Enhanced Content Delivery Network to Improve the QoE

[1]Sachendra Singh Solanky, [2]Sandra Brigit Johnson, [3]Vakkalagadda Eswar Praphul

[1]M.Tech Student SCSE, VIT University Chennai-600048,
[2]M.Tech Student SENSE, VIT University Vellore – 632014,
[3] M.Tech Student SITE, VIT University Vellore – 632014
[1]15sachendra@gmail.com, [2] sandrajohnson1990@gmail.com, [3] eswar.praphul@gmail.com

***Abstract*—Content Delivery Network (CDN) is a distributed system of network servers arranged for more effective delivery of content to end-users. They offer fast and reliable applications and services by distributing content to cache or edge servers located close to user. The primary goal of a CDN is to achieve scalability. It allows dynamic provisioning of resources to address flash crowds and varying traffic needs as well. This leads to the concept of virtualizing a CDN. When several instances of a CDN are integrated into a single box, it is called a virtual CDN which can further reduce the network latency. Partial Resource Caching is another feature that could also help in minimizing the latency. It maintains the integrity of the requested content from the source as well as the caches through the usage of etags. It is also essential to prioritize the traffic, based on its criticality which can be attained through traffic characterization using Differentiated Services Code Point (DSCP) header that replaces the existing TOS /TC field in the IPv4/v6 packets. Thus better Quality of Service (QoS) could be maintained with the customers as guaranteed by their corresponding Service Level Agreements (SLA) thereby ensuring minimum latency.**

***Index Terms*—CDN, Partial Resource Caching, Differentiated Services Code Point (DSCP), latency**

I. INTRODUCTION

A Content Delivery Networks (CDNs) provide services that improve network performance by maximizing bandwidth, improving accessibility and maintaining correctness through content replication. They offer fast and reliable applications and services by distributing content to cache or edge servers located close to users .A CDN has some combination of content-delivery, request-routing, distribution and accounting infrastructure. The content-delivery infrastructure consists of a set of edge servers (also called surrogates) that deliver copies of content to end-users. The request-routing infrastructure is responsible to directing client request to appropriate edge servers. It also interacts with the distribution infrastructure to keep an up-to-date view of the content stored in the CDN caches. The distribution infrastructure moves content from the origin server to the CDN edge servers and ensures consistency of content in the caches. The accounting infrastructure maintains logs of client accesses and records the usage of the CDN servers. This information is used for traffic reporting and usage-based billing [3][4].

