

DTN Research Group
INTERNET-DRAFT
<[draft-irtf-dtnrg-ipn-bundle-xfer-01.txt](#)>
October 2003
Expires April 2004

Robert C. Durst
The MITRE Corporation

**Delay-Tolerant Networking:
An Example Interplanetary Internet Bundle Transfer**

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of [Section 10 of RFC2026](#).

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>.

Abstract

This document presents an example data transfer in a delay tolerant network. The particular delay tolerant network is the Interplanetary Internet, and the purpose of this document is to illustrate the key concepts in the Delay Tolerant Network architecture [[DTNRG03](#)]. The document describes the network and follows the progress of a bundle as it is transferred through that network.

Table of Contents

Status of this Memo.....	1
Abstract.....	1
Table of Contents.....	2
Copyright Notice.....	2
1 Introduction.....	3
2 Rules for forming tuples in the example network.....	3
2.1 Region identification.....	3
2.2 Intra-region identification.....	3
3 Example Network Topology at the Region Level.....	5
4 DTN Gateway routing.....	6
5 Systems participating in example bundle data transfer.....	7
6 End-to-end Transfer.....	9
6.1 Step 0: Application instance registration.....	9
6.2 Step 1: Bundle creation and first-hop transmission....	9
6.3 Step 2: Bundle processing at first-hop destination... 10	
6.4 Step 3: Bundle processing at gateway to destination IPN Region.....	11
6.5 Step 4: Bundle processing at destination.....	11
7 Error Conditions at the Bundle Layer.....	11
8 Normative References.....	14
9 Security Considerations.....	14
10 Author's Address.....	15

Copyright Notice

Copyright (C) The Internet Society (2003). All Rights Reserved.

Acknowledgments

John Wroclawski, David Mills, Greg Miller, James P. G. Sterbenz, Joe Touch, Steven Low, Lloyd Wood, Robert Braden, Deborah Estrin, Stephen Farrell and Craig Partridge all contributed useful thoughts and criticisms to the DTN Architecture. We are grateful for their time and participation.

This work was performed in part under DOD Contract DAA-B07-00-CC201, DARPA AO H912; JPL Task Plan No. 80-5045, DARPA AO H870; and NASA Contract NAS7-1407.

1 Introduction

This document is a companion to the Delay Tolerant Networking architecture definition [[DTNRG03](#)]. The article "Delay-Tolerant Networking: An Approach to Interplanetary Internet" [[IEEE](#)] considers the application of the DTN approach with one based on adaptations of existing Internet infrastructure.

We provide the following example of an end-to-end bundle transfer across a Delay-Tolerant Network. This particular example is based on the Interplanetary Internet: A host on Earth sends a bundle to a destination on Mars. Figure 1 illustrates the network that is the subject of this example - the Interplanetary Internet in this example has five distinct regions interconnected by four DTN Gateways. This example highlights the actions taken by the bundle layer and its interactions with applications and underlying transport protocols.

2 Rules for forming tuples in the example network

Per the DTN architecture [[DTNRG03](#)], application instance ID tuples consist of two parts: a region ID, and a Universal Resource Identifier (URI) [[RFC3305](#)]. Each DTN region has a URI space shared by all DTN nodes within that region. Each DTN region has a unique identifier that is known (or discernable) by all regions in the DTN. This particular delay-tolerant network is the Interplanetary Internet. In this section we will establish the naming rules that permit interoperation within this network. Note that this is only an example, and that not all DTNs must use this particular region identifier space.

2.1 Region identification

All regions in *this* DTN (the example Interplanetary Internet) must share a region identifier space. The DTN region *name* space is hierarchical, dot-separated text, structured such that left to right is leaf-to-root in the tree. The *identifier* space may be exactly this name space, or a DTN may define a translation from the name to a particular identifier, in the same way that names in the DNS may be translated to Internet addresses. For this example, we will use the *names* as the *identifiers*.

The DTN region name space is shown in Figure 2.

