Delay-Tolerant Networking: An Approach to Interplanetary Internet

Scott Burleigh, Adrian Hooke, and Leigh Torgerson, Jet Propulsion Laboratory
Kevin Fall, Intel Corporation
Vint Cerf, MCI
Bob Durst and Keith Scott, The MITRE Corporation
Howard Weiss, SPARTA, Inc.

ABSTRACT

Increasingly, network applications must communicate with counterparts across disparate networking environments characterized by significantly different sets of physical and operational constraints; wide variations in transmission latency are particularly troublesome. The proposed Interplanetary Internet [1], which must encompass both terrestrial and interplanetary links, is an extreme case. An architecture based on a "least common denominator" protocol that can operate successfully and (where required) reliably in multiple disparate environments would simplify the development and deployment of such applications. The Internet protocols are ill suited for this purpose. We identify three fundamental principles that would underlie a *delay-tolerant networking* (DTN) architecture and describe the main structural elements of that architecture, centered on a new end-to-end overlay network protocol called *Bundling*. We also examine Internet infrastructure adaptations that might yield comparable performance but conclude that the simplicity of the DTN architecture promises easier deployment and extension.

INTRODUCTION

Consider a scientist who is responsible for the operation of a robotic meteorological station located on the planet Mars (Fig. 1). The weather station is one of several dozen instrument platforms that communicate among themselves via a wireless local area network deployed on the Martian surface. The scientist wants to upgrade the software in the weather station's data management computer by installing and dynamically loading a large new module. The module must be transmitted first from the scientist's workstation to a deep space antenna complex, then from the antenna complex to a constellation of relay satellites in low Mars orbit (no one of which is visible from Earth long enough on any single orbit to receive the entire module), and finally from the relay satellites to the weather station.

The first leg of this journey would typically be completed using the TCP/IP protocol suite over the Internet, where electronic communication is generally characterized by:

- Relatively small signal propagation latencies (on the order of milliseconds)
- Relatively high data rates (up to 40 Gb/s for OC-768 service)
- Bidirectional communication on each connection
- Continuous end-to-end connectivity
- On-demand network access with high potential for congestion

However, for the second leg a different protocol stack would be necessary. Electronic communication between a tracking station and a robotic spacecraft in deep space is generally characterized by:

- Very large signal propagation latencies (on the order of minutes; Fig. 2)
- Relatively low data rates (typically 8–256 kb/s)
- Possibly time-disjoint periods of reception and transmission, due to orbital mechanics and/or spacecraft operational policy
- Intermittent scheduled connectivity
- Centrally managed access to the communication channel with essentially no potential for congestion

The combination of long signal propagation times and intermittent connectivity — caused, for example, by the interposition of a planetary body between the sender and the receiver — can result in round-trip communication delays measured not in milliseconds or even minutes but in hours or days. The Internet protocols do not behave well under these conditions, for reasons discussed later in this article.

Yet a retransmission protocol of some sort is required to assure that every bit of the new executable module is correctly received. Forward error correction (FEC) can reduce data loss and

The research described in this article was performed under DOD Contract DAA-B07-00-CC201, DARPA AO H912; JPL Task Plan no. 80-5045, DARPA AO H870; and (at the Jet Propulsion Laboratory, California Institute of Technology) NASA Contract NAS7-1407.

0163-6804/03/$17.00 © 2003 IEEE

corruption, but it consumes bandwidth whether data are lost or not, and it offers no protection against sustained outage. Optimum utilization of meager links demands automated repeat request (ARQ) in addition to some level of FEC. What is needed on this part of the route is an ARQ system for efficient retransmission on the link.

Recent developments in deep space communications technology have begun to address this problem. Over the past 20 years the Consultative Committee for Space Data Systems (CCSDS) [2] has established a wide range of standards for deep space communications, including Telecommand and Telemetry wireless link protocols for spacecraft operations. A recent addition to this program is the *CCSDS File Delivery Protocol* (CFDP) [3], which is designed for reliable file transfer across interplanetary distances. (CFDP is discussed in more detail below.) The "CFDP-RP" link ARQ system in Fig. 1 is a hypothetical protocol that would be constructed by, in essence, implementing just the data retransmission procedures specified for CFDP. (Note that although CFDP implementations exist, the CFDP-RP standalone subset has not yet been isolated for the purpose proposed here.)

For the final delivery of the module from the relay orbiters to the weather station on its wireless LAN, TCP/IP might again be the best choice. But now TCP/IP would be running over wireless link protocols, perhaps CCSDS Proximity-1 [4] to the satellites and 802.11b among the landed assets. As in interplanetary space, and in contrast to the wired Internet, data rates on these links are likely to be fairly low,[1] and the potential for congestion will be low for the foreseeable future.

Since no single stack will perform satisfactorily on all segments of the route, no single protocol immediately below the application layer is suitable for end-to-end use in this scenario. How then can the application operate?

This is not an altogether new problem. The three different sets of physical and operational constraints described above define very different networking environments. Protocols that enable effective communication within each of these environments have been developed and are continually being improved, and techniques already exist for forwarding data between environments that differ in less radical ways. For example, IP routers typically convey traffic between adjacent subnets running at different data rates; transport-level proxies can bridge between TCP connections tuned to different data loss profiles and relatively small differences in signal propagation time [5]. But for large differences in the scale of round-trip latency, wholly different transport mechanisms are needed.

