GENSO, SPA, SDR, and GNU Radio: The Pathway Ahead for Space Dial Tone

Ryan L. Buffington*
Schafer Corporation, Albuquerque, NM, 87106

Craig J. Kief†
COSMIAC, Albuquerque, NM 87106

R. Scott Erwin,‡ Joshua F. Androlewicz,§ and James Lyke¶
Air Force Research Laboratory, Kirtland AFB, NM, 87117

This paper introduces "space dial tone" as an alternative model to present ground stations which involve complex, often proprietary, stovepipe equipment and labor-intensive operations. This is compared with the cellular telephony model, which was engineered to produce a user experience that mimicked the simplicity of the earlier land line network and through great effort seamlessly integrated with it, producing a consistency and transparency that greatly enhanced its utility and promoted its popularity. Space dial tone will permit users to transparently network with one or many spacecraft with an engineered user experience that is simple, consistent, and does not require an intimate understanding of the actual communications path through the ground systems. This paper presents the vision of the space dial tone model and how it might be formed through extensions of previously existing ground networks, software defined radios, and interchangeable (plug-and-play) interfaces in hardware, software, and protocols. This model greatly increases the flexibility of existing and new ground stations and promotes the interchangeability and exploitation of advanced radio equipment in future satellites.

I. Introduction

In a traditional Low Earth Orbit (LEO) satellite mission, one of the most important components is the transmission of commands from the ground to the satellite and the transmission of mission data from the satellite to the ground. The average ground station sees a satellite in LEO with an orbital period of 100 minutes less than half a dozen times per day. Most of these passes last no longer than 15 minutes.[1] In the event of inclement weather or equipment failure, a ground station could easily go for 24 hours without access to the satellite. Experimental data stored by the satellite must be retrieved before the satellite runs out of memory, otherwise valuable data will be lost.

As one drives between cities in the United States, it is often possible to talk the entire distance on a cellular telephone. Telephone users have come to expect the network to pass the call from tower to tower transparently as they move, yet satellite owners often settle for much less - one or two home ground stations with no opportunity to communicate with the satellite unless it is directly overhead and no automatic handoffs between ground stations. Cellular communications companies sign roaming agreements to increase the size and coverage of their networks without building additional towers. These agreements allow users to communicate using towers controlled by other companies when they are outside of the range of their home network. Similarly, the coverage of a satellite ground segment can be expanded by allowing satellites

*Engineer/Scientist, Space and Directed Energy Division, 2309 Renard Place SE, Suite 300
†Deputy Director, 2350 Alamo Avenue SE, Suite 100
‡Principal Research Aerospace Engineer, Space Vehicles Directorate, 3550 Aberdeen Avenue SE, Associate Fellow, AIAA
§Electrical Engineering Technician, Space Vehicles Directorate, 3550 Aberdeen Avenue SE
¶Principal Electronics Engineer, Space Vehicles Directorate, 3550 Aberdeen Avenue SE, Associate Fellow, AIAA

to use ground stations owned by other organizations when they are outside of the coverage of their home ground station. If agreements are made with a sufficient number of ground stations, coverage can be made ubiquitous. This will allow the user to access his satellite regardless of where the satellite is in its orbit or his own geographical location. Just as the selection of a cellular tower and the routing of a call through the telephone network is automated, the selection of an appropriate ground station and the connection from the user to the remote ground station can be handled automatically by the network, as shown in Figure 1. We refer to this new paradigm as space dial tone.

Four new capabilities are in development that will help in achieving this vision: the Global Educational Network for Satellite Operation (GENSO), the Space Plug-and-Play Architecture (SPA), Software-Defined Radio (SDR), and the GNU Radio software architecture. Each of these capabilities will be explored in this paper, and a framework for combining them into an initial incarnation of the space dial tone concept is discussed as well as the current status of the experimental efforts to this end.

