

A Study on Different Parameters Affecting Overall Cost of Global Content Distribution Services in Metropolitan Cloud Network

Semanto Mondal(✉) and Rajib Chandra Ghosh

University of Naples Federico II, Naples, Italy
{s.mondal,r.chandraghosh}@studenti.unina.it

Abstract. Due to the advancement of technology and the internet, every digital component is connected which results in a tremendous increase in digital content. To make digital content respectively multimedia content available to potentially large and geographically distributed consumer populations, Global Content Distribution Networks are used. In the era of cloud, several cloud providers provide cloud services to their users by optimizing cloud-based content delivery. There are different factors associated with a cost-effective model such as content delivery cost, computation cost, and storage cost associated with the cloud network [1]. In this study, we have tried to analyze different parameters that can affect the overall cost and performance of different Global Content Distribution services such as the capacity of resource blocks and, the addition of edge computing by applying different network constraints such as flow conservation constraints, flow chaining constraints, communication and computation capacity constraints, maximum delay constraints. We have also tried to experiment with the effect of different network infrastructures on the overall cost of four different Content Distribution services for different service graphs.

Keywords: Global Content Distribution Network (GCDN) · Edge Computing · Service Graph · Network Optimization

1 Introduction

In less than fifteen years, Content Delivery Networks (CDNs) have become key actors in the value and supply chain of the Internet. The main objective of these cloud networks is to maintain Quality of Service (QoS) and quality of Experience (QoE) while maintaining lower cost as well as minimal delay which can be utilized in the field of cloud gaming, content delivery, as well as on-demand video [2, 3].

Cloud networks can be designed with centralized as well as distributed computation resources. With the help of wired as well as wireless network connectivity different digital devices such as servers are connected. In the centralized network all the computation is performed centrally after that the content is delivered to the end users who might reside in a geographic location far away from the original servers. Here several problems arise For example when large content is transferred to a longer distance itneeds a larger capacity

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
O. Gervasi et al. (Eds.): ICCSA 2024, LNCS 14813, pp. 46–64, 2024.
https://doi.org/10.1007/978-3-031-64605-8_4

of the communication channel as well as overall delay also increases significantly. To overcome this problem distributed computation resources such as Edge Computing come into the picture. Caching can also be used to reduce the delivery cost. In services like Professional Video on Demand (e.g.; Netflix), and User Generated Video on Demand (e.g.: Facebook) caching near the user end can be utilized to reduce both usage of resources as well as delay in the network [4]. Other techniques such as computation offloading which refers to the process of transferring the computational task to the computation node from the parent node to make the computation process faster as well as to minimize the resource allocation in the parent node [5, 6, 7].

In this paper, we have tried to minimize the overall cost by using different network infrastructures, by changing the network capacity which resulted also into a change in resource allocation block as well as by changing caching and computation rates. Figure 1 shows the overall workflow of the project for different services:

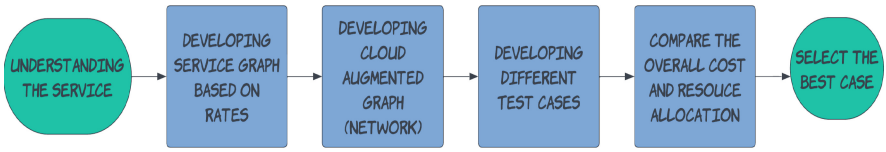


Fig. 1. Overall Workflow

2 Formulation of Cloud Network Flow Optimization

In order to formulate the optimization problem, we leverage the recently proposed cloud network flow optimization framework [8, 9, 10, 11]. The goal is to optimize the configuration of a service over a cloud network. The service is described by a service graph $R = (I, \mathcal{K})$ where I is the set of service functions and $\mathcal{K}$ the set of commodities. In addition $g : \mathcal{K} \rightarrow \mathcal{O}$ refers to a mapping function that maps each commodity to a specific information object. The cloud network is also described by a directed graph $G = (V,E)$ with V vertices and E edges representing the set of nodes and links in the cloud network. The target is to minimize cost and latency.

$$min\ (\sum_{(u,v)\epsilon\varepsilon^a} y_{uv} w_{uv} + \sum_{k\epsilon K^d} l_T^k) \tag{1}$$

Equation (1) is the objection function where $(u, v)\epsilon\mathcal{E}^a$ defines the set over which the minimization is performed [10]. Here, $\mathcal{E}^a$ represents the set of edges in the network. y_{uv} Represents the number of allocated resource blocks from node u to v and w_{uv} represents the cost associated with each resource block for transmitting content from node u to v. Here l_T^k is the total delay for the destination commodities and $l_T^k \leq L^k$ where L^k is the maximum delay constraint per destination commodity. The function aims to minimize the total cost as well as the delay of content distribution over the network considering the following constraints.

$$\sum_{v\epsilon\delta^-(u)} f_{vu}^k = \sum_{v\epsilon\delta^+(u)} f_{uv}^k \ \forall u\epsilon\mathcal{V},\ \ k\epsilon\mathcal{K} \tag{2}$$

Equation (2) is the flow conservation constraint which states that, for a specific content type K, in a node u the total incoming flow is equal to the total outgoing flow in other words what goes in is that comes out [8, 9].

$$f_{pu}^{k} = f_{up}^{l} \forall u \in V,\ k \in K, l \in X(k) \tag{3}$$

Equation (3) is related to the chaining constraint [10] which states that for every vertex u, and every pair of content types k and l associated with u, the flow of content from vertex p to vertex u for content type k is equal to the flow of content from vertex u to vertex p for content type l. $\forall u \in V, k \in K, l \in X(k)$; this part of the constraint states that the equality holds for all vertices u in the set of vertices V, for all content types k in the set of content types K, and all content types l in the set X(k).

