Media-accelerated Global Information Carrier

Engineering Specification

Revision 2.4

January 22, 2002

Authors:	Henry Juszkiewicz, Nathan Yeakel, Shri Amit, Richard Frantz, and Jason Flaks
Contact:	info@gibsonmagic.com
Latest Version:	www.gibsonmagic.com

Revision History

The following table is a record of all public releases of this document.

Revision	***Release Date***	***Comments***
1.0	9/18/99	Initial proposal to the audio industry at the 107th AES.
1.1	2/2/00	NAMM 2000.
2.0	1/20/01	NAMM 2001. Revised packet format.
2.1	9/21/01	Complete rewrite. Revised control protocol and application layer. New sample rate modification protocol.
2.2	10/18/01	Changed flag order in word 12. Added more figures and improved descriptions of all algorithms. Simplified the CTS/MIP control flow protocol. Fixed bug in control message numbering. Added section on Endian requirements.
2.3	11/16/01	Added MIDI message encapsulation and timecode, and a Blob component type. Also, added control packet requirement of at least 12 RX buffers on each port to ensure proper flow control.
2.4	01/22/02	Added copy protection bit. Added reset enumeration from any device on the network. Several minor edits to presentation.

Contents

Figures

Tables

Terms and Abbreviations

The following section lists and defines commonly used terms and abbreviations used in this specification:

100baseT	The 100-Megabit Ethernet physical layer.
A-port	A MaGIC port with a one-to-one wiring with respect to the cable.
A/D	Refers to analog-to-digital conversion, the process by which an analog signal is converted to a digital sample.
ADAT	Alesis Digital Audio Tape. See LightPipe.
ADCCP	Abbreviation for Advanced Data Communications Control Protocol. An ANSI standard, bit-oriented, data-link control protocol.
AES/EBU	Abbreviation for the Audio Engineering Society / European Broadcasting Union. Here refers to the 24-bit, 48 kHz, two channel digital audio standard.
ANSI	American National Standards Institute. The coordinating body for voluntary standards groups within the U.S.
AUTODIN	Acronym for Automatic Digital Network. The United States Department of Defense automatic switching network for telecommunications developed in the 1960s.
Audio Slots	MaGIC supports multiple Sample Rates by providing flexible mapping between samples and channels, called audio slots. For example, slots 1 and 2 are mapped to the first samples of channels 1 and 2 at 48 kHz, and to the first and second samples of channel 1 at 92 kHz.
B-port	A MaGIC port with a signal crossover with respect to the cable.
Bandwidth	The measure of data transmitted per unit of time across a given medium. Common units are bits per second (b/s) or bytes per second (B/s).

Bi-directional	Moving simultaneously in both directions. See Full-duplex.
Bit	The smallest unit of digital data. It has two possible values a logical high or one, or a logical low or zero.
Byte	A commonly used unit of data equal to eight bits.
CRC	see Cyclic Redundancy Check.
CTS	Clear To Send. This bit in the MaGIC packet allows devices to determine the flow of control packets from adjacent devices, based on their individual processing capabilities.
CAT-5	The interconnection cable used for Common Instrument Interface MaGIC physical layer. It is based on the IEEE 802.3 protocol.
Category 5	See CAT-5
Component	A unit of control in a MaGIC device that can issue or interpret a control command.
Control Link	A virtual connection established between two Components on a MaGIC network. It allows a Source Component to control a Target Component of the same type.
Control Message	A message sent by one MaGIC device to another in order to provide configuration or control information.
Cyclic Redundancy Check	A technique that employs an algorithm to compute a value for any arbitrary data. This value is then transmitted with the data allowing the recipient to re-compute and check the value to determine if the transmission was free of errors.
D/A	Refers to digital-to-analog conversion, the process by which digital samples are converted into an analog signal.

Daisy Chain	Network topology in which nodes are linked together such that no node is adjacent to more than two other nodes, and no loops may exist.
Device	See MaGIC device.
Device Address	The network-wide unique unsigned integer address assigned to the device by the STM during enumeration.
Ethernet	See IEEE 802.3.
Enumeration	The process by which the STM assigns a unique unsigned integer Device Address to each device on the network.
Firewire	See IEEE 1394.
Full duplex	The simultaneous exchange of data in both directions across a single connection.
MaGIC Control	The native protocol described in this document that allows MaGIC devices to control parameters on one another.
MaGIC Device	Any device equipped with a MaGIC Link that allows it to exchange bi-directional, fixed-length data and control, at a determined network sample rate.
MaGIC Link	The connection between two adjacent devices on a MaGIC network.
MaGIC Packet	A collection of networking headers, audio, and possibly control information transmitted synchronously across a MaGIC network.
MaGIC Port	The physical point of connection between the medium of transport and the MaGIC device.
Gigabit	An advanced 802.x standard that operates at 1000Mbps.
IEEE 802.3	A bus-based broadcast network with decentralized control operating at 10 or 100 Mbps.

IEEE 1394	A digital multimedia peripheral interface developed by Apple for digitally connecting consumer electronics with personal computers.
Inter-NIC	Network Information Center. The central authority that assigns worldwide unique IP addresses.
Jitter	The phase shifts of digital signals caused by mechanical or electrical imperfections. This deviation can lead towards a lack of synchronization of the signals involved.
Latency	Time delay in signal propagation as measured between the source and the target.
Lightpipe	An eight channel, 24-bit, 48 kHz, low-cost optical digital audio standard developed by Alesis. Also known as ADAT Optical and TossLink.
MAC Address	The worldwide unique Medium Access Control addresses.
MIDI	The Musical Instrument Digital Interface standard.
MIP	Message In Progress. This bit in the MaGIC packet allows adjacent devices to determine the flow of control information between them depending upon their individual processing capabilities.
Network Name	A mnemonic name assigned to a configured MaGIC network for easy identification in user interfaces. This name can be requested by and sent to other devices.
Nibble	A commonly-used unit of digital data equal to four bits.
RJ-45	The connector/port used for the connection based on the IEEE 802.3 Ethernet physical layer.
STM	System Timing Master. Every network automatically selects one device to be the STM. This device is responsible for enumerating the other devices on the network and for providing the sample clock.

Sample Rate	The number of samples transmitted per second. Measured in Hertz (Hz) or kilohertz (kHz).
Source	A Component that can issue a control command.
Star	MaGIC network topology in which Daisy Chains are connected together using a Routing Hub.
Target	A Component that can receive and interpret a control command.
T/DIF	The Tascam Digital Audio Interface. An 8-channel digital audio interface used mainly by in Tascam products.
UDP	User Datagram Protocol. The standard Internet connectionless transport protocol used for sending raw IP datagrams.
Uplink	MaGIC network topology that employs at least two Switching Hubs allowing several MaGIC Links to be multiplexed onto a single cable of a faster physical layer (such as Gigabit).

1. Introduction

1.1 Motivation

The Media-accelerated Global Information Carrier (MaGIC) is motivated by the following goals:

1. **Enhanced real-time digital sonic fidelity:**

   Despite dramatic advances in technology, real-time high-fidelity digital audio has yet to permeate both production and live performance. Increasing demand has motivated little effort to apply modern network technology towards producing superior quality real-time audio devices, at low prices.

   MaGIC uses state-of-the-art technology to provide up to 32 channels of 32-bit bi-directional high-fidelity audio with sample rates up to 192 kHz. Data and control can be transported 30 to 30,000 times faster than MIDI. Added cable features include power for instruments, automatic clocking, and network synchronization.

2. **Interoperability**:

   There is a compelling need in the audio industry for an Open Architecture digital interconnect that would allow audio products from different vendors (musical instruments, processors, amplifiers, recording and mixing devices, etc.), to seamlessly communicate. MaGIC provides the ability to create such networks appropriate for use in a wide variety of environments ranging from professional audio to home music installations.

3. **Complete digital solution**:

   Existing digital systems rely on archaic analog interfaces to connect with other devices. The increasing demand for interconnected devices has resulted in diminished sound quality, caused by repeated analog-to-digital and digital-to-analog conversions. Often these conversions lead to prohibitive size and power product requirements. The situation begs for a universal end-to-end digital solution.

4. **Simple installation and ease-of-use**:

   Most existing systems are difficult to install, lack flexible reconfiguration capabilities, and do not take advantage of user-friendly hardware and software interfaces. MaGIC provides a single cable solution that is trivial to install, requires virtually no maintenance, and offers a data link layer that supports a simple yet sophisticated protocol, capable of offering a superior user experience.

Several digital interconnection specifications such as AES/EBU, S/PDIF, ADAT (Light Pipe), and IEEE 1394 (Fire Wire) have emerged but none satisfy the unique requirements of live audio performances, particularly in the areas of clocking, distance synchronization, and jitter/latency management.

The Media-accelerated Global Information Carrier was commissioned to overcome these limitations of point-to-point solutions by providing inexpensive, end-to-end, and real-time digital sonic fidelity.

1.2 Scope of the Document

This document is written for manufacturers, and hardware and software developers who wish to develop MaGIC compliant devices.

It describes the physical, mechanical, and electrical interconnection based on the 100-Megabit Ethernet physical layer. The document also specifies the network timing and synchronization mechanism, data transport layer, and the control protocol which are independent of the physical layer.

This specification does not supply any information about specific devices or products. It is expected that the development of a product will require a complementary document that would describe the specific features and functions of that product while ensuring compatibility with this specification.

1.3 Historical Background

Henry Juszkiewicz, Executive Director of the MaGIC project, has supported the idea of a digital audio network for nearly a decade. In addition to launching the preliminary Guitar Innovation Group (GIG) studies, Henry has funded extensive research and development in digital audio connectivity at the University of California at Berkeley Center for New Music and Audio Technologies.

The guiding principle of this research was to apply technology invented for computer network products to audio equipment, and develop an interconnect that would be:

?? Reliable over long distances, locally repairable, trivial to install, and simple to use.

?? Capable of supporting multiple audio channels of advanced fidelity audio.

?? Fit for enabling installations to scale beyond the capacity of existing multiple cable solutions and meet the requirements of permanent installations such as live venues and recording studios.

?? Able to provide power for digital instruments thereby removing the need for batteries.

?? Supportive of intuitive control interfaces.

?? Successful in providing extreme simplicity and reliability.

The above stated goals were to be accomplished by augmenting, not diminishing the acoustic, electric, or physical characteristics of musical instruments.

MaGIC is the result of these efforts.

1.4 Document Organization

Chapters 1 and 2 provide a general introduction to the goals, concept, and architecture of the MaGIC standard. It includes an overview of the various interfaces and protocols that comprise MaGIC without subjecting the reader to rigorous details.

Chapters 3 and 4 are directed at the hardware and mechanical engineers designing the physical layer of a MaGIC device.

Chapter 5 is primarily of interest to the firmware and software engineers implementing the packet framework and Ethernet compatibility.

Chapter 6, 7, and 8 describe the MaGIC application layer and is aimed at software engineers implementing enumeration, audio, control, and other related protocols.

Chapter 7 describes the specific properties and requirements of the System Timing Master, which is also of interest to engineers implementing clock and synchronization hardware.

