Applying the Principles of Data Communications to The Development of an Open and Universal IBOC Data Protocol

David P. Maxson
Broadcast Signal Lab
Cambridge, Massachusetts

Paul Signorelli
Impulse Radio
New York, New York

ABSTRACT

The imminent introduction of an In Band On Channel (IBOC) standard by the National Radio Systems Committee (NRSC) for digitally broadcasting program audio and related data in the AM and FM radio bands creates an opportunity for the broadcast industry to surpass competing technologies in the data-casting arena. The NRSC has convened a parallel process to that of its Digital Audio Broadcasting Subcommittee, which is administered by the new Digital Data Broadcasting Subcommittee. This new subcommittee will consider the potential for specifying and possibly adopting a standard for the transmission of all forms of data content over AM and FM IBOC signals. This paper addresses the characteristics of the technology and of the cooperating industries that drive the creation of a uniform data protocol to support many services. The fundamentals of protocol design are applied to the characteristics of the technology to create a framework under which the protocol should be designed. The paper concludes that a seminal protocol has already been written for the medium and it lays out the process by which that protocol can be utilized by the broadcast industry and the consumer electronics industry to provide maximum value to the public.

OUTLINE

1 - WHY DOES IBOC NEED A DATA PROTOCOL?

Interoperability Drives Penetration

Webster's dictionary defines a protocol to be "a set of conventions governing the treatment and especially the formatting of data in an electronic communications system". These conventions are established so that entities involved in manufacturing one or more components of the communication system can do so, without having to work directly with any other entity manufacturing another component of the communication system. These conventions do so by ensuring a component will function with corresponding components that follow the same protocol.

Probably the best and most common example of this in the modern age is the Internet. The Internet is a network that supports a multitude of communication systems. Its two most popular systems are undoubtedly the World Wide Web, and Internet e-mail. Both of these communication systems are based upon a series of interconnected protocols, often referred to as a protocol stack. These protocols allow manufacturers of Internet communications hardware and software: routers, switches, hubs, firewalls, web servers, e-mail servers, web browsers, e-mails clients, and content to do so independent of each other ensuring that their respective products will work together properly.

It is hard to argue with the recipe that has made the Internet and the World Wide Web so successful. It has completely changed the way people live their lives, and although it took a while to get started, when it hit its inflection point the impact was dramatic and irreversible. At the heart of the recipe is a series of free and open protocols, including the most well known of the Internet data protocols: HTTP and HTML. It is these protocols that define the end-user applications of the World Wide Web that people have come to know and love today.

Internet Marketplace Model

Consider this story. The Internet toiled for 20 some odd years as a network used by government agencies and educational institutions to share information. However, it lacked a uniform method for sharing this information. Then in 1989, the World Wide Web was conceived and created by Tim Berners-Lee, a computer specialist from the European Particle Physics Laboratory (CERN). Berners-Lee and his colleagues saw the need for "a collaborative knowledge-sharing tool" that could be used by scientists all over the world to share research. Since then it has grown into the web we know today under the guidance of the World Wide Web Consortium (W3C), a volunteer organization with the responsibility for developing and maintaining common standards.

Arguably the most significant development in the history of the web came in the 1990's with the development of HTTP and HTML and web browsers capable of transmitting and producing graphical images based upon these protocols. Starting with Mosaic and its evolution into Netscape's Navigator and Microsoft's Internet Explorer, these programs allowed users to easily browse information on the web. This led to applications such as Internet e-mail, search engines, webzines, and web portals. The result is a billion dollar industry with a multitude of companies large and small providing a piece of the puzzle. Whether producing browsers or content to fill those browsers, companies are free to make products based upon the open protocols of the World Wide Web. And despite Microsoft's perceived dominance in the browser market, free browsers like Mozilla continue to maintain a significant market share among web aficionados.

Even with the demise of Internet boomers like Pets.com, an abundance of companies were born during the Internet boom of the late 90's that have experienced undeniable success. Companies like Netscape, eBay, Yahoo, Amazon, Priceline, Orbitz, Monster, Etrade, Inktomi, Doubleclick, Register.com, and About.com have become household names and represent just a few in the plus column. Meanwhile, media conglomerates from NBC to the New York Times have been forced to rethink the way they distribute their content. Still further, technology giants like IBM and Microsoft have completely changed their business strategies around the World Wide Web and the sharing of information based upon free and open protocols.

IBOC Can Learn From Internet Success

So, how does the Internet story relate to IBOC? Sitting in their labs, in 1989, it would have been hard for Berners-Lee and his colleagues to imagine the impact their invention would have on the world. Moreover, had it not been for the fact that their invention was flexible, universal, and open for all to contribute to and benefit from, the web would probably still be a communications network used only by scientists and researchers, rather than the most prolific form of communication in our society today.

IBOC needs the kind of same flexible, open, and universal data protocol that the web has in HTTP and HTML if it is to 1) consider the needs of the entire industry; 2) grow to create rich consumer applications; 3) foster a competitive marketplace that drives innovation; and 4) become a true force in the communication age.

2 - WHY DOES IBOC NEED ITS OWN DATA PROTOCOL?

Distinction between the Web and IBOC

Taking as an axiom that IBOC needs protocols that play roles just like HTTP and HTML, then why not simply have IBOC use HTTP and HTML as its data protocols? After all, from a look at the proposed uses of IBOC such as sending text to a radio display or sending images to an LCD screen on a receiver, or even sending audio files to a receiver, it could be concluded that the same type of communication and functionality is taking place. The truth is that much of the proposed communication for IBOC is similar to things seen on the web. However, much of the communication is very different. And that is the problem.

Take, for example, one of the web's simplest functions: rendering an image file. The markup for such a function is as follows:

<img src="http://www.myurl.com/images/logo.gif" alt="myurl logo">

On the surface this seems like a simple operation. Render the logo image file. There are two problems that must be solved when porting this to IBOC however.

Bi-directional Web Protocols

The first problem resides in the fact that HTML (Hyper Text Markup Language) operates along with a bi-directional communications protocol for the web called HTTP (Hyper Text Transport Protocol). Working in tandem HTML represents a document protocol conveying formatting instructions to a web browser, and HTTP is a communications protocol that facilitates the transferring of web data including HTML documents. One rarely sees one protocol operate without the other. The notation *src = "http://www.myurl.com/images/logo.gif"* invokes the

HTTP protocol and asks the browser to call back to the server for the desired file "logo.gif". In stark contrast, IBOC is uni-directional. There are no return messages in the IBOC channel. Therefore, IBOC could not easily support HTTP if at all without completely altering the meaning behind it. At that point it is the same only in name. Also since, much of the functionality of HTML depends upon HTTP it would require significant alterations to it as well.

Protocols Rely on the Way the Medium Works

The second problem with the example and with HTML for IBOC in general can be seen with the designation *alt = "myurl logo"*. HTML was designed to be highly interactive. Users of the web are constantly clicking their mouse to navigate and move through content on the web. Almost any web-based application requires significant interaction on the part of the user. In the above example, the *alt* designation provides users with an alternative description for the image file. This serves as a backup in case the image file cannot be loaded. However, it also provides a description to the user as to the nature of the image if they place their mouse pointer on top of the image and hold it there. This operation is called a "roll-over". It goes to the very nature of the interaction of web applications, or PC applications in general.

IBOC users will very likely be traveling in cars at high speeds, at least initially. There will be very little use for a "roll-over" in these contexts. Yet, this interaction is among the simplest of interactions on the web. Factor in form fields, buttons, hot-links, scrolling, java plug-ins, and expandable and collapsible menus to name a few features. It becomes apparent there is a lot of interaction on the web.

It makes sense for the web. HTTP and HTML were initially designed to facilitate the sharing of documents over inter-connected bi-directional networks. As they evolved new types of interaction were added, special functions, such as security were implemented, and the scope of web applications blossomed. Fundamentally, the form and function of HTML and HTTP were designed to make use of the Internet. They work well with the type of communication seen on the Internet.

The Role of Multimedia in IBOC

Consider another Internet-oriented protocol, for example, the multimedia protocol called SMIL (Synchronized Multimedia Integration Language)[1]? At first glance SMIL offers some hope to those who would like to cut and paste from pre-existing protocols. SMIL is described as a simple protocol for authoring interactive audiovisual presentations. It is typically used for multimedia presentations that integrate streaming audio and video with images, text or any other media type. It was developed to provide a more efficient way for streaming multimedia presentations over the Internet than provided by MPEG (MPEG-1 at the time). SMIL allows a presentation to be divided into its individual components for transport and coordinates their activity via an XML based markup language. This results in lower bandwidth utilization when dealing with presentations that use the same multimedia objects over and over. It would not be helpful for streaming a movie where the multimedia objects are constantly different. In that case, MPEG is still preferred. The most common use of SMIL is in streaming media players such as QuickTime or Real Media Player.

