

A Traffic Engineering Approach for Placement and Selection of Network Services

Reuven Cohen and Gabi Nakibly
 Dept. of Computer Science
 Technion
 Haifa, Israel

Abstract—Network services are provided by means of dedicated service gateways, through which traffic flows are directed. Existing work on service gateway placement has been primarily focused on minimizing the length of the routes through these gateways. Only limited attention has been paid to the effect these routes have on overall network performance. We propose a novel approach for the service placement problem, which takes into account traffic engineering considerations. Rather than trying to minimize the length of the traffic flow routes, we take advantage of these routes in order to enhance the overall network performance. We divide the problem into two sub-problems: finding the best location for each service gateway, and selecting the best service gateway for each flow. We propose efficient algorithms for both problems and study their performance. Our main contribution is showing that placement and selection of network services can be used as effective tools for traffic engineering.

I. INTRODUCTION

As the Internet becomes more prevalent and diverse, there is a growing demand for services that facilitate and enhance interoperability, performance and security of communication between two or more parties. Examples for such services are voice and video conversion, protocol translation, caching, compression, QoS control, authentication, encryption and intrusion detection. Many of these services require the intervention of intermediate service gateways, like firewalls, VoIP gateways, NAT routers, VPN gateways and broadband access servers.

These services are sometime referred to as session-oriented services [1], because they operate on traffic flowing between pairs of source and destination nodes. In the case of a stub AS, some of these source and destination nodes are likely to be edge routers connected to the hosts, whereas in the case of a large transit AS, these nodes are two border routers in the ingress and egress of the AS. The serviced traffic traverses the shortest path from the source to the service gateway, and then the shortest path from the gateway to the destination.

Traditionally, such service gateways have been placed on the boundary of an Autonomous System (AS), since all inter-domain traffic passes through it. However, there is a growing trend to place network services inside the AS. It was first shown by [2] that FTP traffic can be significantly reduced by placing caches in strategic locations inside the AS backbone. Since then, there has been a large volume of work that demonstrate the benefits of well-planned placement strategies in a variety of service contexts [1], [3]–[7]. Such strategies take

into account the distribution of traffic as well as the topology of the AS.

Research on service placement has concentrated mainly on placing the service gateways in a way that minimizes the average length of the traversed routes. Therefore, each flow always selects the service gateway that imposes the shortest possible route. However, this approach does not take into account the reciprocal effect of individual flows, the load imposed on the network links, and the possible existence of hotspots (congested areas) in the network.

Another important trend in recent years is the adoption of traffic engineering inside large ASs. Traffic engineering [8] is related to a set of actions dealing with performance evaluation and performance optimization of operational IP networks. The performance of an operational network is enhanced by addressing traffic oriented performance requirements, such as delay, delay variation, packet loss, and throughput, while utilizing network resources economically and reliably. Most importantly, traffic engineering is used to control and optimize the routing function so that traffic can be steered through the network in the most effective way. This allows to admit more traffic into the network and to support better QoS requirements.

In this paper we consider traffic engineering in the case where some of the traffic flows in an AS have to traverse one or more service gateways before reaching their destinations. These ASs, are likely to suffer from uneven utilization of their resources, *which can only be partially solved if the routes to and from the gateways are selected using traditional traffic engineering schemes*. We propose to facilitate traffic engineering in such networks by judiciously placing the service gateways in the network and selecting the gateway(s) to be traversed by each flow. The main advantages of our scheme compared to traditional traffic engineering schemes are as follows:

- Traditional traffic engineering schemes are usually based on the deployment of an underlying virtual circuit technology, like MPLS, in order to select underutilized links. In contrast, our scheme works very well with the standard shortest path routing.
- The effectiveness of traditional traffic engineering schemes is limited if the gateways are located very close to each other. Our scheme addresses this problem by considering the problem of server placement as an important part of the traffic engineering scheme.

We address the problem by dividing it into the following

two sub-problems:

- 1) The service placement problem: finding the best location for each service gateway.
- 2) The gateway selection problem: selecting the best service gateway to accommodate each flow.

The service placement problem is addressed in the offline context, by considering the long term average distribution of the source-destination traffic for each service type, which can be obtained using traffic matrix estimation techniques [9], [10]. We use this information to decide on the best location for each service gateway. In contrast, the gateway selection problem is addressed in the online setting. That is, each flow is associated with a service gateway, which is determined by current network conditions.

The rest of the paper is organized as follows. In Section II we present detailed scenarios for which our placement and selection schemes are applicable. In Section III we present related work. Section IV presents formal definitions of the problems we address. In Section V we present approximation algorithms, with performance guarantees, as well as efficient heuristics for both problems. In Section VI we present extensive simulation results that demonstrate the significant performance gain achieved by our approach for real and synthetic topologies and various load settings. Finally, Section VII concludes the paper.

II. APPLICATION SCENARIOS

In this section we present two important application scenarios for the problem discussed in the paper.

A. Open Access Networks

Users today are connected to the Internet almost from everywhere, using a variety of wireless and wireline technologies. There is often a distinction between the operator of an access network and the service provider (SP) who provides Internet access and/or other services (like telephony) to the end users. This concept, known as Open Access, is motivated by the following factors:

- The desire for regulations to protect the SPs from anti-competitive practices on the part of the access operators, in order to ensure that access to the Internet and other broadband services remain open to free competition.
- The expertise and assets required from an access operator such as a local or cable company are completely different from those required from an SP.

Figure 1 depicts a typical Open Access network that employs MPLS technology. This network connects the broadband link termination systems (BLTSs) such as cable-modem CMTSs, ADSL DSLAMs, and WiFi access points to the networks of the service providers. One of the most important functionalities of the Open Access network is to decide to which service provider each packet sent by a user should be forwarded. This decision is different from a typical routing decision, because it is not performed according to the IP destination address. This address indicates the final target of the packet, which could be, for example, a web server, that is not necessarily located in the service provider's domain.

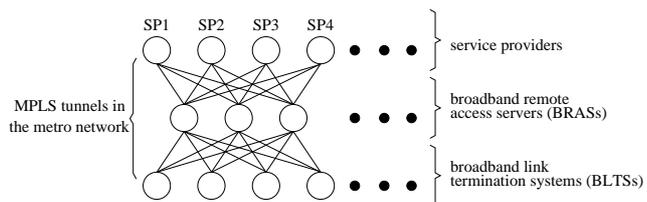


Fig. 1. The schematic structure of a typical open access network

The information on which the decision is based is not included in the BLTS, but rather in another network device, referred to as a broadband remote access server (BRAS). This is for two main reasons:

- There might be hundreds of BLTSs and tens of service providers. Establishing one tunnel, for best-effort traffic, or several tunnels, for multiple QoS classes, between every BLTS and service provider pair, would require too many tunnels.
- The decision as to which service providers a packet should be forwarded is usually based on policy and price considerations which are not in the scope of the BLTSs.

There is at least one MPLS tunnel between every BLTS and BRAS, and one tunnel between every BRAS and service provider. More tunnels can be used to address different service classes. When the BLTS identifies a new flow of a certain user, it needs to select the best BRAS to which this flow will be forwarded. This selection is not dependent on the target service provider since every BRAS has at least one tunnel to every service provider. Obviously, traffic engineering is one of the most important selection criteria, especially for traffic flows that require better than best-effort QoS. This has motivated the model and solutions presented in this paper.

B. VoIP communication through session border controllers

Enterprises and operators that provide VoIP services over their IP networks often use a special device called a session border controller (SBC). When a local user initiates a VoIP call to a remote peer, the call is established through one of the local SBCs, to the destination peer. In certain scenarios, an SBC is involved only in the control plane. However, here we consider the case where it is involved both in the control and data planes. In such a case, two RTP (Real Time Protocol) connections are established for the considered call: between the caller and the SBC, and between the SBC and the remote peer (see Figure 2).

