**Core Flight Sistemi Uygulamaları ve Sistemin CCSDS Protokolleri: Uzay Görevlerinde Modüler Uçuş Yazılım Mimarisi**

İÇİNDEKİLER

[Giriş 3](#_Toc206077997)

[CFS APLICATIONS 4](#_Toc206077998)

[Data Managment 4](#_Toc206077999)

[DS — Data Storage (Veri Depolama/Yönlendirme) 4](#_Toc206078000)

[FM — File Manager (Dosya Yöneticisi) 4](#_Toc206078001)

[CF — CFDP Application (Dosya Transferi) 4](#_Toc206078002)

[MM — Memory Manager (Hafıza Yöneticisi) 4](#_Toc206078003)

[MD — Memory Dwell (Periyodik Bellek Okuma) 4](#_Toc206078004)

[CS — Checksum (Bütünlük Denetimi) 4](#_Toc206078005)

[Advanced Operations 4](#_Toc206078006)

[SC — Stored Command (Depolu/Zamanlanmış Komut) 4](#_Toc206078007)

[SCH — Scheduler (Zamanlayıcı) 5](#_Toc206078008)

[LC — Limit Checker (Sınır Denetleyici) 5](#_Toc206078009)

[HK — Housekeeping (Toplayıcı HK) 6](#_Toc206078010)

[HS — Health & Safety (Sağlık ve Güvenlik) 6](#_Toc206078011)

[CFE SERVİSLRİ 6](#_Toc206078012)

[MISSION C&DH (COMMAND & DATA HANDLING) APPS 9](#_Toc206078013)

[CFE Haberleşme 12](#_Toc206078014)

[CAN – CAN BUS 17](#_Toc206078015)

# Giriş

Uçuş yazılımı, bir uzay aracının görev-kritik işlevlerini gerçek zamanlı ve deterministik biçimde yürüten; güç ve ısıl yönetimi, tutum/yörünge kontrolü, haberleşme, veri toplama/işleme ve anomaliden toparlanma gibi alt sistemleri güvenli bir bütün içinde koordine eden temel yazılım katmanıdır. Yer segmentiyle etkili entegrasyon bu sürecin ayrılmaz parçasıdır; bu bağlamda COSMOS, komut/telemetri akışı, paket izleme ve betik tabanlı otomasyon yetenekleriyle (örn. senaryo yürüteçleri, bekleme/doğrulama kontrolleri) cFS tabanlı uçuş yazılımlarının laboratuvar, donanım-içinde-test (HIL) ve simülasyon ortamlarında hızlı devreye alınmasını ve erken doğrulamasını kolaylaştırır. Bu çerçevede core Flight System (cFS); işletim sistemi ve donanım bağımlılıklarını soyutlayan OSAL/PSP katmanları ile çekirdek yürütme servislerini (cFE: ES, SB, EVS, TIME, TBL) birleştirerek bileşen-tabanlı, mesajlaşma-odaklı bir mimari sunar; böylece görev uygulamaları farklı platformlarda kod değişikliği olmadan derlenip çalıştırılabilir, derleme-zamanı ve çalışma-zamanı yapılandırmaları üzerinden hızla uyarlanabilir ve yer sistemleri/simülasyonlarla bütünleşik test süreçleri kısaltılabilir; sonuç olarak geliştirme ve bakım maliyetleri azalırken, ölçeklenebilirlik ve saha güvenilirliği artar.

**A diagram of a software system

AI-generated content may be incorrect.**

# CFS APLICATIONS

## Data Managment

### DS — Data Storage (Veri Depolama/Yönlendirme)

DS, gelen telemetri akışlarını mesaj kimliği ve filtre kurallarına göre dosyalara yönlendirir, oranlama/koşullu yazım uygular ve dosya rotasyonu ile depolama alanını yönetir; telemetri dosyalarının durumunu ve yazılan paket sayılarını raporlayarak uçuş verisinin düzenli arşivlenmesine ve daha sonra yere indirime hazır tutulmasına imkân verir.

### FM — File Manager (Dosya Yöneticisi)

FM, uçuş dosya sisteminde listeleme, bilgi alma, kopyalama/taşıma/yeni ad verme, silme, birleştirme ve bütünlük (checksum) gibi temel işlemleri sağlar; böylece DS’nin ürettiği dosyaların bakımı, CF ile gönderim öncesi doğrulama ve genel depolama hijyeni telemetri destekli ve komutla yönetilebilir bir şekilde yürütülür.

### CF — CFDP Application (Dosya Transferi)

CF, CCSDS CFDP protokolünü uygulayarak yer–uzay arasında güvenilir dosya gönder/al işlemleri yürütür; zaman aşımı, yeniden deneme, sınıf A/B kipleri, kanal/kuyruk yönetimi ve işlem istatistikleriyle birlikte konfigüre edilebilir dosya yolları üzerinden yazılım/konfigürasyon paketleri veya bilim verilerinin emniyetli aktarımını sağlar.

### MM — Memory Manager (Hafıza Yöneticisi)

MM, RAM/EEPROM üzerinde güvenli okuma–yazma, doldurma, kopyalama ve dosyadan yükleme/dosyaya boşaltma işlemlerini sunar; adres–uzunluk denetimleri ve sonuç telemetrisiyle birlikte hızlı yama, tanılama ve parametre ayarlamaları için düşük seviyede fakat kontrollü bir bakım arabirimi sağlar.

### MD — Memory Dwell (Periyodik Bellek Okuma)

MD, tanımlı adreslerden belirli aralıklarla değer okuyup telemetriye aktarır; hafif bir “scope” gibi çalışarak uygulama sayaçları, durum bayrakları veya hızlı değişen göstergelerin gerçek zamanda izlenmesini mümkün kılar ve dwell tablosu üzerinden örnekleme süresi/adres/uzunluk parametreleri uçuşta yeniden ayarlanabilir.

### CS — Checksum (Bütünlük Denetimi)

CS, yürütülebilir imaj, tablolar ve seçili bellek/dosya bölgelerinin CRC/bütünlük değerlerini hesaplayıp referanslarla karşılaştırarak uçuşta bozulmaları tespit eder; periyodik veya komutla tetiklenen taramalar sonucunda hangi kapsamların geçtiği/kaldığı ve kalan blok sayıları raporlanır, böylece yazılım ve konfigürasyon bütünlüğü sürekli doğrulanır.

## Advanced Operations

### SC — Stored Command (Depolu/Zamanlanmış Komut)

SC uçuşta komutları otomatik yürütmek için iki yöntemi bir arada sunar:

ATS (Absolute Time Sequence) ve RTS (Real-Time Sequence). ATS, yerden yüklenen ve uçuş zamanına “mutlak” damga vurulmuş bir zaman çizelgesidir; SC bu çizelgeyi doğrular, etkinleştirir ve her adımı zamanı geldiğinde Software Bus üzerinden yayınlar. Başlat/durdur/devam, küçük zaman kaydırmaları (bias/slip), gerektiğinde adımı atlama ve A/B bankası arasında geçiş gibi temel yetenekleri sayesinde planlı gece operasyonları gibi akışlar güvenilir ve tekrarlanabilir şekilde çalışır.

RTS (Real-Time Sequence) ise olayla tetiklenen, RAM’de tutulan kısa komut makrolarıdır; tek tek satırlara komutlar (isteğe bağlı küçük gecikmelerle) yazılır ve örneğin bir limit ihlali, sağlık kontrolü uyarısı veya başka bir adım tarafından anında tetiklenebilir. RTS’in güçlü yanı çok hızlı tepki vermesi, birden fazla RTS’in aynı anda koşabilmesi ve enable/inhibit ile kolayca yönetilmesidir; “safing” (güvenli moda geçiş) gibi hızlı toparlama adımlarında tercih edilir. SC’nin durum telemetrisi; aktif ATS ve adım numarası, bir sonraki yürütme zamanı, etkin RTS’ler ve iptal/başarısız adım sayaçları gibi alanlarla yürütmenin sağlık ve ilerleyişini net biçimde raporlar.

