

Resource-Considerate Data Routing Through Satellite Networks

Christopher J. Lowe* and Malcolm Macdonald†

University of Strathclyde, Glasgow G1 1XJ, United Kingdom

DOI: 10.2514/1.I010423

In many envisaged satellite-based networks, such as constellations or federations, there often exists a desire to reduce data latency, increase delivered data volume, or simply exploit unused resources. A strategy is presented that achieves efficient routing of data, in a store-carry-forward fashion, through satellite networks that exhibit delay- and disruption-tolerant network characteristics. This network-layer protocol, termed Spae, exploits information about the schedule of future contacts between network nodes, because satellite motion is deterministic, along with the capacity of these contacts to route data in such a way as to avoid significant overcommitment of data along a resource-limited journey. Results from simulations of a federated satellite system indicate consistent benefit in terms of network performance over other, less-sophisticated, conventional methods, and comparable performance to a packet-optimal, full-knowledge approach.

Nomenclature

C	=	network connectivity
C	=	total journey cost
D	=	matrix of maximum communication range between nodes ($\mathbb{R}^{n \times n}$)
E	=	set of edges in G
E_G	=	subset of edges in \mathcal{G}
G	=	graph containing all nodes and edges
\mathcal{G}	=	evolving graph
H	=	number of hops in a journey
J	=	set of feasible journeys
\mathcal{J}	=	journey between nodes
\mathcal{J}^*	=	highest-value journey
M	=	total number of edges in \mathcal{G}
m	=	number of edges in \mathcal{G}
N	=	total number of nodes in \mathcal{G}
n	=	number of nodes in \mathcal{G}
P	=	set of packets
\mathcal{P}	=	path taken in a journey
Q	=	set of nominal resources
Q'	=	set of virtual resources
R	=	matrix of bandwidth (data rate) between nodes ($\mathbb{R}^{n \times n}$)
r_{ij}	=	entry in R (i th row, j th column)
S_G	=	sequence of subgraphs
S_i	=	sequence of times at which subgraphs occur
\mathcal{S}	=	storage cost matrix ($\mathbb{R}^{1 \times n}$)
S_i	=	entry in \mathcal{S} (i th column)
T	=	time horizon for \mathcal{G}
\mathcal{T}	=	transmission cost matrix ($\mathbb{R}^{n \times n}$)
T_{ij}	=	entry in \mathcal{T} (i th row, j th column)
$t_0(\sigma)$	=	start time of a journey
V	=	set of nodes in G
V_G	=	subset of nodes in \mathcal{G}
α	=	time of packet generation
Γ	=	node availability matrix ($\mathbb{R}^{n \times n}$)
γ_{ij}	=	entry in Γ (i th row, j th column)
δ_{ij}	=	entry in D (i th row, j th column)
ρ	=	volume of a packet
σ	=	times of edges in a journey
τ	=	number of subgraphs
φ	=	time to live of a packet
χ	=	journey feasibility (0, 1)

Received 29 September 2015; revision received 4 April 2016; accepted for publication 11 May 2016; published online XX epubMonth XXXX. Copyright © 2016 by Christopher Lowe. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission. Copies of this paper may be made for personal and internal use, on condition that the copier pay the per-copy fee to the Copyright Clearance Center (CCC). All requests for copying and permission to reprint should be submitted to CCC at www.copyright.com; employ the ISSN 2327-3097 (online) to initiate your request.

*Research Associate, Advanced Space Concepts Laboratory, 75 Montrose Street; christopher.lowe@strath.ac.uk.

†Reader, Advanced Space Concepts Laboratory, 75 Montrose Street; malcolm.macdonald.102@strath.ac.uk. Associate Fellow AIAA.

I. Introduction

RESearch into the performance of data-routing protocols through delay- and disruption-tolerant networks (DTNs) has shown that effective approaches can offer significant value where objectives are some function of data latency (delay), hop count, and/or delivery ratio (proportion of generated data that are finally delivered) [1]. An intermittently connected satellite network, such as that exhibited by federated satellite systems (FSSs) [2], can be categorized as such, where reliable, expeditious delivery of data from source to destination is a driving force behind research in the field. This work introduces a novel routing strategy, called Spae,[‡] which exploits the deterministic contact schedule of the network and expected downstream resource availability, in the form of communication bandwidth, energy, and buffer storage, in an FSS to efficiently route data and maximize mission-level objectives.

II. Related Work

Routing of data through delay- and disruption-tolerant networks (DTNs) with deterministic contact schedules has seen significant attention in recent years, with the introduction of the evolving graph in [3] and a formal definition of DTNs offered in [1]. Since then, there has been much development on network-layer protocols that aim to improve performance in terms of data-delivery attributes, which, in the case of fixed-schedule dynamic networks (FSDNs), often exploit knowledge of the future contact schedule in support of routing decisions. A number of authors have applied modified versions of traditional network shortest-path algorithms for this purpose, typically Dijkstra’s algorithm [4–6], but also breadth-first search [7,8] has been employed to good effect. The journey through an FSDN (i.e., the path taken by a packet/message/bundle) is often classified in terms of value by the arrival time at the destination [4], the number of hops taken to reach the destination [5], the expected communication-based quality of service [9], or a combined generic cost metric based on the end-to-end journey time and number of internode transfers [10]. This metric is used to identify the most attractive next hop node along the journey toward the destination, to which packets are typically forwarded or copied. In the case of deterministic data traffic, whereby the arrival of packets can be predicted in advance, linear programming has been shown to be an effective method of optimizing the routing plan [11,12]. However, it is generally considered unrealistic for this level of knowledge to exist.

Space-based networks are a special case of FSDN in which the orbital motion of nodes can be predicted accurately for long periods into the future, therefore providing stable, deterministic contact event information. This has led to a wealth of interest in the development of routing protocols for space networks, initially for deep-space applications [13,14] and at the transport layer [15,16]. Minimal delay routing is investigated in [17], whereby messages are delivered to the user via a network of satellites and ground station gateways but where no intersatellite or intergateway communication is considered. In [18], application of the store-and-forward better approach to mobile ad hoc networking algorithm [19] is used for routing in an FSS, which uses a DTN version of link-state routing (LSR) [20]. Although LSR was originally developed for mobile ad hoc networks, it offers the ability to update network nodes on potential disruptions and/or node unavailability through message flooding. A shortcoming of LSR, however, is the lack of consideration for downstream resources beyond the first hop, which makes rerouting of data at a downstream node likely in regions of high load. Perhaps one of the most established methods of routing in space networks, originally designed for deep-space networks, is contact graph routing (CGR) [21], which in its original form follows much of the literature in using Dijkstra’s algorithm to identify the next hop node along a journey. Extensions include the consideration of an earliest arrival time objective [22], multiple destinations [23], timeliness effects of potential data-delivery failure due to lossy signal [24], effects on delay from downstream buffer queues [25,26], and an approach to overcome the effects of high-priority data preventing transmission of bulk packets during a scheduled contact [27,28]. Although CGR has strengths in terms of development and deployment, the journey cost functions, mission objectives, and resource considerations could be considered lacking in generality.

A general strategy, in the sense of its independence to network topology, data type, and mobility pattern, is presented here, termed Spae, which is a packet-forwarding algorithm that routes data on a packet-by-packet basis through a DTN. Forwarding decisions are made on a hop-by-hop basis each time contact is made with another node in the network, ensuring robustness to stochastic network changes. Knowledge, or at least estimation, of future resources (e.g., energy, buffer, and communication bandwidth) is necessary for effective operation. Upon making contact, two nodes share information relating to their buffer contents via some handshake interaction, and packets are routed[§] one-by-one, highest priority first, until all packets have been evaluated. The journey along which a packet is routed is selected by Spae as the one with highest expected value, measured as a function of network-level objectives and expected downstream link availability. The virtual resources of edges and nodes along the highest-value journey are then reduced by an amount equal to that demanded by traversal of the packet, such that an apparent resource is suitably lower when routing subsequent packets, during the contact event. Edges along which resources are sufficiently low to prevent traversal of a packet are invisible when routing remaining packets, thus avoiding delivery along journeys with little or no hope of success. In addition, feasible journeys are considered only those that reach the destination before expiration of a packet. Finally, packets are physically transferred to the neighboring (local) node if, and only if, it is the next hop along the selected journey. Spae provides, for the first time, a method of routing packets through DTNs in which resources of generic type are considered from source to destination. Other methods have lacked generality in terms of the resources being considered, end-to-end completeness in the extent to which these considerations are made, or comprehensiveness in the dedicated assessment of each packet in the buffer.

