intended recipients of the bundle actually received it. On the other hand, using the receiver-driven model, if a node registers in a multicast endpoint after a bundle has been sent from the source to that endpoint, it is possible for the late-joining node to receive the multicast bundle. Using a source-driven model, however, the source must know the identity of each destination node before the source initiates the bundle transmission. If the source

knows the identity of a destination node in advance and includes this destination node in the list of intended recipients, it is possible for this late-joining node to receive the multicast bundle; however, if the source is made aware of an additional destination node to which a multicast bundle should be sent after transmission of the bundle has been initiated, the bundle cannot reach that destination node. A separate, unicast transmission of the bundle would have to be initiated to reach that node.

3.2. Custodial Multicast Objectives

The objectives of the custodial multicast extensions defined in this document are to increase the likelihood that

- each custodial multicast bundle that is sent will be delivered to as many of its destination nodes as possible prior to bundle expiration and,

- if a custodial multicast bundle needs to be re-forwarded in the course of being transmitted from source to destination, the cost of reforwarding it from the custodian (in terms of the routing metric in use) will be less than the cost of reforwarding it from the source node or, ideally, though not necessarily, from any previous custodian nodes.

A third objective of custodial multicast delivery, which is not an objective that is addressed in this document, is to enable delivery of bundles to a node whose registration request for the multicast endpoint may be late or delayed. Even though this registration request had not yet propagated through the network sufficiently to graft the destination node onto the distribution tree when the bundle was sent, a custodian of the bundle can forward the bundle toward the late-registering node when the custodian is notified of its registration.

If a bundle has a singleton destination EID, then custodial delivery enables it to be stored in the network until the destination node registers to receive it and notification of this registration request is able to propagate to the custodian (providing the bundle doesn't expire first) so that the custodian can forward the bundle toward the destination. Once such a bundle that is sent to a singleton EID reaches the (single) node that registers with that EID, the bundle may be deleted at its current custodian because it will not need to be delivered to any other destination node. If a bundle has a multicast destination EID, on the other hand, there is no inherent limit to the number of such registrations that may be received. Custodians of multicast bundles may store those bundles in the network until they expire in order to ensure that the bundle will be

available for forwarding to every node that registers late or the registration request of which is delayed. As pointed out in [[refs.wenruiZhao](#)], due to the unique characteristics of frequent partitioning and consequently large transfer delays in DTNs, destination endpoint registration changes during data transfer may be the norm rather than the exception.

The protocol extensions defined in this document focus only on defining mechanisms to meet the first two objectives presented above. They do not address the mechanics of re-forwarding multicast bundles to late-registering nodes.

3.3. Assumptions and Design Principles for Custodial Multicast Extensions

As discussed in the introduction, these extensions address custodial multicast transmission for the receiver-driven multicast model.

A node may be a branching point for a bundle without taking custody of that bundle. It is not feasible to expect or require that every branching point node will always take custody of a bundle. The extensions defined in this document, therefore, take into account the possibility that some branching point nodes may not become custodians of the custodial multicast bundles that they branch.

Although we cannot require all branching nodes to become custodians for bundles that they branch, we do require all branching nodes to maintain a small amount of additional state for each custodially-transferred bundle that they branch. Maintaining this state information enables non-custodial branching nodes to keep track of the custody status of each copy of the bundles that they branch. Given that a bundle's custodian has no awareness of the existence of copies of the bundle that are created at non-custodial branching nodes downstream of it, the non-custodial branching nodes must act as "proxy custodian" in the sense that they must keep track of the custody status of each copy they create and in turn report this status upstream to the bundle's actual custodian.

Custodial branching nodes must store a copy of the bundle, re-forward it if necessary, and maintain custody state per bundle copy that they branch rather than just per bundle (as currently defined in the Bundle Protocol). Non-custodial branching nodes are not required to retain a copy of the bundle or re-forward it, but they are, like custodial nodes, required to maintain custody state per bundle copy that they branch (see [Section 5](#) and [Section 8.2.1](#)).

If a node's forwarding information indicates that it must branch a multicast bundle but it does not have the storage capacity or other

resources necessary to pose as a proxy custodian by maintaining custody state per copy of the bundle that it branches then the node must report this difficulty to the previous upstream custodian or proxy custodian and it must not deliver or forward the bundle as a custodially transferred bundle (see [Section 8.2.1](#)).

Unless a custodian is going to re-forward a bundle when needed, the custodian cannot be of any use in helping to achieve the two objectives that we have identified as our objectives for custodial multicast. Therefore, the extensions we define must include not only the capability to re-forward a multicast bundle when necessary, but the ability to determine when re-forwarding is necessary. In the singleton case, a custodian may receive an explicit notification (in the form of a "Failed" custody signal) or an implicit notification (in the form of an expired custody timer) that it should retransmit a bundle. In the singleton case, a custodian may also receive explicit notification (in the form of a "Succeeded" custody signal) when a bundle of which it has custody has been either delivered or taken custody of by another node. When enough time has elapsed, as measured by the custody timer timing out, during which the custodian has not received a "Succeeded" custody signal for a bundle, the custodian may re-forward the bundle.

In the multicast case, in order for the custodian of a multicast bundle to know whether or not it needs to re-forward that bundle, the custodian need not necessarily know the delivery status of each downstream copy of the bundle. It does, however, need to know whether or not there is at least one downstream copy of the bundle that has not yet been delivered or taken custody of. If there is, and the custody timer for that copy expires, the custodian must re-forward the bundle copy downstream to at least the next-hop node(s) with which the expiring timer is associated. A major component of the extensions that we are defining, therefore, is a mechanism by which a custodian can be notified when all downstream copies of a bundle for which it is custodian have either been delivered or taken custody of. A second component is a mechanism whereby the custodian can be explicitly notified of an undelivered downstream copy of a bundle that has been deleted. In our extensions, analogous to the way that custody signals operate for the singleton case in the Bundle Protocol, receipt of a "Succeeded" custody signal for all copies of the bundle that a custodian has branched indicates that all downstream copies have either been delivered or taken custody of. Failure to receive a "Succeeded" custody signal before a custody timer times out indicates that one or more bundle copies associated with that timer has not yet been delivered or taken custody of (and thus may need to be retransmitted). Receipt of a "Failed" custody signal indicates that the referenced bundle copy was deleted before being delivered or taken custody of (and thus may need to be

retransmitted).

There is an inherent tradeoff between robustness of delivery and bandwidth conservation when custody transfer is requested for a multicast bundle. These extensions give bandwidth conservation priority over robustness of delivery by default, but local policy may override this. Unless overridden by local policy that specifically attempts to adapt the multicast delivery path in response to network dynamism, custodians and non-custodial branching nodes, by default, should not forward a multicast bundle copy for which a successful custody signal has already been received. Bundles should only be re-forwarded to those next-hop nodes that have downstream copies of the bundle that have not yet been delivered or taken custody of. Because the multicast delivery path is assumed to be a tree (see [Section 4](#)), if a node receives a bundle of which it is currently custodian, the bundle is assumed to be in an unintentional loop and must be dropped without sending any concomitant custody signal to the asserted custodian of the received bundle.

