ASTAS: Architecture for Scalable and Transparent Anycast Services

Tim Stevens, Marc De Leenheer, Chris Develder, Filip De Turck, Bart Dhoedt, and Piet Demeester

Abstract: **Native IP anycast suffers from routing scalability issues and the lack of stateful communication support. For this reason, we propose ASTAS, a proxy-based architecture that provides support for stateful anycast communications, while retaining the transparency offered by native anycast. Dynamic resource assignment for each initiated session guarantees that a connection is established with the most suitable target server, based on network and server conditions. Traffic engineering in the overlay can be realized in an effective way due to the dissemination of aggregated state information in the anycast overlay.**

To minimize the total deployment cost for ASTAS architectures, we propose optimized proxy placement and path finding heuristics based on look-ahead information gathered in network nodes. Contrary to a regular ILP formulation, these heuristics allow to optimize proxy placement in large networks. A use case on a European reference network illustrates that lower proxy costs enable proxy deployment closer to the end-users, resulting in a reduced network load.

Index Terms: **Anycast, overlay architecture, path finding, proxy placement, resource intensive services, stateful communication, traffic engineering**

I. INTRODUCTION

IP anycast enables communication between a source host and one member of a group of target hosts, usually the one nearest to the source [1]. As such, anycast is considered a powerful tool for realizing transparent, scalable and reliable communications with connectionless distributed network services. The use of replicated DNS root servers listening to a common—anycast—IP address is an example application where anycast has proven useful [2].

At present, there are limitations that prevent widespread adoption of IP anycast in general, and its adoption for network service provisioning more specifically. First, session-oriented services (including all applications implemented on top of TCP) cannot take advantage of this addressing mode, because subsequent packets from the same source host (and session) may be routed towards a different target host. Another anycast limitation is its poor global routing scalability due to the fact that routes to anycast groups cannot be aggregated: widespread adoption of end-to-end native IP anycast would undoubtedly lead to huge and unmanageable routing tables. Possible solutions for this issue have been proposed by D. Katabi et al. [3] and H. Ballani et al. [4].

In this paper, we present ASTAS: a proxy-based Architecture for Scalable and Transparent Anycast Services. Using this architecture, distributed network services can be scaled to a large number of consumers and resources, and this in a transparent way from an end-user perspective. A scenario where resource-constrained clients delegate resource-intensive tasks to computational resources installed in the network is an example application where this architecture could prove useful. In addition to network state and metrics, the proxy infrastructure uses server state information to forward service requests to the most suitable location, which is not possible using only IP anycast. Server state aggregation in the overlay ensures that state dissemination is achieved in a scalable way. Furthermore, anycast group state changes (e.g., a new anycast group, a failing server) are completely hidden from the routing substrate, thereby maintaining IP routing scalability.

Whereas the PIAS anycast overlay [4] focuses on a global, lightweight anycast solution with a semi-static coupling between an overlay node receiving anycast service requests and an overlay node connecting a target server, we tailor our anycast proxy architecture for the scalable execution of resource-intensive services, where session-based target selection increases resource utilization efficiency. Application layer anycast [5] or other anycast implementations above the network layer such as i3 [6] also support stateful communications, albeit at the expense of losing IP anycast transparency, which is undesirable.

After describing the anycast proxy architecture, we investigate how large the proxy infrastructure should be and where proxies should be placed in the network to accommodate a given client demand in a network-efficient way. For this purpose, we present a near-optimal heuristic approach tackling anycast infrastructure dimensioning and proxy placement in potentially large networks. Based on several parameters, including anycast proxy infrastructure costs, network operational costs and infrastructure component capacities, we transform the problem into a Fixed Charge Network Flow Problem (FCNFP) [7], which is solved by means of a Dynamic Slope Scaling Procedure (DSSP) introduced by Kim and Pardalos [8]. Using this solution technique, we can show that a modest anycast overlay provides an effective solution, even in large networks.

The remainder of this paper is structured as follows. Section II details the anycast proxy architecture. We present the optimization problem and heuristic approaches for infrastructure dimensioning and proxy unit placing in Section III. In Section IV, computational experiments validate the proxy placement and path finding algorithms detailed in the paper. Additionally, simulation results for a large pan-European reference network are discussed. Section V summarizes the main results of this paper.

Manuscript received May 15, 2007; approved for publication by Eiji Oki, JCN Editor, October 16, 2007.

The authors are with the Department of Information Technology, Ghent University - IMEC - IBBT, Gaston Crommenlaan 8 bus 201, 9050 Gent, Belgium, email: tim.stevens@intec.ugent.be

II. ANYCAST ARCHITECTURE

A. IP anycast limitations

Because IP anycast forwards packets to the nearest member of an anycast group, it could prove useful as a transparent *service discovery* primitive. For single request-response services such as DNS, it can even support the entire service and increase service scalability by means of implicit coarse-grained load balancing between the anycast group members.

Despite these promising features, the use of IP anycast is not widely adopted and production use is essentially limited to DNS root server replication [2]. According to Ballani et al. [4], the main reason for this is the *lack of IP routing scalability* inherent to native anycast. First, IP anycast routes cannot be aggregated and widespread adoption would lead to an explosive growth of IP routing tables. Since anycast group members—using the same IP address—can be scattered all over the Internet, a distinct routing entry is needed per anycast group. Secondly, anycast group dynamics (i.e., joining and leaving members) necessitate frequent changes to a relatively slow converging IP routing configuration, eventually leading to network instability. In general, intra-domain and inter-domain routing protocols and algorithms are designed assuming that the underlying network configuration is relatively stable over time. As such, these protocols are not optimized to handle frequent changes to the network topology.

With the intention to use IP anycast for transparently providing scalable remote service execution, two additional limitations arise:

(i) IP anycast does not support session-based communications;

(ii) IP routing is static and does not support multiple constraint routing or traffic engineering in general.