2. Architectural Overview

2.1 Devices

A MaGIC-compliant device is equipped with at least one MaGIC Link through which it can exchange real-time, bi-directional, fixed-length data and control information, at a determined network sample rate. Unless specified otherwise, the term "device" is to be understood as referring to a MaGIC-compliant device.

2.2 Network Topology

MaGIC networks can be arranged in the following three topologies:

1. **Daisy Chain:** As shown in the figure below, a Daisy Chain network refers to devices connected together to form a single chain.

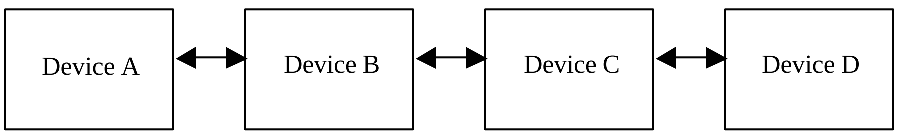


Figure 2-1: Daisy Chain network topology

2. **Star:** As shown in the figure below, a star network is one in which several Daisy Chain networks are connected together using a Routing Hub.

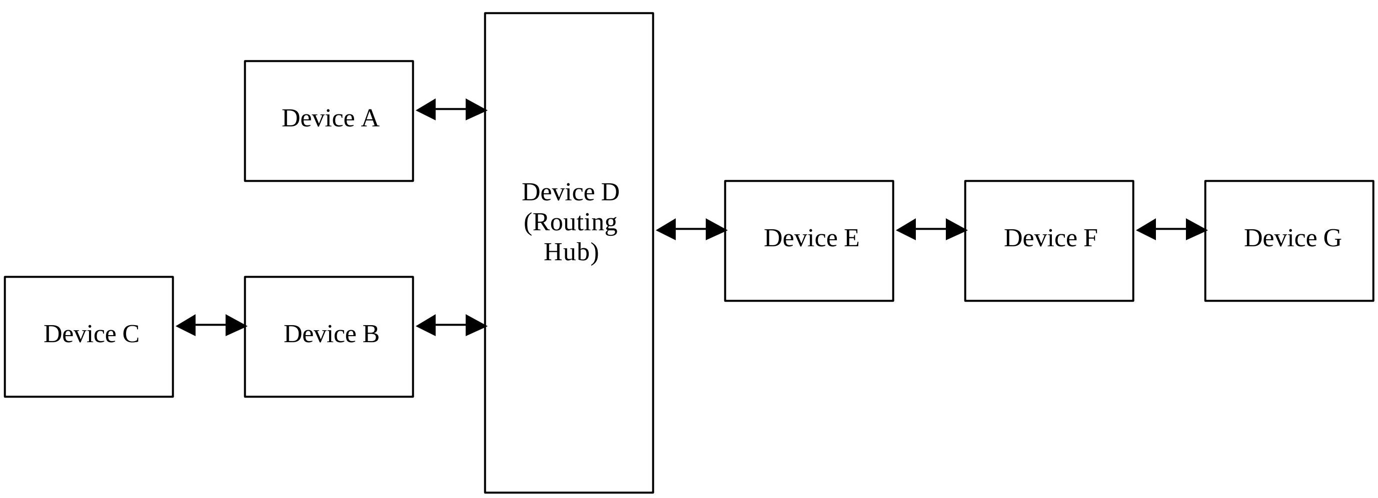


Figure 2-2: Star network topology

3. **Uplink:** As shown in the figure below, an uplink network topology employs at least two Switching Hubs that allow data from several MaGIC Links to be multiplexed onto a single cable.

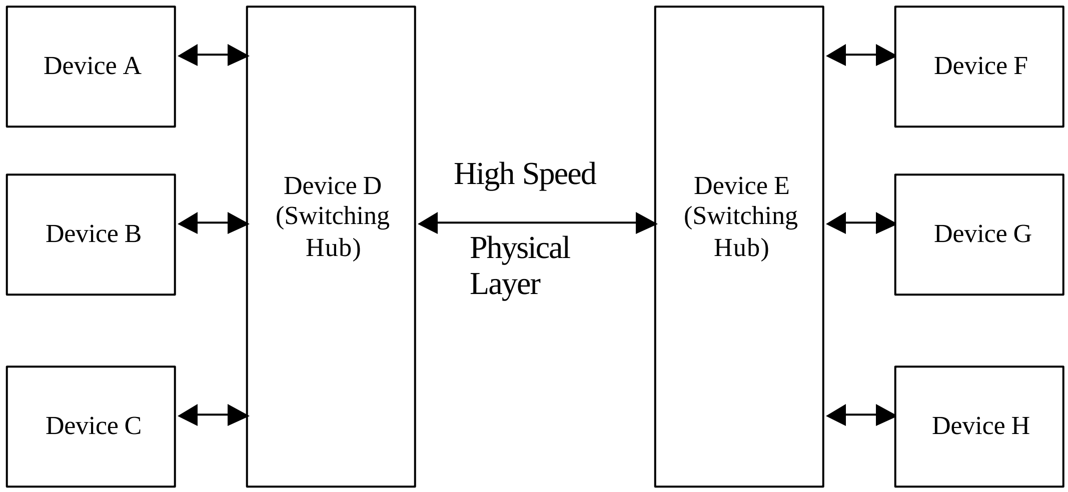


Figure 2-3: Uplink network topology

2.3 Protocol Stack

Not unlike common networking protocols, the MaGIC protocol is stacked into three distinct layers. From the lowest to highest, they are:

1. *Physical Layer*: consists of the mechanical and electrical specifications required to form the physical network. This layer is compatible with the IEEE 802.3 Ethernet physical layer.

2. *Data Link Layer*: as defined by the IEEE 802.3 Ethernet protocol. It views bits transported by the Physical Layer as defined sequences called frames that can be transported across any standard Ethernet-compatible network.

3. *MaGIC Application Layer*: uses the frames transported by the Data Link Layer to encapsulate MaGIC-specific information into packets that allow MaGIC devices to exchange real-time bi-directional audio and control data.

The MaGIC application layer is independent of the two layers under it thereby providing the ability to easily change the mode of physical transport based on available technology.

2.4 Mechanical Interface

The MaGIC protocol is suitable for a variety of physical interfaces. Examples include: the IEEE 802.3 Ethernet physical layer, the high-speed multi-link Optical Interface, wireless interfaces, the Ethernet Gigabit-based physical layer, etc.

This specification only describes the MaGIC Link based on the IEEE 802.3 100-Megabit Ethernet physical layer, which uses standard Category 5 (Cat 5) cables, and RJ-45 connectors.

2.5 Electrical Interface

The electrical component of the Ethernet physical layer is based on a 4b/5b data-encoding scheme, which is then scrambled to eliminate RF 'hot spots', thereby reducing emissions. This is a well known, tested, and documented data transport with a large installed base.

Half of the eight conductors in a Category 5 cable are used for data transport and the other half supply at least 500mA at plus nine volts DC for devices that can operate on limited power.

2.6 Data Link Layer

The MaGIC Data Link Layer is based on the IEEE 802.3 Ethernet Data Link Layer. Data is transmitted between devices at a synchronous rate in discrete fixed-size frames. Each frame is 55 words long, contains the Start of Frame, Source and Destination MAC Addresses, Length, a fixed-size Payload, and CRC fields, and is transmitted at a fixed network sample rate.

2.7 Application Layer

The Application Layer encapsulates a MaGIC packet in the Payload field of the Data Link Layer frame. Each packet consists of thirty-two, 32-bit data slots as 16, 24, 28 or 32 bits of PCM audio. Specific compressed data formats can also be supported. Each individual audio pipe can be reassigned as arbitrary 32-bit data if desired.

The MaGIC packet can also contain control information for processes like network enumeration, sample rate modification, and parameter control. Other control protocols such as MIDI can also be encapsulated.

2.8 System Timing Master

A single source of synchronization is required for the devices on a network to process data in phase with one another. This source is called the System Timing Master (STM). The STM is selected automatically using preset system rules and is responsible for using the enumeration protocol to assign dynamic addresses to all devices available on the network.

The default MaGIC frame timing is 48 kHz with an acceptable tolerance of 80 picoseconds. This timing is locally generated by the STM, and recovered and regenerated by all other devices. The Ethernet signaling rate is asynchronous with the rate at which frames are transmitted.

2.9 Control Protocol

The MaGIC control protocol allows any device to control an arbitrary parameter on another device.

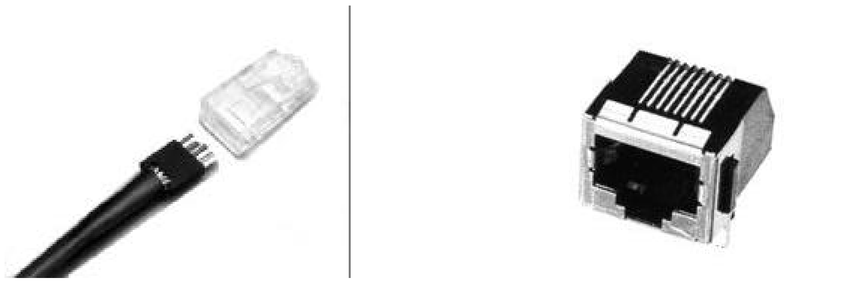
This eliminates the need for predefinition of parameter and controller messages as is common in other protocols such as MIDI.

Non-MaGIC control messages can also be exchanged by encapsulating them in the MaGIC control words. At this time, MIDI is the only defined encapsulated control type.

3. Mechanical

3.1 The MaGIC Link

The IEEE 802.3 implementation of the MaGIC Link is comprised of the RJ-45 connector and 100baseT Category 5 cable.



RJ-45 Cable RJ-45 Port

Figure 3-1: RJ-45 cable and port

3.2 The MaGIC Port

3.2.1 Port Specifications

The Category 5 cable consists of four twisted pairs; two of which are assigned for data transport and the other two for power. Ports must be wired in a one-to-one configuration allowing each conductor to be connected to the same pin on both ports.

MaGIC requires a crossover on one of the two ports as shown in the figure below. This allows data transmitted from one device to be received by another. The port using the standard 568A (one-to-one) wiring is defined as the **A-port**. And, the port using the standard 568B with the signal crossover is defined as the **B-port**.

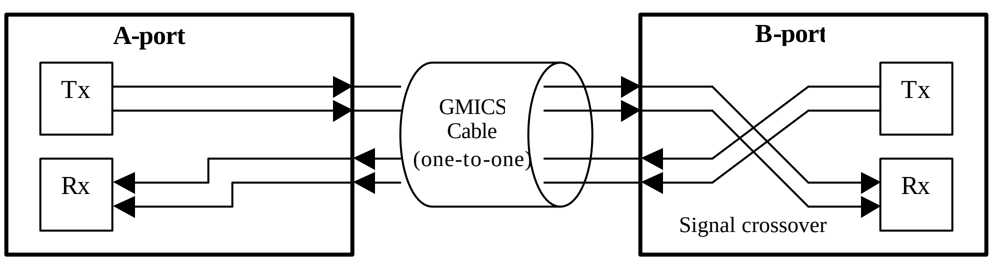


Figure 3-2: The A and B ports

3.2.2 Pin Assignments

Signal Name	Port A pin number	Port B pin number
Transmit Data (TX) +	1	3
Transmit Data (TX) -	2	6
Receive Data (RX) +	3	1
Receive Data (RX) -	6	2
Power Ground	4	4
Power Ground	5	5
Voltage +	7	7
Voltage +	8	8

Table 3-1: Port pin assignments

This pin assignment ensure that

?? Signals are transported over twisted pairs

?? Transmit and Receive signals use the same pins as a NIC card.

