NASA Direct Readout for its Polar Orbiting Satellites

Coronado, Patrick L., Brentzel, Kelvin W.

NASA Goddard Space Flight Center
Greenbelt MD., 20771

What it Means

The two overarching concepts that encompass the content of this chapter are Direct Broadcast and Direct Readout. Fortunately, the conceptual separation is simple; one refers to the space segment, the other to everything on the ground.

Direct Broadcast (DB) is the real-time transmission of satellite data to the ground. As the Earth is observed by satellite instruments, data is formatted and broadcast omni-directionally in a hemispheric pattern to Earth in real-time. Users who have compatible ground receiving equipment and are in direct line of sight to the satellite may receive these transmissions.

Direct Readout (DR) is the process of acquiring freely transmitted live satellite data. As DR technologies have become more affordable and accessible - such as with the onset of the Internet - tools have been developed by the remote sensing community to make satellite data easier to acquire, process, and utilize. As a member of this community, NASA supplies many of these tools to foster global data exchange and scientific collaboration. Live local and regional environmental data, in turn, benefits environmental, commercial, and public interest decision making.

With that stated, when referring to this topic, the term DR is not used, but instead, DB is the overarching descriptor. This was not done by design, but by the human need to simplify terminology. In this chapter, DB will be described briefly to differentiate it from the primary method of acquiring instrument data from Earth remote sensing satellites, such as POES, EOS series, NPP, and future NPOESS satellites. Instead, DR will be the focus where history, methods of acquisition, supporting technologies, and its roadmap will be described. Further, this chapter will address only NASA's satellites including its involvement in the POES DR system's development and how it has, and still is providing a bridge between NASA DB Earth science missions and the broader user community.

Context in History

In the late 1970s the stage was set for DB of satellite data to come into the forefront. But this stage was pre-defined by the first of ten application satellites named TIROS (Television Infra Red Observation Satellite, 1960 - 1965) as depicted in Figure 1. TIROS was launched primarily to provide images of cloud formations. The second satellite had a five-channel radiometer and an infra-red camera which could provide both day and night cloud images. The fourth satellite in this series, launched in Feb 1962, had a low data rate direct broadcast capability called Automatic Picture Transmission (APT).

Normal satellite operations involved recording the instrument measurements onto an onboard tape recorder and the data would later be dumped to a dedicated ground receiving station. This data was then operationally distributed to the broad user community by the National Oceanographic and Atmospheric Administration (NOAA). It was shortly thereafter realized that the satellites making the measurements would make excellent distribution vehicles, and thus the concept of Direct Broadcast of satellite data was developed. In this mode of operation, as the satellite measurements were made, the data was both recorded onboard the satellite and directed to a separate transmitter to be sent directly to the ground whereby any receiving station within the line of sight with the satellite could receive the data in real-time, such as APT.

These first series of satellites lead to the development of the ESSA/TOS (TIROS Operational Satellites, 1966 -1969) series of weather satellites which had advanced video cameras and APT low resolution DB capability. In late 1960's and early 1970's the spectacular Gemini and Sky-lab Photographs had sparked the imagination of the world, but more importantly, the entire science community, which was now able to clearly see the potential that satellite measurements could have on all aspects of Earth science.

During this early stage of satellite remote sensing, scientists and application engineers realized that the varied ways the electromagnetic radiation interacts with matter could provide quantitative measurements of the parameters of interest to each Earth science discipline. Detailed studies of absorption and scatter of the atmosphere were undertaken, and the interaction of radiation with various surfaces was investigated, which in turn lead to the development of unique new satellite instruments that were available via direct broadcast, all-be-it using very expensive DR systems.

The first images of the Earth used visible light-based instruments to show clouds and cloud patterns, ocean features such as boundaries, eddies and currents, and land features such as forest, grasslands and urban centers. The meteorological instruments concentrated on using infrared and microwave measurements which were designed to measure sea surface temperature, cloud top temperature and detailed atmospheric temperature profiles.

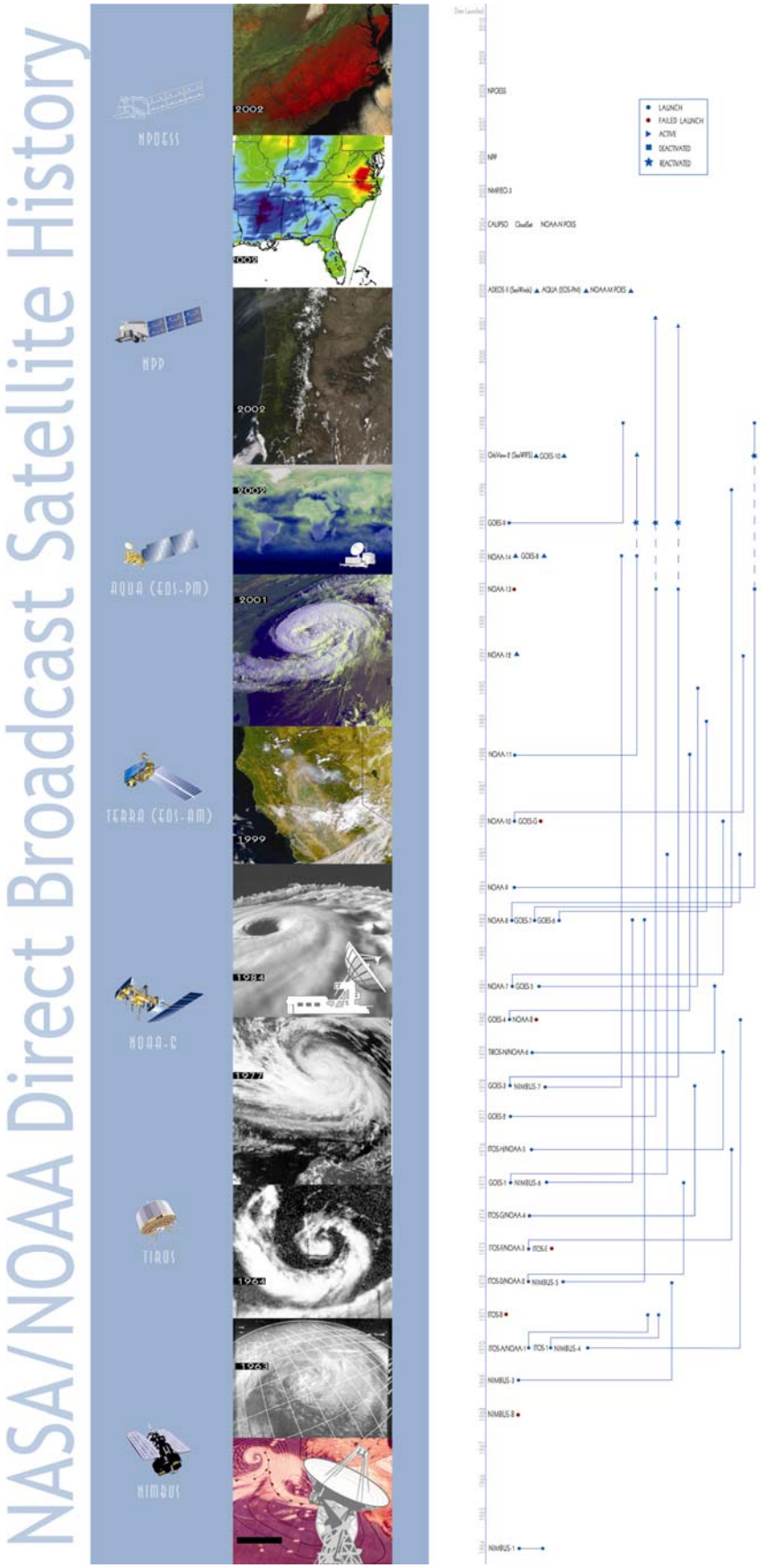
NASA/NOAA Direct Broadcast Satellite History
NPOESS
NPP
AQUA (EOS-PM)
TERRA (EOS-AM)
NOAA-G
TIROS
NIMBUS
2002
2002
2002
2001
1999
1984
1977
1964
1963
LAUNCH
FAILED LAUNCH
ACTIVE
DEACTIVATED
REACTIVATED

Figure 1

An important technology breakthrough occurred with the NIMBUS series of experimental weather satellites where many of these new and innovative suites of instruments allowed experimenters to explore the use of the entire spectrum to monitor and measure a significant range of phenomena not only of atmospheric interest, but of oceans, land and Earth's vegetation.

