2016 5th IEEE International Conference on Cloud Networking

Enhancing Content Distribution through Information-Aware Mechanisms

Walid Benchaita
UPMC Sorbonne Universités
walid.benchaita@lip6.fr

Gioacchino Tangari
University College London
uceegta@ucl.ac.uk

Samir Ghamri-Doudane
Nokia Bell Labs France
samir.ghamri-doudane@nokia.com

Sébastien Tixeuil
UPMC Sorbonne Universités
sebastien.tixeuil@lip6.fr

***Abstract*—Densely-deployed Content Delivery Network (CDN) solutions are used today for delivering a significant fraction of the Internet traffic through replication mechanisms. However, these networks show technological limitations when dealing with the proliferation of rich media-enabled applications such as video streaming.**

This paper introduces a new approach to content delivery incorporating Information-Centric Networking principles without requiring any change in the underlying network. This solution improves content delivery performance and enables the implementation of cost efficient request routing strategies. We also develop an on-line algorithm based on Lyapunov optimization theory which allows to dynamically generate effective strategies in order to satisfy operator's objectives. By using real video streaming request traces, it has been shown that the global hit ratio can be increased up to 100% while delivering the content 30% faster compared to currently deployed content delivery schemes. Furthermore, the proposed approach offers stable behavior during peak traffic.

I. Introduction

Content Delivery Networks (*CDN*s) operate to enhance user experience by moving content closer to the client. Their dense presence across large-scale networks, including the last mile to end-users, allows them to lower delivery delays, reduce long-haul links traffic, and guarantee high availability. DNS resolutions represent the *de-facto* standard for assigning user traffic to one of the available replica (at cache servers), based on proximity metrics, server load conditions and traffic handling costs. However, the explosion of video traffic and the increase of users' expectations (high quality, no buffering and low start-up delay [1]) have been challenging this setup for over a decade. Peaks in traffic workloads can still overstretch the network and caching infrastructures, while the frequent mismatches — due to content-unaware redirections, and also to failures in locating the clients — between clients and replica strongly impact the users experience. Recent solutions have tackled these problems by improving the control on video traffic, *e.g.* through centralized optimization [2], or by addressing the inaccuracy of end-user mapping [3]. However, a common bottleneck for all these solutions lies in the limitations of DNS-based request-routing: *(i)* lack of reactivity (due to DNS architecture and TTLs), *(ii)* additional delays (name resolution and multiple redirections) and *(iii)* low flexibility, since new policies for resolving the client requests need to be reflected in third-party servers.

This issue has been recently challenged by the ICN (Information Centric Networking) paradigm. By naming information at the network layer, ICN proposals enable the deployment of in-network storage and novel routing schemes to facilitate the efficient delivery of contents to the users.[4] However, adapting the current network infrastructure to support one of the *fully-fledged* ICN designs would pose severe changes to today's routers and to the Internet protocol suite [5].

In this paper, we present a new request routing framework which incorporates ICN functions into CDN infrastructures without requiring any change to the underlying network. Our hybrid approach benefits from a faster, yet accurate, name-based routing scheme in which DNS redirections are bypassed, and leverages connectionless transport to avoid static connection bindings. This solution reduces CDN serving costs through more accurate request resolutions (up to the granularity of a single content name). Furthermore, it provides improved flexibility and reactivity, as it allows the CDN operator to disregard third-party services (*e.g.* DNS) when enforcing new request routing strategies. To demonstrate the effectiveness of the proposed framework, we test it using real video streaming traces from a North America CDN operator. The evaluation, performed through a comparison with the current (DNS-based) approach shows a considerable improvement of the performance both in terms of both delivery delays and costs. In addition to this, we show the flexibility of our approach by implementing several request routing strategies that satisfy different optimization goals. The evaluation shows that the workload at the origin server can be reduced by a factor of 4, while the content can be delivered 30% faster compared to the traditional content delivery schemes. Finally, we propose an on-line algorithm that generates new routing configurations to satisfy operator's objectives and timely react to emerging conditions. The algorithm leverages Lyapunov optimization theory [6] to produce stable and cost effective strategies.

The paper is organized as follows. In Section II we present our novel framework based on the introduction of *information-aware* mechanisms. Section III shows the gains brought by our proposal with respect to the current CDN scheme. In Section IV we detail and evaluate an algorithm for the generation of request routing strategies offering improved content delivery performance, while ensuring stability. Finally, conclusions are provided in Section V.

