Delay-Tolerant Networks (DTNs)

A Tutorial

Version 1.1
3/5/03

Forrest Warthman
Warthman Associates
forrest@warthman.com

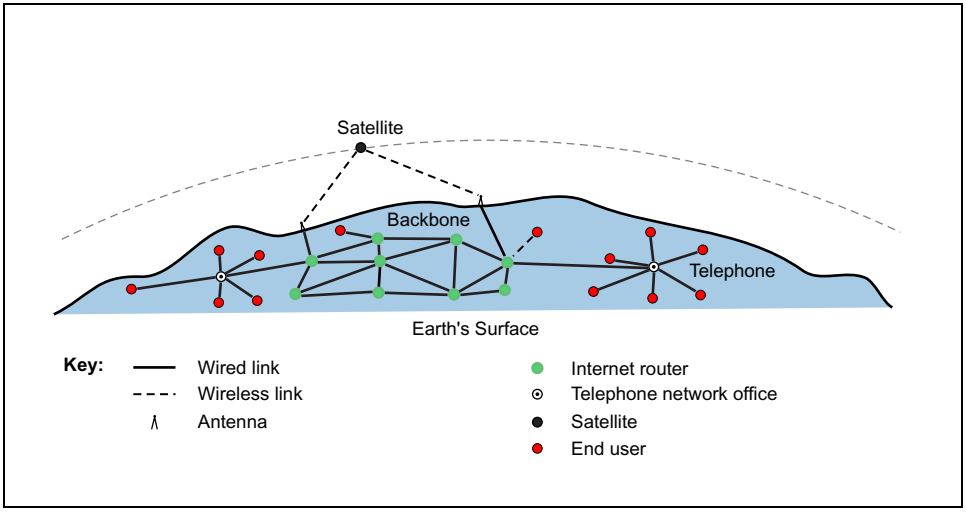
Based on Vinton Cerf, Scott Burleigh, Adrian Hooke, Leigh Torgerson, Robert Durst, Keith Scott, Kevin Fall, Howard Weiss, *Delay-Tolerant Network Architecture*, DTN Research Group Internet Draft, March 2003.

Contents

Today's Internet

The Internet has been a great success at interconnecting communication devices across the globe. It has done this by using a homogeneous set of communication protocols, called the *TCP/IP protocol suite*. All devices on the hundreds of thousands of subnets that make up the Internet use these protocols for routing data and insuring the reliability of message exchanges.

Connectivity on the Internet relies primarily on wired links, including the wired telephone network, although new wireless technologies such as short-range mobile and satellite links are beginning to appear. These links are continuously connected in end-to-end, low-delay paths between sources and destinations. They have low error rates and relatively symmetric bidirectional data rates.



Evolving Wireless Networks Outside the Internet

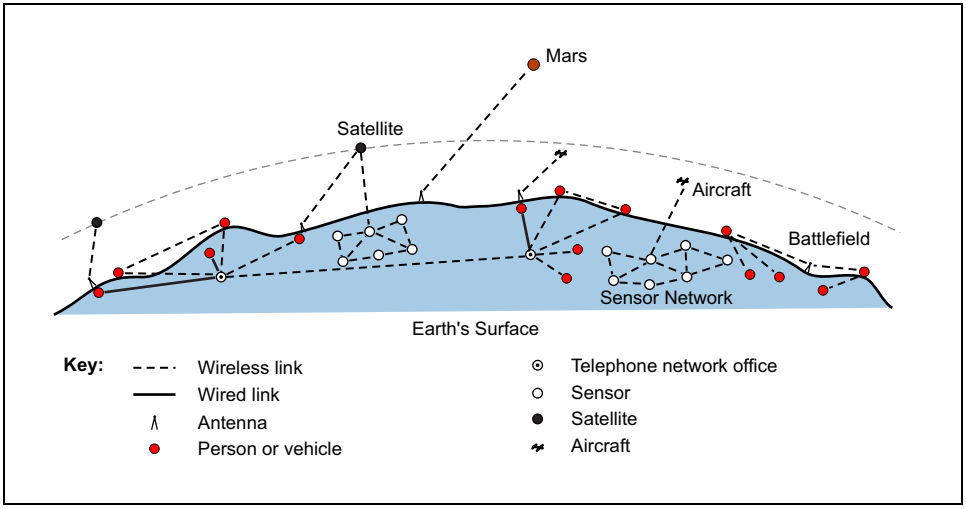
Communication outside of the Internet—where power-limited mobile wireless, satellite, and interplanetary communications are developing—is accomplished on independent networks, each supporting specialized communication requirements. These networks do not use Internet protocols and they are mutually incompatible—each is good at passing messages *within* its network, but not able to exchange messages *between* networks.

Each network is adapted to a particular communication *region,* in which communication characteristics are relatively homogeneous. The boundaries between regions are defined by such things as link delay, link connectivity, data-rate asymmetry, error rates, addressing and reliability mechanisms, quality-of-service provisions, and trust boundaries. Unlike the Internet, these wireless networks support long and variable delays, arbitrarily long periods of link disconnection, high error rates, and large bidirectional data-rate asymmetries.

Examples of wireless networks outside of the Internet include:

- Terrestrial civilian networks connecting mobile wireless devices, including personal communicators, intelligent highways, and remote Earth outposts.
- Wireless military battlefield networks connecting troops, aircraft, satellites, and sensors (on land or in water).
- Outer-space networks, such as the InterPlaNetary (IPN) Internet project, described at *http://www.ipnsig.org*.

Spanning two network regions requires the intervention of an agent that can translate between incompatible networks characteristics and act as a buffer for mismatched network delays.

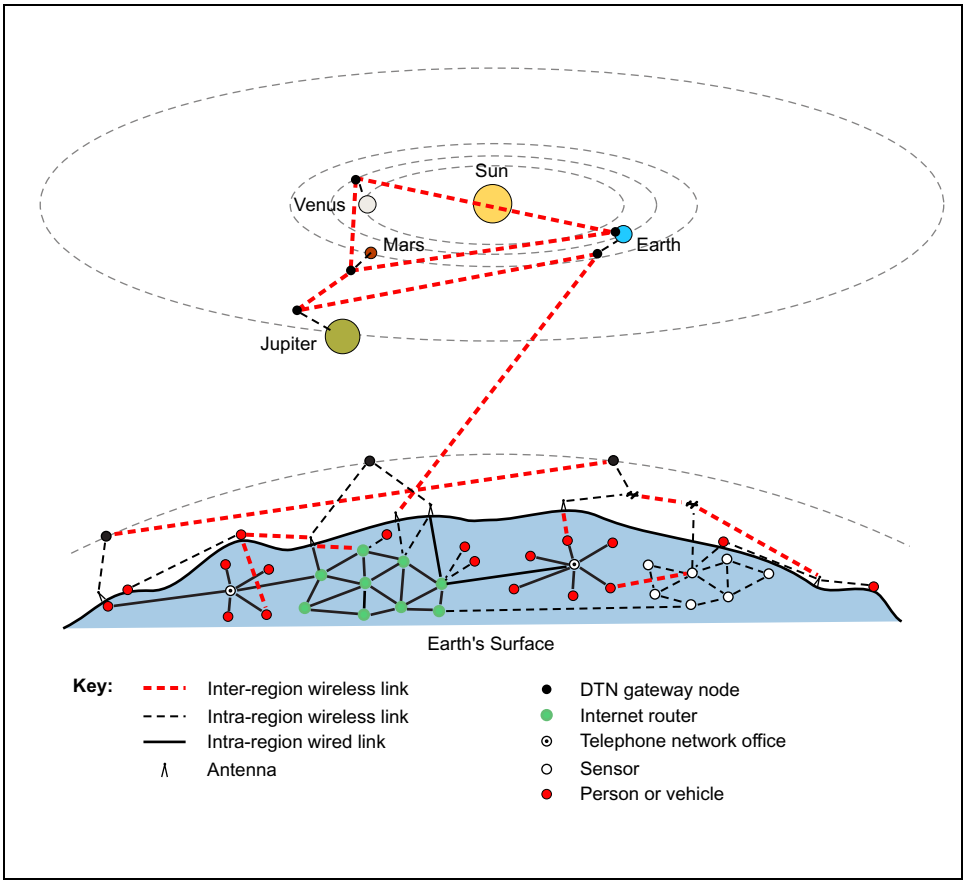


The Concept of a Delay-Tolerant Network (DTN)

A delay-tolerant network (DTN) is a *network of regional networks*. It is an *overlay* on top of regional networks, including the Internet.

DTNs support interoperability of regional networks by accommodating long delays between and within regional networks, and by translating between regional network communication characteristics. In providing these functions, DTNs accommodate the mobility and limited power of evolving wireless communication devices.

The wireless DTN technologies may be diverse, including not only radio frequency (RF) but also ultra-wide band (UWB), free-space optical, and acoustic (sonar or ultrasonic) technologies.



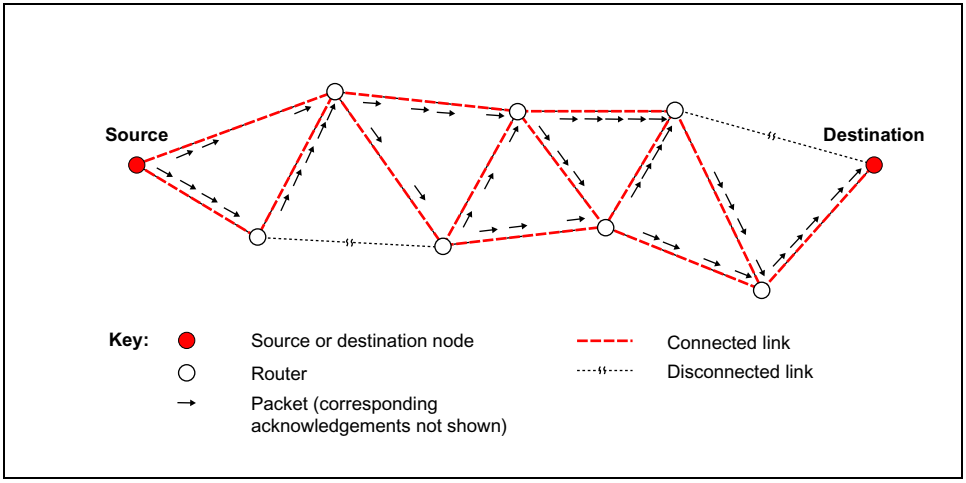
Today's Internet—Packet Switching

Communication on the Internet is based on *packet switching*. *Packets* are pieces of a complete block of user data (e.g., pieces of an email message or a web page) that travel independently from source to destination through a network of links connected by routers. The source, destination, and routers are collectively called *nodes*.

Each packet that makes up a message can take a different path through the network. If one link is disconnected, packets take another link. Packets contain both application-program user data (the payload part) and a header (the control part). The header contains a destination address and other information that determines how the packet is *switched* from one router to another. The packets in a given message may arrive out of order, but the destination's transport mechanism reassembles them in correct order.

The usability of the Internet depends on some important assumptions:

- *Continuous, Bidirectional End-to-End Path:* A continuously available bidirectional connection between source and destination to support end-to-end interaction.
- *Short Round-Trips:* Small and relatively consistent network delay in sending data packets and receiving the corresponding acknowledgement packets.
- *Symmetric Data Rates:* Relatively consistent data rates in both directions between source and destination.
- *Low Error Rates:* Relatively little loss or corruption of data on each link.



