

SDN in the Stratosphere: Loon's Aerospace Mesh Network

Frank Uyeda*

Marc Alvidrez

Erik Kline[†]

Bryce Petrini*

Brian Barritt[†]

David Mandle[†]

Aswin Chandy Alexander

Loon LLC

Mountain View, California, USA

loon-network-paper@googlegroups.com

ABSTRACT

The Loon project provided 4G LTE connectivity to under-served regions in emergency response and commercial mobile contexts using base stations carried by high-altitude balloons. To backhaul data, Loon orchestrated a moving mesh network of point-to-point radio links that interconnected balloons with each other and to ground infrastructure. This paper presents insights from 3 years of operational experience with Loon's mesh network above 3 continents.

The challenging environment, comparable to many emerging non-terrestrial networks (NTNs), highlighted the design continuum between predictive optimization and reactive recovery. By forecasting the physical environment as a part of network planning, our novel Temporospatial SDN (TS-SDN) successfully moved from reactive to predictive recovery in many cases. We present insights on the following NTN concerns: connecting meshes of moving nodes using long distance, directional point-to-point links; employing a hybrid network control plane to balance performance and reliability; and understanding the behavior of a complex system spanning physical and logical domains in an inaccessible environment. The paper validates TS-SDN as a compelling architecture for orchestrating networks of moving platforms and steerable beams, and provides insights for those building similar networks in the future.

CCS CONCEPTS

• **Networks** → **Mobile ad hoc networks**; **Mobile networks**; **Wide area networks**; **Hybrid networks**; **Wireless mesh networks**; **Network dynamics**; **Network mobility**; **Network manageability**; **Programmable networks**; **Network resources allocation**.

KEYWORDS

Project Loon, Stratosphere, Balloon, Non-Terrestrial Network, Temporospatial SDN, TS-SDN, Minkowski

*Now at Google Inc.

[†]Now at Aalyria Technologies, Inc.

SIGCOMM '22, August 22–26, 2022, Amsterdam, Netherlands

© 2022 Copyright held by the owner/author(s).

This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in *ACM SIGCOMM 2022 Conference (SIGCOMM '22)*, August 22–26, 2022, Amsterdam, Netherlands, <https://doi.org/10.1145/3544216.3544231>.

ACM Reference Format:

Frank Uyeda, Marc Alvidrez, Erik Kline, Bryce Petrini, Brian Barritt, David Mandle, and Aswin Chandy Alexander. 2022. SDN in the Stratosphere: Loon's Aerospace Mesh Network. In *ACM SIGCOMM 2022 Conference (SIGCOMM '22)*, August 22–26, 2022, Amsterdam, Netherlands. ACM, New York, NY, USA, 17 pages. <https://doi.org/10.1145/3544216.3544231>

1 INTRODUCTION

Loon's mission was to bring affordable Internet connectivity to the long tail of unconnected people by exploring a radical approach using stratospheric balloons. The population that is not served by traditional telecommunications operators typically live in outlying areas with low population density and often with geographical features that cause terrestrial cell towers to perform poorly. Given a high per-user deployment cost and limited revenue potential, telecom operators have few incentives to deploy existing solutions for users in these areas, thus widening the digital divide [37, 43].

Non-terrestrial networks (NTN) present an opportunity to provide broadband Internet access to large geographic regions. Elevating the height of a wireless transceiver enables a greater coverage area, but until relatively recently, geostationary orbit at ~36,000 km was the only option for placing a transmitter at an altitude higher than a cell tower. Today, NTNs are being pursued using aircraft, High-Altitude Platform Stations (HAPS) in the stratosphere, and satellites at Low Earth Orbit (LEO), Medium Earth Orbit (MEO), and Geostationary Earth Orbit (GEO). The potential capabilities, challenges, and architectures of networks across these altitudes form a rich design space, which have been studied extensively [13, 32, 33, 46]. Planned LEO and MEO satellite constellations [38], at 100s and 1000s of kilometers of altitude respectively, can improve capacity density (i.e. Mbps per km²) by 1-2 orders of magnitude above GEO.

Given the limitations and costs associated with satellite offerings at the time of project founding (c. 2011), Loon's approach was to develop economical HAPS in the form of untethered, high-altitude balloons that could operate at an altitude an order of magnitude lower than LEO. At this altitude it is theoretically possible to provide mobile or fixed broadband services with far greater capacity density than the equivalent services from satellites.

In response to a rapidly evolving connectivity landscape, Loon partnered with mobile telecommunications companies (telcos) to provide 4G LTE mobile network expansion. Deployed in response

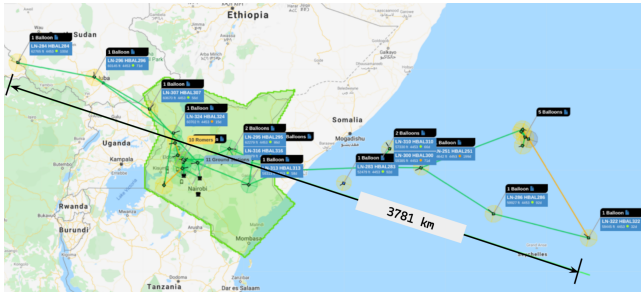


Figure 1: Mesh balloons spanning 3,781 km over East Africa and the Indian Ocean in February 2021. Loon’s meshes regularly contained 20+ balloons spanning 3000+ km.

to natural disasters since 2017¹ and in regular commercial service between August 2020 and March 2021, Loon served TiBs of data directly to the handsets of hundreds of thousands of unique users.

Due to geographical and cost constraints, Loon needed to form long range links between balloons and ground infrastructure. Our platforms used highly-directional steerable beams to form a floating mesh of point-to-point links that connected balloons to ground stations, to each other, and to other moving platforms. The balloons’ constant motion and focused beams necessitated the use of a centralized Temporospatial SDN (TS-SDN) that monitored and forecasted aspects of the physical environment and incorporated them into network planning [7, 8].

Loon’s deployed network drew from work across many areas including SDN [36, 42], MANET [25, 30], and ad-hoc networking using directional links [6, 45]. Though airborne meshes have been conceptualized and proposed [12, 15], there is limited public information on actual production deployments.

This paper goes beyond theory to discuss lessons learned from three years of operating Loon’s production non-terrestrial backhaul network. Based on Loon’s experience, this paper validates TS-SDN as a compelling technology for orchestrating networks of moving platforms and steerable beams, and provides insights for those building future non-terrestrial networks.

2 BACKGROUND

2.1 Loon’s Service Objective & Approach

The service objective for Loon’s first commercial deployment was to maximize the availability of 4G LTE data coverage to a 39,334 km² rural region of Kenya. Deploying terrestrial LTE cell towers to cover this area would have been extremely expensive and orbital platforms are too distant for mobile devices to connect using 4G. Instead, Loon’s approach was to deploy LTE sector antennas and base stations [4] on balloons floating 15-18 km in the sky. Given their altitude and minimal obstructions, these balloons could reliably serve ~5,000 km².

For Loon’s 4G LTE communications payload, illustrated in Figure 2, backhaul was provided from the balloons to our telco partner’s network core. Balloons used long-range point-to-point E band

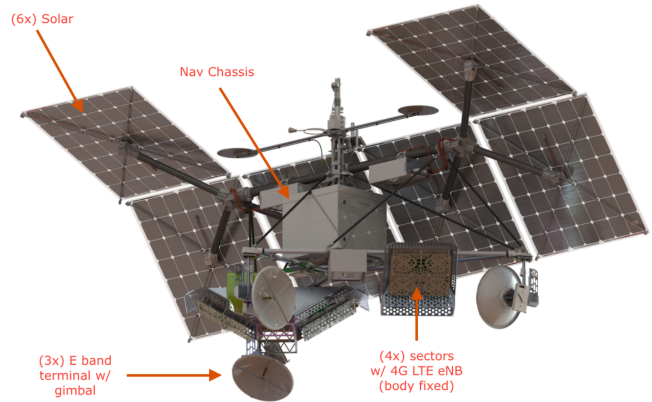


Figure 2: Loon bus with a communications payload containing four 4G LTE sectors, three E band backhaul transceivers, and multiple embedded computers.

radio links to connect to ground stations (GS) and to each other. These E band links are similar to inter-satellite links (ISLs) employed by some LEO satellites in capability and function. Loon placed a small number of ground stations in areas with reliable power and network connectivity. Ground stations acted as gateways between the balloon mesh and wired backhaul networks, multiplexing IPv6 traffic between the balloons in the mesh and Loon’s Edge Compute (EC) infrastructure using an overlay of encrypted tunnels. EC infrastructure comprised services, like extensible Virtual Network Functions (VNFs), and peered with our partner’s core network using private circuits. Figure 3 captures this architecture.

Ethical Concerns. Ethical considerations were at the heart of Loon’s mission to bring connectivity to un(der)served communities around the world. Loon systematically and collaboratively worked with aviation and telecom stakeholders to deploy a commercial service that would reduce the digital divide by providing fast, cost-effective Internet access to inaccessible areas. Further, Loon’s approach adhered to the highest standards of user data protection, employing encryption for all data at rest and end-to-end protection for LTE data in transit. Even when turning down the service, Loon worked with local community groups to retrieve landed balloons, often from very isolated and difficult locations, and to donate all usable equipment, including broadly useful and highly sought after solar panels and batteries. More detail in the Loon Library [3].

2.2 Challenges & Tradeoffs

The design of Loon’s communication platform was challenging because we were developing the communications systems at the same time that we were deploying a novel platform into an extreme operating environment [3]. As with other NTN, Loon had to balance many system-level trade-offs. These affected the design and deployment of the individual balloons, as well as the sizing and deployment of the fleet over time. The desire to learn quickly, paired with the relatively low cost of balloons, led us to deploy systems with known limitations up front and to update them frequently. Of particular importance to the network design were 1) position

¹Deployed in Peru for El Nino flooding in 2017, in Puerto Rico after Hurricane Maria in 2017-2018, and again in Peru after a major earthquake in Loreto in 2019.

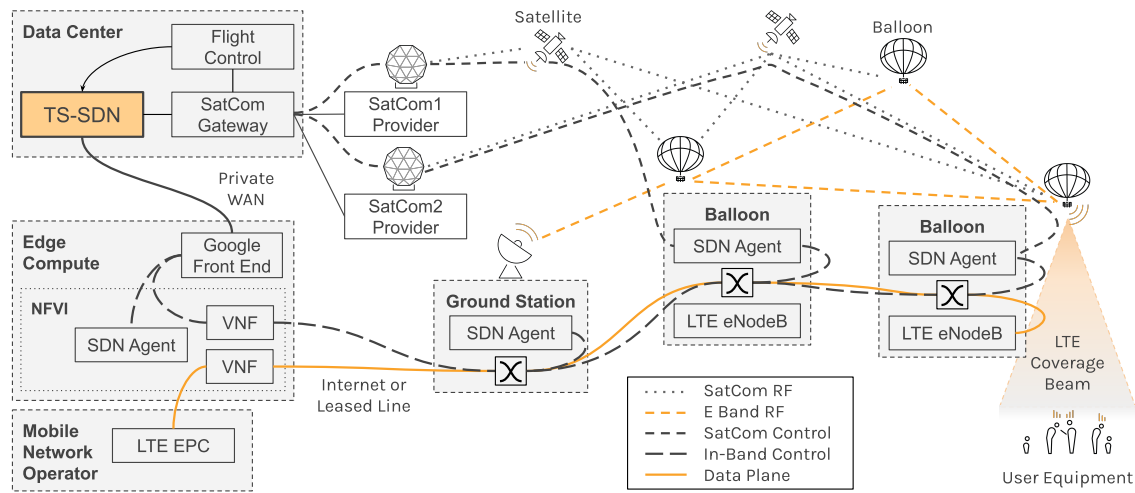


Figure 3: Loon's Data Plane and Tiered Control Plane Architecture.

and motion of balloons, 2) available power, 3) radio transceiver performance, and 4) a means to coordinate network nodes.