978-1-5090-5093-2/16 $31.00 © 2016 IEEE
DOI 10.1109/CloudNet.2016.50

II. Information-Aware Model

A. Toward Information-Aware content delivery

The CDN architectures have deeply matured over the last decade to match the publishers' needs and the end users' requirements. However, they still show technological limitations when facing the unpredictability of the network conditions as well as the skewness of request patterns. This is mainly due the inefficiency introduced by the current request routing scheme (together with the rigidity of connection-full transport).

DNS resolutions are today the *de-facto* standard for assigning user requests to one of the available caches. The role of DNS-based request routing in content delivery is twofold: as a mapping system, it operates to bind a user to a suitable cache based on proximity metrics (e.g. select the nearest cache to the user); as a load balancer, it distributes customers traffic among the available caches depending on server load conditions and traffic handling costs. Recent solutions have aimed at improving the control on traffic, e.g. through centralized optimization [2]. Other proposals have addressed the accuracy of the client mapping, e.g. through enhancement of the user localisation [3]. However, the traditional, DNS-based, approach still constitutes a drawback for the following reasons:

a) Lack of reactivity: each DNS response has a TTL value (time-to-live) defining the temporal validity of the answer cached by ISP DNS and user hosts. Therefore, as long as DNS entries are still valid at user hosts or intermediate DNS servers, routing updates are not taken into account.

b) Additional delays: periodic DNS requests lead to extra delays that slow down the content delivery process, especially when the domain resolution requires several DNS redirections.

c) Unguaranteed reliability: the potential malfunction or disruption of the DNS service, which is not fully controlled by the CDN operator, has an impact on request routing reliability, and thus on the user quality of experience. Compatibility is also an issue when designing new solutions. For example, the solution in [3] could be deployed only for those resolvers supporting the EDNS0 client-subnet extension.

d) Content-unawareness: the name resolution mechanism associates the domain name of the required content item to a server address, disregarding other characteristics contained in its URI, like the specific content name or the corresponding category. This may limit the caching efficiency (reduced cache hit ratio), especially when small caches are used.

e) Static connection bindings: high volume flows (*e.g.* VoD traffic flows) are still hardly manageable (think about the case of user mobility), since once a cache/replica has been assigned to the client, the content is usually delivered chunk by chunk using the same long-lived flow [7].

The inefficiencies of the current delivery systems reveal a mismatch between the *content-oriented* nature of the CDN service and the *host-oriented* setup of the underlying network. This problem has been challenged by a large variety of works in the domain of ICN (Information Centric Networking). All these proposals, despite their differences, converge on three main concepts: *i)* the redesign of names to be location-independent and to make them directly bound to the intent of consumers or publishers, *ii)* on-path caching – e.g. in the CCN/NDN [8][9] architecture every network router also acts as a content cache – and *iii)* name-based routing to the nearest content replica.

By decoupling the data a user wants to access from how the data is delivered, ICN paradigm promises to lower the response times, simplify traffic engineering and improve the support of user mobility. Content delivery networks could largely benefit of the gains produced by the deployment of ICN paradigm, however the implementation of one of the "fully-fledged" ICN designs would require severe changes to the actual network, *i.e.* expensive upgrades of the forwarding devices (that could involve both hardware and software) and updates of the protocol stack. For example, considering NDN, which is one of the pioneering fully-fledged approaches to ICN, a complete implementation of such architecture would entail huge expenditures for transforming common routers in content-routers, as it could require changes on the packet processing pipeline, as well as the addition of expensive storage appliances. Furthermore, the resulting shift of the address space (from one billion IPs to at least one trillion content names [5]) would sharply increase the amount of routing state to be stored at content routers.

B. Design of an Information-Aware CDN

1) **Principles of our approach:** Differently from the related *clean-slate* ICN approaches, the architecture we propose incorporates ICN functions into the CDN infrastructure at a minimal cost, *i.e.* without requiring any modifications of the network physical infrastructure or on the Internet protocol suite. Our solution leverages an *Information-Aware* approach in which all (request/content) routing operations, including the ones at the user/cache interface, operate on the complete name of the requested content content *name*, e.g. the full content URI. Content-aware request routing enables the actuation of new strategies, in which any routing decision considers additional information – with respect to the current CDN user's mapping mechanisms, which basically operates on the domain name and the client (approximated) location – like the specific content's category or video channel. Increased "granularity" (up to the single content/chunk) in request routing operations can improve the precision in content resolution, with benefits on the content distribution performance, as it allows to improve the cache hit-ratio in the case of caches of small size with respect to the content catalog stored at the origin server.