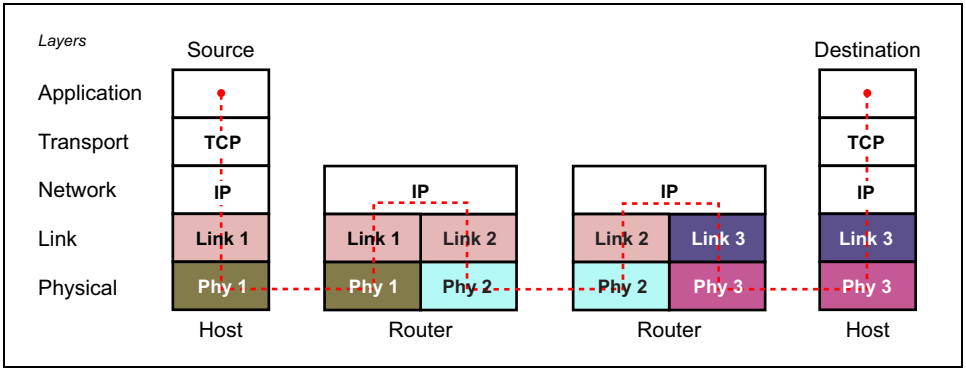
Today's Internet—Protocol Layers

Messages are moved through the Internet by *protocol layers*, a set of functions performed by network nodes on data communicated between nodes. *Hosts* (computers or other communicating devices that are the sources or destinations of messages) usually implement at least five protocol layers, which perform the following functions:

- *Application Layer:* Generates or consumes user data (messages).
- *Transport Layer:* Source-to-destination (end-to-end) segmentation of messages into message pieces and reassembly into complete messages, with error control and flow control. On the Internet, the Transmission Control Protocol *(TCP)* is used.
- *Network Layer:* Source-to-destination routing of addressed message pieces through intermediate nodes, with fragmentation and reassembly if required. On the Internet, the Internet Protocol *(IP)* is used.
- *Link Layer:* Link-to-link transmission and reception of addressed message pieces, with error control. Common link-layer protocols include Ethernet for Local-Area Networks (LANs) and Point-to-Point Protocol (PPP) for dial-up modems or very high-speed links.
- *Physical Layer:* Link-to-link transmission and reception of bit streams. Common physical media include category 5 (cat5) cable, unshielded twisted pair (UTP) telephone cable, coaxial cable, fiber-optic cable, and RF.

Routers—in their function of forwarding data (shown below)—implement only the lower three protocol layers. However, routers also implement the higher layers for routing-table maintenance and other management purposes.

The figure below shows the basic mechanism. Each hop on a path can use a different link-layer and physical-layer technology, but the IP protocol runs on all nodes and the TCP protocol runs only on source and destination end points. Several other Internet protocols and applications are also used to provide routing-path discovery, path selection, name resolution, and error recovery services.

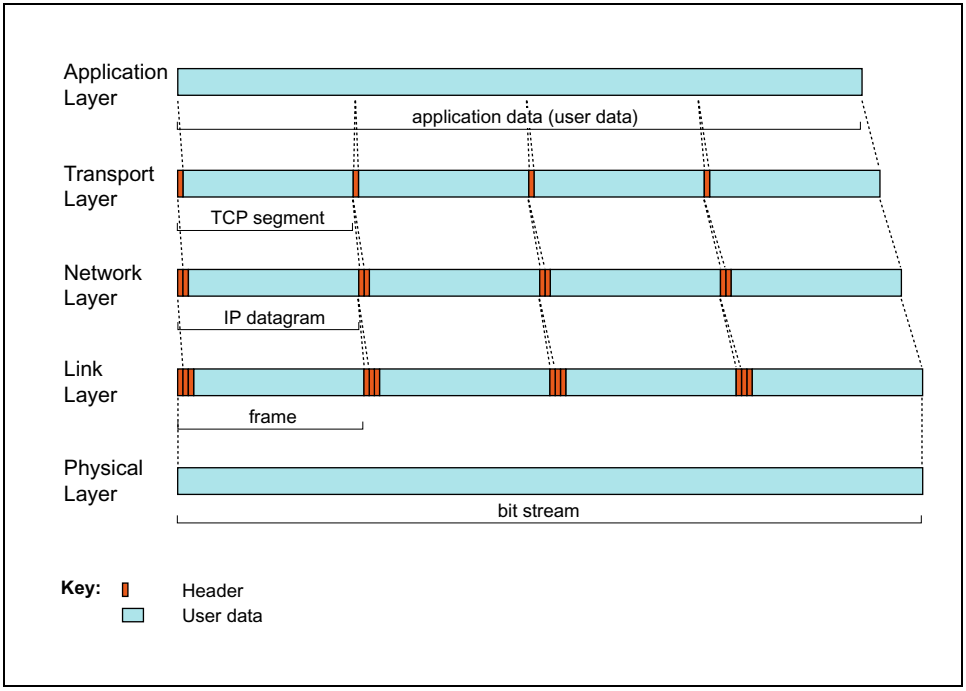


Today's Internet—Encapsulation

The term *packet* is applied to the objects actually sent over the physical links of a network. They are often called *IP packets* because the IP protocol—the only protocol used by *all nodes* on the path—is primarily responsible for directing them, node-by-node, from source to destination along their entire path (page 5).

Packets consist of a hierarchy of data-object encapsulations that are performed by the protocol layers. During transmission, higher-level data and its header are enclosed (encapsulated) in a lower-layer data object, which is given its own header. The headers are used by their respective protocol layers to control the processing of the encapsulated data. Successive headers are added at the source as user data moves down the layer structure (also called the *protocol stack)* from source application to physical layer. Headers are removed at the destination end as data moves up the layer structure to the destination application.

TCP breaks user data into pieces called *segments*. IP encapsulates the TCP segments into *datagrams*, and it may break the segments into pieces called *fragments* (not shown in the figure below). The link-layer protocol encapsulates IP datagrams into *frames*. The physical layer then transmits and receives a sequence of frames as a continuous *bit stream*.

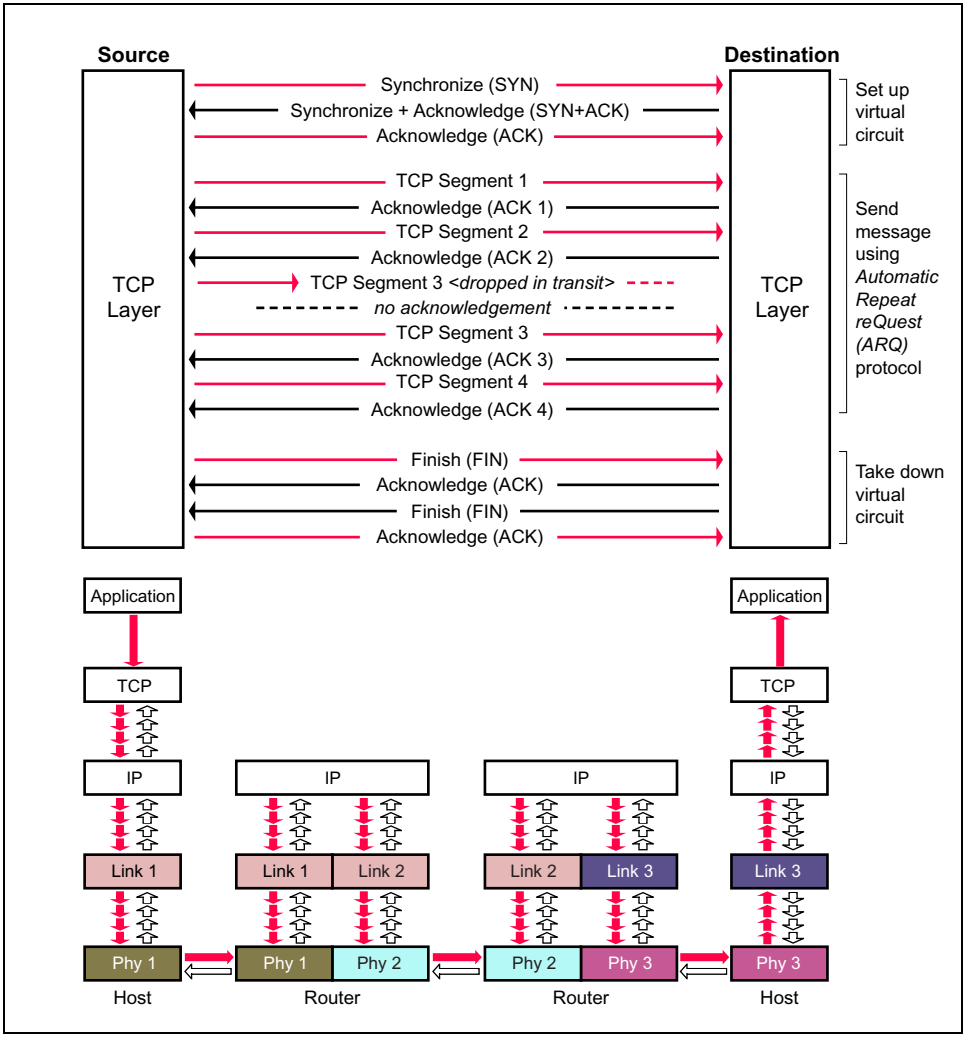


Today's Internet—Conversational Protocols

The TCP protocol is said to be *conversational* (interactive), because a complete one-way message involves many source-to-destination signaling round-trips:

- *Set Up:* A three-way *"Hello"* handshake.
- *Segment Transfer and Acknowledgement:* Each TCP segment (or a few segments) sent by the source is acknowledged by the destination.
- *Take Down:* A four-way *"Goodbye"* handshake.

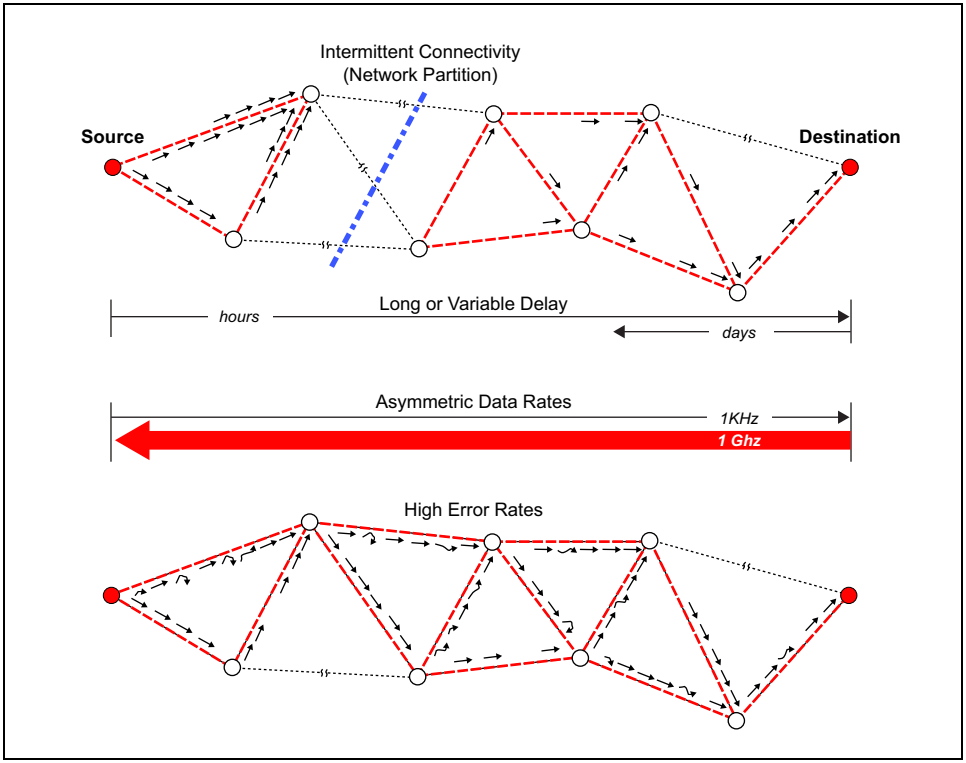
The use of positive or negative acknowledgements to control retransmission of lost or corrupt segments is called an Automatic Repeat reQuest (ARQ) protocol.



