Delay- and Disruption-Tolerant Networks (DTNs)

A Tutorial

Version 2.0
7/23/12

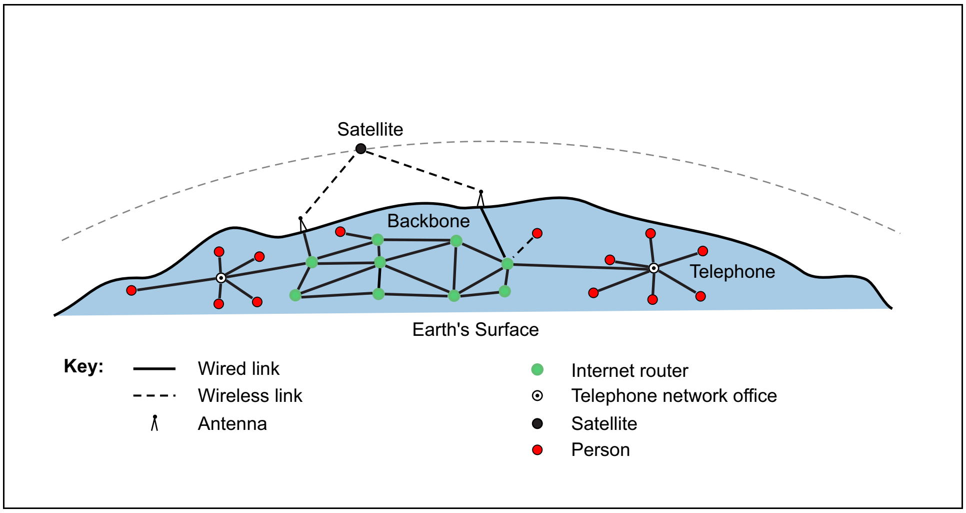
By Forrest Warthman, Warthman Associates, based on technology developed by the Interplanetary Internet Special Interest Group (IPNSIG, http://www.ipnsig.org).

Contents

Today’s Internet

The Internet has been a great success at interconnecting communication devices across the Earth. 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 networks 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 wireless technologies such as satellite and short-range mobile links are also an essential part of the network. These links, as used on the Internet, 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

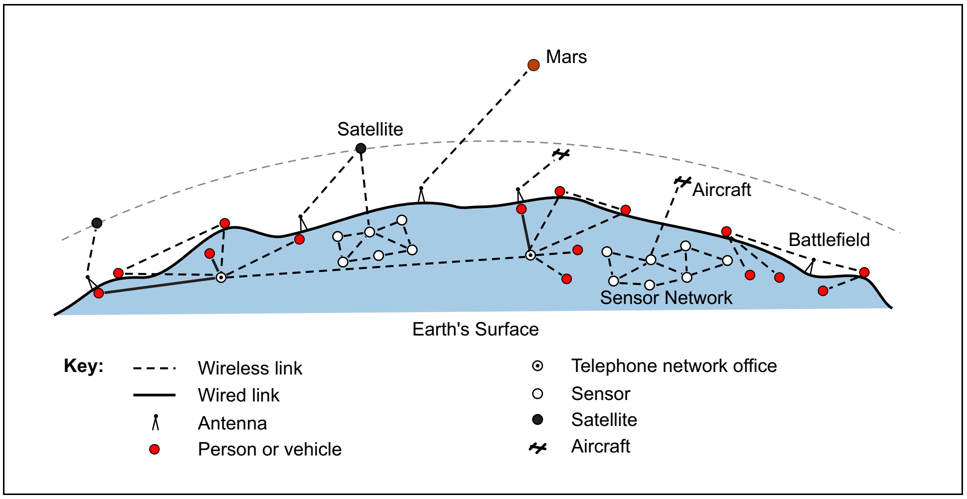
Communication outside of the Internet—where power-limited wireless communications are developing—is done on independent networks, each supporting specialized communication protocols. Most of these networks are mutually incompatible—each is good at passing messages *within* its network, but not able to exchange messages *between* networks.

The nodes of each network communicate among themselves using the same set of communication protocols. Each network also has communication characteristics that are relatively homogeneous—for example, link delay, link connectivity, data-rate asymmetry, error rates, addressing and reliability mechanisms, quality-of-service provisions, and trust boundaries. Unlike the Internet, the evolving wireless networks experience 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:

- Civilian networks on Earth that connect mobile wireless devices, such as networks for intelligent highways, and remote environmental and animal-movement outposts.
- Wireless military battlefield networks connecting troops, aircraft, satellites, and sensors on land and in water.
- Outer-space networks, such as the InterPlaNetary (IPN) Internet project, described at *http://www.ipnsig.org*.

Spanning two networks requires a protocol agent that can translate between network protocols and act as a buffer for mismatched network delays.



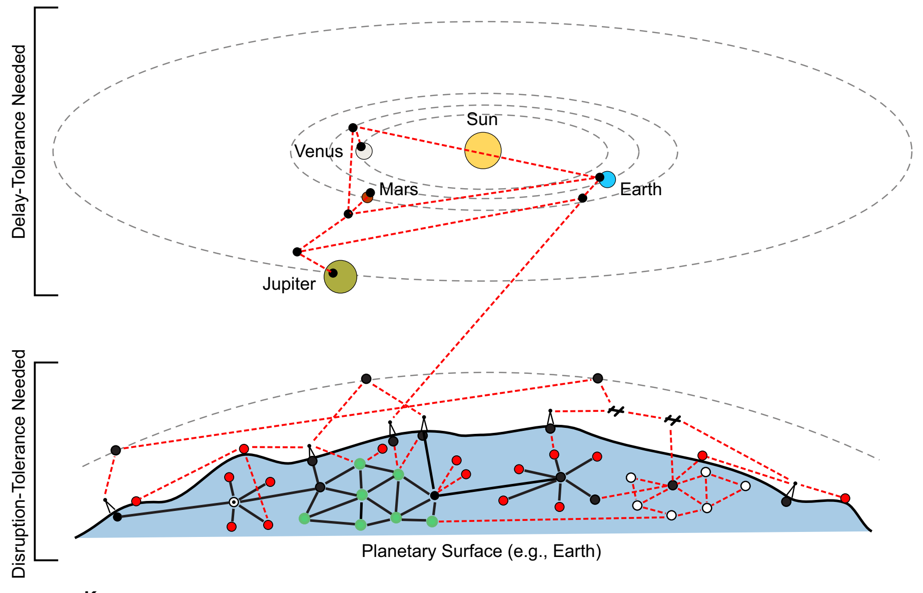
The Concept of a Delay- and Disruption-Tolerant Network (DTN)

A DTN is a network of smaller networks. It is an overlay on top of special-purpose networks, including the Internet.

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

DTNs were originally developed for interplanetary use, where the speed of light can seem slow and delay-tolerance is the greatest need. However, DTNs may have far more diverse applications on Earth, where disruption-tolerance is the greatest need. The potential Earth applications span a broad range of commercial, scientific, military, and public-service applications (page 32).

DTNs can accommodate many kinds of wireless technologies, including radio frequency (RF), ultra-wide band (UWB), free-space optical, and acoustic (sonar or ultrasonic) technologies.



Key:

- - -	Wireless communication link	🔴	Person or vehicle
—	Wired communication link	🟢	Internet router
Antenna symbol	Antenna	⊙	Telephone central office
●	DTN node	○	Sensor
Aircraft symbol	Aircraft with DTN node		

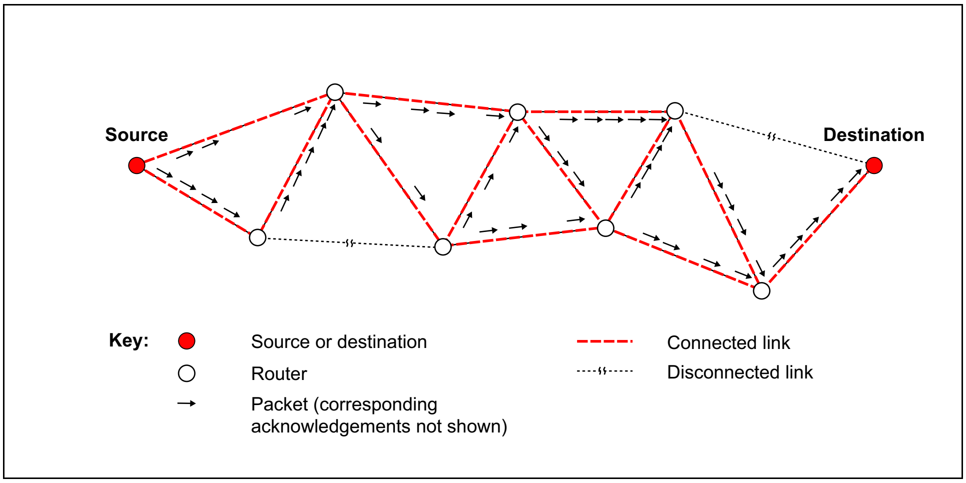
Today’s Internet—Packet-Switching

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. Routers switch the direction in which the packets move. The source, destination, and routers are collectively called *nodes*.

Each packet that makes up a message can take a different path through the network of routers. If one link is disconnected, routers redirect the packets to use an alternate 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—milliseconds, not hours or days—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.