### SCH — Scheduler (Zamanlayıcı)

Scheduler (SCH), cFE TIME servisinden aldığı zaman tabanıyla periyodik işleri deterministik biçimde tetikleyen zamanlayıcıdır; varsayılan olarak 100 Hz’lik bir “minor frame” üzerinde çalışır (her 10 ms bir slot), bu 100 slot 1 saniyelik “major frame”i oluşturur ve major frame senkronu TIME servisinin tipik 1 Hz “tone” senkronizasyonundan türetilir; SCH iki tabloluk bir model kullanır: Message Table (yayımlanacak SB mesajlarının listesi) ve Schedule Table (her slotta hangi mesaj indeksinin, hangi periyot ve faz ofseti ile tetikleneceği), böylece her 10 ms’de aktif slottaki girişler değerlendirilir ve “enable” olan bir giriş için (slot−offset) mod period = 0 koşulu sağlandığında ilgili MsgID Software Bus’a yayımlanır (SCH doğrudan fonksiyon çağırmaz; uygulamalar SB’den “wake-up/req” mesajlarını alır); zaman kayması durumunda derleme-zamanı politikaya bağlı olarak kaçırılan slotları atlama/kaydırma (slip/skip) ya da yetişme (catch-up) davranışı uygulanır ve missed slot, sync loss gibi sayaçlar telemetride izlenir; komut seti tipik olarak NOOP/ResetCounters, belirli schedule entry enable/disable ve tablo yükleme/etkinleştirme işlevlerini içerir, sağlık telemetrileri SlotsProcessed, SlotsMissed, SyncAttempts, EntriesEnabled gibi alanlarla çizelge bütünlüğünü raporlar; yapılandırma esnekliği hem derleme-zamanında hem de çalışma-zamanında (periyot ve ofsetlerin tablo yükleriyle ayarlanması) sağlanır; özetle SCH, HK toplama, sağlık/sınır denetimleri ve veri işleme istekleri gibi tüm periyodik faaliyetleri tekrarlanabilir, fazlanmış ve gözlemlenebilir bir takvim üzerinde yürütür.

### LC — Limit Checker (Sınır Denetleyici)

LC (Limit Checker), periyodik olarak (genelde SCH tetiklemesiyle) tanımlı izleme noktalarını (watchpoint) değerlendirip, bir eylem noktası (actionpoint) mantığıyla otomatik tepki üreten koruma/otonomiya katmanıdır. Çalışma akışı şöyle işler: Uçuş yazılımın telemetri akışından (SB) veya doğrudan bellekten okunacak ölçüm/alan her watchpoint için tabloyla tarif edilir (karşılaştırma işleci: <, ≤, >, ≥, ==, !=, hedef eşik/değer). LC her döngüde ilgili değeri taze telemetriden alır, tanımlı operatöre göre PASS/FAIL sonucunu üretir, art arda kaç FAIL gördüğünü ve toplam değerlendirme sayaçlarını tutar; istenirse histerezis (farklı açma/kapama eşikleri) ve stale veri (tazelenmeyen ölçüm) yönetimi uygulanır. Her watchpoint, bir veya daha fazla actionpoint ile ilişkilidir: actionpoint ARMED durumundayken bağlı watchpoint tetik koşulunu sağlarsa TRIPPED olur ve konfigüre edilmiş aksiyonu icra eder; bu aksiyonlar tipik olarak (i) EVS olayı/kayıt, (ii) komut yayını (SB üzerinden belirli bir uygulamaya), (iii) bir RTS’in (Stored Command kısa dizisi) tetiklenmesi ya da (iv) inhibit/allow bayraklarının değiştirilmesi şeklindedir. Actionpoint’ler latch (kilitlenip sadece yer komutuyla sıfırlanan) veya auto-clear (belirli sayıda PASS görülünce kendini temizleyen) çalışacak biçimde ayarlanabilir; ayrıca global inhibit (tüm aksiyonları duraklat), per-action inhibit (yalnızca seçili eylemleri duraklat) ve test-only/monitor-only (sadece algıla, aksiyon üretme) kipleriyle güvenli devreye alma/deneme yapılır. Tüm bu davranış tablo yükleri üzerinden uçuşta değiştirilebilir: watchpoint tabloları (hangi alan, nasıl kıyas, kaç ardışık ölçümde tetikle) ve actionpoint tabloları (hangi watchpoint tetikleyince ne yapılsın, latch/auto-clear, olay düzeyi) TBL servisleri ile güncellenir. LC’nin HK telemetrisi; her watchpoint’in durumu (PASS/FAIL/STALE), ardışık PASS/FAIL sayaçları, son tetik zamanı, her actionpoint’in state’i (ARMED/TRIPPED/INHIBITED), üretilen olay/aksiyon sayıları, inhibit bayrakları ve hata kodları gibi alanlarla tam görünürlük sağlar. Özetle LC, ölçümü tanımla → kıyası yap → tetik eşiğini “sağlam” bir istatistikle doğrula → gerekiyorsa tek seferlik veya sürekli aksiyonu uygula → durumu telemetriyle raporla döngüsüyle, yanlış pozitifleri azaltan filtreleri ve uçuşta güncellenebilir tablolarıyla koruyucu mantığı güvenli ve izlenebilir biçimde yürütür.

### HK — Housekeeping (Toplayıcı HK)

HK, birden fazla uygulamadan seçili telemetri alanlarını “kopya tablosu” ile tek bir HK paketine derleyerek bant genişliği ve yer ayrıştırma yükünü azaltır; periyodik olarak toplanan bu özet, sistem sağlığının hızlı izlenmesini sağlar ve tablo güncellemeleriyle hangi alanların, hangi ölçek/formatta raporlanacağını çalışma sırasında esnekçe değiştirilebilir.

### HS — Health & Safety (Sağlık ve Güvenlik)

HS, uygulamaların canlılık göstergelerini ve kritik sistem metriklerini izler; eşik aşımları:da olay üretir, gerekirse önceden tanımlı eylemleri (ör. bir RTS ile toparlama, mod düşürme) tetikler ve watchdog mekanizmasıyla kilitlenmeleri saptar; telemetride ihlal, eylem ve izleme sayaçlarıyla birlikte sistemin operasyonel güvenliğine dair bütüncül bir görünüm sunar.

# CFE SERVİSLRİ

1. **GİRİŞ**

Uzay görevlerinde görev bilgisayarı (OBC – On-Board Computer), tüm alt sistemlerin koordinasyonunu sağlayan merkezi beyin konumundadır. Komutların yer istasyonundan alınması, alt sistemlere iletilmesi, sensör verilerinin toplanarak işlenmesi ve yer istasyonuna telemetri olarak gönderilmesi gibi kritik işlevler bu bilgisayar üzerinden yürütülür. Bu süreçlerin güvenilir, esnek ve tekrarlanabilir bir yazılım mimarisiyle yönetilmesi ise görevin başarısı için hayati önem taşır.

NASA tarafından geliştirilen core Flight System (cFS), bu ihtiyacı karşılamak üzere tasarlanmış, modüler ve taşınabilir bir uçuş yazılımı mimarisidir. cFS’nin kalbinde yer alan core Flight Executive (cFE) katmanı, uygulamaların üzerinde çalıştığı çekirdek çalışma zamanı ortamını ve görevden bağımsız temel servisleri sağlar. Böylece farklı görevler için yeniden kullanılabilen, test edilebilir ve donanımdan soyutlanmış bir yazılım altyapısı elde edilir.

