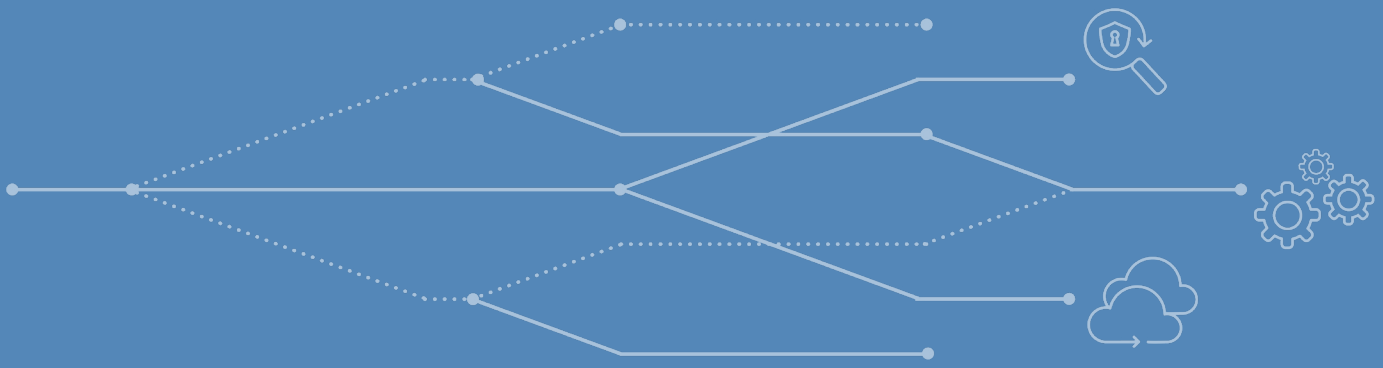
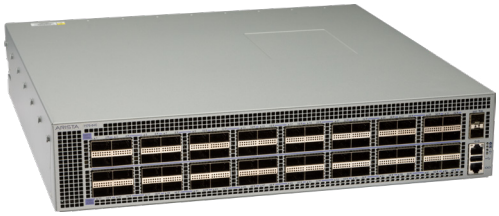


Application Note:

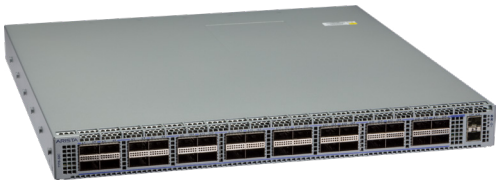
Arista 7170 Stateless Cloud Load Balancer

Introduction

The Arista 7170 Series is a unique platform with a software defined dataplane that enables it to support multiple roles in a network. This application note describes the Stateless Load Balancer role.



Arista 7170-64C:
64x 100GbE QSFP100 ports, 2 SFP+ ports



Arista 7170-32C: and 7170-32CD
32x 100GbE QSFP100 ports, 2 SFP+ ports

The primary challenge with using high performance switch silicon as a load balancer is how to deal with changes in the network topology without disrupting existing TCP connections. Switch chips are very good at processing high numbers of packets, calculating the 5-tuples and hashing multiple flows evenly across multiple devices, however when the number of devices changes this causes a rehashing of flows resulting in some being sent to a different device. This behavior causes no problems in a multi-path network infrastructure, but if used for load balancing across end hosts, results in TCP connections being lost.

Arista's Resilient ECMP is an example of a hashing feature that limits rehashing when a network device fails; however, when used for server load balancing, this technique still causes some number of active flows to be misdirected when a server is added to a server pool.

Traditional load balancers are stateful devices, solving the problem of graceful addition and removal of load balanced servers or devices by tracking all sessions flowing through the load balancer to avoid the need for re-hashing.

As this requires the tracking of a significant amount of state information, the typical solution to this problem is to use x86 or NPU based hardware with large amounts of external memory to track all the flows. The downside is that each CPU or NPU provides limited packet processing throughput, so high bandwidth load balancers tend to consist of large numbers of processors and memory resulting in a large footprint and making them costly to purchase and operate.

On the other hand switching silicon performance is several orders of magnitude higher than a typical CPU but switch chips do not contain enough memory to track the large numbers of individual flows required for stateful load balancing.