In 1972 the first High Resolution Picture Transmission (HRPT) DR system was developed by the NASA Direct Readout Laboratory which was established at Goddard Space Flight Center (GSFC). This HRPT capability enabled the VHRR (Very High Resolution Radiometer) and VTPR (Vertical Temperature Profiling Radiometer) data to be sent directly to DR ground receiving stations. The laboratory made investments in low noise amplifiers and down converters, inexpensive receivers, demodulators and the design and development of a multi satellite frame formatter and ingest board that resided directly on Digital Equipment Corp. VAX 's computer bus. Also at this time, the costly hardware-based image processing systems were replaced with software.

With the advent of Microvax computers, NASA was able to significantly reduce the cost of DR ground stations. A single Microvax computer could now ingest the High Resolution Picture Transmission (HRPT) data stream at 665 Kbps from the new ITOS/NOAA. At this time, the cost of a ground receiving stations was still high, in the order of $1 M to $1.5M dollars, but the expectation of a long continuous series of satellites producing significant data sets with multiple applications made the decision to invest in the new ground systems acceptable for large organizations and institutions.

In 1983, NASA GSFC and NOAA joined forces to develop a complete satellite ground system based entirely on the use of Direct Broadcast data from the ITOS. By 1989 NASA GSFC had made significant improvements to the hardware and software of DR ground receiving systems such that multiple polar orbiting satellites (DMSP- Defense Meteorological Program & TIROS) and Geo-synchronous satellite data from GOES and GMS could now be received, processed and analyzed in near real time in an automated fashion. The DMSP satellites were encrypted, but these satellites had unique microwave instruments which were based on the NIMBUS instruments. In the late 1980's to early 1990's the data products from DMSP satellites were made available to the research community. These products would eventually influence the design of the future NPOESS (National Polar Orbiting Environmental Satellite System).

During this period, many of the advanced DR ground system technologies were transferred to industry. NASA GSFC was able to demonstrate that previous ground systems costing about $1 M could now be procured for less than $300K. Shortly thereafter, the industry would further reduce the cost and complexity such that, at the time of this writing, many basic TIROS and geosynchronous satellite-based ground systems can be purchased for less than $15K as further described in Figure 2.

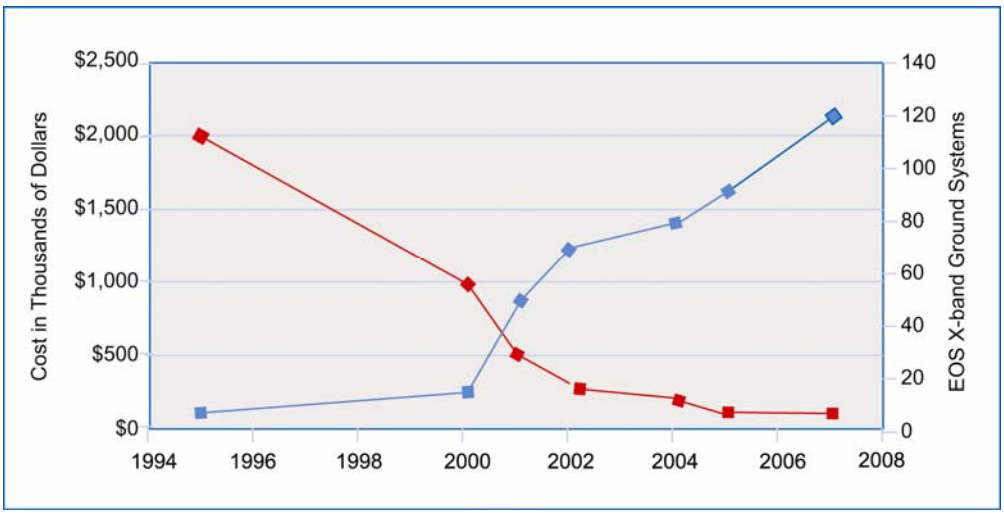

Figure 2.
Correlation of X-Band Ground System Costs to their number increase

Beginning in 1987 NASA GSFC's primary interest in DR was the demonstration of practical applications of DR ground system technologies. This included data mining, distributed processing computers, data archiving and data management technologies. These activities lead to important studies such as the relationship between sea surface temperature anomalies and climate changes (El Nino Southern Oscillation), Global deforestation, sea surface rise, volcano eruptions and sand storm dust distribution, snow and ice distributions and ozone anomalies to name a few of the scientific applications which will lead to the need for the next generation of both research (TERRA,AQUA) and operational (NPOESS) and Geostationary (GOES) satellites.

In the early 90's, the next technology project involving DR ground system development was in support of the Orbview-2 ocean color commercial satellite, formally known as SeaStar carrying the SeaWiFS ocean color instrument. In this project the development of new antenna feed horns allowed both L-band DB data and S-band satellite tape dumps to be received using the same ground system.

At this juncture, the development cycle began to repeat itself, NASA's new research DB satellites, TERRA and AQUA have state of the art instruments, such as, MODIS, HIRS & AMSU. The data rates have increased significantly to an average of 13 to 15 Mbps , and the frequency band has changed from L-band to X-Band (7.8 Ghz-8.5Ghz). Aqua and Terra are sister spacecraft with planned mission lives of 6 years. Terra was launched on December 18, 1999, followed by Aqua on May 04, 2002. Each spacecraft is in a 98-degree inclination near-polar orbit so that its motion is largely in a north or south direction when passing over the Equator. Aqua and Terra each circle the Earth 16 times daily at a nearly constant altitude of about 705 kilometers or 438 miles.

Terra follows a Sun-synchronous orbit across the Equator heading southward at about 10:30 am, local time, when cloud cover tends to be minimal and visibility greatest. Aqua crosses the Equator northbound at about 1:30 pm local time. The

crossing times were selected to gain insight into the daily cycling of precipitation, ocean circulation, and other key scientific parameters.

The next step

These new NASA experimental research spacecraft such as Terra and Aqua are continuing to make the long-term measurements which have been defined by the NASA and collaborating communities. The NOAA operational instruments have been partially defined by these prototype NASA instruments and plans are underway to launch a new generation of satellites, the National Polar-orbiting Observational Earth Satellite System (NPOESS) Preparatory Project (mission) in 2006/7, and NPOESS series between 2009 - 2010.

The NPP mission was formulated by NASA, GSFC and the National Polar-orbiting Operational Environmental Satellite System (NPOESS) Integrated Program Office (IPO).

For NASA, the NPP mission is part of the Earth Observing System (EOS) program that provides measurements of environmental phenomena identified in the EOS Science Plan. Scheduled for launch in October 2006, the NPP mission furnishes remotely-sensed land, ocean, and atmospheric data to the meteorological and global climate change communities as the responsibility for these measurements transitions from existing Earth-observing missions, e.g. Aqua, Terra, to the NPOESS.