?? The two pairs of wires not used in Ethernet networks carry power. This reduces the possibility of damage if a MaGIC device is accidentally connected into a computer network connector.

3.2.3 Valid Configurations

One of the key features of MaGIC is the automatic determination of the System Timing Master. (See chapter 7 for a detailed description). To make that possible, a device may not have more than one A-port but is allowed any number of B-ports.

3.3 The MaGIC Cable

3.3.1 Cable Requirements

The MaGIC Link has been designed to use 100baseT Category 5 cables:

?? Of lengths up to 130 meters

?? That include all four twisted pairs

?? With at least 24 gauge stranded wire

It should be noted that MaGIC uses Category 5 patch cords that are always wired as a one-to-one assembly and not special crossover cables. Cables must be connected between A and B ports, not A to A or B to B.

It is strongly recommended that MaGIC devices provide a mechanism (such as a green LED) to notify the user of a proper connection. This would allow the user to easily detect and rectify incorrectly connected cables.

A MaGIC network should never be wired in such a fashion that any loops exist.

3.4 Dominant Data Flow

While it is true that the MaGIC protocol is symmetrical and bi-directional, there is almost always a dominant direction to the flow of data due to the nature of audio devices. The figure below illustrates a simple network consisting of a guitar, an effects box, and an amplifier.

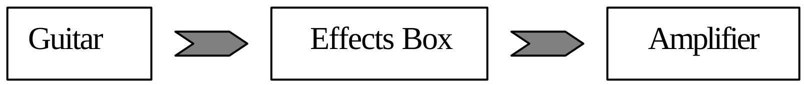


Figure 3-3: Direction of dominant data flow in a simple network

Consider below a more complicated network with guitars, a microphone, and an amplifier connected to a mixer, which is in turn connected to a recording device. When recording the dominant direction is **to** the recorder, and during playback it is **from** the recorder.

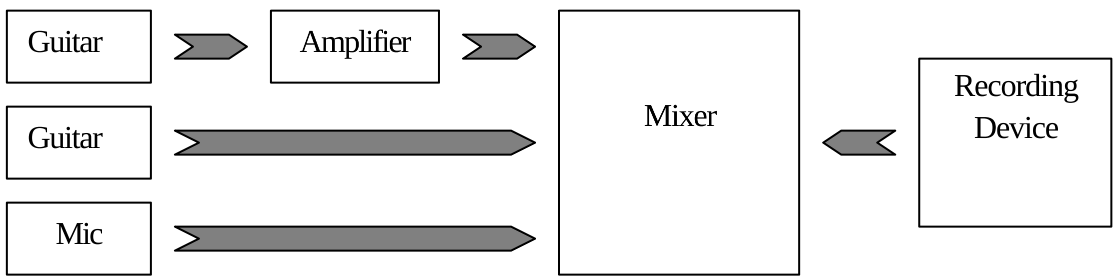


Figure 3-4: Direction of dominant flow in a network involving a recording device

3.5 Robustness and Reliability

In order for MaGIC-enabled devices to be used for live performance, particular attention to the reliability and robustness of the network is required. The following list suggests measures for vendors to increase the quality of their connection equipment.

?? Require highly flexible cables that can sustain repeated twisting, turning, and mechanical stress often experienced in live environments

?? Take into account a reasonable amount of mechanical stress typically induced upon ports and over-stretched or tangled cables.

?? Employ physical supports such as locking clips and special cable ends for the RJ-45 connector to prevent the clip from breaking or from accidental unplugging.

?? Use stranded instead of solid wire cables in all environments except permanent fixed installations.

?? Provide adequate protection against high voltage/current cables running nearby or even bundled with a MaGIC cable.

4. Electrical

4.1 IEEE 802.3 Compatibility

MaGIC shares a common physical layer with Ethernet. It is UDP compatible and has no handshaking protocol or retransmission ability.

Each individual link occupies the entire bandwidth of a discrete 100baseT link in full-duplex mode. This is necessary to provide the bandwidth needed for live synchronous audio.

Therefore, MaGIC may only be said to be compatible with Ethernet at its lowest physical layer of abstraction.

4.2 Timing Parameters

4.2.1 Sample Clock Recovery

The recovered sample clock is based on the incoming sample rate to ensure synchronous data processing, and, if need be, can be multiplied up to the system sample rate.

With the exception of devices with sample rate conversion capabilities, the STM should supply sample timing for other devices on the network with a maximum frame-to-frame jitter of 80 picoseconds.

All other devices must generate their outgoing frames in-phase with the stream of incoming frames. The frame-to-frame jitter of the outbound frames from non-STM devices must not exceed 160 nanoseconds. This is not a measure of accumulated jitter.

4.2.2 Latency Limitations

In order for MaGIC to function as a real-time digital link, audio latency must be contained to a low deterministic minimum.

There are three sources of latency in a MaGIC network:

1. *Physical Layer*: for a 100baseT physical layer this is usually in the range of 10-40 microseconds.

2. *Digital/Analog conversion*: analog-to-digital (A/D) and digital-to-analog (D/A) converters usually add a significant delay of 3,000-10,000 microseconds. Therefore, these converters must be chosen with care particularly for devices used in live performance.

3. *Device processing*: each MaGIC device should use no more than 250 microseconds to process and then forward an incoming audio frame. This limit

refers specifically to the processing of audio between the two digital MaGIC ports A and B, and not between a MaGIC port and an analog port.

4.2.3 Jitter Management

Minimizing jitter is an important requirement in designing circuits for sample rate recovery. The exact amount will vary depending upon the quality required for the specific application but as a general rule each device should manage jitter to less than 80 picoseconds.

4.3 Power

4.3.1 Cable Supply Requirements

A MaGIC cable must supply 18 to 24 Volts power on pins 7 and 8. This ensures a minimum of 9 Volts DC at greater than 500mA across the maximum cable length, as measured at the cable terminus. We strongly recommend supplying 24 Volts DC so as to account for attenuation.

Current limiting should occur at 1 Amp and should not be in the form of a standard fuse. A PolySwitch is highly recommended for protection against over-current conditions that may damage the device. Full power should be restored if possible upon correction of the fault. Each B-port must be independently protected to prevent a single defective link from disabling all others.

All B-ports are required to supply power as specified above.

4.3.2 Powered Device Requirements

Devices powered by the cable supply must operate on a range of voltages from 24 Volts DC to 9 Volts DC. The device must never draw more than 500mA. Proper heat dissipation at 24 Volts DC must be designed as part of the device.

Note that the power supplied by one port must not be used by more than one device within a Daisy Chain. Only end point devices like instruments are permitted to use the power.

Allowing power to pass through a device is attractive but practically unsupportable because it possibly violate the maximum cable length, giving rise to a voltage drop that would limit the voltage at the instrument, to less than the required minimum.

5. Data Link Layer

5.1 Overview

The Data Link Layer is based on the following 32-bit, 55-word Ethernet frame.

✍ Increasing time

<table>
<tr><th>Word</th><th>B31 - B28</th><th>B27 – B24</th><th>B23 - B20</th><th>B19 – B16</th><th>B15 – B12</th><th>B11 – B8</th><th>B7 – B4</th><th>B3 – B0</th></tr>
<tr><td>0</td><td>5</td><td>5</td><td>5</td><td>5</td><td>5</td><td>5</td><td>5</td><td>5</td></tr>
<tr><td>1</td><td>D</td><td>5</td><td>5</td><td>5</td><td>5</td><td>5</td><td>5</td><td>5</td></tr>
<tr><td>2</td><td colspan="8">Destination MAC Address</td></tr>
<tr><td>3</td><td colspan="4">Source MAC Address</td><td colspan="4">Destination MAC Address continued</td></tr>
<tr><td>4</td><td colspan="8">Source MAC Address Continued</td></tr>
<tr><td>5</td><td colspan="4" rowspan="7">Reserved for Networking</td><td colspan="4">Length</td></tr>
<tr><td>6</td><td colspan="4" rowspan="7">Headers</td></tr>
<tr><td>7</td></tr>
<tr><td>8</td></tr>
<tr><td>9</td></tr>
<tr><td>10</td></tr>
<tr><td>11</td></tr>
<tr><td>12</td><td></td><td></td><td></td><td></td></tr>
<tr><td>13 – 53</td><td colspan="8">Payload</td></tr>
<tr><td>54</td><td colspan="8">CRC32</td></tr>
</table>

Table 5-1: The MaGIC Frame

Bits 16-31 of word 12 and words 13-53 comprise the Payload and are described in the next chapter. The following subsections describe all other fields in detail.

5.2 Preamble and Start of Frame

Word	*B31 - B28*	*B27 – B24*	*B23 - B20*	*B19 - B16*	*B15 – B12*	*B11 – B8*	*B7 – B4*	*B3 – B0*
0	5	5	5	5	5	5	5	5
1	D	5	5	5	5	5	5	5

Table 5-2: Preamble and Start of Frame Words

Words 0 and 1 are as described in sections 7.2.3.2 and 7.2.3.3 of CSMA/CD IEEE 802.3 specification [1].

5.3 Source and Destination MAC Addresses

Word	*B31 - B28*	*B27 – B24*	*B23 - B20*	*B19 - B16*	*B15 – B12*	*B11 – B8*	*B7 – B4*	*B3 – B0*
2	**Destination MAC Address**							
3	**Source MAC Address**				**Destination MAC Address continued**			
4	**Source MAC Address Continued**							

Table 5-3: Source and Destination MAC Address Words

Words 2-4 specify source and destination worldwide unique MAC Address. They allow MaGIC devices to remain compatible with existing and future network hardware.

5.4 Length

Word	*B15 – B12*	*B11 – B8*	*B7 – B4*	*B3 – B0*
5	**Length**			

Table 5-4: Length Bytes

Bits 0-15 of word 5 contain the number of bytes following this field except the CRC. As seen in table 5-1 above, in the MaGIC frame this length is a constant 194 bytes (0x00C2).

5.5 Reserved Networking Headers

Word	*B31 - B28*	*B27 – B24*	*B23 - B20*	*B19 - B16*	*B15 – B12*	*B11 – B8*	*B7 – B4*	*B3 – B0*
5	**Reserved for Networking Headers**							
6								
7								
8								
9								
10								
11								
12								

Table 5-5: Reserved Networking Headers Words

Bits 16-31 of word 5, words 6-11, and bits 0-15 of word 12 are reserved for inserting data compatible with the TCP/IP categories, UDP encapsulation, or WAN applications. They are not used in isolated MaGIC networks.

5.6 CRC-32

Word	*B31 – B28*	*B27 – B24*	*B23 - B20*	*B19 – B16*	*B15 – B12*	*B11 – B8*	*B7 – B4*	*B3 – B0*
55	**CRC32**							

Table 5-6: CRC-32 Word

Word 54 of the frame contains a 32-bit Cyclic Redundancy Check (CRC) for the data contained in entire packet. The algorithm is based on the standard CRC-32 polynomial used in AUTODIN, Ethernet, and ADCCP protocol standards. Appendix A contains a source code example of a CRC-32 generation function.

