The Usage of CDN for Live Video Streaming to Improve QoS. Case Study: 1231 Provider

Wella Edli Shabrina, Dodi Wisaksono Sudiharto, Endro Ariyanto, and Muhammad Al Makky
School of Computing, Telkom University, Bandung, Indonesia
Email: wellaedli@gmail.com, {dodiws, endroa, malmakky}@telkomuniversity.ac.id

Abstract—Streaming is a popular technology used by active users for enjoying audio or video. Broadly, this technology needs high bandwidth to carefully keep its Quality of Service (QoS) at a reasonable level. Without enough bandwidth, a problem arises, such as packet loss. This condition can decrease the quality of content delivery. To properly handle that problem, cache technology can be utilized. One type of these technologies is Content Delivery System (CDN). Naturally, the position of CDN has to be placed not far from the user area, so the access time can be faster than the access time when the CDN is not used. Another contributing factor, such as the right video format selecting can provide a good impact. There are two popular formats for live video streaming, such as HLS (HTTP Live Streaming) and RTMP (Real Time Messaging Protocol). This study is going to elaborate on the comparative between HLS and RTMP with CDN and also without it. The result shows live video streaming with CDN has better performance than without CDN.

Index Terms—Live video streaming, content delivery network, packet loss, HTTP live streaming, real time messaging protocol

I. Introduction

Streaming technology can be naturally used for enjoying standard audio or video content from certain providers. Generally, to maintain the quality of those contents delivered, a channel operated has to require high bandwidth. The streaming via a line with limited bandwidth has a risk because the channel can decrease the quality of the contents sent. The academic field of a computer network has identified possible mechanisms to modestly increase QoS (Quality of Service) of those contents delivered. One of the mechanisms is by using a cache [1].

The cache is temporary storage to increase data access speed if the data are reused again [1]. Broadly, the cache usage in the internet world is functioned when a specific user does the surfing, and the functional is traditionally done by a proxy server. Sometimes, this function of a proxy server has a problem when the data saved are bigger than the cache capacity. Frequently, it introduces a delay for video content transferred if the user tries to maintain received data because the data cannot be cached [1].

In notable addition, another contributing factor makes an impact on content delivery, and also its data loading in the client is codec operation [2]. The codec is naturally needed for the video streaming concretion typically loaded by the user application. It is a critical aspect because the target user may employ a device that has limited computation ability, such as a non-PC device.

Presently, there are modern architectures use a familiar pattern like the cache function in proxy server. They are CDN (Content Delivery Network) and NDN (Named Data Network) [3], [4]. Even though the NDN is undoubtedly a new architecture related to how data communication is performed, CDN is a popular mechanism for video streaming up until now.

For video data streaming, two practical methods can be exhibited. The first is on-demand, and the countless other is live streaming [5]. Frequently, the live streaming method has a challenge relatively bigger than the on-demand method because the data delivery has to be done in real-time. So, maintaining related to the data delivery has to be executed with the latency level as small as possible.

For Indonesian people live modestly in foreign countries, an instinctive desire to appreciate digital content typically produced in Bahasa may remain a requirement. For that reason, The 1231 Provider tries to announce a service such as broadcasting by performing live video streaming with HLS (HTTP Live Streaming) or RTMP (Real Time Messaging Protocol). Those streaming protocols permit a significant quality of live video streaming to be sent instantly.

However, when live streaming is performed, the limited bandwidth factor can make the quality of video delivery decreases. Another factor, such as the long-range between the client and the server, creates a long delay for video data delivery. CDN can be properly utilized for video streaming to maintain QoS of content delivery carefully. The stressing point is CDN position has to be as closer as possible to the target user location, so by that way, the throughput created can be maximized.

In addition, there are still a few Indonesia companies that have a core business in live video broadcasting that uses CDN infrastructure. Mainly focusing on Indonesian customers live abroad, such as in California or Tokyo.

In this study, there is a case that 1231 Provider, a local broadcaster in Bandung, Indonesia, has no backbone infrastructure, so the cloud infrastructure is a logical

Manuscript received September 12, 2019; revised March 5, 2020.
doi:10.12720/jcm.15.4.359-366

choice for its successful operation. AWS (Amazon Web Services) is wisely selected because it is a popular cloud provider, and the principal payment can be not as expensive as if 1231 Provider itself builds the backbone. Besides cloud technology, 1231 Provider also exhibits CDN, and both HLS and RTMP to increase video streaming performance. Its clients are placed in California and also in Tokyo.

