Towards A Routing Framework in Ad Hoc Space Networks

Chao Chen*
Department of Engineering,
Indiana University - Purdue University Fort Wayne, Indiana, USA
E-mail: chen@engr.ipfw.edu
*Corresponding author

Zesheng Chen
Department of Electrical and Computer Engineering,
Florida International University, Miami, Florida, USA
E-mail: zchen@fiu.edu

Abstract: The unique characteristics of space networks lead to different research approaches from those in terrestrial networks. In this paper, a routing framework called Space Gateway Routing (SGR) is proposed for routing through different autonomous regions (ARs) in ad hoc space networks. SGR has two integral parts: External SGR (ESGR) and Interior SGR (ISGR). ESGR addresses the delivery of different traffic types through ARs. Inside ESGR, the Location-Predicted Directional Broadcast (LPDB) is proposed for fast delivery of remote control messages and automatic data delivery. The Receiver-Initiated On-demand Routing (RIOR) is proposed for controlled data delivery, where the route discovery is initiated on-demand by the receiver, and routing tables are maintained in soft state at intermediate nodes. ISGR exchanges inter-AR routing information among border routers within an AR and schedules inter-AR message transmission. The Longest Queues (LQ) policy is proposed for contact allocation for AR border routers, and the Minimum Waiting (MW) policy is introduced for scheduling inter-region messages through an AR. Simulation results show that LPDB and RIOR are efficient both in message delivery and power consumption, and a combination of proposed LQ and MW policies achieves good delay and throughput performances.

Keywords: routing; ad hoc networks; space communication networks

Biographical notes: Chao Chen is an assistant professor with the Department of Engineering, Indiana University - Purdue University Fort Wayne. She received her Ph.D. degree from Georgia Institute of Technology in 2005. Her current research interests include routing in mobile ad hoc networks and space-based communication networks, modeling and performance evaluation of wireless opportunistic networks, and network security. Zesheng Chen is an Assistant Professor at the Department of Electrical and Computer Engineering at Florida International University. He received his Ph.D. degree from Georgia Institute of Technology in 2007. His research interests include network security and the performance evaluation of computer networks.

1 Introduction

The developments in space technologies are enabling the realization of scientific missions such as Mars exploration. These missions produce a significant amount of scientific data to be collected from remote space exploration sites and delivered to the Earth. Reliable control is also required to ensure the success of the delivery. An example ad hoc space network architecture shown in Fig. 1 helps to

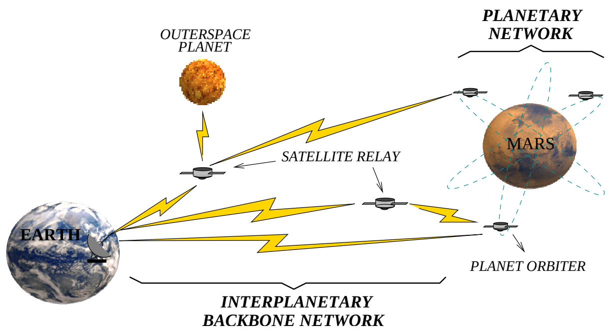


Figure 1: An example ad hoc space network architecture [1].

build a general space network architecture that combines differently challenged parts [1].

The main challenges that affect routing in ad hoc space networks are listed as follows [1]:

- *Long and Variable Propagation Delay:* Space communication links have extremely long and variable propagation delays. Node movement during propagation time must be considered in the process of route computation and message scheduling.

- *Intermittent Connectivity:* Link outage may occur as a result of natural reasons such as planetary body blockage and environmental interference. Furthermore, because of economical reasons, radio transceivers for space communications are shared, and link connectivity is scheduled to be episodic.

- *High Bit Error Rates:* The raw bit error rate can be in the order of 10^{-1} in the deep space environment [2]. Furthermore, burst errors that last in the order of minutes can also be expected.

- *Power Constraints:* The operation of space elements mainly depends on the re-chargeable battery using solar energy [3]. The use of nuclear power has also been explored in space applications. The high cost of nuclear power and the risk of radioactivity release in case of accident, however, prevent it from extensive communication usage.

- *Link Asymmetry:* The space link quality is different in opposite directions. The time-dependent nature of the network topology also causes space links to be asymmetric in delay and stability.

Most of these characteristics are unique to the space communication paradigm and thus lead to different research approaches from those in terrestrial networks. An ad hoc space network is also a special type of Delay-Tolerant Networks (DTNs) [4], where continuous end-to-end connectivity cannot be assumed. When the connectivity patterns of the network are known, Jain et al. formulate the DTN routing problem based on different knowledge about the network topology [5]. The proposed algorithms require error-free communications, and no effective solutions are given when unpredictable link failures occur. Techniques like erasure-coding that use redundancy to cope with failures in DTNs have also been proposed [6, 7]. However, these solutions do not specifically address the challenges in ad hoc space networks. Furthermore, mechanisms for gathering the forwarding information through the network are not discussed in these work.

In this paper, a new routing framework, *i.e.*, *Space Gateway Routing (SGR)*, is proposed based on the hierarchical architecture and specifically addresses the challenges of ad hoc space networks. SGR is illustrated in Table 1 and has two integral parts: *External SGR (ESGR)* and *Interior SGR (ISGR)*. ESGR addresses the delivery of remote control messages and scientific data through different autonomous regions (ARs) in an ad hoc space network, whereas ISGR is executed within an AR. Inside ESGR, two different routing protocols are proposed to address the requirements of different traffic types.The *Location-Predicted Directional Broadcast (LPDB)* is proposed in Section 3.1 for the fast and reliable delivery of remote control messages and automatic data reports. Paths to the destination are calculated en route based on predictable node locations and reachability information. These paths are used to direct and limit the forwarding area of the message broadcast. For controlled data delivery that contains large amounts of scientific data from remote exploration sites back to the Earth and requires high reliability, a combination of reactive and proactive approaches is utilized in the *Receiver-Initiated On-demand Routing (RIOR)* in Section 3.2. The route discovery is initiated on-demand by the receiver, and routing tables are maintained in soft state at nodes along the forwarding area. No end-to-end path is recorded for the data delivery. Link state exchanges during the data transmission process provide nodes with up-to-date path information. Furthermore, we demonstrate by simulation in Section 3.3 that proposed protocols are efficient in both message delivery and power consumption, and are suitable for different traffic types in space networks with respect to their specific requirements. ISGR exchanges inter-AR routing information among border routers within an AR and schedules inter-AR message transmissions. In Section 4, we give the problem definition of contact allocation and traffic dispatching, which are two important functionalities of ISGR. As a first attempt, we propose the *Longest Queues (LQ)* policy for contact allocation for border routers and the *Minimum Waiting (MW)* policy for the dispatching of inter-region messages through an AR. Simulation results in Section 4.3 show that the proposed ISGR policies produce good delay and throughput performances for crossing traffic in an AR.