An Overview of CFDP

One approach to reliable transport that can tolerate extremely long and variable round-trip latency is reflected in the design of CFDP (introduced above). CFDP can operate in either acknowledged (reliable) or unacknowledged mode; in acknowledged mode, lost or corrupted data are automatically retransmitted. CFDP's design includes a number of measures adopted to enable robust operation of its ARQ system in high-latency environments:

- Because the time required to establish a connection might exceed the duration of a communication opportunity, there is no connection protocol; communication parameters are managed.
- Because round-trip latency may far exceed the time required to transmit a given file, CFDP never waits for acknowledgment of one transmission before beginning another. Therefore, the retransmitted data for one file may arrive long after the originally transmitted data for a subsequently issued file, so CFDP must attach a common transaction identifier to all messages pertaining to a given file transmission.
- Because a large number of file transmissions may concurrently be in various stages of transmission, retransmission buffers typically must be retained in nonvolatile storage; this can help prevent catastrophic communications failure in the event of an unplanned power cycle at either the sender or the receiver.

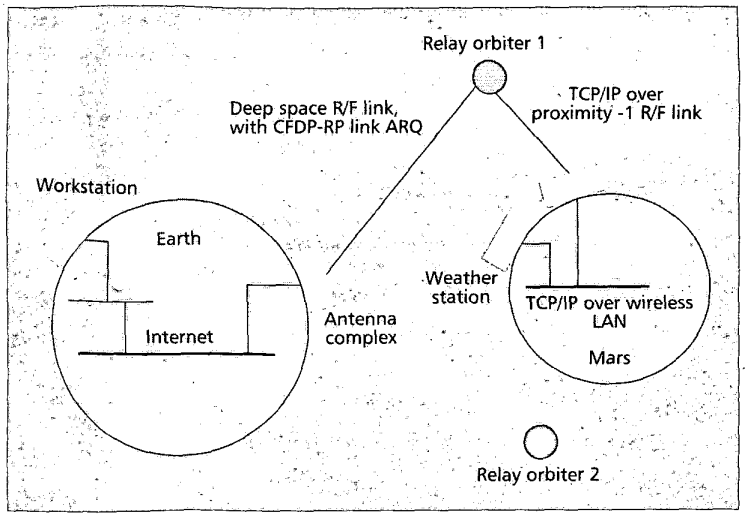

Figure 1. *A high-delay operational scenario.*

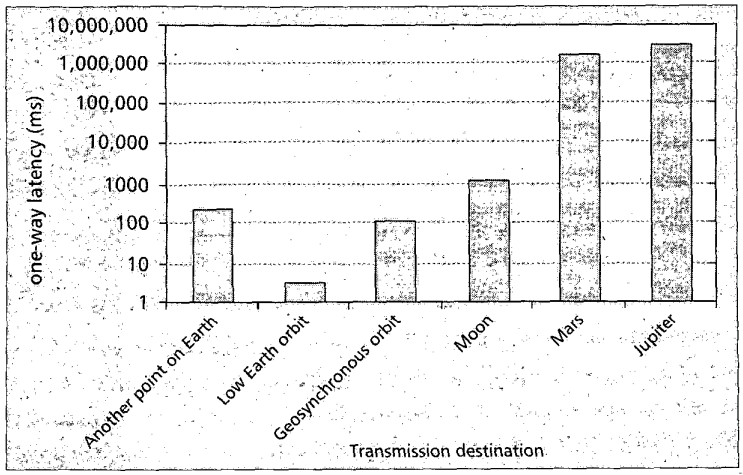

Figure 2. *Maximum latency (transmit from Earth).*

[1] *This is largely due to the difficulty of providing abundant electrical power.*

Many applications operate properly only if data transmission is reliable, i.e., if there is assurance that every item of data issued is successfully delivered to its destination (barring catastrophic infrastructure failure that requires human intervention).

WHY NOT THE INTERNET PROTOCOLS?

The Internet protocols are in general poorly suited to operation on paths in which some of the links operate intermittently or over extremely long propagation delays. The principal problem is reliable transport, but the operations of the Internet's routing protocols would also raise troubling issues. While those issues don't seem insoluble, solutions would entail significant divergence from typical operations in the Internet.

RELIABLE TRANSPORT

Many applications operate properly only if data transmission is reliable; that is, if there is assurance that every item of data issued is successfully delivered to its destination (barring catastrophic infrastructure failure that requires human intervention). As noted earlier, an ARQ system of some sort is needed for this purpose.

The two broadly supported transport layer protocol options offered by the Internet protocol suite are TCP and UDP, both operating over IP. TCP performs ARQ, but it is ill suited for operation over a path characterized by very long signal propagation latency, particularly if the path contains intermittent links:

- TCP communication requires that the sender and receiver negotiate a connection that will regulate the flow of data. Establishment of a TCP connection typically entails at least one round-trip (transmission and response) before any application data can flow. If transmission latency exceeds the duration of the communication opportunity, no application data will flow at all.
- TCP delivers received data to the application only in transmission order. This means that any data loss requiring retransmission will retard the delivery of all data subsequently transmitted on the same connection until the lost data have been retransmitted successfully — at least one round-trip. To avoid this blockage, the sending application's only option is to incur the transmission cost of opening additional parallel connections and distributing transmission across those connections.
- The throughput of TCP itself diminishes with increasing round-trip latency due to the manner in which TCP responds to data loss and handles network congestion [6].

