Evaluating Performance of Content Cache Placement in a Wireless Community Network

Chikomborero Mwenje and Josiah Chavula(✉)

Computer Science Department, University of Cape Town, Cape Town, South Africa
jchavula@cs.uct.ac.za

Abstract. Community networks are often associated with bandwidth constraints. The limited bandwidth capacity in community networks results in higher content delivery time (latency) and reduces quality of service. Unplanned cache placement in the community networks has the potential to result in higher delays and increased network traffic. This study evaluates cache placement and content distribution in a community network using a distributed caching strategy. Latency, throughput and video performance measurements were carried out for geography, delay and hop count cache placement. In this study, hop count cache placement resulted in the lowest average latency, highest average throughput and best video performance. Overall, the study shows lower average latency, higher average throughput and better video performance at the caches compared to the main server. This reinforces the effectiveness of content caching in improving network performance in wireless community networks.

Keywords: Community networks · Content caching · Quality of experience · Quality of service

1 Introduction

The rapid evolution of mobile network communication technologies has led to increased demand for digital content. Over time, investments in digital infrastructure has increased access to Internet with over 80% of people in South Africa covered with 3G Internet [1]. While this is significant progress in enhancing access to digital content, there are still challenges in Internet reliability and affordability. Most South Africans access the Internet using prepaid data through mobile network operators that are among the most expensive Internet service providers in the country [2].

Community networks are increasingly seen as potential solutions to the affordability barrier. Community networks are shared Internet facilities that involve collective maintenance and utilisation of the network infrastructure [3]. Community networks employ network techniques to create computer systems

Y. H. Sheikh et al. (Eds.): AFRICOMM 2021, LNICST 443, pp. 305–318, 2022.
https://doi.org/10.1007/978-3-031-06374-9_20

that help to address connectivity gaps in existing Internet infrastructure. The use of community networks substantially lowers the cost of accessing Internet through lowering the cost of commonly accessed content or localised services in a geographical location [4]. The reduced cost of Internet access comes with a strain in bandwidth of most community networks. This is because low or free access to digital content increases content demand in community networks. This usually surpasses the capability of the bandwidth in these community networks. Research on how to optimise infrastructure of community networks has shown that content caching can streamline traffic in such networks. This can improve overall user experience on these networks [5,6].

Studies have also found that most of the network traffic is recurring content requests from users, usually in the same proximity [7,8]. Also, research shows a strong correlation between the popularity of content and geographical location [8]. This makes it possible for strategic content distribution, reducing latency and consumption of network bandwidth. Past studies showed that content caching based on geographical location increases hit rate by 30% [9], and that content caching reduces traffic load by half [10]. However, content caching is widely applied in web caching while community networks rarely employ content caching technique [6,11].

This study evaluates cache placement and content distribution in a community network using distributed caching strategy. Distributed caching techniques employed in this study will enable network systems to handle multiple caches hence multiple requests on the network will not reduce network performance [12], [13]. Also, the study adopts an Information-Centric Network architecture in the form of Content-Centric Network (CCN) architecture [14]. The caching approach used in this study is an in-network caching approach, to ensure that content is not duplicated in the routers [15]. The cache distribution strategies used are transparent strategy and adaptive strategy [14]. The key objectives of this paper are:

1. To analyse performance of cache placement strategies in terms of retrieval latency and resource utilization
2. To evaluate the effect of content distribution strategies on delay and throughput
3. To evaluate the effect of cache placement and content distribution strategies on Quality of Experience

2 Background and Related Work

Smartphone and digital technologies usage has increased over the past years as technology became cheaper and more accessible [16]. The adoption of smart devices in communities has increased demand for digital content and increased congestion in community networks. Edge computing is increasingly seen as a potential solution to congested networks. It is taking the cloud close to the users by creating replicas of the cloud on a smaller scale [17]. One of the most important forms of edge computing is content caching. Content catching is a performance enhancing strategy designed to store content in servers that are located closer to the users [18]. The same content can be stored in several servers in different locations to ensure that users get access to the content in the least possible time.