For clarity, some of the terminology used within the following sections is introduced next. In terms of network performance, three metrics are considered. Delivery delay is the duration between a packet entering the network and being delivered to a destination. Delivery ratio is the ratio of the total number of delivered packets to the total number of generated packets, and hop count is the number of nodes on which a packet is stored before reaching the destination. With respect to data traffic, the term “utilization” is applied, which denotes the ratio of the total number of generated packets to the total number of packets that could theoretically be downloaded. For example, a utilization of 0.5 would indicate the generation of, say, 100 Mb of data, given that the total download capacity of contacts between nodes and the destination/s is 200 Mb. Time to live (TTL) is the duration between the moment a packet is generated and the moment it would expire. Following expiration, a packet would be dropped from the network. Finally, network connectivity C is used as described in [4] as a measure of internode communication range representing the fraction of total time that a node pair is in contact, averaged over all node pairs in the network (e.g., $C_{ij} = 0.1$ would imply that node i is in view of node j , 10% of the time, and a value of $C = 1$ implies that all nodes are connected at all times).

[‡]Scottish word meaning “to predict/foretell”.

[§]The term “routed” shall be used throughout to represent the virtual assignment of a packet over a particular journey. The packet may or may not traverse the particular journey in reality; however, it enables identification of the ideal next hop.

III. Network Model

In the following sections, the network model is formally described to provide an explicit definition of Spae routing. Spae offers a scale- and time-independent network routing strategy, applicable to any fixed-schedule dynamic network (FSDN) in which information pertaining to expected start time and duration of node-pair contacts is available.

A. Evolving Graph Configuration

A graph $G(V, E)$ is made up of a set of nodes $V = \{v_1, v_2, \dots, v_n\}$ and edges $E = \{e_1, e_2, \dots, e_m\}$ [¶] and represents the union of an ordered sequence of subgraphs $S_G = G_1, G_2, \dots, G_\tau$, which occur at times specified by the sequence $S_t = t_1, t_2, \dots, t_\tau$, respectively, such that $G = \bigcup_{i=1}^\tau G_i$. The evolving graph $\mathcal{G} = (G, S_G, S_t)$ is used to define the dynamic network that exists over the time horizon $T = [t_1, t_\tau]$. Formation of a new subgraph occurs at each change to the set of edges $E_i(G_i)$, i.e., whenever one or more edges are formed or broken, $E_{i+1} \neq E_i, \forall E_i \in E$. For clarity, graph G_i is in place during the interval $[t_i, t_{i+1}]$. An edge $e(u, v) \in E_i(G_i)$ is considered directional and uniquely characterized by its sending node u and receiving node v . Therefore, it follows that identical edges present across one or more adjacent graphs $e(u, v) \in \{E_i, E_{i+1}, \dots, E_{i+k}\}$ represent edges that remain in place while k other edge formations or removals occur elsewhere in the network. The cause of edge formation is arbitrary as long as a future contact schedule is known (e.g., edge formation might result from node proximity in a mobile network or intermittent operation of nodes in a static wireless sensor network).

For each edge e in the evolving graph \mathcal{G} , there exists a nominal capacity $q(e)$, which is the total volume of data able to be sent over e in ideal conditions. The set Q_i represents the capacity of all edges at time i , $|Q_i| = |E_i|, \forall i \in [1, \tau]$. The capacity of the edge between sending node u and receiving node v , from time t_i to t_{i+1} , is

$$q(t_i) = \int_{t_i}^{t_{i+1}} r_{uv}(t) dt, \quad \forall i \in [1, \tau] \quad (1)$$

where $r_{uv}(t)$ is the data rate (bandwidth) at time t from node u to node v . For clarity, we define $\mathbf{R}(t) = \mathbb{R}^{n \times n}$ as the matrix representing the data rate for each node pair at time t , where r_{ij} is the ij th element in \mathbf{R} .

B. Data Traffic

Generally, data are defined as a packet $p(\rho, \alpha, \varphi, v_d)$ with a unique volume ρ , time of generation (birth date) $\alpha \in T$, time to live (TTL) φ (i.e., time since α until expiration), and destination node $v_d \in V$. The objective of almost any routing algorithm is to deliver packets to their destination in such a way that best satisfies some network objective, e.g., minimum end-to-end delay (latency), minimum hop count, or maximum volume delivered. In some networks, packets may be required to pass through specific nodes in the network en route to the destination for security or other reasons. Although it would be possible to include this in Spae routing by replacing the modified version of Dijkstra's algorithm [29] that finds the optimal journey through the evolving graph with an algorithm that found the optimal journey in a Steiner tree problem [23], it is considered beyond the scope of this work.

In most terrestrial systems, the arrival rate of data to the network is stochastic but often can be defined by some probability distribution, which may indeed vary with time or node location. In the case of deterministic data generation,** it is argued that an alternative, bespoke routing algorithm could be developed that exploits this information to maximize the network objective.

C. Journeys

A journey \mathcal{J} through the network, between two nodes (u, v) , is defined by a start time t_0 , a path \mathcal{P} , made up of an ordered sequence of $H + 1$ nodes, $\mathcal{P} = i_0, i_1, \dots, i_H$, where $i_h \in V, \forall h = 1, 2, \dots, H$, and an ordered sequence of times $\sigma = t_1, t_2, \dots, t_H$, where $t_h \in S_t, \forall h = 1, 2, \dots, H$ at which the edges between nodes exist. The total number of hops (transfer of custody between nodes) is $H, i_0 \in \mathcal{P}(\mathcal{J})$ is the source node, and $i_H \in \mathcal{P}(\mathcal{J})$ is the final node. It is clear that $t_h \geq t_{h-1}$, and the receiving node in hop h must be the sending node in hop $h + 1$. Finally, it follows that t_H represents the start time of the contact with the destination.

The cost of a journey is generalized as the sum of the storage cost \mathcal{S}_i onboard each node i and the transmission cost \mathcal{T}_{ij} between each node pair ij , along a journey \mathcal{J} . The total journey cost is therefore

$$\mathcal{C}(\mathcal{J}) = \sum_{h=1}^H \int_{t_{h-1}}^{t_h} \mathcal{S}_{i_{h-1}}(t) dt + \sum_{h=1}^H \mathcal{T}_{i_{h-1}i_h}(t_h) \quad (2)$$

where H is the total number of hops, and $t_h \in S_t$ is the time at which hop h occurs. It is clear that a transmission cost of $\mathcal{T}_{ij} = 0$ for all node pairs would imply that a journey with minimal duration would be the one of least cost.

The availability γ of a node is taken to be the probability that the node will be available for data sharing at a scheduled contact. This may be either constant or time-dependent but must be deterministic/predictable for implementation into Spae. This is considered a reasonable assumption, whereby the availability of a satellite in a FSS could be either published by the operator or calculated as some moving average from historical availability data. The total availability of a journey is therefore formulated as

$$\gamma(\mathcal{J}) = \prod_{h=1}^H \gamma_{i_h}(t_h) \quad (3)$$

where $\gamma_{i_h}(t_h)$ is the availability of the h th hop receiving node, at the time of that hop. It is clear that one or more nodes with a low availability will adversely affect the value of the journey. For completeness, the set $\Gamma(t) = \{\gamma_1, \gamma_2, \dots, \gamma_n\}$ represents the availability of each node at time t , where

[¶]Often, the set of edges E is captured in the form of an adjacency matrix of size $\mathbb{R}^{n \times n}$, whereby a nonzero entry in the i th row and j th column, $e_{ij} \neq 0$, represents a directed link from node i to node j . This is a plausible approach here and would naturally follow through to the evolving graph instances $E_i \forall i \in [1, \tau]$.