$$f_{su}^{k} = \begin{cases} 1, \textit{if } k \textit{ is sourced at node } s (i.e. \forall k \in K^{s}) \\ 0, \textit{otherwise} \end{cases} \tag{4}$$

Equation (4) is the source constraint which states that the content type k originating from source vertex s must follow a specified path to reach vertex u to avoid any sort of bypass of content.

$$f_{ud}^{k} = \begin{cases} 1, \textit{if } k \textit{ is requested at node } d (i.e. \forall k \in K^{d}) \\ 0, \textit{otherwise} \end{cases} \tag{5}$$

Equation (5) is related to destination constraints which ensures that for every content type k associated with the destination vertex d, there is exactly one unit of flow from the vertex u to the destination vertex d.

$$f_{uv}^{k} R_k \leq f_{uv}^{o}\ \forall (u, v) \in \mathcal{E}^{a}, k \in K, o = g(k) \tag{6}$$

$$\sum_{o \epsilon \mathcal{O}} f_{uv}^{o} = f_{uv} \tag{7}$$

Equation (6) is related to the actual flow constraints [9] and ensures that the flow of commodity k through the edge (u,v), does not exceed the flow of the corresponding object o through the same edge considering k is mapped to object o using mapping function g(k) and Eq. (7) ensures that the summation of object flow over edge (u,v) is equal to the total flow.

$$f_{uv} \leq y_{uv} c_{res}; y_{uv} \leq c_{uv}\ \forall (u, v) \epsilon \mathcal{E}^{a} \tag{8}$$

Equation (8) is the communication capacity constraint [9] which states that the total flow on a link (u,v) or an edge is less or equal to the total allocated rate (where the rate is equal to the number of resource blocks multiplied by capacity per block) and number of allocated resource block should be less or equal to the maximum possible resource block for the link or edge (u,v).

$$l^{k} = \sum_{(u,v) \in \mathcal{E}^{a}} l_{uv}^{k} f_{uv}^{k}\ \forall k \in K \tag{9}$$

$$l_T^k = l^k \ \forall k \in K^s \tag{10}$$

$$l_T^k \geq l^k + l_T^l \ \forall k \in K \backslash K^s, l \in X(k) \tag{11}$$

$$l_T^k \leq L^k \ \forall k \in K^d \tag{12}$$

Equations (9), (10), (11) and (12) are related to the delay constraints which set the lower and upper bound to the allowed total delay for delivering any content type or stream of k [9].

$$f_{uv}^k \epsilon \{0, 1\}, f_{uv} \epsilon \mathbb{R}^+, f_{uv}^o \epsilon \mathbb{R}^+, y_{uv} \epsilon \mathbb{Z}^+ \ \forall (u, v) \in \mathcal{E}^a, k \in K, o \in \mathcal{O} \tag{13}$$

Equation (13) is related to the domain of the variables. $f_{uv}^k \epsilon \{0,1\}$ This is related to discrete flow which does not allow splitting of the content flow and maintains a specific path from source to destination. f_{uv} And f_{uv}^o are the flow variables can take any positive real numbers whereas y_{uv} is the resource allocation variable that belongs to any positive integer [9, 10].

We have used the LpProblem class from a Python library named PuLP using liner programming where all the above-mentioned constraints are used to develop the final optimization model which optimizes the cost of different services in different networks. All the constraints are utilized to optimize the overall cost of the model and to understand how different parameters such as resource block capacity, the total number of resource blocks, location of processing, the addition of computation node at the access layer, and transmission rate affect the overall performance of the cloud network.

3 Cloud Augmented Graph and Service Graph Implementation

Three different networking scenarios have been implemented to test different cases and to provide a general conclusion. All these scenarios are for a single metropolitan area with a single core node and multiple edge and access nodes. The communication cost is set at 100 whereas the computation in and computation out costs for the computation nodes are set to 1 and 10 respectively. The overall cost of different networks is compared for varying rates as well as the capacity of the resource block.

In the first network of Fig. 2, there are single communication node in the core and 3 communication nodes in both the edge and access layer. Each communication node in the edge layer is having 3 computation nodes. Source nodes are located both in the core and access layer whereas destination nodes are located only at the access layer. There are no computation nodes in the access layer.

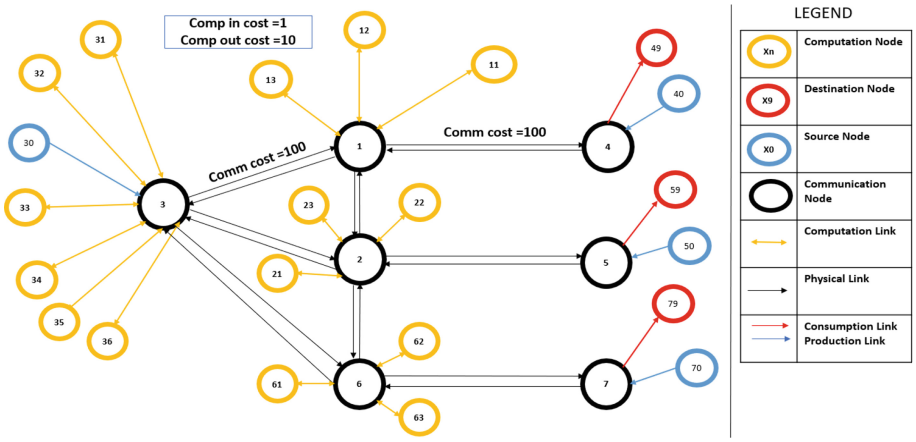


Fig. 2. Cloud Augmented Graph-1

The second network that is Cloud Augmented Graph-2 in Fig. 3 is the extended version of the network one that is also for a single metropolitan area having a single communication node in the core layer and 4 communication nodes in the edge layer. Each communication node of the edge layer has 4 communication nodes residing in the access layer. Here also there are no computation nodes in the access layer and all of these computation nodes belong to the Core and Edge layer.

