2020 6.033 Design Project:

Network Storage Subsystem for ExtraNet

See also DP FAQ and DP Errata

Due Dates and Deliverables

There are four deliverables for this design project:

1) **DP Preliminary Report (DPPR)**: This preliminary report will lay out your key design decisions, including both a functional system design and a sketch of any data structures, storage management, and/or network protocols required to achieve your design. It will not include any significant evaluation. It will be approximately 2,500 words and is due March 20, 2020 at 5:00pm.
2) **DP Presentation**: This presentation will address the feedback received on the DPPR, and any corrections or updates to the design project specification. It will also outline evaluation criteria and use cases you will use later for evaluating your design. The presentation will occur during the week of April 13-27.
3) **DP Report (DPR)**: This will be your full report. It will include your final design, all diagrams appropriate for that, your evaluation of your design and a review of how effectively your design addresses the specified use cases. It will be approximately 6,000 words and is due May 4, 2020 at 11:59pm.
4) **Peer Review**: In Tutorial your team will have done an early "review," providing informal feedback to another team on their design. For this peer review, you will individually review a few specific sections of that (same) other team's final report and will address some specific questions about that report. It will be approximately 250 words and is due May 8, 2020 at 5:00pm.

Your assignment for each of the four parts above will be distributed in separate "assignment" documents.

The preliminary report, final report, and peer review should be submitted via the submission site on the 6.033 website. As with real-life system designs, the 6.033 design project is under-specified, and it is your job to complete the specification in a sensible way given the stated requirements of the project. As with designs in practice, the specifications often need some adjustment as the design is fleshed out. Moreover, requirements will likely be added or modified as time goes on. We recommend that you start early so that you can evolve your design over time. A good design is likely to take more than just a few days to develop. A good design will avoid unnecessary complexity and be as modular as possible, enabling it to evolve with changing requirements.

Large systems are never built by a single person. Accordingly, you will be working in teams of three for this project. Part of the project is learning how to work productively on a long-term team effort. **All three people on a team must be in the same tutorial.**

Although this is a team project, some of the deliverables have individual components. See the individual assignment links for more information.

Late submission grading policy: If you submit any deliverable late, we will penalize you one letter grade per 48 hours, starting from the time of the deadline. For example, if you submit the report anywhere from 1 minute to 48 hours late and your report would have otherwise received a grade of A, you will receive a B; if you submitted it 48 hours and 5 minutes late, you will receive a C.
As stated on the syllabus, you must complete every design project component to pass 6.033.

Space network communication:
The network storage subsystem for ExtraNet

Update: April 29, 2020

1. Introduction

Your team in the space program is taking on a design role for part of the new (slightly fictional) **ExtraNet** project, the extraterrestrial network being design by NASA. ExtraNet will be carrying three categories of data: telemetry (collected data), management (control data for satellites and other objects, routine activities, etc.), and mission critical information to avoid various forms of catastrophes. Both the requirements of these kinds of data and the extraterrestrial communications limitations will play major roles in your design. You will be designing what we call a network storage system to support all this on contract to NASA. For purposes of this study, it is assumed that all elements of the network are under the sole control and management of NASA.[1]

To date all extraterrestrial communication has been directly point to point between terrestrial points and extraterrestrial devices. That means that the path taken by information between Earth and these devices is a direct path (single hop) between the extraterrestrial device and the ground. Examples of the sorts of extraterrestrial devices range from sensors and telescopes collecting data, and rovers, to routine operational information and support systems perhaps housed in antennas, to the devices that must send and receive mission critical information used by both manned and unmanned missions. The remote devices can be as close as the space station, or further to the Moon, Mars, and possibly further out in the Solar System. The ExtraNet plan is that traffic may move through the network of satellites in other paths and in some situations, such as between the Moon and Mars, may not ever reach the Earth.

For communications, this leads to a collection of challenges not generally present in terrestrial communications. The particular challenges include:

- Extreme variability in roundtrip times[2] between end points;
- Extremely long roundtrip times in some cases (e.g. the round trip at the speed of light between Earth and Mars varies between 8 and 40 minutes);
- Intermittent connectivity, both predictable and unpredictable;
- The need to take advantage of scheduled and predicted connectivity opportunities;[3]
- Traffic errors due to a wide variety of influences, such as noise, not only packet loss as in terrestrial networks:
- Extremely limited resources such as bandwidth[4] and power on occasion.

[1] This is a simplification of reality because both there are commercial communications operations occurring and other national and international organizations. We are assuming this is a system solely under the control of NASA.
[2] Roundtrip time is the amount of time it takes for a message to be sent and a response sent back.
[3] For example, an intermediate node may not be able to forward a bundle directly, but might have knowledge that its nearest neighbor will be able to later, so it may pass traffic to that neighbor for future forwarding.
[4] Bandwidth is the maximum rate at which data can be transferred. In our case, different devices will support different bandwidths.

To address these problems the networking community has defined the Bundle Protocol, which includes a requirement for network storage, but no proposed design for it. There is more detail on the Bundle Protocol itself in later sections of this document.

