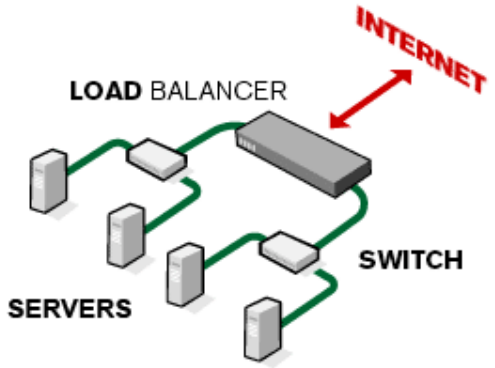
Load Balancing

Introduction

- Explosive Growth of the Internet
  - 100% annual growth rate
- Sites receiving unprecedented workload
  - FB has more than 900 million objects that people interact with (pages, groups, events and community pages)
  - Twitter receives about 50 million tweets per day, which breaks down to about 600 per second.

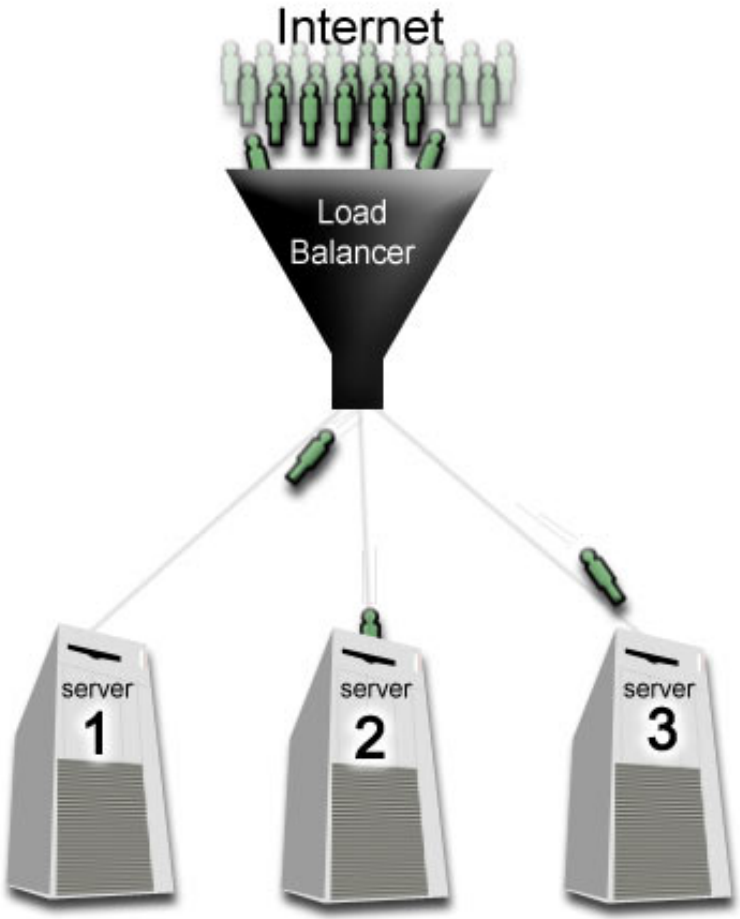
Definitions

- **load balancing** is a technique to spread work between many computers, processes, disks or other resources in order to get optimal resource utilization and decrease computing time.
- **A load balancer** consists of a virtual server (also referred to as vserver or VIP) which, in turn, consists of an IP address and port.
  - A load balancer can be used to increase the capacity of a **server farm** beyond that of a single server.
  - It can also allow the service to continue even in the face of server down time due to server failure or server maintenance.
  - virtual server is bound to a number of physical services running on the physical servers in a server farm.



Virtual Servers

- Different virtual servers can be configured for different sets of physical services, such as TCP and UDP services in general.
- Application specific virtual server may exist to support HTTP, FTP, SSL, DNS, etc.
- The load balancing methods manage the selection of an appropriate physical server in a server farm.



Persistency and Fail-over

- Persistence can be configured on a virtual server; once a server is selected, subsequent requests from the client are directed to the same server.
- Persistence is sometimes necessary in applications where client state is maintained on the server, but the use of persistence can cause problems in failure and other situations.
- A more common method of managing persistence is to store state information in a shared database, which can be accessed by all real servers, and to link this information to a client with a small token such as a cookie, which is sent in every client request.
- case of failure of a service, the load balancer continues to perform load balancing across the remaining services that are UP.
- In case of failure of all the servers bound to a virtual server, requests may be sent to a backup virtual server (if configured) or optionally redirected to a configured URL.

Load Balancers

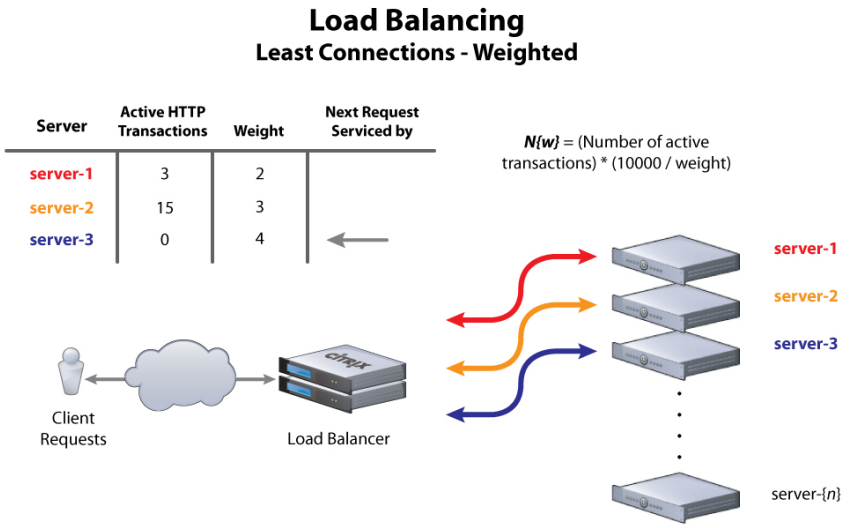
- In computer networking, load balancing is a technique to distribute work load evenly across two or more resources. The purpose of a load balancer is to maintain the system in a state where load on a node is below it's target
  - **Load**: could be storage, bandwidth, etc.
  - **Target**: the load a node is willing to take (ex. capacity, avg. util. + slack)
- Assumptions
  - Nodes are cooperative
  - Only one bottlenecked resource

Why to Load-balance?

