

Application Delivery Controller and Virtualization (18CS7G2)
Instructor: Dr. Vishalakshi Prabhu H, Dept of CSE, RVCE, Bangalore-59
Author of notes: Dr. Vishalakshi Prabhu H, Date :01-Nov-2021
Contents of Unit - 3
Traffic Management: Core principles of traffic management, Multiprotocol Label Switching, DNS and global server load balancing, Content switching, AppQoE, TCP and SSL profiles, Introduction to Optimization and Security.

Core principles Network Traffic Management

Rapid growth and increasing requirements for service quality, reliability, and efficiency have made traffic engineering an essential consideration in the design and operation of large public Internet backbone networks. Internet traffic Engineering (TE) addresses the issue of performance optimization of operational networks. A paramount objective of Internet traffic engineering is to facilitate the transport of IP traffic through a given network in the most efficient, reliable, and expeditious manner possible.

Network Traffic is the density of data present in the network. Communication devices access resources and get requests to carry out some work. So, a lot of request, response, and control data increases Load on the network. Some devices may get delayed in their requirements.

Traffic Management is controlling network traffic requires limiting bandwidth to certain applications, guaranteeing minimum bandwidth to others, and marking traffic with high or low priorities. This exercise is called traffic management. Two major issues that have received attention in TE approaches are quality of service (QoS) and resilience.

NetScaler is in use in many large public organizations all over the world, doing traffic management for all their public web services. In large organizations, we might have over 100 different services masked behind one URL with different policies and features being used. This chapter tries to address some of the content related to Traffic Management as applied in WAN and from Citrix NetScaler perspective.

Organizational Requirements for managing Network Traffic?

1. To guarantee maximum bandwidth to mission critical applications.

2. To block music or video downloads.
3. To block music file sharing and avoid copyright infringement liability.
4. To delay investments in additional network capacity
5. Service monitoring - making sure things keep working.
6. Cost recovery - session times and traffic volumes can provide billing data.
7. Research - an improved understanding of what's happening should allow us to
8. Improve network performance.

Basic performance metrics:

Packet loss, Delay, Throughput, Error rate and Availability

Where should we manage traffic?

Usually deployed at the WAN edge of an enterprise site. The LAN-WAN juncture is also where both Internet and Intranet traffic enters and exits the enterprise.

Processes for Traffic Management

Once a problem of traffic management has been identified, the following tasks should be worked through in detail:

- ☐ Decide what is to be measured.
- ☐ Decide where to put traffic meters.
- ☐ Decide how data will be read from the meters and consolidated into a central database.
- ☐ Decide what reports will be generated, and how they will be published (on demand, via web page.).

Network Traffic Measurement Tools

Local Systems: NETSTAT, TCPDUMP, WIRESHARK

Remote End Systems: MIBS, IF-MIB, SNMP, RMON

Routers: NETFLOW (CISCO), LFAP (ENTERASYS)

Experiments have shown that these tools can currently handle up to several gigabits.

Traffic management approaches

Look from control theory point of view

- Open loop: - These act at source and destination to avoid problems from occurring
- Closed loop: - They act once the problem is on. Based on explicit & implicit feedback

Overview of congestion in network

Congestion: A state of a network resource in which the traffic incident on the resource exceeds its output capacity over an interval of time.

Congestion control: An approach to congestion management that attempts to remedy congestion problems that have already occurred.

Some of the techniques to control congestion are

- Traffic-aware routing
- Admission control
- Traffic throttling
- Load shedding

Quality of Service?

Quality of Service (QoS) is a set of technologies that work on a network to guarantee its ability to dependably run high-priority applications and traffic under limited network capacity.

Using QoS, businesses can prevent disruption in the form of IP packet loss, delays and jitter for VoIP (voice over IP), AoIP (audio over IP) and other real-time communications applications.

More often, QoS will be incorporated within the Service Level Agreement (SLA) given by their service provider. This guarantees a specific level of service. However, tools and techniques can be used independently to achieve QoS.