As further background, in reality, NASA, at this time, is designing and building a less ambitious version of what is proposed.[5] Our ExtraNet extended specification will consist of a set of satellites to support the NASA missions not only to the Moon (both uncrewed and crewed), but is also intended to support missions to Mars and beyond. The network will consist of various types of satellites, probably both Low-Earth Orbit (LEO) and higher-orbit geosynchronous (GEO) satellites. For simplicity, as you will see below, all the LEO satellites will be assumed to be at a single altitude and all the GEO satellites at a single much higher altitude. In addition, their design may include relay nodes to support more challenging situations, such as the antenna on the Moon being on the far side (see Figure 1 below). One of the key features of this design is that the Bundle Protocol (BP) is designed to be a store-and-forward protocol that can handle both long and unpredicted delays in communication without the expense of either retransmission or disruption of communication. In this store-and-forward protocol a satellite or node in the network may hold a "bundle" of data for as long as needed to be able to forward it toward its destination. For purposes of this project, the design of ExtraNet and some of the details of the whole Bundle Protocol and LunaNet have been changed. Thus, although the BP specifications are widely available, take what is described here as authoritative for this project, rather than what you may find in other papers, documents, websites, etc.

Although the Bundle Protocol is described as a store-and-forward protocol, only the forwarding aspect of it is well defined, but the storage aspect is not. Therefore, your task is to design the network storage system to support the overall design needed for the network nodes to execute the protocol, as we will discuss further below. You will be designing a distributed system, but in order to do that you will first need to design the individual nodes of the system[6], and then the distributed coordination among the nodes. Below you will find more details about the network design, some of the specifications for the network storage system, and a set of use cases which must be supported.

2. Background: The ExtraNet Design

In order to set the stage and examine the constraints on your design, we first examine the ExtraNet design as a whole, beginning initially with the hardware and its constraints and then several key software elements that run in that hardware environment, primarily the communications protocol.

A. The Satellites and other hardware

ExtraNet will consist of a set of 100 LEO satellites, a set of 10 GEO satellites at significantly higher altitude, a small number of relay stations, near other solar system bodies and Earth, in our initial case the Moon and Mars, and a set of antennas on the surfaces of the Moon and Mars. Initially, we are assuming one combined antenna and surface system on each. In addition, some data will originate on

[5] The NASA program is to design and build LunaNet, based on the specification and their implementation of the Bundle Protocol. We point this out, because these are real specifications, designs, and implementations.
[6] We suggest that you design the elements of your system in this order, because as of the time of this assignment, we will have mostly discussed only singular centralized systems. As we get to networking and more distributed computational models, you will be better able to address the distributed portions of this design.

the GEO satellites, which will include cameras, telescopes, and other measurement instruments focused on Earth and the Sun. Figure 1 below is a diagram conceptual design and not to scale. Every satellite and relay will have significant long-term storage. Your challenge is to design the organization of that storage, both on the individual nodes, and collectively among the nodes, to improve communication over the simple model in use currently/ The current simplistic approach assumes an extremely simple local storage model and no cooperation or collaboration among the nodes' storage systems. The project is based on the hypothesis that communication using the Bundle Protocol can be improved along a number of metrics, and hopefully with a low overhead cost to doing that.