One of the first efforts towards content caching [17] sought to address the problem of high latency that resulted from users trying to access centralized content. This challenge was driven by increased demand for digital content that exponentially increased content requests thereby overburdening the network resources back-end and front-end of the applications. Research shows that such an increase in content demand causes application crashes and low throughput [19]. This has a negative impact on quality of experience on networks. Content caching is now widely used to improve user experience on networks [17].

2.1 Mobile Edge Computing

Cisco [16] predicted that by 2022 the share of smart devices will be more than thrice the global population. In addition, projections show that about 89% of mobile data traffic will be video streaming [20]. This points to the need for more advanced cloud computing technologies that can resolve high latency problems that result from the massive increase in traffic. Mobile edge computing places network resource management and storage at the edge to reduce latency and mobile energy consumption [21]. Edge computing technologies make use of cloudlets to bring content closer to users [13]. Cloudlet is a mini server connected to a network of smart devices, sharing computational resources at an ultra low latency, mimicking how the cloud operates [22].

With the cloud now replicated and closer to the users, there is need for content and resources that is provided to the users to be customised for the particular population that is accessing the cloudlets. Therefore there is need for the relevant content to be cached in the storage space in the cloudlets.

2.2 Content Catching

The ability of mobile edge computing to provide storage resources is called content caching [23]. However, cache location problem arises when caches are not strategically placed. This incurs a drawback in access latency and bandwidth usage [23]. A greater part of this problem is traffic build-up because of recurring content requests by users [24]. Caching content in cloudlets paired with an ideal cache placement strategy has come with many benefits including cost effectiveness in infrastructure deployment and reduced response time when users request for content.

A number of studies have been done recently on content caching on network edges. *Chen et al.* [25] designed caches in the form of small cell clouds and macro cell clouds to store data about healthcare resources that would be available using a Software Defined Network (SDN) based mobile architecture. The main intention of the caches was to reduce latency during content delivery by prioritising specific packets that could not be delayed. *Chen et al.* [25] focused on integrating data cognitive engine and resource cognitive engine. The user equipment would be a smart health wearable device with sensors. The sensors collected data which would be sent to the cloud for analysis, then the SDN controller would allocate resources according to priority of the most urgent healthcare issue. *Zhang et al.* [26] implemented caching content in vehicles so that users can easily access the content when travelling. The motivation was that vehicular caching enables users to access content much faster as they come across several vehicles on the road. Vehicular caching was mentioned to be less expensive than deploying infrastructure on a large scale. The use of Content Centric Networks (CCN) resulted in efficiency of network energy because it would require less energy as it is based on named data rather than IP addresses hence reduced cache updating. *Zhang et al.* [26] developed an online caching algorithm that enhanced caching in vehicles and resulted in less energy being consumed by the devices. *Gao et al.* [27] stated that caching content in cloudlets is an effective way of reducing latency between the data centre and user leading to an improved network performance. The upshot of a network with a good performance is an enhanced quality of user experience. Cloudlets also result in a prolonged battery life on the users' devices [27]. Caching content in cloudlets is important because it maximises on efficient content reuse [28]. For example, *Chen et al.* [28] highlighted that in most instances, popular content requests are made several times in a certain time period, from media streaming applications.

3 Methodology

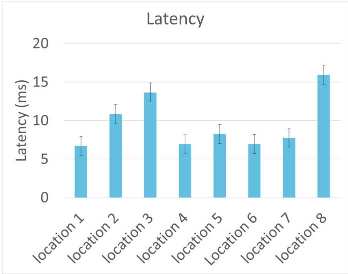
The research was carried out in two phases. The first phase included network measurements for latency and throughput in a wireless community network (name withheld for blind review). Thereafter, a network model of the community network was simulated with content caches added. In the simulated network model, caches were distributed based on three strategies; geography, hop-count and delay. Experiments were then carried out in these different cache placement strategies to obtain throughput, latency and video performance measurements.

3.1 Network Measurements

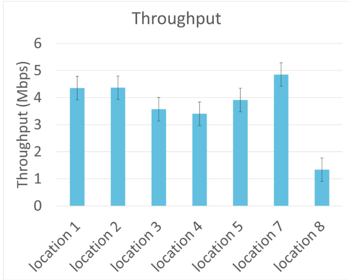
The first stage in the research involved running network measurements in a wireless community network (name withheld for blind review), to measure throughput and latency between access points.