Why a Delay-Tolerant Network (DTN)?

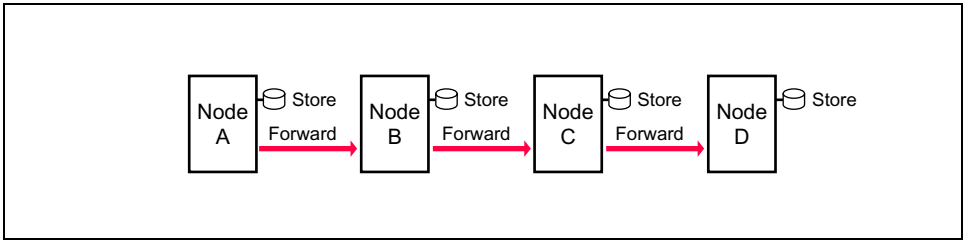
Many evolving and potential networks (page 2) do not conform to the Internet's underlying assumptions (page 4). These networks are characterized by:

- *Intermittent Connectivity:* If there is no end-to-end path between source and destination—called *network partitioning*—end-to-end communication using the TCP/IP protocols does not work. Other protocols are required.
- *Long or Variable Delay:* In addition to intermittent connectivity, long propagation delays between nodes and variable queuing delays at nodes contribute to end-to-end path delays that can defeat Internet protocols and applications that rely on quick return of acknowledgements or data.
- *Asymmetric Data Rates:* The Internet supports moderate asymmetries of bi-directional data rate for users with cable TV or asymmetric DSL access. But if asymmetries are large, they defeat conversational protocols (page 7).
- *High Error Rates:* Bit errors on links require correction (which requires more bits and more processing) or retransmission of the entire packet (which results in more network traffic). For a given link-error rate, fewer retransmissions are needed for hop-by-hop than for end-to-end retransmission (linear increase vs. exponential increase, per hop).



Store-And-Forward Message Switching

DTNs overcome the problems associated with intermittent connectivity, long or variable delay, asymmetric data rates, and high error rates by using *store-and-forward message switching*. This is an old method, used by pony-express and postal systems since ancient times. Whole messages (entire blocks of application-program user data)—or pieces (fragments) of such messages—are moved (forwarded) from a storage place on one node (switch intersection) to a storage place on another node, along a path that *eventually* reaches the destination.



Store-and-forwarding methods are also used in today's voicemail and email systems, although these systems are not one-way relays (as shown above) but rather star relays; both the source and destination independently contact a central storage device at the center of the links.

The storage places (such as hard disk) can hold messages indefinitely. They are called *persistent storage*, as opposed to very short-term storage provided by memory chips. Internet routers use memory chips to store (queue) incoming packets for a few milliseconds while they are waiting for their next-hop routing-table lookup and an available outgoing router port.

DTN routers need persistent storage for their queues for one or more of the following reasons:

- A communication link to the next hop may not be available for a long time.
- One node in a communicating pair may send or receive data much faster or more reliably than the other node.
- A message, once transmitted, may need to be retransmitted if an error occurs at an upstream (toward the destination) node or link, or if an upstream node declines acceptance of a forwarded message.

By moving whole messages (or fragments thereof) in a single transfer, the message-switching technique provides network nodes with immediate knowledge of the size of messages, and therefore the requirements for intermediate storage space and retransmission bandwidth.

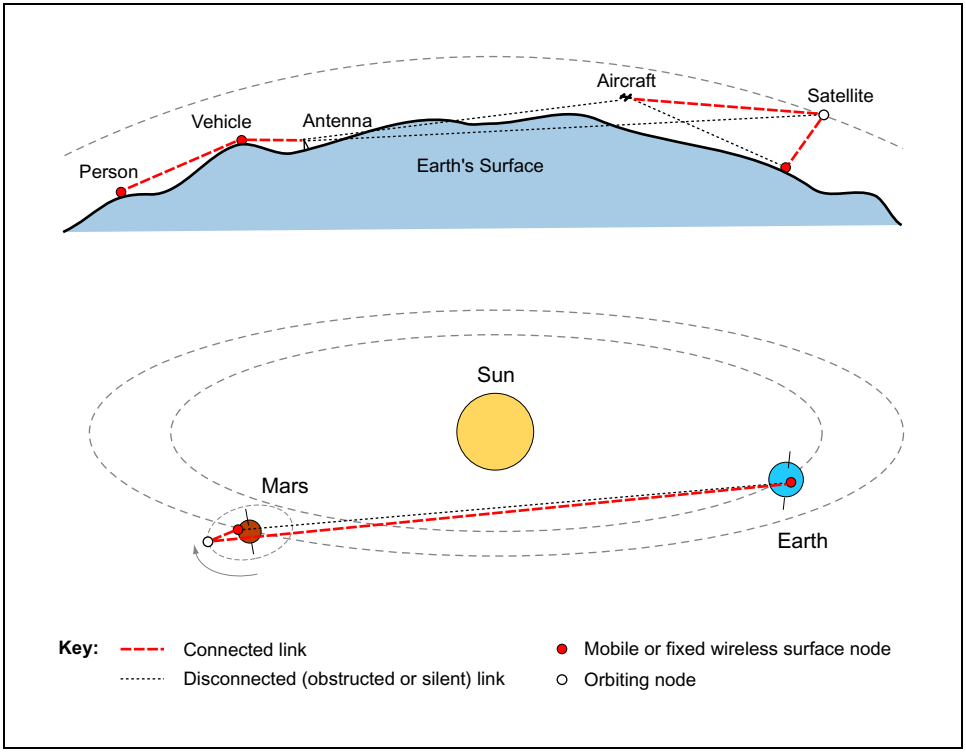
Intermittent Connectivity

A growing number of communicating devices are in motion and/or operate on limited power. This is true in interplanetary space and is becoming much more common on Earth among mobile wireless communication devices.

When communicating nodes are in motion, links can be obstructed by intervening bodies. When nodes must conserve power or preserve secrecy, links are shut down. These events cause *intermittent connectivity*. When no path exists to connect a source with a destination, a *network partition* is said to occur.

On the Internet, intermittent connectivity causes loss of data. Packets that cannot be immediately forwarded are usually dropped (discarded), and TCP may retransmit them with slower retransmission timing. If packet-dropping is too severe, TCP eventually ends the session, which can cause applications to fail.

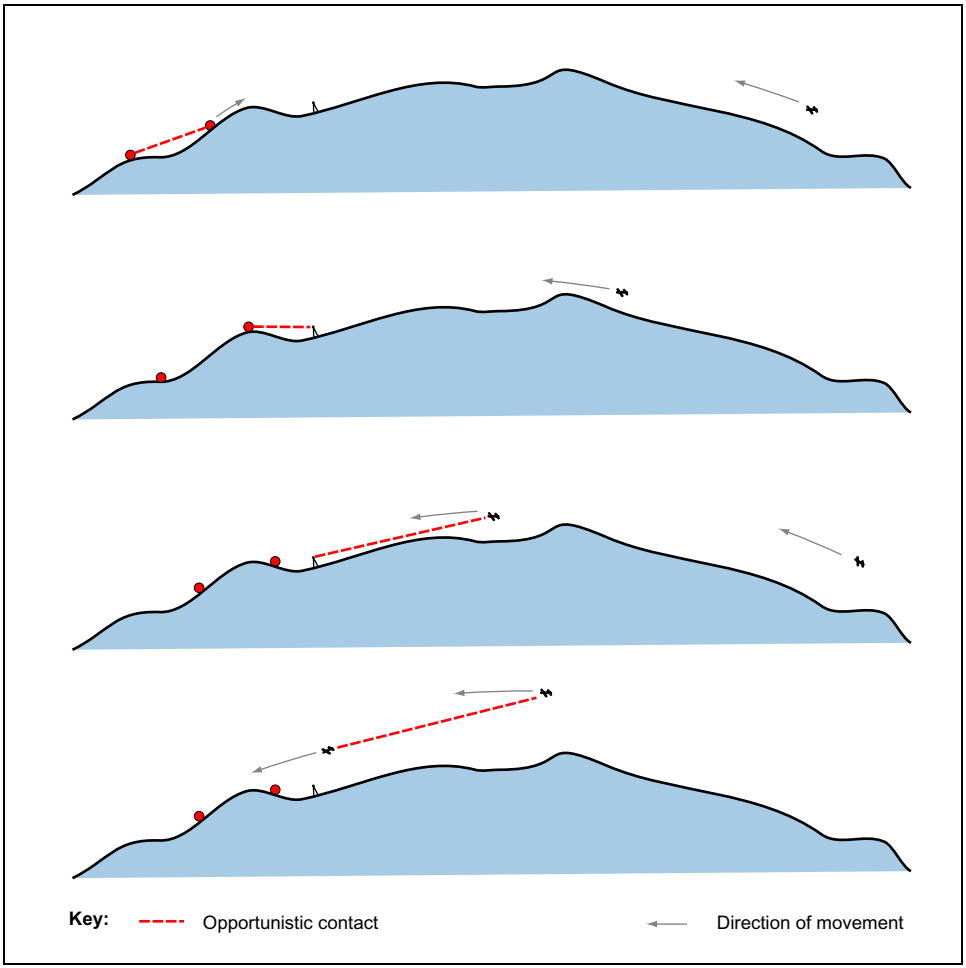
DTNs, by contrast, support communication between intermittently connected nodes by isolating delay with a store-and-forward technique (page 9).



Opportunistic Contacts

Network nodes may need to communicate during *opportunistic contacts*, in which a sender and receiver make contact at an unscheduled time. Moving people, vehicles, aircraft, or satellites may make contact and exchange information when they happen to be within line-of-sight and close enough to communicate using their available (often limited) power.

All of us use opportunistic contacts for communication: when we happen, by chance, to meet certain people with whom we wish to talk, we talk. This same model can apply to electronic communication. For example, wireless Personal Digital Assistants (PDAs) can be designed and programmed to send or receive information when certain people carrying the PDAs come within communication range, or when a PDA is carried past a certain type of information kiosk.

