Journal of Computing & Biomedical Informatics
ISSN: 2710 - 1606

ICASET
2024

Research Article
Collection: Intelligent Computing of Applied Sciences and Emerging Trends

Analyzing Cache Server Placement's Impact On SDN-Based Cooperative Caching

Muhammad Talha Jahangir[1], Aamir Hussain[2*,6], Muhammad Umar Farooq[3], Afaq Ahmad Khan[4], and Hasseeb Ahmad Bhutta[5]

[1]Department of Computer Science, MNS University of Engineering & Technology, Multan, Pakistan.
[2]Institute of Computing, Muhammad Nawaz Sharif University of Agriculture Multan, Pakistan.
[3]National University of Sciences & Technology, Islamabad, Pakistan.
[4]Punjab Tianjin University of Technology, Lahore, Pakistan.
[5]National University of Sciences and Technology, Islamabad, Pakistan.
[6]Department of Computer Engineering, Modeling, Electronics and Systems (DIMES), University of Calabria, Rende Italy.
*Corresponding Author: Aamir Hussain. Email: aamir.hussain@mnsuam.edu.pk

Academic Editor: Salman Qadri **Published:** April 01, 2024

Abstract: Big data introduces many challenges from the service provider's perspective such as increase in latency, delay variance, congestion and network utilization. These days' users require fast speed contents as well as high connectivity. Usersare service aware and dislike network delays. In an attempt to make networks less prone to delays, a bunch of researches do improve throughput enhancement but significant efforts are needed to minimize latency delay and delay variance. New network framework turns into the attractive obligation ofthis technological age. Latest network Infrastructures such as content delivery network (CDN) and software defined network (SDN) assist to determine today's user's concerns. CDN provide Contented Caching and it is valuable practice that Internet Service providers (ISP) utilize at this period. Different cacheserver cooperates with each other and optimally offers the demanded content to the consumer. Challenges in cooperative Caching are where to place the cache server and optimally route the content to client using cache server. In proposed approach, SDN-based Inter-ISP Caching Technique is aspiring to resolve the difficulty of cache server placement which diminishesthe data transmission latency and direction the cache content with optimized mode in the inter-ISP traffic. In proposed approach, it is assumed that different ISPs, which are present in same geographical location, communicate with each other. The specified approach introduces a lightweight algorithm named as Row Based Floyd Warshall algorithm to assist placing the cache server using an optimized method. Cache Controlleris a pioneering area to establish the cooperative caching andcompetently a method to reduce cache miss and recovernetwork throughput with the incidence of cache hit. NS3, whichis advance event-based network simulator is being used for this research. Simulation results demonstrate the efficiency ofSoftware Defined Network based Inter-ISP Caching beneath diverse network circumstances.

Keywords: Big data; Cooperative Caching; Cache Controller Placement; Row based Floyd Warshall algorithm; Software Defined Networking; ISP; CDN.

1. Introduction

In this age, a lot is linked through the internet. According to the [1], the global IP traffic was 7.5 ZB per year by 2023. If we extrapolate the trend from the report, we can estimate that the global IP traffic will be around 10.5 ZB per year by 2025, assuming a compound annual growth rate of 18%. This transfer is owed to swapping big data comprising of Cloud computing technologies, supercomputing, AI based Virtual Reality and other technologies. Degree of video streaming and substance distribution is considerably rising in modern era. For example, Netflix traffic crafts 30% of the US traffic split in climax period [2]. Big data has been extremely dominant in today's system. The traffic is outstanding to swap huge sum of stuffing e.g. photos and videos. In history, we just necessitate the connectivity, computing reserve is fewer resourceful and bandwidth expenditure of connection is not fairly fewer. Inside that occasion, the structure of the internet was too prearranged in the similar method. The people's ambition was little and he used to be pleased on just connectivity. But at the present the diversion of internet, it is totally distorted and the cause following this is the survival of big data. In this age, requirement of internet users has increased, they not only require connectivity but they want great amount of data so that they can do uploading and downloading. Internet should proportionally modify its structural design with today's order. The distributed architecture or nature of old network causes management and control issues [3]. A novel internet replica and fresh structure is required so that it can proficiently sprint today's system resourcefully. New rising expertise such as Content delivery network, Software defined system are altering the image of internet.

Software Defined Networking (SDN) is new-fangled structure and is being established day by day. Software defined Network (SDN) is a structure which grants a central controller during which we can simply direct the Cooperative caching. In customary networks, managing and controlling the data is done manually. In preceding network structural design, formation of routing and forwarding table is extremely expensive because it entails competent algorithm to decide finest trail for traffic. Whenever any changing occurs in network then the data in tables also change and for consistency each network device share updated table to each other. The procedure of distribution tables devours plenty of bandwidth which is not price effectual and as well sluggish along the business model. New structures are necessary which address this difficulty. If traditional networks are compared with SDNs, then SDNs makes it a lot easier to introduce new abstractions in networking, simplifying network management and provide the framework which support evolution [4]–[6]. SDN grants us a central controller which holds control plane whilst Open flow Switches hold data plane. The Central Controller which gives orchestration and manage all the network devices in extremely competent way. Controller recognizes all the outline of topology and if any altering happens in network topology, so all network campaign cross ways the system will not share routing table information in order amongst them but they converse with Controller and vice versa. Central controller competently informs pleasing network machine which is very price effectual and effortlessly convenient. SDN orchestration is incredibly useful for cache server placement to manage the traffic in a very efficient way. SDN Controller can drill the demanded content to pierce into exact cache server quite than the content navigates amongst some routers by extraordinary delay and then discover the destination. To handle the explosive growth of network traffic, many efforts have been performed to depress the bandwidth usage [7]-[10] notably by caching the big data in networks. Cache server cooperates with every other and shapes a group. The machinery of group of cache servers to allocate cache content is described as content delivery system. Content Distribution Network (CDN) is the expertise that has modified the financial system of Internet Service provider's (ISP's) and Internet Exchange Point's (IXP's). Now the Issues comes, how to put the controller and cache server in an ISP's network , how to direct the cache server and how to competently route the contented using SDN technology. According to the financial system, utilization of bandwidth is relative to charge. When ISP's lay cache server then they should remain this fixation in intelligence that cache hit exploited and cache miss diminished. If Cache miss happens, then bandwidth utilization should be negligible. Seeming at all these matters, SDN-based Inter-ISP Caching Technique is used for quicker release of Web substance to the end consumer. In this advance, "Row Based Floyd-Warshall algorithm" is initiated. Furthermore, projected loom as well offers the resolution to path and organizing the cache content in optimized way in SDN based ISP Cache Server.

1.1. Research Contributions

Now here numerous challenges when the cache server was placed in multiple ISPs by using an SDN network. The challenges are:

1) Placement of SDN Controller and Selection of border Open flow Switches in every ISP.

2) When Controller sprints numeral of system submissions such as routing app or firewalls, then it will be congested so how we shun congestion of Controller

3) Placement of cache server is a gruesome task. There are numerous constraints in system such as hop count, through coldness, bandwidth spending, traffic slide, link delay, to gauge the placement of cache server.

4) How to direct the cache memory in ISP's cache server and cache controller

5) What will be the Cache placement Strategy?

6) How to optimally itinerary the demand of content in SDN based Network?

The proposed approach faces all the challenges and introduces a lightweight algorithm which successfully finds the best optimal location of cache server. The main purpose of this paper is to find the best location of cache server in network. After placing cache server in best location, the specified approach tries to reduce the load of cache server by using cache controller. The main purpose of the cache controller is to provide the cooperative caching in the network. The given approach also provides mechanism of cache management and routing the content. Results reveals that the proposed approach which uses Row based Floyd Warshall algorithm provides much better results.

1.2. Paper Organization

Section II introduces the related work that has been carried out in this area. Section III goes into detail about the methodology that has designed and adopted. The proposed methodology has been put to test and results are compiled in Section IV. Conclusions and future works follows.

2. Literature Review

Web caching aims to store website in its cache server. There is a bottleneck present in bandwidth and latency [11]. Now a days web caching is not used due to high bandwidth consumption and the existence of https. In this era, network engineers and scientists are working to minimize the latency [12]. Today we are more concerned about content caching rather than web caching.