For the IPO, the NPP mission presents an opportunity to demonstrate and validate new instruments and science data processing algorithms, as well as to demonstrate and validate aspects of the NPOESS command, control, communications and ground processing capabilities prior to the launch of the first NPOESS spacecraft. The NPP does not officially supply operational data to military users; but for all practical purposes it does so to its civilian support structure, and the general world community.

NPOESS is a tri-agency, Department of Defense (DoD), NOAA and NASA, program that merges two environmental satellite systems into one. Currently, NOAA and the Department of Defense each operate their own systems - the Polar Operational Environmental Satellites (POES) program and the Defense Meteorological Satellite Program (DMSP), respectively.

The plan is for the NPOESS constellation to consist of three polar-orbiting satellites. Each satellite will be designed for a seven-year lifetime and will host 10 to 12 sensor payloads. These sensors will provide a wide range of weather and environmental data to be disseminated worldwide, such as improved three- to five-day, seasonal and interannual weather forecasts, storm tracking, and agricultural crop management. The first launch is scheduled to coincide with the end of the expected lifetimes of the POES and DMSP constellations in 2008.

Who is the DB Community

There is significant reference to the DB community, but for many, they are difficult to identify and quantify, especially since the term DB can mean many things to many people. Within this context, a DB community member is any person or group that uses, benefits from, or contributes to the acquisition of remotely-sensed, DB data. The DB community can be classified in to two categories: DR stations and DB data users. The DR station category can be further segmented into low, mid and high data rate acquisition systems. The low rate (2.4Kbps) DR systems include the APT stations, of which there are over 4,900 world-wide. The mid data rate (665Kbps – 3Mbps) DR stations consist of the HRPT systems of which there are approximately 3000. The last segment constitute the next generation of remote sensing DR stations. These high data rate (13-25Mbps), X-band DR stations, consist of approximately 100 world-wide at the time of this writing.

The second DB community category is the derived product user. This category includes the ground station but have an additional estimated 10,000 users that make direct use of the information derived from DB data. Members of this category range from the commercial sector that provides processing components to the scientist that evaluates the most resent data acquisition results. Other members of this community include universities, the USDA forest service, fisheries, weather forecasters, DoD, NOAA, Dept of Transportation, Commerce and the Interior, NASA, amateurs and foreign meteorological offices, to name a few.

Technologies and Data Flows in DB and DR

Satellite direct broadcast is supported by technologies that are ever evolving. Nevertheless, over the last ten years and for the next twenty, these technologies will evolve within two well defined categories; data packaging and Radio Frequency (RF) communications packaging.

At NASA GSFC, the Direct Readout Project promotes continuity and compatibility among evolving EOS direct broadcast satellite downlink configurations and direct readout acquisition and processing systems. This project bridges the EOS missions with the global direct readout community by establishing a clear path and foundation for the continued use of NASA's Earth science DB data.

From Figure 3, the primary components of DB are depicted. The Consultative Committee for Space Data Systems (CCSDS) is the primary method for packaging on-board instrument data. This standard incorporates robust error-correction methods such as Reed Solomon, as well as a structure for ordering multiple data streams from one or more instruments.

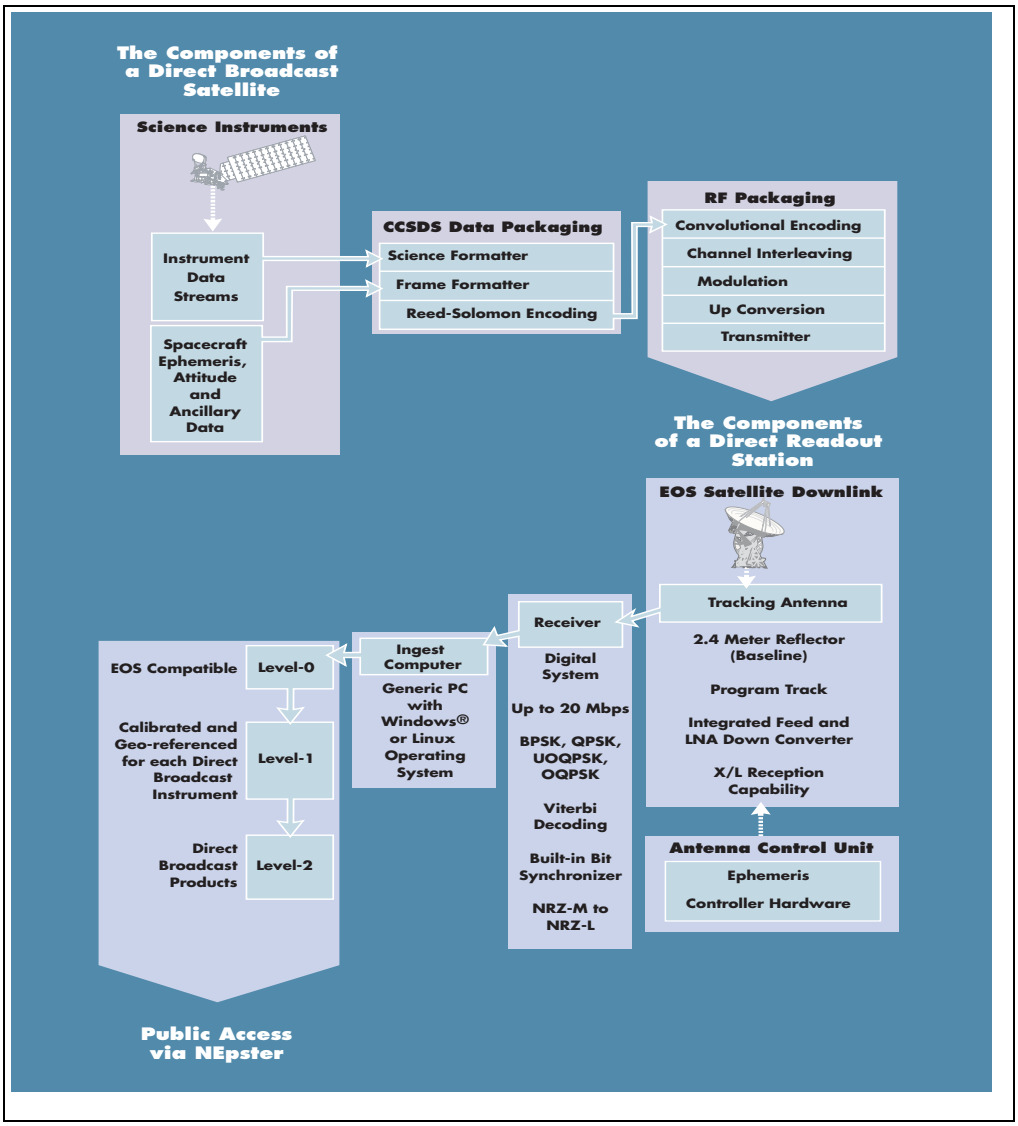

Figure 3.
Functional Components of DB through DR

The second category from Figure 3 constitutes the RF packaging and transmission. Most DB data users identify DB by the link characteristics. As described in the Context in History section of this chapter, DB has been perceptually defined by broadcasting data in the L-band frequencies at a rate of 665Kbps, which is synonymous with HRPT. But as technologies evolve, data requirements increase and new utility for the instrument data are developed, therefore new methods of DB must follow. These advancements are predominantly in the area of data encoding and in the methods for data correction. Current and future designs have settled on multi-phase encoding (QPSK, SQPSK, etc.) with convolutional (1/2 and ¾ rate viterbi) forward error correction. Additional variations of these are coming online, but most importantly, they represent a standard which enables easier transition for the DR end user on the ground from one spacecraft to the next.