**The deterministic case shall be used as an upper bound on performance for comparative purposes in this work, achieved by applying a traffic oracle that knows the priority and location of all traffic in the network.

$|\Gamma| = |V|$. The value $\mathcal{V}(\mathcal{J})$ of a journey is considered to be the inverse of the risk-adjusted cost $\mathcal{V} = \gamma/C$, such that a low-cost journey with low risk (high availability) is of high value. Naturally, in a completely reliable network, the value reduces to the inverse of the cost. The highest-value journey \mathcal{J}^* from the set of feasible journeys $J = \{\mathcal{J}_1, \mathcal{J}_2, \dots, \mathcal{J}_Y\}$, is

$$\mathcal{J}^* = \mathcal{J}_i | \mathcal{V}(\mathcal{J}_i) = \max_{\mathcal{J} \in J} \mathcal{V}(\mathcal{J}) \quad (4)$$

IV. Spae Routing

At the start of a contact between nodes u and v , their respective buffers awaiting transmission $B_u = \{p_{u1}, p_{u2}, \dots, p_{uI}\}$, $I = |B_u|$ and $B_v = \{p_{v1}, p_{v2}, \dots, p_{vK}\}$, $K = |B_v|$ are shared over a handshake interaction during which all packets onboard both nodes are placed into virtual combined buffer (VCB), with $B_{uv} = \{p_{uv1}, p_{uv2}, \dots, p_{uv(I+K)}\}$. The VCB will be replicated on both interacting nodes, such that, during the routing process, each node is aware of the packets being routed by their neighbor. Packets are routed, one-by-one, typically highest priority first, but not necessarily such that other buffer ordering schemes could be employed, until the VCB is empty. The journey identified during this routing process (as described in Sec. III.C) is used to identify the next hop for a packet, from its current custodian, and thus whether custody should be transferred during the current contact (i.e., if the next hop is to the current neighbor during the present contact, then custody is transferred). Before routing of the first packet in the VCB, a virtual resource set Q'_i is created, which is a duplicate of the nominal set Q_i , for each future graph that could potentially be part of the journey for a packet within B_{uv} . This resource set is generic for Spae, such that it could include spare bandwidth, node buffer storage, energy, or other components, and can be described formally as

$$Q'_i = Q_i, \quad i = [k, k + y(p_{uvX})] \quad (5)$$

where k is the graph number in S_G at the current time t_k , and $y(p_{uvX})$ is the number of graphs until expiry of the packet with the longest remaining time before expiry (p_{uvX}). It follows therefore that $t_{(k+y(p_{uvX}))} \leq t_k + \varphi(p_{uvX}) - \alpha(p_{uvX})$ and $t_{(k+y(p_{uvX})+1)} > t_k + \varphi(p_{uvX}) - \alpha(p_{uvX})$. Naturally, the complete set can be defined as $Q' = \{Q'_k, Q'_{k+1}, \dots, Q'_{k+\max(y(p_{uvX}))}\}$. How the available resources are established is independent from how Spae operates, such that they could arrive from an oracle layer that is able to transmit information to all nodes at all times or from onboard estimations considering previous contacts with others in the network. Here, it is assumed that nominal resources (i.e., those attributed to all nodes and edges in \mathcal{G} before consideration of traffic flow) are known by each node. For example, the effect of traffic arriving at a downstream node from elsewhere in the network is ignored such that a particularly busy edge would likely become oversubscribed. Clearly, this might result in situations whereby intended journeys are not achievable by certain packets, which would not happen in a full knowledge scenario, but a tradeoff between estimate complexity and risk of route modification must be made. Indeed, a more-accurate estimation of the expected available resource would offer higher accuracy, at greater computational expense and data overhead. In the method proposed here, the only system traffic required to be shared with the neighbor is a list of packets in the buffer awaiting transmission, including their priority, size (in bits), and intended destination. This is not expected to be significant relative to the payload data and is therefore neglected in this analysis. Furthermore, because the focus of this work is investigation of routing algorithm performance, the physical processes by which nodes connect (physical layer) and the associated protocol over which data are sent (data-link layer) are ignored. This additional level of detail shall form part of future work, which will include the time and overhead for initial contact and handshake interaction as well as tracking of links considering relative velocities and attitudes between node pairs. Indeed, depending on the physical link medium, some contacts are likely to be infeasible due to significant Doppler effects, which will also be considered in the future.

Following routing of packet p along a journey \mathcal{J} , the entries in Q' that correspond to the edges and nodes in \mathcal{J} are reduced by the magnitude required to transfer custody of packet p during that graph, which in the case of edge capacity (bandwidth) is

$$q'_{h-1,h,k} = q'_{h-1,h,k} - \rho_p, \quad k = t_h(\sigma), \quad \forall h = 1, 2, \dots, H \quad (6)$$

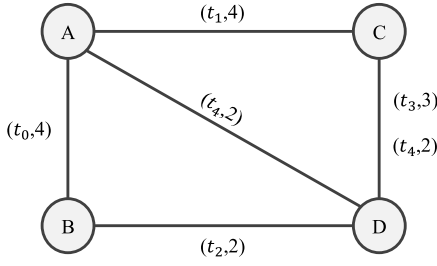
where H is again the number of hops in \mathcal{J} , and k is the subgraph number in G in which the hop takes place. Noteworthy is that, although it would be possible to include other resources such as energy and buffer storage, only edge bandwidth is considered here to illustrate the concept. In the future, both energy and buffer storage shall be incorporated for completeness. Regarding the former, the nominal energy resource would diminish according to the amount required for routing of the packet in question, but exactly how to measure the nominal amount available may require use of heuristics. In the case of a satellite, for example, onboard energy might cycle between a maximum and minimum each orbit, such that a specific amount would be available for data-routing operations per orbit. Should the routing process demand energy beyond this amount, it is expected that some increased cost will be imposed such that the journey value is artificially lowered. Regarding buffer resource, this will be seen to fluctuate as packets are transmitted to and from a node, such that limits will be imposed on how much data can flow into a node, before some are sent onward further along their journey.

Journey evaluation is achieved by searching a subset of the evolving graph using a modified version of Dijkstra's algorithm, adapted for use on time-varying networks. The subset for packet p is defined as the set of graphs that occur before expiration of p , i.e., $[G_k, G_{k+y(p)}]$, following notation used in Eq. (5). A journey may only be selected with the modified Dijkstra's algorithm if it is considered feasible, with respect to packet p . A feasible journey is considered one over which a sufficient level of expected resource exists and which can be completed before expiration of the packet. Therefore, the feasibility χ of journey \mathcal{J} , considering bandwidth as the resource that is potentially limited, can be defined as

$$\chi(\mathcal{J}) = \begin{cases} 1 & |q'_{h-1,h,k} \geq \rho_p \wedge t_H - \alpha \leq \varphi, k = t_h(\sigma), \forall h = 1, 2, \dots, H \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

After either routing all data in the VCB or exploiting all available capacity in both directions of the current contact event, each packet should be transferred to the neighboring node, retained onboard the current custodian, or dropped due to a lack of feasible journeys. At this point, the routing process for this contact is considered complete.

Regarding application and manipulation of the evolving graph described in Sec. III.A, it is assumed that, in the FSS scenario, a satellite has single-point transmission (send) and multipoint reception (receive) capabilities, exhibiting full-duplex operation when a two-way link exists. This is considered a realistic configuration assuming multiple antennas capable of electronic beamforming, while being considerate of energy usage, but other considerations will be investigated in future work.



$$B_A = \{p_{A1(1,5)}, p_{A2(2,2)}, p_{A3(1,1)}\} \quad B_B = \{p_{B1(3,4)}, p_{B2(1,3)}\}$$

Packet	Resource required	Path (\mathcal{P})	Times (σ)	Edge resource before	Edge resource after	Journey cost (C)
p_{A1}	1	A-B-D	t_0, t_2	4, 2	3, 1	3
p_{B1}	3	B-A-C-D	t_0, t_1, t_3	3, 4, 3	0, 1, 0	5
p_{B2}	1	B-D	t_2	1	0	2
p_{A2}	2	A-D	t_4	2	0	4
p_{A3}	1	A-C-D	t_1, t_4	1, 2	0, 1	5

Fig. 1 Schematic example of Spae routing.

