

Ein ganzheitliches dezentralisiertes Betriebskonzept für den Betrieb von PUS-basierten CubeSat-Technologie-Demonstrationsmissionen

S. Tholl ⁽¹⁾, A. Ohndorf ⁽¹⁾, M. T. Knopp, D. Krieger ⁽¹⁾, A. Hauke ⁽²⁾, B. Dachwald ⁽³⁾

⁽¹⁾ Deutsches Zentrum für Luft- und Raumfahrt e.V. (DLR), Kompetenzzentrum für Reaktionsschnelle Satellitenverbringung (RSC³), Eugen-Sänger-Straße 50, 29328, Faßberg, Deutschland

⁽²⁾ Deutsches Zentrum für Luft- und Raumfahrt e.V. (DLR), Raumflugbetrieb und Astronautentraining, Münchener Str. 20, 82234 Weßling, Deutschland

⁽³⁾ FH Aachen, Fachbereich für Luft und Raumfahrttechnik, Hohenstaufenallee 6, 52074, Aachen, Deutschland

ZUSAMMENFASSUNG

CubeSats haben sich von einfachen Technologie-Demonstratoren zu wichtigen Ressourcen für die Weltraumnutzung entwickelt. Mit ihnen kann inzwischen eine Vielzahl komplexer Ziele erreicht werden. Im Gegensatz zu geringer ausfallenden Launch-Kosten, fallen die Betriebsaufwendungen von CubeSats u.a. aufgrund ihrer begrenzten Downlinkfähigkeiten oft höher aus als bei klassischen Satellitenmissionen. Das neuerliche Aufkommen von PUS-gesteuerten CubeSat-Plattformen ermöglicht die Nutzung traditioneller Bodensegment-Infrastrukturen, anstatt für jeden neuen CubeSat ein gesondertes Bodensegment hinzuzufügen. Neben der Einsparung von Betriebskosten weisen diese Systeme jedoch Schwächen auf, die den CubeSat-Flugbetrieb unflexibel machen. Daher wird derzeit im DLR an einer Lösung gearbeitet, welche eine einfache Rekonfiguration etablierter Bodensegment-Infrastrukturen ermöglicht, um besser in das Betriebskonzept aktueller CubeSat-Missionen zu passen. In diesem Beitrag stellen wir den Entwurf eines holistischen CubeSat-Kontrollzentrums vor, das als skalierbares, Dienstbasiertes Bodensegment etablierte Bodensegment-Infrastrukturen in einer dezentralisierten Weise nutzt und es unterschiedlichen Missionsbetriebsentitäten ermöglicht, an der gleichen Mission zur gleichen Zeit zu arbeiten.

SCHLÜSSENWÖRTER

Web-basierter Multi-Missionsbetrieb, Ground Segment as a Service (GSaaS), Kleinstsatelliten, Packet Utilization Standard (PUS), Responsive Space

1. EINFÜHRUNG

Zu Beginn des 21. Jahrhunderts stellten CubeSats, die ursprünglich als Bildungsplattformen eingeführt wurden, einen Wendepunkt in der Raumforschung dar. Trotz ihres anfänglichen akademischen Charakters erkannten der private Sektor und Regierungen schnell ihren wirtschaftlichen Wert, was zu einer Vielzahl von Anwendungen führte. Viele CubeSat-Betriebssysteme zeichnen sich jedoch durch umfangreiche Anpassungen der Infrastruktur und Software aus, was zu eingeschränkter Interoperabilität, begrenzten Mehrbenutzerfähigkeiten und betrieblicher Komplexität führt.

CubeSats mit optischer Raum-Boden-Kommunikation sind anfällig für Wettereinflüsse und technische Einschränkungen, die hochentwickelte Kontrollsysteme erfordern. Gegenwärtige Systeme sind jedoch nicht in der Lage, die erforderliche Effizienz zu gewährleisten, wodurch die Kostenvorteile eines CubeSat-Starts häufig zunichte gemacht werden.

1.1 Motivation und Zielsetzung:

Mit der rasanten Entwicklung von CubeSats von rudimentären Technologiedemonstratoren zu entscheidenden Komponenten für Raumfahrtmissionen sind auch ihre betrieblichen Herausforderungen in den Vordergrund gerückt. Diese Kleinstsatelliten bieten zwar geringere Startkosten, doch ihre Betriebskosten, überschatten oft die Einsparungen. Die

Einführung von PUS-gesteuerten CubeSat-Plattformen verspricht eine Lösung durch Nutzung der traditionellen Bodensegment-Infrastruktur. Diese Plattformen sind jedoch mit zahlreichen Funktionen ausgestattet, von denen viele für einen effektiven CubeSat-Betrieb unnötig sind.

Die oben genannten Herausforderungen werden durch die wachsende Zahl von Satellitenkonstellationen und die Notwendigkeit, Plattform und Nutzlast gleichzeitig zu betreiben, noch komplexer. Ein disruptiver Ansatz ist unerlässlich, um das Versprechen eines kosteneffizienten CubeSat-Betriebs zu verwirklichen und den Zenit der Effizienz zu erreichen, insbesondere wenn dieser in der optischen Kommunikation verankert ist.

Dieser Beitrag befasst sich mit dem Entwurf und der Anwendung eines ganzheitlichen CubeSat-Kontrollsystems, das diese Lücken schließen soll. Wir werden seine Rolle in der bevorstehenden deutschen CubeSat-Technologie-Demonstrationsmission "CubelSL" (ISL steht für Inter-Satellite-Link) untersuchen und seine breiteren Auswirkungen im Bereich des CubeSat-Betriebs diskutieren.

2. HINTERGRUND UND VERWANDTE ARBEITEN

Mit der steigenden Nachfrage nach höheren Datenraten und sicherer Kommunikation wandte sich die CubeSat-Gemeinschaft der optischen Kommunikation zu und läutete damit

eine neue Ära der Weltraum-Boden- und Intersatelliten-Konnektivität ein. Im Jahr 2012 nutzte der CubeSat FITSAT-1 erstmals optische Kommunikation mit Hochleistungs-LEDs für den Downlink (T. Tanaka, 2013). Ein bedeutender Fortschritt in der laserbasierten Raum-Boden-Kommunikation war die OCSD-Mission von der Aerospace Corporation, die 2017 höhere Datenraten als konventionelle RF-Systeme demonstrierte (S. Janson, 2016). Der CubeSat Pathfinder PTD-3 erreichte 2022 mit seiner optischen Nutzlast TBIRD beeindruckende 200 GB/s (C. M. Schieler, 2022) und unterstrich damit das Potenzial von CubeSats für anspruchsvolle Missionen. Ein weiterer Durchbruch gelang 2022 mit der NASA-Mission CLICK-A, die die optische Kommunikation zwischen CubeSats im erdnahen Orbit demonstrierte und das Potenzial der Laserkommunikationstechnologie für CubeSats aufzeigte (K. Cahoy, 2019).

Die Einführung der optischen Kommunikation in CubeSats hat die Türen zu anspruchsvolleren Missionen wie der Erdbbeobachtung mit hochauflösenden Sensoren, der wissenschaftlichen Datenübertragung aus dem All und der sicheren militärischen Kommunikation geöffnet. Da die Nachfrage nach Daten weiter steigt, wird die optische Kommunikation bei künftigen Weltraummissionen wahrscheinlich eine noch wichtigere Rolle spielen und noch größere Herausforderungen für den Missionsbetrieb mit sich bringen.

Für 2023 plant die University of Tokyo eine laserbasierte ISL-Verbindung zwischen dem 6U CubeSat „CubeSOTA“ im LEO und dem GEO-Satelliten ETS9-HICALI mit einer Datenrate von 10 Gbps. Dabei werden optische Bodenstationen (OGS) in Tokio für Space-to-Ground Experimente genutzt (Carasco-Casado, 2019).

Die PIXL-1 Mission mit dem OSIRISv1 Terminal war ein Schlüsselereignis für die Validierung eines deutschen laserbasierten Kommunikationsterminals für CubeSats in Zusammenarbeit mit DLR-KN und TESAT (C. Schmidt, 2022).

Das DLR-Kompetenzzentrum für Reaktionsschnelle Satellitenverbringung (RSC³) plant eine Mission, bei der zwei CubeSats mit optischen Kommunikationsterminals ausgestattet werden, um eine Space-to-Space-Verbindung mittels Laserlicht aufzubauen (D.N. Amanor, 2018). Ein Start der CubeSL-Mission ist für 2024 geplant (B. Rödiger C. F., 2021).

Die optische Kommunikation bietet Vorteile wie Kompaktheit, hohe Datenraten und erhöhte Sicherheit. Herausforderungen sind jedoch die Wetteranfälligkeit und die begrenzte Leistungsfähigkeit von CubeSats (M. Motzigemba, 2019).

Der kooperative Betrieb eines Satelliten durch mehrere Institutionen mit unterschiedlichen Zielen erfordert fortschrittliche Koordinationsprotokolle und Management. Ein ausgedehntes Netzwerk von Bodenstationen ist unerlässlich, um eine konsistente Kommunikation und einen effizienten Missionsbetrieb zu gewährleisten und den vielfältigen Anforderungen gerecht zu werden.

2.1 DER PACKET UTILIZATION STANDARD (PUS)

PUS ist zum Zeitpunkt der Erstellung dieses Artikels im Dokument ECSS-E-70-41C, der European Cooperation for Space Standardisation verankert. Es handelt sich um ein inhärent dienstleistungs-orientiertes Modell, sowohl in der Konzeption als auch in der Umsetzung (ECSS, 2016). Dieser Ansatz bietet durch die Standardisierung der Interaktionen

zwischen Bodensystemen und Bordsoftware eine strukturierte Methode für den Betrieb von Raumfahrzeugen. PUS definiert mehrere Servicekategorien, die jeweils spezifische Betriebsbereiche adressieren, wie z.B. Zeitmanagement, On-board-Monitoring oder Ereignisberichterstattung.

Das serviceorientierte Paradigma von PUS bietet einen strukturierten Ansatz für den Betrieb von Raumfahrzeugen und gewährleistet die Konsistenz von Telekommandos und Telemetrie (R. Arias, 2008). Diese Standardisierung stellt sicher, dass die Struktur und die Art dieser Interaktionen konsistent bleiben, unabhängig von Raumfahrzeug oder Mission. Jeder PUS-Service hat vordefinierte Parameter, die wesentliche Details für seine Ausführung bereitstellen. Dies können verschiedene Servicetypen sein, die Aufgaben wie dem Zurücksetzen eines Bordsubsystems oder dem Umschalten des Betriebsmodus eines Instruments entsprechen. PUS wurde für den grundlegenden Betrieb von Raumfahrtmissionen konzipiert und ist missionspezifischen TMTC-Protokollen (TMTC: Telemetry and Telecommand) in Bezug auf Kommunikation und Dateitransfer überlegen. Um den Missionsbetrieb von unnötigem Ballast zu entlasten, gibt es die Möglichkeit, die Services durch sog. Tailoring an die Bedürfnisse der Mission zu reduzieren und ggf. anzupassen (Masutti, 2017) (TEB, 2016). Zusätzlich sind die PUS-Protokollbibliotheken in verschiedenste Programmiersprachen auf GitHub unter LGPL-Lizenz frei verfügbar.