Here is a SMIL example:

```
<smil>
   <head>
      <layout>
            <region id="smile" top="200" left="100" z-
         index="5" width="100" height="220" />
         </layout>
   </head>
   <body>
      <seq>
            <img id="smile"
         src="http://www.myurl.com/images/thing" />
         </seq>
   </body>
</smil>
```

This markup defines a region of the available screen and then defines a simple sequence that places an image in that region. *Img* with SMIL is similar to the *img* declaration that is seen in the HTML example above, however, SMIL adds some information to it. Again, the language relies upon the bi-directional communication protocol HTTP. The SMIL client needs to understand HTTP in order to render this code. So once again, porting SMIL to IBOC requires defining a lower level uni-directional protocol to replace the functionality of HTTP.

SMIL's similarity to HTML from the examples above begs the question, why did SMIL even need to be created? Why not simply modify HTML to include the functionality needed to make SMIL work? After all, they are both intended to render multimedia information over the web. SMIL simply intends to add some coordination to the multimedia objects.

---

1 http://www.w3.org/TR/REC-smil

The answer is two-fold. For one the applications they produce are quite different in their multimedia capabilities. Web applications tend to be static, and rely upon a great deal of user interaction to trigger events. While web applications do employ multimedia objects (images, movies, audio, etc.), the majority of the data that is rendered continues to be text. SMIL applications on the other hand, are rich with graphical and audio data and are often synchronized with audio or video streams. SMIL applications trigger events on their own without user interaction. The components of the protocols are the same, but their structure and rules are as different as the applications they produce.

Secondly, the act of changing HTML to be used in a different manner, in other words, changing the meaning behind its basic structures, creates a new protocol. The new protocol is now operating along a different evolutionary branch. The users of the original technology do not bear the fruits of the changes and enhancements of the new technology. This is purposeful. It cannot be expected for HTML designers to have to accommodate the whims of SMIL designers and vice-versa. Moreover, it would have been risky to do this since there was an abundance of web content already published when SMIL was invented. SMIL was intended to solve a different problem, despite its similarities to HTML.

The fact is that protocols like HTTP, HTML, and SMIL have a lot to offer IBOC and the IBOC data protocol should look to incorporate aspects of them whenever possible. Doing so can add efficiency to the deployment of IBOC applications and content. However, in and of themselves, these protocols do not represent a data protocol for IBOC and the process of altering them for use in an IBOC system inherently creates a brand new protocol. Moreover, there are many proposed IBOC applications that are completely outside of the scope of what is offered by these Internet protocols. As will be seen later, these protocols become part of a universal IBOC data protocol and not the other way around.

Finally, IBOC imposes its own constraints upon any protocol it will support. The impact of this will be discussed further in section 4. These constraints require IBOC have its own data protocol in the long run even if it borrows from existing protocols. Consider the uniqueness of the IBOC communication system: One-way transmission, point-to-multipoint applications, lightweight devices, limited data bandwidth, high degree of users in a mobile environment, integration with two-way networks at the device, and support for a legacy system to name just a few.

3 - WHY DOES THAT PROTOCOL NEED TO BE STANDARDIZED?

The Role of a Standard

As discussed earlier, a protocol requires that all parties adhere to it for it to be effective. There are two paths that markets take to ensure this—market competition and consensus standardization.

Market competition generally leads to de-facto standards unless the technology is adopted as a formal standard later on. Probably the most well known de-facto standard in technology today is Microsoft’s Windows operating system, which is used by the majority of PC manufacturers and application developers. It is important to note that although Windows is a de-facto standard in the PC industry, companies are free to implement and develop for other PC operating systems (e.g., Apple OS, LINUX, UNIX, etc.) and still do in great numbers.

Many people feel that de-facto standardization is favorable to consensus standardization because of the efficiency of markets. In many cases that is true. Perhaps the PC operating system example makes this point. However, in the establishment of protocols for communication systems, where there are many varied interests, market forces tend to be inefficient and consensus standards are required in order to ensure that all of the interests are considered.

Consider the statement that David A. Balto, an Assistant Director of the Federal Trade Commission, made in a speech on anti-trust law in February of 2001. “Standard setting plays a critical role in network industries because of two factors: consumer expectations and interoperability. Consumer expectations are critical to the success of networks, either existing or emerging ones. The strength of a network’s market power depends upon its users' expectations of the likely behavior of other users of the network. Consumers fear making investments in a network and then becoming "stranded" because there is insufficient consumer acceptance. Standards may alleviate those concerns, by assuring consumers that the network technology will be adopted.”

“Interoperability - the capacity of products of one vendor to communicate or interface with the products of competing suppliers of complementary products -- also plays a critical role in the success of networks. No network is an island, and networks must depend on alliances with producers of complementary products. Interoperability is a core function of most information technology products. Network products such as modems and cellular phones are heavily dependent on interoperability standards. Interoperability standards also play a critical role in overcoming the concerns of stranding and the expectations of those which produce complementary products"[2]

However, Mr. Balto goes on to stress that markets engaged in standardization must find "common ground" within the standards framework to create competition in the market and foster innovation. Whereas communication system protocols necessitate consensus standard setting, that is not a license to create monopolies. Mr. Balto's emphasis on the virtues of competition and interoperability support the position that consensus standards in communications systems should prevent monopolies.

Standard Setting Process

In general, a standard can be said to be a widely accepted and understood practice. By practice, a standard is a system or method that has been sufficiently documented so that it can be replicated from the documentation. Creation of a consensus standard generally involves a "standard setting process" in which the standard is created and agreed upon by all interested parties. A group consisting of these interested parties assembles and runs the standard setting process. There are well known organizations throughout the world that work to establish standards, the IEEE, ANSI, and ISO are just a few.

The IEEE identifies five "imperative" principles of consensus standard setting: Due Process, Consensus, Openness, Balance, and Right of Appeal[3]. Due process involves having well-established publicly declared procedures for establishing the standard and adhering to those procedures in the process of setting the standard. Consensus means that a majority settles all issues that arise during the course of the standard setting and a majority approves the final adoption of the standard. Openness involves ensuring everyone has access to the process. To ensure access all interested and affected parties must be allowed to participate and the results of the deliberations must be made public. Balance ensures that the participants of the standard setting process represent a diverse group. Finally, right of appeal ensures the procedures of the standard setting process provide for the realistic and impartial handling of procedural complaints and technical issues.

The American National Standards Institute (ANSI) adopts these same principles, as part of their guidelines. They also add an additional principle that is worth mentioning: Lack of Dominance. According to ANSI, "The standards development process shall not be dominated by any single interest category, individual or organization. Dominance means a position or exercise of dominant authority, leadership, or influence by reason of superior leverage, strength, or representation to the exclusion of fair and equitable consideration of other viewpoints"[4].

Lack of Dominance goes beyond balance, which really speaks to representation. It goes to the heart of standard setting, that the interests of all stakeholders be considered in the development of the standard. No single entity should dominate.

It is important to note that standards, unlike many other types of written guidelines, have been shown to have legal force and effect. Standards can be used as evidence in courts of law and can also become legal requirements when adopted by governments and by regulatory agencies. A standard can carry far more weight, and the process by which they are developed falls under more scrutiny than other documents and papers.

An Open System Standard

A standard setting process, as described above, generally yields a standard that describes an open system. An open system should describe all required information to duplicate the system, but does not need to provide implementation details, and can even contain patented material. Patents and standards are related in the fact that both strive to describe a system such that a person "skilled in the art" can duplicate the system without undue experimentation. When patents are incorporated into of standards it is typical for the holder of the patent to agree to license the patent on fair and reasonable terms on a non-discriminatory basis to anyone who implements the standard.