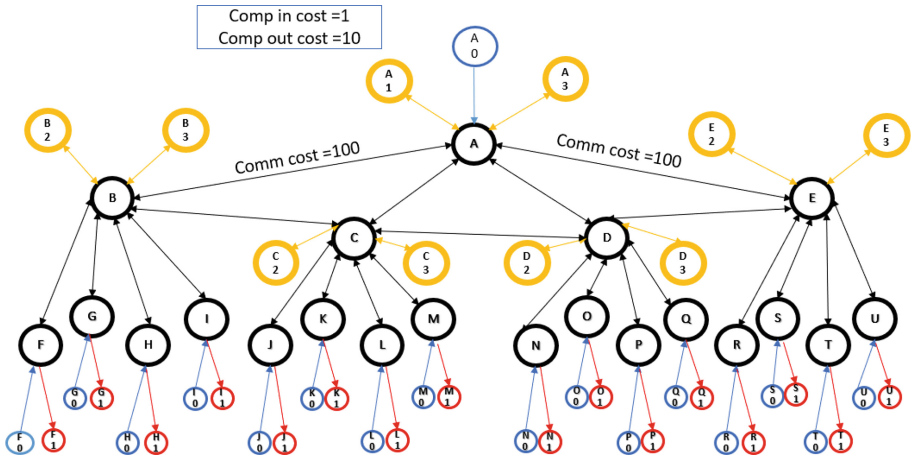


Fig. 3. Cloud Augmented Graph-2

Cloud network-2 is further extended to cloud network-2A in Fig. 4 where two main modifications have been implemented. The first one is, that now in all layers communication nodes have only one computation node which will be responsible for both caching

and processing and the second one is, that computation nodes are also integrated with the communication nodes of the access layer.

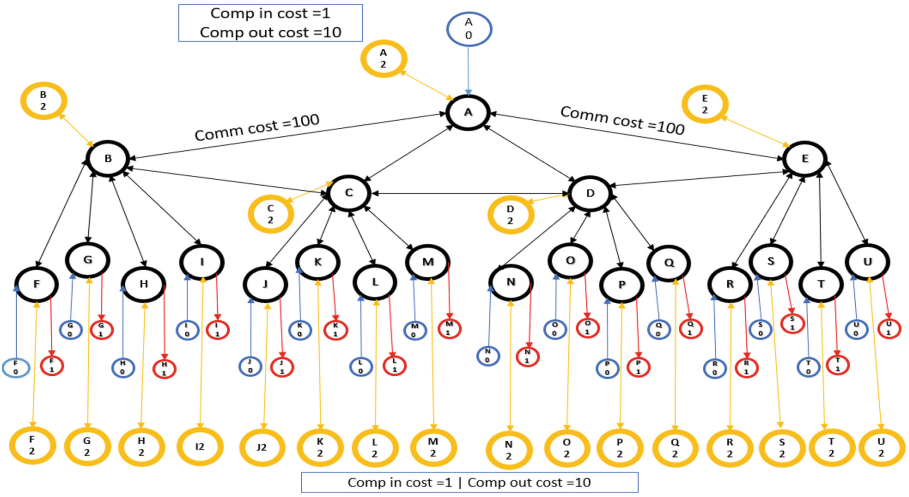


Fig. 4. Cloud Augmented Graph-2A



Fig. 5. Service Graph for Network-1

Figure 5 is the service graph for network-1 where there are 9 commodities and 7 objects. Here a single source and 3 destinations are considered. R_1, R_2, R_3 are the rates associated with caching and processing.

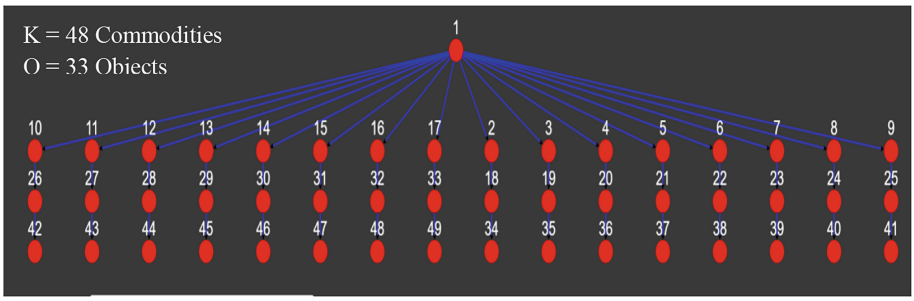


Fig. 6. Service Graph for Network-2 & 2A

Figure 6 represents the service graph for both network-2 and 2A having 48 commodities and 33 objects also has a single source along with 16 destinations.

4 Professional Video on Demand (e.g.: Netflix)

In professional video-on-demand cases, users can watch the content whenever they want. Netflix and YouTube are good examples of these services. Video that the user accesses frequently can be cached near to the user so that next time when any other users are accessing the same content the delivery cost as well as time should be minimized.

Here in this service, the general consideration is that the rate before caching is smaller than the rate after caching. Source functions are located at the core node and the destination functions are at the access nodes. So, we have decided the rate as follows: $R_1 < R_2$, $R_1 = 15$, $R_2 = R_3 = 50$. For all the services the cost associated with the communication link is set to 100 per resource block, 1 for computation in and 10 for computation out. The maximum allowable delay is set as $L^k = 100$.

