Endpoint Naming for Space Delay / Disruption Tolerant Networking

Loren Clare, Scott Burleigh
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive
Pasadena, CA 91109
{Loren.P.Clare, Scott.C.Burleigh}@jpl.nasa.gov

Keith Scott
MITRE Corporation
7525 Colshire Drive
McLean VA 22102
kscott@mitre.com

Abstract—Delay/Disruption Tolerant Networking (DTN) provides solutions to space communication challenges such as disconnections when orbiters lose line-of-sight with landers, long propagation delays over interplanetary links, and other operational constraints. DTN is critical to enabling the future space internetworking envisioned by NASA. Interoperability with international partners is essential and standardization is progressing through both the CCSDS and the IETF.

The DTN architecture, defined in RFC 4838, "uses a flexible naming scheme (based on Uniform Resource Identifiers [RFC3986]) capable of encapsulating different naming and addressing schemes in the same overall naming syntax." Although DTN was originally conceived with a space focus, as the technology underpinning an InterPlanetary Network (IPN), DTN has found increasingly broad application to military networks, wireless sensor networks, village networks, "pocket switched" networks, and peer-to-peer networks. In these latter contexts the generality of the naming structure permits the sophistication of "intentional naming" as recently proposed in Internet Draft draft-pbasu-dtnrg-naming-00.

We argue in this paper, however, that when the application domain is limited to the space context a much simplified naming scheme is preferred: names may be essentially tuples (x, y) where x and y are finite non-negative integers identifying "node number" and "service number". This scheme was demonstrated in the DINET deep space flight experiment, and it is currently implemented on nodes in the NASA DTN Experimental Network (DEN). We discuss the rationale for this constrained naming structure, based on considerations of the space context. Alternative naming schemes may also be accommodated in a space DTN node, but we suggest that this simplified scheme should be the minimal naming mechanism that all space DTN nodes must implement.

A recommendation for a node number assignment strategy is offered which is bandwidth-efficient and fair to agencies/centers/projects using space DTN. Such assignments could be made under the auspices of the Space Assigned Numbers Authority (SANA) now being established by the CCSDS.

[1] IEEEAC paper #1379, Version 2, Updated December 18, 2009

TABLE OF CONTENTS

1. BACKGROUND AND HISTORY

Delay/Disruption Tolerant Networking [1] has been proposed as an internetworking protocol to support space communications. The DTN Bundle protocol [2] provides an internetwork layer that conveys data in an internetwork data unit, called a *bundle*, from one identified network node to another via zero or more forwarding nodes. The Bundle Protocol does not assume continuous connectivity and specifically allows for in-network data storage such as might take place when Earth can transmit to an orbiter which then has to store the data until the orbiter can relay the data to a landed asset.

The Bundle protocol can support a multitude of different endpoint naming schemes, so long as all such schemes adhere to a standard endpoint name identification pattern: principally, all endpoint names must be Uniform Resource Identifiers [RFC3986] of the general form *scheme_name*:*scheme_specific_part*. Little work has been done on how routing will work when source and destination endpoints are identified by names expressed in different schemes, and even the exact use of the default 'dtn' scheme is not fully standardized. This paper presents an approach to endpoint naming that was developed for space internetworking, meeting the needs of the space community while enabling the network to scale up to large numbers of nodes and preserving interoperability with an emerging terrestrial DTN infrastructure.

Comparison with Internet Naming and Addressing

The Internet Protocol (IP) uses 32-bit (IPv4) or 128-bit (IPv6) numeric *addresses* to identify interfaces and route packets. Note that an address is different from a *name*:

- An address has "topological significance", i.e., it identifies a location in some sort of space: a location on a specific street in a specific city, a location in a specific subnet in a specific network, etc. Given a map of the relevant space, the address itself provides all the information that is needed in order to pass information to whatever entity is occupying the identified location, but it does not identify that entity.

- A name identifies an entity to which one might want to pass information, but it provides no information as to the location of that entity. In order to pass information to the named entity we first have to determine its location in some other way.

In the Internet, node names such as www.nasa.gov must be resolved to addresses before communication can begin. This resolution is typically done via the Domain Name Service (DNS), a distributed hierarchical database. Thus before any data can be sent to a named entity (rather than simply to a known location), a database lookup that might involve machines far from the source has to be completed. If the destination entity is *mobile* – that is, its location in the network may change over time (even if its location in physical space does not) – then the results of prior address lookups may lose validity and repeated lookups may be necessary.

To improve the efficiency of IP routing, IP addresses are typically aggregated by assigning addresses with a common prefix to nodes residing in a common subnet. Thus all interfaces whose IP addresses begin with 137.78 can be found in the subnet serving the Jet Propulsion Laboratory in Pasadena, CA. This administrative decision allows routers that are far from JPL to use a single routing table entry for all of the 65,534 possible machines at JPL. IP addresses would have topological significance even if they were not aggregated in this way, but the "maps" (routing tables) within which they are meaningful could not be nearly so concise. This economy in representation of routing information has been essential to the ability of the Internet Protocols to survive the growth of the Internet.