An example of Spae routing is illustrated in Fig. 1. Here, three packets from node A 's buffer and two packets from node B 's buffer are routed to destination D along journeys described in the table on the right of the figure. An evolving graph is shown on the left, where the terms in parentheses (t, n) alongside each edge indicate the time t and expected available resources n . For simplification, all nodes are considered available for transfer, the cost of storage is 1 per time step, and the cost of transmission between two nodes is 1 everywhere, except for transfer to the destination D , which is assumed 0. Each packet ($p_{Ni(j,k)}$) on node N is described as having a resource demand j and priority k within the combined buffer.

Even in this simple case, the routing approach found by Spae is not trivial, with preferred routes exhibiting insufficient resources in a number of cases. For example, the relatively large resource demand from packet p_{B1} requires it to traverse three nodes before reaching the destination, with lower-cost journeys being unavailable due to resource limitations. The result of this routing event would be a transfer of p_{A1} from A to B and p_{B1} from B to A , whereas all other packets remain on their current custodian for transfer at future contacts.

A. Pseudocode

The following pseudocode describes the Spae routing procedure as implemented in software, which is carried out during each node-pair contact event.

Algorithm 1 Pseudocode for Spae routing

Spae — procedure at each node-node contact event
Input: $\mathcal{G}, u, w, \mathbf{R}, B_u, B_w, S, T, \Gamma$
Output: Custodian of each packet in combined buffer post-routing

$$B_{uw} = \bigcup \{B_u, B_w\}$$

$$Q' = Q$$

for $k = 1$ to $|B_{uw}|$ **do**
 $\nu = v(p_k) \in \{u, w\}$
 Find $\mathcal{J}^* = \mathcal{J}_i | \mathcal{V}(\mathcal{J}_i) = \max_{\mathcal{J} \in \mathcal{J}: \chi(\mathcal{J})=1} \mathcal{V}(\mathcal{J})$ using modified Dijkstra($\mathcal{G}, u, \mathbf{R}, S, T, \Gamma$)
if $\chi(\mathcal{J}^*) = 1$
 for $h = 1$ to H , where $H = |\sigma|$ and $\sigma \in \mathcal{J}^*$ **do**
 $q'_h = q'_h - \rho_{p_k}; q' \in Q'$
 end
 if $i_1 = \{u, w\} \neq \nu \wedge t_1(\sigma) = t_0(\sigma)$, where $i_1 = \mathcal{P}^*(1), \mathcal{P}^* \in \mathcal{J}$, **do**
 $v(p_k) = i_1$
 end
else
 drop p_k from buffer
end
end

The pseudocode is also described informally by the flow diagram in Fig. 2.

B. Complexity

During each contact event, Spae can be seen to have a worst-case complexity equivalent^{††} to running a modified Dijkstra's algorithm, for multiple graphs extending over the period of a packets remaining TTL, for each packet onboard the node pair's combined buffer. In big-O notation, this is $O(|B|(|E| + |V| \log |V|))$, where $|B|$ is the number of packets in the combined buffer, $|E|$ is the number of downstream edges (i.e., the number of edges in all future graphs), and $|V|$ is the number of downstream nodes (i.e., the number of nodes in all future graphs). In reality, the complexity is generally not this large because computation can be reduced given that 1) once the resources of each graph (in both directions) of the current contact have diminished below a level required for routing of a packet, routing is terminated because no additional transfer can take place during that contact, and 2) a packet with the same destination node as some previously routed packet can be routed along the same journey, if sufficient resources are expected to remain along that journey. To achieve this, a feasible journey table is maintained, which can be checked at the start of routing each packet.

C. Performance Attributes

Performance of a routing protocol for DTNs is typically measured as some function of delivery delay, hop count, and delivery ratio. Delay is often critical for information such as Earth observation data, machine-to-machine messaging, and satellite-based automatic identification system data. Hop count is often considered a good measure of energy usage because transmission of data requires power from the satellite bus, but in an

^{††}Although execution of the modified Dijkstra's algorithm only represents the "search feasible journeys" and "identify highest-value journey" steps in Fig. 2, the other steps consist of trivial operations that have either constant $[O(1)]$ or linear time complexity with respect to buffer size $[O(B)]$. As such, their complexities can be neglected when considering the overall algorithm complexity.

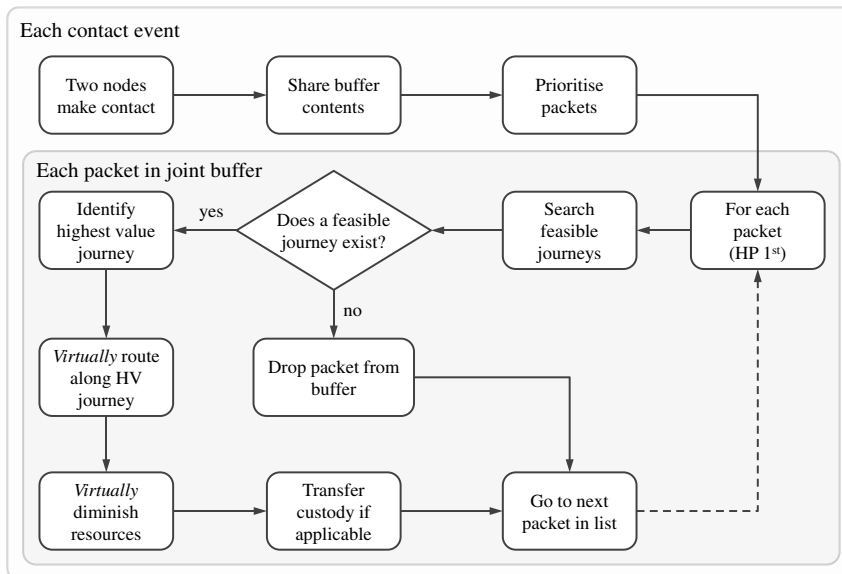


Fig. 2 Spae routing process.

FSS, hop count is also a suitable measure of the potential financial implications of using a relay satellite within the federation. Delivery ratio is effectively a measure of payload operation efficiency, where a low result would indicate significant wasted resource. Indeed, for important and critical data, a high delivery ratio could be considered necessary.

V. Simulation

A simulation environment has been developed^{††} that provides an effective means of evaluating the performance of Spae compared with other routing strategies. The environment provides the necessary physics-based capability for orbit propagation of each satellite and ground station, resulting in the relative positions that are exploited for building the evolving graph. The arrival of data is modeled as a stochastic process following a geometric distribution for interpacket arrival time, which has been shown to accurately represent the arrival of satellite image requests from customers [30] or indeed any process in which the memory-less property is exhibited. It is considered likely that this distribution would accurately model other processes, such as the delivery of machine-to-machine messages; however, worth noting is that Spae operates independently of the data arrival process, such that other distributions could be implemented without issue. The probability of packet generation is defined as a function of network utilization and total network capacity of contacts with ground stations. This approach ensures a comparative level of data in the network, given different network attributes, allowing for effective comparison. In all simulations, 10 satellites form the federation and an availability parameter γ_i of between 0.5 and 1 is selected, which aims to emulate the situations whereby operational constraints may render an agent unavailable for federation involvement without prior warning. Other simulation parameters are outlined in Table 1, where the values of orbit altitude, inclination, and satellite availability are selected at random from the range indicated, for each satellite, during each simulation.