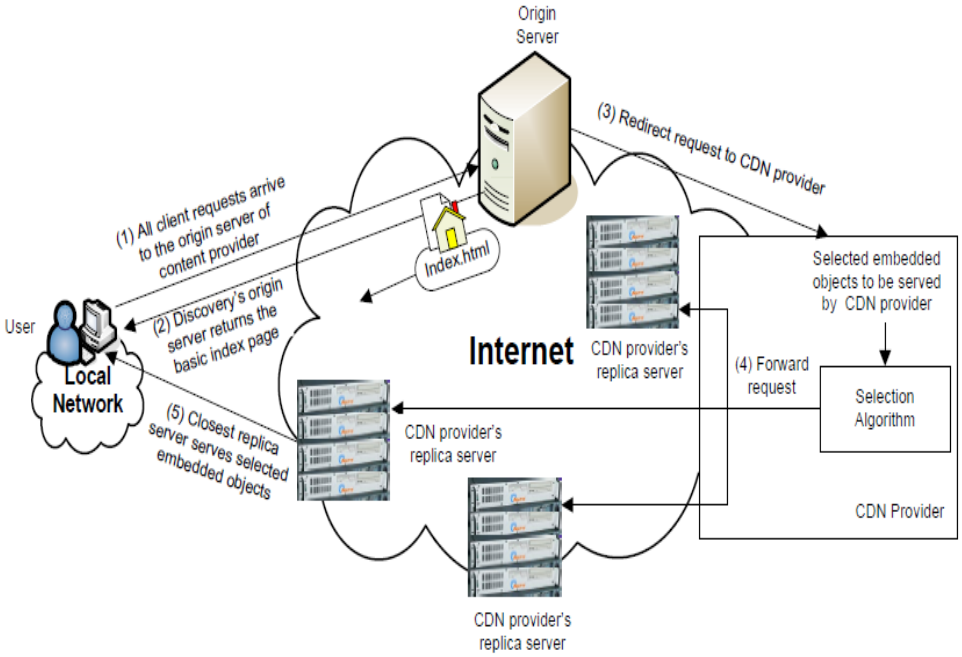

Fig 1: Request-routing in a CDN environment

Fig 1 provides a high-level view of the request-routing in a CDN environment. The interaction flows are: (1) the client requests content from the content provider by specifying its URL in the Web browser. Client's request is directed to its origin server; (2) when origin server receives a request, it makes a decision to provide only the basic content (e.g. index page of the Web site) that can be served from its origin server; (3) to serve the high bandwidth demanding and frequently asked content, content provider's origin server redirects client's request to the CDN provider; (4) using the proprietary selection algorithm, the CDN provider selects the replica server which is 'closest' to the client, in order to serve the requested embedded objects; (5) selected replica server gets the embedded objects from the origin server, serves the client requests and caches it for subsequent request servicing[5].

During hours of congestion, CDN still pose an inefficient delivery in terms of latency. As resources are not cached partially, the partial requests still need to be addressed by the origin server itself which will lead to an overloading. Hence the intermediate servers in the CDN domain should have a mechanism to partially cache the resources which will increase the overall CDN performance further. Additionally the processing time at these intermediates is very high considering the individual flow resource reservations that have to be maintained at every router. This paper adopts two measures to overcome these inefficiencies, which includes:

(i) Resource reservation maintained only at the ingress and egress nodes rather than at the intermediates. This can be implemented with the concept of Differentiated Services

(ii) Overloading of the origin server has to be reduced especially during hours of heavy traffic, hence by partially caching the resources at the intermediates, the requests could be served by itself rather than from the origin server themselves which can further reduce the response time.

Thus CDN customers can have better Quality of Experience (QoE) especially during hours of heavy data flow.

II. IMPLANTATION

(i) Differentiated Services Code Point(DSCP)

Differentiated services or DiffServ is a computer networking architecture that specifies a simple, scalable and coarse-grained mechanism for classifying and managing network traffic and providing quality of service (QoS) on modern IP networks. DiffServ can, for example, be used to provide low-latency to critical network traffic such as voice or streaming media while providing simple best-effort service to non-critical services such as web traffic or file transfers. DiffServ uses the 6-bit Differentiated services Field (DS field) in the IP header for packet classification purposes. The DS field and ECN field replace the outdated IPv4 TOS field.