Interplanetary Internet Naming and Addressing

The original Interplanetary Internet design was based on two-level hierarchical endpoint identification. Each endpoint identifier in the interplanetary internet was a tuple consisting of a *region* and a *region-specific-part*. Regions were intended to be topologically significant and useful for coarse-grained routing; that is, they were partial addresses, somewhat akin to ZIP codes. For example, 'Earth', 'Mars', and 'Moon' might be useful regions. Within each region, the 'region-specific-part' of the identifier was intended to identify a particular node (or interface) within the region; region-specific-parts might or might not have topological significance. Table 1 lists some example region-specific addresses for the three regions.

In this example the Earth uses some version of IP with DNS support in its region-specific-parts, the Moon uses IPv4 addresses with no DNS, and Mars uses a non-IP-based scheme such as "MarsHostXXX" where XXX is a non-topological numeric identifier for a host. Figure 1 shows how the regions might be interconnected.

The rationale behind two-tier hierarchical routing was essentially the same as that behind FishEye Routing [3] – nodes generally need to know more about things close to them than about things far away. Thus in the Interplanetary Internet, a node would be able to route data to all other nodes in its region, but might only know one or two gateways to get to other regions.

Consider a bundle with destination identifier **(Mars, MarsHost001)** in Figure 1. Because nodes outside the destination region (Mars) don't have to be able to interpret the region-specific part of addresses at all, node C on Earth only needed to know how to route data to get the data to the Mars region. In particular, *no* name-to-address translation has to occur at any node on Earth. To achieve the 'regional' routing, node A on Earth would advertise reachability to the Mars region, and node C would simply examine the 'region' part of the endpoint identifier in order to know that the data needed to be routed to node A.

Similarly, MarsHost001 could send data to the destination **(Earth, www.nasa.gov)** and not have to know anything about how to interpret the string www.nasa.gov. In particular, no node on Mars would need to resolve www.nasa.gov to an IP address. The data would travel to Earth, and only once it reached the Earth region would the host name be resolved to an IP address. If the data were routed via the link from MarsHost002 to node A on Earth, the resolution would take place at node A. If the data were routed via the moon, the name-to-address translation would take place at node B (the ingress node into the Earth region).

Using this hierarchical routing scheme, even if the Earth and the moon shared an IP routing space, they would be considered two different regions. Thus (Earth, www.example.com) would be a different node than (moon, www.example.com). They would be differentiated by the location at which the region-specific part of the address was bound to an IP address.

Thus the name-to-address translation didn't need to happen at the source and before data was transmitted; only the first-level resolution to 'Earth' was required. The full resolution to identify a particular destination could take place after the data is already in flight. If the destination region was well-connected, then the name-to-address resolution operation via DNS would be fast once the bundle reached the Earth region, and at that point a destination IP address would be fixed.

Table 1: Example region-specific addresses.

Region	Region-specific part of address
Earth	www.nasa.gov www.ee.ucla.edu
Moon	10.17.23.44 137.79.33.12
Mars	MarsHost001 MarsHost002 MarsHost 003

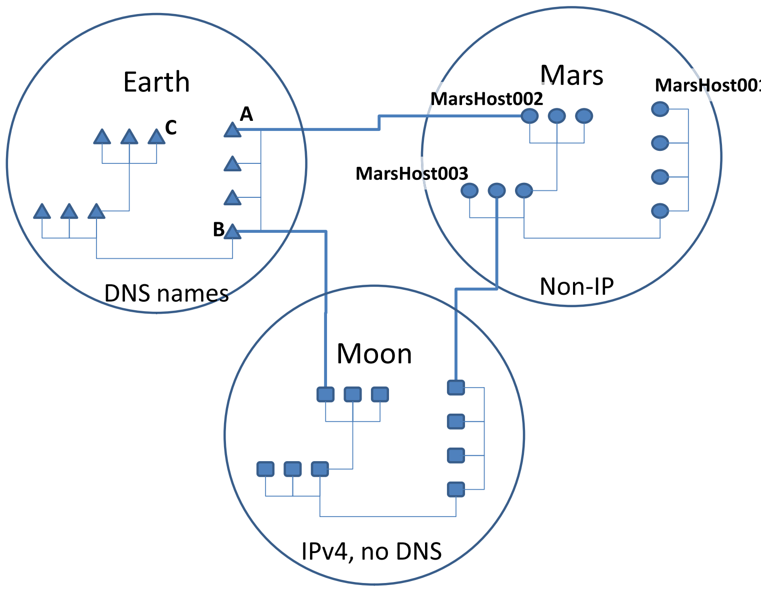

Figure 1. Interplanetary Internet naming using regions.

This *late binding* of addresses to particular hosts in the Interplanetary Internet was important partly because it provided a way of accommodating node mobility in communication environments characterized by lengthy end-to-end delivery latency. Mobile nodes' locations can change at frequent intervals, so long end-to-end latency increases the probability that the location of the node at the time of a bundle's arrival is different from its presumed location at the time of original transmission. Routing strictly by address therefore increases the chance of delivery failure and data loss. Nodes' names typically change far less frequently, so they were safer to use when making routing decisions.

Late binding also had an additional advantage: because region-specific-parts weren't required to have topological significance, endpoints in the network might be identified by something other than location, such as 'role'. Thus one could envision addressing data to 'the current mission operations officer for mission X'. This would require some infrastructure for resolving the mnemonic for the officer to the officer's *current* IP address. "All rovers within some particular region on Mars" might also be expressible if the Mars region's addressing scheme were Mars-location based.