II. Related Works

Some architectures can be used related to the conceptual of the cache, such as a proxy server, CDN, and NDN [3], [4]. In general, the popularity of CDN in practical implementation for video streaming is higher than NDN's popularity [4]. It happens because NDN is still in progress to achieve its stable development [6]. Invaluable addition, CDN is also popular because its usage can minimize packet loss, as a possible result, it can maximize throughput [7].

The published paper of [8] recommends a comprehensive framework to compare NDN and CDN objectively. It typically exhibits cache distribution performed by CDN, or NDN demonstrates significant contribution for QoS. However, the CDN can offer better performance slightly than NDN when the cache capacity needed is large.

In [9] presents RTMP player requires higher CPU power than HLS player because RTMP operates on small frames instead of large chunks, which leads to significantly higher processing overhead.

The study of [10] exhibits HLS media server for multi-bitrate VOD service can increase storage and power efficiency by using real-time transcoding according to client request pattern.

In [11] shows HLS is the most popular commercial protocol. HLS can support live streaming for many devices (in terms of maximum resolution or maximum bitrate can be supported by the devices) depends on the network condition.

The observation of [12] presents HLS has better performance than RTMP related to delay parameter. However, according to the small percentage of packet loss, RTMP works better than HLS.

The RTMP is used as a basic protocol for comparison because it can be stated as the first streaming protocol created by Macromedia (now Adobe). Based on the literature above, this study examines live video streaming by comparing between HLS and RTMP for both conditions (with CDN and without it). CDN is performed by exploring cloud technology provided by AWS (e.g., AWS CloudFront). The parameters explored as performance metrics are throughput and packet loss ratio.

A. *The Cache Architectures on the Internet World*

1) *Proxy server*

The proxy server is performed to save a web page to local cache from the internet temporarily. It happens when the user surfs on the internet, loads the web page for the first time. The proxy server is going to catch that content into the cache instantly. So when another user is going to launch the same page (beforehand the cache is refreshed), then the proxy server is going to serve that page. So, the content is going to be loaded faster than if the page is downloaded from the original site.

2) *Content delivery network*

CDN is a group of nodes connected to each other and acts as content replication placed in strategically position around the world of the computer network. The content cached can be a web page or video, including media streaming [5]. Related to its design, CDN has four purposes goals which are performance, reliability, scalability, and responsiveness [13].

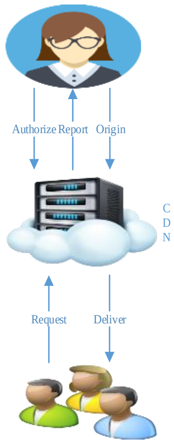

Fig. 1. The caching mechanism of CDN

The CDN mechanism is started when the client requests web content from an origin server. The server is going to respond to the client requesting by sending the web content to the client. (Fig. 1)

In addition, the origin server is going to produce duplications of the web content and forward them to the CDNs. Every content changing in the origin server, the CDN servers are going to update it in their cache storages instantly. Every CDN server can act as a content provider. However, the CDN server closest to the client position has a bigger chance of being the provider. Logically, the content is faster to be downloaded by the client than if the origin site delivers it.

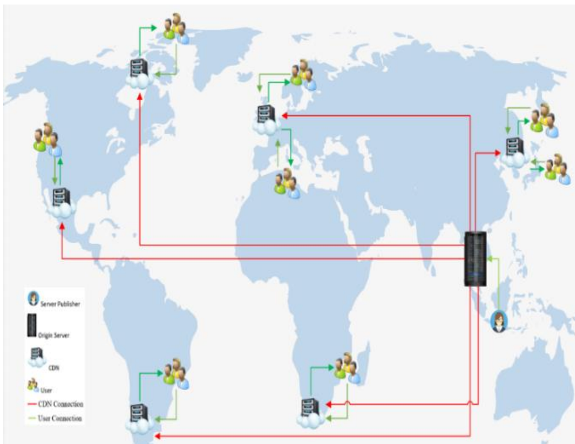

Fig. 2. The illustration of CDN servers

For example, since a client in Japan downloads content from Indonesia, the request to the origin server can be longer than the request to the nearest CDN. It happens due to the distance between the two countries. By using CDN, the requested content place is not as far as to the origin site (located in Indonesia), but it is located in the closest CDN server (located in Japan). (Fig. 2)