- Scale applications / services
- Ease of administration / maintenance
  - Easily and transparently remove physical servers from rotation in order to perform any type of maintenance on that server.
- Resource sharing
  - Can run multiple instances of an application / service on a server; could be running on a different port for each instance; can load-balance to different port based on data analyzed.

Load-Balancing Algorithms

- Most predominant:
  - **least connections**: server with fewest number of flows gets the new flow request.
  - **weighted least connections**: associate a weight / strength for each server and distribute load across server farm based on the weights of all servers in the farm.
  - **round robin**: round robin thru the servers in server farm.
  - **weighted round robin**: give each server 'weight' number of flows in a row; weight is set just like it is in weighted least flows.
- There are other algorithms that look at or try to predict server load in determining the load of the real server.



Server Load-balancing

- Gets user to needed resource:
  - Server must be available
  - User's "session" must not be broken
    - If user must get to same resource over and over, the SLB device must ensure that happens (ie, session persistence)
- In order to do work, SLB must:
  - Know servers – IP/port, availability
  - Understand details of some protocols (e.g., FTP, SIP, etc)
- Network Address Translation, NAT:
  - Packets are re-written as they pass through SLB device.

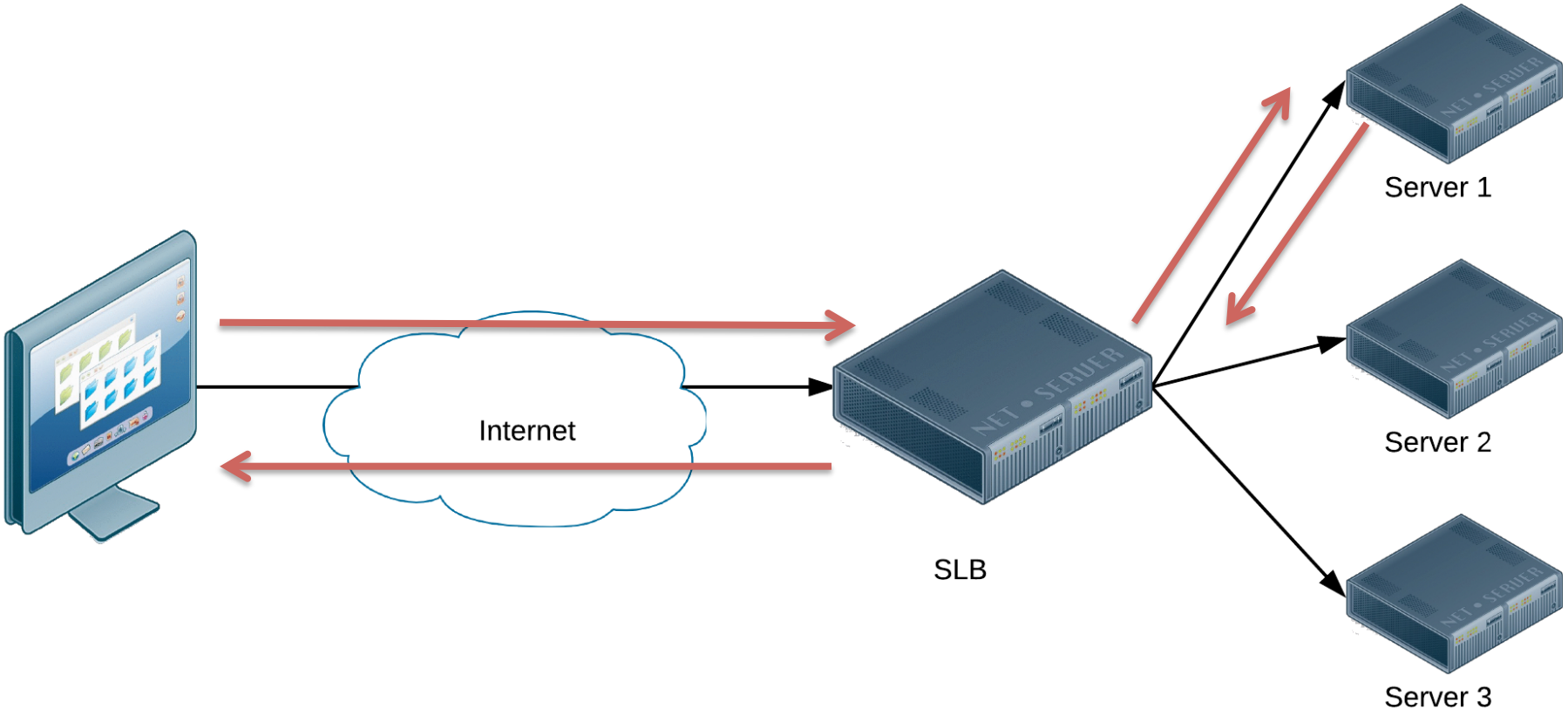
How SLB Devices Make Decisions

- The SLB device can make its load-balancing decisions based on several factors.
  - Some of these factors can be obtained from the packet headers (i.e., IP address, port numbers, etc.).
  - Other factors are obtained by looking at the data beyond the network headers. Examples:
    - HTTP Cookies
    - HTTP URLs
    - SSL Client certificate
- The decisions can be based strictly on flow counts or they can be based on knowledge of application.
- For some protocols, like FTP, you have to have knowledge of protocol to correctly load-balance (i.e., control and data connection must go to same physical server).