Table 1: Proposed Routing Framework in Ad hoc Space Networks.

Space Gateway Routing (SGR)			
External SGR (ESGR)		Interior SGR (ISGR)	
remote control and automatic data reports LPDB	*controlled data delivery* RIOR	*contact allocation* LQ	*traffic dispatching* MW

The rest of this paper starts with a network description in Section 2. *External SGR (ESGR)* and *Interior SGR (ISGR)* are described in detail in Sections 3 and 4, respectively. Finally, Section 5 concludes the paper.

2 Network Description

The properties and assumptions of nodes and links in ad hoc space networks, and the proposed routing framework for this network architecture are described in this section.

2.1 Autonomous Regions (ARs) and AR Nodes

An ad hoc space network is composed of multiple *autonomous regions* (ARs). An AR contains communication entities that are located close (*i.e.*, much shorter than the interplanetary distance) to each other. These regions are called "autonomous" since the local nodes can communicate among themselves using a single common protocol family without intervention from other regions. The Mars planetary network in Fig. 1 is an example of AR. Here, we abstract an ad hoc space network by an AR topology that is composed of AR nodes and AR links.

An *AR node* is an abstraction of one AR. The location of an AR node can be represented by a position within the AR. Let the position of an AR node u at time t be represented by a vector $\vec{r}_u(t)$ originating from the sun. An AR node v is *reachable* by node u and called an *AR neighbor* of node u at time t, if there exists some $\Delta_{uv}(t) > 0$ that satisfies the following condition:

$$||\vec{r}_v(t + \Delta_{uv}(t)) - \vec{r}_u(t)|| = C \cdot \Delta_{uv}(t) < L^{uv}(t), \quad (1)$$

where C is the speed of light and $L^{uv}(t)$ is the reachable range limit between node u and v. If the transmission delay can be omitted, Eqn. (1) states that a signal transmitted at time t from node u can be received by node v at time $t+\Delta_{uv}(t)$. The reachability from u to v at time t is denoted by an *AR link* $l_{uv}(t)$.

2.2 Border Routers

The physical devices in ad hoc space networks that have long haul communication capability are called "*border routers*". Examples of border routers include planet surface elements such as Earth ground stations for NASA's Deep Space Network [8], relay satellites orbiting around planets such as Earth and Mars [9, 10], and other intermediate relay nodes such as mission-specific spacecrafts and relay stations at the Lagrange points of planets like Jupiter and Pluto [11]. These border routers are organized into different ARs according to their locations. As border routers move constantly, abiding by the orbital mechanics, this kind of node mobility is calculable by the knowledge of their trajectory information.

We assume that the border routers in an AR are time-synchronized and time is slotted with length T_{slot}. The local time at different ARs can be translated to a common time, *e.g.*, the coordinated universal time (UTC) [12]. Difference in time synchronization between ARs is omittable, compared to the propagation delay between ARs.

2.3 AR Links

As mentioned previously, an AR link represents the reachability of two AR nodes at a certain time. The signal transmission and reception on AR links are assumed to have the following properties:

- Inter-AR communications use different frequency band from that used for intra-AR communications. Therefore, signals targeted for receivers within the local AR and those to different ARs do not interfere with each other.

- Due to the extremely long distance, inter-AR communications via AR links require huge power consumption, and the cost per second of transmission can become very high. To reduce the transmission cost associated with AR links, directional antennas are used to increase the power efficiency toward targets. Moreover, a border router can only transmit to one AR neighbor (*i.e.*, an AR that is in range of at least one border router in this AR) in a timeslot.

- Omni-directional antennas or multiple directional antennas with different pointing angles are used for signal reception. A border router can receive signals from different ARs simultaneously and differentiate them by their distinct angle-of-arrivals (AOAs). Signals from different border routers in the same AR to the same AR neighbor collide with each other, as their AOAs are approximately the same.

- The incoming signal from an AR neighbor can be picked up by any local border routers that are within the transmission antenna's propagation cone.

As one of the communication properties of AR links, there should be only one border router in an AR that transmits to a specific AR neighbor within a timeslot. Therefore, some scheduling mechanism is needed to allocate the next contact toward an AR neighbor to one local border router. A *contact* describes an allocated time period when a border router can transmit to one AR neighbor whereas other border routers in the same AR cannot. In this paper, the length of a contact is a multiple of T_{slot}. An AR node u is "*connected*" to an AR neighbor v at time t only if v is reachable by u and one of u's border routers is in contact with v at time t. AR links may be intermittent and represented by a series of different contacts at different time.

2.4 Routing Framework

Our proposed routing framework has a similar hierarchical structure as that in the terrestrial Internet. The terrestrial Internet is organized into autonomous systems (ASes). Inside every AS, routing is accomplished through interior gateway protocols (IGPs). Inter-AS routing is based on an exterior gateway protocol (EGP), namely the border gateway protocol (BGP). BGP has two parts: external BGP (EBGP) used between ASes and interior BGP (IBGP) to exchange inter-AS routes within an AS. Similarly, an ad hoc space network can be organized into ARs. Different routing protocols can be developed for intra-AR communications to address specific challenges inside each AR, whereas a common routing protocol is needed for communication across the whole network. For this purpose, we propose a common routing framework, namely *Space Gateway Routing (SGR)*, for communication among ARs through an ad hoc space network. SGR has two integral pieces: *External SGR (ESGR)* and *Interior SGR (ISGR)*, where

- ESGR populates the forwarding information through the space network and selects AR paths for inter-AR messages, and
- ISGR routes inter-AR traffic through an AR and schedules inter-AR message transmission at border routers.

3 External SGR

The objective of space networks is to realize communication among in-space entities, allowing large volumes of scientific data to be collected from remote space exploration sites. The main traffic types carried through ad hoc space networks are:

- *Remote control:* Command and control messages sent from the Earth to remote devices at exploration sites. The delivery of remote control messages is time-sensitive, and requires high reliability acknowledgment.
- *Data delivery:* Scientific data delivered from exploration sites back to the Earth. We further classify the data delivery into two types with respect to the initiator and the service requirements:
  - *Automatic data delivery:* This type of data delivery is initiated by the mission devices at remote exploration sites, reporting mission status and some environmental data typically via repetitive transmissions. Automatic data delivery is time-sensitive and does not have strict reliability requirements.
  - *Controlled data delivery:* The Earth control center actively queries the mission devices for important scientific data delivery. In this application, the Earth center is aware of where to retrieve the scientific data from and initiates the data delivery. This type of data delivery requires a higher level of reliability.

For the above different traffic types that are transmitted through inter-AR communications, we propose

- Location-Predicted Directional Broadcast (LPDB) for remote control and automatic data delivery, and
- Receiver-Initiated On-demand Routing (RIOR) for controlled data delivery.

