Development of a Satellite Beacon Receiving Station

Carlos Guillermo Niederstrasser
Space Systems Development Laboratory
Stanford University
2035 California St., Apt. 32
Mountain View, CA 94040-1967
(650) 694-4667
carlosn@alumni.princeton.edu

Abstract. As part of its space operations research program, Stanford University's Space Systems Development Laboratory (SSDL) is implementing an automated state of health assessment and notification system for spacecraft. On board the spacecraft, this system consists of software that filters telemetry to derive a health assessment and a periodic beacon that broadcasts this assessment to the Earth. Throughout the world, a network of low cost receiving stations receive the beacon signal and relay it to a central mission control center via the Internet. This paper addresses the design and development of a beacon receiving station. Each station is designed to be approximately an order of magnitude lower in price than a conventional two-way ground station. Emphasis is placed on making sure the station is highly autonomous, requiring little or no assistance from the host site. The stations are made up of only three separate components – an antenna, a receiver, and a personal computer. Existing hardware at the host site, such as available computers and network connections are used to further minimize costs.

Introduction

Current trends in the space industry are forcing designers to completely reassess the way spacecraft are operated. By some estimates, sixty percent of a spacecraft's costs are in its operation. To bring mission costs down it is therefore essential to bring operation costs down. One way to accomplish this is by increasing satellite autonomy while reducing reliance on the traditional ground segment of a mission.

Stanford's SSDL is currently developing a satellite beacon system that would reduce the need for large ground stations. In this scenario, the satellite would have considerable autonomy in monitoring its own status and health. When an anomaly, as defined by programmable values or parameters, is detected the satellite takes corrective action. For instance, the satellite might proceed to turn off all non-critical systems and put itself into safe-mode. The status of the satellite is then transmitted in a short beacon signal that can be picked up by any of the beacon network stations. Support from a full two-way station would only be needed if the satellite's state required operator interaction.

The beacon system is composed of three separate segments. The space segment consists of satellites that can transmit a known beacon signal to be received on the ground. The receiving segment is a network of receiving stations that will process the beacon signals from all member satellites. Finally the control segment will take all signals from the receiving stations, and take any action that might be required.

This paper focuses primarily on the receiving segment. In order for the beacon system to be competitive with the current tracking and control stations, beacon stations must be cheaper and provide larger worldwide coverage. This creates several cost, complexity, and operability constrains. Each beacon station must be approximately an order of magnitude cheaper than a full-scale two-way control station. It must be small, robust, and simple in order to be deployed anywhere in the world. Maintenance by the station's host must be kept to a minimum. Finally, it must be able to link with existing communications networks in order to connect to the home control base.

Space Systems Development Laboratory[1]

SSDL was chartered in 1994 to provide world class education and research in all aspects of spacecraft design, technology, and operation. To achieve this goal, SSDL members enroll in a comprehensive academic program composed of coursework, project experience and research investigations. SSDL is actively involved in research in spacecraft operations and automation.

The Satellite Quick Research Testbed (SQUIRT) Microsatellite Program

The SSDL SQUIRT program[2] is a project through which students design and fabricate a real spacecraft capable of servicing low mass, low power, state-of-the-art research payloads. By limiting the design scope of these satellites, the project is simple and short enough so that students can see a full project life cycle and are able to technically understand the entire system. Typical design guidelines for these projects include using a highly modular bus weighing approximately 11 kilograms, a hexagonal form that is roughly 23 centimeters high by 41 centimeters in diameter, amateur radio communications frequencies, and commercial off-the-shelf components. Missions are limited to about one year of on-orbit operation. Since little money is available for operations, a highly automated mission control architecture is being developed.

The Stanford Audiophonic Photographic Infrared Experiment (SAPPHIRE) Microsatellite

Shown in Figure 1, SAPPHIRE is the first SQUIRT spacecraft.[3] Its primary mission is to characterize the on-orbit performance of a new generation of infrared horizon detectors, in addition to flying two student instruments, a digital camera and a voice synthesizer. Student research interests are also driving experiments in nontraditional sensing and automated operations, including the beacon health monitoring system.

