L3DSR Load Balancing

Damodharan Rajalingam

8 December 2012

YAHOO!

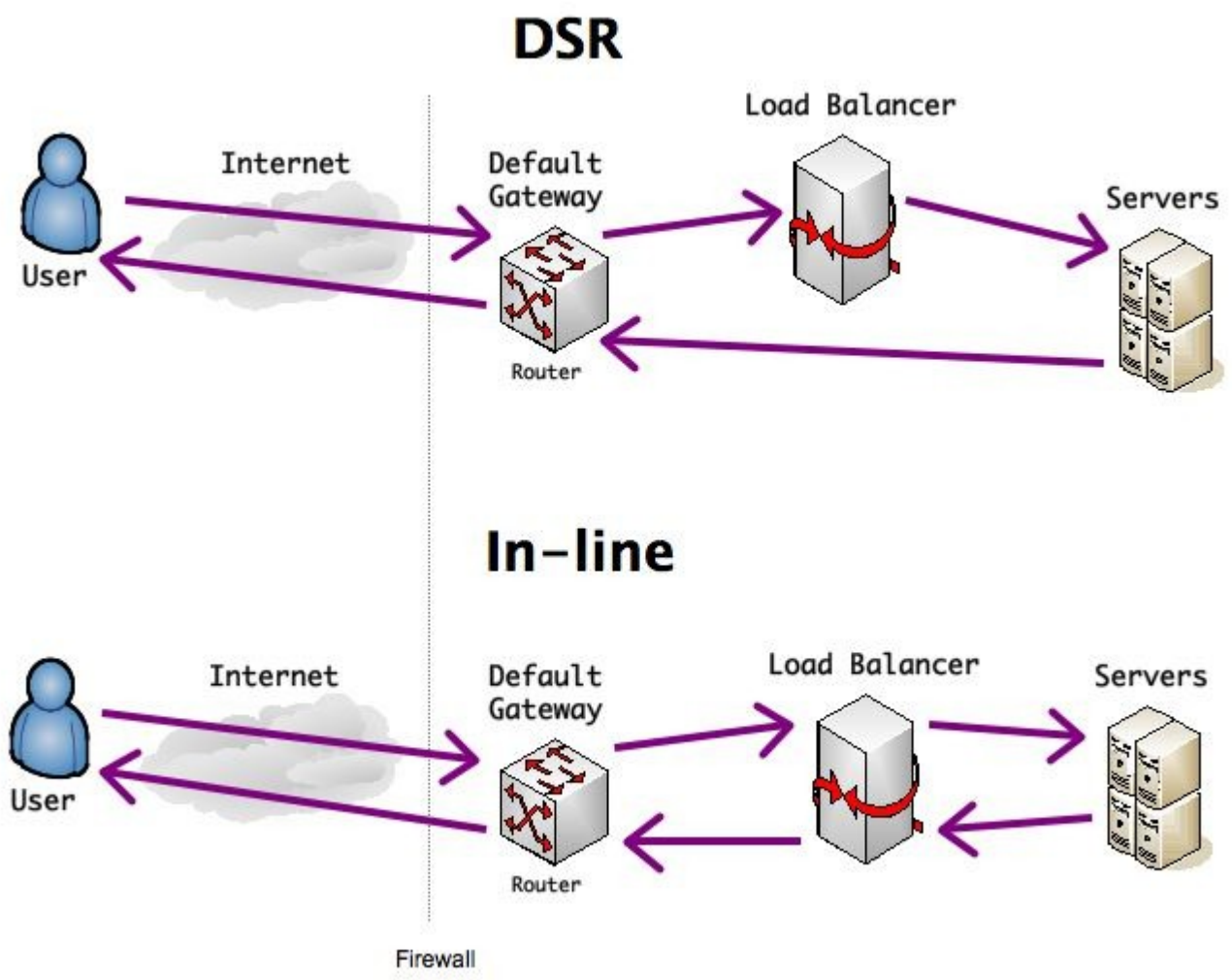
What are LBs?

- *network devices that distribute client requests or sessions among many servers (hosts)*
- Benefits
  - Scalability
  - Manageability
  - Availability
  - Security

How they work?

