

Lunar Relay Satellite Network for Space Exploration: Architecture, Technologies and Challenges

Kul B. Bhasin¹, Anthony W. Hackenberg², Richard A. Slywczak³
NASA Glenn Research Center, Cleveland, OH, 44135

Prasanta Bose⁴
Lockheed Martin Corporation, Palo Alto, CA, Zip Code

Marcos Bergamo⁵
BBN Technologies, Cambridge, MA 02138

and

Jeffrey L. Hayden⁶
PresciPoint Solutions, L.L.C., Littleton, CO, 80123

NASA is planning a series of short and long duration human and robotic missions to explore the Moon and then Mars. A key objective of these missions is to grow, through a series of launches, a system of systems infrastructure with the capability for safe and sustainable autonomous operations at minimum cost while maximizing the exploration capabilities and science return. An incremental implementation process will enable a build-up of the communication, navigation, networking, computing, and informatics architectures to support human exploration missions in the vicinities and on the surfaces of the Moon and Mars. These architectures will support all space and surface nodes, including other orbiters, lander vehicles, humans in spacesuits, robots, rovers, human habitats, and pressurized vehicles. This paper describes the integration of an innovative MAC and networking technology with an equally innovative position-dependent, data routing, network technology. The MAC technology provides the relay spacecraft with the capability to autonomously discover neighbor spacecraft and surface nodes, establish variable-rate links and communicate simultaneously with multiple in-space and surface clients at varying and rapidly changing distances while making optimum use of the available power. The networking technology uses attitude sensors, a time synchronization protocol and occasional orbit-corrections to maintain awareness of its instantaneous position and attitude in space as well as the orbital or surface location of its communication clients. A position-dependent data routing capability is used in the communication relay satellites to handle the movement of data among any of multiple clients (including Earth) that may be simultaneously in view; and if not in view, the relay will temporarily store the data from a client source and download it when the destination client comes into view. The integration of the MAC and data routing networking technologies would enable a relay satellite system to provide end-to-end communication services for robotic and human missions in the vicinity, or on the surface of the Moon with a minimum of Earth-based operational support.

¹ GRC ECANS System Manager, Exploration Systems Mission Directorate, 21000 Brookpark Rd., MS 142-4, Cleveland, Ohio, 44135, AIAA Associate Fellow.

² Computer Scientist, Computing Science Division/Computational Sciences Branch, 21000 Brookpark Road, MS 142-1, Cleveland, OH 44135.

³ Computer Engineer, Communication Technology Division, 21000 Brookpark Road, MS 54-5, Cleveland, OH 44135, AIAA Senior Member.

⁴ Senior Technical Staff, Modeling, Simulation and Information Sciences, LMSSC ATC, Palo Alto, CA 94304.

⁵ Principal Scientist, Internetwork Research Department, 10 Moulton Street, Cambridge MA 02138.

⁶ Space Systems Engineer, 5467 S. Cimarron Road, Littleton, CO, 80123, AIAA Member.

I. □ Introduction

THE lunar orbiting communication and navigation relay infrastructure is expected to be established over a period of time to support both robotic and short term human lunar missions; and will grow to meet the final objectives of long term lunar exploration, habitation, and colonization. Such an infrastructure will require lunar relay satellites to provide lunar vicinity and surface coverage, flexibility and agility in forming connections, and the ability to provide communications and navigation for multiple, concurrent missions. Several architectural options including location, number of satellites, and their orbital mechanics have been well considered in the literature¹. However, in the case of the lower, more stable orbit satellite network, approaches to in-space connectivity and networking to achieve autonomous behavior and the provision of network-based services have not been closely analyzed. A low lunar orbit communications relay constellation system can evolve by including small, inexpensive communication spacecraft as “hitchhikers” on launches by other missions. The low altitude provides shorter communication paths to surface nodes and reduces the on-board communication hardware mass, power, and cost, while maintaining very high data rate link support for surface to surface communications. A relay satellite in low orbit will pass over surface sites relatively quickly, thus requiring more satellites than a constellation of high orbit satellites to provide nearly continuous communication support.

Delay tolerant networking and/or cross-link capability among multiple relay satellites will likely be needed to provide communications to non-visible nodes (on the far-side or in craters) on the Moon. Current networking approaches based on planned communication routes among space assets are inadequate to meet the on-demand and resource optimizing communication constraints, especially with such large number of nodes. Static routing has been the previous method of choice for most routing in space networks. Network engineers manually schedule the routing of data through the ground stations and satellites. This technique works well if the nodes and links as well as traffic characteristics are fixed and known ahead of time. Static routing must be manually reconfigured if new relay satellites and user nodes become available. Newly deployed nodes might also provide an alternate path for communication and better communication services such as higher bandwidth. It is possible to manually update data routing configurations to accommodate new nodes and capabilities, of course. However, when the number of such nodes increases, it becomes increasingly vital to have a system that can adapt itself to network dynamics without requiring manual intervention. Below we describe Autonomous Space Communications Technology (ASCoT) software that can adapt itself to embrace new resources and user nodes as they become available to the space network. This software enables the space network to function gracefully in a dynamic space environment. ASCoT also directs an agile media access control (MAC) schema that, in-turn, directs electronically steered antenna arrays and/or fast pointing dish antennas to autonomously handle the rapidly changing geometry of the communication links among multiple separated nodes in-space or on the surface. The relay satellite also uses the MAC and special protocols to make the ad hoc connections and handoffs to nodes as they dynamically appear and disappear.

II. □ NASA Space Exploration Missions

The Vision for Space Exploration refocused NASA on a new initiative to explore the Moon, Mars, and beyond with safe, sustainable, and affordable missions. NASA's exploration roadmap calls for manned missions to the Lunar surface by 2020 leading to Mars surface exploration. Human presence on the surface will evolve in three stages. In the initial stage, astronauts will be sent to any Earth-facing location on the Moon for a 4-7 day stay. During these missions, no infrastructure will be pre-deployed on the surface. In the next stage, humans will be sent to pre-designated locations on the Moon for long duration, 42-98 day missions. These missions will involve pre-deployed assets such as a habitat, nuclear power station, communications infrastructure, and pressurized rover. In the third stage, the capabilities developed for long-term lunar surface stays will be used to transfer a crew to Mars for longterm exploration. Surface-to-surface communications will play a critical role in all of these missions, enabling collaboration among astronauts and advanced health monitoring and control capabilities.

III. □ Lunar Communication Relay Satellite Network Architecture

A system of communication relay satellites will be needed in orbit about the Moon to support surface-to-surface and surface to Earth communications for the longer duration human missions and for human and robotic missions to lunar sites not visible from Earth.

A. Communication Relay Constellation Considerations

In February 2004, NASA's Space Operations Mission Directorate (SOMD) established a Space Communication Architecture Working Group (SCAWG) to develop communication architecture to support the Human and Robotic Exploration of the Moon.¹ Much of their architecture work focused on relay constellation orbital design, on potential

satellite design, including the use of derivatives of existing satellites, and on the cost of various options. The means of providing on-demand connectivity and network routing functionality were not covered in detail. This paper examines the implementation of a constellation of lower altitude relay satellites into an autonomous communication network that provides on-demand connectivity among surface nodes and with Earth.