According to the International Standards Organization (ISO), "Standards are documented agreements containing technical specifications or other precise criteria to be used consistently as rules, guidelines, or definitions of characteristics, to ensure that materials, products, processes and services are fit for their purpose."[5] In relation to communications networks ISO states "The computer industry offers a good example of technology that needs quickly and progressively to be standardized at a global level. Full compatibility among

---

[2] http://www.ftc.gov/speeches/other/standardsetting.htm
[3] http://www.ieee.org

[4] ANSI Procedures for the Development and Coordination of American National Standards, January 2002 Edition
[5] http://www.iso.org/iso/en/aboutiso/introduction/index.html

open systems fosters healthy competition among producers, and offers real options to users since it is a powerful catalyst for innovation, improved productivity and cost-cutting."6

The NRSC describes a standard as "A document that establishes engineering and technical requirements for processes, procedures, practices and methods". Section 8 of the NRSC procedures manual on standards development goes beyond mere reference to the ANSI process stated above; it calls for its process to be "the same"[7]. There is a clear precedent for the establishment of an open process, such as that employed by ANSI, the IEEE, and ISO, which lead to a standard that describes an open system.

The reason for this precedent goes beyond the technical and the legal. It makes good business sense because it creates an open market that drives innovation. To some this seems contradictory because a standard will artificially favor one solution over the other. However, once that solution is established, the standard creates a level playing field upon which all may compete. This is of particular importance to communication systems where there are so many varied entities involved in implementation.

Consider the history of the web versus closed on-line service providers such as AOL, Prodigy, and CompuServe. The web has thrived, launching commercial enterprises at an unprecedented rate. In the mean time, Prodigy and CompuServe have disappeared, and although AOL has survived and done well despite recent woes, they have done so in many minds because they have embraced the Internet. AOL's killer applications continue to be e-mail and instant messaging because they recognized there was more value in being part of an open communication system than a closed proprietary one. This strategy becomes even more important to them as "always on" broadband connections erode their dial-up business.

In the 1990's IBM recognized this and threw their weight behind open platforms. They abandoned trying to compete in the world of proprietary standards and focused on promoting open system standards and then built a service business behind it. This has lead to yet another resurgence of big blue. Even Microsoft has gotten the picture. Despite their dominance in the PC operating system market they have developed a strategy for their products based upon XML, an open data protocol for transferring data on the Internet, fully recognizing the inevitability of such a move, and its importance to consumer and business applications.

The IBOC Data Standard

The IBOC data standard should be nothing less than an open system standard. The system described should be an open system. Included patented material should be sufficiently disclosed and licensed on fair, reasonable, and non-discriminatory terms. Trade secret material that is essential to the integrity of the standard should be revealed. And those "skilled in the art" should be able to reproduce the system with reasonable efforts. Anything short of this will stifle innovation in the short run and succumb to outside competition in the long run. AOL, Prodigy, and CompuServe stand as examples of the limitations of closed networks. Open system standards-based wireless networks emerge and drive competition among innovators. As history has shown, innovators migrate to the open system.

4 - HOW IS THE PROTOCOL CREATED?

The discussion up to this point has focused upon the need for an open, flexible, IBOC data protocol, that is developed as part of a consensus standard setting process. Attention now turns to the all-important work of developing the protocol. The standard setting process calls for establishing the requirements for implementing IBOC data services. These requirements will be encapsulated in the protocol, so the question must be answered, "What are the requirements of an IBOC data protocol?"

4.1 - UNDERSTAND THE REAL WORLD PROBLEM TO BE SOLVED

To understand the requirements, the "real world" problem must be articulated. The term real world is used because at this point the requirements must be described in terms of how the world will use the protocol. In this case the world is comprised of content producers, data service providers, broadcasters, radio listeners, other wireless device users perhaps, and even other networks (e.g., the Internet). Technical terms are not used to describe the problem at this point.

Data Protocol Rationale

Perhaps the problem can be framed as such:

"Radio is undergoing a transition from a primarily analog medium to a system that supports a robust digital signal. The original purpose of the digital signal is to improve consumer experience of the radio program

6 http://www.iso.org/iso/en/aboutiso/introduction/whyneeded.html

7 http://www.nab.org/scitech/Nrscgeneral/NRSCproceduresmanual.pdf

by digitally reproducing traditional analog broadcasts. However, a significant benefit to this transition is the ability to send broadcast data.

"Wireless data services in other media are taking root and offering consumers a host of services from e-mail to streaming media. Some of these services are subscription based and some of these are free. Some are suited for stationary wireless devices, while still others are ideal for mobile environments. The radio broadcasting industry is looking towards its emerging digital format as an opportunity to compete in this wireless data market operating as both a standalone wireless network and a backbone to other networks.

"The wireless applications envisioned by the industry include services that are highly complementary with existing radio services and others that operate completely independently of existing radio services. Some look to take advantage of the creativity and mass appeal of radio broadcasting in terms of its local content and informative programming. Other services are only looking at IBOC as a low-cost carrier of data. A diverse sampling of potential "use cases" for IBOC is given in appendix A."

The problem, then, is to define a means for maximizing the utilization of the IBOC spectrum, thus inuring to the public benefit. A common protocol, shared by all stakeholders (with suitable protections for private data) must address the diverse needs and uses in a streamlined and standard fashion.

Protocol as a Network Enabler

The attached use cases give a bigger picture of the types of applications that IBOC will enable. The potential of IBOC is expanded significantly when it is combined with other communications networks. Therefore, IBOC data service protocol must be flexible enough to support other existing protocols, as well as add new features as the use of the network expands.

In the Internet example, the original document sharing protocols were all text based; it was only later that multimedia data was added. Even when modern day HTML was introduced, the original versions lacked support for elements such as style sheets, frames, Java applets, and streaming audio and video. All these elements are commonplace on web applications now and were added later without interrupting existing services. Furthermore, their addition launched a whole new array of services, all under the umbrella of the same set of protocols.

4.2 - UNDERSTAND THE CONSTRAINTS IMPOSED BY THE TECHNOLOGY

Once the problem can be described, it is important to add a level of refinement by folding in the constraints that are to be placed upon any solution. For IBOC there are two sets of constraints: The constraints of the transmission system (namely IBOC itself), and the constraints of the device and the device environment.

IBOC Constraints

At the time of writing this paper, the inner workings of FM IBOC are being carried through a standards setting process with the NRSC. AM standardization is to follow. The FM IBOC signal inherently has more bandwidth than the AM IBOC signal due to its greater channel bandwidth. This paper focuses on the structure of the FM IBOC signal, leaving assessment of the unique characteristics of the AM IBOC channel to subsequent analysis.

As a design problem, the IBOC concept is particularly challenging. In 1997 Dr. Brian Kroeger laid out some of the design constraints confronting the radio frequency design of IBOC.[8] The challenge was to occupy the existing broadcast spectrum with new digital signals that would be at once compatible with existing receivers, yet still capable of transmitting sufficient digital bandwidth to support a credible service.

This result of years of development takes the form of sets of OFDM (Orthogonal Frequency Division Multiplexing) carriers straddling the occupied bandwidth of the FM analog signal (with a similar solution designed specifically for AM). These OFDM carriers are modulated with one symbol per carrier, two bits per symbol, 344.5 symbols per second. Every 19th carrier is a reference carrier for synchronization and fast identification of the mode the entire IBOC transmission is operating in.

Modes of Operation

There are several operating modes because there are several ways to configure the transmission. From a radio frequency perspective, there are three basic modes of operation, hybrid, extended hybrid, and all digital.

The hybrid modes insert OFDM carriers on each side of the existing analog signal. Hybrid mode is the least demanding on compatibility with analog reception, placing 190 carriers on each side of the analog signal in the range from 129,361 Hz to 198,402 Hz above and

[8] B. Kroeger, P. Peyla, "Robust IBOC DAB AM and FM Technology for Digital Audio Broadcasting," Proceedings of the 51st Annual Broadcast Engineering Conference (NAB), April 1997

below the station's center frequency. These carriers are the "primary main" carriers and are divided into ten "partitions" of 19 carriers, including one reference carrier per partition. (see chart below)

In extended hybrid mode, one, two or four blocks of 19 additional carriers are inserted closer to each side of the analog signal. These are the blocks of "primary extended" carriers. These carriers are more likely to be compromised by energy in the analog FM carrier than the primary carriers are. Consequently it behooves the system design to segregate these channels for different levels of error correction and for information with different degrees of importance to the broadcast.

variations of logical channel configurations to support different modes of the audio codec. These modes give the broadcaster ways to provide various degrees of audio quality and robustness to the listener.

Service Modes

Initially seven service modes (MP1-MP7) are defined for hybrid and extended hybrid operation, with backward compatible extensibility to 56 additional modes yet to be defined. These modes map certain logical channels to certain groups of RF carriers. The groups of primary RF carriers, shown in Figure 1 below, are 10 primary main partitions or 1, 2 or 4 primary extended partitions. In some cases, logical channels will share the same OFDM partitions in the spectrum by being interleaved on them over time.

