A Novel Alternative to Bundle Protocol for Handling Data Transmission Across Disruption-Tolerant Networks

Caitlyn A. K. Singam

Abstract Communication networks are prone to disruption due to inherent uncertainties such as environmental conditions, system outages, and other factors. However, current communication protocols for state-of-the-art disruption-tolerant networks (DTNs) designed to withstand such conditions are not yet optimized for high performance over long distances, such as those encountered in deep space. Current DTN communication protocols have been documented in the literature as inherently assuming relatively low levels of signal loss, not accounting for end-to-end error rate, and presuming a lack of performance constraints governing optimal communication function. However, these assumptions and constraints frequently do not hold true outside of theoretical scenarios; therefore, there is a need for an improved communication protocol that has the ability to minimize data loss to tolerable levels over an unstable and error-prone communication link. Furthermore, any novel communication protocol should also be able to optimize transmission time: this is because current communication networks for parts of space prone to signal disruptions, particularly deep space, are fairly slow and have a low data rate, since transmitters have to trade speed for accuracy when transmitting data at a particular power level directly from deep space to Earth. Bundle protocol (BP) is an experimental protocol for handling packet transmission through DTN networks that has a number of vocal proponents in the academic and the aerospace community; however, as noted by authors of the protocol, there are a number of key areas of concern associated with BP approach, including, but not limited to, high vulnerability to denial of service (DoS) attacks and issues efficiently handling congestion and flow control schemes implemented across highly variable delay environments. BP, as a protocol which "sits at the application layer of some number of constituent internets", also utilizes internet protocols such as Transmission Control Protocol/Internet Protocol (TCP/IP) and similar alternatives to handle lower-level management of data transfer, and thus inherits the limitations associated with the implementations of such approaches (as well as those that emerge at the interface of protocols at each level), creating further vulnerabilities for potential exploitation by nefarious agents or reductions in system performance due to poor environmental conditions.

C. A. K. Singam (✉)
Department of Bioengineering, University of Maryland, College Park, Maryland 20742, USA
e-mail: csingam@terpmail.umd.edu

C. Cruzen et al. (eds.), *Space Operations*, Springer Aerospace Technology,
https://doi.org/10.1007/978-3-030-94628-9_18

This work concerns the development of a novel protocol for data transmission across delay/disruption-tolerant networks, which is presented as an alternative to the bundle protocol standard. The alternative proposed herein seeks to address some of the limitations seen in bundle protocol and provide a DTN networking option with wider usability, better reliability, and improved immunity to DoS attacks. In particular, the efficacy of the proposed approach, in terms of maintaining both data integrity and transmission speed, was evaluated via simulation against BP and a set of other alternative DTN data handling methodologies from the literature and demonstrated a statistically significant improvement in performance compared to BP and other canonical communication protocols. The result is presented herein in terms of its ramifications for future DTN implementations.

Keywords Communications · Network · Signal optimization · Bundle protocol · Disruption tolerant networking · DTN

Acronyms/Abbreviations

BP	Bundle Protocol
CCSDS	Consultative Committee for Space Data Systems
COA	Course of Action
DoS	Denial of Service
DTN	Disruption-Tolerant Network(ing)
MATLAB	Matrix Laboratory Software
MAVF	Multi-Attribute Value Function
TCP/IP	Transmission Control Protocol/Internet Protocol

1 Introduction

The objective of this analysis was to simulate the performance of three different ad-hoc protocols for disruption-tolerant networking—i.e., the transfer of information through a network of nodes in contexts prone to signal interruption/signal degradation—and to perform a trade-off analysis that would yield a recommendation of the best course of action (COA) for space-based network communication.

This is important for space-based networks in particular since transmitting information over long distances—e.g. directly from the initial node to the destination node—will result in the terminal signal being relatively weak. This is problematic in a high noise (disruption-prone) environment since it will likely result in packet degradation or loss unless signal power is increased to compensate. Given that changing signal power for each transmission is impractical when one's network is located in

space, optimization of the signal route is the best means of ensuring the transmitted signal reaches its destination rapidly and with maximum fidelity.

The recommended COA, as determined at the end of the analysis, is one that optimizes performance (in terms of selected parameters) in a fashion that minimizes error during transmission and transmission time to the greatest extent possible. This analysis takes into consideration the relative value of transmission time and transmission error associated with space-based communication systems (for instance, a scientific mission—the most likely type of user for a space-based relay network [1]—would prioritize data integrity much higher than transmission time since even though a longer transmission time equals greater cost, a lower level of data integrity could result in the mission's scientific objective being compromised [2]) and provides a recommendation accordingly.

In order to evaluate the different design options, two different principal metrics of interest were be taken into account: percent error on receipt (a representation of packet integrity) and transmission time (a representation of transmission speed).

At the end of the analysis, the calculated values for percent error and transmission time for each design option were combined together using a multi-attribute value function (MAVF) to yield a single quantitative value representing the relative value of each solution. The MAVF can produce values ranging from 0 (representing the worst option) to 1 (the optimal design option). The analysis objective was considered satisfied when the design option with the highest MAVF ranking was identified.

Three different design options were considered: bundle protocol, the current state of the art routing protocol for DTN [3] which picks the route that strikes a balance between distance and signal quality; distance-based Dijkstra, which selects a route that minimizes the distance travelled during transmission; and a novel value-based approach, which selects a route with the best overall signal quality and based on the overall value of the message to the user on receipt after accounting for incurred errors and transmission delays. The novel approach described herein is the "signal quality-based Dijkstra", a shorthand for the fact that it seeks to minimize the errors incurred during transmission à la the Dijkstra algorithm.

To evaluate the different design options, a generic/hypothetical space-based network comprised of 10 satellites was generated to route data from an initial node to a destination node. Apart from the distance between the initial node and destination node, which was fixed, the distance and signal quality for the link between any two nodes in the network was instantiated randomly. A Monte Carlo simulation was then used to simulate the routing of 500 packets through the network, with each packet representing one 'sample', or iteration of the Monte Carlo simulation, using the three different routing protocols being evaluated. The Monte Carlo simulation was used to vary the signal quality and inter-node distance associated with each link in the network each step, in order to represent variations due to environmental phenomena and orbital movement.

At the end of each iteration, the transmission time for each packet (based on the distance traveled by the packet, and given that the packet is transmitted as an electromagnetic wave traveling at the speed of light) and the packet's state (intact or damaged) was determined. The aggregate data from the entire simulation was then

used to calculate the mean transmission time and percent error for the overall sample (determined based on the terminal state of each packet on receipt), both of which were in turn fed into a MAVF. The recommended design option was chosen based on which option was associated with the highest MAVF output value. Welch's t-tests were also used to confirm that the differences in metric values observed were in fact statistically significant: i.e. that there was enough of a difference in the performance of the different options for the choice to have a significant impact on overall system performance.

2 System Description

Since the purpose of this analysis was to determine the relative performance of different disruption-tolerant networking protocols for a generic space-based use case, the analysis used a generic, hypothetical network as its system of interest.