Navigation. Balloons floated freely in the stratosphere, but had the ability to change altitude. Loon's Fleet Management Software (FMS) modeled winds at different altitudes, then automatically instructed balloons to change altitude to catch the desired wind currents and drift toward a target over the service region [10]. Since the vehicles had no lateral thrust, the availability of the Loon network was dependent on the FMS's ability to maintain a balanced distribution of balloons over the service region. In stark distinction with deterministic satellite orbits and the directional control of flying UAVs, navigation for Loon was probabilistic due to the stochastic nature of the winds. Balloon trajectory was unpredictable to a meaningful degree, so the network controller formed meshes from whichever balloons were in position.

Balloon motion affected the shape of the network topology and its evolution. Loon operated three ground station sites and dozens of balloons that were continuously seeking the serving region. In total, 100+ backhaul transceivers (2 per ground site; 3 per balloon) could be tasked to form the backhaul mesh. Given n platforms, the possible pairings of antennas can approach $O(n^2)$.

The TS-SDN's "candidate graph" composes the set of all possible links between transceivers on different platforms that are expected to have acceptable characteristics. On average, the candidate graph contained 3275 links, with significant variation in the number of balloon-to-balloon (B2B) and balloon-to-ground (B2G) links, ranging from 0 to 6,595 and 0 to 750, respectively. Figure 4 shows the distribution of hour-to-hour differences in Loon's candidate graph in the final 3 months of service from December 1, 2020 to March 1, 2021. The candidate graph changed in 99.9% of hours with 13% median change. Only 3.5% of minutes saw a stable candidate graph, and at median 10 links changed minute to minute.

Power. Similar to orbital satellites, Loon balloons generated power using solar arrays and stored excess energy in batteries. Though

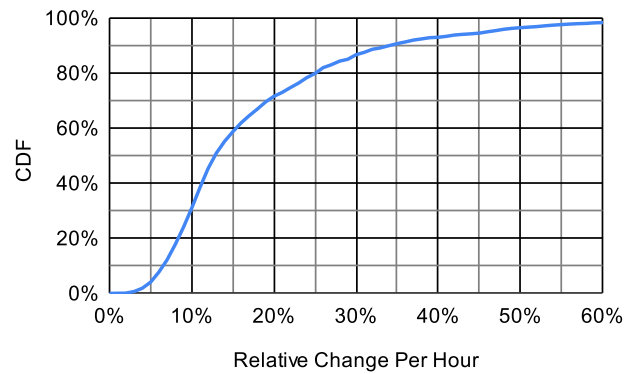


Figure 4: Hour-to-hour deltas in the set of candidate links.

power generation and battery capacities improved across balloon generations, system engineering trade offs for our latest balloons still resulted in insufficient energy storage to power the LTE and backhaul networks through the night. Instead, Loon served from shortly after dawn through the first few hours of darkness each day (approximately 14 hours). As a result, the Loon network had to bootstrap itself every day and gracefully handle communication nodes entering low power states. However, balloons kept a reserve of power for safety critical systems, such as flight control avionics and the satellite communication channels used to receive and relay flight commands and telemetry. Power generation and consumption details can be found in [3]. While balloons could not provide service at night, they could continue to station seek.

Radio Links. Each balloon carried three E band (71-76/81-86 GHz) transceivers, operating in licensed millimeter wave bands (similar to high-band 5G frequencies) and each capable of up to 1 Gbps. High-gain, highly directional antennas were mounted on mechanically

pointable gimbals at the three corners of the balloon's bus. To form a point-to-point link between two balloons or between a balloon and a ground station, antennas on the pairing platforms had to slew to aim at each other. The balloons employed differential GPS for tracking position and orientation [21], which they used for computing antenna pointing angles. Ground stations also used balloon's ADS-B [41] broadcasts to aid their antenna pointing, but unlike other schemes [2], Loon's use of ADS-B was unmodified.

Once a link was established, local control loops continuously tuned the antenna's pointing angle to track the remote transceiver based on received signal strength. In addition, link-local "one-hop" telemetry that included the node's position and heading was passed over established links to enable fast, local link reacquisition when antenna tracking failed.

Each antenna had a range-of-motion of 360° azimuth and an elevation range from nadir (directly below) to +20° above horizontal, allowing for substantial – though not complete – overlap between each antenna's field of regard. This overlap provided some, though incomplete, flexibility when choosing which antennas to task in forming a link. Due to mounting locations and other hardware on the bus, each antenna experienced different occlusions within their field of regard. This restricted antenna choice and added complexity when planning the network.

Ground station deployments required physically secure locations proximate to our service region, but also with access to reliable power and cost effective backhaul. Transceivers were provisioned with higher performance radio systems, mounted within radomes on rooftops or other areas with an expansive field of regard. Despite this, ground stations still experienced occlusions from geological formations, structures and tall trees due to the low pointing elevations required when forming long distance B2G links.

E band (including 5G) transmissions attenuate in the presence of atmospheric moisture such as rain, clouds, or fog [23]. Even if an antenna had line of sight, transient RF attenuation could prevent link formation, degrade link capacity, or cause links to fail. Rain and clouds primarily affected B2G connections and were seasonally prevalent in our tropical service regions. This is significantly more detrimental than the rain fade of Ka and Ku bands used by many satellites for space-to-ground communications. Similar to ISLs, the B2B links typically formed at altitudes above significant weather and atmospheric attenuation.

Despite these challenges, ground stations were able to reliably establish B2G links with balloons at a slant-range (i.e. line-of-sight distance) of 130 km under good weather conditions and maintain them to 250+ km. Balloons were able to establish B2B connections at ranges of 500+ km with a maximum range of 700+ km.

Earlier versions of the bus also utilized dual band (2.4GHz and 5GHz) Wi-Fi 802.11n with long-range, ground angled, fixed antennas for B2G communication. This had different tradeoffs than the E band solution - shorter range, no antenna pointing, less impact from atmospheric attenuation, lower power, and unlicensed spectrum. However, the antenna modifications and protocol tuning necessary to establish even 20-100 km links added unacceptable operational complexity given the reduced reliability and insufficient bandwidth for LTE data plane use.

Command & Control. Loon balloons required a reliable command and control channel for navigation and safety critical operations. Due to the indispensable nature of this channel we chose two commercial satellite communications (satcom) providers (one GEO, one LEO) to create redundant, out-of-band connectivity. The choice of satcom providers was primarily motivated by the requirements of stratospheric vehicle navigation. The FMS could task balloons with hundreds of altitude changes per day, but could tolerate 1-2 minutes one-way latency. The specific satcom providers and services chosen were therefore selected from offerings designed around IoT over satellite – at the lowest cost per message. Alternative satcom offerings with <1sec one-way latency were available at higher cost.

These channels were also key to bootstrapping the communication network. While they provided reliable reachability to the balloons, these channels were expensive and extremely limited in both bandwidth and latency performance. To avoid channel overload we typically were limited to sending less than one 1 KiB message per minute per balloon with multi-minute one-way latency.

The out-of-band paths were complemented by the use of in-band paths constituted by the backhaul mesh network itself. This high bandwidth, low latency connection allowed for exfiltration of log files, high rate telemetry, and interactive debugging. The in-band path was typically only available for a small fraction of any balloon's lifetime, even for balloons in our production service fleet. Poor position, nightly depletion of power, or weather events could all prevent in-band control channels.

We also prototyped a one-hop LoRaWAN [1] device with 350 km of simulated range, and were able to establish bootstrapping links. While never deployed in production, a technology like this would have enabled us to improve the speed and consistency with which shorter bootstrap links could be formed. However, this approach did not have the range to match our longer E band links, meaning that satcom would still be required as a backstop.

2.3 Architecture Overview

To address the challenges above, Loon developed a Temporospatial SDN controller dubbed "Minkowski" that was responsible for centrally planning and actuating the backhaul network. The TS-SDN determined radio resource allocations and wireless link selection, in addition to the other traditional SDN duties such as routing and virtual network function configuration. Core to the TS-SDN was the ability to model the 3-D geometry and RF propagation of the physical world and to anticipate changes as nodes moved and atmospheric conditions changed over time.

The TS-SDN shared the control channels employed by Loon's FMS. Due to its ubiquitous reachability, satcom channels were used to bootstrap the initial E band links to add balloons into the network. Once connected, balloon-based routers established in-band control routes to the ground stations using a MANET protocol. Using the ground station as a gateway, balloons were then able to reach SDN endpoints at Loon's EC. Once the balloons were connected to the EC, the SDN could begin programming the data plane.

Loon's 4G LTE systems requested backhaul provisioning from the TS-SDN to connect eNodeBs on balloons in the serving region to the carrier's evolved packet core (EPC). The TS-SDN orchestrated

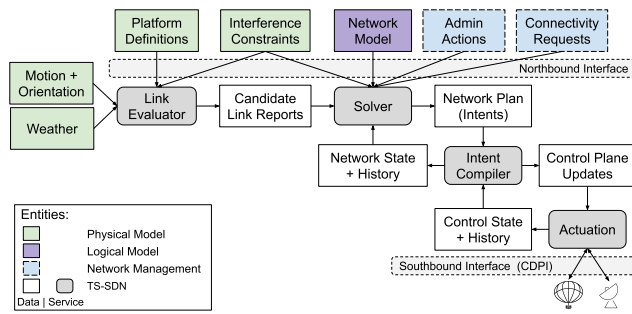


Figure 5: Logical data flow of the TS-SDN.

the topology and implemented the data plane as a network overlay atop the dynamic mesh. This architecture is captured in Figure 3.

3 PREDICTIVE FAILURE RESPONSE

Loon's TS-SDN was engineered to predict failures by modeling the physical environment along with the logical network. Whereas a reactive approach to network failures is the standard in ground-based systems, the dynamic aspects of moving NTN systems increase the chance of failures and can delay both failure detection and recovery. Loon's use of predictive modeling helped the system recover quicker than a reactive-only approach and we planned to take this a step further to avoid most foreseeable failures.

3.1 TS-SDN Design

The TS-SDN aimed to orchestrate the current and future state of the network by forecasting the availability and performance of the physical layer based on 3-D geometry and RF attenuation. Loon's TS-SDN ran in Google datacenters, sending updates to network nodes through a control-data-plane interface (CDPI) [19, 24]. The basic architecture, presented schematically in Figure 5, provided the following services: wireless link modeling, solving for topology and routing, intent generation, sequencing of node updates, and control channel multiplexing. Loon's TS-SDN was built independently. Like other SDN controllers [18, 20], it was programmed with static network entities like interfaces and subnets, and received dynamic route provisioning requests from clients. To model the physical and link layers, it also stored available radio parameters and antenna properties, the 3-D positions and trajectories of platforms over time, and the 3-D volumes of atmospheric conditions and forecasts.

Flight control systems updated balloon positions based on their self-reported GPS location, altitude, and velocity. Trajectory data could also be fed in from Loon's FMS which predicted future positions. Weather data was ingested in real time from rain gauges deployed at GS sites. ECMWF weather forecasts were consumed and processed every 12 hours, upon update. Backstopping these weather data was the ITU-R regional-seasonal atmospheric model [29].

