

Review

DTN and Non-DTN Routing Protocols for Inter-CubeSat Communications: A comprehensive survey

Mohamed Atef Ali Madni ^{1,*}, Saeid Iranmanesh ^{2,*} and Raad Raad ^{1,*} 

¹ School of Electrical Computer and Telecommunication Engineering, University of Wollongong, Wollongong, 2522 NSW, Australia

² School of IT and Engineering, Melbourne Institute of Technology, Sydney, 2000 NSW, Australia

* Correspondence: maarm430@uowmail.edu.au (M.A.A.M.); siranmanesh@mit.edu.au (S.I.); raad@uow.edu.au (R.R.); Tel.: +61434191775 (M.A.A.M.); +61423328275 (S.I.); +6142214761 (R.R.)

Received: 19 November 2019; Accepted: 11 March 2020; Published: 14 March 2020



Abstract: CubeSats, which are limited by size and mass, have limited functionality. These miniaturised satellites suffer from a low power budget, short radio range, low transmission speeds, and limited data storage capacity. Regardless of these limitations, CubeSats have been deployed to carry out many research missions, such as gravity mapping and the tracking of forest fires. One method of increasing their functionality and reducing their limitations is to form CubeSat networks, or swarms, where many CubeSats work together to carry out a mission. Nevertheless, the network might have intermittent connectivity and, accordingly, data communication becomes challenging in such a disjointed network where there is no contemporaneous path between source and destination due to satellites' mobility pattern and given the limitations of range. In this survey, various inter-satellite routing protocols that are Delay Tolerant (DTN) and Non Delay Tolerant (Non-DTN) are considered. DTN routing protocols are considered for the scenarios where the network is disjointed with no contemporaneous path between a source and a destination. We qualitatively compare all of the above routing protocols to highlight the positive and negative points under different network constraints. We conclude that the performance of routing protocols used in aerospace communications is highly dependent on the evolving topology of the network over time. Additionally, the Non-DTN routing protocols will work efficiently if the network is dense enough to establish reliable links between CubeSats. Emphasis is also given to network capacity in terms of how buffer, energy, bandwidth, and contact duration influence the performance of DTN routing protocols, where, for example, flooding-based DTN protocols can provide superior performance in terms of maximizing delivery ratio and minimizing a delivery delay. However, such protocols are not suitable for CubeSat networks, as they harvest the limited resources of these tiny satellites and they are contrasted with forwarding-based DTN routing protocols, which are resource-friendly and produce minimum overheads on the cost of degraded delivery probability. From the literature, we found that quota-based DTN routing protocols can provide the necessary balance between delivery delay and overhead costs in many CubeSat missions.

Keywords: aerospace communications; satellite communications; CubeSats networks; CubeSat swarms; constellations; delay tolerant networks; ad-hoc networks; sensor networks; routing protocols; TCP/IP

1. Introduction

CubeSats are a type of miniaturised satellite designed for space research. These tiny spacecraft are limited in size ($1\text{U} = 10 \times 10 \times 10 \text{ cm}$) and mass (around 1kg for 1U). Satellites are generally classified

on the basis of their mass into the following five categories [1]: (i) Mini-satellites (weight: 100–500 Kg), (ii) Micro-satellites (weight: 10–100 Kg), (iii) Nano-satellites (weight: 1–10 Kg), (iv) Pico-satellites (weight: 0.1–1 Kg), and (v) Femto-satellites (weight: <0.1 Kg). CubeSats fall into the Nano-satellites category. Additionally, in terms of size, CubeSats can be structured in different sizes, 1U–12U, which are the multiplication of 1U, as illustrated in Figure 1.

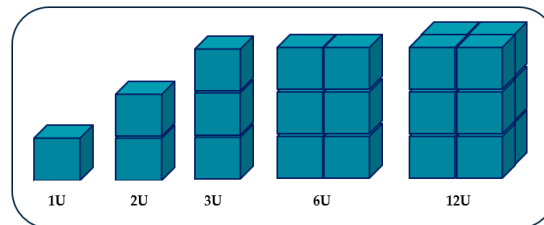


Figure 1. Standard CubeSat Sizes (1U–12U).

In comparison to conventional satellites that cost \$0.1–\$2B, the expenditure for constructing a CubeSat is only \$20–\$200 K and the launching cost of a CubeSat can be as low as \$12,500 [1]. More importantly, new technologies can be tested on CubeSats before using them on more expensive satellites [2]. Hence, it makes economic sense to introduce CubeSats for deployment in Low Earth Orbit (LEO) and in deep space to perform different missions [3]. Over 2200 NanoSats and CubeSats were launched by 2018 and over 3000 NanoSats will be launched in the next six years, according to the Nanosatellites Database [4]. For instance, Planet Labs launched and operated a series of 3U CubeSat constellations for Earth observation, known as Flock missions, with a total number of about 250 CubeSats from 2014 to 2017. Another notable example can be seen in the series launches of Lemur-2 NanoSats, where the mission consisted of 73 3U CubeSats that were developed by Spire Global to serve as a part of their remote sensing commercial satellite constellation that offers global ship tracking and weather observation [5]. Indeed, the largest batch that was launched into space on a single rocket was Flock-3p, which consisted of 88 CubeSats [6]. Figure 2 shows the current and future NanoSats launch statistics.

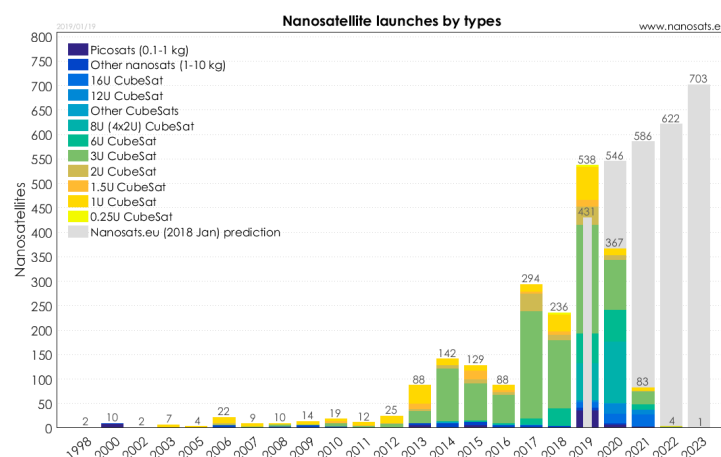


Figure 2. Nano-satellite launch by size statistics [4].

The advancement of miniaturised space technology and the capability of multi-satellite missions have allowed for the establishment of enormous Nano-satellite networks. These satellites can then communicate with each other without human intervention by means of Inter-Satellite Links (ISL), where the ISL provides the essential capabilities of real-time or quasi-real-time communication. Deploying low-cost swarms or constellations of Nano-satellites with ISL opportunities can simultaneously improve the spatial and temporal data collection from different parts of space [7].

CubeSat networks can be deployed in different formations/topologies according to mission objectives. In reference [8], Saptarshi et al. categorised 39 CubeSat missions, according to the mission objectives, satellite number, and type of formation flying. The topology of small satellites can be mostly divided into three types, namely: leader-follower; cluster; and, constellation. In leader-follower missions, all satellites are deployed on the same orbit, are separated by a specific distance and all follow the leader. A cluster mission is when a group of satellites is deployed close to each other on different orbits to cover an appointed area on the Earth, while a constellation mission includes a group of satellites that are distributed in different orbits to provide full coverage of the Earth [1].

In said topologies, the densities of CubeSats and mobility pattern play an important role in forming the connectivity of the network. When the density of CubeSats is high in the network, the probability of discovering a route from a source to a destination increases. In such cases, Non Delay Tolerant (Non-DTN) routing protocols effectively increase the network performance in terms of high delivery ratio, low delay, and low overhead. However, when the density of CubeSats in the network decreases and/or the topology of the network is highly dynamic, the network experiences frequent disconnectivity, which will cause the Non-DTN protocols to not be efficient anymore. This is because the data packets will be dropped when a route is not discovered. Thus, Delay Tolerant (DTN) protocols as a new approach were introduced to store, carry, and forward the data packets when a route is not discovered. In this survey, we consider both DTN and Non-DTN protocols, as CubeSats' missions may create a connected or disjointed network. The relevant space missions DTN and Non-DTN protocols are classified, as depicted in Figure 3. The CubeSat networks create a different network topology based on the intended mission. Both types of protocols are considered where the network is connected or disjointed, respectively. Hence, all of the protocols discussed in this survey can be used in satellite networks, depending on the network resources and conditions. The Non-DTN based routing protocols are divided into two categories that are based on the satellite network architecture, namely: (i) single-layer routing protocols where a certain number of satellites are distributed on one standard orbit, for example, LEO; and, (ii) multi-layer routing protocols, where the satellites are distributed over two or three standard orbits, such as LEO/MEO/GEO. The said protocols can be efficient if the missions create a connected network topology. However, in many missions, the network might become disjointed, which requires the store-carry-forward manner as a route towards a destination may not be contemporaneously discovered. Thus, the DTN protocols can be used. The DTN routing protocols are categorised into three groups, namely: (i) flooding-based protocols, where the number of packet's replicas is dependent on the number of encountered node, (ii) quota-based protocols, where the number of packet's replicas is fixed, and (iii) forwarding-based protocols, where there is only a single copy of the packet throughout the network.

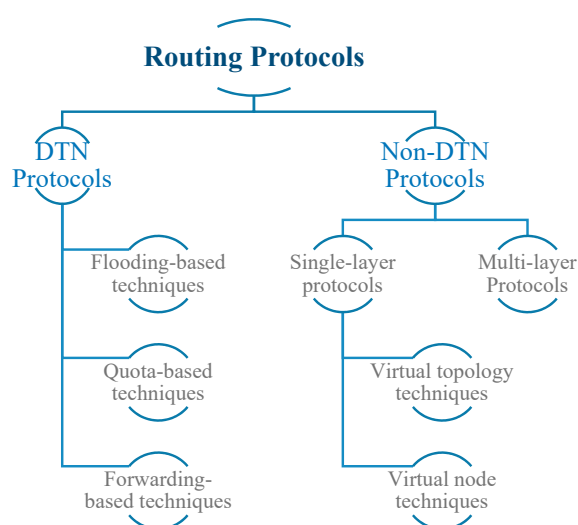


Figure 3. Delay Tolerant (DTN) and Non Delay Tolerant (Non-DTN) routing protocols categories.

A wide-range of routing protocols in CubeSats communications have been studied in recent years due to the economic benefits and greater spatial and temporal performance. Most of the previous work on inter-CubeSat communications has focused on developing physical layer links i.e., antenna design. In Reference [9], CubeSat missions that are based on cross-link communication are surveyed. The main focus is on the physical connectivity among CubeSats, such as frequency allocation. Similarly, in Reference [10], the suitability and the challenges of intra-satellite and inter-satellite links in wireless sensor space networks are discussed. Additionally, different types of planar antennas for CubeSats communication applications are studied in [11–13]. In Reference [1], the communication capabilities of small satellites over inter-satellite links are presented from the view of the last three lower layers of the OSI. In Reference [14], some of the routing approaches for conventional satellite networking are reviewed with a focus on networking challenges in the space environment. Most recently, some features of small satellites and CubeSats are reviewed, including mission objectives, CubeSat subsystems, topology examples, and communication protocols [15]. In Reference [16], energy-aware routing protocols are proposed for inter-CubeSat swarms communication. However, there was no sufficient focus on the different routing protocols that can be used for CubeSat networking.

From the above, prior work has not comprehensively investigated the suitability and applicability of both Non-DTN and DTN routing protocols for the different topologies of CubeSat networks. Hence, this paper discusses and analyses the suitability and scalability of using the Non-DTN based single-layer and multi-layer routing, as well as DTN routing protocols in CubeSat networks. This survey provides a comprehensive review of the state of the art space routing protocols under DTN and Non-DTN architectures. The review qualitatively compared different TCP/IP single-layer and multi-layer routing protocols, as well as DTN routing protocols. The applicability of the compared protocols is discussed for both connected and disjointed CubeSat networks in order to highlight the cases where DTN and Non-DTN protocols can be used. Most of the other CubeSat communication surveys focus on the CubeSats physical layer capabilities to improve the performance of the network; however, this survey investigates routing issues in the network layer level. The performance of the abovementioned routing protocols in different CubeSat network settings is analysed and discussed accordingly.

