Salo, Timothy J. "Proposed Network-Centric Architecture for the Advanced Communications Package (ACP)". *Proceedings of the AMSAT-NA 26th Space Symposium, Atlanta, GA, October 24-26, 2008*. Silver Spring, MD: The Radio Amateur Satellite Corporation, 2008. 33-43. <http://www.saloits.com/papers/AMSAT2008.pdf>

Proposed Network-Centric Architecture for the Advanced Communications Package (ACP)

Timothy J. Salo
Salo IT Solutions, Inc.

Abstract –This paper suggests a system-level architecture for the Advanced Communications Package (ACP). The architecture presented here employs the lessons of the Internet architecture and leverages the Internet protocols to create a network-centric, system-level architecture for the ACP. This high-level architecture is developed by partitioning the required functionality into subsystems, defining interfaces between those subsystems, and assigning the subsystems to hardware platforms. Following that, several fundamental design issues are highlighted, and potential solutions to these challenges are outlined. The resulting system-level architecture should stimulate thought and discussion about the design of the ACP, and may provide a foundation for future ACP design and implementation efforts.

1. Introduction

The Advanced Communications Package is a novel, satellite-based, digital communications system that is enabled by software-defined radio (SDR) technology. The ongoing ACP design work has, up to this point, focused on the lowest communications layers and on the application layer. This paper examines everything that is architecturally in between the lowest communication layers and the ACP applications. The result is a proposed network-centric, system-level architecture. Important benefits of this proposed system-level architecture are that it:

1. Simplifies the design and implementation of the ACP system by placing clear boundaries on the ACP system,

2. Provides a general communications service that supports a broad range of applications, both those that are already planned, as well as not-yet-conceived applications,

3. Opens application development to *any* interested party, by cleanly separating applications from the ACP communications services and by providing a standard interface between the two,

4. Leverages existing Internet protocols and the Internet architecture to enhance interoperability and reduce technical risk, and

5. Helps to identify architectural and design issues that warrant additional study.

2. The Advanced Communication Package (ACP)

The Advanced Communications Package (ACP) uses software-defined radio technology to