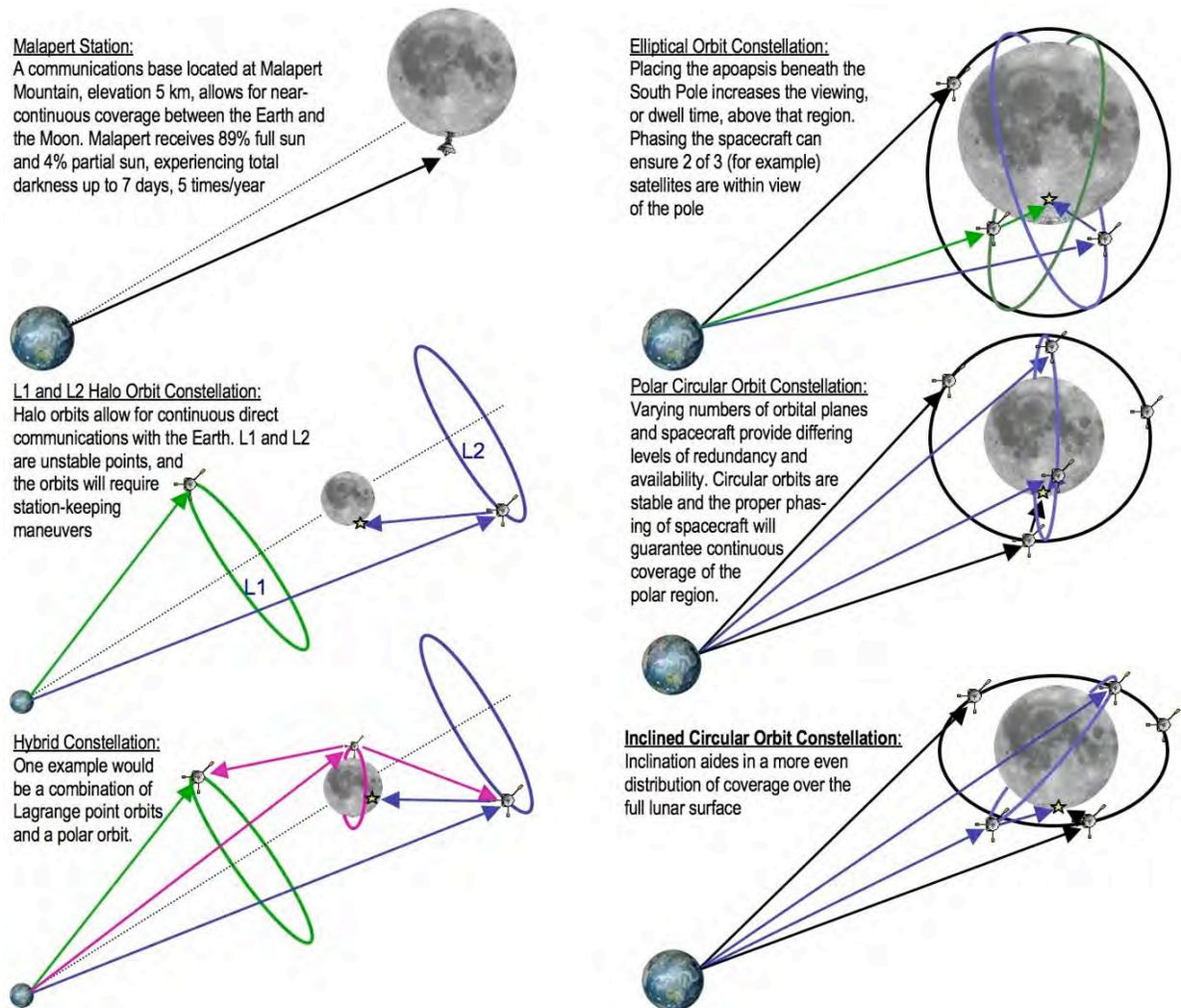


Figure 1. Lunar communication relay and satellite orbit configurations studied by the SCAWG.¹

The orbits studied by the SCAWG are shown in Figure 1. The Malapert Station configuration is a special case that provides nearly continuous direct communication from the lunar South Pole to Earth. That configuration does not use relay satellites and thus cannot support communications to other lunar surface sites. The L1 & L2 halo orbit and hybrid constellations, treated elsewhere², employ larger satellites with higher power radios and larger antennas to handle high data rate (100-1,000 Mbps) communications at a greater distance from the Moon (61,000 km to L1 or L2 from the Moon). The elliptical orbit constellation is, again, a special case constellation for supplying 24/7 coverage of the lunar South Pole and limited coverage for other lunar surface areas. This paper describes communication networking with relay satellites in inclined and polar circular orbit constellations at low to moderate altitudes (50 to 400 km).

B. Autonomous Network Routing Within the Lunar Relay Constellation

An operational view of Lunar Relay Satellites (LRS) in low altitude, inclined circular orbits around the Moon is shown in Figure 2. In this figure, the LRS nearest the center of the Moon is shown as capturing data from Earth and routing that data by crosslink to two other LRSs in a different inclined orbit. Those satellites then pass the data on to

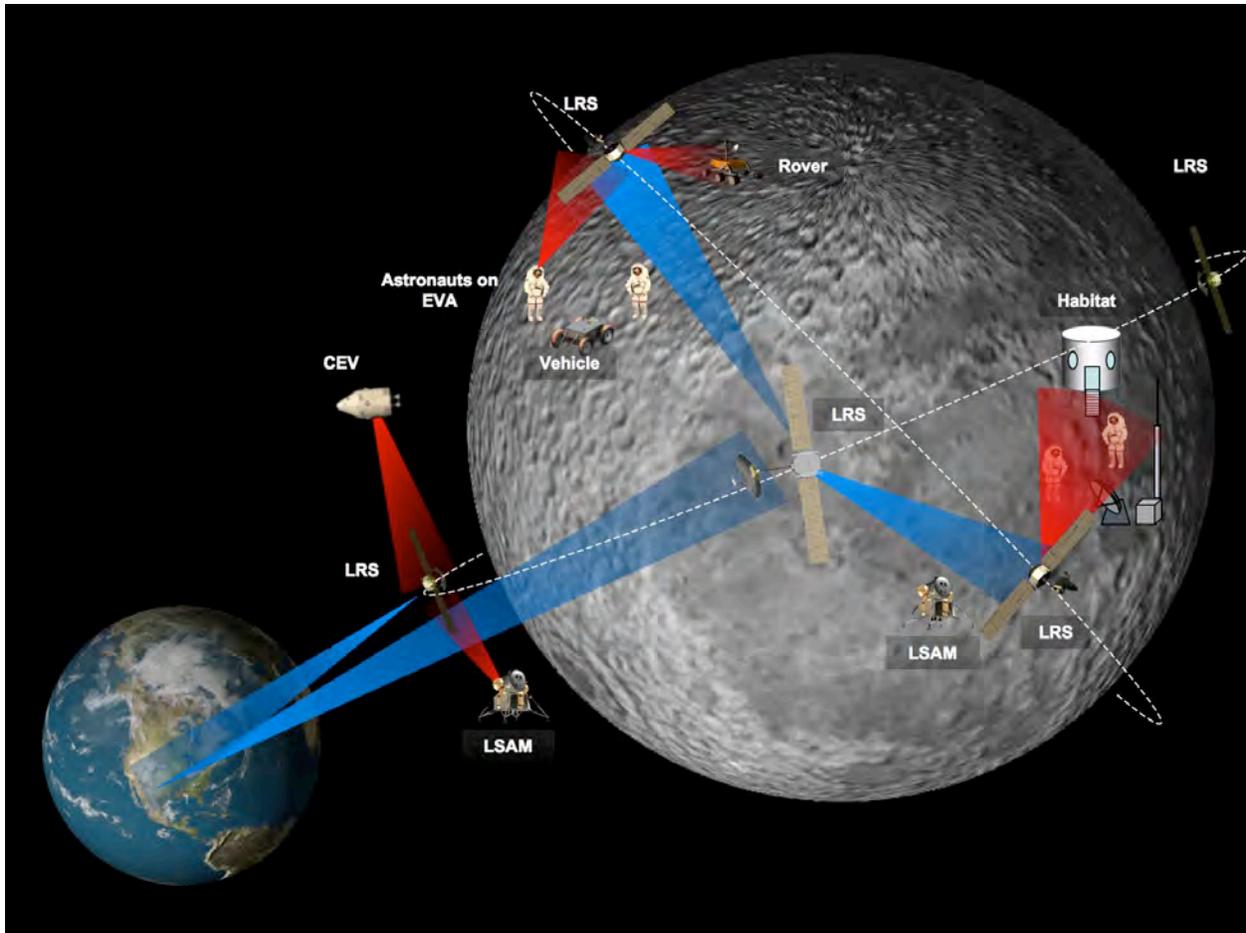


Figure 2. Low lunar orbit relay satellites provide Earth-Moon vicinity-Moon coverage. Blue rays indicate backbone links among relay satellites and Earth. Red rays indicate links to users of the relay network.