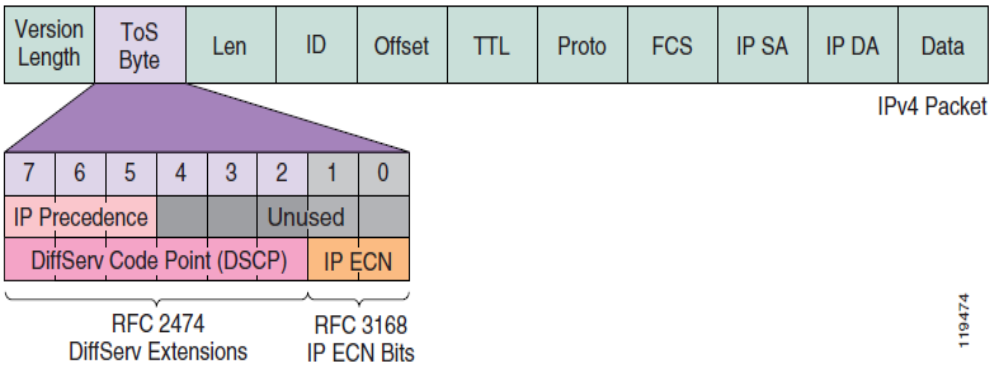

Fig 2: IPv4 packet

DiffServ operates on the principle of traffic classification, where each data packet is placed into a limited number of traffic classes, rather than differentiating network traffic based on the requirements of an individual flow. Each router on the network is configured to differentiate traffic based on its class. Each traffic class can be managed differently, ensuring preferential treatment for higher-priority traffic on the network. The premise of Diffserv is that complicated functions such as packet classification and policing can be carried out at the edge of the network by edge routers who then mark the packet to receive a particular type of per-hop behavior. Core router functionality can then be kept simple. No classification and policing is required. Such routers simply apply PHB treatment to packets based on the marking. PHB treatment is achieved by core routers using a combination of scheduling policy and queue management policy.

Network traffic entering a DiffServ domain is subjected to classification and conditioning. Traffic may be classified by many different parameters, such as source address, destination address or traffic type and assigned to a specific traffic class. Traffic classifiers may honor any DiffServ markings in received packets or may elect to ignore or override those markings. Because network operators want tight control over volumes and type of traffic in a given class, it is very rare that the network honors markings at the ingress to the DiffServ domain. Traffic in each class may be further conditioned by subjecting the traffic to rate limiters, traffic policers or shapers. Fig 3 gives the Logical View of a Packet Classifier and Traffic Conditioner.

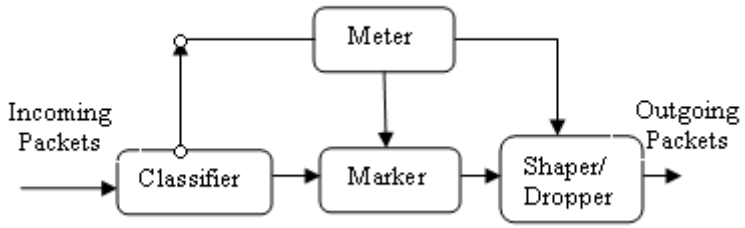

Fig 3: Traffic Classification and Conditioning

Traffic meters measure the temporal properties of the stream of packets selected by a classifier against a traffic profile specified in a Traffic Conditioning Agreement. Further the packet markers set the DSCP field of a packet to a particular code point, adding the marked packet to a particular DS behavior aggregate. Shapers delay some or all of the packets in a traffic stream in order to bring the stream into compliance with a traffic profile. A shaper usually has a finite-size buffer, and packets may be discarded if there is not sufficient buffer space to hold the delayed packets whereas the Droppers discard some or all of the packets in a traffic stream in order to bring the stream into compliance with a traffic profile. This process is known as "policing" the stream.

A network could have up to 64 (i.e. 2^6) different traffic classes using different DSCP. Commonly defined Per-Hop Behaviors (PHB) includes:

- *Default* PHB—which is typically best-effort traffic. The recommended DSCP for the default PHB is 000000_B (0).
- *Expedited Forwarding* (EF) PHB—dedicated to low-loss, low-latency traffic. EF traffic is often given strict priority queuing above all other traffic classes and occupies not more than 30% of the whole traffic.
- *Assured Forwarding* (AF) PHB—gives assurance of delivery under prescribed conditions. The AF behavior group defines four separate AF classes with Class 4 having the highest priority. Within each class, packets are given drop precedence (high, medium or low, where higher precedence means *more* dropping). The combination of classes and drop precedence yields twelve separate DSCP encodings from AF11 through AF43. Traffic that exceeds the subscription rate faces a higher probability of being dropped if congestion occurs. Rather than using strict priority queuing, more balanced queue servicing algorithms such as fair queuing or weighted fair queuing (WFQ) are used. To prevent issues associated with tail drop, more sophisticated drop selection algorithms such as random early detection (RED) are often used
- *Class Selector* PHBs—which maintain backward compatibility with the IP Precedence field.

