

Active Load Sharing Team Performance

Please make the list below jump to its corresponding headings

1. [Introduction](#)
2. [Common Misconceptions](#)
3. [Teaming Protocols](#)
4. [Distribution Algorithms](#)
 - A. [Address Based Algorithms](#)
 - B. [Port Based Algorithms](#)
 - C. [Other Algorithms](#)
 - D. [The Alacritech Way](#)
5. [Glossary](#)
6. [Additional Reading](#)

Introduction

Teaming, also known as port aggregation or link aggregation is fairly simple in concept, but not quite so simple in application. In an ideal world you would be able to form a team, hook up the cables, and instantly multiply your throughput by the number of links aggregated into the team. Unfortunately, it's not that simple for a number of reasons. Hopefully this page will explain the details of link aggregation well enough that you as an end user can determine how best to optimize your teaming configuration to maximize the throughput on your network. For the purposes of this document, we are only going to address load sharing teams that require active packet distribution by aggregators on both ends of the connection.

This page is intended to be a practical guide to link aggregation from the Alacritech perspective. As such, it contains simplifications and suppositions that may not be entirely accurate. If you would like an exact explanation, consult the source documents: (IEEE Std 802.3ad-2000 for 802.3ad Link Aggregation, Cisco Corporation for Fast/Gigabit EtherChannel) or contact Support at support@alacritech.com.

Common Misconceptions

There are a couple of common misconceptions about link aggregation that cause people to have unrealistic expectations of resultant performance. The first is that link aggregation automatically multiplies your network throughput by the number of links in the team, regardless of connection type, environment, etc. This assumption is incorrect. Link aggregation increases your total throughput (when set up properly), but will not increase an individual connection's throughput at all. That is to say; if you can do a file copy at 50MB/sec on a single gigabit link, if you create a team, your speed for that file copy will still be exactly the same. If you do the same copy simultaneously from multiple clients, your total throughput will increase, but the copy time for any individual file will either remain the same or possibly decrease, depending on resource contention. The reason for this is that all of the common link aggregation protocols require that a given conversation must be carried across a single link at a time.

A second misconception relates to the term "Load Balancing". People use the term load balancing as interchangeable with teaming. This is incorrect. Load balancing implies that the owning device makes intelligent decisions based on the bandwidth utilization of the individual ports when it decides which link to send the next packet down. This is untrue for a number of reasons. Among these reasons are, as mentioned above, a given conversation must reside on a single link. Another is that load tends to be highly volatile. What starts out as balanced at the beginning of a conversation may not be balanced a few milliseconds later, thus the return on the additional overhead of active load balancing is questionable. The most important reason though

is that for active balancing to work, both ends of the connection must cooperate with each other, and as yet there is no industry standard mechanism for that cooperation.

Another misconception is that all teaming works the same. This isn't true at all, as neither of the two most popular link aggregation protocols - 802.3ad, nor EtherChannel - specify the conversation distribution algorithm. With implementation left to the vendors, there ends up being a wide variety of distribution algorithms available. As to which conversation distribution algorithm works best in a given environment that discussion is below.

A final point that should be mentioned is that with the exception of true load based distribution, there is no requirement that opposite ends of a teamed connection run the same conversation distribution algorithm. In fact, there are many situations where it's not even desirable.

Teaming Protocols

There are two different link aggregation protocols that are in common use today. Cisco (Fast/Gigabit) EtherChannel, and 802.3AD. Technically, Fast EtherChannel and Gigabit EtherChannel are different protocols, but other than link layer support, they seem to behave the same. Furthermore, Since Alacritech teaming is designed to be compatible with, rather than compliant to, EtherChannel and 802.3AD, we're going to treat them as pretty much the same as well.

The important thing to know about EtherChannel and 802.3ad is that despite the different rules they may have for ensuring packet ordering, prevention of duplicates, extensions, setup, etc., they have a couple of rules in common that define most teaming behavior. Rule one is that a given conversation should only reside on a single link at a time. A conversation may be moved from one link to another, but the timeouts necessary to ensure packet order make it inefficient to do so except in the case of catastrophic failure such as a link down event. Rule two is that the actual algorithm for distribution of conversations among the member ports is left to the implementer. Any switch vendor can choose any conversation distribution algorithm they want as long as it abides by the rules of packet ordering and conversation location.