3.1 Location-Predicted Directional Broadcast

Although the locations of AR nodes are predictable, there exist unpredictable factors in the AR topology caused by different contact schedules at AR nodes, interferences, and power variation. Flooding is the most reliable method for the fast delivery of control messages and automatic data delivery, but at the expense of high network resource and power consumption. Therefore, we limit the broadcast area in space and time.

A control message or an automatic data delivery message contains fields of {`destAR, expireAt`}, where `destAR` identifies the destination AR node and `expireAt` indicates the time constraint set by the application. LPDB is done independently at each AR node and has two parts: *reference AR path computation* and *directional forwarding.*

3.1.1 Reference AR Path Computation

A reference AR path is computed according to the predictable AR topology and locally available information at the source AR. Specifically, given

- a time-varying AR topology $G(V, E(t), t)$, which composes of AR nodes V and a set of AR links $E(t)$,

- source s, destination d, and message arrival time t_s,

- the expected waiting time ω_{sv} at s to its AR neighbor $v, \forall v \in \mathcal{N}^s$, where $\mathcal{N}^s$ is a set of possible AR neighbors of AR s,

compute a fastest traversal AR path, which is a concatenation of possibly time-disjoint AR links $P_{s \to d} = (l_{sv_1}(\tau_0), l_{v_1 v_2}(\tau_1), ..., l_{v_{m-1}d}(\tau_{m-1}))$, where $l_{v_{i-1}v_i}(\tau_{i-1})$ is an AR link at time τ_{i-1}. Then $\pi = (s, v_1, ..., v_{m-1}, d)$ is the topological AR path, and $\tau = (\tau_0, \tau_1, ..., \tau_{m-1})$ shows the departure times at the AR nodes on the path and is computed by $\tau_0 = t_s + \omega_{sv_1}, \tau_i = \tau_{i-1} + \Delta_{v_{i-1}v_i}(\tau_{i-1}) + \epsilon_i, 0 < i \le m-1$, where ϵ_i is the message buffering time at node v_i. The above problem can be solved using an extension of Dijkstra's algorithm in time-dependent networks [13], where the fixed link cost is replaced by the sum of message buffering time and the time-dependent link propagation delay $(\Delta_{uv}(t))$. In the first hop, this buffering time is the locally available expected waiting time (ω_{sv}). With the queuing delay omitted, the message buffering time (ϵ_i) in later hops is approximated by the waiting time for a link to be reachable, which can be calculated by the predictable location information.

The AR path computed in this way only represents the shortest-delay path under the condition that the queuing delay can be omitted at the computed departure time $(\tau_i, 0 < i \le m-1)$ and the intermediate nodes are ready to pick up the messages at the reception time. When scheduling or retransmission delays the messages, however, the computed AR path may not be optimal or exist any more. The computed AR path is used just as a reference to direct message forwarding and is thus called the "*reference AR path*". As the actual message delivery can deviate from the pre-calculated timeline, intermediate AR nodes update the reference AR path as messages traverse the space network.

3.1.2 Directional Forwarding

Message forwarding is limited in *space* and *time*. Specifically, suppose that an AR node v receives a message at time t_0, the topological AR path from v to the destination d is computed as $\pi = (v, v_1, ..., v_{m-1}, d)$, and the departure timeline is $\tau = (\tau_0, \tau_1, ..., \tau_{m-1})$. Then, only AR neighbors that lie within the *forwarding cone* within time interval $[t_0, \tau_0 + T_{thresh}]$ can receive a copy of the message, where T_{thresh} is a parameter set by the application or AR nodes.

As shown in Fig. 2, the forwarding cone contains the space that is within node v's transmission power range L^v and limited within cone angle θ around the axis from v to d. The *cone angle* is calculated by:

$$\theta(\pi, t) = \max_{v_i \in \pi \backslash \{v\}} \{\angle v_i v d + \delta\}, \tag{2}$$

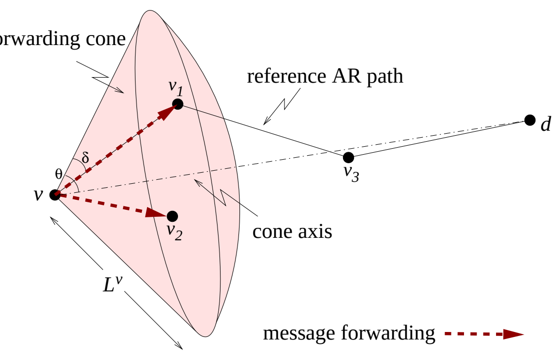


Figure 2: Location-predicted directional broadcast.

where δ is a parameter that controls the width of the forwarding cone. In order to adjust the forwarding cone to the movement of AR nodes, $\angle v_i v d$ is computed by the predicted location of node v_i on the reference AR path, *e.g.*, the location of v_i on the path is represented by $\vec{r}_{v_i}(\tau_{i-1} + \Delta_{v_{i-1}v_i}(\tau_{i-1}))$.

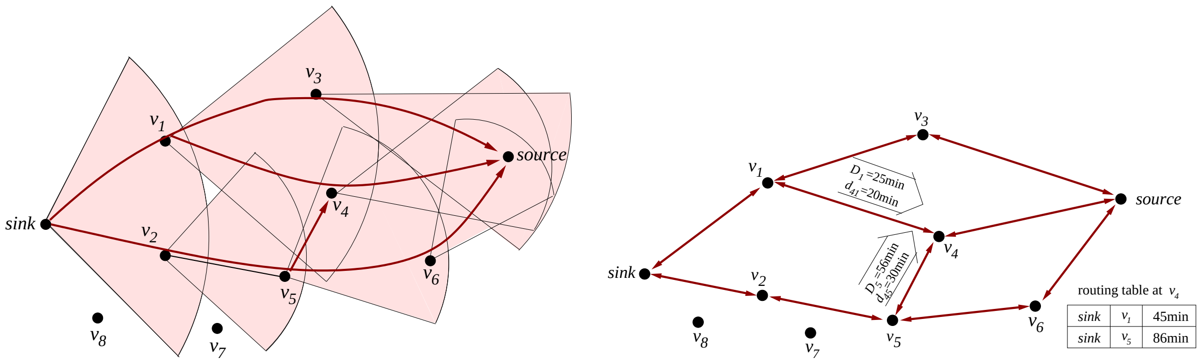
Remarks: The difference between LPDB and traditional location-aware routing protocols like LAR [14] and DREAM [15] is that no network-wide flooding is needed in LPDB to obtain nodes' location information, which can be calculated according to the orbital mechanics. Furthermore, the network connectivity intermittency caused by predictable reasons is addressed by allowing message buffering at the AR nodes. Directional forwarding provides multipath routing near the reference AR path to handle link unreliability and speed up end-to-end delivery.