A Link Evaluator component within the TS-SDN continuously analyzed candidate links between all pairs of transceivers at multiple time steps in the future, up to a configurable time horizon. For each pair of antennas, field-of-view and line-of-sight evaluation pruned candidates incapable of satisfying geometric pointing

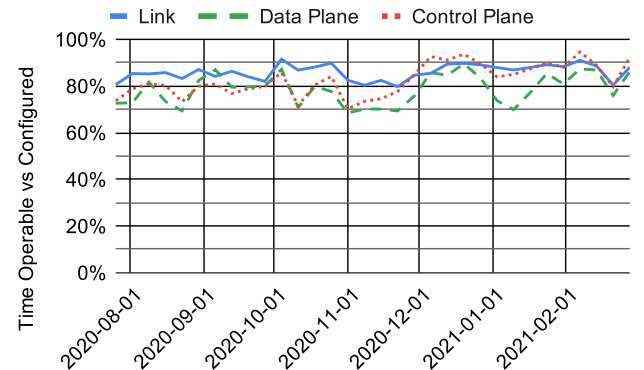


Figure 6: Aggregated node-level reachability metrics.

constraints. For each RF band, the attenuation along the transmission vector was computed, based on an evaluation of free space loss, atmospheric absorption, and moisture attenuation according to ITU-R models [27–29]. For each transmit power level available, transmit and receive antenna gain patterns were used to compute the maximum bitrate with acceptable link margin (specified as a configuration parameter) or the expected link margin for minimal bitrate. The computation was highly parallelizable and distributed across many tasks in a data center. Additionally, the time to compute each report was reduced by caching or precomputing attenuation values for volumes of the atmosphere, and then assembling them using 4-D linear interpolation. To account for uncertainty in our modeling, links just below the acceptable margin were retained and annotated as "marginal". Marginal links were penalized during solving, but attempted when no acceptable links were available.

With these candidate link inputs, a Solver generated intent-based plans for radio resources, topology, and IPv6 src-dest routing policy to form a mesh which considered the feasibility and expected throughput of radio links, the requested backhaul capacity to each node, interference avoidance, and the redundancy of the network. Note that neither link reliability nor duration were optimization targets in this version of the system but could be promising objectives. An actuation component compiled intents into desired per-node configuration, continuously monitored node state, and dispatched commands using the CPDI to align node behavior with the desired intents. Though planned, the Solver and actuation layer lacked the sequencing of updates to avoid temporary routing blackholes.

3.2 Service Performance

The solutions enacted in NTNs are sensitive to change. The fading of a single critical link in the network could prompt a complete overhaul of the desired topology. NTNs must balance adding stability to the topology through the selection of long-lived links with frequent reconfigurations for higher network resilience and utility. In our experience, link reconfigurations were risky as they failed often and had high recovery costs. We biased toward the selection of high utility links and dampened the rate of change by biasing toward topologies that kept established links.

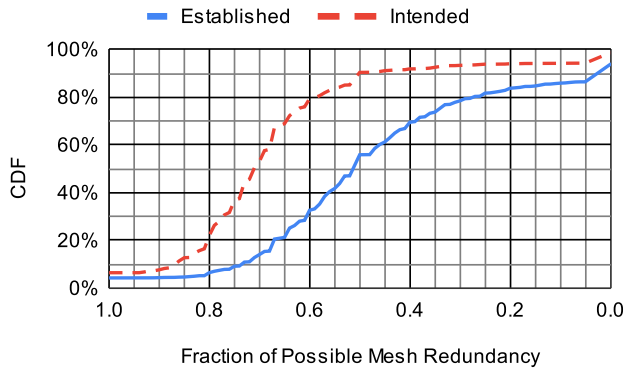


Figure 7: The cumulative distribution of time that redundant links were intended vs established.

Node Reachability. Figure 6 shows the three core component measures of availability for Loon’s network: link layer, in-band control plane (i.e. MANET-routed path from balloon to EC), and data plane (i.e. SDN-configured route from balloon to EPC). Each line reports the ratio of time that the layer was successfully operable over the total potential operable time. For the link layer, for example, we report the fraction of time that the link is installed over the time from the first link establishment command to the withdrawal of the link’s intent.

To a first order, the layers were dependent on one another: data plane connectivity required an operable control layer, and control plane connectivity required an operable link layer. General layering of availability appears clearly before December 2020: the link layer with highest availability and the data plane with the lowest. The dependency layering, however, is not strict. Starting in December 2020, Loon’s TS-SDN could construct a mesh whose in-band control plane connectivity routinely exceeded its link layer reliability. This was due to the rapid recovery from unplanned failures enabled by the establishment of redundant links in the mesh paired with the MANET routing protocol. We expect that with additional improvements, this effect would have been extended to the data plane, with its traffic engineering features and requirements.

Redundancy. Redundant links were critical to improving recovery time. Provisioning balloons with 3 E band antennas proved to be very successful. Not only did this provide redundancy from hardware failures, but it also provided up to 50% additional links to our mesh (see Appendix A). Simulations of 4 or more E band transceivers per node showed diminishing returns that did not justify the added costs.

As a secondary goal, the TS-SDN added redundant links using otherwise idle E band transceivers to enable faster failover. From Figure 7, we observe that 14% of the time the established mesh had no redundancy (e.g. any link failure will disconnect one or more balloons). However, at median, meshes utilize 53% of available transceivers to create additional links - adding 5.5 redundant links. While this is lower than the intended level of redundancy (70% of available transceivers at median), these additional links frequently allowed our in-band control plane to maintain connectivity to the

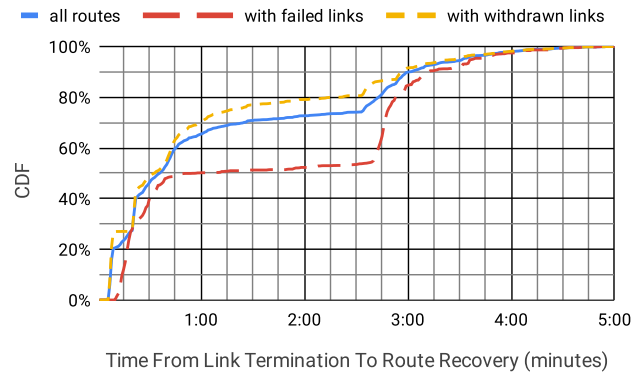


Figure 8: The time to repair broken routes that failed and re-established within 5 minutes.

TS-SDN without requiring that new links be formed, speeding recovery when links failed. We planned to promote topological robustness to be one of the primary optimization objectives in the second version of the solver.

Route Recovery. Routes repair faster when link termination is planned. Figure 8 shows how quickly the TS-SDN was able to recover programmed data plane reachability to individual balloons in the face of anticipated (withdrawn) or unexpected (failed) link termination. Here we look at broken routes which recovered within 5 minutes, representing 45% of all recovered routes. Of these broken route recoveries, 2.9x more co-occurred with withdrawn links than with failed links. We expect that these route-breaking link withdrawals were in anticipation of degrading link quality (e.g. motion leading to line-of-sight occlusions, or out of link range, or areas of increasing rainfall) or to re-optimize the topology toward a higher utility configuration, but we are not able to attribute and quantify the relative frequencies.

Due to the level of redundancy in the mesh and our use of AODV, 75% of recovered routes had control plane breakages of less than 20 seconds and 92.4% of these broken routes recovered without installing a new link. This speaks to the effectiveness of redundant link selection by the Solver, but also indicates that many of these disruptions could have been avoided either using seamless rerouting, pre-programming backup paths, or employing a more traditional routing algorithm such as destination-based routing with source-destination routes overriding as needed. Despite the shortcomings of our implementation, we observe that anticipating link failures consistently improves recovery, restoring network connectivity in 37.8% less time on average.

Takeaway. Reactive recovery mechanisms are always needed in NTN as it is impossible to predict all failures. In our experience, MANET routing protocols were very effective in adapting to topology changes, in turn speeding recovery. Tasking idle transceivers to provide redundancy was a good trade off. Adding even a small number of additional links to the established network pushed our control plane availability above our link level availability.

Given the propensity to disruption and potentially long recovery times, we recommend that network engineers consider incorporating predictive approaches into NTN coordination. While we saw benefit from predictive recovery, we expect that with more development a significant fraction of network breakages could eventually be avoided entirely.

4 HYBRID CONTROL PLANES

Traditional SDN control planes typically assume highly available, reliable, in-order delivery of messages [19]. However, the control channels available to NTNs may have severely constrained availability, latency, or bandwidth. Loon utilized multiple control channels to improve availability and performance, but faced unforeseen challenges coordinating nodes.

4.1 Control Planes

The TS-SDN controller configured the balloon routers using a hierarchy of three control planes, shown in Figure 3. Each balloon control plane tier was progressively more capable than its predecessor, with command latencies moving from minutes down to milliseconds. However, as capability went up, reliability went down.

Balloon Tier 0: Satcom. To bootstrap a disconnected balloon into the network, the base control plane utilized two commercial satellite networks. These highly available channels primarily served safety critical flight operation functions, but the TS-SDN passed ~1KB messages by filling unused slots. With latencies up to minutes, this control plane was only used when higher-performance control planes were unavailable.

The TS-SDN, having computed a topology for some time in the future, would contact a satellite message relay service and pass the minimum amount of information necessary to a balloon node to initiate a link with another balloon or a ground station. An analogous message would be sent to the peer platform, possibly over a different channel. These messages contained a future enactment timestamp, anticipated pointing geometry, transmit and receive channel characteristics, and the identity of the intended peer. Messages were cryptographically signed with key material unique to each balloon to ensure integrity.

Balloon Tier 1: MANET. Once link-layer connectivity was established, Loon used batman-adv [40], an AODV-based protocol [14], to route control plane messages. The ad-hoc routing domain spanned from ground stations up to balloons and among connected balloons.

The primary purpose of this control plane was to allow each balloon router to establish a gRPC [22] connection to a TS-SDN controller endpoint in an EC pod and to maintain that connectivity despite link failures. As long as some path through the mesh existed, batman-adv could repair mesh routing faster than the datacenter-based TS-SDN could react, especially given the highly dynamic environment and limitations of satcom paths. Using this higher performance control plane, the TS-SDN completed its programming of the data plane and received high-rate telemetry.

Balloon Tier 2: SDN. The network architecture and the TS-SDN enabled the ability to program the in-band data plane in response to requests from node management and LTE control applications. Data-plane forwarding for any of the balloon's dedicated /64 IPv6

addresses could be configured and applications could choose which route to use (SDN or MANET) by setting a suitable source address. The SDN-programmed path provided the management and data planes for the LTE eNodeBs and for other attached services (e.g. bandwidth reservations to download large OS updates).

Ground Station Wired Access. Ground Stations were deployed with wired access to at least one EC installation, connected using either a virtual circuit within a partner carrier's network or, most often, over the Internet. With basic connectivity provisioned (IP addressing, routing, and DNS), ground stations reused the same gRPC-based control plane applications as the balloons to establish a secure, authenticated connection to an SDN endpoint. The SDN would enact IPsec tunnels between the GS and requested EC pods, and then program routes between balloon nodes and ECs over these IPsec links as needed.

4.2 Control Plane Composition

Based on the idea of a Control-to-Dataplane Interface (CDPI) from OpenFlow [36], we added extensions to address the challenges of multiple channels and a moving NTN. Our approach exceeded the performance of a prohibitively slow satcom-only design, but introduced significant complexity.

Channel Selection. Existing CDPI protocols allow for multiple control channels to each node, a primary channel and auxiliary channels for better throughput [19], with each command sent over a single channel. Likewise, Loon maintained multiple control channels (2 satcom, 1 in-band) to each balloon, but distributed messages across them without assigning special semantics. The TS-SDN's CDPI frontend assumed some satcom path was available and tracked in-band node reachability using heartbeats transmitted on the gRPC connection from balloons. Similar to other resource-constrained networks [47], the TS-SDN monitored connectivity and directed messages along the lowest latency path. For satcom channels, Loon implemented a CDPI proxy to bitpack messages and a satcom gateway service to route messages using the network with lowest expected delivery time. If in-band was not available to all recipient nodes, then commands enacting an intent used a combination of in-band and satcom paths. Command delivery was tracked and retried as necessary, potentially using a different delivery path.