For cloud network-1 we have considered two different cases as shown in Table 1 in terms of communication link capacity and total amount of resource blocks:

Table 1. Cases for Cloud Network-1 for all services

Case-1	Case-2
Capacity per resource block = 5 Mbps	Capacity per resource block = 50 Mbps
Number of resource blocks = 200	Number of resource blocks = 20
Total capacity of communication link = 1 Gbps	Total capacity of communication link = 1 Gbps

For cloud network-2 we have considered two different cases as shown in Table 2 in terms of communication link capacity and total amount of resource blocks:

Table 2. Cases for Cloud Network-2 & 2A for all services

Case-1	Case-2	Case-3
Capacity per resource block = 5 Mbps	Capacity per resource block = 50 Mbps	Capacity per resource block = 200 Mbps
Number of resource blocks = 200	Number of resource blocks = 20	Number of resource blocks = 5
Total capacity of communication link = 1 Gbps	Total capacity of communication link = 1 Gbps	Total capacity of communication link = 1 Gbps

Table 3 shows the comparison concerning different cases for network 1. Whereas Figs. 7 and 8 show the content flow and allocation of resource blocks as well as the total amount of cost for different links. Due to the increase in resource block capacity, caching got overlapped which resulted in a reduction of overall content delivery cost from 4026 to 727.

Table 3. Performance Comparison on Coud Network-1

Case-1 (5 Mbps)	Case-2 (50 Mbps)
Total Cost = 4026	Total Cost = 727
Total number of resource blocks used 51	Total number of resource blocks used 26
Caching is redundant and at Edge	Caching is overlapped at the core
Total number of links used 22	Total number of links used 18

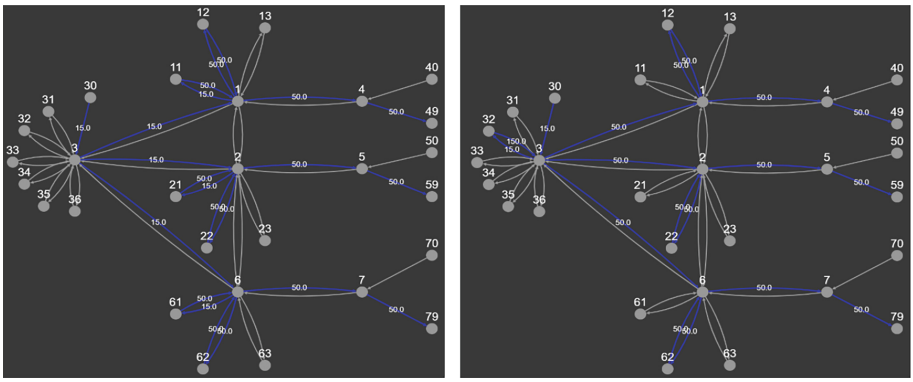


Fig. 7. Content flow graph for cases 1 and 2

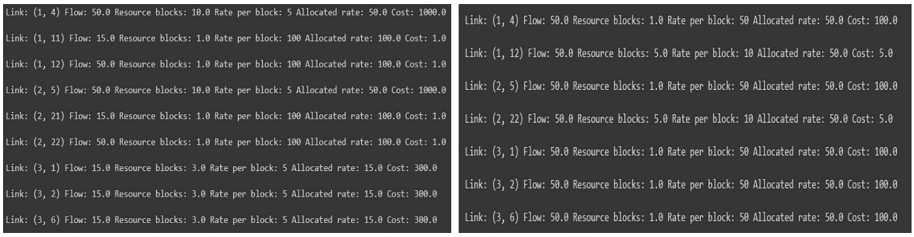


Fig. 8. Allocated resource block and cost per link for cases 1 and 2

Table 4. Rate (R1) vs Cost for communication resource block capacity = 5 Mbps in Network-1

Rate (R1)	Cost	Caching
15	4026	Edge
20	4326	Edge
25	4626	Edge
30	4926	Edge
35	5226	Edge
40	5526	Edge
45	5826	Edge
50	6114	Core

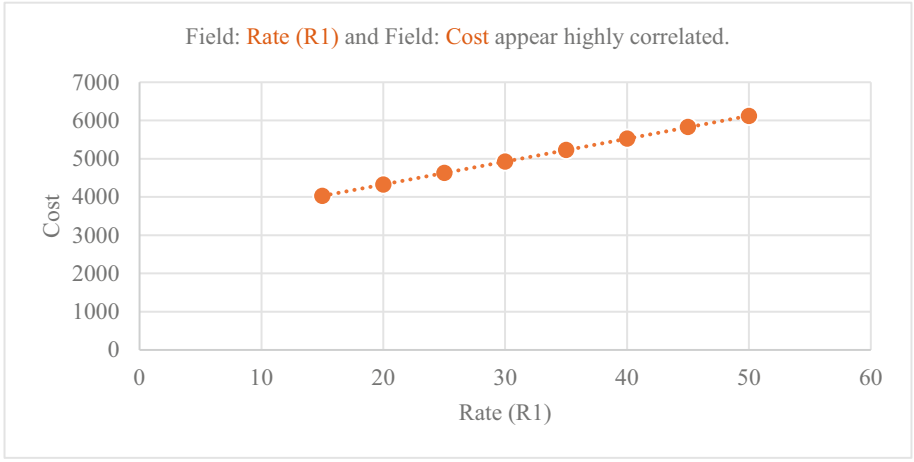


Fig. 9. Relationship between Rate (R1) and Cost in Network-1

Based on Table 4 as well as Fig. 9 it is clear that, if we keep the capacity of the communication resource block constant and change the rate (communication, consumption

and production), with the increase of the rate (R) the overall cost of the network also increases linearly.

In general, when we increase the communication link capacity while keeping the cost the same as smaller resource blocks, caching is done with overlapping near the source that is at the core because it reduces the total number of resource block usage. On the contrary, if we fix the channel capacity to a particular value and chance R1 which is the rate before caching, we can observe that, at some point when R1 = R2 where R2 is the rate after caching, then automatically caching gets overlapped and happens at the core instead of happening at the edge redundantly. The cost remains the same from core to the edge for both catching at the core and edge, but usage of computation resource block reduces when caching is done at the core with overlapping. As a result, to optimize the cost caching moved from edge to core without increasing the channel capacity.