3.2 Receiver-Initiated On-demand Routing

The Earth-controlled data delivery carries scientific data that are usually unprocessed and very large in volume. Therefore, flooding and the directional broadcast in LPDB would consume very high network resources. This type of traffic also requires high reliability, which is difficult to achieve in the space environment without redundant delivery or maintenance of up-to-date routing information. Since the Earth control center knows *when* and *where* this type of data needs to be gathered, we propose the use of on-demand route discovery initiated by the receiver, *i.e.*, the Earth control center. Routing tables at the intermediate AR nodes that are possibly on the data delivery path are built on-demand and maintained in soft state by exploring the link status and the load distribution. The proposed routing protocol is called *Receiver-Initiated On-demand Routing (RIOR)*.

3.2.1 Route Discovery

For convenience, we call the Earth control center as the "*sink*" in this application scenario. The route discovery



(a) `RREQ` message forwarding. (b) `KeepAlive` message exchange.

Figure 3: Route discovery for controlled data delivery.

contains two parts: `RREQ` notification and `KeepAlive` exchange, and routing table maintenance.

1) `RREQ` *notification and* `KeepAlive` *exchange*

At some time long enough (considering the propagation delay between the sink and the data source) before the data delivery will start, the sink initiates route discovery by sending out an `RREQ` control message to the data source AR periodically with interval T_{RREQ}. The delivery of `RREQ` messages follows the LPDB scheme as in Section 3.1.

The reception of the `RREQ` message initiates periodical `KeepAlive` exchange between neighboring AR nodes: *i.e.*, `KeepAlive` requests from the receiving AR node to the message sender, and `KeepAlive` replies in the reverse direction. The `KeepAlive` exchange interval T_{KA} is much smaller than T_{RREQ}. The `KeepAlive` exchanges serve for two purposes: To measure the delay and monitor the quality of AR links, and to build route entries to the sink AR.

As a result of the constant movement of AR nodes, new rounds of `RREQ` message transmission are initiated periodically until the expected data arrives the sink or until a specified timeout limit is reached. Later `RREQ` messages may follow different reference AR paths to the data source. AR nodes on the new forwarding paths exchange `KeepAlive` messages with their neighbors and maintain route entries to the sink. This is to adapt the area of the control message exchange to node movement and AR link condition changes.

2) Routing table maintenance

Routing tables at the intermediate AR nodes are built upon the reception of the `KeepAlive` reply messages. A route entry has three fields: (`sink`, `nh`, `delayToSink`), where `sink` is the destination AR; `nh` specifies one AR neighbor to reach `sink`; and `delayToSink` denotes the delay from the local AR to the `sink` by way of `nh`. There may be multiple route entries for the same `sink`, enabling multipath routing and providing alternate paths when one path fails.

A new route entry is built as follows:

- Upon receiving the `KeepAlive` request, an AR node (u) records the directional link delay (d_{vu}) from the sending node (v). d_{vu} is the time elapsed from the transmission of the request at v to its reception at u.

- Node u selects the minimum `delayToSink` value (D_u) in all entries associated with the same sink.

- A `KeepAlive` reply is sent in the reverse direction (from u to v), containing the link delay (d_{vu}) and the minimum `delayToSink` value (D_u).

- Node v then retrieves the information (d_{vu} and D_u) from the `KeepAlive` reply, and builds a new route entry with parameters of (`sink`, `nh`=u, `delayToSink`=$d_{vu} + D_u$).

Once a route entry is built, the `delayToSink` is updated as the value contained in the latest `KeepAlive` reply to capture the current link property. If an AR node has not received any `RREQ` message from one AR neighbor for a long time, it stops `KeepAlive` message exchange with this neighbor, and the corresponding route entry is removed as well.

The data delivery follows the information in local routing tables. The `nh` with the minimum `delayToSink` value is chosen as the next-hop AR (data can also be forwarded to multiple next-hops). As the `delayToSink` value is augmented with the propagation of the `RREQ` message from the sink to the source, the delay of the previous part of the path that the `RREQ` message traverses may be outdated. The accuracy of this `delayToSink` value therefore decreases as the distance from an AR node to the sink grows. When the `RREQ` message first reaches the source AR, the minimum delay path seen from the source may not be optimal. Nevertheless, `KeepAlive` exchanges continue updating the route state during the data delivery and refining the remaining path toward the sink. As the data message traverses closer to the sink, `delayToSink` approaches its actual value.

`RREQ` message forwarding and `KeepAlive` message exchanges in an example AR topology are depicted in Fig.s 3(a) and 3(b), respectively. The forwarding of the `RREQ` message from the sink initiates `KeepAlive` exchanges between AR neighbors. The AR nodes in the forwarding paths, *i.e.*, nodes v_1 to v_6, build and update route entries to the sink. The routing table at node v_4 is given as an example. Node v_4's routing entries to the sink are built according to the `KeepAlive` replies from v_1 and v_5.

3.2.2 Route Repair

As a data message traverses the network, if an intermediate AR node finds that the `nh` with the minimum `delayToSink` value is not reachable or that it cannot receive an acknowledgment from the `nh` after a certain number (K) of consecutive retransmission attempts, a link failure to this `nh` is detected. In this case, it reroutes the data message to an alternate, possibly longer delay path in the routing table. If no alternate path is available, an `RREQ` message is initiated and sent to the sink periodically with an interval T_{RREQ} using the LPDB scheme. `KeepAlive` exchanges are also initiated between AR neighbors along the forwarding path to the sink. For example, suppose a data message takes the path of (source, v_4, v_5) in Fig. 3 and finds that the link to v_2 is not reachable. As no alternate path is available at node v_5, an `RREQ` message is sent from v_5 to v_7 in v_5's forwarding cone and pass on by v_7 to v_8 and the sink. Periodic `KeepAlive` exchanges are initiated between the new neighboring nodes on the forwarding path. Thus, new paths such as (source, v_4, v_5, v_7, v_8, sink) can be found.

Remarks: RIOR executes reactively to the application requests without network-wide topology propagation. It does not look for a specific route used for the data delivery session as other on-demand routing protocols such as DSR [16] and AODV [17], since this route may be obsolete after the long route discovery phase. Upon the detection of link failures, DSR and AODV notify the sender node, which then restarts the route discovery process. In RIOR protocol, on the other hand, `KeepAlive` messages are utilized to obtain the up-to-date link property, so that the updated route entries reflect more recent delay metrics. The maintenance of multiple route entries provides alternate routing options. New route discovery can be issued at any intermediate node that encounters link failure. These mechanisms help RIOR adapt fast to the changes in ad hoc space networks with long and variable delays. Moreover, RIOR does not assume link symmetry, and all delay measurements are directional.

Directed diffusion [18] proposed for wireless sensor networks is also a receiver-initiated protocol. The feedback-based adjustment and the end-to-end negotiation in directed diffusion, however, are not applicable in space networks with long delays.