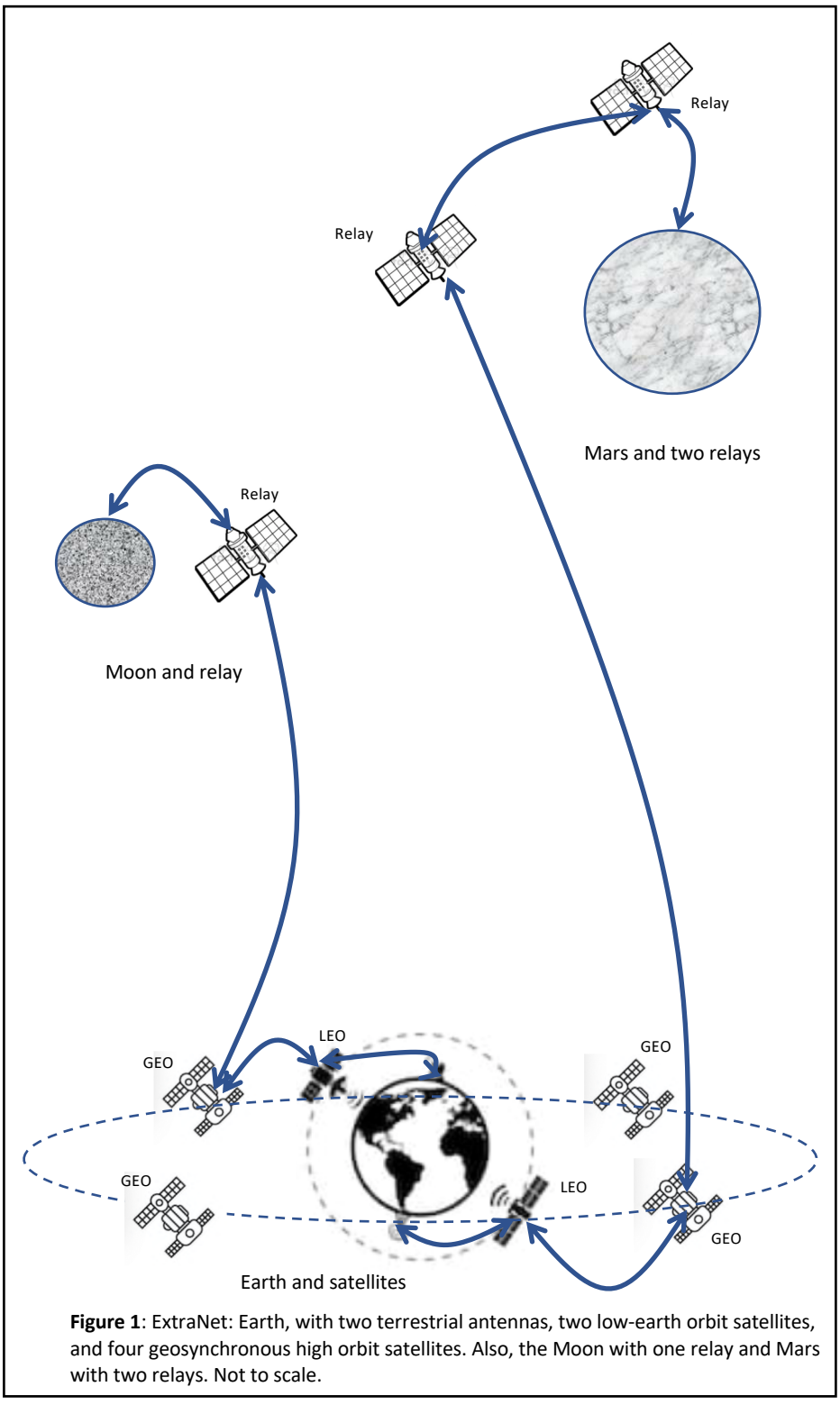


Figure 1: ExtraNet: Earth, with two terrestrial antennas, two low-earth orbit satellites, and four geosynchronous high orbit satellites. Also, the Moon with one relay and Mars with two relays. Not to scale.

In ExtraNet, the LEO satellites are the only ones that have direct communication with Earth antennas. They also can communicate with their nearest LEO neighbors at any given time and with the GEO satellites. In our design, the LEO satellites are in Polar orbits, each of which takes 1.5 hours. At this rate, with the Earth rotating underneath them, each LEO satellite will return to the same location over the Earth once every 24 hours. These satellites are all at an altitude of 833Km. LEO satellites will be able to

communication with antennas on the Earth, their nearest neighbor LEO satellites intermittently, and with different of the GEO satellites at different times, depending on their relative orbits.

The GEO satellites will be in orbits on the Equator, at about 36,000Km from Earth, but are geosynchronous, so they orbit once every 24 hours in sync with the Earth. These satellites, in addition to communicating with the LEO satellites will be able to communicate again with their own nearest neighbors in the set of GEO satellites as well as any relays and in some cases directly with antennas and their associated systems on the Moon, Mars, and possibly later on other more remote bodies. Remember that we use the term *relay* to identify elements of our network that are not specifically orbiting around Earth. The relays may only be needed for some Lunar communications, such as, if the Moon-based antenna is out of direct communication with the GEO satellites. For Mars and more remote communication, the relays will be needed between those remote locations and the GEO satellites. All non-terrestrial nodes will communicate via optical links, which have the characteristic that they are directly from one node to another (point-to-point).[7]

The system specification includes the following constraints on bandwidth, orbit times, the number of each kind of device, and the amount of storage available on each device.

Device Type	**Bandwidth capacity**	**Orbit time**	**Number of devices**	**Amount of storage**
Earth Antenna	2GB/s	-	25	10TB
LEO	2GB/s	1.5 hrs	100	.5TB
GEO	1.2GB/s	24 hours (stationary)	10	.5TB
Relay	622MB/s	None	4 (2 each at Moon/Mars)	.5TB
Moon/Mars Antenna	622MB/s	Revolution time of the body	1 each on Moon and Mars	10TB

Table 1: The physical modules of the system and their specifications.

These bandwidths are in each direction, because our underlying medium and technology is optical. This leads to other communications constraints among the nodes that may be valuable to you:

- At any given time, there is a 100% probability that at least one LEO satellite is communicating with a ground antenna. There is a 50% probability that two are communicating with the ground and a 25% chance that three are, and so forth.
- 90% of the time a LEO satellite can reach one of the GEO satellites. When the LEO satellites are at the poles, their communication may be severely disrupted.
- At each of the Moon and Mars, there are two relays, one that is always in contact with the antenna there and the other that is in contact with the GEO satellites. So there is 100% connectivity there. There will be times when this will require moving the relays with respect to both the Moon or Mars and the Earth satellites in order to maintain connectivity. It is important to remember that although communication between the Moon region and Earth only takes a few seconds at the speed of light, between Mars and Earth that time stretches to between about 4 and 20 minutes <u>each way</u> depending on where in their orbits Mars and the Earth are.

[7] The reason for this is that the optical communication used for these networks is directional, whereas radio waves, as used in much terrestrial communication, are non-directional. For space communication, energy and therefore power are critically important. So, although we are not considering power in this design project, optical communication is the technology being used for long distance communication.

Although these are the numbers at present, NASA hopes to deploy more of each type of device over time. This means that your design should allow for expansions in the capacity of the system.