Table 5. Performance Comparison on Coud Network-2

Case-1 (5 Mbps)	Case-2 (50 Mbps)	Case-2 (200 Mbps)
Total Cost = 17692	Total Cost = 2492	Total Cost = 2489
Total number of resource blocks used 200	Total number of resource blocks used 48	Total number of resource blocks used 45
Caching is redundant and at Edge, and Processing is at Edge	Caching is redundant and at Edge, and Processing is at Edge	Caching is overlapped at the core; Processing is at the Edge
Total number of links used 53	Total number of links used 53	Total number of links used 47

Table 6. Performance Comparison on Coud Network-2A

Case-1 (5 Mbps)	Case-2 (50 Mbps)	Case-2 (200 Mbps)
Total Cost = 6496	Total Cost = 2492	Total Cost = 2489
Total number of resource blocks used 92	Total number of resource blocks used 48	Total number of resource blocks used 45
Caching is redundant and at Access, Processing is at Access	Caching is overlapped and at Edge, Processing is at Edge	Caching is overlapped at the core; Processing is at the Edge as well as the core
Total number of links used 69	Total number of links used 45	Total number of links used 47

Tables 5 and 6 show the comparison of different cases for networks 2 and 2A. For both networks 2 and 2A, we can visualize from the table that for higher capacity total number of resource allocation blocks is reduced due to overlapping of caching at a single node instead of redundant overlapping at different nodes. It is also clear that an increase in communication resource block capacity from 5 Mbps to 50 Mbps results in a significant reduction in cost whereas 200 Mbps couldn't reduce the cost predominantly.

5 Professional Live Video (e.g.: IPTV)

In professional Live Video, the consideration is that there will be one broadcasting channel that broadcasts the content like a live football match and there will be multiple users watching the content from a single metropolitan area. Here the rate before caching function is the same as the rate after caching function and the rates are considered as follows: $R_1 = R_2$, $R_1 = R_2 = 15$, $R_3 = 55$. Source functions are located at the core and destination functions are located at the access layer. The rate after processing is higher than before processing why because here we are considering that there might be some personalized content such as advertisements could be added to the original content as a result the rate should be higher.

Table 7. Performance Comparison on Coud Network-1

Case-1 (5 Mbps)	Case-2 (50 Mbps)
Total Cost = 4008	Total Cost = 1004
Total number of resource blocks used 42	Total number of resource blocks used 17
Caching is overlapped and at Core	Caching is overlapped and at Core
Total number of links used 18	Total number of links used 18

Table 8. Performance Comparison on Coud Network-2

Case-1 (5 Mbps)	Case-2 (50 Mbps)	Case-2 (200 Mbps)
Total Cost = 19288	Total Cost = 4088	Total Cost = 2475
Total number of resource blocks used 212	Total number of resource blocks used 60	Total number of resource blocks used 40
Caching and processing are at the edge, caching is redundant	Caching and processing are at the edge, caching is redundant	Caching is overlapped at the core; Processing is at the Edge
Total number of links used 53	Total number of links used 53	Total number of links used 47

From Tables 7, 8, 9 and Fig. 10 it is visible that if the rate is higher than the individual resource capacity then the usage of resource blocks increases, and overlapping can't reduce the cost. Whereas if the capacity of the communication resource block increases it allows overlapping resulting in a reduction of usage of both links and communication and computation resources as a result we can observe a reduction in overall cost.

Likewise, in professional live video case based on Table 10 as well as Fig. 11 in case of professional live video, if we keep the capacity of the communication resource block constant and change the rate (communication, consumption, and production), with the increase of the rate (R) the overall cost of the network also increases linearly.

For professional live video, we have considered that the commodities before and after caching belong to different objects. We have also tried to check the effect of making

Table 9. Performance Comparison on Coud Network-2A

Case-1 (5 Mbps)	Case-2 (50 Mbps)	Case-2 (200 Mbps)
Total Cost = 6496	Total Cost = 2496	Total Cost = 2385
Total number of resource blocks used 92	Total number of resource blocks used 52	Total number of resource blocks used 41
Caching is redundant and at Access, Processing is at Access	Caching is redundant and at Access, Processing is at Access	Caching is overlapped at the core; Processing is at the Edge
Total number of links used 69	Total number of links used 69	Total number of links used 63

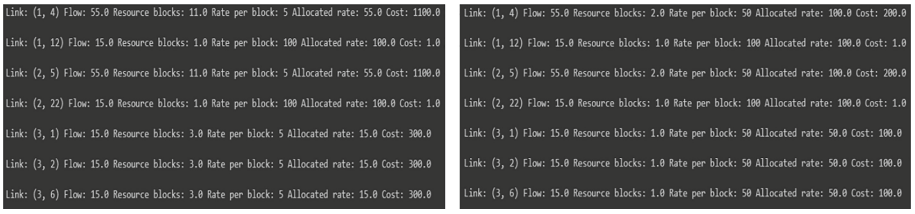


Fig. 10. Allocated resource block and cost per link for cases 1 and 2 in network 1

Table 10. Rate (R3) vs Cost for communication resource block capacity = 5 Mbps in Network-1

Rate (R3)	Cost	Caching
15	2052	Core
20	2382	Core
25	2712	Core
30	3042	Core
35	3372	Core
40	3702	Core
45	4032	Core
50	4362	Core
55	4692	Core

commodities before and after caching as the same object in network 2. In that case, the total number of commodities is 48 and the total number of objects is reduced to 18. The commodities after processing still belong to different commodities because after caching different personalization can be done on the content such as adding personalized advertisement or promotional content. Even though in network 1 caching is happening at the core in case 1, in case 2 caching is happening at the edge when the total number of objects is 33. In Network-1 each access node has a single destination whereas in



Fig. 11. Relationship Between Rate (R3) and Cost in Network-1

network-2 there are 4 destinations in each access node. From Fig. 12 it is clear that the communication rate of content from core to edge will be 60 if caching is happening at the core whereas it becomes 15 if caching is happening at the edge.