The CRC computation and checking is optional. Bit 26 of word 12 (described in section 6.5.2) denotes whether a valid CRC has been computed.

5.7 Endian Format

All data on a MaGIC network must be Big Endian. Any Little Endian device must accordingly swap necessary bytes before sending and after receiving frames.

6. Application Layer

6.1 Overview

The MaGIC Application Layer is based on the following 32-bit, 41.5-word packet used to transport real-time audio and control data. Note that the word indices in the left most column have been preserved with respect to the payload field of the MaGIC frame shown above in table 5-1.

✍ Increasing time

Word	*B31 - B28*	*B27 – B24*	*B23 - B20*	*B19 – B16*	*B15 – B12*	*B11 – B8*	*B7 – B4*	*B3 – B0*
12	Configuration	Bits	Cable Num	S-Rate				
13	Frame Count / Timecode							
14	Audio Valid							
15	Audio Express							
16	Audio Slot 1							
17	Audio Slot 2							
18	Audio Slot 3							
19	Audio Slot 4							
20	Audio Slot 5							
21	Audio Slot 6							
22	Audio Slot 7							
23	Audio Slot 8							
24	Audio Slot 9							
25	Audio Slot 10							
26…47	Audio Slots 11… 32							
48	Control Message				Version		Configuration	
49	Destination Device Address				Source Device Address			
50	Destination Component Address				Source Component Address			
51	Control Data 1							
52	Control Data 2							
53	Control Data 3							

Table 6-1: The MaGIC Packet

The MaGIC packet can be divided into the following sections:

?? *Configuration*: fields that specify the context and configuration in which to interpret the packet.

?? *Audio*: fields containing the audio samples and related control bits. These fields can also contain arbitrary data if needed.

?? *Control*: fields containing control messages and data being exchanged between MaGIC devices.

These sections are described in detail the following paragraphs. For pedagogic continuity the Audio and Control sections have been addressed before the Configuration bits.

6.2 Audio

6.2.1 Audio Valid

Word	B31 - B28	B27 – B24	B23 - B20	B19 - B16	B15 – B12	B11 – B8	B7 – B4	B3 – B0
14	Audio Valid							

Table 6-2: Audio Valid Word

Word 14 of the MaGIC packet is used to determine which audio slots (see section 6.2.3 below) contain valid data. Bits 0-31 of this word are mapped to Audio Slots 1-32 (words 16-47) respectively. For example, if bit 0 were set it would denote valid audio in Audio Slot 1. If bit 1 were set it would denote valid audio in Audio Slot 2, and so on.

6.2.2 Audio Express

Word	B31 – B28	B27 – B24	B23 - B20	B19 - B16	B15 – B12	B11 – B8	B7 – B4	B3 – B0
15	Audio Express							

Table 6-3: Audio Express Word

Much like the Audio Valid word described above, bits 0-31 of word 15 are mapped to Audio Slots 1-32 (words 16-47) respectively. This allows a sample arriving on the corresponding input channel to be expressed unaltered on the mapped output channel. For example, setting bit 0 would forward Audio Slot 1 unchanged to the mapped output channel. If bit 1 were set it the same would happen to Audio Slot 2, and so on.

This feature allows simpler devices within a Daisy Chain to reduce overhead, particularly when multiplexing with a higher bandwidth backbone. By definition, this feature is not applicable to end points in a network.

A hub may or may not respond to these bits depending upon its specific function. For example, it must respond when providing an uplink but may choose to ignore them in the case of a mixer.

Note that where the audio is expressed depends entirely on the input channel to output channel mapping. Setting this bit only ensures that the audio will bypass any processing or alteration.

6.2.3 Audio Slots

Word	B31 – B28	B27 – B24	B23 - B20	B19 – B16	B15 – B12	B11 – B8	B7 – B4	B3 – B0
16	Audio Slot 1 (first sample)							
17	Audio Slot 2 (second sample)							
18	Audio Slot 3 (third sample)							
19	Audio Slot 4 (fourth sample)							
20	Audio Slot 5 (fifth sample)							
21	Audio Slot 6 (sixth sample)							
22	Audio Slot 7 (seventh sample)							
23	Audio Slot 8 (eight sample)							
24	Audio Slot 9 (ninth sample)							
25	Audio Slot 10 (tenth sample)							
26…47	Audio Slots 11… 32 (eleventh – thirty second samples)							

Table 6-4: Audio Slots at 48 kHz

Words 16-47 of the MaGIC packet contain the audio samples. This notion of slots allows MaGIC to support multiple sample rates by providing a flexible mapping between the rate and the channels being transmitted.

As shown in the table above, at the default sample rate of 48 kHz, each audio slot corresponds to a single sample mapped to a single channel. Therefore at this rate, one sample each, thirty-two different channels may be transmitted.

In order to achieve higher fidelity, it is desirable to operate the network at a higher sample rate. At a sample rate of 96 kHz, one channel of audio is assigned two audio slots resulting in a possible transmission of two samples each, belonging to sixteen different channels as shown in the figure below:

Word	B31 – B28	B27 – B24	B23 - B20	B19 – B16	B15 – B12	B11 – B8	B7 – B4	B3 – B0
16	Audio Slot 1 (first sample)							
17	Audio Slot 2 (second sample)							
18	Audio Slot 3 (first sample)							
19	Audio Slot 4 (second sample)							
20	Audio Slot 5 (first sample)							
21	Audio Slot 6 (second sample)							
22	Audio Slot 7 (first sample)							
23	Audio Slot 8 (second sample)							
24	Audio Slot 9 (first sample)							
25	Audio Slot 10 (second ample)							
26…47	Audio Slots 11… 32 (… so on)							

Table 6-5: Audio Slots at 96 kHz

The following table shows the mapping between sample rate, audio slots, and channels transmitted at the various defined MaGIC network sample rates.

Sample Rate (kHz)	Slots per Channel	Total Channels
44.1	1	32
48	1	32
96	2	16
192	4	8

Table 6-6: Sample Rate – Audio Slot – Channel Mapping

Even though MaGIC was envisioned as a digital audio standard, these Slots (words 16-47) can be used to transmit any arbitrary data in any format a sender and recipient can agree on.

6.3 Control

6.3.1 Version Number

Word 49	B31 – B28	B27 – B24	B23 - B20	B19 – B16	B15 – B12	B11 – B8	B7 – B4	B3 – B0
					Version Number			

Table 6-7: Version Number Byte

Bits 8-15 of word 49 of the MaGIC packet are used for specifying the MaGIC protocol version number being used by the network. The 8-bit field should be formatted as follows:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Integer	Integer	Integer	Fraction	Fraction	Fraction	Fraction	Fraction

Table 6-8: Version Number Format

Version numbers are defined in the standard dot notation. Bits 0-4 are used for the fraction and bits 5-7 for the integer.

6.3.2 Control Message

Word 48	B31 – B28	B27 – B24	B23 – B20	B19 – B16	B15 – B12	B11 – B8	B7 – B4	B3 – B0
	Control Message							

Table 6-9: Control Message Bytes

Bits 16-31 of word 48 define the control message being sent. Appendix B lists all the standard control message types defined at this time.

6.3.3 Source and Destination Device Addresses

Word 49	B31 – B28	B27 – B24	B23 - B20	B19 – B16	B15 – B12	B11 – B8	B7 – B4	B3 – B0
	Destination Device Address				**Source Device Address**			

Table 6-10: Source and Destination Device Addresses

Word 49 contains the destination device and the source device addresses in bits 16-31 and 0-15 respectively.

These fields allow a device to address a control packet from itself to another device on the network. As a control packet is sent from one device to another, each device evaluates the Destination Device Address field to determine if it should process the packet. If not, it must forward the packet along the network ensuring that the packet will eventually reach its intended destination(s).

Control packets can also be multicast or broadcasted. The following table lists reserved addresses (not assigned to any device during enumeration) used for this purpose:

Name	Address	Description
System Broadcast	0xFFFF	All devices on a network must process a message with this destination address.
Local Hub Broadcast	0xFFFE	If a hub generates this broadcast it must forward it to all its B-ports. If it receives the message on one of its ports, it should process it and then forward it on all ports except it's A port, and the port it received the message on.
Daisy Chain Broadcast	0xFFFD	All devices on a Daisy Chain must process and forward this broadcast. A hub should only forward it to its B-ports if it generates the message itself or if it receives it on it's A port.
Startup	0xFFFC	Self-assigned startup address for all devices. See chapter 5 for details.
Base	0x0000	Addressed used by the STM. See chapter 5 for details.

Table 6-11: Reserved Device Addresses

6.3.4 Source and Destination Component Addresses

Word	B31 – B28	B27 – B24	B23 - B20	B19 – B16	B15 – B12	B11 – B8	B7 – B4	B3 – B0
50	Destination Component Address				Source Component Address			

Table 6-12: Source and Destination Component Addresses

Word 50 contains the destination component and the source component addresses in bits 16-31 and 0-15 respectively. Components and their function will be described in detail in chapter 8.

These fields (in conjunction with the Source and Destination Device Address) allow a parameter on a device to address a control packet from itself to a parameter on another device on the network.

6.3.5 Control Data

Word	B31 – B28	B27 – B24	B23 - B20	B19 – B16	B15 – B12	B11 – B8	B7 – B4	B3 – B0
51	Control Data 1							
52	Control Data 2							
53	Control Data 3							

Table 6-13: Control Data Words

Words 51 through 53 are designated to transmit supporting data for control messages. Examples are discussed in chapters 7 and 8.

6.3.6 Sending and Receiving Control

The flow of audio in MaGIC is synchronously whereas that of control is not. Audio is present in every outgoing packet issued at the defined network sample rate. Control information, on the other hand, is included in the packet only when needed.

Note that if a certain packet does not contain control, the packet length does not change. Instead, the Control Valid Bit (described in section 6.5.2 below) is set to low to denote that no valid control information is present.

Before sending a control message a device must:

1. Ensure that the adjacent device is ready to receive the message using the CTS and MIP control bits described in section 6.5.1.

2. Setup the validity bits described in section 6.5.2.

3. Issued the message as part of the next outgoing packet on the desired port(s).

Upon receiving a new message a device compares the Destination Device Address field against its own address. If they match it must process the message, otherwise it must forward it to other connected devices thereby ensuring that the packet will eventually reach its destination.

6.4 Configuration

6.4.1 Clear To Send and Message In Progress

Word	*Bit 31*	*Bit 30*
12	**Clear To Send (CTS)**	**Message In Progress (MIP)**

Table 6-14: CTS and MIP Bits

Bits 31 and 30 of word 12 are the Message In Progress (MIP) and Clear To Send (CTS) bits respectively. They allow a recipient device to manage its limited control message buffer space against several possibly faster senders.

The following series of figures illustrate the use of these bits by considering an example of a device X sending a message to an adjacent device Y:

1. Initial state:

All CTS bits are high indicating that anyone can send a message.

All MIP bits are low indicating no messages in progress.