- Clients make a request to a Virtual IP (VIP)
- Load Balancers forward the request to the appropriate server in its rotation
  - Monitor health of servers to know their availability
  - Use a distribution algorithm to select a host
- The servers can:
  - Reply directly to the client (DSR) or
  - Reply to the Load Balancer (Inline)

DSR vs Inline

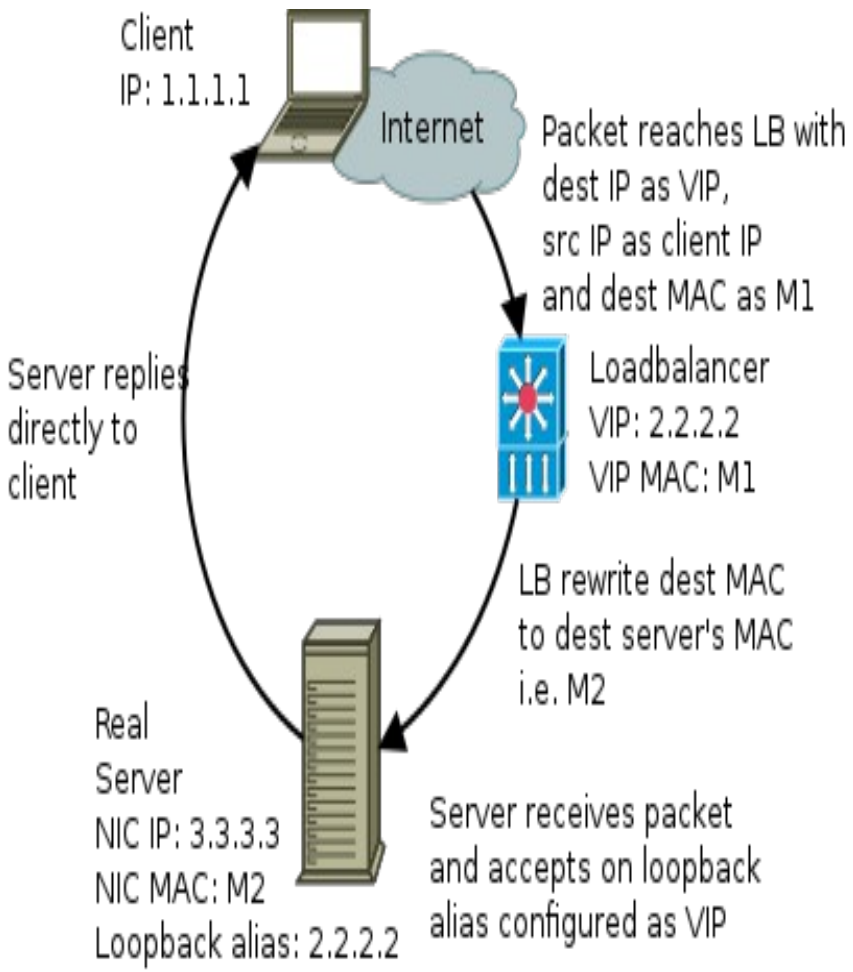


LB Configurations

- Inline
  - DNAT
  - SNAT
- DSR
  - Layer 2

Working of L2 DSR

1. Client sends SYN packet to VIP which lives on LB
2. LB figures out which server should handle the request, changes the destination MAC to that of the selected server (rest of the packet is kept intact including the destination and source ip)
3. Server has VIP IP configured as alias for loopback interface, so it can accept the packet
4. Application on server has to listen to VIP IP address
5. Server directly replies to client with Source IP = VIP IP and Destination IP = Client IP