Section 2 of this paper describes the existing challenges in CubeSats communications and, in Section 3, particular focus is given to Non-DTN based single-layer and multi-layer routing protocols by providing a comprehensive review. In Section 4, we compare the existing DTN routing protocols and highlight the drawbacks and advantages. Lastly, we conclude our survey by raising open research questions from the existing gaps within the current state of the art in Section 5.

2. Challenges in CubeSat Communications

This section discusses the challenges that are raised by the characteristics of CubeSat communication where network resources are limited and the topology of the network is highly dynamic. While the intermittent connectivity is considered to be the main challenge in routing data across CubeSats, some independent challenges regarding communication in space, such as inter-CubeSat links, up/downlinks with the ground station, natural drag, and energy, will also be highlighted. These challenges pose some communication issues that will eventually lead to more intermittent connectivity and poor routing performance.

As discussed earlier, the very small size of the CubeSats enables them to be fit for many missions and be more cost-effective when compared to other satellite options. Unfortunately, the small size does not allow for them to be equipped with a large capacity of resources. The size of the CubeSat antenna, for example, is bound by the required gain and operating frequency, as well as the size of the CubeSat could limit transmission range on some missions, leading to a network with intermittent connectivity. This is more critical when the topology of the network is highly dynamic, because CubeSats, on LEO, are orbiting the Earth with a high speed that causes the contact duration between them to be short. For example, the link between the two will last for about 1.3 s, assuming a negligible channel discovery time, and such a short contact duration will influence the network performance when the data rate is

low, where two CubeSats are traveling in crossing directions at a velocity of 7.8 km/s with a radio range of 5 km. Additionally, as LEO CubeSats can be deployed on different altitudes 200–1200 km above the ground, each satellite can have a different inclination in degrees from 0° – 180° that will specify its trajectory. As such, the encounter period may last longer than if they were orbiting on different orbit inclination if two satellites are orbiting on different altitudes but the same orbit inclination, even if they were on the same altitude. Thus, one important factor that affects the link availability between any pair of CubeSats is its relative motion.

The orbital period of LEO satellites is an additional source of intermittent connectivity in CubeSat networks that results in a short communication period between a CubeSat and the corresponding ground station. This intermittent connectivity comes from a satellite orbital period of about 90–110 min at speed of about 7.8 km/s with an encounter window of about 5–15 min to the ground station during every cycle [17].

The energy budget that was allocated to the CubeSats is another problem that worsens the intermittent connectivity. The limited size and structure of CubeSats do not allow for many solar panels to be fitted, especially with the available surface area being shared with some other instruments, such as antennas and sensors. With the maximum energy producible from the solar panels thus limited, only when the solar cells are pointing to the sun are able to charge the batteries. Indeed, due to the inability to load many CubeSats with altitude determination and stabilization control devices to ensure that the solar cells are pointing towards the sun, the solar cells may provide energy that is even more limited during the limited encounter times with the sun. Consequently, this causes some CubeSats to fail to continue on their mission, because they run out of power quickly and, as a consequence, the failure of one CubeSat might lead to disruption and discontinuity in multiple links. Moreover, the ground station links and inter-CubeSat communication is considered to be a significant source of power consumption. The estimated communication power budget of a CP1 CubeSat mission, for example, is 267 mW, which is a 36.77% of the available produced power from the solar cell [18]. To date, many energy-efficient routing protocol have been proposed to tackle the said issue, which will be discussed in the next chapter.

Where LEO satellites are usually deployed on orbits of 200–1200 km, they are additionally associated with a natural drag that increases when the satellites are orbiting closer to the Earth due to atmospheric drag forces, such as thermospheric heating and cooling, as well as the gravitational forces. This will force CubeSats to change their trajectory over time, such that the meeting time with other CubeSats will also change, which will require an accurate Attitude Control Subsystem (ACS) to maintain both the orbit and the direction of the antenna on the CubeSat body. Even though different ACS techniques were proposed for CubeSat missions, such as low control accuracy active and passive magnetic ACSs, as well as reaction wheels and torque coils [19], the limited resource capability of the CubeSats restrict the application of advanced stability control devices, especially on 1U–3U CubeSats that are commonly used in the large CubeSat constellations, for correcting the orbital drag and stabilizing the rotation of CubeSat. Thus, communication might be impossible because of the trajectory change and the rotation of CubeSats can change the antenna pointing direction, leading to link disruption and data loss, even when two CubeSats are in close proximity.

Beyond the entire abovementioned reasons making routing in CubeSat networks extremely challenging, yet another issue that might cause link disruption in CubeSat networks is the failure of the CubeSat itself. CubeSats are characterized by high failure rates, as most of them are made from COTS components with the impossibility of recovery in the case of failure. The failure of one CubeSat might result in the failure of multiple routing paths, which leads to data dropping and the necessity for routing table recalculations. Table 1 summarizes the routing challenges and implications that CubeSat networks may encounter in LEO, including space communication challenges and the other constraints, due to the CubeSat structure limitations.

Table 1. Operating conditions of CubeSats on Low Earth Orbit (LEO), challenges, and implications.

Challenges	Implications
Intermittent connectivity	<ul style="list-style-type: none"> - Satellites on this orbit are characterized by scheduled predictable/semi-predictable intermittent connectivity, whether for a satellite to ground links or inter-satellite links. - There are no contemporary paths present for satellite and ground station communication or cross-link communication.
Orbital period	<ul style="list-style-type: none"> - LEO satellite orbital velocity ~ 7800 m/s, based on the satellite altitude orbital period of about 90–110 min for 160–1200 km altitudes respectively. - Limited encounter time between satellites which in turns bounds data transfer rate.
Inter-CubeSat links	<ul style="list-style-type: none"> - Transmission range between two satellites, approximately 5–200 km. - The transmission range of inter-CubeSats is bound by cross-link antenna transmission power. - Limited antenna size and capability compared with the conventional satellites. - Limited antenna coverage compared with the conventional satellites.
Up/Downlinks with the ground station	<ul style="list-style-type: none"> - Transmission range between satellite and ground station, approximately 200–1200 km - The transmission range of CubeSats is bounded by the downlink antenna transmit power. - Satellite revisit time Limited antenna size and capability
Altitude and inclination ranges	<ul style="list-style-type: none"> - Orbit altitude rang is 200–1200 km above the Earth and orbit inclination ranges 0°–180°.
Natural drag	<ul style="list-style-type: none"> - Common de-orbiting behaviour leads to changes in orbital height and hence meeting time between CubeSats will also change over time. - Orbiting at lower altitudes increases the drag process. - The drag upsurges with increasing solar activity (sunspots).
High failure rate	<ul style="list-style-type: none"> - Space radiation effects on electronic components, particularly Commercial-off-the Shelf (COTS) components. - Impossibility of recovery under failure.
Energy	<ul style="list-style-type: none"> - Solar cells limited space available on the small size of the CubeSat body. - Small storage batteries. - High power consumption of up/downlinks and cross-links.
Topology density	<ul style="list-style-type: none"> - Satellite dissemination and encounter times.
CubeSat stability on orbit	<ul style="list-style-type: none"> - There is no space on the CubeSats for advanced stability control devices. - Antenna directionality and steering ability.
Data rate	<ul style="list-style-type: none"> - A single CubeSat has limited data rate - CubeSat swarms and constellations can provide a higher overall system data rate, however, networking CubeSats in these systems is challenging and requires advanced routing protocols.

Advantageously, networks of CubeSats serve to provide a wider communication window and coverage, as well as gaining better data rates from the whole network. However, the networks that are usually formed as swarms and constellations require reliable routing protocols to define one or more reliable paths for its users, according to the link metric and QoS requirements. The integrated satellite network structure is consisting of a space segment and the terrestrial network, which creates three main types of routing segments, which are: space segment routing; access routing; and, boundary routing [14]. Of these segments, space segment routing includes data routing between satellites over one or more reliable paths to satisfy system requirements, while access routing controls satellite access to a ground station or a mobile ground user. The election of the access satellite is based on link efficiency regarding elements, such as signal strength and delay. However, the boundary routing is responsible for enhancing the interoperability between the space segment and the terrestrial network, where communication capability mismatch is present. The capability mismatch at the borders between the satellite and ground networks requires a resilient routing protocol to provide reliable communication. Thus, highly sophisticated routing techniques are essential for coping with these routing challenges.

3. Routing Protocols in Connected CubeSat network

Given a connected CubeSat network, the routing protocols can be categorised based on the network topology over one or more standard satellite orbits into two groups, namely: (i) single-layer routing protocols, where the nodes are clustered and communicate within the same level of orbit [20–26]; and, (ii) multi-layer routing protocols where nodes communicate with other nodes from another level [27–34]. These protocols, which were originally proposed for Mobile Satellite Services (MSS) and Satellite Personal Communication Networks (S-PCN), include constellations of conventional satellites to provide voice and data mobile satellite services, such as Iridium and Inmarsat. Protocols that were proposed for these two categories are based on TCP/IP technology and will be comprehensively discussed in the following section with an analysis of each routing protocol.

3.1. Single-Layer Routing for Satellite Networks

Satellites in the single-layer networks are usually disseminated on one orbit at different altitudes. Most of the small satellite swarms and constellations are deployed into the LEO layer that is characterized by high-speed satellites, a short round-trip time, and short contact times between satellites and ground stations. In addition to these characteristics, the Nano-satellites have limited resources, such as energy, low antenna performance, and limited storage.

Single-layer routing protocols are categorised based on two strategies, namely: (i) virtual topology; and, (ii) virtual node. The virtual topology-based routing protocols are dependent on the principle of a ‘space-time-graph’, where the satellite network can be considered over a certain period with this period and then divided into time slots, each of which will represent a static snapshot of the network topology at that time slot. This provides a complete picture of the network with a large number of two-dimensional (2D) network topology snapshots. Thus, defining paths between any pair of nodes will be determined over time while using the current static snapshot topology knowledge as well as the future consecutive static topologies. However, in general, there is no great difference in the topology of connectivity of two consecutive snapshots. The second strategy, virtual node-based routing protocols, considers the connectionless datagram delivery [35]. The movement of a satellite over its orbit is represented by filling the current logical position with the information of the next satellite traveling on the same orbit where it is assumed that the earth is covered with a grid that indicates the logical location of each satellite (node) according to the satellite longitude and latitude. This strategy is based on removing the dynamic component of the satellite network. Hence, the network is represented as a fixed topology that considers the logical locations, as hops and any routing decisions will be made accordingly. Table 2 summarizes the currently proposed techniques for both virtual topology and virtual node of single-layer routing.

Table 2. Single-layer routing algorithms.

Node and Topology-Based Algorithms	Network Type	Topology Information	Decision Criteria	Resources Consumption	Links Recovery
dv-dvtr [20]	Connection-oriented	Predictable /off-line	Link metric	High	No
fsa [21]	Connection-oriented	Predictable /off-line	Link metric	High	No
cemr [22]	Connection-oriented	Predictable /off-line	Link metric	High	No
elb [23]	Connection-oriented	Predictable/dynamic	Link metric /congestion	High	Yes
cal-lsn [24]	Connection-oriented	Predictable/dynamic	Link metric	High	Yes
par [25]	Connectionless	Predictable/dynamic/ Mesh-grid	Historical Info/Link metric	High	Yes
epar [25]	Connectionless	Predictable/dynamic/ Mesh-grid	Historical Info/Link metric	High	Yes
ddra [26]	Connectionless	Predictable/dynamic/ Mesh-grid	Link metric	High	Yes
dra [35,36]	Connectionless	Predictable/logical locations/ Mesh-grid	Node info	Low	No
laor [37]	Connectionless	Predictable/dynamic/ Mesh-grid	Node info	High	No
dodr [38]	Connectionless	Predictable/dynamic/ Mesh-grid	Node info	Medium	Yes
lcra [39] and lcpr [40]	Connectionless	Predictable/logical locations/ Mesh-grid	Node info	Low	No
lrs [41]	Connection oriented	Predictable/dynamic/ Mesh-grid/MSN	Node info/zone info	Medium	No
slsr [42]	Connectionless	Predictable/dynamic/ Mesh-grid/MSN	Node/topology info	Medium	Yes