Based on the improved DB transmission methods and data packet assembly schemes, the end-user has to adapt to and comply with these changes. And even

though the spacecraft changes have been intentionally designed to ease the transition to new satellites and instruments, there is still a significant expense involved with compliance. This will be most evident in the 2004 and 2005 time frame when HRPT users will be transitioning to Metop and subsequently to NPP and NPOESS LRD (Low Rate Data) DB. The Metop and NPOESS DR transition will be made easier with the establishment of The Initial Joint Polar-Orbiting Operational Satellite System (IJPS). The IJPS is the result of a cooperative effort between NOAA and the European Organization for the Exploitation of Meteorological Satellites (EUMETSAT). Transition difficulties also exist at a higher level for the HRPT users that have been migrating to EOS DB, since L-band ground system are not adequate for X-band DB.

The front-end antenna component of the DR system has been and will continue to be, a cost-driver, and therefore a significant emphasis was placed by NASA GSFC to work with the commercial sector to bring these costs down to approximately 15K for an HRPT system and 100K for an X-band system by 2004/5. In doing so, it allowed the end-user to gain further access to NASA and NOAA's real-time satellite data for entrepreneurial and real-time research applications.

Unfortunately, the challenges for end-users to keep up with the times are far from over. After contending with the expensive front-end, the end-user is confronted with several processing requirements. Additionally, software must be upgraded, hardware reconfigured, and design templates updated. In order to address these hurtles, NASA GSFC is providing the user community access to Earth remote sensing data technologies through shared information resources, including design templates that enable the DB community to receive, process, and analyze direct readout data. NASA supplies these tools in an effort to foster global data exchange and to promote scientific collaboration. In turn, real-time local and regional Earth science data benefits environmental, commercial, and public interest decision making.

A DB Model

The Direct Readout Project (DRP) at NASA GSFC plays a vital role as the intermediary between remote sensing missions, such as Terra, Aqua, and the NPP, and the DB community. The project encourages communication and maintains an open-door policy with the commercial and research and development sectors. This two-way information exchange is part of NASA's DR model, depicted in Figure 4. This model provides the DRP with critical information on the equipment currently in use, the technology being developed commercially, and the needs of the direct broadcast community. As a result, the Direct Readout Project is able to report the "state of the community" to the NASA mission while providing the DB community with a two-way dialogue between the mission objectives and the user needs.

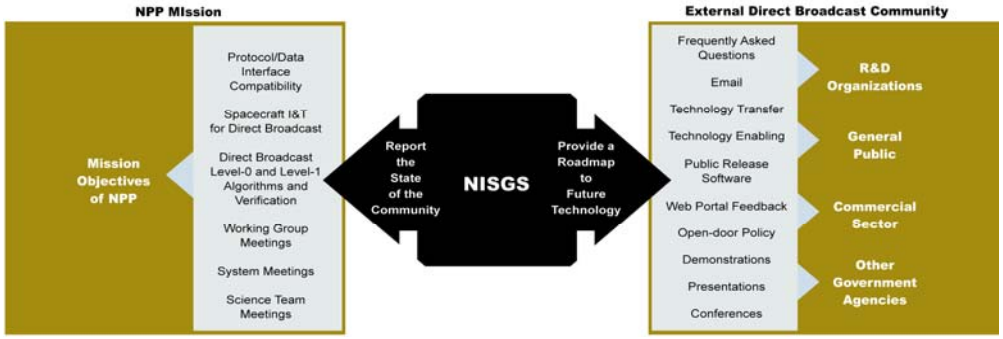

Figure 4
NASA Direct Readout Model Applied to the NPP Mission

An example of an application of the DB model, as depicted in Figure 4, is the NPP In-Situ Ground System (NISGS) development. Using this process, NISGS inherits the Direct Readout Project's knowledge, experience and "lessons learned" from prior missions, bringing continuity from legacy missions to NPP and NPOESS. This experience enables the DRP to predict problem areas in the design and development of NISGS and mitigate them before problems develop. The NISGS model is both, a system architecture, and a methodology. In this methodology, the direct broadcast community is the driving force for DB service-type requirements, which often translates to system design requirements.

The DRP also developed the Mobile NISGS, a mobile DB system, as an accessible prototype and evaluation platform. The Mobile NISGS represents the state-of-the-art technologies used by the commercial and end-user DB communities. This development system allows for immediate compatibility testing of the Direct Readout Project technologies and provides demonstrations to the end-user community and commercialization partners. The Mobile NISGS also interfaces with development DB components of the spacecraft for compatibility and validation of the RF system, protocol and data processing, and ground network interfaces. The Mobile NISGS also serves as a tool for gathering real-time Earth science data in remote areas for instrument calibration and validation campaigns.

Technology Roadmap

One of the primary results of the DR model, as described in Figure 5, is the identification of key technology categories that the DR end-user would have to contend with in order to be compliant with a multi-satellite environment. These categories include: a multi-mission autonomous task manager and dispatcher, CCSDS packet re-assembly and standard data reformatting tools, instrument-specific calibration and geo-registration algorithms, and real-time data distribution mechanisms. NASA GSFC through the DRP and the NPP mission have addressed these key technology categories in the form of specific technologies which are generic in nature and can be integrated into existing or developing DR systems.

As previously described, the DRP serves as the intermediary between missions and the user community, but more so, the project's history of involvement with direct broadcast missions ensures that the legacy of information and technology is

incorporated into future mission efforts. In this way, the DRP brings continuity, cohesiveness, and standardization to these technologies, providing the DR community with a technology roadmap as described in Figure 5.

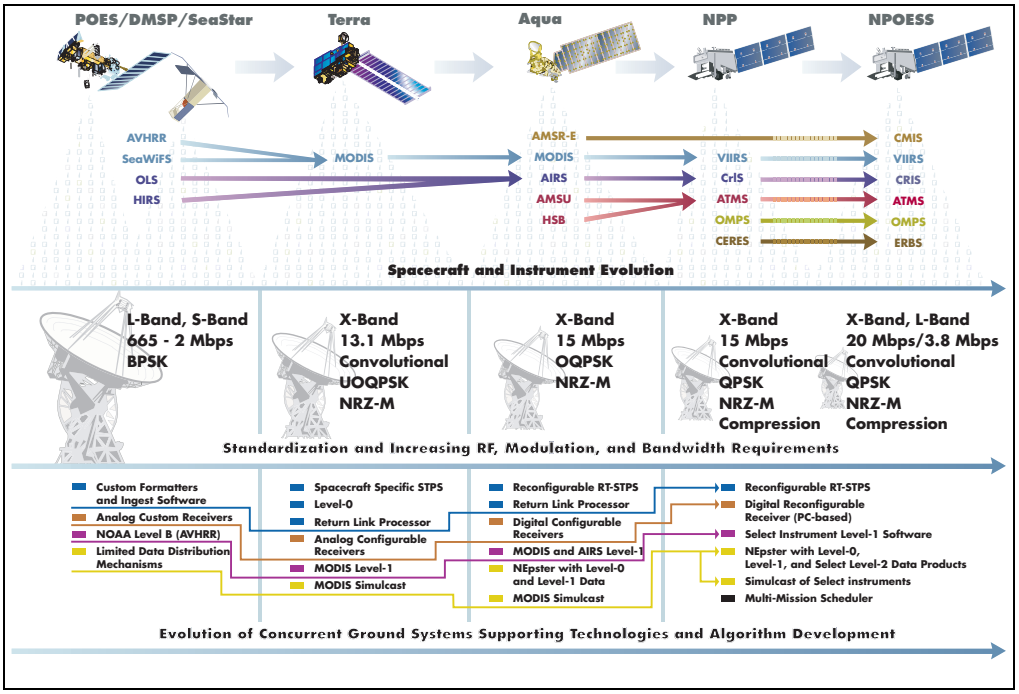

Figure 5
Direct Readout Technology Roadmap

Multi-Mission Scheduler

Within the NISGS example, the DR Multi-Mission Scheduler (MMS) is used as a baseline technology. The MMS is a configurable, client-server control software environment that automates ground system data acquisition, data processing, and data distribution. MMS also provides a documented Application Programming Interface (API) so that it can be easily integrated into any ground system environment.