As represented in Figure 1, in our approach the client's requests are directly handled by the CDN infrastructure, bypassing third-party resolvers like ISP operators' DNS servers or other distributed DNS like OpenDNS [10] or Google Public DNS [11]. This feature allows new policies and strategies to be enforced/reconfigured at very short timescales to match fluctuations of the users demand, reduce the cache workloads and react to network anomalies. In addition to this, further indicators such as the content's popularity or the related

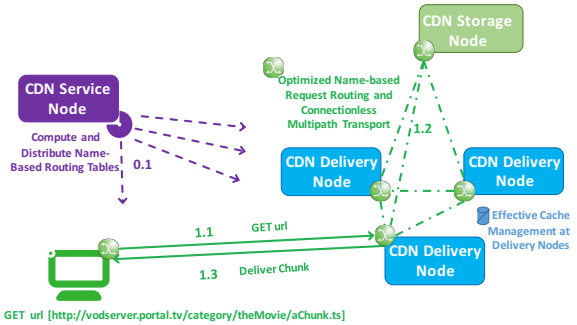


Fig. 1: Overview of Information-Aware CDN

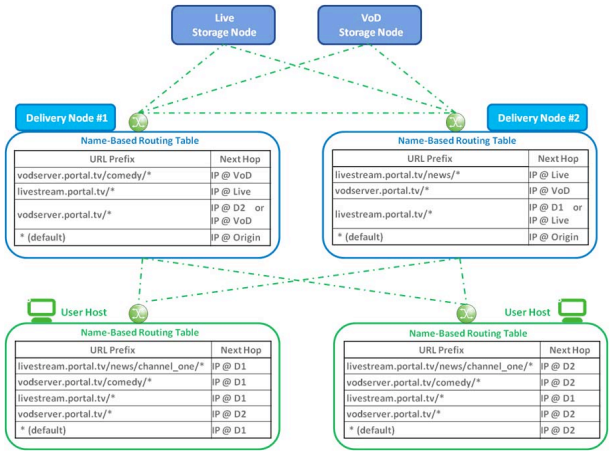


Fig. 2: Name-based request routing

geographical distribution [12] could come into play when deciding along which path, within the cache network, the content should be searched and retrieved.

To support name-based operations, our solution adopts a content retrieval scheme inspired by the CCN/NDN principles. Each user request is routed hop-by-hop along the CDN nodes according to the information maintained by *name-based* routing tables. In addition, each CDN node can save the state of the "un-satitisfied" requests to enable request aggregation and response duplication. When a request hits a content location, the individual requested item is routed back to the client, by default on a symmetric path. Data is delivered through connectionless transport, which is more suitable to a context of client mobility or dynamic network behaviour, since it allows fast path migrations. Table I summarizes the analogies and differences of the proposed solution with respect to the standard CDN mechanisms and to CCN/NDN operations.

2) **Name-based Request-Routing:** The main idea of name-based request routing is depicted in Figure 2. CDN delivery nodes maintain name-based routing tables integrated in their server software suite. This mechanism is also extended to client hosts, for example deployed as a plug-in for the Web browsers or as part of *ad-hoc* the applications developed by the content provider. Alternatively, proxy servers can be used. The routing tables are queried in case of a local cache miss or when the node is not equipped with caching storage. Individual table entries correspond either to a single content chunk (full URL), to a class of items, which can be the set of contents under a specific domain (*e.g. vodserver.portal.tv/**), or one subset of the last, identified by a specific channel or category name (*e.g. livestream.portal.tv/news/channel1/**). To handle a look-up miss, a default resolution is included in every table.

Similarly to the ICN case, our model relies on the use of hierarchical names, here given by the content URLs. When the table is checked against a requested content name, the matching entry is produced by longest-prefix matching. Such entry maps a single next hop or a pool of possible next hops, from which the next hop can be extracted according to a discrete probability distribution. Furthermore, the selected next-hop can be a storage node (including the *origin* server) or a delivery node empowered with caching capabilities.