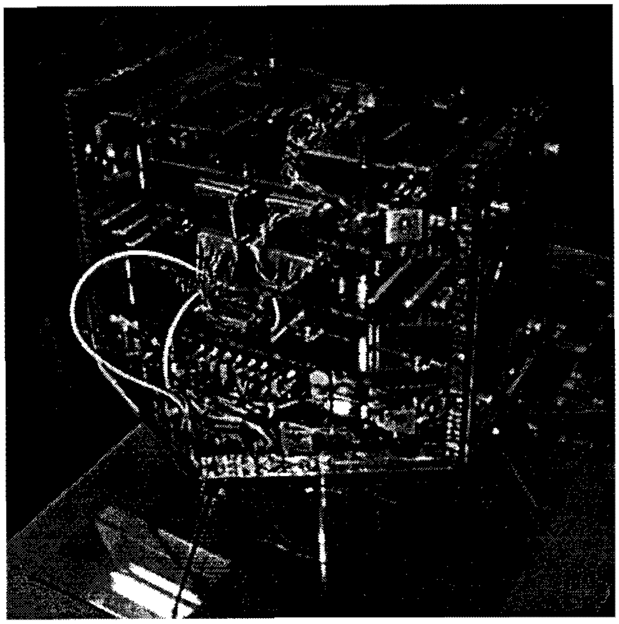

Figure 1: SAPPHIRE

The Automated Space System Experimental Testbed (ASSET) System

The ASSET system[4] is a global space operations network under development within SSDL. The first goal of this system is to enable low-cost and highly accessible mission operations for SQUIRT microsatellites as well as other university and amateur spacecraft. The second goal of this system is to serve as a comprehensive, low inertia, flexible, real-world validation testbed for new automated operations technologies. The basic components of ASSET include the user interface, a control center, ground stations, communications links, and the target spacecraft. During the current developmental phase, a highly centralized operations strategy is being pursued with nearly all mission management decision-making executed in the control center. These tasks include experimental specification, resource allocation throughout the ground and space segment, anomaly management, contact planning, data formatting and distribution, and executive control.

SAPPHIRE and all future SQUIRT satellites will be operated through ASSET. In addition, controllers for a number of other university and amateur satellites have expressed in becoming part of the system. As for ground stations, the Ogden and Stanford ground stations are the first two facilities to be included. Several other stations throughout North America, Asia, and Europe have been identified for future integration.

Beacon-based Health Management[1]

A beacon-based health management concept was first presented in a U.S. Air Force study, Lifeline.[5] It is currently a flight experiment aboard NASA's Deep Space 1 mission[6] and is one of the key technologies for future NASA deep space missions.[7] This concept is being prototyped as a part of the SAPPHIRE mission[8]; its main features are summarized, below. The signal flow for the SAPPHIRE implementation is presented in Figure 2.

SAPPHIRE Health Monitoring

SAPPHIRE will monitor its own sensors, comparing measured values with expected values in a state-dependent limit table. Certain measurands will be validated by