2.2 Intra-region identification

For simplicity in this example, we will assume that all regions use a well-known TCP port in combination with DNS host names as the portion of their entity identifier that locates the application providing the bundle service. The bundle layer application resides above this well-known port (which we arbitrarily choose to be TCP port 6769, a

currently unassigned port in the registered port space). The

Durst

Expires September 2003

[Page 3]

interplanetary backbone region, labeled "ipn" in Figure 2, will certainly NOT use TCP as its underlying transport protocol. For the sake of simplicity in this example, let us assume that the protocol used within the interplanetary backbone region uses the same entity identifier space (IP addresses and port numbers) that the remainder of the network uses.

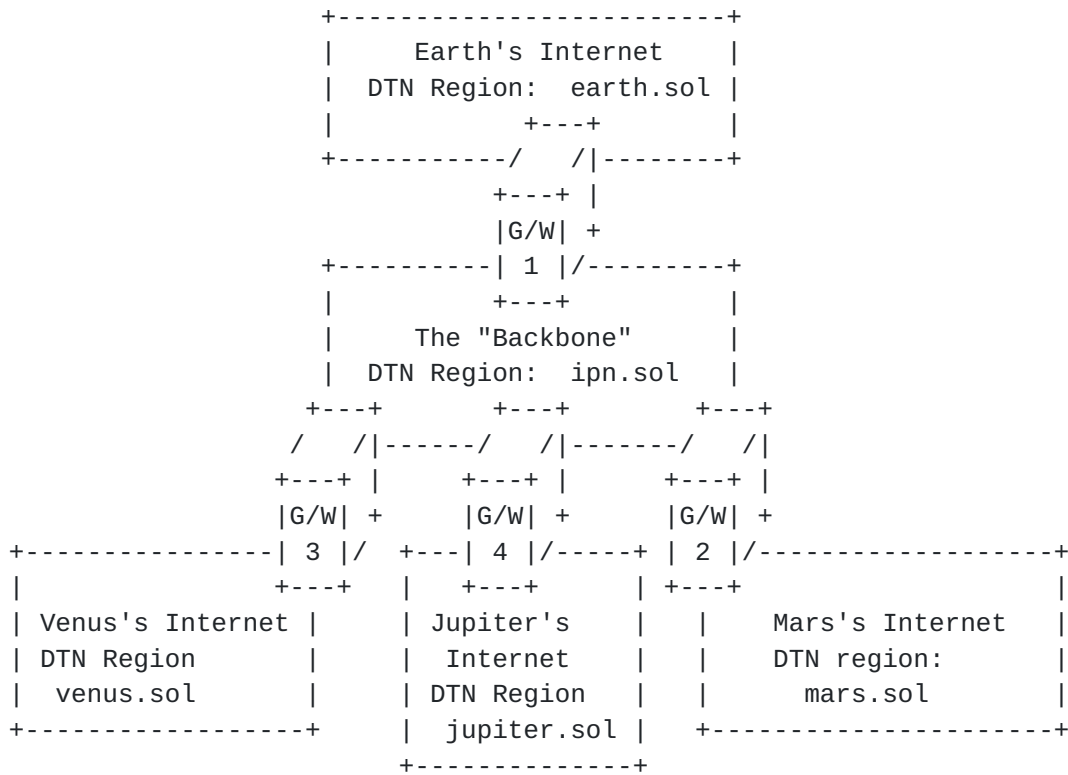


Figure 1. An Interplanetary Internet of Five IPN Regions

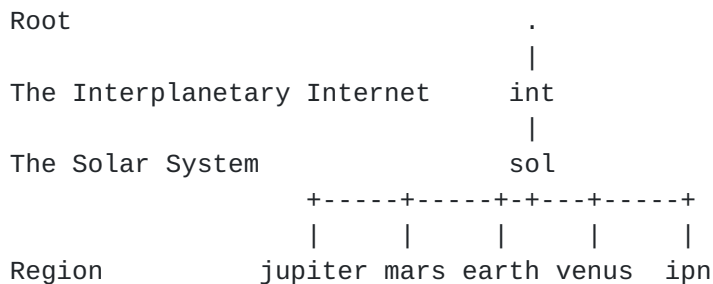


Figure 2. An Example Interplanetary Internet Region Name Space

The mechanism used to demultiplex bundles received by a bundle node is entity-specific, but must be represented in the region-specific

URI. For the sake of simplicity, we will assume that this is a

Durst

Expires September 2003

[Page 4]

process ID that has been incorporated into the URI. [Note: this is a simple, but quite limiting decision, as it has implications on process instantiation and process portability/migration from one entity to the next. We choose it only for simplicity.]

