

Multi-criteria decision algorithms for efficient content delivery in content networks

Andrzej Bęben · Jordi Mongay Batalla ·
Wei Koong Chai · Jarosław Śliwiński

Received: 19 December 2011 / Accepted: 25 July 2012 / Published online: 29 August 2012
© The Author(s) 2012. This article is published with open access at Springerlink.com

Abstract Today's Internet is prominently used for content distribution. Various platforms such as content delivery networks (CDNs) have become an integral part of the digital content ecosystem. Most recently, the information-centric networking (ICN) paradigm proposes the adoption of native content naming for secure and efficient content delivery. This further enhances the flexibility of content access where a content request can be served by *any* source within the Internet. In this paper, we propose and evaluate a multi-criteria decision algorithm for efficient content delivery applicable for content networks in general (among others, CDN and ICN). Our algorithm computes the best available source and path for serving content requests taking into account information about content transfer requirements, location of the consumer, location of available content servers, content server load and content delivery paths between content servers and consumer. The proposed algorithm exploits two closely related processes. The first level discovers multiple content delivery paths and gathers their respective transfer characteristics. This discovery process

is based on long-term network measurements and performed offline. The second process is invoked for each content request to find the best combined content server and delivery path. The cooperation between both levels allows our algorithm to increase the number of satisfied content requests thanks to efficient utilisation of network and server resources. The proposed decision algorithm was evaluated by simulation using Internet scale network model. The results confirm the effectiveness gain of content network architectures that introduce network awareness. Moreover, the simulation process allows for a comparison between different routing algorithms and, especially, between single and multipath routing algorithms.

Keywords Multi-criteria decision algorithms · Content networks · Future internet

1 Introduction

With massive volume of content being accessed over the Internet every day, content networks such as content delivery networks (CDNs) have flourished. Recently, the information-centric networking (ICN) paradigm has received widespread attention with various initiatives targeting the area (e.g., DONA [1], CCN/NDN [2, 3], 4WARD/SAIL [4, 5], PSIRP/PURSUIT [6, 7], and COMET [8, 9]). It has been advocated as the cure for many ills of today's host-centric content distribution Internet (e.g., CDNs [10]). The proposition is that, nowadays, the Internet is no longer used for simple resource sharing but rather for sophisticated content access and dissemination. The current simplistic unicast end-to-end communication model is neither compatible nor efficient for the new generation of Internet applications and services which often requires one-to-many or many-to-many communication mode (e.g., dissemination

The research leading to these results has received funding from the EU FP7 COMET project—Content Mediator Architecture for Content-Aware Networks, under Grant Agreement 248784

A. Bęben (✉) · J. M. Batalla · J. Śliwiński
Warsaw University of Technology,
Warsaw, Poland
e-mail: abeben@tele.pw.edu.pl

J. M. Batalla
e-mail: jordim@tele.pw.edu.pl

J. Śliwiński
e-mail: j.sliwinski@tele.pw.edu.pl

W. K. Chai
Department of Electronic and Electrical Engineering,
University College London,
London, UK
e-mail: wchai@ee.ucl.ac.uk

of popular content, spreading of information in online social networks etc.). Among various research areas of content networks (e.g., naming, in-path caching [11], security, mobility etc.), we focus, in this paper, on the general problem of content selection which relates to the content resolution process and the increased flexibility in the communication models. In the broader context, our work is applicable to general content distribution problem where multiple copies of the same content are hosted in different (geographical) sites (e.g., via surrogate servers). Following the proliferation of various content, including user-generated content, content (or sometimes server) replication has already been included in the repertoire of network services in recent content dissemination architectures such as ICN, CDNs and even peer-to-peer (P2P) networks for scalable and fault tolerant content access and diffusion.

A central question with multiple available content sources across the Internet that has various and possibly dynamic traffic conditions and network capabilities is how to find and select the best available content source to satisfy a content request and guarantee efficient resource use. The current literature in content network seems to use the simplest of metric to decide on the best available server. For instance, [1] uses its anycast primitive which implicitly selects the content source with the lowest domain-level hop count. In [2], the first content found by the content request (i.e., the interest packet) is treated as the best option and being retrieved following the reverse direction. Such approaches hardly guarantee or even reflect the true quality of the content delivery path.

In this paper, we study the problem of multi-source content resolution within a content distribution platform. We argue that (1) hop count alone is not an adequate performance metric for deciding a content source and (2) for Internet-wide server selection, it is not scalable to both resolve content requests and construct the delivery paths on-the-fly (i.e., per request computation). Thus, we investigate a two-phase approach where, in the first phase, a foundation is built on the available delivery paths with their corresponding capabilities and quality of service (QoS) dimensioning parameters to facilitate the server selection decision when the actual request is sent. In the second phase, we develop advanced multi-criteria decision algorithms that use both the information-base already built in phase-1 and dynamic information about the state of the network and the server in order to efficiently find the best content source and/or path to satisfy a content request. Note that our methodology is extensible to accommodate new performance metrics.

Besides contributions on the decision algorithm design, our results offer to the reader a clearer comprehension of the efficiency gain of content network compared to systems without network awareness as the current Internet. Even

when the conclusions are obtained on a comparison basis, the obtained results offer comprehensible information about, among others, the convenience of introducing multipath routing, the importance of using short-term scale information and the influence of different network parameters on the efficiency gain of the system.

The organisation of the paper is the following. In Section 2, we present related works focusing on server selection methods used by content networks. Moreover, we introduce the multi-criteria optimization methods which constitute a base for our algorithm. In Section 3, we present details of our two-phase decision approach. The results of performance evaluation are included in Sections 4 and 5. These sections describe simulation model proposed for evaluation of content networks as well as the description of the experiments, which evaluate the effectiveness of our approach comparing to others. Finally, Section 6 summarises the paper and give outlines on further works.

2 Related works and background

In general, server selection decisions can be made on the client-side (e.g., probing, statistical estimator), server-side (e.g., server push) or by the network infrastructure (e.g., DNS, anycast-enabled routers). An empirical evaluation on client-side server selection algorithms in [12] has found that simple dynamic probing outperforms other common client-side approaches. However, it is worth to say that individual client probing does not scale in an Internet-wide information-centric setting. Server selection in the light of anycast has also appeared as a formidable engineering problem mainly due to susceptible scalability. Most work in this area can be categorized into application and network-layer solutions and majority of them are restricted to specific context such as web service [13] or wireless ad hoc networks [14]. Most relevant to our work is [15] where a global IP-anycast framework is proposed and claimed to be scalable for several millions of global anycast groups. However, in the context of content networks, we are dealing with much finer granularity (measured in terms of number of content rather than groups of servers). The scalability requirements in such scenario indicated the need for some offline pre-computation as a preparation to serve content requests in real time.