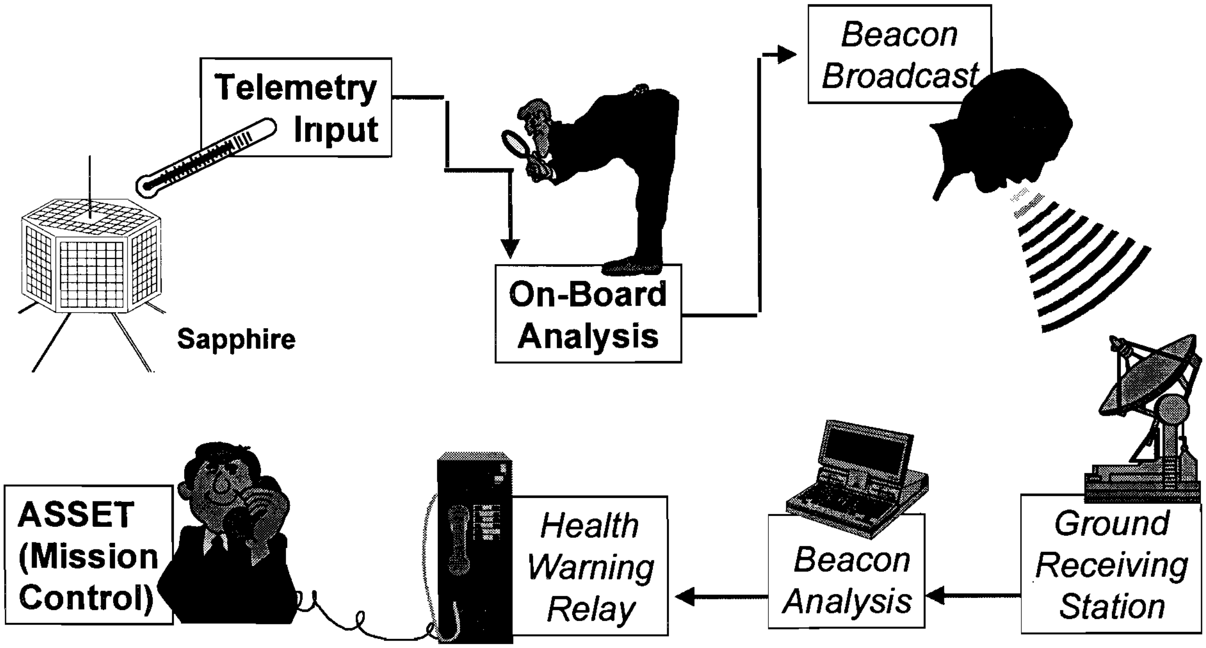

Figure 2: SAPPHIRE Health Beacon Signal Flow

aggregate analysis. For example, the vehicle's configuration prevents all solar panels from seeing the Sun at once; if solar panel measurements indicate that all panels are generating current, then there is good reason to believe that the current sensors are malfunctioning. These modest steps provide SAPPHIRE with an anomaly detection system far more mature than most spacecraft.

Depending on the seriousness of the limit violation, the spacecraft state is assessed to be one of four values. For example, when measurands are within limits, the spacecraft is judged to be *Normal*. Out-of-limits with moderate impact, such as an overheating camera, is considered an *Alert*. Out-of-limits that can rapidly jeopardize the mission elevates the health status to *Critical*. Finally, *Emergency* condition is defined to be an unexplained computer reset. Note that the rules by which measurands trigger the modes, and the limits for each, are defined by the operations team. This ensures that beacon modes are a mapping from spacecraft state to operator action.

Health Beacon Transmission

The beacon is a pulse modulation of the main transmitter carrier, with different pulse widths defined for a one bit and a zero bit. The total transmission time of the beacon message is less than one second. The message is broadcast whenever beacon operations are active, nominally at one-minute intervals. Therefore, spacecraft health is continuously monitored and the health indication is available anytime the spacecraft is within range of a receiving station. The beacon message format is further described in the Beacon Message Specification section.

Receiving Station

SSDL has developed the prototype BACON (Beacon Automated Contingency in Orbit Notification) receiving station, which is the primary focus of this paper. Based on a schedule provided by ASSET, it listens for satellite transmissions. If a beacon signal is received, the pulse modulation is converted into a numeric message and this information is time-tagged and sent via electronic mail to the ASSET mission control.

Mission Control Center

Once mission control receives a beacon monitoring update from a remote station, it logs this information and then takes appropriate action. Depending on the health assessment, there are varied responses, from storing the update in the system database to paging the operator on call and rescheduling the network to contact and recover a failed satellite.

BACON Hardware

Hardware selection for the BACON station was driven by three major factors: the station had to be low cost, relatively autonomous, and use easily available components. These requirements led to a station based around a Personal Computer, using commercial radio scanner technology.

Figure 3 shows a block diagram including the main components of the station. The station contains only three separate pieces of hardware: an antenna, a receiver, and a PC. In addition the PC is equipped with a sound card and is connected to the Internet through the host site's local network.