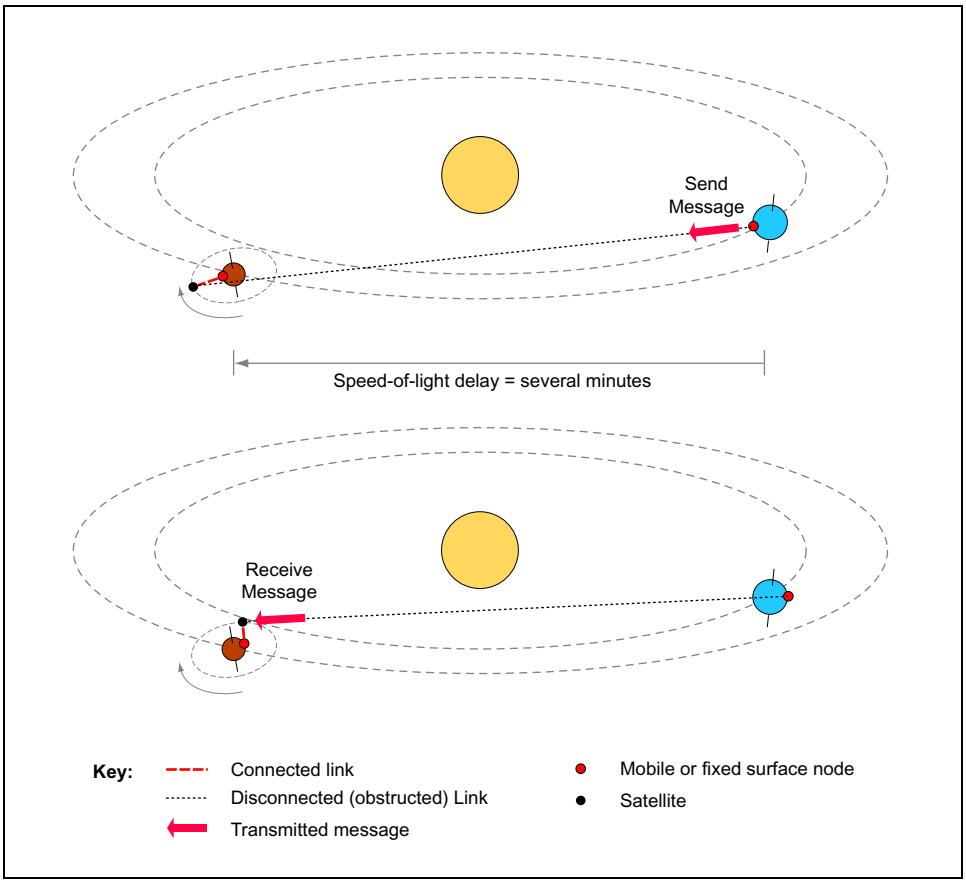


Scheduled Contacts

In space, almost everything is in motion and speed-of-light delays are significant (tens of minutes within our solar system). If potentially communicating nodes move along predictable paths, they can predict or receive time schedules of their future positions and thereby arrange their future communication sessions.

Scheduled contacts may involve message-sending between nodes that are not in direct contact, as shown in the figure below. They may also involve storing information until it can be forwarded, or until the receiving application can catch up with the sender's data rate.

Scheduled contacts require time-synchronization throughout the DTN.

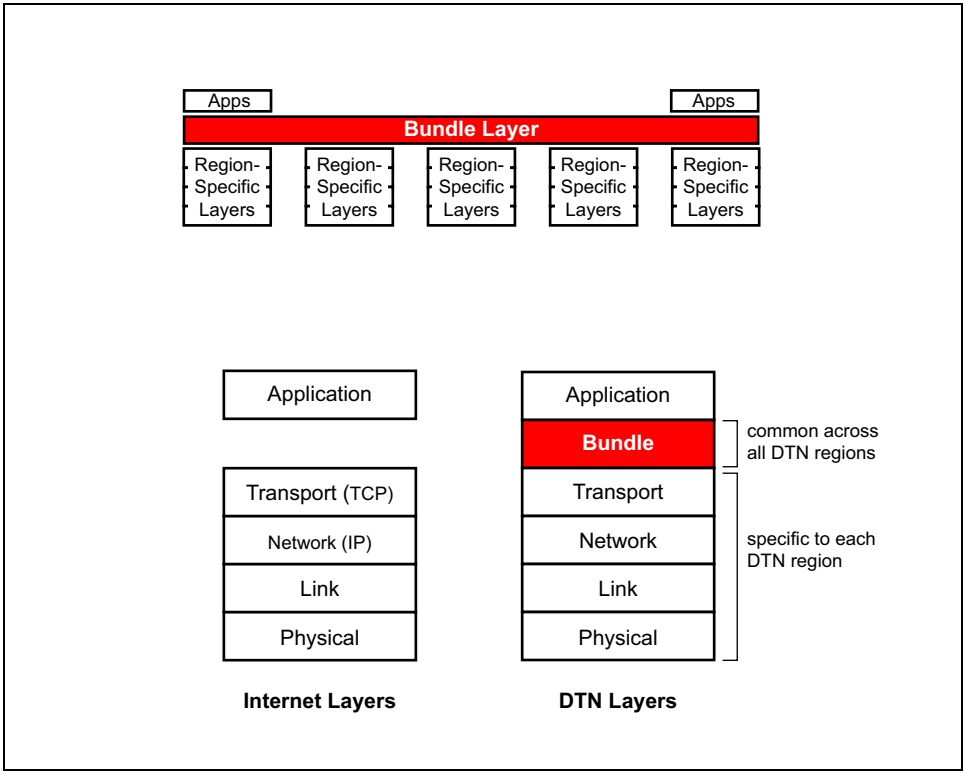


The Bundle Layer

The DTN architecture implements store-and-forward message switching by overlaying a new protocol layer—called the *bundle layer*—on top of heterogeneous region-specific lower layers. The bundle layer ties together the region-specific lower layers so that application programs can communicate across multiple regions.

Bundles are also called *messages* (as in *message-switched)*. The bundle layer stores and forwards entire bundles (or bundle fragments) between nodes. A single bundle-layer protocol is used across all networks (regions) that make up a DTN. By contrast, the layers below the bundle layer (the transport layer and below) are chosen for their appropriateness to the communication environment of each region.

The figure below illustrates the bundle overlay (top) and compares Internet protocol layers with DTN protocol layers (bottom).

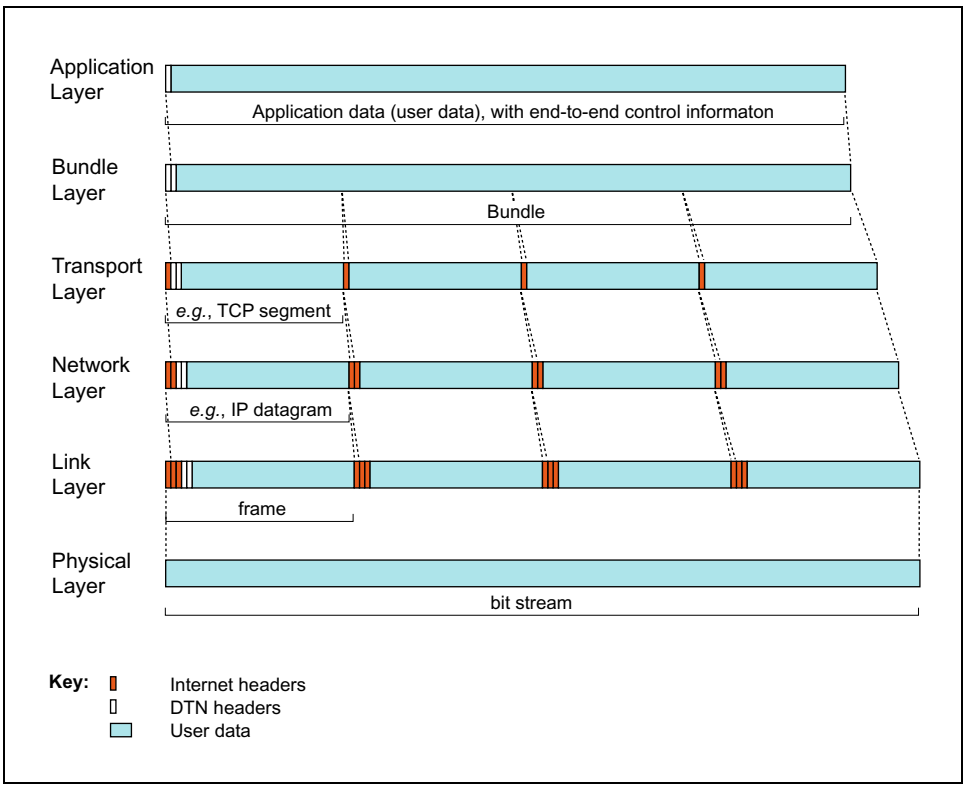


Bundles and Bundle Encapsulation

Bundles consist of three things: (1) a source-application's user data, (2) control information, provided by the source application for the destination application, describing how to process, store, dispose of, and otherwise handle the user data, and (3) a bundle header, inserted by the bundle layer. Like application-program user data, bundles can be arbitrarily long.

Bundles extend the hierarchy of data-object encapsulation performed by the Internet protocols (page 6). The example below shows how bundle-layer encapsulation works in the context of lower-layer TCP/IP protocols.

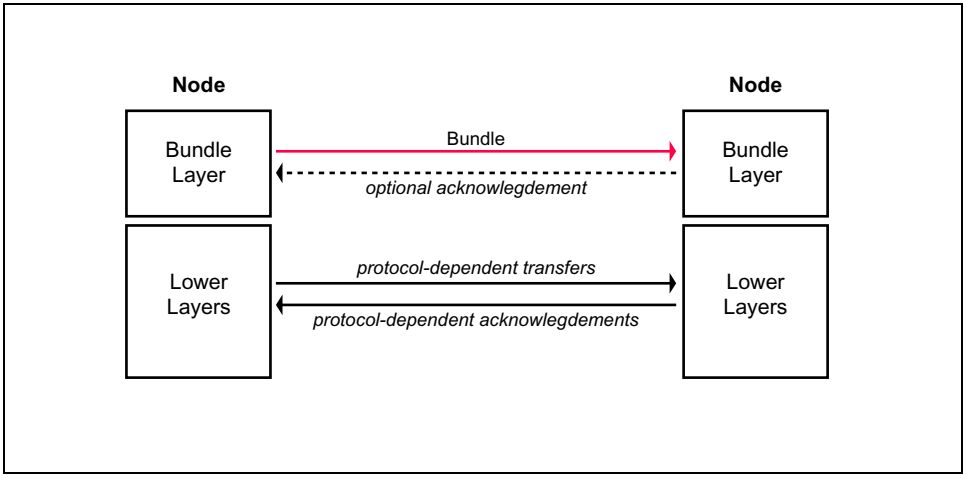
A bundle layer may break whole bundles (whole messages) into fragments (not shown in the figure below), just as an IP layer may break whole datagrams into fragments. If bundles are fragmented, the bundle layer at the final destination reassembles them.



A Non-Conversational Protocol

On intermittently connected links with long delays, conversational protocols such at TCP (page 7) that involve many end-to-end round-trips may take impractical amounts of time or fail completely. For this reason, DTN bundle layers communicate between themselves using simple sessions with minimal or no round-trips. Any acknowledgement from the receiving node is optional, depending on the class of service selected (page 21).

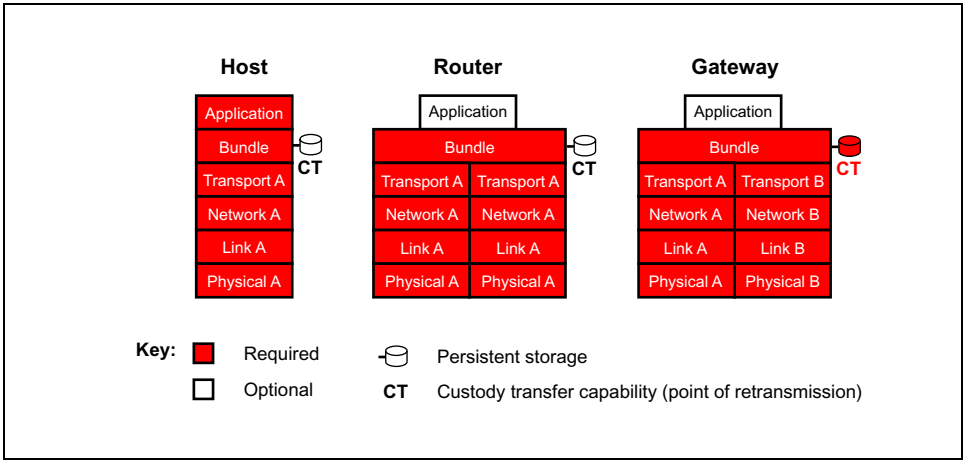
The lower-layer protocols that support bundle-layer exchanges may, of course, be conversational like TCP. But on intermittently connected links with long delays, non-conversational or minimally-conversational lower-layer protocols can be implemented.