By using network and server information in the server and path selection process, the system gains in effectiveness, which is reflected in better quality of experience, improves system resource utilisation and improves the load balancing within the network and between servers. Nevertheless, the selection of server and path on the basis of different information is a complex multi-criteria optimization problem [16, 17]. The basic model of multi-criteria optimization defines

the decision space \mathfrak{X}^i which consists of the decision vectors $x=(x_1, x_2, \dots, x_i)$. Each decision vector contains i decision variables. Any decision variable may have bounded amount of feasible solutions by given constraints. Therefore, the space of decision vectors may also be bounded. Multi-criteria optimization focuses of optimizing a set of k objective functions $\Pi_1(x), \Pi_2(x), \dots, \Pi_k(x)$ which can either be minimized or maximized.¹ The aggregate objective function is simply a vector of these objective functions: $\Pi(x) = (\Pi_1(x), \Pi_2(x), \dots, \Pi_k(x))$ [18]. For each decision vector $x \in \mathfrak{X}$, exists one unique objective vector $y \in \mathfrak{Y}$, where $\Pi: \mathfrak{X} \rightarrow \mathfrak{Y}$ with, $y = (y_1, y_2, \dots, y_k) = \Pi(x) = (\Pi_1(x), \Pi_2(x), \dots, \Pi_k(x))$. In multi-criteria optimization, a solution x'' dominates the solution x' if and only if $\forall k^* \in \{1, \dots, k\} : \Pi_{k^*}(x'') \leq \Pi_{k^*}(x')$ and $\exists k^- \in \{1, \dots, k\} : \Pi_{k^-}(x'') < \Pi_{k^-}(x')$ and a solution x' is called efficient if and only if there is not another solution x'' , which dominates x' . The set of efficient solutions is the Pareto optimal set and the set of all outcome vectors y resulting from $y = \Pi(x)$ where x is an efficient solution, is the Pareto Frontier.

Multi-criteria optimization problems generally are NP-complete. Therefore, one commonly investigated approach is to apply heuristics methods. The simplest heuristic method converts the k -dimensional vector of weights into a single scalar value w by using an appropriate cost function $f(\cdot)$. In this way, the multi-criteria problem is reduced to a single criterion problem, which simplifies the solution computation. On the other hand, this simple heuristic does not guarantee that selected server and path will be the most effective ones since the scalar value w losses information about particular constraints. The recently investigated multi-criteria decision algorithms, presented in [19] and [20], introduce a reference point to rank the Pareto optimal set. By using this method, the aggregate objective function finds the effective solutions of the Pareto optimal set, which are nearest from the given reference solution (generally the reference solution may not be in the Pareto optimal set). As an example, we can think of an algorithm, which prefers balanced solutions (medium server load and medium path load) over solutions with extreme values (low server load and high path load, or high server load and low path load).

3 Proposed multi-criteria decision algorithm

In order to ensure effective content delivery in multi-source environment, we maximize the information used in the server/path selection process by introducing the decision

algorithm at two levels. The first level corresponds to routing process in the network which discovers content delivery paths between domains. For the rest of the paper, we treat a domain as equivalent to an Autonomous System (AS). The routing process is performed offline using long-term information about network topology, Classes of Service offered by individual domain and the corresponding provisioned resources. The outcome of this process is a set of end-to-end content delivery paths between servers' and customers' domains, which are established to meet content transfer requirements of different types of content. The second level of decision process is invoked for each content consumption request. This process selects the content server and content delivery path that are actually used for the delivery of the requested content. Note that decision algorithms operate at different time scales and use different information.

3.1 Decision algorithm at routing level

Current Internet relies on BGP-4 shortest path routing protocol, which establish a single routing path between any two domains. Such approach limits the effectiveness of content delivery because: (1) the network transfers the content regardless of its transfer requirements, what generally leads to degradation of the quality experienced by consumers, and (2) downloading of popular content may provoke network congestion, since single routing path going from content server's to customers' domains may become congested.

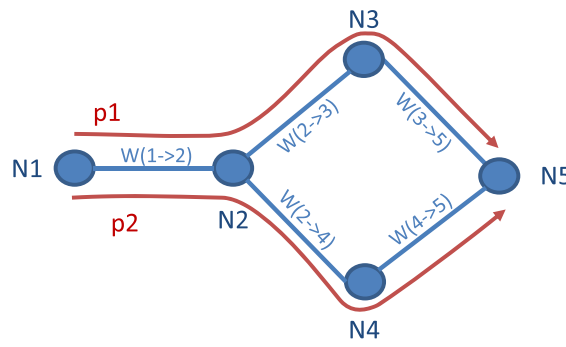
Exploiting the fact that content network aims to create a new network architecture, one may go beyond the above limitations and use multi-constraints and multipath routing. Multi-constraints routing selects paths that satisfy a given set of constraints [21, 22]. Let us consider the network as a directed graph $G(N, E)$, where N represents the set of domains, while E is the set of links. Each link $u \rightarrow v$, $u, v \in N$, $u \rightarrow v \in E$, is characterised by m -dimensional vector of non-negative link weights $w(u \rightarrow v) = [w_1, w_2, \dots, w_m]$. Any path p between two nodes is characterised by vector of path weights $w(p) = [w_1^{(p)}, w_2^{(p)}, \dots, w_m^{(p)}]$, where $w_i^{(p)}$ is calculated as a concatenation of the link weights w_i of each link belonging to path p . The multi-constraints routing finds a set of feasible paths, $f \in F$, going from server domain to consumer domain. The path is feasible if its path weights satisfy the assumed constraints: $w_i^{(f)} < l_i$, $i = 1, \dots, m$, where L is given vector of constraints $L = [l_1, l_2, \dots, l_m]$.

Figure 1 presents an example on how the proposed protocol establishes paths. Domain N1 has two alternative paths (i.e., p1 and p2) going towards domain N5. Each path is characterised by two-dimensional vector of weights. The path weights are calculated as concatenation of links weights.

The multi-constraints routing belongs to the class of multi-criteria optimization problems that are in principle

¹ The problem does not lose generality by the fact that we consider uniquely minimization.

Fig. 1 The illustration of multipath routing with two paths between domains N1 and N5



$$w(p1)=[w_1^{(p1)}, w_2^{(p1)}]$$

$$w_1^{(p1)}=w_1^{(1\rightarrow2)} + w_1^{(2\rightarrow3)} + w_1^{(3\rightarrow5)}$$

$$w_2^{(p1)}=w_2^{(1\rightarrow2)} + w_2^{(2\rightarrow3)} + w_2^{(3\rightarrow5)}$$

$$w(p2)=[w_1^{(p2)}, w_2^{(p2)}]$$

$$w_1^{(p2)}=w_1^{(1\rightarrow2)} + w_1^{(2\rightarrow4)} + w_1^{(4\rightarrow5)}$$

$$w_2^{(p2)}=w_2^{(1\rightarrow2)} + w_2^{(2\rightarrow4)} + w_2^{(4\rightarrow5)}$$

NP-complete. Although different methods for multi-criteria optimization have been proposed, they cannot be directly used in routing protocols because most of them assume independent decision makers operating on its own decision space. On the other hand, routing protocols create paths in distributed manner where decision space of a given domain is determined by information propagated by neighbouring domains. As a consequence, the end-to-end path is a result of the sequence of decisions taken by the involved domains based on their knowledge about paths, local preferences and assumed constraints (local decisions in-path vector routing protocol).