Receiver

Receiver selection proved to be the most challenging aspect of the station's hardware. In addition to being low cost and relatively rugged, the receiver had to be sensitive enough to pick up the weak beacon signals,

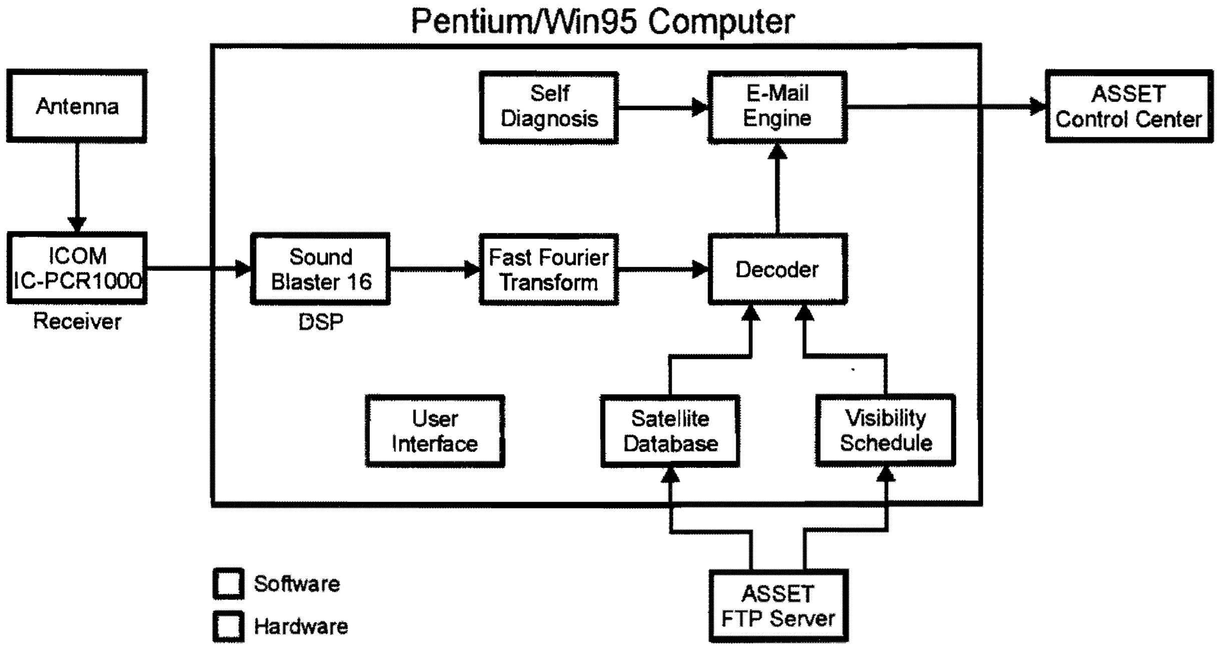

Figure 3: BACON System Diagram

and it had to be computer controllable to allow the station software to tune it.

Although a wide selection of receivers in the 70-cm band is available for HAM use, most are part of expensive transceiver units. Cost becomes even higher if computer control is desired, since this is often viewed as a luxury feature. This lead to a design decision to avoid the HAM units and instead focus on the market for scanners.

Scanners have the advantage of handling a wide range of the broadcast spectrum, and do not have the additional cost added by the transmitter equipment. Of these, some are now emerging in the market that are fully computer-controlled. The scanner selected for the station is the ICOM America IC-PCR1000 receiver.

The PCR1000 has several features that made it particularly attractive. It is one of the cheapest computer-controlled radios available, and has a relatively high sensitivity in the VHF and UHF regions where most amateur radio satellites operate. It has a bandwidth of 6 kHz, which permits simplified tuning to account for the Doppler shift in the satellite's signal. Its small package (the size of a small book) takes very little space, and because it can only be controlled by a computer, there is only one external switch. Finally, the unit uses a standard serial cable to communicate with the host computer, as opposed to some other options that required an interface card to be installed in the computer.[9]

The antenna for the receiver is an eggbeater antenna, popular with amateur radio satellite operators. Eggbeater antennas are ideally suited for low-bandwidth satellite communications since they are essentially omnidirectional, with a gain pattern that is stronger toward the horizon and weaker toward the zenith.

Computer

Early in the design it was decided that an Intel-based Microsoft Windows platform offered the best environment for the station. The drop in prices of personal computers in the last few years allowed the design to include a standard PC without need for any hardware modifications.