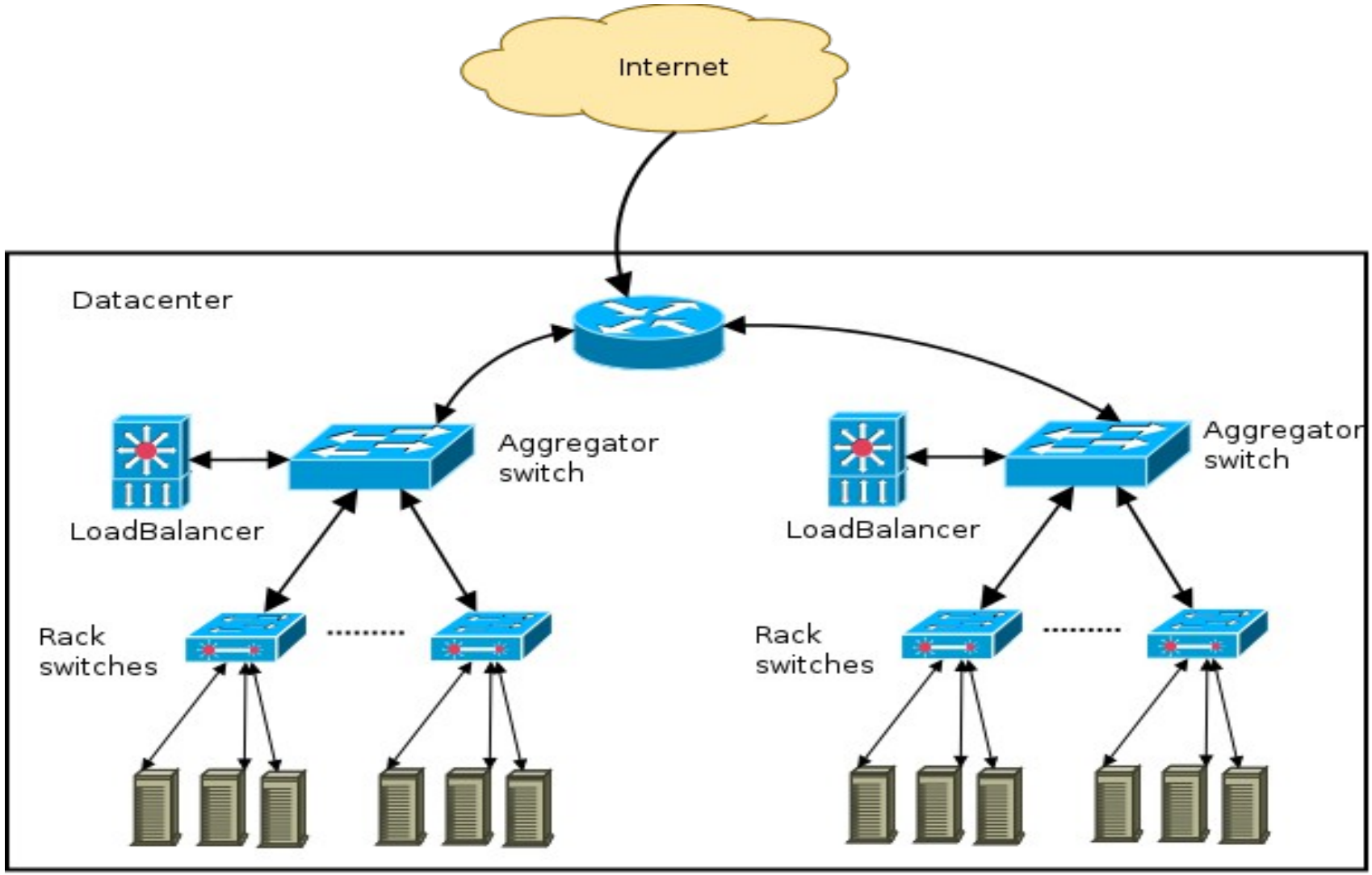
Advantages of L2 DSR

- Only sees inbound traffic. This is a huge win for HTTP type services
- Outbound bandwidth not limited by bandwidth of LB
- LB's can do more VIPs (about 8x more) in comparison to Inline
- Doesn't have return-path problem – no need to use SNAT or default route as with inline VIPs
- The clients' source addresses are preserved

Limitations of L2 DSR

- Cannot inspect L7 details for load balancing decisions (eg. Cookie or url based routing)
- Requires host configuration of VIP IP address as loopback alias
- Server cannot directly respond to ARP requests for VIP
- Port Address Translation is not possible
- Servers behind LB are susceptible to SYN attacks
- *LB and all servers need to be on same L2 segment*
  - Physical location of servers behind VIP restricted
  - Flexibility within datacenter limited

A Simplified View of DC Network (L2 based)