Figure 1a shows the distribution of latencies to each access point, measured from all other access points. The highest average latency was at location 8, with an average of 15.9 ms. Links to location 1 resulted in the lowest latency with an average of 6.72 ms. Latency at location 1 could have been the lowest because location 1 is more centrally located in terms of physical distance and hop count, whereas location 8 is the furthest.

Throughput measurements were also conducted in the community network. This was done using Iperf between all the access points. Figure 1b shows throughput results between access points in 8 different locations in the network. Throughput was highest at location 7, with an average throughput of 4.85 Mbps. The least throughput was recorded at location 8 with an average of 1.34 Mbps.



(a) Latency between access points



(b) Throughput between access points

Fig. 1. Latency and throughput results that would later be used to configure a simulated version of the community network.

3.2 Network and Cache Modelling

The second stage of the research was modelling a virtual network using a Software Defined Network (SDN) controller. Mininet was utilised to create a network of virtual switches, routers, hosts, links and controllers [29]. The network was managed via the Floodlight [30], a component based SDN controller. Floodlight was used to manage flow control with automated route configuration between network devices.

3.3 Experiments

Network measurements were carried out in the emulated community network to compare performance of different cache placement strategies, viz-a-viz, geographic, hop-count, and delay. The measurements were carried out while requesting for the same video that was used in the first stage of the research (hosted on the main server) for an unbiased analysis of the performance.

Geographic Cache Placement: In geographic caching, access points were categorised into three zones as shown in Fig. 2. In each zone the access point most directly connected to the main server was used to host the cache. Performance from each access point was measured by requesting for a video that was hosted on the main server, and also by requesting for the same video from the cache within the zone.

Hop-Count Cache Placement: A hop-count caching was implemented by the use of Dijkstra's algorithm to calculate the shortest path from the main server to all the access points. Caches were then placed on every second hop access point connected from the main server, as shown in Fig. 3. The performance of the network was measured by requesting for the same video that was used in geographic caching, from all the access points in the network.

