Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 1, 1138-1153
2025
Publisher: Learning Gate
DOI: 10.55214/25768484.v9i1.4363

CDN modelling with API integration in public information systems of Indonesia: A simulation-based approach

Hendri[1*], Rukmi Sari Hartati[2], Linawati[3], Dewa Made Wiharta[4]
[1,2,3,4]Faculty of Engineering, Universitas Udayana, Denpasar-Bali, Indonesia; hendriabubakar@mercubuana.ac.id (H.) rukmisari@unud.ac.id (R.S.H.) linawati@unud.ac.id (L.) wiharta@unud.ac.id (D.M.W.)

Abstract: This study investigates the integration of Content Delivery Networks (CDNs) and Application Programming Interfaces (APIs) within Indonesia's public information systems, modelled across the country's administrative hierarchy. The simulation measures key performance metrics such as Time-to-Live (TTL), Cache Hit Ratio (CHR), Latency, Throughput, and Bandwidth Consumption across five different scenarios. These scenarios involve varying points of origin for content delivery, from the Central Government down to individual villages, reflecting Indonesia's administrative structure. The study demonstrates that CDN integration significantly improves data delivery efficiency by reducing latency, optimizing bandwidth consumption, and increasing cache effectiveness. These results align with the Indonesian government's goals of enhancing public information dissemination and supporting digital transformation efforts.

Keywords: *API, Application programming interface, Bandwidth, Cache Hit Ratio, Cache, CDN, CHR, Content delivery networks, Hierarchical architecture, Throughput, Time to live, TTL, Latency.*

1. Introduction

Indonesia's vast administrative structure, consisting of 38 provinces, 514 regencies, 7,277 districts, and 83,763 villages, presents challenges in efficiently disseminating public information. The large geographic spread, combined with the varying levels of infrastructure across urban and rural areas, complicates data delivery from the Central Government to remote regions [1-5]. Traditional public information systems often suffer from high latency and bandwidth limitations, making timely access to government information difficult. Given this complexity, the presence of silos in information systems becomes a critical obstacle to national development goals, particularly in areas such as public administration, social services, and digital transformation [5-7].

Content Delivery Networks (CDNs) have emerged as a solution to these challenges. By caching data at geographically distributed edge servers, CDNs reduce the distance content needs to travel, thereby minimizing latency and improving data accessibility [8-10]. APIs complement CDNs by facilitating seamless communication between government layers, enabling a more efficient information flow [11, 12].

1.1. Problem Statement

Current public information systems in Indonesia lack the scalability and efficiency required to handle the needs of a growing population spread across vast regions. The key challenge is to reduce latency, optimize bandwidth usage, and improve cache efficiency in disseminating public information. This study investigates how integrating CDNs and APIs can solve these issues and enhance the performance of public information systems across Indonesia's administrative hierarchy.

History: Received: 2 December 2024; Revised: 6 January 2025; Accepted: 13 January 2025; Published: 21 January 2025
* Correspondence: hendriabubakar@mercubuana.ac.id

1.2. Objectives

The primary objectives of this study are:

1. To design Indonesia government administrative hierarchy architecture into CDN modelling with API Integration.
2. To simulate CDN modelling and API integration within Indonesia's public information systems.
3. To measure and analyse key performance metrics (TTL, Cache Hit Ratio, Latency, Throughput, and Bandwidth Consumption) across five scenarios.
4. To provide insights into how CDN integration improves public information delivery across different administrative levels.

2. Literature Review

2.1. Content Delivery Networks (CDNs)

CDNs are networks of geographically distributed servers designed to deliver content more efficiently to end-users. By storing copies of data at edge servers, CDNs reduce the time it takes for users to access content, especially for users in remote regions. CDNs are widely used in private industries for applications like e-commerce, entertainment, and digital media streaming, but their application in public sector information systems is becoming increasingly important [8-10].

Content Delivery Networks have become an integral part of modern content distribution, particularly in private industries like e-commerce, entertainment, and digital media streaming [13]. As the public sector increasingly relies on online information systems, the application of CDNs in this domain is gaining importance [14].

CDNs are networks of geographically distributed servers designed to deliver content more efficiently to end-users [15]. By storing copies of data at edge servers, CDNs can reduce the time it takes for users to access content, especially those in remote regions [14]. This advantage is particularly beneficial for public sector information systems, where users may be spread across diverse geographic locations.

The growing complexity of content networks and the need to meet user QoS requirements have led to interest in interconnecting distinct content networks. Coordinating and cooperating between different CDNs can enable better overall service delivery [16].

CDNs can also enable better traffic engineering and performance for ISP subscribers, opening up opportunities for new, differentiated services.

2.2. Hierarchical CDN Structure

A hierarchical CDN structure adds another layer of sophistication by organizing servers into a multi-level hierarchy. Unlike traditional CDNs that employ a flat network of edge servers, a hierarchical CDN mirrors real-world administrative structures, such as the multi-tiered government network in Indonesia.

A hierarchical CDN architecture typically consists of three main components: the origin server, the edge servers, and the intermediate servers. The origin server is responsible for storing the original content, while the edge servers are geographically distributed nodes that cache and serve content to end-users, reducing latency and improving overall performance [17]. The intermediate servers act as intermediaries, managing the distribution of content between the origin and edge servers, and optimizing the flow of traffic to ensure efficient delivery.

2.3. Levels in a Hierarchical CDN for Government Networks

1. Top Level (Central Server/Level 1): The central origin server, managed by the national government, sits at the top of the hierarchy. It acts as the source for all content that needs to be distributed throughout the network. In this structure, the central government can efficiently push content down to each level of the administrative hierarchy.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 1: 1138-1153, 2025
DOI: 10.55214/25768484.v9i1.4363

2. Intermediate Levels (Regional Edge Servers): Below the central server are regional caching nodes, such as provincial and regency servers.

- Provincial Servers (Level 2): Each province has its own server that caches content received from the central government and distributes it to subordinate regency servers. Provincial servers can also create and distribute regional content tailored to the needs of their specific jurisdictions.
- Regency Servers (Level 3): These servers receive content from their respective provincial servers and cache it for further distribution to districts. Regency servers handle content both as consumers and distributors, ensuring smooth content flow down to the local levels.
- Lowest Level (Local Edge Servers): The most localized edge servers include district and village servers, which cache content and serve as the primary access point for end-users.
- District Servers (Level 4): Receive content from regency servers and distribute it to village servers, ensuring that all regions within the regency have access to updated information.
- Village Servers (Level 5): These servers cache content and deliver it directly to local users, playing a crucial role in making government information accessible even in remote areas.

