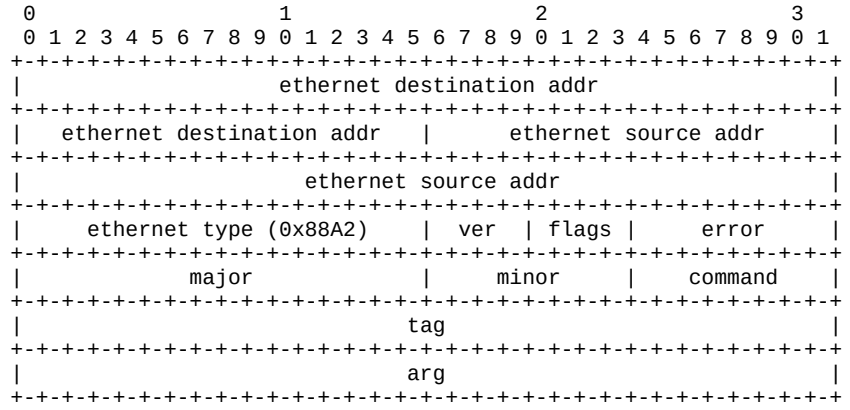
ATA over Ethernet (AoE)

Introduction. AoE is used to achieve a very basic level RPC mechanism between an initiator and a target. The target 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. All values for which a byte order is applicable are in network byte order. Reserved fields must be set to zero in all messages sent.

Common Header Format. An AoE transaction consists of a request message made by an *initiator* sent to a *target*, where it is processed and a response message returned to the initiator using the source address of the request message. Each message contains a header followed by an argument field. The format of the argument field is defined based on the header command code.

AoE is not a connection based protocol. Each message sent to a target should be considered unique and unreliable. It's up to the initiator to resend messages that have gone astray.



AoE Header Format

The messages are carried in Ethernet frames, one message per frame. We include the Ethernet header as part of our definition. AoE has an registered Ethernet type of 0x88A2. The version, *ver* 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. The Query Config Information response includes the version number a target supports.

The *flags* field is four bits. Bit 3 is set when the message is a response. Bit 2 is set in a response message if the associated command message generated an AoE protocol error. The remaing bits are zero, and reserved.

If Flags bit E is set, the *error* field contains an error code defined as follows. A value of 1 means the command is unrecognized. A value of 2 means an improper value exists somewhere in the *args* field. A value of 3 means the target can no longer accept ATA commands. A value of 4 means the target cannot set the config string because it is non-empty. (See Query/Config.) A value of 5 means the target does not understand the version number in *ver*. And a value of 6 means the command can not be completed because the target is reserved.

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

The major field in the header is the target's major address or all ones. (0xffff).

The minor field in the header is the target's minor address or all ones. (0xff).

Any command messages failing either of these two tests must be ignored by the target. The target must supply its major and minor address in every response, even if the corresponding request has a value of all ones.

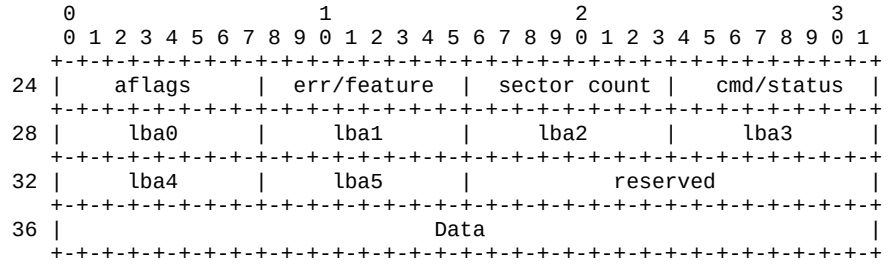
Command. This field contains the command code for the message. The following command codes are defined for this protocol version: 0: issue ATA command, 1: query/config command, 2: MAC mask list command, 3: reserve/release command. Command codes 240-255 are reserved for vendor specific use.

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

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

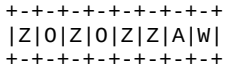
ATA Command (code 0) Command 0 is used to issue an ATA command to an attached ATA device. Any data associated with a command must fit into a single message. Using standard 1520 byte Ethernet frames will limit a device read/write to a maximum of two sectors. Using jumbo frames of 9000 can fit eight sectors into the frame.