The setup of a VoIP call through an intermediate device, rather than peer-to-peer, might increase the end-to-end delay. However, it has some important – and usually more significant – advantages:

- The codec used by the caller can be changed to another codec, either because the new codec is more efficient, or because the older one is not supported by the remote peer. This case is shown in Figure 2(a).
- The NAT (Network Address Translation) problem can be overcome if the local user uses a non-globally unique IP address. This is because the IP packets sent out of the

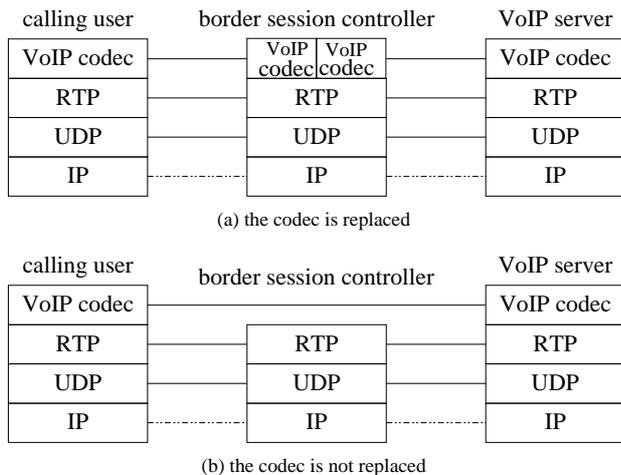


Fig. 2. VoIP communication through a session border controller

local network will carry the globally-unique IP address of the SBC, rather than the non-globally unique IP address of the calling user. This case is shown in Figure 2(b).

- The firewall traversal problem of VoIP sessions can be overcome. Since firewalls are usually closed for UDP traffic, the local administrator can open the firewalls only for UDP connections that are originated in or are destined for the local SBCs, when these are employed. This case is also shown in Figure 2(b).
- Providing QoS from a small set of SBCs to the target peers is easier than providing QoS from each end user to these peers. For example, if MPLS tunnels are used, then the number of required tunnels is smaller by several orders of magnitude.

The concepts presented in this paper in the considered SBC-based VoIP architecture can be taken advantage of as follows. Each VoIP client is configured with a local redirect proxy server. This server only participates in the signaling of the VoIP calls. To set up a new call, the client sends a SIP INVITE message to the proxy. The proxy takes into account the location of the client and of the potential SBCs. It selects one of these SBCs (e.g., using the algorithms proposed in this paper), and responds to the client with a 3XX Redirect message, asking this client to establish the call through the selected SBC. The clients then sends an INVITE message to the selected SBC, and the SBC sends such a message to the remote peer.

III. RELATED WORK

The problem of placing network intermediate devices has been extensively addressed in a vast range of fields. Most notably, much work has been done on the placement of caches, Web proxies [11], [12] and mirror servers [13]–[15]. Usually, these devices respond to the service request themselves; they do not forward any traffic to a destination node. However, as noted by [1], they can also be considered as private cases of session-oriented services in which the volume of traffic changes after it passes through the service gateway. Hence, our results are applicable to such services as well.

Another field where intermediate device placement has been researched is stream processing systems [16], [17]. These distributed systems are composed of autonomous devices that operate on continuous data streams. Each device performs a single function and then passes the traffic to other operators or to the final consumers. Applications include sensor networks, network management, and location-tracking. The goal here is to select appropriate devices while dynamically adjusting to the network load, thereby minimizing service latency and improving resiliency.

The main focus of all the above work is placement and selection of intermediate devices while minimizing bandwidth consumption or minimizing the distance (average or maximum) between the devices and the end users. The most notable algorithms proposed in these works are graph-theoretic algorithms, which are based on approximation algorithms for the K -center and K -median problems [1], [14]. Other algorithms are based on a greedy strategy [13], [14] or on the connectivity degree of the routers and the ASs [13], [15]. However, no work has yet addressed the reciprocal effect of the serviced traffic flows. Ignoring this important issue may result in the creation of network hotspots, which may lead to exponential degradation in the service latency, or even to denial of service.

Some papers, such as [3], [18], suggest placing the service gateways in a distributed manner. This approach is claimed to be scalable, efficient, and adaptable to dynamically changing network conditions. However, it may be more prone to deployment difficulties, as stateful services have to be forwarded from one gateway to another, and clients need to continuously relocate the gateways. Furthermore, it was shown in [6], [19] that Web traffic in the long term is fairly stable, with relatively moderate changes. This implies that determining the location of gateways using long-term traffic distribution statistics can be very effective.

Placement and selection of relay nodes in a network is also addressed in the field of anycast servers. Anycast is a routing mechanism that forwards a packet to at least one node from a set of possible destinations. In [20], the authors investigate selection algorithms of already deployed anycast relay servers which forward the received traffic to the final destination. Their main conclusion is that the best selection algorithm is dependent on the placement algorithm for the servers. Other papers, such as [21], [22], suggest selection algorithms that minimize response time.

Relay nodes are also deployed in sensor and wireless networks. Relay nodes with higher energy capacity are deployed in these networks in order to shorten the transmission ranges of regular, more energy-constrained, nodes and to alleviate the need for them to relay data for other nodes. In [23]–[25], the authors suggest placement algorithms that optimize various metrics such as average node lifetime and average node congestion.

Finally, a concept called N -hub shortest-path routing is presented and explored in [26]. In this routing paradigm, a traffic flow can be routed through up to N intermediate nodes (hubs) before reaching its destination, while traversing the shortest paths between them. N -hub shortest-path routing was

shown to be an effective approach for ensuring load balancing and achieving better utilization of network resources. In this paper we build on this result and leverage the benefits of the N-hub shortest-path routing scheme.

IV. PROBLEM MODEL

The optimization metric we consider maximizes the volume of admitted traffic, subject to an upper bound on the load imposed on each link. This criterion is relevant for service providers that must provide guaranteed bandwidth to some of the flows, such as video or voice streams, which have strict QoS constraints. Our approach can also be applied to several other optimization metrics, such as minimizing the delay, minimizing the average packet loss, or minimizing the maximum load.

The metric selected above is good for achieving load balancing. However as in other traffic engineering schemes our scheme may lengthen the routes taken by the flows. Nonetheless, this has no adverse effect as long as the network resources are indeed better utilized and the total accommodated traffic increases. Moreover, since our scheme reduces the load on overutilized links, it also reduces the queuing delays in the network. Therefore, end-to-end delay is likely to be reduced despite of the longer routes.

We now formulate the two problems described in Section I: placement of service gateways, and selection of service gateways. We assume that the network offers several services, each of which is offered by a set of service gateways. For the sake of simplicity, we assume that each flow demands at most one service, which is rendered by one gateway from a set of gateways that can deliver this service. Still, all the algorithms described in the next sections can be extended to address the case where a flow may demand multiple services and have, therefore, to traverse multiple gateways. Note that a flow may require no service at all. In this case, it should not be routed through a gateway, but through the direct path from its source to its destination. We assume that between every two nodes in the network (end nodes or service gateways) there exists a single predetermined route, to be used by the traffic between them. However, we make no specific assumptions regarding this route. Therefore, such a route can be the “standard” shortest path between the two nodes, or a pre-established MPLS tunnel. To avoid having to forward states from one service gateway to another, and to avoid packet reordering or route oscillation, we require that the same gateway serve each flow for the entire flow duration. A flow serviced by a gateway is said to be routed through that gateway. We call the assignment of hubs (service gateways) to flows in \mathcal{F} “1-hub routing” of \mathcal{F} .

Definition 1: Let $\mathcal{H} \subseteq V$. An instance of 1-hub routing is \mathcal{H} -limited for a given set of flows \mathcal{F} , if only nodes from \mathcal{H} serve as hubs for the flows in \mathcal{F} .