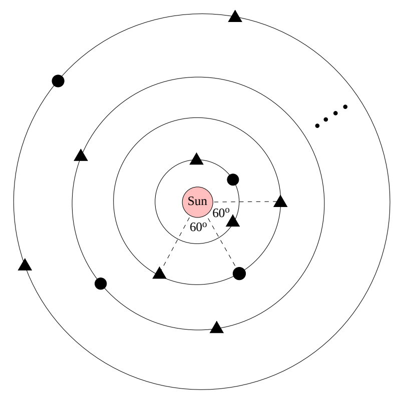
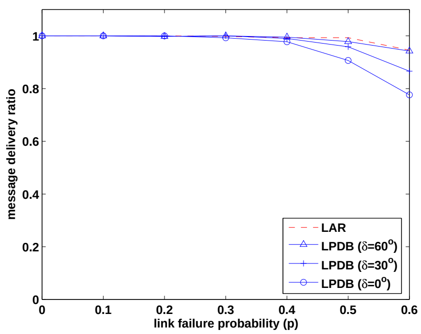


Figure 4: Network model for the simulated ad hoc space network.

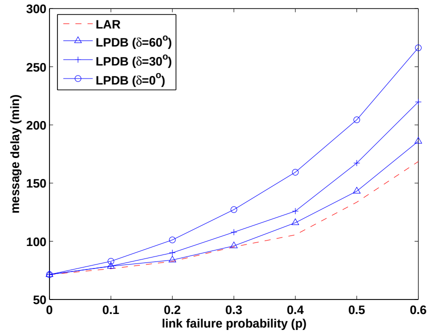
3.3 Performance Evaluation of LPDB and RIOR

An event-driven simulator on C++ is developed to evaluate the performance of proposed routing protocols. The AR nodes in the network model are 9 planet ARs and 18 planetary Lagrange ARs. To simplify the model, all planets move around the sun in circular orbits in the same plane. The orbit radii of the planets are from 1 to 9 AU (1 AU $\approx$ 149,600,000 km), with 1 AU in between. The planet orbiting period $\mathcal{T}$ (in year) is decided by the Kepler's third law: $\mathcal{T}^2 = a^3$, where a is the orbit radius in AU. The Lagrange ARs are on the same orbits, 60^o ahead or behind of the associated planets. The reachable range limits of all AR pairs are set as 5 AU. The network model is shown in Fig. 4, where the dots refer to the planet ARs and the triangles are the planetary Lagrange ARs. All nodes circulate around the sun according to the orbital mechanics.

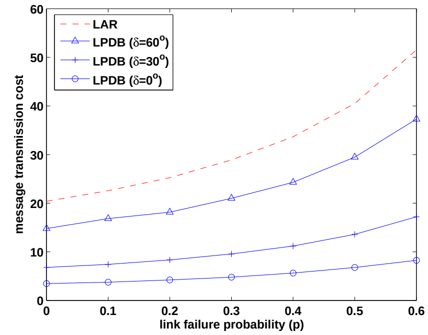
In each simulation round, the initial positions of planet AR nodes are random with central angles uniformly distributed in $[0^o, 360^o)$. Control/Data messages are sent in 1 hour interval. All results are the averages of 20 simulation rounds, each of 1 day long. Control/Data message delivery performances are evaluated under different link failure probabilities (p). Each link between AR neighbors is prone to failure, and the failure is independent across different links. Three message delivery metrics are measured to evaluate the routing performance: the *delivery ratio* and the *average delay* of successfully delivered messages; and the *transmission cost*, *i.e.*, the average number of transmissions for each successful message delivery.



(a) Message delivery ratio.

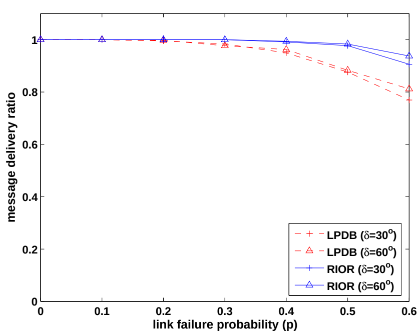


(b) Message delay.

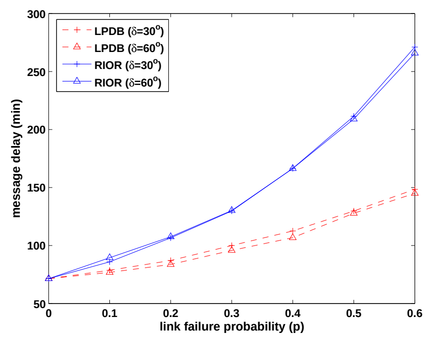


(c) Message transmission cost.

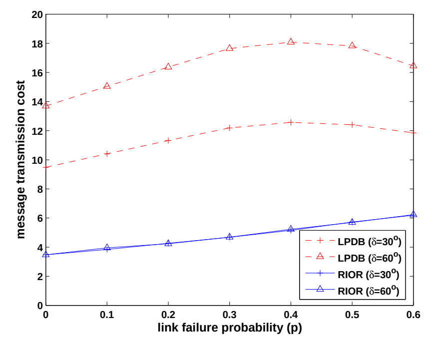
Figure 5: Performance comparison between LPDB and LAR.



(a) Message delivery ratio.



(b) Message delay.



(c) Message transmission cost.

Figure 6: Performance comparison between RIOR and LPDB.

3.3.1 LPDB for Remote Control and Automatic Data Delivery

For remote control and automatic data delivery, we compare LPDB with *Location-Aided Routing (LAR)*, which is the LAR scheme 2 in [14] with $\alpha = 1$ and $\beta = 0$, *i.e.*, a message is forwarded from the current AR node to all neighbors that are closer to the destination. The computation of distance considers the movement of the destination node. To measure the effect of the forwarding cone angle on the performance of the LPDB scheme, three different values of δ are chosen in the simulation. The case $\delta = 0^o$ provides the performance bound for LPDB.

Messages are sent from node 0 (orbit radius = 1 AU) to node 7 (orbit radius = 8 AU) in a store-and-forward manner, with a maximum life time of 6 hours. Each AR node stores a copy of a message and should make sure that every next-hop AR gets a correct copy of the message via per-hop acknowledgment before it removes its local copy.

Fig. 5(a) shows the message delivery ratio of these two schemes under different link failure probabilities (p). For LPDB, the delivery ratio is always higher than 90% when $p \leq 0.5$. As the forwarding cone angle increases, more messages are successfully delivered to the destination. LAR provides a high degree of redundancy for the message delivery. Thus, it achieves low end-to-end delay but with high transmission cost, as can be seen in Fig.s 5(b) and 5(c), respectively. As the forwarding cone angle increases, LPDB's delay and ratio performance gets closer to LAR. Even in the case of $\delta = 60^o$, LPDB can achieve the same delivery ratio but with less transmission cost than LAR at $p = 0.6$.

In summary, LPDB balances between reliability and redundancy, as well as between the delay and the transmission cost. AR nodes can change the forwarding cone angle by adjusting δ, which in turn controls the message delivery performance.

