

RESEARCH

Open Access

SAViNE: social network analysis - inspired content delivery network deployment and experimentation

Aris Leivadeas, Chrysa Papagianni[†] and Symeon Papavassiliou^{*†}

Abstract

Over the last years, the content delivery network (CDN) market has been witnessing globally an increase in the development of cloud-based CDNs, as they constitute a viable and cost-effective alternative to traditional commercial CDNs. Towards that direction, Social-Aware Virtual Network Embedding (SAViNE) for wireless content delivery within the OpenLab project aims at establishing, assessing, and prototyping a novel framework for deploying a CDN over the wireless cloud. Specifically, the goal of SAViNE is to provide experimental validation of the efficiency of social network analysis (SNA) - inspired surrogate server placement strategies, devised for fostering content delivery. The proposed framework has been integrated with the OpenLab experimental facilities, providing a run-time environment for CDN deployment, operation, and performance evaluation over the wireless environment. The framework's feasibility and scalability is experimentally validated. The study is complemented with repeatability evaluation of experimental measurements, related to the resulting CDN's operational efficiency.

Keywords: Wireless testbeds; Experimental validation; Content delivery networks; Surrogate placement

1 Introduction

In today's digital landscape, the mobile channel has emerged as the most significant factor in consumers lifestyle. Rich media content providers are quickly becoming key elements for mobile channel monetization affecting a significant part of the global gross domestic product. To accommodate this traffic, numerous geographically disparate servers are needed, imposing significant cost overheads that are expected to multiply. Towards that end, content delivery network (CDN) providers (e.g., Akamai, Rackspace, etc.) facilitate the search and delivery of content.

Traditional CDNs have proven to be successful at optimizing and accelerating delivery of content. CDNs replicate content at cache and/or replica servers^a, deployed in multiple, geographically diverse locations, closer to the recipients of content [1]. End user requests are redirected to the most appropriate surrogate, based on network-

related criteria (e.g., traffic volume, proximity, etc.) as well as service-oriented criteria (e.g., response time, availability, etc.).

Emerging cloud-based CDNs leverage cloud resources to reduce the cost associated with implementing content delivery services (e.g., Amazon CloudFront, MetaCDN). Cloud-based CDNs promote a different business model where content is distributed on a pay-as-you-go model, supporting economies of scale, while 'CDN as a service' (CDNaaS) emerges in accordance with the 'everything as a service' notion [2,3].

Effective content distribution is crucial in CDNs [1]. Toward this direction, the appropriate placement of surrogates to locations that are closer to end users is considered an important challenge in CDNs [4]. While the problem has been studied extensively in traditional CDNs (e.g., [5-8], etc.), these results cannot be directly applied for cloud-based CDNs [2,9]. Few studies in literature investigate server placement for an overlay cloud-based CDN (i.e., [2,9,10]).

Following a cloud-based CDN approach in conjunction with the current trend of using lightweight (wireless/mobile) handheld devices (e.g., smart-phones,

*Correspondence: papavass@mail.ntua.gr

[†]Equal contributors

Institute of Communication and Computer Systems (ICCS), School of Electrical and Computer Engineering, National Technical University of Athens (NTUA), Greece, Iroon Politechniou, 9, Athens, Greece

tablets etc.) for accessing resource-voracious applications (e.g., media streaming, etc.), the Social-Aware Virtual Network Embedding (SAViNE) framework aims at establishing, assessing, and prototyping surrogate placement techniques for wireless cloud-based CDN deployment. SAViNE's goal is to provide experimental validation on the efficiency of social network analysis (SNA)-inspired surrogate server placement strategies devised for fostering content delivery within the evolving wireless cloud environment. SAViNE serves also as a proof of concept for delivering CDNaas over the Cloud, catering for the cost-efficient delivery of the long tail of the content. The evaluation is performed over a pure wireless multi-hop environment, in order to validate the *ad hoc* deployment of cloud-based CDNs, in the case that underlying wired infrastructure is not available or wireless infrastructure is preferred (e.g., wireless community networks).

1.1 Paper contributions and outline

The scope of this work is to provide experimental validation of the proposed framework for cloud-based CDN deployment, in realistic, large-scale wireless testing environment. The application of surrogate placement algorithms, followed by the deployment of the resulting CDNs, is conducted over a set of Future Internet (FI) research experimental infrastructures, namely (i) the w-iLab.t [11] wireless testbed for functional validation and feasibility check and (ii) PlanetLab [12] for scalability testing, utilizing appropriate techniques to emulate the

wireless environment. These experimental infrastructures are federated within the context of the OpenLab [13] and Fed4FIRE Integrated Projects [14] - as part of the Future Internet Research and Experimentation (FIRE) initiative in Europe [15].

The key contributions of this work are the following;

- The proposed SAViNE framework has been integrated with the OpenLab experimental facilities, providing a run-time environment for CDN deployment, operation, and performance evaluation.
- SAViNE's feasibility is validated by investigating the operational efficiency of the proposed solution in wireless environment, throughout each step of the CDN deployment process (server selection and placement, CDN deployment, CDN operation).
- The validity of the proposed framework is reinforced by investigating SAViNE's repeatability over the experimental infrastructures. Specifically, the repeatability testing methodology is streamlined, describing appropriate experiment scenarios and statistical analysis methods of experimental data.
- The scalability of the proposed solution is examined by incorporating PlanetLab testbed conducting cross-testbed experimentation.
- An appropriate set of performance evaluation metrics are presented, related to the feasibility/operational efficiency and scalability of the proposed framework.

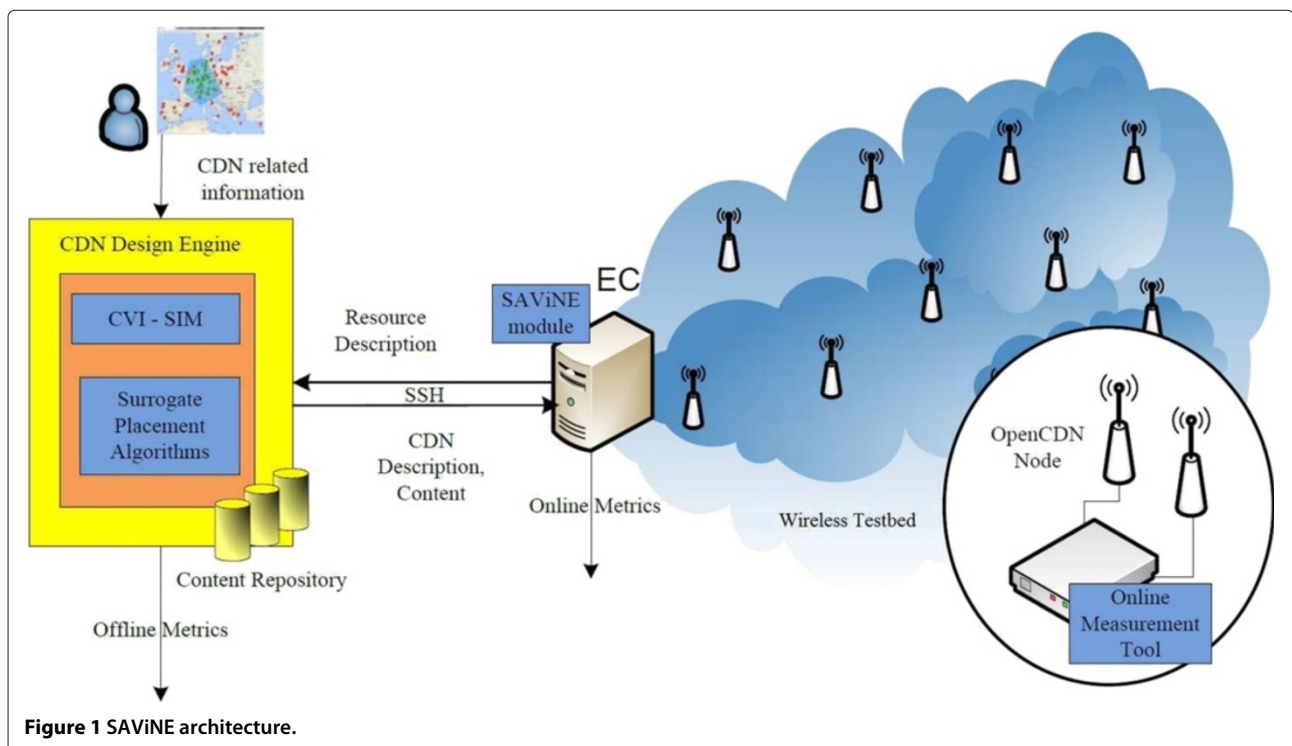


Figure 1 SAViNE architecture.

Table 1 Offline evaluation metrics

| | |
|--------------------------------------|--|
| Mapping cost | The cost of mapping a CDN on the testbed equals the amount of required bandwidth for all content distribution substrate paths as defined by the surrogate placement solution augmented by the computing resources allocated to each surrogate node. |
| CDN deployment cost | The overall cost of deploying a CDN is based on the adopted cost models (retrieval, update, storage [9]). The cost is essentially comprised by (i) the cost for storing the content at the selected sites (ii) the access cost of CDN end users within the area of service (iii) the cost of updating the content at the transit/surrogate servers |
| Number of surrogate servers | The number of nodes that are selected by the placement algorithm to host a surrogate server. |
| Path length | The average number of hops between CDN end users and the origin server in a particular CDN solution. |
| Shortest path betweenness centrality | Shortest path betweenness centrality for a particular CDN solution is defined as the average of individual SPBCs of the selected surrogates comprising the final CDN solution. |

The rest of the paper is organized as follows. In Section 2, a brief insight on the surrogate placement problem and the background work on the heuristics adopted for the particular study are presented. Following in Section 3, a description of the architecture of the SAViNE framework is presented, integrated with the OpenLab experimental facilities. The experimentation results are presented in Section 4, emphasizing on feasibility and scalability validation as well as repeatability evaluation. In Section 5, a set of closing remarks conclude the paper.