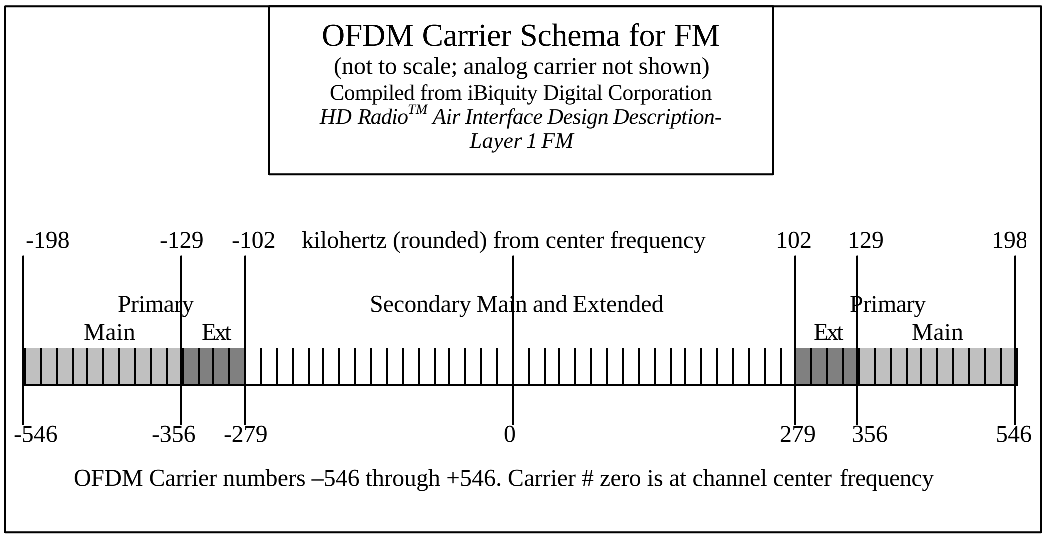

Figure 1

All-digital mode replaces the analog carrier with a low level digital signal comprised of the "secondary main" and "secondary extended" carriers. This increases data throughput, but places demands on the system to provide a fast access digital audio signal to compensate for the loss of the instantaneously available analog signal. As all-digital operation will likely take many years to be widely implemented, it is overlooked for the purposes of this discussion. However, the data protocol design should anticipate future extension to the all-digital mode.

The other factor that determines the mode the station is operating in relates to the configuration of several "logical channels" carried over the OFDM carriers. Since the broadcast primarily exists to provide a reliable digital audio program (called the Main Program Service in the proposed standard), there are numerous

Logical Channels

Logical channels are labeled by letter designations: P1-P3 and S1-S5. The logical P-channels are carried on either the primary main or primary extended hybrid OFDM carriers. The logical S-channels are carried on the secondary OFDM carriers.[9]

[9] There are some other logical channels that offer critical services to IBOC. One is the PIDS logical channel that provides station information on the primary carriers with a lower degree of robustness than the Main Program Service audio, but with a high degree of repetition to provide useful information about the station. The SIDS logical channel provides the same information on the secondary carriers when in all-digital mode.

Channel Characteristics

As the OFDM carrier configuration was selected to optimize performance in a potentially noisy and multipath prone environment, so, too, the preprocessing of the data stream(s) is designed for the communications channel. IBOC, like all communications channel designs, must balance bandwidth, latency, and robustness in its design constraints.

Logical Channels of the Primary Service Modes: Frame Sizes Prior to scrambling, channel encoding and interleaving. Compiled from iBiquity Digital Corporation *HD Radio™ Air Interface Design Description- Layer 1 FM*									
	P1			P2			P3		
Service Mode	Logical Frame size (bits)	No. of Logical Frames per transmitted frame	Assigned to OFDM carrier partitions	Logical Frame size (bits)	No. of Logical Frames per transmitted frame	Assigned to OFDM carrier partitions	Logical Frame size (bits)	No. of Logical Frames per transmitted frame	Assigned to OFDM carrier partitions
MP1	146176	1	PM						
MP2	146176	1	PM				2304	8	PE 1
MP3	146176	1	PM				4608	8	PE 1,2
MP4	146176	1	PM				9216	8	PE 1-4
MP5	4608	8	PM, PE 3,4	109312	1	PM	4608	8	PE 1,2
MP6	9216	8	PM, PE 1-4	72448	1	PM			
MP7	4608	8		146176	1		4608	8	

The transmitted frame is the 1.5-second long frame transmitted by the OFDM partitions. Each OFDM carrier transmits 512 symbols (1024 bits) during the 1.5-second frame. Some logical channels use smaller logical frames that populate the 1.5 second transmitted frame using 16 discrete blocks or 8 block pairs.

PM = Primary Main partitions (10 groups of 19 carriers above and the same below the analog carrier.)
PE = Primary Extended partitions (4 groups of 19 carriers, numbered 1-4, appearing both above and below the center frequency)
The frame sizes described here are raw data capacities that do not account for error protection overhead. See table below for examples of actual throughput data rates.

Table 1

The IBOC data processing in IBOC Layer 1 accomplishes this by establishing a defined throughput for each logical channel, depending on the mode the station is in. This definition assigns both a bandwidth (the frame sizes in Table 1 below) and a robustness (an assigned degree of error protection overhead from convolutional coding).

To minimize errors caused by instantaneous bursts of noise, the logical channels are each interleaved in time across a long frame of data, or within blocks in each frame. This frame is fixed at a 1.5 second duration (rounded figure). Because of this inherent property of the FM IBOC channel, data will have fixed minimum latencies and it will share the robustness of the logical channel to which it is assigned for a given service mode.

Data can be assigned to an unused logical channel if one is available in the selected station mode. Data also will be able to share space with digital audio on a logical channel carrying audio on a "space-available" basis. That is, when there are instantaneously low processing demands on the codec, the size of data packet from the codec will shrink, leaving what is called "opportunistic" bandwidth. The station will also be able to reduce the bandwidth of the codec in 8 kHz steps (from 96 to 88 to 80, etc) to make room for additional data at some trade-off with audio quality. These fixed increments of recovered bandwidth are called "fixed" bandwidth.

Broadcast data, then, can be transmitted in logical channels that contain digital audio taking up fixed and opportunistic bandwidth, and on logical channels not carrying digital audio. This complexity created by the combination of logical channels, degrees of robustness, fixed latency, varying throughput with opportunistic data, and available bandwidth must be considered when developing a data protocol for IBOC.

The raw digital bandwidth of the hybrid and hybrid extended channel is carried on 504 OFDM carriers (180 primary main carriers plus 72 primary extended subcarriers, times two, for upper and lower sidebands). Each carrier operates at ~344.5 symbols per second times two bits per symbol, yielding a raw throughput of

~347 kbps. Below Table 2 depicts reported data throughput rates (“payload”) after accounting for error protection overhead.

The premise of a data protocol design is based on the same fundamental precepts of RF waveform design and digital framing and error protection design. All are engineered specifically for the characteristics of the communications channel on which they operate.

Receiver Constraints

There are two primary constraints that govern receivers. The first constraint deals with the fact that receivers cannot ask the sending component for any information. It is incumbent upon the sender of the information to provide the receiver with all of the information required to render a service. This is significantly different from operations on the web where there is a back-and-forth communication between the sender of the service and the receiver of the service. In fact, in most web services, the receiver initiates the transmission, not the sender. In radio broadcasting, it is completely opposite. Therefore, if a receiver tunes to a channel with multiple data services, it will need a mechanism for deciding which service to render and when. That mechanism could involve user interaction.

	Primary Service Mode Effective Data Throughput Rates (kbps) Compiled from iBiquity Digital Corporation *HD Radio™ Air Interface Design Description- Layer 1 FM*						
Service Mode	MP1	MP2	MP3	MP4	MP5	MP6	MP7
Logical Channel							
P1	98	98	98	98	25	50	25
P2					74	49	98
P3		12	25	50	25		25
All modes but MP1 employ 1,2 or 4 extended hybrid partitions for additional capacity							

Table 2

The second constraint imposed by receivers is that they contain a diversity of HMI (human-machine interface) configurations. This affects the way users can interact with a data service, and the types of services receivers can support.