3) Named data networking

Named Data Networking (NDN) is a new network architecture. The NDN packet carries the data name (Name), not the source address or destination address (e.g., IP address). NDN is intended to replace the host-centric into data-centric architecture [3]. In NDN, the IP address is replaced by Name (data name), and for data communication, NDN uses two types of packages: Interest and Data. Both Interest and Data carry a Name to identify data uniquely. The node requests Data is called as Consumer. On the other hand, the node replies the request is determined as Producer [3].

B. Feature Comparing between Proxy Server and CDN for Caching Files

There are disabilities owned by the proxy server, and CDN can handle them. The first disability is related to its scalability. The proxy server only covers the local area network. On the other hand, CDN projection can be as wide as a wide area network. The CDN cache capacity is larger than the proxy server cache capacity. Consequently, CDN can handle the weakness of the proxy server, such as caching large files (e.g., video files). CDN also suitable for the application needs authentication feature, while the proxy server cannot execute that feature.

By using large cache capacity, CDN can reduce the latency of data streaming, or it can be said that CDN can increase the throughput of data streaming. This condition also happens because CDN distribution can cover a large area network. As a result, the CDN position serves the client request can be the closest one.

Sometimes, CDN can be called as Reverse Proxy. By this operation, CDN is going to cache the content automatically from the origin server. This content is going to be loaded by another client when there is a next request related to it.

C. The Streaming Mechanism of HLS and RTMP Related to CDN Implementation

The HLS format strives to maintain video quality during the streaming session. The video streaming sent is going to be encoded and divided into several segments. Each segment is stored in storage or buffer alternately during the streaming period. However, on the client-side, the sized content in each segment is dynamically adjusted to the ability of the player to display the streaming video [14].

The RTMP format is going to divide the video stream into fragments, and the size of each fragment can dynamically change. It depends on the connection conditions between the client and the server. However, fragment sizes are generally fixed. If the bandwidth decreases to under the lowest limit of the bandwidth size required for content delivery, the client player is going to be continuously interrupted [14]. There is no additional feature to maintain the bitrate of the content received by the player likes in HLS format.

When the CDN is utilized, the impressive performance of streaming video sent can be further improved. It is related to the notable features of the service protocol that supports HLS and RTMP, namely TCP (Transmission Control Protocol). TCP has several features, which are flow control and retransmission. The mechanism of content duplication on CDN makes the distance between server and client is as though reduced. The content duplicated from the original site is going to be placed at the nearest CDN location to the client. So, when the client requests the content, the content delivery mechanism no longer needs to retrieve from the origin server can be farther away. Regarding the flow control feature, this condition can reduce the value of RTT (Round Trip Time) [15].

The existence of the CDN also accommodates client requests. So that, it can reduce the traffic requests to the server. Logically, this condition can improve server performance by avoiding server failure due to too many requests. According to the retransmission feature, if there is a packet sending failure, the retransmission of the packet can be directly served (in part or whole) by the CDN located closest to the client. This operation makes the number of packet loss can be reduced, and it also can increase the throughput automatically [15].

D. Amazon Web Services

AWS is a cloud service platform that proposes resources such as computation, database, content storage, content delivery, and any other functionalities to support business or organization operation. The resources requested can be plotted as flexible as possible to achieve a certain requirement. The payment method uses “pay as you go” to make it can adequately meet the business requirement of SMEs (Small and Medium-sized Enterprises) such as 1231 Provider.

The services of AWS can be divided into three types. They are SaaS (Software as a Service), PaaS (Platform as a Service), and IaaS (Infrastructure as a Service).

- SaaS – is a cloud service that provides any software. The usage of SaaS makes the user has not the flexibility to configure an instance selected because it is a service provided as a default by the cloud provider. The examples of SaaS are Terraform, Cloudflare [16].
- PaaS – is a cloud service that offers a platform to support software development. In this way, the cloud consumer can build an app, make testing, configure the app, and put the app that has been released into the cloud. The example of PaaS is Heroku [16].
- IaaS – is an infrastructure service in the cloud, such as virtualization, bandwidth, and network. IaaS permits

the user to maintain a full control for adding or releasing resources and also installing an application in a virtual machine made. The example of IaaS is AWS Cloudfront [16].