Table 2 Online evaluation metrics

| | |
|--------------------------------|---|
| Response time | User-perceived service response time. |
| Throughput | Surrogate throughput measured on the server wireless interfaces. |
| CPU load | The average load of the surrogate server aggregated over a pre-specified time window. |
| Cache hit ratio | The ratio of the number of times that content is served from the cache to the number of requests for content. |
| Client requests per surrogate | The average number of connected clients per surrogate. |
| Average distance per surrogate | The average distance of the connected clients from the surrogate. |
| SNR | Signal-to-noise ratio is a relative measure of the received signal power compared to the noise power in the wireless channel. |

Table 3 Feasibility validation - w-iLab.t offline metrics

| | SNA-VSP | SNA-GVSP | GS |
|--------------------|---------|----------|-------|
| Mapping cost (%) | - | 16.25 | 23.66 |
| CDN cost (%) | - | 19.62 | 26.66 |
| Surrogates | 6 | 10 | 11 |
| Path length (hops) | 2.54 | 2.34 | 2.65 |
| SPBC | 0.057 | 0.053 | 0.021 |

2 Background work

A decisive factor on the performance of a CDN is the number and location of surrogate servers. Optimizing surrogate placement enables the delivery of high quality services at low prices. Surrogate placement belongs to the NP-complete class of problems [16]. With regard to cloud-based CDNs, the replica placement is a complex, joint problem of building distribution paths and replicating content [9].

Within the SAViNE framework, the location and number of surrogates is optimized, taking into consideration metrics inspired by social network analysis. In SNA, the ‘popularity’ of nodes in networks has been successfully captured by the notion of ‘centrality’ and respective metrics [17]. Centrality in graph theory and network analysis is a quantification of the relative importance of a vertex within the graph. Modelling the network as an undirected weighted graph $G(V, E)$; let $\sigma(j, k)$ be the number of shortest (j, k) paths. Shortest path betweenness centrality (SPBC) [18] c is defined as the ratio of the number of shortest paths traversing node i denoted as $\sigma(j, k|i)$ over the total number of shortest paths in the network.

$$c(i) = \sum_{\forall j,k} \frac{\sigma(j, k|i)}{\sigma(j, k)} \quad (1)$$

For the sake of completeness, a brief description of the SAViNE heuristics introduced in [2] is provided.

Table 4 Feasibility validation - w-iLab.t online metrics

| | SNA-VSP | SNA-GVSP | GS |
|------------------------------------|---------|----------|------|
| Response time (s) | 1.71 | 2.42 | 3.52 |
| Client requests per surrogate | 7.33 | 4.4 | 4 |
| Average distance per surrogate (m) | 5.97 | 5.98 | 6.06 |
| Cache hit ratio | 0.45 | 0.31 | 0.20 |
| Throughput (%) | 46.3 | 5.17 | - |
| CPU load (%) | 2.47 | 0.2 | - |

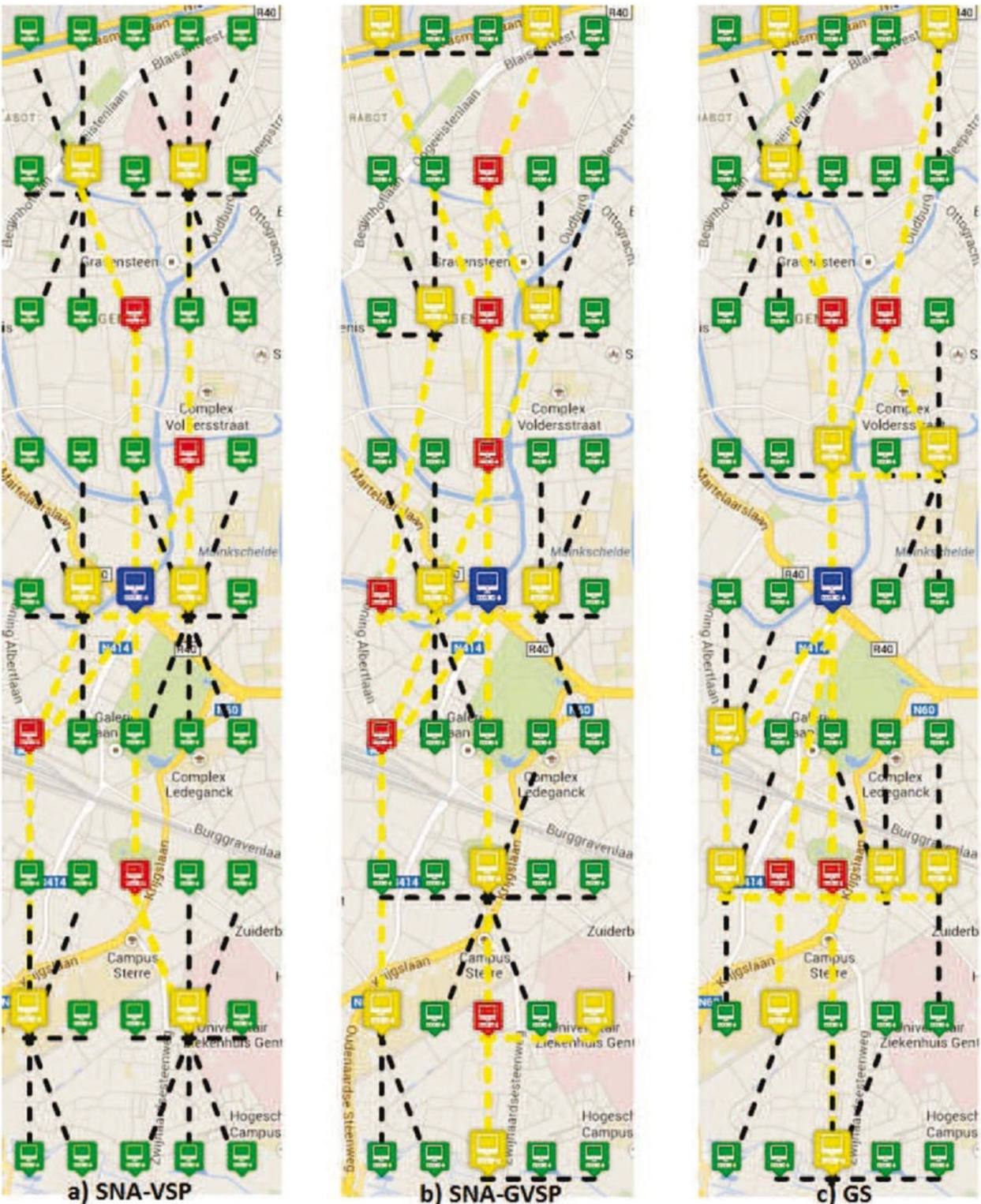


Figure 2 Feasibility validation - w-iLab.t CDN deployment topologies for a) SNA-VSP, b) SNA-GVSP, and c) GS algorithms.

2.1 Surrogate placement

SNA inspired - virtual surrogate placement heuristic

The SNA Inspired Virtual Surrogate Placement (SNA-VSP) is a sub-optimal algorithm, inspired by resource mapping algorithms in virtual networks [19]. The number and location of surrogates is optimized with regard to CDN deployment cost and SPBC, while taking into consideration quality of service (QoS) constraints. CDN deployment cost is comprised of update (upload), storage, and retrieval (download) costs (e.g., [20]) in the cloud environment. For a surrogate that serves user requests, its update cost is incurred by incoming traffic at the node in case of cache miss, and its retrieval cost is incurred by outgoing traffic to serve user requests. In the case that the server relays traffic to other surrogates (transit node), the retrieval cost is incurred by traffic to provision these servers [9]. With regard to QoS, the maximum (routing) distance between the edge server and the end user is usually considered, as it captures the communication quality between the two nodes and can be measured by either hop count or delay. Alternatively, as in this particular case, geographic distance can be used as an indicator of delay [2]. The algorithm is executed in two phases; (i) the surrogate server selection phase and (ii) the content distribution path selection phase.

Surrogate selection

We consider the joint set of potential surrogate/transit and origin(s) servers, along with their communication links, as an undirected physical network graph. The set of N end users is provided, where each user is associated with (i) a specific location determined by geographical coordinates, (ii) content request rate, and (iii) the size(s) of the request. The physical network graph set is augmented by introducing one pseudo-node for each end user, having the same properties (e.g., coordinates, etc.). For each pseudo-node, a cluster is created, with a radius that matches the maximum QoS distance that the corresponding end user can have from a surrogate server. Each pseudo-node is connected with infinite bandwidth to the physical network graph nodes within its cluster, creating a set of pseudo-edges. These pseudo-edges are added to the edge set of the the augmented graph.