The name-based routing tables are produced by one or more CDN service nodes and distributed to all delivery nodes and client hosts (or proxies) in their scope. New tables are generated in response to changing profiles of user demands, server loads, network traffic, cache setups, or provider policies. Every routing table configuration can represent an optimization goal (*e.g.* minimize delivery costs, minimize delivery delays, balance server charge, etc.) or a tradeoff between different strategies. The ability for frequent updates, with fast implementation, of the routing configuration enables the CDN to deal with the unpredictable nature of request patterns and traffic conditions (*e.g.* episodes like Flash Crowds).

III. Evaluation

We evaluate the proposed *Information-Aware* framework in two ways: *i)* a comparative evaluation between the current CDN delivery scheme and our model with a focus on content delivery performance and *ii)* a comparative evaluation of different request routing strategies to demonstrate the flexibility of our solution with respect to realistic operator's objectives. For the experiments we use trace-driven simulations on top of the Omnet++ framework.

A. Simulation Settings

We simulate our model using the topology and the cache configuration of a real CDN operator. The network topology includes 41 nodes and 141 links. Then, we analyze the efficiency of our framework using the following workloads:

- *Initial workload:* A real traffic trace captured from a north American commercial CDN operator. This workload is related to a Live TV service.
- *Popularity workload:* A partially synthetic workload obtained by increasing the request rate of popular contents. We generate three workloads with the following Zipf's law parameters: $\alpha = \{0.8, 0.9, 1.0\}$. Note that the real workload follows a ZipF law with parameter $\alpha = 0.7$.

TABLE I: Comparison of the proposed *Information-Aware* model with current CDN and CCN/NDN approaches

	Traditional CDN	CCN/NDN	Information-Aware CDN
Naming	Hierarchical (e.g. URL). Initial part is a domain name or an IP address	Hierarchical, but names are not necessarily URLs (they may not contain any DNS name/IP address)	Hierarchical, but names are not necessarily URLs (they may not contain any DNS name/IP address)
Request-Routing	Routing of user's requests to the address provided by the DNS resolver. On a cache-miss, an origin location is queried	Straight forward routing using an integrated FIB. Routing state for data is established at routers during request propagation	Request routing driven by software-level name-based routing tables integrated with the CDN constituents and players
Data Routing	Connection-oriented (e.g. TCP)	Connectionless. Data is sent from one end point to another without prior arrangement	Connectionless. Data is sent from one end point to another without prior arrangement
Content Caching	Network of dedicated caches	In path caching using content-routers	Network of dedicated caches

- *Flash crowd workload:* A synthetic workload generated to simulate a heavy load scenario characterized by globally high request rates. Handling the resulting traffic is a challenging task for the CDN.

To test the performance of our solution in terms of user QoE and service costs the following metrics have been used:

- *Average delivery delay:* The average delay to deliver a content to a client from the moment the request is issued to the moment the content is completely received by the user. Such delays constitute a key performance indicator, since they are used as the main criteria by content publishers to choose the CDN operators.[13]
- *Hit ratio:* It is defined as the percentage of requests served directly by the CDN cache without fetching from the origin. A high hit ratio implies a good request routing strategy and better use of the CDN resources
- *Completed requests:* The number of requests completed without re-transmission(s).
- *Origin cache load:* The rate of requests arriving at the origin servers. CDN operators seek to reduce this load, as it implies increasing transit and infrastructure costs.[14]

To highlight the flexibility of the proposed design we define and implement several routing strategies depicting potential operator objectives:

(i) Optimize delivery delays — client requests are directed to the nearest cache server. In case of a cache miss, requests are directed to the next-hop cache on the path toward the origin server;

(ii) Optimize cost efficiency — each cache server is assigned a subset of the content catalog. The requests are routed according to these assignments, which maximize the content availability at the cache servers;

(iii) Trade-off strategy — a hybrid configuration in which the content catalog and the cache servers are split into two classes. Then, client requests are directed to the nearest cache server of the same class.