Bu dokümanda öncelikle cFE’nin ne olduğu ve sunduğu temel servisler ayrıntılı biçimde incelenecek, ardından tipik bir uzay aracı görev veri işleme mimarisinde yer alan Mission Command & Data Handling (C&DH) uygulamaları detaylı olarak ele alınacaktır. Son bölümde ise, uzay görevlerinde sıkça kullanılan CAN (Controller Area Network) veri yolu teknolojisi, çalışma prensipleri ve cFS ortamına entegrasyonu üzerinde durulacaktır.

**cFE**

core Flight Executive (cFE), NASA’nın geliştirdiği core Flight System (cFS) mimarisinin çekirdek çalışma zamanı bileşenidir. Basitçe söylemek gerekirse, cFE; uzay aracındaki uygulamaların (mission apps, device drivers, servisler vb.) üzerinde çalıştığı yazılım işletim ortamıdır.

Ama klasik bir işletim sistemi gibi donanımı doğrudan yönetmez, donanım soyutlamasını OSAL (Operating System Abstraction Layer) aracılığıyla yapar. Yani cFE, donanım ve işletim sistemi (RTEMS, VxWorks, Linux vb.) ile uygulama katmanı arasında bir ara yazılım (middleware) görevindedir.

1. **Amacı**

cFE, taşınabilirlik açısından aynı kod tabanının farklı görevlerde ve farklı donanım ya da işletim sistemlerinde çalıştırılmasına olanak tanır. Örneğin, bir CubeSat projesinde geliştirilen bir uygulama ufak değişikliklerle bir derin uzay misyonuna kolayca adapte edilebilir. Yeniden kullanılabilirlik özelliği sayesinde, temel servisler standartlaştırılarak görevden bağımsız yazılım bileşenleri geliştirilebilir ve böylece görevden göreve değişmeyen “çekirdek” uygulamalar (HK, SC, TBL vb.) tekrar kullanılabilir. Modüler yapısı, yeni görev uygulamalarının (mission-specific apps) kolayca eklenip çıkarılmasına imkân verir; misyon değiştiğinde yalnızca ilgili modüller değiştirilir, çekirdek yapı aynı kalır. Güvenilirlik açısından ise, kritik görevlerde kesintisiz çalışacak sağlam bir mesajlaşma ve yönetim altyapısı sunar; hatalı bir uygulama devre dışı bırakılabilirken diğer uygulamalar sorunsuz şekilde çalışmaya devam eder.

1. **Sağladığı Servisler**
   1. **Executive Services (ES):**

Executive Services (ES), uygulamaların yaşam döngüsünü yönetir; görevleri başlatır, durdurur ve yeniden başlatır, sistemin yeniden başlatma modlarını kontrol eder ve sistem kaynaklarını takip eder. Bu kapsamda uygulama yönetimi ile uygulamaların başlatılması, durdurulması ve yeniden başlatılması; görev (task) yönetimi ile uygulama içindeki görevlerin oluşturulması, önceliklerinin ayarlanması ve yığın boyutlarının yönetilmesi sağlanır. Reset yönetimi kapsamında, yalnızca işlemciyi ve uygulamaları yeniden başlatan (RAM’in korunduğu) Processor Reset veya sistemi tamamen sıfırlayan (RAM’in temizlendiği) Power-On Reset işlemleri gerçekleştirilir. Ayrıca performans izleme özelliği ile belirli kod bloklarının çalışma süreleri performans ID’leri aracılığıyla ölçülür, hafıza yönetimi ile heap alanı kullanımı ve paylaşılan bellek yönetimi yapılır. Örneğin, kritik bir uygulama hata verdiğinde yalnızca o uygulama ES tarafından yeniden başlatılabilir ya da sistemde bellek sızıntısı veya aşırı CPU kullanımı tespit edildiğinde olay kaydı alınarak gerekli düzeltme uygulanabilir.

* 1. **Software Bus (SB):**

Software Bus (SB), uygulamalar arasında publish/subscribe yöntemi ile mesaj (paket) iletimi sağlayan ve cFE’nin kalbindeki iletişim omurgasını oluşturan bir servistir. Bu servis, CFE\_SB\_TransmitMsg() fonksiyonu ile başka uygulamalara paket yayınlama, CFE\_SB\_ReceiveBuffer() fonksiyonu ile pipe üzerinden gelen mesajları alma işlevlerini yerine getirir. Ayrıca uygulamaların belirli Message ID’lere (MID) abone olabilmesini sağlayan abonelik yönetimi, mesaj kuyruklarının oluşturulması ve kapasite ayarlanması gibi queue/pipe yönetimi özelliklerini sunar. Gönderilen ve alınan paketler, CCSDS (Consultative Committee for Space Data Systems) standardına uygun başlık ve veri alanı formatına sahiptir. SB’nin en büyük avantajı, uygulamaların birbirini doğrudan çağırmadan iletişim kurabilmesidir; bu sayede modüller bağımsız çalışır ve aynı mesaj birden fazla uygulamaya iletilebilir. Örneğin, Housekeeping (HK) uygulaması belirli periyotlarla durum telemetrisi gönderir, bu telemetri SB üzerinden Telemetry Output uygulamasına iletilir ve oradan yer istasyonuna aktarılır.

* 1. **Event Services (EVS):**

Event Services (EVS), uygulamaların çalışma sırasında oluşturduğu olay mesajlarını yönetir, filtreler ve gerektiğinde yer istasyonuna iletir. INFO, WARNING, ERROR ve CRITICAL seviyelerinde olay mesajı üretebilir. Aynı olayın gereksiz yere tekrar gönderilmesini önlemek amacıyla oran sınırlama mekanizması kullanır. Olaylar; konsol/terminal, telemetri paketleri veya dosya logları aracılığıyla dağıtılabilir. Her olay için benzersiz bir Olay ID atanarak tanımlama ve takip kolaylığı sağlanır. Örneğin, Limit Checker uygulaması sıcaklık belirli bir eşiğin üzerine çıktığında EVS üzerinden bir ERROR olayı yayınlar; bellek hatası tespit edildiğinde ise CRITICAL seviyede olay mesajı oluşturur.

* 1. **Time Services (TIME):**

Time Services (TIME), uzay aracının zamanını yönetir ve sistem genelinde doğru, senkronize zaman damgaları sağlar. Zaman kaynağı olarak GPS, yer istasyonu senkronu, 1PPS sinyali veya serbest koşu (free run) yöntemleri kullanılabilir. UTC ve TAI zaman sistemleri arasındaki farklar leap second düzeltmeleri ile yönetilir. TIME servisi, telemetri paketlerine otomatik olarak zaman bilgisi ekleyerek veri akışının doğru sırada yorumlanmasını sağlar ve sistem içindeki tüm görevlerin ortak bir zamana göre çalışmasını mümkün kılar. Örneğin, telemetri paketlerine eklenen zaman damgaları sayesinde yer istasyonunda veriler doğru şekilde sıralanabilir; ayrıca, kayıt cihazında (Recorder Manager) zaman bazlı veri kesitleri alınabilir.

* 1. **Table Services (TBL):**

Table Services (TBL), uygulamaların çalışma zamanında kullandığı parametre tablolarının yönetilmesini sağlar. Yer istasyonundan gelen dosyalarla tablolar güncellenebilir (Tablo Yükleme) ve yüklenen tabloların uygulamanın beklediği formatta olup olmadığı kontrol edilir (Doğrulama). Geçerli tabloyu aktif hale getirme (Etkinleştirme) ve mevcut tablo içeriğini dosyaya kaydetme (Döküm) işlevlerini destekler. Ayrıca, aktif ve inaktif kopyalar arasında güvenli geçiş yapılabilmesi için kopya yönetimi sunar. Örneğin, Limit Checker uygulamasının eşik değer tablosu TBL servisi aracılığıyla yerden güncellenebilir veya bir iletişim modülünün frekans ayarları tablodan okunabilir.