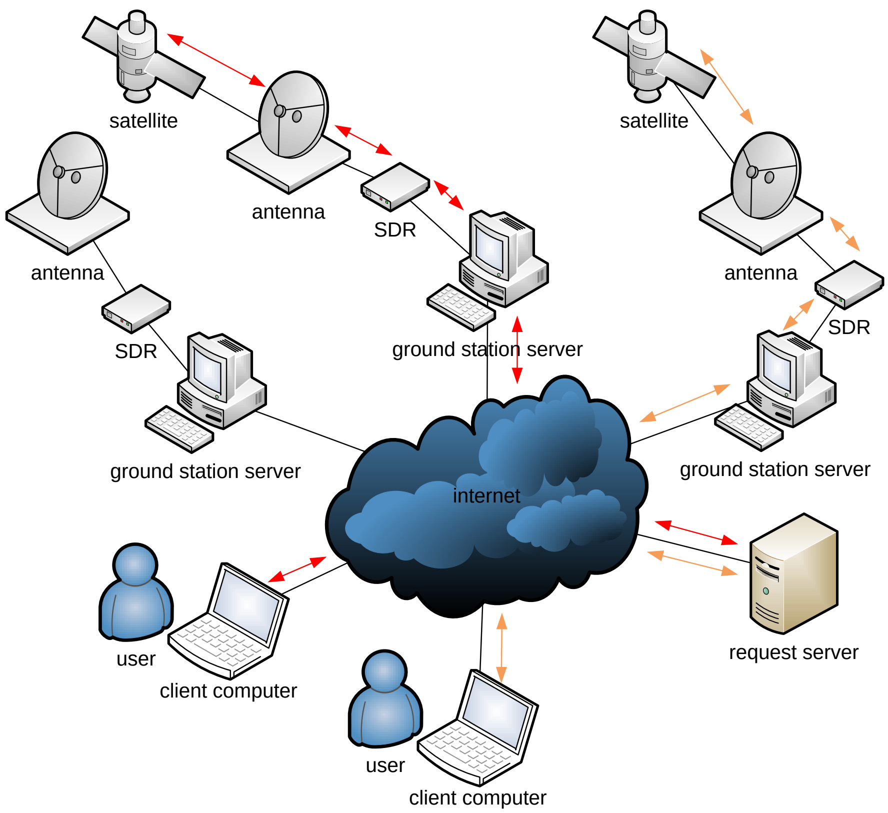


Figure 1. Space dial tone provides a connection between the client and the satellite over the internet. The user is able to communicate with his satellite through a remote ground station even when the satellite is on the other side of the earth. The path between the client and the satellite is determined by the request server, without user intervention.

II. CubeSats: Motivation for the Space Dial Tone Vision

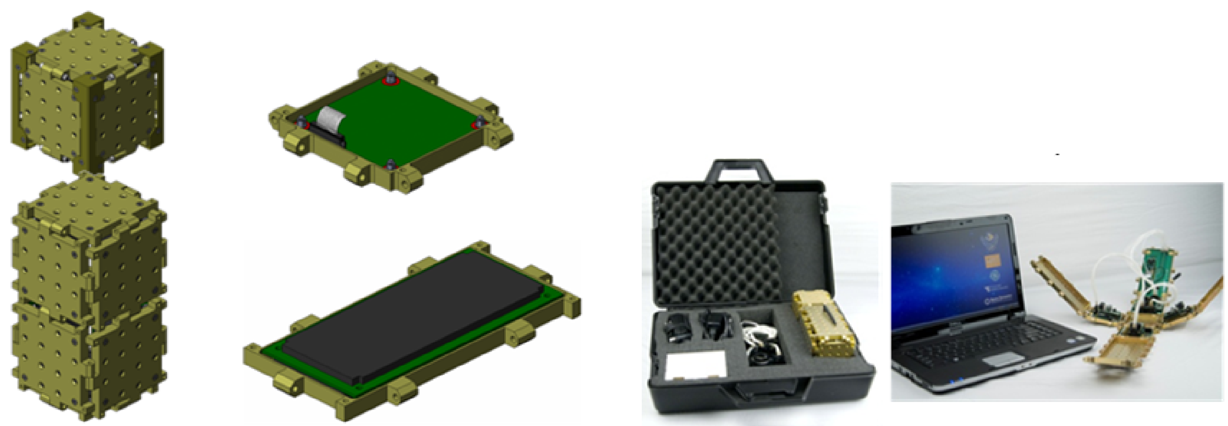
The Cube Satellite or CubeSat was introduced in the late 1990s,[2] initially as an educational concept at Stanford University. Since that time, well over 100 groups worldwide have studied and attempted to develop their own CubeSats, some to the point of successfully launching and operating them.[3] CubeSats are defined as simple parallelepiped spacecraft, where each key dimension is typically an integer multiple of 10 cm. The simplest standard configuration, approximately 10 cm x 10 cm x 10 cm, is referred to as a 1U CubeSat, whereas a 2U CubeSat configuration has dimensions of approximately 20 cm x 10 cm x 10 cm, and a 3U CubeSat configuration has dimensions of approximately 30 cm x 10 cm x 10 cm. The popularity of the 1U and 3U CubeSats can largely be attributed to the advent of a simple dispenser, referred to as the Poly-

Picosatellite Orbital Dispenser, or more commonly the PPOD,[4] which can be viewed as a simple, specialized cargo container, capable of accommodating a total of three 1U equivalent spacecraft (for example, three 1U CubeSats or a single 3U CubeSat). PPODs can be effectively integrated into a variety of launch vehicles to exploit otherwise unused excess lift capacity available in fielding other satellites. The ability to separate the complexity of launch from the complexity of the spacecraft being launched (to first order) has produced an interesting confluence in which launch vehicle integrators have become tolerant of accommodating simple PPOD/CubeSat combinations as ballast alternatives, while CubeSat developers have been emboldened to pursue opportunistic launch opportunities.