Scaling issues with Layer 2 design

- A Layer 2 DSR VIP can only load balance to servers which are on the same Aggregation switch
- As servers are added to VIPs over time these will eventually be on separate Aggregation switches (which have no L2 connectivity)
- When new servers have no space in existing L2 segment it results in shuffling and migration of servers to place all the servers behind the VIP being expanded in same L2 segment

Layer 3 Load Balancing

- To overcome the scalability issue we need Load Balancing to support hosts that are router hops away
- Traditional way to do load balancing in Layer 3 environment is to either use SNAT or DNAT
- DNAT (Destination NAT or Half-NAT) replaces the destination IP with the IP of real server.
  - Servers network must be configured so that reply packets always go through LB so that reply can un-NATed.
- SNAT (Source NAT or Full-NAT) replaces the source IP with LB IP in addition to the modification of destination IP.
  - Server sees requests as coming from LB and client address is lost.
  - LB can add a header to preserve client address but is resource intensive
- In both cases the outgoing traffic has to go through the LB
  - Bandwidth of LB becomes a bottleneck
  - Number of VIPs that can be configured in an LB reduces as SNAT and DNAT are resource intensive

Layer 3 DSR

- Layer 3 DSR is a way of implementing DSR across L3 boundaries
- Provides the efficiency of DSR VIPS while allowing us to scale beyond L2 segments
- L3DSR is a Yahoo! Invention, created by Dave Temkin, Dave Barrow and Igor Gashinsky. Kernel modules written by Quentin Barnes, John Baldwin, David Discher, Jan Schaumann
- Kernel modules open sourced at - https://github.com/yahoo/l3dsr/
- Supported by 3 vendors: A10, Brocade (Foundary), Citrix (Netscaler)

How to do DSR on L3

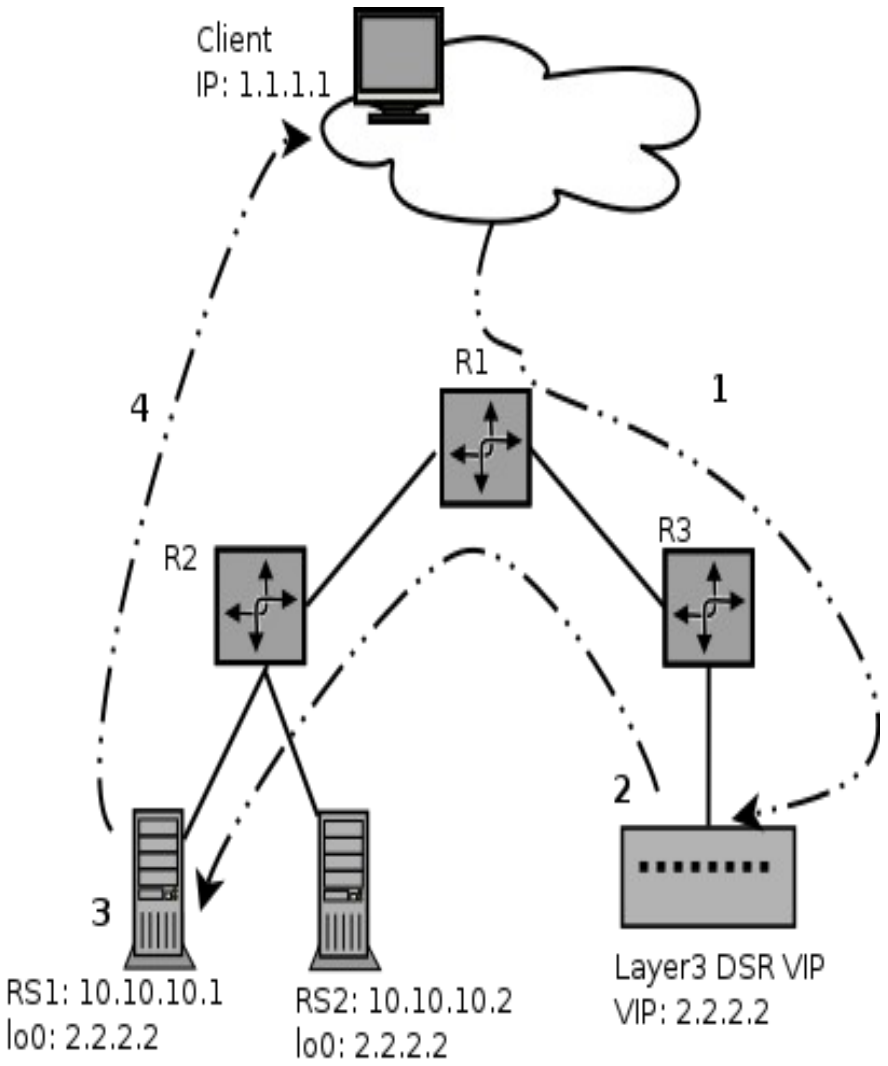
- Server need to know
  - Client source address
  - VIP address for which the request was made
- LB needs to
  - Tell the server behind the VIP the source address of the client
  - Send the request to the IP of the server instead of VIP
  - Tell the server the original destination address (ie the VIP)
- Where to put the extra information?
  - In L2DSR both source and destination IP present in respective address fields
  - Here the destination IP is the NIC IP of server.
  - The DSCP field is used to pass the information about actual destination IP
  - DSCP field is 6-bits wide and is used for TOS, QoS, COS – none of these used inside Yahoo!
  - Map these bits to VIPs