For a more detailed discussion of the objectives, assumptions, and design principles underpinning these protocol extensions for supporting custodial multicast, see [[refs.IEEEcustMcast](#)].

4. Multicast Routing Protocol Requirements and Desired Features

In order for the custodial multicast extensions defined in this document to work correctly, the multicast routing protocol that is used to set up the delivery path for custodial multicast bundles must meet certain requirements, which are discussed in the following subsections.

4.1. Only CMC-nodes may be branching points

As discussed in [Section 3.3](#) and [Section 5.2](#), non-custodial branching nodes are required to have the capability to maintain state information per bundle copy that they branch. Because this special capability is required of all branching nodes, in order for custodial multicast transfer to work correctly, either:

- the network used for custodial multicast delivery must consist only of CMC-nodes, or
- the multicast routing protocols responsible for delivery path formation must be capable of distinguishing CMC-nodes from non-CMC-nodes and they must enforce the restriction that only CMC-nodes are permitted to be branching points.

If there are nodes that serve as branching points for multicast bundles that do not have the capabilities defined in this document, the custody transfer procedures defined in this document will not work correctly. Therefore, in order to support custodial multicast in a network that consists of both CMC and non-CMC nodes, the multicast routing protocols used must enforce the restriction that only CMC nodes are permitted to be branching points in the delivery path. In order for multicast routing protocols to be able to distinguish CMC-nodes from non-CMC-nodes, when CMC-nodes register, they must make known their status as CMC-nodes and this information regarding their status as CMC-nodes must be propagated along with the registration information among the nodes that participate in multicast routing.

4.2. Every bundle copy must be able to be individually accounted for

The extensions defined herein are based on the principle that if a bundle is branched because branching is required to enable the bundle to reach two different destination nodes, then both the original bundle and the branched bundle must be acknowledged with a successful custody signal. On the other hand, if a bundle is branched but both the original bundle and the branched bundle are destined for the same destination node, then only one of these copies of the bundle must be acknowledged with a successful custody signal. If a branching node

forwards two copies of a bundle that are not distinguishable from each other and some custodial downstream node receives both of these copies at the same time, then one of the copies will be deleted and will not be acknowledged with a successful custody signal. The custodian that had branched the bundle will receive only one successful custody signal and, assuming it is expecting two, it will retransmit the unacknowledged bundle copy unnecessarily. On the other hand, if a branching node forwards two copies of a bundle that are distinguishable from one another and some custodial downstream node receives both of these copies at the same time, then because these copies are distinguishable the receiving node may generate two successful custody signals--one for each copy of the bundle that it receives.

The extensions defined herein require that every bundle copy that is destined for its own destination node (such that no other bundle copy is destined for that same destination node) must be able to be individually accounted for in terms of custody signaling. This requirement may be achieved in one of two known ways:

- DTN multicast routing protocols must operate over trees rather than meshes such that a multicast delivery path is always a tree. If the multicast distribution path is guaranteed to be a tree, then it is necessarily the case that every copy of a bundle that is branched is destined for its own destination node and the extensions defined herein are correct in requiring each copy to be acknowledged with a successful custody signal. Any copy that is not acknowledged with a successful custody signal must necessarily be retransmitted until it a successful custody signal is received for it.

- Every bundle copy that is created must contain a unique custodian EID, and nodes receiving two copies of the same bundle that are identical except for their custodian EID fields must send a separate successful custody signal for each copy of the bundle that they receive (to be sent to the EID in the bundle's custodian EID field). In this case, if the multicast distribution path is a mesh rather than a tree, there is no danger that a bundle copy that is received will not be acknowledged with a successful custody signal in the event that the receiving node already has custody of a copy of that bundle that was received earlier, because the earlier copy will have a distinct custodian EID from the recently received copy. Note that this second method of ensuring that each bundle copy will be acknowledged requires receiving nodes to be able to distinguish between two copies of the same bundle that have different custodian EID values, but that are otherwise identical. This is a capability over and above what is required of basic bundle nodes in the Bundle Protocol

specification.

4.3. If convergence layer multicast is used, the forwarding node must know the number of next-hop recipient nodes

If DTN multicast is running over an underlying convergence layer protocol that has inherent best-effort multicast transmission capabilities (e.g. a Local Area Network) and DTN multicast endpoint IDs are associated with underlying convergence layer multicast addresses, a bundle received at a DTN endpoint ID that is bound to an underlying convergence layer multicast address could in turn be multicast out toward its multiple destinations using that convergence layer. Using the best-effort multicast capabilities provided by some convergence layers, there is no way for the forwarder of such a multicast bundle to know in advance the expected number of recipients. Although from the standpoint of the DTN bundle protocol agent only a single copy of a bundle is being sent on a particular multicast-capable convergence layer, in effect there are multiple next-hop nodes that this single copy is intended to reach. When a best-effort multicast-capable convergence layer is used to distribute DTN multicast bundles such that the forwarding node does not know the number of expected next hops that are reachable via the convergence layer, it is impossible for the bundle's custodian to know if and when all destinations have received the bundle.

If unicast forwarding is being used on a given convergence layer, the receipt of a "Succeeded" custody signal associated with a particular bundle copy forwarded using that convergence layer means that all downstream copies of that copy have been either delivered or taken custody of. If multicast forwarding is being used on a given convergence layer then, although only a single bundle is being forwarded, this bundle may have multiple next hops. Therefore, in order for such a configuration to be able to be used to support custodial multicast, the bundle node that is forwarding the bundle onto the convergence layer:

- must be a CMC-node, and

- must know the number of next-hop nodes that the bundle is expected to reach using that convergence layer, so that it can maintain state for this number of copies.

4.4. CMC nodes should keep track of de-registration requests in order to be able to distinguish delayed de-registrations from routing failures

In the same way that there is a delay between when a node registers and when that registration propagates through the network to graft

that node onto the distribution tree, there may also be a delay between when a node de-registers with a multicast EID and when that de-registration propagates through the network to prune that node from the distribution tree. As a result, a situation may arise in which a de-registration request has reached some nodes but not others such that a bundle could be forwarded by a custodian (which has not yet received notice of the de-registration request) and later received at some downstream non-branching node (which has received the de-registration request) that does not have any next-hop nodes to which the bundle should be forwarded (because the node that recently de-registered was the only node downstream of this node that had been in the multicast endpoint). In this situation, the bundle cannot be forwarded because there is no known route to its destination. Nor should it be forwarded, because there is no longer an intended destination downstream of it. If the node at which this situation occurs has kept track of the fact that it received a de-registration notification for the multicast EID in question, it can distinguish this type of situation from a real routing failure. Instead of sending a "Failed" custody signal, the node that cannot forward the bundle because a downstream node has recently de-registered from the multicast EID should send a "Succeeded" custody signal for this multicast bundle, because a "Succeeded" custody signal indicates that there are no remaining copies of the bundle downstream of this node that need to be either delivered or taken custody of. If the node at which this situation occurs has not kept track of the de-registration notifications it has received and so cannot distinguish this situation from a real routing failure, it will send a "Failed" custody signal for this multicast bundle to the bundle's current custodian, and the custodian will re-forward the bundle toward the node at which the routing failure occurred. Eventually either the bundle will time out or the de-registration request notification will propagate to a branching point upstream such that the routing failure will no longer exist. Hence, having nodes in the multicast distribution tree keep track of de-registration requests is not required in order for the custodial multicast extensions to work correctly, but it does optimize bandwidth conservation by eliminating unnecessary custody signal transmissions and bundle retransmissions on the affected branches of the distribution tree. State on de-registration request notifications received does not have to be maintained indefinitely at any given node. De-registered endpoint IDs may be stored with a record of the time at which the de-registration request notification was received. The de-registered endpoint ID may be deleted from the state information when enough time has elapsed to allow the de-registration request to reasonably be assumed to have propagated through the network. Again, if this amount of time is not estimated correctly and the de-registered endpoint ID is deleted too soon from the state information, custody transfer will still work correctly; it will just not be optimized to