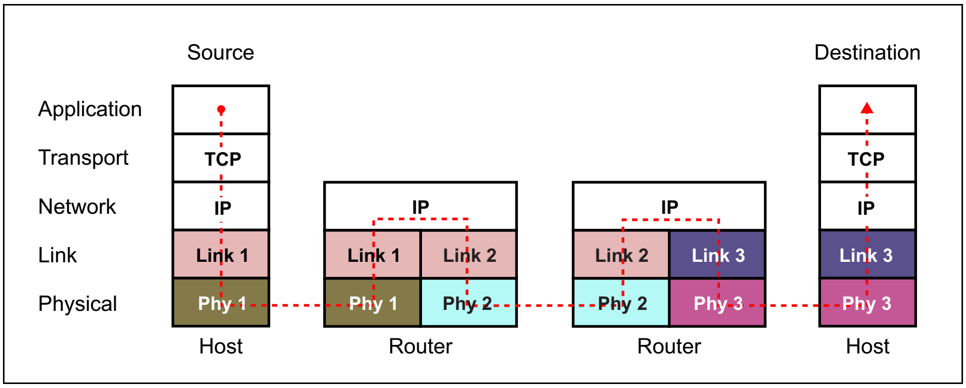


Today’s Internet Protocol Layers

Data is moved through the Internet by *protocol layers*, which are sets of functions performed by network nodes on data communicated between nodes. *Hosts* (computers that are the sources or destinations of data) usually implement at least five protocol layers that perform the following functions:

- *Application Protocol:* Generation or consumption of user data.
- *Transport Protocol:* Segmentation of user data into pieces at the source, and reassembly of the pieces at the destination, with error control and flow control. On the Internet, the *Transmission Control Protocol (TCP)* does this.
- *Network Protocol:* Source-to-destination routing of addressed pieces of user data through intermediate nodes, with fragmentation and reassembly if required. On the Internet, the *Internet Protocol (IP)* does this.
- *Link Protocol:* Link-to-link transmission and reception of addressed pieces of user data, 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 high-speed links.
- *Physical Protocol:* 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.

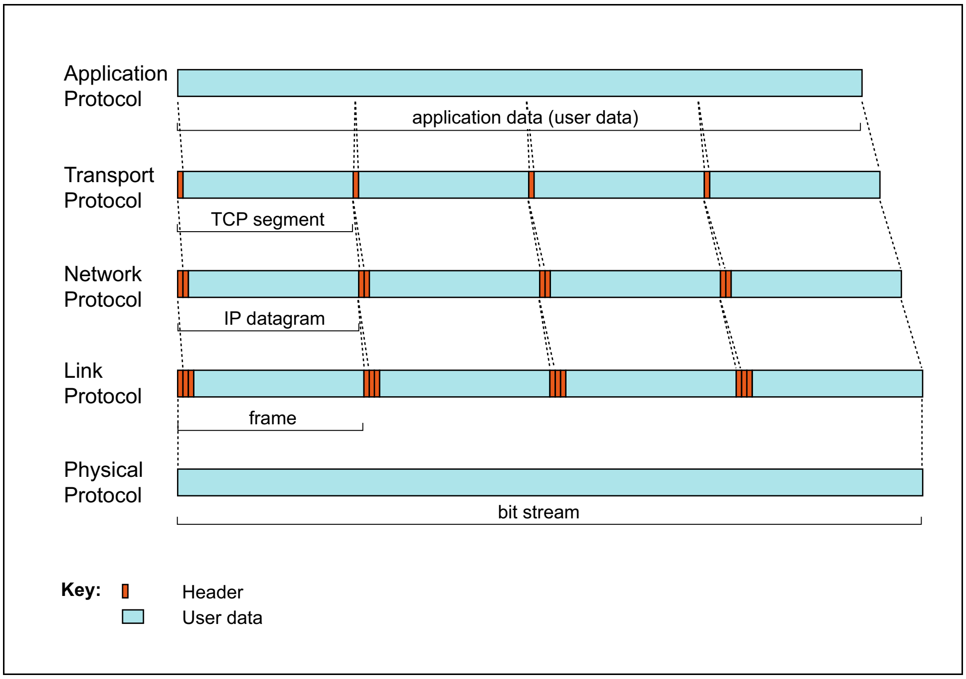
The figure below shows the basic mechanism. Each hop on a path can use a different link-layer and physical-layer protocol, but the IP protocol runs on all nodes and the TCP protocol runs only on source and destination hosts. Routers, in their function of forwarding data (shown below), use only the lower three protocols. Several other Internet protocols and applications are also used to provide routing-path discovery, path selection, name resolution, and error recovery services.



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.

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 protocols to control the processing of the encapsulated data. Successive headers are added at the source as user data moves down the protocol structure (called the *protocol stack)* from the application protocol to the physical protocol on the source. Headers are removed at the destination as data moves up the protocol stack to the application protocol on the destination.

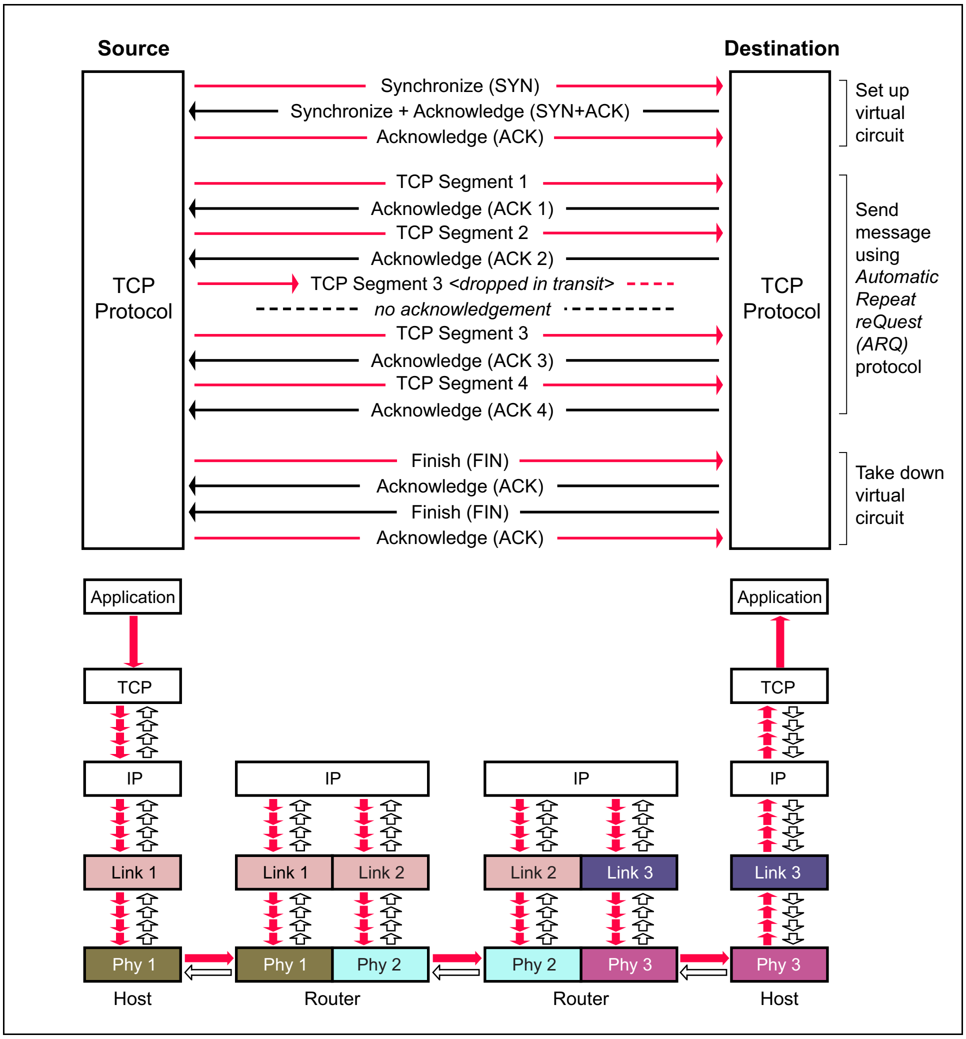
The TCP transport protocol breaks user data into pieces called *segments*. The IP network protocol encapsulates the TCP segments into *datagrams*, and it may break the segments into pieces called *fragments* (not shown in the figure below). The link protocol encapsulates IP datagrams into *frames*. The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



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.

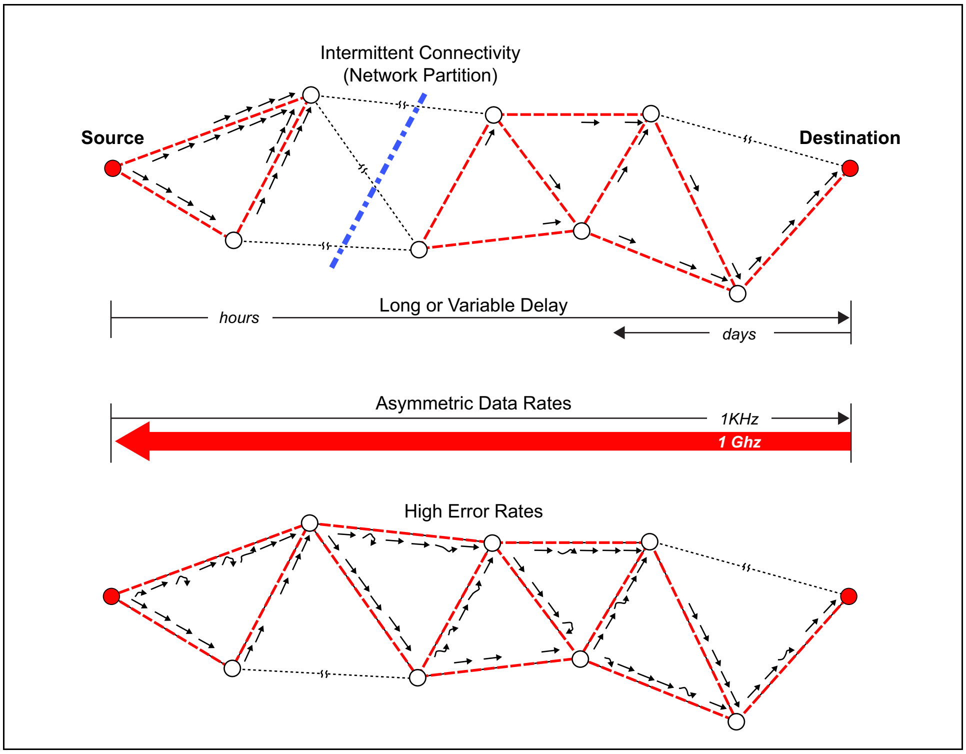
The use of positive or negative acknowledgements to control retransmission of lost or corrupt segments is called an Automatic Repeat reQuest (ARQ) protocol.