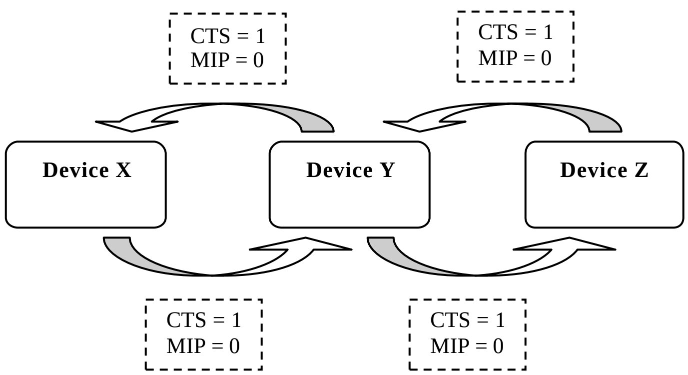


Figure 6-1: Initial state of the CTS MIP control message exchange protocol

2. Sending a message:

Device X raises the MIP bit on the packets outgoing to device Y and starts sending control messages.

Device Y receives and processes the packets.

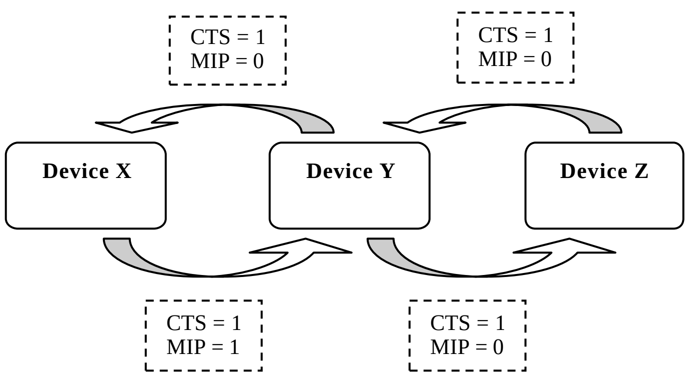


Figure 6-2: Transmitting a message using the MIP bit

3. Controlling the flow:

Device Y runs out of buffer space and lowers the outbound CTS to X to notify X of the same.

X receives that CTS and stops sending control packets, but keeps its MIP bit high to indicate to Y that it still has packets to send.

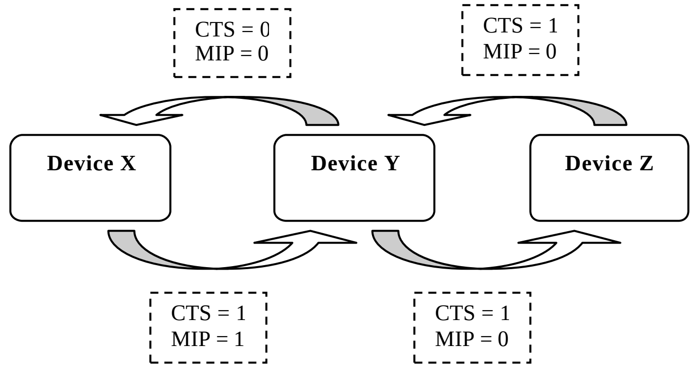


Figure 6-3:Controlling the flow using the CTS bit

Once device Y has emptied its buffers it must raise its outbound CTS bit to X to continue transmission as shown in **2** above. After X has completed its transmission it must lower its outbound MIP bit towards Y and return to the initial state **1** shown above.

In order for this protocol to function correctly, the following rules must be observed:

1. The protocol must be observed every time a control packet passes from a device to its adjacent device.

2. Each device must have the memory required to buffer at least twelve control packets per port at all times. As soon as the available buffer space drops below that, the device must lower its CTS to the sender. Analogously, when the available buffer space rises above twelve control packets, the device can raise its CTS again.

These rules together ensure that each recipient will have adequate time to stop a faster sender if it runs low on available receive buffer space.

6.4.2 Validity

Word 12	*Bit 29* Control Valid (CV)	*Bit 28* Control Data Valid (CDV)	*Bit 27* Joined with Next Valid Frame (JNVF)	*Bit 26* CRC Valid

Table 6-15: Validity Bits

This most significant nibble of word 12 determines whether certain parts of the packet are valid:

?? Bit 29 denotes whether the Control Message in word 48 is valid.

?? Bit 28 denotes whether the Control Data in words 51-53 is valid.

?? Bit 27 denotes whether there are more packets following this one as part of a multi-packet transmission. It is used when more control data than can fit into a single packet is to be transmitted. By setting this bit on all packets except the last one in the sequence, the sender can notify the recipient(s) of the same.

?? Bit 26 denotes whether the CRC defined in word 54 is valid.

These bits have been placed towards the beginning to notify hardware designers of the packet contents as early as possible to allow for efficient designs that can allocate necessary resources to process the packet.

6.4.3 Unused

Bit 25 of word 12 is currently unused.

6.4.4 Floating Point Format

Word	***Bit 24***
12	**Floating Point Format**

Table 6-16: Floating Point Format Bits

Bit 24 of word 12 indicates to the recipient that data in words 16-47 of the packet is formatted according to the IEEE 754 / 854 floating-point standard. When low those words are in standard 32-bit fixed-point format.

The fixed-point format has been chosen as the default to prevent expensive conversions that may be required since most commonly used CODECs do not support floating-point data.

6.4.5 Cable Number

Word	***B23 – B20***
12	**Cable Number**

Table 6-17: Cable Number Nibble

The cable number allows incoming MaGIC streams to be labeled while being multiplexed onto a high bandwidth medium such as a Gigabit Ethernet.

6.4.6 Sample Rate

Word	B19 – B16
12	Sample Rate

Table 6-18: Sample Rate Nibble

This nibble specifies the sample rate at which the packet is being transmitted across the network. The following table lists currently supported rates with corresponding values (to be set in the sample rate nibble of the packet):

Sample Rate (kHz)	Value
44.1	0x1
48 (default)	0x2
96	0x3
192	0x4
Reserved for future use	0x5 – 0xF

Table 6-19: Defined Sample Rates

The default sample rate is 48 kHz. All MaGIC devices are required to startup at that rate.

Increasing the sample rate to 96 kHz allows capable devices to send two samples per packet by reducing the number of audio channels to eight. Similarly, increasing the sample rate to 192 kHz allows capable devices to send four samples per packet by reducing the number of audio channels to four.

Individual devices may be capable of different sample rates. It is therefore necessary for the entire network to agree upon a universally supported sample rate. The protocol described in section 7.4 provides the procedure for modifying the network sample rate.

6.4.7 Frame Count / Timecode

Word	B31 - B28	B27 – B24	B23 - B20	B19 - B16	B15 – B12	B11 – B8	B7 – B4	B3 – B0
13	Frame Count / Timecode							

Table 6-20: Frame Count / Timecode Word

The configuration bits described in the next section determine the content of word 13. This word can either be used as a counter for the number of frames transmitted, or to store Timecode.

When used as a counter at 48 kHz, the number stored in this field will reset every 24.86 hours when it reaches the maximum 0xFFFFFFFF.

6.4.8 Frame Count / Timecode Configuration

Word	Bits 5-0
48	Frame Count/ Timecode Configuration

Table 6-21: Frame Count / Timecode Configuration Bits

Bits 0 and 1 of word 48 determine the content of word 13. The following table lists the options:

Configuration	Value
Frame Count	0x0
MaGIC Timecode	0x1
MIDI Timecode	0x2

Bits 2-5 store the frame rate for Timecode. The following table lists supported rates with the corresponding values:

Frame Rate (Hz)	Value
24	0x0
24.97	0x1
25	0x2
29.97	0x3
30	0x4

Table 6-22: Timecode Frame Rate Values

6.4.9 Copy Protection

In compliance with the Serial Copy Management System requirement specified by the 1992 U.S. federal copyright law, bit 6 of word 48 denotes whether the data is copy protected.

Upon receiving copy-protected data, a device must not retransmit it to any other ports.

6.4.10 Reserved

Bit 7 is currently unused.

7. System Timing Master

The System Timing Master (STM) is the device on a MaGIC network that:

- Provides the sample clock used by all devices to achieve synchronized data processing.

- Enumerates all devices by dynamically assigning them unique addresses.

Selection of the STM is automatic and transparent to the user.

The following diagram shows an STM device X, connected to a slave device Y. Device Y uses the recovered and regenerated sample clock for encoding and decoding frames.

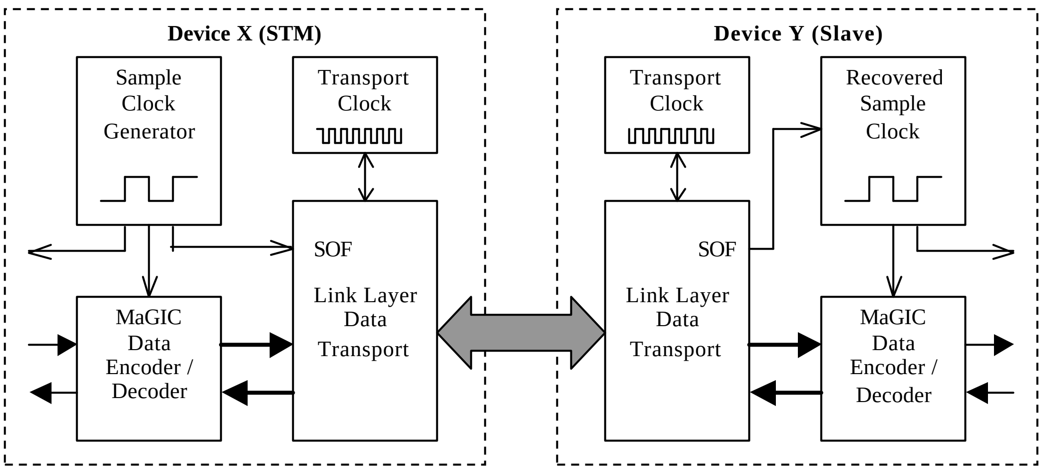


Figure 7-1: Sample and Transport Clock Relationship

7.1 Establishing the STM

The following rules are used to uniquely establish the STM:

1. *A device with only one A-port can never be the STM.*

2. *A device with only B ports will be the STM.*

3. *If all devices in the system contain both A and B ports, then the one device not connected on the A-port will be the STM.*

The following sections will continue to refer to these rules as Rule 1, Rule 2, and Rule 3 respectively.

7.2 Examples

The following figure shows a simple network with a Guitar (one A-port, no B ports) and an end-point Amplifier (no A-port, one B port). Rule 1 above disqualifies the Guitar from being the STM. Rule 2 uniquely identifies the end-point Amplifier as the network STM.

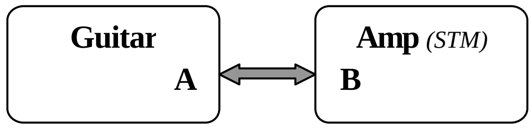


Figure 7-2: Establishing the STM using Rules 1 and 2

Consider below another simple network consisting of a Guitar (one A-port, no B ports), a Stomp Box (one A-port, one B port), and an Amplifier (one A-port, one B port). As in the previous example, Rule 1 disqualifies the Guitar and Rule 2 is not applicable. Rule 3 however, disqualifies the Stomp Box (because it is connected on the A-port) in favor of the Amplifier, which becomes the unambiguous STM.