Discrete Time-Dynamic Virtual Topology Routing (DT-DVTR) is an ATM-based connection-oriented routing scheme for LEO Satellite Personal Communication Networks (S-PCN) that confirms the feasibility of ATM application in LEO [20]. DT-DVTR is a baseline routing technique that uses a ‘space-time-graph’ for modelling the network connectivity, where each time interval represents a static network topology at that time. Correspondingly, all of the available paths can be discovered over the consecutive topology snapshots. Eventually, DT-DVTR runs the Dijkstra shortest path algorithm to find all of the least-cost paths. However, it is assumed that satellites always have periodic movement and, hence, the acquired topology snapshots from one cycle do not change in the next cycle. Although this assumption could be realistic for S-PCNs, the network topology is not always fixed. This is because of the CubeSats restricted resources, meaning that, where a CubeSat fails to continue its mission, the network topologies will be changed over time, especially as a CubeSat cannot maintain its trajectory due to the neutral drag and the lack of altitude and stability control. A Finite State Automaton (FSA) algorithm was proposed for LEO satellite mobile communication networks to address the mentioned issue of DT-DVTR. The FSA algorithm aims to take full advantage of the ISL assignment by modeling the dynamic topology as a finite state machine, where each state represents a single static topology [21]. The FSA combined the problem of routing with that of link assignment and simplified the link assignment problem in the dynamic satellite network to a group of static topology link assignment problems. Subsequently, the single state link assignment problem is solved as a mixed-integer linear programming problem. Iterative optimization methods are applied due to the NP-hardness of the mixed-integer linear programming problem. The results of the FSA algorithm suggested that the simulated annealing method outperformed the other regular link assignment methods. DT-DVTR and FSA algorithms are both connection-oriented routing protocols that consider their network topology based on the predetermined satellite motion calculations. However, neither are efficient in sudden

changes in the topology, because they do not have regular routing table updates. Compact Explicit Multipath Routing (CEMR) is a multipath routing technique for mobile satellite networks [22]. In a similar fashion to the DT-DVTR and FSA algorithms, CEMR models the dynamic topology of the satellite network as a series of discrete-time static topologies. However, CEMR introduces the concept of a compact path identifier (PathID), which includes the ISL identifiers of all hops along the path to the destination. This technique allows for all of the nodes in the path to transmit their packets based on the embedded information in the PathID. A PathID verification process is also implemented in CEMR to regularly evaluate the precision of the PathID. Thus, CEMR can provide lower overheads when compared with other previous source multipath routing or Multiprotocol Label Switching (MPLS) techniques [43]. Contrary to the traditional LEO satellite routing techniques, CEMR considers propagation delay as well as queuing delay as a routing cost metric. Besides this, the possibility of forwarding data over multiple separate paths causes CEMR to provide load balancing, minimize overall delay, and improve throughput. However, packet loss is not considered to be detected for the next hop.

An Explicit Load Balancing (ELB) is a multi-hop routing algorithm proposed to solve the load balancing issue in congested mobile satellite networks [23]. The basic idea of ELB is that adjacent satellites can exchange their congestion status to minimize the load on the congested satellite and select an alternative path with less traffic. The congested satellite broadcasts a “soon to be congested” message to its adjacent satellites in order to minimize their transmission rate through it. The adjacent satellites will respond to that by exploring alternative paths excluding the congested satellite. ELB routing is based on a queue ratio and reduction ratio metrics to provide restored data flow and avoid the congestion that results in data dropping. More recently, a load balancing routing algorithm was proposed for broadband LEO satellite networks [24]. The authors designed and implemented the Cross-layer design and Ant-colony optimization-based Load-balancing routing algorithm (CAL-LSN), which uses the physical layer information to make routing decisions. Mobile agents that are known as “ants” are used to collect link status information. Their proposed multi-objective optimization model considers the smallest bandwidth limit, the upper limit of the LEO satellite network delay tolerance, and link disruption probability.

The discrete time-dynamic virtual topology of the satellite networks typically contains multiple shortest paths between the source and destination. Accordingly, a Priority-based Adaptive Routing (PAR) load balancing routing technique was proposed for non-geosynchronous orbit satellite networks [25]. The priority technique in PAR utilizes the historical information and ISL buffering status to make the decision at each node along the path and form a complete path to the destination. The satellite network is modelled and simulated as a grid that provides multiple shortest paths to the destination. The enhanced version of PAR (ePAR) is also proposed as an alternative to minimize needless data traffic and increase link utilization efficiency. For instance, not only will the congested links be avoided, but by the use of ePAR the links with smaller queues will also be more favourable. Although the PAR uses the minimum hop count as a metric to reach the destination, it can be enhanced to consider the link length and average minimized delays. In terms of data redirection at the congested links, ePAR parameters should be adjusted to ensure a small initial data rate transmission over the alternative links; otherwise, the alternative link itself may become congested. The ePAR is analysed to show the effect of adjusting the priority metric parameters to attain optimal performance; however, this analysis assumes an unlimited buffer size and long ageing periods for prioritization. To observe the application of the ePAR on the CubeSat networks, the buffer size should be adjusted. Besides, the ageing technique that uses the information from the previous time slots should be modified to reduce the storage complexity. Similar to this PAR algorithm, the Dynamic Detection Routing Algorithm (DDRA) examines routing in virtual grid topology that divides the satellite network into small slots of time with two link status changes considered: a predictable change that defines the link alternation between two-time slots and a sudden change that reflects any link congestion or failure at a specific time slot [26]. The advantage of DDRA is that the delay is limited to the topology slot and any

sudden change is handled to avoid any delay coming from the link disruption that is caused by link congestion or failure.

Henceforth, apart from routing protocols based on “virtual topology” that are discussed above, the second category of single-layer “virtual node” routing protocols will be discussed. In satellite networks, especially the CubeSats, which are characterized by high failure rates, the absence of one satellite might lead to multiple link disruptions. Virtual node routing typically considers the network status based on the node status to update the path for better performance, as some dynamic routing techniques cannot handle the regular occurrence of link and node failures. For example, the Location-Assisted On-Demand routing (LAOR) protocol that was proposed for the IP-based LEO satellite systems [36] establishes the path discovery process based on route request and route reply among each pair of source and destination, while considering the shortest paths. Even though on-demand routing might create precise information about the network current situation, high routing overheads are expected due to the flooding of path discovery messages. The weakness of LAOR is that it does not consider the node or link failures required for path reestablishment. However, Destruction-resistant on-Demand Routing (DODR) routing has been introduced to solve this issue by using a node-based repair technique [37]. In the case of failure, the local repair technique of DODR protocol allows for a quick response from the current node right before its failure to define a new path to maintain the stream to the destination based on the information that was collected in the path discovery phase.

A Datagram Routing Algorithm (DRA) was proposed to maintain minimum propagation links in the LEO environment [35,36]. Basically, decisions at DRA are made locally at a node for each individual packet. The network is modelled as a static grid of logical locations above the Earth. Each satellite has two neighbours on the same orbital trajectory and two neighbours on nearby trajectories. The packet travels from one fixed logical location to another until it reaches its destination over the minimum propagation delay path. Routing overhead is zero in DRA due to the absence of signalling for collecting topology information. In Reference [39,40], the Low-Complexity Routing Algorithm (LCRA) and Low-Complexity Probabilistic Routing (LCPR) were both proposed for polar orbit constellations that were based on the concept of logical locations introduced by the DRA. LCRA considers both propagation delay and queuing delay as well as balancing the traffic load to avoid congestion. Likewise, LCPR allows for each node to take its next hop decision based on a minimum propagation delay and the congestion status information collected from its neighbours. The LCPR can reasonably reduce the computational and storage complexity on board when compared to the DV-DVTR and FSA protocols that compute the routing tables off-line and then load them to the nodes.

A localized Routing Scheme (LRS) is used for the LEO satellite network with the network modelled as a mesh-like Manhattan Street Network (MSN) [41]. The LRS aims to save processing resources by reducing the size of routing tables that were calculated onboard a satellite to decrease the overall onboard computation complexity, including paths definitions and exchange. This LRS algorithm utilizes two routing techniques that are divided into intra-zone routing and inter-zone routing. The whole network is divided into zones; any adjacent virtual nodes are grouped in one zone. In the intra-routing stage, LRS elects a node to coordinate the routing within the zone and also selects a minimum hop path to the other zones utilizing the MSN binary method. In the inter-routing stage, data will be routed among the zones that are based on the shortest path; however, the traffic load among the zones is not considered in the inter-zone routing, which might lead to performance degradation. A Satellite network Link State Routing (SLSR) was proposed to address the high overhead and long delays on satellite networks while using the same MSN model [42]. Routing includes two phases with propagation delay routing tables, based on the satellite dynamics, calculated off-line in the first phase. In the second phase, the topology is regularly updated to determine the broken links or failed satellites, as well as any congestion information. An enhanced flooding mechanism is used in the second phase to exchange network information to ensure reliability and reduce the high overhead that is caused by the original flooding technique.

3.2. Multi-Layer Routing Protocols in Connected CubeSat Network

Multi-layer routing was proposed to resolve some of the single-layer routing drawbacks, such as the low persistence and efficiency of the network, and the concept of Multi-Layer Satellite Networks (MLSNs) are introduced to enhance the performance of satellite networks. In the multilayered architecture, satellites are disseminated over two or three standard orbits, such as LEO, MEO, and GEO orbits. Figure 4 illustrates standard satellite orbits.

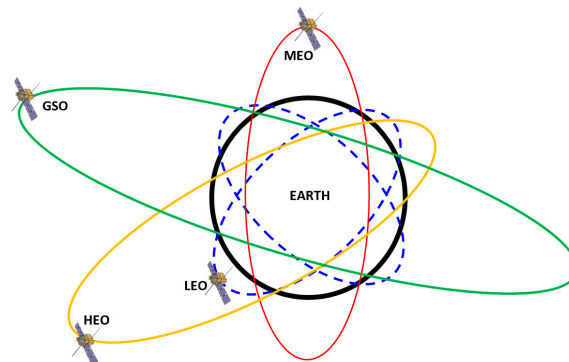


Figure 4. Standard satellite orbits.

In other words, MLSN architecture allows for the satellites in the higher layer to coordinate with those in the lower layer which means that all of the computational and storage complexity will be more relaxed on the lower layer satellites without the need for ground segment intervention. Figure 5 depicts the multi-layer architecture with three layers.

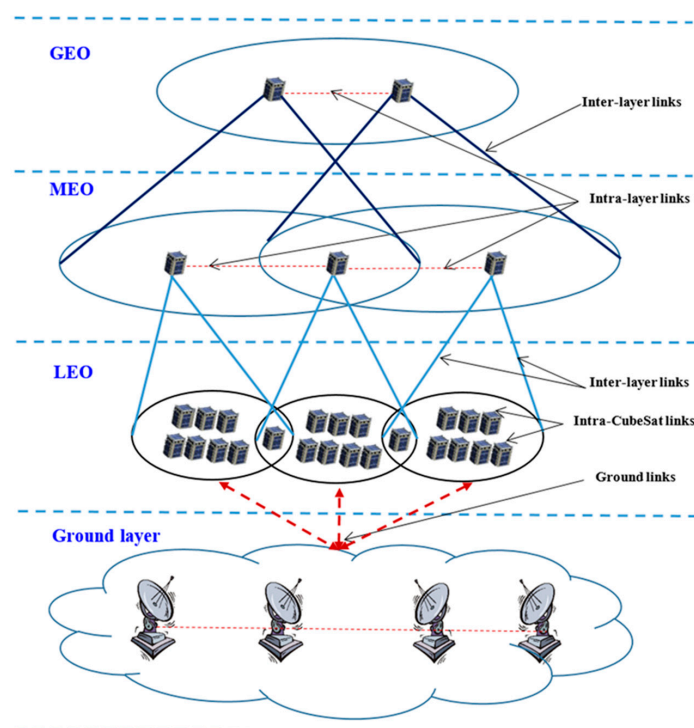


Figure 5. Multi-layer satellite network hierarchy.

In multi-layered architecture, the management of any lower layer is the responsibility of the higher layer, for instance, GEO satellites will manage the MEO satellites, whereas the LEO satellites are accountable to the ground access. While a satellite can communicate with another satellite on the

same orbit using inter-satellite or intra-satellite links, the inter-layer links are used for establishing communication between the satellites on different orbits. Satellites in the higher orbit will have a wider communication window with satellites in the lower orbit, which increases the efficiency of the network management and provides flexible routing.

In general, the routing algorithms proposed for multi-layer satellite networks depend on the master-slave concept. Table 3 summarizes the multi-layer routing algorithms that are proposed for a variety of multi-layered satellite networks.

Table 3. Multi-layer routing algorithms.

