Energy-Aware Data Center Management in Cross-Domain Content Delivery Networks

Chang Ge, Ning Wang, Zhili Sun
Centre for Communication Systems Research
University of Surrey, Guildford, UK
Email: {C.Ge, N.Wang, Z.Sun}@surrey.ac.uk

Abstract—**A content delivery network (CDN) typically consists of geographically-distributed data centers, each containing servers that cache and deliver web contents close to end-users for localized content access purpose. In recent years, optimization of CDN data center energy consumption has attracted increasing research efforts. The key technical challenge is the tradeoff between CDN energy consumption and the content service capability at both the server and the network sides. In this article, we introduce a data center management scheme that effectively reduces the energy consumption of cross-domain CDNs through dynamically provisioning servers and coordinating content delivery operations with respect to dynamic server and network load. The proposed scheme focuses on optimizing the energy-performance tradeoff in two aspects. On one hand, servers in CDN data centers are put to the sleep mode during off-peak hours to save energy. On the other hand, CDN Quality-of-Service (QoS) performance is assured through honoring constraints on servers and network link loads, especially through restricting inter-domain content traffic volume. As a result, the proposed scheme is able to reduce CDN data center energy consumption without compromising its end-to-end QoS performance. According to our experiments based on realistic CDN scenarios, the proposed scheme is able to reduce data center energy consumption by up to 45.9% while achieving desired QoS performance.**

I. Introduction

In a content delivery network (CDN), multiple data centers (DC) containing content servers are strategically established within proximity of Point-of-Presence (PoP) nodes to serve end-users' content requests [1]. As web content objects (such as webpages and videos) are normally replicated and cached at the DCs, they are efficiently delivered to end-users in a more scalable and efficient manner thanks to localized content access. In general, each content request is resolved to either a local (preferred) or a remote DC. The contents are effectively delivered through the virtual path from the DC to the PoP where the user is attached, and the determination of the virtual overlay links' mapping to ISP physical networks are based on the underlying intra- and inter-domain routing configurations. In order to support end-to-end Quality-of-Service (QoS) assurance in content delivery operations, the CDN operator may "rent" bandwidth resources along the virtual links in the overlay across PoP nodes through establishing service level agreements (SLA) with underlying ISPs. In this way, a certain proportion of bandwidth resources in the ISP network is reserved for dedicated use by the CDN infrastructure. Such bandwidth reservation is typically achieved via virtualization techniques [2].

In recent years, energy efficiency in the information communication technologies (ICT) sector has attracted more and more interests from both the research community and the industry, which is especially the case for DCs and CDNs. In the literature, most relevant research works have focused on *standalone* DCs, which involves power management, performance management or both [3]. Meanwhile, there have been research works on reducing CDN energy consumption in recent years, which aim to optimize CDN energy consumption from either a theoretical perspective [4] or a practical perspective [5][6]. However, the scenario of energy saving in large-scale CDN infrastructures that typically cover *multiple inter-connected* ISP domains has not been well addressed.

For the CDN operators, it is crucial to meet end-users' QoS requirements through delivering the requested content objects with short end-to-end delay and assured bandwidth support. Therefore, in order to accommodate the uncertainty in user activities, CDN operators typically keep all DCs and their servers constantly up and running, even during off-peak hours when many servers are idle due to low content request volume. As a result, it is observed in [7] that most modern DC servers are utilized by only around 10% to 50% in average, which are rarely fully utilized during daily workloads. Moreover, a distinct observation is that a server consumes at least 60% of its peak power (around 100$\sim$300 Watts) when it is active, even without serving any content request. In contrast, this figure drops to only 5 Watts when it is configured into the sleep mode [7]. Therefore, it can be inferred that modern DCs suffer from poor energy efficiency due to under-utilized servers, and it is desired for the CDN operators to put some servers to the sleep mode when incoming content request volume is low, for instance during off-peak hours. However, although this technique has been widely applied to *standalone* DCs, this is not a trivial task in the cross-domain CDN scenario where multiple DCs are deployed in a geographically-distributed manner. This is because the decision on servers' sleep mode reconfiguration needs to consider not only the DC servers' service capabilities, but also network resource availability to support end-to-end content deliveries and assure QoS performance in terms of end-to-end delay.