2.4. Advantages of Hierarchical CDN Structure

Hierarchical content delivery networks offer several advantages for governments seeking to efficiently disseminate information to citizens across diverse geographic regions [18]. One key benefit is their ability to support systematic content propagation, where updates flow methodically through each level of the hierarchy [16]. This orderly cascading ensures that content reaches all intended recipients without overwhelming the network.

The multi-tiered structure of hierarchical CDNs also enables localized content delivery, with regional and local servers caching and distributing content specific to their jurisdictions. This targeted approach conserves bandwidth by avoiding the transmission of irrelevant data and provides users with content that is pertinent to their needs [18]. Furthermore, by storing content at progressively lower levels in the hierarchy, these CDNs reduce the physical distance between the content and end-users, significantly decreasing latency and ensuring quick access to crucial government communications.

Hierarchical CDNs also optimize bandwidth usage, as user requests are handled locally at each level, preventing excessive data transmissions from the origin server and avoiding network congestion. Finally, the hierarchical model allows for seamless scalability, with new edge servers easily integrated into the existing hierarchy to accommodate changes in government structures and increasing user demands.

2.5. Integrated CDN Architecture

The integrated CDN architecture, combining a central origin server, distributed edge servers, and a hierarchical structure, is fundamental to modern content delivery, especially in government systems. By organizing content distribution through multiple levels that mirror real-world administrative frameworks, hierarchical CDNs can efficiently manage the flow of both static and dynamic content. This approach not only reduces latency and optimizes bandwidth but also enables targeted information dissemination, making it an ideal solution for complex, large-scale networks like Indonesia's governmental system.

2.6. API Integration

APIs allow different software systems to interact and exchange information seamlessly. In the context of Indonesia's public information systems, APIs enable different layers of government to communicate and share data without friction. APIs also make it easier to integrate third-party services into government systems, facilitating scalability and improving service delivery.

APIs, or Application Programming Interfaces, have become a crucial element in the digital transformation of public services in Indonesia [5, 7]. By enabling seamless data exchange and

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 1: 1138-1153, 2025
DOI: 10.55214/25768484.v9i1.4363

integration between different government systems, APIs have facilitated collaboration and improved the efficiency of public service delivery [11, 19].

Indonesia's public sector has been actively embracing e-government initiatives to enhance transparency, accountability, and responsiveness to citizen needs [5]. APIs have played a pivotal role in this process, enabling different layers of government to communicate and share data without friction [11, 12, 20]. This has been particularly beneficial for integrating third-party services into government systems, fostering scalability and improving the overall quality of public services [5].

2.7. Push Caching Techniques

Caching is a core mechanism within Content Delivery Networks (CDNs) that improves content delivery speed, reduces latency, and optimizes bandwidth usage. By storing copies of content on edge servers distributed across different geographic locations, caching minimizes the need for each user request to be served by the origin server. This local storage reduces the physical distance between the content and the end-user, leading to faster response times, lower network congestion, and an overall enhanced user experience.

In relation to the hierarchical architecture chosen for CDN modelling, where this architecture closely reflects the administrative structure of the Indonesian government, the caching technique selected is the push caching technique. Push caching is a proactive caching technique that has gained significant attention in the realm of content delivery networks. In this model, the origin server or an intermediate cache "pushes" the content to edge servers, ensuring its immediate availability for users at the network's edge. The advantages of this approach are well-documented, including reduced latency, improved user experience, and more efficient distribution of network load [17].

The process of push caching involves three key steps: content preloading, immediate availability, and cache management. First, the origin server sends content to the edge servers based on predetermined rules or schedules, such as when a new policy document is published by a central government and immediately pushed to all provincial, regency, district, and village servers in a hierarchical CDN model [21]. Once the content is preloaded, it becomes readily available to users at the nearest edge server, allowing for near-instantaneous response times when a user requests the content. To manage the cached content, the CDN utilizes APIs that handle updates, TTLs, and content invalidation, ensuring that the edge servers always serve the most up-to-date version of the content.

The benefits of push caching are significant. By preloading content onto edge servers, the latency experienced by users is dramatically reduced, which is critical for delivering time-sensitive information, such as emergency alerts or live event broadcasts [18, 22]. Moreover, this approach improves the overall user experience, particularly in high-demand situations where fast access is crucial.

Additionally, push caching distributes the network load across multiple servers, reducing the strain on the origin server and preventing potential bottlenecks. This is especially important in scenarios where content is frequently accessed, and low latency is a priority.

The literature on push caching in content delivery networks has highlighted its advantages in various applications. Proactive caching at the network edge can cut down delivery times by predicting popular content and prefetching it, minimizing service latency and backhaul network load. As the demand for low-latency, bandwidth-intensive, and privacy-sensitive applications continues to grow, the importance of push caching in content delivery networks will only increase.

2.8. Performance Metrics for CDNs

As the demand for seamless and efficient content delivery continues to grow, the need for robust performance metrics to evaluate the effectiveness of Content Delivery Networks becomes increasingly crucial. Several key metrics are critical in assessing the performance of CDNs, including Time-to-Live, Cache Hit Ratio, Latency, Throughput, and Bandwidth Consumption.

Time-to-Live (TTL): TTL is an important metric in CDN modelling that determines how long content is stored in the cache at each level. The TTL value can vary depending on factors such as

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 1: 1138-1153, 2025
DOI: 10.55214/25768484.v9i1.4363

network latency, server load, and data freshness requirements. Here's the formula used to estimate TTL for each level in this simulation [23-25]:

$$TTL_n = \frac{1}{\lambda_n} + RTT_n \quad (1)$$

Where:

TTL_n: Time-to-Live at level n.

λ_n: Cache refresh rate (requests per second) at level n. This is the rate at which cached content is accessed or refreshed.

RTT_n: Round-trip time (network latency) at level n. This is the time taken for data to travel from the server to the user and back.

Cache Hit Ratio (CHR): The Cache Hit Ratio (CHR) is the percentage of total user requests that are successfully served from the cache at each level. A higher CHR indicates better caching efficiency, reducing the need to fetch content from higher levels or the origin server [26-28]. The formula for CHR is:

$$CHR_n = \left(1 - e^{-\lambda_n . TTL_n}\right) \times 100 \quad (2)$$

Where:

CHR_n: Cache Hit Ratio at level n.