Performance attributes of delay, hop count, and delivery ratio are considered throughout, each shown with respect to varying data TTL, intersatellite link (ISL) range, and ISL bandwidth. Here, Spae routing is compared against four other protocols, which are described in the following in order of sophistication. The first approach, Direct, represents a nonnetworked FSS in which satellites have no ISL capability and thus only consider direct transfer to the ground stations (i.e., all satellites operating in isolation). This is considered to represent a lower bound on performance in terms of delay and delivery ratio but will of course have a hop count of 1 always. The second, Next2see, is a strategy whereby packets are transferred to a local node if it is to make contact with a ground station before the current custodian. The third protocol, NextHop, exploits Dijkstra's algorithm to identify the highest-value journey, along which packets are transferred should the local node represent the next hop. It differs from Spae in that downstream resources are not considered such that downstream congestion is more likely. Finally, in FullKnow, an upper bound on performance is evaluated, where full knowledge of all current and future network properties exist, including traffic generation, future node availability, and resources. As discussed earlier, this approach is considered unattainable in most real-world scenarios. In all of the protocols described previously, limitations on the number of packets transferred over an edge due to finite bandwidth are obeyed, but no limitations on buffer or energy are considered. The properties of each strategy are summarized in Table 2.

VI. Results

Results of simulations carried out using the parameters described in the previous chapter are shown in the following sections. First, the effects of different ISL range and bandwidth are investigated, given no data expiry (TTL = ∞). Next, the effect of the same variables plus that of different TTL values are shown, whereby packet loss is evident due to expiry.

In the graphs showing results of delay and hop count, the data represent the average values from all packets delivered by the network. Furthermore, all plotted data represent the mean average across all 50 Monte Carlo simulations. Finally, standard deviation is seen to be smaller, or comparable, with increasing protocol complexity such that plotting of this metric has been omitted.

A. Infinite Packet Time to Live

In satellite systems of the future, where buffer storage is likely to be cheap and plentiful, given the recent and expected advancements in solid-state storage, dropping of packets due to a full buffer could be considered unlikely. Therefore, to ensure complete data delivery, a high TTL is

^{††}The simulation environment is implemented in MATLAB, which offers a flexible software environment capable of successfully executing network-based analysis. It is recognized that other software environments exist, such as ns3 and OMNeT++, the suitability of which shall be investigated during future work.

Table 1 Simulation parameters

Parameter	Value	Comment
Orbit altitude	400–800 km	Uniform distribution
Inclination	0–100 deg	Uniform distribution
Eccentricity	0	Circular orbit
Ground stations	Kourou, French Guiana; Villafranca, Spain; Perth, Australia	Minimum elevation = 10 deg
Bandwidth (to ground)	150 Mb/s	Fixed rate
Satellite availability γ	50–100%	Uniform distribution
Utilization	0.5	Ratio of upload to (possible) download data volume
Simulation duration	Three days	10 s time steps
Number of simulations	50	Monte Carlo parameter
Storage cost S	1 per second	Arbitrary units
Transmission cost T	1800 per hop	Equivalent to 30 min of storage

Table 2 Routing strategy comparison

Name	Description	Knowledge			
		ISL	Mobility	Resource	Traffic
Direct	No intersatellite data transfer; packets sent direct to destination	×	×	×	×
Next2see	Data are transferred to the local node next to be in contact with a ground station	✓	×	×	×
NextHop	Data are transferred to the local node next along the highest-value journey identified using Dijkstra	✓	✓	×	×
Spae	As described previously	✓	✓	✓	×
FullKnow	Knowledge of all network attributes allows source routing to destination with guaranteed delivery	✓	✓	✓	✓

required to avoid dropping due to expiry. The following two sections show simulation results with infinite packet TTL, in which delivery ratio is 1 throughout (i.e., all data that were generated were delivered).

1. *Effect of Intersatellite Link Range*

Figure 3 shows the effect of an intersatellite link range between 500 and 2500 km, with a constant ISL bandwidth of 150 Mb/s, on both delay and hop count. Spae can be seen to offer comparable delay performance to the full knowledge upper bound, whereas NextHop follows the same trend but at an almost constant additional delay from the two more sophisticated protocols. The delay effectiveness of Spae can be seen to diminish relative to FullKnow, at higher ISL range (higher connectivity), which can be attributed to the greater number of journey options and hence greater likelihood of a congested path being taken. The hop count for the three best performing strategies is comparable, whereas that of Next2see suffers from inefficient routing along suboptimal journeys.

As would be expected, an increase in ISL range has value in terms of delay because contacts that were perhaps not available with shorter range are realized. This is effectively a measure of the network connectivity, which represents the percentage of time node pairs that are in contact, averaged over all node pairs in the network. Statistics of network connectivity are shown in Fig. 4, with respect to ISL range. Noteworthy is that

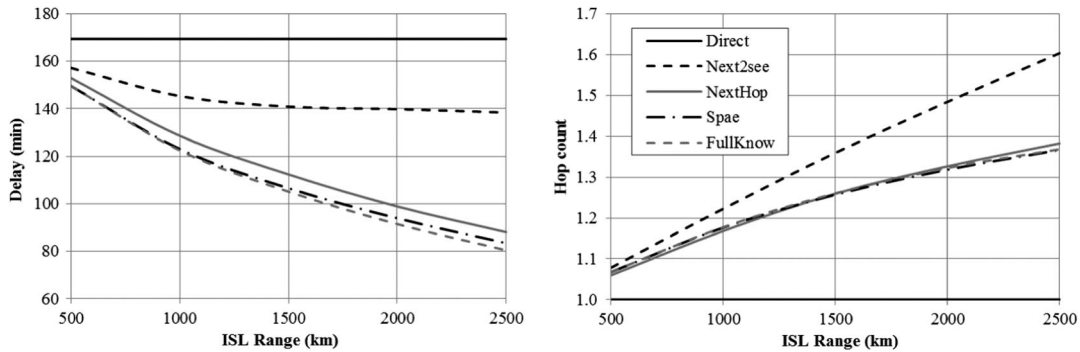


Fig. 3 Effect of ISL range on delay (left) and hop count (right).

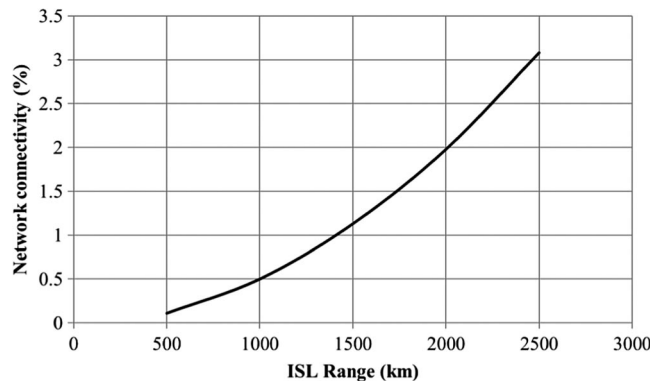


Fig. 4 Network connectivity at various ISL range.

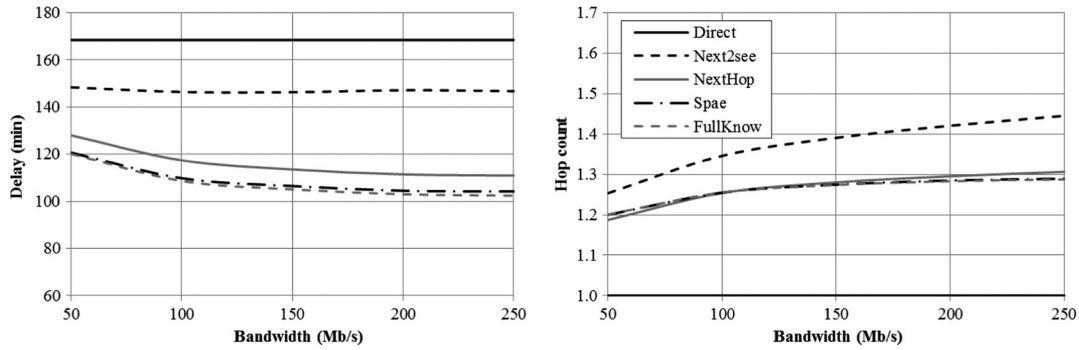


Fig. 5 Effect of ISL bandwidth on delay (left) and hop count (right).