In fact, the requirements for the computer are relatively low by today's standards. If the computer is to be dedicated solely to the station, then a Pentium-class machine running at 90 MHz with a small hard drive and 16 megabytes RAM is enough. In addition, a standard Ethernet card is required to connect to the host site's network. Finally, a Sound Blaster-compatible sound card, described below, provides the interface to the receiver. A monitor is optional, and only required when the station needs human operation.

During development, and additional advantage of using a Windows-based platform became apparent. A BACON station can capitalize on the host site's existing infrastructure by using a computer already present. This greatly reduces the cost of the station, since only the antenna, receiver, and possibly the sound card must be acquired. Of course, hardware requirements in this situation will be higher since the station must now share processing power with other applications.

Digital Signal Processor

Originally the BACON station designed called for an external Digital Signal Processor (DSP) to digitize and process the audio output from the receiver. While the DSP offered great flexibility, it substantially increased the cost, and was complex to program by someone without previous experience.

Upon further study, it was noted that most PCs today come with a built-in and inexpensive DSP in the form of a sound card. The BACON design uses a Sound Blaster 16-bit sound card, the industry standard for PCs. Such cards have sampling frequencies of up to 45 kHz and are easily programmable from within the Windows environment.

Although digital filtering functions are available directly on the Sound Blaster's DSP chip, these are not widely documented. Because of this, the sound card only provides the digitizing of the signal, while the computer's software carries out all the necessary filtering, as explained in the Signal Sampling and Decoding section.

Costs might be further reduced by using anyone of numerous "Sound Blaster compatible" cards. However, software tests to insure true compatibility with the BACON system remain to be done. ICOM has also just released a DSP module for the PCR1000, which might eliminate the need for a sound card.

BACON Software

The software for BACON is written entirely in C++ using the Microsoft Foundation Classes (MFC) on a Windows 95 platform. This platform was chosen for several reasons. Windows 95 is now widespread and a large base of unofficial programming support exists. C++ and MFC provided the least learning curve for the author, while allowing easy access to the sound card driver routines. Although not yet tested, BACON is expected to be compatible with Windows 98 and Windows NT.

Figure 3 shows the general block diagram and interfaces for both hardware and software. All software blocks are highlighted in gray. The station obtains information about the satellites it services directly from the ASSET FTP server. This is provided in the form of a satellite database and a visibility schedule, both of which are described below.

As the signal is sampled by the sound card it is processed by a Fast Fourier Transform algorithm and is then decoded according to the rules established in the beacon specification for the particular satellite that is visible. Once the signal is decoded, the resulting numeric code is passed to the e-mail engine to be forwarded to the ASSET Mission Control Center.

Working in parallel with the decoding algorithms, there is also a self-diagnosis unit. This unit constantly monitors the state of the station and sends email to the control center if it detects any faults. Because of this, the station is largely autonomous only needing human interaction for regular maintenance or if it detects a fault.

Satellite Database

The satellite database is the station's sole source of information regarding the satellites that it services. The database is updated on a weekly basis (more often by using operator interaction) from the main ASSET FTP server. By having a centralized depository of information, any BACON station around the world can have access to the latest data.

The satellite database informs the station's software about certain key attributes of each satellite. Each satellite has a unique three letter call sign assigned to it – SAPPHIRE is SAP, for example. Along with its call sign, the name of the satellite is also included for the benefit of any human operators. Beyond basic identification of the satellite, two key parameters are necessary to enable the station to operate properly: the base frequency of the satellite's beacon, and the format of the beacon. This information permits the station to monitor the satellite and translate its beacon messages into the appropriate numeric codes required by the ASSET mission control center. Further details on the implementation of the actual message are given in the Beacon Message Specification section.

Visibility Schedule

The visibility schedule is the second document that the BACON station obtains from the ASSET FTP server. Like the satellite database, the visibility schedule tells the station what it needs to know in order to operate.

The BACON network can service many satellites, each with its own frequency. Because of this each station must know when to tune its receiver to a particular frequency in order to pick up a satellite during its pass. Each entry in the schedule contains the satellite's call sign followed by the start and finish times of the pass (in GMT), and the maximum Doppler shift during the pass.