Figure 9 shows the observed round trip command latency from submission into the satcom transmit queue until an ACK was received from the payload. In the best case, satcom round-trip latency could be as little as 23 seconds, but combined across our two providers, was 1m27s at the median, 5m47s at the 90th percentile and 14m50s at the 99th percentile. In contrast, the in-band control plane offered up to 987 Mbps of bandwidth with sub-second round-trip latency at the median, 2 seconds at the 90th percentile, and 23 seconds at the 99th percentile.

Time to Enact. Timely topology changes in NTNs are critical for maintaining mesh connectivity and an in-band control plane. As the position of nodes and the viability of links changes, nodes need to converge quickly on a new topology and new routing paths. However, control plane messages may reach nodes at different times, causing some nodes to switch to the new topology while

others remain in the old. Further, the formation of moving point-to-point wireless links requires synchronizing the endpoints to search for each other. In the Loon implementation, this process could take dozens of seconds. To avoid these costly failures, Loon added a "time to enact" (TTE) parameter to CDPI commands to allow nodes to begin topology changes at a consistent time using GPS synchronized clocks.

In-band Side Channel. Traditional SDN controllers expect command responses to be returned over the same channel as the request. However, due to the lower latency of Loon's in-band channel, the TS-SDN was able to quickly infer the result of a command by the presence of in-band CDPI connections. For example, when a balloon was bootstrapped into the mesh, the link establishment command for the connecting balloon and its response would be sent over satcom. However, upon successfully connecting to the mesh, the balloon's SDN agent would immediately establish an in-band connection to the TS-SDN using the batman-adv routed path. This connection request would typically reach the CDPI frontend many seconds before the satcom response arrived, allowing the TS-SDN to infer that the link establishment had succeeded and proceed to program routes to the balloon.

Message Queuing. Existing SDN CDPI protocols do not consider sustained channel congestion nor the desired behavior when messages queue. Due to the limited throughput of our satcom channels, the CDPI messages were queued in the satcom gateway. Further, radio reboots and antenna slewing caused long link acquisition times, blocking enactment of dependent commands. To reduce contention on satcom channels, the CPDI proxy and satcom gateway dropped CPDI messages that 1) would not arrive by the TTE or 2) required in-band connectivity (e.g. forwarding table updates). We relied on the TS-SDN's timeout to retry commands. With multiple parallel control channels, the filtering and prioritization of messages could result in out-of-order delivery.

Challenges. Choosing a TTE that allowed command delivery to all nodes, but did not cause unneeded delay, was challenging. The TS-SDN set the TTE based on the available control channels. For commands using satcom, the 95th percentile of one-way command delivery delay was added to the TTE. If in-band paths were available to all updating nodes, then a three-second delay was added. Not only did the TS-SDN have to consider the channels available to the destination node, but it also had to consider the channels available to all other nodes receiving a command as part of the same intent enactment and set the TTE to the longest delay. The queue depth within the satcom gateway was not visible when setting the TTE and further complicated choosing a TTE value, especially across multiple destination node channels. Ideally, TTE and transmission time would consider the queue depth on all affected nodes.

Once TTE was chosen, it could not be updated. At times the in-band paths flapped on and off. If the TTE was set when only satcom was available, but then in-band paths to all destinations appeared shortly after, we did not upgrade the TTE and retransmit the commands on the faster path. In such a case, one would like the ability to do a quick upgrade of the time-to-enact and retransmit on the high-performance control channel to avoid the long delay.

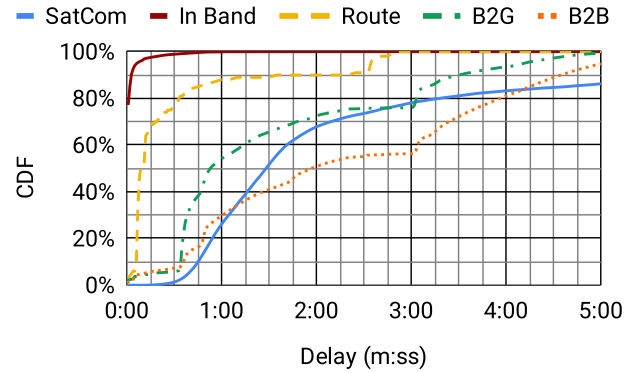


Figure 9: Distribution of time for successful enactments of Link and Route intents versus the round-trip time of control channels.

Since the time needed to enact different commands varied, the TS-SDN set timeouts based on the command type and the channel used. When the TS-SDN didn't get a response back, it cycled through the available channels based on priority, set a new TTE, and retried the command. However, for slow commands like link formation, this resulted in additional delay if the message was dropped from the satcom queue or arrived after its TTE. Mechanisms to promptly notify the TS-SDN of a discarded message would have allowed the TS-SDN to retry sooner, saving valuable time. Further, commands dropped by the satcom gateway should have fanned out to also drop any other related, queued commands. The TS-SDN had no way to cancel enqueued satcom messages that it knew could not succeed.

Performance. Figure 9 highlights the improvement from combining in-band and satcom control channels. If we had used a trivial satcom-only approach, all commands should propagate, enact and report their success no faster than the satcom RTT delay. Adding the in-band channel improved performance for all types of intents. When adding links between in-band accessible nodes, the delay should be dominated by any radio boot up and antenna search time (up to 2m30s). In contrast, any command sent over satcom incurs an extra 3m6s TTE delay to account for 1-way satcom latency.

While including in-band was a large improvement, significant gains were left unrealized. Route updates should have always been sent over in-band channels and quickly enacted, but we observe increased delay in the tail of the distribution possibly due to 1) 3 second in-band TTE, 2) waiting for batman-adv to reconverge, 3) waiting for broken links to reestablish, or 4) incorrect selection of the satcom channel. We expect that similar issues also affected some fraction of link formation commands. In the ideal case, the TS-SDN would sequence routing and topology changes across balloons to maintain balloons' connectivity to the mesh. This would preserve data plane availability and maximize the commands delivered in-band.

Takeaway. The amalgamation of multiple channels was effective at improving the control plane availability and performance beyond

the level offered by any single channel. However, Our satellite solution was unable to meet the performance requirements of the control plane, and we introduced onerous design complexity to overcome its limitations. Based on this experience and current market offerings, we would recommend leveraging lower latency satellite solutions or long range broadcast solutions like LoRaWAN.

5 LONG RANGE POINT-TO-POINT LINKS

Instantiating mesh topologies using long distance, directional point-to-point antennas presents a set of interesting challenges, especially for moving platforms. In a moving networked system, practically all elements change over time and their properties must either be explicitly modeled or detected reactively. While the need for good models of the physical environment can't be overstated, systems also need to detect and adapt when their observations contradict the modeled conditions.

In the face of changes on short timescales, Loon built support into the TS-SDN to model vehicle motion and to enact commands at a specific point in the future, allowing it to exploit projected pointing vectors and expected signal strengths. As with many systems that depend on sky-to-ground point-to-point RF links, weather analysis and forecasting was another particularly active area of exploration and investment. For managing change over longer timescales—obstruction due to new construction or seasonal variation like foliage growth—the system needed to be able to detect when the models of the physical world had gone stale and update them.

Model Validation. Network telemetry data were used to validate and identify models that had gone stale. For example, when installing a new ground station, the site engineers built an obstruction model taking into account surrounding terrain and buildings. These obstruction masks required updating as new buildings rose up. Given the remoteness of some of our ground station sites, sending a technician to periodically recheck our model of the physical surroundings was impractical. Instead, we built tooling to correlate historical link telemetry with antenna pointing vectors to detect stale obstruction masks. A screenshot is shown in Figure 13 in the Appendix. Identification of a systematic skew in the RF measurements and model expectations would trigger remedial action (for example, model update, hardware debugging, site resurvey). Analyzing low-level network state was an effective means to handle the reality of even our "static" models changing over time.

Weather & Data Freshness. Due to our point-to-point links' sensitivity to atmospheric moisture, the Loon team incorporated weather data into the TS-SDN. Discrepancies in the modeled and actual weather, or in signal propagation effects in the field, could have a material impact on the solver's outcome. Thus, we evolved the system to prioritize data freshness when considering solver inputs. For example, preferring weather data from ground station sensors and real time network telemetry proved more accurate than relying on weather forecasts alone. The goals were fourfold: 1) Increase the likelihood of successfully establishing a chosen link, 2) Reduce the link acquisition time by avoiding attenuation-driven retries, 3) Ensure that chosen links could be expected to have a good bit rate and link margin, and 4) Lengthen link lifetime durations by reducing unplanned link failure due to weather.

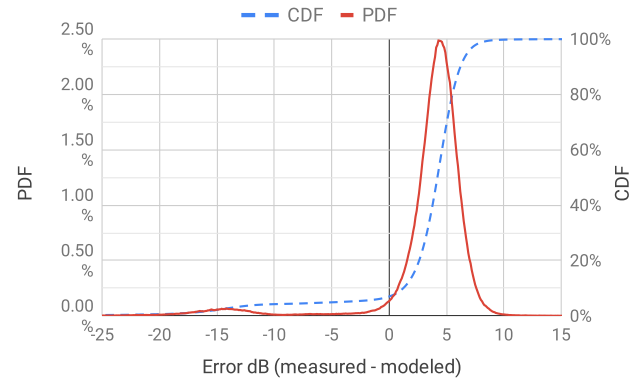


Figure 10: Plot of error between measured and modeled channel attenuation for installed B2B links.

The fundamental approach was to improve the weather data available to the link margin and bitrate estimators in order to provide the Solver with a more accurate model for each pair of transceivers that had line of sight. The variety of manifestations of atmospheric water vapor motivates the need to store and index weather parameters as spatial volumes projected forward in time. There were three vectors we used in an attempt to improve estimates of moisture attenuation: 1) As noted above, we used ITU-R regional, seasonal estimates; 2) We installed weather gauges at ground station sites to provide real time data to the system; and 3) We consumed and incorporated ECMWF forecasts of the weather, both stratospheric and tropospheric.

We invested substantial effort on incorporating weather forecasts into the system, and, given its marginal utility relative to ground station rain gauges, would likely have taken a different approach if we were to do it again. With the seasonal prevalence of thunderstorms in our subtropical service region, the forecasts didn't have sufficient accuracy and fidelity to be relied upon and were not a large improvement over probabilistic models derived from ITU regional and seasonal averages. Instead we would lean further into fusing various weather data using Gaussian Mixture Models [34], feeding observed signal strength measurements as data points back into our weather model, using redundant links to explore and exploit uncertain transmit vectors, and placing local weather radar equipment near the service region to improve the fidelity of atmospheric data. Given the high-degree of moisture-based attenuation caused to many RF bands, we still believe that real-time volumetric weather data is valuable to incorporate into an NTN network solver, but requires more sophistication and better sensors than we brought to bear.

Modeled Link Quality. Figure 10 shows that the modeled and measured signal strength of B2B links rarely matched. Some of this was expected as we intentionally selected a pessimistic level from the ITU-R regional seasonal average model to increase confidence in forming the selected links. This is clearly visible in the 4.3 dB right-shift (more signal measured by the radios than modeled) in the graph. To further increase the chance of successful link formation, Loon deprioritized links within 5dB of the minimum signal strength.