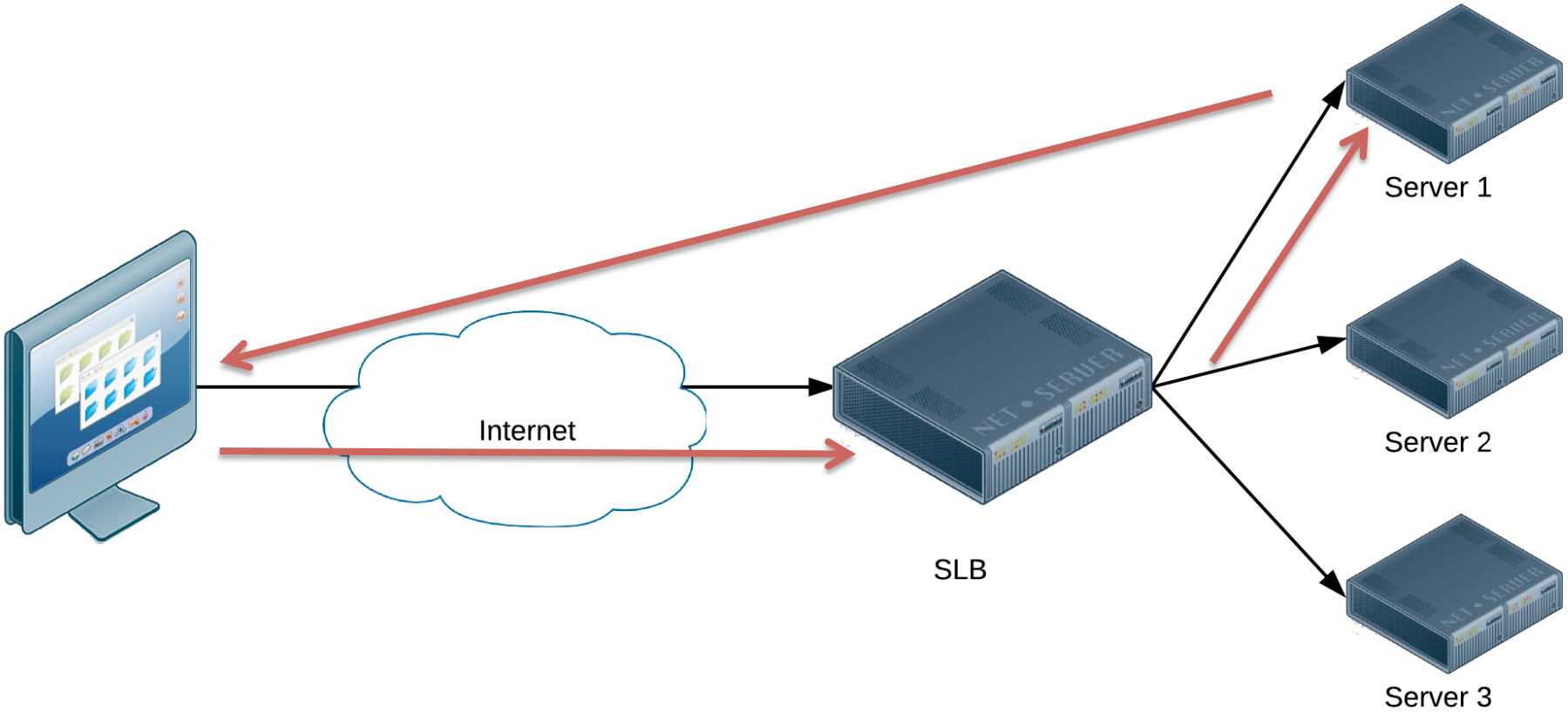
SLB Operation

- When a new flow arrives it determines if virtual server exists.
  - If so, make sure virtual server has available resources.
  - If so, then determine level of service needed by that client to that virtual server.
    - If virtual machine is configured with particular type of protocol support of session persistence, then do that work.
  - Pick a real server for that client.
    - The determination of real server is based on flow counts and information about the flow.
    - In order to do this, the SLB may need to proxy the flow to get all necessary information for determining the real server – this will be based on the services configured for that virtual server.
- If not, the packet is bridged to the correct interface based on Layer 2.