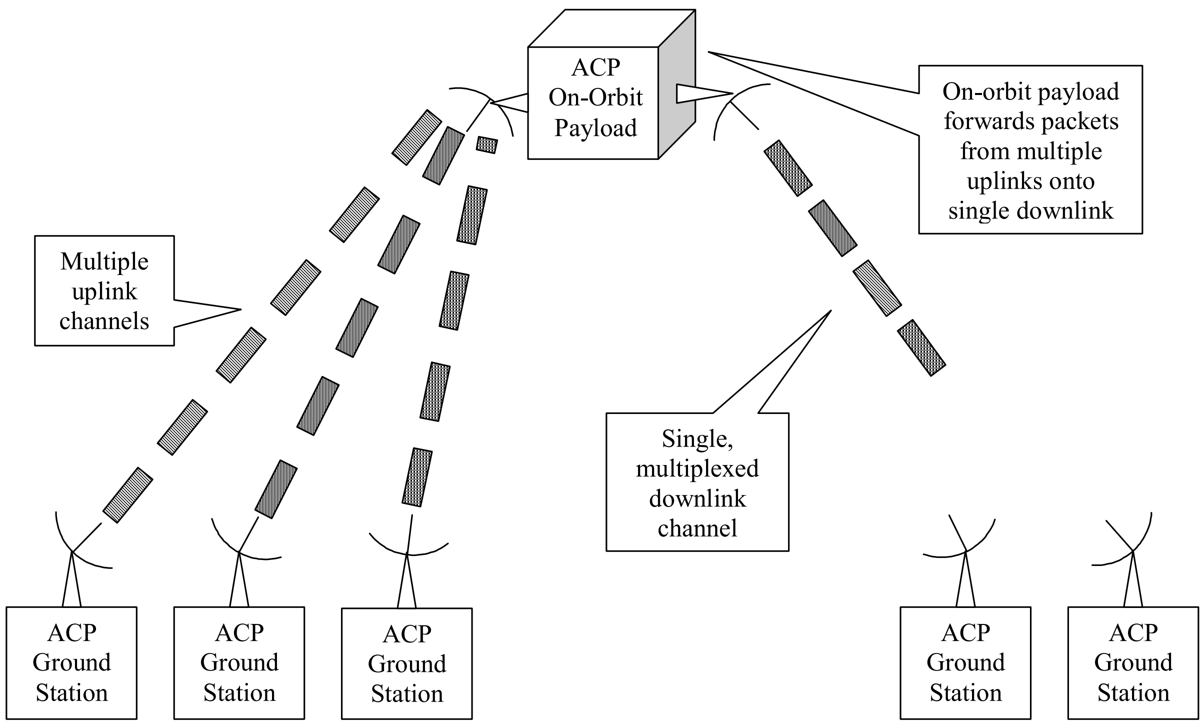


Figure 1. Advanced Communications Package (ACP) Configuration

provide a satellite-based, multi-user, digital communication system [Ettus 2006], [Ettus 2007], [Ettus 2008]. ACP ground stations transmit data to the on-orbit ACP payload on different channels. The ACP payload employs SDR technology to demodulate all of the uplink channels simultaneously. Packets received by the ACP payload on the uplink channels are statistically multiplexed onto a single downlink channel. Each ACP ground station extracts from the single, multiplexed downlink the packets in which it is interested. Figure 1 below illustrates how the ACP payload multiplexes packets received on different uplink channels onto a single, statistically multiplexed downlink channel.

An important design objective of the ACP project is to minimize the complexity of the ACP on-orbit payload. As a rule, intelligence should be placed in the ground stations, rather than in the ACP payload. Consistent with this objective, the ACP payload should do little more than forward packets from the uplink channels onto the downlink channel.

The ACP will support numerous simultaneous channels [Thompson 2008b]. Each ground station will use its own uplink channel (although the ACP payload won't be able to prevent ground stations from sharing an uplink channel, either inadvertently or intentionally).

Multiple classes of ground stations will be supported. Less-capable ground stations will transmit at lower bit rates, while the ACP payload will encode packets destined for less-capable ground stations in a fashion that makes them easier to decode reliably. The ACP payload will forward fixed-length 225-byte packets.

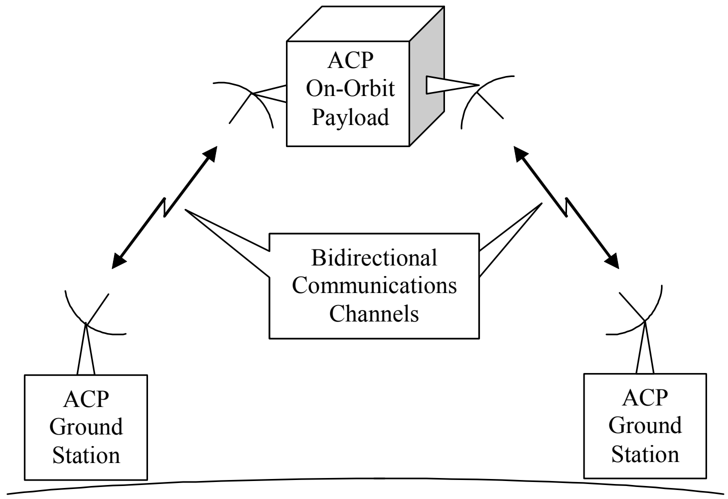


Figure 2. ACP Configuration - Abstracted View

The ACP system is being developed by the Radio Amateur Satellite Corporation – North America (AMSAT-NA). AMSAT anticipates that the ACP payload will be carried on either a high earth orbit (HEO) satellite or a geosynchronous earth orbit (GEO) satellite. Information about the project can be found on the AMSAT Web pages (http://www.amsat.org/).

Most of the design work on the ACP to date has focused on the lowest communications layers (e.g., modulation techniques and forward error correction (FEC) technology) and on the applications that will use the communications services provided by the ACP payload. This paper outlines a proposed system-level architecture that uses the lower-level services provided by the ACP payload in order to support ACP applications, both the applications that have been proposed, as well as those that have not yet been invented.