An end user in the resulting CDN will be served by either (i) a surrogate server in its cluster, on the path to the origin server, or (ii) by the origin server in the case of cache miss, as a non-cooperative pull-based approach is used. Therefore, we define N origin-destination pairs between the origin server and pseudo-nodes in the augmented graph, where communication demands are defined by end-user request pattern (request rate and size) as well as cache misses. A mixed integer programming (MIP) minimum cost N -commodity flow problem

is formulated. The objective is essentially to minimize the overall CDN deployment cost as discussed earlier and maximize the average SPBC of the selected set of surrogates, while the QoS distance requirement for end users is satisfied, ensuring that the capacity of the physical resources is not exceeded. The detailed description of the MIP problem formulation is provided in [2].

Solving the particular flow allocation problem results in assigning end users (pseudo-nodes) to surrogates (physical network graph nodes), thus selecting the appropriate set of surrogates (cloud sites) that will be utilized for the CDN deployment. Since MIP problems are known to be NP-hard and hence computationally intractable, the optimal fractional solution is computed for the problem's linear programming relaxation of the integer variables, which can provide a solution at least as good as the integer one. The relaxed problem can be solved by any suitable linear programming method, in polynomial time. A rounding technique is applied to obtain the integer solution of the MIP problem [19].

Content distribution path selection

Once the set of surrogates have been selected and end user to surrogate assignment has been completed, we compute content distribution paths among origin and surrogate servers, by using a shortest path algorithm taking into consideration capacity constraints of the underlying physical network. Communication demands in the physical network graph are defined by end-user request pattern (request rate and size) as well as cache misses. The selected content distribution paths determine also the set of servers to be used for relaying traffic (transits) in the resulting CDN.

SNA inspired - greedy virtual surrogate placement heuristic

Cronin et al. [6] proposed the transit node heuristic for mirror placement on the Internet, where mirrors are placed on candidate hosts in descending order of out-degree. The heuristic was based on the assumption that nodes with the highest out-degrees can reach more nodes with lower latencies. Following the same incentive for the SNA Inspired - Greedy Virtual Surrogate Placement Heuristic (SNA-GVSP), the out-degree metric has been

Table 5 Temporal repeatability - ANOVA online metrics

| | F | p-value | F crit |
|---------------|-------------|-------------|-------------|
| CPU load | 0.00501221 | 1 | 1.472993464 |
| Throughput | 0.383495729 | 0.998865133 | 1.469073451 |
| Response time | 0.016543423 | 1 | 1.476323778 |
| SNR | 0.056285158 | 1 | 1.477756452 |

replaced by the SPBC metric. Specifically, the goal of SNA-GVSP is to greedily assign end users to surrogate servers, maximizing the average SPBC of the selected set

of surrogates. Each end user is assigned to at least one surrogate server within its cluster, so that the QoS distance constraint is satisfied, while ensuring that the capacity

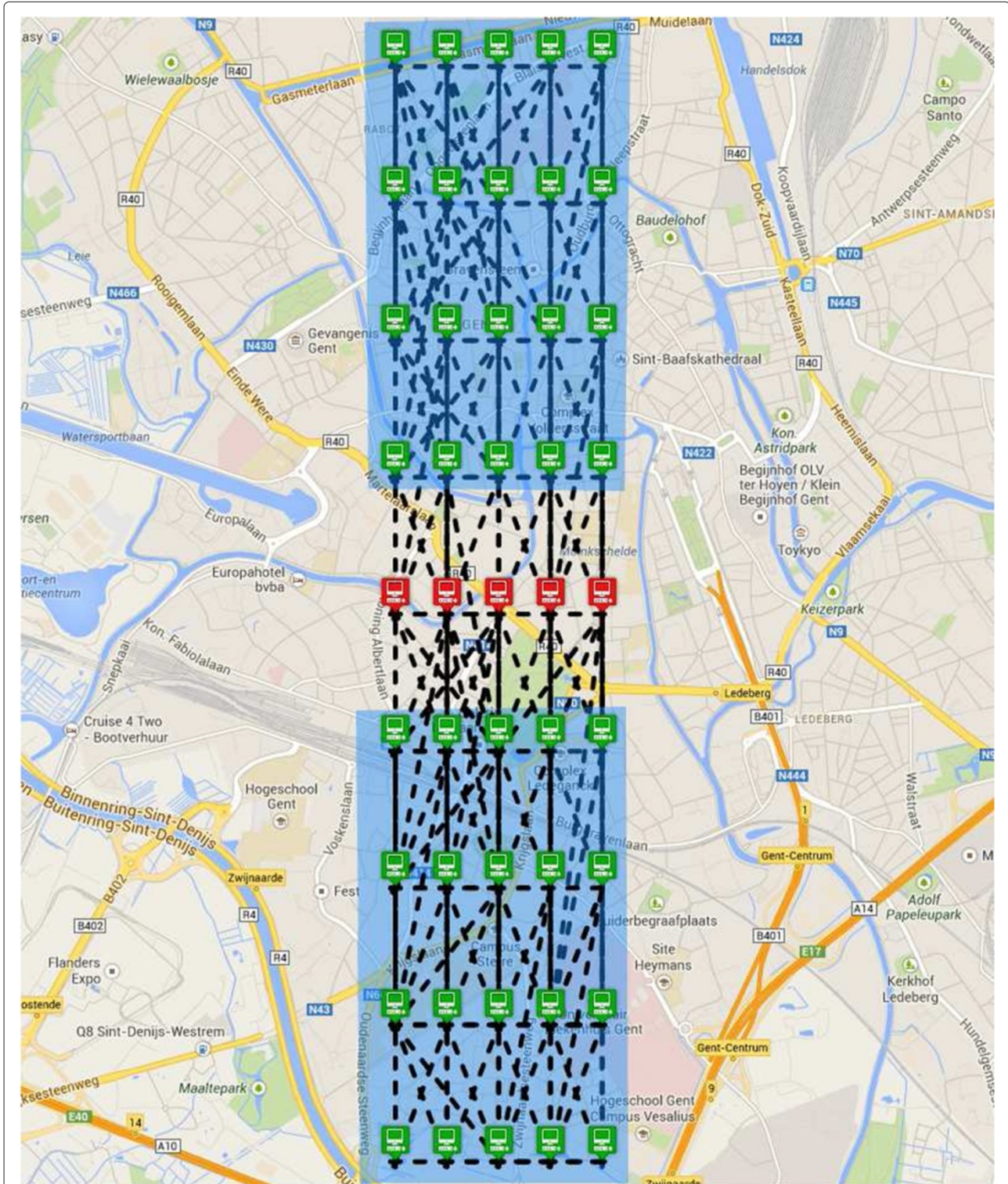


Figure 3 Spatial repeatability - SAVINE symmetric substrate topologies I and II - w-iLab.t Zwijnaarde testbed.

of the physical resources is not exceeded. The detailed description of the server placement algorithm is provided in [2]. Subsequently, to identify the content delivery paths from the origin to surrogates servers, a shortest path algorithm is applied, as the one in the content distribution path selection phase of the SNA-VSP, taking into consideration capacity constraints of the underlying physical network.

3 SAViNE framework: cloud CDN design, deployment, and operation

The SAViNE framework has been integrated with the OpenLab facilities; namely w-iLab.t and PlanetLab testbeds, to conduct experimental validation of the aforementioned server placement techniques. CDN design, deployment, and operation is realized in three consecutive phases:

- *Offline planning* phase: The selected server placement algorithm is refined with the necessary information such as (i) information regarding the testbed's physical infrastructure subject to disclosure policies by testbed providers (e.g., node/site location) and (ii) CDN-related information provided by the user of the SAViNE framework (e.g., the content provider), such as geographical dispersion [21], request rate and requested traffic volume of end users [10], replica size(s), and QoS distance [9].
- *Real-time design* phase: Taking into consideration information acquired during the *offline planning* phase, the server placement algorithm is executed, providing the actual CDN topology that will be deployed on the testbed(s). During this phase, predetermined evaluation metrics (denoted hereafter as *offline metrics*) are gathered. Offline metrics signify (i) the efficiency of the surrogate placement strategy and (ii) the impact of the SNA-inspired metrics on the CDN design.
- *CDN deployment and operation* phase: During this phase, CDN deployment takes place using the results produced from the *real-time design* phase. The resulting CDN is fully operational and administrable by the user of the SAViNE framework. During the actual operation of the CDN, a set of predefined evaluation metrics (denoted hereafter as *online metrics*) are acquired. These are crucial for the evaluation of the deployed CDN in terms of its performance in delivering content to CDN end users.

3.1 Architectural overview

The SAViNE framework is implemented as loosely coupled set of components. Four main entities interact with each other during the CDN design, deployment, and

operation namely the *SAViNE GUI/CDN design engine*, the *SAViNE module*, the *CDN software*, and the *online measurement tool*. The reference architecture is provided in Figure 1.

3.1.1 SAViNE GUI/CDN design engine

The *CDN design engine* is responsible for the *offline planning* and *real-time design* phase. The core of this component is based on the discrete event Java-based simulator, called Simulator for Controlling Virtual Infrastructures (CVI-SIM) [22]. CVI-SIM was initially used to evaluate the performance of the proposed server placement heuristics [2]. It provides an extensible simulation environment that was developed to facilitate research on the control of (virtualized) infrastructures. CVI-SIM acts also as an emulator since it is designed to support importing actual resource specification files (e.g., GENI v3 RSpec [23]).