A route in an \mathcal{H} -limited 1-hub routing is denoted a *1-hub(\mathcal{H}) route*. The routing domain is represented by a directed graph, $G = (V, E)$, where V is a set of routers and E is a set of directed links. A bidirectional link is represented by two counter edges. Let $u(e) \forall e \in E$ be the bandwidth capacity

of link e . Let $\mathcal{F} \subseteq V \times V$ be a set of flows and $t(f) \forall f \in \mathcal{F}$ be the bandwidth demand of flow f . Let S be the set of service types deployed in the network. Let $\mathcal{F}_s \subseteq \mathcal{F}$, $s \in S$, be the subset of flows that require a service of type s . $\forall s_1, s_2 \in S, \mathcal{F}_{s_1} \cap \mathcal{F}_{s_2} = \emptyset$ and $\bigcup_{s \in S} \mathcal{F}_s \subseteq \mathcal{F}$. Let $k_s \leq |V|$ is the maximum number of service gateways for service of type s . Let $p_{uv}, \forall u, v \in V$, be a predetermined path between nodes u and v . A flow (o, d) serviced by a gateway h must be routed on the path p_{oh} and then on the path p_{hd} . A flow (o, d) that does not demand a service must be routed on the path p_{od} . Each gateway can be deployed on one of the network routers. The load of a link is the total traffic the link carries divided by its capacity.

The *placement problem* is defined as follows. For every $s \in S$, find $\mathcal{H}_s \subseteq V$, where $|\mathcal{H}_s| \leq k_s$, for which there exists an \mathcal{H}_s -limited 1-hub routing of $\mathcal{F}_s \subseteq \mathcal{F}_s$, and a direct routing of subset $\mathcal{F}' \subseteq \mathcal{F} \setminus \mathcal{F}_s$, such that the total load imposed on each link does not exceed the threshold L , and the total bandwidth of $\bigcup_{s \in S} \mathcal{F}_s \cup \mathcal{F}'$ is maximized.

The *selection problem* is defined as follows. For every $s \in S$, let $\mathcal{H}_s \subseteq V$ be the set of (already placed) service gateways. For every $s \in S$, for every $f \in \mathcal{F}_s$, either assign a service gateway $h \in \mathcal{H}_s$ or reject it, and for every flow $f \notin \bigcup_{s \in S} \mathcal{F}_s$, either route f directly to its destination or reject it, such that the sum of the bandwidth demands of the admitted flows is maximized, and the load imposed on every link does not exceed the threshold L .

Both problems are NP-complete even for the special case where $|S| = 1$, $\mathcal{F}_1 = \mathcal{F}$ and $\forall u, v \in V p_{uv}$ is the shortest-path between u and v . To prove that the decision variant of the placement problem is NP-complete, consider a slightly different problem: Find a set $\mathcal{H} \subseteq V$, where $|\mathcal{H}| \leq k$, for which there exists an \mathcal{H} -limited 1-hub routing of \mathcal{F} , such that the maximum link load is minimized. The decision variant of this problem is deciding for a given instance whether there exists a suitable set of hubs \mathcal{H} for which there is an \mathcal{H} -limited 1-hub routing for *all* the flows while ensuring that the maximum load on a link does not exceed a threshold L . In [26] it is shown that finding “1-hub routing” under a similar optimization criterion but with is no upper bound on k is NP-complete. This implies that the problem with $k = |V|$ is NP-complete as well. Hence, the problem with an arbitrary value of k must also be NP-complete.

Getting back to our original problem, the decision variant of this problem is deciding for a given instance whether there exists a set $\mathcal{H} \subseteq V$ of hubs, where $|\mathcal{H}| \leq k$, for which there exists an \mathcal{H} -limited 1-hub routing for a subset of flows whose total bandwidth demand is not less than a threshold T , such that the load imposed on each link does not exceed L . Let us assume that there is an optimal polynomial algorithm for this problem. If we take T to be the sum of the bandwidth demands of all the flows in \mathcal{F} , we can decide in a polynomial time whether there exists a suitable 1-hub routing that routes every flow in \mathcal{F} while keeping the maximum load imposed on each link below L . However, such an algorithm can also solve in polynomial time the decision variant of the problem shown earlier to be NP-hard.

It can be easily shown that the offline decision variant of

the selection problem is NP-complete, using, again, a reduction from the routing problem presented in [26].

As already said, we address the selection problem in the online context. A competitive ratio for an online algorithm is defined as the worst case ratio, over all sequences of flows, between the value of the solution found by the algorithm and the value of the solution found by an optimal offline algorithm [27]. In [26] it is shown that if the optimization criterion is to minimize the maximum load, and the flows have a limited and unknown duration, the best and worst competitive ratios are $O(|E|)$. It can be shown that a similar proof also applies for our selection problem. Hence, in the following discussion we assume that each flow has an unlimited duration or, alternatively, known duration. This assumption allows us to differentiate between the performance of the various algorithms by their competitive ratios.

V. ALGORITHMS FOR THE PLACEMENT AND SELECTION PROBLEMS

A. Algorithms for the Placement Problem

In this subsection we present approximation algorithms and heuristics for the placement problem. We first propose an approximation algorithm whose worst-case performance is upper bounded.

Algorithm V-A.1: We start by formulating the problem as an integer linear problem. Let $\mathcal{F}_S = \bigcup_{s \in S} \mathcal{F}_s$. The following variables and parameters are defined:

- x_{if} – a binary variable, whose value is 1 if node $i \in \{V, \phi\}$ is assigned as a hub for the traffic of flow f , and 0 otherwise. The symbol ϕ denotes a dummy hub. Namely, if ϕ is assigned to f , then f does not pass through any hub.
- X_f – a binary variable, whose value is 1 if flow f is admitted, and 0 otherwise.
- h_i^s – a binary variable, whose value is 1 if node i is used as a hub for some flow in \mathcal{F}_s , and 0 otherwise.
- z_{if}^e – for every flow $f = (o, d)$, node $i \in V$ and link e , $z_{if}^e = 1$ if e is on p_{oi} or p_{id} , and 0 otherwise. If $i = \phi$ $z_{if}^e = 1$ if e is on p_{od} , and 0 otherwise.

The target function is:

$$\text{maximize } \sum_f t(f) \cdot X_f$$

subject to the following constraints:

- (a) $\forall f \quad \sum_{i \in \{V, \phi\}} x_{if} \geq X_f$
- (b) $\forall f \in \mathcal{F}_S \quad x_{\phi f} = 0$
- (c) $\forall f \notin \mathcal{F}_S \quad x_{\phi f} = 1$
- (d) $\forall e \quad \sum_{i \in \{V, \phi\}, f} x_{if} \cdot z_{if}^e \cdot t(f) \leq Lu(e)$
- (e) $\forall i \in V, s, f \in \mathcal{F}_s \quad x_{if} \leq h_i^s$
- (f) $\forall s \quad \sum_{i \in V} h_i^s \leq k_s$
- (g) $\forall i \in \{V, \phi\}, f, s \quad x_{if} \in \{0, 1\}, X_f \in \{0, 1\}, h_i^s \in \{0, 1\}$

The linear relaxation of the program allows each variable to be assigned any real value in $[0, 1]$. This implies that the requirement to route the entire bandwidth demand of every flow $f \in \mathcal{F}_S$ on a single route is now relaxed. Note, however, that a flow $f = (o, d) \in \mathcal{F} \setminus \mathcal{F}_S$ (i.e., a flow served by no gateway) still can be routed only on p_{od} . Though not all its

bandwidth demand may be accommodated. After finding an optimal solution for the relaxed linear program, we have for every flow $f = (o, d) \in \mathcal{F}_S$ a set of hubs Γ_f through which $t(f)$ or part of it is routed. An empty Γ_f indicates that f is not admitted. Every hub $h \in \Gamma_f$ defines a route from o to d , which consists of p_{oh} and p_{hd} . Each such a route carries a fraction of x_{if} from $t(f)$. In order to convert the solution to the relaxed linear program into a solution to the original integer program, each hub $h \in \Gamma_f$ is associated with a weight that equals to that fraction. The sum of the weights for every Γ_f equals X_f .