B. Other important elements of the system

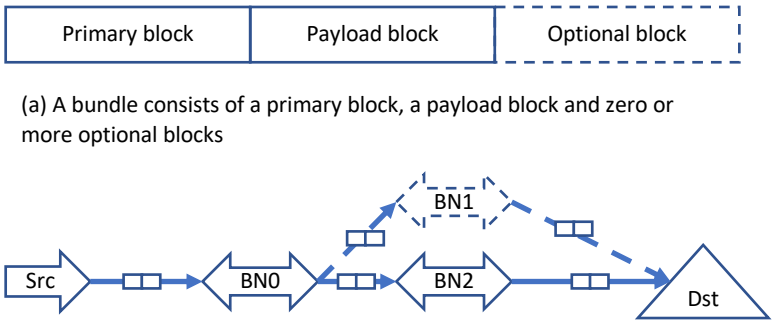
There are two key elements of the system that you will depend on, but will not be designing, the Bundle Protocol and the routing protocol. We describe each of them here briefly. The Bundle Protocol moves data from a source to a destination. The Routing Protocol informs each node that participates in the Bundle Protocol about where to send the traffic as a next step, in order to make progress toward the destination. In addition, it informs each satellite of its two nearest neighbors in its satellite category (either LEO or GEO). These are fictionalized a bit for the purposes of this project. There exist full specifications and implementations of each, but use our specification of them here for your design. Your assignment is to focus on the network storage system, not these protocols.

1) The Bundle Protocol

In this section, we will begin with a simple example to demonstrate a few of the key features of the Bundle Protocol. We will then look at what a bundle is. That will be followed by a description of the Bundle Protocol itself. It is important for you to both understand that the whole Bundle Protocol is a given in your project and for you to consider how it may provide both opportunities and constraints in your own design of your part of the system.

A. Introduction to the Bundle Protocol: An example

Let us begin to understand the Bundle Protocol (BP) through a simple example, as depicted in Figure 2. Below, with the example in mind, we will explain the terminology. It all begins with an application that has a chunk of data called an application data unit (ADU) to send to some other place. For this example, let's assume that this is a piece of email containing birthday wishes for an astronaut on the Moon. The email protocol does not guarantee that all email is delivered. Email messages may be lost; it is a "best effort" protocol. On Earth the BP running in the sender's computer will receive the data from the email program, arrange it into a bundle (the set of bits to be sent, plus any additional information required for sending) to be sent, and forward it to the first bundle



(a) A bundle consists of a primary block, a payload block and zero or more optional blocks

(b) Bundles are stored and then forwarded from the source (Src) through a series of bundle nodes (BN) to the destination (Dst). A bundle may follow more than one path or be fragmented, in which case the fragments may follow different paths. This example has only bundles with no optional blocks.

Figure 2: Example of simple, best-effort delivery of bundles. The Bundle protocol forwards each bundle through a series of bundle nodes, each with network storage, toward the destination. It may try more than one path. It may also fragment the bundle and send the fragments on a variety of paths. Bundle protocol will reassemble the full bundle in either an interior bundle node or at the destination. This example has no responses or acknowledgements.

node, perhaps a LEO satellite, BN0. In the simplest case, the LEO satellite will hold the bundle in its storage until a GEO satellite (BN2) comes into communication, when it will forward it along. That satellite may again store the bundle until the Moon-based antenna is available. It is also possible that the LEO satellite finds that it can reach two GEO satellites (BN1 and BN2) because they are both within the direct communication distance and sends the bundle to both, to increase the odds of it being delivered sooner. How this information is known is the task of the Routing Protocol, which will be discussed below. You can assume that the routing table on each Bundle Node, managed by the routing protocol, will tell you which nodes are within communication each node at any given time.

Below we describe the key aspects of the BP. We begin with the design of a bundle, including its format and the types of bundles. We then discuss the core of the protocol itself. It is important to remember here that the Bundle Protocol as described in the next two subsections is part of what you are provided. Your task is to support the protocol as best you can and to utilize features appropriately to do that. There are a couple of places where you can choose to add features, but you must explicitly justify such additions, because what is specified below is in the formal specification and standards documents and has already been implemented a number of times. Changes to the protocol design and implementation must be extremely compelling.

B. The Bundle

The bundle is the unit of transmission. It is not fixed size, but is limited in size at any point in the network by how many bits can be transmitted between a sender and receiver, both of which are Bundle Nodes (discussed further below) based on both the bandwidth available between them and the amount of time they are in direct communication. A bundle contains a unit of data specified by the application, the Application Data Unit (ADU) using the BP, thus its original size is unconstrained. The application, in creating the ADU has full and exclusive responsibility for formatting and organizing the bits in the ADU. The BP simply receives the set of bits determined by the application. It then packages that information for transmission. A bundle consists of a set of blocks, with differing purposes. (See Figure 2(a).) The fixed-size primary block contains what in another protocol might be the header, so it includes such immutable information as the source, destination, and report-to node addresses, an identifier for the bundle itself, so each bundle is uniquely identified, and a set of flags to indicate choices about how the bundle is to be handled. The source and destination are simply the originating node and the final node. The report-to node is the node to which any responses or error messages might be sent. The flags (specified below) indicate various bits of information such as priority and whether or not response or reporting messages might be expected. Then the bundle contains a single block of data containing the data of the ADU from the requesting application. Finally, the bundle may contain zero or more optional blocks that may provide for “management” information for handling the bundle. These include optional information such as the bundle age or a limit on the number of hops the bundle should experience.