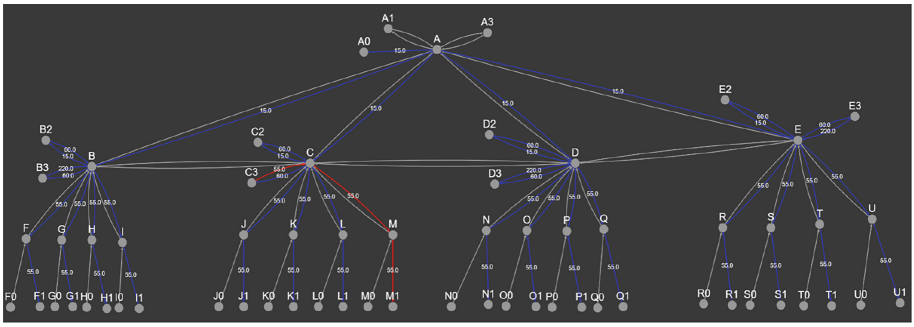


Fig. 12. Content flow graph in network 2

Referring to Fig. 13 when commodities after caching belong to the same object as well as commodities before caching belong to the same object too then caching gets overlapped and moved to the core. The communication rate of content flow from the core to the edge is 15. To reduce overall resource allocation caching overlaps at the core on the contrary processing is at the edge and redundant because these belong to different objects. The overall cost gets reduced from 19288 to 19255 for case 1.

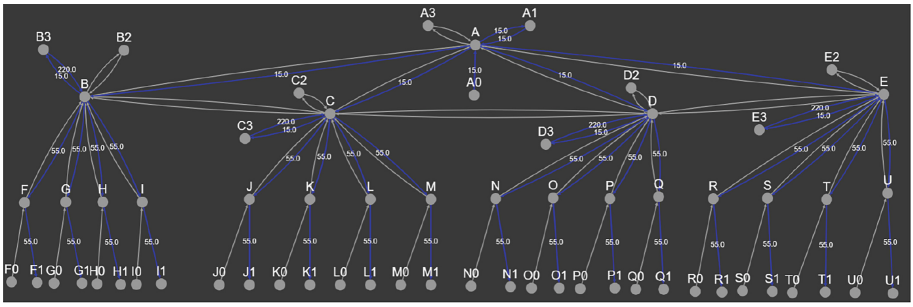


Fig. 13. Content flow graph in network 2 when commodities before and after caching belong to the same object

6 User Generated Video on Demand (e.g.: Facebook)

In general, users generate content on different social media platforms, and other users using the same platform can view this content. Facebook, Twitter, Instagram, and TikTok are a few examples of these services where users can watch the content generated by the other user. The content can be cached to the CDN servers near the user and the user can watch this content whenever they want as they result these videos are known as video on demand.

In this case, the rate before caching is smaller than the rate after caching. The source and destination are both located at the node associated with the access layer. The rates are set as follows: $R_1 < R_2$, $R_1 = 15$, $R_2 = R_3 = 50$.

Table 11. Performance Comparison on Coud Network-1

Case-1 (5 Mbps)	Case-2 (50 Mbps)
Total Cost = 4056	Total Cost = 734
Total number of resource blocks used 51	Total number of resource blocks used 14
Caching is redundant and at the edge	Caching is overlapped and at the edge
Total number of links used 22	Total number of links used 18

Table 12. Performance Comparison on Coud Network-2

Case-1 (5 Mbps)	Case-2 (50 Mbps)	Case-2 (200 Mbps)
Total Cost = 17932	Total Cost = 2732	Total Cost = 2647
Total number of resource blocks used 197	Total number of resource blocks used 47	Total number of resource blocks used 45
Caching and processing are at the edge, caching is redundant	Caching and processing are at the edge, caching is redundant	Caching is overlapped at the edge for the last 8 destinations, Processing is at the Edge
Total number of links used 52	Total number of links used 52	Total number of links used 47

Table 13. Performance Comparison on Coud Network-2A

Case-1 (5 Mbps)	Case-2 (50 Mbps)	Case-2 (200 Mbps)
Total Cost = 6416	Total Cost = 2612	Total Cost = 2612
Total number of resource blocks used 89	Total number of resource blocks used 45	Total number of resource blocks used 45
Caching is redundant and at Access, Processing is at Access	Caching is overlapped, and Processing is at both access and core	Caching is overlapped, and processing is at both access and core
Total number of links used 69	Total number of links used 50	Total number of links used 50

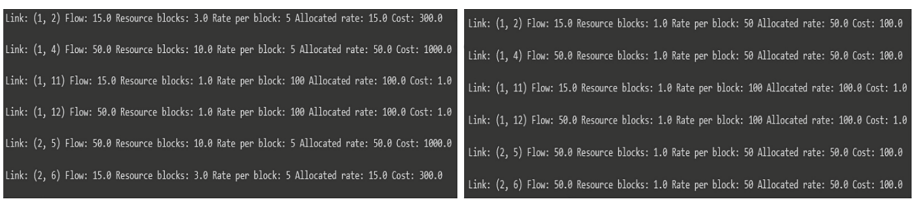


Fig. 14. Allocated resource block and cost per link for cases 1 and 2 in network 1

From Tables 11, 12, 13 and Fig. 14, it can be stated that when we increase communication resource capacity from 5 Mbps to 50 Mbps we can see a significant drop in cost due to overlapping and reduced amount of resource block allocation. Whereas if we increase the communication link capacity further to 200 Mbps we can't see much reduction in cost. So, for this case, 50 Mbps capacity is more suitable meaning case-2 is more appropriate than case-3 and case-1.

