

AoEr10

AoE (ATA over Ethernet)

Status of this Memo

This document specifies a lightweight protocol for accessing an Ethernet attached ATA device.

1. Introduction

AoE is used to achieve a very basic level RPC mechanism between a client and an ATA device server. The server accepts commands and generates responses based on a command code in the AoE header.

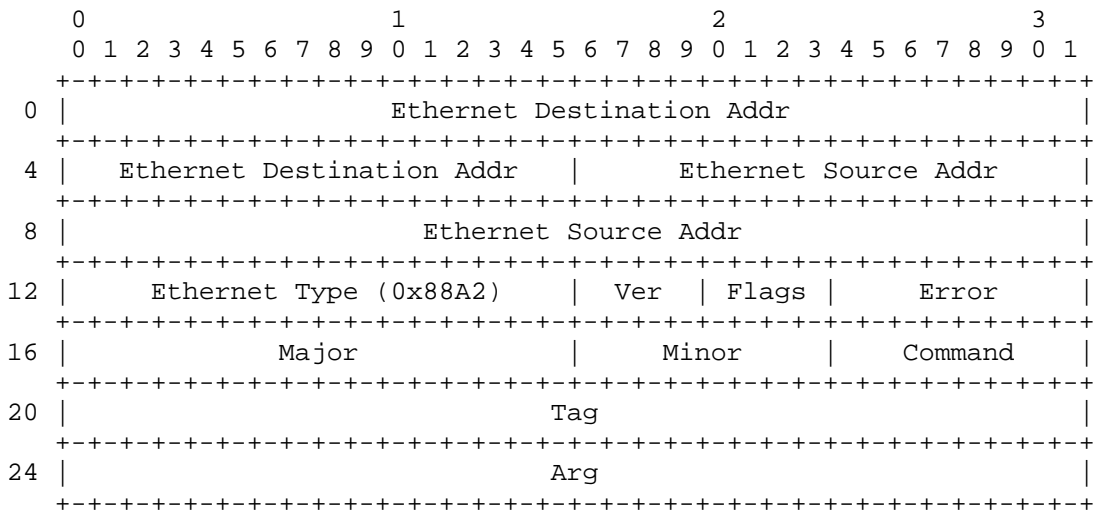
This document describes the format of the AoE header and the command set for protocol version one. All values for which a byte order is applicable are in network byte order. Reserved fields should be set to zero in all messages sent.

Each message contains a header followed by an argument field. The format of the argument field is defined based on the header command code.

1.1. Connections

AoE is not a connection based protocol. Each message sent to a server should be considered unique and unreliable.

2. AoE Header Format



2.1. Ethernet Source, Destination Addresses & Ethernet Type

The standard Ethernet MAC header for IEEE 802.3 Ethernet frames.

AoE has an registered Ethernet type of 0x88A2.

2.2. Ver (Version)

The version field defines the AoE header format as well as a set of command codes. All future versions must contain this field in the header, in this precise location. This document describes version 1. Section 2.6 defines the command codes for this version.

The Query Config Information response includes the version number a server supports.

2.3. Flags

The Flags field contains bitwise flags defined as follows:

```
+---+---+
|R|E|Z|Z|
+---+---+
```

The R bit is set if the message is a response. The E bit is set in a response message if the associated command message generated an AoE protocol error. The Z bits are reserved and must be set to zero.

2.4. Error

If Flags bit E is set, this field contains an error code defined as follows:

Error 1: Unrecognized command code
The server does not understand the code in the Command field.

Error 2: Bad argument parameter
An improper value exists somewhere in the Arg field.

Error 3: Device unavailable
The server can no longer accept ATA commands.

Error 4: Config string present
The server cannot set the config string because it is non-empty (see section 3.2).

Error 5: Unsupported version
The server does not understand the version number specified in Ver.

2.5. Major, Minor

Each AoE server possesses a major and minor address. Before processing the header Command the server must validate its major and minor address with the Major and Minor fields in the header. A server will accept a command message for processing if the following two tests are true:

The Major field in the header is the server major address or all ones (0xffff).

The Minor field in the header is the server minor address or all ones (0xff).

Any command messages failing either of these two tests must be ignored by the server.

The server must supply its major and minor address in every response, even if the corresponding request had all-ones values.

2.6. Command

This field contains the command code for the message. The following command codes are defined for this protocol version:

Command 0: Issue ATA Command

Command 1: Query Config Information

Command codes 240-255 (0xf0-0xff) are reserved for vendor specific use.

2.7. Tag

The Tag field permits a client with the means to correlate responses with their appropriate commands. It is copied into the response message by the server and is otherwise ignored.

2.8. Arg

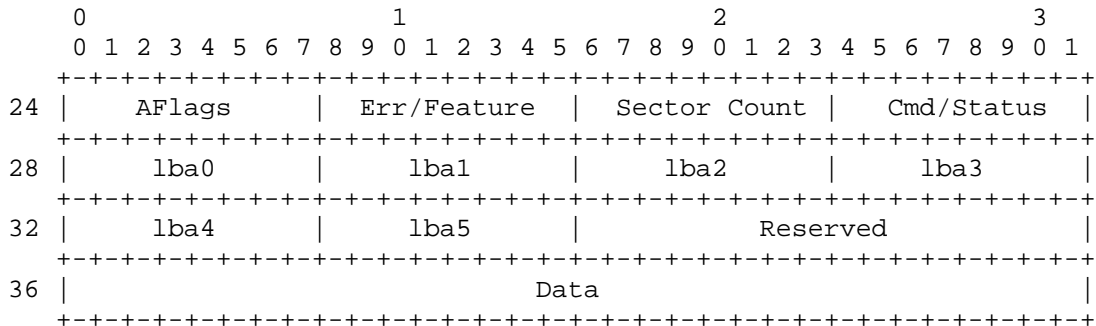
The Arg field contents serve as input for the specified command code.