Approaches to Quality of Service

- Application requirements
- Traffic shaping
- Packet scheduling
- Admission control
- Integrated services
- Differentiated services

Some of the QoS tools in market are

- SolarWinds NetFlow Traffic Analyzer
- ManageEngine NetFlow Analyzer
- Paessler PRTG Network Monitor
- Ntopng

Traffic monitoring: Link-level versus Flow-level

Link-level traffic monitoring: Relies on SNMP statistics maintained by each router for each link. Management station polls each router frequently.

Advantages

- Simple to use and to deploy
- Tools can automate data collection/presentation
- Rough information about network load

Drawbacks

- No addressing information
- Not always easy to know the cause of congestion

Flow-level traffic monitoring: Routers identify flow boundaries. It does not cause problems on cache-based routers but more difficult on hardware-based routers. Higher-end routers rely on sampling. Routers forward the following information inside special packets to monitoring workstation.

Layer-3 flows

- IP packets with same source and destination prefix
- IP packets with same source and destination Autonomous System (AS)
- IP packets with same Border Gateway Protocol (BGP) next hop

Layer-4 flows

- one TCP connection corresponds to one flow
- UDP flows

Advantages are that it provides detailed information on the traffic carried out on some links.

Drawbacks

- Flow information exported to monitoring station
- CPU load is high on routers
- Disk and processing requirements on workstation that collects monitoring data
- Router-based flow aggregation can sometimes help

ROUTING OPTIMIZATION CONCEPTS IN IP NETWORKS

A. Destination-Based vs. Source/Flow-Based Routing

Two fundamentally different routing concepts exist, which strongly influence the optimization procedure and the achievable results: destination-based routing and source- or flow-based routing. Conventional routing protocols such as Open Shortest Path First (OSPF), Enhanced Interior Gateway Routing Protocol (EIGRP), Intermediate System to Intermediate System (IS-IS) follow the next-hop destination-based routing paradigm. Within each router the forwarding decision for an IP packet is based solely on the destination address specified in the packet header. No information about the origin or any other context of the packet is considered. Therefore, this routing procedure is simple and quite efficient. However, it imposes limitations on routing optimization, as illustrated in Figure 1.

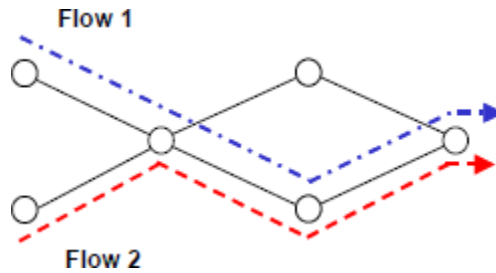


Figure 1

Whenever two traffic flows with the same destination cross each other's way they are merged and sent out over the same interface. This might cause traffic overload on some links, while other links are still only lightly utilized.

To overcome these limitations, new flow-based routing technologies such as Multiprotocol Label Switching (MPLS) have been developed. MPLS makes it possible to establish an overlay routing structure within the IP network – independently of the used routing protocol – and to set up explicit paths for individual traffic flows. Each MPLS-routed IP packet is marked with a special label, which the routers along the way consider for forwarding decisions. Thus, the routing pattern does not depend on the underlying routing protocol, but rather on the label and forwarding information that is stored in the MPLS routers. Instead of determining the path of a packet based on the destination address in a hop-by-hop manner, the path is now fixed by the router, which marks the packet with the appropriate label. This introduces a high degree of “routing freedom” as any desired routing pattern can be obtained.

B. Single-Metric vs. Dual-Metric Routing

In the case of destination-based routing protocols a router determines an outgoing interface based on metric values, which quantitatively describe the distance to a destination node. Most commonly, single additive metrics are assigned to every link and a shortest-path algorithm is used to determine the preferred path from each node to every other node in the network (“single-metric routing”). Link metrics often have physically relevant meanings such as “delay” or “cost”, they can also be used in a generic way purely for the sake of routing optimization. By setting appropriate link metric values, one can indirectly influence and, thus, optimize the routing scheme.