2. DTN NAMING AND ADDRESSING

DTN is essentially a generalization of the original Interplanetary Internet design. It uses absolute Universal Resource Identifiers (URIs) [4] as DTN Endpoint Identifiers (EIDs). The general form of any absolute URI (and therefore any DTN EID) is:

scheme_name:*scheme_specific_part*

EIDs are the tokens by which DTN bundles are routed to their destinations, but they are not necessarily addresses because neither the scheme name nor the scheme-specific-part is required to have topological significance. An EID's scheme-specific-part may have topological significance, depending on the definition of the named scheme, but alternatively it may be a name – or not even a name, but only an expression which must be evaluated in some way in order to be converted into a name or address (or multiple names or addresses). A DTN system can use any URI scheme it chooses, and there are as yet no real conventions as to how different schemes could be used and how they might interact.

The 'default' scheme for DTN endpoint naming is the 'dtn' scheme. The syntax and semantics of the dtn scheme are defined by the DTN2 reference implementation software. dtn scheme syntax is currently as follows:

dtn://*machineID*/*appID*

where

machineID is an ASCII string that identifies a particular computer running a DTN2 protocol stack.

appID is an ASCII string that identifies a particular application using the DTN protocol stack identified by the *machineID*

There is nothing particularly special about the dtn scheme. An implementation of the DTN protocols might choose to implement any number of naming/addressing schemes, and might not implement the dtn scheme at all. However, because all DTN endpoint identifiers are URIs, every implementation must be able to identify the scheme name that is part of an EID. This would allow an implementation that did not know anything about the scheme-specific part of the dtn scheme to at least identify the scheme.

Example DTN scheme EIDs

Thus the following are valid EIDs under the dtn scheme:

dtn://myMachine/dtn_recv

dtn://everyoneWithin100MetersOfMe
[Local personal area communications.]

dtn://rover3.mars.sol/other
[Note: while rover3.mars.sol is formatted as a DNS name, it is just a string to the dtn scheme.]

dtn://allMarsOrbiters/cmdApp

dtn://allSpacecraftInCruise/otherApp
[An example of a destination EID whose membership might change with time.]

dtn://128.29.23.37/dtncpd
[Note: in this context, 128.29.23.37 is NOT an IP address, it's just a string.]

All of these endpoint identifiers are just strings, and although they carry *connotations* (e.g., allMarsOrbiters), such ideas are meaningless without a routing protocol that supports them.

The DTN2 definition of dtn scheme semantics is currently based on the use of *globbing* to match destinations against entries (patterns) in routing tables. Globbing allows for limited wildcard characters. In particular, '*' matches any string, strings in brackets '[', ']' match any single character in the brackets, and '.' matches any single character. Multiple wildcards in a pattern are allowed.

Table 2 shows a sample table-based-router routing table as might be used by the DTN2 reference implementation. In the table-based routing implementation, next hops are identified by references to their outbound links, not the next hop EID. Each link contains information about the next hop address and protocol to be used.

Table 2: Example DTN TableRouter Routing Table

Destination	Next Hop
dtn://myMachine/*	Link1
dtn://*_yellow_*/*	Link2
dtn://*/*	Link3
dtn://lat35.*lon-74.*	Link2
otherScheme:*	Link6

The 'dtn://lat35.*lon-74.*' entry in the table shows an example of using multiple wildcards to designate all machines in a box bounded by 1 degree of latitude and longitude.

The 'otherScheme:*' entry represents a routing table entry that routes all bundles that use the naming/addressing scheme 'otherScheme' out Link6.

3. SPACE REQUIREMENTS CONSIDERATIONS

Although the DTN architecture was originally developed with interplanetary networking primarily in mind, it was quickly realized that DTN offers profound advantages across a wide range of application domains. A large number of valuable developments have ensued in such areas as military networks (including progress through the DARPA DTN program), wireless sensor networks, village networks, "pocket switched" networks, and peer-to-peer networks. The DTN architecture and Bundle Protocol were designed

Table 3. Space vs. Terrestrial DTN Naming Considerations.

Issue	Space DTN Environment	Terrestrial DTN Environment
Capability / Power of Naming	Limited (at least for the short term): need to address 'applications' on 'hosts'	Researchers looking at everything from 'IP-like' to content-based addressing to sensor network queries
Scalability	To 1000s of nodes spread across 10s of agencies	To millions of nodes spread across thousands of administrative entities
Mobility	Limited, planned	Common, unscheduled
Concern with overhead	High	Medium

to accommodate these and other yet-to-be-discovered application domains in a variety of ways, including provision for advanced destination endpoint naming capabilities such as Intentional Naming in DTN [8].

The naming scheme proposed for DTN is space is substantially restricted relative to these extended features. Table 3 provides the rationale for this.