NASA has specified several optional blocks that can be included in a bundle. The interesting ones from our perspective are:

- Previous node: this identifies the node from which this bundle was directly forwarded.
- Bundle age: this is the amount of time since the bundle was created. It is useful, in part because some nodes may not have accurate clocks, but can compute elapsed time. It is in seconds. When this reaches the bundle lifetime as specified in the primary block, the bundle will no longer be forwarded.

- Hop-count: this contains two values, the hop-count limit and the number of hops the bundle has experienced. When the hop-count reaches zero, it is expected that the bundle will no longer be forwarded.

You can decide whether or not these are useful to you and under what circumstances. You are also permitted to design your own (well justified) options, if you find you need others. You will need to make the case for them and explain any tradeoffs in each, this will also require NASA to update their implementation of the protocol, so they are likely to be resistant to changes to the protocol, without strong justification.

A critical feature of the bundle concept is that a bundle can be fragmented if it cannot be transmitted in its current size. Continuing with the email example, it may be the case that the email message has a large attachment. The BP knows nothing about the internal structure of the email message, just that it is quite large. Consider the situation in which the time when BN0 and BN2 are in direct communication is too short to transmit the whole email including the large attachment. The BP at BN0 will split the data into two parts, two bundles each with a primary block, etc. and with an indication that it is a fragment and where it fits into the whole (also in the primary block). Your design will need to determine the size of the first fragment, but the BP itself will handle the fragmentation itself. The first fragment will be small enough to send. The second may be small enough to send as well, or may need a further fragmentation. Once the bundle is fragmented, BN0 will send the first fragment to BN2, and if only two fragments were needed, the second fragment to either BN1 of BN2. The whole original bundle will be reconstituted when all the fragments rejoin, in our example, at the destination. If any fragment does not arrive or is corrupted the email will be lost.

This design leads two types of bundles, the normal bundles carrying an ADU or part of an ADU, and administrative bundles. Administrative bundles are the responses being sent from various bundle nodes in the network to the report-to node. They are prohibited from requesting any reporting, acknowledgements or error messages about their own delivery. They can only be sent with all the reporting sorts of flags turned off. In addition, administrative bundles are specified as being unfragmentable. They are sent as is or dropped.

Now that the main aspects of the bundle model have been described, we can return to the set of flags mentioned above. NASA has specified a particular set of flags for use in the primary block. You may or may not find you have uses for any of them. If you need another type of flag to do something NASA did not think of, you may specify that, but you will need strong justification, because as mentioned above the protocol has already been standardized and implemented. The maximum number of additional flags you can add is four, because the other bits are being used for other things, not part of this design or are being reserved for future extensions. The flags that NASA has decided on are:

Bit #	Flag name	Results
0	Fragment	Bundle is a fragment
1	Administrative record	Bundle is an administrative record
2	No fragment	Bundle cannot be fragmented, it must be sent whole or rejected
3	Custody transfer request	Custody will only be transferred among custody nodes
5	Application ack requested	The application will try to send an ack to the sender
6	Status time requested	Every status report will try to include the current time
7-8	Priority class of service	These will be one of high, medium, or low.

14	Bundle reception status report requested	Every node will attempt to send a “reporting node received the bundle” bundle for this bundle to the “Report-to” node
15	Request report of custody acceptance	A custody accepted bundle will be sent to the “Report-to” node, if custody is accepted
16	Bundle forwarding status report requested	Every node will attempt to send a “reporting node forwarded the bundle” bundle for this bundle to the “Report-to” node
17	Bundle delivery status report requested	Every node will attempt to send a “reporting node delivered the bundle” bundle for this bundle to the “Report-to” node
18	Bundle deletion status report requested	Any node deleting the bundle will send a “reporting node deleted the bundle” bundle for this bundle to the “Report-to” node

Table 2: The optional flags available in the primary block of a bundle.

At each node generating report bundles, in general, a single status report bundle will be created that will include the status of whether the bundle was fully received, whether it was forwarded, whether it was delivered, and whether it was deleted. In some cases, a BN may send several reports to the report-to node if conditions for reporting occur sequentially. If they can be grouped together because they occur close enough in time they will be, but there is no guarantee of this. Remember that you may choose to use all, some or none of these types of reporting. There is no requirement to use any particular ones, but some may provide functionality that you will need to support the use cases specified below.

C. The Core of the Bundle Protocol

The Bundle Protocol (BP) moves bundles from sender to the receiver through a series of Bundle Nodes, each of which supports the BP. The BP is called a *store-and-forward* protocol because it stores the bundle at a bundle node until it can forward it to the next Bundle Node, and that may take some time. It is useful to remember that ADUs and therefore bundles may be different lengths and therefore may need different handling. The current implementation supports only an extremely naïve storage system on each node with no coordination among the storage systems. NASA hypothesizes, and you are being requested to demonstrate that it is possible to design a much more sophisticate storage system that will provide increased throughput (delivering more bits more efficiently), with lower overhead, lower loss, lower rejection, and improved reliability. In addition, although applications may consider ADUs to have an ordering, the BP considers each ADU independently of all others, so they may be transmitted in parallel or out of order. In other words, different email messages in the example above might be sent by different routes and arrive in a different order than they were sent.