conserve bandwidth.

5. Overview of Mechanisms for Supporting Custodial Multicast

The following are circumstances under which a custodian may want to re-forward a multicast bundle:

- The custodian received a "Failed" custody signal for the bundle from a specific node, in which case it wants to retransmit the bundle to that node (and, assuming the network is relatively stable, it may want to conserve bandwidth by encapsulating [[refs.Encaps](#)] the multicast bundle in a singleton bundle for bandwidth-efficient transmission to the node that originally generated the "Failed" custody signal; this node can then de-encapsulate the multicast bundle and forward it further), or
- One or more of the custodian's custody timers for the bundle timed out, in which case it wants to re-forward the bundle on (at least) all downstream branches of the distribution path that are associated with the expiring timer(s).
- Notification of a new or delayed registration for a multicast EID is received.

The protocol extensions defined in this document do not address the third circumstance listed above: re-forwarding bundles to newly-registering or delayed-registration nodes. They do, however, address the first two circumstances listed above by ensuring that the custodian will be able to:

- receive "Failed" custody signals for arbitrary bundle copies from nodes downstream in the delivery path
- be aware of whether or not there exists a downstream copy of that bundle that has not been delivered or taken custody of when its custody timer times out.

Custody status notification is provided to each custodian of a multicast bundle in the same way that it is provided to each custodian of a singleton bundle: by having downstream nodes send either "Succeeded" or "Failed" custody signals to the custodian, as appropriate. In the multicast case, however, the custodian must keep track of the custody status of each copy of each bundle it forwards. When the custodian receives a "Succeeded" custody signal for each of the copies that it branched, the custodian is assured that every downstream copy has either been delivered or taken custody of. Until the custodian receives such a signal for any given copy that it forwarded, it must assume that there is at least one copy of the bundle on that copy's branch of the distribution tree that has neither been delivered nor taken custody of.

Although the custodian expects one "Succeeded" custody signal per bundle copy that it branched, there may be more copies of the bundle created downstream of it. These copies, however, must be kept track of by the non-custodial branching nodes that create them. When all copies created by a non-custodial branching node have either been delivered or taken custody of, the branching node sends a "Succeeded" custody signal to report this to the bundle's previous upstream custodian or non-custodial branching node, and so forth, until the status of every copy that the custodian branched is reported to the custodian. There is no need for the custodian itself to be made aware of every copy of the bundle that is created downstream of it. To achieve this "relayed" custody signal transmission, every non-custodial node that is a branching point for a multicast bundle, upon receipt of that bundle, takes note of its current custodian and then places its own EID into the bundle to list itself as custodian for that bundle before forwarding the bundle (even though it really is not the custodian of the bundle in the sense that it is not storing a copy of the bundle in permanent storage, nor does it consider itself responsible for retransmitting the bundle). In this way, each branching point node is assured that it will receive any custody signals that may be generated for the bundle copies that it branches.

Note that in order for custody signals to be received at custodians, there must be connectivity from the node that is generating the custody signal back to the custodian. This is true for the custody transfer of both singleton and multicast bundles. In the multicast case, there must also be connectivity from the node that is generating the custody signal back to its nearest upstream proxy custodian (if there is one), and from each proxy custodian to its nearest upstream proxy custodian and so forth to the real custodian. If this connectivity does not exist, the protocol will still work, but the custodian may issue one or more custodial retransmissions that are unnecessary and therefore wasteful of bandwidth.

Non-CMC and CMC nodes alike are permitted to originate and deliver custodial multicast bundles, to register in multicast EIDS, and to forward custodial multicast bundles to a single next-hop node. As is made clear in the Bundle Protocol, however, non-CMC nodes, are not permitted to take custody of multicast bundles. In order for a node to be able to become a custodian for a multicast bundle, the node must implement the procedures defined in this specification, which make it a CMC node.

In addition, nodes that implement only the base bundle protocol are not permitted to be branching points in the distribution path for a custodially-transferred multicast bundle. In order for a node to be able to be a branching point in the distribution path of a custodially-transferred multicast bundle, the node must be a CMC

node. The following sub-sections discuss the specific responsibilities that CMC-nodes must fulfill in order to act as custodians or branching points in support of the delivery of custodial multicast bundles.

5.1. Special Responsibilities of Custodial Nodes

When a CMC-node becomes custodian of a multicast bundle, that custodial node takes on responsibilities similar to those that are taken on by a custodian of a singleton bundle. The main difference, however, is that the custodian of a multicast bundle must maintain custody-related state information per copy of that bundle that it branches rather than just per bundle. Specifically, the custodian of a given multicast bundle:

- MUST maintain a list of the copies of that bundle that it has branched along with the EID/convergence layer to which each copy was delivered or forwarded.
- MUST keep track of which of these copies for which it has received a "Succeeded" custody signal.
- MUST retain the multicast bundle that it takes custody of--in persistent storage if possible-- until the bundle expires or (assuming that accommodating late- or delayed-registering destination nodes is not a concern) until it receives a "Succeeded" custody signal for each of the copies of the bundle that it branched (which indicates that all downstream copies of that bundle have either been delivered or taken custody of).
- MUST maintain at least one retransmission timer for the bundle; possibly one timer per copy it has branched.
- SHOULD retransmit the copy (or copies) of the bundle corresponding to the retransmission timer when the retransmission timer expires.
- SHOULD retransmit a copy of the bundle referred to by a received "Failed" custody signal, perhaps encapsulated (see [[refs.Encaps](#)]) in a unicast bundle sent to the "Failed" signal's source.
- MUST destroy a retransmission timer when "Succeeded" custody signals for all bundle copies associated with that timer have been received
- MUST delete a multicast bundle and all its associated custodial state information when the bundle expires

5.2. Special Responsibilities of (Non-Custodial) Branching Nodes

If a node is a branching point for a custodial multicast bundle but it does not take custody of that custodial multicast bundle, the node must pose as a proxy custodian by inserting its own EID into the custodian field of the bundle so that it will receive custody signals (if sent) for all copies of the bundle that it branches. In particular, in order to support custodial delivery of a multicast bundle, a non-custodial branching node:

- MUST maintain a list of the copies of that bundle that it has branched along with the EID/convergence layer to which each copy was delivered or forwarded.
- MUST keep track of the "Succeeded" custody signals received.
- MUST notify the appropriate upstream node (e.g. the node that had been listed as the bundle's custodian when the bundle was received, which is either the bundle's "real" custodian or the most recent proxy custodian that may in turn pass the signal upstream until it eventually reaches its real custodian) when a "Succeeded" custody signal is received for all of the copies of the bundle that it branched (which indicates that all downstream copies of that bundle have either been delivered or taken custody of).
- If custody transfer failure is reported for any of the downstream copies that the bundle branched, the non-custodial branching node MUST generate a replacement "Failed" custody signal and send it to the appropriate upstream node, inserting a Proxy EID extension block (see [Section 7](#)) into this bundle (if there is not one in there already) that identifies the endpoint of the source of the original "Failed" custody signal. (If custody transfer failure is reported for multiple downstream copies that the bundle branched, the non-custodial node may aggregate these by inserting multiple corresponding Proxy EID extension blocks, each of which identifies a different source node, into the replacement "Failed" custody signal that it generates and sends to the appropriate upstream node.)
- Upon receipt of a custodial multicast bundle, determine not only to which next-hop nodes the bundle should be forwarded in order to reach all of the nodes in its multicast endpoint, but also from which of these next-hop nodes (if any) the branching node has already received a "Succeeded" custody signal for this bundle. By default, but subject to network stability conditions and local policy, the bundle SHOULD be forwarded to only those next-hop nodes for which a "Succeeded" custody signal for the bundle has

not been received. (Forwarding the bundle to only those next-hop nodes for which a "Succeeded" custody signal for the bundle has not yet been received rather than to all next-hop nodes is an optimization for bandwidth conservation purposes; it is not a requirement.)

-If the node does not have sufficient resources to maintain the state information required of it in order for it to branch a particular multicast bundle, it MUST send a "Failed" custody signal to the appropriate upstream node. It MAY (subject to policy) reset the custody transfer requested bit and branch the bundle. If the branching node resets the custody transfer requested bit and branches the bundle, the reason code in the "Failed" custody signal MUST be the new reason code, "Bundle Forwarded Non-Custodially". If the branching node does not reset the custody transfer requested bit and branch the bundle, the reason code in the "Failed" custody signal MUST be "Depleted Storage". In addition, if the node's resource depletion is expected to last a while, the node MAY change the way it represents itself to the multicast routing protocol (subject to the specifics of that protocol), thereby actively seeking to not be a branching point in any multicast distribution paths that are set up by that routing protocol.

6. Identifying Bundle Copies

A main purpose of multicasting a bundle to several destination nodes rather than unicasting the bundle to each of those nodes individually is to conserve bandwidth. In keeping with this objective of bandwidth conservation, as mentioned in [Section 3](#), if there is a custody transfer failure of one downstream copy of a bundle whereas all other downstream copies of that bundle have had successful custody transfers or deliveries, it would conserve bandwidth to retransmit the bundle only along those branches of the downstream delivery path that are necessary for the bundle to reach the node at which the custody transfer failure occurred, rather than having that bundle traverse all downstream branches of the delivery tree--even those leading to nodes that have already successfully taken custody of or delivered the bundle. Performing such bandwidth optimization, however, requires branching nodes to be able to distinguish the various copies of a given bundle that they deliver or forward from each other, so that when a custody signal is received, the receiving node can determine not only which bundle it refers to, but which particular copy of the bundle that was branched it refers to. Furthermore, in order to perform bandwidth-efficient retransmissions that target only those next-hop nodes that still have downstream copies of the bundle for which "Succeeded" custody signals have not been received, the node receiving a custody signal must be able to determine to which next-hop node the particular copy of the bundle referred to by that signal had been forwarded. Because we cannot rely on the fact that the custody signal for a given bundle copy will always be returned via the same route along which the bundle was forwarded, the requirement that branching nodes be able to distinguish among bundle copies requires the branching node to somehow mark each bundle copy uniquely. This marking could be accomplished using a to-be-defined bundle extension block, but such a technique would require each branching node to add such a block, thereby adding to the size of the bundle. It would also require that procedures be defined for determining when these blocks could be deleted from the bundle, and in general it would complicate the protocol. Instead, we recommend that a certain portion of the EID that the branching node inserts into the bundle's custodian field be used as a copy identifier. For example, suppose a node is forwarding a multicast bundle to two different next-hop nodes. This forwarding node could be registered in two singleton EIDs, e.g., NodeA:1 and NodeA:2 and it can be uniquely identified by only the scheme name (e.g., "NodeA:") portion the EID [[refs.URI](#)]. The forwarding node places EID "NodeA:1" in the current custodian field of the bundle that it forwards to the first next-hop node and EID "NodeA:2" in the current custodian field of the bundle that it forwards to the second next-hop node. When it receives a custody signal back for this bundle, it can use the EID to which the signal was sent to determine

to which copy of the bundle the signal refers. This technique of distinguishing the multicast bundle copies from each other is conserving of both bandwidth and protocol complexity. Ideally, it would be defined as part of the multicast EID naming scheme and integrated with the routing protocols to the extent that only one registration per node would have to be propagated because only a certain portion of the EID would be used to route to each node.

Note, however, that as topology changes, new EIDS would need to be created to accommodate newly-discovered/encountered neighbors, and old EIDs would not be able to be recovered until after the latest expiration time of any bundle containing that EID. Hence, branching nodes that use this technique to distinguish among bundle copies would also have to keep track of the largest expiration time of the bundles associated with each next-hop EID.

Note also that depending on the type of convergence layer being used to forward the bundle copy, the bundle copy may no longer be identified uniquely when it is received at its next-hop node. If a copy is forwarded on a unicast convergence layer, then it is expected to be received at exactly one next-hop node, and the EID placed in the current custodian field, for example, can be used to identify it uniquely both when it is forwarded and when it is received. If a copy is forwarded over a multicast convergence layer, however, then the EID that is placed in the current custodian field is only guaranteed to identify the copy uniquely when the copy is forwarded; there is no guarantee that this copy will still be uniquely identified by that current custodian EID when it is received, because it may be received at multiple next-hop nodes and thereby become several different copies, all of which have the same EID in the current custodian field. Therefore, if a custodial multicast bundle is forwarded over a multicast convergence layer, the forwarding node must know the number of next-hop nodes to which it is destined (as discussed in [Section 4.3](#)). The EID placed in the current custodial field will be the same in each copy of the bundle received at these next-hop nodes. Therefore, for purposes of keeping track of "Succeeded" custody signals received, the forwarding node must expect the appropriate number of custody signals referring to bundles with this particular current custodian EID. For purposes of retransmission as a result of the custody timer associated with these copies timing out or as a result of the receipt of a "Failed" custody signal associated with one of these copies, a copy re-forwarded on the multicast convergence layer will reach all next-hop nodes that are reachable via that convergence layer and will not be targeted to reach any specific one of these next-hop nodes in particular.