4. CBHE-CONFORMANT NAMING

"Compressed Bundle Header Encoding" [5] (CBHE) is a method for reducing bundle protocol overhead in bandwidth-constrained environments. The scheme-specific part of any URI formed under any CBHE-conformant scheme always has the following structure:

node_number.service_number

The recently introduced 'ipn' scheme conforms to CBHE. The following are valid EIDs under the ipn scheme:

ipn:0.0

ipn:631.0

ipn:233.115

ipn:29874789966.7112

node_number is a unique identifier for a DTN "bundle node", defined in the Bundle Protocol specification as an "entity that can send and/or receive bundles". (Each DTN2 "dtn daemon" is typically a single bundle node.) A node number is not an address: it is a name that happens to be written using the restricted alphabet of the decimal digits 0-9 rather than the entire alphabet of printable ASCII characters.

Node numbers differ from IP addresses in two significant ways. First, IP addresses don't always uniquely identify communicating entities: a single host machine can have multiple IP addresses, one per IP interface. Second, IP addresses have topological significance: as noted earlier, part of the address is a network identifier that can be used for route aggregation in a routing table, provided the addresses are assigned with some care. Node numbers are not addresses at all, as they have *no topological significance*.

service_number is a demultiplexing token used to identify a particular application on a DTN node. Service numbers serve a function similar to that of the appID in the dtn scheme, the protocol number in an IP packet, or the port number in a UDP datagram.

Both node number and service number are nonnegative integers of arbitrary size. That is, node number is not limited to 32 bits as IPv4 addresses are (or 128 bits as IPv6 addresses are), and service number is not limited to 16 bits as TCP and UDP port numbers are (or 8 bits as protocol numbers are).

One disadvantage of CBHE-conformant endpoint IDs is that they aren't especially user-friendly: remembering numbers is generally harder than remembering names. When the numbers are kept fairly small (like spacecraft IDs, or like Interstate highway numbers) this may not be much of a problem; flight mission operators already use such numbers in mission operations every day. Moreover, modern user interfaces built along the lines of Web browsers could be expected to spare users the trouble of remembering or typing endpoint IDs altogether. Still, for some purposes there would likely be a need for a directory service somewhere – something similar to DNS – for looking up correspondences between node numbers and user-meaningful node names. We would expect this directory service to be less dynamic than DNS because, again, the rate at which a node's name changes is typically lower than the rate at which its address (location) changes.

The countervailing advantage is that the use of numeric node identifiers and demux tokens enables endpoint IDs to be compressed into binary (rather than string) form. This has two main benefits:

1. Processing is faster, because integer storage, retrieval, and comparison operations are much quicker than string manipulation.

2. The compressed form of endpoint ID representation can be encoded in transmitted bundles' primary bundle blocks, sharply reducing overhead. For very large bundles this advantage is not a big deal, but for small bundles (e.g., real-time commands, telemetry packets, custody signals) it can improve bandwidth utilization significantly.

Because these economies can be especially important in the resource-constrained communications environment of space flight missions, NASA's space DTN program has adopted the ipn scheme as the mandated minimum capability for endpoint naming in space DTN.

Note that dtn scheme EIDs could be compressed in a similar manner. All that would be required would be for the CBHE compression mechanism to treat dtn scheme EIDs of the form

dtn:// *node_number.service_number*

as CBHE-compliant after ignoring the initial '//' on compression and to reintroduce the '//' on decompression. While the presence of the '//' makes the dtn EIDS non-conformant under the current draft CBHE specification, the mechanism just described would yield the same level of overhead in compressed bundles.

5. CBHE COMPRESSION

The structure of the primary bundle block (the main 'header' of the bundle protocol) is depicted in Figure 3 below.

Numbers in Bundle Protocol blocks are encoded as Self-Delimiting Numeric Values (SDNVs) which enable any size number to be represented using a variable number of octets. The first bit of each octet determines whether the representation continues to a subsequent octet or not. A depiction of this scheme is provided in Figure 2. The values that can be represented in one octet are 0 through 2^7-1; two octets can represent all values 0 through 2^{7+7}-1, etc. The dashed elements of Figure 3 are all SDNVs.

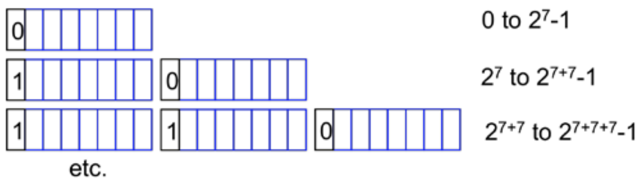

Figure 2. SDNV Numeric Representation Method.

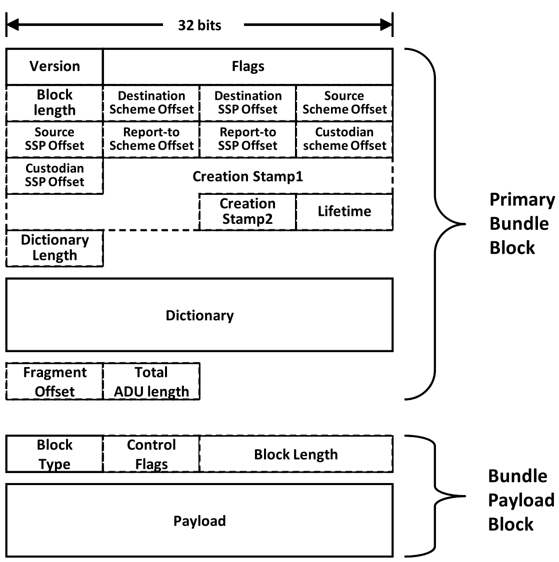

Figure 3. Bundle Protocol Structure.