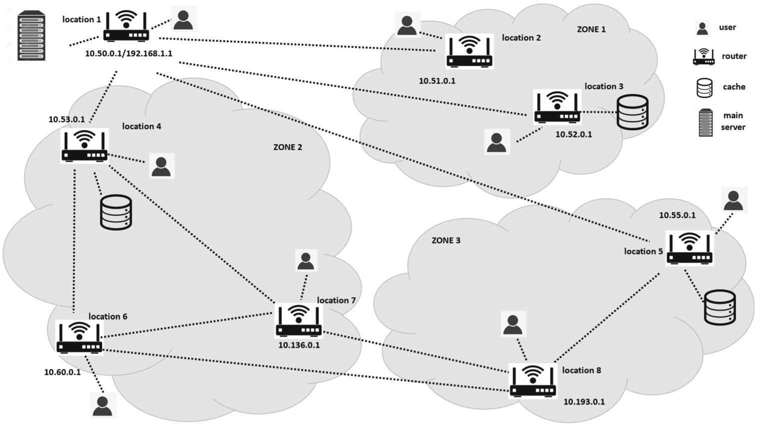


Fig. 2. Geographic caching

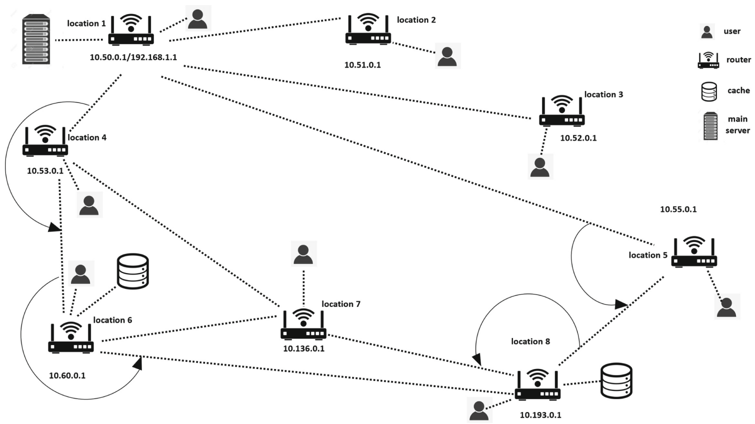


Fig. 3. Cache placement based on Hop-count

Delay-Based Cache Placement: The cache placement in the experiment was based on the round-trip-time (RTT) between each access point and the main server. Access points were thus grouped into high, middle and low delay. Access points within the middle range were selected to host the caches as shown in Fig. 4. Measurements were carried out from each access point to determine the performance of the placement strategy when the video was requested from the caches.

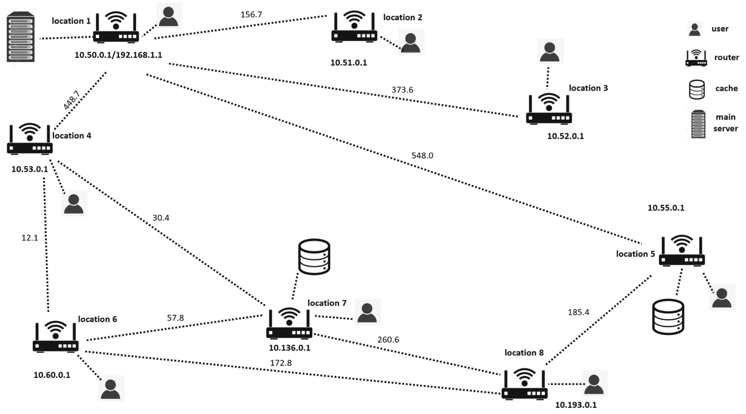


Fig. 4. Cache placement based on delay

4 Performance Results

Network measurement results for latency and throughput were used to simulate a community network in Mininet. In the simulated network, caches were placed according to three cache placement strategies, thus geography, hop count and delay. For hop count cache placement, caches were placed after every 2 hops from the main server. This resulted in 2 caches being placed in the network. In delay cache placement strategy, latency between links in the network was measured. Caches were placed at the access point that had the median delay from the main server. For geography cache placement, the network was separated into 3 geographic zones and a cache was placed in each zone. The evaluation was done by performing latency, throughput and video performance measurements then comparing the outcome from the 3 caching strategies.

4.1 Latency

Latency measurements were carried out to measure the delay between a cache and a user connected to a specific access point. Latency was measured in milliseconds. Ping tests were carried out to measure latency. Best network performance was achieved when latency was lowest. High delay results in poor user quality of experience.

Figure 5 presents average latency per cache placement strategy, between the 8 access points and caches in the network. On average, the main server resulted in the lowest average latency of 1.57 ms, when measurements were carried out from all the 8 access points compared to the 3 caches. However, latency measurements carried out from caches to only access points directly linked to them resulted in lower latency readings compared to the readings recorded at the main server. Thus, the main server seemed to have low latency because it is central when in actual fact latency readings at caches were lower. Thus, caches resulted in better network performance in terms of latency.

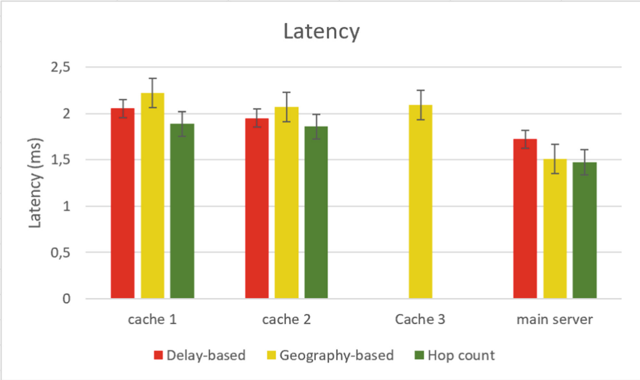


Fig. 5. Latency measured between caches and access points

Comparing the cache placement strategies, Overall, hop count cache placement produced the lowest average latency in the network and geography cache placement resulted in the highest latency. Measurements carried out at Cache 1 showed that hop count cache placement resulted in the lowest total latency with an average of 1.886 ms. Geography cache placement resulted in the highest total latency with an average of 2.217 ms. At cache 2, hop count had the least total latency, with an average of 1.859 ms. The highest total latency was recorded for geography cache placement with an average of 2.067 ms.