In this article, we propose an energy management scheme that is specifically developed for cross-domain CDN infrastructures. The fundamental idea is to strategically reconfigure DC servers to the sleep mode without deteriorating the service

978-1-4799-0354-2/13/$31.00 ©2013 IEEE

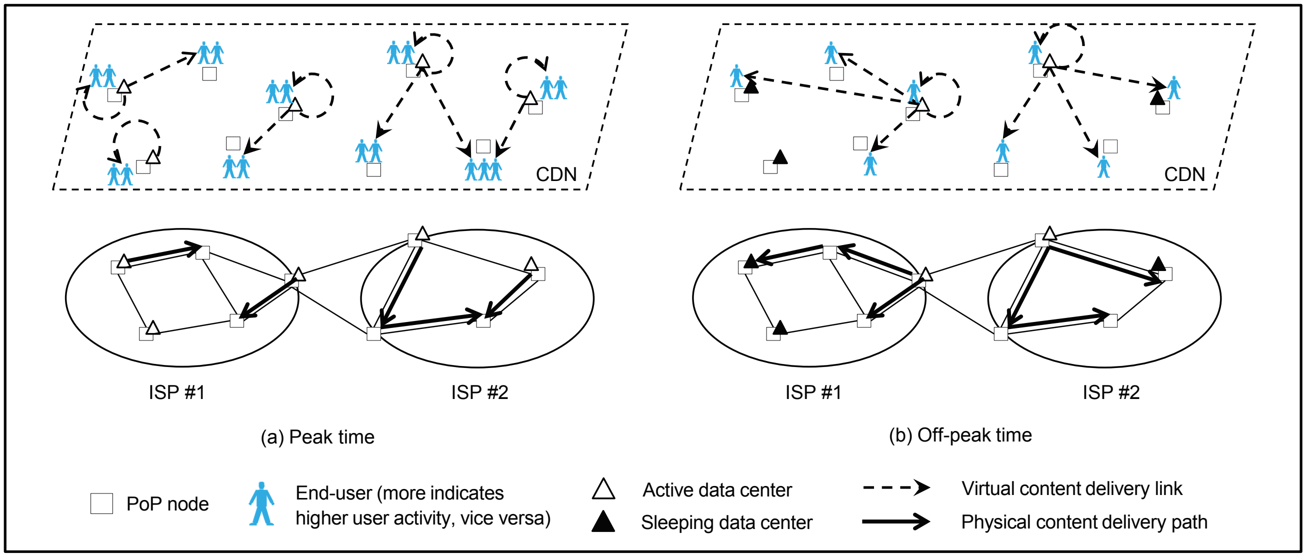

Fig. 1. Energy management scheme illustration: a) Peak time, when all DCs are active and serving user requests; and b) Off-peak time, when some DCs are in the sleep mode and requests are resolved to fewer active DCs.

capabilities on both the DC side and the network side. It is observed that user activities in a CDN typically follow a dynamic but relatively regular daily pattern [8], which implies that during off-peak hours, only a subset of all servers is needed to serve the requests. Therefore, in our scheme, content servers are dynamically provisioned with respect to the present content request volume instead of being kept active all the time. However, the realization of such an operation needs careful consideration on the tradeoff between the DC energy consumption and the CDN QoS performance. On one hand, the proposed scheme tries to put as many servers as possible to the sleep mode, as long as the remaining servers are capable of handling the present content requests. On the other hand, the server status reconfigurations also lead to changes in bandwidth utilization of the underlying ISP networks, as some end-users need to fetch content objects from more remote DCs. Intuitively, such request redirections will incur increased bandwidth consumption due to the growth of inter-PoP content traffic. In order to cope with such energy-performance tradeoff, specific constraints, which keep both DC servers and virtual links under their load capacities, need to be taken into account to avoid QoS deterioration in content consumption sessions with reduced content serving capability.

Moreover, there are some practical issues that are considered by our proposed scheme while performing on/off state reconfigurations on CDN servers. Firstly, during the server state reconfigurations, some ongoing or new content requests may need to be redirected to alternative DCs that are located at different PoP nodes. In a cross-domain CDN, this implies the possibility that the new DC might even be located within a different ISP domain. In this case, it can be inferred that the volume of inter-domain content traffic will be unnecessarily increased. Such a situation is effectively prevented in our proposed scheme which is still locality aware. Secondly, the reliability and lifetime of the server hardware will be affected if servers are turned on or off frequently [5]. In order to avoid such a situation, our scheme not only limits the number of on/off state transitions performed on servers, but also ensures that a server stays asleep for a sufficiently long period of time before it is activated.

II. Energy Management Scheme Overview

We consider large-scale CDN infrastructures that cover multiple autonomous ISP domains. In each domain, there exist a set of PoP nodes where local end-users initiate requests for web content objects. Each incoming content request is resolved to distinct DCs with specific policies. By default, a request is resolved to a local or nearby DC for improved QoS in terms of end-to-end delay [9]. Furthermore, within the specified DC, the request is assigned to a content server according to some load balancing policies (such as round-robin) to prevent servers from being overloaded [5]. These techniques are typically enforced via an overlay control plane in a centralized manner by the CDN operator [1]. In the proposed scheme, we focus on managing the PoP-to-DC request resolution policy across a CDN infrastructure, but it can be easily employed together with other DC-to-server request mapping policies to pursue further energy saving.