The bundle is passed from one Bundle Node (BN) to one or more others to make progress toward its destination. For purposes if increased reliability, time constraints and possibly changing delivery paths, bundles may be duplicated on multiple paths. In addition, fragmentation may occur, as discussed above, anywhere along the path, and the fragments may follow different paths. Fragmentation normally is handled by the protocol itself, so your design will not need to request it explicitly, but the option exists for you to also request fragmentation explicitly, in which case you will need to specify the size of the first fragment. When a BN receives a fragment, it may choose to collect all the other fragments, if they arrive, and reassemble some or all of the bundle into its original form. Again, this is something that the

protocol already does when possible.; it is not something that you need to design. At the latest the bundle will be reassembled at its destination.

In considering the BP, it is important to remember that basic communication using the BP is simple, but best-effort unreliable delivery. One reason for this basic unreliable service is that any other alternative requires notification back through the network acknowledging delivery, partial delivery, etc. by sending *administrative* bundles. Choices you make about whether and how to use these options in the BP are likely to be determined by your design to meet the requirements of the Use Cases later in this document.

Another feature of the BP is that it provides a limited form of ordering on bundles as they pass through each BN, as specified in bits 7-8 of the flags. These bits together allow for specification of one of three levels of "priority": low, medium, and high. The application providing tan ADU will specify to the BP the priority level of each ADU. The protocol specifies that bundles will be given a priority of order from each single source. So, all the bundles from source A that are high priority will be sent out before those from A of medium priority. And they will go out before the low priority traffic from A. This says nothing about whether medium bundles from A will be sent before or after high, medium or low bundles from B, and so forth. Nor does it specify anything about the order of bundles at a single priority from a single source. So, bundles *a, b,* and *c*, all of which are at highest priority from source A may arrive in that order at BN0, but the choice of how they will be sent out from BN0 is not specified. (You may find it useful to make some choices in situations such as this.) The only guarantee is that bundles high priority bundles *a, b,* and *c* from A must be sent before any medium or low priority bundles from A. In the face of this sort of ambiguity, you must decide how best to use the capabilities of the protocol to achieve increased performance and/or reliability in the delivery of bundles and at what "cost" in terms of additional traffic and potential delays.

Related to the issue of "priority" is the issue of queueing. A node can only send bundles sequentially to other nodes. That means that it must pick an absolute ordering on bundles to be sent to each other node. The priority scheme above does not provide you with a complete ordering on all outgoing bundles, in two ways. Among all the bundles being sent by a single source at a particular priority level, the protocol does not provide an absolute ordering. In addition, as mentioned the ordering of bundles from different sources is more or less unconstrained. You will need to explain how each "next" bundle to be sent will be chosen.

Another feature of the BP is that different BNs may take on different levels of responsibility for communication. All BNs will forward bundles to the best of their ability. Some nodes may respond to the requests for administrative bundles to be sent to the report-to node. Some nodes may also be prepared to be *custody* nodes; these are predetermined by NASA and only comprise a subset of all Bundle Nodes. Custody nodes take on an additional responsibility for receiving and delivering a bundle. For a bundle that specifies (through flag bit 3) that custody is requested, as the bundle moves through the series of BNs, as it arrives at a custody node, that custody node will confirm with the previous custody node that it has received the bundle. If bit 3 is set then the "current custody node" field will contain the address of the current custody node. Furthermore, in order to accept a custody bundle, the receiving Bundle Node agrees to retain a copy of the bundle until either it has received notification that a succeeding node has successfully taken custody (including the final destination) or the bundle has timed out. Because the "current custody node" is included in a bundle requesting custody transfer, the node accepting custody must send an administrative bundle to the current custody node, to report custody transfer. Note that it is also possible that bit 15 will be set. This will cause an administrative bundle to be sent to the report-to

node as well, so that progress can be monitored. This provides an enhanced form of increased reliability with respect to store-and-forward. Custody transfer can only occur over a whole original bundle, so if the bundle is fragmented along its path, it will need to be reassembled before custody transfer can occur.

In this context, there are two noteworthy points. First, any bundle node can choose to ignore any request in an arriving bundle for a response (in the form of an administrative bundle) to be sent. They are optional from the perspective of the intervening bundle nodes. Second, these administrative bundles are all sent unreliably. That means they cannot request acknowledgements and reports about their progress and delivery. What this means for you in your network storage subsystem design is that you may choose to have nodes support none, some or all of these reporting options, but you will need to explain how you will be providing the most effective communication improvements through your design.

2) Routing

The other element of the overall system provided for you that will be critical to your subsystem is the management of the information that is used for forwarding bundles. Forwarding is the action of moving a bundle at one BN one step closer to its destination. At each BN, a decision will need to be made about the next step. The routing protocol implemented by NASA provides that "forwarding" information to each BN in a timely way. That means both that the updates are in a compressed form and therefore quite small and incremental, and that they occur once a minute. This protocol and forwarding table management system are provided by NASA, so you can use this subsystem, but do not need to design it. Its task is to make sure that each BN has the information it needs for you to make the appropriate decisions about forwarding and fragmentation.