Yong [13] proposed a Cooperative cache that aims to optimize the traffic by caching big data in networks. They Proposed SDN-based Cooperative Cache Network (SCCN) for ISP networks, which employs SCCN Controller to coordinate distributed SCCN Switches (i.e., cache nodes). In order to maximize the cache hit ratio, they used relaxation and heuristic algorithm. In this Approach, the Openflow Switch contains a special storage for content. The real purpose of SDN is to separate the control plane and forward plane. In the given approach, Openflow Switch is overloaded because now the task of Openflow Switch is not just to forward flow but also manage and storage the content in the Openflow Switch.

Kim [14] suggested a centralized cache server instead of maintaining cache at each router in the network for ICN. The approach basically works on two principles:

• At least one router between two edge routers of a domain should be connected to the cache server

• The cache-hit ratio at the cache server should be larger than or equal to that in pervasive caching

Limitation of this approach is that the Openflow Switch is just connected with cache server. If the network size gets large, the single cache server will not perform well. Furthermore, in pervasive cache, if a router is processing one request, other packets will wait and if those other packets are of real time traffic, they will get delayed and may become useless.

Aubry [15] projected a method that enables the router to send requests to controller when they not have the route. The controller calculates the course and mounts the way on the routers in overturn arrange. Furthermore, a large network is divided into a number of domains and each domain has its own controller. This approach also does not address the issue of cache server placement.

Jan Badshah [16] approach has one limitation, they are using Information-centric networking whose main focus is to change the internet existing architecture. Information-Centric Networking - proposes a future Internet architecture that revolves about the contents being exchanged rather than the communication of hosts and network devices. While, in host centric architecture, a packet is delivered to a given destination address, communication is from one host to another host. Implementation of ICN architecture is very expensive and complex.

We studied OSI layer in many decades and all the network devices, network application, is designed and developed accordingly. If we change the internet architecture, we require to change everything i.e. network applications, network devices, header file of each layer and proposed new standard. ICN is good approach but it requires lot of time to successfully implement into current network.

3. Materials and Methods

The proposed Approach calculates the position of cache server in an ISP and relates routing configuration to increase the hit ratio. Major reason is to put the cache server in best location and then contrast with every placement in particular topology. The proposed approach is supported on SDN based cache. SDN divides the control and forwarding planes and it offers an inner power that is extremely useful for orchestration. SDN is a central controller which holds all information about ISP.

Content of different type is stored in each ISP cache server. The SCCN [13] approach contains the cache management in VSwitch. It makes Openflow Switch overloaded and cause delays. To overcome this issue, the suggested approach uses a simple Openflow Switch and introduces specified cache server in each ISP. To overcome the limitation of single cache server in Kim [14], suggested approach gives three types of SDN based servers such as:

1) Local ISP Controller
2) Cache server
3) Cache Controller

Local ISP server's task is finding the placement of cache server in its local ISP and add the flows into Openflow Switches which helps the traffic to direct towards its cache server. Cache server used to store content in its cache space and also add rules in Openflow Switches to connect with cache controller. Cache controller helps the ISP to coordinate with other ISP's cache controllers. In the proposed approach, placement of cache server is based on row-based Floyd Warshall algorithm. Floyd Warshall algorithm resolves the position of placement problem. Floyd Warshall algorithm presents us a matrix-based computation which can be a competent procedure via multicore processor. Furthermore, every ISP holds a huge figure of connectivity as it is linked with numerous structures such as Switches, routers, and servers. If topologies of an ISP can examine in the shape of diagram theory then ISP topologies is bottomed on dense graph somewhat than a sparse graph. Dijkstra algorithm is also a way to find out the shortest distance between source and destination but it cannot give us a shortest distance among all pair of nodes. In proposed approach, Row Based Floyd Warshall Algorithm is more effective than Dijkstra's Algorithm because it works best for dense graph problem while Dijkstra work is good in sparse graph problem, moreover it can also provide the shortest distance among all couple of nodes. As Mention earlier, the main focus is to find the location of cache server and then distribute content in multiple ISP's to desired location.

3.1. Placement of Cache Server

The Proposed Strategy is bottomed on Floyd Warshall algorithm and Row based Floyd Warshall algorithm. Floyd-Warshall algorithm is used to find the shortest distance of each node with other nodes. It is supposed that three ISP's are in same geographical location and every ISP holds one cache Server. Floyd Warshall algorithm offers a matrix (Dk ij) which gauges the shortest delay of every node of additional node so we obtain three matrices. Every Matrix belongs to all ISPs. Main aim is to decrease the computational time to process the packet and decrease the factor of latency. Multi-core processor can carry out matrix-based computation extremely quick and this information can help for future research works.

Row based Floyd Warshall algorithm assists us to decide which vertex or node is decided as a cache server. Regardless of whether a node can be a controller (θ), switch (S), a client, or a cache server (Cs). Users issue content requests (Req) to the Openflow switch(S). Every user can admittance some ISP Switch(S). The ISP topology was planned according to the graph (G= {V, E}). Here, "V" feels right to network nodes and "E" is the edges. The main purpose to decrease overall delays. It assumed that the bandwidth is same among all nodes and take delay as parameter (μ). Relating Row supported Floyd Warshall algorithm on every topology and located cache server on intended site. Row based Floyd Warshall algorithm is sprint on every ISP's controller to gauge the top trail to put the cache server in preferred ISP. ISP Controller also adds the flow in each Openflow Switch which tells the path of cache server in each ISP. The symbols and their description is documented in Table I.

Placement of cache server in proposed network model is big issue. Two Algorithms Floyd Warshall and Row-based Floyd Warshall algorithm resolve the matter of position of cache server. Floyd Warshall is specified in Algorithm 1 [17].

Table 1. Symbols used in Proposed Approach

Symbol	Description
θ	Controller
S	Openflow Switch
D_{ij}^{k}	Matrix of ith row & jth column giving shortest distance
Ω	Shortest delay
€	Row based vector
c'	Content
C_s	Cache server
b_o	Border Opeflow Switch
R_{eq}	Request for content
R_{es}	Response of content
F_t	Flow table in Switch
μ	Link state information (delay matric)
D_c	Database of controller
D_{cs}	Database of cache server
©c	Cache controller
U	Client who requested for content
I	Internet
Th	Threshold value
tr	time value

Whereas the modified Row-based Floyd Warshall is mentioned in Algorithm 2. We assume that Row-based Floyd Warshall algorithm is based on two limitations:

1) Number of Nodes

2) Metric value (delay) among Nodes Every ISP Controller runs the row-based Floyd Warshall algorithm as well as locates the cache server in its ISP. It ought to be renowned if the topology and metric value (delays value) is modified, then placement of cache server will as well vary.

Algorithm 1. Floyd-Warshall algorithm

Input: number of vertices {v1, v2, ...v*n*}

Input: D: Initial Matrix of each vertices with edges value

Output: Floyd Warshall Matrix ${D_{ij}}^{k}$

```
1    for k = 1 → n do
2        for i = 1 → n do
3            for j = 1 → n do
4                D_ij^k ← min(D_ij^(k-1), D_ik^(k-1) * D_kj^(k-1))
5            Diagonal element of D_ij^k = 0
```

Algorithm 2. Row based Floyd Warshall Algorithm

Input: number of vertices {v1,v2,...vn}

Input: D: Initial Matrix of each vertices with edges value

Output: Optimal placement of Cache server

```
1.   for i = 1 → n do
2.       for j = 1 → n do
3.           row[i] ← sum(d_ij^k )
```

```
        Ω ← min row[1] to row[n]
for i = 1 → n do
    if shortest_delay(Ω) = row[i] then
        select row[i] as Optimal placement
    else
        i←i + 1
```

Placement of cache server is not sufficient, cache memory management and traffic flow or traffic route also takes place as explained in proposed approach. The ISP's_Controller task is just to find the shortest path of cache server in each ISP and configure the flow entries of ISP cache server in Openflow Switch. Furthermore, if a few consumers approach in ISP and ask for to content during such Openflow Switch which is not directly linked with controller then we countenance latency. To resolve the above issue, controller and Openflow Switch is connected though Mesh Topology.