The weight of hub i in every Γ_f is rounded to 1 with probability x_{if}/γ , where γ is some constant larger than 1 whose value will be discussed later, and to 0 with the complementary probability. There might be flows for which more than one hub is chosen for routing the whole demand, and other flows for which no hub is chosen. In the former case, one of the hubs is arbitrarily selected, whereas in the latter case the flow is not admitted for routing. These decisions are made independently for each flow f . A flow $f \in \mathcal{F} \setminus \mathcal{F}_S$ that requires no service is admitted with probability $x_{\phi f}/\gamma$. ■

In appendix A we prove that there exists a value γ for which the above probabilistic algorithm can be transformed to a deterministic one that yields a feasible solution. That is, the number of service gateways does not exceed k and the load imposed on every link does not exceed 1. We also show that the admitted bandwidth of the solution is not smaller than $\Omega(\max\{(y^*)^2/|E|, y^*/\sqrt{|E|}\})$, where y^* is the bandwidth admitted by an optimal solution for the relaxed problem.

Note that in certain cases, when the number of the service gateways is small compared to the number of flows, the service gateways may become congestion hotspots. In these cases, the traffic that can be admitted into the network is bounded by the capacity of the links surrounding the service gateways. This gives rise to the following heuristic, which maximizes this bandwidth.

Algorithm V-A.2: Let $D^v = \min\{D_{in}^v, D_{out}^v\}$, where $D_{in}^v = \sum_{e \in E | e=(n,v)} u(e)$ and $D_{out}^v = \sum_{e \in E | e=(v,n)} u(e)$. We call D^v the capacity of v , and $\sum_{f \in \mathcal{F}_s} t(f)$ the bandwidth demand of s . Set all vertex capacities and service bandwidth demands as unassigned. First, assign the vertex with the highest unassigned capacity to the service with the highest unassigned bandwidth demand. Then, subtract the unassigned capacity of the vertex and the unassigned bandwidth demand of the service accordingly. These two steps iterate until each service is assigned k_s vertices. ■

Note that Algorithm V-A.2 does not take into account the distribution of the flows in the network. It only maximizes the capacity of the service gateways, in order to reduce their load.

As mentioned above, the goal of most placement algorithms proposed in the past is to minimize the average or maximum routes of the flows. Algorithms based on approximations for the K -center and K -median problems have been proposed in [1], [14]. They can be summarized as follows.

Algorithm V-A.3: For every service $s \in S$, construct an instance of the K -median/ K -center problem from an instance of the placement problem. For every service $s \in S$, every flow $f = (o, d) \in \mathcal{F}_s$ is considered as a client. Each router $v \in V$

is considered as a site. The distance between a client and a site is set to the length of the path in G between the source of the corresponding flow and the site, plus the length of the path in G between v and the destination of the corresponding flow. Then, use an approximation algorithm for the K -median or the K -center problem (e.g., [28]) that selects a set of k_s vertices to be used as the locations for the gateways of service s . ■

These algorithms are considered computationally intensive. Hence, a greedy placement strategy was proposed as an efficient alternative [13], [14].

Algorithm V-A.4: For every service $s \in S$, choose the k_s locations in k_s iterations. In each iteration select the location which, in conjunction with the locations selected in previous iterations, imposes the smallest average of route lengths for the flows in \mathcal{F}_s . ■

B. Algorithms for the Selection Problem

We now present algorithms for the online selection problem. When a flow is introduced into the network, the selection algorithm can either reject it or admit it. If necessary, it can also assign to the flow a service gateway. The service gateway is selected from a set \mathcal{H}_s of gateways that offer the required service and whose locations have been determined in advance. The rejection and assignment decisions are made without being aware of forthcoming flows. Furthermore, it is assumed that an active flow cannot be stopped or reassigned to a different gateway.

The selection problem can be translated into a special case of a well-known traffic engineering problem called the *online unsplittable flow problem* [29]. In this problem, every flow may be routed over an arbitrary route. In our problem, a flow may only be routed over a limited set of routes that must pass through one of the service gateways, or over a single direct path. In the following, we extend the results for the general unsplittable flow problem to address our selection problem.

We first present gateway selection schemes employed in previous work (e.g. [1], [13], [14]).

Algorithm V-B.1: If $f \in \mathcal{F}_s$, select from \mathcal{H}_s the gateway that imposes the shortest path on the flow. ■

The main advantages of this algorithm are its simplicity and its ability to minimize the consumption of network resources. Indeed, the scheme performs very well in practice when traffic is distributed evenly across the network.

Another example of an algorithm employed in previous work is the following closest gateway algorithm.

Algorithm V-B.2: If $f \in \mathcal{F}_s$, select the gateway from \mathcal{H}_s that is closest to the source of the flow. ■

This scheme does not require the source to know the network topology, but only its distance to every service gateway.

The above algorithms belong to a class of algorithms that do not take into account the load distribution but only the network topology. These algorithms employ only one path between a source-destination pair, regardless of possible network hotspots. Consequently, they might create congestion if the traffic is not evenly distributed.

A more sophisticated class of algorithms takes into account not only the network topology but also the current load on

every link, [29], [30]. We now adapt the algorithm presented in [29] to our selection problem. The algorithm assumes that $t_{max} \leq u_{min}/P$, where t_{max} and u_{min} are the maximal bandwidth demand and minimal edge capacity, respectively, and $P \geq 2$. The algorithm yields a competitive ratio of $O(P|V|^{1/P})$.

Algorithm V-B.3: Let D denote the length of the longest 1-hub route in G . Choose one of the following two routing methods with equal probability, and route each incoming flow request accordingly.

- 1) Set $\mu = 2D$. If the flow's bandwidth demand is larger than $\frac{u_{min}}{P+1}$, then route the flow over π and set $\forall e \in \pi \quad L(e) = L(e) + \frac{1}{\frac{P u(e)}{u_{min}}}$, if one of following conditions holds:

- $f \in \mathcal{F}_s$ and there exists a 1-hub(\mathcal{H}_s) route π for f for which $\sum_{e \in \pi} (\mu^{L(e)} - 1) < D$ holds,
- $f = (o, d) \notin \mathcal{F}_S$ and $\sum_{e \in \pi} (\mu^{L(e)} - 1) < D$ holds for $\pi = p_{od}$.

Otherwise, reject this flow.

- 2) Set $\mu = (2D)^{1+\frac{1}{P-1}}$. If the flow's bandwidth demand is less than $\frac{u_{min}}{P+1}$, then route the flow over π and set $\forall e \in \pi \quad L(e) = L(e) + \frac{t(f)}{u(e)}$, if one of following conditions holds:

- $f \in \mathcal{F}_s$ and there exists an 1-hub(\mathcal{H}_s) route π for the considered flow for which $\sum_{e \in \pi} (\mu^{L(e)} - 1) < D$ holds,
- $f = (o, d) \notin \mathcal{F}_S$ and $\sum_{e \in \pi} (\mu^{L(e)} - 1) < D$ holds for $\pi = p_{od}$.

Otherwise, reject this flow. ■

It can be shown that this algorithm achieves a competitive ratio of $O(P|V|^{1/P})$ for our selection problem. The proof is similar to the one presented in [29] because the following claims always hold: 1) An admitted flow is routed on a path whose cost is less than D ; 2) A rejected flow has no path whose cost is smaller than D . As in [29], a cascade network, in which all simple routes can also be viewed as 1-hub routes, can be used to show that the above algorithm achieves the best competitive ratio.

One drawback of this algorithm is that the current load on every network link cannot be delivered to the nodes in real time. If this information is not available, the algorithm's performance might be impaired as a result.