4.2 Throughput

Throughput measurements were carried out by running Iperf between caches and access points. Figure 6 shows throughput between access points and caches placed according to geography, hop count and delay cache placement strategies.

Hop count cache placement generally provided better throughput compared to the results obtained for geography and delay cache placement. At cache 1 hop count cache placement resulted in the highest average throughput, with an average of 10.05 Mbps. The lowest throughput was recorded for delay cache placement, with an average of 4.08 Mbps. Delay cache placement resulted in the highest throughput at cache 2 with an average of 8.67 Mbps. The average throughput for delay cache placement was 1.39 Mbps higher than for hop count cache placement. The lowest throughput was recorded for geography cache placement, with an average of 4.57 Mbps. At cache 3 geography cache placement had an average throughput of 7.04 Mbps. The average throughput at cache 3 was lower than the highest for cache 1 and cache 2. At the main server hop count cache placement had the highest throughput with an average of 9.15 Mbps. The lowest throughput was recorded for geography based cache placement, with an average of 7.88 Mbps. Hop count cache placement might have resulted in the highest throughput because caches were evenly distributed in the network. Geography cache placement resulted in the lowest throughput because caches were

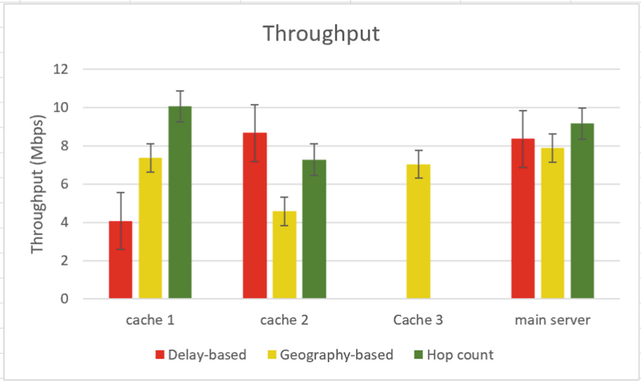


Fig. 6. Throughput between caches and access points

distributed according to geographic locations, when location 2 had more users than location 1 and 3.

4.3 Video Performance

Video performance was measured using AStream, where a nine minute (540 s) video saved in each cache was played from the access points. AStream recorded the time to initial buffering, the earliest time the video started playing the highest resolution of 1080p, and total time to play the whole video. Total time in playing the video included buffer time and network delays. The best quality is when video started playing at its best quality earliest and total video playtime was minimum.

4.4 Buffering

On average, lowest results for initial buffering time in the simulated network were recorded during hop count cache placement. Figure 7 shows that at cache 1, the earliest initial buffering was recorded for delay based cache placement with an average of 1.48 ms. Geography based cache placement resulted in the latest initial buffering at cache 1 with an average of 2.18 ms. At cache 2 the hop count cache placement had the earliest buffering with an average of 1.19 ms and the latest buffering was recorded for geography based cache placement with an average of 2.71 ms.