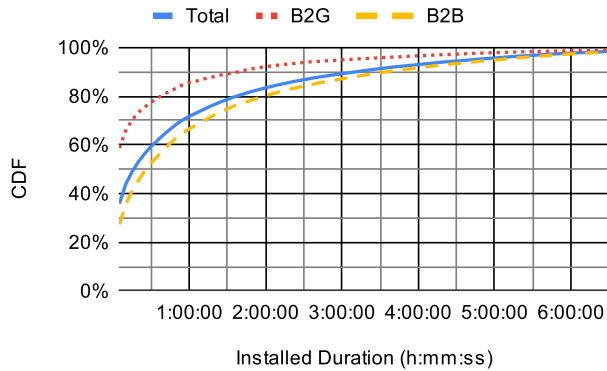


Figure 11: Distribution of link lifetimes.

However, these "marginal" links could still be used if no better options were available.

There is also a visible bump around -14dB , which we suspect mostly represents locking on to side lobes of the antenna pattern. The long tails consist largely of inaccurate weather predictions. The cases where the model was significantly overestimating the quality of the link caused significant operational load to differentiate between link failure due to possible hardware issues like antenna pointing offsets and unrecognized atmospheric conditions. We were focused heavily on further improving the ingestion pipeline for weather data to reduce these errors, as well as limiting their impact by feeding back observed link data into the solver.

Link Lifetime. In our experience, B2G and B2B links differed significantly in formation rate, failure rate, and longevity. We found that 51% of B2G and 40% of B2B links succeeded in their first attempt, but success on retries diminished quickly with 95% of installed links succeeding within 2 and 3 attempts for B2G and B2B, respectively. In both cases 35% of links never succeeded. Since Loon's TS-SDN lacked a feedback loop and relied on modeled data for network planning, links were retried repeatedly. A better policy would have adapted to failures and tried an alternate link if one existed.

Figure 11 shows the lifetime of B2G and B2B links. B2G links were more prone to disruption than B2B with a median duration of 1m45s for B2G links versus 25m55s for B2B links. B2B links were subject to less weather effect, fewer obstructions, and, as a side effect of wind based navigation, both endpoints tended to have correlated motion. Indeed, fully 44.8% of B2G links lasted for shorter than 1 minute. B2B links fared somewhat better though still experienced a significant early mortality rate of 15.0%.

High link mortality rates speak to the need for the TS-SDN to choose links that have both a higher probability of succeeding and are modeled to have a long installed duration. Loon did not implement this paired optimization criteria and instead relied on optimizing for an instant a few minutes in the future. Our relatively slow moving nodes afforded us a window to act and adapt to maintain connectivity, rather than explicitly optimizing for duration.

Recall from Figure 8 that we were able to recover data plane routes significantly more quickly when the TS-SDN requested link

teardown versus reacting to unexpected failures. Looking at the observed end state of all installed links, we found that approximately half (47.4%) failed unexpectedly, but unexpected failures were more prevalent in B2G links (69.2%) vs B2B links (39.2%). This further demonstrates the increased impact of atmospheric phenomena like clouds, and unmodeled obstructions like new construction and seasonal foliage near the ground. It highlights the more brittle nature of ground-terminated links, and our limited success modeling all impacting phenomena.

Takeaway. Physical models are an important part of NTN network planning and management, but are imperfect approximations of the real world. In our experience, physical models differed from our observed measurements in a few key ways: 1) Errors due to inaccurate inputs (e.g. balloon trajectory estimates), 2) Errors due to the limited precision of inputs (e.g. coarse temporal & spatial granularity of weather inputs), 3) Fidelity of the model's approximation (e.g. quantized representations of antenna gain patterns), 4) Uncharacterized variables that were assumed to be static (e.g. new obstructions, hardware performance degradation).

From our experience, we offer a few insights. 1) Network telemetry is a useful mechanism for making observations of the physical environment and is useful for cross-validating and improving physical models, potentially in real time. 2) While systems can be built to account for noise and uncertainty, flagging significant deviations to network operations engineers is an important aspect of detecting and responding to field anomalies. 3) Network orchestration needs to bias away from model-based solutions when real-time observations provide conflicting signals. 4) Onboard feedback loops to correct/recalibrate are important for inaccessible, commercial off the shelf hardware to maintain its performance over its lifetime in a harsh environment (i.e. ~ 300 days in the stratosphere).

6 EXPLAINABILITY

Debugging Loon's production network was a difficult undertaking. The combination of time- and space-dynamic processes that made the TS-SDN necessary also made it hard to reason about the results. There were three primary facets: 1) **What** is the state of the system? 2) **How** did the system reach this state? 3) **Why** did the TS-SDN choose this specific topological configuration?

The TS-SDN's centralized architecture made some aspects of visibility straightforward. It was easy, for example, to directly inspect the current and historical records of network intents and network telemetry. It was impossible, however, for the system to provide consistent and timely information about a disconnected node's state. The stratospheric operating environment and the ever present uncertainty of weather conditions further increased the challenge, even beyond the burden of debugging data center systems without physical access [9]. In addition to limited real-time access through satcom (and no physical visibility), intermittent reachability meant that we often only had retroactive access to logs (i.e. after a node rejoined the mesh).

Loon built visualization tools that captured both physical and logical views of the network. Critically, these added visibility into the relationships between layers of the system, making it easier to connect top-level connectivity problems to low-level failures and vice versa. Adding a time dimension to our visualization and

debugging tools – filtered change-log style views, and a scrubber enabling us to roll time backwards and forward – further enabled us to understand **how** the system reached a particular state.

Even with visibility into the current state and how we got there, there was still a broad gulf to explain **why** the TS-SDN chose a particular configuration. There were a number of reasons why this was so hard to understand.

The Solver, though deterministic, was iterative and dependent on a large set of asynchronous data sources. Its topological solutions were time dependent, and able to change as they gradually enacted across a sequence of solve cycles. Many subsystems worked incrementally, cascading changes across time. Limited versioning and data provenance information restricted our ability to reproduce system states for further debugging. System configurations were time ordering dependent. The links that the Solver had to consider were different if one balloon happened to come into range slightly before another. The solver applied hysteresis to bias toward keeping existing links, moderating the aggregate rate of change in the network (i.e., limiting the effects of slow link acquisition). This made it difficult for human operators to predict or understand the end state that the Solver was working towards. Without understanding the full history and predicted future at each time step of its evolution, it was difficult to understand why a given mesh configuration existed.

Further, there were many dimensions and constraints considered in solving that were invisible in the realized mesh configuration. This led operators to second guess the solver and frequently ask "**why not...**". What was not clear was whether such proposed solutions were possible (e.g. didn't have unseen geometric or RF-based constraints), the amount of disruption that would have been incurred to reach the configuration, and whether it would lead to higher network utility over time. Adding such properties to visualization tools was challenging but critical, as their absence made it difficult to reason about "correct" system behavior versus "bugs".

Given the challenges a simple, handcrafted solver presented, we anticipated these issues would grow substantially harder with subsequent generations of solvers based on Mixed Integer Programming (MIP) and Reinforcement Learning (RL). Explainability remains a broadly open problem, but we provide a few observations and recommendations. 1) Solutions that differ based on the order of events are implicitly harder to reason about. Take care to log comprehensively to enable tracing of path dependent effects. 2) Design solvers and their inputs in a way that enables the reproducibility of network commands in tests and post-hoc analysis. 3) Put individual changes in context by surfacing a near-term goal state from the solver, and the expected sequence of intents to reach it. 4) Improve confidence in solver adjustments by identifying a metric for the value of each given network solution. 5) The value of a good data visualization design should not be underestimated. It empowers network operations to answer "why not" questions, find bugs, and build confidence in correct behavior.

7 FUTURE WORK

Loon's TS-SDN optimized for node connectivity and maximizing bitrate, but many properties like fault tolerance and disruption-free network evolution are also desirable. Work is needed on problem formulations and solving approaches which can express nonlinear

tradeoffs between multiple objectives. Metrics are needed to rate both a network configuration as well as the sequence of steps used to arrive there. TS-SDN modeling and solvers should be extended to use an explicit mapping of shared fate elements and failure domains, and differentiate between types of nodes (airborne, ground, maritime) to exploit node specific capabilities. Further, benefit might be found in modeling link characteristics probabilistically. For example, conditioning link selection on physical models augmented with enactment success rate, link duration, and signal strength measurements would improve performance in a number of dimensions.

The TS-SDN approach could be adapted to other domains with physically or electronically steered beams and moving nodes. Potential applications include flexibly routed and dynamic capacity satellite to ground links, ISL-connected satellite constellations, or traditional aviation and maritime meshes.

As TS-SDN adoption grows, work will be needed to define architectures in which TS-SDN instances, and the assets they control, can coexist and even interoperate with each other. This might be accomplished by delegating control of assets, offering NTN transit as a service, or running multiple TS-SDN instances in hierarchical or federated arrangements.

8 CONCLUSIONS

Loon's experience shows that incorporating a model of the physical world onto the TS-SDN's logical network planning decreased average recovery time for routes recovering within 5 minutes by 37.8% relative to a strictly reactive approach. However, models of the physical world are limited and are sensitive to data quality and freshness problems. For example, integrating weather forecasts into RF link modeling yielded only marginal gains. The analysis of discrepancies between modeled and measured data was critical to identifying model and data quality degradation. This method successfully detected new obstructions near our ground stations, and we anticipated substantial improvements from control loops that would incorporate observed data in link selection.

Loon's hybrid control plane, composed of satcom and in-band channels, enabled a valuable trade-off between availability and performance. Using a MANET protocol to implement mesh redundancy and in-band control channel routing was a good choice that resulted in command enactment substantially faster than a satcom-only design. However, the implementation required several extensions to the TS-SDN's control-to-data-plane interface, which added significant complexity to the overall system.

Debugging the network was exceptionally difficult given the dynamic environment and limited communication channels. Visualizing the system's logical and physical evolution aided in understanding the current state of the network, but solver improvements were needed to associate individual commands with planned topological evolution and service-level objectives.

While there is still ample opportunity for improvement, our experience suggests that TS-SDN is a compelling architecture for orchestrating networks of moving platforms and steerable beams. Loon's TS-SDN proved itself instrumental in enabling an autonomous and dynamic network which allowed us to serve hundreds of thousands of users in remote areas around the world.

ACKNOWLEDGMENTS

This paper represents the collective accomplishments of dozens of collaborators, colleagues, and friends at Loon. Thank you to Filip Zivkovic and Ryan Endacott for their assistance in curating data sets presented here and to Mark Schlagenhaut for his expert copy editing. We would also like to thank Sal Candido, Paul Heninwolf, Scott Moeller, Justin Lannin, & Jeff Mogul for their candid feedback and valuable suggestions. Thank you also to our shepherd, Shyam Gollakota, and our anonymous referees for their comments and guidance.