As of early 2011, dozens of CubeSats have been launched, with varying degrees of success, and the popularity of CubeSats has continued to grow for a number of reasons:

- They are gaining the attention of civilian and military sponsors,[5] who are exploring their utility for a range of simple roles, with the eventual hope of using them to possibly displace other more expensive classes of spacecraft.
- The opportunities for parasitic launch is expected to increase, due to the growing acceptance of rugged, compartmentalized dispensers such as the PPOD.
- Groups that cater to the burgeoning CubeSat development market have emerged that offer a variety of simple components, ranging from radios[6] and power supplies[7] to full development kits.[8]

The Air Force Research Laboratory Space Vehicles Directorate has introduced additional concepts in modularity and intelligent interfaces to promote the interchangeability implied in the CubeSat notion. The resulting set of ideas, referred to as CubeFlow (Figure 2),[9] involved the creation of simple pegboard-like structures (Figure 2(a)) that could be composed from basic building block modules (Figure 2(b)) into many flexible arrangements to form CubeSat and other satellite systems. Complementing the physical modularity, simpler forms of the Space Plug-and-Play Architecture (SPA)[10][11] and development tools were developed and infused into the CubeFlow concept. In this sense, CubeFlow attempted to introduce to CubeSats concepts similar to the plug-and-play (PnP) model of the personal computer, in which most components are freely interchangeable using standard interfaces without need for customized circuitry and software. Teaching kits employing the CubeFlow architecture (Figure 2(c)) have been created and distributed to research organizations for studying both CubeSat and SPA concepts.



(a) (b) (c)

Figure 2. CubeFlow modular architecture for CubeSats. (a) 1U and 2U CubeFlow examples, depicting peg-board style modular structure concepts. (b) Examples of interchangeable components, employing plug-and-play interfaces and the CubeFlow structural format. (c) Cubeflow development kit, featuring CubeFlow structures, integrated spacecraft computer, interface modules, and development tools.

While CubeSats are an attractive concept for budget-strapped research groups, there have been significant barriers affecting their more widespread adoption, even outside of their basic limitations as tiny platforms. One of these has been the lack of simple interchangeability of CubeSat components, a factor that we believe can be ameliorated through concepts such as CubeFlow. A second factor has been the lack of ability to specify orbits on demand, owing to the parasitic nature of most CubeSat launches (i.e., CubeSats are limited to orbits imposed by the primary satellite that drove the opportunity for a particular launch opportunity). We believe that creative solutions such as orbit-changing propulsion modules[12] and dedicated manifests will eventually alleviate this problem, but we shall not discuss those concepts further here.

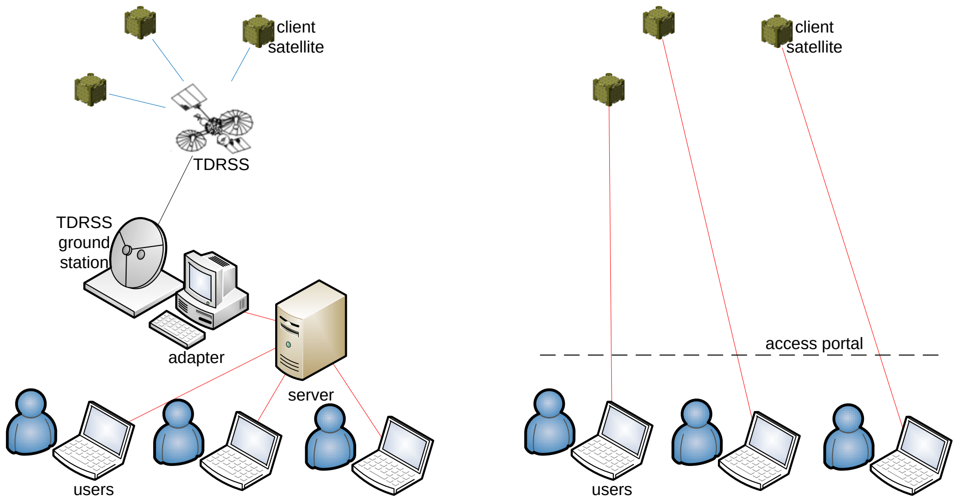
Another factor, in many ways as significant as lack of modularity and problematic launch, is the lack of a uniform communications infrastructure. The development of ground stations, while at some level well-disciplined, is an intensively customized proposition for virtually any new space system. For the most part, present ground stations involve dedicated facilities with complex, proprietary stovepipe equipment, labor-intensive operations, bureaucratic negotiations and arcane, inconsistent user interaction models. While we believe this problem is generic across the global space fleet, its effects are particularly acute for CubeSats. What it means at present is that each new CubeSat or CubeSat constellation must create - often from scratch - a space-ground command, control, and user communications architecture. Though they can take advantage of existing standards, protocols, and in some cases the ground equipment itself, the generation of a communications plan for CubeSats can be a difficult undertaking. Radio equipment must be purchased with a particular ground facility (or set) in mind, and multiple layers of protocols must be negotiated to form a complete operational concept in which commands can be conveyed by a user, usually through dedicated operators, to a spacecraft and telemetry extracted, interpreted, and transported. Changing from one facility to another can introduce wild variations that can ripple across the selection of equipment on the ground, on the satellite, its programming, and a complex, inter-dependent web of associated factors.