For example, some receivers have robust text displays and some have simple text displays. Others may have full graphics displays or no display. Each of these receivers can support different classifications of services, and these classifications overlap. For instance, almost any receiver can support a service providing short text information about the audio portion of the broadcast, as long as it has some display. However, only a graphics capable receiver could render a service that relied upon image data. And only a receiver with programmable buttons can prompt the user for input to the service.

4.3 - DESCRIBE THE PROTOCOL REQUIREMENTS

An understanding of the characteristics of the communications channel, the receiver environment, and the user requirements, as briefly explored above, will inform the development of an appropriate data protocol.

Conclusions

A single object[10] from a data service will be larger than the space in a single frame on a single logical channel in many cases. The lower level protocols will be forced to break up these objects into smaller pieces. This can lead to less reliability in performance for the service because the transport time has increased, thus increasing the chance for errors. Services consisting of many large objects working in unison (e.g. an HTML page strewn with image files) will only exacerbate this situation.

Inversely, a single frame can carry data for more than one data service. This means the channel can have more than one object from a data service in transport at a given time, meaning the channel can carry multiple data services. It also means, the bandwidth available in many circumstances for a single data service will be limited, and sometimes prone to error.

Receivers are passive devices, in terms of the communication channel, and wait for data to be received and then react to it. They have no ability to ask for data retransmission. They have no direct control over how many services they will receive. They are “sessionless,” in that they have no actual connection to the sending device.

Conversely, the sender of the service has no information regarding the specific receivers getting the data service. Although the Internet serves millions of

[10] In this section elements of data services will be referred to as data objects, data service objects, or simply objects. Objects in this context can be thought to be files such as audio files, video files, text files, or even markup files or executable software that are part of a data service. Data services consist of a set of one or more data objects operating in unison.

web users, all connections are one-to-one in nature. The sender and the receiver (roles the browser and the web server play) communicate directly with one another, and each is aware of the other. With IBOC data services, the server needs a way to address the receiver without a direct connection.

Requirements

At the highest level, the data protocol can be divided into two sets of protocols: channel related protocols and service related protocol. Channel related protocol requirements arise out of the characteristics of the channel described above. Service related protocols arise out of the performance requirements of the services as defined by the use cases. However, service related protocols are not immune to channel characteristics.

Channel Requirements

The channel characteristics point to the need for the protocol to allow for the inclusion of error detection, if not error correction information, at the object level. Individual objects need to have a unique ID in the system so they can be properly, disassembled, reassembled, and retransmitted.

Furthermore, if the channel can support objects from more than one data service, and data services can consist of more than one object then the protocol must identify the service that an object belongs to as well. It also stands to reason that if several objects depend upon one another (like the HTML page with the image file example) in order to operate properly this would also need to be indicated by the protocol. In other words, objects may need to be grouped together.

On the web HTML calls back to the server for the image files referenced by a web page (using HTTP) so they arrive at different times in several communications, even if the user sees it as the same action. If the image file has errors, it can be re-sent at the request of the user. The result is that most of the time a web page will load properly even if there are errors in some of the communication, and this is hidden from the user. Only on very slow connections will images appear to load significantly after the main page has loaded. What is more is that the browser knows that there have been errors and can ask for the data again. In other words, it knows it will get good data and can keep the session alive.

However, in a one-way medium that has no sessions, the device has no real way of knowing when the image file that is missing will arrive, or if it will arrive at all. The receiver can use grouping information as well as timeout information assigned to the objects or the services to predict that the file is lost.

As with the web, it is likely that IBOC data service objects will reference one another. The web uses the HTTP protocol to point from one object to another. Recall, however that HTTP uses two-way communication to ask for and send objects. The IBOC data protocol needs a one-way version of this where an object can point to other objects in the data stream or the receiver memory.

Finally, IBOC communications are not uniform. In other words, there is no discrete beginning and end to a communication session. It might be said that communications are intermittent. This differs from the web model where there is a discrete beginning, middle, and end to a communication. With the web, a user requests a web page, a session is initiated between the server and the browser, the page is transmitted to the browser, all referenced files of that page are requested by the browser, and subsequently transmitted, the browser acknowledges reception, the session ends, the page has been loaded.

Consider the person traveling in a car, turning on the radio and tuning a station in the middle of a song. From the perspective of the broadcaster that song and any data related to it has a discrete beginning and end. To the person tuning in during the middle of the song, the experience will be different than for one already tuned in. Data may take time to accumulate and some may be missing. Consequently, IBOC data service objects (or groups of related objects) must have the ability to operate independently to provide a satisfactory experience in the absence of all the data intended for the receiver. It is similar to streaming, except with Internet streaming there is still a session with a discrete beginning and end, and data can be retransmitted at the request of the client.

This means that the IBOC data protocol must repeatedly send service level data that is needed by the object or service level data must be made part of the object data itself. This would include information like the ID of the service and the classification of receiver required in order to render it.

While this example is given for a broadcast service related to the radio programming, the same issues arise with the delivery of independent data services. The creation of a uniform protocol for handling such issues will streamline communications and ensure interoperability while still enabling various services to differentiate themselves.

This section illustrated some types of requirements that emerge from the communication channel properties. The next section does the same for service-related requirements.

Service Requirements

The desired services also inform the creation of protocols. These protocols would allow broadcasters to implement the functionality of various data services. The data services from the use cases can be divided into two distinct groups. The first group represents multimedia applications and the second group represents network applications.

Multimedia Services

Multimedia services can be categorized as services where the receiver is viewed as an entertainment or information reception device, and the user is the desired audience. These services include simple program associated data services such as supplying information about the audio stream (song title, artist name, artist image, advertiser phone number, etc.) to data services that are independent multimedia streams, such as secondary audio services, or news ticker services.

The service provider intends to control the timing and the flow of the information even if there is user interaction involved. Consider for example the field of video game authoring in which the producer creates an entire user experience, yet the user has tremendous flexibility to play within the environment of the game. So, too, broadcasters and other content suppliers will seek general control over and predictability of the IBOC user experience while giving users freedom to interact in their own ways. These services are similar to other Internet and wireless services because they all deliver multimedia content: text, hypertext, image, video, and audio data, etc. While "pushed" services are an option in other media, they are a necessity in the broadcast medium.

It is important to note that the early devices rendering this data will not have the capability to be upgraded. Therefore, formats for multimedia data types must be well established by the protocol. With the web, where the receiving devices (i.e., PC's running browsers) can easily add new software components capable of rendering new multimedia types, IBOC devices will not have this capability for quite some time. Consequently, if new formats or features are to be offered, content providers and receiver manufacturers must work in tandem to get them into the hands of the public.

Like other types of multimedia services, IBOC multimedia services can have menu capability, allowing users to select different options for the service that lead to different data being presented. More advanced services can have filtering capability that gives the user the impression that the service is geared towards them specifically. Filtering in a one-way medium would tend to do so based upon categorization of content. Filtering occurs when the user or the service itself makes a determination to focus on a particular category for presentation purposes. An obvious example would be geographical or time categorized data.

Of course, all of this functionality indicates that the IBOC data protocol must have structures to communicate to the receiving device the purpose and function of the content.

Network Services

Network services are services that simply view the IBOC channel as an efficient way of moving data. The role of the IBOC receiving device is as a "gateway" to another network. A commonly referenced example for this type of application is the fleet service, where the IBOC channel is used to transfer schedule information to a fleet of delivery trucks. Another example is the "IBOC cradle application", where a handheld device is placed in a cradle and overnight data, perhaps a newspaper, is sent to it for use the next day.

These types of services are relevant because of the low cost of sending data to a large number of devices over IBOC compared to other wireless media. Because of this, the IBOC data protocol should be able to carry these services to their host applications, but do not need to understand the functionality of the host applications.

They are relevant to the IBOC data protocol because 1) They share the same pipe as other IBOC services; 2) They share similar requirements at the channel level. For instance, these applications need a way to be identified on the channel. They must identify the type of information being carried. They may have to broken up and reassembled by the lower level protocols, etc.

Furthermore, they represent a significant piece of IBOC data service potential. They should be addressed at the outset, governed cooperatively by the standard, and not taken on ad-hoc. Potential data providers will be attracted to IBOC as a channel for their data when it offers the assurances that Mr. Balto says a standard provides: competition, interoperability, and no stranding.