REFERENCES

- Ferran Adelantado, Xavier Vilajosana, Pere Tuset-Peiro, Borja Martinez, Joan Melia-Segui, and Thomas Watteyne. 2017. Understanding the Limits of LoRaWAN. *IEEE Communications Magazine* 55, 9 (2017), 34–40. <https://doi.org/10.1109/MCOM.2017.1600613>
- Talal Ahmad, Ranveer Chandra, Ashish Kapoor, Michael Daum, and Eric Horvitz. 2017. Wi-fly: Widespread opportunistic connectivity via commercial air transport. *HotNets 2017 - Proceedings of the 16th ACM Workshop on Hot Topics in Networks* (2017), 43–49. <https://doi.org/10.1145/3152434.3152458>
- Aswin Alexander, Marc Alvidrez, Wajahat Beg, Zoe Benezet-Parsons, Léonard Bouygues, Umit Dogruer, Ben Freedman, Jon Grazer, Paul Heninwolf, Derek Herbert, Bryan Ho, Rick Lange, Marc Lelièvre, Don Nguyen, Jonathan Nutzmann, Ken Riordan, Kevin Roach, John Rousseau, Robert Schlaefli, and Sam Truslow. 2021. Loon Library: Lessons from Building Loon's Stratospheric Communications Service. (2021). <https://x.company/projects/loon/the-loon-collection/>.
- Sharath Ananth, Ben Wojtowicz, Alfred Cohen, Nidhi Gulia, Arunoday Bhat-tacharya, and Brian Fox. 2019. System design of the physical layer for Loon's high-altitude platform. *EURASIP Journal on Wireless Communications and Net-working* 2019, 1 (2019), 1–17.
- D. Anipko. 2015. *Multiple Provisioning Domain Architecture*. Technical Report. RFC 7556.
- Gary Atkinson, Xiang Liu, R Nagarajan, S Parekh, and Xiangpeng Jing. 2005. Dynamic topology control in ad hoc networks with directional links. In *MILCOM 2005-2005 IEEE Military Communications Conference*. IEEE, 543–549.
- Brian Barritt and Wesley Eddy. 2015. *Temporospatial SDN for Aerospace Communications*. <https://doi.org/10.2514/6.2015-4656> arXiv:<https://arc.aiaa.org/doi/pdf/10.2514/6.2015-4656>
- Brian Barritt, Tatiana Kichkaylo, Ketan Mandke, Adam Zalcan, and Victor Lin. 2017. Operating a UAV mesh & internet backhaul network using temporospatial SDN. In *2017 IEEE Aerospace Conference*. 1–7. <https://doi.org/10.1109/AERO.2017.7943701>
- Luiz André Barroso, Urs Hölzle, and Parthasarathy Ranganathan. 2018. The data-center as a computer: Designing warehouse-scale machines. *Synthesis Lectures on Computer Architecture* 13, 3 (2018), i–189.
- Marc G Bellemare, Salvatore Candido, Pablo Samuel Castro, Jun Gong, Mar-los C Machado, Subhdeep Moitra, Sameera S Ponda, and Ziyu Wang. 2020. Autonomous navigation of stratospheric balloons using reinforcement learning. *Nature* 588 (2020), 77–82. <https://doi.org/10.1038/s41586-020-2939-8>.
- C. Byrne, D. Drown, and A. Vizdal. 2014. *Extending an IPv6 /64 Prefix from a Third Generation Partnership Project (3GPP) Mobile Interface to a LAN Link*. Technical Report. RFC 7278.
- Mitch Campion, Prakash Ranganathan, and Saleh Faruque. 2018. UAV swarm communication and control architectures: a review. *Journal of Unmanned Vehicle Systems* 7, 2 (2018), 93–106.
- Xianbin Cao, Peng Yang, Mohamed Alzenad, Xing Xi, Dapeng Wu, and Halim Yanikomeroglu. 2018. Airborne Communication Networks: A Survey. *IEEE Journal on Selected Areas in Communications* 36, 9 (2018), 1907–1926. <https://doi.org/10.1109/JSAC.2018.2864423>
- Ian D Chakeres and Elizabeth M Belding-Royer. 2004. AODV routing protocol implementation design. In *24th International Conference on Distributed Computing Systems Workshops, 2004. Proceedings*. IEEE, 698–703.
- Amira Chriki, Haifa Touati, Hichem Snoussi, and Farouk Kamoun. 2019. FANET: Communication, mobility models and security issues. *Computer Networks* 163 (2019), 106877.
- Jonathan Corbet. 2015. *SOCK_DESTROY: an old Android patch aims upstream*. Retrieved 2022-02-02 from <https://lwn.net/Articles/666220/>
- Richard Draves and Dave Thaler. 2005. *Default router preferences and more-specific routes*. Technical Report. RFC 4191.
- Andrew D. Ferguson, Steve Gribble, Chi-Yao Hong, Charles Killian, Waqar Mohsin, Henrik Muehe, Joon Ong, Leon Poutievski, Arjun Singh, Lorenzo Vici-sano, Richard Alimi, Shawn Shuoshuo Chen, Mike Conley, Subhasree Mandal, Karthik Nagaraj, Kondapa Naidu Bollineni, Amr Sabaa, Shidong Zhang, Min Zhu, and Amin Vahdat. 2021. Orion: Google's Software-Defined Networking Control Plane. In *18th USENIX Symposium on Networked Systems Design and Implementation (NSDI 21)*. USENIX Association, 83–98. <https://www.usenix.org/conference/nsdi21/presentation/ferguson>
- Open Networking Foundation. 2015. *OpenFlow Switch Specification*. Retrieved 2022-01-08 from <https://opennetworking.org/wp-content/uploads/2014/10/openflow-switch-v1.5.1.pdf>
- Open Networking Foundation. 2022. *Open Network Operating System (ONOS) SDN Controller for SDN/NFV Solutions*. Retrieved 2022-01-29 from <https://opennetworking.org/onos/>
- Mahanth Gowda, Justin Manweiler, Ashutosh Dhekne, Romit Roy Choudhury, and Justin D. Weisz. 2016. Tracking Drone Orientation with Multiple GPS Receivers. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking* (New York City, New York) (*MobiCom '16*). Association for Computing Machinery, New York, NY, USA, 280–293. <https://doi.org/10.1145/2973750.2973768>
- gRPC authors. 2022. *gRPC*. Retrieved 2022-01-08 from <https://grpc.io>
- J. Din M. J. Alhilali S. L. Jong H. Y. Lam, L. Luini and F. Cuervo. 2017. Impact of rain attenuation on 5G millimeter wave communication systems in equatorial Malaysia investigated through disdrometer data. In *11th European Conference on Antennas and Propagation (EUCAP)*. EUCAP, 1793–1797. doi: 10.23919/EUCAP.2017.7928616.
- Evangelos Haleplidis, Kostas Pentikousis, Spyros Denazis, J Hadi Salim, David Meyer, and Odysseas Koufopavlou. 2015. Software-defined networking (SDN): Layers and architecture terminology. *RFC 7426* (2015).
- Guoyou He. 2002. Destination-sequenced distance vector (DSDV) protocol. *Networking Laboratory, Helsinki University of Technology* 135 (2002).
- Thomas R Henderson, Mathieu Lacage, George F Riley, Craig Dowell, and Joseph Kopena. 2008. Network simulations with the ns-3 simulator. *SIGCOMM demon-stration* 14, 14 (2008), 527.
- ITU-R P.676 2019. *ITU-R P.676: Attenuation by atmospheric gases and related effects*. Standard. International Telecommunication Union (ITU) Radiocommunication Sector (ITU-R), Geneva, CH.
- ITU-R P.838 2005. *ITU-R P.838: Specific attenuation model for rain for use in prediction methods*. Standard. International Telecommunication Union (ITU) Radiocommunication Sector (ITU-R), Geneva, CH.
- ITU-R P.840 2019. *ITU-R P.840: Attenuation due to clouds and fog*. Standard. International Telecommunication Union (ITU) Radiocommunication Sector (ITU-R), Geneva, CH.
- Philippe Jacquet, Paul Muhlethaler, Thomas Clausen, Anis Laouiti, Amir Qayyum, and Laurent Viennot. 2001. Optimized link state routing protocol for ad hoc networks. In *Proceedings. IEEE International Multi Topic Conference, 2001. IEEE INMIC 2001. Technology for the 21st Century*. IEEE, 62–68.
- The kernel development community. 2022. *batman-adv - The Linux Kernel documentation*. Retrieved 2022-01-08 from <https://www.kernel.org/doc/html/v4.15/networking/batman-adv.html>
- Oltjon Kodheli, Eva Lagunas, Nicola Maturo, Shree Krishna Sharma, Bhavani Shankar, Jesus Fabian Mendoza Montoya, Juan Carlos Merlano Duncan, Danilo Spano, Symeon Chatzinotas, Steven Kisseleff, Jorge Querol, Lei Lei, Thang X. Vu, and George Goussetis. 2021. Satellite Communications in the New Space Era: A Survey and Future Challenges. *IEEE Communications Surveys Tutorials* 23, 1 (2021), 70–109. <https://doi.org/10.1109/COMST.2020.3028247>
- Gunes Karabulut Kurt, Mohammad G Khoshkholgh, Safwan Alfattani, Ahmed Ibrahim, Tasneem SJ Darwish, Md Sahabul Alam, Halim Yanikomeroglu, and Abbas Yongacoglu. 2021. A vision and framework for the high altitude platform station (HAPS) networks of the future. *IEEE Communications Surveys & Tutorials* 23, 2 (2021), 729–779.
- Zhengzheng Li. 2011. *Applications of Gaussian mixture model to weather observa-tions*. The University of Oklahoma.
- A. Matsumoto, T. Fujisaki, R. Hiromi, and K. Kanayama. 2008. *Problem Statement for Default Address Selection in Multi-Prefix Environments: Operational Issues of RFC 3484 Default Rules*. Technical Report. RFC 5220.
- Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Pe-terson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. 2008. OpenFlow: Enabling Innovation in Campus Networks. *SIGCOMM CCR* 38, 2 (mar 2008), 69–74. <https://doi.org/10.1145/1355734.1355746>
- Robert Opp. 2021. *The evolving digital divide*. Retrieved July 1, 2022 from <https://www.undp.org/blog/evolving-digital-divide>
- Nils Pachler, Inigo del Portillo, Edward F Crawley, and Bruce G Cameron. 2021. An updated comparison of four low earth orbit satellite constellation systems to provide global broadband. In *2021 IEEE international conference on communications workshops (ICC workshops)*. IEEE, 1–7.
- P. Pfister, E. Vyncke, T. Pauly, D. Schinazi, and W. Shao. 2020. *Discovering Provisioning Domain Names and Data*. Technical Report. RFC 8801.
- Antonio Quartulli, Linus Lüssing, Marek Lindner, Martin Hundebøll, Simon Wunderlich, and Sven Eckelmann. 2022. *WikiStart - Open-Mesh - Open Mesh*. Retrieved 2022-01-08 from <https://www.open-mesh.org/projects/open-mesh/wiki>

- [41] RTCA RTCA. 2002. DO-242A-Minimum Aviation System Performance Standards For Automatic Dependent Surveillance Broadcast (ADS-B), June, 2002. Washington, DC, USA: Radio Technical Commission for Aeronautics (2002).
- [42] Julius Schulz-Zander, Carlos Mayer, Bogdan Ciobotaru, Stefan Schmid, and Anja Feldmann. 2015. OpenSDWN: Programmatic Control over Home and Enterprise WiFi. In *Proceedings of the 1st ACM SIGCOMM Symposium on Software Defined Networking Research* (Santa Clara, California) (SOSR '15). Association for Computing Machinery, New York, NY, USA, Article 16, 12 pages. <https://doi.org/10.1145/2774993.2775002>
- [43] UN Secretary-General. 2020. Road map for digital cooperation: implementation of the recommendations of the High-level Panel on Digital Cooperation: report of the Secretary-General. (2020).
- [44] D. Thaler, R. Draves, A. Mastumoto, and T. Chown. 2012. *Default Address Selection for Internet Protocol Version 6 (IPv6)*. Technical Report, RFC 6724.
- [45] Daniel J Van Hook, Mark O Yeager, and John D Laird. 2005. Automated topology control for wideband directional links in airborne military networks. In *MILCOM 2005-2005 IEEE Military Communications Conference*. IEEE, 2665–2671.
- [46] Peng Wang, Jiaxin Zhang, Xing Zhang, Zhi Yan, Barry G. Evans, and Wenbo Wang. 2020. Convergence of Satellite and Terrestrial Networks: A Comprehensive Survey. *IEEE Access* 8 (2020), 5550–5588. <https://doi.org/10.1109/ACCESS.2019.2963223>
- [47] Iulislou Zacarias, Luciano P Gaspary, Andersonn Kohl, Ricardo QA Fernandes, Jorgito M Stocchero, and Edison P de Freitas. 2017. Combining software-defined and delay-tolerant approaches in last-mile tactical edge networking. *IEEE Communications Magazine* 55, 10 (2017), 22–29.