lunar surface nodes such as rovers, astronauts on Extravehicular Activities (EVA), and to high data rate surface terminals. Each LRS also passes data from surface nodes to other surface nodes or to Earth. The LRS at the Earth-side of the Moon is shown as capturing data from the Crew Exploration Vehicle (CEV) and the Lunar Surface Access Module (LSAM) and then routing the data to Earth. This LRS can also route data among the in-space Crew Exploration Vehicle (CEV), LSAM, and the lunar surface nodes. The on-board routing among communication packages is handled by ASCoT software. This software provides a unique combination of positional-state routing (in three dimensions) and link-state routing (in a very efficient manner). ASCoT also dynamically tracks the source and destination node directions as the LRS orbits the Moon; and doesn't attempt to contact a node if it is not in view. All data that passes through an LRS is first brought to the digital Internet Protocol (IP) level, routed to the appropriate radio, and then modulated onto that radio's RF. Using directions received from the ASCoT software, which has calculated the direction of the destination node, that radio utilizes special Media Access Control (MAC) software to configure its antennas and direct the RF beam at the destination.

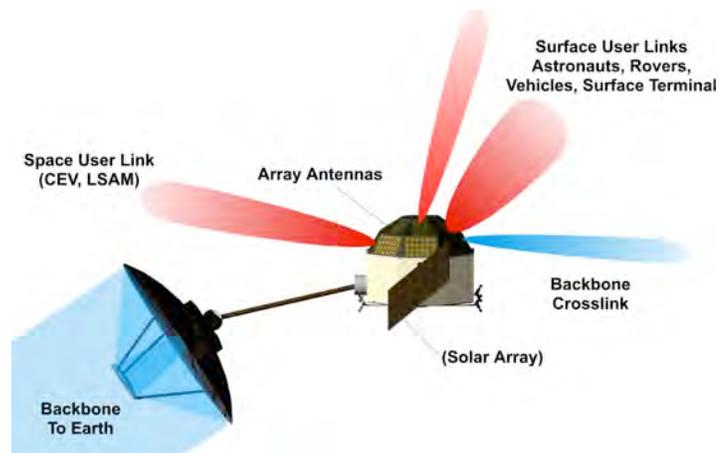


Figure 3. LRS concept using array antennas for local communications and dish antenna for Earth communications.

The beam directing concepts are shown in Figure 3. The MAC layer, discussed below, introduces dynamic coding, beam directing, and power control software to optimize the passing of data to and from nodes at widely varying distances.

IV. □ Forming Networks in Space

A. General Description of Networks in Space and of ASCoT Networking

On satellites providing backbone communication links, ASCoT^{3,4} leverages predictable positioning and dynamics of nodes in space to make routing decisions. Positional Link State Routing (PLS), an extension to the link state routing used in the Internet, enables ASCoT to compute paths using links that will be available even in the future. PLS is unique relative to other link state protocols in two ways. Traditional link state protocols do not use positional information, while PLS does. Furthermore, in PLS, link state information can be proactively disseminated. Such a communication mechanism allows declarative access to sensor data from proximity networks.

ASCoT provides the different Quality of Service (QoS) levels appropriate for data flows with different purpose and functionality. Space networks transport different classes of traffic broadly classified in two dimensions: bandwidth and latency. Traffic can be low or high bandwidth and latency tolerant or intolerant, or combinations thereof. Existing reservations based schemes assume that the most latency intolerant traffic is low bandwidth traffic. While this is true most of the time, there are cases when this is not true. For example, when a rover becomes stuck, high bandwidth telemetry might be necessary with the least data latency to provide remote control for freeing the rover. ASCoT has priority mechanisms that allow flows of different nature to co-exist in the system. For example, there could be a sensing experiment going on in one of the networks on Moon. This experiment generates steady data for the next 10 hours as scheduled days ahead of time. There could be health telemetry in response a rover emergency on a different place on Moon. In this scenario, even though the health telemetry data demands high bandwidth, ASCoT can allow this flow to take as much of the forwarding resources as possible. Any leftover bandwidth and buffer space is used by the science experiment data.

ASCoT has mechanisms to detect and recover from communication outages due to link, node, or network failure. A faulty antenna can make a link unavailable until it is fixed. A satellite node can disappear from the network due to an accident. A section of the network might experience an outage due to disrupted long haul links connecting this sub-network to the rest of the network. Any routing and forwarding system must be able to cope with these failures that can happen without warning. ASCoT can detect failures when they happen. The network propagates this information to the rest of the network efficiently (quickly and economically) so that the network can make adjustment in the paths it computes for flows and to appropriately reallocate/readjust resources for existing commitments.

B. ASCoT Architecture

The architecture of the ASCoT software is middleware that provides an Application Programming Interface (API) to the services ASCoT can provide. The diagram in Fig. 4 shows the main components of the ASCoT Middleware. It also indicates the Application/ASCoT and ASCoT/Runtime-Link Layer interface.

1. Application

Users interact with the network using the application layer functionalities. The application layer provides different types of interface depending on the need: OPNET GUI or a custom Java GUI. Applications make subscribe calls to inject a query to the network. Publish calls are used to push data

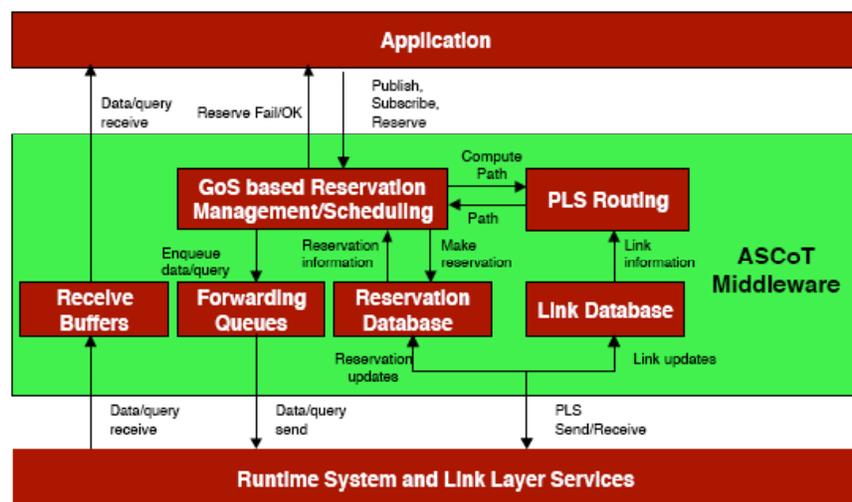


Figure 4. ASCoT Architecture

to the network in response to a query. Applications have event handlers that get invoked when there is data or a query ready to be received. Applications may indicate that the flow related to a subscription is a reserved flow.