Mit PUS können unterschiedliche Bodensysteme nahezu nahtlos mit verschiedenen Raumfahrzeugen zusammenarbeiten. Dadurch wird vermieden, dass für jede Mission eine eigene Schnittstelle entwickelt werden muss. Diese Flexibilität stellt die Skalierbarkeit sicher und macht PUS für Missionen geeignet, die von größeren Raumfahrtsystemen bis hin zu CubeSats reichen, und verbessert die Interoperabilität und Robustheit, indem ereignisgesteuerte Operationen erleichtert werden (L. Stöcken, 2018).

In der Vergangenheit wurde der Weltraumbetrieb von großen Satelliten mit proprietären Kommunikations-Protokollen dominiert, was zu Kompatibilitätsproblemen bei den Bodenstationen und Missionszentren führte. Hier kam PUS ins Spiel. Ursprünglich für größere Raumfahrzeuge entwickelt, wurde PUS zunehmend auch für den aufstrebenden CubeSat-Sektor relevant.

Das European Space Operations Centre (ESOC) und das GSOC sowie das Centre Spatial de Toulouse (CNES/CST) (C. Laroque, 2018) haben PUS frühzeitig eingeführt (M. Pecchioli, 2010). Viele etablierte Anbieter von weltraumprobten CubeSat-Plattformen, verwenden weiterhin individuelle Varianten des CubeSat Space Protocol (CSP). CSP ist ein besonders leichtes und einfaches TMTC-Protokoll für CubeSats, das zu einer Zeit entwickelt wurde, als die Rechenleistung solcher Systeme noch bescheiden war (Wikipedia, 2023). Obwohl es von mehreren CubeSat-Herstellern aufgegriffen wurde, gibt es derzeit kein standardisiertes CSP-Protokoll. Leider sind auch keine Bemühungen zur Standardisierung von CSP geplant (P. A. Lagadrillière, 2023). Dies führt zu Problemen, da die Integration solcher plattformspezifischen CSP-Varianten in Missionsbetriebseinrichtungen für die Nutzung unterschiedlicher CubeSat-Plattformen eine umfangreiche Anpassung der implementierten Systeme, eine erneute Schulung des Personals und eine Neuanpassung der Validierungsprozesse erfordert, um eine nahtlose Integration

dieser Protokolle zu gewährleisten und gleichzeitig die Integrität und Funktionalität der bestehenden Missionen zu erhalten. Angesichts der langjährigen Flugerprobung und der breiten Akzeptanz von PUS, würde dieses Protokoll die Integration von CubeSats in Multimissionsumgebungen langfristig drastisch vereinfachen (L. Grillmayer, 2020) (P. A. Lagadrillière, 2023).

Beispielsweise hat die systemweite Implementierung des PUS-Protokolls am GSOC die Entwicklung eines Software-Dolmetschers zur Übersetzung von CSP in PUS erforderlich gemacht, um den CubeSat PIXL-1 mit der vorhandenen Infrastruktur betreiben zu können. Angesichts des prognostizierten Rückgangs des CubeSat-Marktes bis 2025 (New Space Economy, 2023) ist es zweifelhaft, ob große Raumfahrtzentren von der Implementierung proprietärer CubeSat-Protokolle wie CSP profitieren würden. Dies führte zu der Erkenntnis, dass die Cubesat-Mission vorzugsweise PUS-kompatibel sein sollte.

Seit Anfang 2010 hat CNES die Entwicklung von PUS-kompatiblen CubeSats im Rahmen des Projekts Janus vorangetrieben und sehr erfolgreiche CubeSat-Missionen wie Angels, (S. Salas, 2018) EyeSat (F. Apper, 2019) und Kinéis durchgeführt (Wikipedia, 2023). Etwa zur gleichen Zeit unterstützte die ESA auch die Entwicklung von PUS-kompatiblen Kleinstsatelliten, zunächst mit Proba (S. Ilsen, 2016) und später mit der Mission OPS-SAT (D. Evans, 2014).

Zusammenfassend lässt sich sagen, dass PUS den Aufwand reduziert, Systemänderungen und Personalschulungen minimiert und die Interoperabilität verbessert - ein entscheidender Vorteil in der zunehmend kooperativen Weltraumarena. Während das CubeSat Raumfahrtprotokoll seine Nische hat, weisen die übergreifenden Vorteile von PUS einen vielversprechenden Weg in die Zukunft. Mit der zunehmenden Integration von Raumfahrtmissionen werden Protokolle wie PUS, die den Betrieb vereinfachen und die Kompatibilität verbessern, wahrscheinlich zum Standard für zukünftige CubeSat-Missionen der etablierten europäischen Agenturen. Tatsächlich bieten mehrere neu gegründete Raumfahrtunternehmen bereits von Grund auf PUS-kompatible Mehrzweck-CubeSat-Plattformen an (A.J. Lago, 2019).

3. HERAUSFORDERUNGEN BEIM DERZEITIGEN CUBESAT-BETRIEB

In Kapitel 2 wurde eine Reihe erfolgreicher CubeSat-Missionen vorgestellt und gezeigt, wie die spezifischen Herausforderungen beim Missionsbetrieb stetig gewachsen sind. Diese ergeben sich nicht nur aus den inhärenten Beschränkungen der CubeSat Plattform, sondern zunehmend auch aus den anspruchsvolleren Missionsprofilen, die an diese Plattformen gestellt werden.

Daraus ergibt sich ein Bedarf an komplexeren Betriebsszenarien, die sowohl die wachsenden Anforderungen der Missionen als auch die technischen Grenzen der CubeSats berücksichtigen. So erfordert zum Beispiel der begrenzte Platz für Solarpaneele und Batterien, ein sorgfältiges Energiemanagement, um einen effektiven Betrieb während der gesamten Mission zu gewährleisten, da die für die Systeme und Instrumente des Satelliten verfügbare Energiemenge noch begrenzt ist. Auch schränken die geringe Größe und das geringe Gewicht von CubeSats die Möglichkeiten des Satelliten und die Art der Missionen ein, da nur eine begrenzte Anzahl

von Instrumenten und Ausrüstung an Bord mitgeführt werden kann. Eine weitere Herausforderung ist das Wärmemanagement, da die geringe Größe der CubeSats nur wenig Platz für ein aktives Thermalmanagement lässt und der CubeSat im Orbit extremen Temperaturschwankungen ausgesetzt sind, die seine Lebensdauer erheblich beeinträchtigen.

CubeSat-Bordcomputer haben im Vergleich zu größeren Satelliten auch eine begrenzte Rechenleistung, was bedeutet, dass CubeSats oft Unterstützung vom Bodensegment bei der Durchführung komplexer Datenverarbeitungsaufgaben an Bord benötigen. CubeSat-Bordrechner mit einer PUS-Systemarchitektur haben in diesem Zusammenhang einen entscheidenden Vorteil gegenüber CSP-basierten Systemen, da sie dem Betreiber die Möglichkeit bieten, serviceorientiert in das Softwaresystem einzugreifen.

Insbesondere bei CubeSats, die wie im Fall von CubesatSL auf optische Space-to-Space- und Space-to-Ground-Kommunikation angewiesen sind, kann eine genaue Lageregelung aufgrund des geringen Durchmessers der Reaktionsräder und der geringen Öffnung der Startracker ebenfalls eine Herausforderung darstellen, was die genaue Ausrichtung zu einem komplexen Unterfangen machen und den Betrieb verlängern kann.

Darüber hinaus erlaubt die geringe Größe von CubeSats nur eine begrenzte oder gar keine Redundanz kritischer Subsysteme. Typischerweise sind die Lageregelung und das elektrische Energiesystem (EPS) die ersten Subsysteme, die ausfallen. Infolgedessen haben CubeSats in der Regel eine kürzere Lebensdauer als größere Satelliten, was es noch wichtiger macht, die primären Ziele der Weltraummission so schnell wie möglich zu erreichen, und einfachere, flexiblere Missionsbetriebskonzepte erfordert, um den Erfolg der Mission zügig zu gewährleisten.

Herkömmliche CubeSat-Missionen sind oft auf vordefinierte Zeitpläne, feste Bodenstationen und spezielle Missionskontrollsysteme angewiesen, was die betriebliche Flexibilität einschränkt. Darüber hinaus unterstützen herkömmliche CubeSat Missionskontrollsysteme nicht notwendigerweise einen Datenaustausch in Echtzeit und die Zusammenarbeit zwischen den verschiedenen beteiligten Operationsentitäten, was schnelle Entscheidungen und Anpassungen während der Mission erschwert.

CubeSat-Missionen in einem operationellen Rahmen mit mehreren Entitäten stellen eine ähnlich komplexe Herausforderung dar wie konventionelle Satellitenmissionen, wenn es um die effiziente Zuweisung und Verwaltung von Ressourcen für die verschiedenen operationellen Entitäten geht. Auch hier muss ein erforderliches Maß an Zugangskontrolle geboten werden, um Sicherheitsrisiken und unbefugten Zugang zu missionskritischen Systemen vorzubeugen.

Schließlich erfordern herkömmliche Systeme eine spezielle Software für den Fernbetrieb von Missionen, was selbst für erfahrene Flugoperatoren eine Einschränkung darstellt, die routinemäßig mit den vertrauten Missionskontrollsystemen arbeiten, auf denen sie ausgebildet wurden.

Während herkömmliche Methoden im CubeSat-Betrieb in der Vergangenheit erfolgreich waren, führen die zunehmende Komplexität der Missionen und die sich entwickelnden technologischen Anforderungen zu neuen Herausforderungen. Dazu gehören Einschränkungen bei der operationellen

Flexibilität, Echtzeit-Kollaboration, Zugangskontrolle sowie Schwierigkeiten beim Umgang mit hohen Datenraten und dem Management von Missionen mit mehreren Teilnehmern. Daher ist es entscheidend, innovative Ansätze zu entwickeln, die nicht nur die Missionsdauer auf ein Optimum kürzen, sondern auch die operationelle Effizienz und Sicherheit verbessern, um trotz der begrenzten Lebensdauer von CubeSats das volle Spektrum der geplanten Missionsziele zu erreichen und gleichzeitig die Betriebskosten zu minimieren.

4. DIE MISSION CUBE-ISL

Die CubeISL-Mission ist eine Initiative des DLR zur Entwicklung eines deutschen Beitrags für ein sicheres Hochgeschwindigkeits-Laserkommunikations-Terminal (LTC) für CubeSats. Die Mission wird einer Flotte von zwei identischen 6U CubeSats, die mit einem LTC-Terminal ausgestattet sind durchgeführt, die in einer niedrigen Erdumlaufbahn platziert werden.

Die beiden Hauptziele der Mission sind die Validierung der CubeISL-Nutzlasten und die Demonstration der Funktionalität der optischen ISL. Weiterhin ist die Interoperabilität mit anderen optischen Bodenstationen (OGS) ein wichtiges Missionsziel.

Die Ermöglichung der optischen ISL für die Kommunikation ist ein entscheidender Teil der Mission. Darüber hinaus werden im Rahmen der Mission Messergebnisse und Missionsdaten aus den Experimenten mit dem LTC gesammelt, die als wertvolle Erkenntnisse für die CubeISL-Mission und zukünftige Initiativen im Bereich der optischen Kommunikationstechnologie dienen können.