The 7170 Series Load Balancer solution introduces an alternative load balancing architecture with extremely high throughput that is stateless and able to maintain flow connections during network topology or server availability changes by leveraging the flow state stored in the servers themselves.

7170 Stateless Load Balancer

The 7170 Load Balancer is a layer 4 capable system that distributes incoming network flows to multiple servers or virtual machines anywhere within the network based on the configured policy.

Load balancing is configured by defining one or more service IP addresses with an optional port and protocol, called Virtual IPs (VIPs). Each VIP is associated with a list of servers to distribute the incoming traffic to, known as a 'nexthop' group, which can be shared between multiple VIPs. When incoming client traffic matches the IP address and port associated with a VIP it is forwarded in hardware at line rate using the load balancer pipeline.

The 7170 divides up all incoming flows to a particular <IP address/port/protocol> into a large number of flow-buckets. The flow-bucket is selected using a hash of the address and port-tuple from the packet header. Each flow-bucket is associated with a server and all flows in the flow-bucket are sent to that server.

There are many more flow-buckets than real servers, enabling fair traffic distribution and per server weighting. The number of flow-buckets never changes, only the association between flow-buckets and the servers changes.

By default incoming traffic is evenly distributed across the servers by assigning the same number of flow-buckets to each server. Unequal or weighted distribution across servers can be achieved by allocation different numbers of flow-buckets to different servers.

Server Removal (graceful & unplanned)

When a server is removed from the network all the flow-buckets that were previously allocated to that server are redistributed across the remaining active servers.

There are two scenarios to consider, sudden server failure and graceful shutdown.

In the case of a sudden server failure all existing connections to the failed server are lost - a natural and unavoidable outcome which no load balancer can overcome. The 7170 will however ensure that the clients of a failed server receive a reset terminating the failed connection quickly.

New connections will be handled by the remaining servers in the pool. At this point the 7170 behaves in a similar way to Arista's Resilient ECMP ensuring only disrupted connections are rebalanced.

When a server is removed gracefully, the desired behavior is that existing flows should continue to go to the server being removed and new flows should only be sent to other servers. The old connections will eventually complete and the server can then be shut down without any client impact.

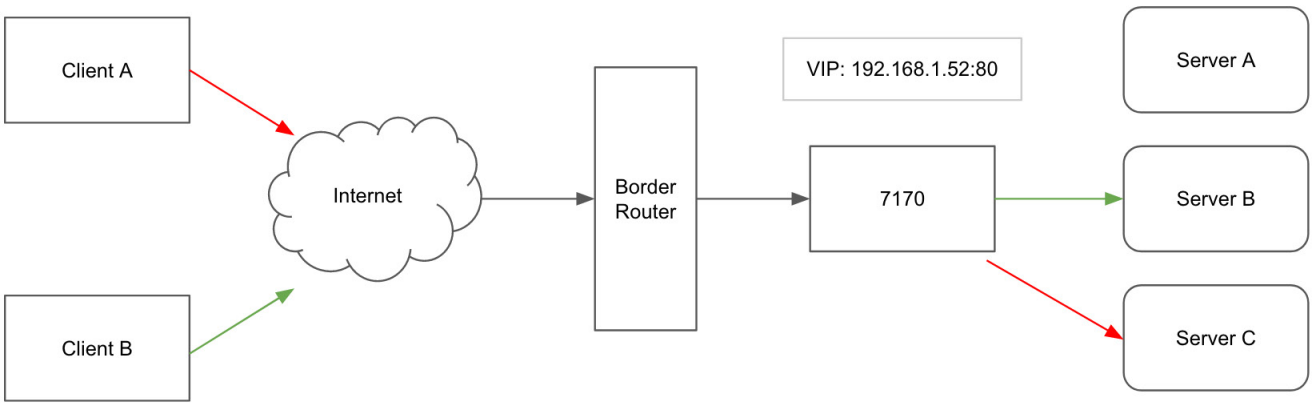


Figure 1

Figure 1 shows two clients connecting to VIP 192.168.1.52 on port 80.

The 7170 hashes the flow from client A to server C and the flow from client B to server B.