7. Proxy EID Extension Block

This section defines the format of the Proxy EID Extension Block that a proxy custodian inserts into a "Failed" custody signal to inform the real custodian that eventually receives the "Failed" signal (or a replacement of the "Failed" signal) the endpoint ID of the node that was originally the source of the "Failed" signal. A bundle may contain multiple Proxy EID Extension Blocks, each of which identifies a node that was the source of a "Failed" custody signal for the identified bundle. For more discussion of when this block is inserted into a bundle containing a "Failed" custody signal and how it is processed, see [Section 8.2.10](#).

The Proxy EID Extension Block uses the Canonical Bundle Block Format as defined in the Bundle Protocol [[refs.DTNBP](#)]. That is, it is comprised of the following elements:

- Block-type code (one byte) - defined as in all bundle protocol blocks except the primary bundle block (as described in the Bundle Protocol). The block type code for the Proxy EID Extension Block is 0x06

- Block processing control flags (SDNV) - defined as in all bundle protocol blocks except the primary bundle block (as described in the Bundle Protocol). SDNV encoding is described in the bundle protocol. In order for custodial multicast transfer to be supported correctly in a network that consists of a mixture of CMC-nodes and non-CMC-nodes, the following block processing control flags MUST NOT be set:

 - Discard block if it can't be processed.

 - Discard bundle if block can't be processed.

The following block processing control flag MUST be set:

- Block contains an EID-reference field.

- EID references - composite field defined in [[refs.DTNBP](#)] containing references to one or more EIDs. Presence of EIDs is indicated by the "block contains an EID-reference field" flag in the block processing control flags, which must be set to 1. Each of the EIDs referenced in this field identify a node that originally generated a "Failed" custody signal for the indicated bundle. The sequence of the EIDs reference in this field correlates with the sequence of custodial signal information records that appears later in this block, insofar as the node identified by a given EID reference generated a "Failed" custody

signal for the indicated bundle with a custody signal reason code and generation time as indicated by the corresponding custody signal information record.

-Block data length (SDNV) - defined as in all bundle protocol blocks except the primary bundle block. SDNV encoding is described in the bundle protocol.

-Block-type-specific data field as follows:

-Ordered sequence of custody signal information records, where a custody signal information record is defined to consist of a (1-byte) status field followed by a (DTN time) Time of signal field. These fields are as defined in the Custody Signals section of the Bundle Protocol [[refs.DTNBP](#)].

The Structure of a Proxy EID Extension Block is as follows:

Proxy EID Extension Block Format:

```

+-----+-----+-----+-----+
| Type  |Flags (SDNV) |EID ref count and list (comp) |Len (SDNV) |
+-----+-----+-----+-----+
|Status |   Time of signal (a DTN time)           |
+-----+-----+-----+-----+

```

Figure 1

8. Bundle Protocol Extensions to Support Custody Transfer and Retransmission for Multicast Bundles

The Bundle Protocol defines custody transfer and retransmission only for singleton bundles. Nodes that implement only the bundle protocol are not capable of generally supporting the custody transfer and retransmission of multicast bundles. They are only capable of supporting the custodial delivery of multicast bundles if they serve neither as custodians nor as branching points in the delivery path. The sections below define the custody transfer and retransmission capabilities that a node must have in order to be CMC and to therefore be capable of supporting custodial transfer and retransmission for multicast bundles regardless of where that node is located in the delivery path.

All nodes that are going to either take custody of a multicast bundle or serve as a branching point of a multicast bundle must implement not only the Bundle Protocol but also the following extensions to the Bundle Protocol in order to be able to support custodial multicast. Nodes that are neither custodians nor branching points may participate in the delivery of custodial multicast bundles without implementing these extensions. If custodial multicast is to be supported in a network that consists of both CMC nodes and non-CMC nodes, the multicast routing protocol is required to build the distribution path with only CMC nodes as branching points.

8.1. Definitions

Custody - Custody of a bundle whose destination is a multicast endpoint is released when notification is received for each copy of the bundle that was delivered or forwarded by the custodian that some other node has accepted custody of the copy, the copy has been delivered at some destination node, or the bundle has been explicitly deleted for some reason, such as lifetime expiration.

8.2. Bundle Processing Steps for Custodial Multicast Bundles

The specific steps for processing a custodial multicast bundle at each node are defined as follows.

8.2.1. Bundle Forwarding

This section augments the Bundle Forwarding section of the Bundle Protocol [[refs.DTNBP](#)]. It defines those steps for forwarding bundles that are specific to custodial multicast bundles.

If the bundle will be forwarded to more than one next-hop node and the forwarding node has not accepted custody of the bundle, then the

node MUST determine if it has sufficient resources to serve as a branching point for the bundle. If it cannot serve as a branching point, the node MUST send a "Failed" custody signal to the upstream node that is listed as the bundle's custodian (which is either the "real custodian" or the most recent proxy custodian that will in turn pass the signal upstream until it eventually reaches the real custodian). It may also (subject to policy) reset the custody transfer requested bit and deliver and/or forward the bundle. If the branching node is delivering or forwarding the bundle (non-custodially), the reason code in the "Failed" custody signal must be the new reason code, "Bundle Forwarded Non-Custodially". If the branching node is not delivering or forwarding the bundle, the reason code in the "Failed" custody signal must be "Depleted Storage".

If the node does have sufficient resources to enable it to serve as a non-custodial branching point, then:

- The node must note the endpoint ID of the bundle's current custodian and store it with the custodial state information associated with the bundle.

- The node must change the current custodian endpoint ID in the bundle's primary block to the endpoint ID of one of the singleton endpoints in which the node is registered. In order to be able to identify each copy uniquely and associate each copy of the bundle with the next-hop node to which it will be delivered or forwarded, the node should use a separate custodian endpoint ID for each copy of the bundle that it delivers or forwards. Alternatively, it may mark the copy in some other as-yet-undefined way. (Note that if the node is forwarding a copy on a multicast convergence layer and this copy is expected to reach multiple next-hop endpoints, each of the copies received at each of these next-hop nodes will have the same custodian EID.)

- The node must also keep track of whether the bundle is being delivered and each of the copies that is being forwarded (both explicit copies and copies that may be engendered by the use of a multicast convergence layer), along with whether or not a "Succeeded" custody signal has been received corresponding to each of these copies. (If a copy is being forwarded on a unicast convergence layer, a single custody signal related to that copy is expected. If a copy is forwarded on a multicast convergence layer, n custody signals associated with that copy are expected, where n is the number of next-hop nodes reachable via that convergence layer in the delivery tree.)