2. *QoS based Reservation Management/Scheduling*

When a reservation request is received, ASCoT must make sure that this request can be accommodated. This module calls the PLS Routing module to first make sure there is an end-to-end path between the source and the sink. Then it looks up the reservation database to make sure that the new flow can be accommodated in all the links that comprise the path. While a packet is sent, this module also checks if the packet belongs to a reserved flow or a best effort flow and sets up appropriate priorities before sending the packet to the forwarding queues.

3. *PLS Routing*

This module is responsible for computing the paths between nodes in the network. The link information is read from the link database. Dijkstra's algorithm was modified so that paths can be computed even when packets need to be buffered at a node because no links are visible for a period of time. Our current implementation minimizes the latency metric. Using other routing metrics requires minimal change.

4. *Receive Buffers*

When application data is received, it is buffered temporarily but otherwise passed to the application with no other processing. If the packets are destined to other nodes in the network, the packets are forwarded to the QoS based Reservation Management/Scheduling module.

5. *Forwarding Queues*

The forwarding queues store packets before they are sent to the network. When packets are added to this queue, they are sorted by the priority (priority queue). Limited buffer size can cause packet drops - be it packets being generated locally or packets being forwarded. If the packets are dropped there is no feedback to any other component. Packets might stay in this queue for two reasons: too many packets for the available bandwidth or the next hop neighbor are not visible until some time in the future. Each interface has its own forwarding queue.

6. *Reservation Database*

There are two sets of information this database stores: (1) reservation metadata for all locally generated reservation and (2) link reservation for all the known links due to all known reservations in the network. The reservation metadata enables a node to reinitiate, repair, or harden a reservation by keeping track of the status of each locally generated reservation. Information about all the committed flows for a link helps a node determine if a new reservation can be accommodated on the link. There are two sources of updates to the reservation database: (1) locally generated reservations and (2) reservation information received from the network through PLS messages. Reservation database (except reservation metadata) is periodically disseminated to the network using PLS messages.

7. *Link Database*

Link database stores the topology information for the network. The times at which a link is available, links capacities at different times, and many other link parameters are stored in the database. Path computation looks up the link database to compute a path with links that are available at the time packets are expected to use that link. There are two sources of updates to the link database: (1) locally inferred and detected links and neighbors and (2) link information received from the network through PLS messages. Link database is periodically disseminated to the network using PLS messages.

8. *Runtime System and Link Layer Services*

The satellite's on-board clock is periodically synchronized with Earth time and sensors that measure the positions of the stars, Earth, Moon and sun can be used to calculate the trajectory, navigation, and link availability on-board. In addition, the run time system provides neighbor discovery, antenna pointing services and helps coordinate the spacecraft attitude.

9. *Link information dissemination*

PLS leverages the largely predictable trajectories of space assets and distributes the link information before the links become available (this is not the case with mobile ad hoc networks). When a link comes up, nodes at each end of the link exchange their link state information; in this manner, whenever nodes could have used a link for communication, they have the necessary information to compute paths. Initial knowledge of the link's existence is obtained either from a lower stack layer or through previously known links during link state dissemination. Over a period of time, nodes accumulate enough link state

```
v.receive_linkstate (links)
  for all l in links
    if exists(known_links, l.(u->v))
      known_links[u->v] = l
    else
      known_links.add(l)
  v.link_refresh()
  for all l in known_links
    if (l.age > LINK_TIMEOUT)
      l.delete()
      l.age++
  send(known_links)
```

Figure 5. Pseudocode1: PLS link information dissemination algorithm

```

// returns true if link l is available at time t
link_available(t, l)
  for all i in known_links[l].intervals
  if (t >= i.start) and (t < i.stop)
  return true
  return false

// compute the latency metric for a given link
latency_weight(u, v, arrival_time)
  for all i in known_links[u->v].intervals
  if i.start <= t && i.stop >= arrival_time
  return link_latency(u,v,now());
  if i.start >= t
  return link_latency(u, v, i.start)
  + (i.start - arrival_time);

// updates the tree if d[u] + (u,v) is shorter
// (according to a given metric) than d[v]
relax(u, v, metric_weight, agg_op)
  newdv = agg_op(d[u], metric_weight)
  if (d[v] > newdv) and
  link_available(arrival_time[u], (u,v))
  d[v] = newdv
  pi[v] = u
  arrival_time[v] = arrival_time[u]
  + latency_weight(u, v, arrival_time[u])

// initialize Dijkstra variables
initialize(G, s)
  for each vertex v in V[G]
  d[v] = inf
  pi[v] = nil
  arrival_time[v] = nil
  d[s] = 0
  arrival_time[s] = 0

// compute the best metric tree
// from source s in graph G
dijkstra(G, w, s)
  initialize(G, s)
  s = \{\}
  q = V[G]
  while (q != \{\})
  u = extract_min(q)
  S = S union {u}
  for each vertex v in adj[u]
  relax(u, v,
  latency_weight(u,v,arrival_time[u]), sum)

```

Figure 6. Pseudocode 2: Algorithm to compute PLS paths

the aggregation operator and latency as the link metric. One can think of using different aggregation operators and metrics with the same algorithm. For example, product aggregation operator for link reliability, sum for energy metric, min for maximum bandwidth, etc. Path computation builds a routing table that lists each node that participated in PLS message exchange and the next hop on a path that maximizes a given metric.

C. Providing Failure Tolerance in the Networks

ASCoT is designed to be robust to link, node, and network failures. Whenever there is any failure, the failure is detected, the failure information is disseminated to the network and appropriate adjustment to the path computation and reservations are made.

Beaconing is at the heart of adapting not only to the dynamics of disruption in line of sight or path noise, but also recovering from failures. Each node sends out beacons at a fixed frequency using all the interfaces. Thus each node

information to compute the shortest paths to each node in the network. Figure 5 shows a simplified description of Pseudocode 1, the dissemination algorithm.

10. Path computation

Traditionally link state protocols use Dijkstra's shortest path algorithm to compute a path to the destination. The algorithm expects a set of links, link weights, and the root node as inputs. To compute the shortest path from the root node, all the links are assigned a weight of 1. Thus, the algorithm computes the shortest path (in terms of number of hops) to all the destinations in the network. This algorithm assumes that all the links are valid and active at all times. However, in space networks many links are intermittent or periodic. To use Dijkstra's algorithm in this environment, the link connectivity must be predicted (part of link state entry for a link) and the link must be valid when the packet arrives. In other words, the algorithm must take into account future topology schedules.

To achieve these goals, the following changes are made to Dijkstra's algorithm. First, instead of hop-count, the desired routing metric is minimized. Second, the algorithm is modified to account for time-varying graphs. The algorithm uses the property that a path between two nodes never needs, at any time t , a link that is not active at t . To ensure this property, Dijkstra's algorithm is modified so that it does not add a link l to the shortest path tree if l is not valid between times t_1 and t_2 where t_1 is the time at which a packet would arrive at the leaf of the tree before adding a new link and t_2 is the time at which a packet would arrive at the end of the tree after adding a new link. Pseudocode 2 (Fig. 6) shows the pseudocode for the augmented Dijkstra's algorithm. This pseudocode uses sum as

has a neighbor table that is constantly updated with trajectory and link characteristics. The basic objective of a hello message is to let the neighbors know that a node is still alive and participating in the network.

The neighbor table is designed such that a neighbor times out if a beacon is not received from the neighbor for too long. This timeout delay is configurable to suit the topology and the network. A small timeout enables a failure to be detected quickly. However, a small timeout is not appropriate if properly functioning neighbors are not able to transmit beacon messages fast enough to refresh the entries in the neighbor table before they time out. If a network is known to be predictable to some extent, a larger timeout is preferred over smaller one. A larger timeout not only avoids wasting too much bandwidth on beaconing but also allows waiting long enough to absolutely make sure that the neighbor or the link has failed before deleting that entry from the neighbor table.