B. Results Analysis

We evaluate the proposed solution across different workload profiles. Figure 3 summarizes the performance of our approach in terms of delivery delay, when compared to the current CDN solution, To accurately quantify the performance

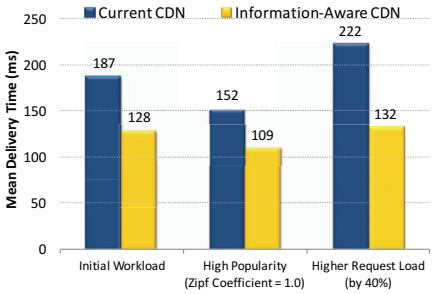
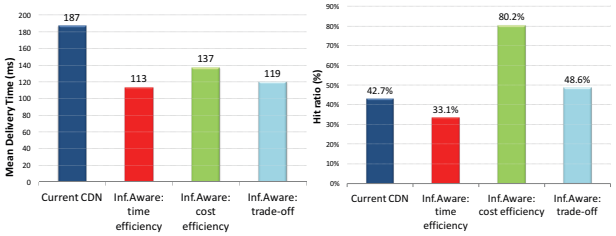
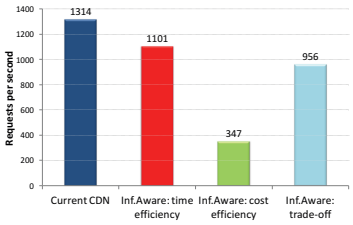


Fig. 3: Mean delivery delay comparison

gains, we have used the actual request routing policy and settings adopted by the CDN operator. As can be observed, our information-aware model significantly outperforms the currently deployed one at different workload conditions. This holds above all during peaks of user demand (Flash crowd workload). These gains are mainly due to the elimination of DNS overhead as well as to the connectionless transport of content data.



(a) Impact of routing strategies on delays(b) Impact of routing strategies on hit ratio



(c) Impact of routing strategies on Origin Load

Fig. 4: Impact of routing strategies

Figure 4 provides a comparative view of these strategies, focusing on: the average delivery delays, the average hit-ratio at delivery nodes (caching efficiency), and the load at the origin server (as good indicator of delivery cost). The results show that all strategies fulfilled the related operator objectives. The *cost efficiency* one is particularly effective, as it reduces the load on the origin server by a factor of 4, while significantly improving hit ratios, yet showing competitive delivery times.

IV. On stability and optimization

CDN operators are concerned about delivery delays as well as offering stable QoE to their clients. A degradation or failure of service could result in the loss of contracts with content providers. Flash crowd events are deemed as the main hurdle in front of a stabilized service offer. These events are characterized by large spikes of traffic that could overload servers and network links. Handling these unpredictable events is challenging for every CDN operator, since the reliability of content delivery service crucially relies on adequate availability of cache servers.

A. Design Goals

We design SORT (Stable and Optimized Request rouTing), an on-line algorithm that generates optimized and stable routing strategy for our information-aware framework. At first, a monitoring system collects at each time slot the performance measurements (round trip times between servers and clients, hit ratio and load from servers, and number of requests from clients) as well as the operator objectives. Secondly, a routing strategy is produced in a *(i)* centralized manner, where the service node generates and distributes the tables to the caches and clients, or in a *(ii)* distributed manner, where the service node sends the relevant measurements together with the operator's objective to the delivery nodes, so the tables can be computed locally. In order to reduce the strategy computation complexity, clients are grouped into clusters depending on their geographical position. Moreover, the content catalog is divided into *flows* based on the content name structure.

The designed algorithm has to meet the following goals:
Reliable service: The strategy produced by SORT algorithm must satisfy all the user requests.
Server load balancing: SORT must ensure continuous service availability and avoid overloads on the CDN servers.
High quality of service: SORT must optimize delivery delays to meet high quality of service standards.
Low cost: The aforementioned goals should be met while minimizing the delivery cost (high hit ratios and reduced load at the origin servers).

B. Formulation

SORT algorithm is called iteratively in a time slotted manner (around one minute per slot) to generate routing strategies respecting the aforementioned goals. Our algorithm receives as input the state of the servers, the network conditions, the users demand and the operator's objective. It outputs the most suitable way to map user requests to cache servers based on the corresponding content names.