In all, the custodial state information that a non-custodial branching point node is required to maintain for each multicast

bundle for which it is a branching point consists of:

- an indication of whether or not the bundle was delivered at this node,
- a list of the next-hop EIDs to which the bundle was forwarded (including the number of next-hop nodes associated with each EID for those next-hop EIDs that are reachable via multicast convergence layers)
- a list of the next-hop nodes for which a corresponding "Succeeded" custody signal has been received, and
- the endpoint ID that was in the current custodian field of the bundle when it was received.
- the largest expiration time of all bundles associated with each next-hop EID (to ensure uniqueness when recovering custodian EID values to be used to distinguish among bundle copies).

This custodial state information must be maintained for each branched bundle until the bundle expires or until the node itself sends a "Succeeded" custody signal for the bundle to the endpoint ID that was in the current custodian field of the bundle when it was received, at which time the information may be deleted.

Note that even though the branching node is not becoming a true custodian for a given multicast bundle, it is inserting its own endpoint ID into the custodian field before forwarding the bundle, so that it may receive either a "Succeeded" or a "Failed" custody signal for each copy of the bundle that it forwards. If a node that is not the custodian of the multicast bundle receives a "Succeeded" custody signal, this MUST be noted in the custodial state information for the bundle, relative to the next-hop branch from which the custody signal was received.

The processing steps to be taken next may be determined by policy, which would be influenced by network stability characteristics and would involve the tradeoff between bandwidth conservation versus robustness of delivery. In the absence of a policy to the contrary, custodial multicast is assumed to favor bandwidth conservation over robustness of delivery. So by default, when a bundle is received, the node must consult its custodial state information to determine for which of the next-hop endpoints to which it should be forwarded, if any, the node has already received a "Succeeded" custody signal for the bundle. If the bundle has already been delivered at this node and a "Succeeded" custody signal received back confirming this delivery, the bundle should not be delivered again. Similarly, the

bundle should not be re-forwarded to any next-hop endpoints for which all expected "succeed" custody signals have already been received for this bundle, because the prior receipt of either of these signals indicates that at some previous time there were no longer any copies of the bundle downstream of that next-hop endpoint that needed to be either delivered or have their custody transferred. If a policy to the contrary is in place that favors robustness of delivery over bandwidth conservation, then when a bundle is received, it should be forwarded out to all next-hop endpoints, even if it has previously been forwarded to some of those endpoints and "Succeeded" custody signals corresponding to one or more of those endpoints have been received back.

8.2.2. Forwarding Contraindicated

This section augments the Forwarding Contraindicated section of the Bundle Protocol [[refs.DTNBP](#)]. It defines the procedures for handling a forwarding contraindication for a bundle whose destination is a multicast endpoint.

As described in [[refs.wenruiZhao](#)], there is a delay, which in a DTN could potentially be substantial, between when a node registers with a multicast EID and when that registration propagates fully throughout the network. (Although [[refs.wenruiZhao](#)] discusses this potentially substantial delay in terms of multicast EIDs, it exists relative to registration requests for singleton EIDs as well.) Analogously, there may also be a substantial delay between when a node de-registers with a (multicast or singleton) EID and when that de-registration propagates fully throughout the network. As a result, it might not be uncommon for a situation to arise in which a de-registration request has reached some nodes but not others such that a bundle could be forwarded by a custodian (which has not yet received notice of the de-registration request) and later received at some downstream node (which has received the de-registration request) that does not have any next-hop nodes to which the bundle should be forwarded (because the node that recently de-registered was the only node downstream of this node that had been in the multicast endpoint). In this situation, forwarding of the bundle is determined to be contraindicated because of the "No known route to destination from here" reason listed in the Status Report Reason Codes of the Bundle Protocol specification [[refs.DTNBP](#)]. According to the Bundle Protocol specification, the bundle protocol agent must determine whether or not to declare a failure in forwarding in this case.

If the bundle protocol agent at which this situation occurs has recorded the fact that it recently received a deregistration request notification for the EID in question, it can distinguish this forwarding contraindication from a forwarding failure. Instead of

declaring a forwarding failure, the bundle protocol agent should send a "Succeeded" custody signal for this multicast bundle. A "Succeeded" custody signal indicates that there are no remaining copies of the bundle downstream of this node that need to be either delivered or taken custody of. In order to be able to distinguish this type of forwarding contraindication from a forwarding error, bundle protocol agents should keep track of de-registration request notifications received in addition to registration request notifications received.

If the bundle protocol agent at which this situation occurs is not recording the de-registration request notifications it receives, it will not be able to distinguish this situation in which forwarding is contraindicated because a downstream node has recently de-registered from the bundle's destination endpoint from a real forwarding failure. Hence, it will declare a forwarding failure and will notify the custodian of the bundle of a custody transfer failure, possibly triggering a custodial retransmission of the bundle.

8.2.3. Forwarding Failed

This section augments the Forwarding Failed section of the Bundle Protocol [[refs.DTNBP](#)]. It defines the procedures for handling forwarding failure for a bundle whose destination is a multicast endpoint.

Procedures for handling failure of custody transfer for a custodial multicast bundle are the same as those for a singleton bundle: the bundle protocol agent MUST generate a "Failed" custody signal for the bundle, destined for the bundle's current custodian; the custody signal MUST contain a reason code corresponding to the reason for which forwarding was determined to be contraindicated.

8.2.4. Bundle Reception

This section augments the Bundle Reception section of the Bundle Protocol [[refs.DTNBP](#)]. It defines the procedures for handling redundancy of custody transfer for a bundle whose destination is a multicast endpoint.

step 1: When a custodial multicast bundle is received, if the bundle has the same source EID, creation timestamp, fragment offset and length (if any), and custodian EID as a bundle that the receiving node is currently custodian of, then by default the bundle's "Dispatch pending" retention constraint SHOULD be removed and the node SHOULD NOT send a "Succeeded" custody signal for this bundle. If the custodian EID of the received multicast bundle is different from that of the otherwise identical bundle of which the receiving

node is currently custodian, however, then the bundle's "Dispatch pending" retention constraint MUST be removed and the node MUST send a "Succeeded" custody signal for this bundle.

8.2.5. Local Bundle Delivery

This section augments the Local Bundle Delivery section of the Bundle Protocol [[refs.DTNBP](#)]. It defines the procedures for reporting custodial delivery for a bundle whose destination is a multicast endpoint.

As soon as the bundle has been delivered, the bundle protocol agent must report custodial delivery for multicast bundles as it does for singleton bundles, namely, by generating a "Succeeded" custody signal for the bundle, destined for the bundle's current custodian.

8.2.6. Custody Transfer

This section augments the Custody Transfer section of the Bundle Protocol [[refs.DTNBP](#)]. It defines the conditions under which a node may accept custody of a bundle whose destination is a multicast endpoint.

As with singleton bundles, the decision as to whether or not to accept custody of a bundle is an implementation matter; however, if the bundle protocol agent has committed to accepting custody of the bundle as described in step 1 of [section 4.2](#) ("Bundle Transmission") of the Bundle Protocol [[refs.DTNBP](#)], then custody must be accepted. If the bundle protocol agent elects to accept custody of the bundle, then it must follow the custody acceptance procedure defined in [Section 8.2.7](#)

8.2.7. Custody Acceptance

This section augments Custody Acceptance section of the Bundle Protocol [[refs.DTNBP](#)]. It defines the procedures for acceptance of custody of a bundle whose destination is a multicast endpoint.