Why a Delay- and Disruption-Tolerant Network (DTN)?

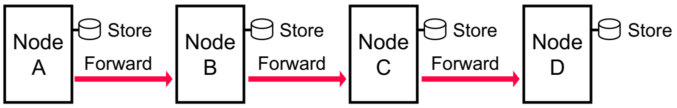
Many evolving and potential communication environments do not conform to the Internet's underlying assumptions (page 6). These environments are characterized by:

- *Intermittent Connectivity:* The absence of an end-to-end path between source and destination is called *network partitioning*. In such cases, communication using the TCP/IP protocols does not work.
- *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 bidirectional data rate for users with cable TV or asymmetric DSL service. But if asymmetries are large, they defeat conversational protocols (page 9).
- *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 retransmission than for Internet-type 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 a very 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, but these systems are not node-to-node 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 and buffers. Internet routers use memory chips and buffers 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 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.

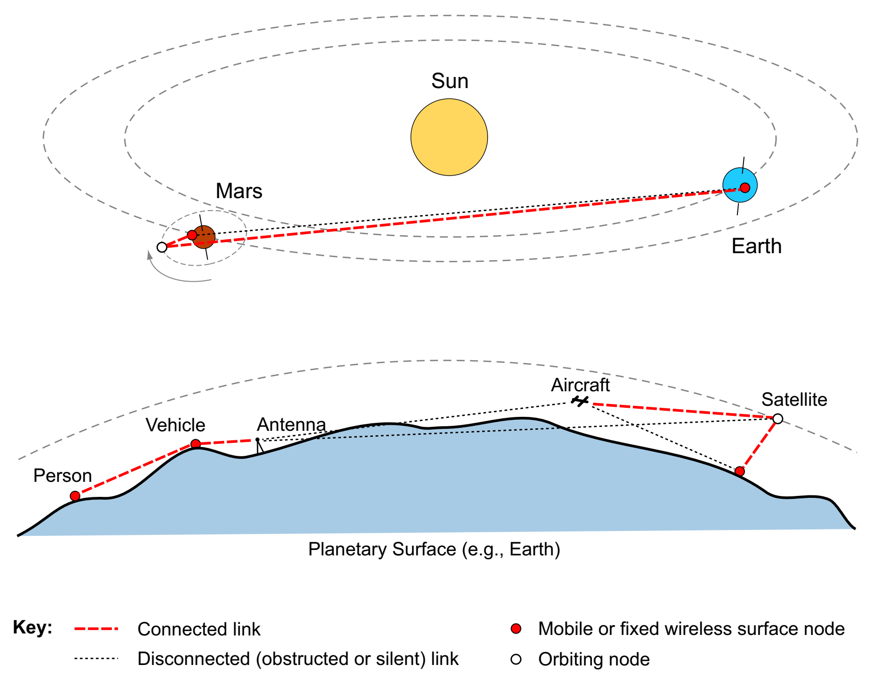
Intermittent Connectivity

A growing number of communicating devices are in motion and operate on limited power. This is true in interplanetary space and is becoming more common on Earth among mobile wireless communication devices, such as cell phones.

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 the TCP protocol may retransmit them with slower retransmission timing. If packet-dropping is too severe, TCP eventually ends the session, which can cause applications to fail.

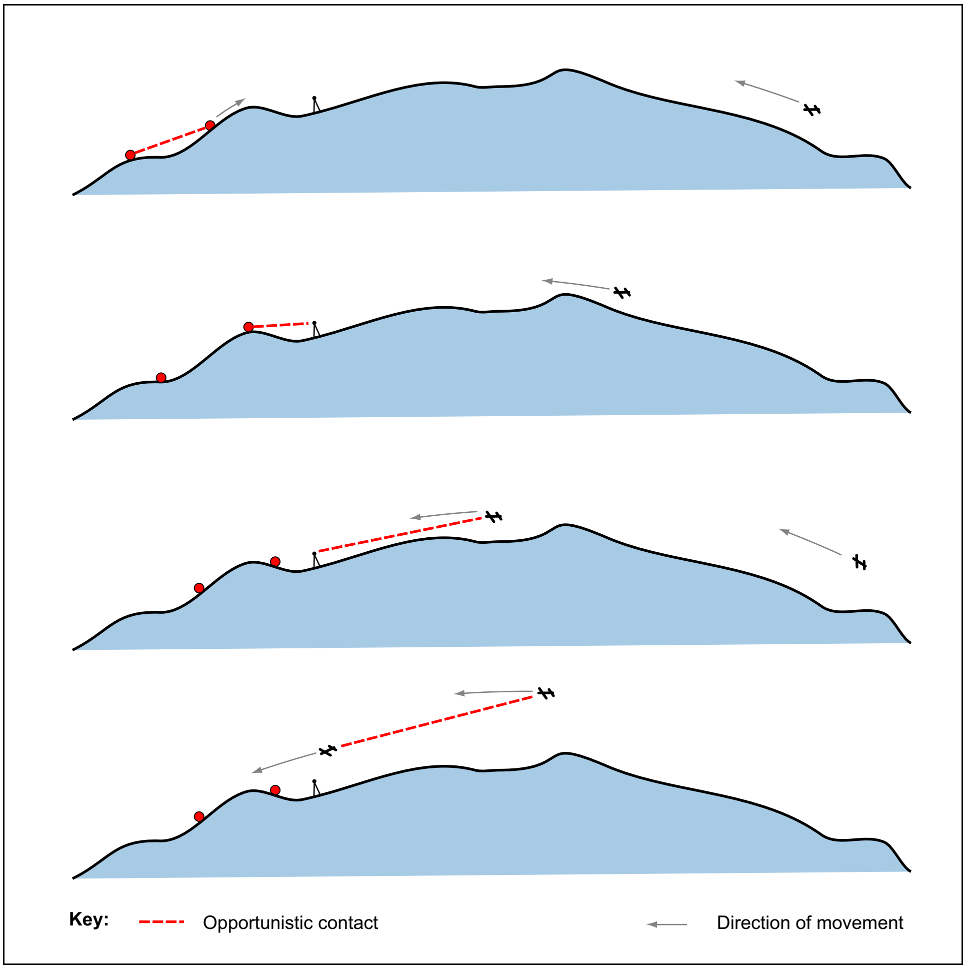
DTNs, by contrast, support communication between intermittently connected nodes by isolating delay and disruptions with a store-and-forward technique (page 11). The intermittent connectivity may be opportunistic (page 13) or scheduled (page 14).


Sun
Mars
Earth
Aircraft
Satellite
Vehicle
Antenna
Person
Planetary Surface (e.g., Earth)
Key:
Connected link
Disconnected (obstructed or silent) link
Mobile or fixed wireless surface node
Orbiting node

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 begin a conversation. This same model can apply to electronic communication. For example, wireless mobile devices such as cell phones can be designed to send or receive information when certain people carrying the mobile device come within communication range, or when the mobile device is carried past an information kiosk.

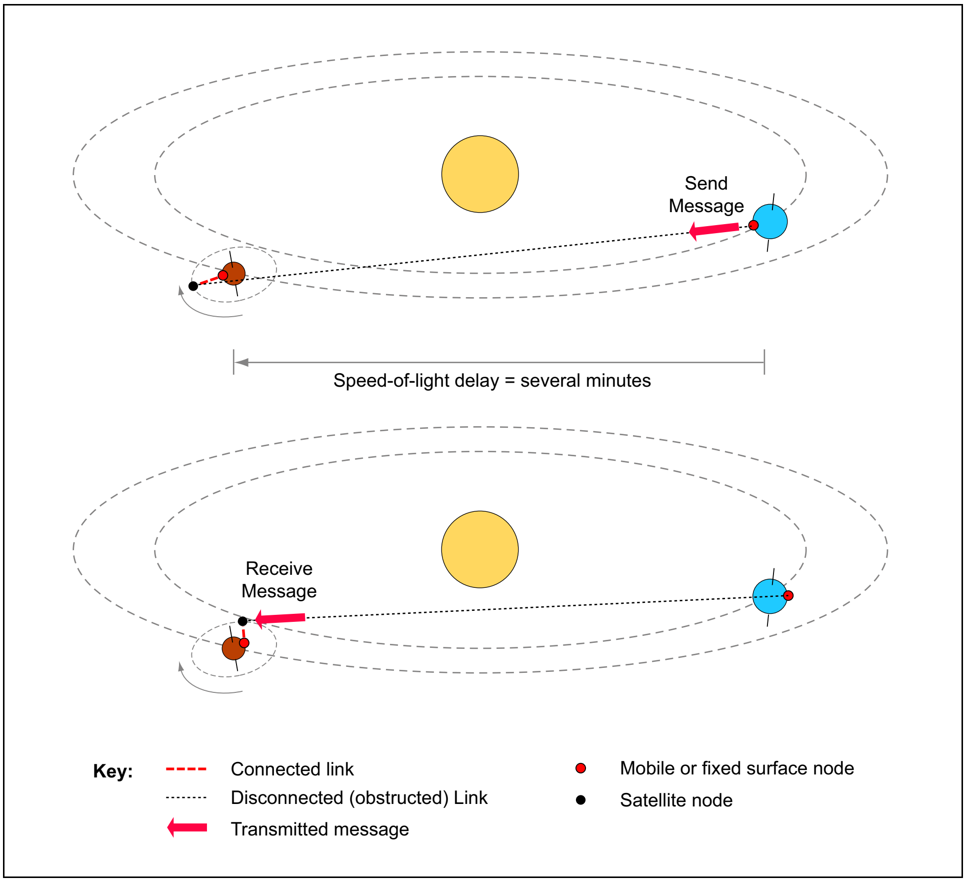


Scheduled Contacts

In space, almost everything is in motion and speed-of-light delays are significant (tens of minutes within our solar system). Potentially communicating nodes move along predictable orbital paths, so 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.