3.2. Remove the load on ISP Controller

Placement of cache server is not enough, cache memory organization and traffic flow or traffic route is as well the errands clarified in projected work. Junbi's approach [18] gives three main reasons why the loads of controllers must be taken into consideration whenever designing a placement policy:

1) Server capacity limitation

2) The delay of message processing in controller

3) Load balancing or Impact of failure of one controller to other controllers in network

It must be noted that, if the load of a controller approaches towards a threshold value, the message processing delay on the controller will rise largely depend upon result in [19]. Moreover, in some scenarios, the failure of largely-loaded controller may affect other controllers [20]. In the proposed topology, the concept of Cache_controller located in Internet Exchange Point (IXP) is introduced. The ISPs_Controller's job is now to discover the best possible trail of cache server in every ISP and arrange the flow entries of ISP cache server into Openflow Switch. ISP's Controller turns out to be overfull if it as well allocates the job of routing and running the management of cache memory. Furthermore, each ISP cache server should require to organize with other's ISP's cache server. To resolve this difficulty, Cache Controller is initiated. Today, each user wants less latency with fast internet connection. Cooperative cache has become a promising technique to optimize the traffic by caching big data in networks. Cache controller can reduce the weight on ISP controller and as well turn into the basis of cooperative caching as it is communal amongst every ISP's cache server as well as holds every sequence of all cache server content keen on its database. The purpose of Cache Controller is to insert the flow into cache server as exposed in Figure 1. It is suggested that every ISP's cache server as well as cache controller friendly through internet.

3.3. Managing the Cache Content

Placement of cache server is not enough but requires such instrument which competently itinerary the demanded transfer from side to side cache servers. Of course, it needs two main segment:

1) Organization of the cache Server memory

2) Optimally direct the demand of consumer currently primary argument is the administration of cache memory. Cache memory administration should have numerous features such as:

1) What should be the storage capacity of cache server and cache controller?

2) In which situation cache server competently put the content in cache server as well as cache controller?

3) If the cache server and cache controller recollection (amount) turn out to be filled then which cache placement strategy will be followed?

In given case, it is considered that the cache server and cache controller can handle bulk of requests, contain very fast memory to save the big data and there is no congestion. In proposed approach it is also suggested that the cache server and cache controller are placed in the ISP network. In the given approach, each ISP contains cache server and each cache server coordinates with each other. Cache server are placed in a distributed fashion. Various distributed cache systems were proposed in [21], [22]. Multi-cache infrastructure mainly consists of hierarchical cache, distributed cache and hybrid cache. Hierarchical Web cache is mentioned in the Harvest project [23]. Similar content placement schemes are explained in [24]–

[26], which purpose to reduce the average access costs. In the proposed approach, the cache server is initially unfilled and whenever any client asks for content, then the cache server demands the similar content from internet. On response cache server send the desirable content to its user and updates its information table without saving the content in its cache server. ISP cache server and cache controller obtain needs from consumer and uphold its database with information table as revealed in Figure 2.

Information table of ISP's cache server holds six field such as hash value of URL content, name of cache server, type of content, content state, counter and timer value. Hash value of URL content can help to search the content. Each content is divided into several chunks and each cache server can hold full content or several chunks of each content in distributed way. Type of content is placed in information database. Type of content contains the info of the content or specific part of content called chunk. Another field is ISPs cache server name. Multiple ISPs cache server send the content and each ISPs cache server has a unique name, so this field is very helpful to find the content of specific cache server. Another field is Content Cache; which determines whether the content is cached or not. The Counter contains the frequency of request. At any time any consumer demands for some kind of content, then the occurrence of ask for similar content located in Counter, for instance, in ISP1 consumer demands software content such as (video player) then cache server modifies its table with the IP address of demanded consumer, IP address of its cache server, sort of content and allocates counter with 1. After a little instance, a few additional consumers as well require for the similar video player then cache server immediately updates the information table by altering the defy worth from one to two. Cache controller too holds the similar in turn table similar to cache server but with little altering. In cache controller in turn table holds IP address of cache server (who propel demand to cache controller rather than consumer IP address).

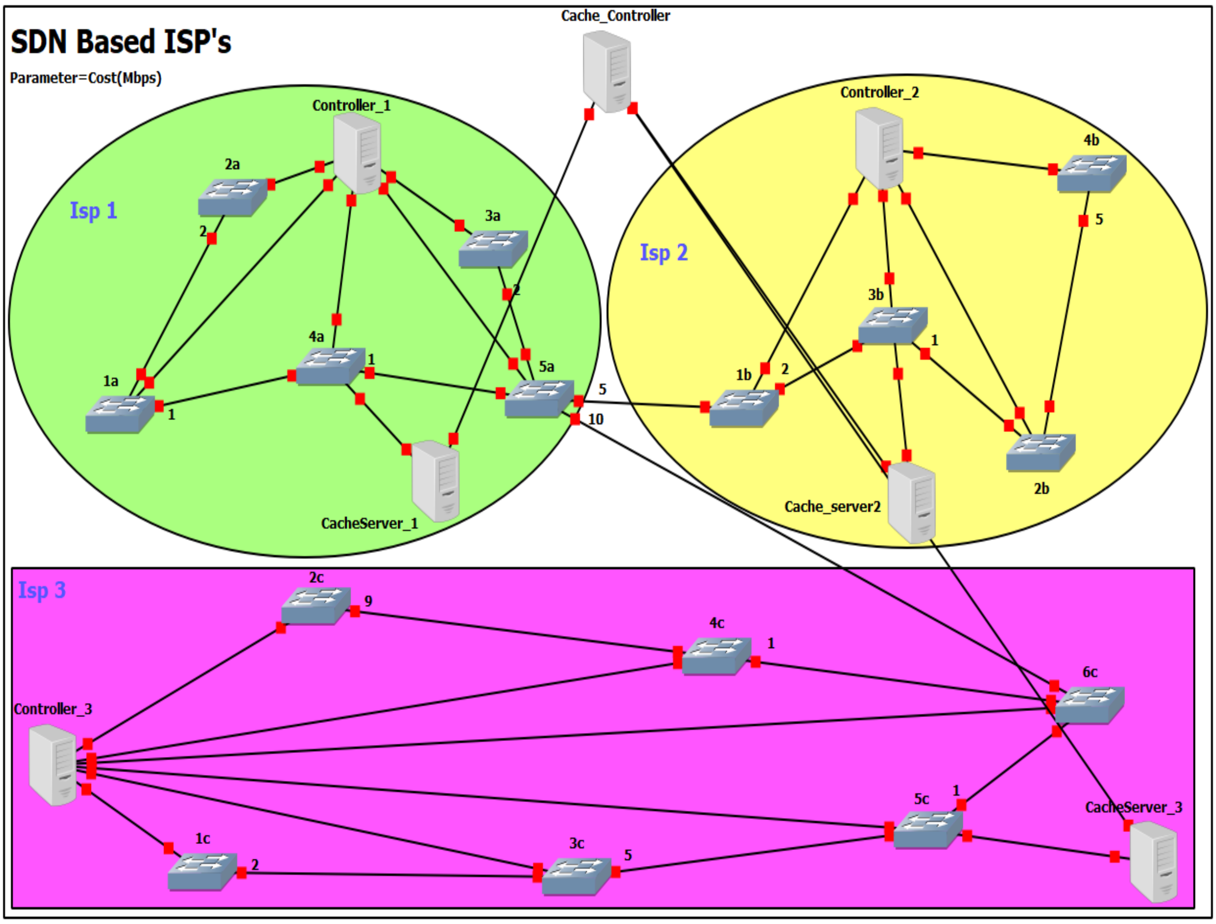
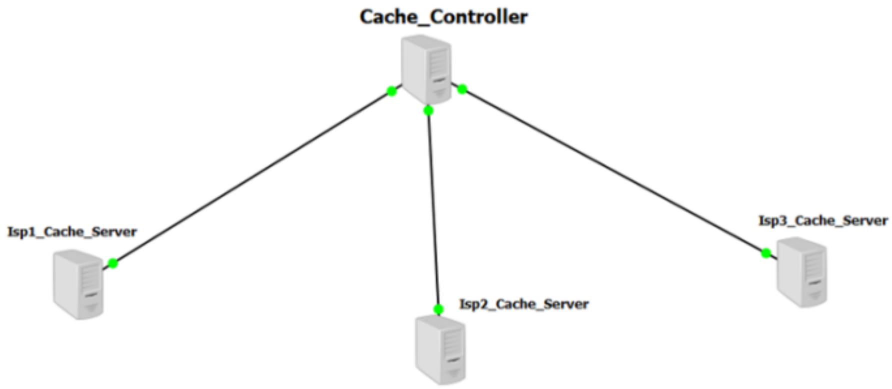