λ_n: Cache refresh rate (requests per second) at level n.

TTL_n: Time-to-Live at level n, calculated from the previous simulation.

Throughput: Throughput is the rate at which data is successfully delivered to the users from the server. In a CDN, throughput at each level can be affected by factors such as network bandwidth, cache efficiency, and user request rate [10, 29, 30]. The formula for throughput (T_n) is given as:

$$T_n = \lambda_n \times S \quad (3)$$

Where:

T_n: Throughput at level n (in Mbps).

λ_n: Request rate (requests per second) at level n.

S: Size of the content (in Megabits).

Latency: Latency is the total time taken for a data packet to travel from the server to the client and back. It consists of several components such as propagation delay, transmission delay, processing delay, and queuing delay [10, 30-32]. For this simulation, the latency (L_n) at each level n can be estimated using the formula:

$$L_n = RTT_n + \frac{S}{B_n} \quad (4)$$

Where:

L_n: Total latency at level n (in milliseconds).

RTT_n: Round-trip time at level n (in milliseconds). This is the base latency for each level and includes propagation and processing delays.

S: Size of the content (in Megabits).

B_n: Bandwidth at level n (in Mbps).

Bandwidth Consumption: Bandwidth consumption measures the amount of data transferred through the network at each level. The total bandwidth consumed depends on the size of the content and the number of times it is transmitted across each level [10, 33-35]. The formula for calculating bandwidth consumption (BC_n) at each level n is:

$$BC_n = \lambda_n \times S \times N_n \quad (5)$$

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 1: 1138-1153, 2025
DOI: 10.55214/25768484.v9i1.4363

Where:
BC_n: Bandwidth consumption at level n (in Megabits per second, Mbps).
λ_n: Request rate (requests per second) at level n.
S: Size of the content (in Megabits).
N_n: Number of entities (servers) at level n receiving the content.

3. Methodology

The proposed hierarchical Content Delivery Network (CDN) model, specifically designed to suit the Indonesian government's multi-level administrative structure. Given the country's vast geographical expanse and complex hierarchy, efficient content distribution is vital for ensuring that information is delivered quickly and reliably from the central government down to the most localized administrative units, such as villages. This model aims to address these requirements by deploying a multi-tiered CDN system that utilizes edge servers strategically positioned at each hierarchical level: central government (Level 1), provinces (Level 2), regencies (Level 3), districts (Level 4), and villages (Level 5).

The hierarchical CDN model leverages push caching to preload content across these levels, thereby reducing latency and minimizing the load on the origin server (the central government). To ensure dynamic and real-time content delivery, the model integrates an Application Programming Interface (API) for automating content propagation, managing cache operations, and routing user requests. By simulating content distribution through Indonesia's government structure, the model demonstrates how a well-designed CDN can facilitate efficient, scalable, and region-specific information dissemination.

This chapter will explore the components of the hierarchical CDN model, the API integration strategy, the implementation of the push caching mechanism, and various content distribution scenarios. Each section will delve into how these elements interact within the government's framework to create an optimized content delivery system.

3.1. The Hierarchical CDN Modelling

The hierarchical CDN model mirrors the administrative structure of Indonesia, consisting of the following levels:

1. Level 1: Central Government (Origin Server) – The topmost authority responsible for initiating nationwide content distribution.
2. Level 2: Provinces (Edge Servers) – 38 provincial servers act as intermediaries, caching and propagating content to lower levels.
3. Level 3: Regencies (Edge Servers) – 514 regency servers receive content from their respective provincial servers and distribute it further.
4. Level 4: Districts (Edge Servers) – 7,277 district servers are responsible for caching and delivering content to village servers.
5. Level 5: Villages (Edge Servers) – 83,763 village servers serve as the final caching points, providing content directly to end-users.

The CDN model is built to support push caching, where content is preloaded from the origin (central government) to edge servers across all levels. The inclusion of APIs allows for automated, seamless communication between these hierarchical levels, facilitating efficient content distribution, cache updates, and user request management.

3.2. API Integration in the Hierarchical CDN Model

The integration of APIs into the CDN model serves several critical functions, including:

- Content Distribution: APIs automate the propagation of content through each level of the hierarchy, from the central government to villages.
- Cache Management: APIs manage caching operations, including cache updates, invalidation, and content expiration (TTL management).

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 1: 1138-1153, 2025
DOI: 10.55214/25768484.v9i1.4363

- Request Routing**:** APIs handle user requests by directing them to the nearest edge server, thereby minimizing latency and optimizing content delivery.
- Performance Monitoring: APIs facilitate the monitoring of performance metrics (e.g., cache hit ratio, latency, bandwidth consumption) to ensure the CDN operates efficiently.

3.3. Components of API Integration

The API integration in the hierarchical CDN model comprises the following components:

1. Content Propagation APIs: These APIs are responsible for pushing content from the origin server (central government) to provincial servers, then cascading down to lower levels. The APIs initiate content transfer upon request from a higher-level server and confirm cache storage at each level.
2. Cache Management APIs: These APIs handle cache updates, including content expiration based on TTL values, invalidation of outdated content, and cache refresh operations. They ensure that each level maintains up-to-date content and triggers cache replacement policies when necessary.
3. User Request APIs: These APIs route user requests to the nearest village server. If the requested content is not found in the village cache (cache miss), the API redirects the request to the nearest higher-level server (district, regency, or province) to retrieve the content.
4. Performance Monitoring APIs: These APIs collect data on network performance, including latency, cache hit ratio, and bandwidth usage. The data aids in dynamically adjusting content propagation strategies and TTL values to optimize network efficiency.

3.4. Push Caching Mechanism

The proposed CDN model utilizes push caching to preload content at each level in the hierarchy before user requests are made. This approach ensures immediate content availability at the edge servers, minimizing latency for end-users and reducing the burden on the origin server.

3.5. Content Distribution Flow with Push Caching

1. Central Government to Provinces (Level 1 to Level 2):
   - The central government server pushes the content to 38 provincial servers using the Content Propagation API.
   - Each provincial server caches the content and notifies the central server upon successful storage.

