

Sunil Yadav

sunilo16@yahoo.com

+91 9670809702

Yet Another Mission Control System (Yamcs)

Space
Application Service (SAS)

Before digging deeper into the application concept, It is crucial to familiarise ourself with some terminologies, that are normally used in to and from communication of remote devices to other applications.

1. Telemetry
2. Telecommand

Telemetry

Telemetry is the collection of measurements or other data at remote points and their automatic transmission to receiving equipment for monitoring.

Telemetry may be commutated to allow the transmission of multiple data streams in a fixed frame.

*A **telemeter** is a physical device used to remotely measure any quantity. It consists of a sensor, a transmission path, and a display, recording, or control device.

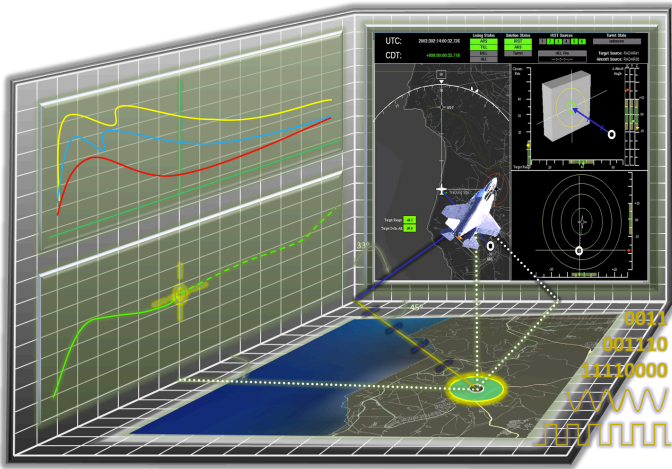
Use of Telemeters in Space Science -

Telemetry is used by manned or unmanned spacecraft for data transmission. Distances of more than 10 billion kilometres have been covered, e.g., by Voyager 1.

Protocol Used -

Space Data Link - CCSDS - 132.0.B.2.

Telecommand



A **telecommand** is a command sent to control a remote system or systems not directly connected (e.g. via wires) to the place from which the telecommand is sent. The *telecommand* can be done in real time or not depending on the circumstances (in space, delay may be off days). Remote devices are programmed to listen and understand the telecommand through which it will be performing its action.

Protocol Used -

Space Data Link - CCSDS - 232.0.B.2.

Introduction to Yamcs

Yamcs, is the central component to monitor and control the remote devices. *Yamcs stores and processes packets, and provides an interface for end-user applications to subscribe to realtime or archived data.*

Yamcs exposes its functionality over a well-documented HTTP-based API. So, tomorrow if Alpha Application wants to have the data from this repository of Yamcs, or Alpha needs the live realtime data from Yamcs, it (Alpha), can either use REST endpoints or it can subscribe to Yamcs via WebSocker and listens to realtime TM data.

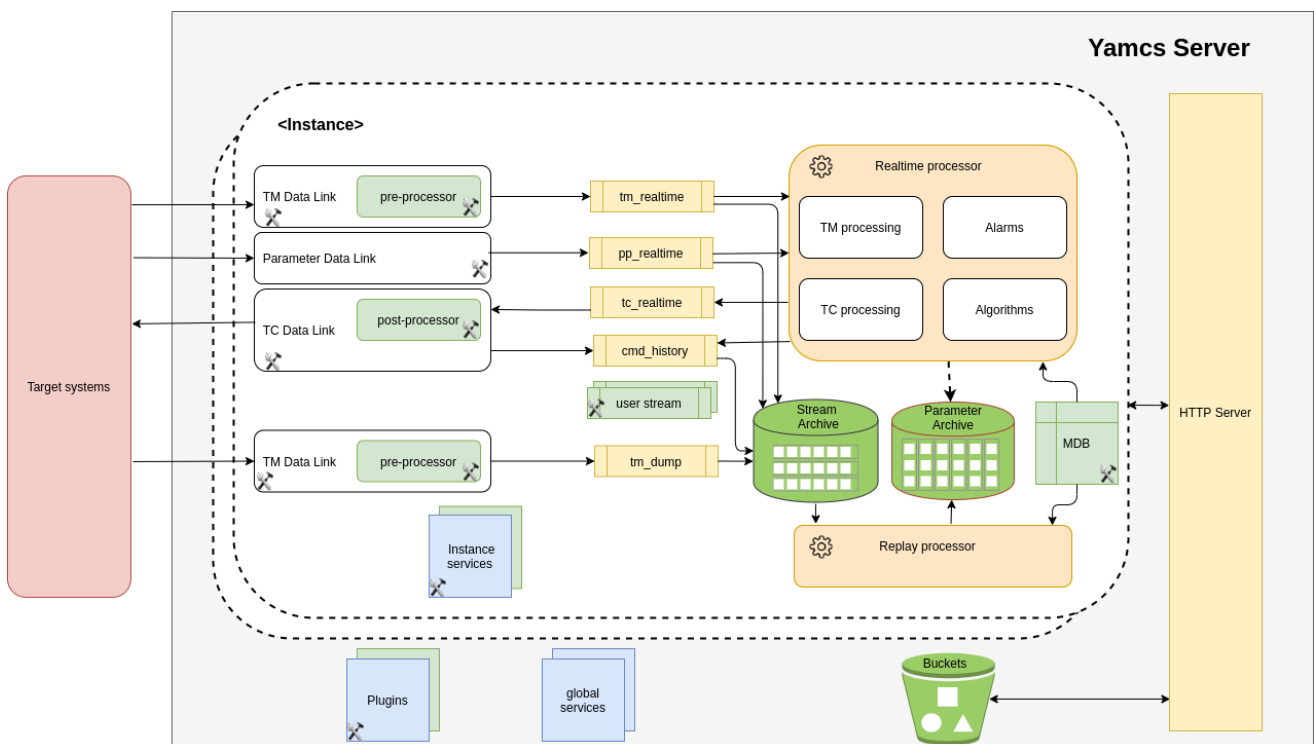
Yamcs Implementation :