4.4 - DESIGN A PROTOCOL

The final steps of the protocol development will be to design a protocol from the requirements stated above.[11] The extensible data service ("XDS") protocol is submitted as an exemplary protocol for use in the IBOC data protocol. The protocol was developed from extensive discussion with the stakeholders, extensive testing in broadcast and simulated environments, using existing receiver hardware, and through careful consideration of the requirements. XDS can serve as the basis for the ultimately adopted protocol.[12]

XDS has the following components:

1. A byte structure for data service object information
2. A byte structure for conveying service and object level identifying information
3. A byte structure for conveying the multimedia or service object
4. A markup language for implementing specific multimedia functions

A typical XDS object is a byte array with the following sections:

Object Header	Service Data

<u>Object Header</u>

The object header contains the identifying information for the service and the object. The service data section holds the service object including multimedia data or application specific data.

The general operation of the device is to analyze the header, make determination as to the nature of the service, its relation to other objects, and its overall requirements and then pass the service data to the appropriate module for execution.

The header fields for an XDS object include the following object and service information:

- Session ID – Identifies the service on the channel
- Session Name – A user viewable name for the service
- Content Type – Identifies the application that the service data is targeted for
- Expiration – Identifies the expiration of the data object or the service
- Error detection – Provides a mechanism for the receiver to detect if there was a transmission error in the object
- Group ID– Provides a mechanism for grouping related objects together
- Interactivity Requirements – The required functionality and applications that must be supported by the receiver to properly render the service
- Synchronization Information – Timing information for synchronizing the beginning of a data object with the audio stream or with time
- Namespace – The "domain" that the service belongs to especially for services broadcast across different channels

Some of the fields of information are mandatory and part of every object. However, in order to preserve bandwidth other fields are left as optional and transmitted only if their non-default values are needed.

Service Data

The service object is intended to be an arbitrary byte string so that as long as the service object can be represented in byte form, any multimedia objects including markup languages like SMIL and HTML can be carried to the receiver. The object header would have to indicate the format of the content so that the receiver could properly interpret the data, and there is a header field to indicate this.

For example, suppose the XDS object is carrying an image file in JPEG format. Image data like GIF and JPEG can be represented in a byte format. The header of the XDS object, in particular, the content-type field would notify the receiver that the file is a JPEG and the receiver would know how to decode and render the image. This assumes that the receiver has the capability to decode and render JPEG files.

This brings up an impotrtant point. The receiver has been sent an image file. How is it to know whether to hold this file in memory for use with another object, or to render it immediately, and what if another data service wants to render something simultaneously on the screen?

Recall the HTML and SMIL discussion in Section 2 above. HTML files and SMIL files make references to

[11] The requirements section is only intended to be a representative sampling of the requirements and do not necessarily represent all of the requirements of the protocol.

[12] A full description of the XDS protocol can be downloaded at http://www.impulseradio.com/standards.

image files. The files are not "embedded" in the markup data. For use in a one-way medium, it was concluded that these files would need to be sent to the receiver and then referenced in the stream or in the receiver's memory. As a result, it will probably not be uncommon for a receiver to get an image file that it should do nothing with until another object references it (or it expires).

Answering these questions, and sorting out these issues, goes to the heart of the purpose of the data protocol. A major role of that protocol is giving data service providers the ability to send commands to the receiver along with the data. Following along with the web, XDS contains a markup language for the conveyance of these commands.

XDS Markup Language

The XDS markup language is based upon the extensible markup language known as XML[13]. XML documents are made up of storage units called "entities", which contain either parsed or unparsed data. Parsed data is made up of characters, some of which form character data, and some of which form markup. Markup encodes a description of the document's storage layout and logical structure. XML provides a mechanism to impose constraints on the storage layout and logical structure. SMIL is another example of an XML based language, and although HTML is not based upon XML, it shares most of the same properties and can be represented in XML with only minor syntactical changes.

The value of XML based instruction sets is that they inherit the design characteristics of the underlying architecture, namely XML and SGML. This condition provides an instruction set architecture that offers the following:

- Is human-legible and reasonably clear.
- Is formal and concise.
- Is easy to create.
- Supports a wide variety of applications.
- Is easy to write programs which process the data.
- Separates data from processing specifications.
- Has system independence and portability of data.

[13] For more information on the XML specification please refer to the Extensible Markup Language (XML) 1.0 specification as defined by the W3C. The specification is available at http://www.w3.org.

Because of this, XML has become a common architecture for moving data around communication networks like the Internet.

The role of the XDS markup language is to provide the receiver with syntactical information regarding the data. In other words, it tells the receiver "what to do" with the data, or better yet, what "it wants" the receiver to do.

At its core, the XDS markup has six categories of instructions, or "tags" as they are referred to in XML nomenclature. However, one of the advantages of basing the XDS markup protocol upon the XML protocol is that XML provides a seamless method for introducing new tags going forward.

The XDS tag categories are:

1. Service Identification
2. Standard Interactivity
3. Display
4. User Navigation
5. Storage
6. Audio Record/Playback

The service identifier information will identify services, and create definitions for the services that define the scope of the service and the relationship between services. A receiver uses the service identifier information to control the interaction of multiple services on the channel. The standard interactivity tags allow service providers to gear a single service to receivers with varying degrees of capability. The display tags inform the receiver that the data encapsulated by the tags is intended for the display (and to what region of the screen for receivers with LCD screens). The user navigation tags allow the service provider to provide the receiver with menu driven data. Storage tags allow the broadcaster to place objects in memory and reference them later on. And finally, audio record and playback allow the broadcaster to add additional audio elements to the broadcast or datacast.

Here are some examples of XDS markup:

```
<display>
    <data>WZZZ Radio</data>
<display>
```

```
<display>
    <data>
        <img drc="getResource://WZZZ.jpg"/>
    </data>
<display>
```

The first piece of markup asks the receiver to send the message "WZZZ Radio" to the receiver's display. This type of markup would work on any receiver with a display capability. The second piece of markup is asking the receiver to display the file "WZZZ.jpg" on the screen. This object would need a receiver with a graphics display.

The advantage of using XML as the basis for the markup language is that a receiver that did not support the "img" tag, namely a receiver without a graphical display, would ignore the "img" tag. In fact, if a broadcaster transmitted the above examples with XDS, each as one object in the order they are given above, it would work on all receivers. The text display receivers would render the first display tag and throw out the second display tag, leaving the message "WZZZ Radio" displayed. Graphical receivers would execute the first display tag and then execute the display tag with the image tag, leaving the display with the image loaded on the screen. However, as one might expect with a markup language, with XDS markup there are more elegant ways of handing such information. For instance, text and graphics can be integrated in receivers so equipped and given combined text and graphics instructions. Other functionality, such as interactive input, audio store and playback, geographic location calls, and the like are readily integrated into receiver design and the markup language as needed.

Use of other Markup Languages

For key presentation elements, XDS has sought to rely upon existing markup language definitions, rather than re-invent the wheel. In particular, the display tags rely heavily upon HTML for the formatting data for LCD screens. The supported HTML tags are listed by the specification, and several of the tags have been modified so that they can be supported by a one-way medium. In particular, these are tags where they utilize HTTP to reference other files, like images.

XDS allows for two ways to support the referencing of one object from another object, as is the case with the HTML tag "img". The first method is by allowing the service provider to embed the file directly in the XDS markup. The second is through the creation of a one-way HTTP-like protocol called "getResource" that instructs the receiver to look for the file in the stream or the receiver's memory.

Advantages of XDS

A major concept governing the design of XDS objects was to keep objects small and as simple as possible. This was done for two reasons. The first reason is that with IBOC, at least initially, there will be limited bandwidth for data services The second reason is that the more efficient the protocol can be with bandwidth, the more data services the channel can support and the more valuable it is as a communication medium.

The getResource protocol is a good example of XDS bandwidth efficiency. The getResource protocol allows larger files to be sent ahead of time at a slower rate, stored in memory, and then referenced at a future time by another XDS object. This allows broadcasters and service providers to insert larger multimedia files (such as images and audio files) into the channel without "clogging" the pipe.

Having smaller objects also makes it easier to synchronize the activity of the objects with other objects and the main program audio and also to maintain continuity between the experience of the user who tunes to the service at any time during the broadcast. To provide rapid access to standing data, an XDS service can consist of a continuing loop of "snippets" of the data the user is intended to view or use. Also, the XDS synchronization cue (based upon the synchronized nature of the IBOC signal) can control the timing and the flow of these objects so that they occur in time with events on the audio program.

For these more sophisticated implementations, all of the processing work and calculation is performed on the broadcast side, where it is easier to add processing power. The receiver only needs to parse small objects, and in general, one at a time, when they are scheduled to run.