Figure 1. Introduction of Cache Controller in Proposed Topology



Hash Value of Content	ISPs Cache Server Name	Type of Content	Counter	Timer Value	Content State

Figure 2. Information table record Entry

It must be renowned that the cache server and cache controller have threshold values which is contrasted by a counter. The timer is as well-worn in the cache server and cache controller which informs the period in which servers recognize requirements from the consumer and following the conclusion of the timer if the threshold will be activated then the cache server and cache controller carry out exact kind of deed. Presume when cache server or cache controller begins, the timer is also initialized with zero signifies that it is well establish. Cache server and Cache controller limit the period of the timer. For instance, the cache server puts timer extent corresponding to two days and the threshold value is identical to a hundred. Inside two days, the cache server just collects the demand of the exact category of content allows "Type A" content as well as revises it in the information table with its admission. After completion of the duration of timer, if the counter value triggers the threshold, it means that the counter value will become equal to 100 or the number of requests approaches to hundred then cache server saves the content in its database and updates its counter value equal to zero. If the counter value is less than 100 then cache server sends the information of the content to the cache controller and initializes its counter value equal to zero.

Likewise, cache controller also puts its threshold value and counter value. The cache controller organizes with the ISP's cache server and obtains in information table evidence from the ISP's cache server. For instance, a cache controller puts a timer worth equivalent to five days and a threshold equivalent to two hundred. Within five days, a cache controller just obtains record of the exact kind of content from three ISPs' cache server. After five day when timer worth restrictions finish, the cache controller examines its database information table. The information table holds entries similar to the cache server IP address, cache controller IP address, kind of content, and its counter. Presume three consumers from dissimilar ISP needs a similar sort of content "Type B". User A from ISP1 propels 40 requirements for "Type A" content, user B from ISP2 propels 200 demands and user C from ISP3 propels 350 requirements. It should be renowned that, in an ISP, some consumers propel demand to cache controller on the behalf of cache server. Cache controller just counts the number of requests of specific type of content and match its value with its threshold value. In this example only ISP2 and ISP3 users meets the threshold value of cache controller. Now, cache controller sets its timer value zero and cache server2 and cache server3 content entry delete from its database. Cache controller automatically request the content from internet and sends content to desirable ISPs cache server to save it. Next time any user that requests for "Type B" content, then cache server does not need to request it from internet because cache controller already provides "Type B" content to its cache server. Cache controller becomes very useful because it saves bandwidth, increases throughput, decreases page load time and most important decreases end to end delay (latency).

3.4. Cache Replacement Strategy

In Yong Cui's approach [17], an increment recording mechanism is designed, which is based on the Least Recently Used (LRU) policy [27] to capture the popularity change of contents. In given approach, if the cache server is filled, then it employs the least recently used (LRU) rule for content substitute. In contrast, content stored within the ISPs cache servers and controller is mostly multi-media content. We therefore follow an adaptive TTL strategy to preserve cache coherence [13]. For simplicity, we assume that the medium is reliable and we ignore the transport layer functionalities.

3.5. Routing the Cache Content

The proposed methodology of traffic routing is mentioned in this subsection. Whenever the cache server gets any request from anywhere, it must share information with the cache controller. At this stage, it is assumed that cache is managed properly, and the system is ready to successfully route the traffic. The choice of content request routing selection exceeds the reach of this paper. It is therefore believed that switches should route the request according to the network routing protocol (e.g. shortest routing of the path) [28]. At the start, cache servers will be empty and start filling the cache of the cache servers with the consumption of time. Each ISP cache server is synchronized with the Cache Controller. When the cache server gets any request from the user it also shares the information of content with the shared cache controller (©c), cache controller updates the information in its database. The cache controller measures the frequency or counter of the requested content. When the cache controller meets its threshold value of the requested counter, it automatically requests the content using the internet. In response to the internet, the Cache controller automatically sends the desirable content to the cache server. The cache server stores a copy of the content and sends the content to the requested client through the corresponding switch.

For instance, the client (μ) such as ISP1_User from any ISP1 demands content (Req), it propels the demand for its ISP Switch such as 1A as shown in Figure 3. After getting demand, the Switch seems to up its onward table. The flow table of the Switch (FT) will have the flow regulation merely of its obtainable ISP cache server. The Cache Server holds the data as well as a database in a sequence of content. Switch 1A forwards the demand to its Cache Server. Now at this phase, multiple situations survive which are clarified below:

1) The Cache Server checks whether content is available in its cache server or not.

2) If the content is present in the cache server then it transmits the content(ç´) to the client with the response (Res) as shown in Figure 4.

3) If Content is present in another ISP, the cache server first checks the content in the cache controller.

Cache controller which provides the cooperative caching and holds the details of the content state of each ISP's cache server. If it is obtained, the attribute of the content state is equivalent to cached then the cache server does not send the request to the internet. Whilst it gets the content from the cache controller, it saves it in its memory and responds to the user as shown in Figure 5.

4) If the content is not present in the Local cache server and content is not present in the cache controller then two cases occur:

a) If the user requests content and the threshold value is triggered then the cache server forwards the request of the user to the internet. Internet communicates with the original server and replies to the cache server. The cache server saves the content and replies to the user as shown in Figure 6.

b) If the user requests content and the threshold value is not triggered then the cache server forwards the request of the user to the internet. Internet communicates with the original server. Instead of replying to the cache server, the internet directly sends the content to the user.

4. Results

The Specified approach uses NS3 real network traces for evaluating the proposed work. NS3 is a time-driven, event-based simulator that gives results at each respective event.

Table 2. .Experiment Metrics

Bandwidth	Traffic Type	MTU	Data Rate	Simulation time
2Mbps	Constant bit rate (CBR)	1000 byte	100Kb/s	3 Minute

In NS3, two significant modules OpenFlow 1.3 [29] and Flow Monitor [30] have been used to control SDN delays and to simulate the network environment respectively with the proposed approach.

A. Experimental Scenario In imitation and belief, the shift of consumer demand of content from Switch 1A to cache server and cache server is located according to Row based Floyd Warshall algorithm shown in Figure 8. In imitation and belief, the shift of consumer demand of content from Switch 1A to cache server and cache server is located according to Row based Floyd Warshall algorithm show in Figure 8. Table II also elaborates on the experimental metrics. After Adding the cache server and client link, the delay in Row based Floyd Warshall matrix is updated as 4A < 1A<5A<2A<5A.

Hence according to Floyd Warshall's algorithm, the cache server should be placed at Openflow Switch 4A. It is worth mentioning that the results are based on two factors:

1) Cache server without any placement algorithm

2) Cache server with Row-based Floyd Warshall placement algorithm.

SDN Controller and Openflow Switch communicate with each other because the controller adds flow to switch which in turn helps the Switch to re-route the traffic back to the user. SDN Controller and OpenFlow communication create constant latency (4.2μs) with each other.

4.1. Results

To demonstrate the performance of the cache server at every location in ISP1, different percentages of cache hits were used, as shown in Figure 10 and 11. First, the SDN-based ISP1 network performance is analyzed in which 100% cache hit is initially set as shown in Figure 4. Results at 100% cache hit produce much less delay which favors the good performance of SDN-based ISP networks. To observe the effect of a cache miss in an SDN-based ISP network, the requested content is traversed to the internet. In this scenario, two cases occur. Firstly, Figure 5 shows the effectiveness of cooperative caching. When some content is present in the cache server and some content is requested from internet, the cache server stores the copy of the content and forwards the request to the user such results as shown in Figure 6. To compare and elaborate the cache placement algorithm we placed the cache server at all the locations in ISP1 as shown in Tables V to VII.