1) Amazon EC2

Amazon Elastic Compute Cloud (Amazon EC2) is a computation platform and acts as a virtual computer that can be configured. The instance can be built by a combination of CPU capacity, memory, storage, and network to cover a requirement needed. The instance can be executed quickly because the state condition is from ready to running [17].

2) AWS CloudFront

CloudFront is a service provided by AWS and acts as global CDN used for distribution of content from the original site [18].

III. Problem Definition

Live video streaming is a challenging service because it needs attention to maintain its service related to its packet delivery process. For Indonesians live abroad, getting the latest news from Indonesia is a desire. However, there are still very few Indonesian companies engage in live broadcasting service.

In this study, the case of 1231 Provider, a small enterprise, is explored. It operates broadcasting in live video streaming with HLS and RTMP formats. The starting point of the broadcaster is located in Bandung, Indonesia, and the clients are placed in Tokyo and California. When the streaming process is run, the limited bandwidth factor and the distance between the server and the remote client makes the video quality decreases. To overcome this problem, 1231 Provider utilizes CDN infrastructure to maintain QoS of the content sent. CDN infrastructure can be performed by utilizing AWS CloudFront. The exploration of the study focuses on the parameters of throughput and packet loss ratio.

IV. Proposed System

This study is going to compare between HLS and RTMP in live video streaming service. Two schemes CDN and the other one (without CDN) are also performed. The performances of them are evaluated by using the tool of Wireshark. Wireshark is a tool performed to capture packet flows on the network. The data captured can be analyzed for observation purposes [19]. The parameters observed in the study are throughput and packet loss ratio.

The proposed system built consists of the camera for video recording, encoder, and media converter for converting the video format without decreasing its quality so it can be compatible with any device loaded it. The last is the CDN itself as a cache of content delivered to the client. The summary of the proposed system can be seen in the following Fig. 3:

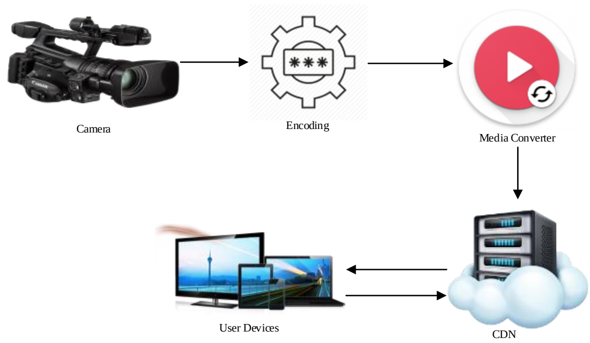

Fig. 3. The architecture of the proposed system

The environment configured for the study remains:

- The first target user is located in California and the other one is placed in Tokyo.
- The 1231 Provider is located in Bandung, Indonesia.
- The broadcasting type is live video streaming.
- The parameters explored are throughput and packet loss ratio.
- The cloud provider used is AWS, and the global CDN service performed is AWS CloudFront.
- The quality of streaming video encoded has a ratio of 1280x720, and it has frame rate of 30 fps. The protocol input is RTMP (H.264 / AVC) with a standard bitrate of 4500 kbps. The output protocols used for broadcasting are RTMP and also HLS. The audio uses AAC encoder with 44.1 KHz.
- Bandwidth performed in the clients is 110 Mbps (download), and 48 Mbps (upload). The clients are utilized by performing AWS instance (Windows 2016) with t2.micro type. The bandwidth executed in the broadcaster is 18.1 Mbps (download), 5.8 Mbps (upload). The tool utilized for measuring the bandwidth of lines is Telstra Speed Test.

A. The Components of Hardware and Software

The software required in this study are:

- Windows 10 Pro – it is used as a laptop operating system.
- Open Broadcast Service – it is a software to connect between the camera and AWS Elemental MediaLive. It is also used to encode video format to any compatible format. It runs on a laptop with Windows 10 Pro operating system.
- AWS Elemental MediaLive – it is a live video processing service that allows video provider to deliver live video. It acts as an encoder with the H.264 compression standard.
- AWS Elemental MediaPackage – it is the original storage in the process of video workflow provides live video streaming.
- AWS CloudFront – it is worked as a global CDN
- Wireshark – it is act as captured packet for the parameters observed.