SLB with NAT



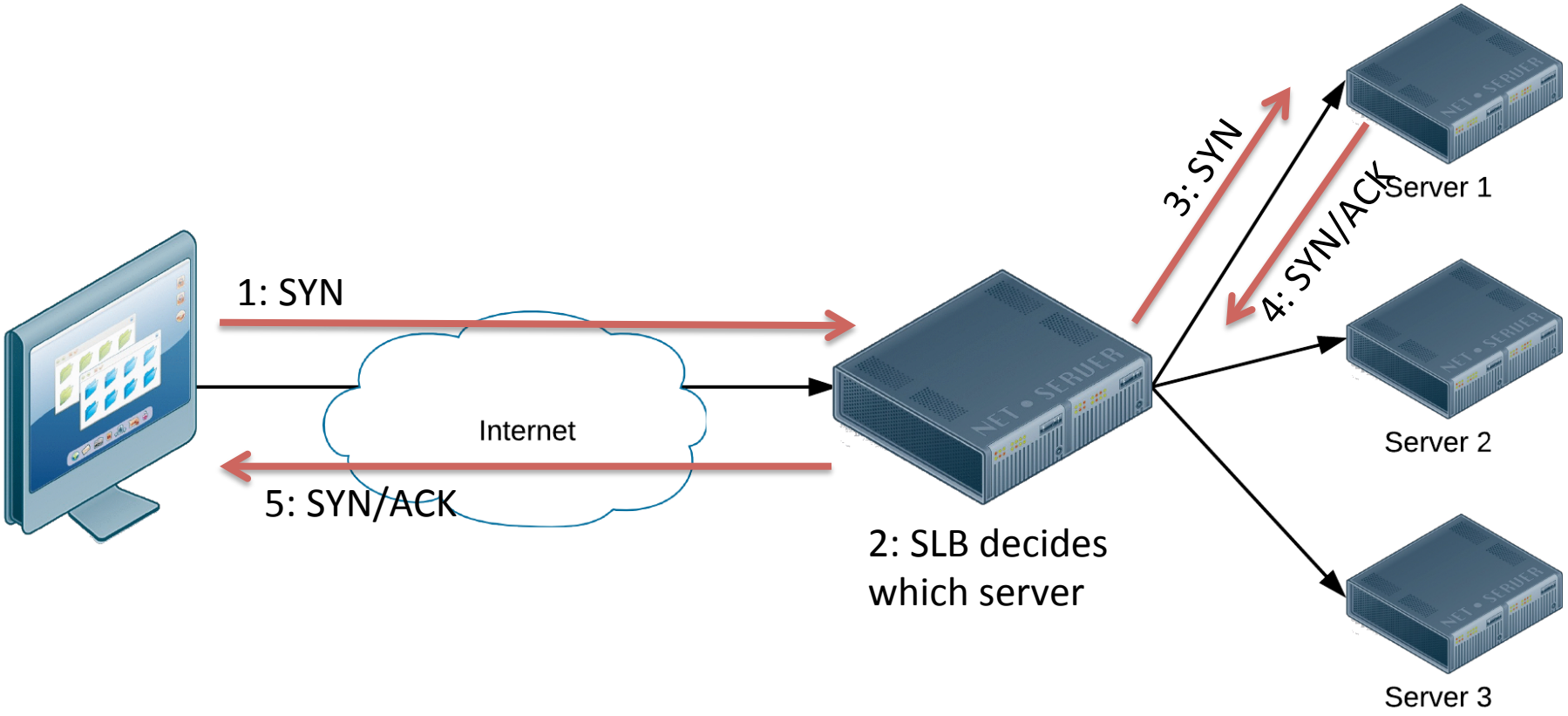
SLB without NAT



Load-Balance: Layer 3 / 4

- Looking at the destination IP address and port to make a load-balancing decision.
- In order to do that, you can determine a real server based on the first packet that arrives.

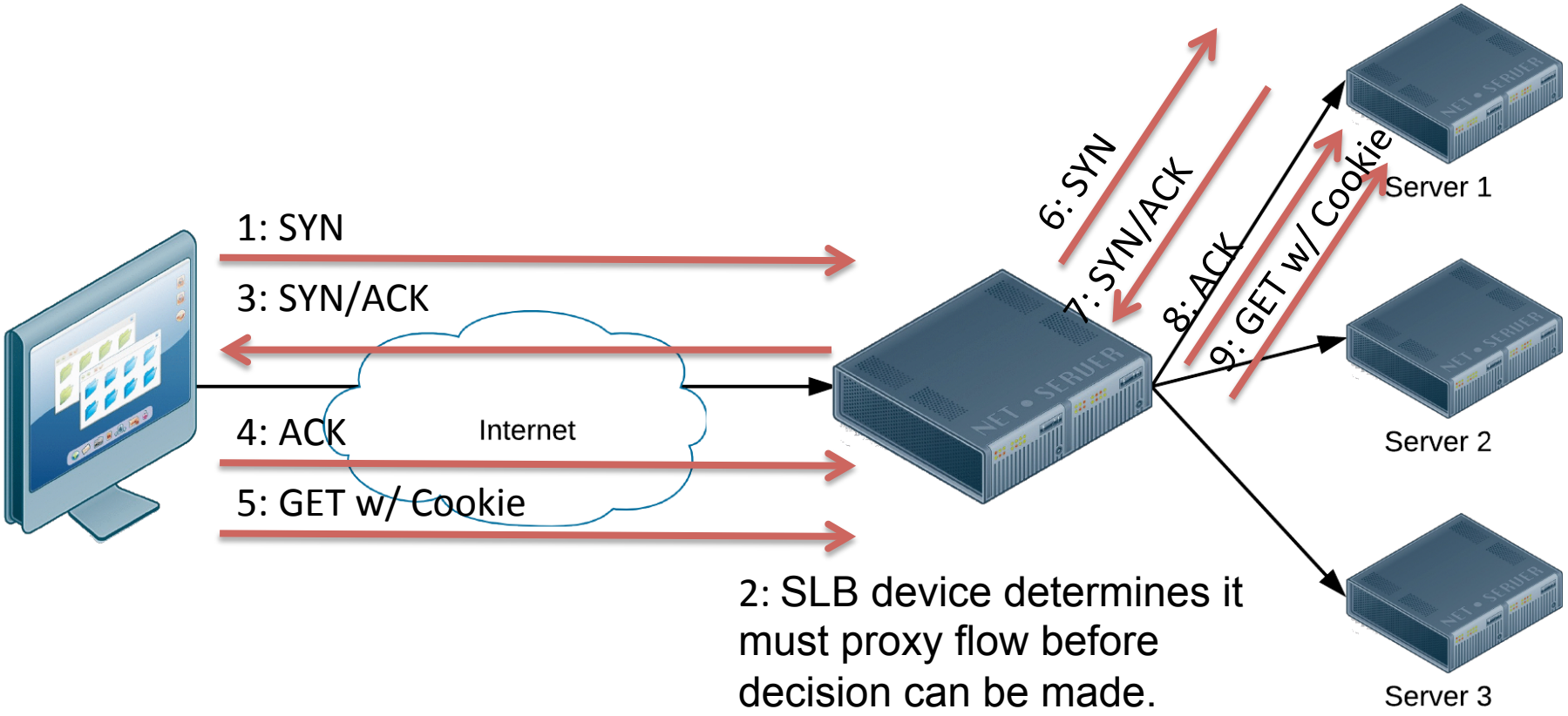
SLB @ layer 3/4


1: SYN
Internet
5: SYN/ACK
2: SLB decides
which server
3: SYN
4: SYN/ACK
Server 1
Server 2
Server 3

Load-Balance: Layer 5+

- The SLB device must terminate the TCP flow for an amount of time BEFORE the SLB decision can be made.
  - For example, the cookie value must be sent by the client, which is after the TCP handshake before determining the real server.