Because PC clocks tend to drift considerably (as much as a minute a day, in some systems) the station's software must constantly monitor its internal clock and compare it against a known source. Fortunately this is a relatively task by using the Internet's Network Time Protocol (NTP). NTP allows a computer to connect through the Internet to a well-known server, and update its internal clock. Such servers are found worldwide and are run by organizations such as the Naval Observatory, NASA, and major universities. They can easily provide and accuracy of one second, which is sufficient for the needs of the station.[10]

Beacon Message Specification

Although BACON is designed to handle relatively simple beacon messages (most implementations call for messages of two bits), some flexibility has been provided in the specification of the message.

Beacon messages are subdivided into discrete segments of 232 ms (see the Signal Sampling and Decoding section), as shown in Figure

4(a). Each message bit consists of two or more of these segments. SAPPHIRE's messages, for instance, have two segments per bit, shown by the darker lines in the figure. Each segment can either be High (RF signal present) or Low (no RF signal) as shown in parts (b) and (c) of Figure 4.

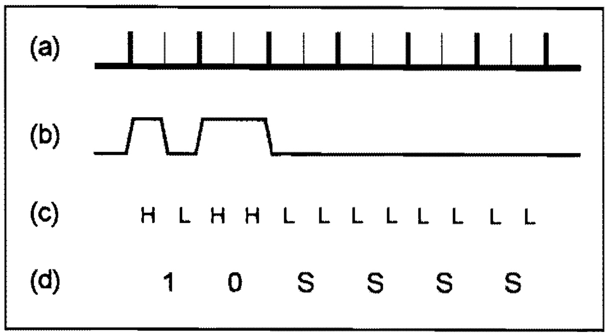

Figure 4: SAPPHIRE's Beacon Signal

Each particular combination of High's and Low's is then translated by the bit format particular to that satellite as shown in part (d). Resulting bits can be 1, 0, or Stop, where stop is always defined as no signal.

Each satellite thus has a translation table unique to its implementation. Table 1, below shows an example of a beacon translation with bits consisting of three segments. Table 2 shows SAPPHIRE's beacon translation with bits consisting of two segments.

Table 1: Sample Beacon Translation

Segment RF signal	LLL	HHL	HHH	All Others
Message bits	Stop	1	0	N/A

Table 2: SAPPHIRE's Beacon Translation

Segment RF signal	LL	LH	HL	HH
Message bits	Stop	N/A	1	0

In addition to the translation from RF segments to message bits, the beacon specification for each satellite also includes how many data bits each message has. Finally, there must also be a number of stop bits – bits in which no RF signal is present. These are required to differentiate beacon messages from other potential RF sources, including the satellite's own downlink.*

Note that the satellite must not take an active role to generate stop bits, it must simply remain silent for the given time period. Figure 4(d) shows that SAPPHIRE requires 2 data bits and 4 stop bits; this combination was empirically determined to produce the best filtering of extraneous signals.

Signal Sampling and Decoding

The sound card of the station is responsible for sampling the audio signal produced by the receiver. Standard PC sound cards can sample at a variety of frequencies. For the BACON station, the lowest sampling frequency of 11.025 kHz is enough to provide the resolution needed, since the audio frequencies of interest are in the 1000 to 4000 Hz range.

Samples are taken in blocks of 2560, creating segments of 232 ms, as mentioned in the Beacon Message Specification section. Each segment is then transformed to the frequency domain by use of a Fast Fourier Transform (FFT) algorithm. Once the signal is in the frequency domain, a search is conducted for the strongest frequency recorded. If its level meets a predetermined threshold above the noise that segment is determined to have had an RF signal present and is labeled as a High.

---

* Unlike larger satellites, SQUIRT-class satellites use their main downlink frequency for telemetry, payload data, and beacon messages.

Provided there is a satellite visible, the station continues to collect samples up to the number listed in the satellite's beacon specification. The signal must now be decoded according to the specification and translated into a numeric message that can be mailed back to the ASSET mission control center. Signal decoding might not always be successful, however. For instance, some bits might get lost due to a weak signal or interference from other sources, or the signal might not be an actual beacon signal. In this case, appropriate note is made of the failure, which is also relayed to the mission control center.

One additional condition might generate an email. If a satellite is not heard at all during a scheduled pass, the station notifies the MMC. This situation might indicate a major problem with the spacecraft or with the station itself.