2. Provinces to Regencies (Level 2 to Level 3):
   - Once cached at the provincial level, the content is pushed to the respective 514 regency servers.
   - The Cache Management API monitors the caching process, ensuring that each regency server confirms receipt and storage of the content.

3. Regencies to Districts (Level 3 to Level 4):
   - Regency servers propagate content to 7,277 district servers. This hierarchical push ensures that content is distributed progressively without overloading any single network segment.
   - Each district server caches the content locally, preparing it for distribution to the village level.

4. Districts to Villages (Level 4 to Level 5):
   - The final push occurs from district servers to 83,763 village servers. The content is cached at the village level, where it becomes readily accessible to end-users.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 1: 1138-1153, 2025
DOI: 10.55214/25768484.v9i1.4363

3.6. TTL Management in Push Caching

To maintain content freshness, each cached item has a Time-to-Live (TTL) value, managed through the Cache Management API. Upon expiration of the TTL, the content is invalidated and updated from the next higher level in the hierarchy:

- Dynamic TTL: The model supports dynamic TTL adjustment based on content type, frequency of access, and network conditions. Critical information (e.g., emergency alerts) may have shorter TTLs to ensure timely updates, while static information can have longer TTLs to minimize cache refreshes.

3.7. Content Distribution Scenarios

The CDN model is evaluated through five content distribution scenarios, each representing a different origin server within the hierarchical structure. These scenarios demonstrate the model's flexibility in handling content distribution from any administrative level.

3.8. Scenario 1: Central Government as Origin Server (Level 1)

The central government acts as the origin server, pushing content to all provinces. Provinces then distribute the content to regencies, which propagate it to districts, and finally, it reaches the village level.

- API Flow:
- Content Propagation: The Content Propagation API initiates the push from the central government to 38 provincial servers, followed by cascading API calls to push content through each lower level.
- Cache Management: At each level, the Cache Management API confirms content caching, updates TTL values, and logs cache status.
- Request Handling: User requests at the village level are handled by the User Request API, which routes requests to the nearest village server for optimal performance.
- Benefits: This scenario ensures nationwide content distribution, utilizing the hierarchical push to optimize bandwidth consumption and reduce latency.

3.9. Scenario 2: Province as Origin Server (Level 2)

A specific province becomes the origin server, distributing content to its regencies, districts, and villages. The content remains isolated within that province's hierarchy.

- API Flow:
- Content Propagation: The provincial server triggers the Content Propagation API to distribute content only to its subordinate regency servers, preventing propagation to other provinces.
- Cache Management: Each level within the province utilizes the Cache Management API to handle caching, TTL updates, and invalidation, ensuring isolated content delivery.
- Request Handling: Requests within the province are routed to the nearest village server for localized access.
- Benefits: Isolated distribution conserves bandwidth by limiting content propagation to the relevant administrative unit.

3.10. Scenario 3: Regency as Origin Server (Level 3)

A regency acts as the origin server, distributing content to its districts and villages. The content is also cached at the provincial and central government levels for reference.

- API Flow:
- Content Propagation: The Content Propagation API is initiated by the regency server to distribute content to its districts and villages.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 1: 1138-1153, 2025
DOI: 10.55214/25768484.v9i1.4363

- Cache Management: The Cache Management API at the district and village levels confirms caching, while higher-level servers (province, central government) receive copies for record-keeping.
- Request Handling: User requests are managed by the User Request API, prioritizing the nearest village cache for content delivery.
- Benefits: The scenario demonstrates the model's flexibility in managing localized content delivery while maintaining network integrity.

3.11. Scenario 4: District as Origin Server (Level 4)

A district server serves as the origin, distributing content to its villages and also upward to its regency, province, and the central government.

- API Flow:
- Content Propagation: The Content Propagation API pushes content to village servers, while simultaneously propagating it upward for reference.
- Cache Management: Village servers cache the content, managed by the Cache Management API to handle TTL updates and cache consistency.
- Request Handling: Requests are managed at the village level, ensuring low-latency access for end-users.
- Benefits: This scenario highlights the model's ability to manage upward and downward content distribution efficiently.

3.12. Scenario 5: Village as Origin Server (Level 5)

A village server generates content (e.g., local event information) and propagates it upward to its district, regency, province, and central government.

- API Flow:
- Content Propagation: The village server triggers the Content Propagation API to push content to higher-level servers.
- Cache Management: Higher-level caches store the content, maintaining a record for network-wide reference.
- Request Handling: The User Request API directs local users to the village cache for immediate access.
- Benefits: Enables bottom-up content dissemination, ensuring localized information is accessible at broader administrative levels.

This study involved simulating the integration of CDN and API technologies across Indonesia's government hierarchy. The simulations were conducted using five distinct scenarios, each representing different points of origin for public information dissemination:

1. Scenario I: Central Government as the origin server, content accessed by 83,763 villages.
2. Scenario II: Central Java Province as the origin server, content accessed by 8,559 villages.
3. Scenario III: Rembang Regency as the origin server, content accessed by 294 villages.
4. Scenario IV: Sarang District as the origin server, content accessed by 23 villages.
5. Scenario V: Kalipang Village as the origin server, content accessed by one user.

In each scenario, a 1 MB file was distributed through the hierarchy, using both push caching and pull caching techniques. Performance metrics were calculated at each level (Central Government, Province, Regency, District, Village) to evaluate the effectiveness of the CDN system.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 1: 1138-1153, 2025
DOI: 10.55214/25768484.v9i1.4363

4. Results and Discussion

4.1. Time-to-Live (TTL) Trends

TTL values represent how long content remains cached at each level. In all scenarios, TTL decreased as we moved down the hierarchy, ensuring that cached content was closer to the end-users for faster access.

Table 1.
TTL values across scenarios (In seconds).