SLB @ layer 5+



Rest of flow continues with Server response.
Note: the flow can be unproxied at this point for efficiency.

Server Feedback

- Need information from real server while it is a part of a server farm.
- Why?
  - Dynamic load-balancing based on ability of real server.
  - Dynamic provisioning of applications.

Server Feedback: Use of Information

- In order to determine health of real servers, SLB can:
  - Actively monitor flows to that real server.
  - Initiate probes to the real server.
  - Get feedback from real server or third party box.
- Availability of real server is reported as a 'weight' that is use by SLB algorithms (e.g., weighted round robin, weighted least connections).
- As weight value changes over time, the load distribution changes with it.

How to Get Weights

- Statically configured on SLB device – never change.
- Start with statically configured value on SLB device for initial start-up, then get weight from:
  - Real server
  - Third party box / Collection Point
    - It is assumed that if a third party box is being used, it would be used for all the real servers in a server farm.

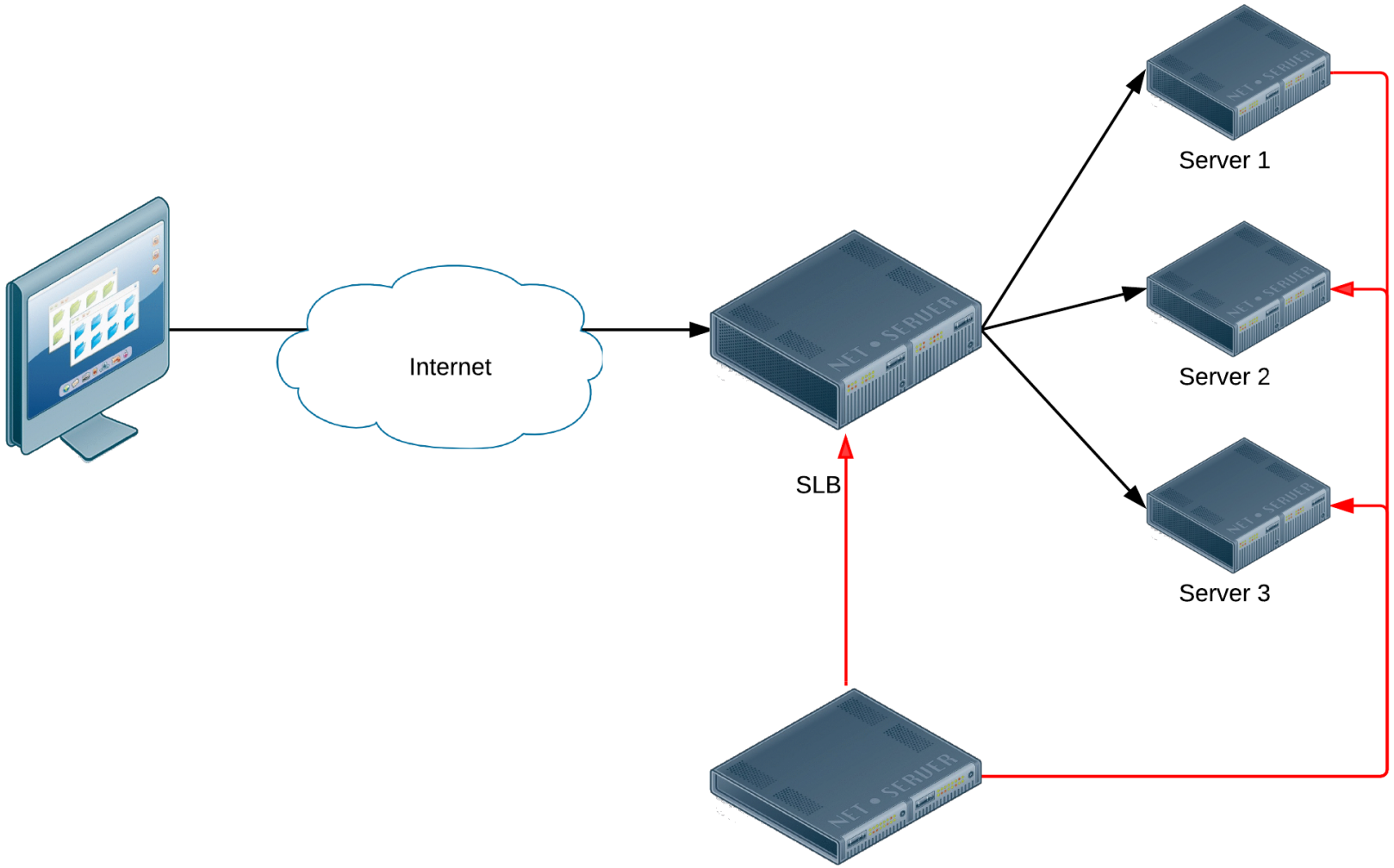
Direct Host Feedback

- Description: Have “agents” running on host to gather data points. That data is then sent to SLB device just for that physical server.
  - Note: agent could report for different applications on that real server.
  - Agent could be based on available memory, general resources available, proprietary information, etc.

Direct Host Feedback

- Pros:
  - Have some way to dynamically change physical server's capability for SLB flows.
- Cons:
  - SLB device must attempt to normalize data for all real servers in a server farm. If have heterogeneous servers, it is difficult to do.
  - Difficult for real server to identify itself in SLB terms for case of L3 vs. L4 vs. L5, etc SLB scenarios.

Third Party Feedback: Network



Host to Third Party Feedback

- Description: Real servers report data to a ‘collection point’. The ‘collection point’ system can normalize the data as needed, then it can report for all physical servers to the SLB device.

- Pros:
  - Have a device that can analyze and normalize the data from multiple servers. The SLB device can then just do SLB functionality.
- Cons:
  - Requires more communication to determine dynamic weight – could delay the overall dynamic affect if it takes too long.