Self-Diagnosis

The station has some basic ability to monitor itself and indicate when it detects a fault. Any faults detected are forwarded by e-mail (provided it is functional) to the mission control center. In this sense the station has its own "beacon" and the MCC treats it like it would any other satellite. The MCC will notify the operator-on-call for the station so that the problem is rectified.

The simplest way in which the station detects a problem is by keeping track of the number of times it has not heard from any satellite. If this number is high, and especially if it covers more than one satellite, it could be a good indicator that something is wrong with the station.

The MMC performs a similar test. If no e-mail is received from the station during a predefined period of time, the MMC assumes that the station has gone off line.

Future implementations of the station will also have more robust testing systems, though some might have a monetary cost due to new hardware. The station could have its own transmitter, effectively providing a reference beacon that would test the entire system. Simple analysis could also be conducted on the signal coming in from the receiver. Changes in this signal could be tracked and could indicate problems with the hardware.

Finally, the station also has a graphical user interface. Although this is technically not part of the self-diagnosis module, it does provided a way for the operator to quickly gauge the performance and status of the station. All important performance parameters are displayed in the GUI, including signal strength, spectrum distribution, RF frequency, and translated beacon messages.

Preliminary Experiments and Results

A preliminary experiment of the beacon system was conducted in May of 1998. The experiment brought together for the first time SAPPHIRE, ASSET, the main SSDL ground station, and the BACON station. The test served as a verification of the beacon concept and provided some early results on the efficiency of carrying out beacon operations vs. normal operations.

During the test the engineering prototype of SAPPHIRE was configured to broadcast a beacon message every minute indicating its state of health. A visibility schedule for the satellite was calculated to simulate SAPPHIRE in a Mir-like orbit, and access to the vehicle was restricted to the windows when the satellite was "over" Stanford.

For the duration of the three-day test two operator teams were assembled. The first team represented the "conventional" operations procedure, with no knowledge of the beacon generated by the satellite. Their support requirement was established at two contacts a day, at least six hours apart. These contacts took place using Stanford's main ground station and occurred when the satellite was visible.

The second set of operators was the beacon team. This group did not have any direct contact with the satellite. Instead, whenever the satellite was visible, it broadcast a beacon signal that was then forwarded via email by the BACON station and the ASSET Mission Control Center. For this test the beacon operators could not react to any fault, they simply recorded the time at which the fault was detected.

For the duration of the test a third party, or Gremlin, created a fault on the satellite. Three fault types were available to the Gremlin, each of which generated a different beacon message. A hard reset of the spacecraft generated an emergency signal, a lowering of the power supply voltage (simulating the onboard batteries) generated a critical signal, and heating of the main payload generated an alert signal. Neither the conventional nor the beacon operators were aware of the Gremlin's schedule.

Although this was a preliminary test, not intended to provide any hard data on reduction of cost or increased data timeliness, some interesting results were obtained. The beacon operator was quoted as saying, "I was able to do this with little effort while completely following a distinct work schedule – not tied down in either time or space." Throughout the test there was no need for the beacon operator to take any action other than regular checks of e-mail.

In terms of timeliness, the two spacecraft reset events introduced by the Gremlin provided the greatest insight. Reset Event One happened at 35:55 (Mission elapsed time). The beacon operator was notified during the 36:10 pass, and noted the event at 37:00, and could have reacted by the next pass at 37:46. (Provided ground stations could be rescheduled on time, of course). The conventional operator did not notice the anomaly until the next scheduled contact during the 39:20 pass, more than three hours after initial notification.

Reset Event Two occurred at 41:05. Because of the orbital dynamics the next pass was at 54:18, at which time the beacon operator was notified. The conventional operator noticed the anomaly during the 57:37 pass. It took the beacon operator about 13 hours to note the fault, while the conventional operator took about 16 hours. This result is particularly interesting when one considers that the event occurred just two hours after the conventional operator had reestablished nominal operations of the spacecraft following the previous reset. Had there been a second beacon station the 41:05 fault might have been detected much sooner, alerting the operators that the resets might be part of a pattern.