MMS is a client-server software package where the client translator sits on each piece of equipment to be controlled. A client template is part of the MMS package. Scheduler is the MMS server. Scheduler provides a central control point for all the ground system software and hardware. The client template is customized, and then installed on each ground system component. Scheduler communicates with each client via a User Datagram Protocol (UDP) socket, and Scheduler uses data and events provided by each client to control data acquisition and processing. Based on Scheduler input, the client starts and stops software tasks local to the device on which it is installed.

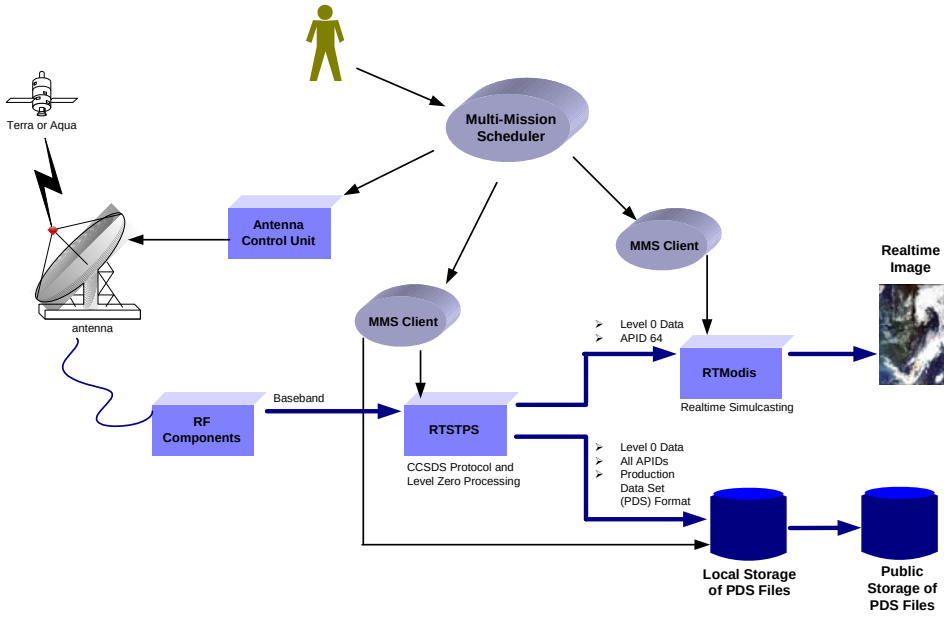

Figure 6
Multi-Mission Scheduler Functional Diagram

Scheduler provides a Graphical User Interface (GUI), which allows the operator to completely configure and control MMS activities. Data from multiple satellites can be acquired, processed, and distributed by defining parameters provided via the GUI.

Real-Time Software Telemetry Processing System (RTSTPS)

RTSTPS provides the baseline technology for NASA standard's Protocol Processing and Level-0 Data Production elements. RTSTPS is a software package that provides front-end processing in support of Earth remote sensing missions. This technology is a pure software implementation of a Consultative Committee for Space Data Systems **(**CCSDS) protocol and level-zero processing system data gateway. It performs protocol and Level Zero processing for legacy and upcoming missions. RTSTPS is platform independent. It is designed as a collection of independent processing nodes, so a user can create standard or custom pipelines by plugging together selected nodes using XML setup scripts. It is written entirely in Java and does not require any special hardware or native library support. Therefore, it runs on any system where Java is installed, yet it runs as fast as C and C++ equivalents. RTSTPS ingests satellite data, performs complete CCSDS protocol processing, and then provides level zero data distribution to end-users. RTSTPS has two mode of operation. In standalone batch mode, the source is a local file. In server mode, RTSTPS runs continuously and waits for raw data input to arrive at a TCP/IP or UDP socket. RTSTPS can send processed data units to a file or directly to a network destination, for which it uses TCP/IP sockets.

To support remote sensing satellites in near real-time, the primary job of RTSTPS is to synchronize CCSDS version 2 telemetry frames (CADUs) and to compose independent CCSDS packet streams, which are distinguished by spacecraft

(SCID), virtual channel (VCID), and application (ApID). Also to support the DB community, it includes a node that creates EOS-compliant PDS level zero files. However, RTSTPS can also receive and identify non-CCSDS frames, and it provides additional CCSDS services to those listed above. Other RTSTPS features include a rate-buffering client, which regulates transmissions for unpredictable or slower network connections, and a graphical viewer to ease configuration and observe processing status. RTSTPS processing has been successfully verified with live TERRA and AQUA MODIS data. The most CPU intensive RTSTPS processing, which includes Reed-Solomon decoding, has been benchmarked at rates in excess of 20 Mbps.

Since RTSTPS has a modular design, it has distinct components that work together to create the telemetry processor. In addition, several utility programs can provide additional functions. Figure 7 shows the layers that create the RTSTPS program. Utility programs exist outside this diagram.

RTSTPS Architectural Layers

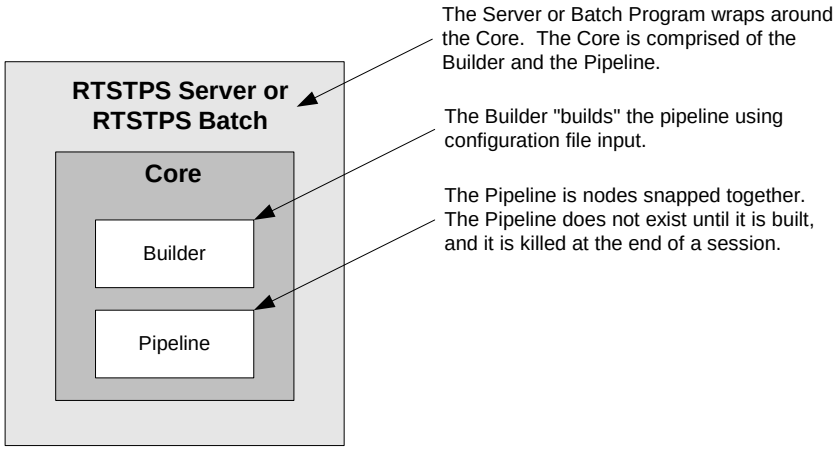

Figure 7

The RTSTPS components and the supporting utility programs can be summarized as follows:

1. RTSTPS Server or RTSTPS Batch Program: The processing component of the RTSTPS is either a standalone batch version, which ingests a single raw data file, or a server (TCP/IP or UDP versions), which runs continually and waits for raw data to arrive at an input socket. To satisfy EOS and NPP requirements, both the server and batch program include a node that creates EOS-compliant PDS files. The following diagram illustrates the Server, which could be replaced by the batch program, and its relationship to two other internal components: Builder and Pipeline. The server loads the XML Configuration file, which is defined below, via the “configuration and commands” input path. It receives telemetry data input through the “data” path.