The proposed multi-constraints and multipath routing protocol creates a set of content delivery paths between server and consumer domains with respect to content QoS requirements. Our protocol follows path vector principle, where each domain advertises its preferred paths to its neighbours. Each path is described by a list of ASes and the corresponding vector of path weights $w(p)$ calculated as concatenation of links weights $w(u \rightarrow v)$. These characteristics allow routing entity to eliminate routing loops and remove unfeasible paths. Furthermore, routing entity removes dominated paths, i.e., paths for which exists another path with all weights $w_i^{(p)}$ lower than weights of paths in question. Remaining paths form a set of preferred paths. In order to achieve optimum solution, all preferred path should be advertised to neighbouring domains. However, for scalability reasons, the number of advertised paths must be limited to some reasonable value. Based on performed experiments, we recommend limiting the number of advertised paths to 3–5 paths. The smaller value limits the gain achieved by multipath routing. On the other hand, dissemination of more than five paths only slightly improve effectiveness but significantly increases the size of routing tables. The key problem in proposed heuristics is to choose the right paths. We believe that selection algorithms should prefer paths with larger distance between weight $w(f)$ and constraint L . In this way, adjacent domains receiving those paths will have a greater chance of finding feasible paths. Therefore, we rank preferred paths using a cost function, $cost_f(\cdot)$, which takes as arguments the vectors of path

weight $w(f)$ and constraint L . Following this ranking, the routing protocol advertises k paths of the lowest cost. Although different functions could be applied, the studies presented in [21, 23–25] point out that the most effective are nonlinear, strict monotonic and convex functions. In our routing protocol, we use Minkowski norm of order r , defined as (1).

$$cost_f(\cdot) = \begin{cases} \left(\sum_{i=1}^m \left(\frac{w_i}{l_i} \right)^r \right)^{\frac{1}{r}}, & w_i \leq l_i \\ \infty, & w_i > l_i \end{cases} \quad (1)$$

This function ranks preferred paths based on the distance of normalized weights $w_i^{(p)}$ from the point zero in m -dimensional decision space. Appropriate tuning of parameter r allows us to influence the shape of cost function.

For r equal to 1, the path cost is linear combination of normalized path weights. Although this function can be easily interpreted, it is insensitive to unbalanced solutions, with extreme weights close to the constraint. Let us consider two exemplary feasible paths f^1 and f^2 , with normalized weights $w(f^1)=[0.4,0.4]$ and $w(f^2)=[0.7, 0.1]$. The costs of both paths equal 0.8, while the probability of exceeding constraints is much higher for path f^2 , because its first component is close to the constraint.

For $r \rightarrow \infty$, the cost is determined by the maximum component of the path's weight $w_i^{(f)}$, while the rest of the components are ignored. So, the cost of two exemplary paths f^3 and f^4 , with normalized weights $w(f^3)=[0.8,0.1]$ and $w(f^4)=[0.8, 0.7]$ equals 0.8, while the probability of exceeding constraints is higher for path f^4 .

In our algorithm, we assumed cost functions with r equal to 4, which is large enough to guarantee sensitiveness to unbalanced solution and it is still low enough to consider impact of all weights.

3.2 Decision algorithm at content consumption level

The second level of decision process selects the best server and path from the available candidates computed in the previous phase. This process is performed, independently

in each consumer domain, upon receiving consumption request in an entity called decision maker. The decision maker is responsible for the selection of the best content server and path based on collected information about candidate servers and paths. Basically, the location of decision maker depends on the architecture of the specific content network. For example, the decision maker can be served by Resolution Handlers defined in DONA [1], Content Mediation Entity designed in COMET [8] or Request Routing Entities used in CDNs [10].

In our algorithm, we consider: (1) the list of content servers that may stream the content and their load, and (2) the list of content delivery paths between particular server and consumer. Each path is characterised by path length, load on the path and QoS parameters. The complex set of parameters used for decision algorithm requires a multi-criteria decision algorithm. However, in contrast to the routing level process, there is no direct influence of one decision maker on another because decision spaces are independent. As a consequence, we may directly apply one of algorithms studied in the Multiple Criteria Decision Analysis [17, 19, 26]. We leverage the multi-criteria decision algorithm presented in [20], which uses some a priori knowledge about the problem in order to select the effective solution. Without loss of generality, our decision algorithm uses three decision variables related to server load, path length and bandwidth. It could be easily extended to accommodate more decision variables. Our algorithm evaluates the impact of particular decision variable using two reference parameters, called reservation level and aspiration level. The reservation level is the hard upper limit for decision variable which should not be crossed by preferred solution. On the other hand, the aspiration level defines the lower bound for decision variable, beyond which preference of evaluated solutions is similar. The decision algorithm consists of three main steps:

Step 1: decision maker creates a decision space, which consists of the list of candidate solutions based on the information about servers and paths. Each candidate solution is a vector of decision variables, which has the following form:

Candidate solution [i]:

serverLoad—this is numerical representation of server status,

pathLength—the path length denotes the number of domains on the path. It is calculated based on the AS path parameter,

bandwidth—maximum supported bandwidth on the path.

Step 2: decision maker calculates the rank value R_i for each candidate using objective function with

reservation and aspiration levels specific for each decision variable.

$$R_i(\cdot) = \min_{k=1,2,3} \left[\frac{r_k - q_k}{r_k - a_k} \right] \quad (2)$$

where: i is the number of candidate, k is the number of decision variable, q_k is current value of decision variable, r_k is a reservation level for decision variable k , while a_k is an aspiration level for decision variable k , which is determined as a multiplication of reservation level by aspiration coefficient α_k , $a_k = \alpha_k \times r_k$.

Step 3: decision maker selects the candidate with maximum rank as the best solution. Note that considered aggregate objective functions may have more than one effective solution into the Pareto optimal set. Thus, some tie-breaking rules (e.g., lower server load is preferred or just random selection) are required to ensure only one solution is selected.

The network operator may tune the behaviour of the decision algorithm by setting reservation and aspiration levels. In particular, he may reduce the importance of given decision variable by setting the aspiration level close to the reservation or even completely exclude given decision variable by setting the aspiration level equal to the reservation level. In this way, each domain may define its own decision strategy based on its own preferences. Below, we discuss exemplary decision strategies which can be enforced by our proposed algorithm.

Strategy 1: random server. This strategy assumes that content server is selected randomly. It reflects situation in the current Internet, where no information about paths and servers is available.

Strategy 2: closest server. This algorithm selects the server which is closest to the user. This approach reflects one of the strategies in the current CDNs [45], when information about server status is not available. In this case, decision algorithm should consider only *path length* (aspiration and reservation levels for *server load* and *bandwidth* are set to 1). In fact, in content network-related literature, this simplistic approach is used in [1] and [2].

Strategy 3: the least loaded server (called best server strategy hereafter). This algorithm considers only the server load. It selects the least loaded server without considering information about paths. This strategy is used by most of P2P content delivery systems as well as some CDNs [45]. In this case, decision algorithm should consider only *server load* (aspiration and reservation levels for *path length* and *bandwidth* are set to 1).

Strategy 4: the best server and path. This algorithm considers both the *server load* and available bandwidth in the bottleneck link. It selects the least loaded server with the path of the best characteristics.

The presented above strategies will be evaluated in Section 5. Note, that proposed decision algorithm allows defining also other decision strategies based on the subset of the available parameters.

4 Validation of decision algorithms in dynamic environment

The combination of both the decision algorithms at routing and at content consumption levels makes feasible, among others, load balancing in servers and network. Let us consider a simple scenario as presented in Fig. 2a. Consumers in domain D1 generate requests of the content C1, which can be found in both domain D3 and D4. The request arrival process is Poisson with mean rate=1.0 req/s. At time $t=500.0$ s, consumers in domain D2 start requesting (Poisson) for content C2 (located uniquely in a server in D4) with the same mean rate=1.0 req/s. The parameters of the simulation are given in Fig. 2b. The selected parameters are simplistic in order to understand the routing reaction of the algorithms to non-stationary congestion phenomena. More sophisticated simulations are presented in Section 5 for evaluating the performance in realistic scenario.