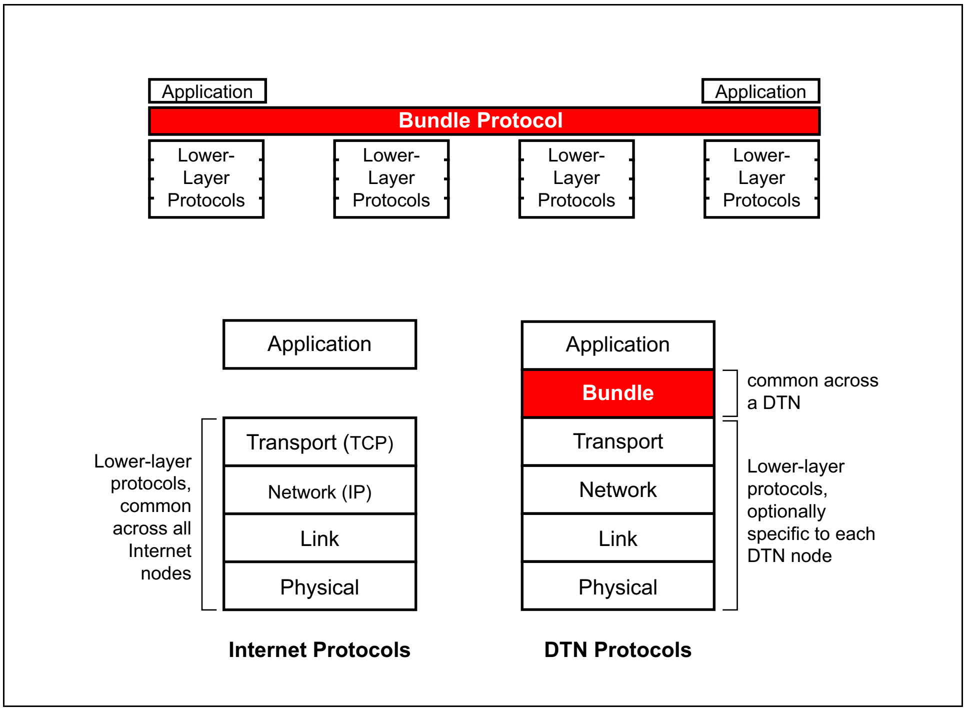

Send Message
Speed-of-light delay = several minutes
Receive Message
Key:
Connected link
Disconnected (obstructed) Link
Transmitted message
Mobile or fixed surface node
Satellite node

The Bundle Protocol

The DTN architecture implements store-and-forward message switching by overlaying a new transmission protocol—called the *bundle protocol*—on top of lower-lower protocols, such as the Internet protocols (page 7). The bundle protocol ties together the lower-lower protocols so that application programs can communicate across the same or different sets of lower-lower protocols under conditions that involve long network delays or disruptions.

The bundle-protocol agent stores and forwards entire bundles (or bundle fragments) between nodes. A single bundle protocol is used throughout a DTN. By contrast, the lower-lower protocols below the bundle protocol are chosen to suit the characteristics of each communication environment.

The figure below (top) illustrates the bundle-protocol overlay and (bottom) compares the Internet protocol stack (left) with a DTN protocol stack (right).

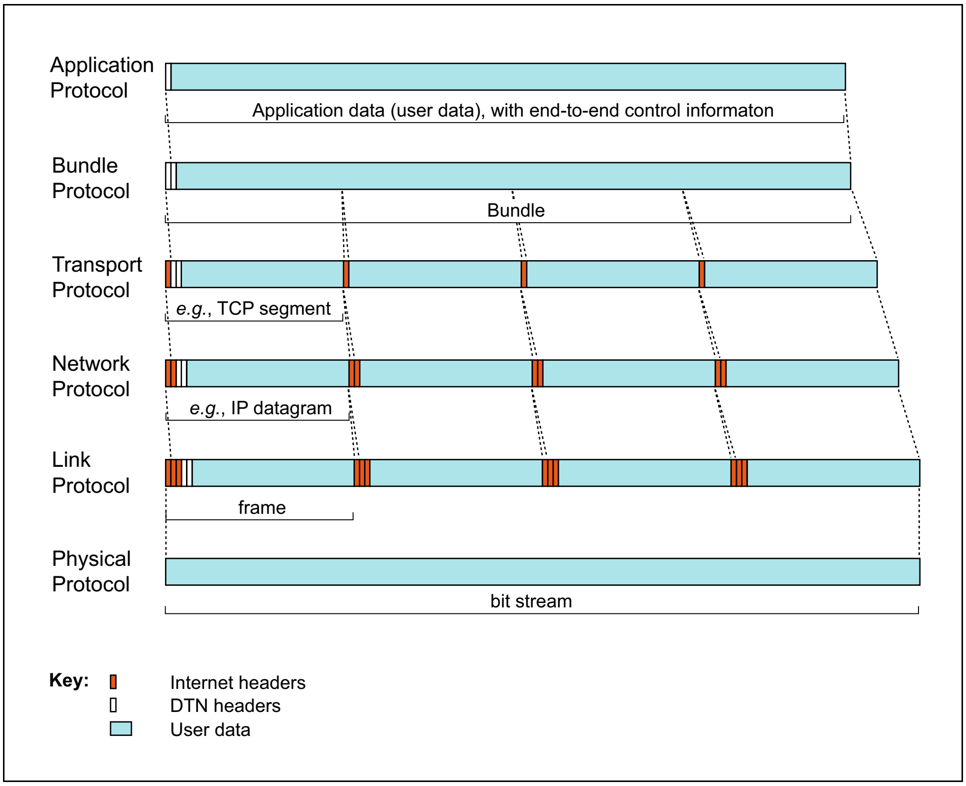


Bundles and Bundle Encapsulation

Bundles consist of three things: (1) a bundle header consisting of one or more DTN blocks inserted by the bundle-protocol agent, (2) a source-application's user data, including control information provided by the source application for the destination application that describes how to process, store, dispose of, and otherwise handle the user data, and (3) an optional bundle trailer, consisting of zero or more DTN blocks, inserted by the bundle-protocol agent (not shown in the figure below). Like application-program user data, bundles can be arbitrarily long.

Bundles extend the hierarchy of data-object encapsulation performed by the Internet protocols (page 7). The example below shows how bundle encapsulation works in the context of the lower-layer TCP/IP protocols. The bundle protocol does not alter the Internet-protocol data; it merely encapsulates the data.

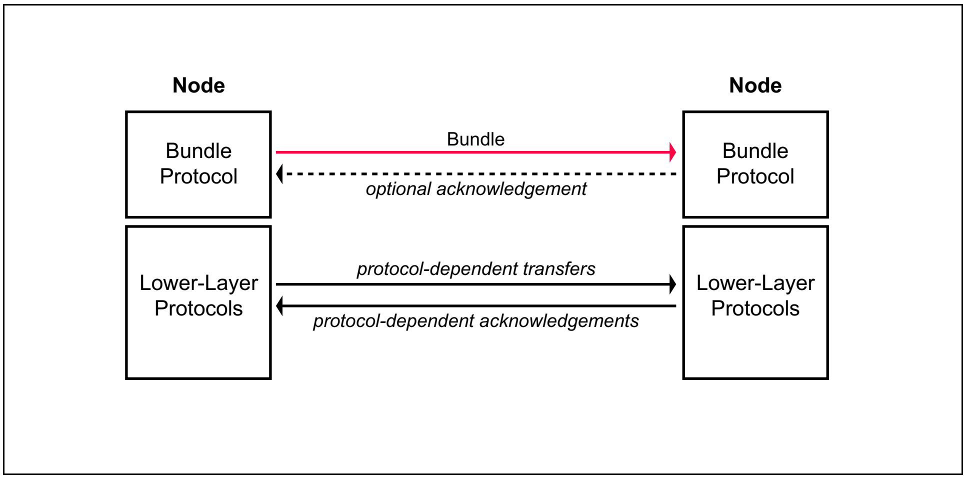
A bundle-protocol agent may break whole bundles into fragments (not shown in the figure below), just as the IP protocol may break whole datagrams into fragments. If bundles are fragmented, the bundle-protocol agent at the destination reassembles them.



A Non-Conversational Protocol

On intermittently connected links with long delays, conversational protocols such at TCP (page 9) that involve many end-to-end round-trips may take impractical amounts of time or fail completely. For this reason, DTN nodes 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 23).