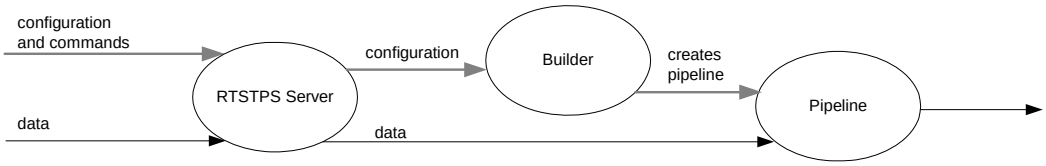

Figure 8
RT-STPS Data Flow

2. Nodes: RTSTPS Nodes define data processing functions. Each node is a Java object, which performs one primary function. Once connected, nodes pass along data units. Each node is independent of all other nodes, and nodes can be modified or new nodes created without having to understand system-wide dependencies or relationships. Nodes implement connection interfaces to create the connectors that allow nodes to be linked together. The interfaces empower a node to be either a sender or a receiver; Sender and Receiver are the two categories of interfaces.

3. Builder: Each time a user loads an XML Configuration file into the server or the batch program, the Builder interprets it to create a pipeline of linked nodes that process raw telemetry into products such as packets.

4. XML Configuration File: The XML Configuration File is a script that is input to the server/batch program. The script tells it which nodes to create and how to link them together. The user can create or modify the configuration file with any text editor.

5. Pipeline: The pipeline is a collection of independent nodes that are linked together to define the data processing path. The XML configuration script defines the pipeline. It tells the server (the Builder specifically) which nodes to create and how to link them together.

6. RTSTPS Utility Programs: The RTSTPS system has several other utilities besides the batch program and the UDP and TCP/IP servers.

    a. *Viewer*: The Viewer is a graphics tool that allows the user to load a configuration file into the server. It also shows status that the Viewer receives from the Server.

    b. *Editor*: The Editor is part of future development. It is a graphics editor that allows users to edit RSTPS configuration files.

    c. *Sender*: The Sender is a small graphics program that sends raw data files to the server. Mainly used for test purposes, it substitutes for the front-end antenna system.

    d. *RAT*: This program is a rate-buffering output client, which regulates transmissions to slow receivers. It can be attached on the output side of the server to spool the server's socket output.

Modes of Operation

The RTSTPS can run in two independent modes. It can run as a stand-alone program that processes a single raw data file (batch mode), or it can run as a server, in which case it runs continuously, listening to a socket and processing spacecraft passes as they occur.

Both modes of processing require an XML Configuration file for system setup. The specific data processing provided with either mode of operation is dependent on the pipeline (consisting of nodes) defined in the configuration file.

Batch Mode

Batch Mode is a run-once program. It reads all data in a source data file, and then produces output units as defined in the configuration file. It terminates when finished. Batch Mode is invoked with a command line that includes the program name "STPSBATCH," a configuration file, and the name of a file that contains the unsynchronized source data.

Once invoked, STPSBATCH creates a data pipeline as seen in figure 9. It then opens the data file and reads chunks of it, which it stuffs into the frame synchronizer node at the head of the pipeline. When it exhausts the file, it terminates.

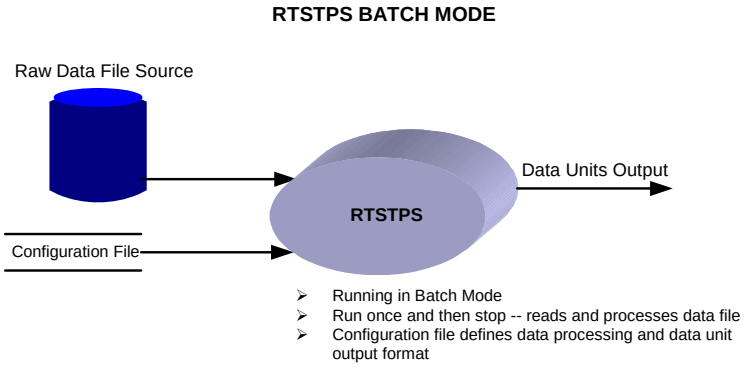

Figure 9

Server Mode

The second mode, Server Mode, runs continuously. It listens for spacecraft data on an input port, and it also listens for commands on a command port, which can be used to load a specific configuration. The server accepts data from a TCP/IP socket, but there is also a second version of the server that connects to a UDP socket.

The server runs continuously. It listens on a TCP/IP socket for connections and spacecraft data. It will accept only one connection at a time. (There is a second server version that uses UDP sockets for data input, but most use the TCP/IP version.)

The server is multi-threaded. One thread represents the pipeline. It receives raw data from the socket and processes it through the pipeline. Another thread listens for commands through a second socket. This is a special java interface called "Remote Method Invocation" (RMI) through which the RTSTPS viewer sends load and shutdown commands to the server. The server also uses this interface to return status information back to the viewer. Finally, there is a third thread called the proxy thread that also allows users to send commands to the server. Unlike the RMI interface, which requires a Java sender, the proxy thread is a pure TCP/IP socket connection, and the commands are text strings. I added this interface so

that programmers could send commands to the server from programs that were not written in Java.

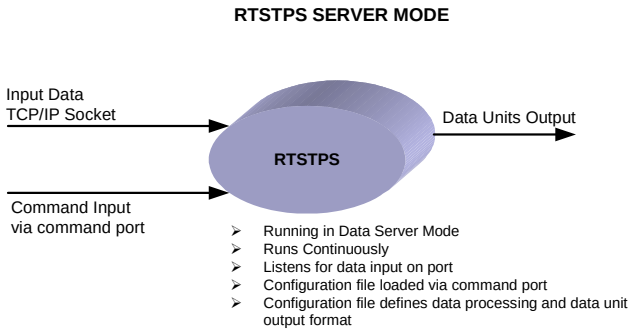

Figure 10

When operating in Server Mode, RTSTPS runs in sessions. A session usually corresponds to a spacecraft pass. Just prior to a pass, either a user or an automated scheduler sends a setup configuration to the RTSTPS server. The server will prepare itself to receive data, which involves creating a pipeline and waiting for data on a TCP/IP socket. RTSTPS begins to process data as soon at it arrives at the socket connection.

If the server receives data before it has been configured, it automatically configures itself. It either uses the last loaded configuration or a default configuration if one has never been loaded. If the user shuts down the server while it still has data to process, it will discard all subsequent buffers.

A shutdown ends a session. Either the user or automated scheduler sends the shutdown command to the server. The server shuts down automatically if the data sender breaks its connection. When the server shuts down, it closes all output files and all output sockets, and then it discards the current pipeline.

A node is a Java object, which performs primarily one function. For example, the Reed Solomon node performs Reed Solomon decoding and correction. When configuring the RTSTPS, the first and only required node is the frame synchronizer. It accepts blocks of unsynchronized telemetry. Figure 11 provides an implementation of nodes and the resultant data flow through the RTSTPS Server/Batch program.