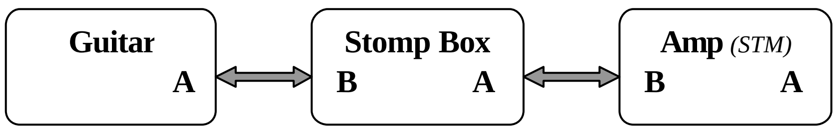


Figure 7-3: Establishing the STM using Rules 1 and 3

The following example consists of a Routing Hub (no A-port, and three B ports) connected to three devices. Again, Rule 1 disqualifies the Instruments and Rule 2 uniquely identifies the Routing Hub as the STM.

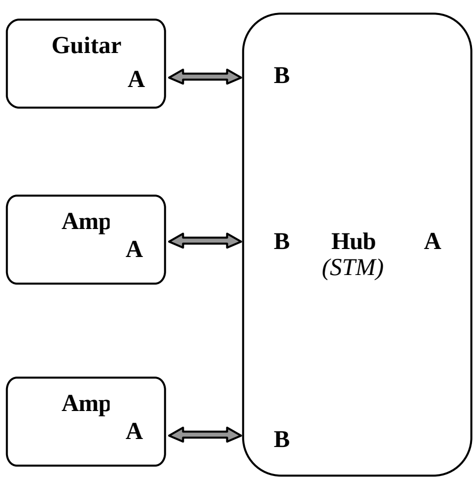


Figure 7-4: Establishing the STM with a Hub using Rules 1 and 2

This final example depicts a relatively complex MaGIC network. The application of Rules 1 and 3 in the same way shown above reveals the Mixer as the unambiguous network STM.

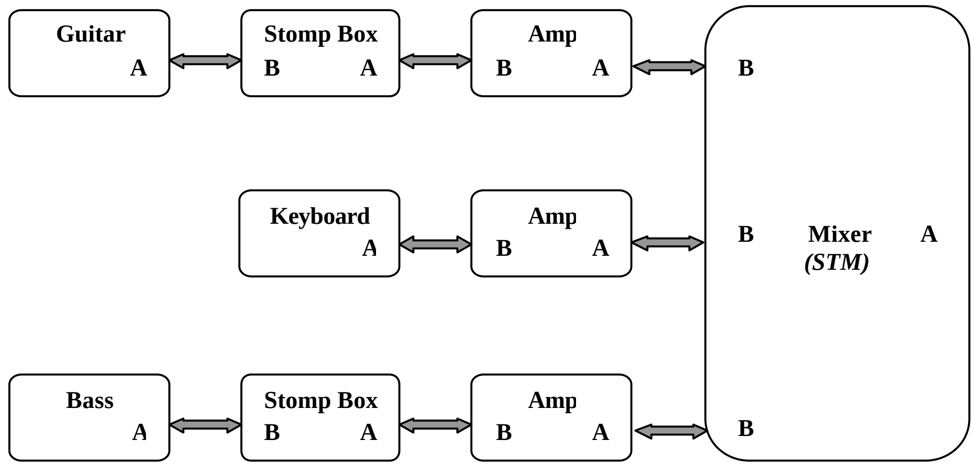


Figure 7-5: Establishing the STM with a Mixer (Hub) using Rules 1 and 3

As the examples above illustrate, all devices must be capable of assuming the role of STM except those with no B-port.

7.3 Device Enumeration

Enumeration is the process in which each device is assigned a unique 16-bit address by the STM. This limits the number of addresses in a MaGIC system to 65,536 (ranging from 0x0 to 0xFFFF). Three addresses are reserved for multicast and broadcast (discussed above in 6.3.3), and one is reserved for startup, leaving the remaining 65,532 addresses available for devices.

7.3.1 Startup

At startup, the STM assigns itself the *STM address* and all other devices assign themselves the *Non-STM startup address*. These addresses are listed in the table below:

Description	**Address**
Non-STM startup Address	0xFFFC
STM Address	0x0000

Table 7-1: STM and non-STM startup addresses

Each device must invalidate the audio using the Audio Valid word (discussed in section 6.2.1) at startup. Once enumeration has been completed, valid audio transmission can be resumed.

7.3.2 Algorithm

7.3.2.1 Control Messages

The following table lists the enumeration messages with corresponding Control Message and Control Data values.

Message	**Control Message**	**Control Data**
Enumerate Device	0x1	Next device address
Address Offset Return	0x2	Device address + 1
Request New Device Address	0x3	None
Reset Enumeration	0x4	None

Table 7-2: Enumeration Message Types

7.3.2.2 Initial Enumeration

1. Following power-up the STM initializes itself as address 0 and issues an *Enumerate Device* message on a connected port with Control Data set to the next address: 1.

2. The next device receives that packet, assigns itself the address 1, and retransmits the packet to the next device in the Daisy Chain with Control Data set to the next address: 2. The process continues until all devices are enumerated.

3. When an end-point is reached, that device must issue an *Address Offset Return* back to the STM with Control Data set to the next address.

4. Upon processing the *Address Offset Return* the STM knows that the network has been enumerated.

5. An STM with multiple B-ports should send the *Enumerate Device* packet down the next B-port only after the *Address Offset Return* packet has been received back from the first. Each daisy chain connected to each of the B-ports is thus enumerated in order.

6. A non-STM device with multiple B-ports must enumerate each daisy chain connected to its B-ports in order, but must only return a single *Address Offset Return* back on the A-port.

The following figure illustrates the sequence of events in the enumeration of a simple MaGIC network:

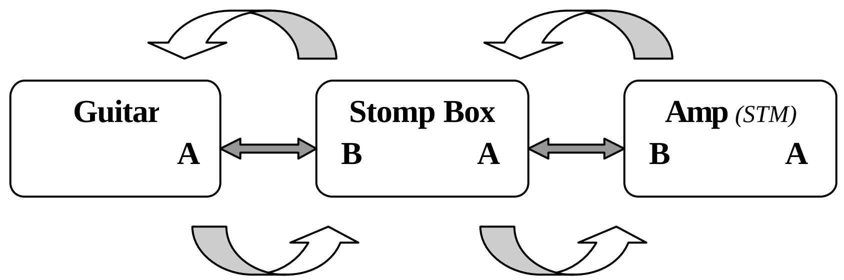


Figure 7-6: Enumeration in a simple MaGIC network

The following figure illustrates the sequence of events in the enumeration of an network with a device equipped with multiple B-ports:

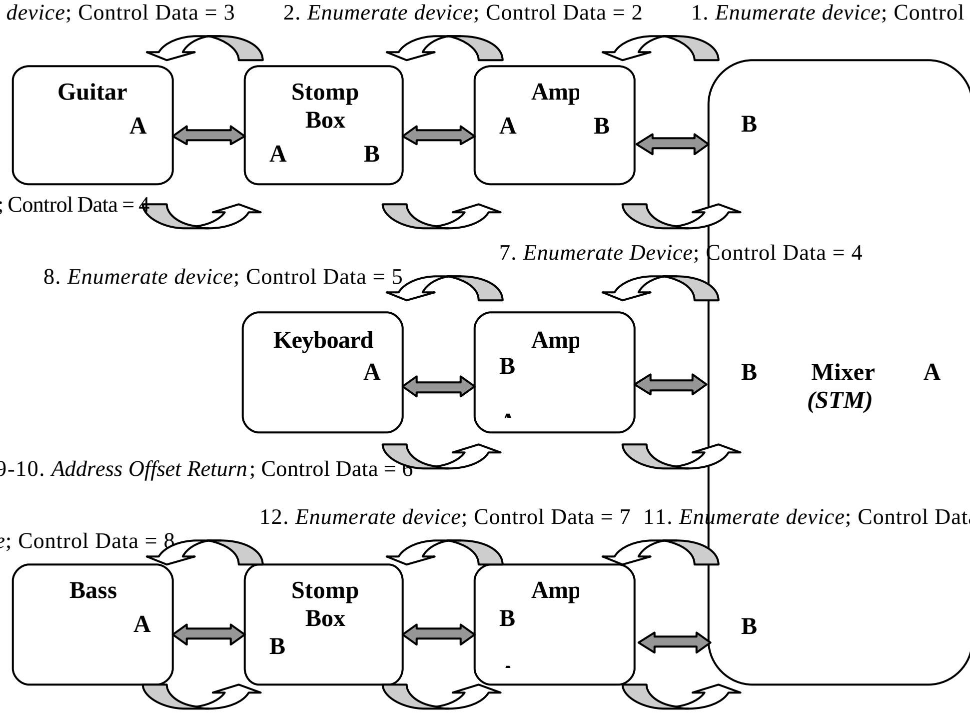


Figure 7-7: Enumeration in a complex MaGIC network

It is required that MaGIC devices broadcast their names on all ports using a *Return Device Name* control packet each time the device enumeration address is modified.

Appendix C contains the pseudo-code for the algorithm described above.

7.3.2.3 *Connecting Ports*

Ports can be connected or disconnected without disrupting the network. This requires the ability to dynamically select a new STM and re-enumerate the network when necessary.

7.3.2.3.1 Connecting the A-port:

1. If the device being connected on the A-port is the STM of its network, it must by Rule 3 relinquish that status by broadcasting a *Reset Enumeration* message to all devices connected to its B-ports.

2. Each device receiving this message must set its address to the startup value of 0xFFFC and forward the message onto any devices connected to its B-ports.

7.3.2.3.2 Connecting the B-port:

1. If the device being connected on the B-port is an STM, it will now be the STM of the new combined network. It must enumerate the network using the protocol described in section 7.3.2.2 above. If it is not the STM, it must issue a *Request New Device Address* to the STM to notify it of the newly connected devices.

2. Upon receiving that request, the STM must issue an *Enumerate Device* message with the Control Data set to the next available device address.

It is required that MaGIC devices broadcast their names using a *Return Device Name* control packet each time a new connection is made.

Appendix D contains pseudo-code for this algorithm discussed above.

7.3.2.4 *Disconnecting Two Arbitrary Ports*

Disconnecting an A-port and a B-port splits one network into two smaller ones. The device with the A-port becomes an STM by Rule 3. It must issue an *Enumerate Device* message to re-enumerate its network.

Appendix E contains pseudo-code for this algorithm discussed above.

7.3.2.5 *Re-enumeration*

Any device can at any time send a *Reset Enumeration* packet to the STM and request that the network be re-enumerated. The STM must first issue a *Reset Enumeration* packet on all ports to force all devices to set their address to the default startup value. It must then restart the enumeration process described in sections above.

7.4 Modifying the Network Sample Rate

7.4.1 Control Message Types

When a network is enumerated, packets are exchanged at the mandatory startup sample rate of 48 kHz. A device capable of a higher sample rate may then request an upgrade to a higher rate.

The following table lists the messages defined for use in this algorithm with corresponding Control Message field values.

Message	Control Message
Request Sample Rate	0x5
Acknowledge Sample Rate	0x6
Reject Sample Rate	0x7
Modify Sample Rate	0x8

Table 7-3: Sample Rate Modification Control Messages

7.4.2 Algorithm

1. Device issues *Request Sample Rate* to STM. The STM then forwards that through the whole network by sending it out on each B-port.