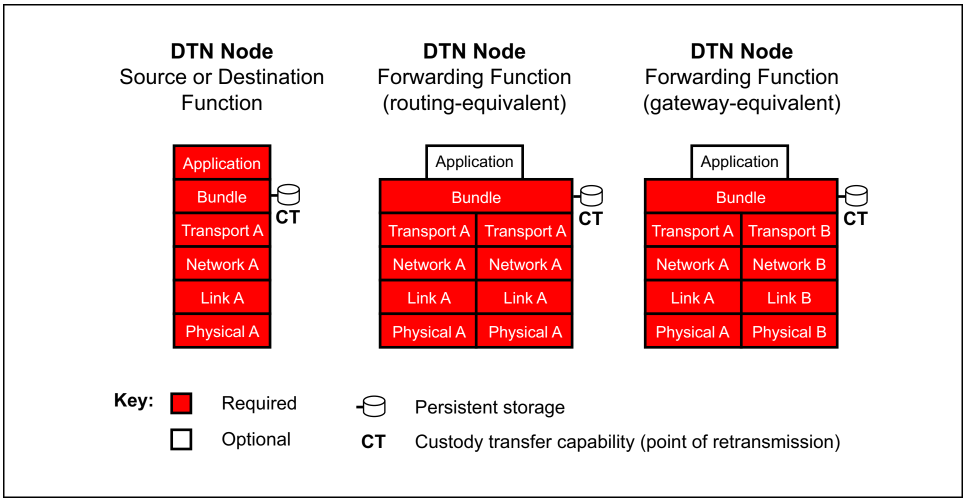
The lower-layer protocols that support bundle exchanges may, of course, be conversational like TCP. But on intermittently connected links with long delays, non-conversational or minimally-conversational lower-layer protocols may be more practical.



DTN Nodes

In a DTN, a *node* is an entity with a bundle-protocol agent overlaid on lower-layer communication protocols (page 15). At any moment, a given node may act as a source, destination, or forwarder of bundles:

- *Source or Destination Function* (figure below, left)—As a source or destination, a node sends or receives bundles to or from another node, but it does not forward bundles received from other nodes. If the node operates over long-delay links, its bundle protocol requires persistent storage in which to queue bundles until outbound links are available. The node may optionally support custody transfers (page 20).
- *Forwarding Function*—A DTN node can forward bundles between two or more other nodes in one of two situations:
  - *Routing-Equivalent Forwarding* (figure below, middle): The node forwards bundles between two or more other nodes, each of which implement the same lower-layer protocols as the forwarding node (the figure below, middle, shows these lower-layer protocols as type "A"). If a forwarding node operates over long-delay links, its bundle protocol requires persistent storage in which to queue bundles until outbound links are available. The node may optionally support custody transfers.
  - *Gateway-Equivalent Forwarding* (figure below, right): The node forwards bundles between two or more other nodes, each of which implement different lower-layer protocols while the forwarding node implements all such protocols (the figure below, right, shows these lower-layer protocols as types "A" and "B"). The node must have persistent storage; support for custody transfers is optional but typically advisable.

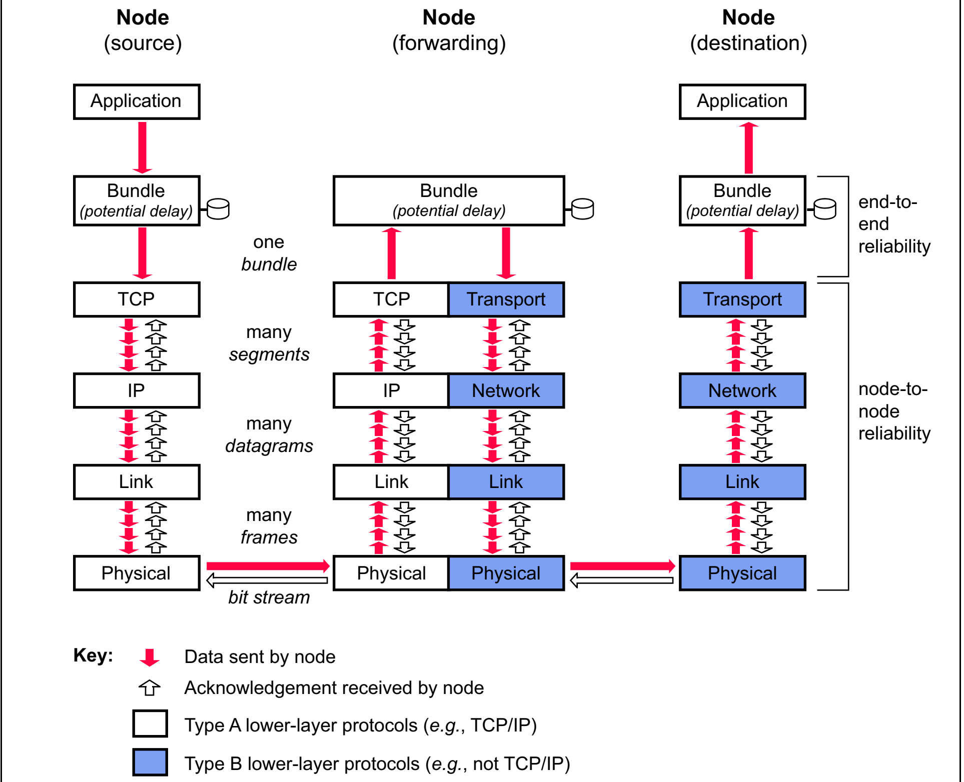


Delay Isolation via Transport-Protocol Termination

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 protocols provide other types of data-integrity services. In a DTN, the bundle protocol relies on these lower-layer protocols to insure the reliability of communication.

However, all DTN nodes terminate lower-layer transport protocols. The bundle-protocol agents thus act as surrogates for end-to-end sources and destinations. The beneficial side-effect is that conversational lower-layer protocols (page 9) are isolated by the bundle protocol from long delays elsewhere in the end-to-end path.

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



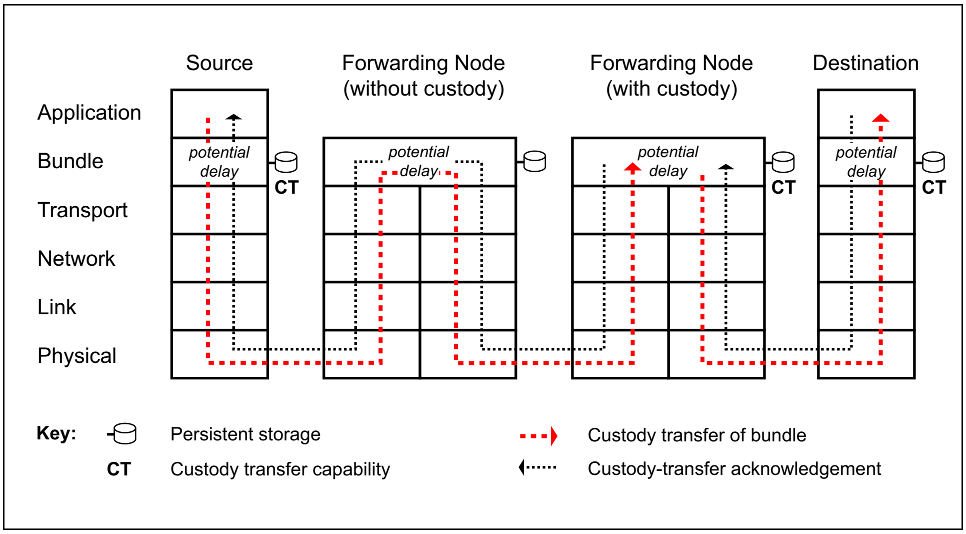
Custody Transfers

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

The bundle protocol supports node-to-node retransmission by means of *custody transfers*. Such transfers are arranged between the bundle-protocol agents of successive nodes, at the initial request of the source application. When the current bundle custodian sends a bundle to the next custodian (not necessarily the next node in the path), it requests a custody transfer and starts a time-to-acknowledge retransmission timer. If the next bundle-protocol agent 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 with 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.

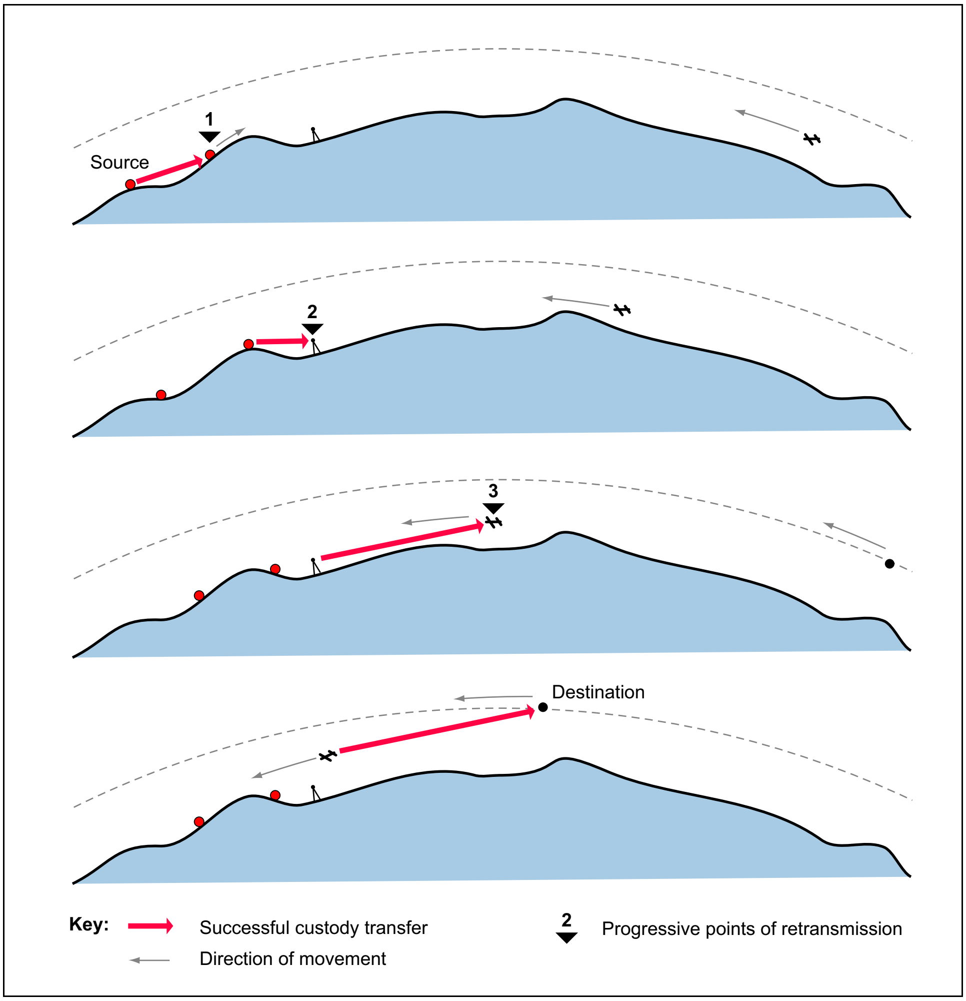
Custody transfers enhance end-to-end reliability, but they do not guarantee it. Further enhancement can be achieved by using both the custody transfer and return receipt services (page 23).