Today, most Internet traffic originates from TCP-based communications. Due to limitation (i), these services cannot take advantage of IP anycast apart from the service discovery feature. Limitation (ii) implies that anycast targets cannot be selected based on volatile network (e.g., congestion) and/or target conditions (e.g., current server load).

Taking into account both the strengths and weaknesses of IP anycast, we propose ASTAS, an Architecture for Scalable and Transparent Anycast Services that is based on PIAS [4].

B. ASTAS overview

The ASTAS overlay infrastructure consists of a combination of two types of nodes: *client proxies* (CP) and *server proxies* (SP). Both client proxies and server proxies are special routers advertising their proximity to the anycast IP range into the routing substrate. By doing this, the proxy routers force IP packets with an anycast destination address to pass through the overlay. When a client initiates a new session to an anycast destination, the closest client proxy registers the new session and selects an appropriate server proxy to forward the request to. The server proxy receiving the new session then selects the most suitable server to handle the request.

Fig. 1 depicts the steps involved in setting up a session between a client and a target anycast server through the proxy system. Step R1 registers a server with unicast address **S** for the service offered by anycast address **A** and port **b**. Note that this registration uses native anycast to reach the *closest* server proxy (SP). At this point, the SP configures an IP tunnel to the unicast address **S**. Next, a client can initiate a session by sending a packet addressed to the anycast service of choice (step 1). When the packet arrives at the *closest* client proxy (CP), it is tunneled to a suitable SP (step 2), where it is tunneled again towards a target server (step 3). The return path (steps 4, 5 and 6) is realized in the same way. *Stateful tunneling* occurs twice in each direction (in CP and SP) and is necessary to guarantee session continuity. The IP tunnels cannot be avoided on the return path because both the CP and SP have to monitor the session state, for which packets have to traverse the system in both directions. This also implies that target servers need to be aware of the ASTAS infrastructure, since an IP tunnel is maintained between each target and its SP, in both directions. However, target servers do not discover the SP unicast IP address and tunnel packets towards the anycast address (step 4).

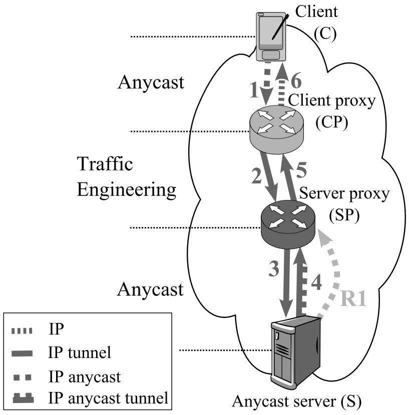
Stateful communications are not explicitly supported by the proposed overlay mechanism since steps 1 and 4 in Fig. 1 cannot guarantee that subsequent packets from the same session arrive in the same proxy. However, in practice there are two reasons why the overlay infrastructure suffices to support stateful communications. First, the number of proxies is relatively small compared to the number of network nodes, meaning that a single link or router failure is unlikely to cause a client (or server) to swap to another proxy node. Secondly, the distance between a client (or server) and its closest proxy node is usually significantly smaller than the end-to-end distance between a client and a server, thereby reducing the chances for a failure on the path segment between client (or server) and proxy node.

C. Comparison with PIAS

ASTAS is inspired by PIAS, the anycast architecture proposed by Ballani et al. Before describing dissimilarities between both architectures, we review the PIAS design goals inherited by ASTAS that native IP anycast cannot fulfill:

1) Ease of joining and leaving: using native IP anycast, servers have to interact with the routing substrate to join or leave; PIAS and ASTAS eliminate this issue.
2) Scale by number of groups: contrary to native IP anycast, PIAS and ASTAS have the possibility to aggregate anycast IP addresses in a single range.
3) Scale by group dynamics: PIAS and ASTAS hide server dynamics (i.e., joining, leaving, status updates) from IP routing to increase routing stability.
4) Target selection criteria: besides proximity, PIAS and ASTAS can select a suitable target server based on load and connection affinity.

An additional PIAS general design objective is global scalability. In this context, the architecture is kept lightweight and proxies exchange as little information as possible. This results in a relatively static target selection mechanism with stateless client proxies forwarding all service requests to the same SP until their long-lived pairing gets invalidated. On the contrary, the proposed ASTAS overlay selects a suitable SP for each initiated session to guarantee optimal service provisioning to the client. As a result, CP need to maintain session state, however.



Step	Packet headers (From → To)
R1	S:b → A:c
1	C:a → A:b
2	CP:C:a → SP:A:b
3	A:C:a → S:A:b
4	S:A:b → A:C:a
5	SP:A:b → CP:C:a
6	A:b → C:a

Fig. 1. Anycast communication through the proxy system. In the table capitals refer to IP addresses and lowercase characters point to the TCP/UDP port used.

Depending on the type of service and client expectations, the overlay of choice can be PIAS or ASTAS. For query-response services (e.g., DNS) or services based on very short sessions (e.g., HTTP) the ASTAS session-based resource selection may be less beneficial, as it will increase the client-server round trip time. Under these conditions, a lightweight semi-static overlay is more efficient and reduces control plane overhead. For services based on relatively long-lived sessions, where service availability is critical and each session consumes a considerable amount of resources, a fine-grained and dynamic solution like ASTAS is preferred.

Fig. 2 depicts the ASTAS changes to the PIAS data path to accommodate session-based resource assignation. Due to the stateless nature of the PIAS CP, a shortcut can be taken for the PIAS return path (see Fig. 2(a), step 5).