The modeled system (as depicted in the domain block definition diagram shown in Fig. 1) was comprised of a network of 10 space-based relay satellites, located at different distances from each other and from the Earth-based ground station. The satellites, which were modeled as nodes in a graph, all had communication links with one another, as well as with an initial node (the space-based asset generating the data being transmitted) and a terminal node (the ground station). The distances in between nodes, as well as the signal quality values for the links between nodes, were determined randomly; the one exception to this was the distance between the transmitting node (noted as a 'deep space asset/probe' in Fig. 1) and the receiving ground station, which were placed at a fixed distance from one another in order to set the scale (i.e. maximum distance) the network would operate at. The value used for the internode distance between the initial and destination nodes was the

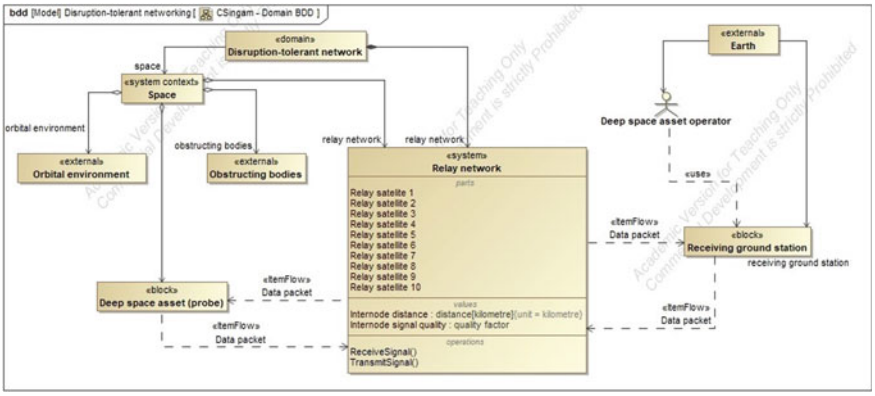

Fig. 1 A domain block definition diagram showing the principal components of the analyzed system

maximum distance from Earth to Titan, 1.27×10^9 km, as an actual mission—the Cassini-Huygens mission—transmitted data back to Earth at that distance [4], and the distance thus is an accurate representation of the conditions that would have to be handled by a space-based network attempting to implement any of the tested routing protocols on a practical basis.

The signal receipt and signal transmission operations that are shown as operations of the satellites within the relay network system in Fig. 1, when taken in aggregate for all the satellites within the network, represent the system behaviour dictated by the network's routing protocol. The system's ability to transmit and receive signals among its component satellites is influenced by two key input factors, internode distance (expressed in kilometers) and internode signal quality (expressed as a percentage of the theoretical ideal for signal performance).

Internode distance, determined by the user's existing asset placement, is a key factor since the time it takes to transmit a signal from point A to point B is, inherently, tied to distance: since the electromagnetic waves used to convey signals travel at the speed of light, c, the time taken to travel between node A and node B is dictated by the distance between the two nodes divided by c. Links with shorter internode distances are thus preferable to those with longer distances, as shorter transmission time is always preferable.

Signal quality is also of interest due to the fact that it is a direct reflection of the role environmental context plays in determining the quality of a link and how much of the transmitted signal gets successfully received by the end node. It represents link quality as a percent of the theoretical ideal/maximum, ranging in value from 0% to 100%, and can be calculated based on historical link performance data and information from environmental models. Though higher signal quality is always preferable, signal quality levels near the theoretical maximum of 100% are rare due to the inherent noisiness of a real-world environment. Disruption-tolerant networks need to take into account, or alternatively be resilient against, the effects of varying and/or poor signal quality lest the data payloads they are transmitting be lost or damaged.

Together, these two factors influence the two main metrics of interest for any transmission that moves through the system: percent error on receipt and transmission time.

Percent error on receipt, or the percentage of packets out of the total number initially transmitted which are damaged and/or lost during transmission, can range from 0 to 100%, but should be as low as possible (i.e. as close to 0% as possible) in order to respect the need for data integrity during transmission. This metric is viable for evaluating the performance of real-world networks since the number of packets and the number of bytes per packet in a given transmission will be defined and kept constant for a particular mission. Thus, when the users of the Earth-based ground station receive packets related to a given mission, they can compare the number of packets (and bytes) received with the number that should have been sent by the mission. Given that the entire point of a transmission is to ensure that the terminal node receives the information that was being transmitted, ensuring that percent error

is as low as possible, and that as much of the data payload as possible is intact, is a high priority.

Transmission time, the amount of time (in hours) it takes for a single packet to travel from the initial transmitting node to the terminal destination (receiving) node, has a theoretical minimum dictated by the time it would take a pulse traveling at the speed of light to travel the straight-line path from initial to terminal node. However, since relay networks route packets through non-linear paths to their destination and increase transmission time beyond the minimum, it is thus important to consider how much additional transmission time the protocol incurs and attempt to keep the overall transmission time as close to the theoretical minimum as possible. For missions where data packets may contain time-sensitive commands, it is imperative to ensure that transmission time does not become excessively long.

Percent error and transmission time are the primary criteria by which the different routing protocols (the design options being evaluated) used by the network can be evaluated, and will be the focus of this analysis.

3 Design Options

The three design options being evaluated as part of this analysis are bundle protocol, distance-based Dijkstra, and signal-quality based Dijkstra. Their relative performance with respect to the two metrics of interest (described above) is shown in Table 1. The exact metric values for each design option were determined via the Monte Carlo simulation.

The details of how each of the design options work, and the parameters prioritized by each, are presented below.

3.1 Bundle Protocol

Bundle protocol is reflective of the store-and-forward methodology put forth by the Consultative Committee for Space Data Systems (CCSDS) [5] for packet routing.

Table 1 A comparison of the relative performance of the design options being evaluated, with regards to the metrics of interest

Metric	Design options		
	Bundle protocol	Distance-based Dijkstra	Quality-based Dijkstra
% error	Medium	High (suboptimal)	Low (optimal)
Transmission time	Medium	Low (optimal)	Medium to high (suboptimal)

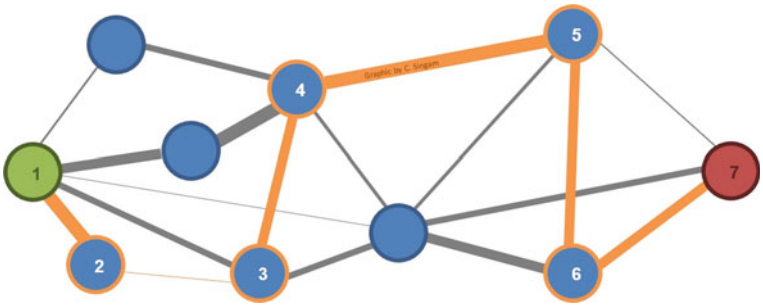

Fig. 2 An example of how bundle protocol routes information through a network from the initial (green) node to the terminal (red) node, with nodes numbered in order of visitation and the path marked in orange. Line widths correspond to signal quality (thicker lines correspond to higher quality links), and distances are to scale

As shown in Fig. 2, in a network following bundle protocol, packets are transmitted from the node they are the currently on to the adjacent/neighbouring node that meets the criteria of being A) closer than the current node to the destination node, and B) having the highest signal quality of the nodes that meet criterion A. This process is repeated until the packet reaches its destination. This typically results in the packets being routed through more nodes than in either of the other two methods, though depending on the length of the links used, this may not necessarily correspond to a longer transmission time or worse signal quality.