Moving Points of Retransmission Forward

The bundle-protocol agent uses reliable underlying transport protocols together, optionally, 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 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).

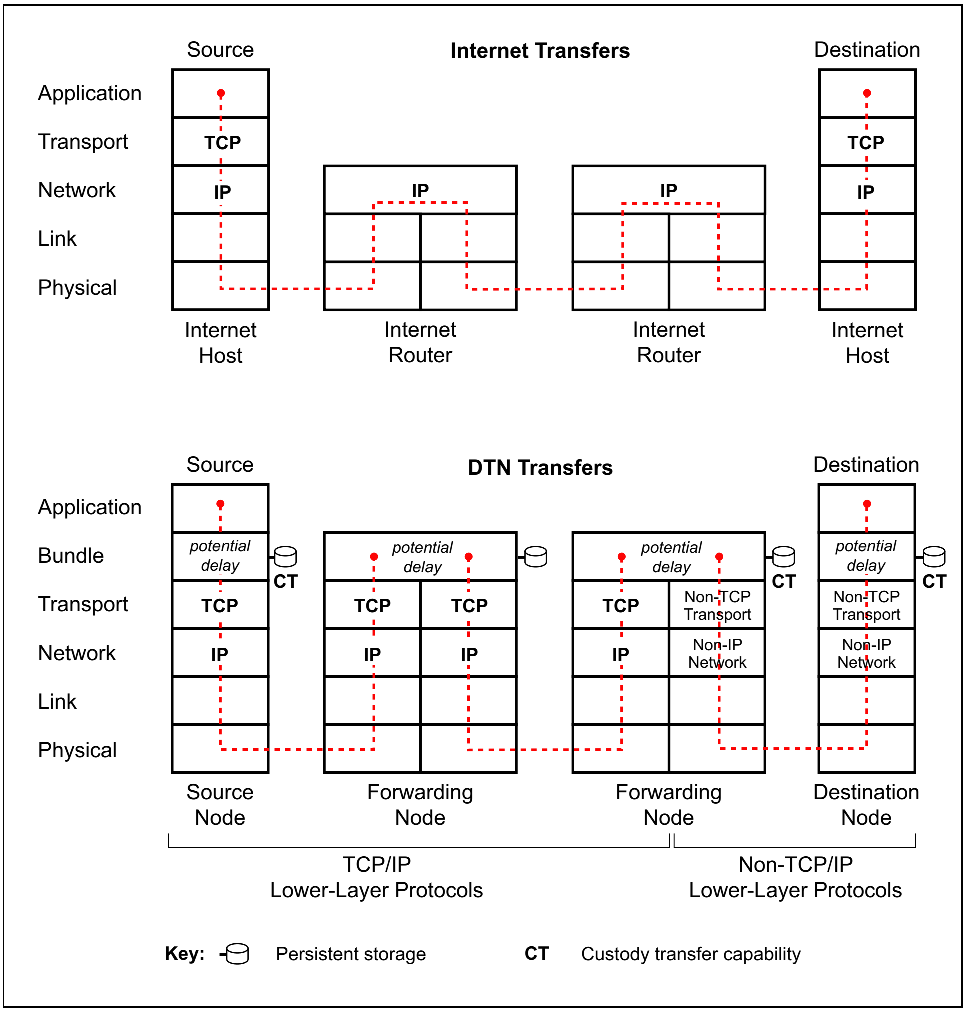


1
Source
2
3
Destination
Key:
Successful custody transfer
Direction of movement
2
Progressive points of retransmission

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 TCP segments (page 7). IP operates at all nodes on the path, where it routes IP datagrams.

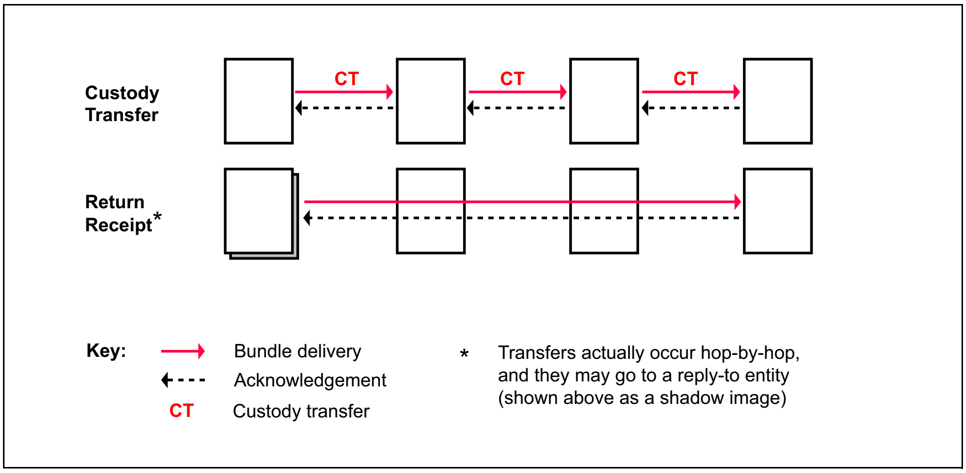
In a DTN, all nodes implement both the bundle protocol and a lower-layer transport protocol. Nodes that forward bundles can implement either the same or different lower-layer protocols on either side of the forwarding; in this respect, their functions are comparable to Internet routers or gateways, respectively.



Classes of Bundle Service

The bundle protocol provides six classes of service for a bundle:

- *Custody Transfer:* Delegation of retransmission responsibility by one node to another accepting node, so that the first node can recover its retransmission resources. The accepting node returns a custodial-acceptance acknowledgement to the previous custodian (page 20).
- *Return Receipt:* Confirmation by the destination to the source, or its reply-to entity, that the bundle has been received by the destination application. Reception by the source, or its reply-to entity, of the return receipt provides end-to-end assurance of delivery.
- *Priority of Delivery:* Bulk, Normal, or Expedited (not shown in the figure below).
- *Time-to-Live:* (not shown in the figure below).


Custody Transfer
CT
CT
CT
Return Receipt*
Key:
Bundle delivery
Acknowledgement
CT
Custody transfer
* Transfers actually occur hop-by-hop, and they may go to a reply-to entity (shown above as a shadow image)

Endpoint IDs

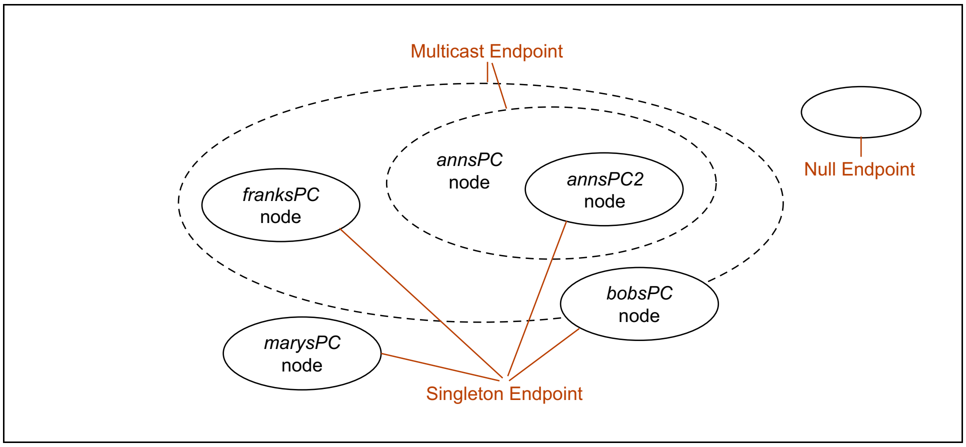
A bundle *endpoint* is a set of zero or more nodes that all identify themselves by the same *endpoint ID*. The common case in which only one node has a given endpoint ID is called a *singleton endpoint*. Every node is uniquely identified by at least one singleton endpoint. Source nodes are always singleton endpoints or null (anonymous source) endpoints, and destination nodes may or may not be singleton endpoints. Endpoints may also be multicast (multiple destination nodes with the same endpoint ID) or null (no nodes). Endpoints may contain multiple nodes, and nodes may be members of multiple endpoints, as shown in the figure below.