2. Each device processes the request and if it can support the requested rate, forwards it on. Otherwise, it returns a *Reject Sample Rate* to the STM. Upon receiving the rejection, the STM forwards it onto the device that issued the initial request and the process ends.

3. When the request reaches an end-point, that device must issue an *Acknowledge Sample Rate* to the STM.

4. Once the STM has received acknowledgements from the Daisy Chains connected to each B-port, it issues a *Modify Sample Rate* through the network. Each device processes this packet, updates its sample rate, and then forwards it onto the next device.

5. When the packet reaches an end-point, that device must return the packet back to the STM.

6. The STM receives the modification packets back from the daisy chains connected to each B-port. It can then determine if the network rate was successfully modified and can end the process.

If the STM receives another request for a sample rate modification while one is in progress it is permitted to discard that request. The responsibility for re-trying rests on the shoulders of the device issuing the request.

All audio must be muted while the sample rate change takes place. How that is done is application dependent and has therefore been left to the discretion of the implementer.

Appendix F contains the pseudo-code for the algorithm discussed above.

7.4.3 Incompatible Devices

If a new device connects to a network operating at a sample rate it does not support, it is required to indicate the problem to the user in a convenient manner and must not transmit any valid audio.

8. Control Protocol

8.1 Overview

A MaGIC network can be viewed as a collection of Components that are capable of controlling or being controlled by other Components, regardless of which physical devices they might be located on. This protocol provides a general mechanism for Components of a certain type to control other Components of the same type on a given network.

8.2 Components

A Component is defined as a unit on a MaGIC device that is capable of generating or interpreting a command. As a simple example, consider a knob (rotary encoder) on a device, and a volume on another device. This protocol would allow the knob to send a command to regulate the volume.

The following subsections discuss the main attributes of a Component in detail.

8.2.1 Attributes

8.2.1.1 Type

There are two types of Components:

Component Type	**Description**
Source	Component that can issue a command
Target	Component that can receive and execute a command.

Table 8-1: Component Types

8.2.1.2 Address

Each device must enumerate its Components and assign them unique unsigned integer addresses between 0 and 65,536. The combination of the 16-bit Device Address assigned during Enumeration (see section 7.3), and this 16-bit Component Address will uniquely identify any Component available on a network.

8.2.1.3 Name

Each Component must be assigned a mnemonic name limited to 16 characters to allow devices with displays to provide named-based access. MaGIC requires the use of 16-bit Unicode format for transmitting text.

8.2.1.4 Parameter Type

Each Component represents a specific parameter. In the example mentioned earlier in section 8.2, the parameter represented by the Source was the knob and the parameter represented by the Target was the volume.

The following table lists the currently defined types of parameters.

Parameter Type	Value
Scale	0x1
Toggle	0x2
MIDI	0x3
Blob	0x4
Reserved for future parameter types	0x5 – 0x3E8

Table 8-2: Control Link Types

Parameter Type is defined as a 16-bit value. It is expected that devices will define application-specific types as long as they do not use the values listed in the above table.

8.2.1.4.1 Scale

A Scale parameter ranges from a minimum to a maximum, and can be modified by a unit value.

In order to control it, the Source must supply the following values to the Target:

1. *Current*: present value of the scale
2. *Minimum*: lowest possible value of the scale
3. *Maximum*: highest possible value of the scale
4. *Unit*: minimum amount by which the scale can be incremented/decremented.

These values are required to be 32-bits each although they do not have to be of a specific type (integer, float, etc). MaGIC-compliant devices must ensure type-independent transmission of control data.

8.2.1.4.2 Toggle

A Toggle parameter is controlled by a single binary value. In order to control it, the Source must supply the following values to the Target:

1. *Current*: present value of the scale

The universal settings of 0 and 1 are used to denote OFF and ON respectively.

8.2.1.4.3 MIDI

A MIDI parameter is a generic type designed for supporting MIDI. By creating Source and Target Components of this parameter type, clients can transmit MIDI messages encapsulated in MaGIC control packets.

Each time a MIDI command is sent, the client must provide the command the number of bytes. No information is required at Component creation time.

8.2.1.4.4 Blob

A Blob parameter is a generic type designed to allow clients to transmit variable amounts of information. Creating Source and Target Components of this type and specifying the number of words to be transmitted is sufficient to deliver the data from the Source to the Target.

8.2.1.5 Control Links

A Control Link is a mapping between a Source and a Target that allows the former to control the latter by sending it control messages in a defined format. A Link can only be formed between a Source and Target of the same Parameter type (Scale with Scale, Toggle with Toggle, etc.).

A Control Link has two pairs of addresses that identify it:

1. Source Device Address and Source Component Address
2. Destination Device Address and Destination Component Address

Note that these map directly into words 49 and 50 of the MaGIC packet. (See sections 6.3.4 and 6.3.4).

8.2.2 Control Messages

The following table lists the Control Messages defined for exchanging information about Components.

Message	Control Message	Control Data 1	Control Data 2	Description
Request All Component Information	0x9	0	None or Parameter Type (table 8-2 above)	Requests information for all Components, or, Components of a specific Parameter type.
Request All Source Component Information	0x9	1	None or Parameter Type (table 8-2 above)	Requests information for all Sources, or, Sources of a specific Parameter type.

Request All Target Component Information	0x9	2	None or Parameter Type (table 8-2 above)	Requests information for all Targets, or, Targets of a specific Parameter type.
Return Component Information	0xA	See below	See below	Supplies information regarding a specific Component
Assign Control Link	0xB	None	None	Sent to a Source and a Target to inform them of a Control Link assignment
Send Control	0xC	See below	See below	Sent by a Source to modify a Target.

Table 8-3: *Component Control Messages*

8.2.3 Algorithm

8.2.3.1 Request Component Information

A device can request information about Components by issuing a *Request Component Information* message. Sending this message involves:

1. Setting the appropriate value in the Control Message field as shown in the table above.

2. Setting one of: 0, 1, or 2 to denote Sources and Targets, only Sources, and only Targets respectively, in the Control Data 1 field.

3. Setting either zero, or a valid Parameter Type (see Table 8-2 above) in the Control Data 2 field.

A device receiving such a message must issue a *Return Component Information* packet back to the sender for each Component that matches the restrictions specified in the Control Data 1 and Control Data 2 fields. For example, if Control Data 1 and 2 were to both contain zeros; this should result in sending a *Return Component Information* message for every single Component. If the values were 2 and 1 respectively, the message would be returned for Targets of type Scale only.

8.2.3.2 Return Component Information

Returning information about a specific Component essentially requires transmitting the current values of each of the attributes listed in section 8.2.1 above. This section describes how this information is encapsulated into a certain number of 32-bit words:

Regardless of Component type, the first two words (set in Control Data 1 and 2 respectively,) will have the following format:

Word Index	Bit Number	Description
0	22-31	Currently unused.
0	21	Component Type: Source or Target.
0	16-20	The number of characters in the Component name. The maximum is 16.
0	0-15	Parameter Type: see table 8-2 above.
1	16-31	Maximum Control Link count.
1	0-15	Current Control Link count.

Table 8-4: *First two words of Return Component Information*

The number of remaining words varies entirely based on the following three categories of data, which must be sent in the order listed below:

1. *Control Links*: For each control link, a word containing a Device address in bits 0-15 and a Component address in bits 16-31 must be included. See section 8.2.1.5 for details.

2. *Name*: For each character a 16-bit Unicode value must be included. Character 0 would occupy bits 0-15 of the first word. Character 1 would occupy bits 16-31, and so on. If the number of characters is odd, then the last 16 bits should be left unused.

3. *Parameter type-specific values*: A Scale parameter requires four 32-bit values (see section 8.2.1.4.1 above). A Toggle only requires one. Users defining their own parameter types must ensure that the values are easily represented in a collection of 32-bit words.

Therefore, the total number of 32-bit data words that have to be transmitted in order to accurately describe a Component is:

Total word count = 2 + Control Link Word Count + Name Word Count + Parameter Type Word Count

A control packet can only contain three 32-bit data words at once. If Total Word Count exceeds three, the words must be sent in separate control packets issued sequentially. The

Joined with Next Valid Frame (JNVF) bit allows packets to be marked as logically contiguous. See section 6.5.2 for details on this bit.

8.2.3.3 Assign Control Link

Any device can assign a Control Link between a Source and a Target on the network. The device making the assignment does not have to be the one with either the Source or the Target. If that is the case, the assigning device must issue the *Assign Control Link* message to both the Source and the Target.

There is no Control Data required for this packet. By setting the appropriate Source and Destination device address fields, the Source and Destination Component address fields, and of course the appropriate Control Message field, the assignment can be made.

8.3 Device and Network Name

Devices must define a mnemonic name. They may also optionally provide the user the option to store a mnemonic network name. The following messages allow devices to request and return these names across the network. Both names must be defined in 16-bit Unicode and have a maximum limit of 16 characters.

Message	Control Message	Description
Request Device Name	0xD	Requests the mnemonic network name.
Return Device Name	0xE	Returns the mnemonic network name.
Request Network Name	0xF	Requests the mnemonic network name.
Return Network Name	0x10	Returns the mnemonic network name.

Table 8-5: Control Messages for Network Name

Neither *Request Device Name* nor *Request Network Name* requires any control data.

Both *Return Device Name* and *Return Network Name* return names in a way similar to the one described above for *Return Component Information*. The format is as follows:

1. *Data word 0*: contains the number of characters in the name to follow.
2. *Data word 1 onwards*: contains 16-bit Unicode values for each character. Character 0 occupies bits 0-15. Character 1 occupies bits 16-31, and so on. If the number of characters is odd, then the last 16 bits should be left unused.

A control packet can only contain three 32-bit data words at once. If the number of words required exceeds three, they must be sent in separate control packets issued sequentially.

The Joined with Next Valid Frame (JNVF) bit allows packets to be marked as logically contiguous. See section 6.4.2 for details.