For the purpose of exploring a system-level architecture for the ACP, it is useful to abstract away the details of the lower communication layers, such as how the uplinks and the down link are multiplexed and the details of the on-the-air encoding of bits and packets. In this abstracted view, illustrated in Figure 2, the ACP payload simply provides a bidirectional, packet-based, digital communications channel between two ground stations. With a little forethought, this simple model can easily be extended to include simplex communications (e.g., systems that transmit sensor data, but don't receive), point-to-multipoint communications (e.g., publish/subscribe disseminations models, such as RSS feeds) and multipoint-to-multipoint communications (e.g., full-duplex multiparty audio or video conferences).

3. Proposed ACP System Architecture: An Initial Refinement

The initial architectural design decisions segment the ground station two major subsystems, identify the boundary of the ACP system, and specify a standard interface between the ACP system and its users.

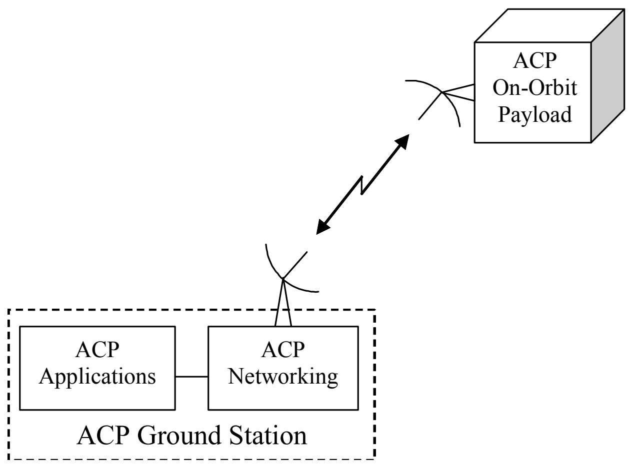


Figure 3. ACP Ground Station – High-Level View

3.1 ACP Ground Station Architecture

An important lesson of the Internet architecture is that it is often highly beneficial to separate the design and implementation of the network-related functionality from that of the systems that use the network [Clark 1988]. Applying this widely employed architectural principle to the ACP ground station yields the structure shown in Figure 3. From this perspective, the ACP ground station is composed of two major subsystems: "ACP Applications" and "ACP Networking". The ACP applications include those that are under consideration, such as text-based messaging, voice communications, and video conferencing [Thompson 2008a]. The ACP architecture and implementation should also permit additional applications to be easily developed in the future.

3.2 ACP System Boundary

Properly specifying the boundary of a system, deciding what functionality is provided by the system and what is not, is often critical to the success of a project. The system boundary shown in Figure 4 places the ACP applications *external* to the ACP system. In this view, ACP applications use the ACP system, but are not part of the ACP system. This architecture emphasizes the ACP system as a networking solution that interconnects applications. (Note that the scope of the ACP Ground Station is smaller than is shown in the previous figure.)