4.2. Analysis

After analyzing the results, various aspects are revealed. Whenever a 100% cache hit occur it produces the best results. But in the rational environment, the chance of occurrence of a 100% cache hit is very rare. Results in Figure 11 show cooperative caching, which indicates that fewer end-to-end delays occur as compared to when the traffic is requested from the internet as shown in Figure 10. Cooperative caching performs much better than sending the request to the internet. An efficient mechanism of cooperative cache is introduced, and the results show that introducing the cache controller, which provides the efficient mechanism of cooperative cache produces much better results than the approach described in [28]. It is also the placement of a cache server in an ISP that affects the consequences of average end-to-end delay. In short, results conclude that cache server in SDN-based ISP networks increases the performance of the network by minimizing the average end-to-end delay, improving user experience, and elaborating the challenges in SDN-based cooperative caching ISP networks.

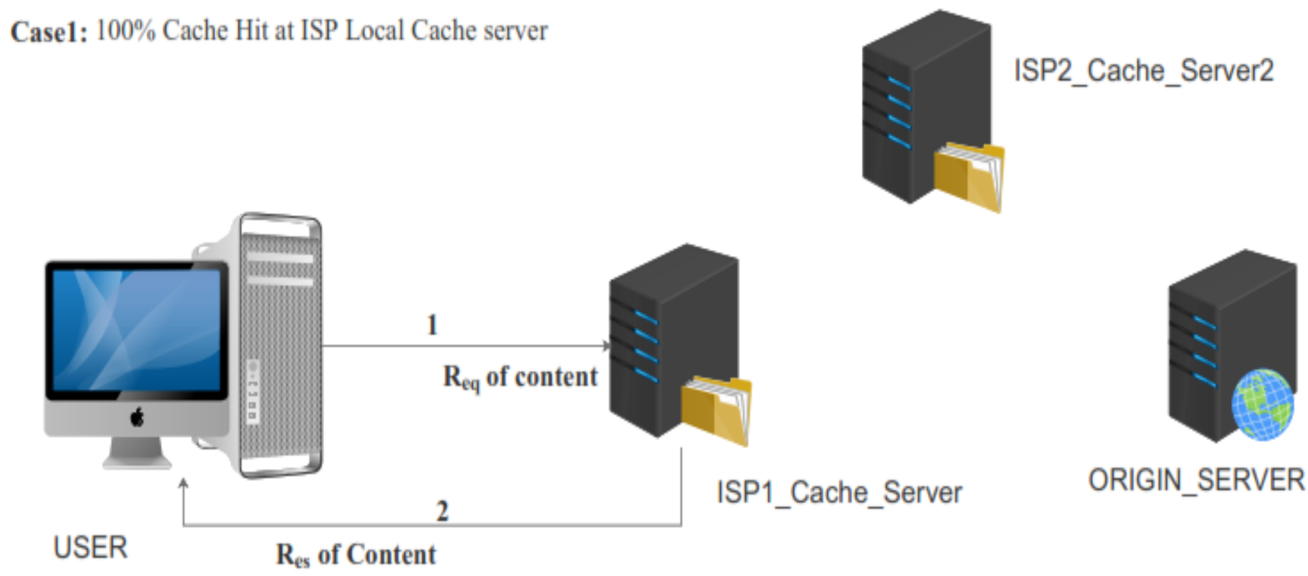


Figure 3. Proposed Scenario at 100% Cache Hit