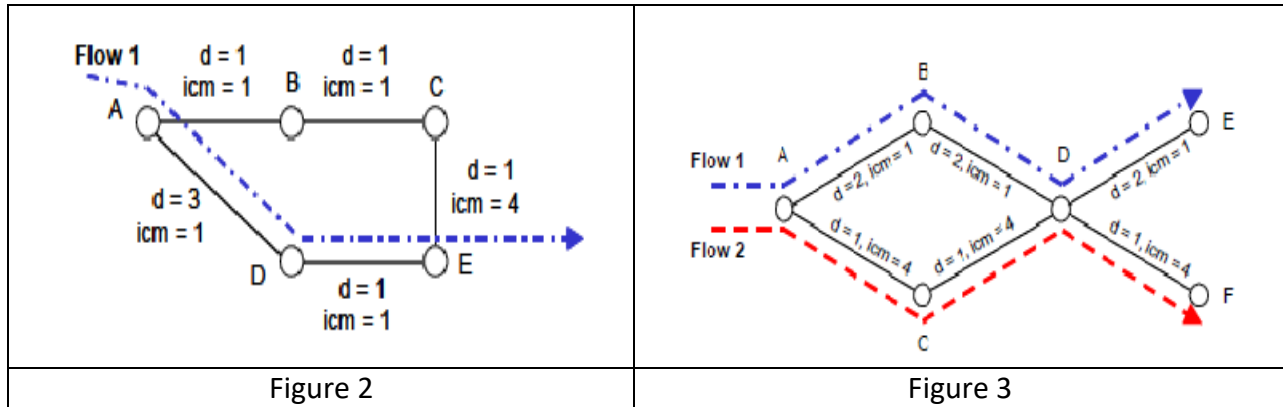
Routing schemes exist, which allow more than one metric considered when computing the length of a path towards a destination node (“multiple-metric routing”). One example is Cisco's routing protocol EIGRP, which incorporates four metric types. However, only two of them are used by default: one additive metric (“delay”) and one concave metric (“bandwidth”). In the following, we will focus on these two metric types and show how they can be used for routing optimization. The distance to a destination node is now computed by the normalized metric

formula

$$M = \frac{1}{\min_i(bw_i)} + \sum_i d_i = \max_i(icm_i) + \sum_i d_i ..$$

Parameter bw_i denotes the bandwidth of link i , while d_i refers to its static delay value. Thus, a router takes the sum of all delay values towards the destination node and adds a bandwidth component, which is the inverse of the smallest bandwidth along the path (“bottleneck”). From all possible path options, it selects the one with smallest path metric M . For further considerations we will refer to the bandwidth component as “inverse capacity metric” icm and take the maximum along the path instead of the reciprocal value of the bandwidth minimum.

Figure 2 illustrates the concept of bandwidth-delay sensitive routing. The distance metric M associated with path A-B-C-E is 7 (delay sum of 3 plus bandwidth component of 4), while path A-D-E has only an overall metric of 5. Therefore, router A selects router D as its next-hop neighbor towards E, and flow 1 is routed over A-D-E.



Depending on the values for d and icm , emphasis is either put on small overall delay (i.e., “short” paths), on high throughput, or on a mixture of both. In case metrics d are substantially larger than metrics icm , the overall path metric is mainly determined by delay values. Only when there are several alternatives with equal smallest delay sum, the bandwidth component really matters. The router then selects the outgoing interface with the largest possible throughput (“widest-shortest path”). In the opposite case ($icm \gg d$), high throughput paths are preferred, and delay is mainly used to break ties (“shortest-widest path”). Whenever the two-link metrics are of the same order, no clear preference is given to either one of them. Link metrics are then used in their most generic form as means of routing optimization without any physically relevant significance.

Routing optimization based on dual-metric protocols has the potential to achieve better results than its single-metric counterpart. Since the second metric introduces some more flexibility, a greater variety of routing patterns can be realized. Note that it is always possible for dual-metric protocols to produce single-metric routing patterns. Simply setting all icm values to the

same constant c would result in a single-metric shortest-path computation with metric function $(M-c)$.