The space communications model is in sharp contrast to the terrestrial dial tone metaphor, where users can be blithely unaware of the nature of cell tower architectures, roaming, transitions to land line carriers, most operational constraints, variations in protocol (CDMA, AMPS, 3G, etc) and equipment (handsets) while still effectively negotiating the placement of phone calls. The telephony infrastructure, while far more complex than most satellite communications systems, is far simpler to access, even across an extremely wide variation in user equipment and service providers. It is easy to see that had cellular telephony followed the space communications model, the results would have been disastrous, since the simple act of placing a phone call would have required intimate awareness of a variety of networks and intensive negotiation, a wholly unscalable proposition.

We argue that the advent of proliferated (and responsive) space access through simple satellite platforms such as CubeSats is placing space communications in a similarly untenable position. It will simply not be possible to spend the additional cost and time preparing a communications concept for many simple designer satellites, much less pay for a dedicated human operator for each of them. As in the case of the cellular telephony system, which were engineered around the idea of supporting the old dial tone metaphor (even though the dial tone itself means nothing to the cellular handset), a space dial tone really is about engineering a user experience. This user experience defines a model that we implement through a combination of technologies, on the ground as well as in the spacecraft.

III. The Space Dial Tone Concept

The concept of how a space dial tone concept would work can be illustrated through simple examples, such as shown in the Tracking and Data Relay Satellite System (TDRSS) example shown in Figure 3. In this case, a number of CubeSat clients independently transmit telemetry through the TDRSS multiple access (MA) mode. These transmissions are picked up by one of the TDRSS platforms, which relay the information to the TDRSS ground station at White Sands Complex (WSC). In this example, a (presently non-existent) portal is established using a server that accepts the groomed feed from all CubeSat clients laundered through the WSC ground station. Since WSC would not natively support a connection to the server, an adapter would be installed at the ground station to accommodate the connection. A number of users log into this portal, which individually authenticates the them against particular CubeSats and allows them access to telemetry bundles, which would presumably be accumulated in a database system. This physical representation (Figure 3(a)) can be virtually replaced with the abstracted connection diagram shown in Figure 3(b). In this diagram, users virtually connect to individual spacecraft and need not be aware of the ground station, server, adapter, or relay satellite. As in the case of the terrestrial cellphone users, they can make an effectively direct connection to their client. The space dial tone concept is based on two principles: the Universal Portal Architecture and portal adapters.



(a) (b)

Figure 3. Tracking and Data Relay Satellite System (TDRSS). (a) Representative physical configuration. (b) Virtualized (user-perceived) configuration.

Universal Portal Architecture

The first of these is that users access spacecraft through a “universal portal architecture”. The simplest model for this portal is a web site (represented by the “server” in Figure 3(a)). In a broader setting, the server comprises a scalable web enterprise architecture, comprised potentially of many individual server machines, in the tradition of standard websites (such as Wikipedia). Users log in and access satellite data directly. Both they and other nodes of the overall network (ground stations through adapters, perhaps eventually individual satellites themselves) authenticate into the server farm. The data could be implemented in traditional relational databases (e.g. SQL, MySQL) and presented to the user in a number of standard forms through a browser interface. As such, users can define trend analysis as web applications directly on the portal or export the data in raw or filtered form for external analysis. The universal portal architecture inherits the richness and power of a web environment. It is, for example, straightforward in this architecture to define text message alerts or Twitter broadcasts when a particular telemetry point exceeds a user defined criteria.

Portal Adapters

Ground portions of existing communications architectures can connect to the authentication server (or server farm) through adapters. These adapters involve a back-end design to translate the dialect of particular ground stations into a uniform protocol that is easily processed by the authentication server system. Since most ground stations involve custom, proprietary designs, the portal adapters require some effort to engineer a compatible connection to the space dial tone system. Done correctly, the adapters are user transparent and can themselves be maintained through the overall portal system.