Routing Algorithm	Protocol Type	Topology Information	Decision Criteria	Resources Consumption	Links Recovery
HSRP [27]	Connection oriented	logical locations/ Mesh-grid/ flooding	Node info/Topology state/resource status	High	Yes
MLSR [28]	Connection oriented	logical locations/ Mesh-grid	Node info/Topology state/resource status	High	Yes
RSSBN [29]	Connection oriented	logical locations/ Mesh-grid	Node info/Topology	High	No
ARS-MLSN [30]	Connectionless	logical locations/ Mesh-grid	Link load	High	No
SGRP [31]	Connectionless	Predictable/ logical locations/ Mesh-grid	Link metrics	Medium	Yes
SARA [32]	Connectionless	Predictable/ logical locations/ Mesh-grid	Link metrics	Medium	Yes
PSNRP [33]	Connectionless	Predictable/dynamic	Link metrics	Medium	Yes
NDRA [34]	Connectionless	logical locations/ Mesh-grid	Data-driven/ Link metrics	High	No

Satellite-Over-Satellite (SOS) architecture was introduced to enhance the routing efficiency over ISLs of broadband and real-time satellite networks [27]. The authors proposed a Hierarchical Satellite Routing Protocol (HSRP) that is a multilayered, dynamic, and QoS-based routing algorithm for high-density SOS networks to provide multimedia and real-time services. In SOS, satellites are assumed to be distributed over multiple orbits and communication between the satellites and the ground stations involves three types of links: inter-orbit links; inter-satellite links; and, ground links. Inter-orbit links are incorporated for communication between different layers and inter-satellite links to connect satellites on the same layer, while the ground links allow for satellites in the lower layer of the network hierarchy to communicate with the ground stations. The SOS network is modelled as a connected grid with logical locations around the Earth to provide full coverage, while assuming that each satellite from the higher layer will have a cluster of lower layer satellite covered by its footprint. Although SOS architecture can provide a better performance, the HSRP flooding mechanism that is used to collect the topology information leads to high overheads and complexity. A Multi-Layered Satellite Routing (MLSR) algorithm was developed for LEO, MEO, and GEO three-layered satellite IP architecture based on the same concept of SOS [28]. With a large number of multilayered satellites, MLSR decentralizes routing table calculations among the high-level management GEO satellites to reduce the computational complexity. MLSR also reduces the communication costs by grouping LEO satellites to update GEO satellites with outlined link delays. MLSR considers propagation delay as well as computation and queueing delays. The link recovery technique in the case of satellite failure is also considered in the routing table calculations. In the same way, a three-dimensional hierarchy satellite network is evaluated while using the OPNET network simulator [29].

In Reference [30], a Multi-Layer Satellite Network (MLSN) that is based on the Walker Delta satellite constellation was developed and tested with the introduced Adaptive Routing Strategy (ARS) with the proposed structure providing continuous connectivity among satellites on the same layer and intermittent connectivity between different layers. The ARS design is based on the Bellman-Ford backwards-forward routing protocol for selecting optimal paths. The path with the least total path weight is considered to be the most optimal path from among all other available paths. A Satellite Grouping and Routing Protocol (SGRP) was proposed for two-layered LEO/MEO structures [31]. The

LEO satellites are modelled as a logical location grid that changes over time in order to show the different snapshots of the topology. The LEO satellites are then grouped according to the footprints of MEO satellites. The MEO satellites are in charge of routing table calculations and updates and these tables are then sent and stored at LEO satellites. SGRP route the data over paths with minimum delay while considering link propagation delay and congestion as well as transmission delays at the satellites. SGRP uses the same recovery technique of MLSR in the case of detected satellite failure. However, only routes that are affected by a satellite failure will be rerouted by the MEO satellites to reduce the computational complexity. A Snap-based Autonomous Routing Algorithm (SARA) was introduced to enhance satellite failure link recovery. When compared to SGRP, SARA achieved a similar propagation delay, but it outperforms SGRP in terms of overhead minimization and boosting delivery rates [32].

A Predictable Satellite Network Routing Protocol (PSNRP) was proposed for two-layer LEO/MEO MLSNs [33]. PSNRP addresses some drawbacks of the previous multi-layered routing protocols for enhancing routing robustness and network survivability. For example, the lack of effective application of satellite movement predictability in the previously mentioned multilayered protocols. Additionally, instead of assigning the routing table management to the MEO satellites, control centres on the ground are employed in conjunction with the MEO satellites to fulfil this task. In this case, the computational burden on the MEO satellites is reduced to avoid any congestion at these major nodes, which may otherwise lead to performance degradation. PSNRP utilizes a simple link recovery technique that separates the user data from control data and floods link states to all LEO satellites as a response to any detected disruption. Additionally, to enhance QoS requirements, such as delay and bandwidth, a routing algorithm based on heuristic techniques was adapted by PSNRP and it is considered to be an NP-complete problem. The simulation results show that PSNRP outperforms SGRP in terms of achieving minimum end-to-end-delay, lower overhead, and robust link recovery.

A Novel Distributed Routing Algorithm (NDRA), which is data-driven-based, was proposed for hybrid LEO/GEO multilayer satellite networks [34]. The algorithm imports a Distributed Routing Algorithm (DRA) that was introduced for virtual node routing and applies it in multilayer architecture. The proposed architecture includes three GEO satellites and 288 LEO satellites to cover the Earth, which is divided into logical locations with the LEO satellites being divided into three groups according to the GEO satellite footprints. These GEO satellites will be used as a one-hop relay to avoid congestion over the LEO links. Additionally, the packets in NDRA are categorised to real-time and non-real-time packets following the packets categorisation idea that was proposed in [44]. The non-real-time packets can be forwarded via the long propagation delay links of GEO, whereas the sensitive real-time packets are forwarded by LEO links. Although NDRA is deemed to have better performance than DRA, this comes at the expense of high overheads at congested satellites [34–36]. In Reference [45], a congestion prediction technique is proposed to maintain both QoS and load balancing.

3.3. Discussion

Single-layer and multi-layer architectures and their related routing protocols have both been discussed throughout this section. These routing protocols have mainly been proposed for conventional satellites that were featured with high communication capability and reliability. Most of these protocols have assumed that the continuous connectivity between satellites can be established and traditional TCP/IP can be modified and used in such environments. The single-layer networking architecture and its associated routing protocols can generally provide better performance when compared with a single satellite, particularly in terms of maximising the connection window with the ground stations, increasing spatial and temporal data collection, and connecting two or more points on the ground through satellites. However, the existing single-layer architecture routing protocols cannot meet CubeSats networking survivability requirements. The alternative concept of routing in a multi-layered satellite network demonstrates itself as being very beneficial for resource-scarce networks when compared with the typical single-layer architecture. Having satellites that function as a backbone support system, with suitable specifications in high layers of a multi-layered system, allows for better

networking capabilities by reducing the intermittent connectivity among the lower layer satellites. Nevertheless, the highly dynamic environment that is created by the mobility pattern of a large number of CubeSats remains challenging, especially with the restriction of limited resources that are available for these satellites, such as radio range, bandwidth, and the high failure rates, which result in intermittent connectivity. Accordingly, traditional TCP/IP based protocols are not reliable, as if a node cannot find a route towards the packet's destination the packet will be dropped, which in turn leads to a lower delivery ratio. Hence, in the next section, DTN routing protocols that tackle the problem of dis-connectivity in a network will be considered.

4. DTN Routing Protocols in Disjointed Satellite Networks

As discussed in [44,46–51], DTN routing protocols are categorized into three groups, namely: (i) flooding-based protocols in which an unlimited number of replicas for each generated bundle can be disseminated throughout the network; (ii) quota-based protocols in which the number of replicas is limited and fixed for every generated bundle; and, (iii) forwarding-based protocols where a single copy of messages/bundles is forwarded onto contacts until it reaches its destination or it has been dropped by the message dropping police. This final group implies that, at any given time, there is only one copy of each generated bundle throughout the network. Although forwarding-based protocols are considered to be resource-friendly protocols at the cost of producing long delays, flooding and quota-based protocols can provide better performance in terms of minimizing delay and increasing throughput. However, flooding-based protocols acquire high overheads because of the high rate of bundle replication and dissemination. This section discusses the features and limitations of these generic protocols and considers their application in CubeSat constellation and swarm environments. Table 4 shows the analysis of DTN routing protocols, which are discussed in [48].

4.1. DTN Flooding-Based Techniques

The well-known flooding-based Epidemic routing is a baseline routing algorithm in the DTN world. The bundles in Epidemic routing are mainly distributed through the network in the same manner as the spread of a viral epidemic [46], which can result in a high probability of messages reaching their destinations in the shortest period if the storage and bandwidth of the nodes are unlimited. However, epidemic routing does lead to wasteful usage of resources in terms of power, bandwidth, and buffers at each node. Besides, message replicas may continue exhausting network resources as a consequence of the lack of negative acknowledgements, even after a copy has already been delivered to its destination. The simple epidemic scheme was enhanced by introducing some bundle dropping policies [50]. In this scheme, the buffer size of each node has been restricted to only be able to store a limited number of bundles based on the assigned dropping policy. Four types of dropping policies were investigated, including Drop-Random (DRA); Drop-Least-Recently-Received (DLR); Drop-Oldest (DOA); and, Drop-Least-Encountered (DLE). The results of their simulation illustrate that DLE and DOA produce the best performance. The DLE policy focuses on dropping bundles according to information concerning node mobility and location, whereas the DOA policy focuses on dropping bundles according to a message delivered duration where the oldest bundles have the highest probability of already having been delivered to their planned bundle endpoints. A (p,q)-Epidemic routing is proposed as another modified form of the simple Epidemic [52]. The (p,q)-Epidemic protocol is designed for networks with random mobility and unpredictable future meetings that make it unreasonable for satellite networks. However, a VACCINE recovery technique that is used to delete needless replicas from the network can be adapted for satellite networks. New schemes were proposed to control flooding in sparse mobile DTNs [51]. These probabilistic schemes are based on Time-To-Live (TTL), in addition to an expiration time that is associated with every bundle. Moreover, after a message is delivered to its destination, a healing process is established to clear all unnecessary message replicas from the network. This healing process is based on a death certificate concept in the context of replicated database maintenance, which was originally introduced by [75].

Table 4. Delay tolerant network routing algorithms.

Protocols	Category	Decision Criteria	Advantages	Limitations	Delivery Ratio	Average Delay	Overhead	Complexity*
Epidemic [46]	Flooding	None	- Simple, no prior knowledge required	- High drop ratio - High overhead ratio	High if resources unlimited	Low if resources unlimited	High	O(n)
PROPHET [49]		History	- Universal and based on the delivery probability	- High drop ratio. - Acts like Epidemic - Low delivery probability	High if resources unlimited	Low if resources unlimited	High	O(n)
Davis et al. [50]		History	- Dropping policies. - Considers limited buffer size	- Incur high traffic load - Considers unlimited bandwidth	High if resources unlimited	Low if resources unlimited	High	O(n)
Harras et al [51]		History	- Reducing the number of messages	- High drop ratio with limited resources	High if resources unlimited	Low if resources unlimited	High	O(n)
(p, q)-Epidemic [52]		None	- Recovery process to clear unnecessary messages. - No prior knowledge required	- High drop ratio with limited resources - High power consumption	High if resources unlimited	Low if resources unlimited	High	O(n)
NECTAR [53]		History	- Less messages traffic	- High drop ratio with limited resources - High power consumption	High if resources unlimited	Low if resources unlimited	High	O(n)
FRESH [54]		History	- Reduced route discovery overheads	- High drop ratio if resources are limited - High power consumption	High if resources unlimited	Low if resources unlimited	High	O(n)
SEPR [55]		History	- Reducing the number of messages. - Power-efficient	- Average drop ratio if resources are limited	High if resources unlimited	Low if resources unlimited	High	O(n)
MobySpace [56]		History	- Less messages traffic	- High drop ratio - High power consumption	High if resources unlimited	Low if resources unlimited	High	O(n+m ²)
MEED [57]		History	- Less messages traffic	- High drop ratio if resources are limited. - High power consumption	High if resources unlimited	Low if resources unlimited	High	O(n+m ²)

Table 4. Cont.