In the context of SAViNE, CVI-SIM facilitates the execution of the proposed SAViNE heuristics based on the set of input parameters set by the user of the SAViNE framework and provides as output the CDN overlay to be deployed, along with CDN deployment related information and *offline metrics* in distinct XML files. The set of input parameters to the SAViNE heuristics include (i) RSpec advertisement files providing details on the location of wireless nodes and their functional properties, (ii) the set of nodes to be used as potential surrogates, (iii) the node to be used as origin server (by default is selected as the node closest to the center of mass of end users [2]), (iv) the content to be distributed, (v) QoS distance, and (vi) site-related unit costs (storage/downloading/uploading).

Via the SAViNE graphical user interface (GUI), the user may initiate each one of the three phases of the experiment as described in the previous section.

3.1.2 SAViNE module

The SAViNE module (Figure 1) is the control entity that initiates the overall experimental process and orchestrates the *CDN deployment and operation phase*. It is developed as a custom shell script, deployed over a single node per testbed (e.g., the experiment controller (EC) for OMF-based testbeds such as w-iLab.t). The module retrieves from the CDN engine all necessary information (e.g. CDN

Table 6 Spatial repeatability - w-iLab.t offline metrics (spatial repeatability)

| | SNA-VSP topology I | SNA-VSP topology II |
|--------------------|--------------------|---------------------|
| Mapping cost (%) | - | 0 |
| CDN cost (%) | - | 0 |
| Surrogates | 3 | 3 |
| Path length (hops) | 1.8 | 1.8 |
| SPBC | 0.083 | 0.083 |

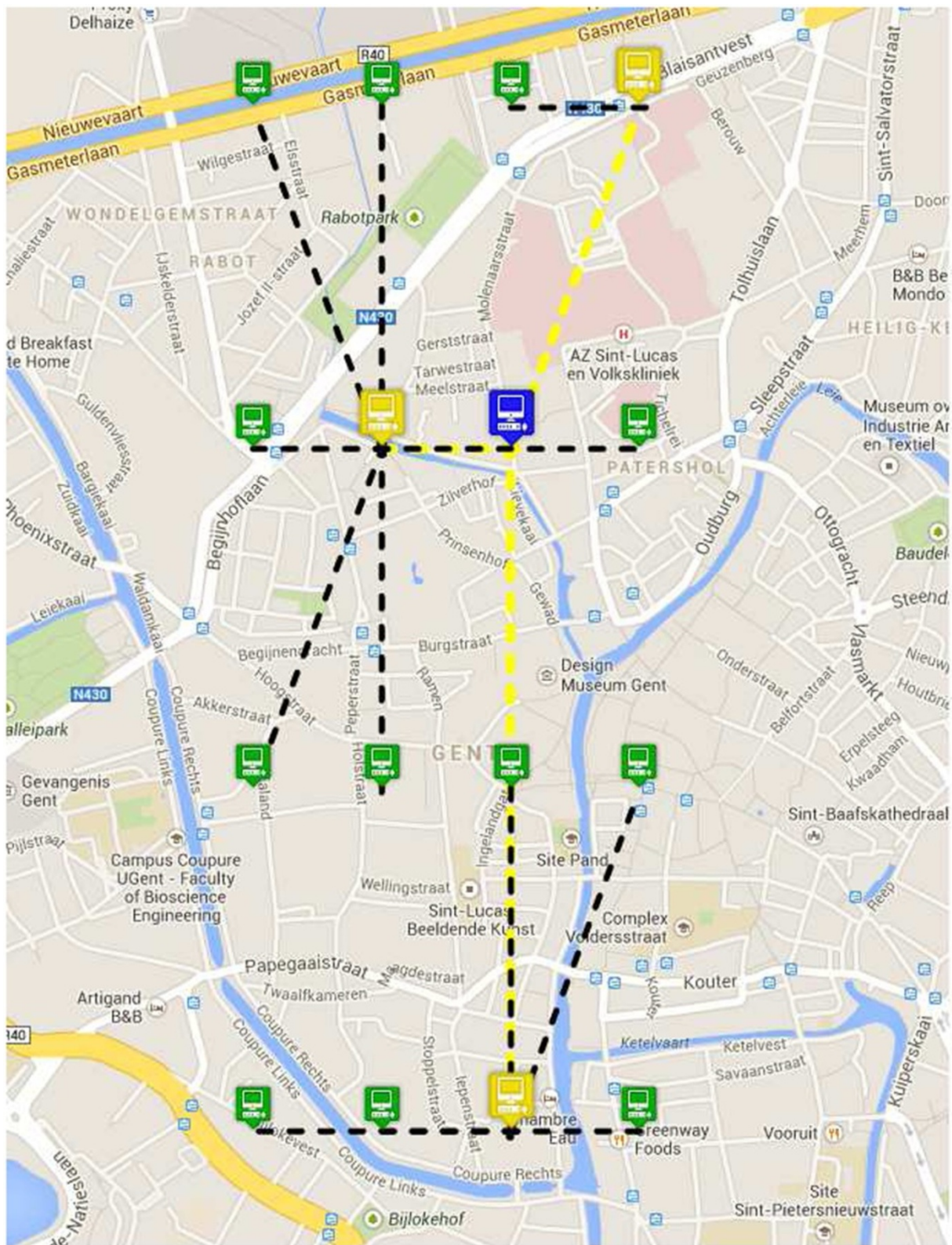


Figure 4 Spatial repeatability - w-iLab.t CDN deployment.

topology, etc.) to deploy the CDN overlay over the testbed and setup its operation. SAViNE module configures each CDN node based on its corresponding role(s) in the experiment (e.g., origin, transit, surrogate, etc.) and transfers the content to the origin server. It is also responsible for the aggregation of *online metrics*, communicating with each of the surrogate servers at the end of the *CDN deployment and operation* phase. The pseudo-code of the script is provided in Algorithm 1.

Algorithm 1 SAViNE Module Script

```

input = CDN_topology.xml
for all line in input do
    find Origin
    find Transits
    find Surrogates
    find Clients
end for
CDN_nodes = Origin.concat(Transits).concat(Surrogates)
for all node in CDN_nodes do
    set configuration files
    set wireless interfaces
    start the OpenCDN software
end for
for all client in Clients do
    set wireless interfaces
    url = request_content(client.content)
    RTSP url
end for
    
```

3.1.3 CDN software

To facilitate the automated deployment of the content delivery scenario, OpenCDN [24] (v0-7-7) software is used for the deployment of the application-level overlay content delivery network. In the OpenCDN terminology, an appropriately defined set of nodes act either as *edge servers* (cache/surrogates) or as *transit servers* - following a cooperative pull-based approach for content outsourcing - under the control of a centralized module named *Request Routing and Distribution Manager* (RRDM). RRDM orchestrates media distribution among CDN nodes and reports to an end-user portal the node address where to route a viewer request. *Origins* provide content and publish metadata, describing it to the portal. For the purposes of SAViNE, the origin/RRDM roles coincide to a single node. Either Apple's open source Darwin Streaming Server (v6.0.3) or VLC streamer Goldeneye is used for streaming content, depending on the testbed involved, and VLC media player is selected as the default client.

3.1.4 Online measurement tool

Apart from OpenCDN software, the online measurement tool (OMT) is deployed at the edge servers. OMT

Table 7 Spatial repeatability - z-Test online metrics

| | z | P(Z <= z) two-tail | z critical two-tail |
|------------|--------------|--------------------|---------------------|
| CPU Load | 0.040192878 | 0.967939356 | 1.959963985 |
| Throughput | -0.099154142 | 0.921015885 | 1.959963985 |

retrieves periodically (i) server CPU load and throughput values and (ii) latency and cache hit ratio values by the OpenCDN log files. Moreover, it captures the link quality for each of the CDN end users the surrogate serves. OMT is also implemented as a custom shell script.

4 Experiment and performance evaluation

The goal of the SAViNE framework is to establish an efficient content distribution overlay in a wireless environment. The experimental evaluation is essentially conducted during the *real-time design* phase and the *CDN deployment and operation* phase. For the sake of comparison, apart from the SNA-inspired heuristics, the greedy site (GS) algorithm [9] is also used as a server placement technique.

4.1 Performance evaluation metrics

As defined in the previous section, performance evaluation metrics have been classified as; *offline metrics* and *online metrics* for the evaluation of the *real-time design* and *CDN deployment and operation* phase, respectively.

Offline metrics, depicted in Table 1, are used to evaluate the efficiency of the server placement techniques. Specifically, these can be compared on the basis of the *CDN deployment cost* and *mapping cost*, the *number of surrogate servers* selected along with other relevant QoS/geographical proximity-related metrics e.g. the *path length* from the end user to the origin server of the CDN. Moreover, they are used to quantify the impact and effectiveness of the adopted SNA-inspired algorithms on the CDN design (e.g., average *SPBC*, the *number of surrogate servers* selected, etc).

Online metrics, presented in Table 2, are commonly used by CDN providers [1] to measure the CDNs ability to serve customers with the desired content.