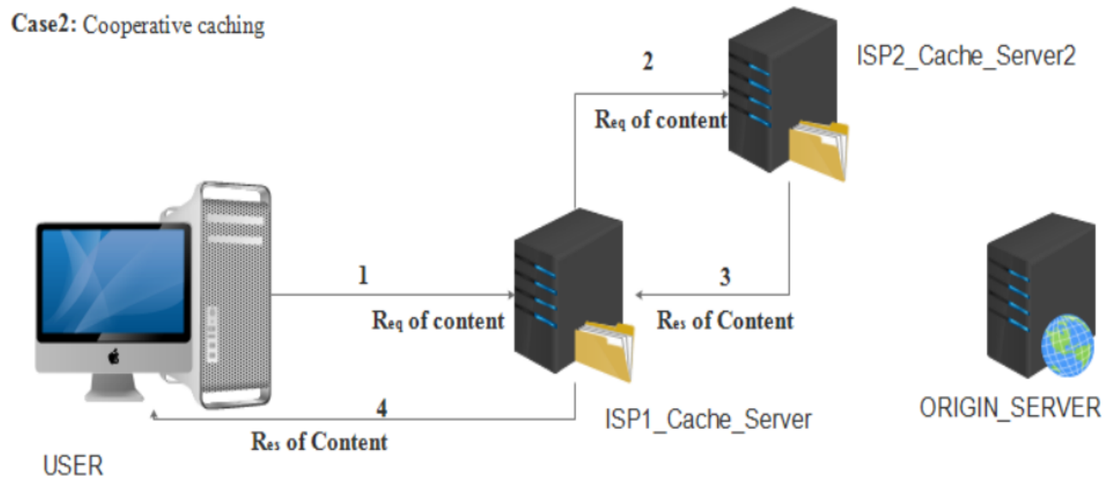


Figure 4. Cooperative Caching

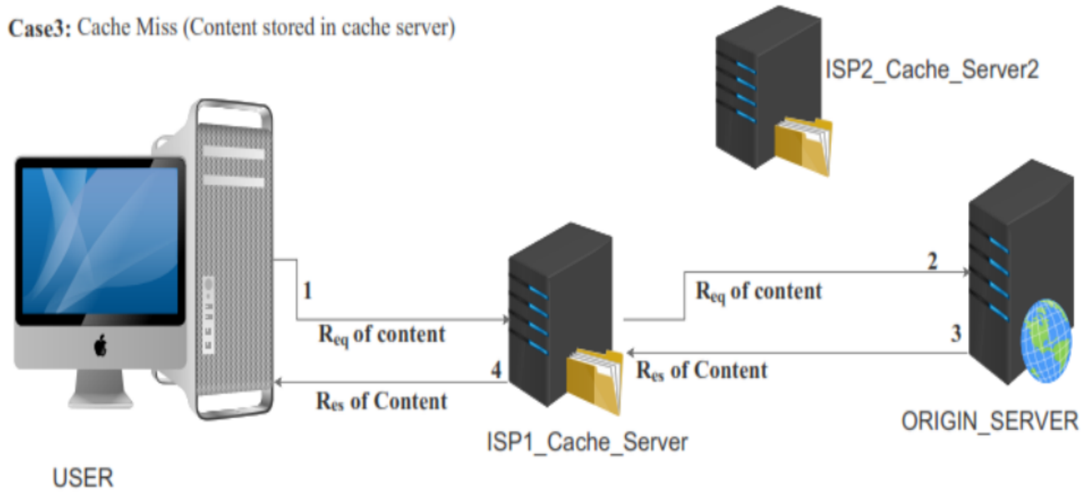


Figure 5. Proposed Scenario when content stored in cache server

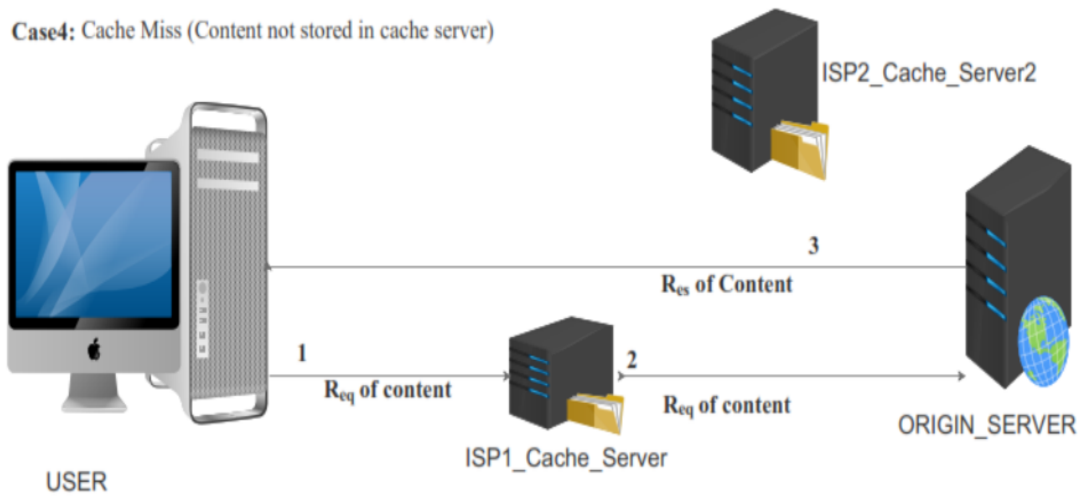


Figure 6. Proposed Scenario when content not stored in cache server

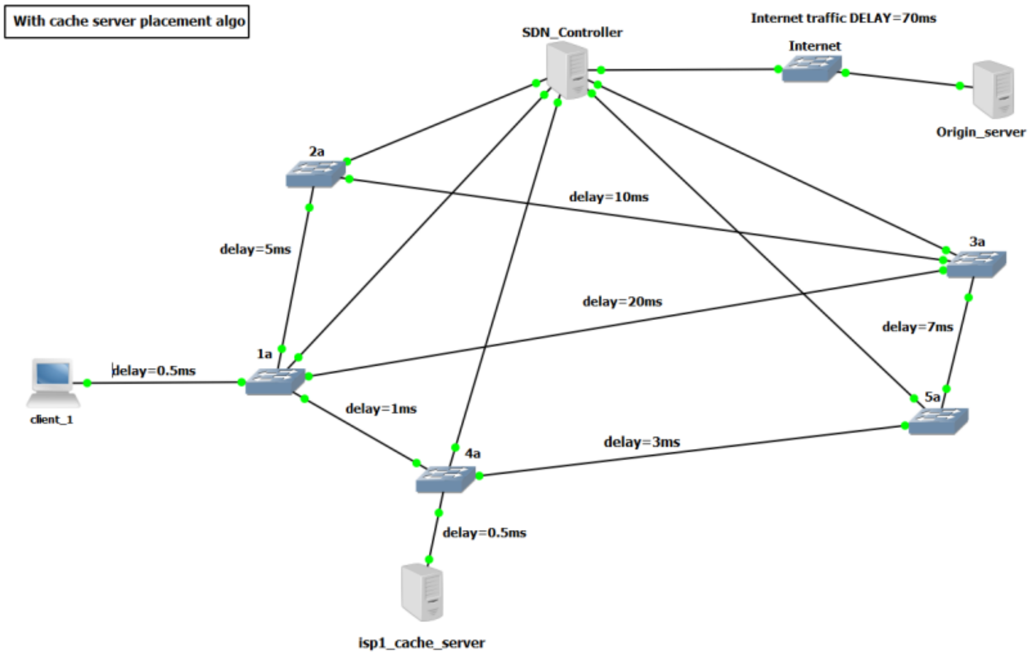


Figure 7. Simulated scenario

Table 3. Simulated scenario with cache server & client link delay

Row-based Floyd Wars hall Algorithm	1A (delay)	2a (delay)	3a (delay)	4a (delay)	5a (delay)	Total Latency
Openflow Switch (1A)	1ms	6ms	12ms	2ms	5ms	1A=
Openflow Switch (2a)	6ms	1ms	11ms	7ms	10ms	26ms
Openflow Switch (3a)	12ms	11ms	1ms	11ms	8ms	3a= 43ms
Openflow Switch (4a)	2ms	7ms	11ms	1ms	4ms	4a= 25ms
Openflow Switch (5a)	5ms	10ms	8ms	4ms	1ms	5a= 28ms
Openflow Switch (1A)	1ms	6ms	12ms	2ms	5ms	1A=
Openflow Switch (2a)	6ms	1ms	11ms	7ms	10ms	2a= 35ms
Openflow Switch (3a)	12ms	11ms	1ms	11ms	8ms	3a= 43ms
Openflow Switch (4a)	2ms	7ms	11ms	1ms	4ms	4a= 25ms

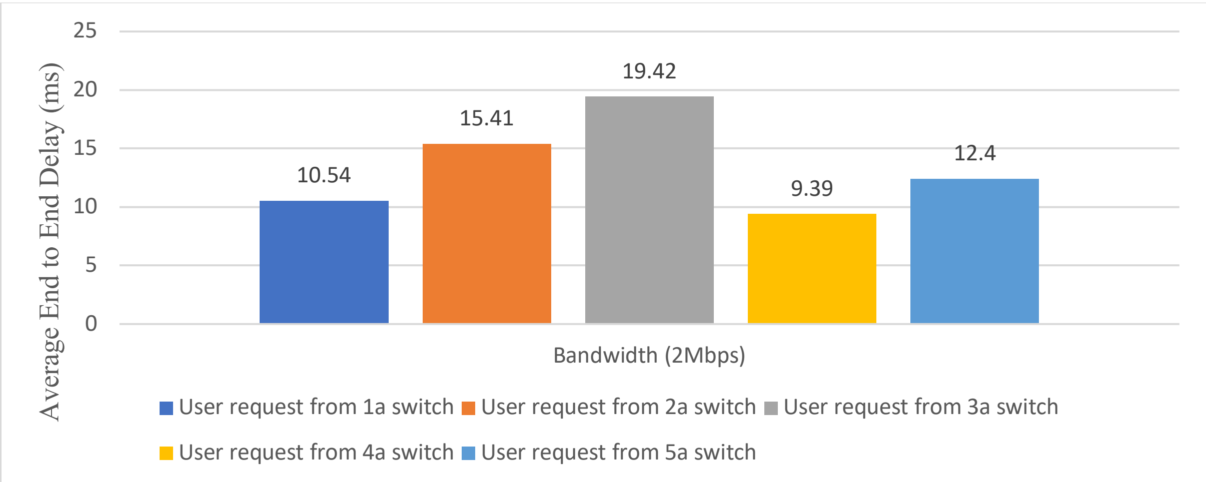


Figure 8. Cache server placed at 4A switch (100% cache hit content from internet)

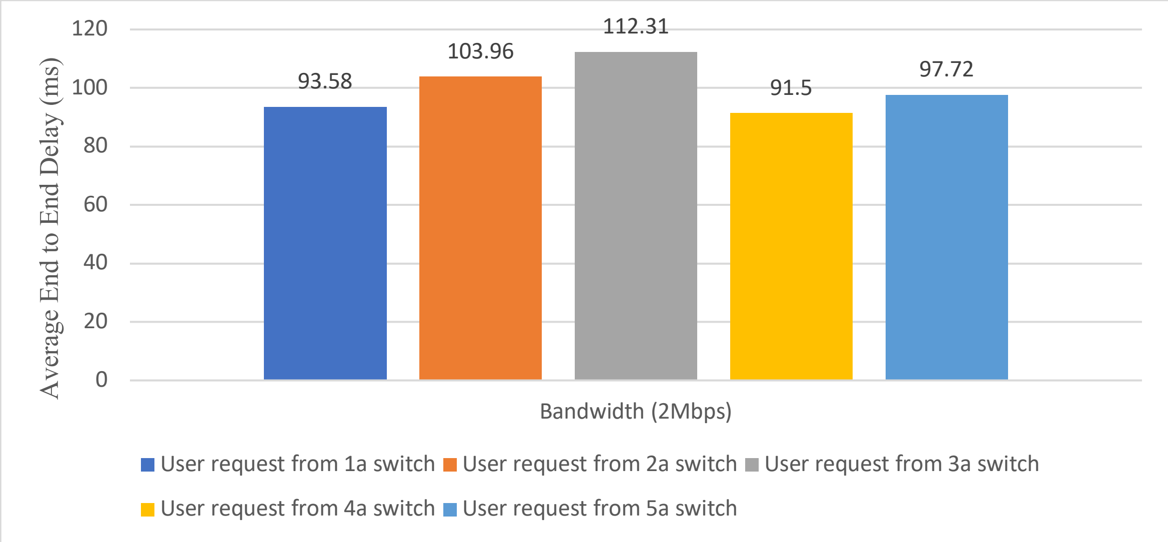


Figure 9. Cache server placed at 4A switch (0% cache hit content from internet)