Die hochfrequenten Experimentdaten werden über mehrere OGS heruntergeladen. Die beiden CubeSats werden auch S-Band RF-Transceiver für den Betrieb der Plattform tragen und sollen auch mit UHF-Sendern ausgestattet werden, die Housekeeping Daten in Form von Auto-Beacons an das SatNOGS Bodenstationen Netzwerk übertragen.

Die Mission soll diese Ziele innerhalb eines Jahres nach dem Start erreichen. Da die beiden CubeSats auf optische ISL und LEO Downlink (OLEODLs) angewiesen sind, wird die akkurate Lageregelung ein zentraler Punkt des Plattformbetriebs sein, da ein direkter Zusammenhang zwischen der Ausrichtungsfähigkeit und der Missionsdauer besteht.

OLEODLs sind für schlechte Wetterbedingungen anfällig. Hier kann die Leistungsfähigkeit des vorgeschlagenen virtuellen Missionsbetriebskonzepts und seine Fähigkeit, schnell von einer OGS zur nächsten zu wechseln, einen entscheidenden positiven Einfluss auf die für den Downlink der Missionsdaten benötigte Zeit haben.

Die kurze Betriebsdauer und die ehrgeizigen Missionsziele stellen eine Herausforderung für die Entwickler des Laserbasierten CubeSat Terminals (LTC), Wissenschaftler und die Flugbetriebsmannschaft der CubeISL-Plattform dar, die ihren Teil der Mission zügig erfüllen müssen. Mit der vorgeschlagenen Systemarchitektur soll die CubeISL-Mission dezentral betrieben werden, so dass die verschiedenen Betriebsentitäten ihre Aufgaben innerhalb der vorgegebenen Missionsdauer erfüllen können.

5. VORGESCHLAGENE SYSTEMARCHITEKTUR

Um den Routinebetrieb der CubeISL-Mission innerhalb des Bodensegments der FH Aachen Space Operations Facility durchführen zu können, wird die bestehende Infrastruktur der FHASOF-Bodenbetriebseinrichtung um eine servicebasierte Softwareinfrastruktur und eine Cloud erweitert, die es den studentischen Flugoperatoren ermöglicht, den Missionsbetrieb außerhalb der Öffnungszeiten der FH Aachen durchzuführen und in kritischen Situationen auf erfahrene GSOC-Flugoperatoren zurückgreifen zu können.

Mit der Entwicklung des Holistic Control Centers (HCC) wurde am GSOC bereits 2016 begonnen (A. Hauke M. P. Geyer, 2018) Basis der Systemarchitektur ist die Datentransportschicht HCC-DT (A. Hauke, 2023), ein auf die Anforderungen des Satellitenbetriebs zugeschnittener Datenbus, der den Datentransport zwischen sog. HCC-DT Knoten über das lokale Netzwerk oder das Internet ermöglicht.

HCC-DT wird verwendet, um die physische Bodensegment-Infrastruktur der FHASOF an der FH Aachen (Funkgeräte, Antennensysteme, Modems, Server) mit den physikalisch abgetrennten Missionsbetriebskonsolen, Clients und Benutzerschnittstellen zu verbinden. Diese werden in einem sog. virtuellen Kontrollzentrum konsolidiert.

Die „natürliche“ Grenze des virtuellen Kontrollzentrums wird durch einen virtuellen Router und Forwarder eines kommerziellen „Infrastructure as a Service“ (IaaS) Anbieters abgebildet. VRF ist eine Technologie in IP-Netzwerken, die es ermöglicht, dass mehrere Instanzen einer Routing-Tabelle gleichzeitig in einem einzigen Router existieren können. Dies ermöglicht die Segmentierung von Routing-Pfaden, ohne dass mehrere Geräte erforderlich sind. Jede VRF-Instanz arbeitet isoliert, als wäre sie ein separater Router, befindet sich aber im selben Gerät. Diese Isolierung stellt sicher, dass jeder VRF seine eigene, unabhängige Routing-Tabelle hat, so dass sich die IP-Adressen der VRFs in einem Gerät überschneiden können. VRFs werden häufig in Verbindung mit Multiprotocol Label Switching (MPLS) VPNs verwendet. Bei der Bereitstellung von VPN-Diensten über eine gemeinsame Infrastruktur können VRFs verwendet werden, um den Datenverkehr und das Routing der einzelnen Missionsbetriebseinheiten zu trennen.

Zurück zum Virtual Router and Forwarder (VRF) des hier beschriebenen virtuellen Kontrollzentrums wird jeder Mission ein sog. virtuelles lokales Netzwerk (vLAN) zugewiesen. Dabei können mehrere vLANs nebeneinander koexistieren, so dass jedes vLAN eine Mission beherbergt. Eine Mission besteht aus mehreren sogenannten virtuellen Maschinen (VMs). Auf diesen VMs wird die missionspezifische Infrastruktur aufgebaut. Dabei erfüllt jede VM eine infrastrukturenspezifische Funktion. Beispielsweise werden die für den Betrieb der Mission notwendigen Benutzeroberflächen auf solchen VMs abgelegt.

Diese speziell für den Missionsbetrieb ausgelegten VMs werden als „virtuelle Cockpits“ bezeichnet. Zwischen den VMs können missionspezifische Daten ausgetauscht werden, da diese VMs über das vLAN verbunden sind. Die Verbindung zwischen der virtuellen Missionsschicht und ihren physischen Bodenstationen erfolgt über eine weitere VM im vLAN, auf der ein HCC-DT Knoten läuft. Dieser Knoten stellt die Wide Area

Network (WAN) Verbindung zum HCC-DT Knoten dieser physischen Bodenstation her. Für die CubelSL Mission befindet sich dieser Knoten im physischen Bodensegment der FHASOF.

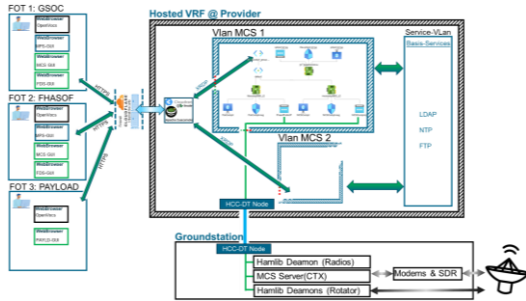


BILD.1 Die vorgeschlagene Systemarchitektur des Ganzheitlichen Kontrollzentrums

Um auf die virtuellen Cockpits der Mission zugreifen zu können, wurde hinter dem VRF eine virtuelle Ubuntu Linux Maschine implementiert, auf der ein Guacamole Server läuft. Apache Guacamole ist ein clientloses Remote Desktop Gateway. Es unterstützt Standard-Fernzugriffsprotokolle wie VNC (Virtual Network Computing), RDP (Remote Desktop Protocol) und SSH (Secured Shell). Clientlos bedeutet hier, dass keine zusätzliche Software auf dem Client-Rechner und den virtuellen Cockpits installiert werden muss, um den Zugriff zu ermöglichen.

Der Zugriff auf die virtuellen Cockpits kann von jedem Rechner aus über einen Webbrowser eines herkömmlichen Office-Rechners erfolgen, vorausgesetzt dass der User gültige Credentials besitzt. Dieser Zugriff kann als digitale Lobby der virtuellen Betriebseinrichtung der Mission angesehen werden, wie in BILD. 2 gezeigt. Durch den Einsatz des Guacamole-Servers und den Betrieb im Browser-Fenster wird die Entwicklung von zweckgebundenen und oft betriebssystemabhängigen Clients überflüssig. Ein weiterer Vorteil der hier beschriebenen Architektur ist, dass die Wartung der Cockpits auf ein Minimum reduziert wird. Guacamole ermöglicht einen robusten und plattformagnostischen Remote-Betrieb.

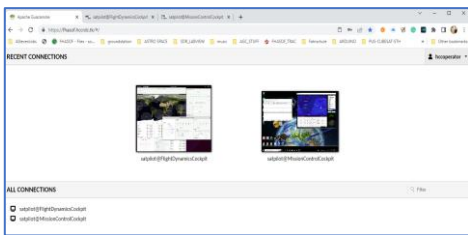


BILD.2 Die digitale Lobby einer Mission. Von hier aus erhält der Operator Zugriff zu den virtuellen Cockpits der Mission. Er sieht nur die VMs auf die er Zugriffsrechte besitzt.

Während die HCC-DT Knoten die Verbindung zwischen dem virtualisierten Mission Control Center und dem physischen Bodensegment über das WAN herstellen, müssen Daten wie Telekommandos (TC), Telemetrie (TM) und Steuerbefehle erst in die HCC-DT Knoten eingespeist werden, bevor sie über diese langen Signalwege transportiert werden können. Da die meisten dieser Softwaresysteme nicht ohne weiteres modifiziert werden können, werden diese Daten mit Hilfe eines softwarebasierten Dateninterpreters in ein für HCC-DT geeignetes Format übersetzt. Zu diesem Zweck wurde die HCC-Bridge entwickelt, (BILD. 3). Diese fungiert als Netzwerkbrücke zwischen Software, Diensten, Geräten und den HCC-DT-Knoten und ermöglicht den Datentransfer zwischen

diesen Anwendungen und dem HCC-DT-System. Dieser Aufbau gewährleistet die nahtlose Kommunikation jedes Teilnehmers mit den HCC-DT-Knoten, wobei sich jeder Teilnehmer an den

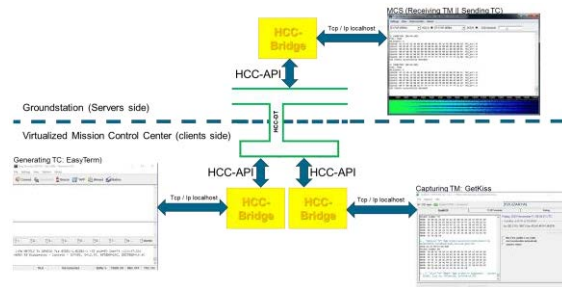


BILD.3 Die HCC-Bridge hilft bei der Anbindung von Teilnehmern an das HCC-DT-Netz

verschiedenen virtuellen Cockpit-Maschinen befindet. Auf den gehosteten Servern sind zwei verschiedene Arten von virtuellen Cockpits in einzelnen virtuellen Maschinen implementiert.

A) Cockpit für Bodenstationsbetrieb (vC-GS)

Das vC-GS dient dem Betrieb der Bodenstation und umfasst mehrere Front-End-Schnittstellen für die Funksteuerung, die Modemkonfiguration und die Satellitennachführung über Gpredict, eine Open-Source-Anwendung (About Gpredict, 2020). Die Verbindung zwischen der Satellitennachführung, dem Antennenrotor und den Funktransceivern wird durch mehrere Instanzen von HCC-Bridges ermöglicht.