When paths fail, the failure is inferred by the nodes in the network as timeouts. The timeouts result in deletion of links from the link database. When a packet is generated or is ready to be forwarded, the path computation function is called. Because the link database is already up-to-date with failed links deleted, the path computation function returns a path that avoids the failed links.

Adapting reservations to the failures is more complex. Reservations allocate resources for a flow from a source to a sink. This end-to-end semantics of reservations allows us to repair reservations when links fail in the network. As long as we are able to find paths that can still provide the same set of resources as was allocated when the reservation was made, the commitments can still be honored. There are two steps to managing reservations in the face of link failures: (a) tearing down existing reservations and (b) making new reservation. Reservation states are flushed from the reservation table in a distributed fashion. When nodes infer that a link has failed, it also checks the reservation database to see if there were any reservations made on that link. If an entry for reservation is found, it is immediately deleted. As failure information propagates to the network, the per link reservation information is also deleted throughout the network. The node that made the reservation initiates the repair of a reservation. By looking up the reservation metadata table, it can compute new paths (that do not include the failed links) and make a new reservation on the links on that path. Thus ASCoT flushes stale reservation information and repairs it as link failures are detected.

V. □ Accessing Space Networks Autonomously

The ASCoT protocol provides the approximate coordinates of surface nodes (e.g., rovers, astronauts performing extravehicular activities (EVA), lunar habitats, etc.), space terminals (e.g., on the CEV, LSAMs, and cargo vehicles) and other relay satellites in view of any given LRS satellite as a function of time. Actual neighbor discovery, link establishment and characterization, inter-node synchronization and traffic scheduling and data multiplexing over such links are performed autonomously by each satellite under the control of a powerful media access control capability formed from a combination of time division multiple access (TDMA) with code division multiple access (CDMA) called TDMA with CDMA encoding multiple access (TCeMA). TCeMA was designed, from ground up, to address the key challenges of surface-to-space and space-to-space communications in Planet-Vicinity Networks composed of multiple satellites with on-board autonomous routing capabilities. TCeMA integrates orbital movement prediction to establish and maintain cross-links, autonomous node discovery to enable dynamically changing self-forming networks of “new” surface terminals and neighbor routing satellites as they come in and go out of proximity,

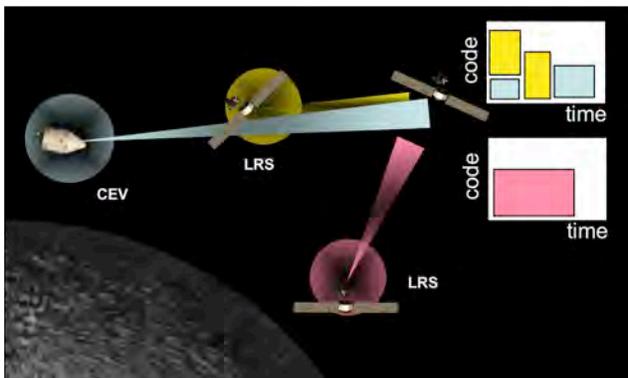


Figure7. The integration of spatial and time-code multiplexing enables full re-use of the available spectrum on the surface/space links.

agile beam-forming spatial multiplexing to maximize re-use of the allocated spectrum, and variable-rate cross-links with multi-code spread spectrum to maximize space access and network connectivity over widely ranging inter-satellite distances. Using ephemerides and coordinates provided by ASCoT, TCeMA derives range and spatial direction of in-view neighbor nodes and autonomously forms the links by pointing and then dynamically activating and focusing the beams toward each neighbor node while continuously adapting the data rate of each link to compensate for dynamic variations in distance. Figure 7 illustrates the integration of spatial multiplexing and time-code multiplexing.

The spatial directions and relative node alignment are measured periodically and, in cooperation with ASCoT, updated and tracked over time. The spectrum is fully reused on links that are spatially isolated. During the times in which satellites become “aligned,” the spectrum is shared using time and/or time-code multiplexing. The encoding of each packet is selected to meet bit error rate (BER) and delay/delay-jitter requirements of each *flow* that compose the aggregate traffic over each cross-link. Multi-code encoding is used to independently adjust the effective data rate of each packet to “close the link” as needed while meeting QoS requirements that may vary from packet to packet. Multi-code encoded bursts are power-efficiently transmitted using a constant envelope split-phase shift keying (SPSK) modulation overlay.

Burst reception is performed using null-steered digital beam-forming in which a receive beam with maximum gain is formed in the direction of a given node while antenna nulls are generated towards all other in-range nodes. The systematic use of null steering on both transmit and receive directions, aside from minimizing interference, simplifies the overall topological structure of the network (from “*mesh*” to a collection of “*independent stars*”), reduces the need for transmitter-receiver “*hand-shakes*” (very costly over long-delay space links) and—very importantly—enables a transmitting node (e.g., an astronaut) to simply *point and shoot* to a selected in-view LRS whenever he has a packet to be sent through that node.

Key space communication challenges addressed by the TCeMA and associated physical layer technologies include: initial neighbor discovery and link establishment to bootstrap the ASCoT routing; dynamic link establishment and rate adaptation with constantly moving neighbors at varying ranges and spatial directions; connectivity between nodes with different power and aperture communication capabilities over a wide range of propagation delays; throughput maximization with efficient use of the available power and allocated spectrum while minimizing the number of modem and transceiver resources on-board the satellite; and efficient support for transmission of isolated and traffic stream packets while satisfying QoS requirements.

In addressing the above challenges, the multiple access capability integrates a number of technologies at the physical-layer, MAC-layer and sub-network (i.e., below IP) levels. The physical-layer technologies are highly optimized for connectivity, power efficiency, and maximum spatial re-use of the available spectrum. They include:

- *Dual-band links*: S-band transmission is used for beaconing, link control while enabling initial exchange of coordinate-related ASCoT information between satellites, and low data rate transmission for ground-space communication by small platforms and sensor nodes. Ka-band communication enables high data rate transmission and spatial reuse of the available spectrum with small-size, high-gain array antennas.
- *Null-steered transmission/reception*: Enables reduced interference burst transmissions among in-range nodes (satellites or lunar-based terminals) while minimizing inter-node coordination over long-delay links.
- *Burst Reception with Digital Beam-forming*: Maximizes connectivity and link availability by enabling satellite to receive—possibly simultaneously—from all in-range neighbors (terminals or satellites).
- *Variable Rate Access Links and Cross-Links*: Enables nodes to maintain connectivity over a wide range of distances by adapting the link data rate to the varying link budgets.
- *Constant-Envelope SPSK Modulation*: Enables efficient use of the available RF power by allowing the operation of the array antenna power amplifiers at or near to saturation.

The MAC-Layer is highly optimized for spectrum reuse through spatial multiplexing and integrates with ASCoT at the sub-network level to keep the network connected (i.e., a single “island”) and to control the degree of connectivity (i.e., number of neighbors) in the constellation. The MAC-layer includes the following capabilities:

- *A Space-Link Frame*: Enables support for space-and-time synchronization (i.e., Direction of Arrival and propagation delay measurements), burst transmission of packet traffic with isolated (datagram’s), intermittent (bursty packet traffic) and recurring (stream traffic flow) characteristics, and for the discovery of new neighbors;
- *A Multiple Access Mechanism*: Integrates Spatial-Division, Time-Division and Code-Division Multiple Access leveraging the direct-sequence orthogonal code multiplexing capabilities of TCeMA.
- *Neighbor Discovery Protocol*: Enables a node to advertise itself, find other nodes, and achieve frame, time-slot and initial frequency synchronization with any other node that that happens to be within range.
- *Network Synchronization Protocol*: Enables each node in the network to, over time, achieve global frame-epoch synchronization while synchronizing its local internal clock and frequency generators to a designated and/or dynamically selected Reference Node.
- *Node Affiliation Protocol*: Enables endpoint terminals (e.g., CEV, astronauts performing EVAs, surface sensors) to find a space-based LRS router node that can relay traffic on their behalf, and perform dynamic handoff as the network topology and geometry changes over time.