3.2 Distance-Based Dijkstra

This design option uses Dijkstra's algorithm to find the path with the shortest distance to the terminal node, which travels through at least one intervening relay node. The stipulation that the recommended path include at least one relay node is to eliminate direct-to-Earth transmission routes, which have already been shown in the literature [6] to be outperformed by relays. This is achieved by instantiating the edge costs of each link as the corresponding internode distance, apart from the link from the initial node to the terminal node, which is instantiated such that the edge cost is significantly higher than the edge cost of any other link and thus resulting in Dijkstra's algorithm rejecting the direct route as a possible path.

The results of this algorithm (exemplified in Fig. 3) usually yield a path that is as close to the straight-line path (marked using a dashed line in the diagram) as possible, resulting in a short transmission time but frequently resulting in a path that includes low-quality links that degrade the signal prior to receipt.

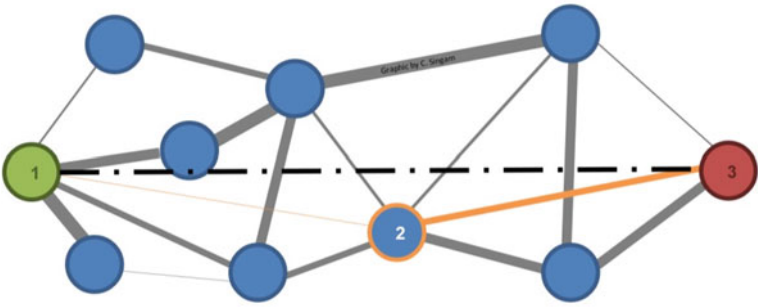

Fig. 3 An example of how the distance-based Dijkstra method routes information through a network from the initial (green) node to the terminal (red) node, with nodes numbered in order of visitation and the path marked in orange. Line widths correspond to signal quality (thicker lines correspond to higher quality links), and distances between nodes are to scale. The dashed line marks the straight-line path from the initial node to the destination

3.3 Signal Quality-Based Dijkstra

This design option evaluates the highest-quality path through the network using the Dijkstra algorithm, using 1—[internode signal quality as a decimal probability] as calculated between each pair of nodes, rather than over an entire path, as edge costs for the network. Since the Dijkstra algorithm selects the path with the minimum cost, and it is desired to maximize the signal quality seen on the chosen route, the edge costs are instantiated as the complement of the parameter of interest: in this case, signal quality.

Since this algorithm uses Dijkstra shortest-path algorithm with complement edge costs, rather than a longest-path algorithm in conjunction with a graph that uses signal quality directly as the edge cost, the results of this algorithm favour paths (exemplified in Fig. 4) with relatively few links and high signal quality. Consequently, it is good at minimizing percent error but since the algorithm favours fewer links, not necessarily shorter ones, it is suboptimal at minimizing transmission time.

4 Simulation Description

The primary objective of this analysis was to identify the mean values associated with each of the design options for the performance metrics of interest, so that a MAVF analysis could be performed and the design options could be quantitatively ranked. However, given that using signal quality-based Dijkstra for space-based networking contexts is, to the best of the author's knowledge, a concept of the author's own devising, no data exists on the performance of such methodologies for the desired use case and in the desired system context. However, given the extravagant financial and scheduling burden that would be involved in constructing any sort

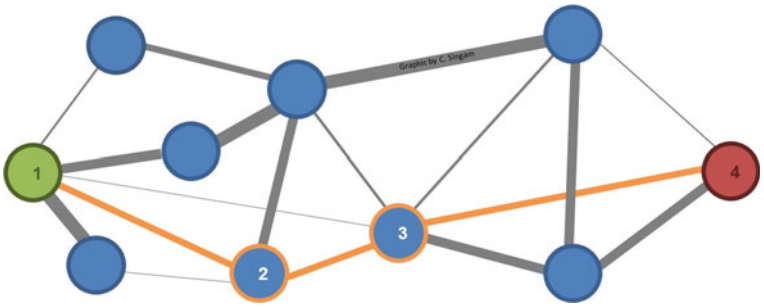

Fig. 4 An example of how signal quality-based Dijkstra routes information through a network from the initial (green) node to the terminal (red) node, with nodes numbered in order of visitation and the path marked in orange. Line widths correspond to signal quality (thicker lines correspond to higher quality), and distances are to scale

of reasonable prototype of a space-based network—or otherwise acquiring even temporary access to existing networks [7]—for testing purposes, obtaining real-world data on the performance of these protocols is, disappointingly, highly impractical. Fortunately for the aerospace community, creating a model of a space-based network and simulating the movement of packets through it using various routing protocols is significantly more practical, providing the data that allows the age-old quandary (of which packet routing protocol performs best in a space-based context) at the heart of this analysis to be answered satisfactorily.

To that end, a multi-step approach was taken to develop a model of sufficient faithfulness to reality as to accurately test the mettle of the three design options of interest. Firstly, the positions of the initial and destination nodes were established in a MATLAB model; the relay network satellites were randomly instantiated so that any one satellite had an equal probability of appearing anywhere between the initial node and destination node.

After each node had been linked with each other node, and all the links had been assigned distances and random signal qualities, the movement of 500 packets through the relay network was simulated using the Monte Carlo simulation method. The packet state and transmission time were recorded for each packet when it was simulated as having reached its destination. Once all 500 packets were simulated as having traveled through the system, the percent error for the population was calculated by totaling the number of packets recorded as having been lost and dividing that by the sample size (500 packets). The mean transmission time associated with each design option was also calculated.

Since each run of the Monte Carlo simulation is associated with a different randomly instantiated graph due to the way the simulation setup was implemented in the MATLAB code, it is possible to run the simulation code multiple times in order to gain an accurate image of how all three design options perform across several different networks. Due to the processing power and time needed to run each simulation, a relatively small sample of 5 runs was used. The values for percent error and

transmission time for each run were collected in Excel, and the mean, standard deviation, and standard error of the mean for the two metrics of interest were calculated for each of the design options.

The obtained 5-run means for percent error and transmission time were subsequently fed into a MAVF for final analysis [8]. The metrics were assigned preference weights per the Parnell swing weight matrix [9]: percent error, as a Parnell 'mission critical/large effect' parameter (a metric which needs to be optimized in order to ensure mission success and which mission success is sensitive to) was assigned a weight of 100; transmission time, as a 'mission effectiveness/small effect' parameter (a metric which can be used to compare the relative worth of design options, but which mission success is not as sensitive to) was assigned a weight of 20. The results of the MAVF for each design option were then directly compared and used to make the final recommendation. Additionally, as a confirmatory measure for whether the designs performed significantly differently from one another, the 5-run data means and standard deviations were used to perform one-tailed Aspin-Welch t-tests [10] to identify whether or not the differences seen between design options for both transmission time and percent error were statistically significant ($p < 0.05$). The equation for the t-statistic comparing the means of samples from two different populations (population 1 and population 2) is:

$$t = \frac{\overline{x_1} - \overline{x_2}}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} \tag{1}$$

where $\overline{x}$ is the mean for a sample, s is the sample's standard deviation, and n is the sample size.