The hardware which are required in this study are:

- Camera – DSLR Canon 650D.

  - o Video resolution: 720p
  - o Audio bitrate : 128 Kbps
  - o Framerate : 30 fps
  - o Encoder : H.264
- Laptop (as a broadcaster) with the specification such as:
  - o 256 GB SDD
  - o 8 GB RAM
  - o Processor Intel Core i7
- Client as EC2 instance with the specification such as:
  - o 320 GB SATA
  - o 2 GB RAM
  - o Processor Intel Atom

B. The Assessment Scenarios

The first scenario is exhibited by performing live streaming broadcasting in the HLS and RTMP formats on the network without implementing CDN. The data streaming sent directly from the broadcaster to the target clients located in California and Tokyo.

The global CDN infrastructure is implemented in the second scenario. When the live video streaming has shown, the replication of the content is going to be executed by the CDN and also forwarded to the client. The throughput and also the packet loss ratio for both scenarios are collected by using Wireshark.

Lastly, all measurement results are going to be compared for analysis.

C. Measurement Parameters

QoS parameters can be utilized to analyze the performance of network service, including CDN performance.

The parameters performed in this study are going to be used for analyzing. The tool for capturing packets is performed by using Wireshark. Several parameters conducted are:

- Throughput – is the number of packets can be delivered to the client in a duration of time [20] [21] [22]. The formulation can be seen in the following equation.

$$Throughput = \frac{the\ quantity\ of\ data\ sent}{time\ duration} \tag{1}$$

- Packet Loss Ratio (PLR) – it is the number of packet loss compared with the number of packets delivered [20] [23]. The packet loss ratio formulation is:

$$Packet\ Loss\ Ratio = P_d/P_s\ x\ 100\ \% \tag{2}$$

which: P_d = the number of packet loss
P_s = the number of packet transmitted

V. RESULT AND DISCUSSION

The comprehensive live video streaming assessments of HLS and RTMP are performed by running the assessment scenarios. Each assessment implementation is carried out in 10 times, and each trial is done in 5 minutes.

A. The Comparison of Throughput Captured by Using CDN and Without Using CDN

The throughput captured in the client locations (e.g. California and Tokyo) can be seen in the following table:

TABLE I. THE THROUGHPUT OF EACH TEST SCENARIO

Assessment Scenario	Video Format	Throughput (California)	Throughput (Tokyo)
Using CDN	HLS	3415.9 kbps	4452.6 kbps
	RTMP	912 kbps	869.2 kbps
Without Using CDN	HLS	2994.7 kbps	3990.4 kbps
	RTMP	677.7 kbps	759.7 kbps

Based on Table I, for HLS video format, if the CDN infrastructure is performed, the result of throughput delivers a higher value than the result of throughput if the CDN infrastructure is unutilized. HLS live video streaming with CDN produces 3415.9 kbps while it is captured in the client position California. On the other hand, HLS live video streaming without using CDN generates 2994.7 kbps while it is obtained in the same place. A similar pattern is shown when the throughput is collected in the client position Tokyo. HLS live video streaming with CDN produces 4452.6 kbps. On the other hand, HLS live video streaming without using CDN generates 3990.4 kbps.

For RTMP, if the CDN infrastructure is conducted, the result of throughput also gives a larger value than the result of throughput if the CDN infrastructure is not functioned. RTMP live video streaming with CDN presents 912 kbps while it is captured in the client position California. On the other hand, RTMP live video streaming without using CDN produces 677.7 kbps while it is obtained in a similar place. The identical pattern is displayed when the throughput is collected in the client position Tokyo. RTMP live video streaming with CDN gives 869.2 kbps. On the other hand, RTMP live video streaming without using CDN generates 759.7 kbps.