- A QoS-oriented Bandwidth Allocation Mechanism that integrates support for long-term bandwidth allocation for rate-based and volume-based application traffic, and for packet flows with different delay and delay-jitter characteristics.

Figure 8 illustrates the TCeMA space-link frame. The overhead area, common to all nodes, includes time slots dedicated for neighbor discovery (DISC), inter-node synchronization (SYNC), and burst transmission announcement and control. The “data” area can be flexibly partitioned and allocated (by each receiving node) to traffic with different priority and QoS characteristics. The overhead time slots, in particular the DISC and SYNC time-slots, play a crucial role in the initial link establishment (i.e., neighbor discovery) and synchronization operations. Triggered by ASCoT, Neighbor Discovery is performed when two nodes become in-view and in-range of each other and typically used (i.e. with “enough” link budget) for

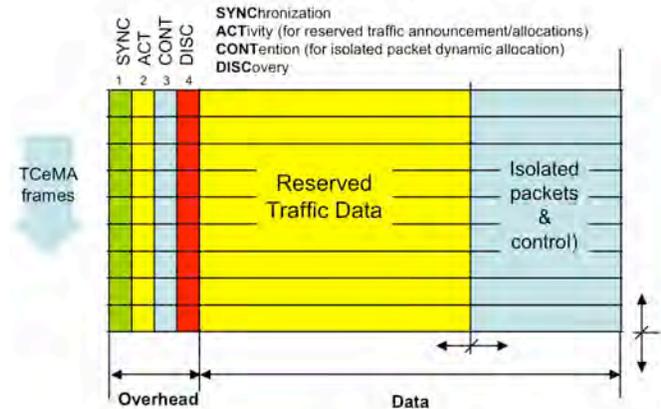


Figure 8 – The TCeMA frame capacity is used mostly for data and includes a small overhead area (<5%) for neighbor discovery, node synchronization and burst control.

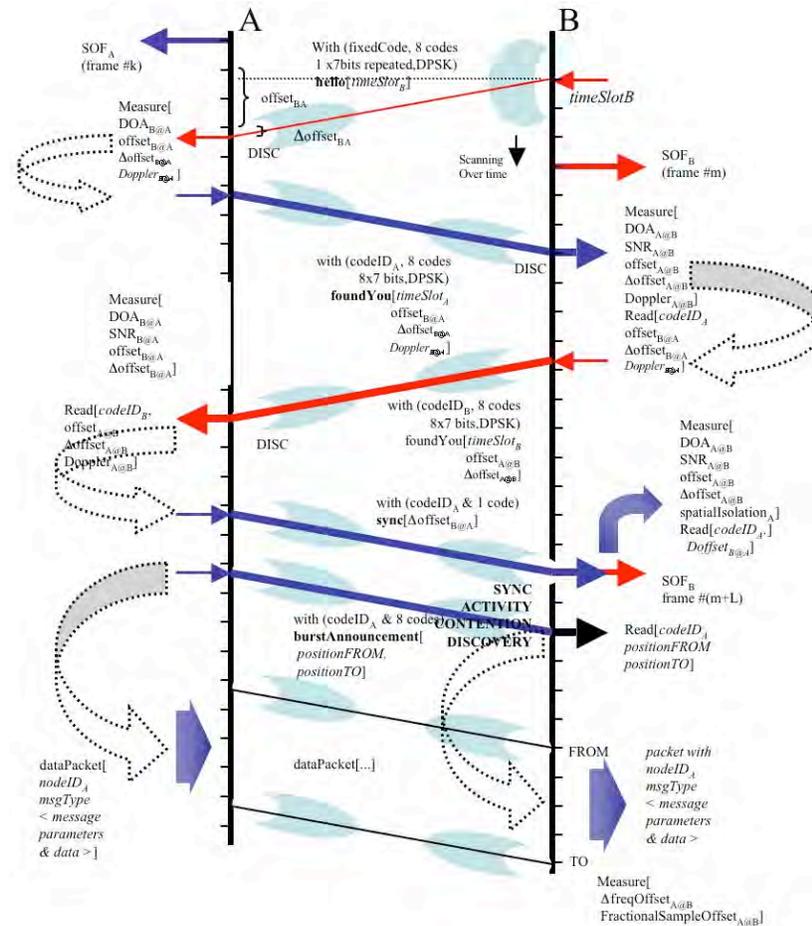


Fig. 9 - Initial link establishment is performed as soon as nodes become in-view of each other and typically ahead of time in which the link can be effectively used for data.

transmission of network traffic data.

We distinguish between two types of neighbor nodes: new and recurring. A ‘new neighbor’ is one for which the node performing Neighbor Discovery has no a priori knowledge of relative position. A ‘recurring neighbor’ is one for which some or all required relative position information (e.g., from ASCoT) is already available. Figure 9 illustrates the initial link establishment between two nodes that just became in view of each other. A node *supports* its own discovery by a “new neighbor” by transmitting HELLO bursts that scan the whole frame. A node *supports* its re-discovery by a ASCoT-aware neighbor by transmitting a HELLO burst with the maximum possible gain towards the recurring neighbor. A node *performs* Neighbor Discovery by systematically receiving the DISC time slot. Successful HELLO burst reception is confirmed by the exchange of a short FOUND_YOU burst that includes controls and assignment for subsequent periodic transmission of SYNC bursts. HELLO, FOUND_YOU and SYNC bursts are transmitted with high

(QAM) of each cross-link. Also illustrated in Fig. 11 are the effective symbol and bit rates achieved on the other two example surface-space access links: 10 Mbit/s (25 codes and 16-QAM); 33 Mbit/s (55 codes and 64-QAM).

The TceMA MAC software performs burst detection by “passing” the incoming signal through a rotating correlator. The rotating correlator transforms the received signal from code-domain (i.e., CDMA) to time-domain (i.e., TDMA). The codes used have a perfect-zero off-phase auto-correlation and the CDMA => TDMA transformation occurs without mutual correlation interference. In addition, when the rotating correlator ‘rotates’ at a rate of one cyclic-shift per channel symbol time, the resulting output signal is an exact replica of an equivalent