4.2 Feasibility validation

In order to validate experimentally the feasibility and operational efficiency of the proposed solution, the SAViNE framework is integrated with the Zwijnaarde

Table 8 Spatial repeatability - t-Test online metrics

| | t | P(T <= t) two-tail | t critical two-tail |
|---------------|--------------|--------------------|---------------------|
| Response time | 0.508503227 | 0.61508386 | 2.048407142 |
| SNR | -1.121891225 | 0.274004776 | 2.073873068 |

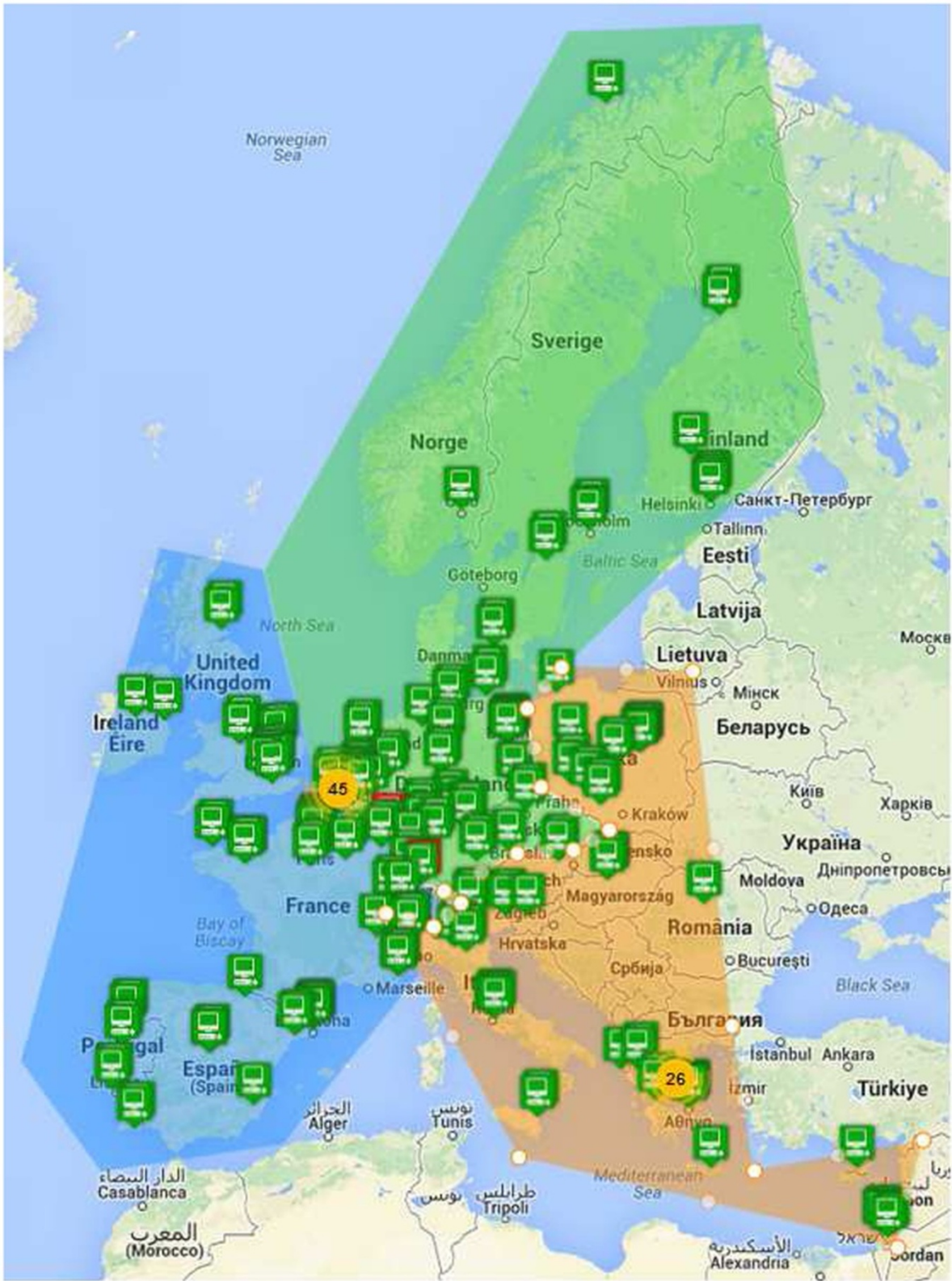


Figure 5 PlanetLab sectors.

testbed at w-iLab.t [11]. An additional objective is to evaluate the impact of the adopted SNA-inspired algorithms on the CDN design.

At the Zwijnaarde testbed, 60 fixed nodes are distributed over an area of 66×20.5 -m room. A thorough description of the testbed facilities is provided in [25]. A subset of 45 nodes is used to evaluate the performance the surrogate placement heuristics. To facilitate the wireless multi-hop environment, the transmit power of the fixed Zwijnaarde nodes is set to 1 mW. During the execution of the *real-time design* phase, different non-overlapping channels are allocated on the different surrogate servers/access points using a graph colouring technique, thus eliminating interference of adjacent access points.

The w-iLab.t nodes act as end users and potential surrogates/transit servers. node number 26 of the testbed is selected as the origin/RRDM server, as described in the SAViNE framework section. A set of four mp4 files is used with a replica size up to 21 MB. Similar to [10], each client is requesting a specific mp4 file at a pre-specified rate. QoS routing distance is restricted to one-hop while a geographic distance constraint of at most 8 m from the surrogate server is set. Furthermore, the unit storage cost is set to 0.18, the downloading cost is set to 0.17, while uploading cost is set to 0.1 (Amazon EU [9]).

4.2.1 w-iLab.t offline metrics

Following the execution of the various algorithms, the values of the *offline metrics* are illustrated in Table 3. The *CDN deployment cost* and *mapping cost* results are presented as the relative percentage differences of the SNA-GVSP and GS values from the SNA-VSP ones.

With regard to CDN deployment over the wireless environment and content replication, SNA-VSP outperforms all greedy heuristics in terms of individual metrics such as the overall *CDN deployment cost* and the *number of surrogate servers*. In the weighted multi-objective formulation of the SNA-VSP, the impact of the term that corresponds to the total cost of computational resources allocated to the set of selected edge servers is stressed. That leads to a decrease in the *number of surrogate servers*, compared to SNA-GVSP, and a slight increase in the *path length* within the service area (average number of hops). Based on the results presented in Table 3, the SNA-VSP leads to reduced *CDN deployment cost*, as it requires the smaller *number of surrogate servers* and employs small content distribution paths. Following a similar trend, the *mapping cost* for the SNA-VSP is reduced, indicating more efficient use of the underlying physical resources.

To study the impact of the adopted SNA-inspired algorithms on the CDN design, we particularly note the effect of the *SPBC* metric. The performance of SNA-inspired surrogate placement algorithms validates the assumption

that by selecting testbed nodes that exhibit larger *SPBC*, it contributes to the enhancement of the inherent characteristics of the CDN solution, as opposed to non SNA-inspired solutions; a fact evident from the comparison of the SNA-VSP/SNA-GVSP and GS algorithm. The SNA-related objective that is maximizing average *SPBC* drives the algorithms to select a smaller number of surrogate servers, re-using ‘popular’ substrate nodes leading to smaller *mapping/CDN deployment cost*. We also notice that by selecting substrate nodes that exhibit larger *SPBC*, it leads to smaller paths in comparison to the GS heuristic. Furthermore, we notice that both SNA-VSP and SNA-GVSP algorithms exhibit similar *SPBC* as they both attempt to maximize the average *SPBC* of the selected set of surrogates. However, *SPBC* is not the only denominator for the SNA-VSP; the algorithm also attempts to minimize deployment cost by taking into consideration the costs for updating/retrieving/storing content on the selected sites (surrogate nodes).

4.2.2 w-iLab.t Online metrics

Following the *offline planning* and *real-time design* phase, the produced CDN topology is sent to the EC in order to be deployed in the actual testbed. The *online metrics* for the experiment conducted are illustrated in Table 4, while the actual CDN topologies produced from the different algorithms are presented in Figure 2. The role of the nodes can be identified by a color code; the nodes in yellow are surrogates, the ones in red are transits, while the end users are illustrated as green and the node in blue is the origin server. The time average of the mean *throughput* and *CPU load* for the different CDN deployments is estimated over a time period of 4 min which is the average duration of a content being served. *Throughput* and *CPU load* are presented as the relative percentage differences of the SNA-VSP and SNA-GVSP values from the GS ones.