The workflow and methodology for the trade-off analysis is summarized in Fig. 5.

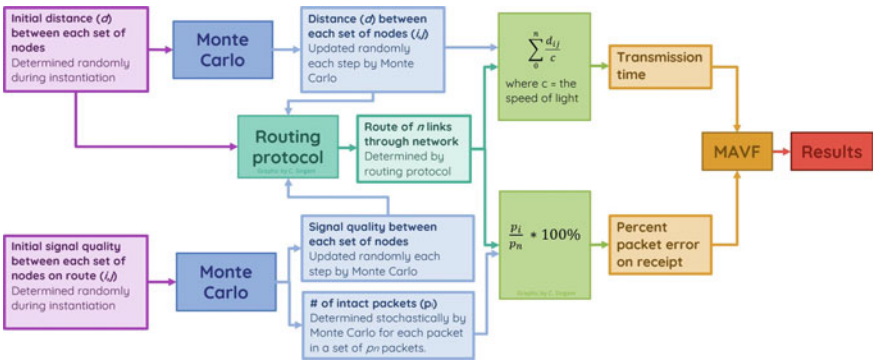

Fig. 5 A combined response model diagram for the entire analysis, showing the initial input factors (purple) and intermediate factors (blue) produced by the Monte Carlo simulation and fed into the routing protocols being evaluated (turquoise). The equations used to calculate the metrics of interest (green) as they are fed into the MAVF (yellow) are also shown

5 Supporting Methods

5.1 Network Graph Models

The relay network that the packets are routed through is represented in the programmatic implementation as a digraph, which is compatible with MATLAB's Dijkstra implementation via the `shortestpath` function [11]. The use of network graphs to model satellite networks is not unusual [12, 13]; however, such graphs are typically used to model control capabilities rather than routing.

The two network digraphs used in this analysis are nearly identical to each other for any given run of the MATLAB program. Each accepts the list of nodes (the 10 relay satellites, the probe, and the ground station) as inputs, as well as arrays representing the links in between nodes (the same for both). The output of these both these models are the fully instantiated and functional digraphs that the Monte Carlo simulation can route its packets through and that the Dijkstra algorithm can be applied to.

As discussed previously, to prevent the direct-to-Earth link (i.e. straight line path from initial to destination node) from being used , the edge cost for the link associated with the direct-to-Earth route on each graph is assigned a value that far exceed the maximum possible value for the edge costs on the other links.

The only difference between the two digraphs is the parameter used as the edge cost for each.

Distance-based graph

Before the graph models are instantiated, each node is assigned an x and y coordinate on the Cartesian coordinate plane (with units in kilometers). Though the coordinates for the initial node and the destination node are fixed so that they are diametrically apart on the coordinate plane and at Titan distance from one another, the relay network's satellites are all assigned coordinates randomly, per a uniform distribution. The lower limit on coordinate distances is defined as 10^4 km (the same as low Earth orbit [14]), and the upper limit is defined by the location of the transmitting probe.

The distance-based digraph uses the randomly assigned position of each node to calculate internode distances and assign edge costs for each link (with the exception of the direct-to-Earth link, as described above). The distance formula is:

$$Distance_{AB} = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2} \tag{2}$$

For convenience in implementation, since the metric of interest is transmission time (not transmission distance)—which is distance divided by a constant (the speed of light)—the edge costs for the distance-based digraph are actually assigned transmission times as edge costs. This does not result in any practical difference in how the model functions or how the Monte Carlo simulation interacts with it; it simply makes programmatic implementation easier.

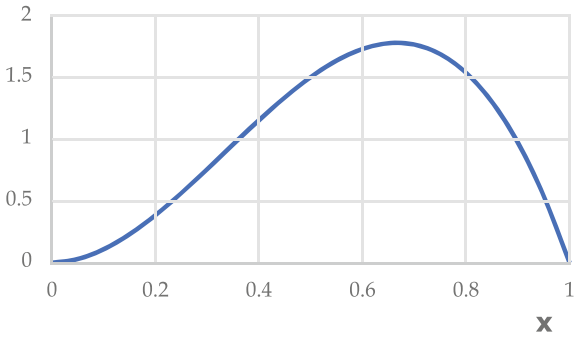

Fig. 6 Beta distribution, a = 3, b = 2. Graph generated using Excel and data from Casio's Keisan online calculator [15]

Signal quality-based graph

Since signal quality can theoretically vary between 0 and 1 (when expressed as decimals instead of as percentages), but must to skew slightly more to the right than a normal distribution (since satellites are designed with antennas, etc. that are optimized for performance in their environments), a beta distribution was used to randomly determine the signal quality to each link in the model. Figure 6 shows the beta distribution with the parameters selected for this analysis (a = 3, b = 2).

The complement of the signal quality values produced by the beta distribution are then calculated using 1-[signal quality] and assigned to the corresponding links as edge costs.

5.2 Monte Carlo Simulation

The Monte Carlo simulation used in this analysis follows standard implementation methodology presented in the literature [16]. The inputs to the simulation are the fully instantiated digraphs described above, with edge costs reflecting the randomly assigned internode distances and signal qualities.

Each iteration of the simulation, a packet is routed through the network as represented by the digraphs. With each step, the packet advances from the node it is currently on to the next node in the path dictated by each routing protocol. In effect, there are three 'copies' of the packet moving through the network simultaneously—each subject to a different protocol, but all experiencing the same conditions. Each 'hop' between nodes is associated with transmission time—added to a running total that is recorded when the packet reaches its destination—and the possibility that the packet has been lost/damaged mid-transmission, which is determined stochastically by comparing a randomly generated decimal (between 0 and 1) with the assigned signal quality (not to be confused with the edge cost) of the link that has been traveled over. If the random decimal is lower than the link quality, the packet is deemed to have survived the 'hop', else it is recorded as having been irreparably damaged/lost. For the sake of data collection (and avoiding the complexities associated with censored data

[17]), 'lost' packets are not removed from the transmission queue and are simulated like intact packets all the way to their destination.

The edge costs are updated each step to represent environmental fluctuations; they are all randomly assigned new values per a normal distribution centered around their initial (i.e. at the start of the simulation) or 'default' value. The digraphs are re-established accordingly with the updated edge costs, and each of the routing protocols re-evaluate their recommended paths based on the node their 'copy' of the packet is currently on and the new edge costs.