The benefit of dual-metric protocols can be demonstrated by means of the network scenario in Figure 3. Assume we have two traffic flows with different destinations, whose paths have several nodes in common. Let A be the first node where the two flows come together, and D be the last common node on their way. While single-metric shortest-path routing would merge the flows at node A and send both either via B or via C , dual-metric routing protocols can achieve the flow pattern given in the figure. For flow 1, the chosen path has a total metric of 7, while the link metrics along the route via C would sum up to 8. For flow 2 the situation is different and the total metrics of the upper and the lower path are 9 and 7, respectively. The trick is to use the inverse capacity metric to make one path option appear more costly for one traffic flow, while for the other flow a larger icm value has no extra effect (since it already experiences high icm values on further links, which the two flows do not share). However, the actual gain in QoS, which EIGRP can achieve over OSPF, depends greatly on the network topology and the traffic demands.

C. Equal-Cost Multi-Path ECMP

Another possible feature of routing protocols, which influences the optimization process, is load sharing. In destination-based routing protocols this capability is often implemented in form of the “equal-cost multi-path” concept. Whenever a router can reach a destination node via several paths with equal metric sums, it splits up the traffic evenly across all corresponding outgoing interfaces. Note, if dual-metric routing protocols are used, “equal cost” does not necessarily mean that all metric components are the same on all load-sharing paths. While metric combinations of several paths might be equal, their delay and bandwidth portions could be quite different.

Shortcoming of conventional IP systems

One shortcoming of conventional IP systems is the inadequacy of measurement functions. For example, a traffic matrix, which is a basic data set needed for traffic engineering, is difficult to estimate from interface statistics on IP routers.

The limitations of intradomain routing control functions are another issue with conventional IP systems. Interior Gateway Protocols (IGPs), such as Intermediate System–Intermediate System (IS-IS) and Open Shortest Path First (OSPF), commonly used to route traffic within autonomous systems (AS) on the Internet, are topology-driven and employ per-packet progressive connection control. Each router makes independent routing decisions using a local instantiation of a synchronized routing area link state database. Route selection is based on shortest path

computations using simple additive link metrics. This approach is highly distributed and scalable but flawed.

The flaw is that these protocols do not consider the characteristics of offered traffic and network capacity constraints when making routing decisions. This results in subsets of network resources becoming congested, while other resources along alternate paths remain underutilized.

Multi-Protocol Label Switching (MPLS)

MPLS traffic engineering provides an integrated approach to traffic engineering. Recent developments in multiprotocol label switching open new possibilities to address some of the limitations of IP systems concerning traffic engineering. Although MPLS is a relatively simple technology (based on the classical label swapping paradigm), it enables the introduction of sophisticated control capabilities that advance the traffic engineering function in IP networks. A particularly interesting aspect of MPLS is that it efficiently supports origination connection control through explicit label-switched paths (LSP).

An explicit LSP is one whose route is determined at the origination node. Origination connection control permits explicit routes to be established which are independent of the destination-based IP shortest path routing model. Once an explicit route is determined, a signaling protocol is then used to install the LSP. Through explicit LSPs, a quasi-circuit switching capability is superimposed on the IP routing model. When deployed in IP over SONET or IP over DWDM configurations, the traditional L3 and L2 functions are virtualized in one network element called the label switching router (LSR). Fewer network elements are required than with overlay alternatives, reliability is increased, and operating costs and queuing delays are reduced.

Additionally, MPLS simplifies network architecture and network design relative to the Overlay model. When MPLS is combined with differentiated services (DiffServ) and constraint-based routing, they become powerful and complementary abstractions for quality of service (QoS) provisioning in IP networks.

MPLS is best summarized as a “Layer 2.5 networking protocol”.

In the traditional OSI model:

- Layer 2 covers protocols like Ethernet and SONET, which can carry IP packets, but only over simple LANs or point-to-point WANs.
- Layer 3 covers Internet-wide addressing and routing using IP protocols.
- MPLS sits between these traditional layers, providing additional features for the transport of data across the network.

In a traditional IP network:

- Each router performs an IP lookup (“routing”), determines a next-hop based on its routing table, and forwards the packet to that next-hop.
- Rinse and repeat for every router, each making its own independent routing decisions, until the destination is reached.