We first illustrate our main idea for energy saving in CDNs in Fig. 1. The conventional request mapping operation is shown in Fig. 1(a), in which five DCs are established to serve content requests from nine PoP nodes within two ISP domains. We can see that since the DCs are attached to five of the PoP nodes, requests from these nodes can be resolved locally without incurring inter-PoP bandwidth consumptions. For the other four PoP nodes, their requests are resolved to the DCs that are the closest to them, which is the common practice in real CDN environments to minimize user-experienced latency

[9].

At present, the CDN operators normally keep all DCs and servers up and running, so that end-users could experience optimized latency when requesting web contents [10]. Although this is necessary at peak hours as illustrated in Fig. 1(a), such a practice may lead to significant waste of CDN energy during off-peak hours. The main reasons are discussed as follows:

- From a single PoP node's perspective, the local end-users' request volume varies over different time periods within a single day, which is usually very low at midnight and early morning. Such a pattern has been observed in [8], which reflects the relatively-steady characteristic of CDN request volume at individual PoP nodes.
- From a CDN's perspective, content requests originate from geographically-distributed PoP nodes that are attached with end-users. Considering the difference in the global time zones among the PoP nodes or ISP domains, as well as the daily fluctuating pattern in end-user activities described above, it is unlikely that all domains experience high request volume simultaneously if we consider cross-continental ISPs (which is the case in typical large-scale CDNs).

Based on the two reasons above, it can be inferred that during off-peak hours, only a subset of servers is sufficient to serve all requests due to the low content request volume from local end-users. In this case, the remaining servers can be safely reconfigured to the sleep mode without introducing negative impact on the CDN performance, which will lead to considerable gains in energy saving of CDN data centers. Such a scenario is illustrated in Fig. 1(b), which shows that the servers in three DCs are put to the sleep mode during off-peak time in order to save energy. Meanwhile, all requests are served by the remaining two active DCs.

While considering energy savings, we also aim to assure the CDN performance in terms of end-to-end QoS. Such an objective is achieved by considering the following two aspects.

Firstly, basic performance assurance is provided through preventing DC servers and virtual content delivery links from reaching their full load capacities. As servers are responsible for handling user requests and delivering content objects, their response time degrades rapidly as they approach load capacities, which in turn affects the CDN QoS performance. Regarding CDN virtual links, as illustrated in Fig. 1, they are used to establish mappings between DCs and PoP nodes that are attached with end-users. Furthermore, network traffic incurred by content delivery will be mapped to underlying ISP networks for the actual delivery. Therefore, limiting traffic over virtual links reduces the risk of congestion in ISP networks, which has implications to end-to-end QoS performance.

Secondly, the CDN performance is further assured through avoiding the generation of unnecessary inter-domain content traffic. This is based on the following considerations. As DC servers are being put to the sleep mode, some requests will have to be redirected to an alternative DC, which could be located in another ISP domain. This should be avoided because a) inter-domain content delivery incurs significantly longer network distance, which increases end-to-end delay; and b) since inter-domain links are more critical compared to intra-domain links, they are more vulnerable to traffic congestions. Therefore, in order to reduce risk of congestion and assure user-experienced latency, inter-domain content delivery is avoided in our scheme when saving CDN energy. Such a scenario is illustrated in Fig. 1(b), in which content delivery sessions are restricted within each domain as three of the five DCs are put to sleep.

III. Energy Management Framework and Policies

In this section, we present the framework of our energy management scheme with relevant policies and show how it is embedded in a typical CDN platform. We also discuss some practical considerations that need to be taken into account when applying the scheme in real CDN environments. For the sake of clarity but without losing generality, we base our discussions on the scenario of Akamai networks [1], which is one of the largest CDN operators in the world.