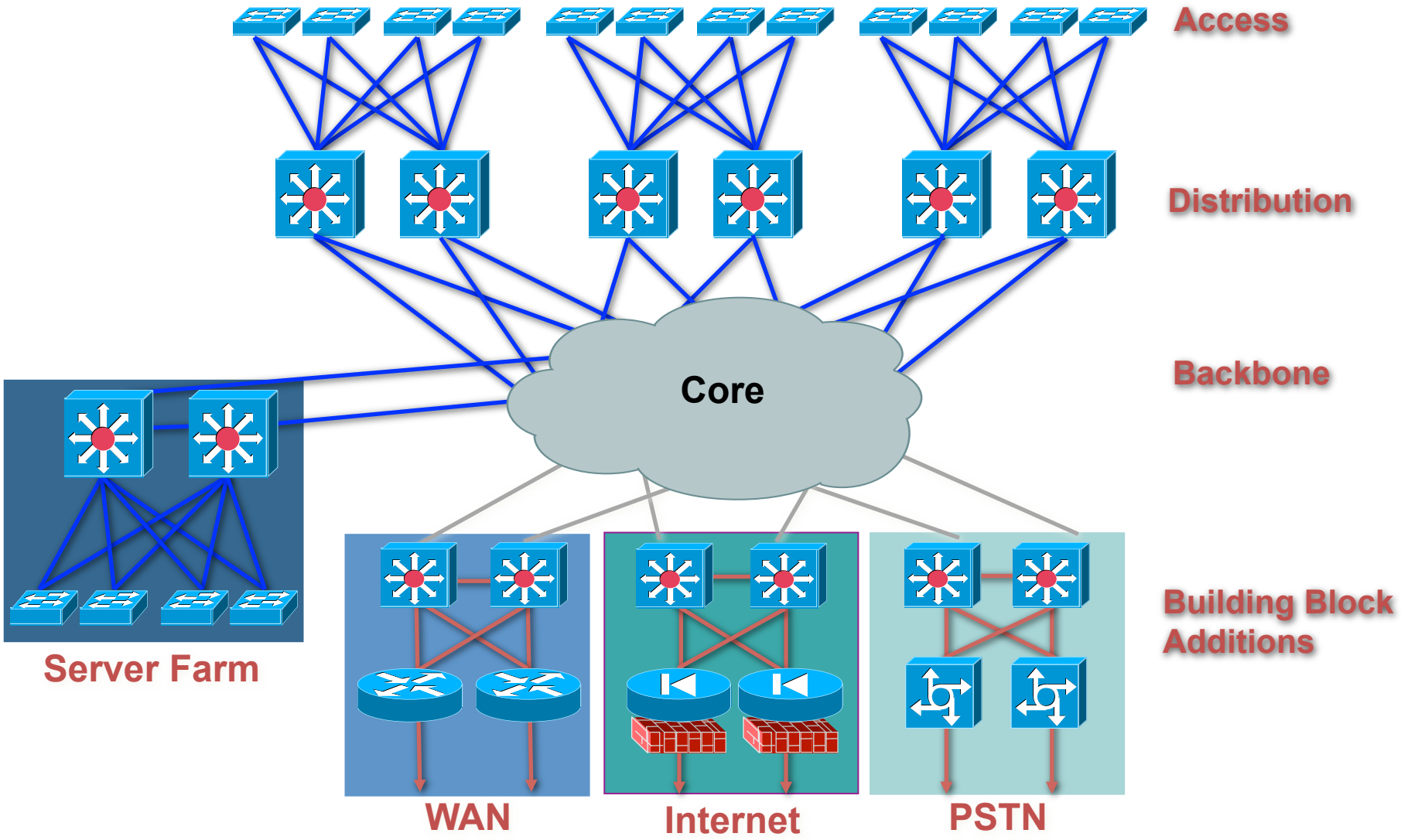
Web Server Load Balancing

- One major issue for large Internet sites is how to handle the load of the large number of visitors they get.
- This is routinely encountered as a scalability problem as a site grows.
- There are several ways to accomplish load balancing
- For example in WikiMedia load is balanced as:
  - Round robin DNS distributed page requests evenly to one of three Squid Cache servers
  - Squid cache servers used response time measurements to distribute page requests between seven web servers.
  - In addition, the Squid servers cached pages and delivered about 75% of all pages without ever asking a web server for help.
  - The PHP scripts which run the web servers distribute load to one of several database servers depending on the type of request, with updates going to a master database server and some database queries going to one or more slave database servers.

Network Load Balancing

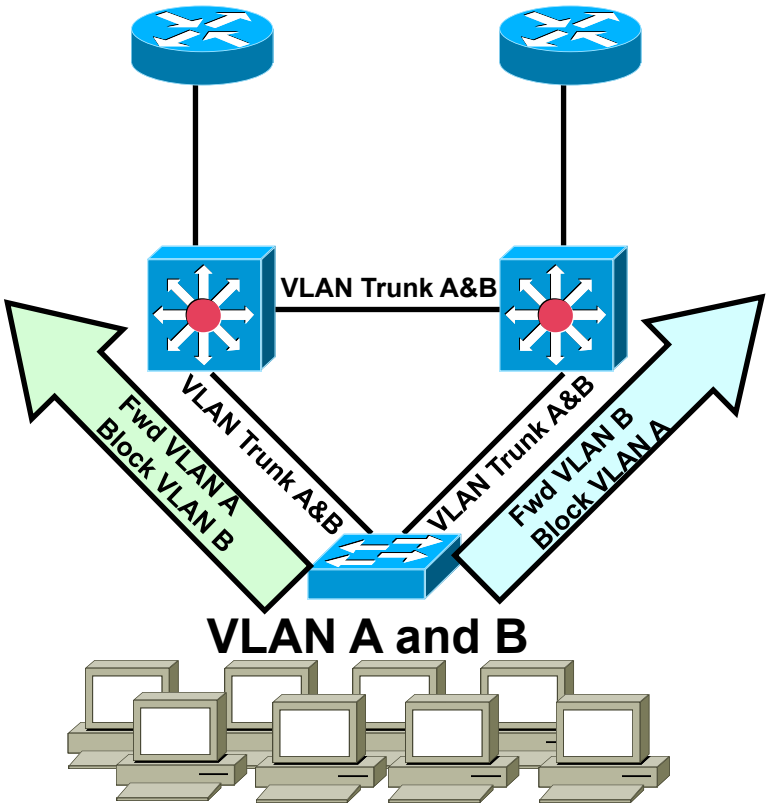
- More then just Load Balancing
- Use Cases
  - Failover
  - Multiple ISP providers
  - Channel bounding & Layer 2 & Layer 3 load-balancing using Network Equipment's (Switches and Routers)
- Layer 2
  - Mostly using spanning tree protocol
- Layer 3
  - HSRP, VRRP, GLBP