MPLS does “label switching” instead:

- The first device does a routing lookup, just like before:
 - But instead of finding a next hop, it finds the destination router.
 - And it finds a pre-determined path from “here” to that final router.
- The router applies a “label” based on this information.
- Future routers use the label to route the traffic without needing to perform any additional IP lookups.
- At the destination router, the label is removed, and the packet is delivered via normal IP routing.

Originally, it was intended to reduce IP routing lookups. Label switching (or “tag switching”) lookups use exact matching. The idea was to have only the first router do an IP lookup, then all future routes in the network could do exact match “switching” based on a label. This would reduce load on the core routers, where high-performance was the most difficult to achieve, and distribute the routing lookups across lower speed edge routers. The ability to control where and how traffic is routed on your network helps to manage capacity, prioritize different services, and prevent congestion.

MPLS Working – Basic Concepts

Label Switched Path (“LSP”)

- One of the most important concepts for the actual use of MPLS.
- Essentially a unidirectional tunnel between a pair of routers, routed across an MPLS network.
- An LSP is required for any MPLS forwarding to occur.

Label Edge Router (“LER”) or “ingress node”.

- The router which first encapsulates a packet inside an MPLS LSP.
- Also, the router which makes the initial path selection.

Label Switching Router (“LSR”) or “transit node”

- A router which only does MPLS switching in the middle of an LSP.

Egress Node

- The final router at the end of an LSP, which removes the label.

P – Provider Router

- A core/backbone router which is doing label switching only.
- A pure P router can operate without any customer/Internet routes at all.
- This is common in large service provider networks.

PE – Provider Edge Router

- A customer facing router which does label popping and imposition.
- Typically has various edge features for terminating multiple services like Layer 3- Virtual Private Network (L3VPN), L2VPN / Pseudo wires and Virtual Private LAN Service (VPLS).

CE is the “Customer Edge”, the customer device a PE router talks to.

MPLS Signaling Protocols

To use an LSP, it must be signaled across your routers. An LSP is a network-wide tunnel, but a label is only a link-local value. An MPLS signaling protocol maps LSPs to specific label values.

There are two main MPLS routing protocols in use today:

- Label Distribution Protocol (“LDP”): A simple non-constrained (doesn’t support traffic engineering) protocol.
- Resource Reservation Protocol with Traffic Engineering (“RSVP-TE”): A more complex protocol, with more overhead, but which also includes support for traffic-engineering via network resource reservations.

Most complex networks will need to use both protocols. LDP is typically used by MPLS VPN (data transport) services. But RSVP-TE is necessary for traffic engineering features. Most networks will configure LDP to tunnel inside RSVP.

MPLS Label Stacking

MPLS labels can also be stacked multiple times. The top label is used to control the delivery of the packet. When destination is reached, the top label is removed (or “popped”), and the second label takes over to direct the packet further. This is illustrated in Figure 4.

Some common stacking applications are:

- VPN/Transport services, which use an inner label to map traffic to specific interfaces, and an outer label to route through the network.
- “Bypass” LSPs, which can protect a bundle of other LSPs to redirect traffic quickly without having to completely re-signal every LSP, in the event of a router failure.

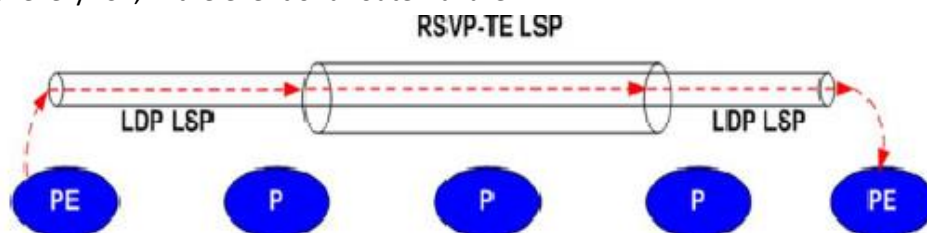


Figure 4

Traffic trunks

One feature of traffic engineering in IP networks is the traffic trunk concept. A traffic trunk is an aggregation of traffic belonging to the same class. It is essentially an abstract description of traffic that allows certain attributes of the traffic to be parameterized. It is independent of the

underlying technologies. The problem of mapping traffic trunks onto a given network topology is a central issue of traffic engineering. In MPLS networks, traffic trunks are mapped onto the Network topology through the selection of routes for explicit LSPs. The terms LSP tunnel and traffic engineering tunnel (te-tunnel) are commonly used to refer to the combination of traffic trunk and explicit LSPs in MPLS.