A. Centralized Energy-Aware CDN Management

Since a CDN is a virtual overlay network built on top of physical ISP networks, it is essentially an application layer infrastructure over the Internet. In Fig. 2, we illustrate the functional blocks of a typical CDN platform with energy-awareness, which involves the following key components:

- **Monitoring Agents (MA)**. In a CDN, it is necessary to continuously monitor the conditions of DCs, servers and virtual links, since it is important that such information is up-to-date so that other components are able to make appropriate decisions on CDN management. The MA units keep reporting necessary context information above to data collection and analysis units and real-time mapping units, so that responses can be immediately made to cope with the events occurred in the CDN.
- **Data Collection and Analysis (DCA)**. The DCA unit is responsible for collecting data for general routine purposes such as logging, analyzing and billing.
- **Mapping Scoring (MS)**. The MS unit is part of the request mapping system in a CDN, and its responsibility includes creating an up-to-date Internet topology map including network connectivity, latency and loss information. The map is continuously refreshed by the MS unit.
- **Real-Time Mapping (RTM)**. The RTM unit is the other part of the CDN request mapping system. It creates a map with only DCs, servers and end-users, so that content requests are resolved to the best DC and server. It is responsible for request resolutions at both PoP-to-DC and DC-to-server levels, which are based on real-time information received from the MS and MA units. The RTM unit then instructs the DNS (domain name system) to resolve individual requests to their designated DCs and servers.
- **Communications and Control System (CCS)**. CCS is the channel for disseminating management information

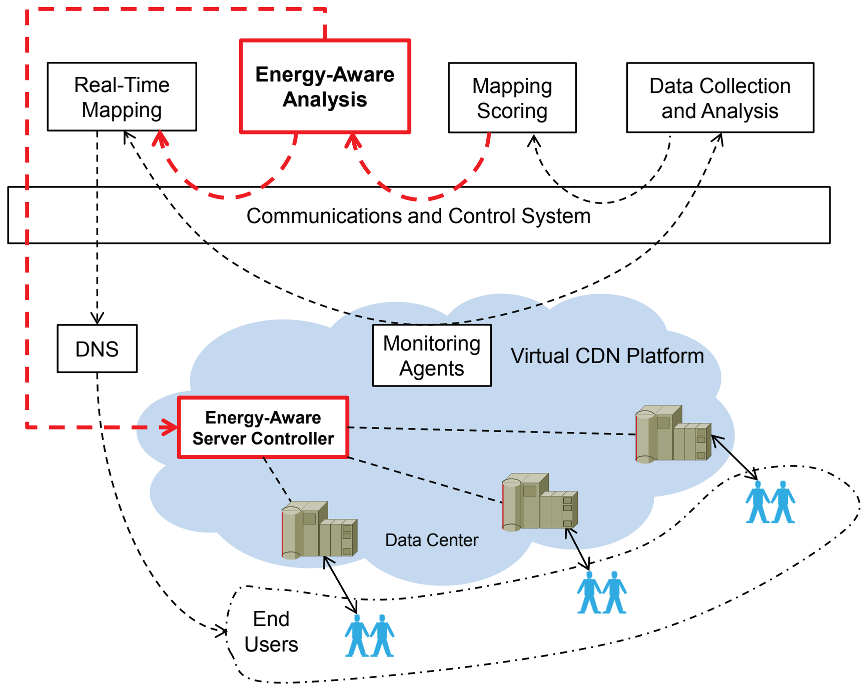

Fig. 2. Functional block diagram of proposed energy-aware CDN data center management system, based on [1]

such as control messages and status updates. As shown in Fig. 2, all management instructions between the functional units need to go through CCS.

So far we have described the basic functional components in a typical CDN platform. The question is - how can our proposed scheme be embedded in such an infrastructure? In Fig. 2, we also illustrate two additional management units that enable our scheme to be deployed in a CDN.

- **Energy-Aware Analysis (EAA)**. The EAA unit is the key component that is needed in our scheme, which is used to determine which specific servers need to be turned on or put to sleep. In order to make such a decision, it takes real-time CDN information as an input from the MS unit. With this information, The EAA unit is able to determine *how many* servers need to be turned on or off. Next, the CDN operator specifies a request-mapping policy to EAA, which is then used to determine mappings from user PoPs to DCs and servers to achieve optimized energy consumption and assured CDN QoS performance. Two examples of such request-mapping policies will be described in the next subsection. Afterwards, the EAA unit sends the mapping instructions to the RTM unit, which in turn instructs DNS to resolve the user requests.
- **Energy-Aware Server Controller (EASC)**. As a controller unit, EASC is built on top of the CDN platform to establish control over DCs and servers. As EASC receives instructions from the EAA unit on state reconfiguration of servers, it sends instruction signals to the specified servers for on/off status change. Such a technique is readily available in modern CDN platforms [1].

With the illustration of Fig. 2, we have shown that our energy-saving scheme is able to be embedded in modern CDN infrastructure in the form of functional blocks. By coordinating with other management components, our scheme is capable of utilizing real-time CDN information to make decisions to optimize the CDN energy consumption with assured QoS performance.

B. *Request Mapping Policies*

In the EAA unit, the CDN operator is able to specify different request-mapping policies that are suitable for CDNs with different characteristics and requirements. In this subsection, we show two examples of such policies, *i.e.,* Min-Energy-Dist and Min-Energy-DC.