## MISSION C&DH (COMMAND & DATA HANDLING) APPS

Mission Command & Data Handling (C&DH) uygulamaları, bir uzay aracının yer istasyonu ile komut ve telemetri alışverişini yöneten, sistem zamanlamasını koordine eden ve kritik veri yolları ile alt sistem iletişimini sağlayan yazılım bileşenleridir. Bu uygulamalar, cFE’nin sağladığı çekirdek servisler üzerinde çalışır ve görev-özel (mission-specific) olarak geliştirilir.

C&DH katmanı, uzay aracının “merkezi sinir sistemi” gibi düşünülebilir; yerden gelen talimatların doğru alt sisteme iletilmesini, alt sistemlerden gelen durum ve bilimsel verilerin toplanıp doğru formatta yere gönderilmesini sağlar. Ayrıca zaman senkronizasyonu, veri yolu kontrolü ve hata yönetimi gibi görevler de bu katman tarafından yürütülür.

**A. Görevdeki Önemi**

Mission C&DH uygulamaları, uzay aracı ile yer istasyonu arasındaki tüm komut ve telemetri trafiğinin kontrolünü sağlar. PDU, yük (payload), haberleşme ve güç sistemi gibi alt sistemlerle veri yolu üzerinden iletişim kurarak alt sistem arabirimi görevini üstlenir. Tüm uygulamalar ve veri paketleri, ortak bir zaman referansıyla çalışacak şekilde zamanlama ve senkronizasyon süreci yürütülür. Görüş penceresi dışında toplanan veriler kaydedilir ve uygun zamanda yer istasyonuna iletilerek veri kaydı ve oynatma işlevi yerine getirilir. Ayrıca, komut doğrulama, hata tespiti ve olay raporlama gibi mekanizmalarla güvenlik ve hata yönetimi sağlanır.

**B. cFE ile İlişkisi**

Mission C&DH uygulamaları, cFE’nin sunduğu servisleri yoğun şekilde kullanarak görev verimliliğini ve güvenilirliğini artırır. SB (Software Bus), komut paketlerini dağıtır ve telemetri paketlerini toplar; EVS (Event Services), hatalar, durum değişiklikleri ve uyarılar için olay mesajları üretir; TIME (Time Services), tüm veri ve olaylara doğru zaman damgası ekler; TBL (Table Services), görev parametreleri, eşik değerler ve veri yolu haritalarını tablolar aracılığıyla yönetir; ES (Executive Services) ise uygulamaların başlatılması, durdurulması ve yeniden başlatılmasını sağlar. Bu entegrasyon sayesinde C&DH uygulamaları donanımdan bağımsız şekilde çalışabilir ve farklı görevlerde kolayca yeniden kullanılabilir.

1. **Command Ingest**

Command Ingest uygulamasının görevi, yer sisteminden (örneğin COSMOS, EGS veya özel yer yazılımları) gelen komut paketlerini almak, doğrulamak ve Software Bus (SB) üzerinden ilgili uygulamalara dağıtmaktır. Bu süreçte, TCP/UDP, seri hat, MIL-STD-1553 veya SpaceWire gibi iletişim kanallarından CCSDS uyumlu komut paketleri alınır. Paketler, CRC kontrolü, paket uzunluğu doğrulaması, APID/MID kontrolü ve komut zaman penceresi geçerliliği gibi adımlardan geçerek doğrulanır. Geçerli paketler, SB üzerinde ilgili Command MID ile yayınlanır. Hatalı paketler tespit edildiğinde ise, Event Services (EVS) üzerinden uyarı veya hata mesajı üretilir. Örneğin, yer istasyonu bir “Payload Activate” komutu gönderdiğinde, Command Ingest uygulaması bu paketi alır, doğrular ve SB’ye iletir; ardından ilgili yük uygulaması bu komutu işler

1. **Telemetry Output**

Telemetry Output uygulamasının görevi, uygulamalardan gelen telemetri paketlerini toplayarak yer istasyonuna iletmektir. Bu kapsamda, uygulama Software Bus (SB) üzerinde gerekli Telemetry MID’lere abone olur ve aldığı telemetri paketlerine CCSDS başlık ekler. Ardından, Time Services (TIME) aracılığıyla paketlere doğru zaman damgası uygulanır. Hazırlanan paketler, TCP/UDP, seri hat, MIL-STD-1553 veya SpaceWire gibi iletişim kanalları üzerinden yer istasyonuna gönderilir. Bant genişliği yönetimi sayesinde, öncelikli telemetriler (örneğin acil sağlık bilgileri) her zaman iletilirken, düşük öncelikli telemetriler bant daraldığında ertelenir veya düşürülür. Örneğin, Housekeeping uygulaması her saniye sağlık durumu paketleri üretir; Telemetry Output uygulaması bu paketleri alarak yer istasyonuna yollar.

1. **Time Manager**

Time Manager uygulamasının görevi, uzay aracının zamanını yönetmek, senkronizasyonu sağlamak ve Time Services (TIME) ile koordineli çalışmaktır. Bu amaçla, GPS, yer istasyonu senkronu veya 1PPS sinyali gibi zaman kaynaklarını kullanır. Zaman düzeltmeleri iki şekilde yapılabilir: step yöntemi ile ani düzeltme veya slew yöntemi ile kademeli düzeltme. Ayrıca Leap Second yönetimi ile UTC/TAI farkı güncel tutulur. Telemetri paketlerine eklenen zaman kalitesi bayrakları, zaman bilgisinin geçerliliğini belirtir. Örneğin, yer istasyonu bir zaman senkronizasyon komutu gönderdiğinde, Time Manager sistemi UTC’ye eşitler ve TIME servisi tüm paketlere güncel zaman damgasını ekler.

1. **1553 Bus Controller**

1553 Bus Controller uygulamasının görevi, MIL-STD-1553 veri yolu üzerinde Bus Controller (BC) rolünde çalışarak uzay aracındaki alt sistemlerle iletişim kurmaktır. Bu kapsamda, çerçeve yönetimi ile majör ve minör zaman çerçevelerini yürütür, RT (Remote Terminal) yönetimi ile alt sistemlerin adreslenmesini ve veri okuma/yazma işlemlerini gerçekleştirir. Mode Codes aracılığıyla veri yolu üzerinden mod değiştirme komutları (örneğin test modu) gönderilebilir. Ayrıca, hata yönetimi kapsamında yeniden iletim, parity hatası ve Manchester kodlama hatalarının kontrolü yapılır. Örneğin, yük alt sistemi belirli bir RT adresinde tanımlıysa, BC bu adrese periyodik olarak veri talebi gönderir ve aldığı telemetriyi Software Bus (SB) aracılığıyla ilgili uygulamalara iletir.

1. **Space Wire**

SpaceWire uygulamasının görevi, SpaceWire bağlantısı üzerinden veri alışverişini yönetmektir. Bu kapsamda, link yönetimi ile portların açılıp kapatılması, veri iletim hızının ayarlanması ve bağlantı sağlığının kontrolü yapılır. Paket yönlendirme işlevi, SpaceWire adreslerinin ilgili cFE Message ID’lerine eşlenmesini sağlayarak verinin doğru uygulamaya ulaşmasını temin eder. Ayrıca, akış kontrolü sayesinde buffer yönetimi yapılarak paket kayıpları önlenir. Örneğin, kamera modülü SpaceWire üzerinden görüntü verisi gönderdiğinde, bu uygulama paketi alır ve Software Bus (SB) aracılığıyla telemetri olarak ilgili sistemlere iletir.