LSP tunnels allow the performance of an operational network to be optimized in various ways. For example, if congestion problems caused by suboptimal routing are detected, LSP tunnels can be rerouted to alleviate the problem. LSP tunnels can be parameterized, and network resources can be allocated to them based on those parameters. Multiple LSP tunnels can be created between two nodes, and the traffic between the nodes divided among the tunnels according to some local policy. LSP tunnels permit the introduction of flexible and cost-effective survivability options.

Statistics derived from LSP tunnels can be used to construct a rudimentary traffic matrix. LSP tunnels can be used to redistribute traffic to address congestion problems caused by shortest path IGPs.

- ❑ Remember, an LSP is a “tunnel” between two points in the network. Under RSVP, each LSP has a bandwidth value associated with it. Using constrained routing, RSVP-TE looks for the shortest path with enough available bandwidth to carry a particular LSP.
- ❑ If bandwidth is available, the LSP is signaled across a set of links. The LSP bandwidth is removed from the “available bandwidth pool”. Future LSPs may be denied if there is insufficient bandwidth. They’ll ideally be routed via some other path, even if the latency is higher.
- ❑ Existing LSPs may be “preempted” for new higher priority LSPs. You can create higher and lower priority LSPs, and map certain customers or certain traffic onto each one. This isn’t traditional QoS, no packets are being dropped when

MPLS Fast Reroute

MPLS Fast Reroute improves convergence during a failure. By pre-calculating backup paths for potential link or node failures.

In a normal IP network

The best path calculation happens on-demand when a failure is detected. It can take several seconds to recalculate best paths and push those changes to the router hardware, particularly on a busy router. A transient routing loop may also occur, as every router in the networks learns about the topology change.

With MPLS Fast Reroute

The next best path calculation happens before the failure occurs. The backup paths are pre-programmed into the router FIB awaiting activation, which can happen in milliseconds following failure detection. Because the entire path is set within the LSP, routing loops cannot occur during convergence, even if the path is briefly suboptimal.

DNS and Global Server Load Balancing (GSLB)

Load balancing is a technique that is used for distributing the workload evenly across computing machines, networks, processing units etc. It basically enhances utilization of resources and enables maximum throughput with minimum response time hence avoiding overloading of a single server.

An extension of the same technology is the Global Server Load Balancing (GSLB). With this technology the Internet traffic can be distributed among different data-centers located at different locations of the planet. For instance, large organizations such as Facebook, eBay, and Microsoft use this technology to load-balance their web services. This might be for proximity reasons, because a user might be redirected to the closest available resource, or to keep redundant services available in case of datacenter failures.

This technology is highly efficient in avoiding local downtimes. There is a Master GSLB which monitors the health and responsiveness of the Slave sites, which in turn identifies the sites that is available and can offer the best response.

Ideally GSLB would include:

1. Redirection of requests to other closest sites (domain controller -DC) in case the usual DC or ~~server~~ does not respond.
2. Forwarding the visitor requests to the site that is closer to the place from where the request is raised from a geographic point of view.
3. In case a threshold is reached at a site, the requests are forwarded to other site which may be at a different geo-location. Its algorithm would calculate the shortest distance from when the request has been raised.

GSLB is based upon the use of DNS. So for instance when a user wants to go to www.citrix.com, that user's DNS client will do a number of queries against the different DNS servers, until it gets a response from an authoritative source for that domain.

An example might be where NetScaler, which is the authoritative nameserver for that domain, responds with an A-record to that user that is one of the load-balanced vServers attached to the domain. Before that record is handed out, NetScaler has done an

evaluation of the health state of the different services the user is trying to access, using SNMP based load monitors, **Metric Exchange Protocol (MEP)**, and explicit monitors.