- **Min-Energy-Dist**: This policy simultaneously optimizes DC energy consumption and network distances traversed by content delivery sessions. In other words, as the number of servers that are actively serving requests is minimal, requests are resolved to local or nearby DCs to optimize network distance and end-to-end delay. Under this policy, user requests are resolved in a distributed manner, and it is likely that every DC in the CDN will have some servers running to serve requests.
- **Min-Energy-DC**: While minimizing DC energy consumption, this policy aims to consolidate active servers into as few DCs as possible. In other words, content requests are resolved to the fewest possible number of active DCs, and the remaining DCs will have no active server running within them. This policy corresponds to the scenario illustrated in Fig. 1(b).

These two policies both have their own advantages and disadvantages. Regarding Min-Energy-Dist, since every DC has a subset of servers running, servers in each DC are provisioned in a more flexible way. For example, if the request volume suddenly increases in a specific area, the DCs in that area will have the flexibility to activate their own sleeping servers to handle the spike in local demand. However, since every DC has some servers running, the overall DC operational costs (*e.g.,* cooling system energy consumption) will be kept at relatively high levels. Regarding Min-Energy-DC, consolidating active servers to fewer DCs leads to easier DC and server management from a centralized point of view. Furthermore, if all servers in a DC are not needed for a certain period of time, then the entire DC could be scheduled for maintenance or sleeping for a given period of time, which introduces possibility of further saving in DC operational costs. However, the CDN becomes more prone to sharply-increased request volumes as entire DCs are being scheduled for sleeping mode reconfigurations. Therefore, Min-Energy-Dist is more suitable for CDNs with more dynamic user demands, where optimizing CDN QoS is prioritized over saving energy consumption. On the other hand, Min-Energy-DC is better for CDNs with regular user activity patterns, where DCs can be safely reconfigured to the sleep mode during off-peak hours without the need to worry about sudden increase in request volumes.

C. Working as a Whole System

After presenting the detailed information on the functional components and relevant operational policies, we now briefly describe how they work together as an energy-aware CDN management system. As MA and DCA continuously report up-to-date condition information on CDN network, DCs and servers to MS, the management system is always aware of the current CDN condition. Hence, MS forwards updated information to EAA, which is the key component of our scheme for decision-making purpose. EAA has two main responsibilities. Firstly, as EAA takes input from MS (as illustrated in Fig. 2), it makes decisions on which specific DC that each content request is resolved to, and send corresponding request mapping instructions to RTM. RTM then instructs DNS to resolve individual content requests to their designated DCs. Secondly, after EAA makes decisions on request mapping, it is able to determine which specific servers are safe to be turned on or off. It then instructs EASC to remotely configure the specified servers to on or off status. Such a management process repeats as MA and DCA keep refreshing CDN condition information in the system. As a result, the CDN energy consumption is optimized by EAA's instructions to RTM and EASC on request mapping and server on/off reconfigurations respectively. Meanwhile, the CDN QoS performance is assured during EAA's decision-making process by considering server and network load conditions.

It is worth noting that during the decision-making process above, EAA also aims to limit the number of on/off state transitions of the servers, so that wear-and-tear effect on server hardware is minimized. Such an objective is achieved via the following approaches. Firstly, as EAA is aware of the current status of servers, it tries to maximize the continuity in server on/off status during reconfigurations. In other words, on/off state reconfiguration will not be performed on a server unless the current request mapping policy requires so. Secondly, EAA also ensures that after a server has entered the sleeping mode, it will stay asleep for a sufficiently long period of time before it is powered on (unless the request volume suddenly increases and requires the server to be activated). Henceforth, the reliability and lifetime of CDN hardware are optimized by EAA while it saves energy consumption.

D. Content Delivery Session Redirection

At the moment when server on/off state reconfigurations take place, some ongoing content consumption sessions may need to be redirected on the fly to alternative servers as their originally-designated servers are affected. Generally, there are two different options for such session redirections:

- The session is redirected to another server located in the same DC. In this case, the redirection is seamlessly achieved through instructing the new server to deliver the content on behalf of the previous one. This is similar to the fast local server fault-recovery mechanism used by Akamai networks, and normally no human-perceived disruption to the ongoing content consumption sessions will take place.
- The session is redirected to another server located in a different DC. Normally, the ongoing content consumption sessions will firstly be disrupted while being switched to another DC. Furthermore, after the session is established between end-user and the new DC, QoS degradation may also occur due to increased network distance. In order to avoid a sudden and dramatic QoS deterioration for ongoing content consumption sessions in the above process, it would be better to "suspend" the server through letting it finish all the ongoing content delivery sessions, but not mapping any more new request to it until the server has finished all these ongoing sessions and hence is able to go to sleep. In this way, no ongoing session will be disrupted. Such a suspension mechanism has also been used by Akamai networks to fix servers with partial hardware faults.