3. Command Codes

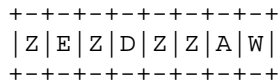
3.1. Command 0, Issue ATA Command

Command 0 is used to issue an ATA command to an attached ATA device. Any data associated with a command must fit into a single message. Using standard 1520 byte Ethernet frames will limit a device read/write to a maximum of two sectors. Servers are not required to understand the ATA command they issue.

The AoE Arg field is formatted as follows:



AFlags is defined as follows:



The W bit should be set when the ATA command requires data to be written to the device. The A bit should be set when a write request is to be done asynchronously. The D bit is as defined in the ATA Device/Head register and is only evaluated when the E bit is set. The E bit defines this command as an LBA48 extended command. The Z bits are reserved and must be set to zero.

The W bit and the Sector Count field together determine whether data is to be transferred to or from the device:

If data is to be written to the device, the W bit must be set and Data must contain Sector Count * 512 bytes of data.

If data is to be read from the device, the W bit must be cleared and Sector Count must specify the number of sectors to be read from the device. If the command succeeds, the Data field in the response message will contain the data read.

If no data is to be transferred, Sector Count must be zero and the W bit is ignored.

If both the A and W bits are set, the server may cache the write request in memory and respond immediately, returning the Arg field unchanged. The server may issue the hardware request whenever convenient provided a subsequent read of the same sector returns the cached data. If the server experiences a power failure the data in the server's write cache may be lost. No response is sent when the write request completes, even if an error occurred.

If the E bit is set, the server will transfer the contents of each field to the ATA device registers as follows:

```

Device          <- (AFlags & 0x50 | 0xA0)
LBA Low         <- lba3
LBA Low         <- lba0
LBA Mid         <- lba4
LBA Mid         <- lba1
LBA High        <- lba5
LBA High        <- lba2
Sector Count    <- 0
Sector Count    <- Sector Count
Err/Feature     <- Err/Feature
Cmd/Status      <- Cmd/Status

```

AFlags bits E,D correspond to Device register bits L,D. They are or'd with 0xA0 to force obsolete bits to 1 as per the ATA spec.

If the E bit is not set, the server will transfer the contents of each field to the ATA device registers as follows:

```

Device          <- lba3
LBA Low         <- lba0
LBA Mid         <- lba1
LBA High        <- lba2
Sector Count    <- Sector Count
Err/Feature     <- Err/Feature
Cmd/Status      <- Cmd/Status

```

Excepting asynchronous writes, a response is not generated until the ATA command has completed, whether by success or by failure. Upon ATA command completion, the ATA device registers are copied into each field of the response as follows:

Err/Feature	<- Err/Feature
Sector Count	<- Sector Count
lba0	<- LBA Low
lba1	<- LBA Mid
lba2	<- LBA High
Cmd/Status	<- Cmd/Status

This sequence is the same whether the E bit is set or unset.

If the E bit is set, the following ATA device registers will additionally be copied into the lba3, lba4, and lba5 fields after setting the HOB bit in the ATA Device Control register.

lba3	<- LBA Low
lba4	<- LBA Mid
lba5	<- LBA High

All remaining fields in the response retain the values from the corresponding command.

3.2. Command 1, Query Config Information

Command 1 retrieves configuration information from the server and in certain cases sets it. The command and response Arg fields are formatted as follows:

	0								1								2								3															
	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
24	Buffer Count								Firmware Version																															
28	Sector Count				AoE				CCmd				Config String Length																											
32	Config String																																							

Buffer Count

The maximum number of outstanding messages the server can queue for processing. Messages in excess of this value are dropped.

Firmware Version

The version number of the server firmware.

Sector Count

If nonzero, this field specifies the maximum number of sectors the server can handle in a single ata command request. A value of zero is equivalent to a value of two for backwards compatibility.

AoE

The AoE protocol version the server supports.

CCmd

Config string query / set subcommand

Config String Length

The length of the following config string.

Config String

The server configuration string, of maximum length 1024.

In a command message the fields Buffer Count, Firmware Version, and AoE should be set to zero by the client and ignored by the server. The remaining fields may be used to query and set the server's config string.

The proper values for CCmd are as follows:

CCmd 0: read config string

Read the server config string without performing any test and respond.

CCmd 1: test config string

Respond only if the argument string exactly matches the server configuration string.

CCmd 2: test config string prefix

Respond only if the argument string is a prefix of the server configuration string.

CCmd 3: set config string

If the current server config string is empty, set the server config string to the argument string and respond. If the current server config string is not empty, return a response with Flags bit E set and Error set to 4.

CCmd 4: force set config string

Set the server config string to the argument string and respond.

In the response the server must supply its current values for all fields.

When a server boots and is ready to process commands it should broadcast a Query Config Information response message with a tag of zero.

Authors' Address:

The Brantley Coile Company, Inc.
220 College Ave. 7th Floor
Athens, GA 30601

{sah,brantley}@borf.com

Acknowledgements:

A special thanks to Norman Wilson for many proof readings and suggested clarifications.