3.3.2 RIOR for Controlled Data Delivery

We compare our RIOR scheme with LPDB, where the parameter δ in LPDB controls the width of the forwarding cone, whereas δ in RIOR controls the forwarding cone of `RREQ` messages. The transmission cost refers to the average number of data message transmission for each successful delivery. We set $T_{RREQ} = 1$ hour, $T_{KA} = 10$ minutes, and $K = 3$. Data messages are sent from node 7 to node 0. The maximum life time for data messages is 10 hours. No multipath forwarding is utilized in RIOR in our simulation.

Fig. 6(a) shows that RIOR always keeps the delivery ratio higher than 90%, even when p is as high as 0.6. Compared with RIOR, LPDB is less reliable especially when p is high. This is because LPDB does not provide any route repair under link failures. As a benefit of the multipath transfer, LPDB results in a lower message delay than RIOR as in Fig. 6(b), thus is more suitable for messages that require fast delivery. When the transmission cost is concerned, as seen in Fig. 6(c), RIOR costs much less data overhead, which is around 30% to 50% of LPDB with $\delta = 60^o$ and 30^o, respectively. Note that the transmission cost of LPDB does not show consistent growth as p increases. When p grows beyond 0.4, the message delivery ratio starts to decrease fast, more messages get dropped or lost early on the delivery path, and only “lucky” ones reach the destination. As the result, the average transmission cost for successful end-to-end deliveries is lower if compared with the cases when p is smaller.

4 Internal SGR

Forwarding cone computation and routing table maintenance for inter-AR traffic are carried out by External SGR (ESGR), whereas the actual information exchange and message delivery between ARs are functions of Interior SGR (ISGR). ISGR directs the inter-AR traffic through each AR by way of *border routers*, as long as they can reach the next-hop AR neighbors.

As described in Section 2.3, in a timeslot with length T_{slot}, a border router can only transmit to one AR neighbor, and only one border router in an AR can transmit to a certain AR neighbor. To avoid signal collision, a *contact allocation* policy is called to schedule the contacts for each border router to its AR neighbors. Meanwhile, a *traffic dispatching* policy is needed to direct each incoming message to an egress router.

In the presence of intermittent links, border routers need to decide not only the next-hop destination but also the time at which to send a message. The routing function in border routers is conceptually described in [19] and has three parts: the contact scheduler, the route evaluation algorithm, and the dispatcher algorithm. In our paper, the second is addressed in ESGR, whereas the first and the last are included in ISGR in our routing framework.

4.1 Problem Modeling

Suppose that the number of border routers in an AR is N and the transmission to a specific AR neighbor can be thought as a queuing model consisting of a single server and N parallel queues, where the server is the AR neighbor and each queue corresponds to a border router. If an AR has M AR neighbors, the transmission to these neighbors contains M such queuing models with inter-dependent queue lengths and server working schedules.

ISGR is executed inside each AR. Consider AR u, given

- $\mathcal{B}^u$, a set of border routers in AR u,
- $\mathcal{N}^u(t)$, u's AR neighbors set at timeslot t,
- $Q_{iv}(t)$, the queue length of a border router $i \in \mathcal{B}^u$ to an AR neighbor $v \in \mathcal{N}^u(t)$ at timeslot t,
- $R_{iv}(t)$, a binary variable describing the reachability of an AR neighbor $v \in \mathcal{N}^u(t)$ by a border router $i \in \mathcal{B}^u$ at timeslot t [1].

The two major functionalities can be described as:

- *Contact allocation:* For each border router $i \in \mathcal{B}^u$, decide its target AR neighbor at timeslot t, $T_i(t) \in \{\mathcal{N}^u(t), e\}$, where e stands for the IDLE mode. According to the assumptions in Section 2.3, no more than one border router can simultaneously transmit to the same AR neighbor, *i.e.*, if $j, i \in \mathcal{B}^u$, $j \neq i$, and $T_i(t) \neq e$, then $T_j(t) \neq T_i(t)$. For a specific border router i, the allocated values of $T_i(t)$ give its contact schedule.
- *Traffic dispatching:* For an incoming message ξ arriving at t in AR u with next-hop AR neighbor v, select $E_\xi(t) \in \mathcal{B}^u$, the border router that performs as its egress router.

The objective function can be the maximum AR throughput, the minimum buffering delay of incoming messages, or load sharing among border routers, etc., depending on application requirements or resource availability.

4.2 Possible Solutions

We propose two simple centralized policies for the problem of contact allocation and traffic dispatching, respectively. The objectives of these policies are high message throughput and low buffering delay. For simplicity, these solutions assume that all contacts from any border router to any AR neighbor have the same link capacity. However, our proposed solutions can be easily extended to the case that the AR links have different capacity values.

1) Longest Queues (LQ) policy for contact allocation: Allocate the next timeslot to the border routers that can reach and also have the longest queues associated with the AR neighbors. The goal for this policy is to transmit as much inter-AR traffic load as possible, thus to approach maximum throughput.

[1]For clarity, we say that an AR neighbor $v \in \mathcal{N}^u(t)$ is reachable by a border router $i \in \mathcal{B}^u$ at timeslot t, *i.e.*, $R_{iv}(t) = 1$, if Equation (1) in Section 2.1 is satisfied in timeslot t and the signal transmission from i to v is not blocked by the body of AR u.

The LQ policy is executed at a *contact allocator*, which contains the queuing information of all border routers. In detail, before the start of each timeslot, every border router reports to the contact allocator the queue lengths associated with its AR neighbors. The comparison of queue lengths considers different lengths of messages. For example, if there are two messages in a queue, with lengths of 4 KB and 6 KB, then the queue length is 10 KB. A simplified version of the LQ policy is executed as follows:

Algorithm 1 LQ policy

Input: $Q_{iv}(t), \forall i \in \mathcal{B}^u, \forall v \in \mathcal{N}^u(t)$
Output: $T_i(t), \forall i \in \mathcal{B}^u$

Set $\mathcal{S} = \mathcal{N}^u(t)$; $T_i(t) = e, \forall i \in \mathcal{B}^u$
while $\mathcal{S} \neq \emptyset$, **do**
  $Q^* = \max_{(i \in \mathcal{B}^u, v \in \mathcal{S})}\{Q_{iv}(t) \mid T_i(t) = e, R_{iv}(t) = 1\}$
  if $Q^* = 0$, **then**
    break
  end if
  $(i^*, v^*) = \arg\max_{(i \in \mathcal{B}^u, v \in \mathcal{S})}\{Q_{iv}(t) \mid T_i(t) = e, R_{iv}(t) = 1\}$
  Allocate $T_{i^*}(t) = v^*$
  $\mathcal{S} = \mathcal{S} \backslash v$
end while

2) Minimum Waiting (MW) policy for traffic dispatching: Direct a message to the border router that is expected to have the minimum waiting time to serve new traffic. The goal of this policy is to achieve shorter delay.