GSLB also consists of different components:

- GSLB sites (that represent a geographical location or datacenter)
- GSLB services (that are linked to a load-balanced vServer)
- GSLB vServers (that consist of multiple GSLB services served from GSLB sites)
- Domains (that on NetScaler are either authoritative or act as a proxy on its behalf)
- MEP used between nodes in each site to exchange information about the state and load on the site

MEP is a proprietary protocol used by NetScaler to communicate different GSLB site metrics, network metrics and persistence info to the other GSLB sites. Communication with MEP happens on port 3011 or 3009 for secure communication. If we want to use load-balancing methods such as proximity or RTT, we need to have MEP enabled; if not, it will fall back to round robin.

Figure 5 depicts an example with a small design layout of GSLB using MEP.

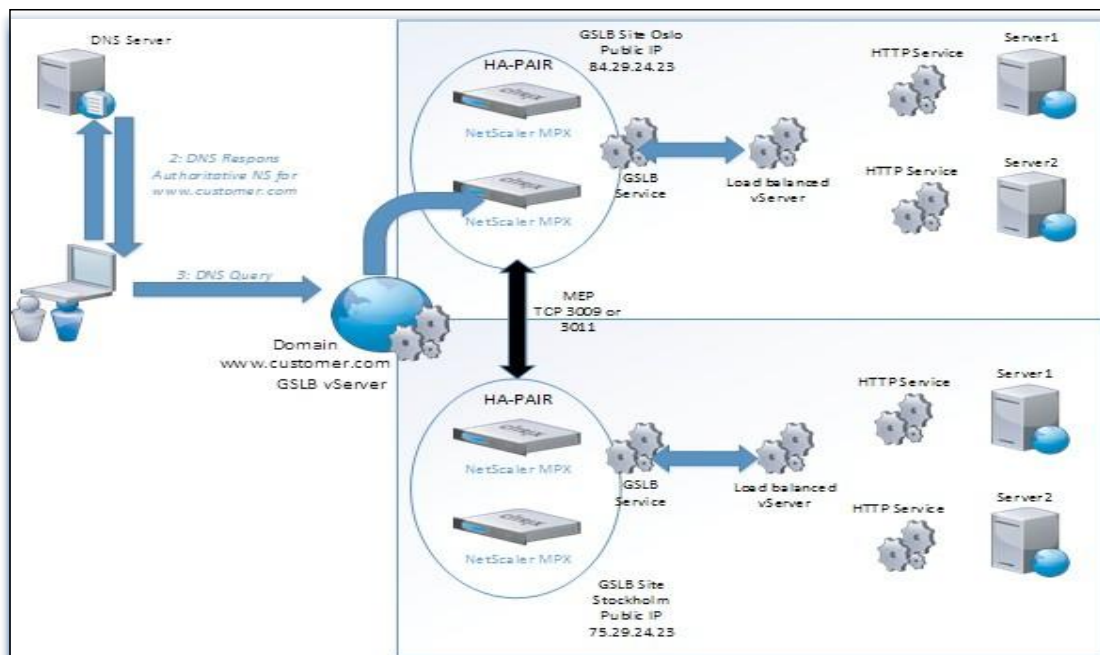


Figure 5

GSLB Scenarios and Requirements:

In GSLB, a website or services are hosted on different servers located at different geo-locations. Let's consider, a visitor is browsing www.sample-uk.com from US. In usual web hosting, this request would be sent straight to the server where it is hosted, and the request would be served from the UK server only. But in Global Load Balancing architecture, since the website is hosted on servers across different geo-locations, the user trying to access the website from US, would be served from the server that is in US. The GSLB in a regular time interval checks the health of the slave servers for providing the best response hence directing the traffic to those servers.

There aren't any special requirements for implementing GSLB for your website. You simply need to choose a server as you do in the usual way, but since you intend to apply load balancing across the globe, you are required to select servers in different locations. Your website data is placed on servers of your choice in all the selected locations.

Note: For Global Load Balancing, it is recommended to choose servers of equivalent configurations.