Another difference with PIAS can be found in the communication phase between SP and target anycast server (see Fig 2, steps 3 and 4), where ASTAS prefers IP tunneling over network address translation (NAT), thereby preserving end-to-end connectivity. This is important to attain total transparency towards clients, indispensable for IPsec support and application layer services that experience difficulties traversing NAT gateways. Contrary to NAT, IP tunneling requires cooperation and configuration of target servers, however. In any event, the anycast overlay is not transparent for participating servers due to control plane interaction with the SP (e.g., server registration), so IP tunnel setup between the SP and the server is not considered as an extra limitation.

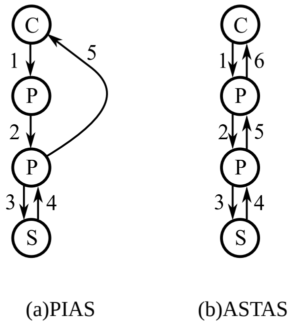
D. Scalable state dissemination

During the initiation phase of an ASTAS session, two important decisions determine which target server eventually processes a client request: first, the CP decides to which SP the request is forwarded; subsequently, the SP selects the actual target. If optimizing end-to-end path length or coarse-grained load balancing would be the ultimate goal of the anycast overlay, generally each CP could statically tunnel all service requests to its nearest SP without session management. When end-to-end Quality of Service (QoS) is required, SP selection should take into account both network and resource availability. Using the ASTAS infrastructure, resources can frequently update their SP with up-to-date status information, which can be distributed by the SP to the CP in an aggregated way. As such, CP have an accurate global view on SP fitness, while SP possess detailed information on the status of each participating server.

The rate of change of resource availability is the key metric to determine state dissemination scalability of the ASTAS overlay. For relatively long-lived sessions, server state does not change very frequently, resulting in modest control plane requirements. The inter-proxy control plane load is further reduced because the SP rate of change of resource availability is inversely proportional to its number of connected servers. One example of a service characterized by long-lived sessions is interactive online gaming, where user session duration can be modeled by an exponential (Weibull) distribution with an average duration of several minutes [9]. Other popular examples include VoIP and VoD services.

III. PROXY PLACEMENT AND PATH FINDING

In this section, we provide a heuristic approach to investigate how large the proxy infrastructure should be and where proxies should be placed in the network to accommodate a given client demand in a network-efficient way. The ultimate goal is to balance architecture investment costs and network operational costs associated with the overlay.



(a)PIAS (b)ASTAS

Fig. 2. PIAS versus ASTAS data path

Table 1. Model parameters

Variable	Description
$G(V, E)$	network topology, V and E denote the sets of vertices and edges
S	set of source sites $s_i (S \subset V)$
T	set of target sites t_j $(T \subset V)$
w_e	edge weight $(\forall e \in E)$
d_i	aggregated demand from source site s_i (units of flow)
c_j	capacity of target t_j (units of flow)
$f^{\cdots}$	fixed charge for opening a CP (f^{CP}), SP (f^{SP}) or target (f^{t_j}) edge
$u^{\cdots}$	unit processing cost for a CP (u^{CP}), SP (u^{SP}) or target (u^{t_j})

A. Problem statement

Equipped with the anycast architecture outlined in Section II, we wish to determine how many proxies are needed and where they should be attached to the network for a given client and server configuration. More formally, given a network $G(V, E)$, a set of source sites $S \subset V$ and their static demands d_i, a set of server sites $T \subset V$ and their capacities c_j, edge weights $w_e : e \in E$, determine how many CP (resp. SP) are needed, and where they should be attached to the network. Additionally, determine which target sites t_j need to be opened. The optimization process should balance network operational costs (related to flow unit processing costs for regular edges (w_e) and flow unit processing costs for proxies and servers $u^{\cdots}$), proxy infrastructure costs (determined by the fixed charge f^{CP} (resp. f^{SP}) associated with each CP (resp. SP)), and server site opening costs f^{t_j}. The parameters for this optimization problem are summarized in Table 1.

B. Solution techniques

In [10], we address the optimal placement and dimensioning of such an anycast architecture using an integer linear program (ILP) solved by a branch-and-bound algorithm [11]. Unfortunately, due to the complexity of the formulation, an exact solution can only be computed for relatively small networks (up to 300 nodes). For this reason we propose two heuristic methods to solve this problem: CP and SP *separated* and *combined* optimization. Contrary to the global optimization performed by the exact ILP, both heuristics decouple the proxy placement problem from the traffic engineering between the proxies (see Fig. 1), which results in a two-step optimization plan:

(i) Find suitable CP and SP locations and determine which target sites to use;

(ii) Optimize the flow between CPs and SPs.

In fact, step (ii) does not contribute to the proxy placement and dimensioning optimization, but allows us to examine the efficiency (optimality) of the proxy locations determined in step (i). Additionally, once the architecture is deployed, this steady state information can be used to steer online traffic engineering components. Using the proxy locations provided by step (i), a regular ILP minimizes the total network flow transportation cost between the proxies in step (ii):

Minimize

$$f(x) = \sum_{e \in E} w_e x_e$$

subject to

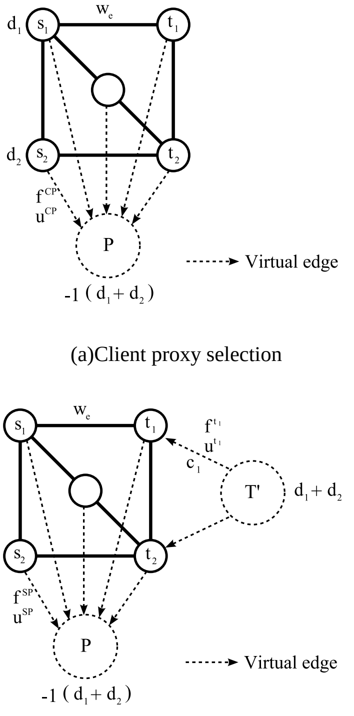
$$\sum_{e \in out(n_i)} x_e - \sum_{e \in in(n_i)} x_e = b_i, \forall n_i \in V \tag{1}$$

$$x_e \geq 0$$

In Eq. 1, b_i stands for the residual flow in node n_i, which is positive for a CP, negative for a SP, or zero for a regular node. In the ILP, x_e denotes the flow over edge $e \in E$. For this formulation, we assume all edges are unidirectional. If this would not be the case, bidirectional edges can easily be replaced by two unidirectional edges.