Once all three 'copies' of the packet reach their destination, the transmission time (calculated based on distance traveled and independent of simulation time) and packet state for each copy is recorded. The digraphs are then reset to their state at the start of the simulation after each iteration for the sake of consistency between samples/packets, and the iteration ends. Each run of the Monte Carlo simulation simulates the movement of 500 packets through the system (i.e. there are 500 samples/iterations). This was determined to be a sufficient number of samples based on a cumulative running mean (CRM) plot for transmission time for all three routing protocols being evaluated: the mean for bundle protocol, the most variable of the three, stabilized after around 250 samples, which means that 500 samples provides a wide contingency margin (Fig. 7).

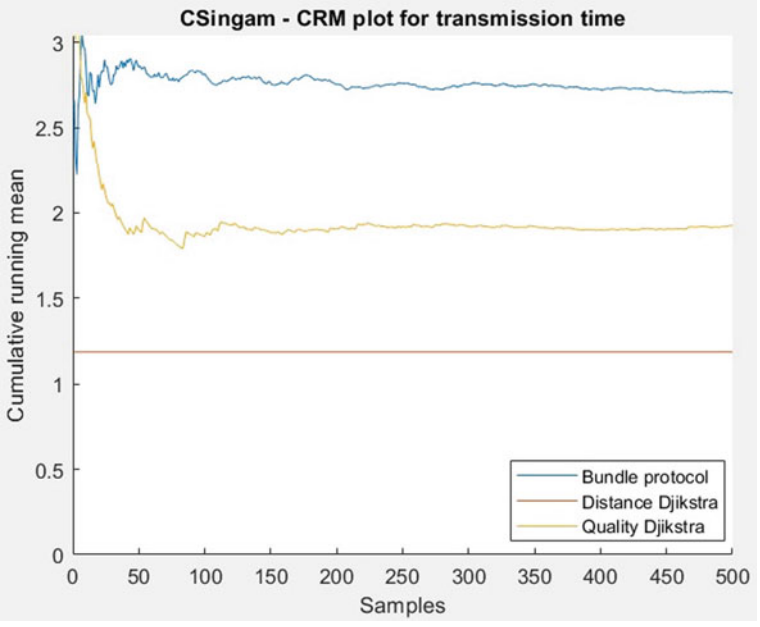

Fig. 7 Cumulative running mean plot for the Monte Carlo simulation, showing stabilization of the mean for all three protocols after around 250 samples

6 Simulation Verification and Validation

6.1 Verification and Validation Methodology

Given that the foundation for the analysis presented herein is a simulation, and thus a generalized representation of the actual behaviour of a real-world satellite network, it was necessary to perform verification and validation testing in order to ensure that the simulation was able to produce accurate and reliable results on a consistent basis as ascertained via verification testing, and that the simulation was able to provide the data needed to compare the performance of the different routing protocols which was verified via validation testing.

To this end, a series of element-level and system-level formal verification and validation tests were performed in order to make a final determination of whether the simulation was of an acceptable level of accuracy.

Development and testing, as with many software projects, followed a spiral development pattern, with individual elements being developed and tested before integration and testing of the larger integrated system. The deterministic module, which is the simpler of the two modules, was developed and white-box tested first, followed by the stochastic module. Due to the presence of a shared interface between the two modules—where parameters from the stochastic module are passed to the deterministic module to perform calculations within a single Monte Carlo event step—the stochastic module's development was nonetheless closely tied to that of the deterministic module, with both elements white-box tested simultaneously. Once integration of the deterministic module and the stochastic module was complete, an element-level integration black box test was performed. Given the successful black-box integration test the system was verified and validated via a final system-level black-box test.

Verification was be performed against the system requirements, and validation was done based on whether or not the simulation was able to provide the requisite information to perform the trade-off analysis the simulation was designed to support.

The elemental testing was conducted informally through code inspections and demonstrations to ensure that the code was functional, whilst the black-box validation testing of the integrated system was conducted formally (involving the development of formal test reports and collection of test data).

The measures of effectiveness (MOEs) that were be used to evaluate the simulation are described in Table 2.

6.2 Verification and Validation Success

The simulation was run through a number of test cases (Tables 3 and 4) as part of the black-box testing process, and the results compared to the manually-derived solutions for recommended route, transmission time, and predicted percent error for a particular network configuration. Path recommendations were verified based on

Table 2 The list of criteria used to evaluation whether the DTN simulation met the standard for acceptability

MOE	Definition
Requirements compliance	Percentage of system requirements met by the simulation, as determined via verification testing
Requirement deviance	Percentage of system requirements for which the system has received waiver(s). This is separate from requirements non-compliance, as deviations from the requirements require justification and stakeholder buy-in (specifically from the user)
Deterministic mode accuracy—protocol percent error	The deterministic mode result for protocol percent error divided by the result determined via manual calculations. (Fraction expressed as a percentage)
Deterministic mode accuracy—transmission time	The deterministic mode result for transmission time divided by the result determined via manual calculations. (Fraction expressed as a percentage)
Fidelity of protocol implementation	Accuracy of the paths recommended by the simulation's implementation of each routing protocol. This MOE is calculated by manually tracing the path recommended by each protocol, and comparing the ordered list of nodes visited with the path generated by the simulation's implementation of the same routing protocol. The fraction of nodes in the simulated path that match the order and ID of the manually identified path (out of the total number of nodes in the path) represents fidelity
Stochastic fidelity—protocol percent error	The average standard deviation error seen in a stochastic result for the metric of protocol percent error for a given set of inputs, relative to the deterministic result for those same inputs
Stochastic fidelity—transmission time	The average standard deviation error seen in a stochastic result for the metric of transmission time for a given set of inputs, relative to the deterministic result for those same inputs
Utility for trade-off analysis	The number of metrics produced by the simulation that can be used in the trade-off analysis, divided by the number of parameters needed to perform the trade-off analysis (i.e. evaluating if the simulation can support user needs)

the most common observed path for each routing protocol, compared against the average network state across all iterations. If the most common recommended path matched the expected path then there was said to be no deviation.

Both the deterministic and stochastic components of the simulation's calculation functions were verified and validated. The deterministic mode results were expected

Table 3 Part of the set of test cases used to exercise the deterministic mode of the DTN simulation

Test case	Max satellite position (km)	Transmitter position (km)	Min. link quality	Max. link quality
1	100	100	0	1
2	100	1,000,000	0	1
3	1,000,000	100	0	1
4	1,000,000	1,000,000	0	1
5	100	100	0.4	0.5
6	100	1,000,000	0.4	0.5
7	1,000,000	100	0.4	0.5
8	1,000,000	1,000,000	0.4	0.5

Table 4 Part of the set of verification test cases used to exercise the stochastic mode of the DTN simulation

Test case	Max satellite position (km)	Transmitter position (km)	Min. link quality	Max. link quality	Monte Carlo iterations
1	100	100	0	1	1000
2	100	1,000,000	0	1	1000
3	1,000,000	100	0	1	1000
4	1,000,000	1,000,000	0	1	1000
5	100	100	0.4	0.5	1000
6	100	1,000,000	0.4	0.5	1000
7	1,000,000	100	0.4	0.5	1000
8	1,000,000	1,000,000	0.4	0.5	1000
9	1,000,000	1,000,000	0	1	500
10	1,000,000	1,000,000	0	1	10

to match the manually calculated values exactly (or to be within 0.1% of the manually-calculated values in the case of rounded values). In the Monte Carlo results, the manually-derived values were compared to the steady-state Monte Carlo values (after running the simulation for 1000 iterations per test case) to see if they stabilized within ±5% of the predicted value.