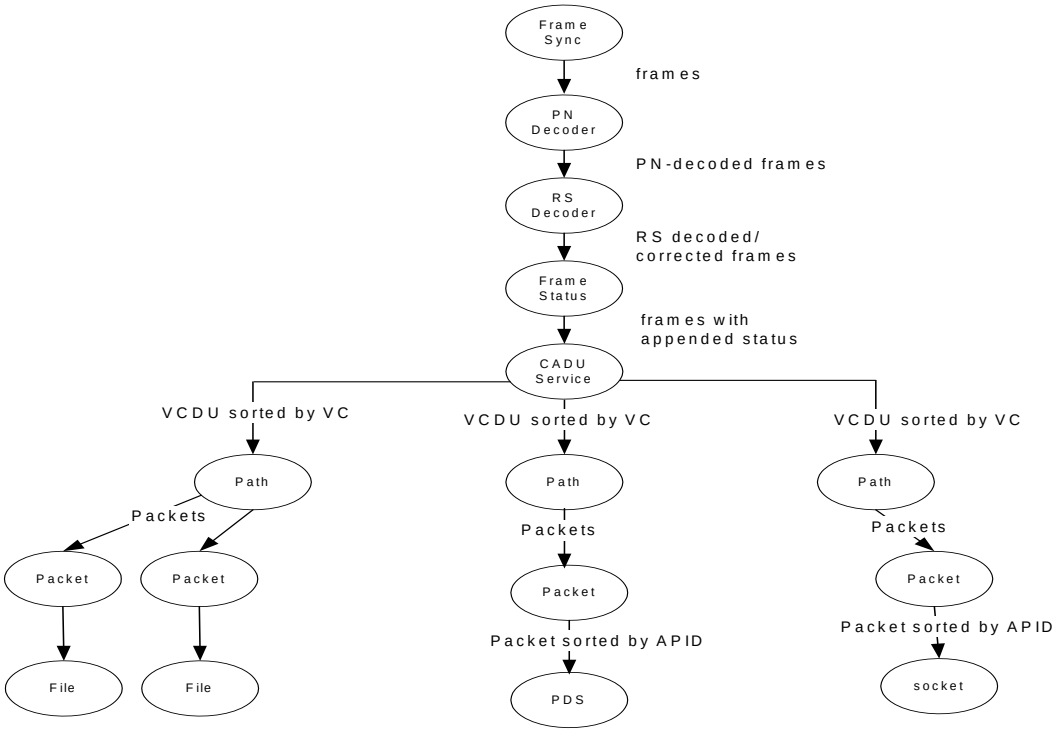

Figure 11

RT-STPS Server/Batch Node Implementation

The behavior of any one node is independent of all other nodes. A node has one input "port" and zero or more output ports. With multiple output ports, a node broadcasts. For data input, a node implements a frame, packet, or unit Receiver Interface. For data output, a node implements a Sender Interface. Since a node is isolated from all other nodes and can be defined to use any interface, it node can be plugged together with little regard to what is being plugged. Once connected, nodes pass along units. The term unit is used loosely to define all items such as frames, packets, VCDUs, and MPDUs.

Simulcast

Simulcast constitutes a component of a satellite DR system that enables real-time Earth science data utility and sharing. The system is designed to read the full multi-instrument CCSDS packet stream from a front-end satellite receiving station, create reduced data volume intermediate products, distribute the products simultaneously to multiple remotely networked workstations, and provide an appropriate interactive near-real-time display of instrument data.

The design and development of the simulcast system is based largely on interest and lessons derived from a similar software program known as RTModis (for Real-time Modis) depicted on Figure 6 in the MMS section. The RTModis system was developed in 1999 to support the direct readout sites for the then, soon

to be launched, TERRA satellite. The RTModis system successfully processed data beginning with the very first DB transmission from the TERRA satellite and is in operational use today providing real-time display of MODIS data for both the TERRA and AQUA satellites at both Mission Operation's Centers.

To support multiple remote displays and to minimize the network bandwidth requirement, the system is designed around a client/server model. A data processing component reads, decodes, decommutates, and performs any other process necessary to create intermediate data products. These products are designed to provide a reduced bandwidth representation of the instrument data as well as performance and state of health information for the spacecraft, instrument, and ground station.

The remote clients connect to the data processing server and receive the intermediate products as they are generated. The clients will render the intermediate products and provide an interactive interface to the end user. The remote display capabilities range from updating textual data about the spacecraft or other system components, to enhanced image products generated from instrument data.

The system is implemented in Java to provide cross-platform support and maintainability. The system will receive data from the front-end system. The data processing server functionality can be broken up across several processes or computers to spread out the necessary workload of generating intermediate products defined by the end-user; single band and image composites are the available public baseline.

The current EOS satellites, TERRA and AQUA, as well as the next generation programs, NPP and NPOESS, will all transmit data using standards accepted by the Consultative Committee for Space Data Systems (CCSDS). There are several hardware and software (including a Java implementation) systems that process raw serial data from an antenna system and produce CCSDS packet streams. These streams can be filtered to include a subset of the instrument data, including the possibility of different streams each for different instruments. The filtering/grouping field for the CCSDS packet is the Application Process Identifier (APID).

The primary input for the Simulcast system is a CCSDS packet stream. The overall goal of the Simulcast Server is to produce intermediate products for each instrument. The processing required for this task may overwhelm one physical computer, or other considerations may make it desirable to handle different instruments on different machines or different processes. Hence, the first Simulcast component, a CCSDS Router, filters and groups the input stream to multiple streams and routes them to different instances of product Extractors described in Figure 12. The same Router functionality will be present within the product Extractors to direct the packets from different instruments to the proper processing routines.

At one extreme there can be a Server for each APID producing an intermediate product for that instrument. Similar operational configurations lead to the need for some coordination regarding the delivery of the intermediate products to the

connecting Clients. To accomplish this, one additional component is added to the client/server system, the Client Coordinator. This component provides a single point of contact for remote clients and data processing servers. Information will be exchanged to direct the clients to the proper server for the desired product. This also provides the added benefit of allowing the data processing to be moved to backup machines or even alternate antenna sites without burdening the client with reconfiguration.

One additional feature is derived from the need to protect the operational environment from denial of service and other network attacks. The most efficient way to do this is to isolate the majority of resources behind tight network security and present only the needed resources to the wider Internet community. To accomplish this, a distinct element will reflect the intermediate products to the multitude of remote clients; in effect being a single client to the data processing servers and reflecting the intermediate products to the end user display clients. This design also addresses the concerns of providing the resources to handle many different legitimate display clients.

The end system results in a programming paradigm that is a grown trend: A set of general processing components forming a complete system via a coordinating configuration. This paradigm relates very well with the general principles of object oriented programming and distributed processing. In fact, using Java and TCP/IP client/server technology as a backbone of development, pulls a system design in this general direction; a happy synergy within the design cycle.

In its simplest form (Figure 12) some of the components only add processing overhead and do not serve a useful purpose.

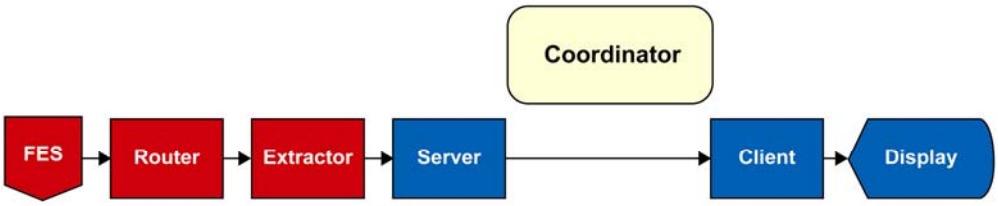

Figure 12
Simulcast block diagram (simple configuration)

However, the key to this system is designed extensibility. The operational system, as described in Figure 13, includes a backup front-end system (FES) that is transparent to the end user, split product generations processors (Extractors), an isolated server to reflect products generated behind a firewall, multiple remote clients, and a remote client which processes and reflects products via a Web Server.