7 User Generated Live Video (e.g.: Facebook Live)

This service follows similar methodologies to professional live video. In services like Facebook Live, the user can go live and start sharing video content whereas other users can watch this content live. We have designed this service for a single metropolitan area where there will be a single source who can produce live content and multiple users can watch this content. So, this is a single source-multiple-destination scenario.

Here the rate before caching function is the same as the rate after caching function and the rates are considered as follows: $R_1 = R_2$, $R_1 = R_2 = 15$, $R_3 = 55$. Both source functions as well as destination functions are located at the access layer. Like professional live video service, the rate after processing is higher than before processing why because here we are considering that there might be some personalized content such as advertisements could be added to the original content as a result the rate should be higher.

Table 14. Performance Comparison on Coud Network-1

Case-1 (5 Mbps)	Case-2 (50 Mbps)
Total Cost = 4345	Total Cost = 1034
Total number of resource blocks used 52	Total number of resource blocks used 17
Caching is redundant and at the edge	Caching is overlapped and at the edge
Total number of links used 20	Total number of links used 18

Table 15. Performance Comparison on Coud Network-2

Case-1 (5 Mbps)	Case-2 (50 Mbps)	Case-2 (200 Mbps)
Total Cost = 19528	Total Cost = 4328	Total Cost = 2715
Total number of resource blocks used 212	Total number of resource blocks used 60	Total number of resource blocks used 39
Caching and processing are at the edge, caching is redundant	Caching and processing are at the edge, caching is redundant	Caching is overlapped at the edge; Processing is at the Edge
Total number of links used 52	Total number of links used 52	Total number of links used 46

From Table 14 and Fig. 15 for network-1 it can be stated that when we increase communication resource capacity from 5 Mbps to 50 Mbps we can see a significant drop in cost due to overlapping and reduced amount of resource block allocation. Whereas from Tables 15 and 16 it's clear that increasing the communication link capacity further to 200 Mbps from 50 Mbps reduces the cost significantly for networks 2 and 2A. So, for this case 50Mbps capacity is more suitable meaning case-2 is more appropriate than case-1 for network 1 on the contrary case-3 is more suitable for both network 2 and network 2A.

Table 16. Performance Comparison on Coud Network-2A

Case-1 (5 Mbps)	Case-2 (50 Mbps)	Case-2 (200 Mbps)
Total Cost = 6416	Total Cost = 2616	Total Cost = 2564
Total number of resource blocks used 89	Total number of resource blocks used 51	Total number of resource blocks used 35
Caching is redundant and at Access, Processing is at Access	Caching is redundant and at Access, Processing is at access	Caching is overlapped at edge and access; Processing is at both access and edge
Total number of links used 68	Total number of links used 68	Total number of links used 45

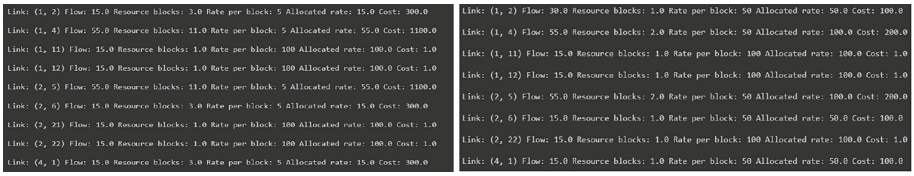


Fig. 15. Allocated resource block and cost per link for cases 1 and 2 in network 1

8 Overall Comparison for Network 2 & 2A

Cloud Network 2A is the modified version of Cloud Network 2. The main difference is that, in a cloud network 2A computation node is also added to the access layer along with the core and edge. A comparison has been made concerning the communication resource block, computation resource block, total cost, and maximum delay for destination commodities for four different services with respect could augmented graphs 2 and 2A. From these comparisons we can answer the following research questions:

RQ1: Weather increase in capacity reduces total delivery cost considering different rate requirements.

RQ2: Whether adding a computation node at the access layer reduces the overall cost of the network.

From the dual axis plots in Figs. 16 and 17, we can answer both the research questions that we proposed earlier. We have considered the data of professional live video service to generate the below plots. According to the plot in Fig. 16, as communication block capacity increases, both the cost and resource allocation reduce. Alongside, Fig. 17 demonstrates that in cloud network 2A the cost is less compared to that of cloud network 2 for different communication block capacities.

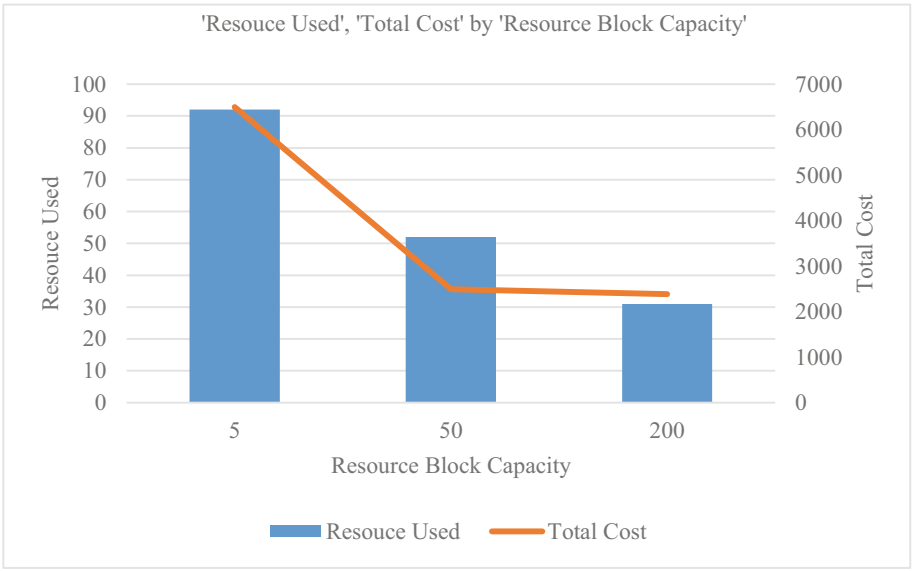


Fig. 16. Total Resource Used and Total Cost vs Resource Block Capacity in Professional Live Video