To overcome this drawback, algorithms with some a priori knowledge about forthcoming flows are presented in [31] and [32]. The algorithm in [31] knows only the source-destination pairs of future flows; it predicts where hotspots are and diverts traffic to less congested areas. The algorithm in [32] also knows the average traffic between every pair of nodes, additional information which leads to improved performance over [31]. We take advantage of this result, and present the following heuristic for the selection problem.

Algorithm V-B.4: Given a set of service gateways \mathcal{H}_s $s \in S$, solve the relaxed linear program presented in Section V-A while setting the variable h_i^s to 1 if $i \in \mathcal{H}_s$, and to 0 otherwise. The target value of the solution is a lower bound for the optimal offline selection problem. Let x_{if}^* and X_f^* be

the values assigned by the solution to variables x_{if} and X_f , respectively. Admit a flow f with probability X_f^* . If f requires a service, gateway i is selected for the flow with probability x_{if}^*/X_f^* . ■

In order to solve the relaxed linear program, a source node need only know the long-term average traffic distribution. It does not require up-to-date information regarding the load of the links.

VI. SIMULATION STUDY

In this section we evaluate the performance of the placement and selection algorithms presented in Section V. We use router-level AS topologies, generated with the Barabasi-Albert model [33], using the BRITE simulator [34]. The Barabasi-Albert model captures two important characteristics of AS topologies: incremental growth and preferential connectivity of new routers to well-connected existing routers. These characteristics yield a power-law degree distribution of the routers.

To validate our results, we also use actual ISP topologies, as inferred by the RocketFuel project [35]. The bandwidth for each link is based on [36]. The Rocketfuel topologies include between several dozens to several hundreds routers, whose average link degrees is less than 3. For each synthetic or real AS topology, we generate a traffic matrix according to a power-law distribution. A traffic matrix and a network topology form together one instance of the placement problem. For each such an instance, we determine the locations of the service gateways using the following algorithms:

- 1) The probabilistic approximation algorithm, Algorithm V-A.1, referred to in the following as Prob.
- 2) The maximum bandwidth location heuristic, Algorithm V-A.2, referred to in the following as Max-BW.
- 3) The K -median algorithm, Algorithm V-A.3, referred to in the following as K -median.
- 4) The greedy algorithm, Algorithm V-A.4, referred to in the following as Greedy.
- 5) A random algorithm, which chooses the location for each gateway randomly from the entire set of network nodes, referred to in the following as Rand-Loc.

To solve the linear programs in algorithm Prob, we use the Lp_Solve software [37].

Due to lack of space, most of our simulations focus on the case where all the flows require the same service. At the end of the section we validate these results we with simulation results for the more general case. A sequence of flows is generated using the average traffic distribution given by the traffic matrix. Taking the average traffic between two vertices, we determine the average life-time of every flow, the average bandwidth demand for every flow, and the average flow inter-arrival times. Throughout the simulation study we change the offered load in the network by adjusting the network-wide average of these three parameters.

Each flow is associated with the following parameters: a source node, a destination node, bandwidth demand, arrival time, and time duration. The network topology, the sequence of flows, and the locations of the k service gateways form together one instance of the selection problem. For each such

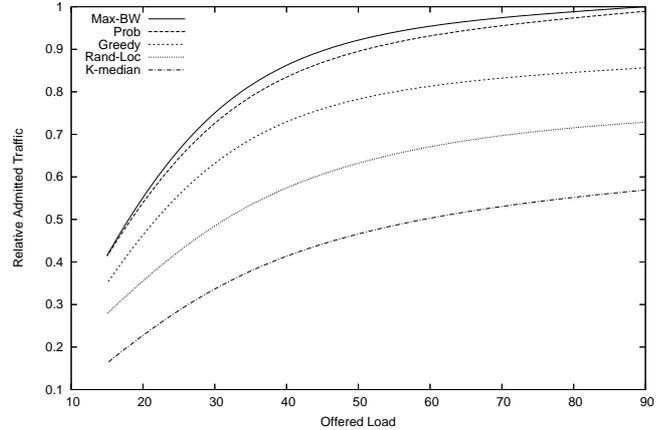


Fig. 3. The potential effectiveness of the various placement schemes

instance, we determine the allowable traffic volume in the network using each of the following selection algorithms:

- 1) Shortest Path, Algorithm V-B.1, referred to in the following as SP.
- 2) Closest gateway, Algorithm V-B.2, referred to in the following as Closest.
- 3) Algorithm V-B.3, which uses the exponential link weights, referred to in the following as Exp.
- 4) The optimal estimation heuristic, Algorithm V-B.4, referred to in the following as Est-Opt.
- 5) A random algorithm that chooses a random service gateway, referred to in the following as Rand-Sel.

For each selection scheme, the flow is admitted if the route through the selected gateway does not violate any of the network capacity constraints. Otherwise, it is rejected.

Every simulation scenario in this section is tested about 400 times: 10 different network topologies were generated and 40 traffic matrices used for each topology. For all simulation results, the widths of the confidence intervals of the expected admitted traffic (using a confidence level of 95%) were less than 5% of the average value.

We start by evaluating the placement algorithms. For each set of gateway locations selected by a placement scheme, we solve the relaxed offline selection problem described in Section V-B. The optimal solution for this problem gives an upper bound on the traffic volume that can be admitted into the AS for the specific gateway locations. This allows us to evaluate the potential effectiveness of every placement scheme, regardless of the selection scheme. Figure 3 depicts the results for a medium-size AS, with 50 routers whose average degree is 4 links. Each placement scheme produces a set of 7 gateway locations. The y -axis represents the total accepted traffic divided by the maximum admitted traffic from all simulation scenarios. The x -axis represents the offered load, which is the average amount of bandwidth that needs to be serviced at any given time between two nodes (i.e., the average bandwidth demand per flow multiplied by the average flow life-time divided by the average flow inter-arrival time).

It is evident from the graph that Prob and Max-BW have

the best potential effectiveness. The potential effectiveness of Prob is attributed to its ability to place the gateways while taking the traffic distribution into account. The fact that Max-BW has roughly the same potential effectiveness is somewhat unexpected, because it does not take into account traffic distribution. In this simulation setting, however, we note that the service gateways create bottlenecks in the network. Therefore, by increasing the bandwidth around the gateways, Max-BW is able to increase the admitted traffic. To verify this, we also measure what percentage of the rejected flows were rejected because of congestion on the links of the service gateways. On the average, this percentage is 75% for Max-BW, and 85% for every other algorithm. This proves that a majority of the rejected flows is indeed attributed to congestion around the service gateways.

The potential effectiveness of Greedy and K-Median is, respectively, 20% and 45% less than the potential effectiveness of Prob and Max-BW. This is because these algorithms aim to minimize the average length of the chosen routes. Hence, they place the gateways close to congested areas in the AS. K-median performs worse than Greedy because K-median is much more successful in minimizing the average route length. The Rand-Loc scheme performs better than K-median, since it places the gateways uniformly throughout the AS, thereby allowing some degree of load balancing.

Next, we evaluate the combined performance of the placement and selection algorithms. Again, we consider a medium-size AS with 7 service gateways and 50 routers whose average degree is 4. For every set of service gateways determined by the placement algorithms, we apply each of the selection algorithms presented above. The graphs in Figure 4 depict the results. To compare the performance of the various combinations, we use the following relative performance metric: the ratio between the bandwidth of flows admitted by the placement and selection algorithms and the bandwidth of flows admitted by the same placement algorithm combined with the optimal solution of the relaxed offline selection. This relative performance forms the y -axis of all the graphs in Figure 4, while the offered load forms the x -axis.

First, it is evident from all the graphs that the relative performance of each selection algorithm is roughly the same for all placement algorithms. The selection algorithm that exhibits the best performance is Est-Opt. For every placement algorithm, Est-Opt admits flows whose bandwidth is almost 80% of the bandwidth of flows admitted by the optimal solution. This can be explained by the fact that Est-Opt is the only selection algorithm that is aware of the long-term average traffic distribution. Surprisingly, Rand-Sel performs roughly the same as Exp, despite the fact that Exp is aware of the AS topology and the current link loads. This can be attributed to the fact that all gateway links are highly congested. Therefore, it does not matter which gateways are selected, as long as the load is evenly distributed among them.