DTN Nodes

In a DTN, a *node* is an entity with a bundle layer. A node may be a *host*, *router*, or *gateway* (or some combination) acting as a source, destination, or forwarder of bundles:

- *Host*—Sends and/or receives bundles, but does not forward them. A host can be a source or destination of a bundle transfer. The bundle layers of hosts that operate over long-delay links require persistent storage in which to queue bundles until outbound links are available. Hosts may optionally support custody transfers (page 18).
- *Router*—Forwards bundles *within* a single DTN region (page 22) and may optionally be a host. The bundle layers of routers that operate over long-delay links require persistent storage in which to queue bundles until outbound links are available. Routers may optionally support custody transfers.
- *Gateway*—Forwards bundles *between* two or more DTN regions and may optionally be a host. The bundle layers of gateways must have persistent storage and support custody transfers. Gateways provide conversions between the lower-layer protocols of the regions they span.

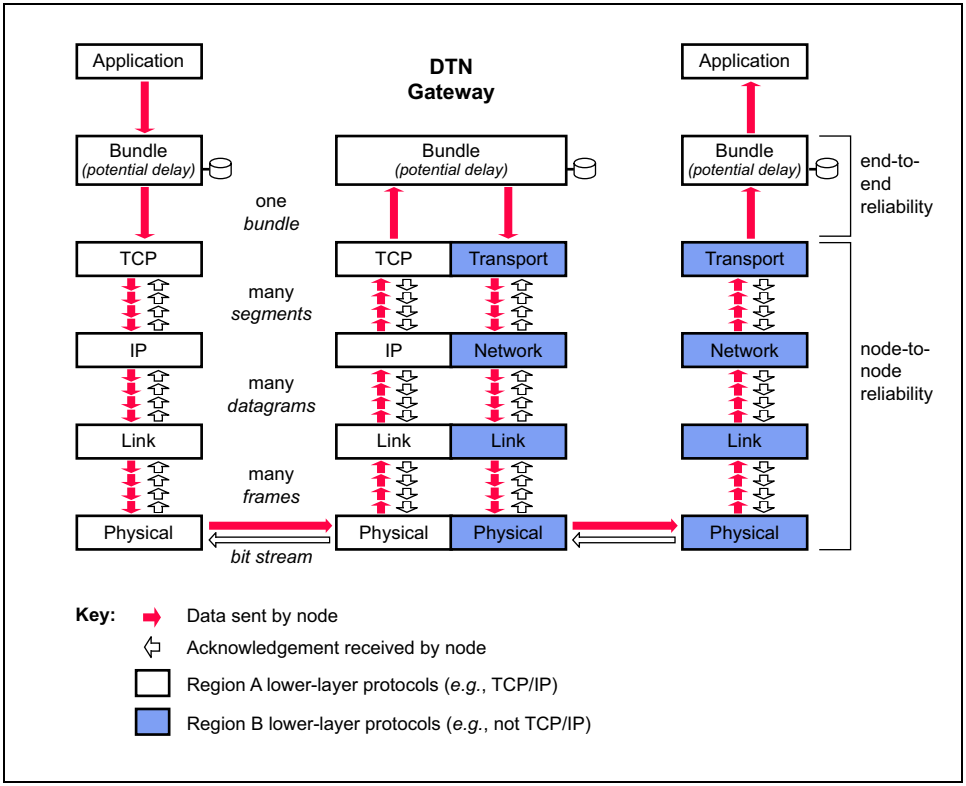


Delay Isolation via Transport-Layer Termination

On the Internet, the TCP protocol provides end-to-end (source-to-destination) reliability by retransmitting any segment that is not acknowledged by the destination. The network, link, and physical layers provide other types of data-integrity services. In a DTN, the bundle layer relies on these lower-layer protocols to insure the reliability of communication.

However, DTN routers and gateways—nodes that can forward bundles within or between DTN regions, respectively—*terminate transport protocols* at the bundle layer. The bundle layers thus act as surrogates for end-to-end sources and destinations. The side-effect is that conversational lower-layer protocols (page 7) of low-delay regions are *isolated at the bundle layer* from long delays in other regions of the end-to-end path.

The bundle layer alone supports end-to-end messaging. Bundles are typically delivered atomically, from one node to the next, independent of other bundles except for optional responses, although a bundle layer may break a single bundle into multiple bundle fragments.



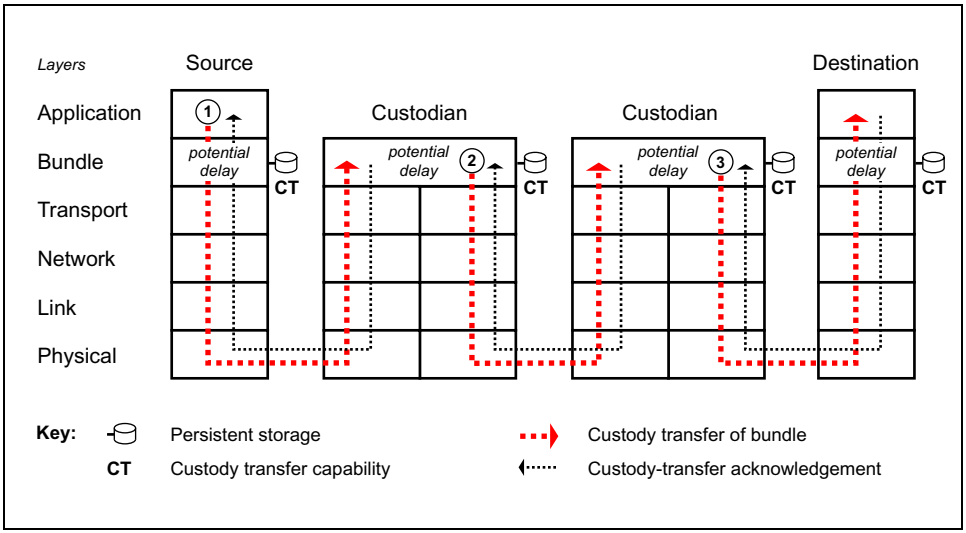
Custody Transfers

DTNs support node-to-node retransmission of lost or corrupt data at both the transport layer and the bundle layer. However, because no single transport-layer protocol (the primary means of reliable transfer) operates end-to-end across a DTN, end-to-end reliability can only be implemented at the bundle layer.

The bundle layer supports node-to-node retransmission by means of *custody transfers*. Such transfers are arranged between the bundle layers of successive nodes, at the initial request of the source application. When the current bundle-layer custodian sends a bundle to the next node, it requests a custody transfer and starts a time-to-acknowledge retransmission timer. If the next-hop bundle layer accepts custody, it returns an acknowledgment to the sender. If no acknowledgment is returned before the sender's time-to-acknowledge expires, the sender retransmits the bundle. The value assigned to the time-to-acknowledge retransmission timer can either be distributed to nodes with routing information or computed locally, based on past experience transmitting to a particular node.

A bundle custodian must store a bundle until either (1) another node accepts custody, or (2) expiration of the bundle's time-to-live, which is intended to be much longer than a custodian's time-to-acknowledge. However, the time-to-acknowledge should be large enough to give the underlying transport protocols every opportunity to complete reliable transmission.

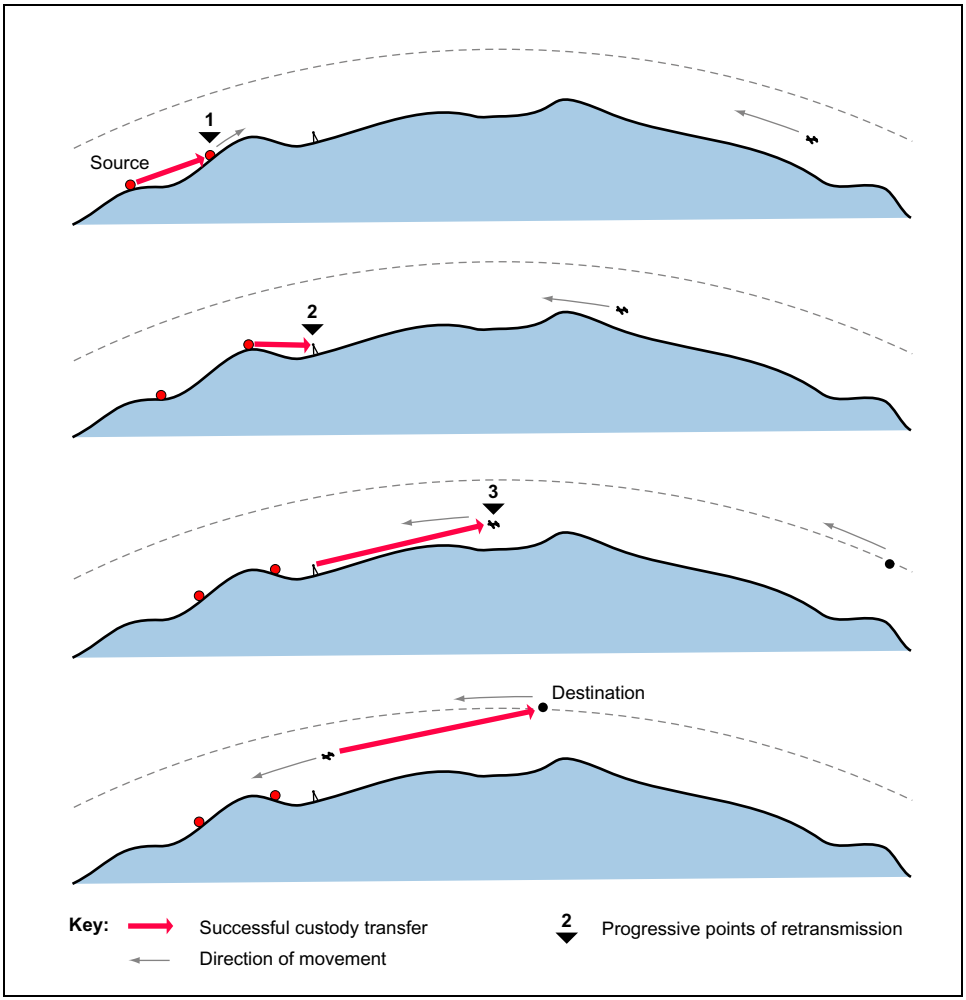
Custody transfers do not provide guaranteed end-to-end reliability. This can only be done if a source requests *both custody transfer and return receipt* (page 21). In that case, the source must retain a copy of the bundle until receiving a return receipt, and it will retransmit if it does not receive the return receipt.



Moving Points of Retransmission Forward

The bundle layer uses reliable transport-layer protocols together with custody transfers to move points of retransmission progressively forward toward the destination. The advance of retransmission points minimizes the number of potential retransmission hops, the consequent additional network load caused by retransmissions, and the total time to convey a bundle reliably to its destination.