B. The Comparison of Packet Loss Ratio Captured by Using CDN and Without Using CDN

The packet loss ratio captured in the client locations (e.g. California and Tokyo) can be seen in the following table:

TABLE II. THE PACKET LOSS RATIO OF EACH TEST SCENARIO

Assessment Scenario	Video Format	PLR (California)	PLR (Tokyo)
Using CDN	HLS	0.01 %	0.08 %
	RTMP	0.05 %	0.04 %
Without Using CDN	HLS	0.58 %	0.33 %
	RTMP	0.69 %	0.19 %

Based on Table II, for HLS video format, if the CDN infrastructure is utilized, the result of the packet loss ratio produces lower value than the result of the packet loss ratio if the CDN infrastructure is unperformed. HLS live video streaming with CDN generates 0.01% while it is captured in the client place California. On the other hand, HLS live video streaming without using CDN produces

0.58% while it is obtained in the same place. A similar pattern is exhibited when the packet loss ratio is taken in the client spot Tokyo. HLS live video streaming with CDN generates 0.08%. On the other hand, HLS live video streaming without using CDN produces 0.33%.

For RTMP, if the CDN infrastructure is executed, the consistent result of the packet loss ratio also gives smaller value than the result of the packet loss ratio if the CDN infrastructure is not utilized. RTMP live video streaming with CDN gives 0.05% while it is captured in the client position California. On the other hand, RTMP live video streaming without using CDN produces 0.69% while it is obtained in the same place. A similar pattern is exhibited when the packet loss ratio is collected in the client position Tokyo. RTMP live video streaming with CDN gives 0.04%. On the other hand, RTMP live video streaming without using CDN generates 0.19%.

The CDN can properly accommodate a lot of high traffic by strategically placing a new server closest to the target client so that it can minimize packet loss. Logically, by progressively reducing packet loss, it can naturally create high throughput.

VI. Conclusion

Based on the successful implementation, it is positively confirmed the CDN is going to undoubtedly affect the quality of the video streaming service by progressively reducing packet loss and modestly improving throughput. It naturally makes the existing channel can be used optimally for both RTMP and HLS live video streaming services. This study also proves that cloud technology makes the enterprise categorized as SME is possible to provide the service of live video streaming. AWS CloudFront as a feature of global CDN infrastructure can be used for maintaining the service of live video streaming better than if the CDN is unperformed for live video streaming. For future research, CDN method can be combined with another techniques, especially for mobile content.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

W. E. Shabrina carries out system design and its performance measurements. D. W. Sudiharto's contributes positively to identify the problem definition and explore the practical solution. E. Aryanto does project estimation and M. A. Makky acts as an administrator. All authors have agreed to the final version.

Appendix A HLS Format with CDN in California

CALIFORNIA	Throughput (kbps)	Packet Loss (%)
Observation-1	3677	0
Observation-2	3462	0
Observation-3	3527	0
Observation-4	3287	0
Observation-5	3249	0
Observation-6	4535	0.1
Observation-7	2599	0
Observation-8	3549	0
Observation-9	2387	0
Observation-10	3887	0
Average	3415.9	0.01

Appendix B HLS Format with CDN in Tokyo

TOKYO	Throughput (kbps)	Packet Loss (%)
Observation-1	4842	0.1
Observation-2	4952	0.1
Observation-3	4741	0.1
Observation-4	4721	0.1
Observation-5	4781	0.1
Observation-6	2878	0
Observation-7	5449	0.1
Observation-8	3368	0
Observation-9	4623	0.1
Observation-10	4171	0.1
Average	4452.6	0.08

Appendix C HLS Format without Using CDN in California

CALIFORNIA	Throughput (kbps)	Packet Loss (%)
Observation-1	4060	0.5
Observation-2	2494	0.7
Observation-3	2436	0.3
Observation-4	2656	0.8
Observation-5	3694	0.2
Observation-6	2109	0.6
Observation-7	2140	1
Observation-8	3912	0.6
Observation-9	3347	0.6
Observation-10	3099	0.5
Average	2994.7	0.58

Appendix D HLS Format without Using CDN in Tokyo

TOKYO	Throughput (kbps)	Packet Loss (%)
Observation-1	1567	0.1
Observation-2	4932	0.3
Observation-3	5265	0.3
Observation-4	5242	0.5
Observation-5	5212	0.4
Observation-6	4325	0.2
Observation-7	3537	0.3
Observation-8	4700	0.3
Observation-9	1956	0.5
Observation-10	3168	0.4
Average	3990.4	0.33