It is also evident that SP and Closest exhibit the worst performance. This is, of course, because both schemes select only one gateway for each source-destination pair, which leads to poor load balancing. Closest is inferior to SP since its selection imposes longer routes, which consumes more

network resources.

We can conclude from Figure 4 that the placement algorithms Prob and Max-BW combined with the selection algorithms Est-Opt, Exp and Rand-Sel exhibit better performance than every other combination.

We now take a closer look at the performance of the above mentioned combinations, by changing the number of service gateways and the link density. Again, we consider two AS types, each with 50 routers. One type has an average link degree of 4 while the other has an average link degree of 7. We change the number of service gateways for every AS and measure the traffic admitted by the algorithms. Figure 5 depicts the results. The y -axis in the graphs represents the admitted traffic divided by the offered load, while the x -axis represents the number of gateways.

It is evident from both graphs that when the number of service gateways increases, so does the volume of admitted traffic. This indicates, again, that when the offered load is sufficiently high, the gateways are likely to create bandwidth bottlenecks. As the number of gateways increases, the bandwidth they can service increases as well, and their immediate vicinity becomes less congested. When the link density is 4, the increase in traffic admitted as a result of the increase in the number of gateways is more moderate. This is because when there are many service gateways, the bottlenecks are shifted from the gateway areas to other areas of the AS. However, when the link density is 7, the AS has much greater capacity, and the gateways impose again bandwidth bottlenecks. From Figure 5(a) we can see that when the service gateways no longer constitute the bandwidth bottleneck, Max-BW does not perform as well as Prob, and Exp performs better than Rand-Sel. This is because the gateways are less congested, so that judicious placement and selection algorithms that take into account the traffic load – Prob and Exp, for example – perform better. Another important point is that when the link density increases, Exp performs better relatively to the other algorithms. This is related to the network links being less congested. Hence, a prudent decision that takes into account the varying network loads achieves better load balancing.

To validate the results from the synthetic graphs, we run similar simulations, but this time over real AS topologies as inferred by the RocketFuel project [35]. The results for these simulations are shown in Figure 6. The topology studied in Figure 6 is of the Exodus ISP, which consists of 80 routers and 147 links (link density = 1.8). Comparing Figure 6 to Figure 5 reveals that the considered combinations of algorithms have similar performance to what we found in the synthetic graphs. To further validate our results, we run simulations with a different demand model, called the gravity model [38], [39]. In this model the offered load from node u to node v is proportional to the product of the total traffic volume exiting u and the total traffic volume entering v . The results for these simulations are shown in Figure 7. The AS topology in this study has 30 routers and link density of 4. It is generated using the BRITE simulator just as described previously. For each router in the AS we use the Zipf distribution to determine the total amount of traffic entering and exiting the router. By comparing Figure 7 to Figure 5 we see that the considered

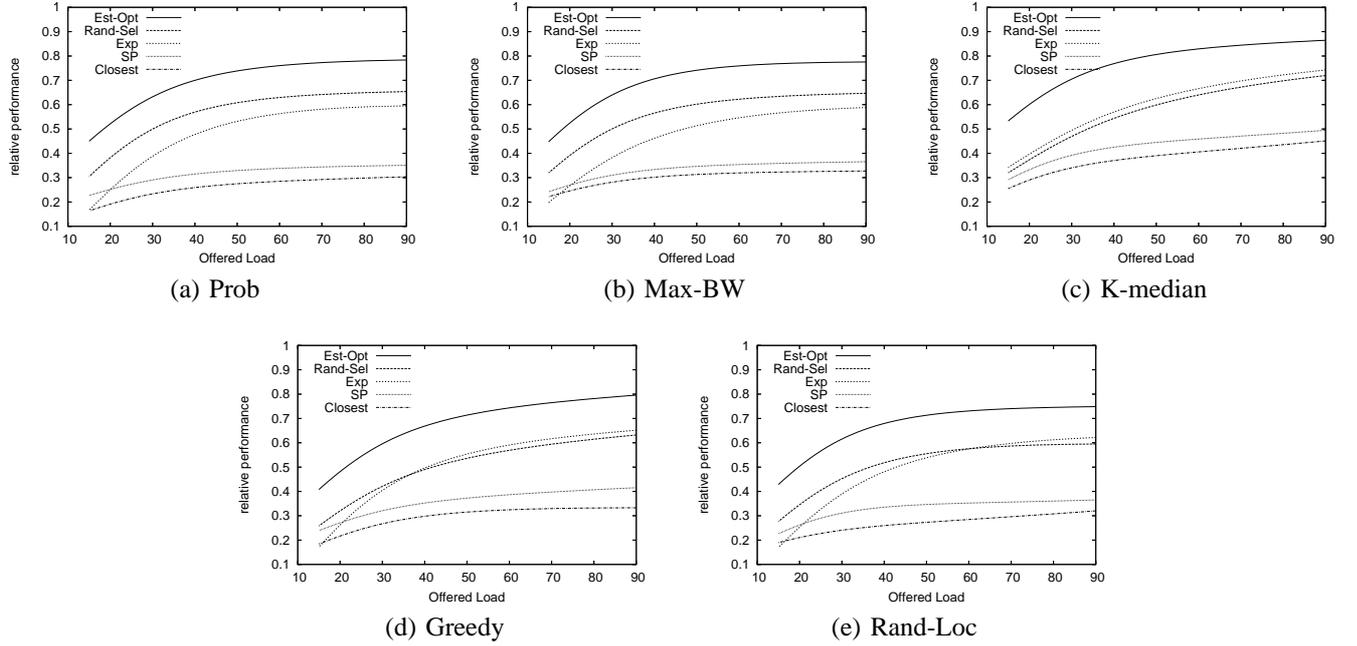


Fig. 4. The relative performance for every combination of placement and selection algorithms

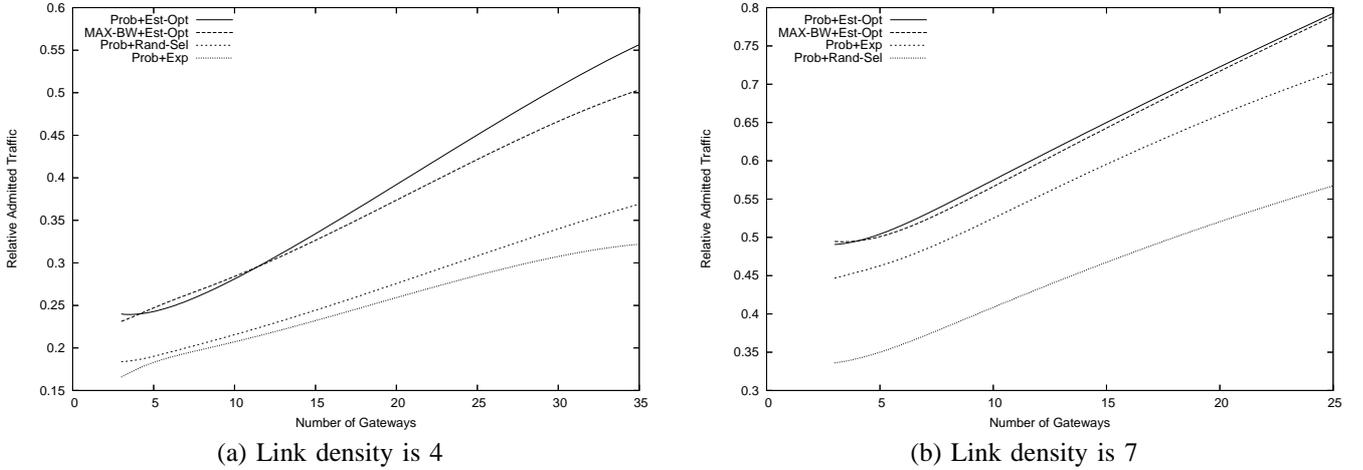


Fig. 5. The admitted traffic for selected combinations of algorithms, as a function of the number of gateways

combinations of placement and selection algorithms have similar performance to what we found before.