Protocols	Category	Decision Criteria	Advantages	Limitations	Delivery Ratio	Average Delay	Overhead	Complexity*
MaxProp [58]	Flooding	History	- Less messages traffic	- High drop ratio limited. - High power consumption	High if resources unlimited	Low if resources unlimited	High	$O(n + m^2)$
MV [59]		History	- Less messages traffic	- High drop ratio if resources are limited. - High power consumption	High if resources unlimited	Low if resources unlimited	High	$O(n + m^2)$
PREP [60]		History	- Less messages traffic	- High drop ratio if resources are limited. - High power consumption	High if resources unlimited	Low if resources unlimited	High	$O(n + m^2)$
Spray and Wait [61]	Quota	None	- Simple and recourses friendly	- High drop ratio if resources are limited. - High power consumption	High if resources unlimited	Low if resources unlimited	Medium	$O(r)$
Spray and Focus [62]		None	- Simple and recourses friendly	- High drop ratio if resources are limited. - High power consumption	High if resources unlimited	Low if resources unlimited	Medium	$O(r)$
ORWAR [63]		History	- Simple and recourses friendly	- High drop ratio if resources are limited. - High power consumption	High if resources unlimited	Low if resources unlimited	Medium	$O(r)$
Bulut et al. [64]		None	- Simple, no prior knowledge required	- High drop ratio if resources are limited. - High power consumption	High if resources unlimited	Low if resources unlimited	Medium	$O(r)$
UDM [65]		History	- Recourses friendly	- High drop ratio if resources are limited. - High power consumption	High if resources unlimited	Low if resources unlimited	Medium	$O(r)$
MCSR [66]		History	- Recourses friendly	- High drop ratio if resources are limited. - High power consumption	High if resources unlimited	Low if resources unlimited	Medium	$O(r)$

Table 4. Cont.

Protocols	Category	Decision Criteria	Advantages	Limitations	Delivery Ratio	Average Delay	Overhead	Complexity*
MFC [67]	Quota	Space-time graph	- Recourses friendly	- High computational complexity	High if resources unlimited	Low if resources unlimited	Medium	$O(m^2 + m^2)$
EBR [68]		History	- Recourses friendly	- High drop ratio - High power consumption	High if resources unlimited	Low if resources unlimited	Low	$O(r)$
DBRP [44]		History	- Recourses friendly	- High drop - High power consumption	High if resources unlimited	Low if resources unlimited	Low	$O(r)$
AMRT [69]		History	- Recourses friendly	- High drop ratio - High power consumption	High if resources unlimited	Low if resources unlimited	Low	$O(r)$
MBRP [47]		Space-time graph	- Recourses friendly	- High computational complexity	High if resources unlimited	Low if resources unlimited	Low	$O(m^2 + m^2)$
RCM [70]	Forwarding	Space-time graph	- Recourses friendly	- High computational complexity	Low	High	Low	$O(m^2 + m^2)$
Huang et al. [71,72]		Space-time graph	- Recourses friendly	- High computational complexity	Low	High	Low	$O(m^2 + m^2)$
Merugu et al [73]		Space-time graph	- Recourses friendly	- High computational complexity	Low	High	Low	$O(m^2 + m^2)$
CAR [74]		History	- Recourses friendly	- High drop ratio - High power consumption	Low	High	Low	$O(n)$

* Where m, n and r represent the number of nodes, number of nodes without the given bundle, and number of bundle replicas respectively in the computational complexity column.

A PROPHET routing protocol that employs probabilistic routing utilizing the history of contacts and the mobility of nodes is proposed [49]. This approach considers using the benefits of non-random mobility, which may include typical real-world scenarios, such as the periodic satellite movement. A global unique bundle ID should be assigned to all of the bundles to identify the bundles that have already been seen as well as any unnecessary re-transmissions. In the same manner as the TTL field in the IP packets, there is also a hop count field that is associated with each bundle to specify a bundle detouring threshold. The hop count is considered to be one of the most important metrics affecting protocol resources utilisation. In contrast, for networks with random probability patterns, Oliveira and Albuquerque proposed a NECTAR protocol [53], which is a history-based routing technique that uses the previous contacts with neighbouring nodes and stores this information as a neighbourhood index. The neighbourhood index is determined based on a distance measure, which is the number of hops and the ageing measure that specifies the number of time slots where two nodes are out of the communication range. In the encounter time, the nodes will exchange the neighbouring information and choose the node that had the previous contact with the destination, as it may have a quicker likelihood of future contact. While NECTAR does not require any prior global knowledge regarding the topology, all of the nodes cache current neighbouring contacts to minimize resources harvesting, where a discarding policy is introduced due to the assumption of limited storage availability. Each node has two counters: one to count the number of received bundles and another to count the elapsed time for each bundle since its arrival. The product of these counters is used to calculate the ageing index and bundles with a higher ageing index are dropped first. Bundles with the least hop count to their destination will have higher delivery priority, according to the message scheduling policy of NECTAR. The NECTAR showed a better performance when compared with the Epidemic and PROPHET protocols in terms of having higher data rates and lower resource consumption in networks with restricted resources. Even though some features of NECTAR are reasonably suitable for CubeSat networks, maintaining all of the routing computations onboard and not considering the periodicity of satellite movement make it unfavourable. FResher Encounter Search (FRESH) was originally proposed for the concept of routing based on historical encounters and an ageing index for Mobile Ad-hoc Network (MANET) by [54].

A Shortest Expected Path Routing (SEPR) was introduced to solve the routing problem in semi-connected ad-hoc networks that include randomly dynamic nodes with limited resources [55]. Instead of flooding data, like in Epidemic, routing in SEPR uses Expected Path Length Metric (EPLM) to reduce the message copies, where each node develops its topology map by exchanging link-state information with its neighbouring nodes. Once a complete topology map is collected, the node runs the Dijkstra algorithm to define all of the expected path lengths among all pairs in the network. No recovery techniques were developed to deal with link and node failures, even though these protocols considered the limitation of the resources of the nodes.

In a similar manner, MobySpace uses Euclidean space as a method to improve the forwarding decisions in the generic DTN routing protocols for networks that have predictable mobility [56]. Each node is defined in the Euclidean space by coordinates that reflect the probability of its appearance in all different locations. The forwarding decision is made according to historical topology information. Based on this information, MobySpace selects a suitable next hop that is based on the mobility pattern similarity of the node as compared with the mobility of the destination node. Where two nodes have a similar encounter history, this means that they regularly meet with the same connection nodes due to being close to each other in their mobility pattern. Delivery probability is directly proportional to the pattern similarity.

A Minimum Estimated Expected Delay (MEED) algorithm was proposed to calculate the estimated average delay to the next-hop nodes for networks with unpredictable behaviour, yet assumes that an encounter with a node in the past indicates the probability of a future encounter [57]. In MEED, global network topology information is gained by flooding link state messages while using a link-state routing algorithm with routing tables dynamically updated after each encounter to consider any unpredictable

encounters. MEED outperforms other protocols that have predetermined global knowledge, in terms of providing a minimum delay path, even without prior knowledge about the topology. From here, MEED could be a suitable candidate for dense CubeSat networks, while assuming that there is no restriction on the onboard storage. However, this algorithm still generates high overheads due to the flooding of link state messages and no recovery technique has been proposed.

MaxProp is a restricted flooding-based algorithm, which, although originally proposed for a vehicular DTN environment, can be generalised to any other DTN [58]. MaxProp is an extension of the MV protocol [59]. The main idea of MaxProp is in the decision mechanisms proposed for forwarding and dropping messages, where the messages are prioritised based on the probability of meeting the destination in the future. Based on historical information, higher probability is given to a message according to its delivery probability, for example, if it has a minimum hop count to the destination. MaxProp also developed some techniques to boost the delivery rates and decrease delays. In the encounter time, those messages with one hop away from their destination are delivered first while routing information will be given second priority. Additionally, all delivered messages are acknowledged by flooding acknowledgment messages, which are much smaller than the data itself, to all of the other nodes not only the source—a technique that serves to clear any other copies of the message from the buffers. MaxProp uses a hop list in the message that includes all of the encountered nodes to avoid sending the message to the same node twice, including the source node. When compared with the Dijkstra algorithm with predicated mobility, MaxProp showed better performance in achieving a lower average latency and higher delivery rates [58]. While some of these proposed techniques are favourable in satellite network scenarios while considering the periodic mobility of satellites, MaxProp still produces high overheads in a restricted resources network and path recalculation in the case of node failure is not considered. Similarly, a PRioritized EPidmic (PREP) was designed to mitigate the load on resources by assigning a scheduled priority to each message, but the transmission overheads are still excessive [60].

4.2. DTN Quota-Based Techniques

DTN quota-based techniques were introduced to moderate the negative effect of flooding data and acknowledgment throughout the network by controlling the number of message replicas produced. A Spray and Wait is one of the DTN routing protocols that tries to take the advantage of both the high delivery ratio that is achieved by replication-based routing and the intelligent resources utilisation of forwarding-based routing [61]. It attains the intelligent employment of network resources by using a limited number of replicas that are distributed throughout the network. This can be accomplished by injecting a small number of copies into the network and waiting until the message reaches its destination while accounting for delay circumstances. This protocol has two phases: Spray and Wait. All of the created messages are assigned a number L that shows the maximum acceptable number of copies. In the Spray phase, the source is accountable for delivering a copy of the message to the L selected nodes (relays). These L nodes will hold the message in the Wait phase until the destination comes into direct contact with one of them. The Spray and Wait protocol is divided into two main types: Vanilla and Binary. The two types are almost the same, except in the way that L replicas spread to L selected nodes in the Spray phase. The vanilla type is simple with the source sending copies of the message to the first $(L - 1)$ selected nodes it contacts after creating the message. In the Binary type, while the source follows the same process as with the Vanilla type, the selected nodes will then send 50% of the number of copies to the future encountered nodes, if they do not have already copies of the message. The advantage of the Binary Spray and Wait is that messages are more quickly distributed than the Vanilla type and expand their reach to destinations that are two hops away from the sources. The Binary Spray and Wait protocol has been approved as the best among all Spray and Wait schemes in regards to the minimum expected delay.

Spray and Focus enhanced the performance of Spray and Wait by developing a Focus phase instead of the Wait phase [62]. In the Focus phase, each node that has received a message from the Spray

phase will search for another intermediate node with a higher probability of meeting the destination to forward the message to it, instead of just waiting to directly meet the destination. This is obtained by using a utility function that counts the elapsed time since the last encounter of each two nodes. A small elapsed time indicates the proximity mobility pattern among the two nodes and incorporating this information increases the probability of delivery in the future. These timers are based on the technique that was originally proposed in the FRESH protocol [54].

Similar to Spray and Wait, an Opportunistic Routing with Window-Aware Replication (ORWAR) utilises the concept of a limited number of the message copies, except that copies of the message will not be randomly disseminated to the first encountered nodes [63]. Instead, the ORWAR evaluates the contact window at the node to select the most suitable message to be forwarded first. The contact window is calculated based on the node speed, movement pattern, and radio capabilities. The selection criteria are based on the utility per bit mechanism and, thereafter, the algorithm selects the message with the size that is suitable for the contact window to ensure complete message delivery and avoid retransmissions that will reduce the resources consumption, such as the power and bandwidth from the overall network view. The ORWAR evaluation against other DTN algorithms, such as Epidemic, Spray and Wait, and MaxProp suggests ORWAR's performance is better in providing greater delivery rates with less overhead. Consideration of the message size and the contact window for networks with asymmetric data rates and variable message lengths, such as the CubeSat networks, is crucial. In another similar work in [64], the proposed multiple spraying technique divides the deadline time span for a message to be delivered to its destination into multiple smaller periods. Every single period will contain the Spray and Wait phase and each message is associated with Time-To-Live (TTL) as a deadline to consider the minimum possible Wait phase. Thus, a limited number of replicas are injected for the Wait phase to start and, if no acknowledgment is received at the end of the Wait phase, the algorithm injects more replicas to raise the delivery probability. Similarly, the Utility-based Distributed routing algorithm with Multi-copies (UDM) allows for a source node to deliver multi-copies to a fixed number of its adjacent nodes [65]. After that, these intermediate nodes forward the message copies to nodes with a higher probability of encountering the destination node according to the utility function. The UDM algorithm varies from Spray and Wait in two main aspects. Firstly, instead of using direct routing to deliver the packet to the destination after injecting a finite number of replicas into the Spray phase, predicted routing that is based on a utility function is used. Secondly, UDM incorporates some buffer control techniques to handle packet congestion. Resultantly, the UDM can provide reduced overheads with a reasonable delivery delay when compared with Epidemic and Spray and Wait algorithms.