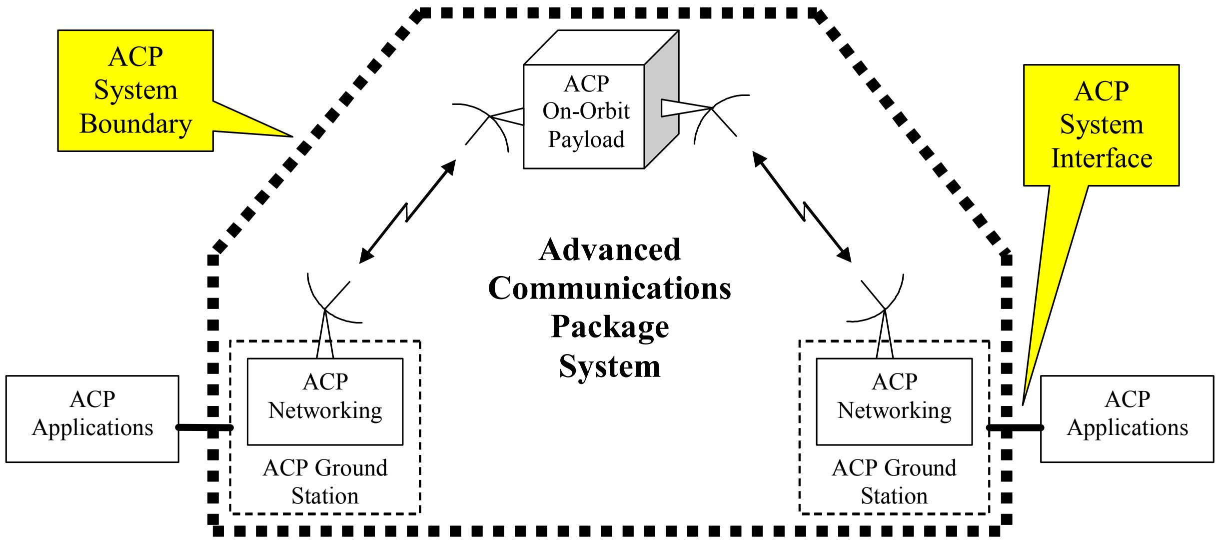


Figure 4. ACP System Boundary and Interface

3.3 ACP System Interface

A significant benefit of this particular specification of the ACP system boundary is that the ACP applications simply view the ACP system as a network that interconnects them with each other, a perspective summarized in Figure 5. The next fundamental design question is: What is the interface between the ACP applications and the ACP system? Describing the ACP system as a network makes one answer obvious: Use the Internet protocol (IP). Specifically, the ACP system forwards IP packets between end systems that host ACP applications. More directly, the ACP system appears to the ACP applications (and to the machines hosting these applications) to be simply an IP network.

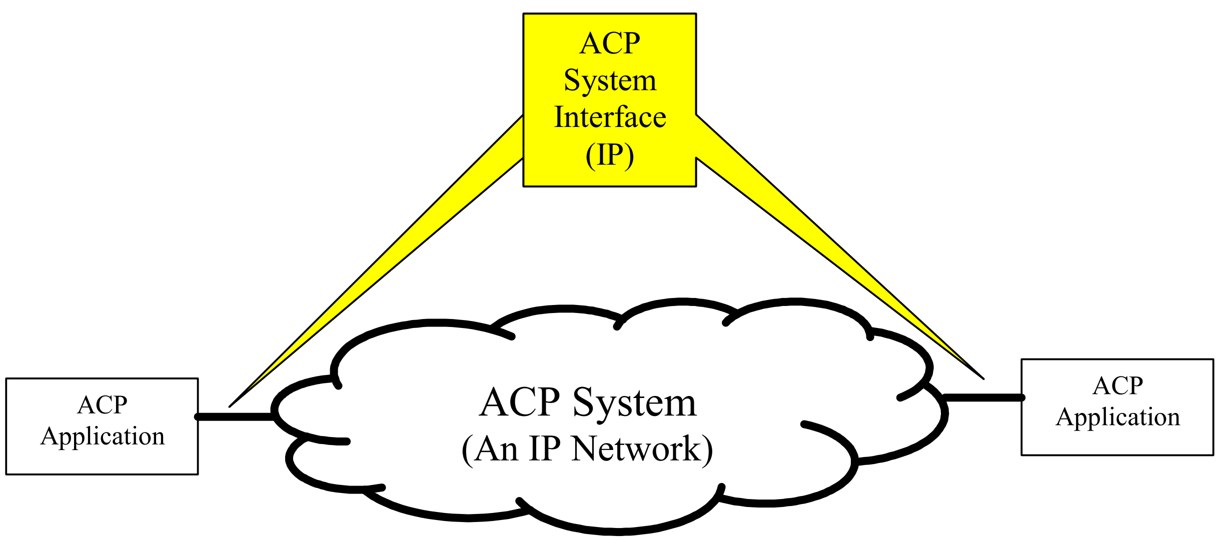


Figure 5. ACP Application Perspective of the ACP System

3.4 Assigning Ground Station Subsystems to Hardware

The architectural decisions described above facilitate the implementation of the ACP applications and of the ACP ground station on different hardware platforms. The ACP ground station can be hosted on a machine that is tailored to its needs, without regard to the needs of the application software, and conversely.