3 Example Network Topology at the Region Level

It is important to have some understanding of the routing that is required at the DTN Gateways, so it is important to understand the topology of the network at the region level. Unlike typical terrestrial communications, the Interplanetary Internet's (IPN's) *long-haul* communication links are directional, mobile, and highly scheduled. This is important, because directionality combined with mobility means that a transmitter and receiver must track each other in order to establish and maintain a communication link. In the IPN, much of the mobility is due to orbital mechanics and is therefore relatively predictable. However, this means that nodes that we would normally consider to be fixed, such as antennas on the surface of the Earth, are actually highly mobile as a result of the Earth's rotation and its revolution around the Sun. (In this example, we confine ourselves to our local solar system, and do not consider the motion of our sun relative to celestial bodies outside our solar system.)

We can describe the predictable aspects of node mobility with an ephemeris, a table of the positions of celestial bodies at specified intervals of time. Both a directional sender and receiver must each know the other's position in order to establish a link between the pair.

In addition, these communication resources are highly scheduled. It is not sufficient for a receiver to point at a prospective target and just wait -- for example, a terrestrial node will typically have to point at several targets sequentially, and an interplanetary node will typically not have enough power to just wait for incoming messages. Rather, a schedule of communication *opportunities* must be calculated and then refined with planned communication *contacts*. A communication opportunity establishes that the endpoints could establish a link if they were pointing at each other at the proper times. We refer to a planned communication contact as an agreement (although not irrevocable) between the two parties to establish contact and communicate for a defined period of time.

The protocols for establishing the schedule of communication contacts between all pairs of possible communicants will evolve -- from something primarily done manually to something more automated as the real Interplanetary Internet grows.

The scheduled nature of connectivity in the Interplanetary Internet, particularly across the deep-space links, means that at the time of a

bundle's arrival at a DTN Gateway, some or all of the possible
outbound routes may be "down." The gateway must store the bundle

Durst

Expires September 2003

[Page 5]

until the appropriate link is available and then transmit the bundle over that link. One of the fundamental differences between the Interplanetary Internet and the terrestrial Internet is this inherent use of store-and-forward mechanisms in routing bundles. With that in mind, let us consider the routing decisions made at some of the DTN Gateways in Figure 3.

4 DTN Gateway routing

Bundle routing at a DTN gateway will typically have to deal with a mix of continuously available links and intermittently available links. Routing across a continuously available link is a relatively straightforward activity. Routing in the presence of intermittently available links can be significantly more complex, as the gateway will need to decide not only the next hop destination but also the time at which to send the bundle. Conditions elsewhere in the network may make it more desirable to send a bundle to a next-hop destination that is not yet accessible, even when an alternative route is currently available.

The intermittent links in this example provide connectivity among the DTN Gateways that are part of the ipn.sol.int region. The contacts among gateways in this region are scheduled. As is currently the case, Gateway 1 (the Earth gateway) has scheduled contacts with each of the other DTN gateways in the ipn.sol.int region (the "backbone"), but there is no direct contact among any of the DTN gateways on Venus, Jupiter, or Mars. Recall that this communication uses directional antennas, so it is generally not possible to communicate with two different entities at once unless they are in the same field of view. Power availability on the remote planets is an issue, so transmitters and receivers are turned on only during a contact. Further, the speed of light delays are nontrivial, skewing transmit and receive times between remote gateways. Table 1 presents a schedule of contacts that is used in the example. All times are referenced to Gateway 1, and that nodes remote to Gateway 1 (that is, Gateways 2, 3, and 4) are not scheduled to turn on their transmitters until they receive a transmission from Gateway 1. The information in this table is completely hypothetical, and the speed of light delays are plausible, but completely back-of-the-envelope. One-way light times between Gateway 1 and Gateways 3, 4, and 2 are 4 minutes, 32 minutes, and 10 minutes, respectively. Those details are perhaps interesting but not the point of the example.

Note that any bundles sent by GW3 after 1156 GW1 time cannot be acknowledged before the next contact, because the bundle will arrive at GW1 after the end of GW1's transmission time (1200). Since the next contact between GW1 and GW3 might be the subsequent Monday, the acknowledgement delay might be VERY long. Bundles sent by GW4 after 1358 cannot be acknowledged during the current contact, because they

will not be received before GW1's transmission time ends at 1430.