For step (i), the actual proxy placement, we propose two network transformations that allow us to reformulate the problem as a *Fixed Charge Network Flow Problem* (FCNFP). Besides the variable cost for using an edge (based on the amount of flow and the edge weight), in a FCNFP each edge has a fixed charge to *open* it. Due to its generality, the FCNFP has many practical applications, including network design and plant location. The FCNFP is formulated as follows:
Minimize

$$f(x) = \sum_{e \in E} f_e(x_e)$$



(a)Client proxy selection

(b)Server proxy selection

Fig. 3. Transformed network topology for separated client and server proxy optimization.

subject to

$$\sum_{e \in out(n_i)} x_e - \sum_{e \in in(n_i)} x_e = b_i, \forall n_i \in V \tag{2}$$

$$0 \leq x_e \leq c_e \tag{3}$$

where c_e is the maximum capacity of edge $e \in E$ and $f_e(x_e)$ is defined as follows:

$$f_e(x_e) = \begin{cases} 0, & x_e = 0 \\ f_e + u_e x_e, & x_e > 0 \end{cases} \tag{4}$$

In this case, the residual flow b_i in node n_i depends on the actual network transformation (see Fig. 3 and Fig. 4). In general, sources have positive residual flow, targets have negative residual flow and transit nodes have zero residual flow. In Eq. 4, f_e and u_e denote the fixed charge and unit flow transportation cost for edge $e \in E$. For the proxy placement problem, regular edges have $f_e = 0$ and $u_e = w_e$. For proxy-edges, f_e equals the proxy installation cost. Additionally, we will assume that all regular edges $e \in E$ are uncapacitated, i.e., in constraint 3, $c_e = \infty$ for non-proxy edges (see later).

Fig. 3(a) depicts the first transformation to approximate the proxy location optimization by a FCNFP: the original network is drawn in bold and the cumulative arcs and nodes for the transformation are plotted using dotted lines. Two source nodes s_i with demand d_i are located on the left and two possible targets t_j with capacity c_j on the right. By assigning a fixed charge f^{CP} to all virtual edges connecting the virtual sink P to the real network nodes, solving the FCNFP yields the optimal CP locations; each virtual edge with positive flow reveals a CP location. SP locations are discovered using a similar transformation shown in Fig. 3(b), with one additional virtual node T' representing the server locations. Virtual edges connecting T' to the server sites are capacitated to reflect server capacity limitations. Virtual edges between regular network nodes and the virtual sink P carrying positive flow reveal potential SP locations.

The separated optimization heuristic finds CP and SP locations using two independent FCNFP instances reflecting the transformed networks depicted in Fig. 3. Since clients and servers connect to the nearest proxy node without making a distinction between CP and SP, an additional merging routine is necessary to guarantee correctness: a client proxy is supposed to connect to a CP, whereas a target server has to connect to a SP. This is achieved by augmenting CP or SP proxies to proxies supporting both where necessary. After applying this routine, unused proxies can be dropped.

Combined optimization applies both network transformations together to create a single FCNFP instance. This approach is depicted in Fig. 4. Initially, each arc to the virtual proxy node P carrying positive flow is both a CP and SP. Unused proxy functionality is removed after solving the FCNFP. In this case, the fixed charge f^P associated with opening a virtual proxy edge can be determined based on the cost to combine a CP and SP on the same node.

C. Look-ahead based optimization

Thus far the heuristic proxy placement does not consider the cost associated with flow passing through the path segment between CP and SP. In a way, the combined FCNFP (see Fig. 4) takes both client and target site locations into consideration for a proxy placement decision, but nevertheless this does not necessarily optimize the proxy locations for the flow between CP and SP.

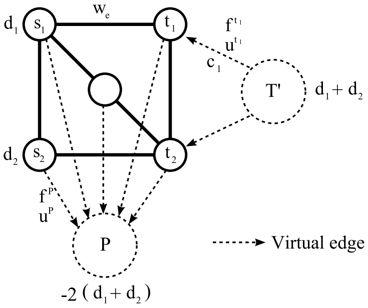


Fig. 4. Transformed network topology for combined optimization.

In this section we refine the FCNFP transformation by manipulating the proxy unit processing costs based on look-ahead information. Instead of assigning the same unit processing cost u^{CP} (resp. u^{SP}) to all potential CP (resp. SP) locations, the unit processing costs for each location $v \in V$ are defined as follows:

$$u^{CP,v} = u^{CP} + \text{look-ahead}(v)$$

$$u^{SP,v} = u^{SP} + \text{look-ahead}(v)$$

We propose two intuitive heuristics for the look-ahead function: Avg(x, v) and Max(x, v). Avg(x, v) selects the x closest nodes from $S \cup T$ relative to v and subsequently computes the average distance from these nodes to v. Max(x, v) is defined in a similar way, but computes the maximum shortest path distance from the x closest nodes to v.

Both heuristics provide a rough estimate of the expected flow forwarding costs *after* the proxy has been reached, tainted with locality information of neighboring sources or targets. If proxy fixed charges are high, usually less proxies are installed and a greater number of clients or servers is aggregated per proxy. This can be reflected in both heuristic functions by raising the value of x.