The primary bundle block contains a number of offsets into a *dictionary* containing the concatenation of all the scheme names and scheme-specific parts of the various endpoint IDs carried by the bundle. This dictionary provides one level of compression in that if all of the addresses share a particular scheme, that scheme appears once in the dictionary and is pointed to by the various scheme offsets. This doesn't help with scheme-specific parts of names that happen to be long ASCII strings, however.

CBHE-conformant BP names are able to substantially reduce the size of the Primary Bundle Block. When CBHE compression is applied, the node and service numbers of the various endpoints are encoded directly in the 'offset' fields of the primary bundle block, which no longer identify offsets within the dictionary; the dictionary is omitted, substantially reducing the length of the primary block. Figure 4 depicts the difference in Primary Bundle Block contents between non-compressed and CBHE-compressed primary blocks.

To illustrate the overhead savings provided by CBHE compression, consider a set of nodes where a source with endpoint ID dtn://a.b is sending bundles to a destination with endpoint ID dtn://c.d, with no report-to endpoint (so that the report-to EID is dtn:none) and current custodian dtn://e.f.

All of the strings that form pieces of the source and destination EIDs fall within 127 bytes of the start of the primary bundle block, so all of the offsets can be represented by 1-byte SDNVs. The required dictionary entries would be as shown in Table 2.

Table 2: Dictionary Entries for non-compressed example

Dictionary Entry	Comment
dtn	Scheme name, common to all EIDs, included once
//a.b	SSP of the source EID
//c.d	SSP of the destination EID
none	SSP of the report-to EID
//e.f	SSP of the custodian EID

Thus the dictionary occupies 29 bytes (each of the strings is NULL-terminated).

A comparable example using the ipn naming scheme might use ipn:1.2 as the source EID, ipn:3.17 as the destination EID, 0.0 ("dtn:none") as the report-to EID, and 5.0 as the current custodian EID. When CBHE compression is applied, the numbers in the EIDs are inserted directly into the 'offset' fields of the primary bundle block and the dictionary has zero length.

The 'base' length for the Primary Bundle Block given the

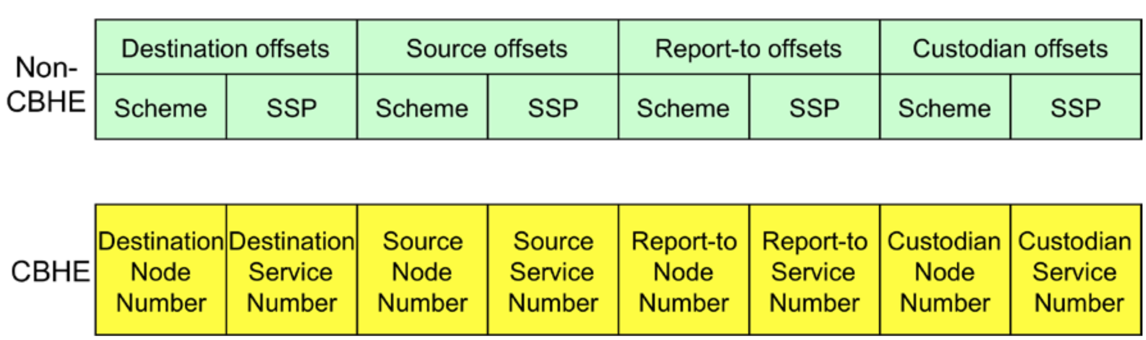

Figure 4. Comparison of Contents of Primary Bundle Block Offsets with/without CBHE Compression

current date (which affects the length of the Creation Timestamp) is 19 bytes (18 bytes for the bulk of the primary bundle block plus 1 byte of dictionary length). The payload block requires a block type (1 byte), block processing control flags (1 byte SDNV), and an SDNV indicating the payload length. Thus the minimum total overhead when using the dtn scheme would be 51 bytes, vs. only 22 bytes for a CBHE-compressed bundle.

Figure 5 shows the overhead of the primary bundle block as a percentage of overall length for different bundle payload lengths.

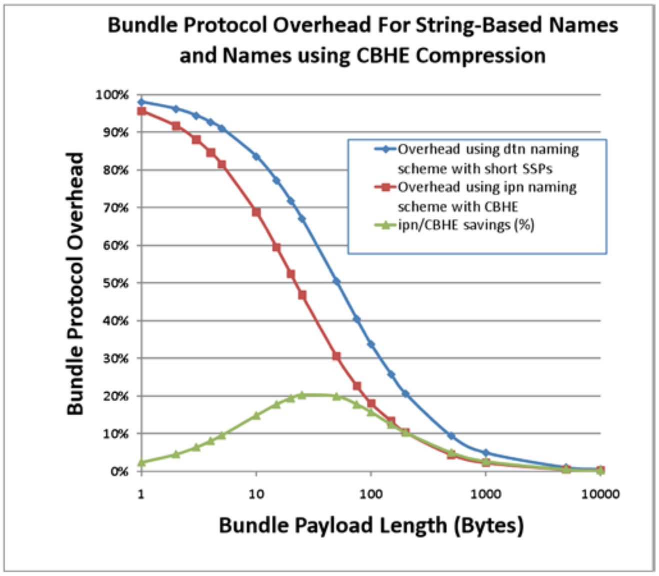

Figure 5. Bundle Protocol Overhead

6. Node and Service Number Assignment

Given a decision to adopt CBHE-conformant endpoint naming for DTN operations in space, further decisions must be made as to the methods by which node numbers are assigned to nodes and service numbers are assigned to services.