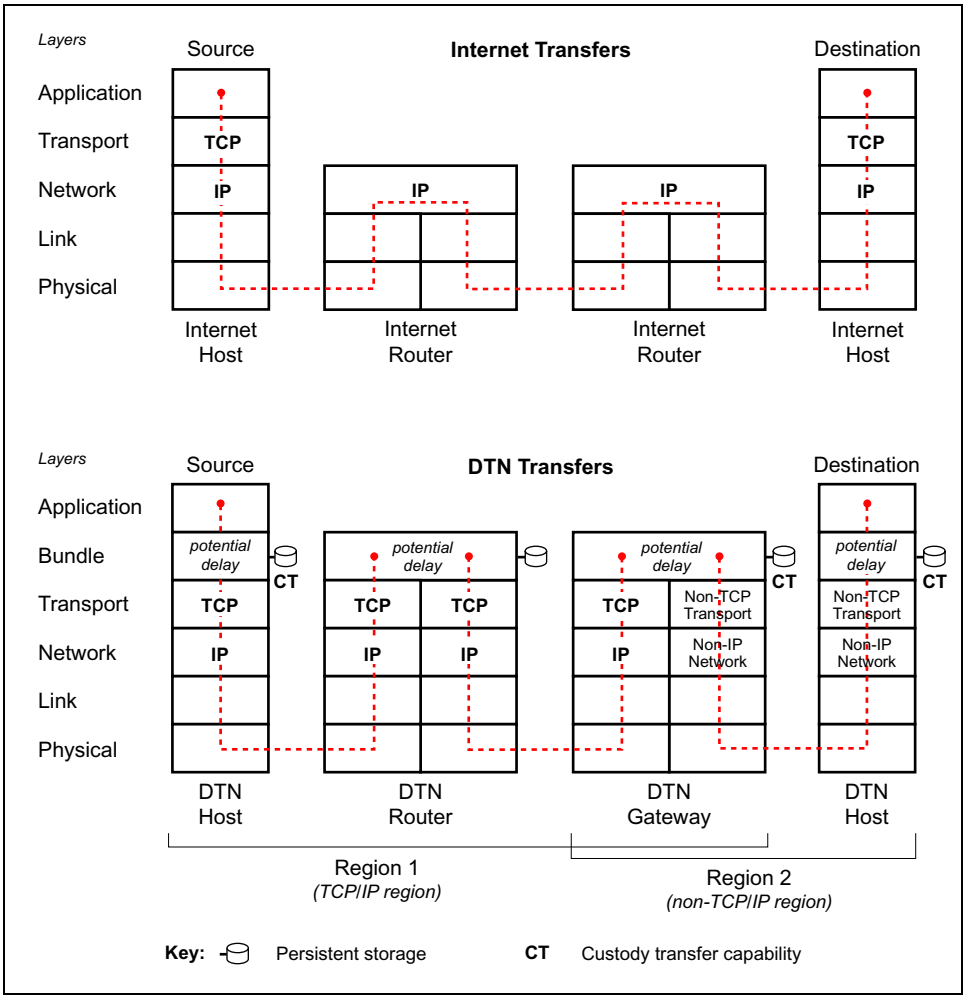
This benefits networks with either long delays or very lossy links. For paths containing many lossy links, retransmission requirements are much lower for hop-by-hop retransmission than for end-to-end retransmission (linear increase vs. exponential increase, with respect to hop count).



Internet vs. DTN Routing

On the Internet, the TCP and IP protocols are used throughout the network. TCP operates at the end points of a path, where it manages reliable end-to-end delivery of message segments. IP operates at all nodes on the path, where it routes message datagrams. Internet routers do not require a transport layer for routing, but they implement transport and application layers (not shown) for routing-table maintenance and other management purposes.

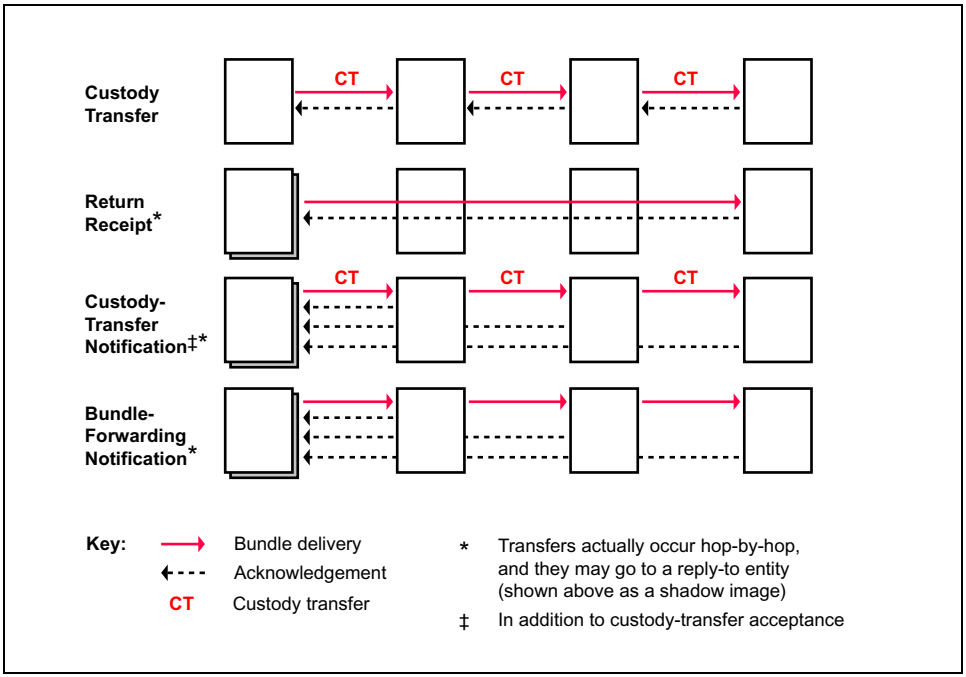
In a DTN, the protocol stacks of all nodes include both bundle and transport layers. DTN gateways have the same double-stack layers as DTN routers, but gateways can run different lower-layer protocols (below the bundle layer) on each side of their double stack. This allows gateways to span two regions that use different lower-layer protocols.



Classes of Bundle Service

The bundle layer provides six classes of service (CoS) for a bundle:

- *Custody Transfer:* Delegation of retransmission responsibility to an accepting node, so that the sending node can recover its retransmission resources. The accepting node returns a custodial-acceptance acknowledgement to the previous custodian (page 18).
- *Return Receipt:* Confirmation to the source, or its reply-to entity, that the bundle has been received by the destination application.
- *Custody-Transfer Notification:* Notification to the source, or its reply-to entity, when a node accepts a custody transfer of the bundle.
- *Bundle-Forwarding Notification:* Notification to the source, or its reply-to entity, whenever the bundle is forwarded to another node
- *Priority of Delivery:* Bulk, Normal, or Expedited.
- *Authentication:* The method (e.g., digital signature), if any, used to verify the sender's identity and the integrity of the message.

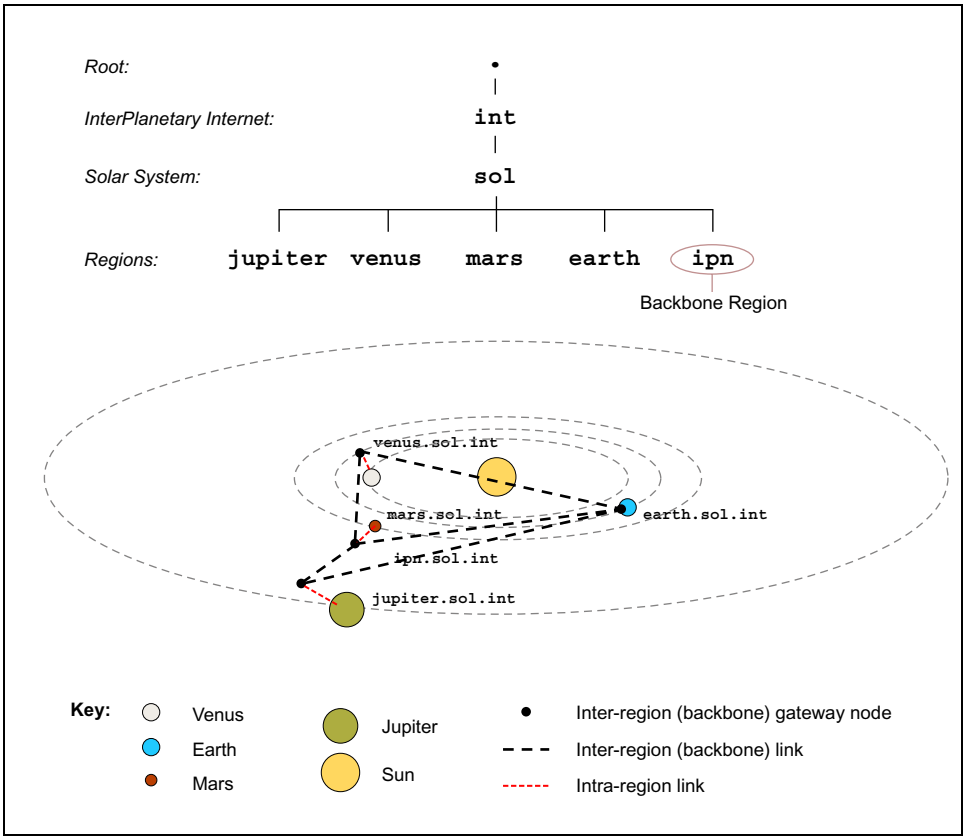


DTN Regions

A DTN is a network of networks, where each of the “networks” is a *region* in which communication characteristics are homogeneous (page 2). For example, a region can be the Earth’s Internet, a wireless personal digital assistant (PDA) network, a sensor network, a military tactical network, an intelligent highway, the surface of a planet, or a spacecraft.

Each region has a unique *region ID* which is knowable among all regions of the DTN and is part of each node’s name. DTN gateways have membership in two or more regions and are the only means of moving messages between regions.

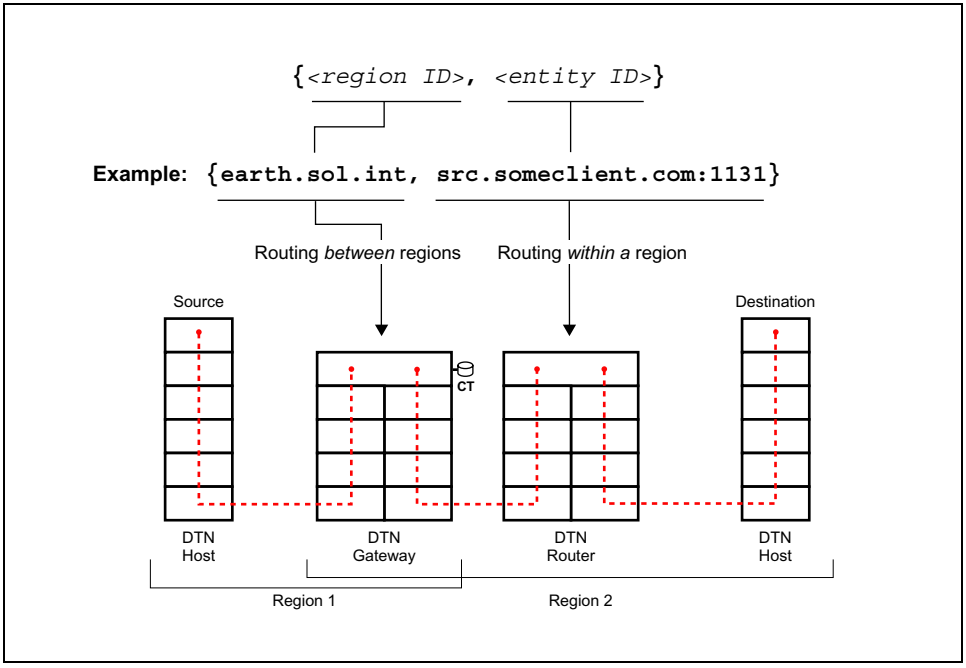
The figure below shows some of the possible regions of the IPN Special Interest Group’s InterPlaNetary (IPN) Internet concept, along with the region name-space hierarchy. The `ipn.sol.int` region forms the IPN backbone of gateways on long-haul links.