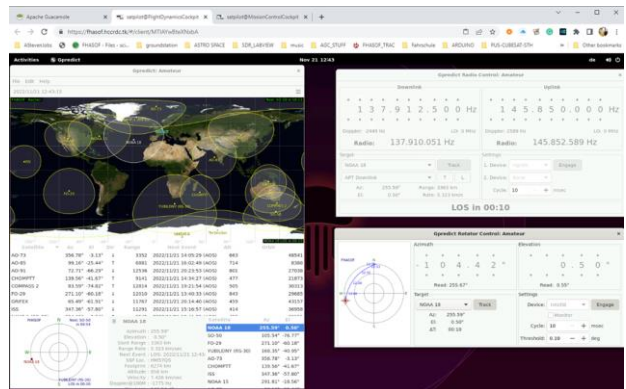


BILD.4: vC-GS auf dem gehosteten Server für AX.25 Amateurfunk Satellitenbetrieb

Beispielsweise übersetzt eine Bridge die Antennenausrichtungskordinaten aus Gpredict und gibt sie in den HCC-DT-Knoten ein. Die Daten werden dann über das WAN an den HCC-DT Knoten des physischen Bodensegments übertragen. Nach dem Empfang leitet eine HCC-Bridge-Instanz im physischen Bodensegment die Daten an die entsprechenden Gerätetreiber der Bodenstation weiter, zu denen der Rotctld-Hamlib-Daemon für Antennenausrichtungsgaräte und der Rigctld-Hamlib-Daemon für die Übermittlung der dopplerkorrigierten Uplink- und Downlink-Frequenzen gehören können. Hamlib ist eine Softwarebibliothek, die eine standardisierte Schnittstelle für die Steuerung von Antennenrotoren und Funktransceivern bereitstellt und damit die Entwicklung von Amateurfunkanwendungen vereinfacht und die Interoperabilität zwischen verschiedenen Softwareanwendungen und

unterschiedlicher Funkgerätehardware fördert (Bargmann, 2021).

B) Cockpits für Satellitenbetrieb (vC-MCS)

Das vC-MCS enthält die Missionsbetriebskonsolen und Benutzerschnittstellen, die Clients wie GECCOS, Protos, SatMon, (M. Hobsch, 2022), EGS-CC (M. Geyer, 2012), OpenMCT (NASA) oder ähnliche sein können. Für den Betrieb von Amateurfunk-Satellitenmissionen wurden EasyTerm, UISS, GETKISS, OnlineKISS und die Satelliten-Telemetrie-Interpreter-Anwendungen einiger prominenter Amateurfunk-CubeSats wie Ops-Sat (ESA), Griffex (NASA) und Funcube installiert.

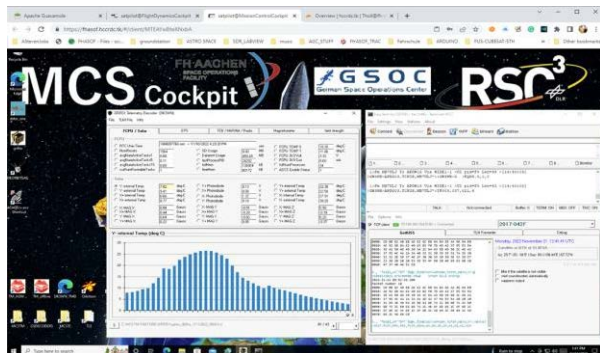


BILD.5: vC-MCS auf dem gehosteten Server für AX.25 Amateurfunk Satellitenbetrieb

EasyTerm wurde vom Funkamateurer UZ7HO entwickelt und ist eine Terminalanwendung zur Kommunikation mit Terminal Node Controllern (TNCs) über TCP/IP unter Verwendung des KISS-Protokolls (Keep It Simple and Stupid), einem Standardprotokoll im Amateurfunk. Ein TNC (Terminal Node Controller) ist ein Gerät, das digitale Daten für die Übertragung per Funk in analoge Signale umwandelt und beim Empfang analoge Signale wieder in digitale Daten für die Weiterverarbeitung im Computer umwandelt. Dieses Programm wird häufig verwendet, um Telecommandos (TC) zu AX.25-kompatiblen Amateurfunksatelliten zu senden. Als Alternative wurde das Programm UISS (Universal ISS) installiert. UISS ist eine Terminalanwendung, die speziell für das Schreiben und Lesen von APRS-Nachrichten zu und von der Internationalen Raumstation (ISS) von Funkamateurer ON6MU entwickelt wurde. UISS verbindet sich zu diesem Zweck mit dem in der Bodenstation vorhandenen AX.25 TNC oder Soundmodem. Soundmodem ist eine Software basierter TNC, der speziell für das Encodieren und Decodieren von AX.25 kodierte Paket-Radio Frames von Amateurfunker UZ7HO entwickelt wurde. Die Anwendung OnlineKISS dient der Archivierung von Telemetriedaten (TM) von Amateurfunksatelliten. Es handelt sich um ein kompaktes Softwareprogramm zur Speicherung von KISS-Frames in einer Datei. Diese Dateien können später mit GETKISS, einem generischen dateibasierten KISS-Frame-Reader, oder mit dem entsprechenden dateibasierten Satelliten-Telemetrie-Interpreter gelesen werden. Mehrere Instanzen von HCC-Bridges stellen die Verbindung zwischen EasyTerm, OnlineKISS und der TNC her. Die erste HCC-Bridge gibt die KISS-Frames von EasyTerm an den HCC-DT-Knoten weiter, von wo aus, die Daten über WAN an den HCC-DT-Knoten des physikalischen Bodensegments übertragen werden. Beim Empfang am Bodensegment holt eine HCC-Bridge-Instanz die KISS-Frames vom Knoten ab

und leitet sie an den entsprechenden TNC weiter. Ein ähnlicher Prozess wird für die Telemetrie-Rückkanäle verwendet: Die mit Telemetriedaten beladenen KISS-Frames werden vom TNC in den HCC-DT-Knoten des physischen Bodensegments eingefügt und dann über das WAN zum HCC-DT-Knoten des virtualisierten Mission Control Centers transportiert. Im vC-MCS holt die HCC-Bridge die KISS-Frames vom Knoten ab und leitet sie an die jeweilige Archivierungsanwendung weiter: OnlineKISS, UISS oder EasyTerm.

5.1 Anbindung zu SatNOGS

Die CubelSL Mission ist nicht für einen 24/7 Multi-Missionsbetrieb ausgelegt. Dies stellt besondere Anforderungen an die CubeSat Plattform, wie z.B. die Fähigkeit zur Selbstdiagnose (FDIR; Fault Detektion and Isolation) und ggf. das automatische Umschalten in einen Safe Mode, bis die First Level Flight Operations Crew wieder verfügbar ist. Zusätzlich sollte die Plattform unabhängig davon in periodischen Abständen Auto Beacons mit einem reduzierten Umfang an Housekeeping-Informationen über das UHF-Amateurfunkband aussenden.

Die in diesem Artikel vorgeschlagene Systemarchitektur soll mit dem SatNOGS-Bodensegment verbunden werden. SatNOGS (Satellite Networked Open Ground Stations) ist ein weltumspannendes dezentrales Netzwerk von Bodenstationen, das auf Open-Source-Prinzipien basiert. Es besteht aus einer globalen Gemeinschaft von Teilnehmern, die ihre eigenen Bodenstationen dem SatNOGS Netzwerk zur Verfügung stellen und die von den Satelliten empfangenen Telemetriedaten über eine gemeinsame Telemetriedatenbank austauschen, der sog. SatNOGS-DB (J. Nicolas, 2021). Die von CubelSL weltweit empfangenen Telemetriedaten können rund um die Uhr aus dieser Datenbank abgerufen werden. Voraussetzung ist, dass die CubelSL Satelliten, amateurfunkkompatible und offene, für den Amateurfunk zugelassene Übertragungsprotokolle im UHF-Band aussenden (IARU, 2023).

SatNOGS verwendet GNU-Radio als generisches Empfangsmodem für die Dekodierung von Satellitendaten. Die Integration von GNU-Radio stellt die Decodierung der Satellitenfunksignale sicher. Hierdurch entfällt die kostenintensive Anschaffung von Cortex-Modems.

Im Laufe der Mission wird die SatNOGS Telemetriedatenbank eine umfangreiche und globale Kollektion von Telemetriedaten zur CubelSL-Mission gesammelt haben. Dies könnte entscheidend dazu beitragen, das Verhalten beider Satelliten unter verschiedenen Bedingungen zu verstehen, und unschätzbare Erkenntnisse für die Optimierung des Missionsbetriebs liefern.

Die Integration von GNU-Radio und dem SatNOGS-Netzwerk erweitert nicht nur die operationellen Fähigkeiten der vorgeschlagenen Systemarchitektur, sondern bettet sie auch in ein globales kollaboratives Ökosystem ein und setzt damit ein neues Paradigma für den Betrieb deutscher CubeSat-Missionen am DLR.

6. TESTBETRIEBKAMPAGNEN DES VIRTUELLEN KONTROLLZENTRUMS

Um das hier vorgestellten virtuelle Kontrollzentrum und dessen Architektur grundsätzlich zu bewerten, wurden zwei Testbetrieb-Kampagnen mit dem APRS-Repeater der ISS

durchgeführt. In der ersten Testkampagne wurde das Virtuelle Kontrollzentrum getestet, welches zuvor auf einem IaaS Service Provider ausgebreitet wurde. Hierfür wurde auf eine Dreimonatigen kostenlosen IaaS-Service Modalität zurückgegriffen. HCC-DT wurde zunächst durch einen sog. HCC-DT-Simulator nachgebildet, um den beabsichtigten Netzwerk-Mechanismus zu testen. Dieser stellt zwei Interfaces zur Verfügung zum Verbinden von Applikationen über TCP/IP, ähnlich wie die HCC-Bridges, kann jedoch Pakete nur unverschlüsselt weiterleiten. HCC-DT-Simulator stand ab Mai 2023 zur Verfügung. Im Anschluss darauf wurde die zweite Testkampagne mit der ISS durchgeführt, in welcher der HCC-DT-Simulator nicht in Verbindung mit einem virtuellen Kontrollzentrum, sondern über zwei Rechner im lokalen Netzwerk der Bodenstation zum Einsatz kam.

Die ISS umkreist die Erde seit 1998 und dient als Weltraumlabor für verschiedene wissenschaftliche Experimente. Eine ihrer Initiativen, ARISS (Amateur Radio on the International Space Station), ermöglicht Amateurfunkkontakte im VHF-Band zwischen der ISS und der Erde zu machen. Eine Besonderheit von ARISS ist der APRS-Repeater (Automatic Packet Reporting System). APRS ist ein digitales Kommunikationssystem, welches das AX.25-Paket Radio Protokoll für den Informationsaustausch zwischen Amateurfunkstationen verwendet.

In der vorliegenden Studie wurde die Internationale Raumstation (ISS) als Testobjekt ausgewählt, da sie aktuell das einzige Raumfahrzeug darstellt, mit dem Up- und Downlink-Tests durchgeführt werden können. Das Bodensegment ist eine zugelassene Amateurfunkstation und verfügt über lizenzierte Funkamateure, wodurch ein schneller Beginn des Testumsetzungsprozesses ohne zusätzliche Genehmigungen ermöglicht wurde. Zunächst wurde aber die Robustheit des Netzwerks der vorgeschlagenen Systemarchitektur quantitativ bewertet.

6.1 Bewertung der Robustheit des Netzwerks der vorgeschlagenen Systemarchitektur

Mit dem HCC-DT-Simulator konnte im Vorgriff auf das HCC-DT begonnen werden, die beabsichtigten Netzwerk-Mechanismen für den Missionsbetrieb zu testen und erörtert werden, ob und wie über den Umweg der HCC-Bridges, bestehende Missionsbetriebssoftware an das echte HCC-DT angeschlossen werden könnte, da die Missionsbetriebssoftware im Allgemeinen nicht Quelloffen vorliegt.