We now examine the case where not all the traffic in the AS demands a service. Such flows are not routed through a service gateway, but traverse the shortest-path between their source and destination. Such flows restrict our ability to balance the load in the AS. To investigate the effect of these “direct flows” on the total admitted traffic, we run simulations where the ratio of the traffic that requires a service varies from 20% to 100% of the total traffic. We run the three best combinations of algorithms as previously found: Prob+Exp, Prob+Est-Opt and Max-BW+Est-Opt. In addition, we examine the two best placement algorithms, Max-BW and Prob, with the commonly used SP selection algorithm, and the most commonly used combination of algorithms: K-median and SP. The AS in these

simulations has 30 routers, link density of 4, and 7 service gateways. Figure 8 depicts the results. The y -axis in the graphs represents the admitted traffic divided by the maximum traffic admitted during the simulation. The x -axis represents the fraction of traffic that has to be serviced.

In Figure 8 it is most important to note that even when only 20% of the traffic needs to be routed through a service gateway, the performance differs significantly for two sets of combinations of algorithms. The first set includes the three combinations of algorithms that do not use SP for routing. The relative admitted traffic for this set is over 0.7. The second set includes the combinations of algorithms that do utilize SP. For this set the relative admitted traffic is less than 0.5. This implies that judicious placement and selection of service gateways in the AS may yield a significant performance gain

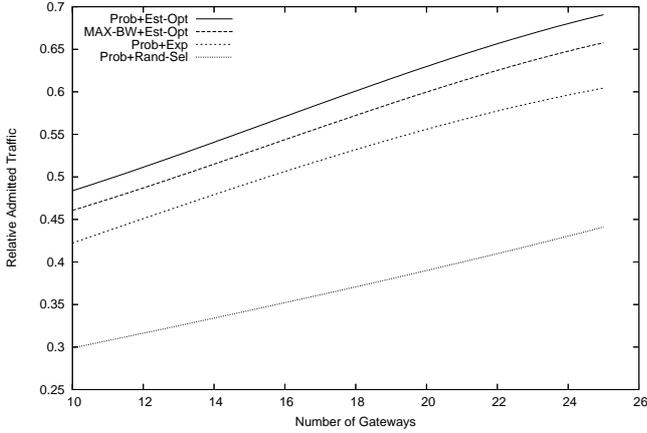


Fig. 6. The admitted traffic for the best combinations of algorithms, as a function of the number of gateways, for the Exodus ISP

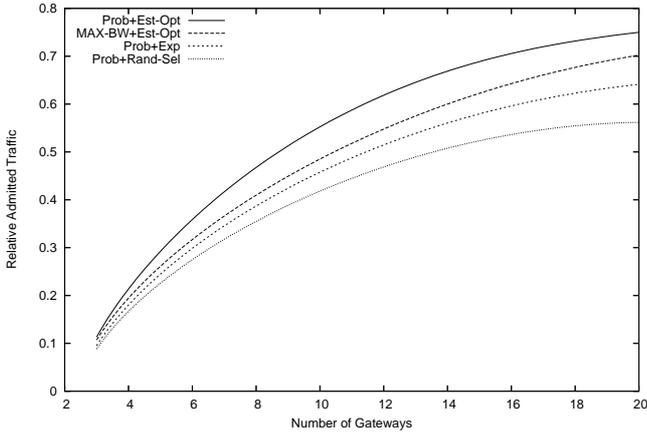


Fig. 7. The admitted traffic for the best combinations of algorithms, as a function of the number of gateways, for the gravity demand model

even if only a small portion of traffic needs to be routed through them. As the fraction of serviced flows increases, the performance difference between the two sets of algorithms increases as well. In fact, the traffic admitted by the first set of combinations increases while the traffic admitted by the second set does not. This is due to the fact that even though the gateways are placed judiciously in the AS, the SP selection algorithm does not try to load balance the traffic.

As noted above, the main drawback of Exp is that it requires every node to have information on the current load of every network link. This information can be obtained via an extension to a link state routing protocol, like OSPF-TE [40]. However, this information cannot be always obtained in real-time, mainly because update packets cannot be sent immediately after every change in the link load. We now study the performance of Exp under this practical constraint. Figure 9 depicts the performance of Exp as a function of the update interval. As in the previous simulations, short-term traffic flows are created between every two nodes in accordance with the long-term average offered load imposed by the traffic matrix.

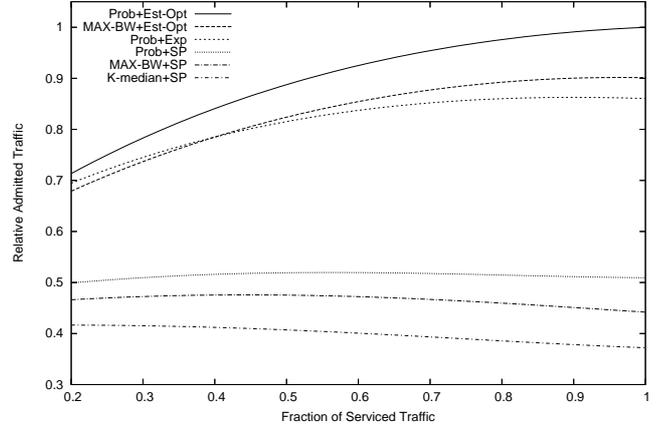


Fig. 8. The admitted traffic for some combinations of algorithms, as a function of the fraction of serviced traffic

The average long-term offered load between every two nodes is determined from the following parameters: the bandwidth demand per flow, the lifetime of the flows, and their inter-arrival time. These parameters are generated for each flow using an exponential distribution. The y -axis in the graph represents the admitted traffic normalized to the case where this update interval is 0. The x -axis represents the update interval normalized to the average inter-arrival time of flows with the same source and destination. In this simulation, Exp is used in conjunction with Prob. However, we saw similar behavior of Exp with the other placement algorithms.

First, it is evident that as the update interval increases, the admitted traffic decreases. This is due to the fact that nodes have a greater probability to choose routes that are no longer available. As the update interval increases, the nodes make such wrong decisions for longer periods of time. It can be seen that when the normalized update interval is 32, the admitted traffic drops by half. However, when the normalized update interval is less than 4, the performance loss is less than 4%. This implies that Exp can tolerate inaccurate information for short periods of time while achieving roughly the same performance level as with accurate information.

As discussed earlier, the selection algorithm Est-Opt and the placement algorithm Prob rely on a priori knowledge of the long-term average traffic distribution in the AS. The simulation results presented above are based on the assumption that these algorithms have accurate information regarding this distribution. However, in reality these algorithms will probably have only a rough estimate of this distribution. In the following we evaluate the effect of inaccurate information on the effectiveness of these algorithms. We add uniformly distributed random noise to the long-term average distribution of the traffic. Figure 10 depicts the performance of the three combinations: Prob and Est-Opt, Max-BW and Est-Opt, and Prob and Exp. We consider an AS with 30 routers whose link density is 4, and with 7 gateways. The y -axis in the graphs represents the admitted traffic divided by the admitted traffic when accurate information regarding the long-term traffic distribution is available. The x -axis represents the noise