An endpoint ID is a uniform resource identifier (URI) text string using the syntax `<scheme_name>:<scheme-specific_part>`, in which the scheme name is either *dtn* or *ipn*. The scheme-specific part comes in two flavors—application-specific, used to identify a source or destination node, and administrative, used when forwarding bundles from node to node. Here are some examples:

▪ `dtn://bobsPC/files`	(application-specific)
▪ `dtn://bobsPC/`	(administrative)
▪ `ipn:81.2`	(application-specific)
▪ `ipn:81.0`	(administrative)

The *dtn* scheme is the original scheme, designed for terrestrial and interplanetary networks, and the *ipn* scheme was added later specifically for interplanetary networks. But either scheme can be used in either context. The *ipn* scheme has interplanetary advantages in that it supports economical compression of bundle-header encoding, per RFC 6260.

Endpoints do not have physical location. They are simply a set of nodes that may be the source or destination of bundles. See RFC 5050 for details.



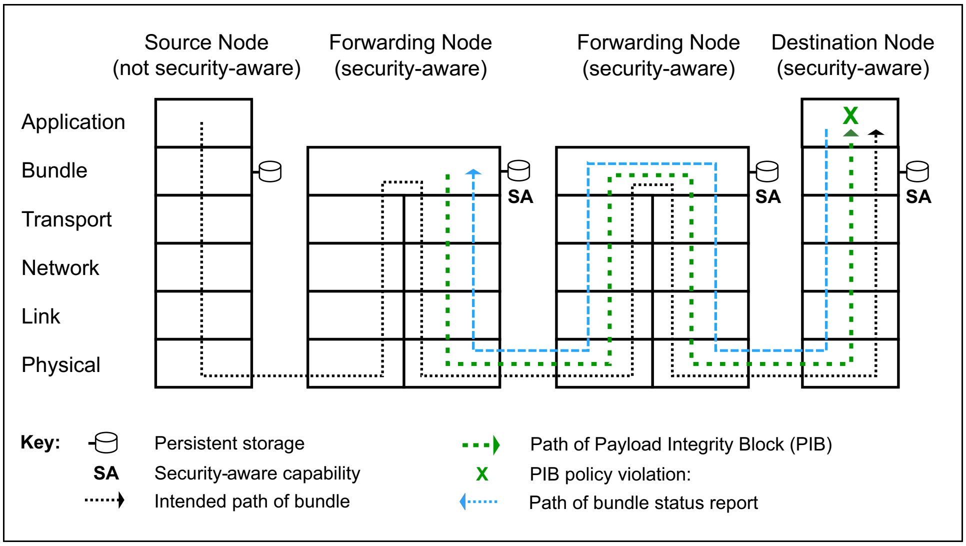
The Bundle Security Protocol (BSP) provides data integrity, authentication, and confidentiality. Only security-aware (SA) nodes with the capacity to originate and process BSP security blocks provide these services. Processing may require some action on a bundle—which may include discarding it—to conform with that node's security policy.

The BSP defines four types of security blocks:

- *Bundle Authentication Block (BAB):* Ensures bundle authenticity and integrity along the path from forwarding SA node to the next receiving SA node. If a BAB is used, its header must be the last header applied to a bundle.
- *Payload Integrity Block (PIB):* Ensures integrity of payload by verifying the signer.
- *Payload Confidentiality Block (PCB):* Encrypts bundle payload.
- *Extension Security Block (ESB):* Provides security for non-payload blocks.

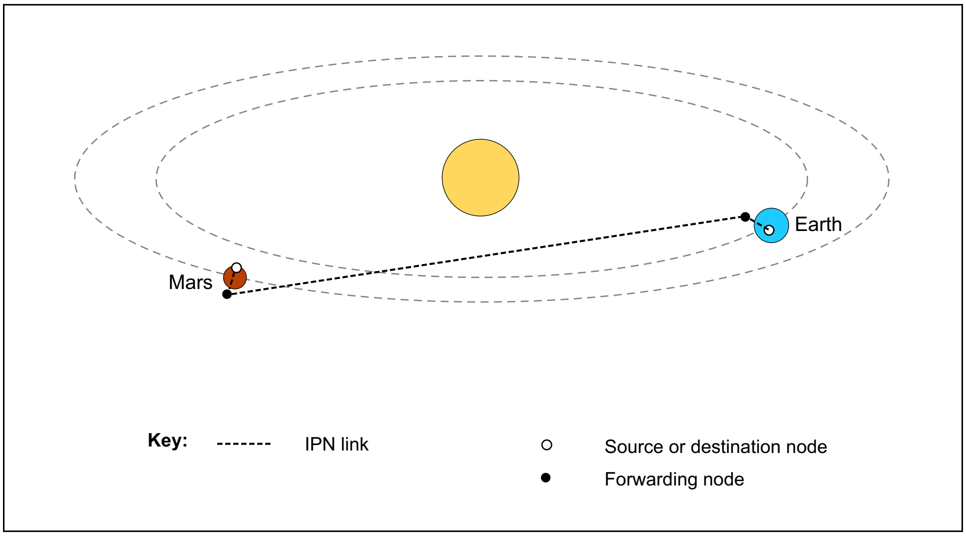
The default security policy for SA nodes requires a BAB. If an SA node has no security policy, only a BAB is required. By default, a BAB policy violation at an SA receiving node requires deletion of the bundle. This default BAB requirement is intended to protect the DTN from DDoS attacks by ensuring the maintenance of trusted relationships between adjacent SA nodes. See RFC 6257 for details.

The figure below shows the path of a PIB block. If a PIB policy violation occurs at an SA receiving node (the green X), the policy determines the node's actions, which may include the sending of a bundle status report to the originating node, as shown in the figure.


Source Node (not security-aware)
Forwarding Node (security-aware)
Forwarding Node (security-aware)
Destination Node (security-aware)
Application
Bundle
Transport
Network
Link
Physical
SA
SA
SA
X
Key:
Persistent storage
SA
Security-aware capability
Intended path of bundle
Path of Payload Integrity Block (PIB)
X
PIB policy violation:
Path of bundle status report

An Interplanetary (IPN) Example

The Internet Society's IPN Special Interest Group's InterPlaNetary (IPN) Internet, described at *http://www.ipnsig.org*, is a DTN that uses the *ipn* scheme name. The next six pages show how a message might be sent from a source on Earth to a destination on Mars, using two intermediate forwarding nodes. All nodes are identified by their associated endpoint IDs.



The forwarding decisions are made on a node-by-node basis. Nodes do not have IDs, but all nodes have at least one singleton endpoint ID. The table below shows the nodes, and their singleton endpoint IDs, that are accessed in the figure above and the pages that follow. The endpoint IDs that end in zero (e.g., ipn:81.0) are administrative IDs, used in forwarding.

Node	Endpoint IDs	
Earth Source	`ipn:81.2`	(application-specific ID)
Earth Forwarding	`ipn:81.0` `ipn:49.0`	(administrative ID) (administrative ID)
Mars Forwarding	`ipn:49.0` `ipn:65.0`	(administrative ID) (administrative ID)
Mars Destination	`ipn:65.7`	(application-specific ID)

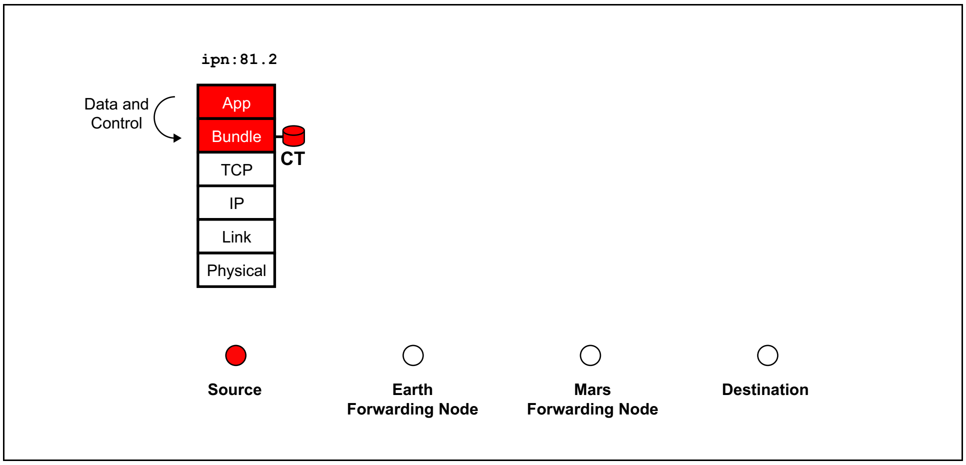
Before transfers begin, and on an on-going basis, the bundle-protocol agents of all network nodes synchronize time among themselves. This is needed for consistent calculation of contact schedules and bundle time-to-live throughout the DTN.

Step 1: Bundle Creation at Source

The source application invokes its bundle-protocol agent, 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-protocol agents that handle the transfer.

Item	Value
Source	`ipn:81.2`
Destination	`ipn:65.7`
Class of service	• Custody transfer • Normal priority • Time-to-live = 36 hours
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-protocol agents.