We consider multipath routing decision algorithm for two paths. So, the consumers in D1 can get the content C1 situated at D3 by: p1 or p2 paths, whereas the content C1 situated at D4 can be transferred by p3 or p4 paths, as indicated in Fig. 2a. When the simulator allots a new request, the streaming server and path for serving the request are selected by considering both strategies: (1) random server and random path and (2) server load (reservation and aspiration levels are $r_{server}=1.0$ and $a_{server}=0$, respectively) and current available bandwidth in bottleneck link ($r_{bottleneck}=100$ Mbps; $a_{bottleneck}=0$ Mbps), where the

bottleneck links are L1, L2 or L3, see Fig. 2a. The server load is the relation between current served connections and CPU capacity of the server (100 connections served in parallel). The appropriate server is loaded by one more connection and the occupied bandwidth in the corresponding bottleneck link is increased by the streaming bandwidth of the content (100 kbps for both the contents), during the streaming duration of the content (100 s for both the contents).

Figure 3 shows the occupied bandwidth in the bottlenecks L1, L2 and L3 (left Y-axis of the figure) as well as the load of the two servers at D3 and D4 (right Y-axis) over the simulated period. Figure 3a considers the random server and random path strategy. We observe that bottleneck L1 and server at D3 do not depend on the requested service in domain D2. So, the selected strategy does not adapt to the request process and no load balancing is achieved neither in bottlenecks nor in servers.

The second strategy, presented in Fig. 3b, shows load balancing in network and servers thanks to our proposed decision algorithm. Note that the load in all links is the same regardless of content request calls. The same occurs for the server load in both the servers (in D3 and D4). When customers in D2 begin requesting content, the links L2 and L3 as well as server in D4 start over-loading. So the new requests from D1 “prefer” the link L1 and the server in D3 achieving, in this way, balancing of bottlenecks and servers.

We can conclude that the decision algorithm positively reacts to the non-stationary congestion phenomena of the system, i.e., it may adapt to changing conditions.

5 Performance evaluation

In this section, we focus on the performance evaluation of our proposed multi-criteria decision algorithm, which selects the best content source. We compare our algorithm with other methods used in content networks in the current Internet. This evaluation is performed in an exhaustive

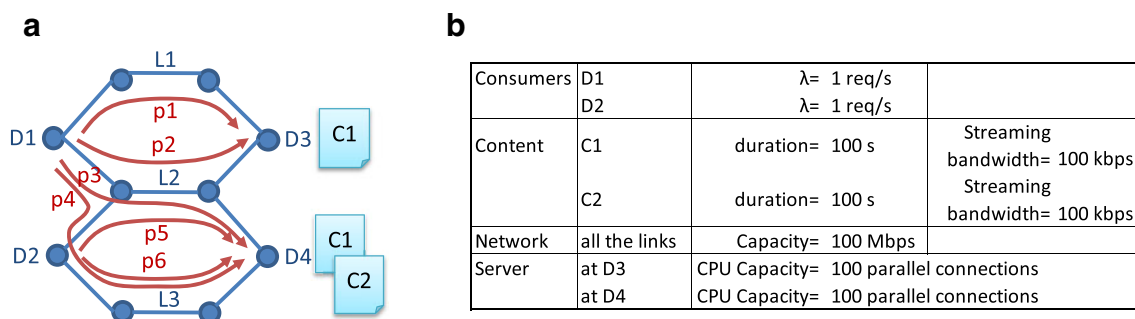


Fig. 2 Scenario and parameters of the simulations for validation of decision algorithms. a Simulation scenario b simulation parameters

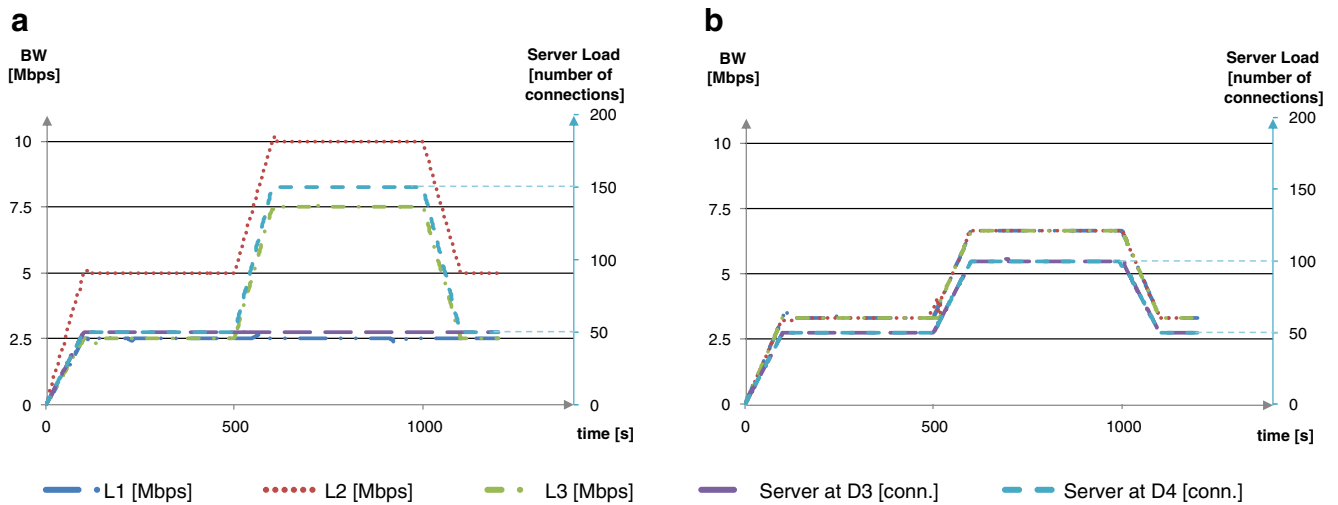


Fig. 3 Results of validation tests in dynamic scenario. **a** Random path and random server strategy. **b** Server load and available bandwidth strategy

model of the Internet from the point of view of video content consumption. In this environment, we simulate the content request arrivals and analyse content delivery performance for each content request. The merit of our studies lies in the comparison of the different server and path selection algorithms presented in the above sections. However, in order to get results that are close to the reality, we performed an exhaustive immersion in the state of the art of video content delivery systems, taking input from the currently available CDN setup.

5.1 A model of video content consumption in the internet

The proposed model for video content consumption covers: (1) large-scale model of Internet topology, (2) content server location and characteristics, and (3) rules for distribution of content replicas. Below, we present details of our model.

5.1.1 Network topology

The Internet topology is based on AS-level data provided by Caida [27] from January 2011. We consider three-level hierarchical Internet topology (Tier-1, 2 and 3) according to the business relationship between domains (provider–customer and peering relationships). The resulting topology has 36,000 domains with 103,000 inter-domain links, where approximately half of them are stub domains. We assume that consumer population in a given domain is proportional to the length of network prefixes advertised by this domain. Using data provided by [28] and RIPE [29], we created a histogram presented in Fig. 4. Moreover, this data confirmed that Tier 3 domains advertise majority of prefixes, while Tier-1 and Tier-2 domains advertised less than dozens of prefixes.

Finally, regarding link capacities, we observe that operators such as China Telecom and others currently connect to consumers with 1 Gbps links [30]. Therefore, we assume uniformly distributed probability density function $U[0.5, 1.5]$ Gbps for links connecting Tier-3 domains. Furthermore, we assume the links connecting Tier-2 to its providers or peers to be ten times greater (i.e., $U[5.0, 15.0]$ Gbps) and, finally, $U[50.0, 150.0]$ Gbps links for connecting Tier-1 peering domains. Note that in our model these capacities are dedicated only to video traffic.