A more generalized exposition of the space dial tone concept is depicted in Figure 4. In principle, the broadest class of commercial, amateur, and military waveforms could be accommodated eventually as nodes in the space dial tone system. In this diagram, protocol labels (such as “Z1”, “Y1”) are generically assigned based on four broad categories.

1. Z-series

   Spacecraft radio frequency waveform stacks (a waveform set combined with underlying protocols) are assigned labels from the Z-series. The connection to plug-and-play modularity is enforced in this concept through the notion that modular spacecraft radios are similar to modems on a personal computer. In a personal computer, access to the internet, whether through a dial-up connection, wired

Ethernet, or wireless, is more or less treated the same to applications (e.g., the same browser is used in any of these cases). As such, we would expect modular CubeSat radios to respect the interface definitions of a plug-and-play radio template that accommodates all possible variations of an overall class of spacecraft communications devices. Just as a personal computer may use several modems, a modular spacecraft can use several modular plug-and-play radios. Of course, additional flexibility is afforded when one physical radio is capable of emulating several Z-series protocols, as we would expect with very sophisticated software definable radios.

2. ZB-series

   We also arbitrarily denote a ZB-series which depict the customized protocols that would exist between a particular ground station and its adapter. Included among these are test only configurations, not intended for flight, which exist for the ease and convenience of testing a CubeSat in an office or laboratory environment. Of course, these Z and ZB-series protocol stacks are not directly visible to the space dial tone system, but must be interpreted through ground stations designed specifically for this purpose, which eventually decode a base protocol.

3. U-series

   The base protocols denoted as a U-series, refer to those protocols that are ingestible to servers in the universal portal architecture, as conveyed by ground adapters. In Figure 4, these U protocols are shown as being transported over the standard internet, which is a useful but non-essential requirement (for cases where integration over private or classified networks is necessary).

4. Y-series

   These are the user protocols of the space dial tone networks. Y-series protocols are the only protocols that most users would directly interact with. Hence the Y-series protocols define the user experience of the space dial tone model suggested in Figure 3(b). The Y1 protocol refers to the standard login portal, whereas other web-based or web-triggered transports (e.g., SMS messaging) could be considered as other Y-series protocols.

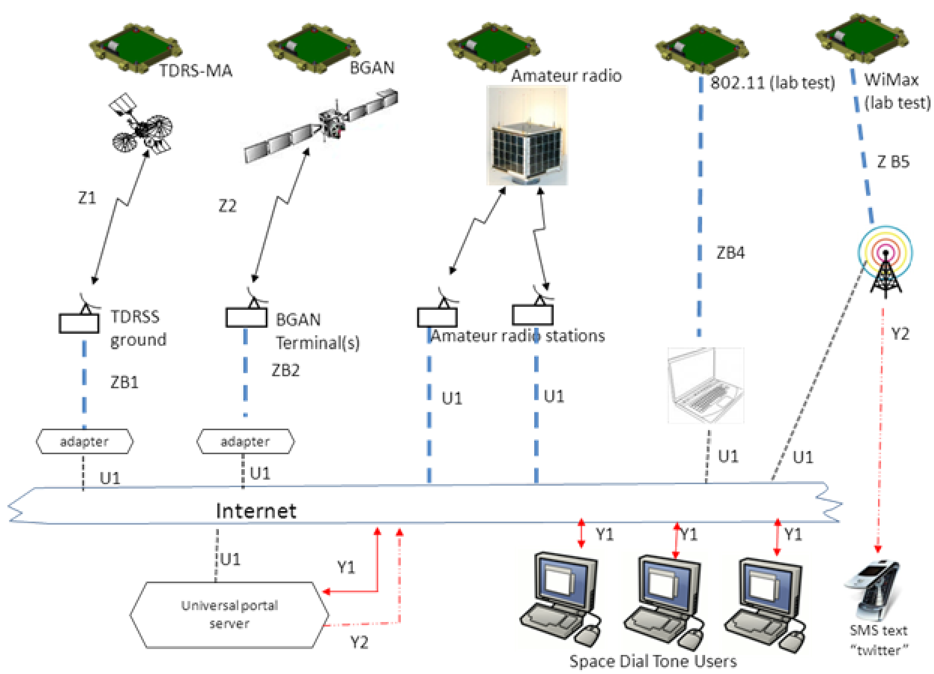


Figure 4. Generalization of the space dial tone concept. Z protocols are from radios in spacecraft to the ground. 2B protocols pertain to connections between ground stations and adapters. U protocols connect adapters to universal server. Y protocols are the only protocols that the users directly interface.

IV. GENSO

Although the CubeSat specification has reduced the cost of launching small satellites, it has not reduced the costs of the ground segment. Traditionally, users of small satellites would build a single small ground station before launch. This ground station would be idle during the majority of the mission, since a satellite in LEO is visible during only a few short passes per day. The data throughput between the satellite and the ground was limited by these small communication windows. Alternatively, arrangements with amateur radio operators or another ground station could be made, but these agreements required the remote operator to attend his equipment and manually send the data.