Durst

Expires September 2003

[Page 6]

Similarly, bundles sent by GW2 after 1150 cannot be acknowledged during the contact.

Table 1. Contact schedule for Gateway 1.

Day	Destination	GW1 Send Time	GW1 Receive Time	Destination Send /Receive Time
Monday	GW3	0900-1200	0908-1208	0904-1204
Tuesday	GW4	1300-1430	1404-1534	1332-1502
Wednesday	GW2	1000-1200	1020-1220	1010-1210

DTN Gateways 2, 3, and 4 have entries in their contact schedules for the contacts with Gateway 1.

5 Systems participating in example bundle data transfer

Figure 3 is a revision of Figure 1 that highlights those portions of the Interplanetary Internet that participate directly in the bundle transfer example. Also shown in Figure 3 are the source and destination for the transfer and the Domain Name System equivalents in the terrestrial Internet (DNS 1), in the IPN "Backbone" (DNS 2), and in the Mars Internet (DNS 3). This figure will serve as the basis for the bundle data transfer example.

Table 2 provides the full host names of each of the primary elements in Figure 3. For this example, we hypothesize a URI scheme named "tcp" that is used to identify the Internet namespace, and a scheme named "dsn" that is used to identify a Deep Space Network namespace. Recall that in this example all bundle layer applications reside on TCP port 6769. The portion of the entity identifier used to demultiplex to the application using the bundle service has been omitted until the applications are discussed in [section 6.1](#). Host name tuples are in the form {region, administrative part}.

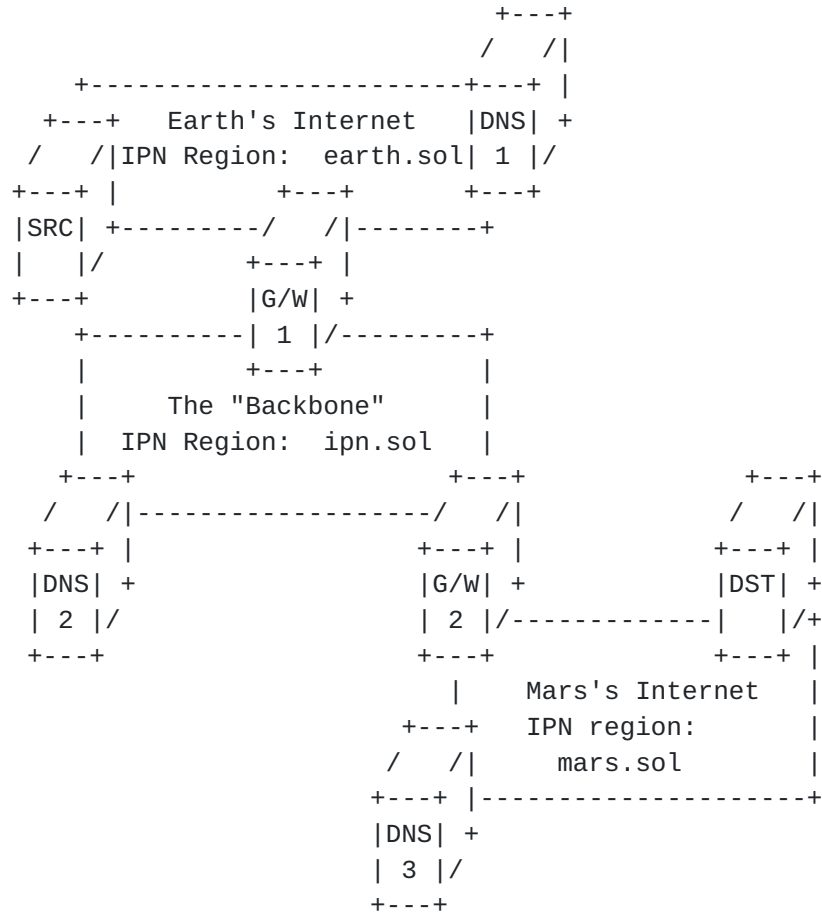


Figure 3. Interplanetary Internet showing principal systems.