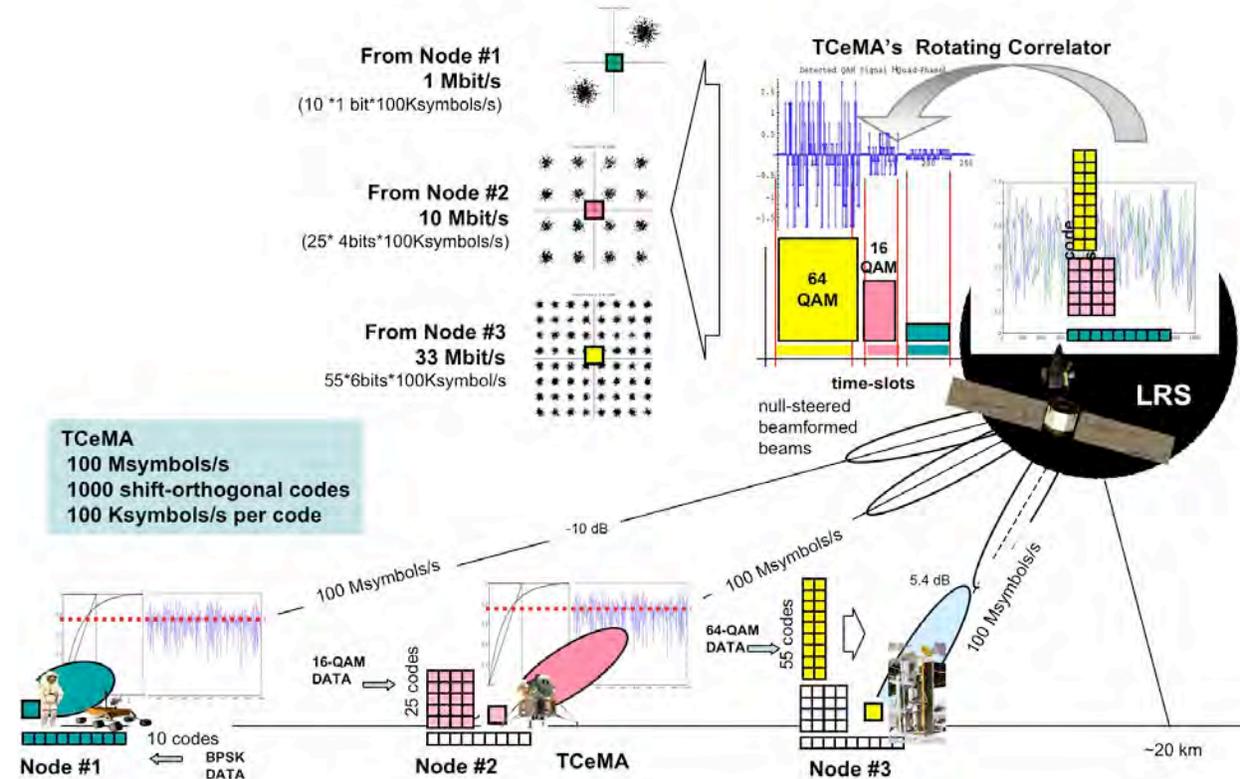


Figure 11. With TceMA, a spacecraft can receive simultaneously from multiple neighbors at difference effective symbol rates while using different amplitude-and-phase modulation in each cross-link

TDMA multiplexed signal in which the number of codes per TceMA-encoded burst equals the number of symbol times per TDMA burst. The advantage of TceMA over TDMA is that the detected signal results with a *processing gain* inversely proportional to the number of codes. In addition, with TceMA, the signals received from different spacecraft DO NOT have to be perfectly time-synchronized with each other. The temporal synchronization requirements for TceMA are similar to the synchronization requirements of TDMA, with time uncertainties ranging typically from a couple to tens of channel symbol times.

VI. □ Usage Scenarios

The difficulty for future lunar communications will be influenced by a number of parameters that have not been experienced by NASA in the previous Apollo era missions. These difficulties can be met and overcome by using the knowledge that NASA has been developing through years of research on leveraging terrestrial networking for space applications. NASA must meet the following challenges:

Communications with the Far Side of the Moon: During previous excursions, NASA has always maintained a direct communications path with previous lunar missions. With the adoption of a networking infrastructure, NASA has the ability to migrate away from point-to-point architectures and develop hop-to-hop based architectures where each intermediate node will route to the next best path. This will allow NASA to explore the far side of the Moon where point-to-point communications would not be possible.

Harsh Communications Environment: In the past 30 years, the focus of NASA has been on LEO-type missions where latencies and errors rates have not been excessive. With lunar missions, both latencies and error

rates will increase to the point where most traditional terrestrial protocols will fail. Technologies must account for these limitations by either developing new protocols or extensions to those existing. One of the more notable protocols is the Delay Tolerant Networking (DTN), which permits nodes to act as Store and Forward devices during periods where communications does not exist. With Mars communications, these factors will escalate.

Quality of Service (QoS): In a network environment, multiple data will be sharing and competing for resources. In the terrestrial Internet, QoS has been a significant research topic, but has not been widely implemented. In the space environment, NASA will have to determine how QoS will play a role so that higher priority data (e.g., commanding, scheduling, etc.) will have precedence over lower priority data (e.g., e-mails, science data downloads, etc).

A. Scenario for a Human on the Moon Communicating with Earth

The first scenario for consideration is human communications from the lunar surface to a ground station on Earth. Some of the considerations could include responses to commanding data, science data download, health and safety transmittals, etc. These types of data will have different corresponding QoS designations that must be considered. For this scenario, the assumption is that an astronaut will be sending voice data to Earth. As shown in Figure 2, lunar assets will be well connected. The nominal data path will be from the astronaut to the LRS and then transmitted to Earth, but, while this path is nominal, it must be determined by communication relay network whether that communications path is available. The communications scenario can be divided among the elements as follows:

1. Astronaut on the far side of the Moon initiates a conversation with Earth through the LRS constellation

The astronaut begins the scenario by talking into the headset and initiating the data transmission. Voice recognition software in the space suit radio (SSR) understands that the astronaut is calling a specific node (person or group of persons) on the Earth and attaches the appropriate packet headers to the voice over Internet protocol (VoIP) data to direct the data to that destination.

The SSR handles routing and bandwidth management. The SSR determines that the data must be routed to Earth and the next best hop will be to an LRS. The SSR requests access to an over-flying LRS - in a similar fashion to dialing a VoIP phone. The SSR conforms to the TCeMA MAC protocol to access an over-flying LRS. If an LRS is not in view, the voice data is captured in local space suit storage for automatic transfer when a link is formed (the storage also acts as a “black box” in the event the astronaut becomes incapacitated and cannot respond in real time when a link is established).

2. LRS constellation handles the VoIP call

The ASCoT software aboard each LRS is aware of the instantaneous positions of the rest of the constellation, other in-space nodes in the vicinity of the Moon, the Earth, and the location of the lunar surface nodes. These locations are updated and propagated throughout the constellation periodically. The in-view LRS uses TCeMA protocols in the low-rate S-band and beam focusing capabilities to listen to the surface area where the astronauts are known to be. The LRS grants the astronaut’s SSR request for access. It then has three options for handling the received data: if Earth is accessible and the data has sufficient priority for transmission, it can forward the data; if Earth is not in-view and the data has sufficient priority for transmission, the ASCoT software configures the LRS radio to send the data around the LRS constellation by shortest path to Earth; or if higher priority data is being transmitted or the next hop is not in view, which might be the case while the architecture is being established, the LRS node will store the data until connectivity is established and then ASCoT will forward the data. In most cases, the LRS will cross-link the data to the LRS in front of it in the same orbital plane since the forward satellite will likely be in-view or coming in-view of the Earth next. To maintain established communication links, ASCoT will direct LRS to LRS handoffs and choreograph the radio antenna systems as the constellation passes over a communicating lunar node location. Using the routable networking architecture, the data will normally not have to be stored due to asset access concerns. This will provide on-demand, end-to-end data communication services between entities anywhere on the Moon and those on the Earth.

3. VoIP data is captured and routed to the Earth destination

The ground station on Earth receives the data from the LRS that is in-view, reduces it to baseband and then to IP and places it on Earth-based networks. The data can be routed to the final destination using the terrestrial Internet on either an open or closed network. If the data is sensitive and routing will be through an open network, encryption or security protocols are employed.