There are ideally two methods how GSLB can be configured:

- A. GLB – Global Load Balancing
- B. DNS Load Balance

A. GLB – Global Load Balancing:

In the technique of GLB, we use a primary server also known as the Master Server and is connected to Slave Servers located in different geo-locations i.e., different Domain Controllers (DC's). Whenever a request is raised, the Master Server checks the location from where it was initiated. Then cookies are set, and it redirects the request to the nearest server location.

Process Flow:

- 1) The user (in this case user is in US) types the website URL www.sample-uk.com in the browser. Since global load balancing is applied, the request would first go to the Master server which is in UK datacenter.
- 2) The Master Server would send a query to the requester for querying IP address, upon confirmation of the user's IP address; cookies are set which contain every information on the user system.
- 3) This information is forwarded to the Master Server which is in the meantime checking the health and level of responsiveness of the slave servers.
- 4) Now the Master Server in the GSLB architecture would redirect the request to nearest Slave Server (i.e., in US).
- 5) The user gets a response and can view the website which is served from the US ~~server~~ and not the UK servers.

All these processes happen in a fraction of seconds without the knowledge of the users. A

constant synchronization takes place between Master and Slave servers across all the DC's.

B. Global DNS Load Balance:

Load Balancing can be fulfilled using DNS. In this the response to the DNS requests depends on requester's geographic IP location and the response from DNS. Let's consider a similar example as above: If a user requests for www.sample-uk.com, it is sent to GLB DNS Server which would check the IP of the visitor. Then it would identify the server that is closest to the location or to the one that has the minimum response time, and the traffic is forwarded or redirected.

Process Flow:

- 1) The user based in Malaysia tries to access the site www.sample-uk.com. In this case the DNS Request is sent to the ISP and further to the GSLB DNS Server.
- 2) Then, the GSLB DNS Server forwards the request to Name Servers of all the Countries or to the GSLB Server located various geo-locations.
- 3) Within fraction of second, the GSLB receives the response from every server, it then analyses the response time and identifies the Closest Server Location from the Users Request and then redirects it to that server.
- 4) The users in this way get connected to the closest server. In this scenario, the user request is initiated from Malaysia, hence the closest server as identified by GSLB is in India, and hence the website gets served from server located in the Indian DataCenter.

Balancing algorithms for GSLB

There are two options that are specific for GSLB: `STATICPROXIMITY` and `RTT`. `STATICPROXIMITY` can use a custom location database to determine where to send clients, or we create our own custom entries.

`RTT` is a dynamic proximity load balancing method that measures the delay between a client's local DNS server and the data resource it is trying to access. After it has measured the `RTT`, NetScaler then exchanges the metric with the other sites using MEP and then sorts out the `RTT` between the different sites and the client. Then it makes its load balancing decision and will dynamically redirect the client request to the GSLB site that has the lowest `RTT`. NetScaler uses different probes to determine the `RTT`. By default, it uses `PING` first; if that fails it tries a `DNS UDP` probe. If that does not work, it tries a `DNS TCP` probe.

If all probes fail or `RTT` metrics are not available, NetScaler will fall back to round-robin load-balancing.

If we want to change the order of the probes, we can do so using the following command:

```
set gslb parameter -ldnsprobeOrder DNS PING TCP
```

This might be necessary for instance if we do not allow external PING traffic via the firewall or UDP.

After we have configured the load balancing method, we must add a domain to the vServer. That can be done under the domain pane in the vServer options.

GSLB is a complex feature that might often lead to headaches if not properly configured. A good way to start troubleshooting a configuration is to look at the domains configured in the GSLB setup and look at the **visualizer**; Visualizer will give a graphical overview of how the configuration is for that domain name.

DataStream

The DataStream feature provides us with an intelligent mechanism for load balancing SQL traffic to backend endpoints. This might for instance be based upon READ and WRITE requests, where READ requests are forwarded to some dedicated backend SQL servers while WRITE requests are dedicated to a master cluster.

This can also be performed down to a database level, where we can load-balance specific databases, or can even be based upon other attributes such as usernames, packet size, and so on. The DataStream feature is only supported for MySQL and MS SQL databases.