Operating TCP end to end over a path comprising multiple links, some of which may be "long" or intermittent, presents additional difficulties. TCP retransmission is end to end, and end-to-end retransmission delays the release of retransmission buffer space. For example, consider a three-hop route, ABCD, where the one-way signal propagation latency is 500 ms on the AB hop, 8 min (480,000 ms) on the BC hop, and 100 ms on the CD hop. (Imagine A is a Mars rover, B is a Mars orbiter, C is a tracking station on Earth, and D is a scientist's workstation somewhere on the Internet.) Retransmission is possible only if the original sender of a message retains a copy until it is confident that retransmission will not be necessary (e.g., the destination notifies it that the message has arrived). If retransmission is hop by hop — that is, performed between the endpoints of each link individually — A can release its retransmission buffer space after about 1000 ms: 500 ms for the transmission from A to B, then 500 ms for the acknowledgment from B to A. If retransmission is end to end, A's retransmission buffer must be retained for 961,200 ms (AB, BC, CD, DC, CB, BA). Data loss would further delay the release of retransmission buffer space by at least one round-trip, consumed by the retransmission request and the retransmission of the message; loss of either the request or the response would, of course, make matters even worse.

This in turn means that the *amount* of retransmission buffer space required at the endpoints of the route is increased by end-to-end retransmission. For optimum link utilization, the links should at all times be carrying as much data as they can. Assume that the maximum data rate on each link is 1 Mb/s. In the worst case, where A is the source of all traffic, links are underutilized if A is not issuing data at 1 Mb/s. Suppose A is issuing one 256 kb low-resolution camera image every 250 ms. At that rate, assuming no data loss, A's aggregate retransmission buffers will consume up to 1 Mb of memory if retransmission is hop by hop: after 1000 ms following the start of operations, the acknowledgments from B will be causing the release of space at the same rate that original transmissions from A are consuming it (256 kb every 0.25 s, as the images are issued, received, and acknowledged). But if retransmission is end to end, acknowledgments from D won't start releasing A's retransmission buffers until 961,200 ms following the start of operations; A's retransmission buffers will consume over 961 Mb of memory.

This effect becomes increasingly significant with increasing transmission latency on any subset of the links, and for remote robotic communication assets operating in power-constrained environments it is highly undesirable. It increases the amount of storage required for a given level of performance and thus increases demand for power, both of which increase cost.

UDP-based approaches are also unsatisfactory. The absence of ARQ in UDP means that all responsibility for data acknowledgment and retransmission is left to the layer above UDP, either the application or some standard "middleware" system (e.g., RPC, RMI, RTP) the application uses. Reinventing retransmission in applications is costly. Standardizing retransmission procedures in reusable middleware is more economical, but end-to-end retransmission in such middleware would be no more satisfactory than TCP's end-to-end retransmission, for the same reasons.

ROUTING PROTOCOLS

The Internet routing system enables routers to choose the best paths for packet forwarding. This system is implemented as a hierarchy to improve its scalability. At the top level of the hierarchy, path selection is resolved by the bor-

der gateway protocol (BGP) operating between IP address aggregates grouped into autonomous systems (ASs). Within an AS, such other routing protocols as Open Shortest Path First (OSPF), the International Organization for Standardization's (ISO's) Intermediate System to Intermediate System (IS-IS), or Cisco Systems' EIGRP are used. These protocols select paths in a changing topology where more than one path may be available at a given time. For this purpose, they require timely updates from agents at various locations in the network. Most have timeouts; if they do not receive routing messages from agents at regular intervals, they assume loss of connectivity to those agents. They also assume that the network is not partitioned: losses of connectivity are assumed to be structural rather than operational and temporary, and no network elements to which there is currently no direct or indirect connectivity will be included in any computed path.

BGP is built on TCP, so its performance in high-delay environments is limited by the TCP operational issues discussed above; BGP performs poorly when TCP is unable to keep a connection established [7]. Moreover, the distributed route computation algorithms themselves may be adversely affected by inaccurate timeout interval estimates. Premature timeouts lead to false negative conclusions about network connectivity, while tardy timeouts delay the detection of connectivity losses and may thus result in unsuccessful routing decisions.

A more serious problem is posed by the transient partitioning of networks in which long delays are caused by *scheduled* (intermittent) connectivity, where network links are created or removed in a predictable way. Since at any single moment there may currently be no direct or indirect connectivity to the destination at all — even though planned connectivity episodes may address those lapses (while perhaps introducing new ones) in a predictable way in the future — normal IP route computation may in some cases be impossible.

DELAY-TOLERANT NETWORK ARCHITECTURE

It is this analysis that leads us to propose an architecture based on Internet-independent middleware: use exactly those protocols at all layers that are best suited to operation within each environment, but insert a new overlay network protocol between the applications and the locally optimized stacks. The overlay protocol serves to bridge between different stacks at the boundaries between environments in a standard manner, in effect providing a general-purpose application-level gateway infrastructure that can be used by any number of applications. By exploiting the ARQ capabilities of the local protocols as discussed later, the overlay protocol can offer applications an end-to-end data transmission service that is both reliable and efficient.

In order to be generally useful, the overlay network protocol that is exposed to applications must be able to ensure reliable transmission between application counterparts separated by an arbitrary number of changes in environmental character; communications must traverse an arbitrary sequence of environments imposing sharply different sets of operating constraints. The protocol must therefore be a "least common denominator" with no mandatory elements that make it inherently unsuitable in any networking environment.

In particular, the design of the overlay protocol must not be based on any end-to-end expectation of:

- Continuous connectivity
- Low or constant transmission latency
- Low error rate
- Low congestion
- High transmission rate
- Symmetrical data rates
- Common name or address expression syntax or semantics
- Data arrival in transmission order

Yet for optimum end-to-end performance we want to be able to take advantage of any of these favorable circumstances that are present, wherever they are present.

Working from these considerations, we have identified three fundamental principles of delay-tolerant networking (DTN) architecture:

- **A postal model of communications**. Because transmission latency can be arbitrarily long, reliance on negotiation, query/response, or any other sort of timely conversational interchange is inadvisable, in both the overlay network protocol and the applications themselves. Insofar as it is possible, the data transmitted through the network should constitute self-contained atomic units of work. Applications should issue messages asynchronously, not wait for the response to one message before sending the next.