Future Research and Conclusions

So far preliminary tests with the ASSET and SAPPHIRE system seem to validate the beacon concept. The use of a low cost, low power receiving station for state-of-health monitoring appears to be a viable option for decreasing mission operations cost and improving health data timeliness.

At this point more experimentation is needed to provide better understanding of the cost and timeliness factors involved. To this end, a second operational test of the ASSET-SAPPHIRE system is planned during July and August of 1998. Operations of SAPPHIRE will again be conducted by two separate teams, but will last at least four weeks. In addition four different beacon stations distributed worldwide will be simulated by the one at Stanford. This experiment will more closely simulate real operation of the satellite in orbit and will provide better results based on operator time, station costs, and mean time between failures.

Future improvements for the BACON station include enhanced ability to accommodate Doppler shift, additional autonomy, and a priority scheme to service multiple satellites. In addition, OPAL, SSDL's second satellite, will also be outfitted with a beacon health system.

Additional research is ongoing in SSDL focusing on the applicability of beacon systems for larger satellites, including LEO, GEO, and deeps space missions. Research is also being conducted in related fields such as health data summary and onboard fault detection.

Acknowledgements

The author wishes to thank Professor Robert Twiggs, Chris Kitts, Mike Swartwout, and Brian Engberg for their guidance as well as the rest of the ASSET and SAPPHIRE teams for their dedication and efforts. The Jet Propulsion Laboratory and NASA Ames Research Center are acknowledged for their support and assistance with this research project. This work has been performed in partial satisfaction of graduate studies at Stanford University.

References

1) Swartwout, Michael A, Carlos G. Niederstrasser, Christopher A. Kitts, Rajesh K. Batra, and Kenneth P. Koller. "Experiments in automated Health Assessment and Notification for the SAPPHIRE Microsatellite." *Proceedings of SpaceOps '98: The Fifth International Symposium on Space Mission Operations and Ground Data Systems*, Tokyo, Japan, June 1-5, 1998.

2) Kitts, Christopher A., and Robert J. Twiggs, "The Satellite Quick Research Testbed (SQUIRT) Program," *Proceedings of the Eighth Annual AIAA/USU Conference on Small Satellites*, Logan, Utah, September 16-22, 1994.

3) Lu, Richard A., Tanya A. Olsen, and Michael A. Swartwout, "Building 'Smaller, Cheaper, Faster' Satellites Within the Constraints of an Academic Environment," *Proceedings of the Ninth Annual AIAA/USU Conference on Small Satellites*, Logan, Utah, September 19-22, 1995.

4) Kitts, Christopher A., "A Global Spacecraft Control Network for Spacecraft Autonomy Research." *Proceedings of SpaceOps '96: The Fourth International Symposium on Space Mission Operations and Ground Data Systems*, Munich, Germany, September 16-20, 1996.

5) Schultz, Michael, "Lifeline: A Concept for Automated Satellite Supervision," *Proceedings of the Software Technology for Satellite Autonomy Workshop*, Albuquerque, NM, June 22-25, 1993.

6) Sherwood, Rob, Jay Wyatt, Alan Schlutsmeyer, and Mike Foster, "Flight Software Implementation of the Beacon Monitor Experiment on the NASA New Millennium Deep Space 1 (DS-1) Mission," *Second International Symposium on Reducing the Cost of Spacecraft Ground Systems and Operations*, Oxfordshire, United Kingdom, July 1997.

7) Wyatt, E. Jay, and John B. Carraway, "Beacon Monitoring Approach to Spacecraft Operations", *Reducing the Cost of Spacecraft Ground Systems and Operations*, Oxfordshire, United Kingdom, September 27-29, 1995

8) Swartwout, Michael A., and Christopher A. Kitts, "A Beacon Monitoring System for Automated Fault Management Operations", *Proceedings of the Tenth Annual AIAA/USU Conference on Small Satellites*, Logan, Utah, September 16-19, 1996.

9) *IC-PCR1000 Specifications*. ICOM America, Inc. January 19, 1998. <http://www.icomamerica.com/receivers/IC-PCR1000specs.html>

10) *Time Synchronization Server*. EE/CIS Computing Systems at the University of Delaware. May 22, 1998. <http://www.eecis.udel.edu/~ntp/>