Consider the case where server B is gracefully removed from the network, the connection between client B and server B must be maintained however new flows must not go to server B.

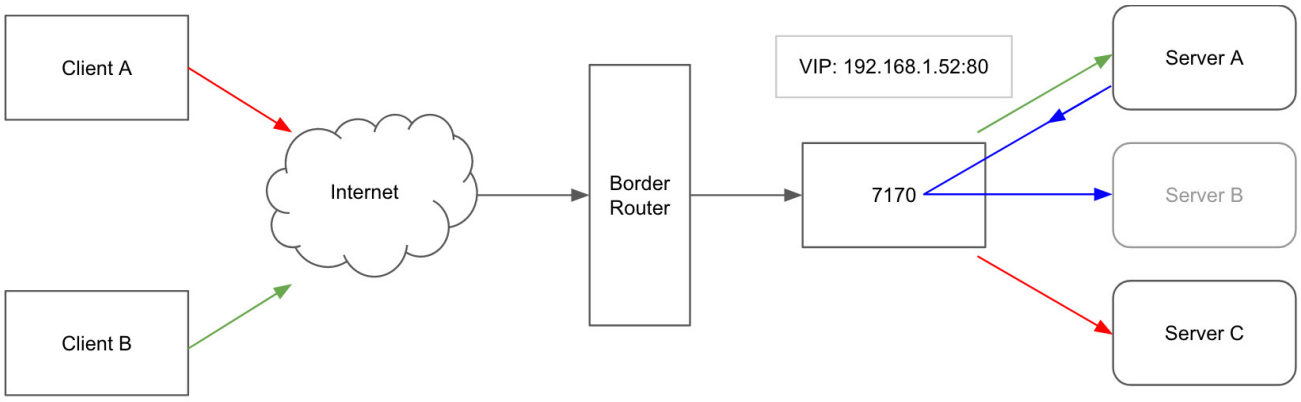


Figure 2

Figure 2 shows the traffic flow while Server B is being gracefully withdrawn from the network.

Once server B is administratively withdrawn, a new hash table is calculated omitting server B. Its flow-bucket[1] is reassigned to server A resulting in all traffic destined for server B now going to server A.

In a normal situation, Server A would not recognize the flow from client B and the flow would fail. However, in the stateless load balancer implementation, Server A instead forwards the unknown packet back to the 7170.

This forwarding is achieved by a custom kernel module on server A, without this module an unrecognized TCP frame would cause a RST to be sent to the client. The custom kernel module instead sends the packet back to the 7170 to a well known IP address and decrements a TTL in the packet.

When it receives the unrecognized packet from client B back from server A it uses this flow-bucket history to determine that server B was the previous server managing the flow-bucket that client B hashed to.

Upon receiving the packet back from server A it looks up the history of the flow-bucket and sends the packet to server B which recognizes the connection.

The 7170 can store up to 254 different servers in the hash history for each flow-bucket. This allows for the same hash entry to churn multiple times (i.e. through multiple additions and removals) and still guarantee the connection will be maintained.

Typically a TCP connection will naturally complete before the hash history records are exhausted, however the administrator can also configure a flow timeout value to flush the history for server B's flow-buckets.

This implementation removes the need for the 7170 to store individual flows, instead it is only required to store the server history for each flow-bucket. Since there is no storage of per-flow state, the 7170 load balancer can operate at full throughput regardless of the amount of active flows being distributed.

Hitless Server Addition

When a server is added to the network, the 7170 must re-assign flow buckets from existing servers to accommodate the new server. This means new and existing client connections that hash to this flow-bucket will now be forwarded to the new server. New connections will be handled as normal by the new server, however existing connections that have been rebalanced will not be recognized by the new server.

Following the process described for server removal, these flows will be sent back to the 7170 by the custom kernel module and the 7170 will use the flow-bucket history to send the unrecognized flow to the server that previously handled the flow-bucket allowing the client to complete the connection. This daisy-chaining process enables hitless moves, adds and changes without the need for central storage of the state of every flow.

[1] This example is simplified, in the real switch there is more than one flow-bucket reassignment and the flow-buckets will be reassigned to many different servers.