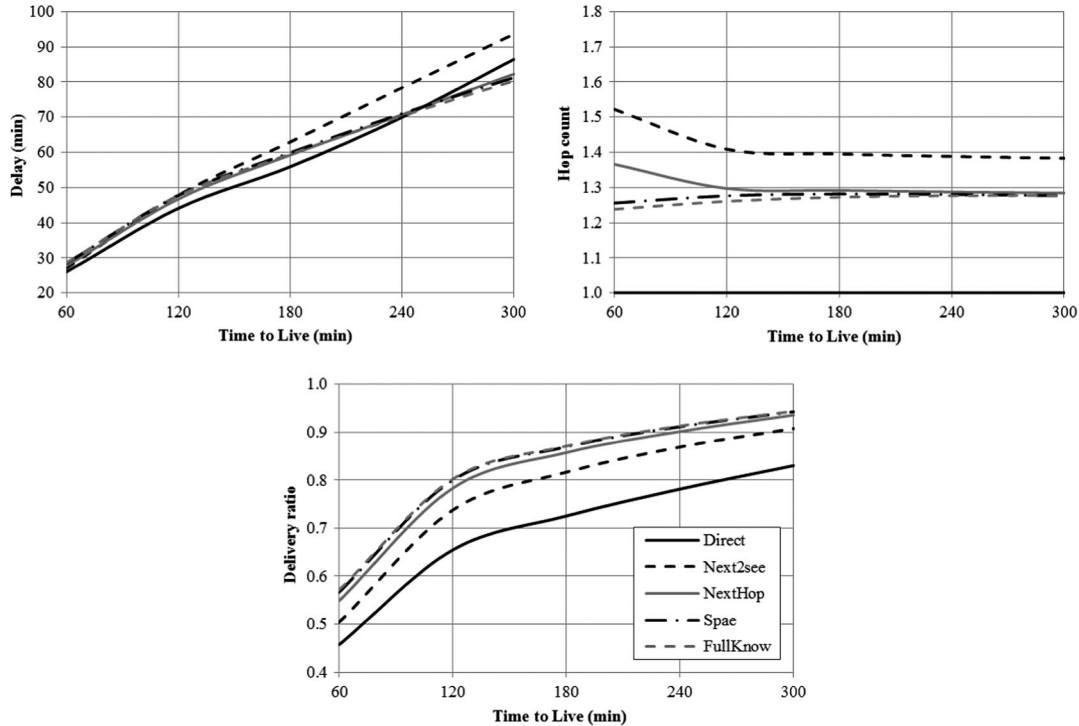


Fig. 6 Effect of time to live on delay (top left), hop count (top right), and delivery ratio (bottom).

calculation of connectivity here considers all connections between satellite pairs (i.e., an effective node availability of 1 everywhere) and no connections to ground stations.

2. Effect of Intersatellite Link Bandwidth

Results for various ISL bandwidth (BW) are shown in Fig. 5, with a constant ISL range of 1500 km^{§§} [31]. It is clear that a bandwidth above ~100 Mb/s is of limited value in this scenario, which suggests that the data intended for transfer between satellites are achievable at this rate, and thus the benefit of additional BW could be considered limited. The hop count trend is similar to that seen with varying levels of ISL range (Fig. 3), whereby the three most sophisticated algorithms are comparable.

B. Finite Time to Live

In the following analysis, a finite TTL results in the situation whereby a packet would be dropped from a node's buffer at the point of expiry or, in the case of NextHop, Spae, and FullKnow, at the point where no feasible journey is found within the remaining TTL.

1. Effect of Finite Time to Live

Given an ISL range of 1500 km and bandwidth of 150 Mb/s, performance at different TTL values is shown (Fig. 6). As would be expected, the average delay increases with TTL, which itself represents the maximum possible delay because the packet would be dropped should delay exceed this value. Although the delay performance from the more sophisticated algorithms appears generally worse than those of less complexity, it becomes clear that this is a result of a greater delivery ratio and, thus, greater volume of data delivery. Indeed, Spae again offers comparable results to the packet optimal in delivery ratio, with NextHop exhibiting a consistently lower delivery volume. This can be attributed to the greater frequency of situations whereby packets are routed along overly congested paths, which then need rerouting, and therefore expire before reaching the destination. The higher hop count seen in Next2See and NextHop at low TTL verifies this hypothesis.

^{§§}Considering current technologies and expected advancements in RF (and optical) communication systems, this range is representative of intersatellite links in a federated satellite system.

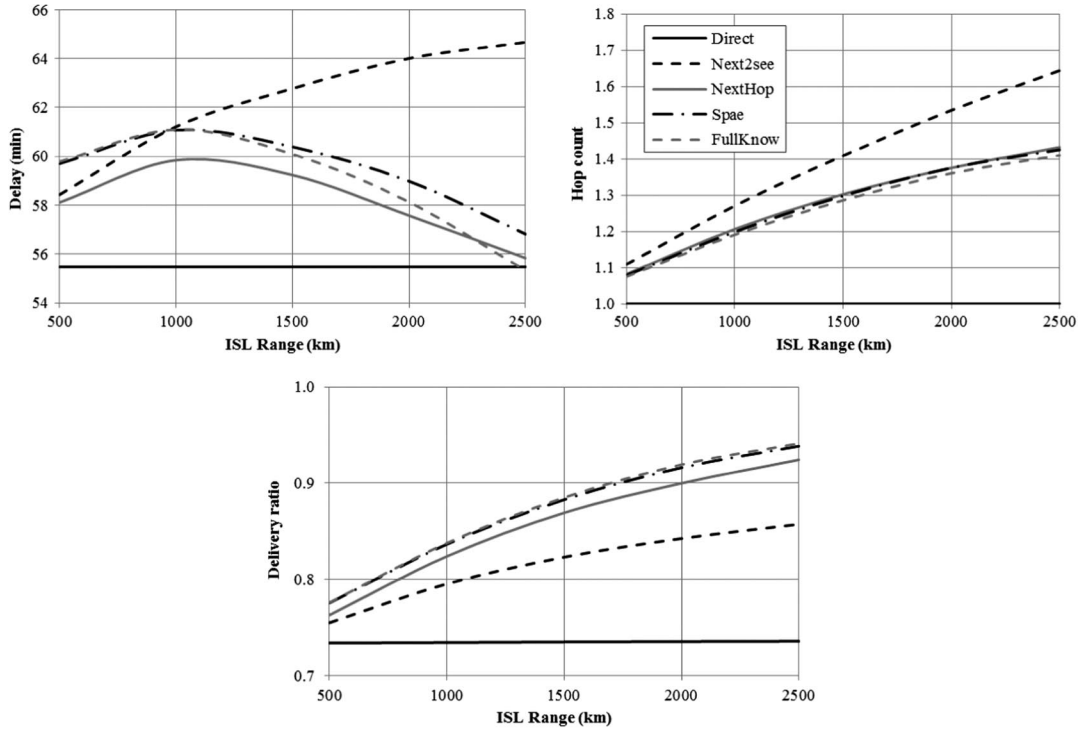


Fig. 7 Effect of ISL range on delay (top left), hop count (top right), and delivery ratio (bottom).

2. Effect of Intersatellite Link Range

The effect of ISL range is shown in Fig. 7 with an ISL bandwidth of 150 Mb/s and TTL of 180 min. Noteworthy is the fact that the axis scale in the plot showing delivery delay (top left) is significantly smaller than that of the plot in the previous section (Fig. 6, top left). ISL range appears to have a relatively small effect on delay in general; however, it is clear that a trend toward lower delay is emerging at greater levels of connectivity for all protocols except Next2see. This exception is due to Next2see’s lack of awareness of lower delay journeys becoming available at higher ISL range, a property shared by the other protocols. The result is inefficient routing of packets toward nodes that subsequently identify other, more promising nodes at a later stage, which also explains the consistently higher hop count. Interesting is the turning point in the ISL range versus delay plot, for NextHop, Spae, and FullKnow. This is due to bandwidth over contact events up to this point being generally saturated with data, which offers little scope for routing packets effectively around congestion. Beyond this point, however, additional connections appear between nodes that were previously out of view, such that less often are the edges saturated, allowing the most favorable journey to be taken by more packets, resulting in a delay reduction. For hop count, an increase in ISL range results in an increase in overall performance, but at a decreasing rate. This is