One possible "bootstrap" mechanism for assigning node numbers would be based on the use of a three-octet SDNV for each node number. The high-order octet of each such SDNV would identify one of 127 *naming authorities*. A zero value in this high-order octet would signify the "null authority" and would be reserved for research purposes. The other 126 naming authorities might be space agencies, instrument teams, launch services, etc. Each naming authority would be authorized to assign up to 16383 node identification values – occupying the two low-order octets of the node number SDNV – to DTN nodes in its administrative purview. Figure 6 depicts this node number assignment mechanism.

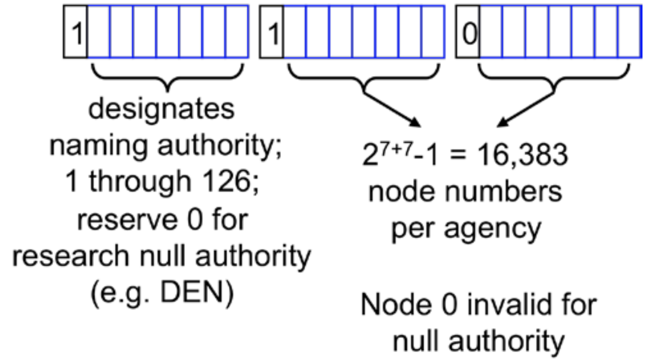

Figure 6. Space EID Naming Assignment Method.

Obviously this rather simple-minded approach has its drawbacks: for example, as described it is not extensible to more than 126 designated naming authorities nor to node populations in excess of 16383 nodes subject to any single authority. The SDNV structure offers one potential work-around for these difficulties: this entire numbering structure could be viewed as a special case of a more general 4-octet-based structure. The high-order octet of such a 4-octet node number would identify a node number assignment mechanism, opening many more possible node number ranges, and a zero value in the high-order octet would simply signify the "bootstrap" mechanism described here. (A zero value in the high-order octet of an SDNV is equivalent to the omission of that octet.)

It must be emphasized, however, that the approach described in this section is merely a proposal. Negotiation of an international consensus resolution of this issue is a matter for future work.

In any case, we expect that node number assignment will be delegated to assignment authorities in some way. It is expected that the designation of the naming authorities themselves will be managed through the Consultative Committee for Space Data Systems (CCSDS). A draft outlining the role and responsibilities of the Space Assigned Numbers Authority (SANA) has been generated [7]. SANA is analogous to IANA, which controls numbers for Internet protocols, assigns the Country Code Top Level Domains, and maintains the IP Address allotments. As specified in the SANA responsibilities draft:

> SANA assigns and registers CCSDS protocol parameters and other CCSDS objects as directed by the criteria and procedures specified in CCSDS documents. SANA is the core registrar and first-level authority for CCSDS registries.

We envision that service numbers would be allocated similarly to port numbers in the Internet. That is, we expect that some number of service numbers will be reserved for 'well-known' services, probably registered at SANA, while the rest will be available for ephemeral use.

7. INTERNETWORKING MULTIPLE SCHEMES

While we argue that space DTN nodes should be required to implement the ipn scheme, the DTN architecture is designed to allow internetworking among nodes employing different schemes. In this section we use several simple examples illustrate this.

In each of the following three figures, we consider a scenario in which there are two Mission Operations Centers (MOCs) on Earth, two Ground Stations (GSs) on Earth, two relay orbiters at Mars, and two rovers on the surface of Mars. There are two space agencies, depicted red and blue, which own one each of these element types. The subsequent three figures are distinct in regard to the naming schemes that are employed.

In Figure 7, all nodes use the standard dtn scheme. The bundle routing table for the red MOC node is shown for the "next hop" (or next Convergence Layer peer) to be used to reach the blue rover and similarly for the red orbiter. In the figures 'BLUE_GS', 'RED_GS', 'BLUE_ORB', 'RED_ORB' and 'BLUE_MOC' are endpoints expressed in the schemes of the DTN convergence layers used to communicate with the next DTN hop. Because the convergence layers in the example use TCP and UDP, these represent IP addresses or DNS names resolvable by the various DTN nodes.

Figure 8 shows the same scenario but with the ipn naming scheme used. Finally, Figure 9 shows the case where the dtn scheme is used on Earth and the ipn scheme used in space; bundle routing tables are shown for a MOC and a rover.