Table 2. Host name tuples (entity ID demultiplexing info omitted).

Host	IPN Regions	Host name tuples
SRC	earth.sol	{earth.sol, tcp://src.jpl. nasa.gov:6769}
IPN G/W1	earth.sol	{earth.sol, tcp://ipngw1. jpl.nasa.gov:6769}
	ipn.sol	{ipn.sol, dsn://ipngw1. jpl.nasa.gov:6769}
IPN G/W2	ipn.sol	{ipn.sol, dsn://ipngw2. nasa.mars.org:6769}
	mars.sol	{mars.sol, tcp://ipngw2. nasa.mars.org:6769}
DST	mars.sol	{mars.sol, tcp://dst.jpl. nasa.gov:6769}

6 End-to-end Transfer

6.1 Step 0: Application instance registration

Before the beginning of this example, all of the bundle nodes in the network exchange their signed key material and credentials with next hop neighbors. These materials are cached for subsequent use. The applications at the source and destination register with their local bundle layers, providing similar key material and credentials to the bundle layer, and receiving in return their application instance identifiers. The destination has registered its IPN-standardized well-known demultiplexing id of "8," which becomes part of the entity ID used to refer to this particular application. The source has registered a demultiplexing identifier "0x1763421A" (a hexadecimal number) that (arbitrarily) corresponds to its process ID.

6.2 Step 1: Bundle creation and first-hop transmission

The application on the source host in Figure 3 has data that it wishes to send to the destination on Mars. The exact content of this data is opaque to the bundle transfer, but assume that it contains all of the information necessary to accomplish some desired function. That is, assume that application-specific instruction for storage, handling, error processing, and disposal accompanies whatever data object is to be operated upon. The application invokes the bundle layer, supplying it the information shown in Table 3.

The bundle agent verifies the signature, then creates a bundle and stores it in persistent storage (on disk or other non-volatile memory). There are some fields of the bundle header that the bundle agent must supply: the Sending Timestamp, the Source Host name tuple, the Custodian name tuple, and the time to live. (The application may state a time after which the data are obsolete, but the actual time-to-live field in the bundle header uses the application's data in combination with network restrictions on time-to-live to initialize this field properly.) The bundle layer consults routing tables and notes that its next-hop destination to reach the mars.ipn.sol region within the earth.ipn.sol region is tcp://ipngw1.jpl.nasa.gov:6769. (Since a host may reside in multiple IPN Regions, the source host name tuple is a function of the outbound route selected. The bundle layer uses this information to complete the Source Host and Custodian name tuples prior to transmission.) Having verified the application's signature, it incorporates this into the authentication information of the bundle header and appends its own credentials. It further notes that it has a continuous connection to that gateway, and transmits the bundle to it, retaining a copy for custodial storage. The bundle layer starts a retransmission timer for the bundle and awaits a custodial transfer acknowledgment.

Table 3. Information supplied by source application.

Item	Value	Description
Destination DTN tuple	{mars.sol, tcp://dst.jpl.nasa.gov:6769/8}	IPN Name tuple of the destination. Note that appl. demuxing ID 8 is supplied.
Source application demultiplex identifier	0x1763421A	Value used to identify the appropriate instance of the source application for response processing (becomes part of the source tuple)
Class of service	Custodial delivery, normal priority, data obsolete in 36 hours.	The services requested from the bundle layer.
Signature	Opaque	Digital signature of the app.-supplied information
User data	N/A	

6.3 Step 2: Bundle processing at first-hop destination

When the IPN Gateway {ipngw1.jpl.nasa.gov, earth.sol} receives the bundle via TCP, it checks the signature of the previous router and determines that the bundle has been forwarded by a legitimate source. Further, since this is the security boundary for the Interplanetary Internet, it verifies the signature of the sending application, requesting a copy of the application's public key from a secure distribution service if it does not already have one cached. Having verified that the application is authorized to access the deep-space portion of the Interplanetary Internet, the bundle layer replaces the signature block of the bundle layer entity with its own, leaving the application's signature block intact. It stores the received bundle on persistent storage (disk). The bundle layer consults the contact schedule and determines that the appropriate next-hop destination is in the "ipn.sol" IPN Region: {ipn.sol, dsn://ipngw2.nasa.mars.org:6769}, and that the next contact is the following Wednesday. The bundle layer manifests the bundle on the contact for that Wednesday.