B. Scenario for Humans on the Moon to Communicate Across a Broad Expanse of the Moon

This scenario assumes the communicating humans are not in line of sight of each other and so must communicate through the LRS constellation. The scenario does not discuss communication through surface networks.

1. Astronaut initiates a conversation with another in the lunar vicinity through the LRS constellation

As in the above scenario, the astronaut initiates this scenario by talking. The voice recognition software in the radio (SSR, vehicle, or habitat radio – depending where the astronaut is) understands that the astronaut is calling a specific person elsewhere on the Moon and attaches the appropriate packet headers to the voice over Internet protocol (VoIP) data to direct the data to that destination.

The ASCoT software in the radio determines that the data must be routed through the LRS constellation to get to the ultimate destination. The radio obtains access to the constellation as describe in the above scenario.

2. LRS constellation handles the VoIP call

The ASCoT software aboard each LRS is aware of the instantaneous positions of the source and destination nodes at their different locations on the lunar surface. These locations are updated and propagated throughout the constellation periodically. The in-view LRS uses TCeMA protocols in the low-rate S-band and beam focusing capabilities to listen to the surface area where the astronauts are known to be. The LRS grants the astronaut's SSR request for access. This LRS also has three options for handling the received data: if the destination astronaut is accessible and the data has sufficient priority for transmission, it can forward the data directly to her/him; if the destination astronaut is not in-view and the data has sufficient priority for transmission, the ASCoT software configures the LRS radio to send the data around the LRS constellation by shortest path to LRS that is over the destination; or if higher priority data is being transmitted or the next hop is not in view, which might be the case while the architecture is being established, the LRS node will store the data until connectivity is established with the destination astronaut and then ASCoT sends the data down. To maintain established communication links, ASCoT will direct LRS to LRS handoffs and choreograph the radio antenna systems as the constellation passes over both source and destination lunar node locations. This provides on-demand, end-to-end data communication services between entities anywhere on the Moon.

VII. Conclusion

NASA faces many technical challenges in its efforts to evolve an autonomous, robust, adaptive, delay and error tolerant networking architecture at minimum cost to support the solar system exploration imperatives in the "Vision for Space" directive from President Bush. In addition, NASA (with its private sector and international partners) will face financial uncertainties that impinge on how these technical challenges will be addressed. In this milieu, NASA is exploring many technology trade-offs and its plans are tentative and uncertain. Thus, it is unclear now what solar system networking architecture and technologies NASA will actually implement and evolve over the next twenty years or more.

This paper and others can take advantage of the current flux in NASA planning to propose advanced technologies to help NASA resolve the technical challenges. The current flux gives these technologies a better chance of being seriously considered and adopted by mission planners to actually fly on future space hardware.

To further that process of getting advanced ideas considered by NASA in a timely manner, this paper discussed the synergy generated by combining these advanced, innovative networking technologies into an integrated package:

- Advanced media access control (a new MAC with TCeMA) and
- Unique position-dependent/link-state data routing (ASCoT).

The MAC technology allows a node to self-discover its neighbors. It also provides variable-rate links and manages simultaneous communications with multiple in-space and surface clients at varying and rapidly changing distances with efficient power utilization.

The ASCoT networking technology uses attitude sensors, a time synchronization protocol and occasional orbit-corrections to maintain awareness of its instantaneous position and attitude in space as well as the orbital or surface location of its communication clients. ASCoT also manages intelligent and autonomous data routing among communications nodes by uniquely combining a modified form of positional routing (in 3-D) with efficient link-state routing managed by the MAC layer utilizing TCeMA. This positional, link-state (PLS) concept handles data movement among any number of multiple clients (including the Earth) that may be simultaneously in view; and if not in view, the relay satellite system will temporarily store the data from a client source and download it when the destination client comes into view.

The integration of the MAC and data routing networking technologies would enable a relay satellite system to provide end-to-end communication services for robotic and human missions in the vicinity, or on the surface of the Moon with a minimum of Earth-based operational support.

References

- ¹Schier, J. S., Rush, J. J., Williams, W. D., Vrotsos, P., "Space Communication Architecture Supporting Exploration and Science: Plans and Studies for 2010-2030," AIAA *1st Space Exploration Conference: Continuing the Voyage of Discovery*, 2005, Orlando, FL, URL: https://www.spacecomm.nasa.gov/doc_repository/architecture/AIAA_architecture.pdf [cited 28 April, 2006]
- ²Kulkarni, T. R., Dharme, A., Mortari, D., "Communication Architecture and Technologies for missions to Moon, Mars, and Beyond," AIAA *1st Space Exploration Conference: Continuing the Voyage of Discovery*, 2005, Orlando, FL, URL: http://www.aa.washington.edu/courses/aa420/references/Comm_AIAA-2005-2778.pdf [cited 28 April, 2006]
- ³Bose, P., Tkachenko, A., Govindan, R., Gnawali, O., "ASCoT: Autonomous Space Communications Technology Final Report", Report to NASA Glenn Research Center, June 2005
- ⁴Gnawali, O., Polyakov, M., Bose, P., Govindan, R., "Data Centric, Position-Based Routing in Space Networks," 2004 IEEE Aerospace Conference, Big Sky, MT.
- Bergamo, M., "Wireless Data Communications System," U.S. Patent No. 6,104,708, August 15, 2000.
- Preuss, R., Bergamo, M., "Data Communications System and Hybrid Time-Code Multiplexing," U.S. Patent No. 6,590,889 B1, July 8, 2003.
- Bergamo, M., "Constant Envelope Signal System and Method," patent pending.
- Bergamo, M., "High-Throughput Distributed Spacecraft Network (Hi-DSN: TCeMA Scheduling: Algorithm for Burst Transmission Time Calculation," Project Report, NASA Contract No. NAS3-01101, September 3, 2003.
- Bergamo, M., Hain, R., "High-Throughput Distributed Spacecraft Network (Hi-DSN: A Full Capability Simulation Study Using OPNET," Project Report, NASA Contract No. NAS3-01101, March 26, 2004.
- Bergamo, M., Ramanathan, S., "High-Throughput Distributed Spacecraft Network (Hi-DSN): System Design," Project Report, NASA Contract No. NAS3-01101, February 14, 2003.
- Bergamo, M., "High-Throughput Distributed Spacecraft Network (Hi-DSN): Physical-Layer API Specification," Project Report, NASA Contract No. NAS3-01101, October 23, 2003.
- Bergamo, M., "High-Throughput Distributed Spacecraft Network (Hi-DSN): MAC-Layer Design," Project Report, NASA Contract No. NAS3-01101, March 30, 2004.
- Hain, R., Ramanathan, S., "High-Throughput Distributed Spacecraft Network (Hi-DSN: Comparing Multiple Access Protocols for the Distributed Spacecraft Network: A Simulation Study using OPNET," Project Report, NASA Contract No. NAS3-01101, December 2002
- R. Hain, R., Ramanathan, S., Bergamo, M. and T. M. Wallett, "Comparison of TCeMA and TDMA for Inter-satellite Communications using OPNET Simulation," OPNET Work 2003, August 25 – 29, 2003.
- Bergamo, M., Dempsey, H., Strayer, T. W., "SpaceVPN Architecture: Space Network Environment, High-Throughput Distributed Spacecraft Network Services and Ground-Space VPN Buildout," Project Report, NASA AIST Contract No. NNC05CB16C, March 29, 2004.
- Bergamo, M., "High-Throughput Distributed Spacecraft Network: architecture and multiple access technologies, Computer Networks (Elsevier), April 2005, pp. 725-749.
- Bergamo, M., High-Throughput Distributed Spacecraft Network for SpaceVPNs: Reference Payload Specification," Project Report, NASA AIST Contract No. NNC05CB16C, March 10, 2006.