Each DTN node has a two-part name, consisting of a *region ID* (or region name) and an *entity ID* (or entity name). Routing *between* regions is based only on region IDs, which are bound to their corresponding addresses throughout the DTN. Routing *within* regions is based only on entity IDs, which are bound to their corresponding addresses only within that region. Thus, each region uses a different mapping of entity IDs to addresses, and no bandwidth is needed to copy name-address mappings between regions.

Gateways belong to two or more regions and move bundles between regions. Thus, gateways have multiple region IDs. Region IDs use the same name-space syntax as the Internet's Domain Name System (DNS).

An *entity* may be a host (a DTN node), an application instance, a protocol, a URL, a port (used to find the bundle service on a host) and potentially a token (used to find a particular application instance that is using the bundle service), or something else.

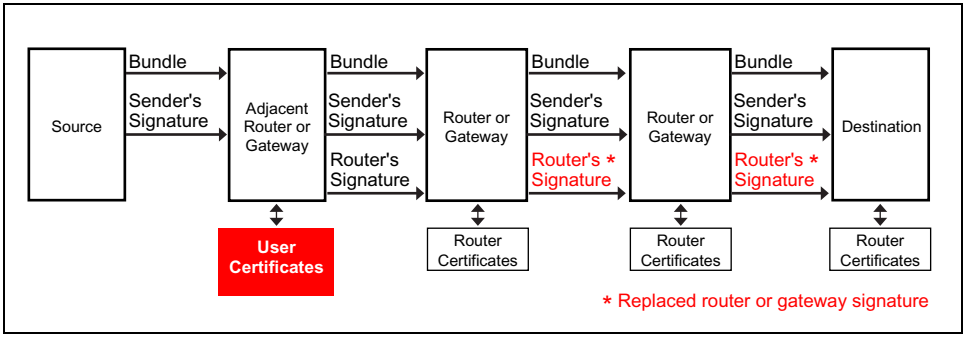


Most network security methods attempt to mutually authenticate user identities and the integrity of messages, but they do not attempt to authenticate the routers that forward information. In DTNs, forwarding nodes (routers and gateways) are also authenticated, and sender information is authenticated by forwarding nodes, so that network resources can be conserved by preventing the carriage of prohibited traffic at the earliest opportunity.

In public-key cryptography, for example, each user has a private and public key-pair. A *certificate* is a file, digitally signed by a trusted Certificate Authority (CA), confirming the user's identity and containing a confirmed copy of the user's public key. In DTN's, both users and forwarding nodes have key-pairs and certificates, and the certificates of users also indicate their class-of-service (CoS) rights (page 21). Senders can sign their bundles with their private key, producing a bundle-specific digital *signature*. The signature allows receivers—using the sender's public key—to confirm the authenticity of the sender (*i.e.*, that it was they who actually sent the message), the integrity of message (*i.e.*, that the message has not been tampered with), and the sender's CoS rights.

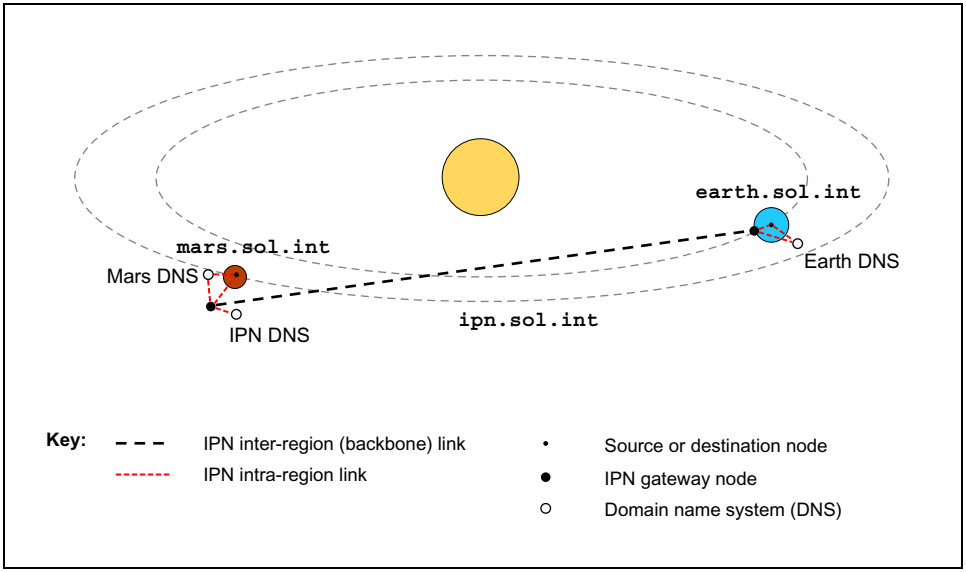
Using public-key cryptography as an example, the security steps are:

1. The source sends its bundle, together with its bundle-specific signature, to an adjacent forwarding node. If that node does not already have a copy of the senders certificate, it obtains one from the sender or a CA.
2. The forwarding node that *first* receives the sender's bundle (shown below as the *Adjacent Router or Gateway)* verifies the sender's identity and CoS rights, using its stored copies of adjacent-user certificates and CA public keys (shown below as the *User List)*. Then, the forwarding node replaces the sender's signature with its own signature (shown below as *Router's Signature)* and forwards the information.
3. Each subsequent forwarding node verifies only the identity of the previous forwarding node, using its stored copies of adjacent-router certificates and CA public keys (shown below as *Router List)*. Then, it replaces the prior-node's signature with its own signature and forwards the information.



An Interplanetary (IPN) Internet Example

The Internet Society's IPN Special Interest Group's InterPlaNetary (IPN) Internet, described at *http://www.ipnsig.org*, is a DTN. The next six pages show how a message might be sent from Earth to Mars in the IPN. The example uses three regions connected by two gateways, with a Domain Name System (DNS) for each region.



The table below shows the names of nodes accessed in the example. For simplicity, all bundle-layer applications in the Earth and Mars regions use the TCP transport protocol and reside at TCP port 6769.

Node	IPN Regions	Node Names
Source	`earth.sol.int`	`{earth.sol.int, src.jpl.nasa.gov:6769}`
Earth Gateway	`earth.sol.int` `ipn.sol.int`	`{earth.sol.int, ipngw1.jpl.nasa.gov:6769}` `{ipn.sol.int, ipngw1.jpl.nasa.gov}`
Mars Gateway	`ipn.sol.int` `mars.sol.int`	`{ipn.sol.int, ipngw2.nasa.mars.org}` `{mars.sol.int, ipngw2.nasa.mars.org:6769}`
Destination	`mars.sol.int`	`{mars.sol.int, dst.jpl.nasa.gov:6769}`

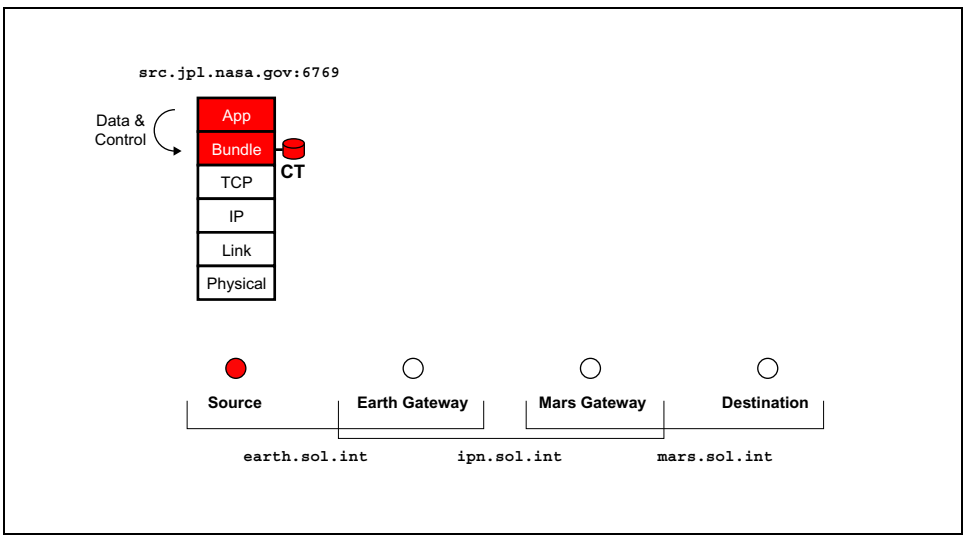
Before transfers begin, and on an on-going basis, the bundle layers of all network nodes synchronize time among themselves. This is needed for consistent calculation of contact schedules and bundle time-to-live throughout the DTN.

Step 1: Bundle Creation at Source

The source application invokes its bundle layer, requesting transfer of a bundle with a header as shown in the table below. The source's user data includes instructions to the destination application for processing, storage, disposal, and error-handling of the data. This user data is not visible to the bundle layers handling the transfer.

Item	Value
Source	`{earth.sol.int, src.jpl.nasa.gov:6769}`
Destination	`{mars.sol.int, dst.jpl.nasa.gov:6769}`
Class of service (CoS)	• Custody transfer • Normal priority • Time-to-live = 36 hours
Signature	*<bundle-specific encrypted signature using source's private key>*
User Data	Application-specific data, including instructions to the destination application for processing, storage, disposal, and error-handling. (User data is not visible to bundle layers.)

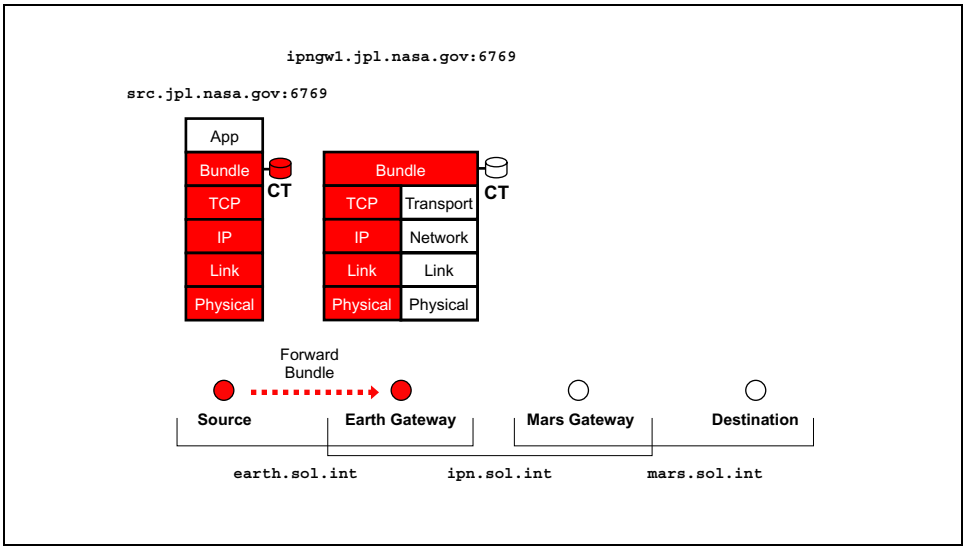
The source bundle layer verifies the source's signature, creates a bundle, appends its own signature after the bundle header, and stores the result in persistent storage. The storage is required, even if an immediate forwarding opportunity exists, because the bundle layer has accepted a *custody transfer* and must therefore be prepared to retransmit the bundle if it does not receive acknowledgement, within the bundle's time-to-acknowledge (page 18), that the subsequent custodian has received and accepted the bundle.