If the ACP applications and the ACP ground station are implemented on different machines, then the interface between them must include a physical link. Inasmuch as the applications and the ground station will exchange IP packets, the most general, applicable physical link is Ethernet. But, this shouldn't be a serious constraint: pretty much every readily available machine supports, or can easily be enhanced to support, an Ethernet interface and the IP protocol suite.

The machine selected for the ground station probably won't have a graphical user interface (GUI) or graphics hardware. And, the machine won't need a lot of memory (because it won't host the applications) and it might not even require a hard drive (perhaps using flash memory instead).

The machine that hosts the ACP applications can be, well, pretty much anything that is able to run the application software and that has IP software and an Ethernet interface. Presumably, most ACP applications will run under Windows or Linux, or under both. But, the application developers are free to choose. In fact, the developers of each application are free to make their own choices about which machine, operating system, and development environment to use, provided that their choices support IP over Ethernet.

4. ACP System-Level Architecture

The system-level architecture proposed here can be summarized succinctly as: The ACP system appears to external devices to be an IP network. More emphatically, external systems are, for the most part, unaware that they are communicating over a satellite-based communications system, rather than over, for example, the Internet.

This architecture naturally divides the ACP development effort into three projects:

- **ACP Applications** This project will develop applications that use the IP networking services provided by the ACP system. It ought to be staffed by software developers who think about human computer interaction (HCI) and graphical user interfaces (GUI). This team will develop application software that runs under Windows, Linux and maybe even MacOS.

- **ACP Ground Station** The objective of this work is to implement the functionality that makes the ACP system appear to external devices to be an IP network. This team will include developers who view the world in terms of packet formats and network protocols. The software they develop may operate in a more specialized environment, perhaps an embedded variant of Linux.

- **ACP On-Orbit Payload** The ACP payload must operate in a very demanding, inaccessible environment. These developers will be concerned with creating highly reliable software that can operate onboard a spacecraft. Fortunately, the functionality that this team must implement is limited to forwarding packets between the uplinks and the downlink, and perhaps a few other tasks that can't easily be performed elsewhere.

This proposed architecture makes the objective of the ACP development clear: the ACP system must look like an IP network; to its users, the ACP system must be *indistinguishable* from any other IP network. The test of whether the ACP development is complete and whether the development team has been successful is very simple: Does the ACP system exhibit the behaviors expected of an IP network?

This IP-based system architecture dramatically simplifies the development of ACP applications. To the extent that application developers are assured that the ACP system behaves like an IP network, they can develop and test their applications using any convenient IP network – like the Internet. If these applications work well over the Internet, but poorly with the ACP system, the first question should be whether the ACP system properly mimics the operation of an IP network. Of course, the ACP system *will* exhibit a few peculiarities that can't be controlled, such as high latency and perhaps high packet-loss rates. But, these unique characteristics can easily be identified. Furthermore, numerous techniques have been developed to mitigate the effects of high-latency paths and high packet-loss rates.

Another benefit of this architecture is that it makes it easy for applications to execute on existing machines. For users, deploying an ACP application should be no more difficult than installing a new software package on an existing machine (assuming, of course, that the machine is running a recent, common operating system and has adequate memory, storage and processing power). (Note that the ACP ground station networking functionality is likely to be deployed on a dedicated machine, but that is really an implementation decision that is up to the ACP development team.)

This architecture opens the development of new ACP applications to almost any motivated person or group. Little specialized knowledge is required, beyond that necessary to develop Internet-enabled software. And, no permission or special dispensation is necessary. The ACP system will transport any IP packets generated by an application, no questions asked. This open application development environment should stimulate the creation of numerous ACP applications, and may even attract new (and much needed) members to the amateur radio and amateur satellite communities.

By this point, it should be clear that even existing networked applications will be able to use the ACP system, and probably without modification. Some applications will require configuration, and some may require that servers be deployed. But, servers, whether they are Web servers or servers that support voice-over-IP (VoIP) networks, can use the ACP system as well – it's just another IP network. In fact, the ACP development team doesn't even need to implement any voice functionality – users could simply connect inexpensive VoIP phones or VoIP software (similar to the Skype software) to the IP services provided by the ACP system.

But, simply asserting that the ACP system will behave like an IP network doesn't mean that this is feasible, or even that it is possible. The remainder of this paper demonstrates the feasibility of this architecture by suggesting how the most significant functionality expected of IP networks might be provided.