A further benefit of XDS is that it can be used as a carrier of other markup languages. This is particularly advantageous. As receivers grow in sophistication they may desire to receive new data formats. The XDS architecture allows them to be included as tag sets while still allowing them to utilize existing XDS commands and functions.

For example, a SMIL file can be sent to a receiver using the storage tags, placed in memory on the receiver, and using synchronization, be rendered at a time desired by the broadcaster. This might be an important capability in making SMIL files work on IBOC because of the limited amount of bandwidth offered by IBOC. Applications like HTML and SMIL lend themselves towards larger documents referencing even larger multimedia files. And even though SMIL has its own synchronization capability, it may need to be given a point of reference to the main program audio. The synchronization function of XDS can accomplish this.

Secondly, the XDS structure allows for any byte array to be carried as a service object and provides a way in the object header for identifying the type of object. Therefore, besides allowing the channel to carry HTML and SMIL objects, the protocol can be allowed to carry other XML based languages to receivers, such as XML structures that are carrying traffic incident information to a navigation system.

Also, the XML structure can be used to create semantic structures for IBOC data services. These types of structures can be used to produce the filtering effects described earlier.

All of these elements can be incorporated into the protocol as IBOC data services evolve and the need for them arises. This can occur seamlessly within architecture intended to accommodate extension of the protocol without interrupting existing capability.

5 - CONCLUSION

As the NRSC convenes the process of developing a protocol for broadcast data services, there are many important factors ranging from the procedural to the technical that must be addressed by the committee. This white paper has attempted to frame those issues calling for an open process, following in the footsteps of organizations like ANSI, the IEEE, and ISO. A rationale has been given for such an approach and for the scope of the protocol that should be developed. Examples of communication systems and their underlying market principles have demonstrated the successful and unsuccessful paths the market can take. Namely, the paper has favored an open system protocol versus a closed system protocol. The specific limitations of the hybrid FM IBOC channel were discussed to illustrate the constraints on an IBOC data protocol design. Finally, the requirements of the data protocol as well as a suggested solution have been given.

APPENDIX A
REPRESENTATIVE SAMPLE OF IBOC USE CASES

Code	Category	Function	Description
DLS001	DLS	Broadcaster sends a music download to a device.	The broadcaster sends a special music service that consists of music that users can purchase by subscribing to the service.
DLS002	DLS	Device uses profile to select downloaded music	The device uses a profile to determine if the user is entitled to the music.
DLS003	DLS	Device writes downloaded music to a permanent storage device	Device selects music by profile from a download service and stores the information on a permanent storage unit on the device.
DLS006	DLS	Device plays music from a download service	User selects option to play download music and the device reads selections from the storage device, whether it is a CD, permanent storage, removable flash memory, etc., and plays them.
DLS008	DLS	Broadcaster sends a news download to a device	The broadcaster sends a special news service that consists of news that users can purchase by subscribing to the service.
DLS011	DLS	Service provider transmits an electronic version of a periodical	Service provider transmits an electronic version of a magazine or a newspaper that the device will then record for later viewing or hearing by the user.
DLS013	DLS	Service provider transmits an electronic version of a book	Service provider transmits an electronic version of a magazine or a newspaper that the device will then record if the user has subscribed for it, for later viewing or hearing by the user.
EAS001	EAS	User receives a warning about an emergency situation.	An emergency service provider alerts the community at large via one more IBOC stations.
EAS002	EAS	Send a weather alert	A weather alert is sent to a car alerting the driver to an impending weather event.
EPG001	EPG	Broadcaster transmits an electronic program guide of channel services	The broadcaster transmits a special service that indicates all of the services available on the channel and gives a description for each, and the user views them on the receiver.
EPG002	EPG	Receiver displays program guide information based upon device capability	The receiver scans the information in the electronic program guide and only displays the services that the device is capable of handling.

Code	Category	Function	Description
EPG003	EPG	Receiver displays program guide information based upon profile	The receiver scans the information in the electronic program guide and only displays the services that the user has indicated they are interested in by their profile.
EPG004	EPG	User selects a service from the electronic program guide	The user chooses a service from the electronic program guide and the device starts rendering that service.
FLM001	FLM	Fleet Management Schedule Distribution	A set of schedules and objectives for a specific fleet of vehicles is broadcast via the IBOC network.
FLM002	FLM	Request for mission status	A request for a status update from a specific fleet of vehicles is broadcast via the IBOC network.
LBP001	LBP	Broadcaster sends location encoded traffic information to be integrated with on-board GPS	A broadcaster sends traffic information encoded by zone and the user sees the traffic that is effecting their current zone and surrounding zones where the zone information is supplied by the on-board navigation system of the vehicle.
LBP003	LBP	Broadcaster sends location encoded ad information to be integrated with on-board GPS	A broadcaster sends an ad for a chain of stores or restaurants in a city encoded by zone and the user sees the information for a store or restaurant in their zone where the zone information is supplied by the on-board navigation system of the vehicle.
MSG001	MSG	Service provider sends a numeric page	Service Provider broadcasts a numeric page to an addressable device and the device with the matching address receives the message and displays it.
MSG002	MSG	Service provider sends a text page	Service Provider broadcasts a text page to an addressable device and the device with the matching address receives the message and displays it.
MSG003	MSG	Service provider sends updated stock quotes	Service provider sends updated stock quotes to addressable devices and the devices with appropriate addresses receive the messages and display them.
MSG004	MSG	Service provider sends updated financial information	Service provider sends updated financial news to addressable devices and the devices with appropriate addresses receive the messages and display the news.
MSG005	MSG	Service provider sends updated news information	Service provider sends news headlines to addressable devices and the devices with appropriate addresses receive the messages and display them.
MSG006	MSG	Service provider sends updated sports information	Service provider sends sports scores to addressable devices and the devices with appropriate addresses receive the messages and display them.
MSG007	MSG	Device uses profile information to customize messaging service	Device receives message service data encode by areas of interest and the device uses a service profile to display only the area of interest to the user.
LBP002	(LBP)	Broadcaster sends location encoded traffic information to be integrated with IBOC GPS	A broadcaster sends traffic information encoded by zone and the user sees the traffic that is effecting their current zone and surrounding zones where the zone information is supplied by an IBOC based GPS calculation.
MSG008	MSG	User receives a page and responds on a two-way network	The device receives a page via the IBOC network and displays it to the user, the user responds on a two-way network to the entity originating the page with the source address information included in the message.
LBP004	(LBP)	Broadcaster sends location encoded ad information to be integrated with IBOC GPS	A broadcaster sends an ad for a chain of stores or restaurants in a city encoded by zone and the user sees the information for a store or restaurant in their zone where the zone information is supplied by an IBOC based GPS calculation.
GPS001	(GPS)	User uses IBOC data to calculate global position	User uses IBOC broadcast data from three locations to triangulate global position.
ODP001	ODP	Broadcaster supplies on-demand visual traffic reports	A broadcaster sends visual traffic information that is stored on the receiver and viewed at the request of the user.
ODP002	ODP	Broadcaster supplies on-demand visual weather reports	A broadcaster sends visual weather information that is stored on the receiver and viewed at the request of the user.
ODP003	ODP	Broadcaster supplies on-demand visual news reports	A broadcaster sends visual news information that is stored on the receiver and viewed at the request of the user.
ODP004	ODP	Broadcaster sends an on-demand traffic report	A broadcaster sends a traffic report that is stored on the receiver and played back by the user at their request.
ODP005	ODP	Broadcaster sends an on-demand weather report	A broadcaster sends a weather report that is stored on the receiver and played back by the user at their request.