5.1.2 Content server location

For the analysis of the distribution of content servers within the network, we refer to the 50 largest video content providers and CDNs, which are, among others: Level(3), Global Crossing, LimeLight, Akamai, AT&T, Comcast and Google [31]. The number of content servers in these

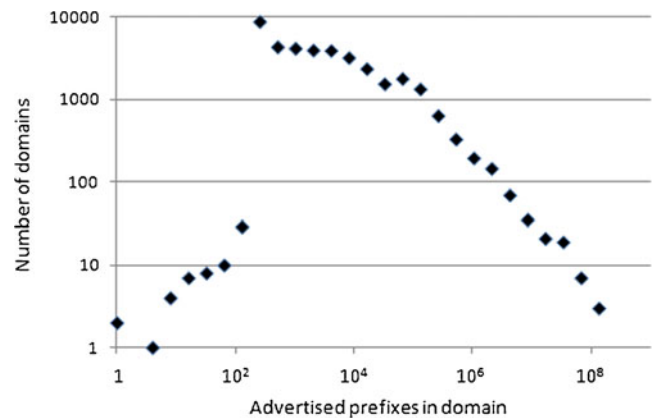


Fig. 4 Histogram of advertised prefixes

domains corresponds to the information provided in public statements and white papers. For the remaining domains, we assume uniformly distributed probability density function $U[50,150]$ for assigning the number of content servers to the domains due to the fact that Akamai serves 1,000 domains (as Youtube, QuickTime TV, etc.) with 84,000 servers [32] that mostly are multimedia servers. The total number of servers in the modeled network is approximately 200,000 servers.

The characteristics of the content server of interest are the maximum number of concurrent connections that may be served (limited by server disk I/O and network bandwidth [34–36]) and the maximum number of contents stored. Current commercial servers may serve from 50 up to 1,000 concurrent connections [33, 36, 37]. For simplicity, we assume that all servers in the network have the same number of maximum concurrent connections but this number varies from one simulation to another. On the other hand, commercial servers differ much in storage capacity. A medium server may have 600 GBytes content storage capacity [37], which approximately translates to 100 titles since one High Definition 2-h movie may have a size of 5–8 GBytes [38]. Consequently, our model assumes that servers store around 20 million copies of content.

5.1.3 Content characteristic and distribution

The features of the video contents are based on data from [39]. We analysed the duration of the movies for 5,000 most popular titles. The results indicate that mean duration of the movie is around 4,100 s. For each content, we attached a random value of streaming bandwidth comprised between 2,600 and 3,400 kbps, imitating the range of bandwidth the videos in Netflix Canadian network are streamed [40].

There are two possible techniques for assuring load balancing in content networks, which are striping and

replication. Striping consists of partitioning the content between different servers, whereas replication consists of copying the content. In most content replication approaches, the number of copies of given content depends on its popularity and it is widely accepted for video distribution in the Internet follows the Zipf's law [30, 35, 41]. In our model, the skew parameter of the Zipf's formula equals 0.2 as suggested in [30]. As explained above, the total number of copies stored in the servers is around 20 million. In order to have these copies (exactly 19,332,562 copies), the most popular content is copied 17,000 times and the rest follows the Zipf's formula. The distribution of copies in the servers is another key point in content networks [42], since a good distribution strategy may definitely increase the efficiency of the system. For simplicity, we assumed a random strategy subject to the condition that no more than one copy of given content may be in any server.

5.1.4 Model summary

Table 1 presents the summary of the model parameters presented above. Although the range of parameters to be modeled in video content Internet is broad, we believe our model reflects closely the reality.

5.2 Content delivery simulations

The simulation process runs as follows: within the video content consumption model described above, consumers request for content following specific request arrival process (detailed Section 5.2.1). The decision maker is the responsible of selecting the server and path for serving the request depending on the evaluated strategy (detailed in Section 5.2.3). The load of the selected server is increased by the requested bandwidth for whole duration of the content. In the same fashion, all

Table 1 Parameters of the model for content distribution in the Internet

Network topology		
Number of domains	~36,000 domains	Sources: [28, 29]. About the half are stub domains
Number of links	~103,000 links	Sources: [28, 29]
Server characteristics		
Number of servers	~200,000 servers	Source: Akamai [32] and CDN
Capacity of servers	100 titles	Sources: [37] and [38]
Content characteristics		
Number of content files	5,000 titles	Source: Film Web Inc. [39]
Number of copies	20,000,000 copies	
Mean duration of content	4,100 s	Source: Film Web Inc. [39]
Streaming bandwidth	$U[2,600,3,400]$ kbps	Source: NetFlix [40]
Content popularity	Zipf's law (skew parameter=0.2)	Source: [30]

the links of the selected path are loaded by a value equal to the bandwidth of the content during a time equal to the content duration. We assume that the connection bandwidth is independent of the state of the network during the content delivery, which is feasible for delivery of streaming content.

Whenever any link or server went beyond the capacity threshold, we consider that all the connections carried by the over-loaded link and/or all the connections currently served by the over-loaded server as unsuccessful (i.e., QoS was not guaranteed). Regardless of the state of the path and server, the request is served and the content is delivered by the selected path. When a connection is terminated, all relevant loads are taken off its respective link(s) and server. Finally, we measure the ratio of successful connections defined as the ratio of the number of successfully completed requests to the total number of content requests.

5.2.1 Request arrival process

The number of content request generated from each domain is proportional to the advertised prefixes of the domain. The arrival process for content requests is amply dealt in the literature; it depends on the type of content and type of application with most models suggesting a Poisson arrival process for short time scale (e.g., for IPTV applications in [43] and for Video on Demand applications in [44]). Some authors point out a modified Poisson process [30] for arrival rate. In our simulations, we applied the Poisson arrival model without considering non-stationary effects such as diurnal/night traffic. Each request is attached to a specific content following the Zipf's law for content popularity.

5.2.2 Routing protocols

The routing algorithms considered in the simulations are:

1. Single shortest path,
2. Multi shortest path,
3. Single bandwidth-based path,
4. Multi bandwidth-based path.

In all experiments, we use the routing protocol defined in Section 3.1. However, we adapt its behaviour by setting appropriate decision variable, i.e. path length or bandwidth, as well as by tuning the number of advertised paths, i.e. one or five paths. In order to evaluate how many preferred paths should be advertised, we assess the effectiveness of proposed protocol in the reference scenario where entire list of feasible paths was propagated. In this case, the protocol is more effective since there is no loss of information in the intermediate domains. Anyway, the results confirmed that

effectiveness of routing protocol only slightly increases when protocol advertises more than five paths. Therefore, in our experiments we use five alternative paths.

Single shortest path protocol offers one shortest path between server and consumer domains. Therefore, in our routing model we randomly select one of the shortest paths. Multi shortest path protocol offers five of the shortest paths between server and consumer domains and for each request one of them is selected following the rules of the decision algorithm. Single bandwidth-based path routing protocol offers the best path between server and client domains, where “best path” refers to the path with highest capacity in the bottleneck link. Multi bandwidth-based path routing protocol offers five best paths between server and client domains.

5.2.3 Decision strategies

In our experiments we consider the following decision algorithms:

1. Random server and random path, which combined with shortest single path routing protocol, reflects the current Internet;
2. Closest server and random path, which combined with shortest single path routing protocol, reflects the strategy of some current CDNs [45];
3. The least loaded server (called best server) and random path, where the least loaded server is the one that currently serves fewer requests;
4. The best server and the path with more available bandwidth in the bottleneck link (called best path).