ATA reads/writes to a target must be gated by the reserve list. If a target cannot be accessed due to reservation restriction, AoE error 6 must be returned.



ATA Command Format

AFlags is defined as follows:



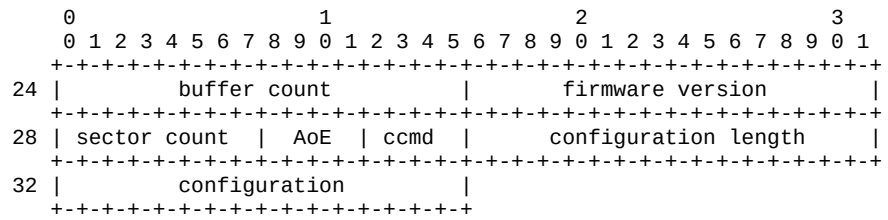
The W bit must be set when the ATA command requires data to be written to the device. The A bit must be set when a write request is to be done asynchronously. The O bits are obsolete and are ignored. The Z bits are reserved and must be set to zero.

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 suc-

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

The target will behave as if it has transfered the contents of each field corresponding to the ATA registers as implied by the register names. The response should contain values as if the ATA operation has taken place. For example, *cmd* is the ATA command on the transaction and the same field in the response is the *status*. Note that the *sector count* register is only one byte, not two as in the ATA spec.

Query/Config (command 1) Query/Config provides a mechanism for locating a target. It serves the function of Address Resolution Protocol for IP, with the additional beneif that it can query targets who have been configured to be used by specific initiators. A 1,024 byte string can be set, queried and cleared to provide this function.



Query/Config Command Format

Buffer count is the maximum number of outstanding messages the target can queue for processing. Messages in excess of this value may be dropped. *Firmware version* is the version of the firmware in the target. *Sector count*, if non-zero, is the maximum number of sectors the target can handle in a single ATA request. A value of zero is equivalent to a value of two. *AoE* is 1. *Ccmd* is the particular query/config command to execute (see below). *Configuration length* is the number of bytes in the configuration buffer. *Configuration* is a number of 8-bit bytes

The configuration is a sequence of 8-bit bytes. While it is common to have them be UTF-8 strings, they are not limited to displaying text.

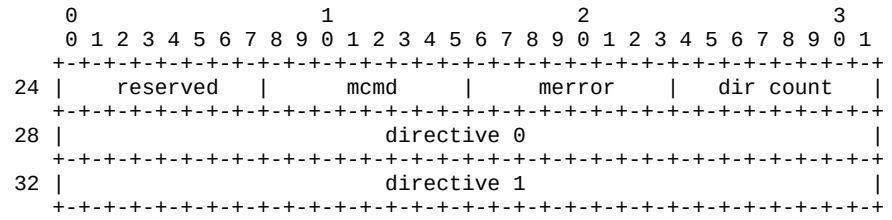
On a request message, the fields buffer count, firmware version and AoE must be set to zero by the initiator and are ignored by the target. When an initiator sends a query/config request message to a target, the ccmd code determines its behaviour.

A **ccmd of 0** causes the target to return the current configration. A **ccmd of 1** will cause the target to response to the request only if the configuration in the request message completely matches the target's configuration. The target will respond to a **ccmd of 2** if the request's configuration is a valid prefix of the target's configuration. In other words, all the bytes in the request message configuration must match the bytes in the target's configuration, but doesn't have to be as long as the target's configuration. This can be used to find subsets of configurations in a network of targets.

Two commands set the configuration. **Ccmd of 3** sets the configuration of a target **only if** the target configuration length is currently zero. This is an atomic command. If two initiators try to set the configuration of a single target, only one will succeed. **Ccmd of 4** will force the configuration on the target to be set unconditionally.

Ethernet Address Fencing (command 2). Normally a target will respond to any initiator's request. However, a target can limit the intiators to which it will respond, causing all other initiators to be fenced off. This is done by setting an Ethernet address mask list. When this list is non-empty the target will only respond to the request messages from an Ethernet address in the mask list. All otheres are maked out.

The Ethernet Address Fencing command has a 32 bit header followed by a number of address directives.