5. Potential ACP Design Solutions

In this section, potential solutions are proposed for the most fundamental ACP system design issues. What follows is *not* a proposed design for the ACP system. Rather, this material is merely intended to demonstrate the feasibility of implementing the network-centric, system level ACP architecture proposed above. The actual ACP system design may, in some cases, employ alternative technologies.

5.1 End-to-End Routing

The principle function of the ACP system is to forward IP packets between end systems, namely hosts running ACP applications. Figure 6 illustrates the end-to-end path between *Host A* and *Host B*, including the addresses assigned to interfaces and partial, stylized route tables. Readers with an exposure to IP routing should be able to convince themselves that this information is adequate to forward packets from *Host A* to *Host B*. The next challenge is to dynamically maintain the appropriate information in the route tables.

5.2 Active Ground Station Announcements

One way to construct the necessary route tables is for every active ground station to periodically transmit Active Ground Station Announcements via the ACP payload. Perhaps, a special control channel could be dedicated to this function. If these announcements contain the following information, then every ground station will be able to create an up-to-date route table for the whole system:

- **Ground Station Address** The address of the ground station's ACP interface would be announced. Using the example above, *Ground Station A* would include its address ($Addr_3$).

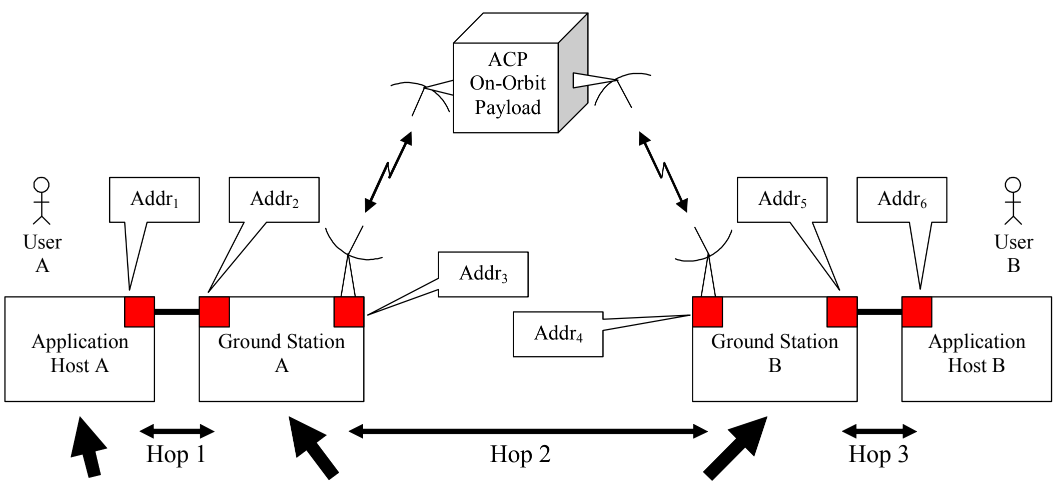
- **Ground Station Name** Ground stations could be assigned names, as well as addresses. Perhaps, these names could be of the form *host*.*call*.amsat.

- **Host Address and Name** The other ground stations also need to know about the hosts to which this ground station can connect. In this example, the announcements created by *Ground Station A* might include an address/name pair something like ($Addr_1$, "HostA.ab0do.amsat").

- **Text Description** While names are more informative than addresses, descriptive text might be even more useful. Perhaps, this example announcement will include the following text: "ACP design video conference". Or, perhaps this sort of information could be disseminated in an Available Resource Announcement.

- **Transmitter Identification** While the precise requirements for transmitter identification warrant additional study, the call sign of the transmitting station might also be included in this announcement (e.g., "AB0DO).

Each ground station could use a traditional routing protocol (e.g., the Route Information Protocol, RIP) to distribute the route information learned from the Active Ground Station Announcements to the hosts to which it provides service. Note that this Active Ground Station Announcement also provides some other capabilities that will be discussed shortly.