The identifiers of source, destination, report-to, custody nodes, and all BNs within the network are addresses.[8] They are used to figure out where a bundle should be forwarded next to move it on its path toward its destination as well as the amount of time this connection is available. When a bundle arrives at a bundle node, forwarding of the bundle to the next bundle node will be done by looking up the address of the final destination in the forwarding table and learning about next possible steps, as well as the amount of time for which this is valid. In addition, this element of the system provides two other kinds of information: 1) the two closest neighbors of the same type of device as itself, and 2) custody forwarding information. The custody forwarding information will contain three facts, the address of the next custody node in a "custody" path, the address of the next hop Bundle Node toward the next custody node, and the amount of connection time remaining for communication with the next hop Bundle Node. How the information is provided to this forwarding table is outside your design challenge; just assume that each node receives forwarding information about addresses, and if it does not, then it will drop the bundle, either silently or with notification, if the source specified that it required notification. It is important to remember that next addresses and neighbors' addresses will change over time. It is the job of the routing protocol to keep that information up to date for you.

The forwarding table provides the normal approach to forwarding a bundle toward its destination. If a bundle is a "custody" bundle the custody path will be the default path. If the bundle is not a custody bundle, then the default path will be the non-custody path. Note that both may be valid for any

[8] This is a simplification of the Bundle Protocol as specified, but for purposes of this project, we will simply use addresses.

destination, and since the forwarding table does not know about the bundle itself, it provides all this information. On top of that you may decide that moving a bundle in another direction will provide some improvement in delivery of the bundle. This is something that you would need to design on top of the existing protocol. Thus, for example, if you decide that it would be better to send fragments of a bundle along different paths, or duplicating a bundle on several paths, that must be part of your design. The NASA code does not do that at present.

3. Your challenge: The Network Storage subsystem

The network storage system consists of a composite of the network storage available on each BN and any communications necessary for coordination and cooperation among BNs with respect to operating a "shared" network storage service. Your task is to design both the individual nodes and coordination management among the Bundle Nodes' storage systems, to provide a distributed network storage that will improve performance over a system with no such distributed coordination.

Within a single node, storage will need to be organized and managed. There are three kinds of influences on your design, the size of the storage available on the node, input/output limitations of the individual node, and transport policies (the degree of urgency for the traffic, the tolerance to lost information, etc.). We will discuss each of these separately.[9]

A. Storage

The most basic constraint on the storage is its size. Each satellite and relay will have only .5TB of storage. Your storage system will need to store and retrieve bundles as efficiently as possible. You will have a design decision to make. It is possible that under some circumstances, duplicate bundles may arrive at a node. In terms of how you organize and manage your bundles, you might choose to allow duplicates in order to increase other features, such as speed of insertion, or you may decide to eliminate duplicates in order to utilize your storage most efficiently. There may be other similar design tradeoffs. In addition, there will be times when decisions will need to be made. One type of decision will be which bundle to forward next when a transmission opportunity arises. As mentioned above, for each bundle source there will be up to three priority queues, but there is no specified ordering within a queue. In addition, there will be up to three queues for each source and the ordering among them is unspecified. A second will be what to do if the network storage is full and new traffic arises. Should something be dropped in favor of the new bundle? Should the new bundle be instantly moved to a neighbor? These will be determined by policies as discussed below. Your local design will need to accommodate these.

B. Communications limitations

The second constraint or influence on the storage on a node is the combination of the bandwidth between the communicating nodes of interest and the amount of time when they are in direct communication. These will determine how much traffic can arrive and how much can be moved either onwards or to a collaborating neighbor. The closer nodes are to Earth, the higher the bandwidth they support. Each class of device (LEO, GEO, relay, and remote antennas) will support the same bandwidth

[9] In such systems, power utilization is a critical characteristic. For purposes of this design project we set that aside, because the primary power consumption will derive from driving the lasers and that is outside the scope of the project.

in all directions (up, down and to neighbors). That said since the bandwidth on a link must be limited by the bandwidth of the lower capacity end of the link, a particular node may be constrained in one direction by a lower capacity next node. Thus, a GEO satellite, which itself has a capacity of 1.2GB/s will only be able to communicate with a relay at 622MB/s, because the relay is more limited.

C. Application requirements

The third constraint derives from requirements of the application. These may include urgency of delivery, tolerance to incomplete delivery of all ADUs, and other delivery expectations. To examine this in more detail, we consider three types of traffic, defined by the types of applications that use them:

- Routine management data with infrequent disruptions for mission critical information
- Telemetry, both routine and infrequently critical
- Extremely large transmissions such as software distributions and updates.

The management applications may involve specific commands for relocation, directional adjustments to focus on particular data, commands to rovers, etc. This traffic is expected to be relatively low volume, and, most of the time, non-critical. This type of data is intolerant of inaccuracies, both incorrect and missing data, but, we will assume, not critically urgent. On rare occasions these "communications" channels will become mission critical, as in the case of impending Solar flares.

Telemetry is data that is collected from sensors. In our case, the primary telemetry will be from sensors collecting data for the National Oceanographic and Atmospheric Administration (NOAA), although in the longer run there may be telemetry from devices based on the Moon, Mars or another body as well as telescope date, as well as observations of Sun activity, to watch for Sun spots and Solar flares. For our purposes in this project, we will assume that the telemetry data will be collected on the GEO nodes and will include data on both ground focused data, such as visible light and infrared images, and Solar information, used in part to detect impending Solar flares. Some of the data may be both tolerant to lapses and delays, while others, in the event of more immediate critical events (such as Solar flares) will have significant urgency to them. The volume of these types of data will be extremely high.