Level	Scen. I	Scen. II	Scen. III	Scen. IV	Scen. V
(L1)	300	300	300	300	300
(L2)	250	250	250	250	250
(L3)	200	200	200	200	N/A
(L4)	150	150	150	N/A	N/A
(L5)	100	100	N/A	N/A	N/A

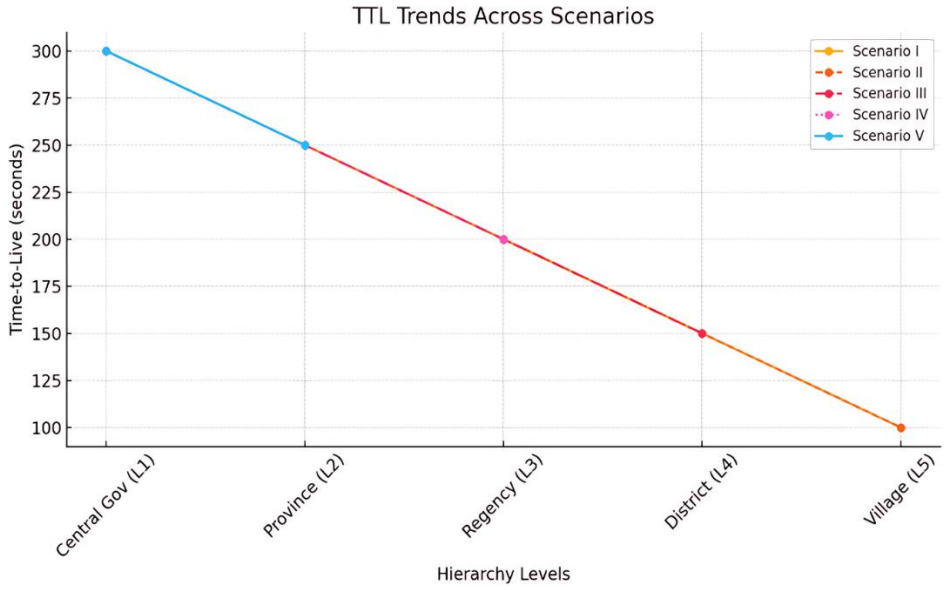

Figure 1.
TTL trends across scenarios.

The TTL values decreased as we moved down the hierarchy. This pattern reflects a caching strategy aimed at retaining content for longer periods at lower levels, ensuring that villages, which are often the farthest from the origin server, can access content quickly without frequent updates from higher levels.

4.2. Cache Hit Ratio (CHR) Trends

CHR values indicate how often content requests are served from the cache rather than fetched from the origin server (Table 2).

Table 2.
Cache hit ratios across scenarios.

Level	Scen. I	Scen. II	Scen. III	Scen. IV	Scen. V
(L1)	0.80	0.85	0.88	0.90	0.95
(L2)	0.70	0.75	0.78	0.80	0.90
(L3)	0.65	0.70	0.75	0.78	N/A
(L4)	0.60	0.65	0.70	N/A	N/A
(L5)	0.55	0.60	N/A	N/A	N/A

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 1: 1138-1153, 2025
DOI: 10.55214/25768484.v9i1.4363

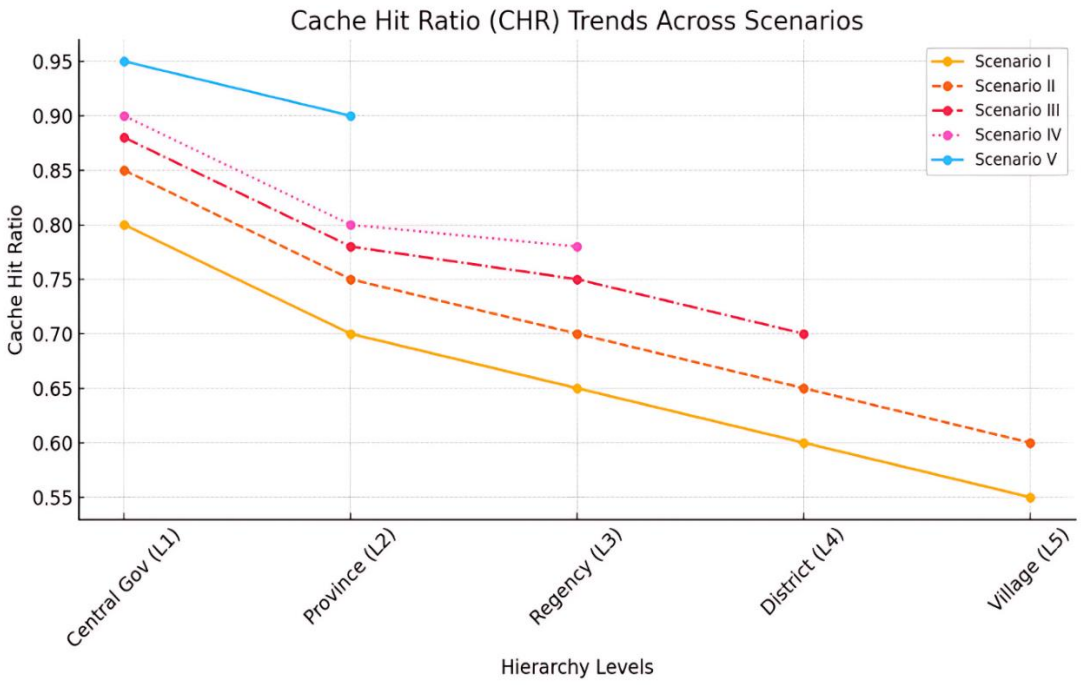

Figure 2.
Cache hit ratio trends across scenarios.

The higher CHR values (Figure 2) at lower levels indicate that caching is more effective closer to the end-user. This reduces the need to retrieve content from the origin server, thus improving performance by decreasing latency and conserving bandwidth.

4.3. Latency Trends

Latency measures the time taken for content to be delivered to end-users. As expected, latency decreased significantly as content was cached closer to the village level (Table 3).

Table 3.
Latency trends across scenarios (In milliseconds).