Both options could take place in the two request mapping policies we have described, *i.e.,* Min-Energy-Dist and Min-Energy-DC. In Min-Energy-Dist, since all requests are

TABLE I
REPRESENTATIVE USER ACTIVITY SCENARIOS

#	USA (GMT-8 to GMT-5)		EU (GMT to GMT+2)	
	Time	User Activity	Time	User Activity
#1	[20:00, 23:00]	Medium	[04:00, 06:00]	Off-Peak
#2	[04:00, 07:00]	Off-Peak	[12:00, 14:00]	Peak
#3	[16:00, 19:00]	Peak	[00:00, 02:00]	Medium

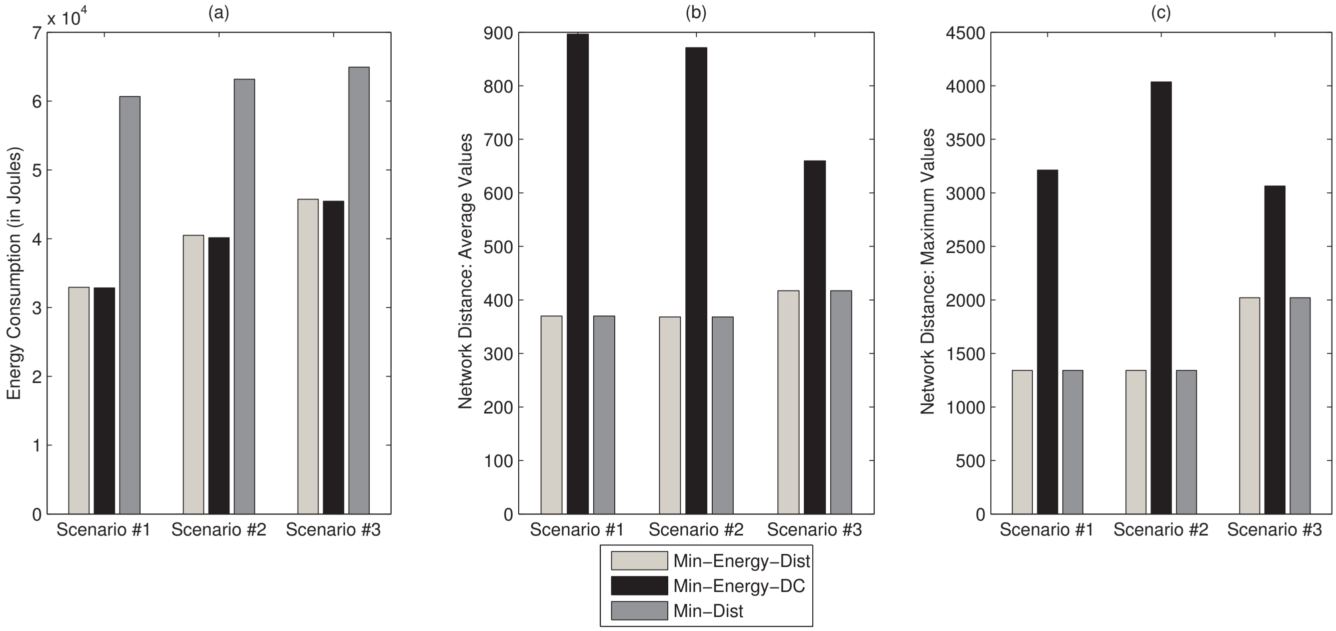

Fig. 3. Performance comparison among schemes with respect to different user activity scenarios: a) energy consumption; b) average network distance; and c) maximum network distance

resolved locally, the policy will firstly try to identify an alternative server in the same DC for seamless diversion, and an ongoing session will only be diverted to another DC if a local alternative server is not available. In Min-Energy-DC, it is more likely that ongoing sessions will be diverted to alternative DCs as active servers are consolidated to fewer DCs. However, under either policy or scenario, ongoing content consumption sessions will not be disrupted based on the techniques we have used.

IV. Experimental Results

In order to evaluate the performance of the proposed scheme, we use the real topologies from GEANT [11] and Internet2 networks [12], which are two interconnected autonomous domains in Europe and US respectively. Altogether, there are 34 PoP nodes that are distributed over Europe (25 nodes) and US (9 nodes), as well as 55 (bi-directional) network links interconnecting them (including 5 inter-domain links). We set 9 DCs to be deployed in the network (5 in GEANT and 4 in Internet2) and assume a 1:1 over-subscription ratio, which means the DC servers have *just* enough capabilities to serve all users in the network *simultaneously*. With respect to the conventional DC deployment strategy in a CDN [9], these 9 DCs are deployed to the 9 PoP nodes whose associated cities have the highest local populations.