In considering alternative contacts for the bundle, the dispatcher

checks the time-to-live in the bundle, which was 36 hours from the time of initial submission to the bundle agent at the source, to

Durst

Expires September 2003

[Page 10]

ensure that the route selected is consistent with the time-to-live requirements of the bundle. The bundle transport functionality of the bundle agent in ipngw1 accepts custody of the bundle, updates this information in the bundle header, and informs the source that has done so. The source's bundle agent deletes its custodial copy of the bundle. When the time arrives for contact with the ipngw2.nasa.mars.org DTN gateway to be established, the convergence layer establishes that contact via a protocol suited for reliable transfer across very long distances, referred to here as the Long Haul Transport Protocol (LTP). When informed that the contact is available, the bundle layer posts its bundles to the convergence layer for transmission via LTP.

6.4 Step 3: Bundle processing at gateway to destination IPN Region

The Mars gateway, {mars.sol, dsn://ipngw2.nasa.mars.org:6769}, receives the bundle from the Earth gateway via LTP. It verifies the signature block of the Earth gateway, then replaces that signature block with its own. It stores the bundle in persistent storage and accepts custody of the bundle, signaling back to the Earth gateway that it has done so. When the notification of acceptance reaches the Earth gateway, ipngw1 deletes its custodial copy. The Mars gateway consults its routing tables to find an outbound contact to forward the bundle over. It determines that the appropriate next hop is the destination itself, that the proper transport protocol is TCP, and that the destination is accessible immediately. The gateway verifies that the time-to-live has not expired, and forwards the bundle to the destination.

6.5 Step 4: Bundle processing at destination

The bundle layer at the destination entity receives the bundle via TCP, verifies the signature block of the IPN G/W2, stores it on its own persistent storage, and accepts custody of the bundle from IPN G/W2. The bundle agent "awakens" the destination application process identified by the Destination Application demultiplexing portion of the entity ID. This may involve creating a new instance of a server from a daemon process, signaling an idle running process, or reinstantiating a process that has been suspended with its state stored on persistent storage. The bundle agent deletes the copy of the bundle from persistent storage when the application has received it. The destination application may generate an application-layer acknowledgment in a new bundle and send it to the source.

7 Error Conditions at the Bundle Layer

This section describes the error conditions that may arise at the bundle layer during bundle creation and transport. When these errors occur within the sender's DTN region, it may be possible to conduct a

near-real-time dialog to correct them before the bundle is forwarded.
We say 'may be possible' because even if two nodes are in the same

Durst

Expires September 2003

[Page 11]

IPN domain, they may not have real-time connectivity. An example of such a situation would be if a lander were on the opposite side of the planet from its DTN gateway, and used bundles to communicate with the gateway through a low altitude orbiter, with the orbiter itself serving as a bundle agent.

Table 4: Error conditions at the bundle layer.

Error	Description	Places where Error can Occur
1*	Unknown destination region	Source Bundle Node
2*	Invalid Source App.	Source Bundle Node
3*	Bundle Parameter Syntax Error	Source Bundle Node
4*	Bundle Parameter Semantic Error	Source Bundle Node
5	Insufficient buffer space	Any bundle node
6*	Time exceeded	Any bundle node other than the source
7*	Source Entity Access Denied	Any bundle node other than the source
8*	Invalid Entity ID in Destination Name	IPN gateway serving destination DTN region
9*	Invalid Destination App.	Destination
10*	End-to-end Access Denied	Destination

The errors that can occur at the bundle layer are shown in Table 4. Error numbers marked with an asterisk (*) are reported back to the sending application (or to the location specified in the reply-to field of the bundle header, if it differs from the sending application).

* Unknown Destination Region: This error occurs when the source bundle node is directed to create a bundle destined for a DTN Region that is not recognized. Note that only the DTN Region part of the destination name has to be interpretable outside the destination's DTN Region. In particular, the entity identifier of the destination name need not be interpretable to the source (assuming the source and destination are in different DTN

Regions), so it cannot necessarily be checked when the bundle is created. (It is possible in the case of a mis-configured DTN

Durst

Expires September 2003

[Page 12]

router that this error might occur at a location other than the Source Bundle Node. However, this is an error condition within the bundle router that should be corrected and the bundle routed, rather than an error that should be reported to the sending application.)