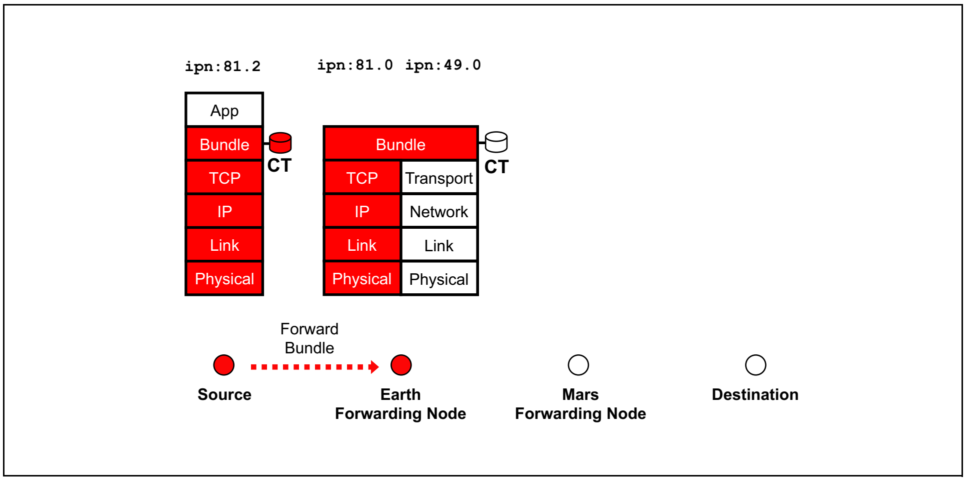
The source bundle-protocol agent creates a bundle and stores the result in persistent storage. The storage is required, even if an immediate forwarding opportunity exists, because the bundle-protocol agent has accepted a *custody transfer* and must therefore be prepared to retransmit the bundle if it does not receive acknowledgement, within the bundle's time-to-acknowledge (page 20), that the subsequent custodian has received and accepted the bundle.



Step 2: Transmission by Source

The source bundle-protocol agent consults its forwarding table and finds that the Earth forwarding node `ipn:81.0` is the next hop capable of accepting custody transfers on a path toward the destination, and that TCP is the proper transport protocol for this low-Earth-orbit node. The source bundle-protocol agent also determines that it has a continuous connection to the Earth forwarding node.

The bundle-protocol agent transmits a copy of the bundle to the Earth forwarding node via TCP, starts a time-to-acknowledge retransmission timer (page 20), and awaits a custody-transfer acknowledgment from the forwarding node.



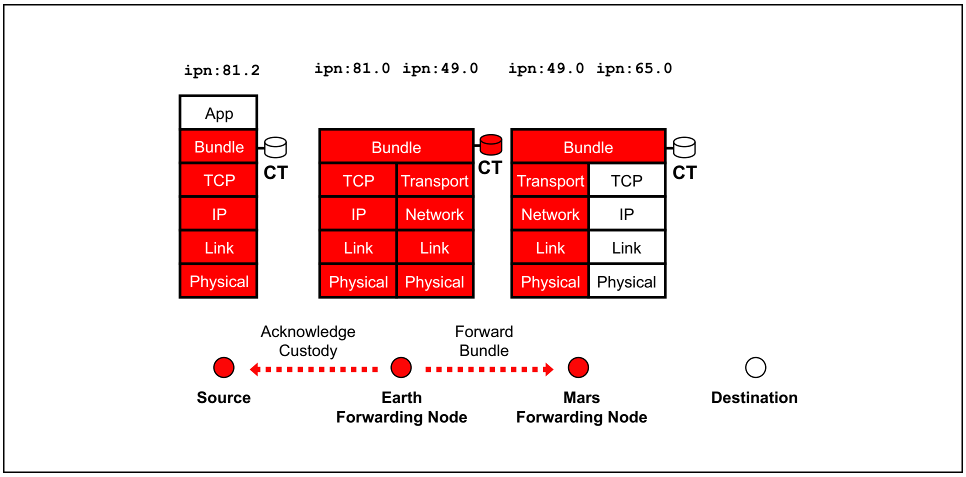
Step 3: First-Hop Bundle Processing and Forwarding

When the bundle-protocol agent at the Earth forwarding node receives the bundle via TCP, it may terminate the TCP session (page 19). Then it stores the received bundle in persistent storage.

The bundle-protocol agent at the Earth forwarding node consults its forwarding table and finds that the Mars forwarding node `ipn:49.0` is the next hop capable of accepting custody transfers on a path toward the destination. It determines that the Mars forwarding node will be accessible at 1100 the following day, confirms that the bundle's time-to-live (page 27) is suitable for this hop's delay, and adds the bundle to its list for forwarding to that hop.

The bundle-protocol agent at the Earth forwarding node then accepts custody of the bundle, updates this information in the bundle header, and confirms this by acknowledgement to the source bundle-protocol agent, which deletes its custodial copy of the bundle.

At the next-hop contact time, the bundle-protocol agent at the Earth forwarding node establishes contact with the bundle-protocol agent at the Mars forwarding node via the appropriate long-haul transport protocol and forwards the bundle.



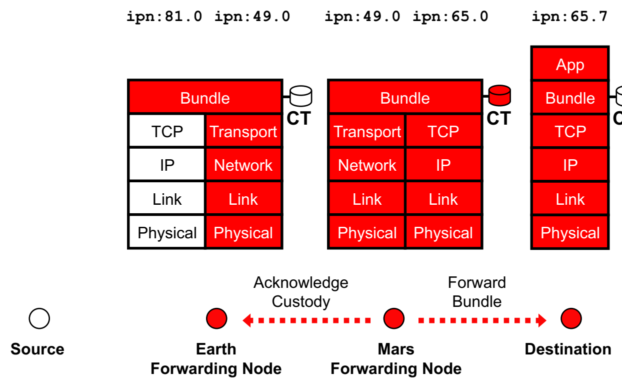
Step 4: Second-Hop Bundle Processing and Forwarding

When the bundle-protocol agent at the Mars forwarding node receives the bundle, the agent stores the received bundle in persistent storage.

The bundle-protocol agent at the Mars forwarding node consults its forwarding 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 27) is suitable for this hop's delay.

The bundle-protocol agent at the Mars forwarding node then accepts custody of the bundle, updates this information in the bundle header, and confirms this by acknowledgement to the bundle-protocol agent at the Earth forwarding node, which deletes its custodial copy of the bundle.

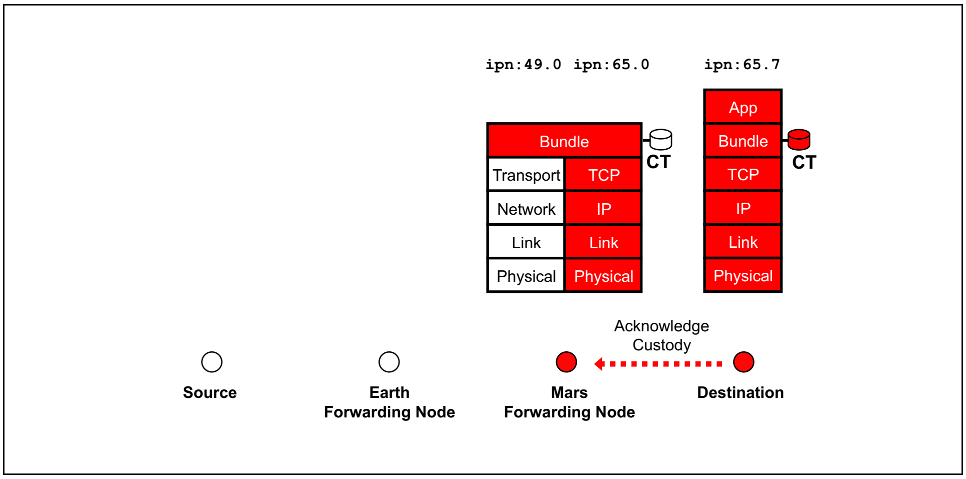
The bundle-protocol agent at the Mars forwarding node then establishes contact with the destination bundle-protocol agent via TCP and forwards the bundle.



Step 5: Bundle Reception by Destination

When the destination bundle protocol receives the bundle via TCP, it may terminate the TCP session. Then it stores the received bundle in persistent storage, accepts custody of the bundle, and confirms this by acknowledgement to the bundle protocol at the Mars forwarding node, which deletes its custodial copy of the bundle.

The destination bundle-protocol agent then awakens the destination application identified by the `ipn:65.7` endpoint ID. Depending on the control part of the user data sent by the source, the destination application may generate an application-protocol acknowledgment in a new bundle and send it to the source.



Potential Applications of DTN Technology

The DTN store-and-forward message switching architecture is a generalization of work originally conceived to support the InterPlaNetary Internet (IPN). The primary goals are interoperability across network environments, and reliability capable of surviving hardware (network) and software (protocol) failures.

Although DTNs were originally conceived for interplanetary use, they may have a far greater number of applications on Earth. Here is a short summary of the possible applications:

- Space Agencies: International Space Station communication (currently operational for research), interplanetary communication, future space-debris monitoring.
- Military and Intelligence: Mobile ad-hoc networks (MANETs) for wireless communication and monitoring, cargo tracking, search and rescue communication, unmanned aerial vehicle (UAV) communication and control.
- Commercial: Cargo and vehicle tracking (by road, rail, sea, and air), in-store and in-warehouse asset tracking, data transactions (e.g., financial, reservations), agricultural crop monitoring, processing-plant monitoring, communication in underground mines.
- Public Service and Safety: Security and disaster communication, search and rescue communication, humanitarian relief monitoring, smart-city event-response, smart transportation networks, smart electric-power networks, global airport-traffic control, infrastructure-integrity monitoring, unmanned aerial vehicle (UAV) communication and control, remote learning.
- Personal Use: Personal monitoring and communication in wilderness and urban areas, fire-and-forget text messaging.
- Environmental Monitoring: Animal migration, soil properties and stability, atmospheric and oceanographic conditions, seismological events.
- Engineering and Scientific Research: Network subject-matter experts, academic research by faculty and students.

References and Bibliography

The Internet Research Task Force's Delay-Tolerant Networking Research Group (DTN-RG), *http://www.dtnrg.org.*

The InterPlaNetary (IPN) Internet Project, described on the Internet Society's IPN Special Interest Group's site, *http://www.ipnsig.org.*

S. Burleigh, *Compressed Bundle Header Encoding (CBHE)*, RFC 6260, http://www.ietf.org, May 2011.

K. Scott, S. Burleigh, *Bundle Protocol Specification*, RFC 5050, http://www.ietf.org, November, 2007.

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