1. **Power & Support**

Power & Support uygulamasının görevi, güç dağıtım birimi (PDU) veya elektrik güç sistemi (EPS) ile iletişim kurarak güç yönetimini sağlamaktır. Bu kapsamda,telemetri toplama işleviyle gerilim, akım ve sıcaklık ölçümleri elde edilir. Hat yönetimi sayesinde belirli güç hatları açılıp kapatılabilir. Koruma mekanizmaları ise aşırı akım veya gerilim durumlarında devreye girerek ilgili hatların otomatik olarak kapanmasını sağlar. Örneğin, güneş paneli voltajının düşmesi durumunda PDU, kritik olmayan alt sistemleri kapatarak enerji tasarrufu sağlar ve görev sürekliliğini korur.

**7) Recorder Manager**

Recorder Manager uygulamasının görevi, verilerin Spacecraft Data Recorder (SDR/SSR) veya başka bir kalıcı depolama cihazına kaydedilmesini ve gerektiğinde oynatılmasını yönetmektir. Bu doğrultuda, kayıt işlevi ile belirli telemetri MID’leri seçilerek depolanır, oynatma işlevi ile ise yer istasyonunun talebi üzerine kayıtlı veriler iletilir. Depolama yönetimi sayesinde, depolama alanı dolduğunda eski veriler silinebilir veya üzerine yazılabilir. Örneğin, uydu Dünya’nın görüş alanı dışındayken tüm bilim verilerini kaydeder; görüş alanına girdiğinde ise Telemetry Output aracılığıyla bu verileri oynatıp yer istasyonuna iletir.

**8) EDAC Memory Scrub**

EDAC (Error Detection and Correction) uygulamasının görevi, bellekteki tek bit hatalarını düzeltmek, çoklu bit hatalarını tespit etmek ve bu durumları raporlamaktır. Bu amaçla, tarama işlevi ile RAM bölgeleri periyodik olarak kontrol edilir. Hata düzeltme mekanizmaları olarak ECC/EDAC teknikleri kullanılarak tek bit hataları otomatik olarak düzeltilir. Ayrıca, raporlama kapsamında hata istatistikleri Event Services (EVS) ve telemetri aracılığıyla yayınlanır. Örneğin, kozmik radyasyon kaynaklı bir tek bit hatası oluştuğunda EDAC bu hatayı düzeltir ve olay EVS üzerinden kayıt altına alınır.

# CFE Haberleşme

1. **CCSDS Nedir ve Neden Kullanılır?**

Uzay görevlerinde yer istasyonu ile uzay aracı arasında güvenilir, uyumlu ve standart bir veri iletişimi sağlamak amacıyla CCSDS (Consultative Committee for Space Data Systems) standartları kullanılmaktadır. CCSDS, farklı üreticilere ait sistemlerin ortak bir dilde haberleşmesini mümkün kılar. Standart, paketlerin belirli bir başlık yapısına sahip olmasını, gerektiğinde segmentlere bölünmesini ve bu parçaların frame içinde fiziksel iletişim hattı üzerinden iletilmesini tanımlar. Böylece hem komut hem de telemetri verileri görev yazılımı ile yer istasyonu yazılımı arasında düzenli ve güvenilir bir şekilde aktarılabilir.

metin, ekran görüntüsü, sayı, numara, yazı tipi içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.

|  |  |  |  |
| --- | --- | --- | --- |
| **Alan Adı** | **Bit/Octet Boyutu** | **Açıklama** | **Amaç/Neden** |
| **Version Number** | 3 bit | Paket yapısının sürümü. | Alıcının paketi doğru sürüm formatıyla yorumlamasını sağlar. |
| **Type Indicator** | 1 bit | Paketin veri mi yoksa kontrol mü olduğunu belirtir. | Paket türünü ayırmak ve işleme kararını doğru yapmak için. |
| **Secondary Header Flag (Pckt. Sec. Hdr. Flag)** | 1 bit | Paket ikincil başlığının varlığını gösterir. | İkincil başlık varsa ek bilgiler paket içinde taşınabilir. |
| **Application Process Identifier (APID)** | 11 bit | Paket hangi uygulama sürecine ait, onu tanımlar. | Çoklu uygulamaların aynı kanalda veri gönderebilmesi için. |
| **Grouping Flags** | 2 bit | Paketin grup içindeki sırasını gösterir (ilk, devam, son, grup yok). | Paketlerin birden fazla parçadan oluştuğu durumlarda sıralama ve birleştirme için. |
| **Source Sequence Count** | 14 bit | Paketin kaynak dizin numarası. | Paketin sırasını ve eksik paket kontrolünü sağlamak için. |
| **Packet Data Length** | 2 octet | Paket veri alanının uzunluğu - 1. | Alıcının veri alanının kaç byte olduğunu bilmesi için. |
| **Packet Secondary Header** | 1–65536 octet (opsiyonel) | Paket ikincil başlık bilgileri (zaman, format, yardımcı veri). | Paketle ilgili ek bilgileri taşımak için. |
| **Source Data Field** | 1–65536 octet | Asıl veri alanı. | Gönderilen gerçek telemetri/veri bilgisi burada bulunur. |

Bu yapının kullanılma sebepleri şunlardır:

* + Minimum alan, maksimum bilgi: Bit sayıları, olabilecek en küçük boyutta gerekli bilgiyi taşımak için seçilmiştir.
  + Sabit başlık boyutu: 6 byte’lık sabit uzunluk, donanım ve yazılım tarafında parse etmeyi kolaylaştırır.
  + Modüler yapı: APID sayesinde farklı uygulamalar tek veri hattında ayrıştırılabilir.
  + Hata tespiti: Sequence Count ile paket kaybı anlaşılır, Sequence Flags ile segment birleştirme yapılır.

1. **CCSDS Veri Yapısı ve İletim Katmanları**

**Packet:** CCSDS’nin mantıksal veri birimidir; Primary Header (APID, paket tipi, uzunluk, sıra numarası), Secondary Header (opsiyonel; zaman damgası veya ek bilgiler) ve Data Field (telemetri veya komut verisi) alanlarından oluşur.

**Segment:** Büyük paketler tek seferde iletilemeyecek kadar büyük olduğunda parçalanır; Sequence Flags ile paketin tek parça (Unsegmented), ilk parça (First Segment), devam (Continuation Segment) veya son parça (Last Segment) olduğu belirtilir.

**Frame:** Fiziksel iletim birimidir; içinde bir veya birden fazla paket ya da segment bulunabilir. Frame Header (frame sayısı, sanal kanal kimliği- VCID) ve Error Control Field (CRC ile hata tespiti) içerir.

|  |  |  |  |
| --- | --- | --- | --- |
| **Katman** | **Görev** | **Örnek Protokol/Veri** | **Veri Birimi / Açıklama** |
| Application Layer | Komut/telemetri işleme | COSMOS–cFS arayüzü, CFDP | **Veri** – Kullanıcı uygulamalarının ürettiği ham veri veya mesaj |
| Transport Layer | Paketleme ve iletim | TCP / UDP | **Segment** – Veriyi ağ katmanına iletilmek üzere parçalara böler; TCP segmentleri güvenli ve sıralıdır, UDP segmentleri hızlı ama kontrolsüzdür |
| Network Layer | Yönlendirme | IP, ICMP, ARP | **Packet** – Segmentleri alır, hedef IP’ye göre yönlendirir; başlık kısmında adres bilgisi bulunur |
| Data Link Layer | Frame, hata kontrol | CRC, MAC | **Frame** – Paketleri fiziksel katmanda taşımaya uygun şekilde çerçeveler; MAC adresi ve hata kontrol bilgisi içerir |
| Physical Layer | Fiziksel taşıma | Kablo, radyo, optik | **Bit** – Frame’i elektrik, ışık veya radyo sinyali olarak iletir |

**Application Layer:** Kullanıcıya veya uygulamaya veri sağlayan en üst katmandır. CFDP (CCDSD File Delivery Protocol) gibi protokoller bu katmanda çalışır. COSMOS ve cFS haberleşmesinde, kullanıcı arayüzünden gönderilen komutlar ve alınan telemetri verileri bu katmanda işlenir.