Multilayer Network Design

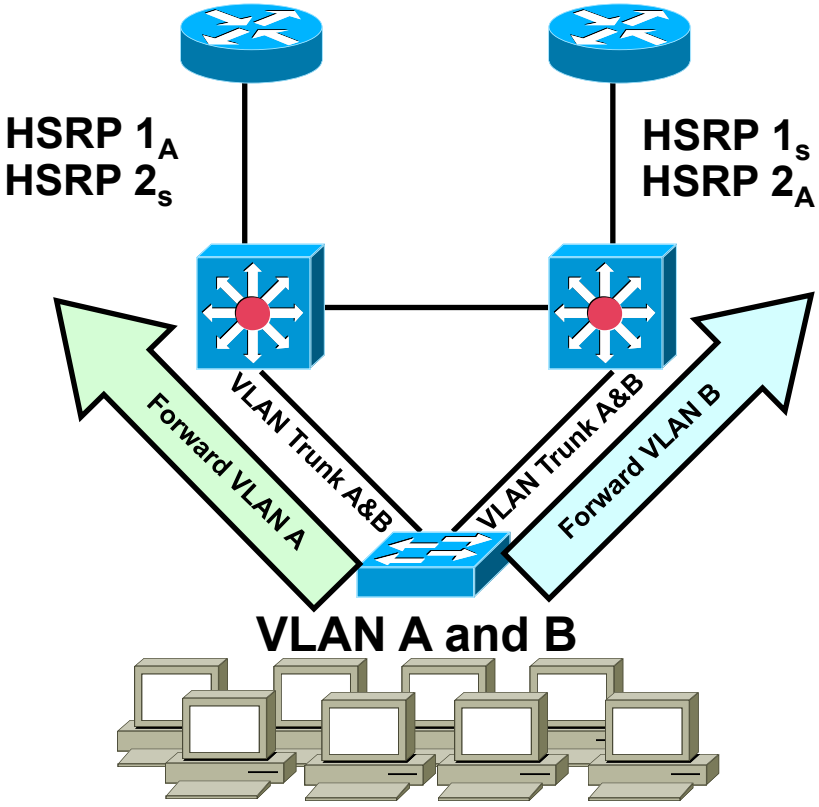


Multi-VLAN Load Balancing Methods

Layer-2 Mode Load Balancing



Layer-3 Mode Load Balancing



First Hop Redundancy Schemes

- Hot Standby Router Protocol (HSRP)
  - Cisco informational RFC 2281 (March 1998)
- Virtual Router Redundancy Protocol (VRRP)
  - IETF Standard RFC 2338 (April 1998)
- Gateway Load Balancing Protocol (GLBP)
  - Cisco designed, load sharing, patent pending

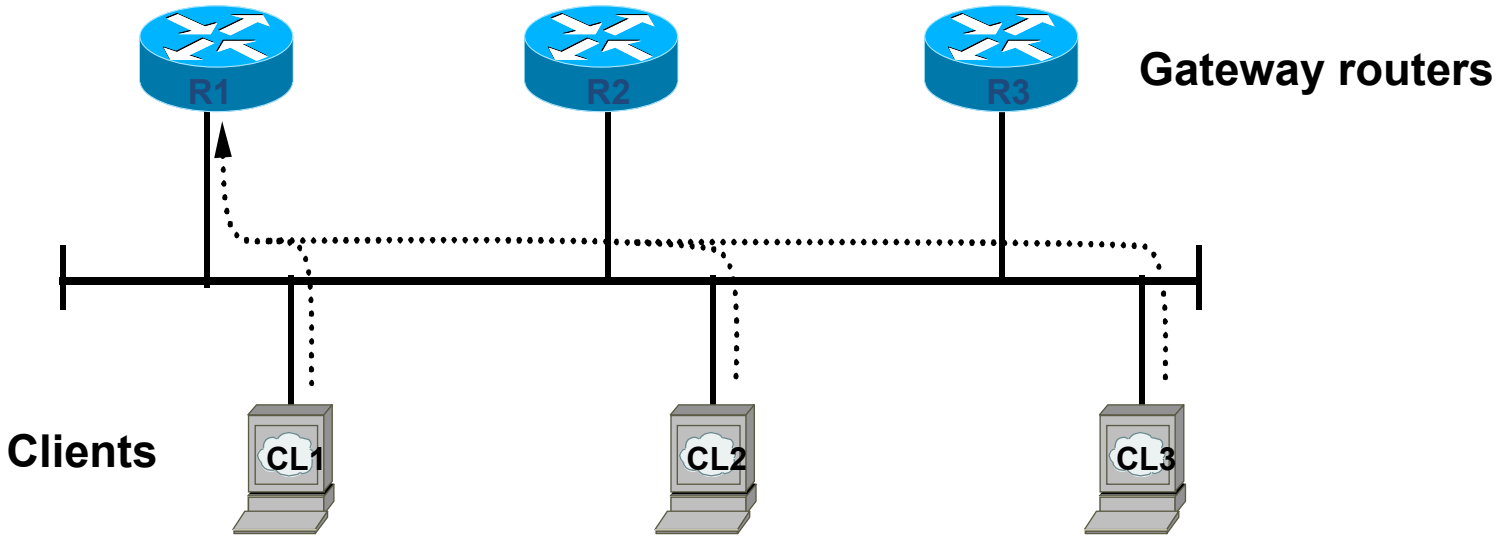
HSRP

- A group of routers function as one virtual router by sharing ONE virtual IP address and ONE virtual MAC address
- One (Active) router performs packet forwarding for local hosts
- The rest of the routers provide “hot standby” in case the active router fails
- Standby routers stay idle as far as packet forwarding from the client side is concerned