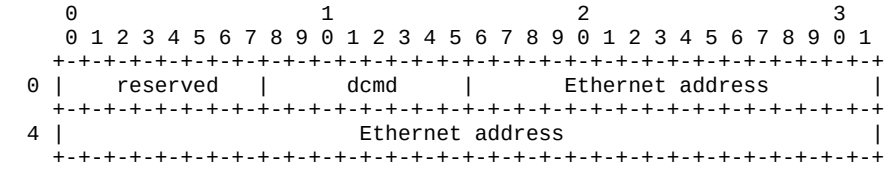


Ethernet Address Fenching Command

Mcmd informs the target if the request is to read the list, in which case the value is 0, or edit the mask list, in which case the value will be a 1. The *merror* byte will be non-zero in the case of an error. A value of 1 indicates that something unspecified was wrong. The value 2 indicates that the *dcmd* value was unknown and an merror of 3 indicate the list is full and a directive tried to install a new address.

Dir count contains the number of directive entries in a request and the number of addresses in a response.

Each Ethernet address directive is eight bytes.



Ethernet Address Directive

Dcmd is the directive command; 0 means empty directive entry, 1 indicates an Ethernet mask should be added to the list, and a 2 indicates the Ethernet address should be deleted.

On a request message with mcmd set to zero, the initiator will return a list of directive entries with the Ethernet addresses filled in.

The target will process the list of directives for a request messages if the *mcmd* field is a 1. Targets will evaulate the directives in the order they appear in the request. If an error is incountered the target should cease processing the request and return a response with the *dir count* field set to the directive that caused the error, and the *merror* field set to the cause of the error.

If no errors occurred in processing the directive list, the target must return the Ethernet address list in the response, including setting *dir count* field to the number of addresses. It is not an error to add an Ethernet address to a list more than once. Similarly, it is not an error to remove an Ethernet address from a list that does not contain the address. Targets must ignore duplicate additions and deletions.

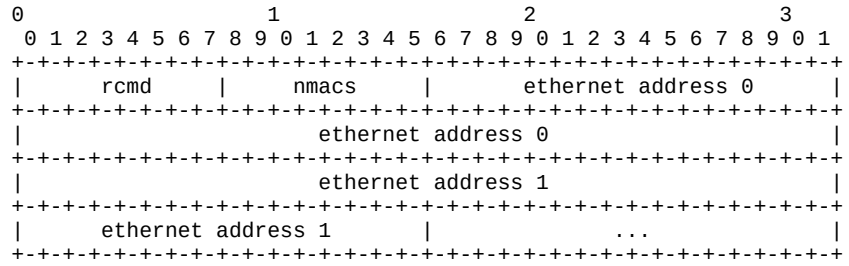
There is no provision for reading a list containing more than 255 entries. Targets supporting this command are recommended to limit their mac mask lists to 255 entries for clarity. A target is not

required, however, to support 255 mask list entries.

It should be noted that if an initiator adds addresses to a list and the initiator is not included in the list, the initiator will no longer be able to communicate with the target. It is recommended that any modification of the list include the initiator source Ethernet address as the first directive.

Target Reservations (command 3) Some initiators would like to share a target. To do this a means needs to be provided to coordinate access between initiators accessing the same target. Reservations are a way of doing that.

When reserved, an AoE target will only process ATA read and write commands if the source mac address in the command frame is in the reserve Ethernet address list; other commands (query config, ata ident, etc) are processed regardless of the source mac address. When a target cannot be accessed due to target reservation, an AoE error is returned with an error code of 6: Target is reserved.



Reservation Command Format

All operations return the current Ethernet address list of initiators allowed to do reads and writes on the target. As expected, *rcmd* contains the operation required of the target. A *rcmd* value of **0** is a no-op and is used to retrieve the current address list. *Rcmd* **1** sets the current list to the set of addresses in the request message and removes the previous list. A reservation can be removed by setting the target to an empty list. To make the the reservation atomic, only initiators in the reservation list can set the list. If two initiators try to set the list at the same time, only one will succeed. The other will get AoE error 6.

A host can force set the reservation list with a *rcmd* command of **2**. It's primary use is to modify a stale reserve list.

Nmacs has the count of Ethernet addresses in a message.