**Transport Layer:** Uygulamalar arasındaki veri aktarımını yönetir. TCP (Transmission Control Protocol) güvenilir veri iletimi sağlar, paketlerin sırasını korur ve eksik paketleri yeniden gönderir. UDP (User Datagram Protocol) ise daha hızlıdır ancak paketlerin iletilme garantisini vermez.

**Network Layer:** Verinin bir cihazdan diğerine IP (Internet Protocol) adresi kullanarak ulaşmasını sağlar. Ağlar arası yönlendirme (routing) bu katmanda yapılır. Bu katmanda IP, ICMP (Internet Control Message Protocol) ve ARP (Address Resolution Protocol) gibi protokoller kullanılır.

**Data Link Layer:** Aynı ağ üzerindeki iki cihaz arasında doğrudan bağlantı kurar. Veriyi frame (çerçeve) formatına çevirir ve hata tespiti için CRC (Cyclic Redundancy Check) kullanır. Adresleme için MAC (Media Access Control) adresleri kullanılır. Ethernet ve Wi-Fi bu katmanda çalışır.

**Physical Layer:** Verinin fiziksel ortamda iletilmesini sağlar. Elektrik sinyalleri, ışık darbeleri veya radyo dalgaları kullanılır. Ethernet kablosu, fiber optik ve Wi-Fi bu katmana örnektir. Burada yalnızca 1 ve 0’ların taşınması gerçekleşir

1. **COSMOS ve cFS Nasıl Haberleşir?**

COSMOS ile cFS (Core Flight System) arasındaki haberleşme, genellikle ağ üzerinden TCP veya UDP protokolleri kullanılarak gerçekleştirilir. cFS, görev yazılımlarını çalıştıran uç taraf (space segment) olarak görev yapar ve CCSDS standardına uygun paketler üretir veya alır. COSMOS ise yer istasyonu (ground segment) yazılımı olarak çalışır, bu paketleri ağ üzerinden alır, çözümler ve kullanıcıya anlamlı veriler olarak sunar. Haberleşme sürecinde cFS, telemetri (TM) verilerini belirlenen port ve IP adresi üzerinden COSMOS’a gönderir. COSMOS bu verileri, tanımlı paket formatı ve CCSDS başlık bilgilerini kullanarak ayrıştırır. Komut gönderme işlemleri de ters yönde, yani COSMOS’tan cFS’e yine aynı protokol ve paket yapısı üzerinden yapılır. Böylece hem telemetri takibi hem de komut kontrolü gerçek zamanlı olarak sağlanmış olur.

1. **TCP ve UDP Kullanımı ve PDU Yapısı**

COSMOS ile cFS arasındaki veri iletimi, uygulama ihtiyaçlarına göre TCP (Transmission Control Protocol) veya UDP (User Datagram Protocol) üzerinden gerçekleştirilir. Bu protokoller, veri paketlerinin iletim yöntemini belirler ve iletişimin güvenilirliğini veya hızını etkiler.

* **UDP:** Bağlantısız bir protokoldür; paketleri hızlı şekilde iletir ancak kayıp veya sıralama bozulmalarını düzeltmez. Bu nedenle, zaman kritik ve veri kaybına toleranslı telemetri iletiminde tercih edilir.
* **TCP:** Bağlantı tabanlıdır; veri paketlerinin doğru sırada ve eksiksiz ulaşmasını garanti eder. Onaylama ve yeniden iletim mekanizmaları kullanır, bu yüzden UDP’ye göre daha yavaştır.

**PDU (Protocol Data Unit)**, protokol katmanında işlenen veri birimidir. COSMOS–cFS haberleşmesinde bir PDU, genellikle bir CCSDS paketi veya segmenti içerir. PDU, veri iletimi sırasında mantıksal bir birim olarak kullanılır ve gerektiğinde ACK (Acknowledgment) veya NACK (Negative Acknowledgment) mekanizmalarıyla güvenli iletim sağlanır.

**ACK (Onay):** Gönderilen verinin alıcı tarafından başarılı bir şekilde alındığını bildiren mesajdır. Alıcı, veri eksiksiz ve doğru olduğunda göndericiye ACK gönderir. Bu sayede gönderici, verinin güvenli bir şekilde ulaştığını bilir.

**NACK (Olumsuz Onay):** Gönderilen verinin eksik, hatalı veya bozuk olduğunu bildiren mesajdır. Alıcı bir paket hatalı veya eksik aldığında göndericiye NACK gönderir ve gönderici bu veriyi yeniden iletir.

1. **CFDP (CCSDS File Delivery Protocol)**

CFDP, uzay görevlerinde dosya aktarımı için kullanılan CCSDS standardına ait bir protokoldür. Yer istasyonu ile uzay aracı arasında büyük veri setlerinin veya görev sırasında üretilen dosyaların güvenilir bir şekilde iletilmesini sağlar. CFDP, hem acknowledged (onaylı) hem de unacknowledged (onaysız) çalışma modlarını destekler. Onaylı modda her dosya bloğu gönderildikten sonra karşı taraftan alındı bilgisi beklenir, eksik veya bozuk bloklar yeniden gönderilir. Bu yöntemle veri kaybı önlenir. Onaysız modda ise dosya hızlı şekilde gönderilir ancak eksik veri kontrolü yapılmaz, bu yöntem genellikle kritik olmayan veriler için kullanılır. CFDP, paket boyutu sınırlı olan uzay iletişim kanallarında dosyaları parçalara ayırarak (segmentation) iletir ve hedefte tekrar birleştirir. COSMOS–cFS ortamında, görev verilerinin veya yazılım güncellemelerinin uzaktan yüklenmesinde CFDP tercih edilir.

1. **Communication Flow (Veri İletim Süreci)**

Yer istasyonu ile uzay aracı arasındaki veri iletimi, katmanlı bir yapı üzerinden gerçekleşir. Kullanıcı COSMOS arayüzü üzerinden bir komut gönderdiğinde, bu komut öncelikle Uygulama Katmanı (Application Layer) tarafından işlenir. Daha sonra Transport Layer’nda TCP veya UDP protokolü ile paketlenir. Paket, Network Layer üzerinden hedef IP adresine yönlendirilir ve Data Link Layer tarafından çerçeveye dönüştürülüp hata kontrol bilgileri eklenir. Son olarak, Physical Layer veriyi kablo, radyo veya optik iletişim hattı üzerinden taşır.

Telemetri verileri uzay aracından yer istasyonuna ulaştığında aynı katmanlar ters sırayla işlenir. Bu katmanlı yapı, farklı sistemler ve protokoller arasında uyumlu ve güvenilir iletişimi sağlar.

## CAN – CAN BUS

Controller Area Network (CAN), iki telli diferansiyel iletim tekniğini kullanan, çok düğümlü, öncelik tabanlı ve hata toleranslı bir veri yolu standardıdır. İlk olarak otomotiv endüstrisi için geliştirilmiş olsa da, düşük maliyet, yüksek güvenilirlik ve basit topolojisi nedeniyle uzay görevlerinde de yaygın olarak kullanılmaktadır.