In Reference [66], a Multi-Copy Spray Routing (MCSR) protocol was proposed. MCSR uses the same concept of wisely spraying a finite number of replicas at the first stage, but then forwards the message to its destination in the second phase while using the single-copy forwarding technique. Using this strategy, MCSR can form an equilibrium by reducing the number of replicas of flooding-based algorithms and reducing the extensive delay produced by single-copy forwarding algorithms. More recently, Reference [68] proposed a heuristic-based algorithm that forwards the limited replicas to nodes that have a larger number of encounters with other nodes in their history, as this implies that these nodes also have a higher probability of encountering the destination nodes. However, under these circumstances, if the destination node is isolated and in a low node density area, the packet might never be delivered. Reference [44] proposed a heuristic-based algorithm that gives more weight to the nodes that encounter particular destinations to overcome this problem. Hence, if a node has a low frequency of encounters with other nodes, but has evidenced several encounters with a destination, that node is allocated a higher weighting to receive more replicas of packets that were associated with that destination. In another more recent work by [69], Adaptive Message Replication Technique (AMRT) that fits onto quota protocols was proposed. AMRT assigns different numbers of replicas to each generated packet based on the congestion that exists amongst the sender's neighbours. In Reference [47], a Mobility Based Routing Protocol (MBRP) was introduced to assess encountered nodes according to their logged mobility patterns to forward a restricted number of packet copies. In

the case of disjointed space-time graph, the MBRP uses mobility patterns to evaluate encountered nodes reachability, unlike history-based routing protocols, which uses prior encounters for future contact estimation.

While some quota-based algorithms only rely on retrieved information from the space-time graph for data routing in networks with mobile nodes, Reference [67] developed Multi-copy Forwarding in Cyclic (MFC) MobiSpace for DTNs with periodic mobility. This MFC extends the Expected Minimum Delay (EMD) that was originally proposed by the authors for a single-copy forwarding scenario [70], where the network is modelled as a state-space graph before an optimal stopping technique is used to extract the extended EMD.

4.3. DTN Forwarding-Based Techniques

In forwarding-based routing, one copy of each message is forwarded on an optimised path until it reaches the designated destination. In Reference [74], a Context-aware Adaptive Routing (CAR) is proposed as a solution for problems through intermittent connectivity in random mobility DTNs. CAR estimates the next hop based on the prediction techniques of the Kalman Filter algorithm and the utility function. The prediction techniques used in CAR do not require the nodes to keep track and store information on the history of the encounter with other nodes, an advantage that makes the CAR algorithm suitable for networks with limited resources. The CAR can reduce the transmission overheads when compared with other single-copy protocols, as well as with multi-copy protocols, such as Spray and Wait. In contrast, Liu et al. proposed a novel Expected Minimum Delay (EMD) metric and EMD-based routing protocol that they called Routing in Cyclic MobiSpace (RCM) for networks with periodic mobility [70,76,77]. This expected delay time represents the minimum time that is needed for an optimal forwarding scheme to transmit a message over a path to a destination at a certain time in a network that is characterised by its cyclic and intermittent connectivity. Being inspired by the Euclidean space of MobySpace originally introduced in [56], the Cyclic MobiSpace network has been modelled as a probabilistic space-time graph, where each encounter time of a node with other nodes is assumed from historical encounter information or previous awareness regarding the network topology, with an assumption that the mobility pattern will not subsequently change. Proceeding to interpret this network model as a probabilistic state-space-graph, it is comprised of a different state for each encounter of each node to remove the time dimension. Subsequently, the Markov Decision Process (MDP) has been applied to obtain the EMDs of the messages over time.

The routing problem in wireless networks that are characterised by predictable mobility is investigated in [73], where these networks suffer from the unavailability of continuous paths between all nodes. The continuous movement of nodes leads to a frequent absence of end-to-end paths between a source and destination at a particular time instant, which in turn makes using ordinary ad hoc routing protocol inappropriate. Although node movements may create a path with the progression of time between any two nodes in the network by using the store, carry, and forward techniques, finding these paths is very challenging, especially in wireless networks with large numbers of nodes. In such a network, a node's mobility can be predicted by a finite time limit or based on the periodic movement of nodes. A space-time graph model was proposed as a solution to this problem of routing in predictable wireless networks [73]. In particular, the routing problem is being resolved by extracting a space-time routing table that was derived from node movement over time to consider the time and place in which a node can forward the message to the next hop or destination where forwarding a message to the next hop becomes a function of both the destination and time. In this model, the network topology is captured over time to determine the connected parts of the network on different time slots. A Shortest Paths in Space and Time (SPST) routing algorithm was designed based on this space-time graph model to specify appropriate next hops and upcoming neighbours to minimise the end-to-end delay. The SPST algorithm was then compared with three other routing algorithms—Hot Potato Routing (HPR), Most Frequent Neighbour Routing (MFN), and Epidemic Routing (ER)—which are based on heuristics. A network of 128 nodes with 61 links was examined with only one message

in the network in the first case and with realistic message traffic in the second case. The lack of continuous connectivity between some nodes invalidates traditional ad hoc routing protocols and this path could be repeatedly disrupted, even if there was a path between two nodes. In terms of successful message delivery, SPST is able to deliver almost all of the messages created because, even though the network is not fully connected in certain times, there were space and time paths which form a source and destination connection over time. While these paths were detected by the SPST routing algorithm, it is still possible for SPST to fail in identifying some space-time paths within a particular time horizon. The performance of the other three heuristic algorithms was comparatively insignificant, where, although they disseminate the data through the network until it reaches its destination, they otherwise fully consume the resources of the node. While the performance of the HPR, MFN, and ER could be enhanced by increasing resource capacity, this would be at the expense of increasing resource exploitation. Some structural features of the network topology, such as node density, can also affect the performance of these heuristics. In comparing these routing protocols, the SPST also achieves a higher performance in terms of end-to-end delivery delay, with the authors assuming that the prediction of upcoming links is perfect and the links on the extracted space-time graph are continuously reliable for connection, for example, all of the packets on spatial or temporal links will be delivered and without any errors.

In reference [72], the problem of topology control (TC) was studied for DTNs with predictable movements, where the network topology can be known prior based on the predictable mobility of nodes over time. Previous studies have mostly concentrated on routing and data dissemination, however, and, as maintaining the efficient and dynamic topology of the DTN becomes difficult with numerous participants of wireless nodes, the authors modelled the time-evolving of DTN as a directed space-time graph, which can show spatial and temporal data regarding the network nodes. The challenge with the topology control problem is in discerning how to extract a sparse space-time graph from the original directed space-time graph while maintaining the connectivity of the network, such that any two nodes can still communicate with each other. The extracted space-time graph, in this case, will have a minimised total cost when compared to the original one. An example of a DTN network where a sparser structure is needed is the interplanetary space DTN, as it is very expensive to build high-density networks [78]. It is necessary to wisely organise the DTN topology by minimising the number of communication links in order to reduce the energy consumption or any other costs of the network. To that end, the TC problem has primarily proven to be an NP-hard problem by relating it to the directed Steiner tree (DST) problem [72,79]. Subsequently, the TC problem is defined as a problem of extracting a sparser space-time graph from the original space-time graph while maintaining the network connectivity and minimising the total cost. Accordingly, three greedy-based algorithms were proposed to be able to maintain the connectivity over time, while considerably decreasing the total cost of network topology. These three algorithms were known as: “Union of Least Cost Path Algorithm” (ULCP); “Greedy Algorithm Based on Least Cost Path”; and, “Greedy Algorithm Based on Least Density Bunch”. For these algorithms, a group of edges was repetitively added to the topology to link one or more pairs of nodes in the space-time graph. In each iteration, one least-cost path was added to link one pair of nodes, as with the second algorithm or bunch of paths were added to link multiple pairs, as with the third algorithm. The near-optimal solution can be theoretically achieved by the third algorithm. However, this new definition of the TC problem is quite dissimilar to the typical space-time-routing [73,77], which focus on finding the most cost-efficient space-time route between one pair of source and destination nodes. The authors also discussed the possibility of addressing the topology control problem in undirected delay-tolerant networks by using the proposed greedy algorithms.

The cost-efficient topology control problem (CETC) is also defined in [71], with a new factor being added to the TC problem that the cost of the least-cost path that links any two nodes in the extracted space-time graph should be more cost-efficient when compared with the least cost path in the original space-time graph. Towards this aim, two new greedy algorithms: Greedy Algorithm to

Delete Links (GrdDL) and Greedy Algorithm to Add Links (GrdAL), were introduced to minimise the total network topology cost while ensuring the connectivity and the cost-efficient paths among any two nodes during time evolution. All of the aforementioned algorithms were investigated by conducting simulations for each algorithm on both the random DTN networks generated by a classical random graph generator and the real-world DTN networks based on CRAWDAD statistics [80]. The simulation results show that the proposed algorithms can save, in cases of the low-density network, approximately 50 percent of the network cost and 50 percent of the number of edges; however, about 95 percent of costs and edges can be saved in the high-density networks. Even though the results demonstrate a proportional efficiency of the proposed methods, some limitations and weaknesses still exist. Firstly, the CETC considers the network from time slot (0) to time slot (T) on the space-time graph and packets were generated at the time (0). In some cases, packets that may arrive in the middle slots may not be delivered to the destination at the end of the period (T) on that extracted space-time graph. Secondly, the complexity of some algorithms is high, so the algorithm might experience long delays if the number of the nodes and the number of the time slots is increased. Thirdly, the TC problem is defined for predictable DTNs with the consideration of time evolution, where there was an assumption regarding the reliability of future links to deliver data without any errors or distortion and, additionally, no link recovery techniques were suggested. In reality, such a strong assumption might be acceptable for particular forms of DTNs, but it cannot be applied to general DTNs, as, practically speaking, the satellite wireless links are unreliable because of the instability of wireless channels. Although the mobility in some DTN can be prior predicted according to historical statistics or the nature of user movement, in some cases these predictions can be incorrect and, as such, it would be quite remarkable to investigate the TC problem with the consideration of DTN routing reliability as a new and important factor. Finally, all of the proposed algorithms have aimed to reduce the number of links on the extracted space-time graph to minimise the costs. However, this process can reduce the performance of the network in aspects, such as routing. Therefore, it is important to study the effect of cost minimisation resultant from TC on the routing performance to consider more optimum routing. In References [81,82], heuristic-based DTN algorithms were proposed for drones flight path optimization. These algorithms optimise the route for both consignment and data delivery with higher delivery efficiency compared with other DTN algorithms such as EBR and Epidemic.

4.4. Discussion

Flooding based routing algorithms are considered to provide the highest delivery rates and the minimum end-to-end delay when compared with other quota-based protocols; however, these protocols exhaust the network with large overheads. In quota-based routing protocols, the focus was on how an algorithm can mitigate the high overheads that are produced by flooding-based algorithms, enhance the network performance by increasing the delivery rate, and minimise the end-to-end delay. Most of these algorithms build a complete knowledge about the networks using historical information that is collected over time from exchanging data and acknowledgments between nodes. Resultantly, these algorithms still yield high overheads for networks with limited resources, such as the CubeSat network. However, no link recovery techniques have been proposed in these algorithms, and the lack of consideration for regular link failure that is caused by congestion and node failure might lead to network degradation, especially in networks with restricted resources. Furthermore, most of these protocols are proposed for unpredictable mobile networks, losing the advantage of predictable or semi-predictable satellite movement in developing routing tables. It has been proven through some of the aforementioned research that networks with global knowledge tend to have better performance, while, on another hand, forwarding-based routing that is based on the space-time graph guarantees the delivery of a message if there is a complete path between a source and its corresponding destination. The forwarding-based routing protocols also save network resources due to no message replication, although this is at the cost of high delay and low delivery rates.

5. Conclusions and Research Questions

Small satellite networks, particularly CubeSat networks, create a challenging environment for efficient data routing while considering that both the CubeSats mobility model, which creates a highly dynamic network that causes the network to be disjointed and the limitations of the CubeSat network, such as the size of the CubeSats, energy, radio range, and bandwidth, resulting in intermittent connectivity among the satellites over time. This paper highlights the main challenges in satellite communications, specifically when the network suffers from intermittent connectivity. Routing is the most challenging issue in satellite communications, where traditional routing protocols do not work.