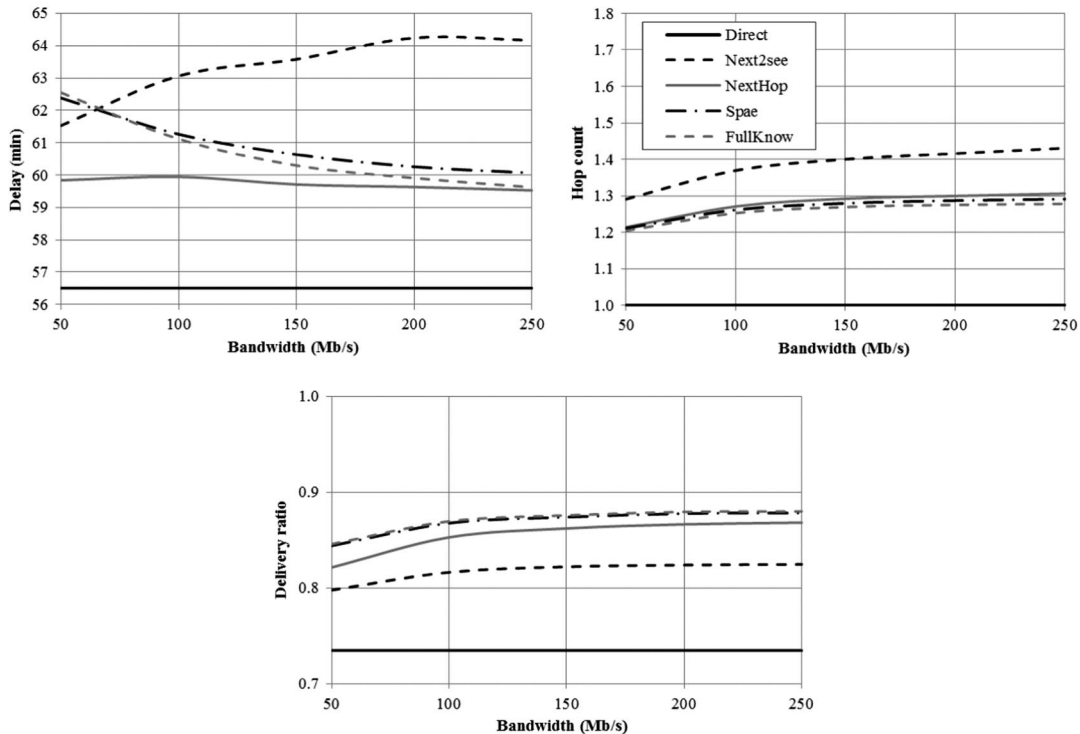


Fig. 8 Effect of ISL bandwidth on delay (top left), hop count (top right), and delivery ratio (bottom).

expected, given that hop count would reach some maximum value at a particular ISL range, before reducing to 1 at the point that ISL range is large enough for all nodes to be in continuous contact with a destination. In a similar manner, the delivery ratio would continue to increase beyond the range shown here, until the network reaches a level of connectivity that enables delivery of all data before its expiry.

3. Effect of Intersatellite Link Bandwidth

The final set of results (Fig. 8) shows the effect of varying ISL bandwidth while maintaining constant ISL range of 1500 km and a finite TTL of 180 min. As seen in the infinite TTL case (Fig. 5), the benefit of ISL bandwidth greater than 100 Mb/s in this scenario is limited, again indicating that the majority of packets that want to be transferred during a contact are able to be transferred. However, it is considered probable that a more congested network would benefit from greater bandwidth because the data traffic will likely be higher. Spae continues to exhibit comparable results to the packet-optimal FullKnow strategy, whereas delivery ratio from the NextHop protocol remains consistently lower throughout. Next2see is once again seen to exhibit different trends to the other strategies, which is attributed to its lack of effective journey awareness. This becomes more pronounced at higher bandwidth because more packets are inevitably sent along poor route choices.

VII. Conclusions

The routing protocol introduced here, called Spae, is shown to offer an effective approach to routing data packets through delay-tolerant networks that exhibit deterministic contact schedules. In particular, it is posed that Spae is particularly well suited to satellite constellations or federated satellite systems (FSSs), in which node mobility is well known and stable, but repetition of the network topology is not necessarily guaranteed. The addition of resource consideration, which is typically omitted in traditional Dijkstra-based routing algorithms, enables performance comparable to a full knowledge strategy that presents a hypothetical upper bound. Simulations carried out on various configurations, which consider finite and infinite packet lifetimes, various ISL ranges, and various ISL bandwidths, verify consistently good performance from Spae.

In the case of infinite packet lifetime, the effect of resource consideration is a general improvement in performance (joint combination of delay and hop count) compared with other methods. This is due to the reduced frequency of routing over potentially congested journeys, which avoids excessive rerouting having to take place. With finite packet lifetime, Spae typically ensures greater delivery ratio, which can be attributed again to the more effective routing of packets along optimal journeys with fewer attempts. The fewer rerouting instances that are necessary due to a failed delivery attempt, the lower the delay is for any individual packet and thus a greater likelihood of it being delivered before expiry.

Future work shall focus on the consideration of additional layers in the communication stack, in particular at the data-link and physical layers, and how different approaches may affect performance. This should include assessment of contact initiation and overhead associated with the handshake interaction between each node pair at the start of each contact event. In addition to this, the demand on the network, in both time and storage, relating to sharing of resource and buffer information shall be taken into consideration. Although the resource consideration approach taken in Spae provides an interesting first step into the use of additional parameters for data routing, it is expected that better estimates of downstream resources can be obtained and shall be considered during future studies.

The use of alternative, discrete-event modeling architectures shall be investigated further to exploit their network-specific capabilities. Further to this, an in-depth investigation into the effects of different network utilization shall be carried out, such that performance under low, medium, and high traffic loads is better understood. Comparison against other available protocols is also anticipated to identify benefits and drawbacks of Spae in this context.

One limitation that will be addressed during future work is the limited ability of Spae to identify the optimal next hop (for sending data) when two or more downstream contacts are available simultaneously (i.e., outdegree > 1). In most cases, and indeed in the investigations presented in this article, this condition is rare and has a negligible impact on results; however, for a network with higher connectivity, this should be addressed. Scalability of Spae shall be investigated further in the future, including the effects of network size, connectivity, and packet number. It is expected that, because of the dependence on Dijkstra's algorithm and the packet-by-packet routing approach, Spae is better suited to small networks, whereas larger networks comprising hundreds of nodes would be better serviced by other, less information-intensive protocols.

Finally, an increase in the model fidelity to incorporate other layers in the protocol stack should be carried out. Specifically, the physical and data link layers are important components that will help to identify overhead requirements of Spae and the mobility-specific limitations associated with contact initiation, such as the impact from Doppler effects on the signals.

References

- [1] Fall, K., "A Delay-Tolerant Network Architecture for Challenged Internets," *Proceedings of the 2003 Conf. on Applications, Technologies, Architectures, and Protocols for Computer Communications — SIGCOMM '03*, ACM, New York, 2003, pp. 27–34.
doi:10.1145/863956.863960
- [2] Golkar, A., and Lluich, I., "The Federated Satellite Systems Paradigm: Concept and Business Case Evaluation," *Acta Astronautica*, Vol. 111, June–July 2015, pp. 230–248.
doi:10.1016/j.actaastro.2015.02.009
- [3] Xuan, B., Ferreira, A., and Jarry, A., "Computing Shortest, Fastest, and Foremost Journeys in Dynamic Networks," *International Journal of Foundations of Computer Science*, Vol. 14, No. 2, 2002,
doi:10.1142/s0129054103001728
- [4] Jain, S., Fall, K., and Patra, R., "Routing in a Delay Tolerant Network," *Proceedings of the 2004 Conf. on Applications, Technologies, Architectures, and Protocols for Computer Communications — SIGCOMM '04*, ACM, New York, 2004, pp. 145–158.
doi:10.1145/1030194.1015484
- [5] Gnawali, O., Polyakov, M., Bose, P., and Govindan, R., "Data Centric, Position-Based Routing in Space Networks," *Proceedings of the IEEE Aerospace Conf.*, IEEE Publ., Piscataway, NJ, 2005,
doi:10.1109/AERO.2005.1559422
- [6] Demmer, M., and Fall, K., "DTLSR: Delay Tolerant Routing for Developing Regions," *Proceedings of the Workshop on Networked Systems for Developing Regions*, ACM, New York, 2007,
doi:10.1145/1326571.1326579
- [7] Mundur, P., Sookyoung, L., and Seligman, M., "Routing for Data Delivery in Dynamic Networks," *Proceedings of the IEEE Military Communications Conf. MILCOM*, IEEE Publ., Piscataway, NJ, 2007,
doi:10.1109/MILCOM.2006.302555
- [8] Di Nicola, A., and Giaccone, P., "Performance Limits of Real Delay Tolerant Networks," *Wireless On-Demand Network Systems and Services (WONS)*, IEEE Publ., Piscataway, NJ, 2008, pp. 149–155.
doi:10.1109/WONS.2008.4459370