Another point that needs to be mentioned is that both EtherChannel and 802.3ad include an optional automated control protocol - PAGP for EtherChannel, and LACP for 802.3ad. Alacritech teaming is incompatible with these control protocols so they must be disabled on any ports connected to Alacritech teams.

Distribution Algorithms

Below are descriptions of some of the most commonly used packet distribution algorithms. Some vendors may use combinations or variations of one or more of the algorithms described here, depending on the traffic type. When in doubt, check the vendor's documentation.

A. Address Based Algorithms

Destination MAC

In this method, the Ethernet packet is examined, and the last few bits of the destination MAC address are used to determine which port on which to transmit the packet. This method is very popular with low-end switch vendors who do not support multiple packet distribution algorithms.

Advantages: Fast and cheap. Since the switch already has to know the destination MAC in order to perform normal switching operations, this method incurs almost no additional overhead.

Disadvantages: Only appropriate for switch-to-switch connections. Since it is a requirement that all members of a link aggregation group have the same MAC address, any traffic from the switch to a directly connected host (or router) would all travel across the same link. In the case of Alacritech, where the requirements of TCP offload dictate that we must reply on the same port that we receive on for a given connection, this is doubly harmful and results in all traffic flowing across a single port, totally eliminating any possible performance improvement that might be gained by teaming.

Recommended for: Only use this method if it is the only one available.

Source MAC

In this method the Ethernet packet is examined, and the last few bits of the source MAC address are used to determine which port on which to transmit the packet.

Advantages: Relatively fast and cheap. Not as cheap as destination MAC, but still fairly shallow in the packet, so not much of a performance overhead.

Disadvantages: Not so good for switch-to-switch connections if most of the traffic up stream originates from a single host or router.

Recommended for: Switch-to-host

Source-Destination MAC (also known as XOR or SA/DA)

The last few bits of the source MAC and destination MAC addresses are XORed together to determine which port on which to transmit the packet.

Advantages: Combines the advantages of source MAC and dest MAC.

Disadvantages: Like all MAC based methods, works best if all traffic originates and terminates on the local LAN

Recommended for: All local LAN traffic, switch-to-switch or switch-to-host. **This is the method that Alacritech recommends for connection to our cards.**

Destination IP

Uses the destination IP address to determine the port. Similar to destination MAC, except that it will work well for switch-to-router connections.

Advantages: Works for WAN traffic

Disadvantages: No good for host connections

Recommended for: Switch-to-router, or switch-to-switch if there are multiple targets. Source-dest IP is better though.

Source IP

Uses source IP address to determine port. Similar to source MAC.

Advantages: Works for WAN traffic. IP addresses tend to be more uniformly distributed than MAC address, so will often give better "balance" than source MAC.

Disadvantages: Depends on a good distribution of upstream hosts.

Recommended for: Switch to host, or router to switch.

Source-Destination IP

Uses both the source and destination IP address to determine port. Probably uses an XOR on the last few bits. Some vendors may also include source or destination port numbers in the calculation for UDP or TCP traffic.

Advantages: Tends to give the best distribution on traffic. Is the most versatile for IP traffic. Works for WAN traffic.

Disadvantages: Has the highest overhead in terms of switch processor power. Probably not an issue with modern hardware.

Recommended for: Environments where you want to have fine control of traffic flow. Since it is much easier to change an IP address than a MAC address, in an environment where traffic patterns are well known and stable, you can choose host IP addresses such that your traffic is distributed across the switch ports in the fashion that you desire. Also quite useful if significant traffic originates outside the local LAN.

- B. **Port Based Algorithms** Port based algorithms are based on the physical port that the conversation comes in on.

Same Port (aka tit-for-tat)

This method will only be found on network endpoints. If a switch were to use this method it is probable that all traffic will flow on a single link.