D. DSSP approximation

The FCNFP is a well-known subclass of the minimum concave-cost network flow problems and unfortunately, it is known to be $\mathcal{NP}$-hard [8]. To overcome the computational complexity inherent to the FCNFP, approximation techniques have been developed. In this paper, we employ the Dynamic Slope Scaling Procedure (DSSP) proposed by Kim and Pardalos [8]. For this reason, we provide a brief overview of this technique. For an in-depth evaluation of this approach including performance evaluations, we refer to [8, 12].

Essentially, the DSSP provides an *iterative* scheme to compute successive *linear approximations* for the original FCNFP. In the k^{th} iteration, the DSSP solves the following linear program:

Minimize

$$\bar{f}^k(x) = \sum_{e \in E} \bar{u}_e^k x_e$$

subject to the same constraints (2) and (3) as the original FCNFP. To find an *initial solution* $\bar{x}_e^0$ ($e \in E$), the coefficients are chosen as follows:

$$\bar{u}_e^0 = u_e, \forall e \in E$$

Besides the initial solution, we also need a coefficient updating scheme and a stopping criterion. The *updating scheme* is defined as follows:

$$\bar{u}_e^{k+1} = \begin{cases} u_e + \frac{f_e}{\bar{x}_e^k}, & \text{if } \bar{\mathrm{x}}_\mathrm{e}^\mathrm{k} > 0 \\ u_e, & \text{if } \mathrm{k} = 0 \text{ and } \bar{\mathrm{x}}_\mathrm{e}^\mathrm{k} = 0 \\ max_{1 \leq l \leq k}\{\bar{u}_e^l | \bar{x}_e^{l-1} > 0\}, & \text{if } \mathrm{k} > 0 \text{ and } \bar{\mathrm{x}}_\mathrm{e}^\mathrm{k} = 0 \end{cases}$$

Now, suppose that at iteration k,

$$\bar{x}_e^{k-1} = \bar{x}_e^k, \forall e \in E$$

From this point, all consecutive solutions are exactly the same because

$$\bar{u}_e^k = \bar{u}_e^{k+1}, \forall e \in E$$

Therefore, no further improvement is possible and the procedure should be halted. After the *stopping criterion* has been satisfied, the best solution can be identified by scanning through the DSSP history and selecting the solution that yields the minimum cost for the original FCNFP.

IV. EVALUATION RESULTS

In this section, we evaluate the performance of the heuristic approach for designing the ASTAS proxy infrastructure. For relatively small networks, we can validate the results obtained by the heuristics by comparing them with exact solutions. Subsequently, Section IV-B discusses proxy placement behavior in a large European reference network. In order to keep this discussion focused, we limit ourselves to the evaluation of the separated heuristic, augmented with the optimization alternatives described in Section III-C. All FCNFP instances are solved by the DSSP approximation technique discussed in Section III-D.

A. Validating the heuristic method

For validating the heuristics, two classes of random graphs are used: Barabási-Albert random graphs[1] and square lattices, both with discrete uniformly distributed edge weights w_e drawn from the set {0.1, 0.2, ..., 0.9}. These two types of graphs cover a wide range of random graphs: lattices are artificial networks with a regular structure, whereas B-A graphs are small world networks that are often used to model large networks (e.g., the Internet) in a realistic way.

A comparison between the heuristics and the exact optimization ILP [10] is depicted in Fig. 5. Results are averaged out over hundred iterations; input parameters for the optimization model are shown in Table 2. Routers providing network access to client sites and target sites are selected randomly and total client demand equals total server capacity. Proxy unit processing costs u^{CP} and u^{SP} are varied using the heuristics discussed in Section III-C. From each node's perspective, we define $\mathrm{Avg(x\%closest)}$ as the average distance to the x% closest sources and target servers. $\mathrm{Max(x\%closest)}$ is defined in a similar way, based on the look-ahead functions proposed in III-C. On Fig. 5, we chose to display all costs relative to the average cost to forward one unit of flow (i.e., a session) over one edge. On the x-axis, "proxy cost" equals the fixed charge for installing either a CP or SP (which are assumed to have equal cost). The following conclusions are drawn:

Table 2. Parameter values for validating the heuristics

Parameter	$\lvert V\rvert$	$\lvert S\rvert$	$\lvert T\rvert$	d_i	c_j
Value	100	10	10	100	100

1) All heuristics follow the same trend as the exact solution: an increasing fixed charge for installing a proxy (either a CP or SP) leads to less proxies being installed and a growing path stretch. By switching on the extra optimizations, near-optimal proxy placement can be achieved.
2) It is possible that heuristics provide a lower path stretch than the optimal solution, at the expense of installing more proxies. As such, the combined solution cost is at least as high as the cost computed by the exact ILP.
3) The most radical optimization strategy that assigns a proxy unit processing cost equal to the maximum shortest path distance from each node to the closest half of sources and targets ($\mathrm{u} = \mathrm{Max(50\%closest)}$), yields the best results when proxy unit costs are high. In this case, nodes located on a less favorable position have proxy unit processing costs that are too high to justify the fixed charge for installing a proxy.
4) Less impacting look-ahead information (e.g., $\mathrm{u} = \mathrm{Avg(20\%closest)}$) yields solutions with a higher number of proxies being installed (and a higher infrastructure cost) . Generally, this leads to a smaller path stretch.
5) The heuristic approach performs significantly better for small-world graphs than for the artificial square lattices. Fortunately, real large networks obey small-world properties [14].

