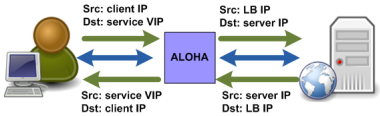
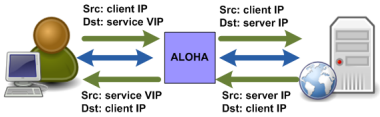
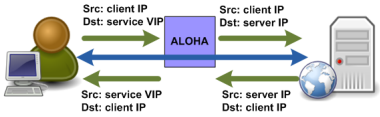
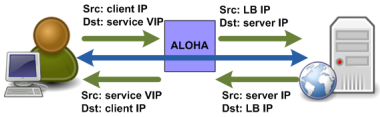
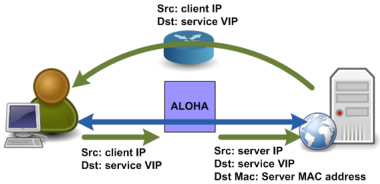
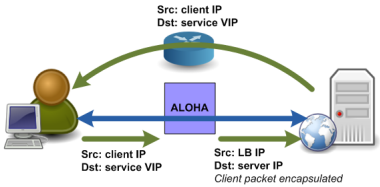
Blue arrows: TCP connections
Green arrows: IP address changes

Name	Presentation	Flow	Pros	Cons	Usage
Layer7 Reverse Proxy	The **Load-Balancer** acts as a **Reverse-Proxy** between the **client** and the **server**. Two TCP connections are established: 1. One between the **client** and the **Load-Balancer** 2. One between the **Load-Balancer** and the **server**		• non intrusive • improve security • allows protocol inspection and validation • clients and servers can be in the same subnet/vlan	• servers don't know the client IP address at TCP layer • limited to hundred of thousands of connections • bandwidth limited by **Load-balancer** capacity	• application layer advanced features • SSL offloading • improve application protection • very simple deployment
Layer7 Transparent Proxy	The **Load-Balancer** acts as a **Reverse-Proxy** between the **client** and the **server**. Two TCP connections are established: 1. One between the **client** and the **Load-Balancer** 2. One between the **Load-Balancer** and the **server** The **Load-Balancer** spoofs the **client IP** address when establishing the TCP connection to the **server**.		• improve security • allows protocol inspection and validation • **servers** know the **client IP** at TCP layer	• intrusive: the traffic from the **server** to the **client** must pass through the **Load-Balancer** • **clients** and **servers** can't be in the same subnet/vlan • takes more resources than L4 modes • bandwidth limited by **Load-balancer** capacity	• application layer advanced features • SSL offloading • improve application protection • **client IP** address mandatory at TCP layer
Layer4 Destination NAT	In **Destination NAT** mode, the **Load-balancer** forwards packets between **clients** and **servers** by changing the destination IP address of each packets. The TCP connection is established directly between the **client** and the **server**.		• fast load-balancing • **servers** know the **client IP** address at the TCP layer • allows millions of connections	• intrusive: the traffic from the **server** to the **client** must pass through the **Load-Balancer** • **clients** and **servers** can't be in the same subnet/vlan • bandwidth limited by **Load-balancer** capacity	• when response time matters • when nothing but the default gateway of the servers can be changed
Layer4 Full NAT	In **full NAT** mode, the **Load-Balancer** forwards packets between **clients** and **servers** by changing both **source IP** and **destination IP** address of each packet. The TCP connection is established directly between a **client** and a **server**.		• fast load-balancing • non intrusive • **clients** and **servers** can be in the same subnet/vlan	• **servers** don't know the **client IP** address at the TCP layer • bandwidth limited by **Load-Balancer** capacity	• when response time matters • very simple deployment
Layer4 DSR or gateway	DSR stands for Direct Server Return In DSR mode, the **Load-Balancer** forwards packets to the **servers** without changing anything in it but the destination MAC address: the new MAC address is server one. Two asymmetric flows happen: 1. **Inbound**: **client** ==> **Load-Balancer** ==> **Server** 2. **Outbound**: **server** ==> **client** The **servers** must have the **service IP** configured on a loopback to be able to accept the requests.		• fast load-balancing • allows millions of connections • allows huge outbound bandwidth • **servers** know the **client IP** address at the TCP layer • **clients** and **servers** can be in the same subnet/vlan	• intrusive: **service IP** must be configured on a loopback in each **server** • The **Load-Balancer** must have an interface in the **server** vlan	• when response time matters • when bandwidth matters
Layer4 IP Tunnel	The **IP tunnel** mode looks like the **DSR** mode, except that traffic between the **load-balancer** and the **server** can be routed. The **load-balancer** encapsulates the client packet in an IP tunnel established with the **server**. Two asymmetric flows happen: 1. **Inbound**: **client** ==> **Load-Balancer** ==> **Server** 2. **Outbound**: **server** ==> **client** The **servers** must have the **service IP** configured on a loopback to be able to accept the requests.		• fast load-balancing • allows millions of connections • **servers** know the **client IP** address at the TCP layer • **clients** and **servers** can be in the same subnet/vlan	• intrusive: an IP tunnel must be setup between the **Load-Balancer** and the **server** • intrusive: **service IP** must be configured on a loopback in each **server** • The **Load-Balancer** must have an interface in the **server** vlan	• when response time matters • when bandwidth matters • when the **load-balancer** can't have an interface in the **server** vlan
Layer4 / Layer7 Transparent listening	This mode is also known as **VIPless**. The **service IP** is not configured on the **Load-Balancer** itself, but it is routed by the routers to the **Load-Balancer**. The **Load-Balancer** will match traffic on the fly, applying any of the Load-Balancing rules. It is compatible with all modes above.		• affect only **client** to **Load-Balancer** traffic • A single vrrp IP address per **Load-Balancer cluster** is required, whatever the number of service IP it hosts • adding new service IPs means adding a route on the core routers • the **Load-Balancer** can be hidden and unreachable from internet while load-balancing	• requires networking skills • the **service IP** address can't be pinged since not configured anywhere	• when a huge number of **service IP** is required