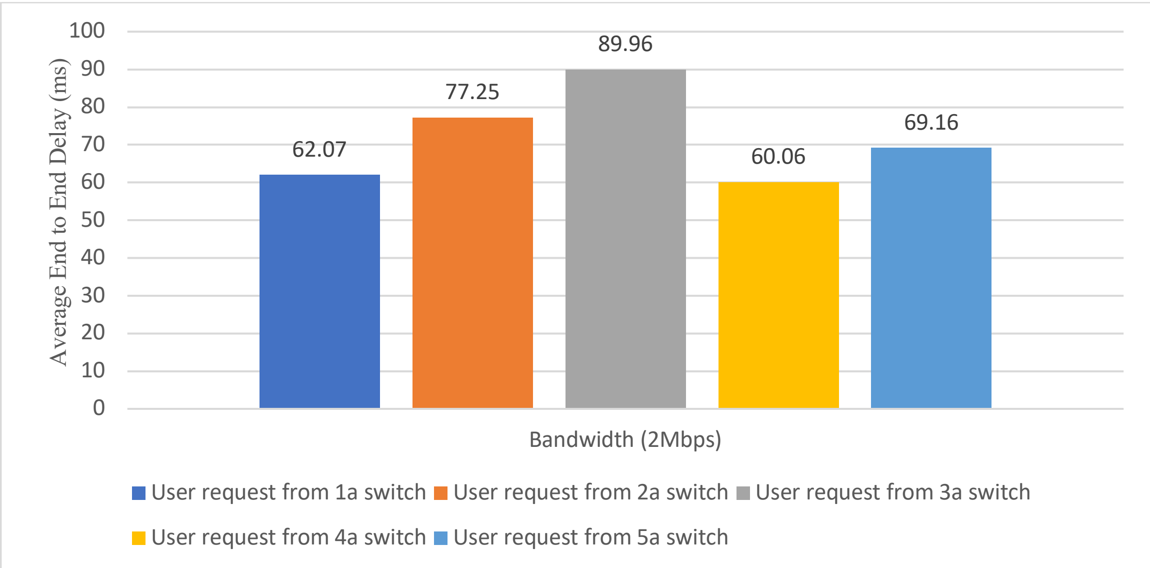


Figure 10. Cache server placed at 4A switch (cooperative caching)

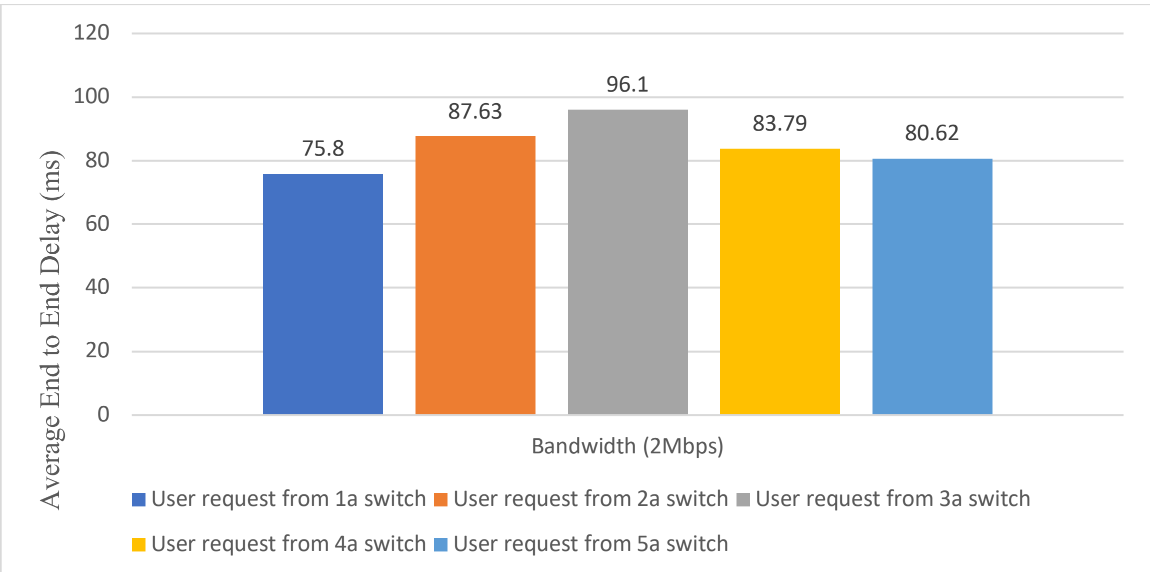


Figure 11. 10% Cache_Server1, 30% Cache Hit Cache_Server2, 60% From Internet

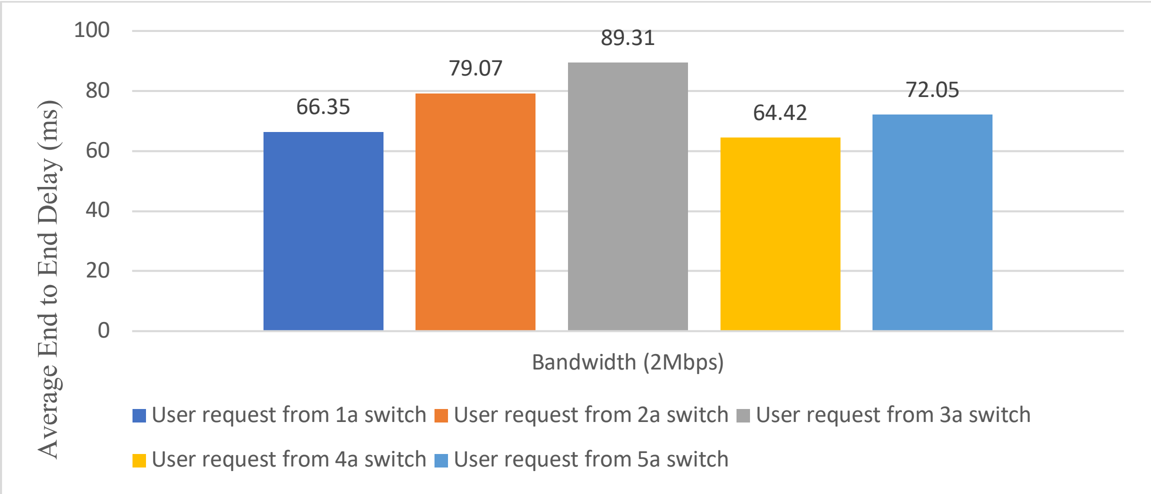


Figure 12. 10% Cache_Server1, 60% Cache Hit Cache_Server2, 30% from Internet

Table 4. Average end-to-end delay at 0% cache hit using 2Mbps

Cache server Location	Request from 1A	Request from 2A	Request from 3A	Request from 4A	Request from 5A
1A	91.54ms	101.95ms	114.43ms	93.64ms	99.88ms
2A	101.95ms	91.55ms	112.36ms	104.05ms	110.28ms
3A	114.43ms	112.36ms	91.55ms	112.36	106.12ms
4A	93.58ms	103.96ms	112.31ms	91.5ms	97.72ms
5A	99.88ms	110.28ms	106.12ms	97.8ms	91.55ms

Table 5. Average end-to-end delay at 50% cache hit using 2Mbps

Cache server Location	Request from 1A	Request from 2A	Request from 3A	Request from 4A	Request from 5A
1A	50.47ms	58.18ms	67.42ms	52.02ms	56.64ms
2A	58.18ms	50.47ms	65.89ms	59.73ms	64.35ms
3A	67.42ms	65.89ms	50.47ms	65.89ms	61.26ms
4A	52.06ms	59.68ms	65.86ms	50.44ms	55.06ms
5A	56.64ms	64.35ms	61.26ms	55.1ms	50.47ms

Table 6. Average end-to-end delay at 100% cache hit using 2Mbps

Cache server Location	Request from 1A	Request from 2A	Request from 3A	Request from 4A	Request from 5A
1A	9.39ms	14.41ms	20.42ms	10.4ms	13.4ms
2A	14.41ms	9.39ms	19.42ms	15.41ms	18.42ms
3A	20.42ms	19.42ms	9.39ms	19.42ms	16.41ms

4A	10.54ms	15.41ms	19.42ms	9.39ms	12.4ms

5. Conclusion

Big data holds an enormous quantity of information which devours additional bandwidth and if the user has little bandwidth, then big data will practice the latency to the client. A cache server and a new network framework named SDN resolve plenty of challenges of big data. A cache server can reduce the weight on real server and add to throughput by lessening the end-to-end delay (latency). Cooperative caching produces better results than standalone cache servers in ISP. Whilst SDN structure is greatly improved than the customary network as SDN has the orchestration which can simply put the flows in an Openflow switch and train them where to direct the traffic. In this background, projected three ISPs, and each ISP holds one cache server to decrease the weight of a solitary cache server. Opening of cache server is not able to reduce the latency but it is as well requiring planning the algorithm which optimally directs where to put the cache server in the topology. In the proposed model, a lightweight row-based Floyd Warshall algorithm is set up to resolve the placement of the cache server. The imitation consequences shaped diagrams which demonstrated that via the row-based Floyd Warshall algorithm, the cache server boosts the network performance by augmenting the throughput and reducing the latency (end-to-end delay) in the network.