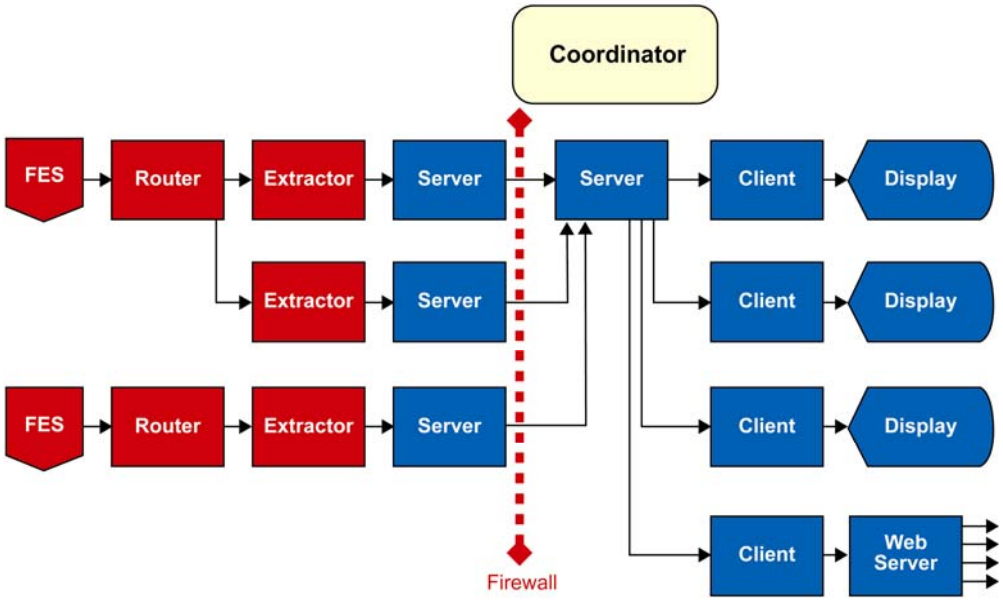

Figure 13
Simulcast block diagram (operational configuration)

NEpster

NEpster is last in a set of technologies needed to support a DR multi-mission environment as described in the Technology Roadmap section. Nepster is a software system that stores information about instrument products. Internet users, through their browsers, can search the system and download for specific products. NEpster also includes an assortment of agents who periodically probe remote data sites to collect product information. Nepster is not designed to store the actual products. Instead it stores the information that describes the products (the "metadata").

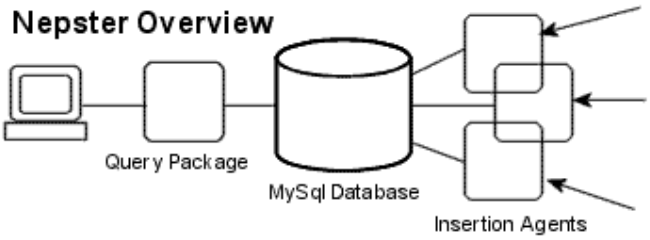

Figure 14

The core element of the Nepster system is a MySql database as described in Figure 14, although any commercial database would work. The database stores information about products, product types, and collection stations. It is not meant to be a complete instrument archive but is intended to hold only recent information, typically 1 to 30 days of current data which is dependent on how long the source-data-keeper retains the data.

The insertion agents are responsible for putting new product information into the database and removing obsolete product information. Each agent is customized to probe one specific site. Most agents use FTP to examine a site's products, but some process local directories, which their remote sites have loaded with current product information. All agents periodically probe their sites, usually every few hours.

The query package's function is to provide browser support for querying the database and downloading products. It contains a small server that plugs into a larger web server.

The Future of DB and DR

It has been said that the use of DR will slowly fade away with the advent of high bandwidth network access. This statement taken in a vacuum may logically seem accurate, but the reality is that there are many other factors contributing to the decision by the end user to invest resources into having their own DR system. Documented instances include emergency DR in network-isolated and/or remote areas where environmental information is required within minutes – currently not possible otherwise. Increase in network bandwidth and reduced access costs will nevertheless affect the way direct readout users utilize direct broadcast data and derived products.

From a programmatic, mission level point-of-view, it is the author's belief that DB will never go away for the simple fact that DB has proven to be an indispensable backup to the primary store and forward method of obtaining high resolution global data. Because of the innate and built –in robustness of the DB system design on-board the spacecraft, DB data has been used as a gap filler when satellite data relays fail and access to near-real-time global data is delayed do to data processing backlogs. This last scenario can be counter-argued by the fact that computational systems and network bandwidths will increase, but as described in Figure 15, data volumes are commensurate with computational power. In other words, we're barely keeping up. And when one considers expected data volumes and computational processing requirements for the next 20 years, the rate of increase will not be decreasing.

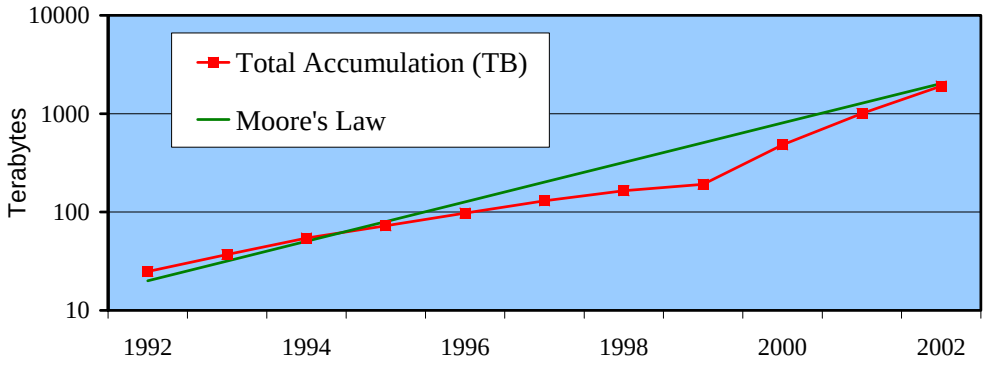

Figure 15
Archived Data in Earth Observing System Data & Information System
By permission from Chris Lynnes at the NASA Distributed Active Archive Center

There is also an economic factor when considering the longevity of DR. When analyzing the DR system growth trend, it is clear that not only is cost significantly decreasing but there is a direct correlation between a step decrease in cost and a step increase in DR systems as described in Figure 2.

On the other hand, there will be a significant change in how the majority of DB users will gain access to near-real-time products; after all, the point of DB is to get a data product and/or product to the end user as soon as possible over their area. With the on-set of the NPOESS, a comprehensive, all-be-it not new, method of acquiring, processing and distributing near-real-time data products will be established. This concept will also be mirrored by many additional profit and non-profit organizations which will take advantage of real-time raw data access to generate and distribute near-real-time value added products. This "new" concept, fueled by increase in network bandwidth and global access, is fundamentally a regression to the centralized, large data processing system's model with a new distribution twist, which includes broadcasting of data products, of much lower volume, to the end user via ubiquitous communication methods.

Acknowledgment

Research and development discussed in this chapter have been supported by the NPOESS Preparatory Project (NPP) and Dr. James Dodge at NASA former Code YS Earth Sciences Division. Special recognition is also given to Mr. Charles Vermillion and Mr. Fran Stetina, former NASA civil servants for their pioneering work in DB and DR systems.

References

P. Coronado, K. Brentzel, C. Harris, "NPP In-Situ Ground System: Bridging Technologies from EOS to NPOESS", International Telemetry Conference. Las Vegas, NV, November, 2003

P. Coronado, D. Jacob, K. Brentzel, "Multi-Mission Scheduling and Dispatching System", International Telemetry Conference. Las Vegas, NV, November, 2003

P. Coronado, A. Lunsford, K. Brentzel, C. Harris, "Simulcast: A Method for Real-Time Viewing of Direct Broadcast Instrument Data Over TCP/IP", International Telemetry Conference. Las Vegas, NV, November, 2003

P. Coronado, P. Ghuman, K. Brentzel, "A High Data Rate Satellite Remote Sensing Ingest System", International Telemetry Conference October, 1998.

P. Coronado, W. Campbell, “Applying Information Technologies to Facilitate Information Access and Regional Development”, Tech 2007 NASA Annual Conference, March, 1997.

Wallach, Jeff, National Oceanographic and Atmospheric Atministration, “User’s Guide for Building and Operating Environmental Satellite Receiving Stations, DOC Publication, July 1997