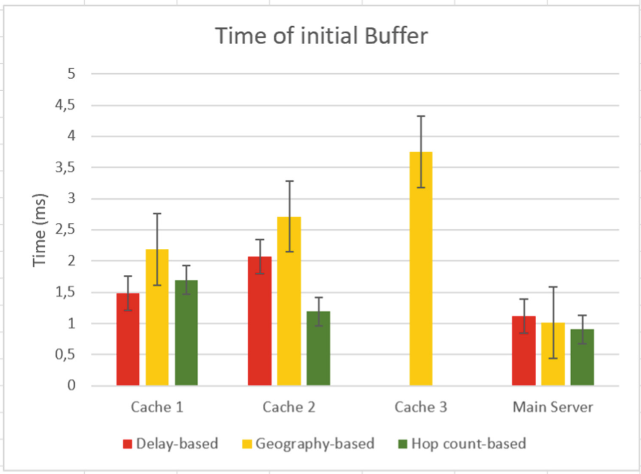


Fig. 7. Time when buffering initially occurred while playing the video

4.5 Video Quality

Figure 8 shows the results for the time at which the video started playing its best quality. The average time the video started playing on its best quality latest was at cache 3 with an average of 35.20 s. Overall, hop count cache placement resulted in the best video quality starting to play earliest. At cache 1, the best video quality started playing earliest in hop count cache placement with an average of 33.9 s. The best quality played latest in delay based cache placement with an average of 35.8 s. For cache 2, the earliest the best quality was played was in hop count cache placement at an average of 34.07 s. The latest time the best video quality was played at cache 2 was in geography based cache placement at an average of 35.50 s. At cache 3, geography cache placement the best video quality started playing at an average of 35.20 s. From the main server, results showed that hop count cache placement had the best video quality played earliest at an average of 33.52 s.

4.6 Video Play Time

Figure 9 shows the average total video playtime in the simulated network. Cache 1 resulted in the lowest average total playtime, with an average of 597.01 s when the video was played from all the 8 access points. The highest average total playtime when the video was played from all the access points was recorded at the main server with an average of 597.03 s.

Overall, hop count cache placement resulted in lower average delay while accessing content from the caches, higher throughput and better video performance compared to geography cache placement and delay cache placement. Hop count cache placement took lesser time from the time the video started playing

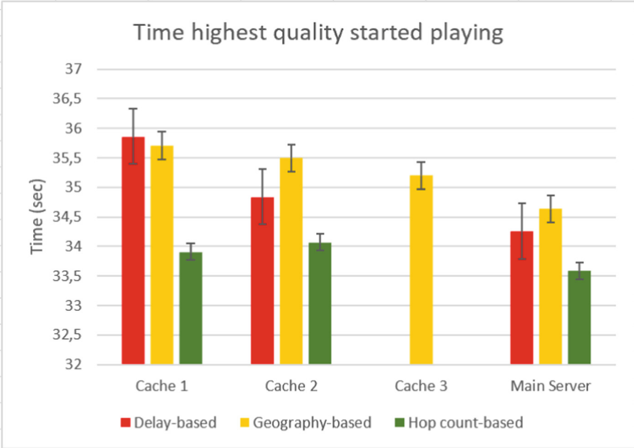


Fig. 8. Time when highest video quality started playing

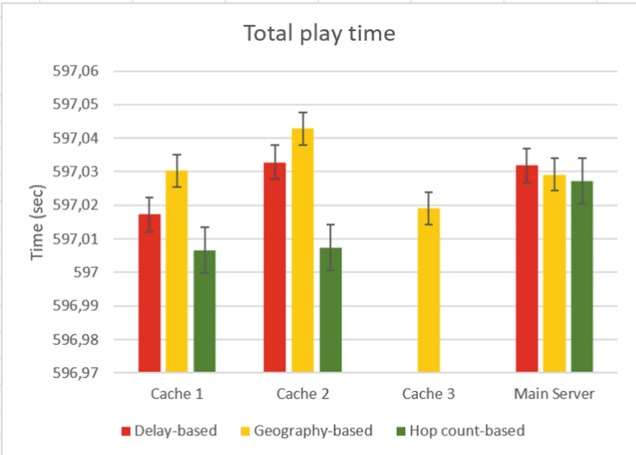


Fig. 9. Total time it took to play the full video

for buffering to occur and for the highest quality to start playing. Average total video playtime in hop count cache placement was the least compared to delay and geography cache placement.

5 Discussion

The aim of this research was to determine effectiveness of content caching in community networks. This was achieved by measuring network performance and quality of service. The network performance was measured by performing latency and throughput tests in the network. User quality of experience was evaluated through video streaming performance while connected to the network through an access point. Video performance was measured based on initial buffer time, the duration to reach highest video quality, and total video playtime.

Latency, throughput and video performance measurements were carried out between the main server and access points in 8 locations in the network. Upon comparing the results obtained from measurements done at the caches and the main server in the simulated network, average latency at the main server was higher than average latency recorded at the caches for all 3 caching strategies. Thus the caches resulted in lesser time for initial buffering to occur, less time taken for video to start playing on its best quality and less time for the video to complete playing compared to the main server.