Testing revealed that the final version of the simulation successfully passed all tests without issue and within acceptable time limits, as evinced by the results shown in Tables 5 and 6.

Table 5 Some of results obtained after black-box testing the deterministic components of the DTN simulation, showing T (transmission time) and E (percent error). N represents no observed deviation; the tilde symbol (~) indicates an observed result that is identical to the expected result

	Expected T/observed T/deviation			Expected E/observed E/deviation			Expected path/observed path/deviation		
Test case	Bundle protocol	Distance Dijkstra	Quality Dijkstra	Bundle protocol	Distance Dijkstra	Quality Dijkstra	Bundle Protocol	Distance Dijkstra	Quality Dijkstra
1	3.7e−7/~/0	1.1e−7/~/0	1.3e−7/~/0	45/~/0	12/~/0	23/~/0	1,3,11,12/~/N	1,6,12,/~/	1,9,12/~/
2	2.8e−3/~/0	1.3e−3/~/0	2.4e−3/~/0	23/~/0	48/~/0	29/~/0	11,6,10,8,3,12/~/N	1,8,12/~/	10,6,12/~/N
3	1.4e−3/~/0	1.1e−3/~/0	1.4e−3/~/0	7/~/0	47/~/0	3/~/0	2,6,3,1/~/N	1,5,3,12/~/N	1,5,3,12/~/N
4	1.8e−3/~/0	1.4e−4/~/0	1.8e−3/~/0	9/~/0	57/~/0	4/~/0	1,11,3,12/~/N	1,3,12/~/	1,10,2,12/~/N
5	1.5e−3/~/0	1.2e−3/~/0	1.5e−3/~/0	19/~/0	22/~/0	19/~/0	1,2,12/~/N	1,6,12/~/N	1,3,12/~/N
6	1.7e−4/~/0	1.1e−4/~/0	1.7e−4/~/0	62/~/0	53/~/0	8/~/0	1,4,8,6,12/~/N	1,9,12/~/	1,11,12/~/ N
7	3.6e−7/~/0	1.2e−7/~/0	1.4e−7/~/0	12/~/0	51/~/0	3/~/0	1,8,3,10,4,12/~/N	1,11,12/~/ N	1,2.12/~/
8	2.1e−3/~/0	1.7e−3/~/0	1.9e−3/~/0	56/~/0	34/~/0	34/~/0	1,3,11,2,12/~/N	1,5,12/~/	1,5,12/~/N

Table 6 Some of results obtained after black-box testing the stochastic components of the DTN simulation, showing T (transmission time) and E (percent error). N represents no observed deviation; the tilde symbol (~) indicates an observed result that is identical to the expected result

	Expected T/observed T/deviation			Expected E/observed E/deviation			Expected path/observed path/deviation		
Test case	Bundle protocol	Distance Dijkstra	Quality Dijkstra	Bundle protocol	Distance Dijkstra	Quality Dijkstra	Bundle protocol	Distance Dijkstra	Quality Dijkstra
1	3.8e−07/~/1	1.3e−07/~/2	1.4e−07/~/3	34/~/2	12/~/3	28/~/2	1,4,10,12/~/N	1,5,12,/~/N	1,10,12/~/N
2	2.8e−03/~/1	1.2e−03/~/1	2.1e−03/~/1	23/~/3	48/~/2	29/~/2	10,6,2,3,12/~/N	1,7,12/~/N	10,4,12/~/N
3	1.9e−03/~/1	1.0e−03/~/2	1.3e−03/~/1	6/~/2	51/~/1	2/~/0	3,6,2,1/~/N	1,4,11,12/~/N	1,10,7,12/~/N
4	5.5e−04/~/1	9.6e−05/~/1	3.1e−04/~/1	59/~/3	84/~/4	36/~/2	1,10,2,12/~/N	1,9,12/~/N	1,3,12/~/N
5	1.6e−03/~/1	1.3e−03/~/1	1.7e−03/~/2	19/~/2	22/~/0	19/~/1	1,3,9,11,8,12/~/N	1,11,12/~/N	1,4,12/~/N
6	1.6e−04/~/1	1.1e−04/~/1	1.6e−04/~/1	62/~/4	53/~/4	8/~/2	1,3,9,11,8,4,2,7,12/~/N	1,7,12/~/N	1,4,12/~/N
7	3.7e−07/~/1	1.0e−07/~/2	1.3e−07/~/1	12/~/3	51/~/2	3/~/3	1,3,11,2,7,12/~/N	1,3,2 12/~/N	1,2,8,0.12/~/N
8	2.8e−03/~/1	1.2e−03/~/1	1.7e−03/~/1	56/~/4	34/~/4	34/~/3	1,11,3,,9,8,12/~/N	1,2,12/~/N	1,2,12/~/N
9	5.5e−04/~/2	9.6e−05/~/2	3.0e−04/~/2	56.6/~/1	80/~/4	38/~/4	1,10,2,12/~/N	1,9,12/~/N	1,3,12/~/N
10	5.2e−4/5.5e−4/12	4.8e−5/4.5e−5/17	1.6e−04/2.1e−4/21	56/40/16	84/90/14	40/36/4	1,3,11,2,12/~/N	1,7,12/~/N	1,4,12/~/N

7 Analysis

The results for a single run of the Monte Carlo simulation are summarized in Table 7. As seen from the results in the table, bundle protocol performed moderately well in terms of percent error but was the worst in terms of transmission time, while distance-based Dijkstra performed the best in terms of transmission time (with minimal variation as well) despite being the worst in terms of percent error. Quality-based Dijkstra, however, outpaced both the other two design options in terms of percent error and was better than bundle protocol, though not distance-based Dijkstra, in terms of transmission time.

Figures 8, 9 and 10 show the most frequent route taken by packets following bundle protocol, distance-based Dijkstra, and signal quality-based Dijkstra respectively for a given network. As is evident from the graphics, the MATLAB implementations of all three protocols behaved in the manner expected, which provides confidence that they were implemented correctly.

Since these are results for a single network, however, they do not reflect the variability in performance seen for each of the design options across different network configurations. Table 8 shows the performance of each of the three design options across five separate runs of the Monte Carlo simulation (and five different networks). The performance of the three routing protocols relative to each other remains consistent, with signal quality-based Dijkstra showing the best results for percent error, distance-based Dijkstra performing the best in transmission speed, and bundle protocol showing moderate results for both data integrity and the worst transmission time.

Notably, distance-based Dijkstra still performs worse than the current state-of-the-art methodology, bundle protocol, in terms of percent error. The other alternative protocol being evaluated, signal quality-based Dijkstra, performs better than bundle protocol across both metrics.