The commonly defined Per hop Behaviors are shown in Table 1

DSCP Name	Binary	Decimal	IP Precedence
CS0	000 000	0	0
CS1	001 000	8	1
AF11	001 010	10	1
AF12	001 100	12	1
AF13	001 110	14	1
CS2	010 000	16	2
AF21	010 010	18	2
AF22	010 100	20	2
AF23	010 110	22	2
CS3	011 000	24	3
AF31	011 010	26	3
AF32	011 100	28	3
AF33	011 110	30	3
CS4	100 000	32	4
AF41	100 010	34	4
AF42	100 100	36	4
AF43	100 110	38	4
CS5	101 000	40	5
EF	101 110	46	5
CS6	110 000	48	6
CS7	111 000	56	7

Table 1: DSCP Class Definitions

With the DSCP definition especially during hours of heavy traffic, user can prioritize their packets in the DS domain thereby reducing the latency.

(ii) Partial Resource Caching

In the current implementations, the requested byte-range (partial content) is returned to the client from the origin directly (but never cached). This is not efficient means of acquisition when the requesting entity is for example a Video on Demand(VOD) cluster serving cable customers where a different set of delivery appliances(DA) are delivering the content to the client[7][8].

By employing Partial Resource Caching with CDN, as a user agent requests for a partial content, it gets cached to the delivery appliance either on the disc or RAM depending on whether its a live or Video on Demand content. Hence for the subsequent requests the content could be served from the caches if the content is not yet expired, else a revalidation is enabled. For optimizing the live content, the origin server hit rate can be reduced with the addition of three headers:

a. Volatile Storage
b. Delete on Expire (DoE)
c. Prioritize hit rate

When a user requests for a live content it gets cached in the memory. By introducing the above headers, the cached content becomes stale after DoE period. Hence it gets deleted after the specified time. Otherwise, when a new request arrives for the same content after the maxage period, a revalidation has to occur which will delete the already cached content and make a replacement in the cache.

By prioritizing the hit rate, latency gets further limited. Consider a scenario where different clients are making simultaneous requests for a live content Based on the request arrival, the initial request will be processed at the origin which will then cause buffering of packets at the delivery appliance. Depending on the bandwidth availability it can serve the requests either by unicast or multicast schemes. HTTP Live Streaming Protocol is generally used for content streaming based on the user adaptability. Apart from this, CDN supports Adaptive Bit rate schemes. Fig 4 depicts the flow mechanism for a live content delivery during busy hours.

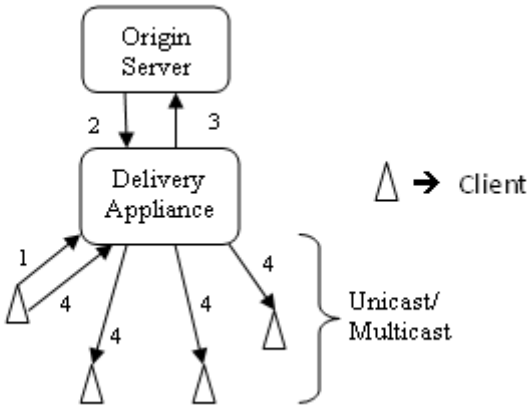

Fig 4: Prioritizing hit rate

III. RESULTS

The Performance measurement of a CDN is done to measure its ability to serve the customers with the desired content and/or service. Performance measurement offers the ability to predict, monitor and ensure the end-to-end performance of a CDN. Typically five key metrics are used to evaluate the performance of a CDN[5]. This includes:

- Cache hit ratio: It is defined as the ratio of the number of cached bytes versus total bytes requested. A high hit rate reflects that a CDN is using an effective cache policy to manage its caches.
- Reserved bandwidth: It is the measure of the bandwidth used by the origin server. It is measured in bytes and is retrieved from the origin server.
- Latency: It refers to the user perceived response time. Reduced latency signifies the decreases in bandwidth reserved by the origin server.
- Surrogate server utilization: It refers to the fraction of time during which the surrogate servers remain busy. This metric is used by the administrators to calculate CPU load, number of requests served and storage I/O usage.
- *Reliability:* Packet-loss measurements are used to determine the reliability of a CDN. High reliability indicates that a CDN incurs less packet loss and is always available to the clients.