We use the content request and traffic traces from the ClarkNet WWW server (an ISP covering Washington DC metro area) which is publicly available at the Internet Traffic Archive [13], and apply them to all PoP nodes according to the following approach. Firstly, we assume that end-users at all PoP nodes initiate content requests by following the same *pattern* as shown in the trace. Thereafter, the exact number of requests at each PoP node is approximated by applying a specific scaling factor to the request volume in the trace, which is proportional to the PoP's local population. This is necessary as the population associated to each PoP can vary substantially in reality.

As previously described, the proposed scheme takes into account time zone differences among PoP nodes in the CDN. Therefore, in the experiments, we identify three representative snapshots that reflect the changes in the CDN request volume within a single day while considering the geographical factor. For each snapshot, we calculate the overall request volume over one-minute period. These snapshots are listed in Table I, where they are referred to as scenarios #1, #2 and #3.

With the inputs specified above, we compare the performances of the following schemes.

Firstly, with the objective of minimizing DC energy consumption, we compare the request mapping policies of **Min-Energy-Dist** and **Min-Energy-DC**, which have been described in the previous section. Meanwhile, we also consider a reference scheme as a benchmark comparison:

Min-Dist: The scheme, as described in [14], optimizes content delivery efficiency through mapping requests locally or nearby. However, no energy awareness was considered and all servers are kept active regardless of the current request volume, which matches the common practice in DCs [10].

The experimental results are shown in Fig. 3, in which the following two metrics are compared in each of the three user activity scenarios:

- **Energy consumption** (in Joules): the amount of energy consumed by each scheme over the one-minute period,

TABLE II
DATA CENTER LOAD PERFORMANCE DETAILS

Data Center ID		GEANT (Europe)					Internet2 (US)			
		DC #1	DC #2	DC #3	DC #4	DC #5	DC #6	DC #7	DC #8	DC #9
Scenario #1	Min-Energy-Dist	66.7%	58.9%	93.3%	45.6%	73.3%	51.1%	37.8%	25.6%	13.3%
	Min-Energy-DC	100%	100%	100%	0%	34.4%	0%	34.4%	0%	92.2%
Scenario #2	Min-Energy-Dist	30.0%	18.9%	30.0%	15.6%	25.6%	100%	63.3%	62.2%	23.3%
	Min-Energy-DC	0%	18.9%	0%	100%	0%	68.9%	0%	100%	80.0%
Scenario #3	Min-Energy-Dist	16.7%	36.7%	55.6%	25.6%	37.8%	100%	100%	100%	60.0%
	Min-Energy-DC	0%	0%	100%	0%	68.9%	87.8%	73.3%	98.9%	100%

which is used to evaluate *energy-saving performance*.

- **Network distance**: the network distances traversed by content objects in each scheme while they are being delivered from DCs to end-users. This is calculated through aggregating the weights of links along the content delivery paths, which are proportional to real end-to-end delay according to the settings of GEANT and Internet2 networks [11][12]. Both *mean* and *maximum* values are investigated to reflect both average and worst cases in end-to-end delay, which are used to evaluate the CDN QoS performance.

It can be immediately observed from Fig. 3(a) that the proposed scheme can substantially reduce energy consumption of the servers in CDN data centers. In the three user activity scenarios, energy consumption has been reduced by 45.9%, 36.4% and 30.0% respectively. Generally speaking, the higher the content request volumes are, the less energy consumption can be saved as more active servers are needed to serve the requests.

From Fig. 3(b) and (c), it can be seen that regarding the average and maximum network distances traversed by content requests, Min-Energy-Dist and Min-Dist have achieved the same results. This shows that the Min-Energy-Dist policy is capable of significantly reducing DC energy consumption without compromising the CDN QoS performance. Compared with these two policies, Min-Energy-DC has increased network distances while achieving the same amount of energy saving as Min-Energy-Dist. This is because requests are mapped to fewer DCs instead of being resolved locally, and some requests need to travel among PoP nodes in order to be served.

In order to further indicate how Min-Energy-Dist and Min-Energy-DC perform, we show how the DC loads (in percentage) differ in these two policies under the three user activity scenarios in Table II. It can be observed that in Min-Energy-Dist, the requests tend to be distributed evenly among the DCs. This is because the DCs are deployed in a distributed fashion among all PoP nodes, and all requests are resolved to either a local or the nearest DC to optimize network distance. In contrast, it is shown in Table II that Min-Energy-DC tries to resolve requests to fewer DCs while pushing them toward their load capacities but without overloading them. Under such a policy, some DCs in the areas with medium or low user activity will not have any running server in them.