6. Future Work

As a prospective effort, the examination of several heuristics can be studied to further decrease the end-to-end holdup (latency) as well as augment the throughput of the projected approach. Information-centric networking (ICN) will also be worn in assumed work. Furthermore, plans can also be devised to experiment with the results of a specified approach in a real physical environment.

References

1. C. Systems, “Cisco 2022 annual report,” 2022.
2. R. Amin, M. Reisslein, and N. Shah, “Hybrid sdn networks: A survey of existing approaches,” IEEE Communications Surveys & Tutorials, vol. 20, no. 4, pp. 3259–3306, 2018.
3. D. Kreutz, F. Ramos, P. Veríssimo, C. Esteve Rothenberg, S. Azodolmolky, and S. Uhlig, “Software-defined networking: A comprehensive survey,” ArXiv e-prints, vol. 103, 06 2014.
4. W. Xia, Y. Wen, C. H. Foh, D. Niyato, and H. Xie, “A survey on softwaredefined networking,” IEEE Commununication Surveys & Tutorials, vol. 17, no. 1, pp. 27–51, 2015.
5. B. Wang, Y. Zheng, W. Lou, and Y. T. Hou, “DDoS attack protection in the era of cloud computing and software-defined networking,” in 2014 IEEE 22nd International Conference on Network Protocols, pp. 624–629, IEEE, 2014.
6. F. Hu, Q. Hao, and K. Bao, “A survey on software-defined network and openflow: From concept to implementation,” IEEE Communications Surveys & Tutorials, vol. 16, no. 4, pp. 2181–2206, 2014.
7. V. Pacifici, F. Lehrieder, and G. Dán, “Cache capacity allocation for bittorrent-like systems to minimize inter-isp traffic,” in 2012 Proceedings IEEE INFOCOM, pp. 1512–1520, 2012.
8. Y. Huang, T. Z. J. Fu, D. M. Chiu, J. C. Lui, and C. Huang, “Challenges, design and analysis of a large-scale P2P-VoD system,” in Proceedings of the ACM SIGCOMM 2008 conference on Data communication, vol. 38, pp. 375– 388, 08 2008.
9. S. Borst, V. Gupta, and A. Walid, “Distributed caching algorithms for content distribution networks,” in Proceedings 29th IEEE International Conference on Computer Communications (INFOCOM 2010, San Diego CA, USA, March 15-19, 2010), (United States), pp. 1–9, Institute of Electrical and Electronics Engineers, 2010.
10. M. Qian, Y. Wang, Y. Zhou, L. Tian, and J. Shi, “A super base station based centralized network architecture for 5g mobile communication systems,” Digital Communications and Networks, vol. 54, 03 2015.
11. S. Sharif, M. H. Y. Moghaddam, and S. A. H. Seno, “Adaptive cache content placement for software-defined internet of things,” Future Generation Computer Systems, vol. 136, pp. 34–48, 2022.
12. W. Rafique, A. S. Hafid, and S. Cherkaoui, “Softcaching: A framework for caching node selection and routing in software-defined information centric internet of things,” Computer Networks, vol. 235, p. 109966, 2023.
13. Y. Cui, J. Song, M. Li, Q. Ren, Y. Zhang, and X. Cai, “Sdn-based big data caching in isp networks,” IEEE Transactions on Big Data, vol. 4, no. 3, pp. 356–367, 2018.
14. D. Kim and Y. Kim, “Enhancing ndn feasibility via dedicated routing and caching,” Computer Networks, vol. 126, 07 2017.
15. E. Aubry, T. Silverston, and I. Chrisment, “Srsc: Sdn-based routing scheme for ccn,” in Proceedings of the 2015 1st IEEE Conference on Network Softwarization (NetSoft), pp. 1–5, 2015.
16. J. Badshah, M. Kamran, N. Shah, and S. A. Abid, “An improved method to deploy cache servers in software defined network-based information centric networking for big data,” Journal of Grid Computing, vol. 17, 06 2019.
17. S. Hougardy, “The floyd-warshall algorithm on graphs with negative cycles,” Inf. Process. Lett., vol. 110, pp. 279–281, 04 2010.
18. G. Yao, J. Bi, Y. Li, and L. Guo, “On the capacitated controller placement problem in software defined networks,” Communications Letters, IEEE, vol. 18, pp. 1339–1342, 08 2014.
19. A. Tootoonchian, S. Gorbunov, Y. Ganjali, M. Casado, and R. Sherwood, “On controller performance in software-defined networks,” in Proceedings of the 2nd USENIX conference on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services, vol. 54, pp. 10–10, 04 2012.
20. G. Yao, J. Bi, and L. Guo, “On the cascading failures of multi-controllers in software defined networks,” in 2013 21st IEEE International Conference on Network Protocols (ICNP), pp. 1–2, 10 2013.
21. S. Sanadhya, R. Sivakumar, K.-H. Kim, P. Congdon, S. Lakshmanan, and J. P. Singh, “Asymmetric caching: Improved network deduplication for mobile devices,” in Proceedings of the 18th Annual International Conference on Mobile Computing and Networking, Mobicom ’12, (New York, NY, USA), p. 161–172, Association for Computing Machinery, 2012.
22. A. Chankhunthod, P. Danzig, C. Neerdaels, M. Schwartz, and K. Worrell, “A hierarchical internet object cache,” in Proceedings of the 1996 annual conference on USENIX Annual Technical Conference, pp. 153–164, 01 1996.
23. S.-H. Shen and A. Akella, “An information-aware qoe-centric mobile video cache,” in Proceedings of the 19th annual international conference on Mobile computing & networking, pp. 401–412, 09 2013.
24. Chankhunthod, P. Danzig, C. Neerdaels, M. Schwartz, and K. Worrell, “A hierarchical internet object cache,” in Proceedings of the 1996 annual conference on USENIX Annual Technical Conference, pp. 153–164, 01 1996.
25. W. Li, E. Chan, G. Feng, D. Chen, and S. Lu, “Analysis and performance study for coordinated hierarchical cache placement strategies,” Computer Communications, vol. 33, no. 15, pp. 1834–1842, 2010.
26. Y. Kim and I. Yeom, “Performance analysis of in-network caching for content-centric networking,” Computer Networks, vol. 57, no. 13, pp. 2465– 2482, 2013
27. M. Mangili, F. Martignon, and A. Capone, “A comparative study of content-centric and content-distribution networks: Performance and bounds,” in 2013 IEEE Global Communications Conference (GLOBECOM), pp. 1403–1409, 2013.
28. S. Podlipnig and L. Böszörmenyi, “A survey of web cache replacement strategies,” ACM Comput. Surv., vol. 35, pp. 374–398, 12 2003.

29. D. Applegate, A. Archer, V. Gopalakrishnan, S. Lee, and K. K. Ramakrishnan, "Optimal content placement for a large-scale vod system," IEEE/ACM Transactions on Networking, vol. 24, no. 4, pp. 2114–2127, 2016.
30. L. Chaves, I. Calciolari Garcia, and E. Madeira, "Ofswitch13: Enhancing ns-3 with openflow 1.3 support," in Proceedings of the 2016 Workshop on ns-3, pp. 33–40, 06 2016.
31. G. Carneiro, P. Fortuna, and M. Ricardo, "Flowmonitor - a network monitoring framework for the network simulator 3 (ns-3)," Proceedings of the Fourth International ICST Conference on Performance Evaluation Methodologies and Tools, 01 2009