The MW policy can be written as:

Algorithm 2 MW policy

Input: message ξ with next-hop AR v arriving AR u at t
Output: egress router $E_\xi(t) = \arg\min_{i \in \mathcal{B}^u}\{\omega_{iv}\}$

To execute the MW policy, a central *traffic dispatcher* needs to gather queuing information from all border routers. The estimation of expected waiting time (w_{iv}) is based on the following information: the contact schedule in the current timeslot, current queue lengths at each border router (assuming first-in-first-out scheduling), and the reachability schedules of all border routers. So w_{iv} consists of the waiting time for the nexthop AR v to be reachable and the time for node i to finish serving messages in its queue to AR neighbor v. The former can be decided by node i and AR v's trajectory information, whereas the latter needs to consider the current queuing information and the AR link capacity.

Message ξ is encapsulated and sent to the selected egress router $E_\xi(t)$, which then performs de-capsulation. If $E_\xi(t)$ is not allocated to AR neighbor v at the current timeslot t, the message is buffered and waiting for future contact opportunity. After the decision of the egress router, the expected waiting time of AR node u to AR neighbor v is represented by the waiting time from the egress router to v, *i.e.*, $\omega_{uv} = \omega_{iv}$, where $i = E_\xi(t)$. ω_{uv} is also provided to ESGR and helps compute the reference AR path in LPDB, as described in Section 3.1.1.

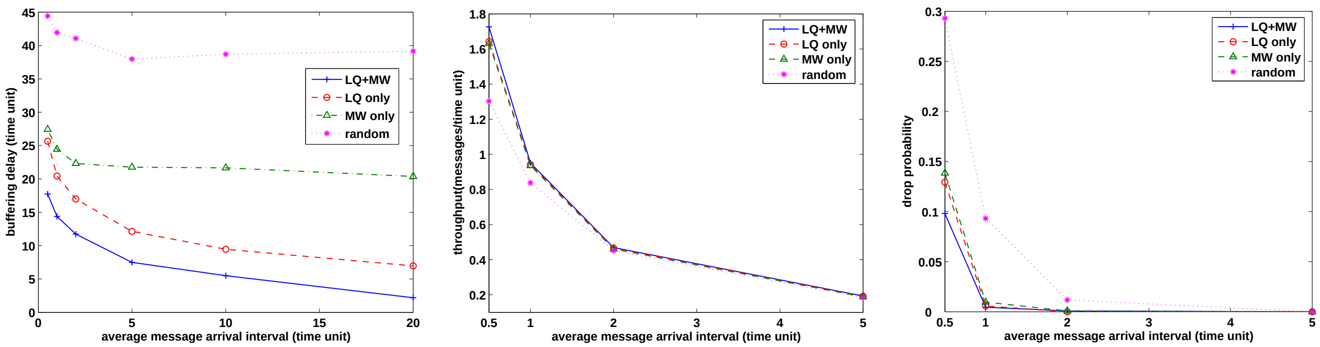
4.3 Performance Evaluation of ISGR

The evaluation of ISGR is done by modeling the contact allocation and traffic dispatching processes in a single AR u, which has N border routers and M possible AR neighbors. To simplify the evaluation, it is assumed that these N border routers have the same reachability pattern towards the AR neighbors, *i.e.*, the binary reachability variables towards an AR neighbor v satisfy $R_{iv}(t) = R_{jv}(t) = R_v(t), \forall i, j \in \mathcal{B}^u$, where t specifies any timeslot, and the reachability pattern in each timeslot is generated randomly and independent from each other.

Four combinations of contact allocation and traffic dispatching policies are evaluated:

- *LQ+MW:* Our proposed *Longest Queues (LQ)* policy combined with our *Minimum Waiting (MW)* policy, which are explained in Section 4.2.
- *LQ only:* LQ policy with random traffic dispatching, *i.e.*, incoming messages randomly choose one of the N border routers as the egress router.
- *MW only:* Random contact allocation with MW policy, *i.e.*, the allocation of contacts in the current timeslot is only based on the knowledge of the reachability schedule, while the queue lengths at border routers are ignored. The contact to a certain AR neighbor v is randomly allocated to a border router if the reachability variable $R_v(t)$ at the current timeslot t is 1.
- *Random:* Random contact allocation and random traffic dispatching without considering any queuing information.

In our simulation, messages are of the same fixed length, and the message arrival process is Poisson with an arrival rate of λ. A message randomly chooses one of the M AR neighbors as its next-hop. The message transmission rate μ is fixed. The parameters used in the simulation are: $N = 3, M = 5$, $T_{slot} = 10$ timeunit, $\mu = 1$ message/timeunit. The queue limit for each AR neighbor at border routers is set to 10 messages. If a message finds that the queue at the egress router towards the next-hop AR neighbor exceeds the queue limit, this message is dropped due to buffer overflow. We evaluate the performance of the above four combinations of ISGR policies under different values of message arrival rate (λ). The results are averages of 10 simulation runs, where each run lasts for 1000 timeunits. The performance metrics under evaluation are the *message*



(a) Message buffering delay. (b) Message throughput. (c) Message dropping probability.

Figure 7: Performance comparison among different ISGR policies.

buffering delay, the *message throughput*, and the *message dropping probability*.

Figure 7(a) shows the delay performance of the four policy combinations. If we ignore the queuing delay and remove the restrictions on the transmission on AR links, *i.e.*, a border router can transmit to different AR neighbors, and different border routers can transmit to the same AR neighbor at the same time, then an incoming message can be transmitted at the earliest time that its next-hop AR neighbor becomes reachable to a border router. This time is referred to as the "minimum transmit bound." The depicted "message buffering delay" in the figure (y-axis) is the difference between the actual message transmit time and this minimum transmit bound. It accounts for the portion of the message delay that is caused by contact allocation and traffic dispatching. From this figure, it can be seen that as the message arrival rate decreases, *i.e.*, the average message arrival interval increases, messages are buffered for a shorter period of time. Our proposed "LQ+MW" policy causes minimum buffering delay among the four combinations. When only LQ or MW policy is implemented, the message buffering delay is also reduced compared to that under the random case.

The message throughput performance is shown in Figure 7(b), which also confirms that the "LQ+MW" policy achieves the highest message throughput. The message dropping probability as a result of buffer overflow is shown in Figure 7(c), where the queue limit is set to 10 messages. The dropping of messages starts (for all cases except random) when the message arrival interval decreases below 2 timeunit. The "LQ+MW" policy causes the least message dropping among the policies. The "LQ only" and "MW only" policies also have much less dropping compared to the random policy. When we set the average message arrival interval to 5 timeunits and change the queue limit, the message dropping probabilities of these policies except the random case are similar when the queue limit decreases, where the "LQ+MW" policy still achieve the least dropping as depicted in Figure 8. It can also be concluded from this figure that message dropping can be effectively controlled by increasing the queue limit. This is easy to achieve by employing large storage space at border routers.