Step 2: Transmission by Source

The source bundle layer consults its routing table and finds that the Earth gateway `{earth.sol.int, ipngw1.jpl.nasa.gov:6769}` is the next hop capable of accepting custody transfers on a path toward the destination, and that TCP is the proper transport protocol. The source bundle layer also determines that it has a continuous connection to the Earth gateway.

The bundle layer transmits a copy of the bundle to the Earth gateway via TCP, starts a time-to-acknowledge retransmission timer (page 18), and awaits a custody-transfer acknowledgment from the gateway.



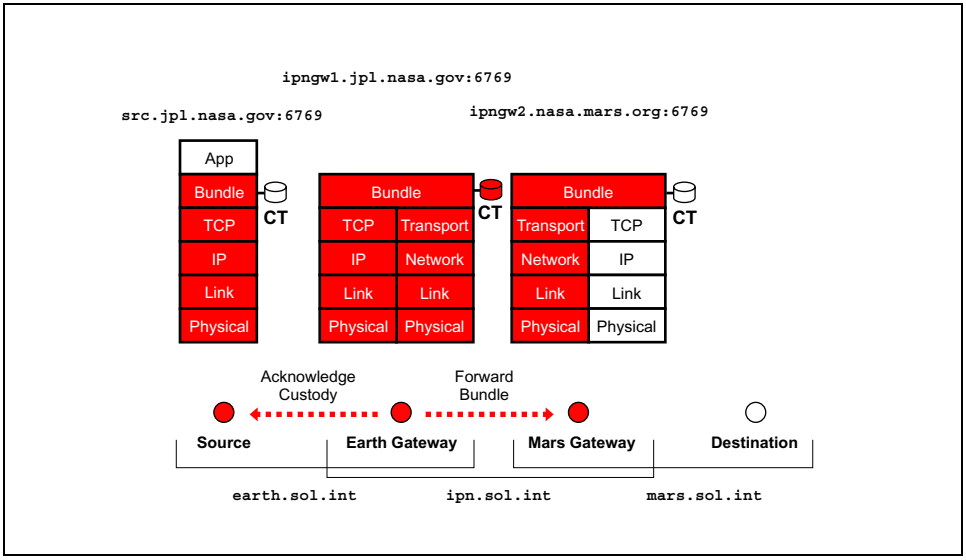
Step 3: First-Hop Bundle Processing and Forwarding

When the Earth-gateway bundle layer receives the bundle via TCP, it terminates the TCP session (page 17). Since this is a security boundary for the Interplanetary Internet, the Earth-gateway bundle layer also verifies the source application's signature and class-of-service (CoS) rights, using its stored copies of adjacent-user certificates and certificate-authority (CA) public keys or obtaining such certificates and keys as needed, and it compares the signature to its access-control list. After confirming the appropriateness of the transfer, the Earth-gateway bundle layer replaces the signature of the source bundle layer with its own, leaving the source-application's signature intact. Then it stores the received bundle in persistent storage.

The Earth-gateway bundle layer consults its routing table and finds that the Mars gateway `{mars.sol.int, ipngw2.jpl.nasa.mars.org:6769}` is the next hop capable of accepting custody transfers on a path toward the destination. It determines that the Mars gateway will be accessible at 1100 the following day, confirms that the bundle's time-to-live (page 26) is suitable for this hop's delay, and adds the bundle to its contact list for forwarding to that hop.

The Earth-gateway bundle layer then accepts custody of the bundle, updates this information in the bundle header, and confirms this by acknowledgement to the source bundle layer, which deletes its custodial copy of the bundle.

At the next-hop contact time, the Earth-gateway bundle layer establishes contact via the appropriate long-haul transport protocol and forwards the bundle.



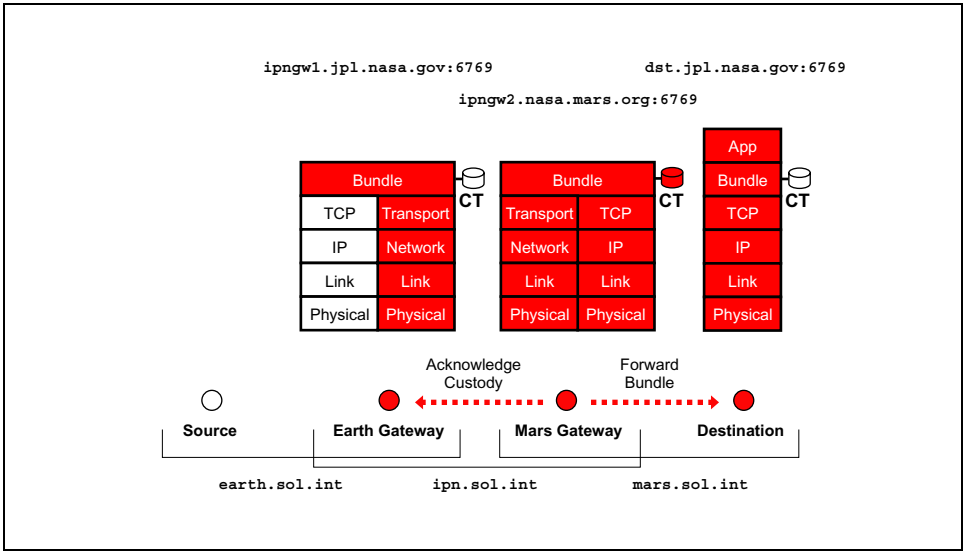
Step 4: Second-Hop Bundle Processing and Forwarding

When the Mars-gateway bundle layer receives the bundle, it terminates the long-haul transport session, and checks the signature of the Earth-gateway bundle layer, using its stored copies of adjacent-router certificates and certificate-authority (CA) public keys. It determines that the bundle has been forwarded by a legitimate source, and replaces the signature of the Earth-gateway bundle layer with its own, leaving the source-application's signature intact. Then, it stores the received bundle in persistent storage.

The Mars-gateway bundle layer consults its routing table and finds that the destination itself is the next hop. It determines that the destination is accessible immediately, that the proper transport protocol is TCP, and confirms that the bundle's time-to-live (page 26) is suitable for this hop's delay.

The Mars-gateway bundle layer then accepts custody of the bundle, updates this information in the bundle header, and confirms this by acknowledgement to the Earth-gateway bundle layer, which deletes its custodial copy of the bundle.

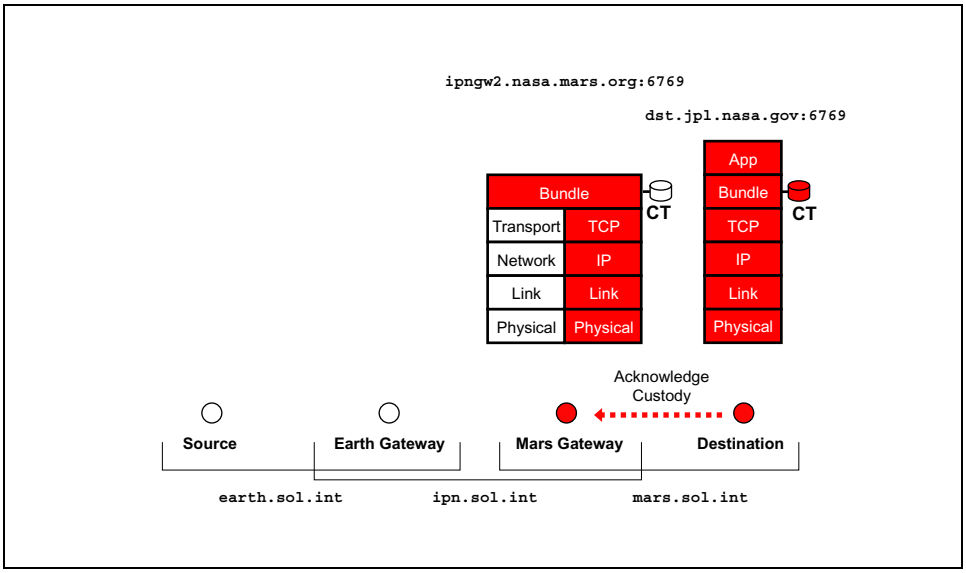
The Mars-gateway bundle layer then establishes contact with the destination bundle layer via TCP and forwards the bundle.



Step 5: Bundle Reception by Destination

When the destination bundle layer receives the bundle via TCP, it terminates the TCP session and checks the signature of the Mars-gateway bundle layer, using its stored copies of adjacent-router certificates and certificate-authority (CA) public keys. It determines that the bundle has been forwarded by a legitimate source. Then it stores the received bundle in persistent storage, accepts custody of the bundle, and confirms this by acknowledgement to the Mars-gateway bundle layer, which deletes its custodial copy of the bundle.

The destination bundle layer awakens the destination application identified by the entity ID. Depending on the control part of the user data sent by the source, the destination application may generate an application-layer acknowledgment in a new bundle and send it to the source.



More Information

The delay-tolerant network (DTN) architecture is a generalization of work originally conceived to support the InterPlanetary Internet (IPN). The description and examples presented here illustrate the basic way in which a DTN can use store-and-forward message switching in many types of environments. The primary goals of a DTN are interoperability across network environments, and reliability capable of surviving hardware (network) and software (protocol) failures.

More information about the DTN architecture is available at:

- The Internet Research Task Force's Delay-Tolerant Networking Research Group (DTNRG), at:
  - *http://www.dtnrg.org*
- The InterPlaNetary (IPN) Internet Project, described on the Internet Society's IPN Special Interest Group's site at:
  - *http://www.ipnsig.org*

Bibliography

V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, H. Weiss, *Delay-Tolerant Network Architecture*, DTN Research Group Internet Draft, Draft 2, <draft_irtf_dtnrg_arch_02>, March 2003.

Kevin Fall, *A Delay-Tolerant Network Architecture for Challenged Internets*, Intel Research Berkeley, Technical Report IRB-TR-03-003.

S. Burleigh, V. Cerf, R. Durst, K. Fall, A. Hooke, K. Scott, L. Torgerson, H. Weiss, *Bundle Layer Protocol Specification*, V 0.4, 9/6/2002, *http://www.dtnrg.org/specs/blps-0.4.pdf*.

Adrian J. Hooke, *Interplanetary Internet*, IPN Special Interest Group, (*http://www.ipnsig.org/reports/ISART9-2000.pdf*), September 2000.

Scott Burleigh, Vint Cerf, Bob Durst, Adrian Hooke, Keith Scott, Eric Travis, Howard Weiss, *The Interplanetary Internet: Status and Plans*, DARPA Next-Generation Internet (NGI) Network, (*http://www.ngi-supernet.org/NGI-PI-2001/Cerf.pdf*), January 2002.

Scott Burleigh, Vint Cerf, Bob Durst, Adrian Hooke, Robert Rumeau, Keith Scott, Eric Travis, Howard Weiss, *The Interplanetary Internet: The Next Frontier in Mobility*, IPN Special Interest Group, (*http://www.ipnsig.org/reports/INETPlenary-06June01.ppt*), June 2001.

Robert C. Durst, Patrick D. Feighery, Keith L. Scott, *Why not use the Standard Internet Suite for the Interplanetary Internet?*, IPN Special Interest Group, (*http://www.ipnsig.org/reports/TCP_IP.pdf*).

K. Fall, *Delay-Tolerant Networking for Extreme Environments*, Intel Research, Berkeley, CA (*http://www.ipnsig.org/reports/Kevin-paper.pdf*).

The InterPlanetary Internet Bulletin, IPN Special Interest Group, (*http://www.ipnsig.org/reports/IPN-Bulletin-Feb0102.pdf*), January 2002.

Index