Eine detaillierte Analyse der TCP/IP-Verbindung wurde durchgeführt, um die Robustheit und Qualität der beabsichtigten Netzwerkmechanismen im Hinblick auf den bevorstehenden APRS-Überflug der ISS zu charakterisieren. Dazu wurden zwei Testprogramme, "Sender" und "Receiver", in der Programmiersprache C entwickelt, um kontrolliert, Pakete definierter Größe vom Sender über das TCP/IP-Protokoll zum Receiver zu senden und zu empfangen (S. Tholl, 2023). Das Programm "Sender" schreibt eine Datei, welche die Paketnummer, die Bytegröße der gesendeten Pakete, den Unix-Zeitstempel in Millisekunden Auflösung, wann das Paket gesendet wurde, und den Paketinhalt enthält. Dabei ist das Paket vom Senderprogramm mittels Seed-Key, pseudo-randomisiert worden, was später Rückschlüsse über seine Erstellung und sein Herkunftsort erlaubt. Das Programm "Receiver", hingegen, schreibt eine Datei mit den gleichen

Informationen für die empfangenen Pakete. Hier wird u.a. auch der über NTP (Network Time Protocol) zeitsynchronisierte Unix-Zeitstempel des empfangenen Pakets aufgezeichnet.

Die beiden Programme "Sender" und "Receiver", waren an beiden Enden des HCC-DT-Simulator-Netzes miteinander verbunden. Sender lief auf einem Rechner in der physischen Bodenstation, und der Empfänger wurde in einem virtuellen Cockpit gestartet.

Durch den Vergleich zwischen dem gesendeten und empfangenen Daten war es möglich, eine Bewertung der Netzwerkleistung durchzuführen. Mehrere Metriken wurden berechnet, darunter die Gesamtzahl der gesendeten und empfangenen Pakete, die Anzahl der Pakete, die unverändert oder modifiziert ankamen, die Laufzeit jedes Pakets, die durchschnittliche Laufzeit, die Standardabweichung der Laufzeiten, der Paketverlust, der Durchsatz und die Fehlerrate. Der Durchsatz ist ein Maß für die erfolgreich empfangenen Bits pro Sekunde und die Fehlerrate beschreibt die Ratio der Anzahl modifizierter Pakete über die Gesamtzahl aller angekommenen Pakete.

Zusätzlich wurden die Laufzeitverteilung, die Paketgrößenanalyse, die Zeitreihenanalyse der Laufzeiten und die Jitterberechnung durchgeführt. Die Laufzeitverteilung stellt die Streuung oder Variation der Zeit dar, die Pakete benötigen, um in einem Netzwerk vom Sender zum Receiver zu gelangen. Sie gibt Aufschluss über die Konsistenz und Effizienz der Datenübertragung, indem sie zeigt, wie sich die Laufzeiten der einzelnen Pakete unterscheiden. Die Zeitreihenanalyse der Laufzeiten (Order-based) untersucht die Abfolge der Laufzeiten von Paketen anhand ihrer Übertragungsreihenfolge und ermöglicht die Identifizierung von Trends, Mustern oder Anomalien im Zeitverlauf. Sie bietet einen chronologischen Überblick über die Entwicklung oder das Verhalten von Übertragungsverzögerungen bei aufeinander folgenden Paketen. "Order-based" bezieht sich auf die Analyse von Daten auf der Grundlage der Reihenfolge, in der die Pakete auftreten. Jitter ist ein Maß für die Schwankungen der Transitzeiten zwischen aufeinander folgenden Paketen. Es ist von entscheidender Bedeutung für die Stabilität von Echtzeitanwendungen wie Packet Radio.

6.2 Test-Parameter

Im Vorgriff auf den bevorstehenden ISS/APRS Testbetrieb mit dem HCC-DT-Simulator wurden Testparameter gewählt, die einen typischen ISS-Überflug beim Senden von APRS-Paketen abbilden. Der APRS-Repeater der ISS sendet die zu übertragenden Daten mit 1200 bps in AFSK-Modulierung (AFSK: Audio Frequency Shift Keying) unter Verwendung des AX.25 Data Link Layer Protokolls. APRS-Nachrichten werden von Funkamateuren üblicherweise absichtlich klein gehalten, um die Wahrscheinlichkeit einer erfolgreichen Übertragung zu erhöhen. Demzufolge wurde die Senderapplikation so konfiguriert, dass sie 475 pseudo-randomisiert generierte Pakete mit einer Größe von 256 Bytes in einem Intervall von 400 Millisekunden zwischen den einzelnen Paketen sendete. Die gewählte Rate und Paketgröße entspricht etwa der doppelten durchschnittlichen Bytegröße typischer APRS-Nachrichten, die von der ISS weitergeleitet werden. Die APRS-Pakete kommen für gewöhnlich in einer Taktfrequenz zwischen 1

Sekunde bis 10 Sekunden mit einer variablen Packetgröße zwischen 80 bis 128 Bytes an.

6.3 Test-Ergebnisse

Der Test dauerte insgesamt 190,2 Sekunden, wähen dessen 475 Pakete übertragen wurden. Die grundlegenden Metriken zeigten, dass alle von dem Programm „Sender“ gesendeten Pakete unverändert mit der vorgegebenen Größe von 256 Bytes zum Programm „Receiver“ ankamen, also mit einer Fehlerquote von 0 %. Die durchschnittliche Transitzeit der Pakete betrug 48,5 Millisekunden, mit einer Standardabweichung der Laufzeiten von 14,8 Millisekunden. Bei dem durchgeführten Test betrug der gemessene Jitter 13 Millisekunden.

Der Durchsatz (Throughput) betrug in diesem Versuch 5278.85 Bps. Die effektive Bandbreite des Netzes betrug während des Tests also mindestens 5.28 Kbit/s. Hier ist es aber wichtig zu beachten, dass die tatsächliche Bandbreite des Netzwerks wesentlich höher sein könnte, da das Design des Senders/Receiver Versuchs nur auf den zweifachen Wert des maximal zu erwartenden Datendurchsatzes des APRS-Repeater der ISS limitierend ausgelegt wurde.

Das Histogramm in Bild. 6 veranschaulicht die Verteilung der Transitzeiten für die Pakete. Die meisten Pakete haben Laufzeiten, die sich um den Durchschnittswert von $48,5 \pm 14,8$ Millisekunden gruppieren.

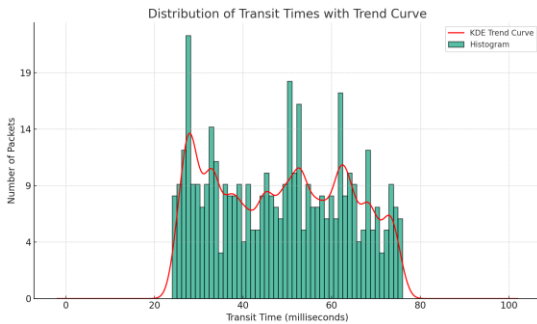


BILD. 6 Transiente Zeitverteilung

Das Zeitreihendiagramm in Fig. 6 zeigt die Laufzeiten für jedes Paket in der Paketsequenz. Es wurde beobachtet, dass etwa die ersten 30 Pakete starke Schwankungen aufwiesen.

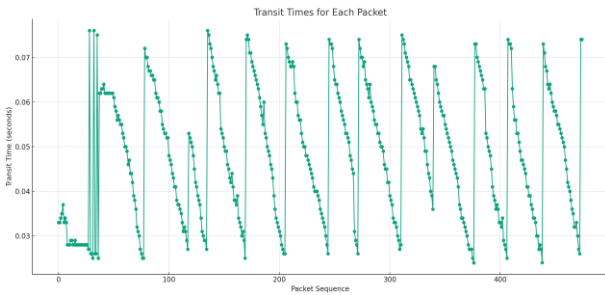


BILD.7 Transitzeiten - Zeitreihen

Die anfänglichen Schwankungen in BILD 7 könnten auf die Netzwerkinitialisierung zurückzuführen sein, bei der die Netzwerkgeräte zu Beginn einer Verbindung Prozesse wie ARP-Auflösung (ARP: Address Resolution Protocol), Routing-Pfadbestimmung oder Verbindungsaufbau durchlaufen. Diese anfänglichen Prozesse können zu Verzögerungen oder Schwankungen der Paketlaufzeiten führen. Andere Ursachen

können durch den sogenannten "TCP Slow Start" entstehen. Bei TCP-Verbindungen wird ein Mechanismus verwendet, der die Kapazität des Netzwerks überprüft. In dieser Phase beginnt TCP mit einem kleinen Staufenster und erhöht die Fenstergröße allmählich, bis es die optimale Kapazität des Netzwerks gefunden hat oder bis Paketverluste auftreten. Diese anfängliche Erkundungsphase kann zu unterschiedlichen Laufzeiten führen.

Nach dieser Initialisierungsphase, die etwa 12,01 Sekunden dauerte, schwanken die Transitzeiten im Allgemeinen nur noch um den zuvor berechneten Mittelwert von etwa 48,5 Millisekunden. Das Diagramm zeigt, wie die Transit-Zeiten der Pakete von Höchstwerten zwischen 75-80 Millisekunden innerhalb der nächsten 10-30 Pakete linear auf einen Wert von etwa 20-30 Millisekunden heruntergehen, steigen dann aber wieder abrupt auf den Höchstwert. Es zeichnet sich hierbei ein periodisches Muster einer Kippfunktion (Sägezahnform) in den Transitzeiten ab. Der Grund dieses Verhaltens muss noch weiter untersucht werden.

In der nächsten Graphik ist der Jitter für jedes Paket als Zeitreihe dargestellt. Da die ersten 30 Pakete Teil der oben besprochenen Initialisierungsphase des Experiments waren, wurden diese Pakete entfernt. Die rote gestrichelte Linie stellt den durchschnittlichen Jitter dar und die grünen gestrichelten Linien stellen die Standardabweichung über und unter dem durchschnittlichen Jitter dar:

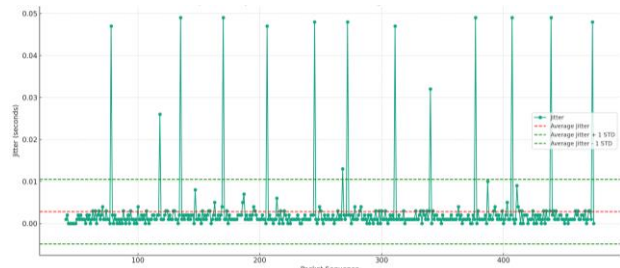


BILD. 8 Jitter-Diagramm über alle Pakete nach der Initialisierungsphase

Der mittlere Jitter über alle Pakete nach der Initialisierungsphase beträgt ca. $2,82 \pm 7,68$ Millisekunden. Der durchschnittliche Jitter und seine Standardabweichung verringerten sich leicht von $3,34 \pm 9,2$ Millisekunden auf $2,82 \pm 7,68$ Millisekunden, nachdem die ersten 30 Pakete entfernt wurden. Dies könnte darauf hindeuten, dass die ersten Pakete zusätzlichen Jitter verursacht haben, wie bereits weiter oben diskutiert wurde.

6.4 Diskussion der Ergebnisse

Die Leistungsfähigkeit des Netzwerks ist entscheidend für seine Eignung als Kontrollzentrum für die Raumfahrt. Die Durchführung dieses Experiment wurde auf die Anforderungen für einen APRS-Überflug mit der ISS zugeschnitten.