For example, a delay-tolerant request for transmission of a file would not initiate a dialog as in FTP. It would instead bundle together into a single message not only the name of the requested file, but also — unprompted — all other metadata that might be needed in order to satisfy the request: the requesting user's name and password, encoding instructions, and so on.

In recognition of this general model, the units of data exchanged via the DTN overlay network protocol are termed *bundles* (which are functionally similar to email messages); the protocol itself is named *Bundling*.

- **Tiered functionality**. The protocols designed for use within various environments already exploit whatever favorable circumstances the environments offer while operating within their constraints, so the DTN architecture relies on the capabilities of those protocols to the greatest extent possible. The Bundling protocol, one layer higher in the stack, performs any required additional functions that the local protocols typically cannot.
- **Terseness**. Bandwidth cannot be assumed to be cheap, so the DTN protocols are designed to be taciturn even at the cost of some processing complexity.

The main structural elements of DTN architecture, derived from these principles, are as follows.

The protocols designed for use within various environments already exploit whatever favorable circumstances the environments offer while operating within their constraints, so the DTN architecture relies on the capabilities of those protocols to the greatest extent possible.

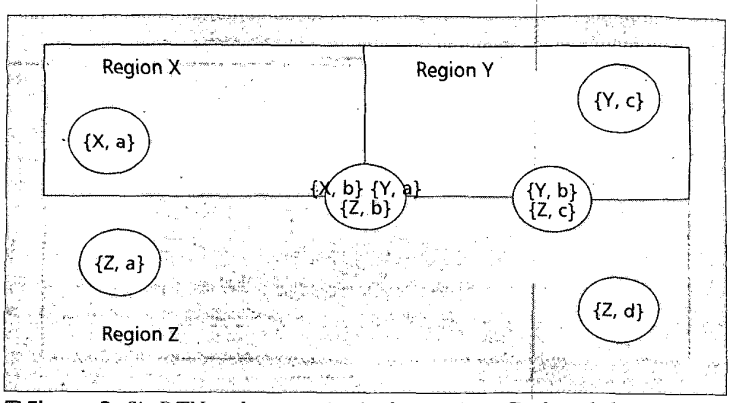

Figure 3. *Six DTN nodes operating in three regions. Each node has one interface for each region within which it operates; each interface is identified by a tuple expression comprising the applicable region ID and the interface's unique identifier within that region. For example, the node at the center of the figure is a gateway that can forward bundles between any two of the regions shown: its interface in region X is {X, b}, its interface in region Y is {Y, a}, and its interface within region Z is {Z, b}.*

Tiered Forwarding — The communication assets on which Bundling protocol engines run (analogous to the hosts and routers in an IP-based network) are termed *DTN nodes*. A *DTN region* is informally defined as a set of DTN nodes that can communicate among themselves using a single common protocol family that is suitable for the networking environment in which all of the nodes must operate.

The DTN architecture generally relies on regional network-layer protocols, such as IP in Internet-like regions, for the forwarding of bundles among DTN nodes within each regional network. The forwarding of bundles among DTN nodes that are in different regions is performed by Bundling. *Gateway* nodes straddling the boundaries between regions are critical to this function (Fig. 3): a gateway node has an interface in each of the adjacent regions (i.e., it can communicate using both regions' protocols) and can therefore convey a bundle between the regions by reading on one interface and writing on the other.

Bundling's store-and-forward operation must differ from that of other network protocols in that outbound data may need to be stored, not for milliseconds in dynamic memory, but for hours or days in nonvolatile media. This *deferred transmission* may be unavoidable because continuous link connectivity cannot be assumed: the link on which an outbound bundle must be transmitted may not be established until some time in the future, depending on node mobility (e.g., in mobile ad hoc networks), power management (e.g., in sensor networks), orbital dynamics (in deep space), and so on.

Tiered Naming and Addressing — In order for a bundle to reach its destination within a given region, it must be tagged with a destination identifier that enables it to be forwarded by applicable regional protocols to the appropriate destination DTN node. That is, the destination identifier of a bundle must map in some way to an address (or equivalent) in that region's address space (or equivalent).

But in order for that bundle to be handed to the applicable regional protocols for delivery, it has to reach the region in which the destination DTN node resides. For this purpose, an additional addressing element is required: the name (or other ID) of the destination region itself, which is used for forwarding at the Bundling layer.

So the source and destination expressions of bundles must be concatenated identifiers, termed *tuples*, comprising both region identifiers and also regional destination identifiers that can be mapped to regional addresses (or equivalent):

{region ID, regional destination identifier}

Regional destination identifiers are *late bound*; that is, they are mapped to regional addresses (or equivalent) only upon arrival at the destination region, rather than at the time of original transmission. This has two advantages for DTN nodes that are the sources of bundles:

- The nodes need not understand all possible regional identifier systems in order to issue bundles. Since the forwarding protocols in different regions may be different, it is possible that destination identifier syntax and mapping algorithms may also vary by region.[2] Late binding enables new regions with new naming and addressing systems to be added without impact on previously deployed nodes.
- Where identifier mapping operations rely on querying servers (e.g., the Internet's Domain Name System), the issuance of a bundle is not delayed by the time needed to complete a mapping query.

Tiered Routing — The network protocols operating within regional networks are already supported by the routing protocols designed for those regions. The forwarding performed by Bundling must be supported by new routing protocols.