By conducting this comprehensive survey on the state-of-the-art DTN and Non-DTN routing protocols, some issues that are related to applying existing routing protocols on CubeSat networks were identified. Accordingly, single-layer and multi-layer routing protocols continue to face additional issues. In single-layer routing, each CubeSat must maintain its routing table, which will consume some of its limited energy and will be also associated with some computational delay and storage complexity. The multi-layer protocols can solve this issue by transferring the routing table costs from the core satellites to a limited number of satellites in the higher layer. These higher-layer satellites should be featured with higher performance capabilities, so that each one of them can coordinate with a certain number of satellites in the lower layer. However, this poses another issue regarding increasing the traffic on cross-layer links where congesting such important links could raise the possibility of the failure of one or more of these backbone links, leading to the unfortunate result of severe degradation and actually jeopardise the mission. The assumption of network connectivity and the application of IP based routing protocols in the existing single-layer and multi-layer routing protocols make them inefficient for the disjointed CubeSat networks, which results in DTN routing protocols being adopted as a reasonable alternative. Based on the mission requirements, different DTN routing protocols can be chosen, where DTN routing protocols are suitable for different space missions, such as near the earth CubeSat networks, deep space networks, and interplanetary CubeSat network.

DTN routing protocols can work on top of both single-layer and multi-layer CubeSat networks. This will open new avenues for routing optimisation under the new communication conditions of CubeSat networks that are based on the intended mission objectives. Accordingly, current DTN routing protocols have been compared and analysed to find that, while the flooding-based routing can provide the highest delivery rates and minimum delivery delay, it is still inefficient for use in the resources-scarce CubeSat networks. This has been concluded because of the very expensive overheads that may harvest these limited resources and that conversely lead to rapid network degradation. The DTN quota-based routing protocol can be used to overcome the limitations of flooding by limiting the number of the message copies; however, quota-based routing protocols still produce high overheads that may lead to high congestion in such a restricted environment. Alternatively, the DTN single-copy forwarding protocols can be used and they will not acquire any overheads, as there is only one copy of the message to be forwarded to the destination. Unfortunately, these single-copy forwarding protocols suffer from long delays and a low delivery ratio. The DTN history-based routing protocols can build global knowledge regarding the whole network by exchanging information with all encountered satellites and using negative feedback. These negative acknowledgments will actually produce more traffic in the overall network and, in some cases, there is insufficient time for negative acknowledgments, especially during short encounter periods or when the path between a source and a destination no longer exists. Besides, most of the reviewed routing protocols are unable to differentiate between link failure that is caused by link congestion and the failure caused by the failure of the satellite itself. Therefore, consideration for which QoS management techniques, such as load balancing, dropping policies, and link recovery in the case of satellite failure, can be used to enhance routing performance in the challenging environment is crucial.

For the future, a High Data Rate and Low Power Consumption Shortest Path (HDRLP-SP) DTN routing protocol will be designed and implemented for CubeSat networks. In addition, the MEO/LEO Multi-Layered Hybrid Integrated CubeSat System (ML-HICS) architecture will be designed and two

new DTN routing protocols will be proposed accordingly, namely: (i) DTN HDRLP-SP single-copy forwarding-based routing protocol over ML-HICS architecture; and, (ii) DTN HDRLP-SP multi-copy quota-based routing protocol over ML-HICS architecture.

Author Contributions: Conceptualization, M.A.A.M., R.R. and S.I.; methodology, M.A.A.M.; software, M.A.A.M.; validation, M.A.A.M., R.R. and S.I.; formal analysis, M.A.A.M., R.R. and S.I.; investigation, M.A.A.M.; resources, M.A.A.M.; data curation, M.A.A.M.; writing—original draft preparation, M.A.A.M.; writing—review and editing, M.A.A.M., R.R. and S.I.; visualization, M.A.A.M. and R.R.; supervision, R.R.; project administration, R.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Antunes, S. *DIY Satellite Platforms: Building a Space-Ready General Base Picosatellite for Any Mission*; O'Reill Media: Sebastopol, CA, USA, 2012.
2. Radhakrishnan, R.; Edmonson, W.W.; Afghah, F.; Rodriguez-Orsorio, R.M.; Pinto, F.; Burleigh, S.C. Survey of Inter-Satellite Communication for Small Satellite Systems: Physical Layer to Network Layer View. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 2442–2473. [CrossRef]
3. Nag, S.; LeMoigne, J.; Weck, O.D. Cost and risk analysis of small satellite constellations for earth observation. In Proceedings of the IEEE Aerospace Conference, Big Sky, MT, USA, 1–8 March 2014; pp. 1–16.
4. Kulu, E. Nanosatellite Database. Available online: <https://www.nanosats.eu/index.html#contact> (accessed on 4 July 2019).
5. Lemur-2 Nanosatellite Constellation of Spire Global. Available online: <https://directory.eoportal.org/web/eoportal/satellite-missions/l/lemur> (accessed on 17 August 2018).
6. The Flock Earth Observing Constellations. Available online: https://space.skyrocket.de/doc_sdat/flock-1.htm (accessed on 16 August 2018).
7. Budianu, A.; Castro, T.J.; Meijerink, A.; Bentum, M.J. Inter-satellite links for cubesats. In Proceedings of the IEEE Aerospace Conference, Big Sky, MT, USA, 2–9 March 2013; pp. 1–10.
8. Saptarshi, B.; Giri, P.S.; Rebecca, F.; Daniel, M.; Soon-Jo, C.; Fred, H. A review of impending small satellite formation flying missions. In Proceedings of the 53rd AIAA Aerospace Sciences Meeting (AIAA SciTech: American Institute of Aeronautics and Astronautics), Kissimmee, FL, USA, 5–9 January 2015.
9. Paul, M.; Janise, M. A survey of communication sub-systems for intersatellite linked systems and cubesat missions. *J. Commun.* **2012**, *7*, 290–308.
10. Xie, S.; Lee, G.X.; Low, K.S.; Gunawan, E. Wireless Sensor Network for Satellite Applications: A Survey and Case Study. *Unmanned Syst.* **2014**, *2*, 261–277. [CrossRef]
11. Tubbal, F.; Raad, R.; Chin, K.-W. A Survey and Study of Planar Antennas for Pico-Satellites. *IEEE Access* **2015**, *3*, 2590–2612. [CrossRef]
12. Tubbal, F.; Raad, R.; Chin, K.W.; Butters, B. S-band Planar Antennas for a CubeSat. *Int. J. Electr. Eng. Inform.* **2015**, *7*, 559–568. [CrossRef]
13. Tubbal, F.; Raad, R.; Chin, K.-W.; Matekovits, L.; Butters, B.; Dassano, G. A high gain S-band slot antenna with MSS for CubeSat. *Ann. Telecommun.* **2018**, *74*, 223–237. [CrossRef]
14. Qi, X.; Ma, J.; Wu, D.; Liu, L.; Hu, S. A survey of routing techniques for satellite networks. *J. Commun. Inf. Netw.* **2016**, *1*, 66–85. [CrossRef]
15. Davoli, F.; Kourogiorgas, C.; Marchese, M.; Panagopoulos, A.D.; Patrone, F. Small satellites and CubeSats: Survey of structures, architectures, and protocols. *Int. J. Satell. Commun. Netw.* **2018**, *37*, 343–359. [CrossRef]
16. Madni, M.A.A.; Communication, T.S.O.D.I.A.W. Energy-Aware Routing for CubeSat Swarms. *Int. J. Digit. Inf. Wirel. Commun.* **2018**, *8*, 150–155.
17. Cakaj, S.; Kamo, B.; Lala, A.; Rakipi, A. The Coverage Analysis for Low Earth Orbiting Satellites at Low Elevation. *Int. J. Adv. Comput. Sci. Appl.* **2014**, *5*, 6–10. [CrossRef]
18. Arnold, S.S.; Nuzzaci, R.; Gordon-Ross, A. Energy budgeting for CubeSats with an integrated FPGA. In Proceedings of the 2012 IEEE Aerospace Conference, Big Sky, MT, USA, 3–10 March 2012; pp. 1–14.
19. Polat, H.; Virgili-Llop, J.; Romano, M. Survey, Statistical Analysis and Classification of Launched CubeSat Missions with Emphasis on the Attitude Control Method. *J. Small Satell.* **2016**, *5*, 513–530.

20. Werner, M. A dynamic routing concept for ATM-based satellite personal communication networks. *IEEE J. Sel. Areas Commun.* **1997**, *15*, 1636–1648. [[CrossRef](#)]
21. Chang, H.S.; Kim, B.W.; Lee, C.G.; Min, S.L.; Choi, Y.; Yang, H.S.; Kim, D.Y.; Kim, C.S. FSA-based link assignment and routing in low-earth orbit satellite networks. *IEEE Trans. Veh. Technol.* **1998**, *47*, 1037–1048. [[CrossRef](#)]
22. Bai, J.; Lu, X.; Lu, Z.; Peng, W. Compact explicit multi-path routing for LEO satellite networks. In Proceedings of the Workshop on High Performance Switching and Routing, Hong Kong, China, 12–14 May 2005; pp. 386–390.
23. Taleb, T.; Mashimo, D.; Jamalipour, A.; Hashimoto, K.; Nemoto, Y.; Kato, N. SAT04-3: ELB: An Explicit Load Balancing Routing Protocol for Multi-Hop NGEOSatellite Constellations. *IEEE Globecom* **2006**, *2006*, 1–5.
24. Suvarna, G.; Chandrashekhar, R. Study of Load Balancing Routing Algorithm for Low Earth Orbit Satellite Networks. *Int. J. Innov. Appl. Stud.* **2014**, *11*, 198–205.
25. Korcak, O.; Alagoz, F. Analysis of Priority-based Adaptive Routing in Satellite Networks. In Proceedings of the 2005 2nd International Symposium on Wireless Communication Systems, Siena, Italy, 5–7 September 2005; pp. 629–633.
26. Tan, H.; Zhu, L. A Novel Routing Algorithm Based on Virtual Topology Snapshot in LEO Satellite Networks. In Proceedings of the IEEE 17th International Conference on Computational Science and Engineering, Chengdu, China, 19–21 December 2014; pp. 357–361.
27. Jae-Wook, L.; Jun-Woo, L.; Tae-Wan, K.; Dae-Ung, K. Satellite over satellite (SOS) network: A novel concept of hierarchical architecture and routing in satellite network. In Proceedings of the 25th Annual IEEE Conference on Local Computer Networks. LCN 2000, Tampa, FL, USA, 8–10 November 2000; pp. 392–399.
28. Akyildiz, I.F.; Ekici, E.; Bender, M.D. MLRS: A novel routing algorithm for multilayered satellite IP networks. *IEEE/ACM Trans. Netw.* **2002**, *10*, 411–424. [[CrossRef](#)]
29. Zhao, Z.; Zhang, J.; Sun, F.; Li, L.; Chen, I.M. Routing strategy for satellite-based networks with a hierarchical space grid architecture. In Proceedings of the Fourth International Conference on Information Communications and Signals, Singapore, 15–18 December 2003; pp. 1939–1943.
30. Li, H.; Zhang, Q.; Zhang, N.; Zhang, Y.; Xu, H. Adaptive Routing Strategy in Multi Layer Satellite Communication Networks. In Proceedings of the 2006 7th International Symposium on Antennas, Propagation & EM Theory, Guilin, China, 26–29 October 2006; pp. 1–4.
31. Chen, C.; Ekici, E. A Routing Protocol for Hierarchical LEO/MEO Satellite IP Networks. *Wirel. Netw.* **2005**, *11*, 507–521. [[CrossRef](#)]
32. Liang, J.; Zhang, J.-W.; Xiao, N. Research and Simulation on an Autonomous Routing Algorithm for GEO_LEO Satellite Networks. In Proceedings of the Fourth International Conference on Intelligent Computation Technology and Automation, Shenzhen, China, 28–29 March 2011; pp. 657–660.
33. Liu, H.; Sun, F. Routing for predictable Multi-Layered Satellite Networks. *Sci. China Inf. Sci.* **2013**, *56*, 1–18. [[CrossRef](#)]
34. Feng, X.; Yang, M.; Guo, Q. A novel distributed routing algorithm based on data-driven in GEO/LEO hybrid satellite network. In Proceedings of the 2015 International Conference on Wireless Communications & Signal Processing (WCSP), Nanjing, China, 15–17 October 2015; pp. 1–5.
35. Ekici, E.; Akyildiz, I.F.; Bender, M.D. Datagram routing algorithm for LEO satellite networks. In Proceedings of the IEEE INFOCOM. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064), Tel Aviv, Israel, 26–30 March 2000; pp. 500–508.
36. Ekici, E.; Akyildiz, I.; Bender, M. A distributed routing algorithm for datagram traffic in LEO satellite networks. *IEEE/ACM Trans. Netw.* **2001**, *9*, 137–147. [[CrossRef](#)]
37. Papapetrou, E.; Karapantazis, S.; Pavlidou, F.-N. Distributed on-demand routing for LEO satellite systems. *Comput. Netw.* **2007**, *51*, 4356–4376. [[CrossRef](#)]
38. Xuezhi, J.; Lixiang, L.; Pei, Z.; Dapeng, W. A destruction-resistant on-demand routing protocol for LEO satellite network based on local repair. In Proceedings of the 2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), Zhangjiajie, China, 15–17 August 2015; pp. 2013–2018.
39. Liu, X.; Yan, X.; Jiang, Z.; Li, C.; Yang, Y. A Low-Complexity Routing Algorithm Based on Load Balancing for LEO Satellite Networks. In Proceedings of the 2015 IEEE 82nd Vehicular Technology Conference (VTC2015-Fall), Boston, MA, USA, 6–9 September 2015; pp. 1–5.