APRS liegt im AX.25-Format vor und ist mit 1200bauds AFSK-moduliert. In diesem Fall kann von einer Eins-zu-eins-Zuordnung zwischen Symbolen und Bits ausgegangen werden. Dies entspricht einer Bitrate von 1200 bps. Bei dieser Bitrate beträgt die Übertragungszeit für ein 80 Byte großes Paket ca. 533 Millisekunden und für ein 128 Byte großes Paket ca. 853 Millisekunden. Vergleicht man diese Zeiten mit den ermittelten Netzwerk Metriken, so liegt die mittlere Transitzeit von 48,5 Millisekunden deutlich unter den üblichen Übertragungszeiten von APRS-Paketen. Dies deutet darauf

hin, dass das Netzwerk in der Lage ist, die empfangenen Pakete effizient und mit minimaler zusätzlicher Verzögerung weiterzuleiten. Darüber hinaus beträgt der durchschnittliche Jitter nur 2,82 Millisekunden. Dieser geringe Jitter bedeutet, dass das simulierte HCC-DT-Netzwerk eine konsistente Kommunikation ermöglicht, die für Echtzeitanwendungen geeignet ist. Darüber hinaus zeigten die Daten keine Paketverluste oder Fehler, was auf die Zuverlässigkeit des Netzwerks hinweist.

Zusammenfassend kann festgestellt werden, dass das Netzwerk für eine APRS-Paketübertragung im AX.25-Format mit der ISS bei 1200 Baud gut geeignet ist, insbesondere für Paketgrößen zwischen 80 und 128 Byte. Weitere Netzwerk-Tests sind notwendig, um das Netzwerk auf die Eignung von Raumfahrteinsätzen mit deutlich höherem Durchsatz, wie es bei der CubelSL Mission gefordert ist, zu charakterisieren.

7. TESTBETRIEB DES VIRTUELLEN KONTROLLZENTRUMS MIT HILFE DER ISS

Der Testbetrieb wurde in zwei Testphasen aufgeteilt, die nun im Folgenden beschrieben werden.

7.1 Vorbereitung des virtuellen Kontrollzentrums und der Bodenstation

Die erste Testkampagne fand während einer drei Monateigen kostenneutralen Testphase des IaaS-Anbieters im Januar 2023 statt. Gegenstand des Tests war die Systemarchitektur und die Netzwerkmechanismen des Virtuellen Kontrollzentrums. Hierfür wurde eigens ein vLAN eingerichtet, so dass dieser Test als eigenständige Mission betrachtet wurde. Für den Betrieb wurden daher zwei VMs eingerichtet, ein Cockpit für die Bodenstation und ein Cockpit für den Satellitenbetrieb. Für die Einrichtung der Cockpits wurde ein Tool-Set für Amateurfunksatelliten verwendet, wie es in Kapitel 6 vC-MCS beschrieben ist.

Danach wurde die physische Bodenstation vorbereitet. Hier wurde zunächst eine WebSDR Station aus Goonhilly (GB) im Webbrowser geöffnet. WebSDR ist eine Online-Plattform, mit der Radioquellen von einem SDR-Receiver über den Webbrowser empfangen werden können. Mit WebSDR können Radiosignale empfangen aber nicht gesendet werden. WebSDR Receiver können durch sog. „Cat-Control“ (Computer Aided Transceiver Control) gesteuert und konfiguriert werden. Es handelt sich hier um eine Schnittstelle, die es ermöglicht, den Receivermodus zu ändern und Frequenzen zu wechseln, um beispielsweise die infolge von Dopplerverschiebung beeinflusste Empfangsfrequenz des ankommenden Signals von der ISS nachzuführen. Zu diesem Zweck wurde eine vom Autor entwickelte Anwendung verwendet, welche die Rigctld Steuerbefehle für WebSDR übersetzt (S. Tholl, Hamlib2WebSDR).

Als nächstes wurde das Soundmodem gestartet. Damit die von WebSDR empfangenen demodulierten Funksignale als Audiosignale zum Soundmodem gelangen, wurde der virtuelle Audiotreiber Virtual Cable (vbCable) verwendet. Virtual Cable ist ein virtueller Audiotreiber, der Audiosignale zwischen verschiedenen Anwendungen auf einem Computer routet. Hier wurde es verwendet, um den digitalen Audioausgang des WebSDR mit dem digitalen Audioeingang des Soundmodems zu verbinden und die Dekodierung der AFSK-Signale zu ermöglichen.

Anschließend wurde eine Linux-VM mit dem Rotctld-Hamlib-Daemon gestartet. Dabei wurde Rotctld für die Interaktion mit einem Dummy-Antennenrotor konfiguriert, da WebSDR keine Antennenrotorsteuerung zur Verfügung stellt. Hier wurde lediglich die potenzielle Fähigkeit einer Antennenrotorsteuerung per Fernzugriff über das virtuelle Kontrollzentrum getestet.

Die Netzwerk-Verbindung zwischen dem virtuellen Kontrollzentrum und der physischen Bodenstation wurde mittels Remote SSH Port Forwarding realisiert. Diese Methode ermöglicht im Allgemeinen die Umleitung eines lokalen Ports auf einem Computer hin zu einem Port auf einem entfernten Server unter Verwendung von SSH als Tunnel und erlaubt einen sicheren Zugriff auf Dienste auf dem entfernten Server, als wären sie lokal verfügbar. Mit Hilfe der SSH-Portweiterleitung wurden sichere Tunnelverbindungen eingerichtet, die es EasyTerm und OnlineKISS ermöglichten, über das WAN mit dem Soundmodem auf dem physischen Rechner der FHASOF-Bodenstation zu kommunizieren. Gpredict wurde ebenfalls über das WAN mit Hamlib verbunden. Um die Übertragung der Pakete von der physischen Bodenstation zum virtuellen Kontrollzentrum qualitativ analysieren zu können, wurden die vom Soundmodem erfolgreich dekodierten APRS-Pakete sowohl auf der Bodenstation als auch im vC-MCS Cockpit archiviert.

7.2 Performance des Überflugs der ISS über Goonhilly

Der Test wurde mit einem Überflug der Bodenstation Goonhilly (UK) mit einer erwarteten maximalen Elevation von 82° durchgeführt. Es wurde darauf gewartet, dass die ISS AFSK-Signale über Goonhilly sendet. Die ersten Pakete trafen bei 4° Elevation ein. WebSDR demodulierte die empfangenen AFSK-Audiosignale und speiste sie in das Soundmodem ein. Das Soundmodem dekodierte daraus die APRS-Pakete als Textnachrichten. Diese wurden in den entsprechenden Remote SSH Port Forwarding Tunnel eingespeist und zum vC-MCS Cockpit weitergeleitet. Das letzte Paket wurde bei einer Elevation von 7° empfangen.

Die Fernsteuerung des Dummy-Antennenrotors und von WebSDR funktionierte während des Überflugs wie erwartet ohne bemerkbare Verzögerung. Der WebSDR-Empfänger erhielt zu jedem Zeitpunkt des Überflugs die korrekte Dopplerverschiebung vom virtuellen Kontrollzentrum. Somit konnten auch alle im Spektrogramm gut erkennbaren AFSK-Radiosignale demoduliert und von Soundmodem dekodiert werden.

7.3 Bewertung des virtuellen Kontrollzentrums

Eine Analyse nach dem Überflug konnte bestätigen, dass alle von der physischen Bodenstation übertragenen Pakete vollständig und unverändert sowohl auf dem Bodenstationsrechner als auch im Archiv des vC-MCS Cockpit angelegt wurden. Leider erlauben die verwendeten Anwendungen der Amateurfunk-Missionskontrolle keine weitergehende Analyse. Es stellte sich auch heraus, dass die Systemuhr der Rechner in der Cloud gegenüber der Systemuhr der Bodenstationscomputer einen leichten Zeitversatz aufwies. Das Herausrechnen dieser Differenz wäre kein Hindernis gewesen, jedoch zeichnen die verwendeten Anwendungen die Ankunft der Pakete nicht in Millisekunden Genauigkeit auf. Dies ist aber die erforderliche Zeitauflösung, um das Netzwerk während des Versuches charakterisieren zu können.

8. TESTBETRIEB DES HCC-DT-SIMULATORS MIT HILFE DER ISS

Der HCC-DT-Simulator stand ab Mai zur Verfügung. Um die Leistungsfähigkeit des HCC-DT-Simulators zu testen, wurde es auf einem Rechner des FHASOF Bodensegments getestet.

8.1 Vorbereitung des Testbetriebs des HCC-DT-SIMULATOR

Der HCC-DT-Simulator wurde auf einem verfügbaren Rechner der FHASOF installiert, der die Funktion eines HCC-DT-Knotens und der HCC-DT-Bridges übernahm. Die UISS-Software und das Soundmodem wurden auf zwei separaten Windows-Rechnern im lokalen Netzwerk installiert. Normalerweise kommuniziert UISS direkt mit dem Soundmodem über einen TCP/IP Port. Für diesen Test wurde jedoch eine Netzwerkverbindung vom Soundmodem zur HCC-Bridge und eine weitere Verbindung von UISS zur zweiten HCC-Bridge hergestellt. Die beiden HCC-Bridges tauschten dann die zu übertragenden Daten über den HCC-DT-Simulator aus.

Um die AFSK-Signale der ISS empfangen zu können, wurde das VHF-Antennenarray auf dem Dach der FH Aachen mit dem Funkgerät ICOM IC-910H verbunden. Das ICOM IC-910H ist ein Amateurfunkgerät, das sowohl für das VHF- (2m) als auch für das UHF- (70cm) Amateurfunkband entwickelt wurde und sich besonders durch seine Fähigkeiten im Satellitenbetrieb auszeichnet. Es kann sowohl Funksignale empfangen als auch senden. Die Cat-Control erfolgte über den Rigctlid-Hamlib-Daemon. Dieser erhielt die Frequenzen für die Korrektur der Dopplerverschiebung vom Satellitentracker über eine DDE-Hamlib-Schnittstelle in Echtzeit während des Überflugs.

Um eine Übertragung der demodulierten AFSK-Signale zu gewährleisten, wurde der analoge Audioausgang des Funkgerätes über ein Audiokabel mit dem Line-Eingang der Soundkarte des Bodenstationscomputers verbunden, auf dem das Soundmodem installiert war. Zusätzlich musste im Soundmodem der Line-Ausgang als Audioquelle ausgewählt werden. Beim Test wurde darauf verzichtet, die Befehle vom Satellitentracker über den HCC-DT-Simulator zu senden. Der Grund dafür war, dass der Simulator nur zwei HCC-Bridges unterstützt, die bereits von Soundmodem und UISS belegt waren.

8.2 Der Überflug der ISS über Aachen

Während eines Überflugs über Aachen mit einer maximalen Elevation von 84° wurde auf AFSK-Signale von der ISS gewartet. Die ersten Signale wurden bei 10° Elevation empfangen. Das ICOM IC-910H demodulierte die AFSK-Audiosignale. Diese Signale wurden dann über seinen Audioausgang über das Audiokabel an den Audioeingang der Soundkarte an Soundmodem übertragen. Das Soundmodem wandelte diese Signale in APRS-Textnachrichten um. Diese Textnachrichten wurden über die HCC-Bridge in den HCC-DT-Simulator eingespeist und von dort an die andere HCC-Bridge zur UISS-Software übertragen, wie in Bild 9 gezeigt.