Code	Category	Function	Description
ODP006	ODP	Broadcaster sends an on-demand news report	A broadcaster sends a news report that is stored on the receiver and played back by the user at their request.
PAD001	PAD	Advertiser places visual advertisements	An advertiser places an ad during regular programming that is viewed on the screen of on a receiver by a user.
PAD002	PAD	Advertiser places supplemental audio advertisement with on-demand audio	An advertiser places an audio ad at the beginning and/or end of a piece of audio that is stored on the receiver and played by the request of the user.
PAD003	PAD	Advertiser enhances main program audio ad with visual data	An advertiser has data related to the information conveyed in the audio portion of an advertisement appear on the visual display of the receiver as the ad is heard by the user.
PAD004	PAD	Advertiser enhances program audio ad with supplemental audio data	An advertiser has supplemental audio data stored on a receiver for a period of time after an ad is aired to allow the user to retrieve pertinent information (such as prices, locations, phone numbers, etc.) related to the ad at a later time.
PAD005	PAD	Broadcaster enhances music programming with supplemental data	A broadcaster has data related to a song being aired appear on the visual display of the receiver.
PAD006	PAD	Broadcaster enhances music programming with supplemental audio data	A broadcaster has audio information related to a song stored on the receiver for a period of time it has aired to be played back upon request of the user.
PAD007	PAD	Broadcaster triggers visual traffic report on receiver	A broadcaster sends visual traffic information that is displayed on the receiver and viewed by the user.
PAD008	PAD	Broadcaster triggers visual weather report on receiver	A broadcaster sends visual weather information that is displayed on the receiver and viewed by the user.
PAD009	PAD	Broadcaster triggers visual news report on receiver	A broadcaster sends visual news information that is displayed on the receiver and viewed by the user.
PER001	PER	Service provider personalization request to the user	A service provider sends a request using the IBOC network to users asking the user to personalize the service using criteria particular to that service.
PER002	PER	User receives customized services	A broadcaster sends a data service with information that is customized by areas of interest to the user, the receiver uses on on-board profile to only display information of interest to the user as indicated by the user.
PER003	PER	User enters profile information into the device manually	A user enters profile information for a service using the input of a receiver device.
PER004	PER	User enters profile information on a web server	A user enters profile information for a service at a web site and the receiver uses a two-way network connection to retrieve the profile.
PER005	PER	Service profile is permanently stored on the receiver	The device receives profile information and that profile is stored on a permanent storage unit on the device.
PER006	PER	Service profile is stored on a web site.	The device receives profile information and the information is posted to a web site using a two-way network connection.
PER007	PER	Service profile is stored on a removable flash memory card.	The device receives profile information and the information is stored on a flash memory card that can be removed from the device.
PER008	PER	Service profile is stored on a CD.	The device profile is written to a CD and the receiver reads the profile information from the CD.
PER009	PER	Service profile is stored on a tape.	The device profile is written to a tape and the receiver reads the profile information from the tape.
RCM001	RCM	Radio station conducts a contest	Radio station sends data eliciting a response to a contest from one or more users in order to win something where the user responds to the contest via a two-way connection on the device.
RCM002	RCM	Radio station conducts a poll	Radio station sends data eliciting a response to a poll question from one or more users where the users responds to the contest via a two-way connection on the device.
RCM003	RCM	Radio station conducts a poll integrated with profile information	Radio station sends data eliciting a response to a poll question from one or more users where the users responds to the contest via a two-way connection on the device and the device transmits profile information back to the station.
RCO001	RCO	Service provider sells pre-recorded audio	The service provider transmits data that allows the user to purchase pre-recorded audio that is associated with the song or

Code	Category	Function	Description
			artist being played by the radio.
RCO002	RCO	Service provider sells concert tickets	The service provider transmits data that allows the user to purchase concert tickets that are associated with artist being played by the radio.
RCO003	RCO	Guest author sells a book	The broadcaster transmits data that allows the user to purchase a book by a guest being interviewed by the on-air personality.
RCO004	RCO	Radio station conducts promotion on behalf of a guest	The broadcaster transmits data promoting events on behalf of an artist whose song is being aired, or who is being interviewed by on-air personality.
RCO005	RCO	Retailer sells advertised merchandise	A retailer transmits data that allows the user to purchase merchandise related to the advertisement is being played on the radio.
RCO006	RCO	User conducts purchase via the device	The user interacts directly with the device to engage a commerce transaction.
RCO007	RCO	User manually inputs commerce instructions on the device	The user uses buttons on the receiver to initiate the commerce transactions.
RCO008	RCO	User uses voice response to initiate a commerce transactions	The user uses a voice response capability of a device to initiate a transaction.
RCO009	RCO	Device conducts transaction via a two-way connection	The device transmits the purchase instructions to an appropriate service provider using a two-way network connection and responds back to the user as to the result.
RCO010	RCO	Device conducts transaction via a flash memory card	The device records the purchase instructions to an appropriate service provider using a removable flash memory card which is later placed in a machine with a two-way network connection to transmit the information to the service provider.
RCO011	RCO	Device conducts a sale via bluetooth	The device transmits the purchase instructions to a machine with a two-way network connection using bluetooth. The purchase instructions are sent by the machine to the service provider.
RCO013	RCO	User requests more information on a product	The user interacts with the device to request more information about a portion of the broadcast such as an ad, a song, a guest, a station promotion, etc.
SAS001	SAS	Broadcaster transmits a free supplementary music program	A broadcaster sends secondary audio program consisting of a different selection of music than the main audio programming that the user can optionally choose to listen to.
SAS002	SAS	Broadcaster transmits a for-fee supplementary music program	A broadcaster sends a secondary audio program consisting of music that the user can subscribe to by entering an authorization code into the listening device.
SAS003	SAS	Broadcaster transmits a free supplementary news program	A broadcaster sends a secondary audio news program that the user can optionally choose to listen to.
SAS004	SAS	Broadcaster transmits a for-fee supplementary news program	A broadcaster sends a secondary audio news program that the user can subscribe to by entering an authorization code into the listening device.
SAS005	SAS	Broadcaster transmits a free supplementary weather program	A broadcaster sends a secondary audio weather program that the user can optionally choose to listen to.
SAS006	SAS	Broadcaster transmits a for-fee supplementary weather program	A broadcaster sends a secondary audio weather program that the user can subscribe to by entering an authorization code into the listening device.
SAS007	SAS	Broadcaster transmits a free supplementary traffic program	A broadcaster sends a secondary audio traffic program that the user can optionally choose to listen to.
SAS008	SAS	Broadcaster transmits a for-fee supplementary traffic program	A broadcaster sends a secondary audio traffic program that the user can subscribe to by entering an authorization code into the listening device.
SAS009	SAS	Broadcaster transmits a supplementary location based traffic program.	A broadcaster sends a secondary traffic program than is encoded with zone information and the user will hear the traffic information that effects the zone they are in and surrounding zones.
SDS001	SDS	Broadcaster transmits a free data news program	A broadcaster sends a data news program that the user can optionally choose to view on the receiver.
SDS002	SDS	Broadcaster transmits a for-fee data news program	A broadcaster sends a data news program that the user can subscribe to by entering an authorization code into the receiving device.

Code	Category	Function	Description
SDS003	SDS	Broadcaster transmits a free a data weather program	A broadcaster sends a weather program that the user can optionally view.
SDS004	SDS	Broadcaster transmits a for-fee data weather program	A broadcaster sends a data weather program that the user can subscribe to by entering an authorization code into the receiving device.
SDS005	SDS	Broadcaster transmits a free data traffic program	A broadcaster sends a data traffic program that the user can optionally view.
SDS006	SDS	Broadcaster transmits a for-fee data traffic program	A broadcaster sends a data traffic program that the user can subscribe to by entering an authorization code into the receiving device.
SDS007	SDS	Broadcaster transmits a location based traffic program.	A broadcaster sends a data traffic program than is encoded with zone information and the user will see the traffic information that effects the zone they are in and surrounding zones.
TEL001	TEL	Update service station information	The on-board service station information for the city or region a car is traveling in is updated via an IBOC network.
TEL002	TEL	Send a telematics weather alert	A telematics weather alert is sent to a car's telematics weather system on the IBOC network.
TEL003	TEL	Update point-of-interest information	The on-board point-of-interest information for the city or region a car is traveling in is updated via the IBOC network. This information includes names, locations, phone numbers, and even menus.
TEL006	TEL	Stolen car retrieval	A telematics message is sent via the IBOC network to cars in a city or region asking for a specific car to report its location via its two-way network where the car or cars being asked to respond have been stolen.
TEL009	TEL	Request traffic pattern information from vehicles by zone	A message is broadcast via the IBOC network asking vehicles to respond with location information via their two-way connection if they are in the zone indicated in the message.
TEL011	TEL	Update navigation information	Updates to the on-board guidance database for a region a car is traveling in are made via the IBOC network.
TEL012	TEL	Update traffic information for on-board navigation system	Traffic information is broadcast via the IBOC network that is coded by zone so that the on-board guidance system of the car can help a vehicle avoid potential traffic jams.
TEL014	TEL	Vehicle theft countermeasures initiation	A user notifies the police vehicle was stolen and a message is sent to the vehicle via an IBOC network to initiate the on-board theft countermeasures.
TSS001	TSS	Service provider encodes a portion of the broadcast for recording purposes	Service provider encodes a portion of a broadcast so that a device can determine the beginning and end of a program so that the device may record and label the program.