APPENDIX E RTMP FORMAT WITH CDN IN CALIFORNIA

CALIFORNIA	Throughput (kbps)	Packet Loss (%)
Observation-1	671	0.1
Observation-2	806	0.1
Observation-3	828	0.1
Observation-4	754	0
Observation-5	648	0.1
Observation-6	662	0
Observation-7	1255	0
Observation-8	672	0
Observation-9	671	0.1
Observation-10	806	0.1
Average	828	0.1

APPENDIX F RTMP FORMAT WITH CDN IN TOKYO

TOKYO	Throughput (kbps)	Packet Loss (%)
Observation-1	601	0
Observation-2	969	0
Observation-3	996	0.1
Observation-4	1056	0
Observation-5	1533	0.1
Observation-6	1159	0.1
Observation-7	825	0.1
Observation-8	605	0
Observation-9	634	0
Observation-10	314	0
Average	869.2	0.04

APPENDIX G RTMP FORMAT WITHOUT USING CDN IN CALIFORNIA

CALIFORNIA	Throughput (kbps)	Packet Loss (%)
Observation-1	534	0.5
Observation-2	590	0.8
Observation-3	556	0.8
Observation-4	583	0.6
Observation-5	572	0.8
Observation-6	529	0.5
Observation-7	1200	0.5
Observation-8	516	0.7
Observation-9	1080	1
Observation-10	617	0.7
Average	677.7	0.69

APPENDIX H RTMP FORMAT WITHOUT USING CDN IN TOKYO

TOKYO	Throughput (kbps)	Packet Loss (%)
Observation-1	1059	0.2
Observation-2	214	0.4
Observation-3	920	0.3
Observation-4	1496	0.2
Observation-5	577	0.1
Observation-6	821	0.1
Observation-7	556	0.1
Observation-8	623	0.1
Observation-9	879	0.2
Observation-10	452	0.2
Average	759.7	0.19

REFERENCES

[1] G. S. Paschos, G. Iosifidis, M. Tao, D. Towsley, and G. Caireieee, "The role of caching in future communication systems and networks," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 6, pp. 1111–1125, 2018.

[2] F. Loh, F. Wamser, T. Hossfeld, and P. Tran-Gia, "Quality of service assessment of live video streaming with a remote-controlled drone," in *Proc. 4th IEEE Conf. Netw. Softwarization Work. NetSoft*, 2018, pp. 464–469.

[3] D. W. Sudiharto, A. Herutomo, and Y. N. Rohmah, "The comparison of forwarding strategies between best route, multicast, and access on named data networking (NDN). case study: A node compromised by the prefix hijack," *J. Commun.*, vol. 12, no. 7, pp. 426–432, 2017.

[4] P. Maille, G. Simon, and B. Tuffin, "Impact of revenue-driven CDN on the competition among network operators," in *Proc. 11th International Conference on Network and Service Management (CNSM)*, 2015, pp. 163–167.

[5] A. Lertsinsrubtavee, P. Mekbungwan, and N. Weshsuwannarugs, "Comparing NDN and CDN performance for content distribution service in community wireless mesh network," in *2014 Asian Internet Engineering Conference (AINTEC)*, 2014, pp. 43–50.

[6] X. G. M. Ge, C. Zhen, C. Junwei, G. Zhenhua, and J. Yixin, "A tentative comparison on CDN and NDN," in *Proc. IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2014, pp. 2893–2898.

[7] M. Taha, "A novel CDN testbed for fast deploying HTTP adaptive video streaming," in *Proc. Int. Conf. Mob. Multimed. Commun.*, 2016, pp. 65-71.

[8] M. Mangili, F. Martignon, and A. Capone, "A comparative study of content-centric and content-distribution networks: Performance and bounds," in *Proc. GLOBECOM - IEEE Glob. Telecommun. Conf.*, 2013, pp. 1403–1409.

[9] B. Wang, X. Zhang, G. Wang, H. Zheng, and B. Y. Zhao, “Anatomy of a personalized livestreaming system,” in *Proc. ACM SIGCOMM Internet Measurement Conference (IMC)*, 2016, pp. 485–498.