Table 9 shows the results of the t-test, confirming that all of the differences observed in Table 8 between the different design options are in fact statistically significant (i.e. that choosing one option over another would result in a substantial difference in the metric) for both percent error and transmission time. Thus, performing a MAVF analysis is reasonable since it has been established that the choice of design option does have a statistically significant impact on the metrics of interest.

Table 7 Summary of the results from a single run of the Monte Carlo simulation, showing the mean, standard deviation, and standard error of the mean for the two metrics of interest

Method	Percent error (%)	Transmission time (h)		
	Mean	Mean	Standard deviation	Standard error
Bundle protocol	57.6	2.703	0.994	0.0444
Distance-based Dijkstra	77.8	1.1882	0.000129	5.759e−06
Quality-based Dijkstra	36.4	1.926	0.970	0.0434

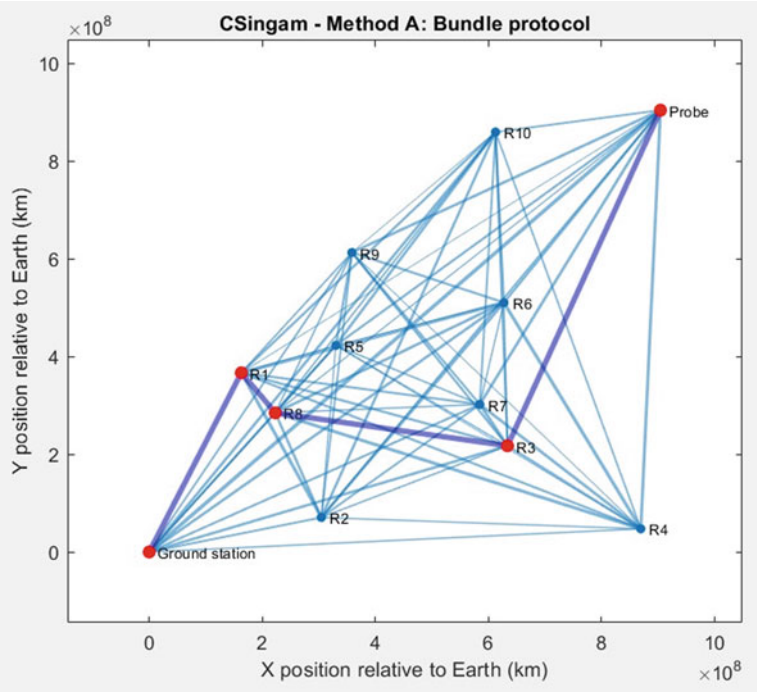

Fig. 8 Most frequent path taken by packets using bundle protocol for the network associated with the results in Table 2

Based on the results seen in Tables 7 and 8, and taking the bundle protocol results as a benchmark for performance (what with it being the current preferred routing protocol for DTN contexts), it is almost unnecessary to perform a MAVF analysis since signal quality-based Dijkstra outperforms bundle protocol across both metrics whereas the other methodology, distance-based Dijkstra, only outperforms bundle protocol in one metric (transmit time) and in fact performs significantly (as shown in Table 9) worse than baseline with regards to percent error.

Nonetheless, the MAVF rankings are provided in Table 10 to provide clear, unequivocal rankings of each design option.

The MAVF results show quality-based Dijkstra to be the best option, with a MAVF value that is over three times higher than the 2nd best option (bundle protocol). Distance-based Dijkstra ranks the worst despite having the best value for transmit time.

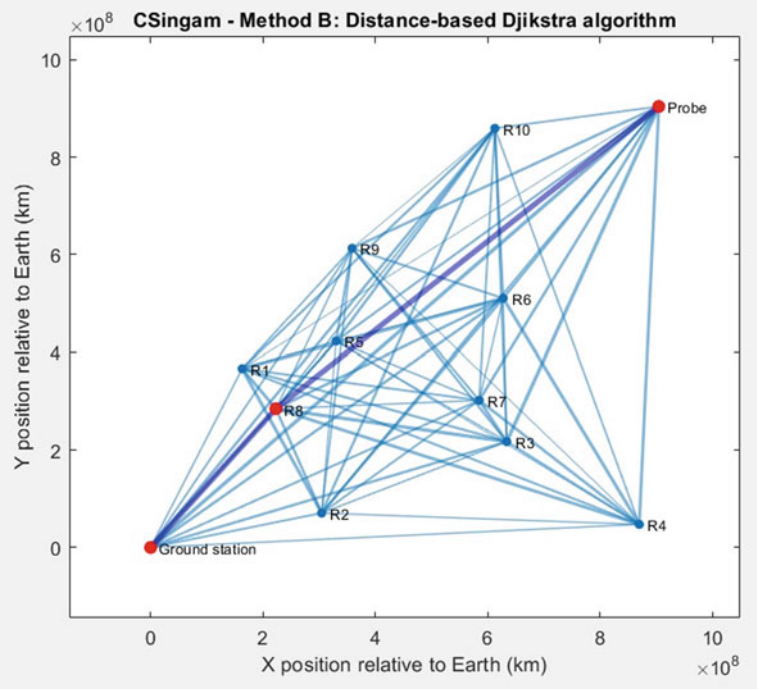

Fig. 9 Most frequent path taken by packets using distance-based Dijkstra for the network associated with the Table 2 results

8 Recommendations

Since the MAVF results are meant to serve as a means of quantifying the relative practical value of the design options, it is also worth noting the methodologies which are not worth implementing (i.e. that performed worse than the baseline). Distance-based Dijkstra performed worse than the baseline in a mission-critical metric (percent error), effectively eliminating any value the protocol's short transmission time might have had—after all, a rapid transmission has little utility if it risks compromising mission success significantly more than is considered standard.

Thus, it would be more accurate to assign an effective value of 0 to for the metric of transmission time for the distance-based Dijkstra methodology; this would yield the corrected MAVF table seen in Table 11.

The recommended course of action is to use signal-based Dijkstra as the routing protocol for space-based DTN applications. In addition to being the best option to minimize transmission error, it also performs moderately well in terms of transmission time and outperforms the current standard for routing protocols, bundle protocol, across both metrics. The Welch's t-test results indicated that switching a network over from using either of the two other design options to using signal quality-based Dijkstra would yield a statistically significant change in metrics, most notably an