Table 2 shows the values of reservation and aspiration levels for the parameters used in the simulations. The only strategy with two parameters is “best server/best path” where we believe that server load parameter is more crucial than bottleneck BW, i.e., between low loaded server and low loaded path, we select the former. Because of this, the reservation and aspiration level follows the formula (3), since the relation between aspiration and reservation level indicates the “a priori” importance of the parameter.

$$\frac{a_{\text{server_load}}}{r_{\text{server_load}}} \left(\frac{a_{\text{bottleneck_BW}}}{r_{\text{bottleneck_BW}}} \right)^{-1} \quad (3)$$

In order to simplify the decision process when a new request arrives to the system, the group of the feasible solutions (Pareto optimal solution set) is reduced to 100 random servers (wherever they are) and one or five paths depending on the assumed routing.

Table 2 Reservation and aspiration levels in the simulations

Strategy	Parameter	Reservation level (r)	Aspiration level (a)
Random server/random path	Not applicable		
Closest server/random path	Path_length	$r_1=100$	$a_1=0$
Best server/random path	Server_load	$r_1=1.0$	$a_1=0.0$
Best server/best path	Server_load	$r_1=1.0$	$a_1=0.0$
	Bottleneck_BW	$r_2=1500.0$	$a_2=1.5 \times 10^5$

5.3 Simulation results

As aforementioned, a content delivery is considered as unsuccessful if one or both of the following is involved: over-loaded path or over-loaded server. Depending on the system state, one cause or the other becomes significant. In order to obtain the most complete results, we first investigate limit cases where both the causes appear simultaneously, and then we focus on deep analysis of working point.

5.3.1 Analysis of system under limit cases

For analysing the limit cases, we first performed tests for very high capacity threshold of the servers (10^8 concurrent connections) and capacity threshold in the link as assumed in the model of video content consumption. Figure 5a presents the success ratio (relation between successful request and all attempts) for the four investigated decision strategies with single shortest path routing protocol. It can be observed that the request arrival rate for which the overload starts is in the range of 500 request/s.

In the following tests, we set to find the value of server capacity threshold for which the servers began to be overloaded in the range of 500 request/s. The link capacity threshold was infinite in all links in order to avoid overloaded links. Afterwards, we increased the server capacity threshold (the same value in all the servers) until we observed unsuccessful connections in the expected range, which occurred for a threshold of 200 concurrent

connections. The results of these tests for single shortest path routing are presented in Fig. 5b.

All the presented tests were repeated 5 times and all the results have confidence intervals fewer than 3 % of the mean values at the 95 % confidence level. For clarity purposes, we do not present the confidence intervals in the figures.

As we may observe in Fig. 5b, the two latter strategies (i.e., best server/random path and best server/best path) offered the same results since there were no bottleneck links in these simulations. In Fig. 5a, one could think that the strategies “closest server/random path” and “best server/random path” should offer the same results since the server is not the bottleneck in these simulations; nonetheless, “closest server” strategy makes sure that links are less used since the selected server is often in the consumer domain and no inter-domain link is loaded by the connection. Because of this, the results for this strategy are slightly better.

For other routing algorithms, the server load threshold value for which server and link overload appear simultaneously is also in the range of 200 concurrent connections. Therefore, in the next simulations, we set the threshold value of server load equal to 200 concurrent connections. Note that this value depends on the assumed scenario and thus, only valid for the presented results.

5.3.2 Analysis of system under working point

The next results compare the four decision algorithms and the four routing algorithms under working conditions defined from analysis of limit cases. Figure 6 presents the

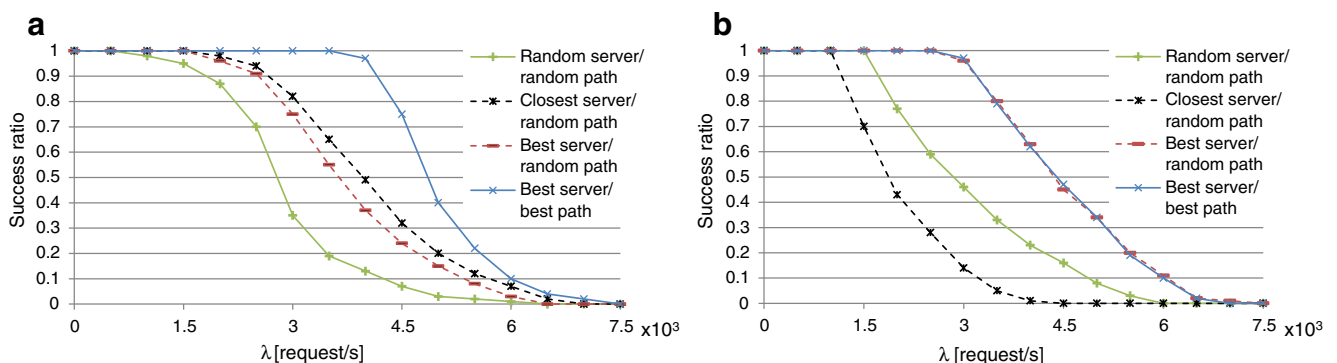


Fig. 5 Success ratio for shortest single path routing algorithm and four decision algorithms. **a** Server threshold= 10^8 concurrent connections and link threshold=1 Gbps; **b** server threshold=200 concurrent connections and link threshold= 10^8 Gbps

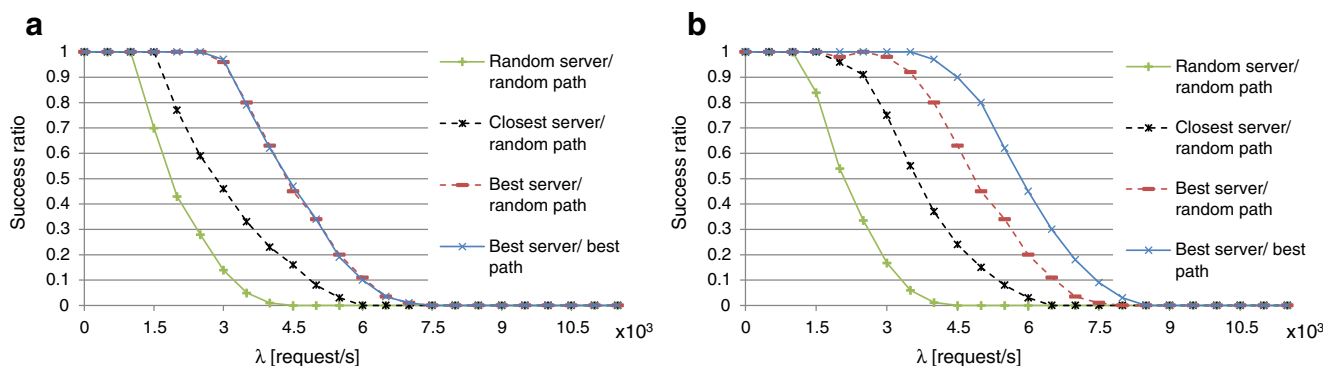


Fig. 6 Success ratio for the four decision algorithms and shortest path routing algorithm. Server threshold=200 concurrent connections; link threshold=1 Gbps. **a** Shortest SINGLE path routing algorithm; **b** shortest MULTI path routing algorithm

results for shortest (a) single and (b) multi path routing algorithms, whereas Fig. 7 presents the results for best (a) single and (b) multi path routing algorithms.

As we may observe, the shape of the curves are different compared to those in Fig. 5. The reason is the overlap of both overload effects: in servers and in links. The low confidence interval of the tests (fewer than 3 % of the mean values at the 95 % confidence level) validates the results.