40. Liu, X.; Jiang, Z.; Liu, C.; He, S.; Li, C.; Yang, Y.; Men, A. A low-complexity probabilistic routing algorithm for polar orbits satellite constellation networks. In Proceedings of the 2015 IEEE/CIC International Conference on Communications in China (ICCC), Shenzhen, China, 2–4 November 2015; pp. 1–5.
41. Chan, T.-H.; Yeo, B.S.; Turner, L. A Localized Routing Scheme for LEO Satellite Networks. In Proceedings of the 21st International Communications Satellite Systems Conference and Exhibit (International Communications Satellite Systems Conferences (ICSSC): American Institute of Aeronautics and Astronautics, Yokohama, Japan, 15–19 April 2003.
42. Yan, H.; Zhang, Q.; Sun, Y. A Novel Routing Scheme for LEO Satellite Networks Based on Link State Routing. In Proceedings of the 2014 IEEE 17th International Conference on Computational Science and Engineering, Chengdu, China, 19–21 December 2014; pp. 876–880.
43. Donner, A.; Berioli, M.; Werner, M. MPLS-Based Satellite Constellation Networks. *IEEE J. Sel. Areas Commun.* **2004**, *22*, 438–448. [[CrossRef](#)]
44. Iranmanesh, S.; Raad, R.; Kwan-Wu, C. A novel destination-based routing protocol (DBRP) in DTNs. In Proceedings of the International Symposium on Communications and Information Technologies (ISCIT), Gold Coast, Australia, 2–5 October 2012; pp. 325–330.
45. Nishiyama, H.; Kudoh, D.; Kato, N.; Kadowaki, N. Load Balancing and QoS Provisioning Based on Congestion Prediction for GEO/LEO Hybrid Satellite Networks. *Proc. IEEE* **2011**, *99*, 1998–2007. [[CrossRef](#)]
46. Vahdat, A.; Becker, D. *Epidemic Routing for Partially Connected ad hoc Networks*, Technical Report CS-200006; Duke University: Durham, NC, USA, 2000.
47. Iranmanesh, S.; Chin, K.-W. A novel mobility-based routing protocol for semi-predictable disruption tolerant networks. *Int. J. Wirel. Inf. Networks* **2015**, *22*, 138–146. [[CrossRef](#)]
48. Iranmanesh, S. Data Management in Delay Tolerant Networks. Ph.D. Thesis, School of Electrical, Computer and Telecommunications Engineering, University of Wollongong, Keiraville, Australia, 2014.
49. Lindgren, A.; Doria, A.; Schelén, O. Probabilistic routing in intermittently connected networks, ACM SIGMOBILE mobile computing and communications review. *ACM SIGMOBILE Mob. Comput. Commun. Rev.* **2003**, *7*, 19. [[CrossRef](#)]
50. Davis, J.A.; Fagg, A.H.; Levine, B.N. Wearable computers as packet transport mechanisms in highly-partitioned ad-hoc networks. In Proceedings of the Fifth International Symposium on Wearable Computers, Zurich, Switzerland, 8–9 October 2001; pp. 141–148.
51. Harras, K.A.; Almeroth, K.C.; Belding-Royer, E.M. Delay tolerant mobile networks (DTMNS): Controlled flooding in sparse mobile networks. In Proceedings of the Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications Systems, Athens, Greece, 9–14 May 2004; Springer: Berlin/Heidelberg, Germany, 2005; pp. 1180–1192.
52. Matsuda, T.; Takine, T. (p,q)-Epidemic routing for sparsely populated mobile ad hoc networks. *IEEE J. Sel. Areas Commun.* **2008**, *26*, 783–793. [[CrossRef](#)]
53. Oliveira, E.d.; Albuquerque, L.D. NECTAR: A DTN routing protocol based on neighborhood contact history. In Proceedings of the ACM symposium on applied computing, Honolulu, HI, USA, 8–12 March 2009; pp. 40–46, 1529290: ACM.
54. Dubois-Ferriere, H.; Grossglauser, M.; Vetterli, M. Age matters: Efficient route discovery in mobile ad hoc networks using encounter ages. In Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing, San Francisco, CA, USA, 1–5 December 2003.
55. Kun, T.; Qian, Z.; Wenwu, Z. Shortest path routing in partially connected ad hoc networks. In Proceedings of the Global Telecommunications Conference, San Francisco, CA, USA, 1–5 December 2003.
56. Leguay, M.; Friedman, T.; Conan, V. DTN routing in a mobility pattern space. In Proceedings of the Acm Sigcomm Workshop On Delay-Tolerant Networking, Philadelphia, PA, USA, 22–26 August 2005; 2005; pp. 276–283.
57. Jones, E.; Li, L.; Schmidtke, J.; Ward, P. Practical Routing in Delay-Tolerant Networks. *IEEE Trans. Mob. Comput.* **2007**, *6*, 943–959. [[CrossRef](#)]
58. Burgess, J.; Gallagher, B.; Jensen, D.; Levine, B.N. MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks. *INFOCOM* **2006**, *6*, 1–11.
59. Burns, B.; Brock, O.; Levine, B.N. MV routing and capacity building in disruption tolerant networks. In Proceedings of the IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies, Miami, FL, USA, 13–17 March 2005; 2005; Volume 1, pp. 398–408.

60. Ramanathan, R.; Hansen, R.; Basu, P.; Rosales-Hain, R.; Krishnan, R. Prioritized epidemic routing for opportunistic networks. In *Proceedings of the the 1st International Mobisys Workshop On Mobile Opportunistic Networking*; ACM: New York, NY, USA, 2007; pp. 62–66.
61. Spyropoulos, T.; Psounis, K.; Raghavendra, C.S. Spray and wait: An efficient routing scheme for intermittently connected mobile networks. In *Proceedings of the ACM SIGCOMM Workshop On Delay-Tolerant Networking*, Philadelphia, PA, USA, 22–26 August 2005.
62. Spyropoulos, T.; Psounis, K.; Raghavendra, C.S. Spray and Focus: Efficient Mobility-Assisted Routing for Heterogeneous and Correlated Mobility. In *Proceedings of the Fifth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PerComW'07)*, New York, NY, USA, 19–23 March 2007; pp. 79–85.
63. Sandulescu, G.; Nadjm-Tehrani, S. Opportunistic DTN routing with window-aware adaptive replication. In *Proceedings of the 4th Asian Conference on Internet Engineering*, Pratunam, Bangkok, Thailand, 18–20 November 2008; pp. 103–112.
64. Bulut, E.; Zijian, W.; Szymanski, B.K. Cost-Effective Multiperiod Spraying for Routing in Delay-Tolerant Networks, *Networking. IEEE/ACM Trans.* **2010**, *18*, 1530–1543. [[CrossRef](#)]
65. Li, Z.; Shen, H. Utility-Based Distributed Routing in Intermittently Connected Networks. In *Proceedings of the 37th International Conference on Parallel Processing*, Portland, OR, USA, 9–12 September 2008; pp. 604–611.
66. Spyropoulos, T.; Psounis, K.; Raghavendra, C.S. Efficient routing in intermittently connected mobile networks: The multiple-copy case. *IEEE/ACM Trans. Netw.* **2008**, *16*, 77–90. [[CrossRef](#)]
67. Liu, C.; Wu, J.; Cardei, J. Message forwarding in Cyclic MobiSpace: The multi-copy case. In *Proceedings of the IEEE 6th International Conference on Mobile Adhoc and Sensor Systems*, Orlando, FL, USA, 12–15 October 2009; pp. 413–422.
68. Nelson, S.C.; Bakht, M.; Kravets, R. Encounter-based routing in DTNs. In *Proceedings of the INFOCOM 2009*, Rio de Janeiro, Brazil, 19–25 April 2009; pp. 846–854.
69. Iranmanesh, S.; Saadati, M. A novel congestion control technique in delay tolerant networks. *Int. J. Interdiscip. Telecommun. Netw.* **2018**, *10*, 20–35. [[CrossRef](#)]
70. Liu, C.; Wu, J. Routing in a cyclic mobispace. In *Proceedings of the 9th ACM International Symposium On Mobile Ad Hoc Networking And Computing*, Hong Kong, China, May 2008; ACM: New York, NY, USA, 2008; pp. 351–360. [[CrossRef](#)]
71. Minsu, H.; Siyuan, C.; Ying, Z.; Yu, W. Cost-Efficient Topology Design Problem in Time-Evolving Delay-Tolerant Networks. In *Proceedings of the Global Telecommunications Conference (GLOBECOM 2010)*, Miami, FL, USA, 6–10 December 2010; pp. 1–5.
72. Huang, M.; Chen, S.; Zhu, Y.; Wang, Y. Topology Control for Time-Evolving and Predictable Delay-Tolerant Networks. *IEEE Trans. Comput.* **2013**, *62*, 2308–2321. [[CrossRef](#)]
73. Merugu, S.; Ammar, M.H.; Zegura, E.W. *Routing in Space and Time in Networks with Predictable Mobility*; Georgia Institute of Technology: Atlanta, GA, USA, 2004.
74. Musolesi, M.; Mascolo, C. CAR: Context-Aware Adaptive Routing for Delay-Tolerant Mobile Networks. *IEEE Trans. Mob. Comput.* **2008**, *8*, 246–260. [[CrossRef](#)]
75. Demers, A.; Greene, D.; Hauser, C.; Irish, W.; Larson, J.; Shenker, S.; Sturgis, H.; Swinehart, D.; Terry, D. Epidemic algorithms for replicated database maintenance. In *Proceedings of the Sixth Annual Acme Symposium On Principles Of Distributed Computing*, August 1987; ACM: New York, NY, USA; pp. 1–12. [[CrossRef](#)]
76. Liu, C.; Wu, J. Practical Routing in a Cyclic MobiSpace. *IEEE/ACM Trans. Netw.* **2010**, *19*, 369–382. [[CrossRef](#)]
77. Xuan, B.B.; Ferreira, A.; Jarry, A. Computing shortest, fastest, and foremost journeys in dynamic networks. *Int. J. Found. Comput. Sci.* **2003**, *14*, 267–285. [[CrossRef](#)]
78. Burleigh, S.; Hooke, A.; Torgerson, L.; Fall, K.; Cerf, V.; Durst, B.; Scott, K.; Weiss, H. Delay-tolerant networking: An approach to interplanetary Internet. *IEEE Commun. Mag.* **2003**, *41*, 128–136. [[CrossRef](#)]
79. Zosin, L.; Khuller, S. On directed Steiner trees. In *Thirteenth Annual ACM-SIAM Symposium on Discrete Algorithms*; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 2002; pp. 59–63.
80. Scott, J.; Gass, R.; Crowcroft, J.; Hui, P.; Diot, C.; Chaintreau, A. CRAWDAD Data Set Cambridge/Haggle (v. 2006-01-31), CRAWDAD Wireless Network Data Archive. 2006. Available online: <https://crawdad.org/> (accessed on 1 November 2019).

81. Iranmanesh, S.; Raad, R.; Raheel, M.S.; Tubbal, F.; Jan, T. Novel DTN Mobility-Driven Routing in Autonomous Drone Logistics Networks. *IEEE Access* **2020**, *8*, 13661–13673. [[CrossRef](#)]
82. Iranmanesh, S.; Raad, R. A Novel Data Forwarding Strategy for a Drone Delay Tolerant Network with Range Extension. *Electronics* **2019**, *8*, 659. [[CrossRef](#)]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).