Appendix A: Cyclic Redundancy Check

The following is an example of a CRC-32 generation function written in C:

```
/*
 * crc32h.c -- package to compute 32-bit CRC one byte at a time using
 *             the Big Endian (highest bit first) bit convention.
 *
 * Synopsis:
 *   void gen_crc_table (void):
 *      Generates a 256-word table containing all CRC remainders for
 *      every possible 8-bit byte. It must be executed (once) before
 *      any CRC updates.
 *
 *   unsigned update_crc (unsigned long crc_accum, char *data_blk_ptr,
 *                        int data_blk_size):
 *      Returns the updated value of the CRC accumulator after
 *      processing each byte in the addressed block of data.
 *
 * It is assumed that an unsigned long is at least 32 bits wide and
 * a char occupies one 8-bit byte of storage.
 *
 * The generator polynomial used for this version of the package is
 * x^32+x^26+x^23+x^22+x^16+x^12+x^11+x^10+x^8+x^7+x^5+x^4+x^2+x^1+x^0
 * as specified in the Autodin/Ethernet/ADCCP protocol standards.
 * Other degree 32 polynomials may be substituted by re-defining the
 * symbol POLYNOMIAL below. Lower degree polynomials must first be
 * multiplied by an appropriate power of x. The representation used
 * is that the coefficient of x^0 is stored in the LSB of the 32-bit
 * word and the coefficient of x^31 is stored in the most significant
 * bit. The CRC is to be appended to the data most significant byte
 * first. For those protocols in which bytes are transmitted MSB
 * first and in the same order as they are encountered in the block
 * this convention results in the CRC remainder being transmitted wit
 * the coefficient of x^31 first and with that of x^0 last (just as
 * would be done by a hardware shift register mechanization).
 *
 * The table lookup technique was adapted from the algorithm described
 * in Byte-wise CRC Calculations, Avram Perez, IEEE Micro 3, 4(1983).
 */

#define POLYNOMIAL 0x04c11db7L

static unsigned long crc_table[256];

void gen_crc_table()
  /*
   * Generate the table of CRC remainders for all possible bytes:
   */
{
        register int i, j;
        register unsigned long crc_accum;

        for (i = 0; i < 256; i++) {
          crc_accum = ( (unsigned long) i << 24 );
```

```
          for (j = 0; j < 8; j++) {
            if (crc_accum & 0x80000000L)
                crc_accum = ( crc_accum << 1 ) ^ POLYNOMIAL;
            else
                crc_accum = ( crc_accum << 1 );
          }

          crc_table[i] = crc_accum;
        }

        return;
}


unsigned long update_crc(unsigned long crc_accum, char *data_blk_ptr,
                         int data_blk_size)
  /*
   * Update the CRC on the data block one byte at a time
   */

{
        register int i, j;

        for (j = 0; j < data_blk_size; j++) {
          i = ( (int) ( crc_accum >> 24) ^ *data_blk_ptr++ ) & 0xff;
          crc_accum = ( crc_accum << 8 ) ^ crc_table[i];
        }

        return crc_accum;
}
```

Appendix B: Control Message Quick-reference

The following table lists all defined control messages, their corresponding values, and a brief description of their purpose.

Control Message	Name	Description
0x0	N/A	Invalid.
0x1	Enumerate Device	Used by the STM to enumerate other devices on the network.
0x2	Address Offset Return	Used by an end-point device to notify the STM of end of Daisy Chain.
0x3	Request New Device Address	Used by an end-point device to request the STM to enumerate newly connected devices.
0x4	Reset Enumeration	Used by an STM relinquishing its control due a new network connection.
0x5	Request New Sample Rate	Used by a device broadcasting a request for a sample rate change.
0x6	Acknowledge New Sample Rate	Used by a device to notify the STM that it is capable of the requested sample rate.
0x7	Reject New Sample Rate	Used by a device to notify the STM that it is not capable of the requested sample rate.
0x8	Modify New Sample Rate	Used by the STM to notify all devices of a change in the network sample rate.
0x9	Request Component Info	Used by any device that wishes to gather information on the Components available on a network.
0xA	Return Component Info	Used by a device to provide information about its Components.
0xB	Assign Control Link	Used to assign a Control Link between a Source and a Target.
0xC	Remove Control Link	Used to remove a Control Link between a Source and a Target.
0xD	Send Command	Used by a Source to set the Target to a new value.
0xE	Request Device Name	Used to request the mnemonic device name from another device on the network
0xF	Return Device Name	Used to return the device name.
0x10	Request Network Name	Used by a device to request the mnemonic network name from another device on the network
0x11	Return Network Name	Used by a device to return the network

		name.
0xFFFF	N/A	Reserved

Appendix C: Enumeration Pseudo-code

The following pseudo-code describes the initial network enumeration algorithm discussed in section 7.3.2.2.

General constants:

```
ENUMERATE_DEVICE              = 0x0001;
ADDRESS_OFFSET_RETURN         = 0x0002;
REQUEST_NEW_DEVICE_ADDRESS    = 0x0003;
RESET_ENUMERATION             = 0x0004;

STM_ADDRESS       = 0x0000;
STARTUP_ADDRESS   = 0xFFFC;
BROADCAST_ADDRESS = 0xFFFF;
```

STM Device Enumeration:

```
Device.address           = STM_ADDRESS;
Device.nextDeviceAddress = Device.address + 1;

SEND_MESSAGES: For each B Port {
      SendMessage(Destination address = STARTUP_ADDRESS,
                  Source address  = Device.address,
                  Control message = ENUMERATE_DEVICE,
                  Control data 1  = Device.nextDeviceAddress);

      Message aor = Get Address Offset Return message;
      Device.nextDeviceAddress = aor.controlData1;
}
```

Non-STM Device Enumeration:

```
Device.address = STARTUP_ADDRESS;

Message ed = Get the Enumerate Device message;
Device.address = ed.controlData1;
Device.nextDeviceAddress = ed.controlData1 + 1;

Goto SEND_MESSAGES;

SendMessage(Destination address = ed.sourceAddress,
            Source address  = Device.address,
            Control message = ADDRESS_OFFSET_RETURN,
            Control data 1  = Device.nextDeviceAddress);
```

Appendix D: Port Connection Pseudo-code

The following pseudo-code describes the arbitrary port connection algorithm discussed in section 7.3.2.3.

General Constants: see Appendix C above

New connection on the A-port or Processing a Reset Enumeration Message:

```
if (Device.address = STM_ADDRESS) {
      Device.address = STARTUP_ADDRESS;

      For each B Port {
            SendMessage(Destination address = BROADCAST_ADDRESS,
                        Source address      = Device.address,
                        Control message     = RESET_ENUMERATION);
      }
}
```

New connection on the B port:

```
if (Device.address = STM_ADDRESS) {
      Follow the STM Device Enumeration algorithm in Appendix C above
}

else if (Device.address != STM_ADDRESS
            && Device.address != STARTUP_ADDRESS) {
      SendMessage(Destination address = STM_ADDRESS,
                  Source address      = Device.address,
                  Control message     = REQUEST_NEW_DEVICE_ADDRESS);
      }
}
```

Processing a Request New Device Address Message:

```
Message rnda = Get the Request New Device Address Message;
SendMessage(Destination address = STARTUP_ADDRESS,
            Source address  = Device.address,
            Control message = REQUEST_NEW_DEVICE_ADDRESS,
            Control data 1  = Device.nextDeviceAddress);

Message aor = Get Address Offset Return message;
Device.nextDeviceAddress = aor.controlData1;
```

Appendix E: Port Disconnection Pseudo-code

The following pseudo-code describes the arbitrary port disconnection algorithm discussed in section 7.3.2.4.

General Constants: see Appendix C above

Disconnection on the A-port:

```
if Device is capable of being an STM {
      Device.address = STARTUP_ADDRESS;

      For each B Port {
            SendMessage(Destination address = BROADCAST_ADDRESS,
                        Source address      = Device.address,
                        Control message     = RESET_ENUMERATION);
      }

      Follow the STM Device Enumeration algorithm in Appendix C above;
}
```

Appendix F: Sample Rate Modification Pseudo-code

The following pseudo-code describes the arbitrary port disconnection algorithm discussed in section 7.4.2.

General Constants and Global Variables: see Appendix C above

```
MSR_REQUEST            0x0005
MSR_ACKNOWLEDGE        0x0006
MSR_REJECT             0x0007
MSR_MODIFY             0x0008
```

Issuing the request from an arbitrary device to STM:

```
SendMessage(Destination address = STM_ADDRESS,
            Source address  = Device.address,
            Control message = MSR_REQUEST,
            Control data 1  = Device.higherSampleRateCode);
```

Processing the request by STM:

```
Message msr = Get the Modify Sample Rate message;

If STM is capable of the sample rate specified by msr.controlData1 {
      On each B-port {
            SendMessage(Destination address = BROADCAST_ADDRESS,
                        Source address  = Device.address,
                        Control message = MSR_REQUEST,
                        Control data 1  = msr.controlData1);
      }
}

else Terminate the sample rate modification process.
```

Processing of the modify sample rate message sent by the STM and received by each device on the A-port:

```
Message msr = Get the Modify Sample Rate message from A-port;

If device is capable of the sample rate specified by msr.controlData1 {
      if Device has no connected B ports, then on A-port {
            SendMessage(Destination address = msr.sourceAddress,
                        Source address  = Device.address,
                        Control message = MSR_ACKNOWLEDGE,
                        Control data 1  = msr.controlData1);
      }

      else on all B-ports {
            SendMessage(Destination address = BROADCAST_ADDRESS,
                        Source address  = Device.address,
                        Control message = MSR_REQUEST,
                        Control data 1  = msr.controlData1);
      }
}
```

```
else on A-port {
      SendMessage(Destination address = STM_ADDRESS,
                  Source address  = Device.address,
                  Control message = MSR_REJECT,
                  Control data 1  = msr.controlData1);
}
```

Processing of the acknowledge sample rate message received by each non-end point device on the B-port:

```
Message msr = Get the Modify Sample Rate message from B-port;

If acknowledge message has been received from all other B-ports {
      SendMessage(Destination address = BROADCAST_ADDRESS,
                  Source address  = Device.address,
                  Control message = MSR_ACKNOWLEDGE,
                  Control data 1  = msr.controlData1);
}
```

Processing of the acknowledgements and/or rejections by the STM:

```
Message msr = Get the Modify Sample Rate message;

If msr.controlMessage == MSR_REJECT {
      Terminate the sample rate modification process.
}

else if msr.controlMessage == MSR_ACKNOWLEDGE && the same
acknowledgement has been received from all other B-ports {
      From this time forward set the audio valid bits for all packets
            to zero;

      SendMessage(Destination address = BROADCAST_ADDRESS,
                  Source address  = Device.address,
                  Control message = MSR_MODIFY,
                  Control data 1  = msr.controlData1);
}
```

Processing of the modify sample rate message sent by the STM and received by each device on the A-port:

```
Message msr = Get the Modify Sample Rate message with controlMessage =
MSR_MODIFY;

From this time forward set the audio valid bits for all packets to 0.

Configure the device to operate with the new sample rate specified by
msr.controlData1;

if Device has no B ports {
      Send the same message ‘msr’ back onto the A-port.

      Set the audio valid bits to 0xFFFF and start transmitting audio
      at the new sample rate.
}

else Send the same message ‘msr’ out as-is on all B ports.
```

Processing of the modify sample rate message sent by the STM, after it has been through the Daisy Chain once and is received by each device on a B port:

```
Message msr = Get the Modify Sample Rate message with controlMessage =
MSR_MODIFY;

if Device has multiple B ports and this message has been received back
from all other B ports {
      Forward the same message ‘msr’ back onto the A-port.

      Set the audio valid bits to 0xFFFF and start transmitting audio
      at the new sample rate.
}
```

Appendix G: References

For a thorough understanding of MaGIC it is recommended that the following documents be reviewed:

1. IEEE 802.3 – 1998, Part 3: Carrier-sense multiple-access with collision detection (CSMA/CD) access method and physical layer specifications.

2. MIDI 1.0 Detailed Specification, Version 96.1 – March 1996

3. Byte-wise CRC Calculations, Avram Perez, IEEE Micro 3, 4 (1983).

4. Copyright Law of the United States of America. 1992. http://www.loc.gov/copyright/title17/92chap10.html