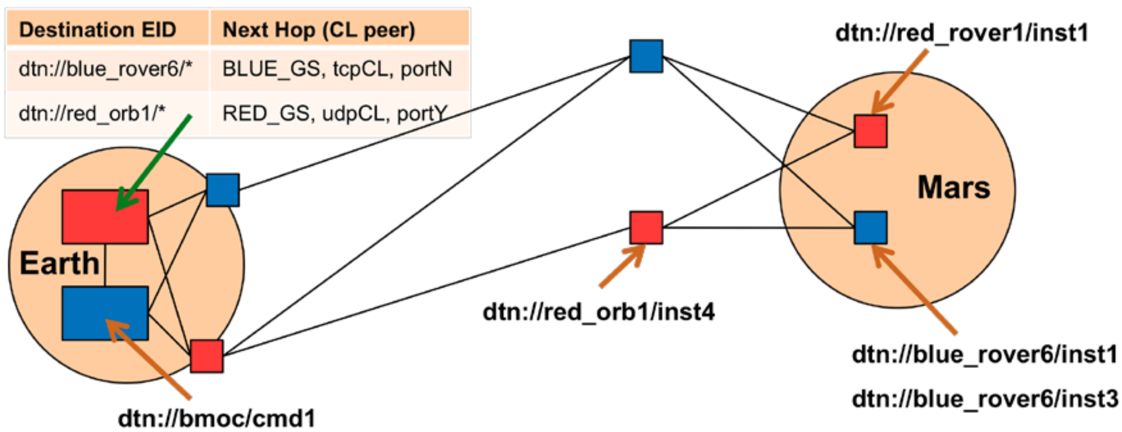

Figure 7. Example Routing with Standard dtn Naming.

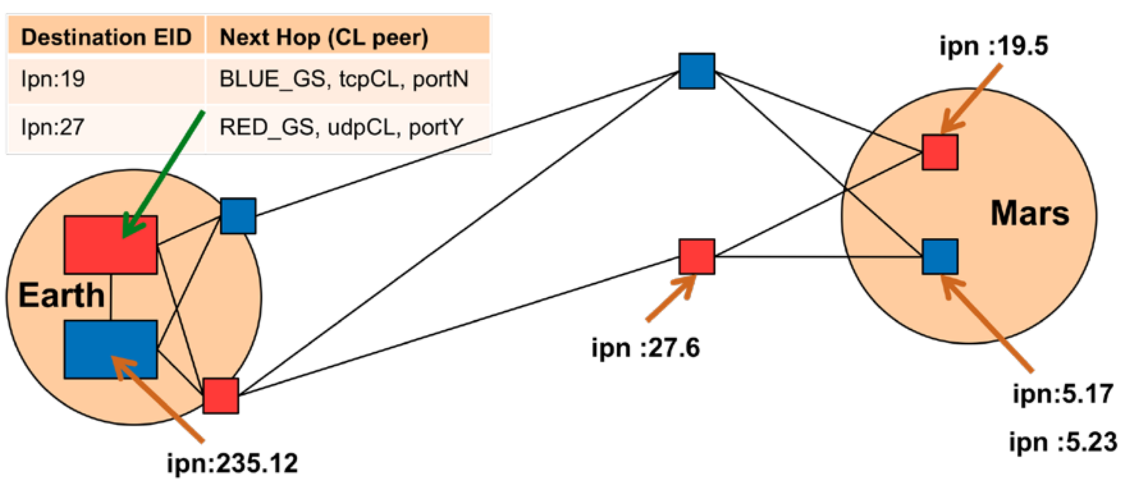

Figure 8. Example Routing with ipn Naming.

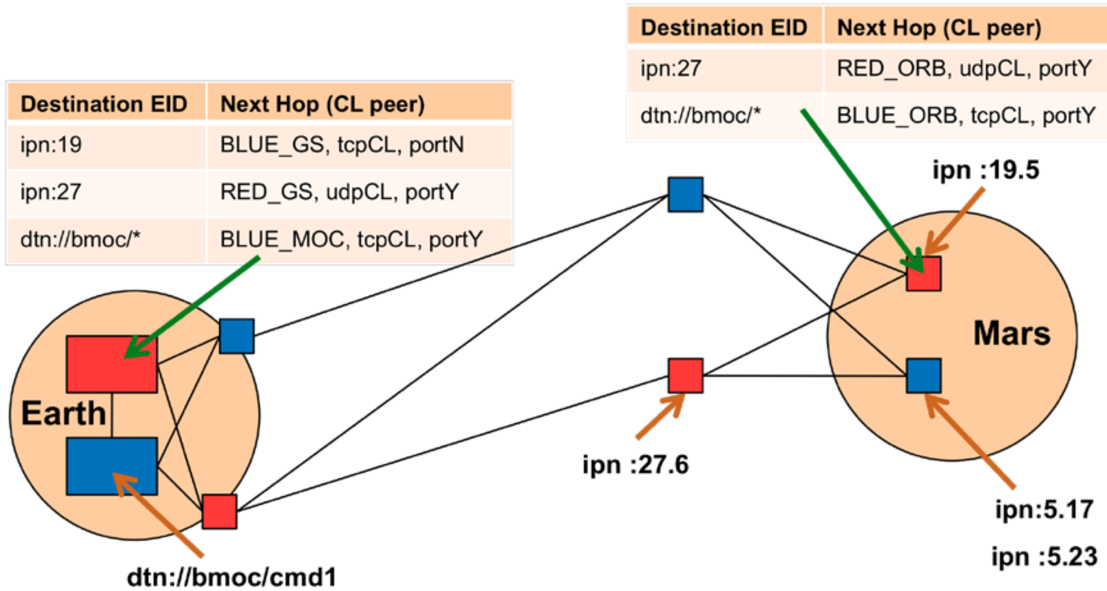

Figure 9. Example Routing with Mix of ipn and Standard dtn Naming.