The Global Educational Network for Satellite Operation[13] (GENSO) system is a software networking standard which allows a user to communicate with a spacecraft by using a remote ground station which has a clear view of the spacecraft. Communications between the client computer and the ground station server are conducted across the internet. A request server authenticates and schedules the requests from the clients, so that no user can monopolize the use of the network. In order to earn the privilege of using remote ground stations, each client must contribute to the network by permitting other users to access his ground station when it is idle.

GENSO permits missions which are impossible using a single ground station. It is possible to track and communicate with spacecraft over other areas of the world, increasing the amount of experimental data which can be transmitted to the ground. It provides a secondary method of communication with the satellite during poor weather or equipment failure. Rather than sitting idle, each ground station associated with the network can be used to its maximum capacity. Each satellite can remain in constant communication with the ground, as long as it is visible to a GENSO station, making it possible to monitor telemetry data over the maximum amount of the orbit and identify problems and issue commands faster. GENSO allows the transmission of data to become more frequent and effective.

A. Reference design

Although the GENSO standard contains a hardware abstraction layer which makes it possible to write drivers for new ground station equipment such as radios and antenna rotators, many of the active nodes on the network currently use the same hardware. This hardware is summarized in Table 1.

Table 1. Typical GENSO hardware[14]

Component	Model
Radio	Icom IC-910H
TNC	Kantronics KPC-3+
Antenna rotator	Yaesu G-5500
Antenna controller	Yaesu GS-232B

The antenna is connected to the radio which brings the signal down to baseband and passes the baseband signal to the terminal node controller (TNC). The TNC is a device used to send and receive digital data using the AX.25 communications standard. This standard is used by amateur radio operators to communicate over packet radio networks, and it is frequently used to communicate with small spacecraft. The TNC is connected to the ground station server using a serial interface. The azimuth and elevation of the antenna are controlled in order to keep the antenna pointed at the spacecraft as it moves across the sky. The ground station server sends serial commands to the antenna rotator controller, which actuates the antenna rotator.

B. Limitations

The hardware in Table 1 does an excellent job of receiving AX.25 packets, but it limits the capabilities of the GENSO network, preventing it from receiving other waveforms. The TNC is a specialized piece of hardware which is only capable of receiving AX.25 packets, making it impossible to receive signals which use another protocol or modulation scheme. For example, many amateur spacecraft such as RS-15, AO-7, VO-52, RS-30, ITUpSAT1, BEESAT, and SwissCube send a telemetry data using Morse code. Other amateur spacecraft such as ARISS, HO-68, SO-50, AO-27, SO-67, AO-51, and AO-16 have an analog FM repeater.[15] Some satellites, such as AO-27, support digital voice using D-Star.[16] Since one of the stated goals of the amateur radio service is to promote experimentation,[17] it is possible for amateur satellites to use experimental communications protocols and modulations. The hardware TNC lacks the flexibility to send or receive these signals over the GENSO network.

Groundstations associated with the GENSO network must be extremely flexible in order to accommodate all of these communication schemes. However, equipping all of the GENSO ground stations with hardware to send and receive these waveforms would be cost prohibitive. The introduction of each new waveform would require hardware upgrades. It is unlikely that the volunteer station operators would be willing to continually upgrade their ground station hardware in order to support satellites launched by other groups. However, if support for new waveforms could be introduced using an automated software update, the capabilities and flexibility of the GENSO network would be greatly improved.

In the United States, the band plan[18] distributed by the American Radio Relay League (AARL) reserves portions of five bands specifically for amateur satellite communications, as shown in Table 2. The Icom IC-910H radio is capable of communicating using the frequencies in Table 3. Note that this radio is only capable of communicating on two of the five possible bands. Optionally, the radio can be upgraded to add 23 centimeter operation.

Table 2. Amateur frequencies reserved for satellites

Band	Description	Range (MHz)
10 m	Satellite downlinks	29.300-29.510
2 m	OSCAR (new)	144.30-144.50
2 m	OSCAR (old)	145.80-146.00
70 cm	Satellite (international)	435.00-438.00
23 cm	Satellite uplinks (WARC '79)	1260-1270
12 cm	Satellite	2400.0-2403.0
12 cm	Satellite high-rate data	2403.0-2408.0
12 cm	Satellite	2408.0-2410.0
12 cm	Satellite	2430.0-2433.0
12 cm	Satellite high-rate data	2433.0-2438.0

Table 3. Icom IC-910H frequency coverage

Band	Direction	Range (MHz)
2 m	TX	144.0-148.0
	RX	136.0-174.0
70 cm	TX	420.0-450.0
	RX	420.0-480.0