In particular, route computation at the Bundling layer must be sensitive to future link establishment opportunities, or *contacts*. Contacts may be anticipated in a variety of ways:

- They may be scheduled by explicit network management, either manual or automated.
- They may be discoverable in real time within regions in which signal propagation delays are small.
- They may be predictable based on region-specific structural awareness, such as knowledge of mobility patterns or orbital dynamics.
- They may be computed stochastically based on prior contact history.

Different anticipated contacts may be characterized by different data rates or other transmission constraints.

Tiered ARQ — The DTN architecture depends on regional transport protocols such as TCP, or reliable link protocols such as CFDP-RP, for assured transmission of bundles among DTN nodes within each regional network. Efficient ARQ relies on the accurate computation of timeout intervals: premature retransmission wastes bandwidth, but waiting too long to

[2] *In sensor networks, for example, an attribute-based naming scheme may serve the same purposes as node addresses [8].*

retransmit degrades throughput and results in excessive allocation of storage to retransmission buffers. Because the algorithms for those computations may be radically different in different regions, the concatenation of reliable transmissions within adjacent regions is the most efficient mechanism for achieving reliability end to end in DTN.

However, this mechanism may not be sufficient. Large round-trip transmission latencies within a region, whether due to long signal propagation times or interruptions in connectivity, may result in aggregate retransmission buffer sizes at a given node that exhaust available resources. When retransmission buffers must be prematurely released, or a retransmitting node simply crashes, reliable regional transmission fails. To guard against such failures, the DTN architecture identifies an additional ARQ mechanism that may be implemented at the Bundling level: a node that explicitly "takes custody" of a bundle guarantees that it can and will devote sufficient resources to retain a copy of the bundle until some downstream node subsequently takes custody of it. This enables *custodial retransmission* in the event that no such notice of custody transfer arrives.

This retransmission device can be viewed as a "safety net" that should rarely reissue bundles. The timeout intervals it operates on must be worst-case estimates to prevent costly unnecessary retransmission, so it cannot be efficient enough to supplant the regional ARQ systems.[3]

Tiered Security — One implication of the DTN principle of terseness is that performance degradation due to unauthorized consumption of DTN resources (transmission bandwidth, storage, and processing cycles) must be minimized. For this purpose, the exchange of bundles between adjacent nodes may be subject to verification of cryptographic credentials wherever this is deemed necessary by network administrators. This *mutual suspicion* model cannot prevent the introduction of unauthorized traffic into the network, but by suppressing propagation of that traffic we can at least contain its impact.

Above Bundling, applications may require user data authentication, integrity, and confidentiality services. However, any such services that rely on key management techniques based on querying key servers or negotiating shared keys will not be efficient over long-latency or intermittent links. Since we will need to solve this problem in Bundling to implement mutual suspicion, this DTN solution might be offered in support of these application services as well. One possibility, inspired by S/MIME and PGP, would be to send a certificate containing cryptographic key material with each bundle. This technique might violate our terseness principle, though, since certificates range in size from 450 bytes to 8 kbytes.

Tiered Congestion Control — Congestion avoidance and control measures as needed are generally built into regions' existing communications infrastructures. Within the Internet, carefully engineered congestion avoidance is one of the key features of TCP. In admission-controlled environments, on the other hand, congestion is a management problem rather than an issue of protocol: access to links is scheduled and controlled, so competition for link access is resolved by reservation rather than contention. In deep space operations, for example, it takes place during operations planning rather than in real time.

DTN architecture relies on the effectiveness of these regional measures. It remains to be seen whether or not the control of congestion within each region individually has the effect of minimizing congestion across all regions collectively. If not, an additional congestion control mechanism will be needed at the Bundling layer.

Resilient Delivery — The ultimate source and destination of a bundle are providers or consumers of services (*service agents*), typically — but not always — taking the form of processes, tasks, or threads. At the extreme, end-to-end transmission latency for a bundle in a delay-tolerant network might be so long that the destination service agent is no longer running at the time the bundle arrives, or the source service agent is no longer running at the time a reply arrives. DTN nodes must therefore be equipped for not only deferred transmission but also *deferred delivery*: the final destination node may need to retain a given bundle in local nonvolatile storage until such time as its destination service agent starts (or restarts) and announces its readiness to receive data. It may even be necessary for Bundling to take responsibility for *reanimating* destination service agents — invoking operating system services itself to start or restart them, possibly with some state information — so that inbound bundles can be successfully delivered.

Postal Service Levels — Guided by the principle of postal communications, we look to literal postal operations for ideas on the different qualities of service DTN applications might find useful. Classes of service offered by the U.S. Postal Service have evolved over hundreds of years to meet the needs of millions of users exchanging information in a non-conversational manner. We propose to offer DTN applications a simplified subset of those services:

- Three levels of delivery priority: low, standard, and high
- Three postal service notifications, all of which can optionally be sent to a specified "reply-to" service agent rather than to the original sender
  - ¶Notice of initial transmission (i.e., notice of mailing)
  - ¶Notice of delivery to the ultimate destination application (i.e., return receipt)
  - ¶Report of route taken (i.e., delivery record)

Result — The architecture that results from the integration of all these structural elements is highly adaptable and extensible, yet simple. The flow of data between a scientist and a spacecraft using a Bundling-based "Interplanetary Internet," for example, might look something like Fig. 4.

This retransmission device can be viewed as a "safety net" that should rarely reissue bundles. The timeout intervals it operates on must be worst-case estimates to prevent costly unnecessary retransmission, so it cannot be efficient enough to supplant the regional ARQ systems.

[3] We recognize that this feature of the DTN architecture might seem to contradict the end-to-end principle that has served the Internet well for many years. We would argue, though, that in fact it conforms perfectly to the end-to-end argument as originally articulated by Saltzer, Reed, and Clark [9]: regional ARQ is indeed a "low-level mechanism" that can be justified only as a performance enhancement, but in the case of DTN this performance enhancement may in practice be indispensable.