A comparison between the cache placement strategies was done to determine which strategy performed best in the simulated network. Latency, throughput and video performance measurements were carried out for geography, delay and hop count cache placement. Amongst the 3 caching strategies implemented in the network simulation (geography, latency and hop count) the best performing caching strategy was hop count cache placement. Average latency recorded in the simulated network showed that hop count cache placement had the lowest average latency(1.74 ms) and the highest average latency (1.97 ms) was recorded during geography cache placement. Hop count cache placement produced the highest average throughput (8.83 Mbps) in the simulated network, while geography cache placement produced the lowest average throughput of 6.71 Mbps. Still comparing the 3 cache placement strategies, the best video performance was recorded for hop count cache placement as it had the lowest average initial buffer time (1.27 ms), lowest average time highest quality started playing (33.83 s) and lowest average total playtime(597.01 s).

Overall, content distribution under cache placement strategies relatively enhanced network performance. This suggests that content caching strategies are effective in improving performance in community networks. Comparing the various caching approaches, the results suggested that hop count-based caching had the highest performance in the three dimensions investigated in this study.

6 Conclusion

This research examined the effectiveness of cache placement strategies on network performance. The analysis also sought to identify the best cache placement strategy if content caching produced better network performance compared to the readings observed on the community network. All 3 cache placement strategies, thus delay cache placement, geography cache placement and hop count cache placement, resulted in a better network performance based on latency, throughput and user quality of experience when playing a video online. This demonstrated that cache placement could improve network performance and user experience. Content caching in the simulated community network resulted in an improved user quality of experience with reduced latency and increased throughput. Caching video content resulted in the users viewing better quality videos. The next level of the analysis showed that hop count-based cache placement produced the best network performance compared to both geography and delay-based cache placement strategies.

The outcome of this research was an optimal cache placement and content distribution strategy for the community network. Content was cached in routers that were allocated storage space. Implementing the content distribution strategies in this study enabled communities to get access to content easily and develop faster, without straining their financial capacity.

References

1. GSMA. Mobile internet connectivity 2019 sub-saharan africa fact sheet. In: Mobile Internet Connectivity 2019 Sub-Saharan Africa Fact sheet, pp. 1–2. GSMA (2019)
2. Mathur, A., Schlotfeldt, B., Chetty, M.: A mixed-methods study of mobile users' data usage practices in south africa. In: Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing, pp. 1209–1220. ACM (2015)
3. Internet Society. Community networks. https://www.internetsociety.org/issues/community-networks/. Accessed 30 April 2021
4. Hadzic, S., Phokeer, A., Johnson, D.: Townshipnet: a localized hybrid tvws-wifi and cloud services network. In: 2016 IEEE International Symposium on Technology and Society (ISTAS), pp. 1–6. IEEE (2016)
5. Melamed, S., Bigio, Y.: Bandwidth savings and qos improvement for www sites by catching static and dynamic content on a distributed network of caches. US Patent Application 10/332,842, 1 July 2004
6. Wang, J.: A survey of web caching schemes for the internet. ACM SIGCOMM Comput. Commun. Rev. **29**(5), 36–46 (1999)
7. Phokeer, A., Densmore, M., Johnson, D., Feamster, N.: A first look at mobile internet use in township communities in South Africa. In: Proceedings of the 7th Annual Symposium on Computing for Development, p. 15. ACM (2016)
8. Brasche, G.P., Fesl, R., Manousek, W., Salmre, I.W.: Location-based caching for mobile devices. US Patent 7,519,470, 14 April 2009
9. Ma, G., Wang, Z., Zhang, M., Ye, J., Chen, M., Zhu, W.: Understanding performance of edge content caching for mobile video streaming. IEEE J. Sel. Areas Commun. **35**(5), 1076–1089 (2017)
10. Wang, Y., Li, Y., Wang, W., Song, M.: A locality-based mobile caching policy for D2D-based content sharing network. In: 2016 IEEE Global Communications Conference (GLOBECOM), pp. 1–6. IEEE (2016)
11. Barish, G., Obraczke, K.: World wide web caching: trends and techniques. IEEE Commun. Mag. **38**(5), 178–184 (2000)
12. Johnson, T.A., Seeling, P.: Browsing the mobile web: device, small cell, and distributed mobile caches. In: 2015 IEEE International Conference on Communication Workshop (ICCW), pp. 1025–1029. IEEE (2015)
13. Liu, D., Yang, C.: Energy efficiency of downlink networks with caching at base stations. IEEE J. Sel. Areas Commun. **34**(4), 907–922 (2016)
14. Abdullahi, I., Arif, S., Hassan, S.: Survey on caching approaches in information centric networking. J. Netw. Comput. Appl. **56**, 48–59 (2015)