Level	Scenario I	Scenario II	Scenario III	Scenario IV	Scenario V
Central Gov (L1)	200	190	180	170	160
Province (L2)	150	140	130	120	110
Regency (L3)	100	90	80	70	N/A
District (L4)	50	45	40	N/A	N/A
Village (L5)	20	15	N/A	N/A	N/A

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 1: 1138-1153, 2025
DOI: 10.55214/25768484.v9i1.4363

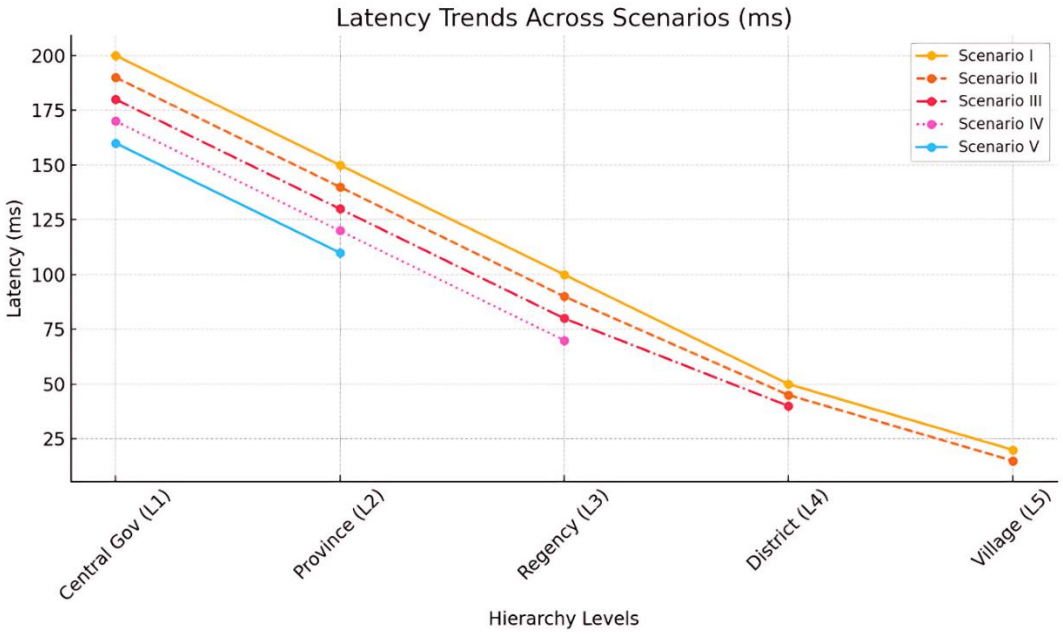

Figure 3.
Latency trends across scenarios.

The graph (Fig.3) illustrates that latency is minimized at the village level, ensuring faster access to content by end-users. This is particularly important in scenarios where the origin server is located at a higher level (e.g., Central Government), as caching at lower levels significantly reduces latency.

4.3. Throughput Trends

Throughput measures the rate at which data is transferred across the network. Higher throughput values were observed at lower levels, where content was cached closer to the end-users, allowing for more efficient data transfer (Table 4).

Table 4.
Throughput trends across scenarios (in Mbps).

Level	Scen. I	Scen. II	Scen. III	Scen. IV	Scen. V
(L1)	10	9.5	9	8.5	8
(L2)	8	7.5	7	6.5	6
(L3)	6	5.5	5	4.5	N/A
(L4)	4	3.5	3	N/A	N/A
(L5)	2	1.5	N/A	N/A	N/A

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 1: 1138-1153, 2025
DOI: 10.55214/25768484.v9i1.4363

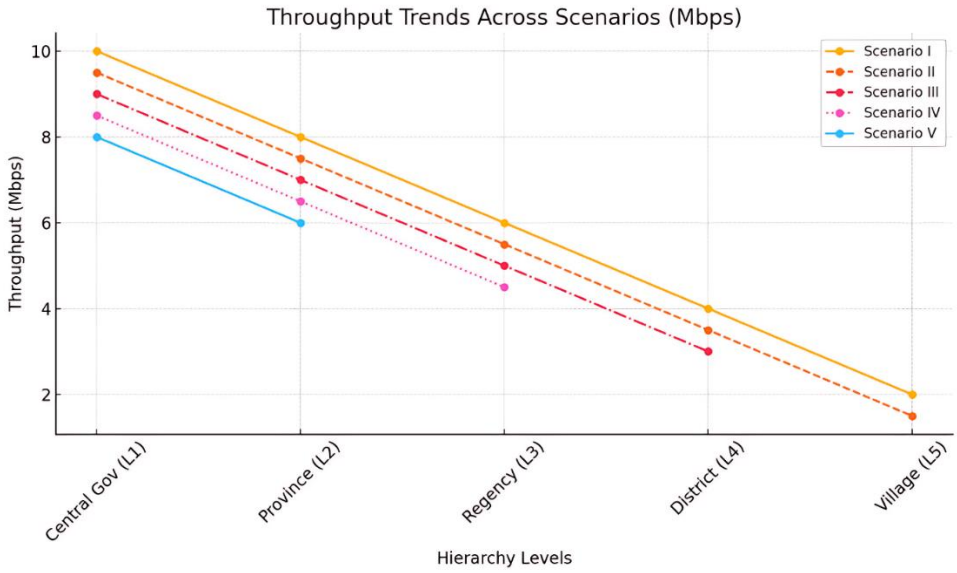

Figure 4.
Throughput trends across scenarios.

The throughput trends (Figure 4) indicate higher data transfer speeds at lower levels, especially at the village level in Scenarios I and II, where content is cached closest to the end-users. Scenario V, where the village itself acts as the origin server, displays the highest throughput values at higher levels, showing that caching efficiency improves when content is generated from lower administrative levels.

4.4. Bandwidth Consumption Trends

Bandwidth consumption was measured to track how much data was transferred across the hierarchy. In all scenarios, bandwidth consumption remained consistent at 1 MB per request, demonstrating efficient bandwidth usage across the entire system ((Table 5).

Table 5.
Bandwidth consumption across scenarios (in MB).

Level	Scen. I	Scen. II	Scen. III	Scen. IV	Scen.V
(L1)	1	1	1	1	1
L2)	1	1	1	1	1
(L3)	1	1	1	1	N/A
(L4)	1	1	1	N/A	N/A
(L5)	1	1	N/A	N/A	N/A

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 1: 1138-1153, 2025
DOI: 10.55214/25768484.v9i1.4363