Other Bundling functions could be performed at the link layer, either by the new virtual interfaces or by a new "reliable link" infrastructure (RLI) whose capabilities would be provided to them.

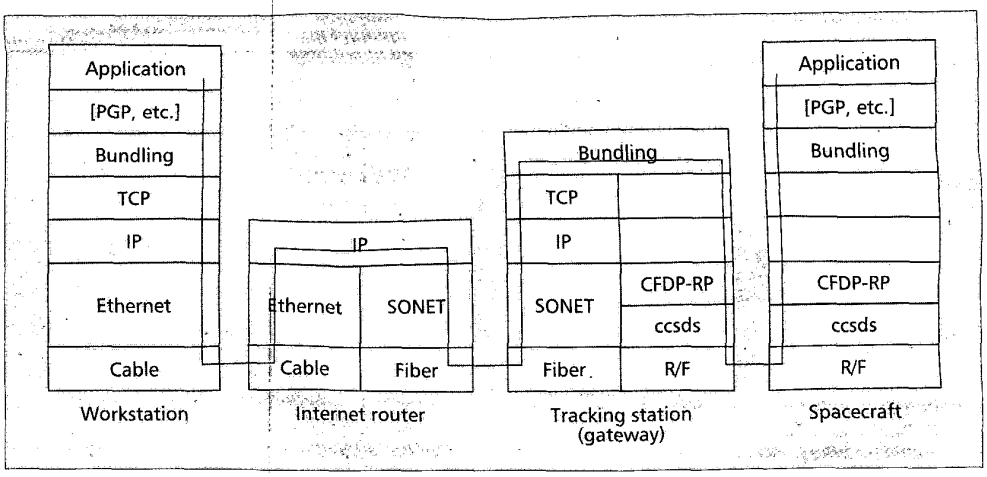

Figure 4. *An example of data flow in a Bundling-based Interplanetary Internet.*

DTN WITHOUT BUNDLING

A possible objection to this architecture is that it departs from the Internet model, which is defined by the end-to-end use of IP rather than Bundling. In this section we consider the development of supporting infrastructure that would enable UDP/IP to function in much the same way as Bundling. Such infrastructure would enable the deployment of DTN built on familiar Internet capabilities with no protocol modification.

We recall that IP itself is an overlay network protocol that mediates between different link layer protocols. Suppose one built a "reliable link" system that used TCP/IP *tunnels*, and suppose one then built IP virtual interfaces to this TCP/IP reliable link tunnel (RLT) system, and also to CFDP-RP and other systems that we have called regional protocols. This would give IP the end-to-end reliability over heterogeneous links that characterizes Bundling (Fig. 5).

Tiered naming and addressing, including the late binding of names to addresses at the destination (rather than source) router, is possibly the most challenging Bundling capability to replicate within the Internet model without protocol modification. The approach considered here is to carry regional destination identifiers as URLs in HTTP 1.1 layered on top of UDP/IP. If DTN gateway nodes' IP addresses are kept relatively stable so that they can be, in effect, used as region identifiers, URL resolution at the HTTP layer of the destination region's gateway node can determine the IP address of the final destination; the HTTP service can then accomplish the final intraregional hop of the end-to-end route.

Other Bundling functions could be performed at the link layer, by either the new virtual interfaces or a new reliable link infrastructure (RLI) whose capabilities would be provided to them. RLI capabilities would include:

- Management of nonvolatile storage, which would enable deferred transmission by the virtual interfaces and thus *tiered forwarding*.
- Custodial retransmission, giving us *tiered ARQ*.
- Mutual suspicion functions, yielding *tiered security*.
- Support for deferred delivery and service agent reanimation by virtual interfaces, yielding *resilient delivery*.
- Postal service notifications, informed by RLI's custody awareness. These notifications, together with the proposed differentiated services capabilities [10] of the Internet, would give us *postal service levels*.

The scope of delay tolerance in the Internet would grow as the new virtual interfaces and supporting RLI were added to hosts and routers.

CONCLUSION

Part of the appeal of the non-Bundling DTN approach would be its familiarity to application developers. Delay-tolerant applications would still need to be engineered with DTN architectural principles borne in mind, but at least the interface to the DTN technology would be one that has been used to implement any number of Internet applications over the past few decades.

However, we view the apportioning of Bundling functions among an array of new virtual links, a new link layer reliability infrastructure, and a variety of addressing and HTTP service expedients as a diffuse, fragile, and costly solution to the problem of delay tolerance in networking. By instead encompassing all this new capability in a single application-layer Bundling service, we are able to develop, debug, and exercise the technology without impact on the lower layers of existing hosts and routers. Porting to different platforms is relatively easy, often little more than a matter of recompilation. As a result, we can fairly rapidly and inexpensively configure large and complex DTN networks for our research. In short, the simplicity of the current Bundling architecture appears to have practical benefits as well as offer the prospect of easier expansion and extension.