[10] S. Hur, S. Cho, Y. Kim, and J. S. Sim, “A power- and storage-efficient HLS media server for multi-bitrate VOD services,” in *Proc. IEEE International Conference on Big Data and Smart Computing (BigComp)*, 2017, pp. 193–198.

[11] P. Chakraborty, S. Dev, and R. H. Naganur, “Dynamic HTTP Live Streaming method for live feeds,” in *Proc. International Conference on Computational Intelligence and Communication Networks (CICN)*, 2015, pp. 1394–1398.

[12] R. G. Guniganti and S. Ankam, “A comparison of RTMP and HTTP protocols with respect to packet loss and delay variation based on QoE,” M.S. thesis, Blekinge Institute of Technology, Sweden, 2012.

[13] Z. Guomin, X. Changyou, and C. Ming, “A distributed multimedia CDN model with P2P architecture,” in *Proc. Int. Symp. Commun. Inf. Technol. Isc.*, 2006, pp. 152–156.

[14] P. Juluri, V. Tamarapalli, and D. Medhi, “Measurement of quality of experience of video-on-demand services: A survey,” *IEEE Commun. Surv. Tutorials*, vol. 18, no. 1, pp. 1-20, 2016.

[15] S. Bohacek, “A stochastic model of TCP and fair video transmission,” in *Proc. - IEEE INFOCOM*, 2003, pp. 1134–1144.

[16] ‘Providers’, Terraform. [Online]. Available: https://www.terraform.io/docs/providers/index.html

[17] W. Stalling, *Operating Systems: Internals and Design Principles*, 9th ed., Pearson, 2018.

[18] I. Bermudez, S. Traverso, M. Mellia, and M. Munafo, “Exploring the cloud from passive measurements: The Amazon AWS case,” in *Proc. - IEEE INFOCOM*, 2013, pp. 230–234.

[19] R. Das and G. Tuna, “Packet tracing and analysis of network cameras with Wireshark,” in *Proc. 5th Int. Symp. Digit. Forensic Secur. (ISDFS)*, 2017, pp. 1-6.

[20] S. Park and S. H. Jeong, “Mobile IPTV: Approaches, challenges, standards, and QoS support,” *IEEE Internet Comput.*, vol. 13, no. 3, pp. 23–31, 2009.

[21] M. Gidlund and J. Ekling, “VoIP and IPTV distribution over wireless mesh networks in indoor environment,” *IEEE Trans. Consum. Electron.*, vol. 54, no. 4, pp. 1665–1671, 2008.

[22] K. Noh, S. Yoon, and J. Heo, “Performance analysis of network coding with raptor codes for IPTV,” *IEEE Trans. Consum. Electron.*, vol. 55, no. 1, pp. 83–87, 2009.

[23] I. Haryono, D. W. Sudiharto, and A. G. Putrada, “QoS improvement analysis of VoIP service which uses overlay network. Case study: Calling AWS VoIP gateway from Bandung, Indonesia,” in *Proc. Int. Semin. Appl. Technol. Inf. Commun. (iSemantic)*, 2018, pp. 381–387.

Wella E. Shabrina. She received Bachelor of Engineering from School of Computing, Telkom University, Bandung, Indonesia in 2020. Her research interests include Computer Network and Cloud Computing.

Dodi W. Sudiharto was born in Jakarta, Indonesia, in 1976. He received the Bachelor of Engineering from Department of Electrical Engineering, University of Indonesia (UI), Depok, Indonesia in 2001 and Master of Science in Information Technology from Computer Science, University of Indonesia (UI), Jakarta, Indonesia in 2004. His research interests include Computer Network, Netcentric, and Internet of Things (IoT).

Endro Ariyanto. He received Bachelor of Engineering from Department of Electrical Engineering Gadjah Mada University, Yogyakarta, Indonesia in 1994 and Master of Informatics Engineering from Institute of Technology Bandung, Indonesia in 2000. His research interests include Advanced Computer System and Real Time Systems.

Muhammad A. Makky. He received Bachelor of Computer Science from Gadjah Mada University, Yogyakarta, Indonesia in 2009 and Master of Informatics Engineering from Institute of Technology Bandung, Indonesia in 2012. His research interests include Advanced IT Enterprise Architecture, IT Governance, Service Computing, Service Oriented Architecture.