Based on Fig. 3 and Table II, it is shown that Min-Energy-Dist optimizes CDN energy consumption with respect to locality-based policy, which introduces some flexibility to the CDN server provisioning as every DC is active with a subset of servers running. On the other hand, Min-Energy-DC also optimizes CDN energy consumption, but through resolving requests to fewer DCs. Although it causes tradeoff in network distances, it introduces easier DC and server management as well as possibility of further saving in DC operational costs. Overall speaking, these two policies are able to save the same amount of energy under the same scenario, and it is up to the CDN operators to determine which one is more suitable with respect to their specific CDN requirements.

V. SUMMARY

In this article, we have introduced a novel energy management scheme that is able to effectively reduce CDN energy consumption. Based on the present request volume in the CDN, our scheme provisions the minimal number of active servers to serve content requests and reconfigures the rest to the sleep mode for energy saving purposes when the content demand is low. The CDN QoS is assured through honoring constraints on load capacities of servers and virtual links in the CDN, as well as restricting inter-domain content traffic to avoid increased end-to-end delay. Meanwhile, reliability and lifetime of CDN hardware is assured through avoiding frequent on/off state transitions of servers. From a practical perspective, the proposed scheme can be built as functional blocks and be embedded in modern CDN management platforms. Our experiments, which are based on realistic CDN scenarios and traffic traces, have demonstrated that the proposed scheme can achieve the energy reduction of up to 45.9% without compromising the CDN performance in terms of end-to-end delay.

ACKNOWLEDGEMENT

This work was partially funded by EU FP7 EVANS Project (PIRSES-GA-2010-269323).

REFERENCES

[1] E. Nygren, R. K. Sitaraman, and J. Sun, "The Akamai network: a platform for high-performance internet applications," *SIGOPS Oper. Syst. Rev.*, vol. 44, pp. 2–19, August 2010.

[2] N. Feamster, L. Gao, and J. Rexford, "How to lease the internet in your spare time," *SIGCOMM Comput. Commun. Rev.*, vol. 37, pp. 61–64, January 2007.

[3] C. Ge, Z. Sun, and N. Wang, "A survey of power-saving techniques on data centers and content delivery networks," *IEEE Communications Surveys & Tutorials*, vol. PP, no. 99, pp. 1 –21, 2012.
[4] L. Chiaraviglio and I. Matta, "Greencoop: cooperative green routing with energy-efficient servers," in *Proc. ACM e-Energy'10*. New York, NY: ACM, 2010, pp. 191–194.
[5] V. Mathew, R. K. Sitaraman, and P. Shenoy, "Energy-aware load balancing in content delivery networks," in *Proc. IEEE INFOCOM'12*. Florida, OL: IEEE, March 2012, pp. 954–962.
[6] P. X. Gao, A. R. Curtis, B. Wong, and S. Keshav, "It's not easy being green," in *Proc. ACM SIGCOMM'12*. New York, NY, USA: ACM, 2012, pp. 211–222.
[7] L. A. Barroso and U. Holzle, "The case for energy-proportional computing," *Computer*, vol. 40, no. 12, pp. 33–37, 2007.
[8] P. Gill, M. Arlitt, Z. Li, and A. Mahanti, "Youtube traffic characterization: a view from the edge," in *Proc. ACM IMC'07*. New York, NY, USA: ACM, 2007, pp. 15–28.
[9] R. Krishnan, H. V. Madhyastha, S. Srinivasan, S. Jain, A. Krishnamurthy, T. Anderson, and J. Gao, "Moving beyond end-to-end path information to optimize cdn performance," in *Proc. ACM IMC'09*. New York, NY, USA: ACM, 2009, pp. 190–201.
[10] C. Ge, N. Wang, and Z. Sun, "Optimizing server power consumption in cross-domain content distribution infrastructures," in *Proc. IEEE ICC'12*, June 2012, pp. 2628 –2633.
[11] Geant project home. [Online]. Available: www.geant.net
[12] The internet2 network. [Online]. Available: www.internet2.edu/network/
[13] Traces in the internet traffic archive. [Online]. Available: http://ita.ee.lbl.gov/html/traces.html
[14] J. M. Almeida, D. L. Eager, M. K. Vernon, and S. J. Wright, "Minimizing delivery cost in scalable streaming content distribution systems," *IEEE Trans. on Multimedia*, vol. 6, no. 2, pp. 356–365, 2004.