Work on the architecture for the Interplanetary Internet, which has since been generalized to DTN architecture, began in early 1998. A new Research Group for Delay Tolerant Networking was chartered within the Internet Research Task

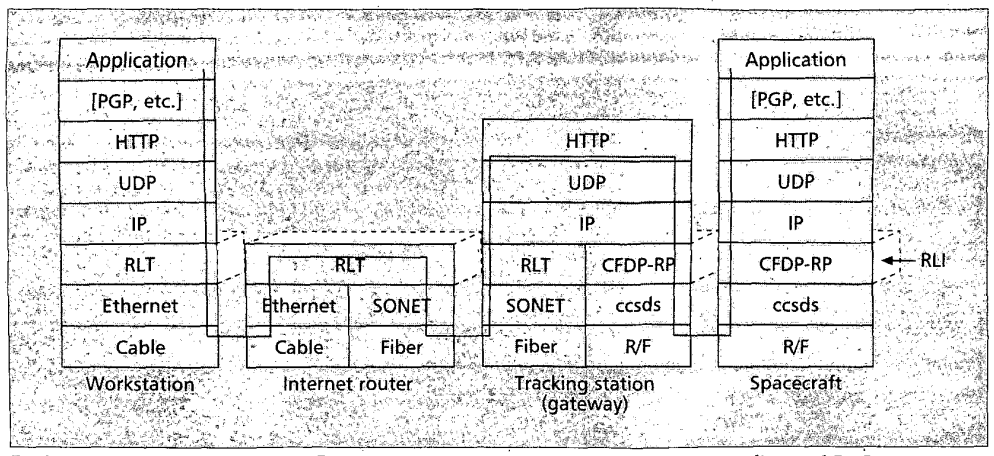

Figure 5. *An example of data flow in an Interplanetary Internet based on tunneling and RLI.*

Porting to different platforms is relatively easy, often little more than a matter of recompilation. As a result, we can fairly rapidly and inexpensively configure large and complex DTN networks for our research.

Force in October 2002. Two editions of an Internet draft describing our architecture (in far more detail than this article) have been offered for peer review, and an initial specification for the Bundling protocol has been drafted.

In addition, prototype implementations of Bundling and supporting software have been developed over the past year. The end-to-end prototype system has been demonstrated to be tolerant of system reboots and simulated regional connectivity lapses lasting up to 60 min, and we know of no structural obstacle to tolerating much longer interruptions. We plan to make the source code for these implementations publicly available as open source.

References

[1] http://www.ipnsig.org/reports/memo-ipnrg-arch-00.pdf
[2] CCSDS, http://www.ccsds.org
[3] CCSDS File Delivery Protocol (CFDP), CCSDS 727.0-B-1, Jan. 2002; http://www.ccsds.org/documents/pdf/CCSDS-727.0-B-1.pdf
[4] Proximity-1 Space Link Protocol (Prox-1), CCSDS 211.0-R-3.2, Red Book, Sept. 2002; daft doc. available: http://www.ccsds.org
[5] M. J. Zukoski and R. Ramirez, "A Transport Protocol for Space Communications," *The Edge*, The MITRE Corp., vol. 2, no. 3, Nov. 1998, http://www.mitre.org/pubs/edge/november_98/11_98.pd, pp. 1–4.
[6] J. Padhye *et al.*, "Modeling TCP Throughput: A Simple Model and its Empirical Validation," *ACM SIGCOMM '98 Conf. Apps., Tech., Architectures, and Protocols for Comp. Commun.*, 1998.
[7] A. Shaikh *et al.*, "Routing Stability in Congested Networks: Experimentation and Analysis," *Proc. SIGCOMM 2000.*
[8] J. Heidemann *et al.*, "Building Efficient Wireless Sensor Networks with Low-Level Naming," *Proc. Symp. Op. Sys. Principles*, Oct. 2001, pp. 146–59.
[9] J. H. Saltzer, D. P. Reed, and D. D. Clark, "End-to-End Arguments in System Design," *ACM Trans. Comp. Sys. 2.4*, Nov. 1984, pp. 277–88.
[10] S. Blake *et al.*, "An Architecture for Differentiated Services," Internet RFC 2475, Dec.1998.

Biographies

SCOTT BURLEIGH (Scott.Burleigh@jpl.nasa.gov) has 29 years of experience in software development, of which the past 17 have been spent at NASA's Jet Propulsion Laboratory. In 1988–1989 he 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 has subsequently worked on high-speed telemetry data acquisition and management systems, optimization of machine learning algorithms, distributed real-time spacecraft simulation, an embeddable object persistence system, software fault tolerance, and sensor webs. He is international rapporteur for the CCSDS File Delivery Protocol (CFDP) specification. He was also the developer of the first implementation of CFDP. He is past Chair of the Interplanetary Internet Special Interest Group (IPNSIG) of the Internet Society and is a member of the Delay-Tolerant Networking Research Group of the Internet Research Task Force. He holds a B.A. in english literature from the State University of New York at Albany.

KEVIN FALL is a senior research scientist at Intel Corporation's Berkeley Research Center. Prior to Intel, he was chief network architect for NetBoost Corporation, an adjunct professor at the University of California, Berkeley (UC Berkeley), and a staff scientist with Lawrence Berkeley National Laboratory's Network Research Group. His research interests include network protocol architecture, operating systems, and performance analysis. He received M.S. and Ph.D. degrees in computer science from the University of California, San Diego in 1991 and 1994, respectively, and earned a B.A. degree in computer science from UC Berkeley in 1988.

VINTON G. CERF [F] is senior vice president of architecture and technology for MCI. He is the co-designer of the TCP/IP protocols and the architecture of the Internet. In December 1997, President Clinton presented the U.S. National Medal of Technology to Cerf and his partner, Robert E. Kahn, for founding and developing the Internet. During his tenure from 1976 to 1982 with the U.S. Department of Defense's Advanced Research Projects Agency (DARPA), he played a key role in leading the development of Internet and Internet-related data packet and security technologies. He serves as chairman of the board of the Internet Corporation for Assigned Names and Numbers (ICANN). He served as founding president of the Internet Society from 1992 to 1995 and is a Fellow of the ACM, the American Association for the Advancement of Science, the American Academy of Arts and Sciences, the International Engineering Consortium, the Computer History Museum, and the National Academy of Engineering. He is a recipient of numerous awards and commendations in connection with his work on the Internet. These include the Marconi Fellowship, the Charles Stark Draper award of the National Academy of Engineering, the Prince of Asturias award for science and technology, the Alexander Graham Bell Award presented by the Alexander Graham Bell Association for the Deaf, the Silver Medal of the International Telecommunications Union, and the Library of Congress Bicentennial Living Legend medal. He holds a Bachelor of Science degree in mathematics from Stanford University, and Master of Science and Ph.D. degrees in computer science from the University of California at Los Angeles (UCLA).