- * Invalid Source Application: If the source authentication information fails, the source bundle layer responds with an Invalid Source Application error.
- * Bundle Parameter Syntax Error: The source bundle layer may check the syntax of some of the bundle-creation parameters (i.e. it may ensure that the end-to-end and DTN access security certificates are well-formed, etc.) If a parameter is found to be syntactically incorrect or obviously and definitely erroneous, the bundle layer will report a Bundle Parameter Syntax Error back to the source that includes, at a minimum, the parameter that caused the error.
- * Bundle Parameter Semantic Error: If the source bundle layer can identify a particular bundle creation parameter as being well-formed but unserviceable, it will report a Bundle Parameter Semantic Error to the source application that includes, at a minimum, the parameter that caused the error.
- * Insufficient Buffer Space: If a bundle node does not have sufficient buffer space to accept a bundle, it drops the bundle and generates an Insufficient Buffer Space error. Note that a bundle node may choose to drop lower priority bundles in order to make room for higher priority ones. This error is not propagated back to the source.
- * Time Exceeded: If the time-to-live field (either the source-provided TTL or the internal bundle TTL) expires, the source is notified with a Time Exceeded message. These errors are propagated to the source application.
- * Source Entity Access Denied: This error indicates that the source entity does not have access to a needed resource at a DTN node. The source might not be authorized to use the node at all, or it might not have authorization to use a particular interface required by the bundle. Source Entity Access Denied errors indicate that the source is not **authorized** to use a particular resource; other errors (e.g. Insufficient Buffer Space) indicate that a particular resource is **unavailable** to a bundle. For example, an entity on the surface of a planet might be authorized to communicate, using the bundle protocol, with another entity on the other side of the planet via a low-altitude orbiter that is also an IPN gateway. The sender might not, however, be authorized to send bundles across interplanetary space. In this case bundles sent to the orbiter destined for the other side of the planet would not cause errors, while any bundles with off-planet destination addresses would. Source Entity Access Denied errors are propagated back to the source application.
- * Invalid Entity ID in Destination Name: Once a bundle has reached

its destination DTN Region, the Entity ID part of the destination name can be parsed. If the Entity ID of the destination name is

Durst

Expires September 2003

[Page 13]

not valid, the source is notified with an Invalid Administrative Destination Name error message.

- * Invalid Destination Application: If the destination bundle layer cannot instantiate the destination application (based on the demultiplexing information the region-specific entity ID of the bundle), it notifies the source application with an Invalid Destination Application error message.
- * End-to-End Access Denied: If the bundle destination does not accept the bundle due to an authentication or access-control error, the source is notified with an End-to-End Access Denied Message.

8 Normative References

[DTNRG03] Cerf et al, "Delay-Tolerant Network Architecture," ([draft-irtf-dtnrg-arch-03.txt](#)), October 2003. Available from <http://www.dtnrg.org>.

9 Informative References

[IEEE] Burleigh et al, "Delay-Tolerant Networking: An Approach to Interplanetary Internet," IEEE Communications Magazine, June 2003, pp 128-136. Available from <http://www.dtnrg.org>.

10 Security Considerations

Security is an integral concern of the Delay Tolerant Network Architecture. The infrastructure protection elements of the Delay Tolerant Network Architecture are still evolving, and this example tracks the state of the security architecture in [[DTNRG03](#)].

11 Acknowledgments

The author gratefully acknowledges the contributions of the following individuals, who either helped to walk through this example on various whiteboards or assisted in making the documentation of it more readable: Scott Burleigh, Jet Propulsion Laboratory; Vint Cerf, Worldcom; Keith Scott, MITRE; Kevin Fall, Intel Corporation; Howard Weiss, SPARTA, Inc.; and Leigh Torgerson and Adrian Hooke, both of the Jet Propulsion Laboratory.

12 Author's Address

Robert C. Durst
The MITRE Corporation
7515 Colshire Drive M/S H300
McLean, VA 22102
Telephone +1 (703) 883-7535
FAX +1 (703) 883-7142
Email durst@mitre.org

Please refer comments to dtn-interest@mailman.dtnrg.org