Alternative Hash History Database

As an alternative to returning unknown flows to the 7170, it is possible to implement an external shared flow bucket history database, providing each server the ability to directly lookup which server previously owned a connection. This solution can help to optimise traffic flows for cases where the load balancer is located remotely from the real servers.

Health Monitoring

The 7170 Load Balancer includes a suite of server health monitoring tools to identify failed servers. Failed devices are removed from the active server list, and recovered servers may be automatically added, removing the need for additional tools to monitor the health of servers.

Servers can be also placed into maintenance mode to be drained of traffic administratively via the 7170 CLI/eAPI or by communicating a state change via the server's response to the health monitor probe.

Traffic Forwarding and Redundancy

Servers can be directly attached to the 7170 on the same subnet, or the 7170 can tunnel traffic to the remote servers in any reachable subnets using a VXLAN like header. In the case of tunneling, the traffic is routed normally using the routing configuration on the 7170. The 7170 load balancer supports all EOS standard routing protocols including BGP, OSPF and static routing configuration.

Return traffic from the servers can go back through the 7170 or bypass the 7170 using a Direct Server Return (DSR) strategy. The 7170 does not need to process the return traffic for any load balancing purpose so DSR enables the 7170 to load balance efficiently to a widely distributed set of servers.

Traffic that does not match a VIP is routed or bridged using normal EOS routing/bridging rules.

Unlike stateful load balancers which require state synchronization (usually implemented in pairs) to overcome a load balancer failure, the stateless nature of the 7170 provides many more options for redundancy without the need for synchronization.

For example, when using a layer 3 implementation with DSR, any number of 7170 devices can be deployed anywhere in the network. With each 7170 having a consistent load balancing configuration and advertising the same set of VIPs via BGP, client traffic will reach its closest 7170 through normal routing policy and each 7170 will load balance in a consistent way.

The failure of an individual 7170 simply results in one advertisement for the VIP being withdrawn and traffic that would have normally been routed to the failed device being sent to an alternative 7170. The same mechanism also enables 7170s to be gracefully withdrawn from service for maintenance without creating a dependency on other load balancers.

Scaling

The 7170 supports up to 1,000 VIPs with a total of 16,000 real servers shared between the VIP addresses for both IPv4 and IPv6 VIPs and real servers.

The 7170 forwards traffic in hardware at line-rate, therefore offering up to 6.4Tbps of throughput in currently available models.

Since the stateless nature of the platform requires no synchronization or active-standby approach to resilience, each additional 7170 added to the infrastructure provides incremental linear scaling in total capacity for the load balanced applications.

Conclusion

Traditional Application Delivery Controllers (ADC) offer a rich set of processing logic and offload capability for homogenous data center environments, where applications and services have not been designed with scale-out in mind. The extra functionality comes with the penalty of complex products with limited throughput and higher capex and opex costs.

Modern high-bandwidth scale-out applications such as web farms, cloud infrastructure and microservices are already intrinsically designed for elastic growth and end-to-end security. Much of the functionality provided by traditional ADCs is not required and instead cost effective scaling, higher throughput and advanced redundancy models are key.

The 7170 Load Balancer focuses on these scale-out use cases, providing a unique platform for multi-terabit load balancing, combined with Arista EOS's industry leading reliability and flexible programmability to integrate smoothly into large scale highly automated environments.

Santa Clara—Corporate Headquarters
5453 Great America Parkway,
Santa Clara, CA 95054

Phone: +1-408-547-5500
Fax: +1-408-538-8920
Email: info@arista.com

Ireland—International Headquarters
3130 Atlantic Avenue
Westpark Business Campus
Shannon, Co. Clare
Ireland

Vancouver—R&D Office
9200 Glenlyon Pkwy, Unit 300
Burnaby, British Columbia
Canada V5J 5J8

San Francisco—R&D and Sales Office 1390
Market Street, Suite 800
San Francisco, CA 94102

India—R&D Office
Global Tech Park, Tower A & B, 11th Floor
Marathahalli Outer Ring Road
Devarabeesanahalli Village, Varthur Hobli
Bangalore, India 560103

Singapore—APAC Administrative Office
9 Temasek Boulevard
#29-01, Suntec Tower Two
Singapore 038989

Nashua—R&D Office
10 Tara Boulevard
Nashua, NH 03062