L3DSR – How it works

- LB and Servers need to agree on DSCP <=> VIP mapping. For example DSCP value 0x000100 maps to VIP address 2.2.2.2
- On the incoming packet LB sets the DSCP field to the value corresponding to the VIP according to the known value
- LB also changes the destination address to the server's real IP address and keeps the client's source address
- Server checks DSCP bit
- Server rewrites the destination address according to the known mapping to appropriate VIP
- VIP configured on loopback alias as with L2DSR
- Application binds to VIP address
- Server responds to client;s source address from VIP

L3DSR Packet Flow

	ToS bits	Source IP	Destination IP
1	0x0	1.1.1.1	2.2.2.2
2	0x4	1.1.1.1	10.10.10.1
3	0x4	1.1.1.1	10.10.10.1 (to) 2.2.2.2
4	0x0	2.2.2.2	1.1.1.1



L3DSR – Benefits

- Location independence of servers within data center
  - We can physically move hosts across different Layer 2 segments
  - We can rebuild hosts on different Layer 2 segment, then use it to replace broken hosts
  - We can add new hosts to existing VIPs without concern for physical location / IP allocation withing same L2 segment
- Preservation of clients' source address
- DSR performance

L3DSR – Limitations

- There is not L7 load balancing capabilities like L2DSR
- Configuration more involved than L2DSR
  - Requires kernel module, iptable setup and loopback configuration
- Use of DSCP precludes using of QoS inside colo
- Additional overhead in network latency
  - 0.04 ms additional latency in our experiments
- Limited OS and load balancing vendor support
- Like L2DSR this is also susceptible to SYN attacks
- DSCP <=> VIP mappings must be tracked

L3DSR Health Checks

- Health checks need to be able to check if iptables plugin/kernel module is in place and working correctly
- LB sends a HTTP request with following details:
  - Source IP: LB_IP
  - Destination IP: Server_IP
  - DSCP: DSCP mapping for that VIP
- Server replies
  - Source IP: VIP
  - Destination IP: LB_IP
  - Status code: 200 OK
- Due to destination address rewriting, source/destination on LB does not match. This means LB cannot use regular TCP stack for this health check

L3DSR for IPv6

- L3DSR for IPv6 is implemented in a similar fashion to IPv4 – it uses the DSCP field (bits 4-7 of 1st octet and bits 0-1 of 2nd octet) in the IPv6 header (RFC2474)
- Yahoo Linux and FreeBSD teams are developing kernel modules
- Brocade, A10 and Citrix are all implementing IPv6 L3DSR in future code revisions
- Long term plans to develop with vendors IPv6 header extension instead of using DSCP field – might include rich information like VIP IP, ports translated etc
-

References

- Github repository: https://github.com/yahoo/l3dsr
- This presentation uses lots of information from following:
  - https://github.com/yahoo/l3dsr/blob/master/docs/nanog51.pdf
  - http://www.slideshare.net/cwestin63/l3-dsr-lspepresentation20120119

Q & A