C. Alternate Implementation of User Protocols

Another interpretation of the space dial tone was provided by a team from the University of Michigan.[19] Their concept, referred to as WolvComm, defined the authentication server equivalent using a Linux, Apache, MySQL, PHP (LAMP) architecture, with a simple U1 type protocol, extended from the Consultative Committee for Space Data Systems (CCSDS). The novel twist in their approach was the use of the commercial file-syncing technology provided by Dropbox,[20] which would allow users to implement satellite commands and retrieve telemetry through a simple file input/output (I/O) scheme. Presumably, one could develop client applications by simply manipulating files in the Dropbox, which would synchronize with the ground adapters and authentication servers at the rate established by the Dropbox server system. The novelty of this scheme is that the clustering of users and satellites would be definable by simply implementing sharing privileges in a master Dropbox account.

V. Software Defined Radio

A software defined radio is a radio in which several of the key components which are traditionally implemented in hardware, such as coding, modulation, filtering, and error correction, are implemented in software. Since the key features of the radio are defined using software, it is possible to quickly and dynamically reconfigure the radio to perform new tasks.[21] Unlike traditional radios, a software radio receiver digitizes the received waveforms as soon as possible using a fast analog-to-digital converter (ADC). A software radio transmitter keeps the signal digitized for as long as possible before sending the signal through a digital-to-analog converter (DAC). Since the processing of the waveforms is accomplished in software, a single software radio may have the same functionality as multiple hardware radios.

A GENSO station which uses a hardware radio and an AX.25 TNC is only be capable of communicating with satellites which communicate using AX.25. However, ground stations which use an SDR such as the Universal Software Radio Peripheral 2 (USRP2) can be programmed to transmit or receive any signal which is within the frequency range and bandwidth of the radio and antennas. Updates containing the software for communicating using new waveforms could be automatically shared among GENSO stations. Using this system, support for new waveforms could be added remotely without the need for hardware upgrades, or the intervention of the operator of each ground station.

Unlike the Icom IC-910H radio which is only capable of communicating in the 2 meter and 70 centimeter bands (and optionally the 23 centimeter band), the USRP2 uses modular daughterboards which allow it to communicate over a wide range of frequencies. A combination of three USRP2s with different daughterboards would permit communications in all five of the amateur satellite bands. The USRP2 contains two 14-bit 100M sample/sec ADCs, two 400M sample/sec DACs, and a FPGA for high speed operations like decimation and interpolation.

The USRP2 is typically controlled using software developed using the GNU Radio environment. Simple applications can be created using the GNU Radio Companion (GRC), a graphical tool used to draw radio flow graphs. GNU Radio uses a system of signal sources, signal sinks, and processing blocks which make up a signal flowgraph. The flowgraph describes the flow of a signal from a signal source to a signal sink. Blocks describe the signal processing steps which are applied to the digitized signal. GRC allows a person with signal processing background but little programming background to build a radio system using a drag-and-drop interface. After the design is complete, GNU Radio Companion converts the graphical description of a flowgraph into a Python file describing the same system. Signal sources, sinks, and blocks are usually written using a compiled language such as C++ to ensure fast computational speed. Due to the open source nature of GNU Radio, it is possible to write custom blocks which implement new signal processing operations.

VI. Implementation

We have constructed two satellite ground stations capable of communicating with amateur satellites. The primary ground station is capable of transmitting and receiving. It has antennas for the 2 meter, 70 centimeter, and 12 centimeter bands. It was originally set up using the standard GENSO hardware in Table 1 and successfully operated as a node on the GENSO network in that configuration. The station is also equipped with two USRP2 radios.

Although the USRP2 radios have not yet been integrated into the GENSO network, a custom program in GNU Radio has been developed to operate the USRP2, antenna rotators, and antenna polarization relays. The software communicates with a satellite tracking server on the network in order to obtain information on the position of the satellite in the sky and its expected Doppler shift. A custom GNU Radio block was developed which controls the antenna rotator over a serial interface. The Doppler shift obtained from the tracking server is used to adjust the frequency of the USRP2 to compensate for the Doppler shift created by the motion of the satellite. The antennas each contain a relay which switches between right-hand and left-hand circular polarization. A custom GNU Radio block was developed to control these polarization relays by using the general-purpose I/O (GPIO) pins on the USRP2.

The software radio at this ground station has successfully received and demodulated FM signals from amateur satellites in LEO, including AO-27, AO-51, and HO-68. These satellites transmit in the 70 centimeter band. The software required to demodulate additional waveforms, including AX.25, is currently undergoing development. The integration of this software into GENSO could allow the USRP2, antenna rotator, and antenna relays to be controlled remotely, and new modulators or demodulators to be added remotely.