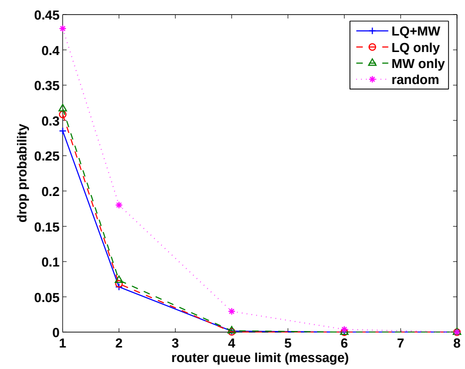


Figure 8: Message dropping probability under different queue limits.

In summary, the combination of LQ and MW policies achieves low message buffering delay, high message throughput, and low message dropping probability by considering contact schedules and queue lengths at border routers.

5 Conclusions and Discussions

In this paper, we propose a routing framework, called Space Gateway Routing (SGR), for routing through different autonomous regions in ad hoc space networks. To address the challenges in the space communication environment and meet the application requirements of control and data delivery, Location-Predicted Directional Broadcast (LPDB) and Receiver-Initiated On-demand Routing (RIOR) are proposed in the realm of External SGR (ESGR). LPDB provides the fast delivery of messages with considerably smaller sizes and lower reliability requirements; whereas RIOR achieves higher data delivery reliability when there is no strict time constraint. LPDB and RIOR can be utilized to deliver different types of traffic in ad hoc space networks, addressing specific requirements in each application scenario. Simulation results show that LPDB and RIOR are efficient in both message delivery and power consumption. For contact allocation and traffic dispatching, which are two important functionalities of Interior SGR (ISGR), we give the problem definition and further propose two preliminary policies: the Longest Queues (LQ) policy and the Minimum Waiting (MW) policy, respectively. Simulation results show that a combination of proposed LQ and MW policies achieves good delay and throughput performances.

Further exploration of the routing framework is needed to address the following issues:

- The contact allocation and the traffic dispatching policies are correlated to each other. For instance, the traffic dispatching policy directs traffic to different border routers, and thus affect the traffic arrival rates and queue lengths at each border router, which are important decision factors of the contact allocation policy. On the other hand, the contact allocation policy affects the message buffering time at each border router, which in turn influences the decision of the traffic dispatching. Therefore, we plan to study these two policies jointly to achieve the best performance.
- The performance of ESGR is affected by the contact allocation and traffic dispatching policies in ISGR. For example, in LPDB, the longer the messages need to wait at an AR to be serviced, the less accurate the computed reference AR path will be. RIOR requires periodical `RREQ` and `KeepAlive` message exchange between AR neighbors; the scheduled property of link contacts between AR neighbors would probably delay the exchange, which in turn affects the timely propagation of routing information. Possible improvements include priority-setting different types of messages (*e.g.*, `RREQ` and `KeepAlive` control messages, and application data messages) for queuing and bandwidth reservation for applications with certain QoS requirements.

REFERENCES

[1] I. F. Akyildiz, O. B. Akan, C. Chen, J. Fang, and W. Su, "InterPlaNetary Internet: state-of-the-art and research challenges," *Computer Networks Journal (Elsevier Science),* vol. 43, no. 2, pp. 75-112, Oct. 2003.

[2] R. C. Durst, P. D. Feighery, and K. L. Scott, "Why not use the standard Internet suite for the Interplanetary Internet?" *http://www.ipnsig.org/reports/TCP_IP.pdf.*

[3] H. Oman, "Deep space travel energy sources," *IEEE Aerospace and Electronics Systems Magazine,* vol. 18, no. 2, pp. 28 -35, Feb. 2003.

[4] S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, and K. Scott, "Delay-tolerant networking: an approach to Interplanetary Internet," *IEEE Communications Magazine*, vol. 41, no. 6, pp. 128-136, June 2003.

[5] S. Jain, K. Fall, and R. Patra, "Routing in a delay tolerant network," in *Proc. of ACM SIGCOMM*, pp. 145-158, Aug.-Sept., 2004.

[6] Y. Wang, S. Jain, M. Martonosi, and K. Fall, "Erasure-coding based routing for opportunistic networks," in *Proc. of ACM SIGCOMM workshop on Delay-tolerant networking (WDTN)*, pp. 229-236, August, 2005.

[7] S. Jain, M. Demmer, R. Patra, and K. Fall, "Using redundancy to cope with failures in a delay tolerant network," in *Proc. of ACM SIGCOMM*, pp. 109-120, August, 2005.

[8] Deep Space Network, *http://deep-space.jpl.nasa.gov/dsn*

[9] W. Ivancic, W. M. Eddy, L. Wood, D. Stewart, C. Jackson, J. Northam, and A. da Silva Curiel, "Delay/Disruption-tolerant network testing using a LEO Satellite," *8th Annual NASA Earth Science Technology Conference (ESTC 2008)*, University of Maryland, June 2008.

[10] R. C. Hastrup, R. J. Cesarone, J. M. Srinivasan, and D. D. Morabito, "Mars Comm/Nav MicroSat network," *paper presented at the 13th AIAA/USU Conference on Small Satellites,* Aug. 1999. USA.

[11] K. Bhasin and J. L. Hayden, "Space Internet architecture and technologies for NASA enterprises," *International Journal of Satellite Communications,* vol. 20, pp. 311-332, 2002.

[12] NASA/MSFC/ESSD - Coordinated Universal Time, *http://www.ghcc.msfc.nasa.gov/utc.html*

[13] A. Orda and R. Rom, "Shortest-path and minimum-delay algorithms in networks with time-dependent edge-length," *Journal of the ACM*, vol. 37, no. 3, pp. 607-625, July 1990.

[14] Y-B Ko and N. H. Vaidya, "Location-aided routing (LAR) in mobile ad hoc networks," *ACM/Kluwer Wireless Networks,* vol. 6, no. 4, pp. 307-321, 2000.

[15] S. Basagni, I. Chlamtac, R. Syrotiuk, and B. Woodward, "A distance routing effect algorithm for mobility (DREAM)," in *Proc. of IEEE/ACM MOBICOM '98,* Oct. 1998.

[16] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad-hoc wireless networks," *Mobile Computing,* edited by T. Imielinski and H. Korth, Kluwer Adademic Publishers, pp. 153-181, 1996.

[17] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in *Proc. of IEEE Workshop on Mobile Computing Systems and Applications,* pp. 90-100, Feb., 1999.

[18] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, "Directed diffusion for wireless sensor networking," *IEEE/ACM Transactions on Networking,* vol. 11, no. 1, pp. 2-16, Feb. 2003.

[19] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, E. Travis, and H. Weiss, "Interplanetary Internet (IPN): architectural definition," *Internet draft, draft-irtf-ipnrg-arch-00.txt,* May 2001.