Route Table Host A	
Addr	**Next Hop**
$Addr_2$	Enet
$Addr_6$	$Addr_2$

Route Table G. S. A	
Addr	**Next Hop**
$Addr_4$	ACP
$Addr_6$	$Addr_4$

Route Table G. S. B	
Addr	**Next Hop**
$Addr_6$	Enet

Figure 6. End-to-End Routing in the ACP System

5.3 Resolution of Names to Addresses

Humans generally prefer domain names (e.g., "www.amsat.org") to numeric IP addresses (e.g., 128.54.16.15). The Active Ground Station Announcement described above could be used to make the devices in the ACP system appear as if they were part of the Internet Domain Name System (DNS). Each ACP ground station could implement a DNS resolver, software that translates domain names into numeric IP addresses. Hosts that want to use the ACP system (e.g., machines that host ACP applications) would consult the ACP resolver anytime that they need to translate a DNS name to an address. The ACP resolver would translate ACP names (e.g., names of the form *host*.*call*.amsat) to IP addresses, based on information learned from the Active Ground Station Announcements. The ACP resolver would forward traditional DNS names to Internet DNS servers. As a result, applications could treat names assigned to ACP devices as if they were Internet domain names. In fact, the applications won't even know whether a name is associated with an ACP device or with a traditional Internet device.

5.4 Segmentation and Reassembly (SAR) Protocol

Because ACP packets are much smaller than IP packets, a simple protocol is required to segment IP packets into multiple ACP packets, and to reassemble those ACP packets back into an IP packet. The ATM Adaptation Layer 5 (AAL5) may provide a useful example.

5.5 Uplink Channel Status Announcements

ACP ground stations will need to determine which uplink channel they should use. One approach is to let each ground station pick a channel that doesn't appear to be currently in use, and then deal with any collisions that might occur. Uplink Channel Status Announcements, messages that describe the status of the uplink channels, could be transmitted on the downlink periodically. Perhaps, each packet on the downlink could include the number of the uplink channel on which it was received. A single control ground station could monitor the downlink channel and periodically transmit Uplink Channel Status Announcements on the uplink channel, enabling every ground station to easily learn the status of the uplink channels.

6. Conclusion

This paper proposes a network-centric, system-level architecture in which the ACP system provides general networking services to its users, ACP applications. By appearing to be an IP network, the ACP system can support existing Internet-enabled applications, as well as open the development of new ACP applications to any interested party. This model has applicability well beyond satellite communications. It is beneficial to terrestrial wireless digital communication systems that use technologies similar to the ACP. In fact, this model is useful for *any* digital communications system. The benefit is the same: nearly any existing

Internet-enabled software or device can, with minimal effort, use the services offered by these communications systems.

7. References

1. [Ettus 2006] Ettus, Matt. "Proposed Eagle Advanced Communications Downlink Design" in *Proceedings of the AMSAT – North America 2006 Space Symposium*. San Francisco, CA, October 5-8, 2006, pages 195-200.

2. [Ettus 2007] Ettus, Matt. "Advanced Communications Package", slides presented at AMSAT – North America 2007 Space Symposium. Pittsburgh, PA, October 26-28, 2007. <http://www.delmarnorth.com/namaste/meetings/ACP2007.pdf>

3. [Ettus 2008] Ettus, Matt. "Advanced Communications Package", slides prepared for ACP 2008 Planning Meeting [cancelled]. Summer 2008. <http://www.delmarnorth.com/namaste/meetings/ACP2008PlanningMeeting.pdf>

4. [Clark 1988] Clark, D., *The design philosophy of the DARPA internet protocols*, ACM SIGCOMM Computer Communication Review, Volume 18 , Issue 4 (August 1988), Pages: 106 – 114

5. [Thompson 2008a] Thompson, Michelle. "Namaste: A Ground Station Project Supporting the Advanced Communications Package on Eagle and Intelsat", presented at Dayton Hamvention, Dayton, OH, May 16-18, 2008. Presentation and notes available at: <http://www.amsat.org/namaste/>.

6. [Thompson 2008b] Thompson, Michelle, Phil Karn, and Paul Williamson. *Namaste Link Analysis*, version 3.3. June 22, 2008. <http://www.delmarnorth.com/namaste/requirements/Namaste_Link_Analysis.pdf>