8.3 Bewertung des ISS-Überflugs über Aachen

Nach dem Überflug zeigte die Analyse, dass alle vom Soundmodem dekodierten Pakete korrekt und unverändert an UISS übertragen wurden. Es ging kein Paket verloren und alle waren unbeschädigt. Für weitere Analysen gelten jedoch die

gleichen Einschränkungen wie zuvor. Für den Flight Operator, der bereits mehrfach das Automatic Packet Reporting System (APRS) der ISS genutzt hat, war kein Unterschied zwischen den Bodenstationen mit und ohne HCC-DT-Simulator feststellbar.

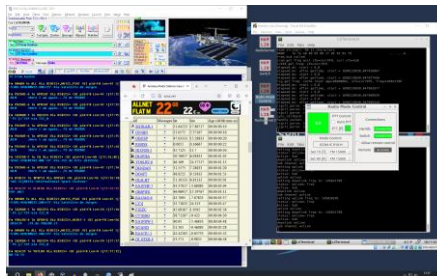


BILD.9 Screenshot von UISS während des Ende-zu-Ende Tests mit dem HCC-DT-Simulator während des Überflugs der ISS. Die UISS-Anwendung (links) zeigt empfangene APRS-Nachrichten von der ISS in Gelb und APRS-Nachrichten, welche an das

FHASOF-Boden-segment (Callsign: DL0FHA) gerichtet sind, in Rot an. Im Browser-Fenster ist das Online-Bulletin der Amateurfunk-Community zu sehen, das bestätigt, dass die von FHASOF via HCC-DT-Simulator im Uplink zur ISS gesendeten APRS-Nachrichten vom ISS-Repeater empfangen wurden.

Trotz der zusätzlichen Schritte bei der Einrichtung der Software durch die neue Komponente gab es im Betrieb keinen erkennbaren Unterschied zur gewohnten Einrichtung. Selbst wenn die Softwarekomponenten indirekt Daten austauschten, verhielten sie sich so, als würden sie direkt miteinander kommunizieren. Dies wurde als Positiv angesehen.

9. SCHLUSSFOLGERUNG

CubeSats sind aus der modernen Raumfahrt nicht mehr wegzudenken. Durch ihre kostengünstige Funktion als Technologiedemonstratoren sind sie zu einem unverzichtbaren Bestandteil der kommerziellen Raumfahrt geworden. Ihre zunehmende Popularität und Vielfalt mit ihren unterschiedlichen Anforderungen erfordert standardisierte, effiziente Betriebsinfrastrukturen. Dies wirft die Frage auf, wie CubeSats optimal in bestehende Kontrollzentren integriert werden können.

Motivation und Ziel dieses Artikels war es, eine Lösung für einen effizienteren und integrativen Betrieb von CubeSats aufzuzeigen. Das Hauptziel bestand darin, die Effizienz von CubeSat-Missionen zu erhöhen und die Betriebskosten zu senken. Es wurde deutlich, dass standardisierte Protokolle wie PUS entscheidend sind, um den CubeSat Missionsbetrieb besser in die Infrastruktur bestehender Kontrollzentren zu integrieren und den Betrieb zu vereinfachen.

Ein besonderer Aspekt dieses Artikels spiegelt die vielfältigen Möglichkeiten der CubeSat-Technologie durch die parallele Nutzung eines konventionellen Amateurfunkbodensegments und die Unterstützung professioneller, standardisierter Verfahren für den Missionsbetrieb wider. Diese Doppelstrategie schlägt eine Brücke zwischen kostengünstiger Technologie für experimentelle Projekte und den hohen Anforderungen von Raumfahrtmissionen. Auf diese Weise wird die CubeSat-Technologie zu einem vielseitigen Werkzeug, das sowohl innovative Forschung als auch einen robusten und standardisierten Betrieb ermöglicht. Dies ist insbesondere für kleinere Missionen oder Missionen mit begrenztem Budget von Vorteil.

Die Herausforderungen des CubeSat-Betriebs wurden ebenfalls hervorgehoben. Es wurde betont, wie wichtig die rasche Erfüllung dieser Aufgaben für den Erfolg solcher Missionen ist, vor allem in Anbetracht der begrenzten Lebensdauer aufgrund des Mangels an Redundanz und der zunehmenden

Komplexität der Aufgaben. Als Lösung schlagen wir ein Kontrollzentrum vor, das den gleichzeitigen Einsatz verschiedener Missionsbetriebsentitäten ermöglicht, wodurch die Missionsdauer, insbesondere bei räumlich verteilten Einsätzen, erheblich verkürzt werden kann.

In diesem Zusammenhang wird eine Systemarchitektur vorgestellt, die großen Wert auf Modularität und Anpassungsfähigkeit legt, um den vielfältigen Anforderungen von CubeSat-Missionen gerecht zu werden. Ein Schlüsselement ist der HCC-DT, ein Datenbus, der das virtuelle Kontrollzentrum von der physischen Bodenstation trennt und somit den essenziellen Datenaustausch sowie eine Dezentralisierung und Aufteilung der auszuführenden Aufgaben in mehrere Teilaufgaben zwischen den beiden Umgebungen ermöglicht. Ergänzt wird dies durch die HCC-Bridge, die als entscheidende Kommunikationsbrücke zwischen den Benutzerschnittstellen und dem HCC-DT fungiert, um eine nahtlose Software- und Hardwareintegration zwischen dem virtuellen Kontrollzentrum und den Elementen der physischen Bodenbetriebseinrichtungen zu gewährleisten.

Hinzu kommen die Cockpits, integrierte Benutzerschnittstellen, die eine individuelle und intuitive Steuerung der Mission ermöglichen. Ein weiterer wichtiger Schritt in der Architektur ist die Integration von SatNOGS, einem Netzwerk von Bodenstationen. Diese Integration stellt sicher, dass CubeSats über eine erweiterte und zuverlässige Kommunikationsabdeckung kostenneutral verfügen, die insbesondere in Zeiten, in denen der Satellit sich selbst überlassen wird, an Bedeutung gewinnt.

Um die Funktionalität und Effizienz des virtuellen Kontrollzentrums zu überprüfen, wurden sowohl qualitative als auch quantitative Tests durchgeführt. Die Ergebnisse des quantitativen Netzwerktests demonstrierten die Belastbarkeit des Netzwerks. Besonders hervorzuheben ist die Stabilität der Transitzeiten, und des niedrigen Jitters im einstelligen Millisekunden-Bereich. Dies deutet darauf hin, dass auch Echtzeitanwendungen auf dem Netzwerk betrieben werden könnten. Das System hat sich insbesondere bei den Überflügen mit dem APRS-Repeater der ISS bewährt.

Die Systemarchitektur wurde mit typischen Softwaresystemen für den Amateurfunksatellitenbetrieb in einem virtuellen Kontrollzentrum bei einem IaaS-Anbieter in Verbindung mit dem HCC-DT Simulator und einer physischen Bodenstation getestet. Es wurden zwei Überflüge mit dem APRS-Repeater der ISS absolviert. Ein Test wurde ohne das HCC durchgeführt, wobei die Verbindungen zwischen den Komponenten des virtuellen Kontrollzentrums und denen der physischen Bodenstation über SSH Remote-Port-Forwarding hergestellt wurden.

Der zweite APRS-ISS Überflug wurde mit dem HCC-DT-Simulator durchgeführt, wodurch das Konzept verifiziert werden konnte. Obwohl jedes der beiden Experimente für sich genommen nur begrenzte Erkenntnisse lieferte, zeigte ihre kombinierte Betrachtung das Potenzial des Systems und lässt den Schluss zu, dass die in diesem Artikel vorgeschlagene Systemarchitektur zur Erstellung eines ganzheitlichen virtuellen

Kontrollzentrums durchaus implementiert werden und in dieser Richtung weitergefordert werden sollte.

Es stellt sich die Frage, was passiert, wenn man sich zwischen den beiden Extremen bewegt: dem einfachen, auf Amateurfunk basierenden Missionsbetrieb und dem hoch entwickelten Missionsbetrieb. Unsere Untersuchungen deuten darauf hin, dass ein hybrider Ansatz, der Elemente aus beiden Welten vereint, das Potenzial hat, die Effizienz und Zugänglichkeit von CubeSat-Missionen deutlich zu verbessern. Damit könnten neue Perspektiven für die Raumfahrtindustrie eröffnet werden, insbesondere im Hinblick auf die Entwicklung kostengünstiger und flexibler Missionsbetriebsmodelle.

10. AUSBLICK

Das virtuelle Kontrollzentrum lief im Rahmen eines kostenlosen Abonnements mit VMs, die mit sogenannten Shared-CPU's ausgestattet waren, bei denen Prozessorressourcen mit anderen Teilnehmern geteilt wurden. Dadurch erhält man schwankende Leistung und die maximale CPU-Leistung steht nur selten zur Verfügung, was für zeitkritische Anwendungen im Missionsbetrieb problematisch sein kann. Die Migration von einem kostenlosen IaaS-Angebot zu einem Premium-Angebot wird daher eine wichtige Stellschraube sein.

Es wird ebenfalls Untersuchungen geben, wo eine lokale IaaS Cloud aufgebaut, die mittels VPN-Portal zugänglich gemacht wird.

In diesem Artikel wurde nur eine detaillierte Netzwerkcharakterisierung des HCC-DT Simulators im LAN der Bodenstation durchgeführt. Interessant wird sein, wie sich der Jitter der ausgetauschten Pakete zwischen dem virtuellen Kontrollzentrum beim IaaS-Provider und der Bodenstation verhält.

Der HCC-DT Simulator wird in naher Zukunft durch die finale Version von HCC-DT ersetzt. Es verfügt über einen deutlich erweiterten Funktionsumfang, optimierte Kommunikationsprotokolle und hohe Sicherheitsstandards.

Die verwendete Missionsbetriebssoftware archiviert ihre Daten nur im Sekundenbereich. Dies ist eine zu geringe zeitliche Auflösung für eine quantitative Bewertung des Netzes im laufenden Missionsbetrieb. Daher muss in den nächsten Versuchsreihen eine Zeitaufzeichnung im Millisekunden Bereich durchgeführt werden. Hier muss ggf. eine Missionsbetriebssoftware entwickelt werden.

In naher Zukunft wird das Projekt die Übertragbarkeit des vorgestellten Systems auf CubeSat-Flotten und komplexere Missionen untersuchen. Der Schwerpunkt der Arbeit wird auf der Anpassung des virtuellen Cockpits und des Missionskontrollsystems für CubeSat liegen.

Das virtuelle Kontrollzentrum und die vorgeschlagene Architektur stellen einen vielversprechenden Ansatz für den Betrieb von CubeSats dar. Die durchgeführten Tests unterstreichen die Effizienz und Effektivität des Systems. Es besteht jedoch weiterhin Bedarf an Forschung und Entwicklung, um das volle Potenzial des Systems auszuschöpfen.