CDNs are deployed according to the results of the surrogate placement algorithms. *Response time* is related to the content distribution *path length* and (inversely) to the *cache hit ratio*, with SNA-VSP and SNA-GVSP having

Table 9 Scalability validation - countries/sectors/number of hosts - PlanetLab testbed

| Region | East | West | North |
|--------------|-------------|-----------------|-----------------|
| | Poland (17) | Spain (8) | Germany (28) |
| | Italy (9) | France (22) | Finland (5) |
| | Israel (4) | UK (11) | Belgium (8) |
| | Greece (13) | Portugal (3) | Czech (2) |
| | Hungary (3) | Ireland (3) | Sweden (4) |
| | Austria (4) | Switzerland (7) | Denmark (2) |
| | Cyprus (1) | | Netherlands (1) |
| Total | 51 | 54 | 50 |

Table 10 Scalability validation- w-iLab.t/PlanetLab trial setup

| Trial number | Nodes | Origin/RRDM server |
|--------------|---|---------------------|
| 1 to 3 | 96 nodes/w-iLab.t and east sector of PlanetLab Europe | planet1.elte.hu |
| 4 to 6 | 150 nodes/w-iLab.t, and east and west sectors of PlanetLab Europe | planetlab-1.imag.fr |
| 7 to 9 | 200 nodes/w-iLab.t, and east, west, and north sectors of PlanetLab Europe | planetlab-1.imag.fr |

lower *response times* than the GS heuristic. In the SNA-VSP algorithm, where there is high contention of clients connected in each surrogate (*client requests per surrogate*), it is more likely for more clients to request the same (cached) content. Hence, SNA-VSP presents elevated values of *cache hit ratio*. On the other hand, *path length* (Table 3) translated to actual average substrate path lengths measured in meters is smaller for the SNA-VSP (SNA-VSP: 19.29 m, SNA-GVSP: 23.07 m, GS: 26.69 m) due to the actual selection of surrogates (Figure 2) that reduces the *average distance per surrogate*. *CPU load* is correlated to the clients served per surrogate; as the *number of surrogates* decreases (SNA-VSP), each surrogate serves more users thus leading to higher (mean) *CPU load*. On the other hand, the *throughput* is related to *average distance per surrogate*, leading to a higher mean value for the SNA-VSP.

4.2.3 Discussion

Overall, SNA-VSP outperforms all greedy heuristics (SNA-GVSP, GS) in terms of most important metrics,

related to CDN design and deployment (e.g., *CDN deployment cost, mapping cost, number of surrogate servers* used) as well as CDN operation (e.g., *response time, cache hit ratio, throughput*). Results reveal the efficiency of using a more sophisticated server placement algorithm and the proposed formulation. Naturally, the reduced *number of selected surrogate servers* and *path length* reduces the *CDN deployment cost* and the cost of allocating substrate resources for the testbed provider. On the other hand, one must note that adopting SNA features in a surrogate placement scheme contributes to the enhancement of the inherent characteristics of the CDN solution; a fact evident from the comparison of the SNA-VSP and SNA-GVSP to the GS algorithm. The experimental evaluation of the surrogate placement algorithms validates the simulation results obtained by the assessment of the proposed solutions [2] and complements them by demonstrating that they improve the CDN operation.

4.3 Repeatability evaluation

To support the feasibility and operational efficiency validation of the SAViNE framework over w-iLab.t testbed, we need to make sure that the experimental results are repeatable. Repeatability is an expectation that an experiment performed under the same conditions in the same environment produces the same results [26]. Based on the definition provided in [27], the conducted experiments (*offline/real-time design/CDN deployment and operation phase*) measure:

- Temporal repeatability: identical trials (30) were conducted over the span of 1 week in order to

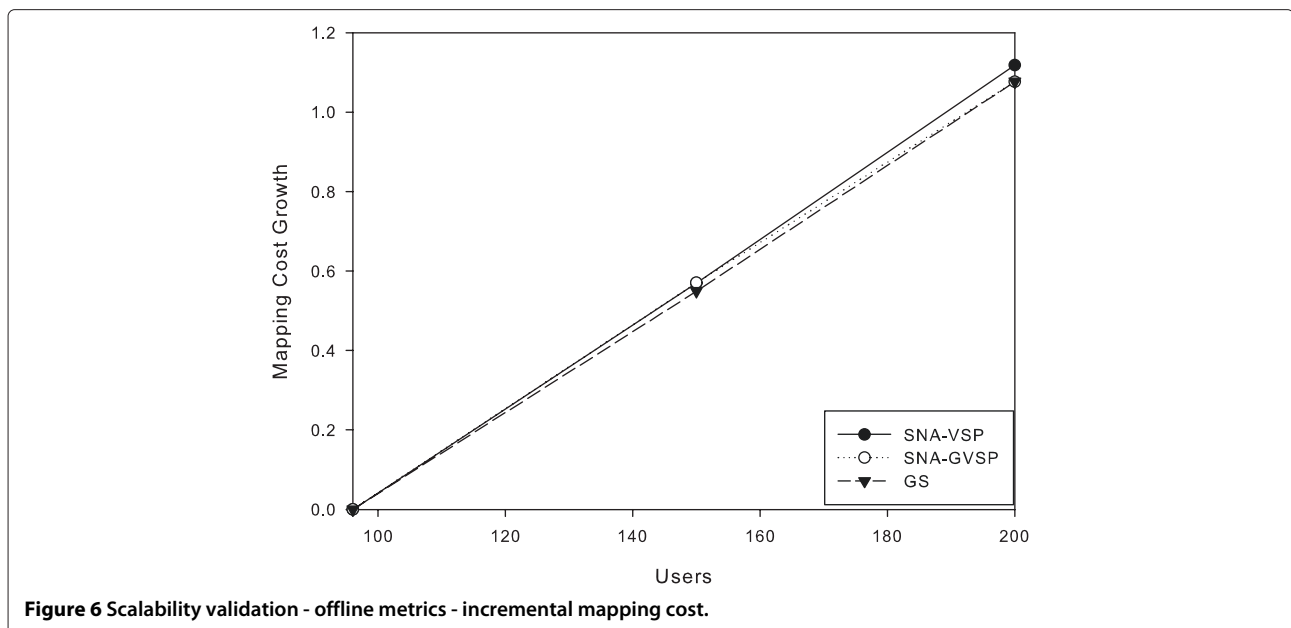
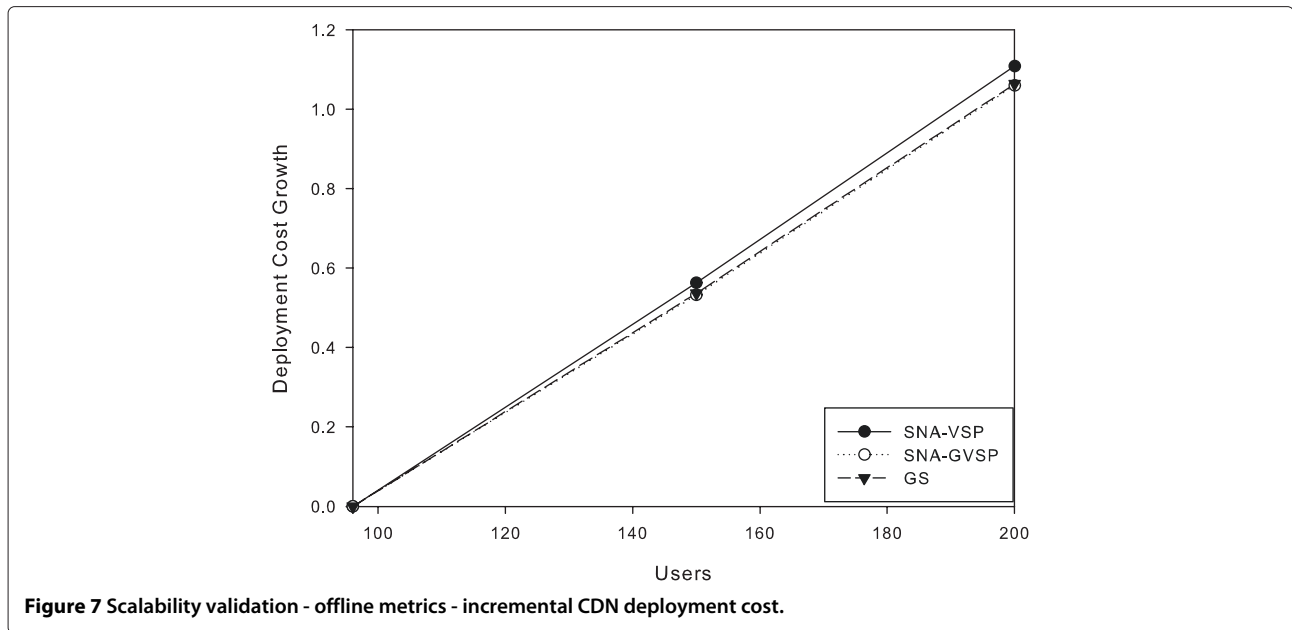


Figure 6 Scalability validation - offline metrics - incremental mapping cost.



capture time variations. For each trial, the same set of nodes and settings were used.

- Spatial repeatability: identical trials were conducted over two symmetric substrate topologies in order to capture possible spatial effects.

The SNA-VSP algorithm has been used for the execution of the experiment trials.

4.3.1 Temporal repeatability

The experiment setup for the Zwijnaarde testbed at w-iLab.t is identical to the one described for feasibility validation. In the following, we report the results for 30 trials chosen out of the duration of a week. The *offline/real-time design* phase results to identical *offline* results and CDN design, hence these are not provided. In order to compare whether the results of the CDN *deployment and operation* phase are repeatable, we adopted one-way ANOVA statistical data analysis, given that the assumptions of normality and homogeneity of variance stand [28]. ANOVA stands for analysis of variance and it is used to compare the average values of more than two independent groups (treatments). The null hypothesis is that the group means are equal. In this particular case, we consider 30 groups, referring to the 30 trials. The test statistic is *F*-test. A low *p*-value for this test indicates the evidence to reject the null hypothesis in favor of the alternative. The critical *p*-value for accepting or rejecting null hypothesis is set to 0.05. Four ANOVA tests are performed, considering the various online metrics as the response variable (average *CPU load*, average *throughput*, average *response time*, and average *SNR*). The

results of the ANOVA tests are presented in Table 5. According to the results, the data provides efficient evidence to conclude that at the 0.05 level of significance, the null hypothesis stands. Therefore, the various trials in time provide results with no statistically significant difference. Hence, the SAViNE experiment results are repeatable over the Zwijnaarde testbed.