First Hop Redundancy with HSRP

R1- Active, forwarding traffic; R2, R3 - hot standby, idle

HSRP ACTIVE	HSRP STANDBY	HSRP LISTEN
IP: 10.0.0.254	IP: 10.0.0.253	IP: 10.0.0.252
MAC: 0000.0c12.3456	MAC: 0000.0C78.9abc	MAC: 0000.0cde.f123
vIP: 10.0.0.10	vIP:	vIP:
vMAC: 0000.0c07ac00	vMAC:	vMAC:



IP: 10.0.0.1	IP: 10.0.0.2	IP: 10.0.0.3
MAC: aaaa.aaaa.aa01	MAC: aaaa.aaaa.aa02	MAC: aaaa.aaaa.aa03
GW: 10.0.0.10	GW: 10.0.0.10	GW: 10.0.0.10
ARP: 0000.0c07.ac00	ARP: 0000.0c07.ac00	ARP: 0000.0c07.ac00

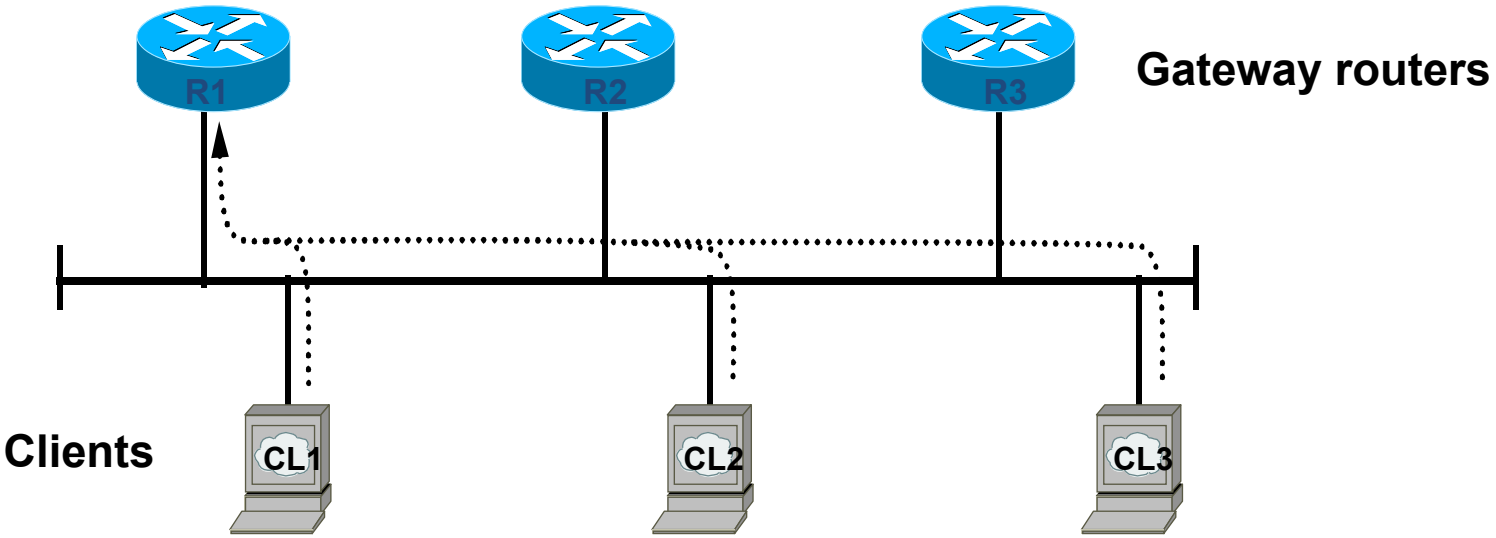
VRRP

- Very similar to HSRP
- A group of routers function as one virtual router by sharing ONE virtual IP address and ONE virtual MAC address
- One (master) router performs packet forwarding for local hosts
- The rest of the routers act as “back up” in case the master router fails
- Backup routers stay idle as far as packet forwarding from the client side is concerned

First Hop Redundancy with VRRP

R1- Master, forwarding traffic; R2, R3 - backup

VRRP ACTIVE	VRRP BACKUP	VRRP BACKUP
IP: 10.0.0.254	IP: 10.0.0.253	IP: 10.0.0.252
MAC: 0000.0c12.3456	MAC: 0000.0C78.9abc	MAC: 0000.0cde.f123
vIP: 10.0.0.10	vIP:	vIP:
vMAC: 0000.5e00.0100	vMAC:	vMAC:



IP: 10.0.0.1	IP: 10.0.0.2	IP: 10.0.0.3
MAC: aaaa.aaaa.aa01	MAC: aaaa.aaaa.aa02	MAC: aaaa.aaaa.aa03
GW: 10.0.0.10	GW: 10.0.0.10	GW: 10.0.0.10
ARP: 0000.5e00.0100	ARP: 0000.5e00.0100	ARP: 0000.5e00.0100

GLBP Defined

- A group of routers function as one virtual router by sharing ONE virtual IP address but using Multiple virtual MAC addresses for traffic forwarding
- Provides uplink load-balancing as well as first hop fail-over
- IP Leadership feature

GLBP Requirements

- Allow traffic from a single common subnet to go through multiple redundant gateways using a single virtual IP address
- Provide upstream load-balancing by utilizing the redundant up-links simultaneously
- Eliminate the need to create multiple vLANs or manually divide clients for multiple gateway IP address assignment
- Preserve the same level of first-hop failure recovery capability as provided by HSRP

First Hop Redundancy with GLBP

R1- AVG; R1, R2, R3 all forward traffic