Bir uzay aracında CAN, özellikle güç sistemi (PDU/EPS), termal kontrol, sensör modülleri, mekanizma kontrol üniteleri gibi alt sistemler ile görev bilgisayarı arasındaki fiziksel iletişim hattı olarak görev yapar.

Ancak tek başına CAN, yalnızca ham veri paketlerini taşır. Bu verilerin anlamlandırılması, yönetilmesi ve görev akışına entegre edilmesi, cFE ve Mission C&DH Apps katmanları tarafından sağlanır.

* 1. **CAN’ın Temel Özellikleri**

Controller Area Network (CAN) veri yolu, tüm düğümlerin aynı iletişim hattını paylaşarak çalıştığı bir yapıya sahiptir. Veri iletimi, yayın (broadcast) yöntemiyle gerçekleştirilir; böylece gönderilen mesaj, veri yolundaki tüm düğümlere ulaşır. Öncelik mekanizması, mesaj kimlikleri (ID) üzerinden arbritaj ile belirlenir; burada daha düşük sayısal ID’ye sahip mesajlar daha yüksek öncelikli kabul edilir. Hata yönetimi açısından, CRC kontrolü, hata çerçeveleri, otomatik yeniden iletim ve hata sayaçları gibi mekanizmalar kullanılarak iletişim güvenilirliği artırılır. Klasik CAN protokolünde hız 1 Mbit/s’ye kadar çıkabilirken, CAN FD (Flexible Data-Rate) ile 64 byte’a kadar veri taşınabilir ve veri aşaması hızları daha yüksek seviyelere ulaşabilir.

* 1. **cFE ile İlişkisi**

core Flight Executive (cFE), CAN donanımını doğrudan yönetmez; bunun yerine iletişim süreci Mission C&DH uygulamaları üzerinden yürütülür. CAN hattından gelen veriler, önce C&DH uygulamaları tarafından alınır ve ardından Software Bus (SB) aracılığıyla diğer uygulamalara dağıtılır. SB’ye gönderilen komutlar ise C&DH uygulamaları tarafından CAN çerçevelerine dönüştürülerek fiziksel veri yoluna iletilir.

Bu süreçte Time Services (TIME), CAN mesajlarına zaman damgası ekler; Table Services (TBL), CAN ID ↔ Message ID eşlemelerini ve veri çözümleme formatlarını tablolarla yönetir; Event Services (EVS) ise CAN hattındaki hataları, bus-off durumlarını ve iletişim kesintilerini olay mesajı olarak raporlar.

Böylece, cFE görev yazılımı CAN protokolüne bağımlı olmadan çalışabilir. İletişim köprüsü görevini gören uygulama değiştirilerek, sistem kolayca başka veri yollarına (ör. MIL-STD-1553, SpaceWire) uyarlanabilir.

* 1. **Mission C&DH Apps ile İlişkisi**

Mission C&DH (Command & Data Handling) uygulamaları, cFE ile CAN Bus arasında yazılım köprüsü görevi görerek, uzay aracındaki veri akışının iki yönlü olarak sağlanmasını mümkün kılar.

Bu uygulamalar; komut yönlendirme işleviyle yer istasyonundan gelen ve Software Bus (SB)’ye düşen komutları CAN formatına çevirerek ilgili alt sistemlere iletir. Telemetri toplama işlevinde ise CAN üzerinden gelen sensör veya alt sistem verileri alınır, SB mesajlarına dönüştürülerek diğer uygulamalara dağıtılır. Hata izleme mekanizmaları sayesinde CAN hattındaki hata çerçeveleri, bus-off durumları ve veri kayıpları tespit edilip raporlanır. Ayrıca, oran ve öncelik kontrolü ile CAN trafiği görev ihtiyaçlarına göre sınırlandırılır ve kritik paketler önceliklendirir.

Örneğin, Power & Support App, EPS modülünden CAN üzerinden voltaj ve akım telemetrilerini alır, SB’ye iletir. Command Ingest App ise yer istasyonundan gelen “hat kapatma” komutunu SB’den alır, CAN paketine dönüştürüp PDU’ya gönderir.

* 1. **Veri Akışında cFE – C&DH – CAN Zinciri**

Yer istasyonundan gelen komut paketi, Command Ingest uygulaması tarafından alınarak Software Bus (SB) üzerinden yayınlanır. İlgili Mission C&DH uygulaması bu komutu SB’den alır ve CAN formatına dönüştürür. Ardından komut, CAN veri yoluna gönderilir ve hedef alt sistem (örneğin PDU) tarafından işlenir. Alt sistem, yanıt veya telemetri paketini CAN üzerinden geri yollar. Bu paket, C&DH uygulaması tarafından alınarak SB’ye telemetri mesajı olarak iletilir. Son olarak telemetri, Telemetry Output uygulaması aracılığıyla yer istasyonuna gönderilir. Bu süreçte cFE, tüm veri yönlendirme ve servis altyapısını sağlarken; Mission C&DH Apps format dönüşümü ve cihaz/alt sistem protokol uyarlamasını gerçekleştirir, CAN Bus ise verinin fiziksel olarak taşınmasını sağlar.

**References**

[1] NASA Goddard Space Flight Center, *core Flight System (cFS) Overview*, NASA, 2023. [Online]. Available: https://cfs.gsfc.nasa.gov/

[2] NASA, *core Flight Executive (cFE) User’s Guide*, NASA, 2022.

[3] NASA IV&V Program, *NASA Operational Simulator for Small Satellites (NOS3) Documentation*, NASA IV&V, 2021. [Online]. Available: https://nos3.readthedocs.io/

[4] Consultative Committee for Space Data Systems (CCSDS), *Packet Utilization Standard (PUS)*, CCSDS, 2020.

[5] Robert Bosch GmbH, *CAN Specification Version 2.0*, Stuttgart, Germany: Bosch, 1991.

[6] International Organization for Standardization, *ISO 11898-1:2015 — Road vehicles — Controller area network (CAN) — Data link layer and physical signalling*, ISO, 2015.

[7] European Space Agency (ESA), *SpaceWire Standard ECSS-E-ST-50-12C*, ESA, 2019.

[8] U.S. Department of Defense, *MIL-STD-1553B Standard — Aircraft Internal Time Division Command/Response Multiplex Data Bus*, 1986.

[9] CAN in Automation (CiA), *CANopen Application Layer and Communication Profile, CiA Draft Standard 301*, CiA e.V., 2021.

[10] National Instruments, *Controller Area Network (CAN) Overview*, National Instruments, 2020.

[11] R. Haywood, et al., “Fault-Tolerant Design of Spacecraft Data Handling Systems,” in *Proc. IEEE Aerospace Conf.*, 2019.

[12] Consultative Committee for Space Data Systems (CCSDS), *Space Packet Protocol (CCSDS 133.0-B-2)*, Blue Book, Issue 2, Washington, DC: CCSDS Secretariat, 2012.

[1] “NOS3 Scenario with cFS,” NOS3 Documentation. [Online]. Available: https://nos3.readthedocs.io/en/latest/Scenario\_cFS.html. [Accessed: Aug. 14, 2025].

[2] “COSMOS Protocols Documentation,” Ball Aerospace. [Online]. Available: https://ballaerospace.github.io/cosmos-website/docs/v5/protocols. [Accessed: Aug. 14, 2025].

**CCSDS Nedir?**

CCSDS (Consultative Committee for Space Data Systems), NASA, ESA, JAXA gibi uzay ajanslarının oluşturduğu, uydu ile yer istasyonu arasındaki veri iletişimini standartlaştıran protokoller topluluğudur.  
Amaç, farklı kurumların geliştirdiği sistemlerin aynı “dili” konuşmasını sağlamaktır.