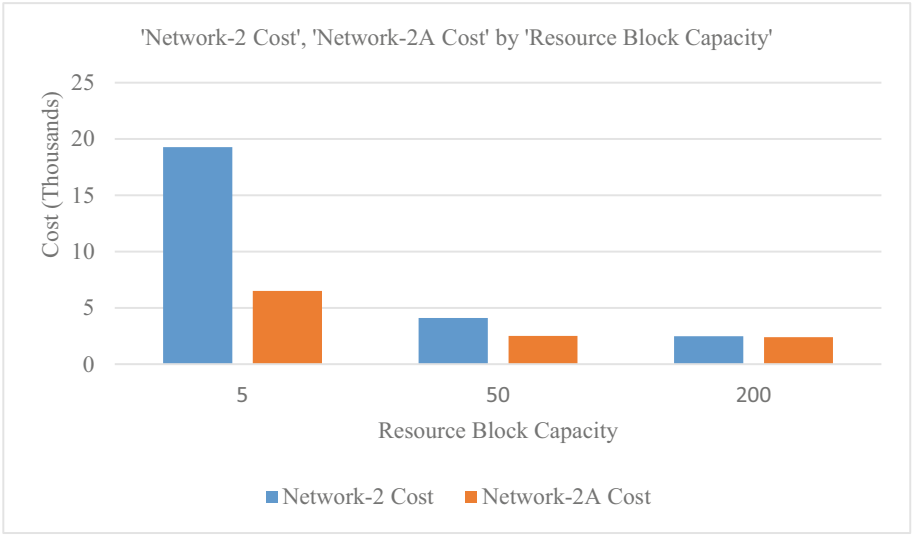


Fig. 17. Total Cost in Network 2 and 2A for different Communication Block Capacity in Professional Live Video

9 Conclusion

As the utilization of resources increases the overall delivery cost also increases. An increase in capacity reduces the usage of resources hence reducing the overall costs whereas an increase in rate for constant resource capacity linearly increases the cost. Overlapping reduces the use of links, which reduces the overall usage of resources as a result of overall cost reductions but to achieve this the capacity of resources should be large enough based on the requirements. For a larger network, adding a computation node at the access layer reduces the overall cost. The work focuses on a single metropolitan area which can be extended to multiple metropolitan areas using the same optimization concepts in future work.

References

1. Balasundaram, S.R., Banu, S.: Cost effective approaches for content placement in cloud CDN using dynamic content delivery model. Int. J. Cloud Appl. Comput., 01 July 2018
2. Llorca, J., Sterle, C., Tulino, A.M., Choi, N., Sforza, A., Amideo, A.E.: Joint content-resource allocation in software defined virtual CDNs. In: 2015 IEEE International Conference on Communication Workshop (ICCW), London, UK, pp. 1839–1844 (2015)
3. Cai, Y., Llorca, J., Tulino, A.M., Molisch, A.F.: Compute - and data-intensive networks: the key to the metaverse. In: 2022 1st International Conference on 6G Networking (6GNet), Paris, France (2022)
4. Llorca, J., Tulino, A.M., Varvello, M., Esteban, J., Perino, D.: Energy efficient dynamic content distribution. IEEE J. Sel. Areas Commun. **33**(12), 2826–2836 (2015)
5. Prerna, D., Tekchandani, R., Kumar, N.: Device-to-device content caching techniques in 5G: a taxonomy, solutions, and challenges. Comput. Commun. **153**, 48–84 (2020). ISSN 0140-3664
6. Lin, H., Zeadally, S., Chen, Z., Labiod, H., Wang, L.: A survey on computation offloading modeling for edge computing. J. Netw. Comput. Appl. 102781 (2020)
7. Jiang, C., Cheng, X., Gao, H., Zhou, X., Wan, J.: Toward computation offloading in edge computing: a survey. IEEE Access **7**, 131543–131558 (2019)
8. Barcelo, M., Llorca, J., Tulino, A.M., Raman, N.: The cloud service distribution problem in distributed cloud networks. In: 2015 IEEE International Conference on Communications (ICC), London, UK, pp. 344–350 (2015). https://doi.org/10.1109/ICC.2015.7248345
9. Barcelo, M., Correa, A., Llorca, J., Tulino, A.M., Vicario, J.L., Morell, A.: IoT-cloud service optimization in next generation smart environments. IEEE J. Sel. Areas Commun. **34**(12), 4077–4090 (2016)
10. Feng, H., Llorca, J., Tulino, A.M., Raz, D., Molisch, A.F.: Approximation algorithms for the NFV service distribution problem. In: IEEE INFOCOM 2017 - IEEE Conference on Computer Communications, Atlanta,GA,USA, pp. 1–9 (2017)
11. Poularakis, K., Llorca, J., Tulino, A.M., Tassiulas, L.: Approximation algorithms for data-intensive service chain embedding. In: Proceedings of the Twenty-First International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing (Mobihoc 2020). Association for Computing Machinery, New York, NY, USA, pp. 131–140 (2020)