B. Proxy placement in a European reference network

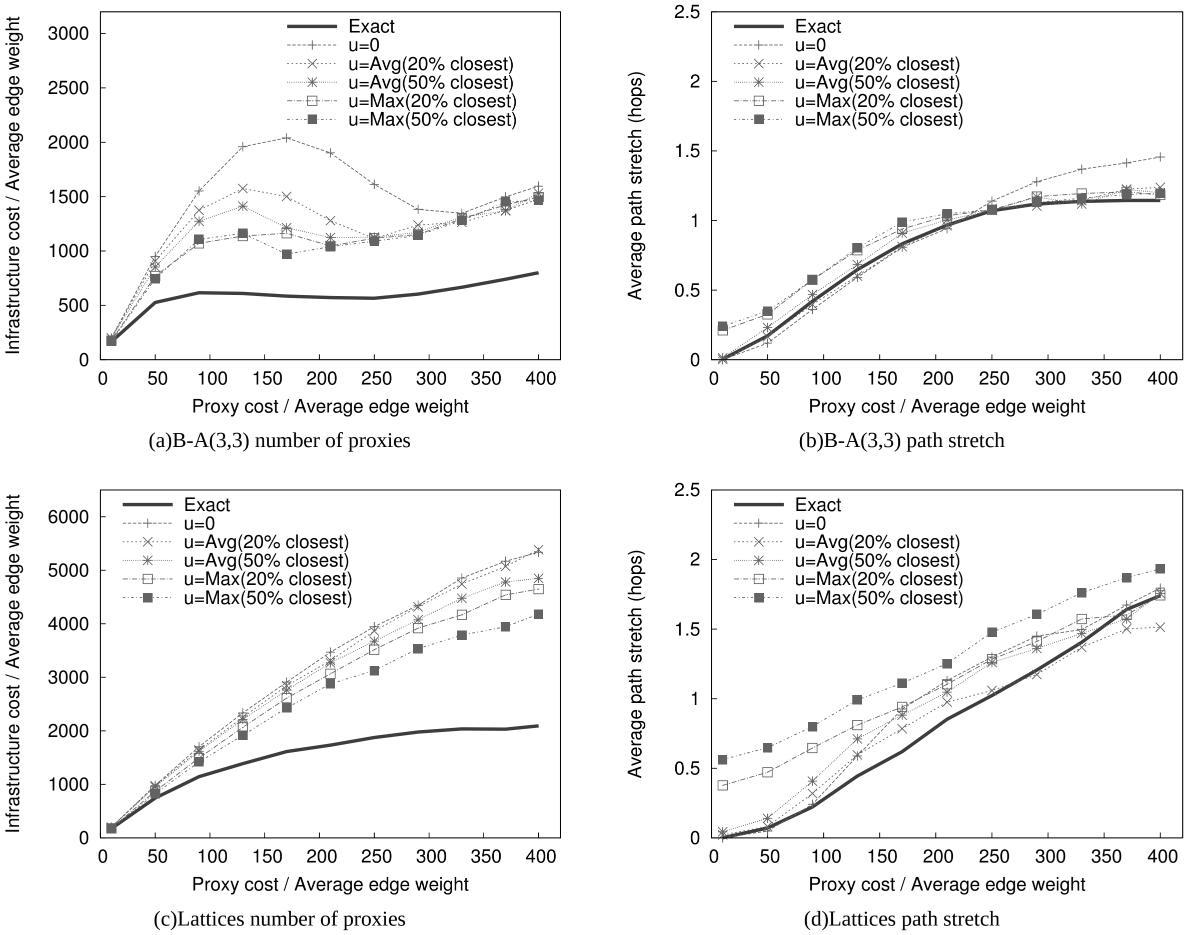
Contrary to the exact ILP, the heuristic approaches enable optimized anycast infrastructure dimensioning and placement for large networks consisting of a few thousands of nodes. In this section, we take advantage of this feature to investigate proxy placement in a pan-European context. Starting from a European reference core network as described in [15], this network is expanded by attaching subnetworks to each city core node. These subnetworks are modeled as B-A(2,2) graphs and represent regional or national networks. The pan-European core network interconnecting major European cities is depicted in Fig. 6(a).

Input parameters for the European simulation model are provided in Table 3. The number of clients is assumed to be larger than the number of resources, while the total resource capacity

Table 3. Pan-European network: input parameters

Parameter	$\lvert V\rvert$	$\lvert S\rvert$	$\lvert T\rvert$	d_i	c_j
Value	1528	20	10	2000	4100

[1] In this paper **B-A(x,x)** stands for the class of Barabási-Albert random graphs [13] with x initial nodes, and during the growing process new nodes are connected by x edges



(a)B-A(3,3) number of proxies

(b)B-A(3,3) path stretch

(c)Lattices number of proxies

(d)Lattices path stretch

Fig. 5. Comparison of exact optimization and heuristic approaches for 100 node Barabási-Albert graphs and square lattices. For each class of graphs, results are averaged out over 100 instances. The total infrastructure cost and the path stretch are related to the cost of installing a single proxy.

is slightly (2.5%) overdimensioned. Comparison with an exact solution is not feasible due to the problem size. Therefore, heuristic results are compared with a static proxy configuration where proxies are deployed in the core nodes. Each dark core node in the European network (see Fig. 6(a)) indicates CP and SP availability in that node.