ROBERT C. DURST is a senior principal engineer in the Networking and Communications Engineering Department of MITRE Corporation's Information Systems and Technology Division. He is responsible for MITRE's support of NASA's Jet Propulsion Laboratory, developing communication protocols for providing Internet access to near-Earth spacecraft and deep space assets. He is an active member of the

The end-to-end prototype system has been demonstrated to be tolerant of system reboots and of simulated regional connectivity lapses lasting up to 60 minutes, and we know of no structural obstacle to tolerating much longer interruptions.

Interplanetary Internet development effort at JPL and the Internet Research Task Force's Delay-Tolerant Networking Research Group, and is leading an effort to generalize their architecture to broader classes of delay-tolerant communication in extreme networking environments. He is the principal investigator for MITRE's Mobile Ad Hoc Networks on the Transformed Army research project, has led MITRE's support to various DARPA research programs, and is actively developing routing protocols for Army satellite networking. His research interests include design and evaluation of distributed computer communication systems and protocols, networking in intermittently connected environments, multicast data transmission, congestion and error control in mobile ad hoc networks, and satellite communications. He received a Bachelor of Science degree, cum laude, in electrical engineering from the University of Missouri in 1978.

ADRIAN J. HOOKE is a principal at the NASA/Caltech Jet Propulsion Laboratory in Pasadena, California. His 37 years of continuous space mission experience began in industry in the mid-1960s when he was intimately involved with launching the Apollo Lunar Modules at the Kennedy Space Center. In December 1969 he joined JPL and was a member of the flight control teams for the Mariner 9 and 10 missions to Mars, Venus, and Mercury. After working on the definition of the Voyager onboard data system and leading the design of the SEASAT end-to-end data system, he joined the staff of the European Space Agency to work on the flight operations architecture for the Shuttle/Space-Lab program. He rejoined JPL in 1977 to focus on the development of new technology in the area of standardized space data communications protocols. He is one of the founders of the international Consultative Committee for Space Data Systems (CCSDS) and has led the development of international standards for Packet Telemetry, Packet Telecommand, and the extension of the terrestrial Internet protocol suite into space. He is currently the chairman of the CCSDS Engineering Steering Group as well as the U.S. representative to the ISO committee dealing with space data systems. Since 1998 he has managed the Interplanetary Internet project at JPL. He holds a B.Sc. (Honors) in electronic and electrical engineering from the University of Birmingham, England. He is a registered Chartered Engineer (C. Eng.) with the IEE in London and a European Engineer (Eur. Ing.) with FEANI in Paris. He has been awarded two NASA Exceptional Service Medals.

KEITH SCOTT is a principal engineer in the MITRE Corporation's Center for Innovative Computing and Informatics. Before joining MITRE, he worked for NASA's Jet Propulsion Laboratory in Pasadena, California, on TCP performance over mobile satellite channels, Mars relay communications, and interspacecraft communication for constellation missions. He is currently the task leader for MITRE's Next Generation Space Internet work applying advanced Internet technologies to spacecraft communications. His research interests include self-organizing networks, congestion control in mixed-loss environments, Internet QoS, application of Internet technologies to space communication, and delay-tolerant networking. He received a B.S. in electrical engineering and computer science from UC Berkeley, and M.S. and Ph.D. degrees in electrical engineering from UCLA.

LEIGH TORGERSON has over 30 years of experience in aerospace systems engineering. After eight years of service as a naval aviator, he joined the Lockheed Skunk Works in 1980, and worked on reconnaissance avionics systems for 10 years. Joining CalTech's Jet Propulsion Laboratory in 1990, he has since concentrated on spacecraft and mission systems engineering. His experience at JPL has included working as avionics system engineer and deputy spacecraft team chief on Mars Observer and spacecraft bus system engineer on Mars Global Surveyor. He has performed mission operations and information systems engineering tasks on several other projects, including most recently the UHF telecom system on the Mars Exploration Rovers. He has been involved with protocol research and development for the past five years as contract technical manager for the Space Communications Protocol Standard (SCPS) project and CCSDS File Delivery Protocol (CFDP), and is currently developing a protocol test laboratory capability to enable protocol testing in space hardware and over R/F links. He holds a B.S. in engineering from UCLA and an M.S. in aeronautical systems from the University of West Florida.

HOWARD WEISS has 27 years of experience in information security. He spent 14 years at the National Security Agency and at NSA's National Computer Security Center performing information security research on multilevel secure operating systems and networks. He has spent the last 13 years at SPARTA Inc., a small-business high tech firm, providing information security solutions for a wide variety of government and commercial customers. He has been involved in information security for space for the last eight years, first developing the Space Communications Protocol Standard (SCPS) Security Protocol and currently developing the security architecture of the Interplanetary Internet. He has participated in the development of security standards in the Internet Engineering Task Force (IETF) and chairs the Ad Hoc Security Working Group in the Consultative Committee for Space Data Systems (CCSDS). He holds a B.S. in electrical engineering from the Polytechnic Institute of New York and an M.S. in computer science from Johns Hopkins University.