- [9] Kandhalu, A., and Rajkumar, R., "QoS-Based Resource Allocation for Next-Generation Spacecraft Networks," *Proceedings of the 2012 IEEE 33rd Real-Time Systems Symposium*, Vol. 6, IEEE Publ., Piscataway, NJ, Dec. 2012, pp. 163–172.
doi:10.1109/rtss.2012.68
- [10] Tasiopoulos, A. G., Tsirar, C., and Toumpis, S., "Optimal and Achievable Cost/Delay Tradeoffs in Delay-Tolerant Networks," *Computer Networks*, Vol. 70, Sept. 2014, pp. 59–74.
doi:10.1016/j.comnet.2014.05.006
- [11] Alonso, J., and Fall, K., "A Linear Programming Formulation of Flows over Time with Piecewise Constant Capacity and Transit Times," Intel, 2003.
- [12] Bocquillon, R., and Jouglet, A., "Data Transfer in Delay-Tolerant Networks," *Proceedings of the 8th International Conf. on Broadband and Wireless Computing, Communication and Applications*, IEEE Publ., Piscataway, NJ, 2013, pp. 355–359.
doi:10.1109/BWCCA.2013.62
- [13] Burleigh, S., Hooke, A., Torgerson, L., Fall, K., Cerf, V., Durst, B., Scott, K., and Weiss, H., "Delay-Tolerant Networking: An Approach to Interplanetary Internet," *IEEE Communications Magazine*, Vol. 41, No. 6, 2003, pp. 128–136.
doi:10.1109/mcom.2003.1204759
- [14] Cerf, V., Burleigh, S., Hooke, A., Torgerson, L., Durst, R., Scott, K., Travis, E., and Weiss, H., "Interplanetary Internet (IPN): Architectural Definition," Internet Engineering Task Force, 2001.
- [15] Farrell, S., and Cahill, V., "Evaluating LTP-T: A DTN-Friendly Transport Protocol," *Proceedings of the 2007 International Workshop on Satellite and Space Communication, IWSSC'07*, IEEE Publ., Piscataway, NJ, 2007, pp. 178–181.
doi:10.1109/IWSSC.2007.4409412
- [16] Wood, L., Eddy, W. M., Ivančić, W., McKim, J., and Jackson, C., "Saratoga: A Delay-Tolerant Networking Convergence Layer with Efficient Link Utilization," *Proceedings of the 2007 International Workshop on Satellite and Space Communication, IWSSC'07*, IEEE Publ., Piscataway, NJ, 2007, pp. 168–172.
doi:10.1109/IWSSC.2007.4409410
- [17] Cruz-Sánchez, H., Franck, L., and Andre-Luc, B., "Use of Store and Forward Metrics for Service-Oriented Routing in Satellite Constellations," *Proceedings of Satellite and Space Communications, IWSSC*, IEEE Publ., Piscataway, NJ, 2008, pp. 80–84.
doi:10.1109/IWSSC.2008.4656753
- [18] Lluch, I., Grogan, P. T., Pica, U., and Golkar, A., "Simulating a Proactive Ad-Hoc Network Protocol for Federated Satellite Systems," *Proceedings of the IEEE Aerospace Conf.*, IEEE Publ., Piscataway, NJ, 2015, pp. 1–16.
doi:10.1109/AERO.2015.7118984
- [19] Delosières, L., and Nadjm-Tehrani, S., "BATMAN Store-and-Forward: The Best of the Two Worlds," *Proceedings of the 2012 IEEE International Conf. on Pervasive Computing and Communications Workshops, PERCOM Workshops 2012*, IEEE Publ., Piscataway, NJ, March 2012, pp. 721–727.
doi:10.1109/PerComW.2012.6197608
- [20] Clausen, T., Jacquet, P., Adjih, C., Laouti, A., Minet, P., Muhlethaler, P., Qayyum, A., and Viennot, L., "Optimized Link State Routing Protocol (OLSR)," Network Working Group, 2003.
- [21] Burleigh, S., "Contact Graph Routing," Internet Engineering Task Force, 2009.
- [22] Seguí, J., Jennings, E., and Burleigh, S., "Enhancing Contact Graph Routing for Delay Tolerant Space Networking," *Proceedings of the IEEE Global Communications Conf.*, IEEE Publ., Piscataway, NJ, 2011,
doi:10.1109/GLOCOM.2011.6134460
- [23] Birrane, E., Burleigh, S., and Kasch, N., "Analysis of the Contact Graph Routing Algorithm: Bounding Interplanetary Paths," *Acta Astronautica*, Vol. 75, June 2012, pp. 108–119.
doi:10.1016/j.actaastro.2012.02.004
- [24] Bezirgiannidis, N., Burleigh, S., and Tsaoussidis, V., "Delivery Time Estimation for Space Bundles," *IEEE Transactions on Aerospace Electronic Systems*, Vol. 49, No. 3, 2013, pp. 1897–1910.
doi:10.1109/taes.2013.6558026
- [25] Bezirgiannidis, N., and Tsaoussidis, V., "Predicting Queueing Delays in Delay Tolerant Networks with Application in Space," *Proceedings of the 12th International Conf. on Wired & Wireless Internet Communications (WWIC 2014)*, Springer, 2014, pp. 228–242.
doi:10.1007/978-3-319-13174-0_18
- [26] Bezirgiannidis, N., Tsapeli, F., Diamantopoulos, S., and Tsaoussidis, V., "Towards Flexibility and Accuracy in Space DTN Communications," *Proceedings of the 8th ACM MobiCom Workshop on Challenged Networks — CHANTS '13*, ACM, New York, 2013, pp. 43–48.
doi:10.1145/2505494.2505499
- [27] Bezirgiannidis, N., Caini, C., Padalino Montenero, D. D., Ruggieri, M., and Tsaoussidis, V., "Contact Graph Routing Enhancements for Delay Tolerant Space Communications," *Proceedings of the 7th Advanced Satellite Multimedia Systems Conf. and the 13th Signal Processing for Space Communications Workshop (ASMS/SPSC)*, IEEE Publ., Piscataway, NJ, 2014, pp. 17–23.
doi:10.1109/ASMS-SPSC.2014.6934518
- [28] Bezirgiannidis, N., Caini, C., and Tsaoussidis, V., "Analysis of Contact Graph Routing Enhancements for DTN Space Communications," *International Journal of Satellite Communications and Networking*, 2015,
doi:10.1002/sat1138
- [29] Dijkstra, E., "A Note on Two Problems in Connexion with Graphs," *Numerische Mathematik*, Vol. 1, No. 1, 1959, pp. 269–271.
doi:10.1007/bf01386390
- [30] Chen, W., Palmer, P., Mackin, S., and Crowley, G., "Queueing Theory Application in Imaging Service Analysis for Small Earth Observation Satellites," *Acta Astronautica*, Vol. 62, Nos. 10–11, 2008, pp. 623–631.
doi:10.1016/j.actaastro.2008.01.026
- [31] Lluch, I., and Golkar, A., "Satellite-to-Satellite Coverage Optimization Approach for Opportunistic Inter-Satellite Links," *Proceedings of the IEEE Aerospace Conf.*, IEEE Publ., Piscataway, NJ, 2014,
doi:10.1109/AERO.2014.6836307