4.3.2 Spatial repeatability

In order to define whether different assignment of nodes for different experiment trials produces similar results, two symmetric topologies (substrates I and II illustrated in Figure 3) are selected. Each time, we consider no interference from the residual testbed nodes within proximity.

Table 6 reveals that the algorithms provide similar CDN solutions over symmetric substrate topologies during the *offline/real-time design* phase.

Figure 4 illustrates the graphical representation of the surrogate placement solution, which is the same for the two symmetrical grids. The identical solution provided for the two topologies, including three surrogates in each grid and the same average *path length*, also results in equal mapping and *CDN deployment costs*.

Table 11 Scalability validation - offline metrics - mapping and CDN deployment cost

| Number of users | Mapping cost(%) | | CDN cost (%) | |
|-----------------|-----------------|-------|--------------|-------|
| | SNA-GVSP | GS | SNA-GVSP | GS |
| 96 | 21.77 | 27.36 | 24.30 | 30.39 |
| 150 | 21.70 | 25.73 | 24.25 | 28.12 |
| 200 | 19.38 | 24.87 | 21.12 | 27.60 |

Table 12 Scalability validation - offline metrics - number of surrogate servers

| Number of users | SNA-VSP | SNA-GVSP | GS |
|-----------------|---------|----------|----|
| 96 | 12 | 21 | 24 |
| 150 | 19 | 32 | 36 |
| 200 | 26 | 42 | 46 |

In order to compare whether the results of the *CDN deployment and operation phase* are repeatable, we adopted the (i) two-tailed *z*-test for the *CPU load* and the *throughput* where the sample size is sufficiently large and (ii) the paired *t*-test for the *response time* and the *SNR*, given that the required assumptions stand. With regard to the *CPU load* and the *throughput*, the null hypothesis states that the population means for the two groups (trials) are identical and the difference is due to chance. The results of the *z*-tests with significance level $\alpha = 0.05$ are provided in Table 7. Both estimated *z* values lie within the corresponding range $[-1.96, 1.96]$ ($p > 0.05$), respectively. The null hypotheses cannot be rejected; hence, it is safe to assume that the trials are repeatable. In addition, with regard to the *response time* and *SNR*, the null hypothesis stands. The results of the *t*-tests with significance level $\alpha = 0.05$ are provided in Table 8. Both estimated *t* values lie within the corresponding critical range ($p > 0.05$), respectively; the null hypotheses cannot be rejected. Hence, it is safe to assume that the SAViNE experiment results are repeatable over symmetric topologies at the Zwijnaarde testbed.

4.4 Scalability validation

To conclude on the experimental evaluation of the SAViNE framework, its feasibility is validated over a large-scale experimentation environment. Taking into account the limitation in the number of nodes of the wireless testbeds like the *w-iLab.t* testbed, PlanetLab Europe (PLE) is also utilized.

PlanetLab platform supports an emulation system (*dummynet*) to facilitate reproducible network conditions and wireless emulation [29]. In the context of SAViNE, *dummynet* is used for the purpose of emulating a large-scale wireless environment, with similar characteristics to the *w-iLab.t* testbed. PLE natively supports *dummynet*

Table 13 Scalability validation - online metrics - 96 nodes

| | SNA-VSP | SNA-GVSP | GS |
|------------------------------|---------|----------|------|
| Response time (s) | 2.03 | 2.52 | 3.80 |
| Client request per surrogate | 7.83 | 4.47 | 3.91 |
| Cache hit ratio | 0.50 | 0.37 | 0.28 |
| CPU load (%) | 12.79 | 3.4 | - |

Table 14 Scalability validation - online metrics - 150 nodes

| | SNA-VSP | SNA-GVSP | GS |
|-------------------------------|---------|----------|------|
| Response time (s) | 2.57 | 3.17 | 4.32 |
| Client requests per surrogate | 7.73 | 4.59 | 4.08 |
| Cache hit ratio | 0.52 | 0.38 | 0.31 |
| CPU load (%) | 12.43 | 3.0 | - |

as a kernel module in all nodes, configurable from the sliver through a command-line tool. A subset of *offline* and *online* metrics are measured, for a varying set of CDN end users (100 to 200 at a step of 50). The area that PlanetLab Europe spans is split in three sectors as shown in Figure 5. Specifically, east, west, and north sectors are formed, each having approximately 50 nodes. Table 9 provides the countries and the number of nodes that each sector contains. We assume that the nodes in each sector have a connectivity degree of 30% to create the emulated multi-hop experimentation environment that matches the connectivity of the *w-iLab.t* testbed.

The same input parameters (e.g., end-user request patterns, etc.) as in the case of *feasibility validation* were used in the particular set of trials. Furthermore, Table 10 provides additional parameters, specific to each trial (nine in total).

4.4.1 *w-iLab.t/PlanetLab* offline metrics

Following the execution of the various algorithms, a subset of the *offline* results are presented. Specifically, the incremental *CDN deployment cost* and *mapping cost* are depicted in Figures 6 and 7. The CDN deployment cost and mapping cost results are presented as the relative percentage differences of the SNA-GVSP and GS values from the SNA-VSP ones, in Table 11. Moreover, the *number of surrogates* is presented in Table 12.

As it is expected, the three algorithms show a linear growth of associated costs with the number of users due to the symmetrical experimental sectors (similar number of nodes and connectivity) involved (Figures 6 and 7). The deviation in the cost growth among the algorithms depends on the underlying physical topology. Following the trend of *feasibility validation*, the SNA-VSP cuts down CDN deployment cost (Table 11) primarily by reducing the *number of surrogate servers* (Table 12).

Table 15 Scalability validation - online metrics - 200 nodes

| | SNA-VSP | SNA-GVSP | GS |
|-------------------------------|---------|----------|------|
| Response time (s) | 2.66 | 3.28 | 4.40 |
| Client requests per surrogate | 7.57 | 4.66 | 4.17 |
| Cache hit ratio | 0.52 | 0.38 | 0.31 |
| CPU load (%) | 11.52 | 3.2 | - |

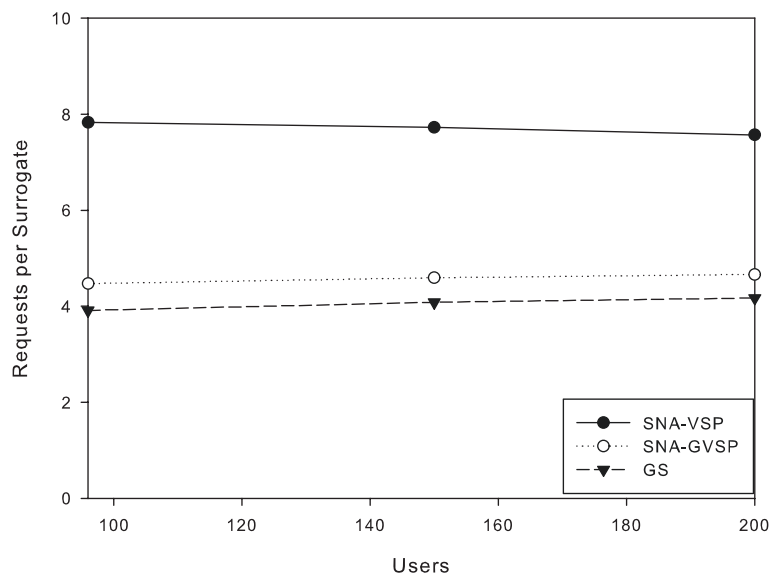


Figure 8 Scalability validation - online metrics - incremental client requests per surrogate.

4.4.2 w-iLab.t/PlanetLab online metrics

The produced CDN topology is deployed in the w-iLab.t/PlanetLab testbeds. The *online metrics* for the experiment trials conducted are illustrated in Tables 13, 14, and 15. The time average of the mean *CPU load* for the different algorithms is estimated over a time period of 4 min which is the average duration of the content being served. *CPU load* is presented as the relative percentage difference of the SNA-VSP and SNA-GVSP from the GS values.

The SNA-VSP algorithm presents a high contention of clients connected in each surrogate (*client requests per surrogate* - Tables 13, 14, and 15) and a smaller number of selected surrogates (Table 12). However, the incremental *client requests per surrogate* (Figure 8) for all algorithms present an almost constant behavior due to the linear increase of surrogates with the number of users, indicating the scalability of the proposed experiment. Moreover, when there is high contention of clients connected in each surrogate, it is more likely for more clients to request

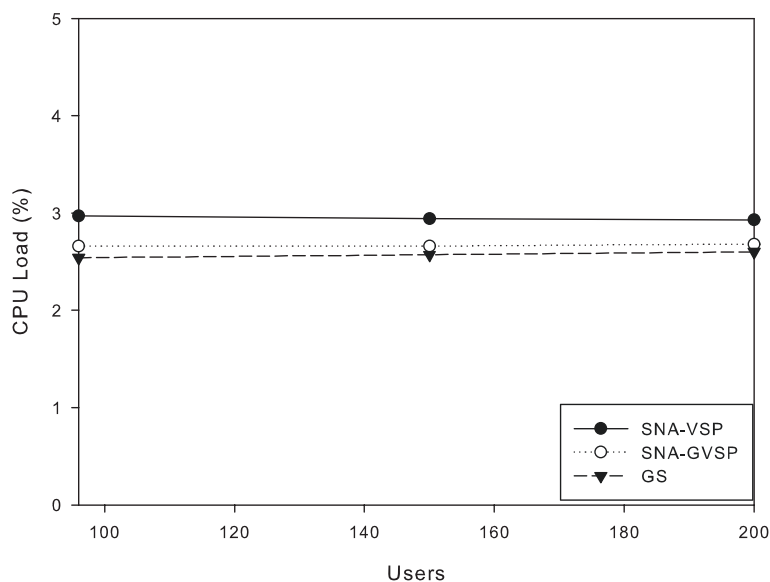


Figure 9 Scalability validation - online metrics - CPU load.

the same (cached) content. Hence, *cache hit ratio* follows a similar trend to the requests per surrogate metric, between the trials for the various algorithms.