The retention constraint "Custody Accepted" must be added to the bundle.

If the "request custody acceptance reporting" flag in the bundle's class of service field is set to 1, this flag MAY be ignored.

The bundle protocol agent must generate a "Succeeded" custody signal for the bundle, destined for the bundle's current custodian.

The bundle protocol agent must assert the new current custodian for

the bundle. It does so by changing the current custodian endpoint ID in the bundle's primary block to the endpoint ID of one of the singleton endpoints in which the node is registered.

The bundle protocol agent must establish and maintain state information to keep track of each of the copies of this bundle that it is either delivering or forwarding, the EIDs to which they are being delivered or forwarded, the number of next-hop nodes expected to be reached at each EID, and whether or not a "Succeeded" custody signal has been received corresponding to each of these copies.

The bundle protocol agent MAY set one custody transfer countdown timer per bundle or it MAY set one custody transfer countdown timer per one or more copies of this bundle that it is branching; upon expiration of one or more of these timers prior to expiration of the bundle itself and prior to custody transfer success for the associated copy (or copies) of the bundle, the custody transfer failure procedure must be followed for that copy (or those copies) of the bundle. The manner in which the countdown interval for such timers is determined is an implementation matter.

The bundle SHOULD be retained in persistent storage if possible.

8.2.8. Custody Release

This section augments the Custody Release section of the Bundle Protocol [[refs.DTNBP](#)]. It defines the procedures for release of custody of a bundle whose destination is a multicast endpoint.

As with singleton bundles, when custody of a multicast bundle is released, the "Custody accepted" retention constraint must be removed from the bundle and all custody transfer timers and other state information that have been established for this bundle must be destroyed.

8.2.9. Custody Transfer Success

This section augments the Custody Transfer Success section of the Bundle Protocol [[refs.DTNBP](#)]. It defines the procedures for determining custody transfer success for a bundle whose destination is a multicast endpoint.

Upon receipt of a "Succeeded" custody signal the receiving node MUST note in its custodial state information for this bundle the receipt of this signal in conjunction with the associated copy of the bundle that the node had branched, because the receipt of this signal means that there are no downstream copies of that copy of the bundle that still need to be either delivered or have their custody transferred.

When a node has noted in its custodial state information for a given bundle that such successful notification has been received for every copy of that bundle that the node branched, meaning that there are no downstream copies of any copy of the bundle that remain to be delivered or taken custody of, then the node MUST check its custodial state information for the referenced bundle to determine whether it is in fact the "real" custodian for that bundle or whether it is just a proxy custodian.

If the node is the real custodian of the bundle then custody of the bundle must be released as described in [Section 8.2.8](#).

If the node is a proxy custodian of the bundle, then the node MUST generate a "Succeeded" custody signal for the bundle, destined for the node that it had noted in its custodial state information as being the bundle's current custodian when the bundle was received. It MAY destroy all custodial state information for the bundle. (The node may want to retain this custodial state information until the bundle expires so that if the node receives the bundle again it will know exactly to which next-hop nodes it was forwarded, assuming this information is useful for efficiently transmitting bundles to late-joining nodes.)

8.2.10. Custody Transfer Failure

This section augments the Custody Transfer Failure section of the Bundle Protocol [[refs.DTNBP](#)]. It defines the procedures for determining custody transfer failure for a copy of a bundle whose destination is a multicast endpoint.

Custody transfer for a multicast bundle copy is determined to have failed at a custodial node when either (a) that node's custody transfer timer that is associated with that copy of the bundle (if any) expires or (b) a "Failed" custody signal for the bundle copy is received at that node. Only custodial nodes have custody transfer timers for bundles; non-custodial branching nodes do not. Both non-custodial branching nodes and custodial nodes, however, may receive "Failed" custody signals.

Upon receipt of a "Failed" custody signal, the receiving node must check its custodial state information for the referenced bundle copy to determine whether or not it is in fact the "real" custodian for that bundle copy or whether it is just a proxy custodian for that bundle copy.

If the receiving node is a proxy custodian of the bundle, then it must start a signal aggregation timer for this bundle if one has not already been started.

If it has not already done so, the receiving node must generate a new bundle. The receiving node's own EID must be inserted as the new bundle's source, and the EID of the node that the receiving node had noted in its custodial state information as being the referenced bundle's current custodian when the bundle was received must be inserted as the new bundle's destination. The "Failed" custody signal administration record must be inserted as this new bundle's payload.

If the bundle that contained the "Failed" custody signal that was received also included a Proxy EID extension block, this extension block must be included in the new bundle that is generated. If the bundle that contained the "Failed" custody signal that was received did not include a Proxy EID extension block, then the node must insert a Proxy EID extension block into the new bundle that is generated. In either case, the node must insert a custodial signal information record into the Proxy EID extension block of the new bundle that is generated.

The procedures for inserting a custodial information record into a proxy EID extension block are as follows:

- The value inserted into the Status field of the custodial signal information record must be the value of the status field of the "Failed" custody signal that was just received.

- The value inserted into the Time of Signal field of the custodial signal information record must be the value of the Time of signal field of the "Failed" custody signal that was just received.

- Strings for the Scheme and SSP of the source EID of the bundle containing the "Failed" custody signal that was just received must be inserted into the dictionary of the bundle containing the proxy EID extension block (if not already there).

- Offsets to the above EID reference pair must be inserted into the EID references field of the Proxy EID extension block following all other such EID references that are in this block, and the EID references count must be incremented by one.

If, on the other hand, the receiving node had already generated a new bundle, it must insert a custodial signal information record into the Proxy EID extension block of this new bundle using the procedures defined above.

When the signal aggregation timer for this bundle expires or when custody signals corresponding to all copies of the bundle that were

branched at this node are received (whichever comes first), the node must forward the new bundle. Note that this new bundle contains a "Failed" custody signal that refers to a specific bundle and it is destined for the upstream node that had been noted as being the current custodian of the referenced bundle when the bundle was received. This new bundle that is forwarded will contain a Proxy EID extension block with one or more custody signal information records, each of which corresponds to the origination of a "Failed" custody signal for a different downstream copy of this bundle by the downstream nodes indicated in the EID references field.

If the receiving node is the real custodian of the bundle copy then, as noted above, custody transfer of the referenced bundle copy is determined to have failed. If the "Failed" custody signal contains a Proxy EID extension block, then the EID of the node(s) that originally generated the "Failed" custody signal(s) is/are identified in the EID references field of the Proxy EID extension block; otherwise, the EID of the node that originated the custody signal is identified in the Source EID field of the bundle containing the "Failed" custody signal.