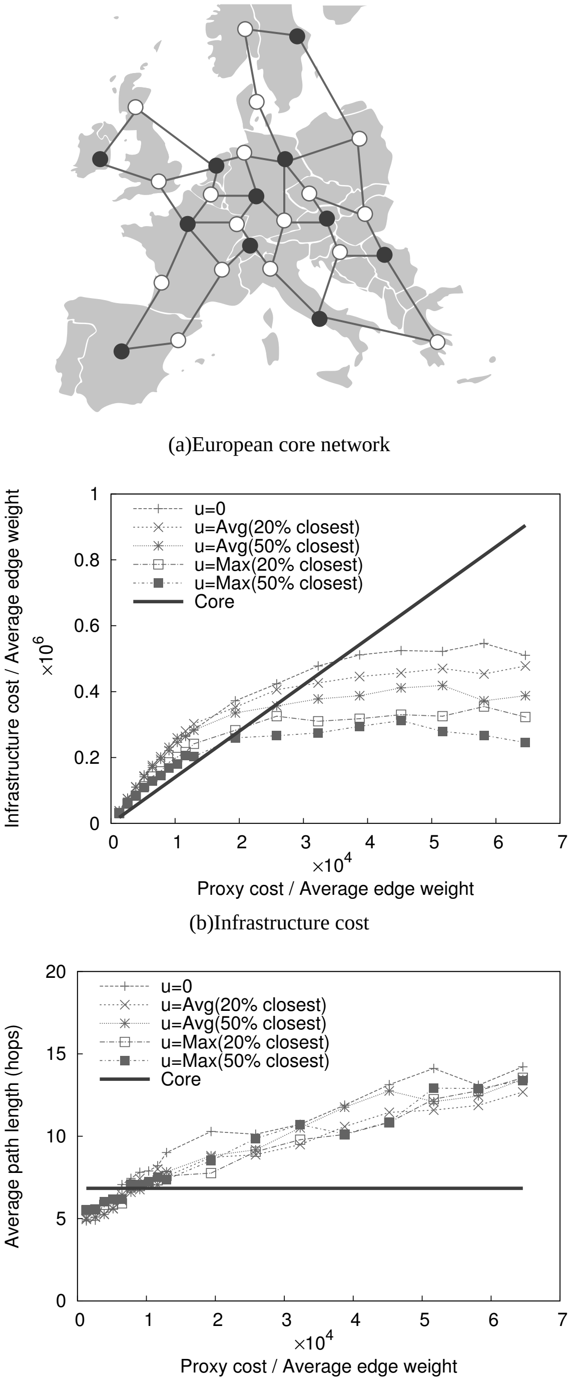
The results presented in Fig. 6 lead to the following observations:

1) Also in the European network, heuristics equipped with look-ahead information perform better than the blind separated heuristic. Again, more pessimistic look-ahead information leads to fewer proxies being installed and a larger path stretch.
2) We have noticed earlier that if proxy installation costs are high compared to network operational costs, $\mathrm{u} = \mathrm{Max}(50\%\mathrm{closest})$ performs best, heuristics with more optimistic look-ahead information install significantly more proxies.
3) In general, high proxy unit costs lead to a small number of proxies being installed in the core nodes, because of their central position. If proxy unit costs are smaller, heuristic dimensioning algorithms significantly reduce the network load (cost) by installing extra proxies on different locations.

V. CONCLUSIONS

Routing scalability concerns and the lack of stateful communications support in native IP anycast prevent its widespread adoption to realize transparent, scalable network services. In this paper, we presented the ASTAS proxy architecture to overcome these issues. Contrary to native anycast, these proxies hide anycast group dynamics from the routing substrate and insert a single aggregated anycast route into the IP routing infrastructure. Moreover, server selection can take into account both network and resource state, which is not possible using IP anycast or a semi-static static proxy infrastructure. Evidently, this opens up new opportunities for anycast traffic engineering and anycast QoS routing.

In this paper, we provide a near-optimal heuristic approach to investigate how large the proxy infrastructure should be and where proxies should be placed in the network to accommodate a given client demand in a network-efficient way. Contrary to an ILP-based formulation, the heuristic can be applied to large networks. Dimensioning studies in a large European reference network have shown that a relatively small number of proxies suffices to effectively accommodate an anycast-based service provisioning platform.



(a)European core network

(b)Infrastructure cost

(c)Average path length

Fig. 6. Results of the dimensioning heuristics in a European reference network.

ACKNOWLEDGMENTS

The work presented in this paper is supported by the EU through the IST Project Phosphorus (`www.ist-phosphorus.eu`). The Phosphorus project is funded by the European Commission under the FP6 contract no. 034115.

Marc De Leenheer thanks the IWT for their financial support through his PhD scholarship. Chris Develder and Filip De Turck thank the FWO for their post-doctoral grant.

REFERENCES

[1] C. Partridge, T. Mendez, and W. Milliken, “RFC 1546: Host Anycasting Service,” November 1993.

[2] S. Sarat, V. Pappas, and A. Terzis, “On the Use of Anycast in DNS,” *SIGMETRICS Performance Evaluation Review*, vol. 33, no. 1, pp. 394–395, June 2005.

[3] D. Katabi and J. Wroclawski, “A Framework for Scalable Global IP-Anycast (GIA),” *ACM SIGCOMM Computer Communication Review*, vol. 30, no. 4, pp. 3–15, October 2000.

[4] H. Ballani and P. Francis, “Towards a Global IP Anycast Service,” *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 4, pp. 301–312, October 2005.

[5] E. Zegura, M. Ammar, Z. Fei, and S. Bhattacharjee, “Application-Layer Anycasting: A Server Selection Architecture and Use in a Replicated Web Service,” *IEEE/ACM Transactions on Networking*, vol. 8, no. 4, pp. 455–466, August 2000.

[6] I. Stoica, D. Adkins, S. Zhuang, S. Shenker, and S. Surana, “Internet Indirection Infrastructure,” *IEEE/ACM Transactions on Networking*, vol. 12, no. 2, pp. 205–218, April 2004.

[7] P. Gray, “Exact Solution of the Fixed-Charge Transportation Problem,” *Operations Research*, vol. 19, no. 6, pp. 1529–1538, October 1971.

[8] D. Kim and P. Pardalos, “A Solution Approach to the Fixed Charge Network Flow Problem Using a Dynamic Slope Scaling Procedure,” *Operations Research Letters*, vol. 24, no. 4, pp. 195–203, May 1999.