15. Ioannou, A., Weber, S.: A taxonomy of caching approaches in information-centric network architectures. Elsevier Journal (2015)
16. Cisco VNI Forecast: Cisco visual networking index: Forecast and trends, 2017–2022. White paper, Cisco Public Information (2019)
17. Dilley, J., Maggs, B., Parikh, J., Prokop, H., Sitaraman, R., Weihl, B.: Globally distributed content delivery. IEEE Internet Comput. **6**(5), 50–58 (2002)
18. AVI Networks. Content caching definition. In Content Caching Definition, AVI Networks VMware, p, 1. http://avinetworks.com/glossary/content-caching/. Accessed 1 June 2020
19. Chetty, M., Sundaresan, S., Muckaden, S., Feamster, N., Calandro, E.: Measuring broadband performance in south africa. In: Proceedings of the 4th Annual Symposium on Computing for Development, p.1. ACM (2013)
20. Cisco Visual Networking Index. Global mobile data traffic forecast update, 2017–2022 white paper. Technical report (2019). https://www.cisco.com/c/en/us. Accessed 19 June 2020
21. Mao, Y., You, C., Zhang, J., Huang, K., Letaief, K.B.: A survey on mobile edge computing: the communication perspective. IEEE Commun. Surv. Tutorials **19**(4), 2322–2358 (2017)
22. Jararweh, Y., Tawalbeh, L., Ababneh, F., Dosari, F.: Resource efficient mobile computing using cloudlet infrastructure. In: 2013 IEEE 9th International Conference on Mobile Ad-hoc and Sensor Networks, pp. 373–377. IEEE (2013)
23. Nuggehalli, P., Srinivasan, V., Chiasserini, C.-F., Rao, R.R.: Efficient cache placement in multi-hop wireless networks. IEEE/ACM Trans. Netw. **14**(5), 1045–1055 (2006)
24. Müller, S., Atan, O., van der Schaar, M., Klein, A.: Context-aware proactive content caching with service differentiation in wireless networks. IEEE Trans. Wireless Commun. **16**(2), 1024–1036 (2016)
25. Chen, M., Qian, Y., Hao, Y., Li, Y., Song, J.: Data-driven computing and caching in 5G networks: architecture and delay analysis. IEEE Wireless Commun. **25**(1), 70–75 (2018)
26. Zhang, Y., Li, C., Luan, T.H., Fu, Y., Shi, W., Zhu, L.: A mobility-aware vehicular caching scheme in content centric networks: model and optimization. IEEE Trans. Veh. Technol. **68**(4), 3100–3112 (2019)
27. Gao, Y., Hu, W., Ha, K., Amos, B., Pillai, P., Satyanarayanan, M.: Are cloudlets necessary? School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA, Technical report, CMU-CS-15-139 (2015)
28. Chen, M., Hao, Y., Hu, L., Huang, K., Lau, V.K.: Green and mobility-aware caching in 5G networks. IEEE Trans. Wireless Commun. **16**(12), 8347–8361 (2017)
29. Sharma, K.K., Sood, M.: Mininet as a container based emulator for software defined networks. Int. J. Adv. Res. Comput. Sci. Softw. Eng. **4**(12) (2014)
30. Floodlight group. Floodlight openflow controller (oss). https://github.com/floodlight/floodlight. Accessed 13 Dec 2020