Language - Java

Server - Netty Server

Database - RockDB

Preferred OS - Linux. In Yamcs, Remote



DevicesSystems is represented through a set of parameters, which are sampled at regular interval. i.e Remote device is sending its Telemetry Data at a fixed interval in particular format based on CCSDS protocol (binary packets), and Yamcs is listening to those data and parsing it.

Parameters can either be received directly from the remote device or can be computed locally by **algorithms**, I.e Engineered Parameters.

These parameters that are coming to Yamcs, has certain expiration as well, after the time is exceeded the remote devices is considered as lost and its status becomes unknown.

A telecommand is made up by a name and a number of **command arguments**.

Some commands can have an elevated **significance**, which may mean that a special privilege or an extra confirmation is required to send the command. Once the command has been sent, it passes through a series of execution stages. Users are not enforced to use these predefined-stages.

Instances in Yamcs

Yamcs application allows us to monitor/control the different payloads or satellites at same time, for each of those remote devices we have to create an instance.

Data Link

Data link is a component responsible to connect the remote device to the ground (target system). One instance can have multiple data links. Data link are of three types -

Telemetry	<i>Binary Chunk coming from remote device to the system, for mission control.</i>
Telecommand	<i>Reverse of Telemetry, this is sending instructions back to remote device after analysis.</i>
Parameters	<i>Processed Data, that helps in checking for limits and for safe keeping for future analysis.</i>

In Yamcs, there are built-in data links like one that is listening for data at port 10015 over UDP.

Streams

Streams are the components inside Yamcs, that transports tuples. While switching from One Component to other component, we can change the stream based on our requirement to get specifics data relevant to that component only.

Yamcs Processor

Processor is where most of the monitoring and control parts are done. Here Packets are transformed from binary chunks to the parameter based documents, limits and alarms are monitored, commands are generated and verified. There may exist multiple processors in single instance.

Mission Database

Mission database is the repository that holds description of telemetry and telecommand, that contains limits, alarms, algorithms, relevant constraints and pre-post verification.

Services

Yamcs, takes the configuration through Yaml, configuration file.

1. The global definition, that are needed at server level and can be accessed by all present instance is written in - `yamcs.yaml`
2. The instance specific configuration like in and out data link with their ports are written in - `yamcs.{instance}.yaml`
3. Processor specific configuration are written in - `processor.yaml`

We can change the default working nature of Yamcs application by creating our custom class that implements the defined interfaces.

Plugins

Plugins are the custom jars, that can be loaded to Yamcs, that alters with the default functionality of Yamcs, to make Mission Specific Functionality.

Stream Archive

Realtime archive, TM and TC data are inserted as soon as it gets it over the stream.

Parameters Archive

Optimised storage of parameter, for its retrieval, when asked in limited set over a longer period of time. Archive is not realtime, it gets its data after the Reply processor, processes the stream archive data.

Buckets

Data Object Storage facility, that is used to store file and documents that are received from the On-Board system. We can make use of these via REST endpoints, that Yamcs, exposed.

* Yamcs, is the server tool, that accepts data from Remote devices over CCSDS protocol, parses it, stores it and facilitates the end-user to make mission critical decisions and analysis on the data. It contains its constraints, breach of that by any parameter will create an alarm and attentions the operator for the immediate check on the remote device. It also contains Mission Specific data in order to verify and generate telecommand, that will be sent as an instruction to the remote device. Yamcs, exposes quite a lot of REST endpoints and it also facilitates for WS endpoints, that different application can make use for its analysis or visuals or for giving instruction to device.