VII. Conclusions and Future Work

Operators of small satellites can improve their connectivity with their spacecraft by pooling the resources of their ground station with the resources of other operators into a global network. This approach may eventually provide worldwide communications coverage for small satellites in LEO. The use of software-defined radios would improve the flexibility of this network, and allow support for new waveforms to be added without hardware upgrades or intervention of an operator at every ground station. This would allow the operators of small ground stations to contribute to the larger goal of establishing space dial tone.

A driver to interface the USRP2 and the GENSO hardware abstraction layer must be written before the USRP2 can be controlled remotely. The software radio code for additional waveforms, such as AX.25, needs to be written so that the USRP2 can communicate with a larger number of amateur satellites. Finally, the USRP2 radios have only been used for receiving transmissions. Before transmissions can be made, a power amplifier needs to be added between the USRP2 and the transmit antenna, and the software used to transmit a signal needs to be completed.

Acknowledgments

We would like to thank to Jim White, WD0E, of Colorado Satellite Services for helping us to improve the performance of our antenna system.

References

[1]Pierucci, L., *Satellite Personal Communications for Future-generation Systems: Final Report: COST 252 Action*, Springer, 2002.

[2]Nugent, R., Munakata, R., Chin, A., Coelho, R., and Puig-Suari, J., "The CubeSat: The Picosatellite Standard for Research and Development," *AIAA Space 2008 Conference & Exposition*, San Diego, CA, 2008.

[3]Chin, A., Coelho, R., Brooks, L., Nugent, R., and Puig-Suari, J., "Standardization Promotes Flexibility: A Review of CubeSats Success," *AIAA 6th Responsive Space Conference*, Los Angeles, CA, 2008.

[4]Nason, I., Puig-Suari, J., and Twiggs, R., "Development of a Family of Picosatellite Deployers Based on the CubeSat Standard," *IEEE Aerospace Conference Proceedings*, Big Sky, MT, 2002, pp. 457–464.

[5]Klamper, A., "NRO Embraces Cubesats for Testing Advanced Technologies," *Space News*, 10 August 2009.

[6]"AstroDev Helium Radios," `http://www.astrodev.com/public_html/downloads/datasheet/Helium-Specification.pdf`, 26 July 2009.

[7]Clyde Space, "3U CubeSat EPS," `http://www.clyde-space.com/cubesat_shop/eps/12_3u-cubesat-eps`, February 2011.

[8]`http://cubesatkit.com`, February 2011.

[9]McNutt, C. and Lyke, J., "CubeFlow: A Modular Open Systems Architecture for CubeSats," *Proceedings of the 7th Responsive Space Conference*, Los Angeles, CA, 27-29 April 2009.

[10]Lyke, J., "Plug-and-play as an Enabler for Future Systems," *AIAA Space Conference*, September 2010.

[11]Lyke, J., Mee, J., Bruhn, F., Chosson, G., Lindegren, R., Lofgren, H., Schulte, J., Cannon, S., Christensen, J., Hansen, B., Vick, R., Vera, A., and Calixte-Rosengren, J., "A Plug-and-play Approach Based on the I2C Standard," *Proceedings of the 24th Annual Conference on Small Satellites*, Logan, UT, 8-10 August 2010.

[12]Moore, G., Holemans, W., Huang, A., Lee, J., McMullen, M., White, J., Twiggs, R., Malphrus, B., Fite, N., Klumpar, D., Mosleh, E., Mashburn, K., Wilt, D., Lyke, J., Davis, S., Bradley, W., Chiasson, T., Heberle, J., and Patterson, P., "3D Printing and MEMS Propulsion for the RAMPART 2U CUBESAT," *Proceedings of the 24th Annual Conference on Small Satellites*, Logan, UT, 8-11 August 2010.

[13]European Space Agency, "Global Educational Network for Satellite Operations," `http://www.esa.int/esaMI/Education/SEMK003MDAF_0.html`.

[14]"Ground Station Hardware," `http://www.genso.org/index.php/ground-station-hardware`.

[15]Radio Amateur Satellite Corporation, "Satellite Frequencies," `http://www.amsat.org/amsat-new/satellites/frequencies.php`, February 2011.

[16]Wyrick, M., "AO-27 Does D-Star," `http://www.ao27.org/AO27/index.shtml`.

[17]Federal Communications Commission, "Amateur Radio Service," 47 CFR 97.1.

[18]American Radio Relay League, "Band Plan," `http://www.arrl.org/band-plan-1`.

[19]Freyberg, M., Heckathorn, C., Lyjak, A., and Mucino, J. E., "Wolverine Communications," *Course AOSS/AERO 583 student project presentation, University of Michigan*, May 2010.

[20]`http://www.dropbox.com`, May 2010.

[21]Reed, J. H., *Software Radio: A Modern Approach to Radio Engineering*, Prentice-Hall, 2002.