[9] W. Feng, F. Chang, W. Feng, and J. Walpole, “A Traffic Characterization of Popular On-Line Games,” *IEEE/ACM Transactions on Networking*, vol. 13, no. 3, pp. 488–500, June 2005.

[10] T. Stevens, F. De Turck, B. Dhoedt, and P. Demeester, “Achieving Network Efficient Stateful Anycast Communications,” in *Proceedings of the 21st International Conference on Information Networking (ICOIN 2007)*, Estoril, Portugal, January 2007.

[11] G. Nemhauser and L. Wolsey, *Integer and Combinatorial Optimization*. New York, United States: Wiley-Interscience, 1988.

[12] D. Kim, X. Pan, and P. Pardalos, “An Enhanced Dynamic Slope Scaling Procedure with Tabu Scheme for Fixed Charge Network Flow Problems,” *Computational Economics*, vol. 27, no. 2–3, pp. 273–293, May 2006.

[13] A. Barabási and R. Albert, “Emergence of Scaling in Random Networks,” *Science*, vol. 286, pp. 509–512, October 1999.

[14] R. Albert and A. Barabási, “Statistical Mechanics of Complex Networks,” *Reviews of Modern Physics*, vol. 74, no. 1, pp. 47–97, January 2002.

[15] S. De Maesschalck, D. Colle, I. Lievens, M. Pickavet, P. Demeester, C. Mauz, M. Jaeger, R. Inkret, B. Mikac, and J. Derkacz, “Pan-European Optical Transport Networks: An Availability-based Comparison,” *Photonic Network Communications*, vol. 5, no. 3, pp. 203–225, May 2003.

Tim Stevens received his M.Sc. degree in Computer Science from Ghent University, Belgium, in June 2001. Until July 2003, he was a database and system administrator for the Flemish public broadcasting company (VRT). At present, Tim Stevens is a Ph.D. student affiliated with the Department of Information Technology of Ghent University. His main research interests include future access network architectures, IPv6, Quality of Service (QoS) and traffic engineering in IP networks, and anycast-based services.

Marc De Leenheer received his M.Sc. degree in Computer Science Engineering from Ghent University, Belgium, in June 2003. He is now a research assistant and Ph.D. student affiliated with the Department of Information Technology at Ghent University and has received a scholarship by the IWT (Institute for Innovation in Science and Technology-Flanders). His main interests include modeling and optimization of Grid management architectures, specifically in the context of photonic networks.

Chris Develder received the M.Sc. degree in computer science engineering and a Ph.D. in electrical engineering from Ghent University (Ghent, Belgium), in July 1999 and December 2003 respectively. From October 1999 on, he has been working in the Department of Information Technology (INTEC), at the same university, as a Researcher for the Research Foundation - Flanders (FWO), in the field of network design and planning, mainly focusing on optical packet switched networks. In January 2004, he left University to join OPNET Technologies, working on transport network design and planning. In September 2005, he re-joined INTEC at Ghent University as a post-doctoral researcher, and as a post-doctoral fellow of the FWO since October 2006. Since October 2007 he holds a part-time professor position at the same institute. He was and is involved in multiple national and European research projects (IST Lion, IST David, IST Stolas, IST Phosphorus, IST E-Photon One). His current research focuses on dimensioning, modeling and optimising optical Grid networks and their control and management. He is an author or co-author of over 45 international publications.

Filip De Turck received his M.Sc. degree in Electronic Engineering from the Ghent University, Belgium, in June 1997. In May 2002, he obtained the Ph.D. degree in Electronic Engineering from the same university. At the moment, he is a part-time professor and a post-doctoral fellow of the F.W.O.-V., affiliated with the Department of Information Technology of the Ghent University. Filip De Turck is author or co-author of approximately 135 papers published in international journals or in the proceedings of international conferences. His main research interests include scalable software architectures for telecommunication network and service management, performance evaluation and design of new telecommunication services. He is in the program committee of several conferences and regular reviewer for conferences and journals in this field.

Bart Dhoedt received a degree in Engineering from the Ghent University in 1990. In September 1990, he joined the Department of Information Technology of the Faculty of Applied Sciences, University of Ghent. His research, addressing the use of micro-optics to realize parallel free space optical interconnects, resulted in a PhD degree in 1995. After a 2 year post-doc in opto-electronics, he became professor at the Faculty of Applied Sciences, Department of Information Technology. Since then, he is responsible for several courses on algorithms, programming and software development. His research interests are software engineering and mobile & wireless communications. Bart Dhoedt is author or co-author of approximately 150 papers published in international journals or in the proceedings of international conferences. His current research addresses software technologies for communication networks, peer-to-peer networks, mobile networks and active networks.

Piet Demeester finished his Ph.D. thesis at INTEC, Ghent University, in 1988. At the same department he became group leader of the activities on metal organic vapor phase epitaxial growth for optoelectronic components. In 1992 he started a new research group on broadband communication networks. The research in this field has already resulted in more than 300 publications. In this research domain he was and is a member of several program committees for international conferences, such as ICCCN, the International Conference on Telecommunication Systems, OFC, ICC, and ECOC. He was Chairman of DRCN '98. He was chairman of the Technical Programme Committee ECOC '01. He has been Guest Editor of three special issues of IEEE Communications Magazine. He is also a member of the Editorial Board of Optical Networks Magazine and Photonic Network Communications. He was a member of several national and international Ph.D. thesis commissions. He is a member of ACM and KVIV. His current research interests include multilayer networks, QoS in IP networks, mobile networks, access networks, grid computing, distributed software, network and service management, and applications (supported by FWO-Vlaanderen, the BOF of Ghent University, the IWT, and the European Commission). He is currently a full-time professor at Ghent University, where he is teaching courses in communication networks. He has also been teaching different international courses.