Algorithm 1 Stable Optimized Request rouTing (SORT)

At the beginning of each time slot t :

- **Input:** load (*i.e.* rate of received requests) of each server J; operator's objective parameter V.
  From all server J: hit ratio for each flow K, round trip time to each client I, round trip time to the origin, number of received requests per flow and per client of the last slot: $A_{I,K}(t)$.
- **Output:** requests assignment $U_{I,J,K}(t)$

Solve the Lyapunov drift minimization problem in order to obtain the optimal allocation $U_{I,J,K}(t)$ that satisfies:

1) Serve all requests:
   $\sum_{I,K}(A_{I,K}(t)) = \sum_{I,K,J}(U_{I,J,K}(t))$
2) Avoid overload at server J:
   Load($\sum_{I,K}(U_{I,J,K}(t))$) < 100%
3) Reduce the average delivery Delay:
   Min($\sum_{I,J,K}(Delay(U_{I,J,K}(t))$))
4) Respect cost limit:
   $\sum_{I,J,K}$Cost($U_{I,J,K}(t)$) < V * CostLimit

Update name-based routing tables ($U_{I,J,K}$)/($A_{I,K}$)
Dispatch tables to every server J

Input: *Round trip time:* each server J measures $RT_{J,I}(t)$ (round trip time between itself and clients cluster I) and $RTO_J(t)$ (round trip time to the origin).
Hit ratio: each server J processes its log files to produce the hit ratio $H_{J,K}(t)$. This is performed for each flow K.
Clients requests: we approximate the client demand $A_{I,K}(t)$ by using the request rates from the previous time slot, classified by pairs <client cluster I, content flow K>.
Operator's objective: this is associated with a parameter V that may vary from 0% (time optimization strategy) to 100% (cost efficiency strategy). Low cost is represented by a reduced load at the origin servers (highly efficient use of caches).
Output: $U_{I,J,K}(t)$, which represents the amount of requests from clients within cluster I for content flow K to be directed to cache server J. This can be easily translated into a name-based routing table entry that assigns the incoming requests for the flow's content names to one of the CDN delivery nodes.
Formulation We use Lyapunov optimization theory to convert the stability concerns into a maximization problem[15]. Minimizing the Lyapunov drift allows to find the optimal allocation $U_{I,J,K}(t)$ that ensures full requests service, non overloaded servers and limited average delivery delays. Produced allocations result in costs $cost(U_{I,J,K}(t))$. To bound such cost we incorporate into our maximization objective a penalty function[6] pushing the objective toward higher hit ratio (low cost) $F(t) = \sum(U_{I,J,K}(t) * H_{J,K}(t))$.

C. Comparative Performance

To evaluate the performance of SORT we test its behaviour under unstable workloads and network conditions.

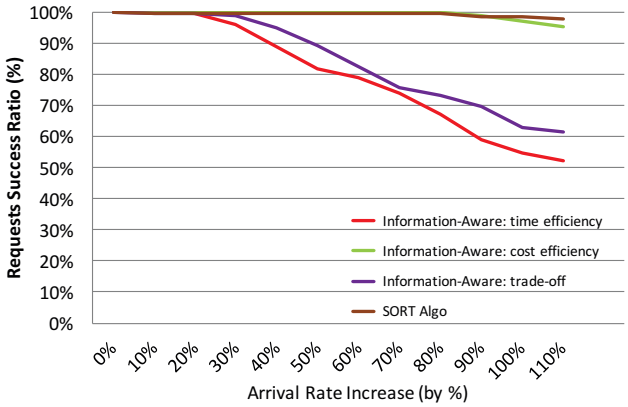


Fig. 5: Completed requests

1) Under heavy traffic: We are interested in the percentage of requests completed without retries. The results for different routing strategies are depicted in Figure 5. As can be seen, increasing the request rate raises the proportion of failed requests, since it poses a high burden on the origin servers that results in frequent connection timeouts. This holds above all for the "time efficiency" strategy, while in the case of the SORT algorithm the performance looks unaffected, since SORT periodically generates new configurations that improve the overall hit-ratio of caches, with a consequent sharp reduction of redirections to the origin server.

2) Under challenging network conditions: Figure 6 shows the average delivery delay in case of failures occurring on a single server within the cache network. The proposed algorithm manages to keep these delays low where other predefined strategies fail. This is mainly due to the reactive reconfigurations produced by SORT which allow effective adaptions to emerging network conditions like server failures, yet satisfying the efficiency requirements of request routing, *i.e.* to select the most suitable servers to avoid overloads and guarantee acceptable delays for all the requests.