The *CPU load*, as in the case of the single testbed, is correlated to the number of surrogate servers and clients served per surrogate. As the *number of surrogate servers* decreases (SNA-VSP), each surrogate serves more users thus leading to higher *CPU load* (Tables 13, 14, and 15). Thus, SNA-VSP presents higher *CPU load* than the other two algorithms. Moreover, *CPU load* remains approximately stable with regard to the number of users (Figure 9) since a topology with similar characteristics is incrementally added to the one used in each experiment trial.

Response time is lower in the case of SNA-VSP and SNA-GVSP than the GS heuristic. Moreover, the mean values are related to CDN topology obtained, based on the design goals set (e.g. maximum one hop routing distance between the client and the surrogate, span of the geographical experimentation area) - hence, a small increase in the response time is noticed as the experimentation area grows.

4.4.3 Discussion

Overall, the SAViNE performance evaluation over the hybrid w-iLab.t/PlanetLab topology reveals the scalability of the proposed SNA-VSP heuristic (*offline metrics*), witnessed by the acquired CDN solution (*online metrics*). SNA-VSP manages to cut down on the deployment cost for deploying a CDN in comparison to SNA-GVSP and GS while the particular metric scales linearly with the number of users. In addition, the operational characteristics of the resulting CDN designs provide sufficient evidence that SNA-VSP manages to select efficiently the appropriate surrogates that will serve the CDN users, enhancing the operational efficiency of the CDN.

5 Conclusion

The SAViNE framework for wireless content delivery has been integrated with a selected set of FIRE experimental facilities, within the OpenLab initiative, providing a runtime environment for CDN deployment, operation, and performance evaluation. The main goal of this study is to experimentally validate its feasibility and operational efficiency over the w-iLab.t testbed. Complementary to the *feasibility validation*, SAViNE is evaluated over a large-scale wireless environment (*scalability validation*) comprised of the heterogeneous w-iLab.t/PlanetLab platforms. Moreover, experimental results at the indoor wireless environment at Zwijnaarde testbed/w-iLab.t prove to be repeatable (*repeatability evaluation*), both with regard to the time variations (temporal repeatability) and spatial effects (spatial repeatability). The involved testbeds (w-iLab.t and PLE) provided the means to validate the simulation results obtained on the assessment of the proposed

surrogate placement algorithm, on the initial theoretical study [2].

Endnote

^aAlso denoted as edge servers or surrogates.

Competing interests

The authors declare that they have no competing interests.

Acknowledgements

This work has been partially supported by the European Community Seventh Framework Programme (FP7), OpenLab project (INFSO-ICT-287581), and Fed4FIRE project (INFSO-ICT-318389).

Received: 28 February 2014 Accepted: 6 November 2014

Published: 24 November 2014

References

1. A-MK Pathan, R Buyya, in *Content Delivery Networks, Lecture Notes in Electrical Engineering*. A taxonomy of cdns, vol. 9 (Springer Berlin Heidelberg, 2008), pp. 33–78
2. C Papagianni, A Leivadeas, S Papavassiliou, A cloud-oriented content delivery network paradigm: Modeling and assessment. *IEEE Trans. Dependable Secure Comput.* **10**, 287–300 (2013)
3. Y Jin, Y Wen, G Shi, G Wang, A Vasilakos, in *Proceeding of Computing, Networking and Communications (ICNC): 934-938 Feb 2012*. CoDaaS: An experiment cloud-centric content delivery platform for user-generated contents (Maui, 2012), pp. 934–938
4. GMH Yin, X Liu, C Lin, Content delivery networks: a bridge between emerging applications and future ip networks. *Netw. IEEE.* **24**(4), 52–56 (2010). doi:10.1109/MNET.2010.5510919
5. L Qiu, VN Padmanabhan, GM Voelker, in *Proceedings of IEEE Infocom: 1587-1596 April 2001*. On the placement of web server replicas (Alaska, 2012), pp. 433–441
6. E Cronin, S Jamin, C Jin, AR Kurc, D Rax, Y Shavitt, Constraint mirror placement on the internet. *IEEE J. Select. Areas Commun.* **7**, 31–40 (2002)
7. X Jia, D Li, K Hu, W Wu, D Du, Placement of web-server proxies with consideration of read and update operations. *Comput. J.* **46**, 378–390 (2003)
8. T Bektas, I Ouveysi, R Buyya, M Pathan, A Vakali, Mathematical models for resource management and allocation in cDNs. *Lecture Notes Electrical Eng. Content Deliv. Netw.* **9**, 225–250 (2008)
9. F Chen, K Guo, J Lin, TF La Porta, in *Proceedings of IEEE Infocom: 433-441 March 2012*. Intra-cloud lightning: Building CDNs in the cloud (Orlando, 2012), pp. 433–441
10. J Broberg, R Buyya, Z Tari, MetaCDN: Harnessing storage clouds for high performance content delivery. *J. Netw. Comput. Appl.* **32**, 1012–1022 (2009)
11. W-iLab.t Wireless Testbed. <http://www.crew-project.eu/portal/wilabdoc>.
12. PlanetLab Testbed. <http://www.planet-lab.eu/>.
13. OpenLab Integrating Project. <http://www.ict-openlab.eu/>.
14. Fed4FIRE Integrating Project. <http://www.fed4fire.eu/>.
15. FIRE, Future Internet Research and Experimentation. <http://www.ict-fire.eu>.
16. TA Neves, LMA Drummond, LS Ochi, C Albuquerque, E Uchoa, Solving replica placement and request distribution in content distribution networks. *Electron. Notes Discrete Math.* **36**, 89–96 (2010)
17. MEJ Newman, *Networks: An Introduction*. (Oxford University Press, UK, 2010)
18. D Katsaros, N Dimokas, L Tassioulas, Social network analysis concepts in the design of wireless ad hoc network protocols. *IEEE Netw.* **24**, 23–29 (2010)
19. M Chowdhury, MR Rahman, R Boutaba, Vineyard: Virtual network embedding algorithms with coordinated node and link mapping. *Netw. IEEE/ACM Trans.* **20**(1), 206–219 (2012). doi:10.1109/TNET.2011.2159308
20. K Kalpakis, K Dasgupta, O Wolfson, Optimal placement of replicas in trees with read, write, and storage costs. *IEEE Trans. Parallel Distributed Syst.* **12**, 628–637 (2001)
21. P Radoslavov, R Govindan, D Estrin, Topology-informed internet replica placement. *Comput. Commun.* **25**(4), 384–392 (2002). doi:10.1016/S0140-3664(01)00410-8

22. C Papagianni, A Leivadeas, S Papavassiliou, V Maglaris, C Cervello-Pastor, A Monje, On the optimal allocation of virtual resources in cloud computing networks. *IEEE Trans. Comput.* **62**, 1060–1071 (2013)
23. GENI v3 RSpecs. <http://www.geni.net/resources/rspec/3>
24. A Falaschi, D Monster, I Dolezal, M Krsek, in *Proceeding of Terena Networking Conference: 1-11 June 2004*. Academic Streaming in Europe: Report on TF-Netcast (Rhodes, 2004), pp. 1–11
25. S Boukaert, B Jooris, P Becue, I Moerman, P Demeester, The IBBT w-ilab.t: a large-scale generic experimentation facility for heterogeneous wireless networks. *Testbeds Res. Infrastructure. Dev. Netw. Commun.* **44**, 7–8 (2012)
26. ASTM E177 - 13 Standard Practice for Use of the Terms Precision and Bias in ASTM Test Methods, *Annual Report*. (ASTM International, West Conshohocken, PA, 2013). www.astm.org
27. S Ganu, H Kremos, R Howard, I Seskar, in *Proceeding of IEEE Tridentcom: 153-160 Feb 2005*. Addressing repeatability in wireless experiments using ORBIT testbed (Trento, 2005), pp. 153–160
28. JA Rice, *Mathematical Statistics and Data Analysis*. (Wadsworth, Belmont, 2007)
29. M Carbone, L Rizzio, An emulation tool for PlanetLab. *Comput. Commun.* **34**, 1980–1990 (2011)

doi:10.1186/1687-1499-2014-198

Cite this article as: Leivadeas et al.: SAViNE: social network analysis - inspired content delivery network deployment and experimentation. *EURASIP Journal on Wireless Communications and Networking* 2014 **2014**:198.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com