The general conclusion of the results is that the increased knowledge about links and servers improves the efficiency of the system. Specifically, the combination of appropriate routing algorithm together with appropriate decision algorithm increases the effectiveness of the system. So, for example, for the assumed model and scenario, best multi path routing and best server/best path strategy ensure 90 % of successfully delivered contents for arrival request up to 8.3×10^3 request/s, whereas in current Internet strategies (shortest single path routing and random server/random path strategy) this same percentile is obtained only for arrival request fewer than 1.0×10^3 request/s.

We also observe that current CDNs strategies (single shortest path routing and closest server/random path strategy) reaches 90 % success ratio for 2×10^3 request/s, whereas

current Internet strategy reaches only 40 % for the same arrival rate. Strategies which use more information about network (as best server/best path) obtain 100 % successful content delivery for this case.

The strategies which use dynamic (online) information of the network and/or servers offer better results than static information, which suggests the need of monitoring systems in order to improve efficiency of the decision process. However, best single path results are not much better than shortest single path results because, in single path routing, the decision process cannot use this information since it cannot select the path. The efficiency for best multi path routing is much higher, which shows the importance of providing multi path routing protocol in networks delivering content.

By comparing results for the random server/random path strategy with shortest single path and shortest multi path routing algorithms, we conclude that, in the current Internet, multi path protocol does not introduce significant effectiveness gain.

In general, we can say that multi path routing protocol introduces the load balancing feature, which improves efficiency in all cases, not only in situations where overloaded

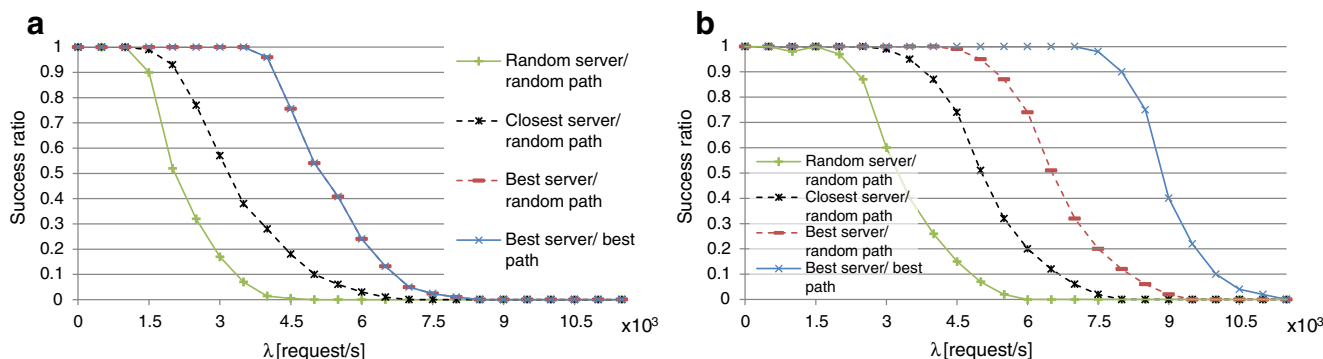


Fig. 7 Success ratio for the four decision algorithms and best path routing algorithm. Server threshold=200 concurrent connections; link threshold=1 Gbps. **a** Best SINGLE path routing algorithm; **b** best MULTI path routing algorithm

link/server occurs. The same direction is followed by decision algorithms, such as best server/best path, which introduce repartition of the requests between servers and paths.

Let us remark that in best server/best path algorithm, we could tune reservation and aspiration levels obtaining a certain gain in efficiency, but this tuning depends on the assumed scenario and it is out of the scope of this paper.

5.3.3 Notes about the trustworthiness and reliability of the results

The complexity of the simulations prompts the question on the trustworthiness of the simulation process itself. We devoted much effort to understand the behaviour of the simulation process. After validating the correctness of our simulations in small scenarios, we concentrated on the possible reasons which could distort the results.

The high confidence of the results (lower than 3 % of the mean values at the 95 % confidence level) may be explained by the wide extension of the assumed model (video content Internet) as well as the fact that all the simulation tests counted at least 10^{11} served content requests. This should be sufficient to ensure dynamism in the states of servers and links (empty, loaded, overloaded) are accounted for, i.e., both servers and links have changed state many times during each simulation test. In order to confirm this point, we performed tests for checking time range dependence.

For this, we analysed the behaviour of one randomly selected server and one randomly selected link. The number of served connections by the server through time as well as the capacity used in the link can be considered as stochastic processes, which are characterised by two dimensions: space and time. With regard to the space dimension, we assume that simulations behave as real content delivery in the network (for simplicity purposes) and we focus on verifying whether simulations and real networks behave similarly over time. In real networks, there appear many effects such as multiplexing with other traffic streams, which substantially reduce the time dimension dependence. By “time dimension dependence” we understand how much the state of given server, or link, in time t influences the state in time $t + \Delta t$. Therefore, we should check short-range dependence also in our simulations. For this, we investigated and confirmed that the autocorrelation function for different lags of time in both server (number of served connections) and link (used capacity) decays slower than exponential function.

In conclusion, we verified that the simulation process does not enter in “loop stance” and the simulation process is trustworthy. However, the trustworthiness of the simulation process does not indicate that the results are reliable. The reliability of the results depends on the made assumptions, which are numerous in the presented simulations. Anyway, the comparison-based simulations credibly show

the importance of network level information in increasing the efficiency of the systems. Our scope was checking whether more information ensures efficiency gain. By the term “ensure”, we mean that in any case fewer information provides better results and on the other hand, more information provides significant improvement in efficiency. The results confirmed these two aspects.

6 Summary

The proliferation of commercial and user-generated content has fostered the establishment of various CDNs and P2P networks and most recently, motivated research on information-centric networking paradigm. In this paper, we focused on designing efficient content source selection algorithm which decides the best available source and path for serving content requests for these content access and distribution platforms. We proposed and evaluated the multi-criteria decision algorithm, which exploits two closely related processes. The first process, which operates offline in a long term, discovers multiple content delivery paths and gathers their respective transfer characteristics. The second process, which is invoked for each content request, combines available information about network and server condition for selecting the best content server and delivery path.

The simulation results confirmed that the two-level algorithm provides more information to the selection of server and path. This results in a higher percentile of satisfied content requests by improving utilisation of network and server resources. When the number of content requests increases, then the two-level algorithm makes feasible load balancing in both network and servers, avoiding or slowing down overload conditions. Load balancing is achieved also in situations of normal load which might be an interesting feature for network and content service operators.

Further work will focus on the effect of the parameter setting in the efficiency gain in order to provide the best decision algorithm in content networks. Moreover, we started to analyse the optimization of multi-criteria decision algorithm by tuning the values of reference and aspiration levels for given set of parameters. In [46], the first results are presented. At last, when both the set of parameters and the tuning of reference levels are optimized, then it will be possible to study the difference between optimum and heuristic methods.

Acknowledgment The authors would like to thank all partners from EU FP7 COMET project for their support and fruitful discussions.