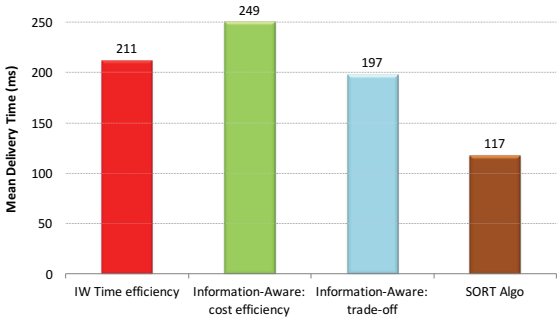


Fig. 6: Average delivery delays during a server failure

V. Conclusion

This paper proposed a novel approach to content delivery aiming at reducing the drawbacks of the currently used schemes (DNS-based request routing, static connection bindings). Our solution, which leverages Information Centric Networking principles, introduces a faster yet accurate request routing scheme and provides significant improvements in terms of reactivity and flexibility.

Moreover, we developed an on-line algorithm based on the Lyapunov optimization theory that generates efficient request routing strategies by exploiting the content popularity indicators. The proposed approach has a low computational time and, hence, can perfectly adapt to changing network conditions. The conducted simulations have demonstrated the promising potential of the proposed scheme, shown by a reduction of the delivery delay up to the 30% and an increase of the hit ratio – a measure of cost-effectiveness – by up to 100%.

References

[1] A. Balachandran, V. Sekar, A. Akella, S. Seshan, I. Stoica, and H. Zhang, "Developing a predictive model of quality of experience for internet video," in *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4. ACM, 2013, pp. 339–350.

[2] M. K. Mukerjee, D. Naylor, J. Jiang, D. Han, S. Seshan, and H. Zhang, "Practical, real-time centralized control for cdn-based live video delivery," in *the 2015 ACM SIGCOMM Conference*. ACM, 2015, pp. 311–324.

[3] F. Chen, R. K. Sitaraman, and M. Torres, "End-user mapping: Next generation request routing for content delivery," in *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*. ACM, 2015, pp. 167–181.

[4] A. V. Vasilakos, Z. Li, G. Simon, and W. You, "Information centric network: Research challenges and opportunities," *Journal of Network and Computer Applications*, vol. 52, pp. 1–10, 2015.

[5] D. Perino and M. Varvello, "A reality check for content centric networking," in *the 2011 ACM SIGCOMM Workshop on Information-Centric Networking (ICN 2011)*, 2011, pp. 44–49. [Online]. Available: http://doi.acm.org/10.1145/2018584.2018596

[6] M. J. Neely, *Stochastic Network Optimization with Application to Communication and Queueing Systems*, U. of Southern California, Ed. Morgan & Claypool, 2010.

[7] M. Wichtlhuber, R. Reinecke, and D. Hausheer, "An sdn-based cdn/isp collaboration architecture for managing high-volume flows," *Network and Service Management, IEEE Transactions on*, vol. 12, no. 1, pp. 48–60, 2015.

[8] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *Proceedings of the 5th international conference on Emerging networking experiments and technologies*. ACM, 2009, pp. 1–12.

[9] V. JACOBSON, D. K. SMETTERS, J. D. THORNTON, M. PLASS, N. BRIGGS, and R. BRAYNARD, "Networking named content," *Communications of the ACM*, vol. 55, no. 1, pp. 117–124, 2012.

[10] O. https://www.opendns.com/.

[11] https://goo.gl/p8cfJm, "Google public dns."

[12] D. Tuncer, M. Charalambides, R. Landa, and G. Pavlou, "More control over network resources: An isp caching perspective," in *Network and Service Management (CNSM), 2013 9th International Conference on*. IEEE, 2013, pp. 26–33.

[13] M. Z. Shafiq, A. R. Khakpour, and A. X. Liu, "Characterizing caching workload of a large commercial content delivery network."

[14] S. Hasan, S. Gorinsky, C. Dovrolis, and R. K. Sitaraman, "Trade-offs in optimizing the cache deployments of cdns," in *INFOCOM, 2014 Proceedings IEEE*. IEEE, 2014, pp. 460–468.

[15] W. Benchaita, S. Ghamri-Doudane, and S. Tixeuil, "Stability and optimization of dns-based request redirection in cdns," in *Proceedings of the 17th International Conference on Distributed Computing and Networking*. ACM, 2016, p. 11.