8. CONCLUSIONS

This paper has described recent progress in specifying the naming of Endpoint Identifiers (EIDs) in Delay/Disruption Tolerant Networking (DTN) in the domain of space communications. A simple scheme, called "ipn," is to be implemented by all NASA space DTN nodes, based upon rationale that includes need for processing and transmission economy and recognition that the number of entities in space will be relatively limited. Alternatives yielding greater functionality are not precluded, and the general DTN architecture and its implementation via the Bundle Protocol will allow broad flexibility and evolvability in naming, e.g., enabling interoperation with alternative schemes used on terrestrial networks.

The space DTN EID naming scheme has been defined as a pair of integers *x.y* which identify the node number and the service number, analogous to host name and port number in the IP protocols. This representation is highly compressible within the bundle protocol via Compressed Bundle Header Encoding, reducing the size of the primary bundle block to as little as 27 bytes.

A proposed node number assignment strategy is presented which is bandwidth-efficient and fair to agencies/centers/projects using space DTN. Node number assignment authorities may be designated by the Space Assigned Numbers Authority (SANA) being developed by the CCSDS.

9. ACKNOWLEDGEMENTS

The research described in this paper was carried out in part at the Jet Propulsion Laboratory, California Institute of Technology and the National Aeronautics and Space Administration.

10. REFERENCES

[1] RFC 4838, Delay-Tolerant Networking Architecture, V. Cerf et. al., April 2007.

[2] RFC 5050, Bundle Protocol Specification, Scott, K. and Scott Burleigh, November 2007.

[3] Scalable Routing Strategies for Ad hoc Wireless Networks, Iwata, Atsushi, Chiang, Ching-Chuan, Pei, Guangyu, Gerla, Mario, and Tsu-wei Chen, IEEE Journal on Selected Areas in Communications, August 1999.

[4] RFC 3986, Uniform Resource Identifier (URI): Generic Syntax, T. Berners-Lee, R. Fielding, L. Masinter, January 2005.

[5] "Compressed Bundle Header Encoding (CBHE)," (work in progress), Internet Draft draft-irtf-dtnrg-cbhe-03, S. Burleigh, November 12, 2009.

[6] dtn2 is found at http://sourceforge.net/projects/dtn/

[7] Space Assigned Numbers Authority – Roles, Responsibilities, Policies and Procedures. Draft CCSDS Yellow Book, Issue 0.2, CCSDS 313.0-Y-0.2, May 2009. Washington, DC.

[8] "Intentional Naming in DTN," (work in progress), Internet-Draft draft-pbasu-dtnrg-naming-00, P. Basu et al., May 22, 2009.

[9] RFC 2101, IPv4 Address Behaviour Today, B. Carpenter et al., February 1997.

[10] RFC 3467, Role of the Domain Name System (DNS), J. Klensin, February 2003.

[11] RFC 1498, On the Naming and Binding of Network Destinations, J. Saltzer August 1993.

[12] Shoch, John F., "Inter-Network Naming, Addressing, and Routing," IEEE Proc. COMPCON Fall 1978, pp. 72-79. Also in Thurber, K. (ed.), Tutorial: Distributed Processor Communication Architecture, IEEE Publ. #EHO 152-9, 1979, pp. 280-287.

11. BIOGRAPHIES

Loren Clare *is the supervisor for the Communications Networks Group at the Jet Propulsion Laboratory. He obtained the Ph.D. in System Science from the University of California, Los Angeles in 1983. His interests include wireless communications protocols, self-organizing systems, network systems design, modeling and analysis, and distributed control systems. Prior to joining JPL in May 2000, he was a senior research scientist at the Rockwell Science Center, where he acquired experience in distributed sensor networks, satellite networking, and communications protocols for realtime networks supporting industrial automation.*

Scott Burleigh *is a Principal Engineer at the Jet Propulsion Laboratory, California Institute of Technology. Scott developed one of the Laboratory's earliest systems based on Internet technology: a data distribution server that supported near-real-time analysis of science instrument data returned from the Voyager 2 encounter with the planet Neptune. He was a co-author of the specification for the Consultative Committee for Space Data Systems and File Delivery Protocol, an international standard for file transfer over interplanetary distances. A member of the Delay-Tolerant Networking (DTN) research group of the Internet Research Task Force, Scott is co-author of the specifications for both the DTN bundle protocol and also the DTN Licklider Transmission Protocol for delay-tolerant data block transmission ARQ at the data link level.*

Keith Scott *is a Principal Engineer in the MITRE Corporation's Communications Network Engineering and Analysis department. He has been involved in DTN since 1998, is co-author of the Bundle Protocol Specification, and chairs the CCSDS Space Internetworking Services Delay Tolerant Networking working group. His research interests include routing in delay tolerant networks and applications of DTN technologies to challenged networks such as tactical networks and MANETs.*