This paper use latency as the parameter for performance measurement. Consider a scenario where many clients are requesting content from Origin Server (i) with and without enabling DSCP and (ii) with and without enabling Partial Resource Caching. Here the performance is measured with respect to the parameter latency. Graph in figure 5 displays the results with and without implementing above discussed enhancements.

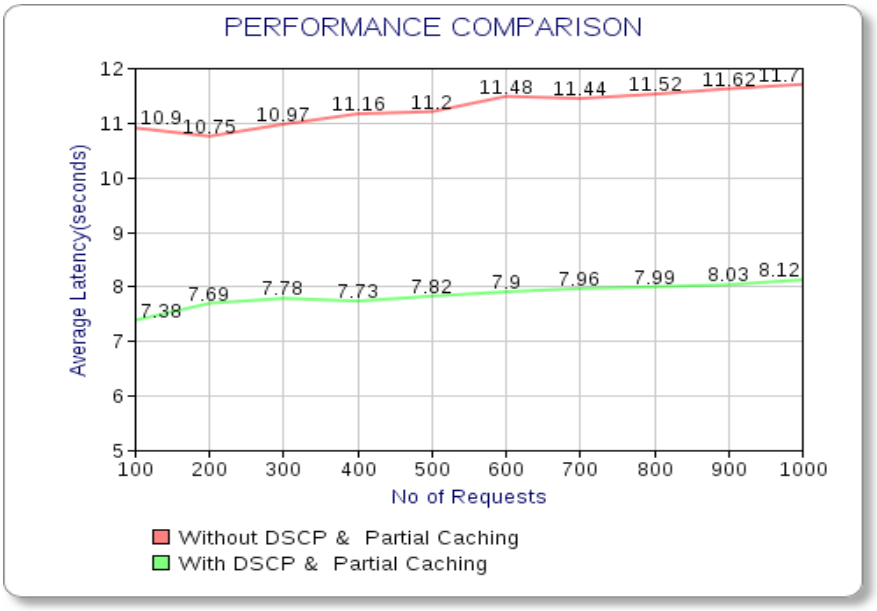

Fig: 5 Performance Comparisons

IV. ACKNOWLEDGMENT

We are highly grateful to our respective guides Dr. L. Jeganathan, Prof. Noor Mohammed V. ,Prof. N. Sivakumar who helped us during the entire research work done by us.

REFERENCES

[1] An Architecture for Differentiated Services RFC2474.

[2] "RFC 2475, Architecture for Differentiated Services", S. Blake et. al.

[3] D. Xu, S. S. Kulkarni, C. Rosenberg, "Analysis of a CDN-P2P hybrid architecture for cost-effective streaming media distribution," Multimedia Systems, Vol. 11, No. 4, 2006, pp. 383-399.

[4] L. Bent, M. Rabinovich, G. M. Voelker, Z. Xiao, "Characterization of a large web site population with implications for content delivery," World Wide Web, Vol. 9, 2006, pp. 505-536.

[5] R. Buyya, A. K. Pathan, J. Broberg, Z. Tari, "A case for peering of content delivery networks," IEEE Distributed Systems Online, Vol. 7, Issue 10, 2006. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.

[6] M. Pathan, R. Buyya, "Resource discovery and request-redirection for dynamic load sharing in multi-provider peering content delivery networks," Journal of Network and Computer Applications, Vol. 32, 2009, pp. 976-990.

[7] G. H. Petit, D. Deloddere, and W. Verbiest, "Bandwidth resource optimization in video-on-demand network architecture", *Proc. 1st Int. Workshop Community Networking Integrated Multimedia Services to the Home*, pp.91 -97 1994

[8] D. Deloddere W. Verbiest, and H. Verhille, "Interactive video on demand", *IEEE Commun. Mag.*, pp.82 -88 1994