APPENDIX

Appendices are supporting material that has not been peer-reviewed.

A MESH REDUNDANCY FORMULA

Given a topology containing B balloons, G ground stations (for $G > 0$), and L links, the minimum number of links required to provide each balloon a route to a ground station is $L_{min} = B$. Given that each balloon has 3 transceivers, the maximum number of possible links (without regard for the geometric or RF feasibility of links) is $L_{max} = \text{floor}(\frac{G+3B}{2})$. Thus the number of possible redundant links is $L_{max} - L_{min}$ and the fraction of possible redundant links utilized is $\frac{L - L_{min}}{L_{max} - L_{min}}$.

B SOLVING

The TS-SDN Solver's objective was to maximize the number of connectivity requests satisfied subject to physical, policy, and priority constraints. The output of each round of topology solving was the set of links (i.e. transceiver pairs) to enact, along with a time at which to enact, that achieved the best theoretic utility. Additionally, route and tunnel intents were emitted on top of the installed topology.

The inputs to the solver included: the set of connectivity requests (with source and destination platforms and desired bitrate), the mapping of transceivers to platforms, and the aforementioned set of candidate link reports (i.e. expected link margin and bitrate for each pair of transceivers which had line of sight and geometric pointing ability at each transmission band). Though not strictly required, the TS-SDN could also examine the existing set of established links (so as to, for example, prioritize candidate topologies that minimized disruption).

The solver imposed several logical constraints on candidate topologies, including: each transceiver may only be paired with at most one other transceiver, and paired transceivers must use non-interfering channels.

Given a set of center frequencies and channel bandwidths, F , a set of transmit powers, W , and a set of transceivers, T , the inputs to

the solver for a given time slice included the set of links, L , where each candidate link, $l_{i \rightarrow j} \in L$, consisted of:

$$l_{i \rightarrow j} = \{t_i, t_j \in T, f \in F, w \in W, b_{\text{modelled}}, m_{\text{modelled}}\}$$

where:

- (1) t_i and t_j are not both attached to the same platform and have line-of-sight free from any known obstructions,
- (2) f and w are link parameters t_i is capable of transmitting,
- (3) f and w are link parameters t_j is capable of receiving, and
- (4) the modelled bit rate, b_{modelled} , and link margin, m_{modelled} , are characteristics determined by the previous link evaluation phase.

Loon nodes were able to forward traffic internally between any pair of transceivers. This connectivity was assumed and not explicitly modelled, though explicit modelling would also have been a valid approach.

The solver was also given a set of connectivity requests, C , for connectivity between nodes in the set of all nodes, N , where each $c_{x \rightarrow y} \in C$ consisted of:

$$c_{x \rightarrow y} = \{n_x, n_y, b_{\min}\}$$

where:

- (1) $n_x, n_y \in N$,
- (2) $n_x \neq n_y$, and
- (3) b_{\min} is the minimum required bit rate.

Any given connectivity request $c_{x \rightarrow y}$ is theoretically satisfiable when a set of candidate links $L_{x \rightarrow y} \subseteq L$ exists such that:

$$\min\{b_{\text{modelled}} \in L_{x \rightarrow y}\} \geq b_{\min}$$

Additional constraints were applied to candidate topologies, e.g., verifying non-interference of $f_{i \rightarrow j}$ and $f_{j \rightarrow i}$. The chosen topology of the previous time slice was also input, and used to prioritize candidate topologies that minimized disruption. The output was a candidate topology for the given time slice, i.e. the set of links $L_{\text{candidate}} \subseteq L$ that maximized the utility of satisfiable connectivity requests. From this the set of nodes and transceivers to be tasked could be derived and radio and route commands issued.

Given access to a low-latency cache of link budget reports, greedy heuristics proved to be a simple mechanism for determining the radio resources to deploy in a demand-aware manner at any instant in time. For each instant in time, one can employ the following iterative algorithm:

```

mark all possible links as "viable"
estimate the utility of all viable links
while there exist viable links with positive estimated utility do
  add highest utility link  $l_{i \rightarrow j}$  to solution set
  mark as "inviable" any links incompatible with  $l_{i \rightarrow j}$ 
  estimate the utility of all viable links
end while

```

The utility of the resulting network topology is highly dependent on the choice of efficient link utility estimation heuristic. One intuitive heuristic is to route each traffic source to its destination on a graph of all viable links, and then take the sum of each link's carried traffic to be that link's utility.

If those routes are determined with respect to link costs that encourage continuity of link selections (i.e. hysteresis) and discourage

selection of short-lived links, the heuristic can yield topologies that are adaptive to demand and robust to dynamic link availability.

C NETWORK MANAGEMENT

Data Plane. Each node in the Loon network was assigned its own global unicast IPv6 /64 prefix and all addressable services associated with the node were numbered from within this prefix. For example, at each EC installation every virtualized network function (each EPC service) was assigned its own address from within the EC's /64. Similarly, each network-reachable compute node in the constellation of computers that formed the balloon payload provisioned addresses from the balloon's prefix, especially: the control plane management node and each of the attached eNodeB nodes (typically two eNodeB computers each controlling 2 sector antennas).

The TS-SDN enacted data plane connectivity by issuing commands to control plane agents at all relevant nodes, primarily in the form of full source-destination route instructions and IPsec tunnel establishment parameters. IPsec tunnels were configured between Ground Stations and EC pods, and IPsec sessions were also provisioned between balloon eNodeBs and NFVI nodes in EC installations (routed over GS-EC IPsec tunnels as necessary). These eNodeB IPsec sessions carried all required mobile network traffic (e.g. S-1 and GTP-u), and thus both LTE network control and data planes were routed over the Loon network data plane.

SDN-programmed IPv6 routes were hardware-accelerated to support line-rate forwarding and minimize power consumption. For these same reasons, the ad hoc control plane was deemed an unsuitable backup path for unscheduled traffic rerouting. Indeed, a primary motivation for the use of full source-destination routing was to make sure that traffic flows stayed on assigned paths to meet resource reservation requirements. With fewer B2G links than B2B links, and even fewer ground station-to-EC links, use of destination-only routing to a handful of edge pods would likely not have resulted in optimal use of diverse sky-to-ground path (depending on available mesh connectivity).

Network Provisioning. Management of the network was highly automated. The TS-SDN exposed an NBI implemented as a gRPC service to other automated data center systems, such as LTE service management and the FMS. Service requests for backhaul transit were submitted to the SDN to provision connectivity across the network and establish data plane routing.

For example, the LTE management stack running in the data center would automatically request backhaul for a balloon's eNodeB to the regional mobile telecommunication carrier's primary LTE MME when it detected that the balloon was in a good location for serving users, had sufficient power, would comply with regulatory constraints, and would not interfere with other balloon's coverage patterns. The requests specified "flow classifier" matching rules, the required bandwidth, and the desired path redundancy. The system was designed to choose topologies and assign routes such that routes with the same "redundancy group" tag would seek disjoint paths. Combined with LTE features like SCTP multi-homing and S1-Flex, this added redundancy to the data plane and was key to our strategy of building a network whose availability exceeded that of individual connections.

Administrative Drains. Drain requests were another key NBI concept which allowed for the temporary exclusion of network nodes from the data plane by rerouting production traffic around the drained node. Drain requests could be specified with enactment times and actuation policies which allowed the system to gracefully orchestrate automated functions such as low power transition, software updates, and other automated maintenance/calibration. For example, to implement an "Opportunistic" drain, the SDN controller would passively wait for a node to naturally lose all traffic, then latch that state. Given the dynamism and constraints of our network, we could expect every node to become fully disconnected from the mesh every night, if not before as balloons moved around. This allowed us to loosely schedule maintenance - software updates in particular - so that they wouldn't impact end-user service. Policies to "deter" traffic from traversing a node until it was drained, or to immediately force traffic from a node were also supported. In addition to automated requests, the production engineering team could also manually request drains to override the system for troubleshooting, experimentation or planned maintenance.

D MANET SELECTION AND IPV6 IMPLICATIONS

A study of mesh network protocol behaviors in the Loon environment [8] was undertaken using ns-3 [26], comparing Ad hoc On-Demand Distance Vector (AODV), Destination-Sequenced Distance-Vector Routing (DSDV), and Optimized Link State Routing Protocol (OLSR) protocols. Since Loon nodes do not need control plane connectivity to other Loon nodes, only a connection to an SDN controller endpoint, convergence time for a route to/from a small set of specified SDN endpoints was a characteristic of particular importance. Both AODV and DSDV protocols exhibited good convergence times, but AODV protocol design resulted in overall lower overhead (no need to build a full routing table for arbitrary balloon-to-balloon connectivity).

When surveying AODV-based approaches, "B.A.T.M.A.N. advanced" (batman-adv) presented several advantages. Being a Linux kernel module [31] with acceptable code maturity made it easy to begin experimentation. More importantly, operating as a virtual Layer 2 network interface with traffic encapsulated in its own EtherType allowed for safely isolating this MANET traffic from data plane traffic, minimizing the possibility of adverse interaction.

This design created a virtual L2 broadcast domain spanning all established wireless links. Ground Stations were configured to be batman-adv gateways to enable balloon clients to identify and sort GS-based connectivity according to kernel-exported batman-adv metrics, e.g. Transmit Quality (TQ). To minimize conflict with balloon-assigned IPv6 addressing, routing of which was controlled by the SDN for data plane functions, addressing was explicitly associated with each GS using unique, frequent IPv6 Router Advertisement (RA) messages: each Ground Station advertised a dedicated /64 through its batman-adv interface and operated in a "64share" [11] mode (similar to IPv6 tethering on some mobile devices). GS RAs did not advertise a default router lifetime, since they did not provide IPv6 Internet connectivity, and instead advertised access to a preferred EC pod, to which an IPsec tunnel had been established, using Route Information Options (RIOs) [17].