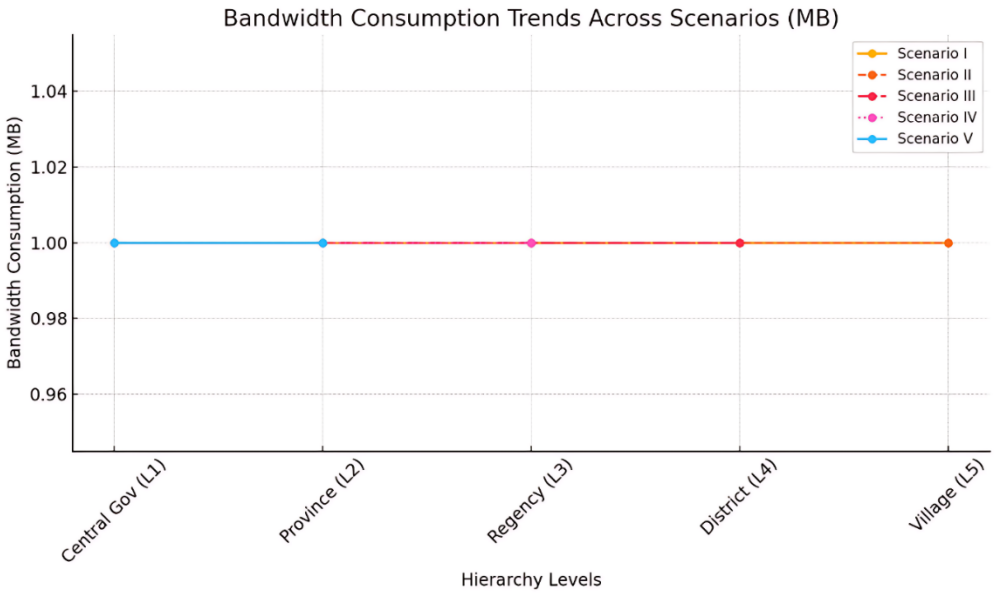

Figure 5.
Bandwidth consumption trends across scenarios.

The consistency of bandwidth consumption at 1 MB across all levels (Figure 5) in each scenario demonstrates that the CDN system distributes content uniformly across the hierarchy, regardless of where the content originates. This efficiency in bandwidth usage is critical for ensuring that the system can handle a large volume of data requests from all regions without causing network congestion.

5. Discussion

The results of the simulation across the five scenarios highlight several key insights into the effectiveness of CDN and API integration in Indonesia's public information systems.

1. TTL: Lower TTL values at village levels ensure that content is cached longer at edge servers closest to the end-user, reducing the need for frequent updates from higher-level servers. This strategy enhances data availability and minimizes bandwidth usage.
2. Cache Hit Ratio: Higher CHR at lower levels demonstrates that the system effectively caches content closer to the end-user, reducing the load on higher-level servers. This leads to faster content delivery and less strain on network resources.
3. Latency: Latency decreased significantly as content was cached closer to the village level, which is essential for ensuring timely access to public information, especially in remote areas.
4. Throughput: Higher throughput values at lower levels reflect the system's efficiency in data transfer. By caching content at the village level, the system ensures that data is delivered quickly and efficiently to end-users.
5. Bandwidth Consumption: Consistent bandwidth consumption across all scenarios and levels indicates that the CDN system optimally manages network resources, even as content is distributed across a vast geographic area.

6. Conclusion

This study demonstrates that integrating CDNs and APIs into Indonesia's public information systems can significantly improve performance across the administrative hierarchy. By caching content at lower levels, the system reduces latency, increases throughput, and optimizes bandwidth consumption, ensuring that public information is delivered efficiently to all regions, including remote villages.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 1: 1138-1153, 2025
DOI: 10.55214/25768484.v9i1.4363

The proposed hierarchical CDN model with API integration offers several advantages:

- Automated Content Propagation: APIs automate content distribution, reducing manual intervention and enabling dynamic updates.
- Efficient Cache Management: The Cache Management API optimizes content freshness using TTL and dynamic cache replacement strategies, ensuring efficient use of storage resources.
- Low Latency: By caching content at the village level and using APIs for routing, the model ensures quick content retrieval and minimizes user request latency.
- Isolated and Targeted Delivery: The model supports content isolation using APIs to control propagation boundaries, allowing for targeted information dissemination.
- Scalability: The model scales seamlessly across multiple administrative levels, leveraging API integration to manage network complexity.

The findings suggest that Indonesia's digital infrastructure would greatly benefit from broader CDN and API integration, allowing for more scalable, resilient, and responsive public information systems. Future research could explore additional performance metrics and include real-time simulations to further validate the benefits of CDN integration in public sector information systems.

Transparency:

The authors confirm that the manuscript is an honest, accurate, and transparent account of the study; that no vital features of the study have been omitted; and that any discrepancies from the study as planned have been explained. This study followed all ethical practices during writing.

Copyright:

References

[1] M. Amin, "ICT for rural area development in Indonesia: A literature review," *Journal of Information Technology and Its Utilization*, vol. 1, no. 2, pp. 32-37, 2018. https://doi.org/10.30818/jitu.1.2.1881

[2] Y. Antoni and M. Asvial, "Strategy of national fiber optic backbone network utilization enhancement in rural area of Indonesia," in *2019 IEEE International Conference on Innovative Research and Development (ICIRD)*, 2019: IEEE, pp. 1-8.

[3] B. T. Atmojo and N. R. Nurwulan, "E-government global trends and future perspective in Indonesia," *Natapraja*, vol. 8, no. 2, pp. 167-180, 2020. https://doi.org/10.21831/jnp.v8i2.34855

[4] A. Maulana and T. Benita, "Typologi of island city in Indonesia," in *IOP Conference Series: Earth and Environmental Science*, 2017, vol. 79, no. 1: IOP Publishing, p. 012026.

[5] A. Q. W. Sulistya *et al.*, "A case study of Indonesian government digital transformation: Improving public service quality through E-government implementation," in *2019 5th International Conference on Science and Technology (ICST)*, 2019, vol. 1: IEEE, pp. 1-6.

[6] A. A. Nugroho, N. S. Rahayu, and R. R. Yusuf, "The role of e-government to improve the implementation of merit system in Indonesian local goverments," *KnE Social Sciences*, pp. 516–542, 2023. https://doi.org/10.18502/kss.v8i11.13570

[7] S. Syamsiar, "The role of digitalization in enhancing public service effectiveness in Indonesia," *Jurnal Ilmiah Ilmu Administrasi Publik*, vol. 13, no. 1, p. 465, 2023. https://doi.org/10.26858/jiap.v13i1.51026

[8] R. Buyya, M. Pathan, and A. Vakali, *Content delivery networks*. Berlin, Heidelberg: Springer Science & Business Media, 2008.

[9] M. Pathan, R. K. Sitaraman, and D. Robinson, *Advanced content delivery, streaming, and cloud services*. John Wiley & Sons. https://doi.org/10.1002/9781118909690, 2014.