This method sends outgoing packets on the same physical port that the incoming conversation was initiated on. If initiating a conversation, then an arbitrary port is chosen. When a reply is received on a port other than the conversation was initiated on, the conversation will be moved to the new port.

This is the method that Alacritech uses. Our implementation will be explained in more detail below.

Advantages: It is fairly simple to implement, and it maximizes compatibility other teaming protocols and algorithms.

Disadvantages: Can sometimes result in all traffic going across a single physical port, while the other ports are idle.

Recommended for: Generally this algorithm will only be available on host interfaces and where it is available, it will be the only mode available.

C. Other Algorithms

Round Robin Packet

Packets are distributed iteratively across all the ports.

Advantages: Traffic is evenly divided among all the ports in the team, maximizing throughput.

Disadvantages: Completely violates protocols in that a conversation isn't kept on a single port, and packets are not guaranteed to arrive in sequence. This problem can be overcome if both ends of the connection are using the exact same rules for distribution. In the absence of a standard, this requires that both endpoints are from the same vendor.

Recommended for: Should only be used in a single vendor environment for switch-to-switch or switch-to-router communications.

D. The Alacritech Way

As mentioned above, Alacritech uses the "Same Port" method for conversation distribution. We do this because the mechanics of TCP offload combined with link aggregation require it. With TCP offload, all TCP context resides on the card (and our multiport gigabit products are logically multiple cards). If we didn't send and receive on the same port, we would have to be continuously flushing the connection back to the host, then handing it out to the other port. This would be slower than not having TCP offload at all.

Another thing that is important to know about the current implementation of the Alacritech teaming driver is that it tracks connections by MAC address. Since most of the time there is a one-to-one correspondence between MAC and IP (unless routers or virtual servers are involved) this isn't usually a problem. The reason we recommend that you set your switch to do source-dest MAC distribution is to prevent port hopping when there will be extensive conversations between the host containing an Alacritech team and multiple virtual servers that all have the same MAC address.

When initiating a conversation across a teamed interface, the initial packet will go out through the first team member port (as defined by the operating system). If the reply comes back on a different port, we will move the conversation to that port until the conversation is terminated or it is moved by the switch.

Glossary

802.3ad: An IEEE standard for link aggregation. Part of the 802.3 Ethernet standard.

Aggregator: A process on a switch, router, or host that controls the distribution of packets across the multiple ports that constitutes a link aggregation group.

Conversation: A set of MAC frames transmitted from one end station to another, where all of the MAC frames form an ordered sequence, and where the communicating end stations require the ordering to be maintained among the set of MAC frames exchanged. (See IEEE 802.3, Clause 43.) In practical terms, most traffic is either UDP or TCP. Each in a single TCP send or UDP datagram is part of a single conversation.

Fast EtherChannel: A proprietary link aggregation protocol owned by Cisco Corporation.

Gigabit EtherChannel: A proprietary link aggregation protocol owned by Cisco Corporation.

LACP: Link Aggregation Control Protocol. A control protocol for 802.3ad Link Aggregation that allows for automatic team formation and other functions.

Link Aggregation: The process by which ports are formed into a Link Aggregation Group.

Link Aggregation Group: A group of links that appear to a MAC Client as if they were a single link. All links in a Link Aggregation Group connect between the same pair of Aggregation Systems. One or more conversations may be associated with each link that is part of a Link Aggregation Group. (See IEEE 802.3, Clause 43.)

PAgP: Port Aggregation Protocol. A control protocol for Cisco Fast EtherChannel and Gigabit EtherChannel that allows for automatic team formation and other functions.

TCP Session: Effectively all traffic on a TCP socket pair (Source IP/port <-> Destination IP/port) from initial SYN to final FIN/RST.

Team: Link Aggregation Group.

Teaming: Link Aggregation.

Trunk: Link Aggregation Group.

Additional Reading

Understanding EtherChannel Load Balancing and Redundancy on Catalyst Switches TCP/IP Illustrated, Volume I" [W. Richard Stevens, Addison Wesley Longman Inc., 1994]