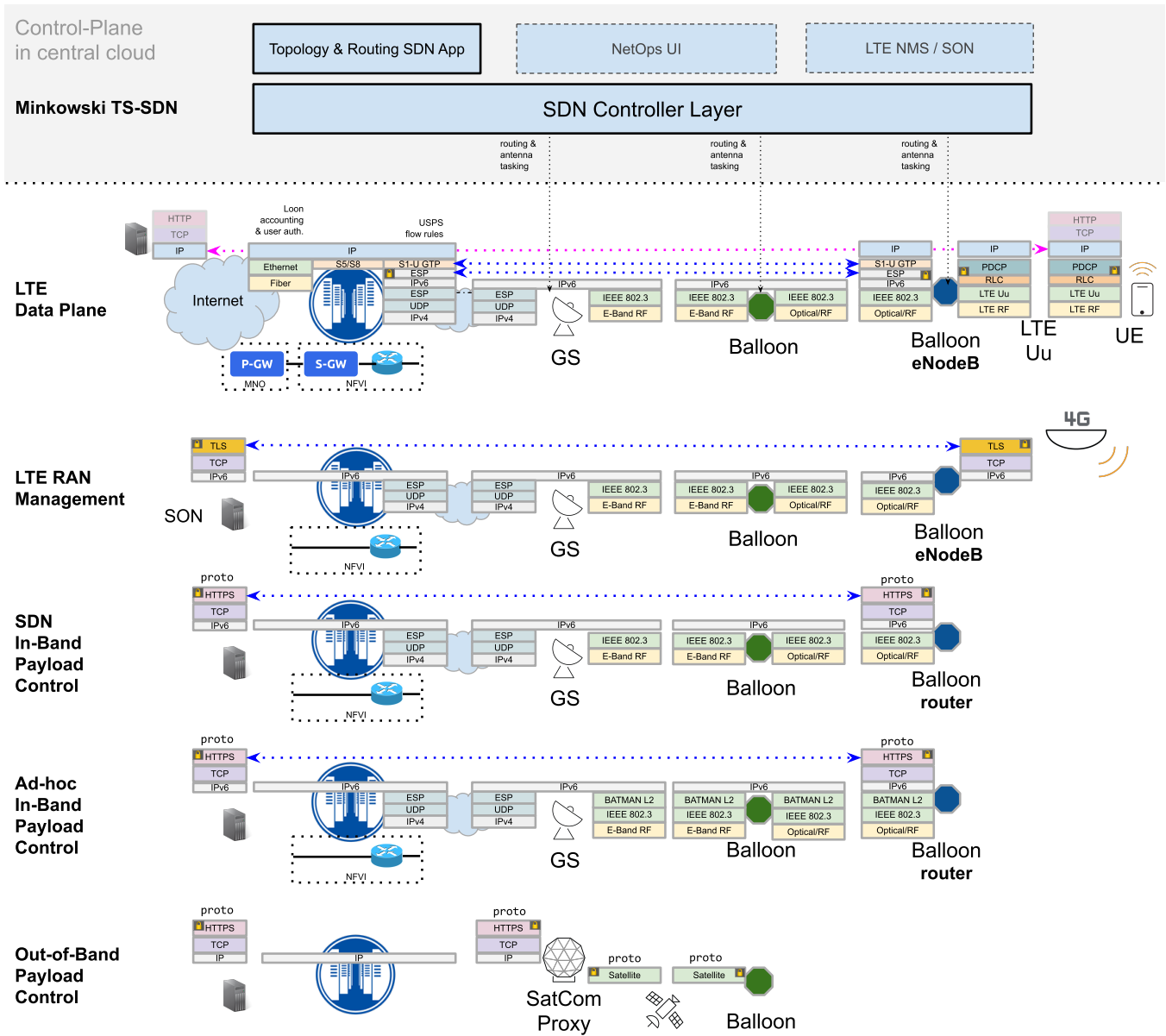


Figure 12: Network architecture for Loon's data plane and control plane.

Despite the absence of advertised IPv6 Internet connectivity, any balloon choosing to configure IPv6 addresses from with the Prefix Information Options (PIOs) of multiple ground stations would face the types of challenges described in RFC 5220 [35] section 2, chiefly: the problem of source address selection in coordination with next hop router selection given that the Linux kernel did not have an implementation of RFC 6724 [44] Rule 5.5 ("[p]refer addresses in a prefix advertised by the next-hop"). Because the SDN did not program a fully connected mesh of $O(n^2)$ IPsec tunnels between GS and EC nodes, EC reachability from a balloon was critically tied to source address and next hop GS selection (as there was otherwise no guaranteed return path).

To function correctly in this environment, RA processing was moved into a user space process. All RAs were sorted according to batman-adv gateway metrics and, in the absence of a reachable, previously selected GS, the RA associated with the "best" GS gateway was selected for application (i.e., formation of addresses from the PIO, programming of routes according to the RIOs, etc.). Once selected, as long as the gateway continued to be reachable, other RAs were examined and held in reserve but not used for provisioning any address and route information, thus dampening connectivity flapping. To ensure fast connectivity changes when a new GS/RA had been selected was to SOCK_DESTROY [16] all sockets using

old source addresses, triggering control plane applications to reinitiate gRPC connections. Altogether this "one working RA at a time" approach yielded a relatively simple host implementation, though if full Provisioning Domain [5, 39] support was readily available it would likely have been preferred.

E ARTIFACT APPENDIX

Abstract

The Loon Network artifact is a dataset consisting of internal state from the TS-SDN and network telemetry gathered from serving commercial traffic and R&D experiments.

Scope

Data provided can be used to reconstruct a subset of the graphs in this paper. With the provided data, we hope that others can draw additional insights and propose alternative link selection algorithms that improve mesh properties.

Contents

Loon collected real world data using system telemetry from Loon assets (balloons, ground stations, etc). These logs were extracted from various storage systems and processed for external publication.

The data is presented in a tabular format, where each table is stored as a comma-separated values ASCII file (i.e. CSV format). Each file is compressed using bz2 (bzip2, a block-sorting file compressor. Version 1.0.8, 13-Jul-2019).

There are 5 tables in total:

- (1) **Network connectivity probes** (backhaul.csv) contains result of network reachability probes from the ground to nodes within the network via different layers of the network's control and data planes across points in time.
- (2) **Link intents change log** (link_intents.csv) contains state transitions of each attempted link. A Link Intent is created by the TS-SDN to indicate its desire for a link between two node's interfaces, and to track the state of the link over time.
- (3) **Transceiver Link Reports** (link_reports-*.csv) contains the evolution of the TS-SDNs candidate graph. Each Transceiver Link Report records the forecasted radio link performance and the sources of attenuation for a given set of transceiver parameters for a time in the future. Forecasted RF performance incorporates the expected spatial geometry of the nodes at the forecast time, and the forecasted weather along the transmission vector. Transceiver Link Reports are packaged in per-hour-recorded files.
- (4) **eNodeB stats** (enodebstats.csv) contains data service download and upload statistics provided to users per eNodeB. The eNodeB is the 4G component that manages sector antennas of the LTE base station.
- (5) **Flight regions** (flight_regions.csv) contains geographic region locations for each flight across points in time.

Hosting

Open source datasets from Loon's TS-SDN and production network deployment are hosted by Zenodo and publicly accessible at <https://doi.org/10.5281/zenodo.6629754>.

Requirements

Software. The data set provided via Zenodo has been compressed using Bzip2 which should be available with modern Unix/Linux systems. Cloud analytic platforms, such as Google Cloud's BigQuery, will allow for fast interactive querying.

Hardware. The data set has an uncompressed size of 191 GB. Processing systems being used to analyze the data should have sufficient storage, memory and processing power to handle datasets of this magnitude.

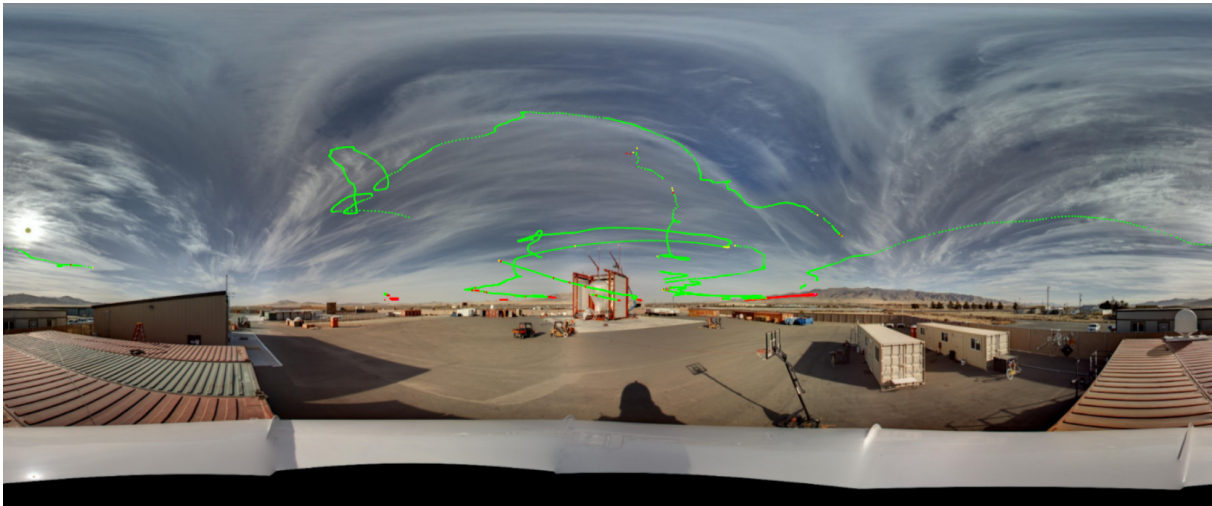


Figure 13: Link status overlaid on a ground station’s field of regard. The pointing vectors where links observed good or poor signal strength are represented by green and red dots, respectively. Note that signal diminished as pointing vector is obstructed by obstacles such as terrain or structures.

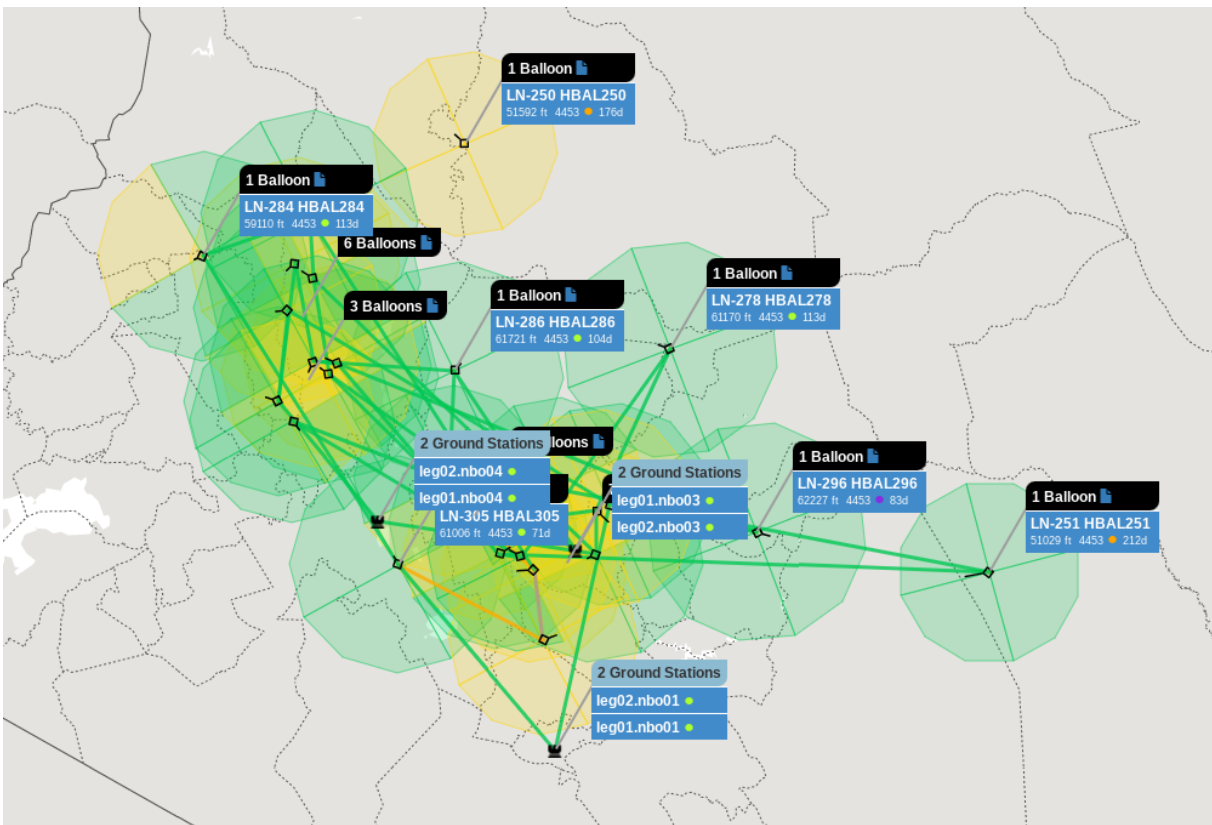


Figure 14: LTE coverage cones for a flock of balloons over Kenya. Each of the balloon’s 4 LTE sectors were independently enabled to avoid self interference or transmission into non-permitted regions. Green sectors are enabled. Yellow sectors are in standby mode. The green lines represent established E band backhaul connections.