Open Access This article is distributed under the terms of the Creative Commons Attribution License which permits any use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

References

1. T. Koponen, M. Chawla, B-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker and I. Stoica, “A Data-oriented (and Beyond) Network Architecture,” in Proc. ACM SIGCOMM’07, Kyoto, Japan, Aug. 2007.
2. V. Jacobson, D. K. Smetters, James D. Thornton, Michael Plass, Nick Briggs, Rebecca L. Braynard, “Networking Named Content,” Proc. ACM CoNEXT 09, 2009, pp. 1–12.
3. NDN – Named Data Networking project, <http://www.named-data.org/>
4. P. A. Aranda et al., “Final Architectural Framework”, June 2010. <http://www.4ward-project.eu/>
5. Scalable and Adaptive Internet Solutions (SAIL) project. <http://www.sail-project.eu/>
6. D. Trossen et al., “Conceptual Architecture: Principles, Patterns and Sub-components Descriptions”, May 2011. <http://www.fp7-pursuit.eu/PursuitWeb/>
7. P. Jokela, A. Zahemszky, C. E. Rothenberg, S. Arianfar and P. Nikander, “LIPSIN: Line Speed Publish/Subscribe Inter-networking”, Proc. ACM SIGCOMM’09, Barcelona, Spain, August 2009.
8. W. K. Chai, et al., “Final specification of mechanisms, protocols and algorithms for the content mediation system”, Jan 2012, <http://www.comet-project.org>
9. Chai WK et al (2011) CURLING: content-ubiquitous resolution and delivery infrastructure for next-generation services. IEEE Communications Magazine 49(3):112–120
10. Buyya R, Pathan M, Vakali A (2008) “Content Delivery Networks”, ISBN 978-3-540-77886-8. Springer, Germany
11. Chai WK, Psaras I, Pavlou G (2012) Cache less for more in information-centric networks. IFIP Networking 2012, Czech Republic
12. S. G. Dykes, K. A. Robbins and C. L. Jeffery, “An Empirical Evaluation of Client-side Server Selection Algorithms,” Proc. IEEE INFOCOM ‘00, Tel-Aviv, Israel, March 2000.
13. Zegura E, Ammar MH, Fei Z, Bhattacharjee S (2000) Application-layer Anycasting: a server selection architecture and use in a replicated web service. IEEE/ACM Trans on Netw 8(4):455–466
14. Lenders V, May M, Plattner B (2008) Density-based Anycast: a robust routing strategy for wireless ad hoc networks. IEEE/ACM Trans on Netw 16(4):852–863
15. Katabi D, Wroclawski J (2000) “A Framework for Scalable Global IP-Anycast (GIA),” Proc. ACM SIGCOMM’00. Stockholm, Sweden
16. K.S. Pradyumn, “On gradient based local search methods in unconstrained evolutionary multi-objective optimization”. Proc. of the 4th international conference on Evolutionary multi-criterion optimization, March 05–08, 2007, Matsushima, Japan.
17. Ehrgott M (2005) Multicriteria optimization. Springer, New York
18. Messac A, Melachrinoudis E, Sukam CP (2000) Aggregate objective functions and Pareto frontiers: required relationships and practical implications. Optimization and Engineering Journal, Kluwer Publishers 1(2):171–188
19. Wierzbicki AP, Makowski M, Wessels J (2000) Model-based decision support methodology with environmental applications. Kluwer, Dordrecht
20. A. Wierzbicki, “The use of reference objectives in multiobjective optimization”. Lecture Notes in Economics and Mathematical Systems, vol. 177. Springer-Verlag, pp. 468–486
21. Masip-Bruin X et al (2006) Research challenges in QoS routing. Computer Communications 29(5):563–581
22. Kuipers F et al (2004) Performance evaluation of constraint-based path selection algorithms. IEEE Network 18(5):16–23
23. G. Cheng, “The revisit of QoS routing based on non-linear Lagrange relaxation”, Int. J. Commun. Syst., vol. 20, 2007
24. Khadivi P, Samavi S, Todd TD (2008) Multi-constraint QoS routing using a new single mixed metrics. J Netw Comput Appl 31 (4):656–676
25. Mieghem P, Kuipers FA (2004) Concepts of exact QoS routing algorithms. IEEE/ACM Trans Netw 12(5):851–864
26. Y. Sawaragi; H. Nakayama and T. Tanino. “Theory of Multiobjective Optimization”. Mathematics in Science and Engineering, vol. 176 Academic Press Inc. ISBN 0126203709. 1985.
27. The Cooperative Association for Internet Data Analysis, <http://www.caida.org/>.
28. University of Oregon Route Views Archive Project David Meyer, <http://archive.routeviews.org/>.
29. RIPE Network Coordination Centre, “Routing Information Service”, <http://www.ripe.net/data-tools/stats/ris/ris-peering-policy>
30. H. Yu, D. Zheng, B. Zhao and W. Zheng, “Understanding User Behavior in Large-Scale Video-on-Demand Systems”. In Proc of EuroSys 2006
31. Labovitz C, Iekel-Johnson S, McPherson D, Oberheide J, Jahanian F, Karir M (2009) ATLAS Internet Observatory 2009 Annual Report. Arbor Networks Inc., University of Michigan and Merit Network Inc, Ann Arbor
32. Akamai Technologies, Inc., “Facts & Figures”, http://www.akamai.com/html/about/facts_figures.html
33. Bidgoli H (2004) The Internet Encyclopedia (Editor-in-Chief). John Wiley & Sons, Inc, Hoboken, p 502. ISBN 0471222046
34. A. Nimkar, C. Mandal and C. Reade, “Video Placement and Disk Load Balancing Algorithm for VoD Proxy Server”, In the proceedings of 3rd IEEE international conference on Internet Multimedia Services Architecture and Applications, 2009, pp. 141–146.
35. Hitachi, Ltd., “Hitachi VOD Server”. ©2010, <http://www.hitachi.com/products/it/network/SDP/>.
36. NetUP Inc., “IPTV solutions by NetUP: Video on Demand & Virtual Cinema”, ©2011, <http://www.netup.tv/en-EN/vod-nvod-server.php>.
37. VBrick Systems Inc., “VBrick Enterprise Media System”. ©2010, http://www.vbrick.com/docs/vbrick_datasheet_VOD-W-family.pdf.
38. J. Donovan and N. Faris, Akamai Technologies, Inc., “Digital Movie Traffic on the Akamai Network Increases Three Fold”, March 2010, http://www.akamai.com/html/about/press/releases/2010/press_031610_1.html
39. Film Web Inc., homepage: <http://www.filmweb.pl/>
40. K. Florance, Netflix Content Delivery, “Netflix Performance on Top ISP Networks”. January 2011, <http://techblog.netflix.com/2011/01/netflix-performance-on-top-isp-networks.html>.
41. K. Gummadi et al., “Measurement modeling and analysis of a peer-to-peer file sharing workload”. Proc. of SOSIP. October 2003.
42. Kangasharju J, Roberts J, Ross KW (2002) Object replication strategies in content distribution networks. Computer Communications 25(4):367–383
43. M. Cha, P. Rodriguez, J. Crowcroft, S. Moon, and X. Amatriain, “Watching television over an IP network”. Proc. of the 8th ACM SIGCOMM Conference on Internet Measurement, Vouliagmeni, Greece, October 20–22, 2008.
44. Yu H, Zheng D, Zhao B, Zheng W (2006) Understanding user behavior in large-scale video-on-demand systems. SIGOPS Oper Syst Rev 40(4):333–344
45. M. Hofmann, R. Leland, R. Beaumont, “Content Networking: Architecture, Protocols, and Practice”, Morgan Kaufmann Publisher. ISBN 1-55860-834-6. 2005.
46. Mongay Batalla J, Begen A, Chen Y (2012) Optimization of decision process in network and server-aware algorithms. IEEE Network 2012, Italy, 978-1-4673-1391-9/12 ©2012