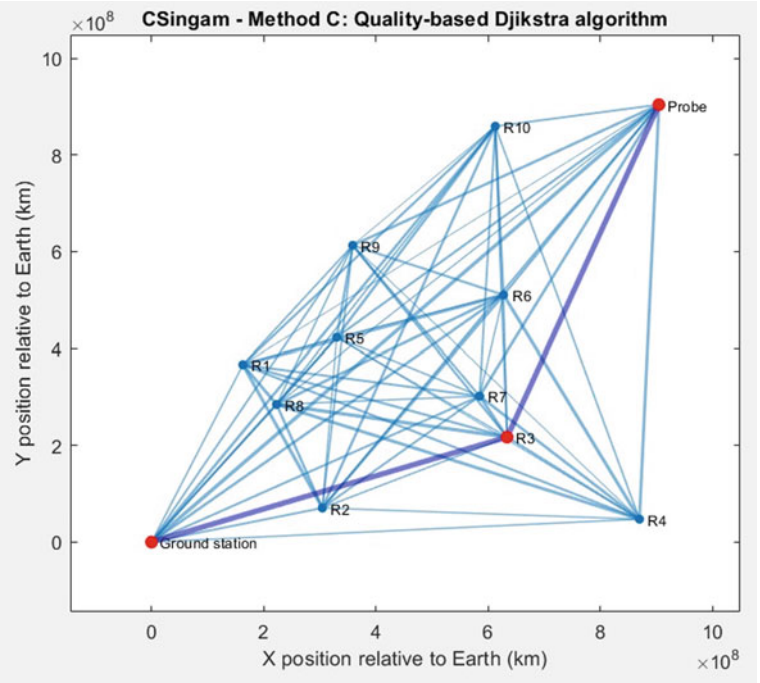

Fig. 10 Most frequent path taken by packets using quality-based Dijkstra for the network associated with the Table 2 results

Table 8 Summary of the results from five runs of the Monte Carlo simulation, showing the mean, standard deviation, and standard error of the mean for the two metrics of interest

Method	Percent error (%)			Transmission time (h)		
	Mean	Standard deviation	Standard error	Mean	Standard deviation	Standard error
Bundle protocol	56.440	7.410	3.314	3.120	0.684	0.306
Distance-based Dijkstra	64.200	4.864	2.175	1.189	0.004	0.002
Quality-based Dijkstra	41.040	4.498	2.011	1.820	0.280	0.125

increase in data integrity on receipt (the more critical of the two parameters). Given that routing protocols are implemented in networks via software, and that software updates can be readily pushed remotely to satellites (by virtue of being communication instruments in and of themselves), it is feasible to implement a change in routing protocols without any hardware modifications to the system.

Table 9 Welch's t-test results comparing each of the design options with each of the other design options for both percent error and transmission time. Green cells indicate that the calculated *p*-value meets the standard for statistical significance ($p < 0.05$)

Welch t-test results: percent error				Welch t-test results: transmission time			
Vs	Bundle protocol	Distance-based Dijkstra	Quality-based Dijkstra	Vs	Bundle protocol	Distance-based Dijkstra	Quality-based Dijkstra
Bundle protocol	–	0.0914	0.0102	Bundle protocol	–	0.00170	0.00500
Distance-based Dijkstra	0.0914	–	8.758E−05	Distance-based Dijkstra	0.00170	–	0.00437
Quality-based Dijkstra	0.0102	8.758E-05	–	Quality-based Dijkstra	0.00500	0.00437	–

Table 10 MAVF results for each of the three design options based on the metric means from the 5-run dataset

MAVF analysis			
Design option	Value based on percent error	Value based on transmit time	MAVF
Bundle protocol	0.33506	0	0.279217
Distance Dijkstra	0	1	0.166667
Quality Dijkstra	1	0.672774	0.945462

Table 11 The MAVF analysis results, with the values corrected for practicality

MAVF analysis			
Design option	Value based on percent error	Value based on transmit time	MAVF
Bundle protocol	0.33506	0	0.279217
Distance Dijkstra	0	0	0
Quality Dijkstra	1	0.672774	0.945462

9 Conclusions

The area of signal routing for disruption-tolerant networks is an area of active research and burgeoning interest, particularly with the varied applications for both DTNs in general and optimized signal routing approaches in particular. Given the ubiquity of communication systems in globalized society and the inevitability of unforeseen circumstances, absent a surprise confirmation of the existence of Laplace's demon, it is to be expected that existing interest in transferring information efficiently in adverse environments and variable conditions will continue into the foreseeable future as well.

The proposed signal quality-based Dijkstra routing method has been shown to be a methodology that performs well across both DTNs and standard-condition (continuous connectivity) scenarios, making it an ideal candidate for implementation across a variety of applications. As a methodology that has been shown to surpass the performance of existing standards, and that demonstrates benefits in terms of key metrics, security against DoS attacks, and from a network architecture standpoint—including improving network security posture, minimizing individual user risk, and increasing resiliency against sub-optimal traffic scenarios—it offers great promise for use with future DTN implementations. It is anticipated that the methodology can be refined further in future work so as to provide even greater flexibility and performance across any number of different DTN-relevant contexts, as well as greater security against hostile actor takeover.

References

1. McMahon A, Farrell S (2009) Delay-and disruption-tolerant networking. IEEE Internet Comput 13(6):82–87
2. Gertz M, Csaba G (2002) Monitoring mission critical data for integrity and availability. In: Working conference on integrity and internal control in information systems. Springer, Boston, MA, pp 189–201
3. Caini C, Cruickshank H, Farrell S, Marchese M (2011) Delay-and disruption-tolerant networking (DTN): an alternative solution for future satellite networking applications. Proc IEEE 99(11):1980–1997
4. Matson DL, Spilker LJ, Lebreton JP (2003) The Cassini/Huygens mission to the Saturnian system. In: The Cassini-Huygens mission. Springer, Dordrecht, pp 1–58
5. CCSDS Secretariat (2010) Rationale, scenarios, and requirements for DTN in space. CCSDS, Washington, D.C.
6. Ibid
7. NASA (2009) NASA's mission operations and communications services. AO NNH09ZDA007O, January 2009. https://deepspace.jpl.nasa.gov/files/6_NASA_MOCS_2014_10_01_14.pdf
8. Dyer JS, Sarin RK (1979) Measurable multiattribute value functions. Oper Res 27(4):810–822
9. Parnell GS, Trainor TE (2009) 2.3. 1 using the swing weight matrix to weight multiple objectives. INCOSE Int Symp 19(1):283–298
10. NCSS. Two sample T-tests allowing unequal variance. https://ncss-wpengine.netdna-ssl.com/wp-content/themes/ncss/pdf/Procedures/PASS/Two-Sample_T-Tests_Allowing_Unequal_Variance.pdf
11. MathWorks. "digraph: Graph with directed edges" (n.d.). https://www.mathworks.com/help/matlab/ref/digraph.html
12. Ulybyshev Y (1998) Long-term formation keeping of satellite constellation using linear-quadratic controller. J Guid Control Dyn 21(1):109–115
13. Zhou J, Hu Q, Friswell MI (2013) Decentralized finite time attitude synchronization control of satellite formation flying. J Guid Control Dyn 36(1):185–195
14. Chang HS, Kim BW, Lee CG, Min SL, Choi Y, Yang HS, Kim CS et al (1998) FSA-based link assignment and routing in low-earth orbit satellite networks. IEEE Trans Veh Technol 47(3):1037–1048
15. Casio. "Keisan Calculator" (n.d.) https://keisan.casio.com/exec/system/1180573226
16. Paxton P, Curran PJ, Bollen KA, Kirby J, Chen F (2001) Monte Carlo experiments: design and implementation. Struct Equ Model 8(2):287–312
17. Buckley J, James I (1979) Linear regression with censored data. Biometrika 66(3):429–436