The third kind of data is system maintenance and updates. Primarily this will be software updates. These will have slightly different requirements. They must have several guarantees including completeness (all the bits must arrive), correctness, delivery to all the correct destinations, and then a guarantee that the updates were installed correctly. Needless to say, these may be quite large sets of data.

D. Some general thoughts

As discussed above your overall objective is to design a coordinated, distributed network storage system, to support sharing of network resources to improve network performance above and beyond what can be provided by each node managing its own network storage independently. You will need to think about design choices both within each local network storage device (on each Bundle Node) as well as the coordination among these individual storage units. You will also need to think whether and how to best utilize (or not) the various options within the BP for increased reliability and reporting. It is important to remember that there is no absolute right answer here. All design decisions are likely to have tradeoffs and repercussions. Part of your task is to recognize, evaluate, and justify the tradeoffs that you select. In the next section we discuss two aspects of that evaluation, a set of possible evaluation criteria and a set of use cases that will help both drive and justify your design choices. You must consider all the use cases in your design.

4. Design criteria and evaluation

Evaluation of your design will be critical to convincing your readers that your design is acceptable. It comes in two forms. The first is overall performance and limitations. This may include such criteria as the space allocation in storage for your design choices of how to store bundles, an estimate of how much computation (and possibly reorganization) your network storage design requires, the tradeoff between dropping and immediately pushing a bundle to a neighbor, when the local storage is full, and so forth. It will also include any limits to scaling that you can identify. These are about overall expected performance. In addition, you are provided below with three specific use cases. In your final report you will need to explain how you will support each (and all together) effectively, and any limitations on those.

A. Evaluation metrics

This section is intentionally brief at this point, but will be expanded in later documents. That said, there are a number of types of metrics you might consider.

- Network overhead: consider how much bandwidth beyond the actual transmission of bundles your design will require. This may be affected by lateral movement among satellites at the same altitude in order to achieve some performance tradeoff, as well as any reporting you may require.
- Times to completion of delivery of bundles
- The tradeoffs between time and space of managing the storage your organization of the local BN storage capability
- The costs (bandwidth and time) of your design choice for how to handle the handing off of a bundle to a neighbor satellite (i.e. if a BN is a GEO satellite, and the choice is to hand off a bundle to another GEO satellite, what does it take to do this).
- Improvements in delivery of bundles using your coordinated network storage facility. This may be in terms of time, bandwidth, or other metrics.

B. Use cases

In addition, your design must be evaluated in terms of three specific use cases, as well as any others you deem valuable to making your case. In addition to evaluating the effectiveness of your design in supporting each of these, because these all can be occurring simultaneously, you will need to discuss their interoperability. The three use cases are:

a) Routine communications: Assume there are two people resident on the Moon in different places, as well as two people resident on Mars in different locations. On a regular but not continuous basis, Ground Control wants to check in with them, send them task responsibilities, send and receive email between them and family, and so forth. 5% of the time there will be simultaneous messages between these people and Ground Control. The messages will range in size from 1KB to 10MB (might be a video between family members). For email and other personal messages there are no delivery time constraints as long as they arrive in a reasonable time-frame, but for task related messages, there is a requirement that they arrive within 10 minutes of their expected time. It is important that at the application layer (email, video streaming, wake-up calls, etc.) that all messages be received completely. Each person will be given his or her own tasks, not connected with the others. The average rate of the traffic between each person and Ground Control is 1KB per minute, but it may be bursty. There will be

very rare occasions (such as the Solar Flare situation in Use Case b)) when some of these small messages will become extremely critical and will require both as fast and as reliable delivery as possible for human safety.

b) Telemetry transmissions, mission critical or not: In our simplified example, telemetry date (from sensors, telescopes, etc.) is collected by devices on the GEO satellites. Types of data will include visible light and infrared images of the ground, and Sun energy images to be used for Solar flare predictions. The volumes of these data are:

Data type	Source	Destination	# Files	Size per file	Frequency
Solar	GEO	Earth	100	200MB	1 min
Earth images	GEO	Earth	25	10MB	5 min

Table 3: Telemetry information

These data will have different levels of urgency and accuracy. It is desirable to deliver as many of these as possible, but on average at least one in five is necessary from each satellite. The Solar energy data also tolerates low delivery rates unless the data has changed by 5% or more. In particular, although the data collected will sample across the full electromagnetic spectrum, soft x-ray is a valuable indicator of the precursor stage to a Solar Flare. For our purposes we will assume that this soft x-ray data comprises 1% of the total data. When the trigger occurs, it will become critical that all Solar measurements be sent both accurately and efficiently. They represent a potential impending Solar Flare, and the data must be fully analyzed on the ground and then the astronauts warned to take protective action. While the Solar measurements do not indicate criticality, the telemetry data will still need to be sent, but it will not be as urgent.

c) Large transmissions requiring guaranteed correct delivery: On a fairly frequent basis, software updates will be needed on all of the different types of devices, LEO satellites, GEO satellites, relays and the facilities on the Moon and Mars. For each software update, it is critical that every target device receive and within 30 minutes initiate the update process. In addition, each device must confirm correct installation of the updates. The maximum sizes and frequencies of these are:

Destination	**Size**	**Frequency**
LEO satellites	4GB	1/month
GEO satellites	16GB	2/month
Relays	1GB	Every 2 months
Moon and Mars equipment	100GB	1/month

Table 4: Software update sizes and frequencies.