CCSDS, iletişimi katmanlı bir yapıda tanımlar:

1. Space Packet Protocol → Uygulama verisinin (komut, telemetri) yapısı
2. Transfer Frame Protocol → Bu paketlerin veri linkinde taşınması
3. Physical Layer → RF, fiber, UDP gibi fiziksel iletim ortamları
4. Ek servisler → Dosya aktarımı için CFDP, zaman senkronizasyonu vb.

**Space Packet Nedir?**

Space Packet, uygulamanın ürettiği mantıksal veri paketidir.  
İki ana türü vardır:

* Telecommand (TC) → Yer istasyonundan uyduya giden komutlar
* Telemetry (TM) → Uydudan yer istasyonuna gelen durum/ölçüm verileri

Yapısı:

* Primary Header
  + Versiyon, paket tipi (TM/TC), APID (hangi uygulamaya ait), paket numarası, uzunluk
* Secondary Header (opsiyonel)
  + Görev özel bilgiler (zaman etiketi vb.)
* Packet Data Field
  + Asıl veri (komut kodu, telemetri ölçümleri vb.)

Space Packet tek başına iletilemez; bir taşıyıcıya (frame) konması gerekir.

**Transfer Frame Nedir?**

Transfer Frame, Space Packet’i fiziksel kanalda taşımak için kullanılan sabit boyutlu çerçevedir.  
Bir nevi kargo kutusu gibidir: İçine mektubu (packet) koyar, dışına adres ve kontrol bilgileri ekler.

Yapısı:

* Frame Header
  + Virtual Channel ID (VCID), Frame Count (çerçeve sırası), MAP ID
* Frame Payload
  + Space Packet’in tamamı veya bir kısmı
* Error Control Field (CRC)
  + Hata tespiti için kullanılır
* FEC kodları (opsiyonel, fiziksel katman öncesi eklenir)
  + Hata düzeltme için kullanılır

**Packet → Frame Dönüşümü Ne Demek?**

1. Uygulama (veya yer istasyonu) Space Packet üretir.
2. Link katmanı (frame builder) bu Space Packet’in önüne Frame Header, arkasına CRC ekler.
3. Böylece sabit boyutlu bir Transfer Frame oluşur.
4. Fiziksel katmanda (RF, UDP vb.) gönderilir.

Yani “dönüşüm” aslında “kapsülleme”dir. Paket yok olmaz; sadece çerçevenin içine konur.

**CRC ve FEC**

* CRC (Cyclic Redundancy Check) → Hata tespit eder. Alıcı, gelen frame’in CRC’sini kontrol eder; uyuşmazsa frame bozuk demektir.
* FEC (Forward Error Correction) → Fazladan kodlama bitleri ekleyerek hataları düzeltebilir. Uzay görevlerinde yeniden gönderim pahalı olduğundan çok önemlidir.

**Yer İstasyonundan Uyduya Akış (Telecommand Örneği)**

1. **COSMOS**: Operatör komut seçer → Space Packet (TC) oluşturulur.
2. **Link Katmanı**: Packet → Transfer Frame’e kapsüllenip CRC eklenir.
3. **Fiziksel Katman**: RF modülasyonu (gerçek görevde) veya UDP (NOS3’te) ile gönderilir.
4. **Uydu Link Katmanı**: Frame senkronu bulunur, CRC kontrol edilir, Space Packet çıkarılır.
5. **CI (Command Ingest)**: Space Packet’i alır, APID’ye göre **SB (Software Bus)**’a publish eder.
6. **SB**: Mesajı ilgili uygulamaya yollar.
7. **Application**: Komutu işler, gerekirse telemetri cevabı oluşturur.

**link service**

Gerçek uyduda bu işi modem + frame parser + packet router yapıyor,  
NOS3’te ise bunun yazılım simülasyonu link service adıyla çalışıyor.

1. **Veri alımı**
   * Yer istasyonundan gelen **CCSDS Transfer Frame**’leri UDP/TCP soketinden alır.
   * Bu veri fiziksel RF yerine ağ üzerinden gelir.
2. **Frame işlemleri**
   * **Frame Header** okunur (VCID, Frame Count).
   * **CRC kontrolü** yapılır (hatalıysa loglar veya atar).
   * **Frame Payload** içindeki **Space Packet** çıkarılır.
3. **CFS’e iletim**
   * Çıkarılan Space Packet, **Command Ingest (CI)** uygulamasına ham olarak gönderilir.
   * CI burada APID’ye bakar ve SB’ye publish eder.
4. **Ters yönde (telemetri)**
   * CFS’ten gelen Space Packet’leri alır.
   * Bunları **frame builder** ile CCSDS Transfer Frame’e sarar.
   * UDP/TCP ile yer istasyonuna yollar.

**Segment Nedir?**

CCSDS’te **segment**, **tek bir Space Packet’in tamamının bir CCSDS Transfer Frame’e sığmaması** durumunda, o packet’in parçalanarak birden fazla frame içinde taşınan **alt bölümlerine** verilen isimdir.

Yani:

* **Packet** = uygulama verisi (örneğin 8000 byte’lık bir telemetri paketi)
* **Frame** = sabit boyutlu taşıyıcı (örneğin 1115 byte)
* **Segment** = bu packet’in frame’lere sığacak şekilde bölünmüş parçaları

**Neden segment gerekiyor?**

* CCSDS Transfer Frame boyutu sabittir
* Eğer Space Packet bundan büyükse, tek frame’e sığmaz.
* O zaman:
  + İlk frame: packet’in **başlangıç segmentini** taşır.
  + Orta frame(ler): packet’in **orta segmentlerini** taşır.
  + Son frame: packet’in **son segmentini** taşır.

**Segment bilgisi nerede tutuluyor?**

* Segment durumu, **Frame Header** içindeki **First Header Pointer (FHP)** ve bazı “segmentation flags” ile belirtilir.
* Örneğin:
  + **FHP** = 0 → Bu frame’de yeni bir packet başlıyor.
  + **FHP** = 2047 → Bu frame’de yeni packet başlamıyor, devam ediyor.
  + Segment flag → “başlangıç / devam / son / tek segment” bilgisini verir.

**Hangi noktada segment halinde?**

* **Yer istasyonundan gönderirken**:  
  Link katmanındaki **Frame Builder**, Space Packet’in uzunluğuna bakar.
  + Tek frame’e sığarsa → “tek segment” olarak koyar.
  + Sığmazsa → parçalara böler, segment bilgilerini header’a yazar.
* **Uydu tarafında**:  
  **Frame Parser** gelen frame’lerdeki segment bilgilerini kullanarak parçaları birleştirir.
  + Segmentler tamamlandığında Space Packet yeniden oluşturulur.
  + Ondan sonra CI’ye verilir.

**Örnek**

Diyelim ki:

* Space Packet uzunluğu: 2500 byte
* Transfer Frame kapasitesi: 1000 byte payload

Gönderim:

1. Frame #1: Başlangıç segment (ilk 1000 byte)
2. Frame #2: Orta segment (sonraki 1000 byte)
3. Frame #3: Son segment (kalan 500 byte + boş alan)

Alım:

1. Frame Parser segment bilgilerini okur.
2. 3 frame’den veriyi sırayla birleştirir.
3. Tam Space Packet oluşunca CI’ye verir.

**Özet:**

* **Segment**, büyük bir packet’in frame’lere bölünmüş parçasıdır.
* Segment bilgisi **frame header** içinde taşınır.
* Segmentasyon **link katmanında** olur, CI bundan önceki aşamada çözülür.