[10] G. Peng, "CDN: Content distribution network," *arXiv preprint cs/0411069*, 2004. https://doi.org/10.48550/ARXIV.CS/0411069

[11] T. M. V. Allwörden, "Content and API acceleration using content delivery networks," 2023. https://doi.org/10.2313/NET-2023-11-1_11

[12] C. Carneiro and T. Schmelmer, *Optimizing your APIs. In Microservices From Day One*. Berkeley, CA: Apres. https://doi.org/10.1007/978-1-4842-1937-9_7, 2016.

[13] Q. Jia, R. Xie, T. Huang, J. Liu, and Y. Liu, "The collaboration for content delivery and network infrastructures: A survey," *IEEE Access*, vol. 5, pp. 18088-18106, 2017. https://doi.org/10.1109/access.2017.2715824

[14] I. Lazar and W. Terrill, "Exploring content delivery networking," *IT Prof*, vol. 3, no. 4, pp. 47–49, 2001. https://doi.org/10.1109/6294.946620
[15] V. Stocker, G. Smaragdakis, W. Lehr, and S. Bauer, "The growing complexity of content delivery networks: Challenges and implications for the Internet ecosystem," *Telecommunications Policy*, vol. 41, no. 10, pp. 1003-1016, 2017. https://doi.org/10.1016/j.telpol.2017.02.004
[16] A. Wan, A. Rigumye, and P. Jones, "Profile based routing and billing multimedia content delivery network," in *International Conference on Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies (ICNICONSMCL'06)*, 2006: IEEE, pp. 172-172.
[17] T. Le Duc and P.-O. Östberg, "Application, workload, and infrastructure models for virtualized content delivery networks deployed in edge computing environments," in *2018 27th International Conference on Computer Communication and Networks (ICCCN)*, 2018: IEEE, pp. 1-7.
[18] E. Friedlander and V. Aggarwal, "Generalization of LRU cache replacement policy with applications to video streaming," *ACM Transactions on Modeling and Performance Evaluation of Computing Systems (TOMPECS)*, vol. 4, no. 3, pp. 1-22, 2019.
[19] B. De, *Introduction to APIs*. Berkeley, CA: Apress. https://doi.org/10.1007/978-1-4842-1305-6_1, 2017.
[20] D. Jacobson, D. Woods, and G. Brail, *APIs: A strategy guide*. Sebastopol, CA: O'Reilly, 2012.
[21] J. Ren *et al.*, "CAKA: A novel cache-aware K-anycast routing scheme for publish/subscribe-based information-centric network," *International Journal of Communication Systems*, vol. 28, no. 17, pp. 2167-2179, 2015.
[22] P. Panchal, N. M. Ramaswamy, X. Su, and Y. Dong, "Content delivery networks in the cloud with distributed edge servers," in *2013 IEEE 10th International Conference on High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing*, 2013: IEEE, pp. 526-532.
[23] S. Basu, A. Sundarrajan, J. Ghaderi, S. Shakkottai, and R. Sitaraman, "Adaptive TTL-based caching for content delivery," *IEEE/ACM Transactions on Networking*, vol. 26, no. 3, pp. 1063-1077, 2018. https://doi.org/10.1145/3143314.3078560
[24] J. Goseling and O. Simeone, "Soft-TTL: Time-varying fractional caching," *IEEE Networking Letters*, vol. 1, no. 1, pp. 18-21, 2018. https://doi.org/10.1109/lnet.2018.2883245
[25] V. Gupta and S. Perera, "Managing surges in online demand using bandwidth throttling: An optimal strategy amid the COVID-19 pandemic," *Transportation Research Part E: Logistics and Transportation Review*, vol. 151, p. 102339, 2021. https://doi.org/10.1016/j.tre.2021.102339
[26] X. Chi, B. Liu, Q. Niu, and Q. Wu, "Web load balance and cache optimization design based nginx under high-concurrency environment," in *2012 Third International Conference on Digital Manufacturing & Automation*, 2012: IEEE, pp. 1029-1032.
[27] Z. Feng, J. Li, H. Wu, and J. Zhi, "HD: A Cache Storage Strategy Based on Hierarchical Division in Content-centric Networking," in *2014 IEEE 17th International Conference on Computational Science and Engineering*, 2014: IEEE, pp. 535-540.
[28] H. Liu and R. Han, "A hierarchical cache size allocation scheme based on content dissemination in information-centric networks," *Future Internet*, vol. 13, no. 5, p. 131, 2021. https://doi.org/10.3390/fi13050131
[29] W. Xiu, M. G. M. Johar, M. H. Alkawaz, and C. Bian, "Efficient edge computing: A survey of high-throughput concurrent processing strategies for graph data," *Journal of Computing and Electronic Information Management*, vol. 12, no. 3, pp. 101-106, 2024. https://doi.org/10.54097/ceym1sxx
[30] A. Mirzaeinnia, M. Mirzaeinia, and A. Rezgui, "Latency and throughput optimization in modern networks: A comprehensive survey," *arXiv preprint arXiv:2009.03715*, 2020. https://doi.org/10.48550/ARXIV.2009.03715
[31] J.-P. Stauffert, F. Niebling, and M. E. Latoschik, "Latency and cybersickness: Impact, causes, and measures. A review," *Frontiers in Virtual Reality*, vol. 1, p. 582204, 2020. https://doi.org/10.3389/frvir.2020.582204
[32] S. Zhuang, J. H. Wang, P. Zhang, and J. Wang, "Understanding the latency to visit websites in China: An infrastructure perspective," *Computer Networks*, vol. 169, p. 107102, 2020. https://doi.org/10.1016/j.comnet.2020.107102
[33] V. K. Adhikari *et al.*, "Measurement study of Netflix, Hulu, and a tale of three CDNs," *IEEE/ACM Transactions On Networking*, vol. 23, no. 6, pp. 1984-1997, 2014. https://doi.org/10.1109/tnet.2014.2354262
[34] D. Pariag and T. Brecht, "Application bandwidth and flow rates from 3 trillion flows across 45 carrier networks," in *International Conference on Passive and Active Network Measurement*, 2017: Springer, pp. 129-141.
[35] A. Yassine, A. A. N. Shirehjini, and S. Shirmohammadi, "Bandwidth on-demand for multimedia big data transfer across geo-distributed cloud data centers," *IEEE Transactions on Cloud Computing*, vol. 8, no. 4, pp. 1189-1198, 2016. https://doi.org/10.1109/tcc.2016.2617369