11. REFERENZEN

- A. Csete, S. C. (2020). Evaluation of SDR Boards and Toolchains Final Report. Libre Space foundation. Noordwijk: esa. https://gitlab.com/librespacefoundation/sdrmaker-space/sdrevall/-/raw/master/Report/pdf/Evaluation_of_SDR_Boards-1.0.pdf abgerufen
- A. Hauke. (2023). GSOC's Service-Oriented Ground System "HCC" - Status and First Experiences from Sounding Rocket Missions. In S. Conf.s (Hrsg.), SpaceOps 2023, (S. 1-13). Dubai, UAE. <https://elib.dlr.de/195401/> abgerufen
- A. Hauke, M. P. (2018). Towards a Modular and Flexible New Ground System. In 2. S. Conf. (Hrsg.), Interactive-Mission Operations Systems Infrastructures and Roadmaps I (S. 1-8). SpaceOps Conf.s. doi:doi:10.2514/6.2018-2627
- A.J. Lago, D. L. (2019). In-orbit results of an ECSS Packet Utilization Standard implementation for small satellites: on-board software and mission control software for LUME-1. SSSIF2020 Spanish Small Satellites International Forum. Málaga, Spain.
- About Gpredict. (19. 10 2020). Abg. 10. 8 2023 vo <http://gpredict.oz9aec.net/>
- B. Rödiger, C. F. (2021). Miniaturized Optical Intersatellite Communication Terminal – CubeISL. 2021 IEEE International Conf. on Communications Workshops (ICC Workshops) (S. 1-5). Montreal, QC, Canada: IEEE. doi:10.1109/ICCWorkshops50388.2021.9473747
- B. Rödiger, R. R. (2022). Acquisition Concept for Inter-Satellite Communication Terminals on CubeSats. In DLR (Hrsg.), Small Satellites Systems and Services - The 4S Symposium 2022 (S. 1-12). Vilamoura, Portugal: CNES, ESA. <https://elib.dlr.de/187119/>
- Bargmann, N. (4. 2 2021). Ham Radio Control Libraries. Abgerufen 10.8 2023 <https://github.com/Hamlib/Hamlib/wiki>
- C. Laroque, C. S. (2018). Implementation of the PUS services for the ISIS product line. SpaceOps Conf.s 2018, (S. 1-12). Marseille, France. doi:10.2514.6.2018-2400
- C. M. Schieler. (2022). TBIRD 200-Gbps CubeSat Downlink: System Architecture and Mission Plan. 2022 IEEE International Conf. on Space Optical Systems and Applications (ICSOS) (S. 181-185). Kyoto City, Japan: IEEE. doi:10.1109/ICSOS53063.2022.9749714
- C. Schmidt, B. R. (2022). DLR's Optical Communication Terminals for CubeSats. IEEE International Conf. on Space Optical Systems and Applications (ICSOS) (S. 175-180). Kyoto, Japan: IEEE.
- Carrasco-Casado, M. T. (2019). Intersatellite-Link Demonstration Mission between CubeSOTA (LEO CubeSat) and ETS9-HICALI (GEO Satellite). In IEEE (Hrsg.), IEEE International Conf. on Space Optical Systems and Applications (ICSOS), (S. 1-5). doi:10.1109/ICSOS45490.2019.8978975
- D. Estevez. (2023). gr-satellites. (R. t. Docs, Herausgeber) Abg. 23. 08 2023 <https://gr-satellites.readthedocs.io/en/latest/index.html>
- D. Evans, M. M. (2014). OPS-SAT: A ESA nanosatellite for accelerating innovation in satellite control. SpaceOps 2014 Conf., (S. 1-11). Pasadena, CA. doi: 10.2514.6.2014-1702
- D.N. Amanor, W. E. (2018). Inter-Satellite Communication System based on Visible Light. IEEE Transactions on Aerospace and Electronic Systems, 54(6), 28888-2899. doi:10.1109/TAES.2018.2832938
- ECSS. (2016). ECSS-E-ST-70-41C Telemetry and telecommand packet. (e. S. ESA-ESTEC, Hrsg.) Noordwijk, The Netherlands: ESA Requirements and Standards Division.
- eoPortal. (21. 10 2016). D-SAT CubeSat mission — demonstration of a decommissioning device. (ESA, Hrsg.) <https://www.eoportal.org/satellite-missions/d-sat#context>
- F. Apper, A. R. (2019). Eye-Sat: A 3U student CubeSat from CNES packed with technology. 33rd Annual AIAA/USU Conf. on Small Satellites, (S. 1-10). Utah State University, Logan, UT.
- F. Sellmaier, T. U. (2022). Spacecraft Operations 2nd Edition (2. Ausg.). Springer.
- H. Borowski, K. R. (2010). Responsive access to space: Space Test Program Mission S26. 2010 IEEE Aerospace Conf., (S. 1-8). Big Sky, MT, USA. doi:10.1109/AERO.2010.5447016
- IARU. (2023). AMATEUR SATELLITE FREQUENCY COORDINATION REQUEST INSTRUCTIONS AND APPLICATION. (T. I. Union, Hrsg.) https://docs.google.com/viewer?url=https%3A%2F%2Fwww.iaru.org%2Fwp-content%2Fuploads%2F2020%2F06%2Fiaru_amateur_satellite_coordination_request_v40.doc
- J. Nicolas. (2021). SatNOGS: Towards a Modern, Crowd Sourced and Open Network of Ground Stations. In P. o. Conf. (Hrsg.), Second European GNU Radio Days, 2(1), S. 1-2. <https://pubs.gnuradio.org/index.php/grcon/article/view/100>
- K. Cahoy, P. G. (2019). The CubeSat Laser Infrared Crosslink Mission (CLICK). icsso proceedings Inter. Conf. on Space Optics (ICSO 2018), (S. 1-12). Chania, Greece.
- K. Devaraj. (2019). The Cubesat with One of the World's Fastest Satellite Radios. Planet Labs. Abg. 2023 <https://www.planet.com/pulse/b14-the-cubesat-with-one-of-the-worlds-fastest-satellite-radios/>
- L. Grillmayer, S. A. (2020). Integrating the Cubesat Space Protocol into GSOC's Multi-Mission Environment. 34th An. Small Sat. Conf. Utah State University, Logan, <https://digitalcommons.usu.edu/smallsat/2020/all2020/181/>
- L. Stöcken, S. W. (2018). PROCESS AS PRELIMINARY STAGE FOR SYSTEM DESIGN AUTOMATION. Deutscher Luft- und Raumfahrtkongress 2018. Meersburg, Germany.
- M. A. Boettcher, B. M. (2016). Low-Cost Approach for a Software-Defined Radio Based Ground Station Receiver for CCSDS Standard Compliant S-Band Satellite Communications. In A. V.-I. Technology (Hrsg.), IOP Conf. Series: Materials Science and Engineering (S. 1-10). IOP Publishing. doi:10.1088/1757-899X/152/1/012033
- M. Hobsch. (2022, 3 1). *Portofolio MCS*. - Mission Control and Data Systems: https://www.dlr.de/rb/PortalData/38/Resourcen/dokumente/leistungen/DLR_RB_Portfolio_MCS.pdf
- M. Geyer, M. Pecchioli, A. Walsh, J.M. Carranza, R. Blommestijn, M.C. Charneau, M. Geyer, Christian Stangl, P. Parmentier, H. Eisenman, J. Ruetting, (2012). Objectives and

- Concepts of the European Ground Systems Common Core (EGS-CC). SESP 2012: Simulation and EGSE, (S. 1-12). ESTEC Noordwijk.
- M. Motzigemba, H. Z. (2019). Optical Inter Satellite Links for Broadband Networks. 2019 9th International Conf. on Recent Advances in Space Technologies (RAST), (S. 509-512). Istanbul, Turkey. doi:10.1109/RAST.2019.8767795
- M. Pecchioli, E. M. (2010). Towards File-Based Operations for ESA Missions. (S. 1-15). SpaceOps 2010. doi:https://doi.org/10.2514/6.2010-2190
- Masutti. (2017). An international network of 50 CubeSats for multi-point, in-situ measurements in the lower thermosphere and re-entry research. VON KARMAN INSTITUTE FOR FLUID DYNAMICS, QB50. <https://cordis.europa.eu/project/id/284427/reporting/de>
- NASA. OpenMTC. Abg. 2023. 08 08 <https://nasa.github.io/openmct/>
- New Space Economy. (14. 7 2023). A TIMELINE OF THE SMALLSAT MARKET – PAST, PRESENT, AND FUTURE. <https://newspaceconomy.ca/2023/07/14/a-timeline-of-the-smallsat-market-past-present-and-future/>
- O. Sanny, B. R. (3. 3 2020). Spacecraft Collision Avoidance Using Aerodynamic Drag. Journal of Guidance, Control, and Dynamics, 43(3), 1-7. doi:10.2514/1.G004518
- P. A. Lagadrillière, T. B. (2023). Confronting or linking CSP and CCSDS? A view on how to operate small satellites today. 37th Annual Small Satellite Conf. (S. 1-7). Logan, USA: Small Satellite Conf.. <https://digitalcommons.usu.edu/smallsat/2023/all2023/173/>
- R. Arias, F. N. (2008). Lessons Learned from an Onboard ECSS PUS ObjectOriented Implementation. In AIAA (Hrsg.), SpaceOps 2008 Conf. (Hosted and organized by ESA and EUMETSAT in association with AIAA). 2008-3524, S. 1-11. AIAA. doi:10.2514.6.2008-3524
- S. Erwin. (6. 1 2021). SPACENEWS. Abg. 25. 7 2023 DARPA satellites damaged at processing facility ahead of SpaceX launch: <https://spacenews.com/darpa-satellites-damaged-at-processing-facility-ahead-of-spacex-launch/>
- S. Ilsen, D. G. (2016). PROBA: Through smart operations, small satellites can be great. SpaceOps Conf. 2016, (S. 1-17). Daejeon Korea.
- S. Janson, R. W. (2016). The NASA Optical Communication and Sensor Demonstration Program: Initial Flight Results. The Aerospace Corporation 29th Annual AIAA/USU Conf. on Small Satellites, 1-8.
- S. Salas, H. D. (2018). ANGELS SmallSat: Demonstrator for new French product. American Institute of Aeronautics and Astronautics, Inc., 1-11. doi:10.2514.6.2018-2731
- S. Tholl. (01. 08 2023). HCC-Bench - Yet another application to test your network robustness on the fly. <https://github.com/sachatholl/HCCBench> abgerufen
- Schreyer, N. E. (15. 2 2022). Wie schläft es sich im All, Herr Maurer? Abg. 20. 8 2023 https://www.fh-aachen.de/studium/studiengaenge/design-ma/aktuelles-details?tx_news_pi1%5Baction%5D=detail&tx_news_pi1%5Bcontroller%5D=News&tx_news_pi1%5Bnews%5D=20115&cHash=5a0f6df1f0eafdf91bdac7eff4db6fa
- T. J. Dijkema, M. W. (2. 8 2019). DSLWP Amateur Radio Team. Abg. 23. 8 2023 <https://www.camras.nl/en/blog/2019/dslwp-amateur-radio-team-2/>
- T. Tanaka, Y. K. (2013). Overview and operations of CubeSat FITSAT-1 (NIWAKA). 6th International Conf. on Recent Advances in Space Technologies (RAST), (S. 887-892). Istanbul, Turkey. doi:10.1109/RAST.2013.6581339
- TEB. (2016). Tailored ECSS Engineering Standards for In-Orbit Demonstration CubeSat Projects. ESA/ESTEC.
- Wikipedia. (25. 6 2023). CubeSat Space Protocol.