Upon determination of a custody transfer failure for a particular multicast bundle copy, the action taken by the custodian depends on the nature of the failure:

- If custody transfer failure is inferred from expiration of a custody transfer timer or is asserted by a "Failed" custody signal with the "Depleted storage", "No timely contact with next node on route", or "Bundle Forwarded Non-Custodially" reason code, the bundle protocol agent SHOULD retransmit the bundle. It may re-forward the bundle to all next hop nodes appropriate for reaching nodes in the multicast endpoint, it may re-forward the bundle only to the next hop node(s) to which the copy of the bundle referenced in the "Failed" custody signal or associated with the expired timer had been forwarded, or it may encapsulate the multicast bundle in a singleton bundle for bandwidth-efficient transmission to the node that originally generated the "Failed" custody signal if a "Failed" custody signal had been received (See [\[refs.Encaps\]](#)).

- If custody transfer failure is asserted by a "Failed" custody signal with the "Redundant Reception", "Destination EID unintelligible", "Block unintelligible", or "No known route to destination from here" reason code, the node SHOULD NOT retransmit the bundle.

8.2.11. Bundle Deletion

This section augments the Bundle Deletion section of the Bundle Protocol [[refs.DTNBP](#)]. It defines the steps in deleting a custodial bundle whose destination is a multicast endpoint.

If the retention constraint "Custody accepted" currently prevents this bundle from being discarded, then:

- Custody of the node is released as described in [Section 8.2.8](#)

- A bundle deletion status report citing the reason for deletion must be generated, destined for the bundle's report-to-endpoint ID.

All custodial state information being stored for this bundle MUST be destroyed.

8.3. Reception of Custody Signals

This section augments [section 5.3](#) of the Bundle Protocol [[refs.DTNBP](#)]. It defines the procedures that must be followed when custody signals for multicast bundles are received.

For each received custody signal for a multicast bundle that has the Custody Transfer Succeeded flag set to 1, the administrative element of the application agent must direct the bundle protocol agent to follow the custody transfer success procedure in [Section 8.2.9](#).

For each received custody signal for a multicast bundle that has the Custody Transfer Succeeded flag set to 0, the administrative element of the application agent must direct the bundle protocol agent to follow the custody transfer failure procedure in [Section 8.2.10](#).

9. Custodial Multicast and the Retransmission Block

Use of the DTN Retransmission Block [[refs.Retrans](#)] to identify bundles that are custodial retransmissions so as to distinguish them from replays is optional. If a node receives a bundle that

- has a multicast destination endpoint,
- has its "custody transfer requested" Bundle Processing Flag set, and
- includes a retransmission block

then if that node will not become a custodian of the bundle but it will serve as a non-custodial branching point, then in addition to changing the current custodian endpoint ID in the bundle's primary block to the endpoint ID of one of the singleton endpoints in which the node is registered (as described in Bundle Forwarding [Section 8.2.1](#)), the node must also change the EID reference field in the Retransmission Block to refer to this same singleton EID in which the node is registered. Hence, both the custodian field and the Retransmission Block will refer to a singleton EID that identifies the Proxy Custodian rather than one that identifies the "real" custodian.

10. Performance Impact

The mechanisms defined in this document attempt to conserve network bandwidth and minimize the complexity of enhancements to the Bundle Protocol. They increase the amount of state that will need to be maintained at those bundle nodes that support these mechanisms.

The amount of state that a given custodial node or non-custodial branching node may have to maintain is potentially large. A node may not have sufficient storage space to store this state information for each bundle until the bundle expires. if a node runs out of room and cannot keep track of for which bundle copies it has received "Succeeded" custody signals and for which it has not, then if the node receives a retransmitted bundle, it will have to re-forward the bundle along all downstream branches of the delivery tree. This is not incorrect; just inefficient. Nodes should be provided with a default algorithm for defining how long state information must be maintained and, if they run out of room for state information, for determining which state information gets deleted and which gets saved.

11. Security Considerations

Although there are two documents that pertain to providing security within DTN [[refs.DTNBPsec](#)], [[refs.DTNsecOver](#)], both of these documents address the security of the Bundle Protocol when used to transmit bundles from a single source to a single destination. They do not address the security of multicast delivery within the bundle protocol. Therefore, how to extend the Bundle Security Protocol to provide end-to-end protection for bundles that are sent from a single source to multiple destinations is yet to be determined. The use of the Bundle Authentication Block (BAB) to provide hop-by-hop security protections for bundles is expected to remain largely unchanged when applied protecting multicast delivery. The Confidentiality Block (CB) and the Payload Security Block (PSB), however, because they provide end-to-end security, would be expected to require adaptation to provide protection between a single source at one end of the transmission and multiple destinations at the other end.

This document does not consider the security aspects of enabling or preventing a node from registering or de-registering with a particular multicast endpoint ID, and thereby joining and leaving a particular multicast group, although we acknowledge that security considerations regarding joining and leaving groups are present and significant.

12. IANA Considerations

None at this time. If the bundle protocol becomes a standards track protocol, then we may want to consider having IANA establish a register of block types.

13. References

13.1. Normative References

[refs.[RFC2119](#)]

Bradner, S. and J. Reynolds, "Key words for use in RFCs to Indicate Requirement Levels", [RFC 2119](#), October 1997.

[refs.DTNBP]

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

[refs.Encaps]

Symington, S., Durst, R., and K. Scott, "Delay-Tolerant Networking Bundle-in-Bundle Encapsulation", [draft-irtf-dtnrg-bundle-encapsulation-06.txt](#), work-in-progress, August 2009.

[refs.DTNBPsec]

Symington, S., Farrell, S., Weiss, H., and P. Lovell, "Bundle Security Protocol Specification", [draft-irtf-dtnrg-bundle-security-08.txt](#), work-in-progress, March 2009.

[refs.Retrans]

Symington, S., "Delay-Tolerant Networking Retransmission Block", [draft-irtf-dtnrg-bundle-retrans-block-05.txt](#), work-in-progress, April 2009.

13.2. Informative References

[refs.DTNarch]

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

[refs.DTNsecOver]

Farrell, S., Symington, S., Weiss, H., and P. Lovell, "Delay-Tolerant Network Security Overview", [draft-irtf-dtnrg-sec-overview-06.txt](#), work-in-progress, March 2009.

[refs.wenruiZhao]

Zhao, W., Ammar, M., and E. Zegura, "Multicasting in Delay Tolerant Networks: Semantic Models and Routing Algorithms", August 2005.

[refs.IEEEcustMcast]

Symington, S., Durst, R., and K. Scott, "Custodial Multicast in Delay Tolerant Networks", January 2007.

[refs.URI]

Burners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, [RFC 3986](#), January 2005.

Authors' Addresses

Susan Flynn Symington
The MITRE Corporation
7515 Colshire Drive
McLean, VA 22102
US

Phone: +1 (703) 983-7209
Email: susan@mitre.org
URI: <http://mitre.org/>

Robert C. Durst
The MITRE Corporation
7515 Colshire Drive
McLean, VA 22102
US

Phone: +1 (703) 983-7535
Email: durst@mitre.org
URI: <http://mitre.org/>

Keith L. Scott
The MITRE Corporation
7515 Colshire Drive
McLean, VA 22102
US

Phone: +1 (703) 983-6547
Email: kscott@mitre.org
URI: <http://mitre.org/>