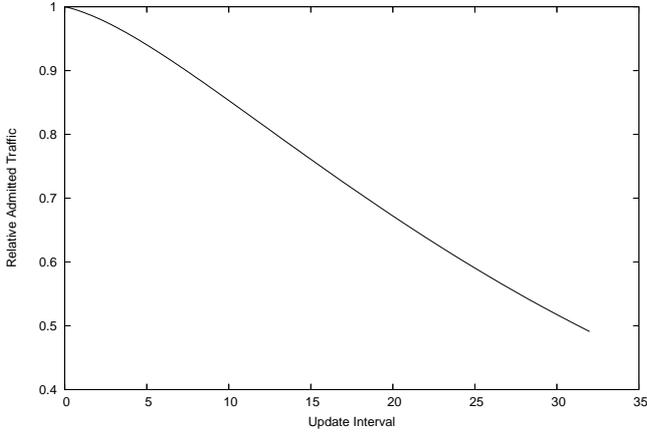


Fig. 9. The relative performance of Exp, as a function of the update interval (gateways were placed using Prob)

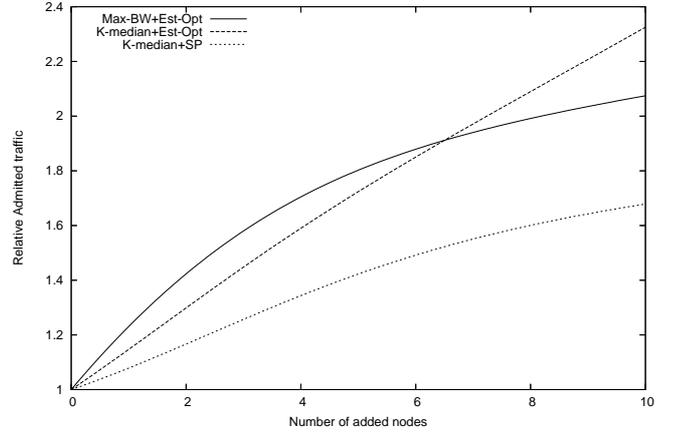


Fig. 11. The effect of AS evolution on admitted traffic

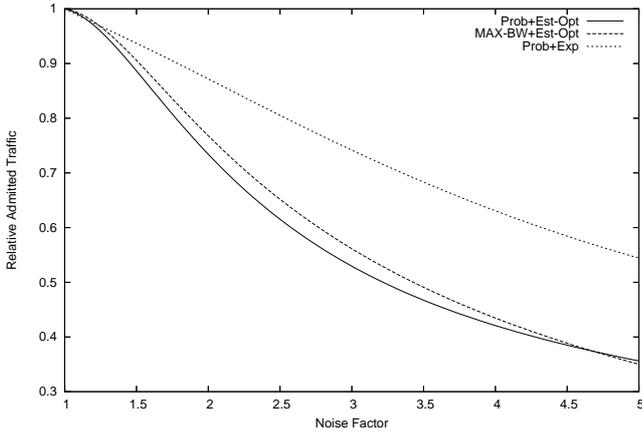


Fig. 10. The admitted traffic for selected combinations of algorithms, when random noise is added to the traffic distribution information

magnitude. A noise magnitude of X means that if the real volume of traffic between a source-destination pair is t , then the algorithms know an inaccurate estimate of that volume that ranges uniformly between t/X and tX .

As expected, Figure 10 reveals that as the random noise increases, the effectiveness of the algorithms decreases. Furthermore, the performance decrease is more drastic when Est-Opt is used as a selection algorithm. For example, the admitted traffic is reduced by almost 66% for a noise factor of 5. This suggests that the sensitivity of Est-Opt to traffic estimation errors is high. When Prob is used with Exp, the performance is less drastically affected. For example, when the noise factor is 5, the admitted traffic decreases by 40%. This is because Exp does not rely on traffic estimates.

Next, we evaluate the effect of AS evolution on the admitted traffic if the gateways are not dynamically relocated in accordance with topological changes. Figure 11 depicts the performance of three combinations of placement and selection algorithms: Max-BW and Est-Opt, K-median and Est-Opt, and K-median and SP. We consider an AS with

30 routers with average link degree of 4 and 7 gateways. The y -axis represents the admitted traffic as the AS evolves, divided by the admitted traffic for the baseline AS. The x -axis represents the number of nodes added to the AS since the last placement of the gateways. The new nodes are added to the AS topology according to the Barabasi-Albert model [33]. As the AS topology evolves, so does the traffic distribution, because the new nodes inject more traffic into the network.

From Figure 11 it follows that as the number of new nodes and the offered load in the AS increase, so does the admitted traffic. Adding 10 new nodes to the network increases the admitted traffic by 67%, 132%, and 107% when using the algorithm combinations of K-median and SP, K-median and Est-Opt, and Max-BW and Est-Opt, respectively. The combination of K-median and SP has the most moderate increase. This is because the load on the service gateways is not properly balanced, and hot-spots in the AS are not relieved. The other two combinations utilize the capacity of the gateways more efficiently, with a sharper increase in the admitted traffic. Hence, we conclude that algorithms Max-BW and Est-Opt not only outperform the other algorithms, but also utilize the network resources more efficiently when the AS evolves.

Finally, note that in the model as presented in Section IV, the bandwidth demand of the flows is rigid. However, there are cases where the bandwidth demand of a flow is flexible. For example, during the startup of a multimedia session, it can change its demand by using different codecs. To examine the effect of such flexibility on the performance of the various algorithms, we run some simulations where bandwidth demand of a flow is halved if the original demand cannot be satisfied. This process repeats up to 3 times or until the demand can be satisfied. The profit gained by admitting a flow is relative to the flow's satisfied bandwidth. Our results show that the combinations of Prob+Est-Opt, Prob+Exp, and Max-BW+Est-Opt have the best performance yielding a relative admitted traffic of 0.81, 0.77, 0.75, respectively. The K-median+SP combination has the worst performance in the non-rigid model, with a relative admitted traffic of only 0.3. These results are in

agreement with those reported earlier for the original (rigid) model. Another result for the non-rigid model is that with the Prob+Est-Opt, Prob+Exp, and Max-BW+Est-Opt combinations the total volume of admitted traffic increases by 60%-70%, whereas with the K-median+SP combination it increases by only 10%-20%.

VII. CONCLUSIONS

We proposed a novel approach for addressing the problems of placement and selection of service gateways. Rather than considering the need to route traffic to its destination through the gateways as a burden that has to be minimized, we take advantage of this need for the sake of traffic engineering. We translated the problems of gateway placement and selection to optimization traffic engineering problems whose objective is to maximize the admitted throughput. In this context, both problems are NP-complete. For the placement problem, we presented a probabilistic approximation algorithm (Prob) and an efficient heuristic (Max-BW). For the selection problem we presented an algorithm whose competitive ratio is bounded (Exp), as well as a simpler heuristic (Est-Opt).

We then conducted a detailed simulation study to examine the performance of these algorithms. We showed that when the service gateways create bandwidth bottlenecks in the network, a simple placement heuristic that maximizes the connectivity of the gateways (Max-BW) yields the best results. When the service gateways are not bandwidth bottlenecks, a placement algorithm that takes into account the expected traffic distribution (Prob) yields the best results. The selection algorithm that was shown to have the best performance is Est-Opt. This algorithm needs to know only the long-term average of the traffic distribution. Finally, the combinations of Est-Opt with Max-BW or with Prob yield significant improvement over algorithms that only try to minimize the length of the traffic routes through the gateways.

Our main conclusion is that placement and selection of network services can be employed as an effective tool for traffic engineering.

REFERENCES

- [1] S. Choi and Y. Shavitt, "Proxy location problems and their generalizations," in *International Workshop on New Advances of Web Server and Proxy Technologies*, Providence, RI, USA, May 2003.
- [2] P. B. Danzig, R. S. Hall, and M. F. Schwartz, "A case for caching file objects inside internetworks," in *Proceedings of the ACM SIGCOMM*, San Francisco, CA, September 1993, pp. 239–248.
- [3] H. Jamjoom, S. Jamin, and K. Shin, "Self organizing network services," in *University Michigan CSE-TR-407-99*, 1999.
- [4] A. Heddaya and A. Mirdad, "Webwave: Globally load balanced fully distributed caching of hot published documents," in *Proceedings of IEEE ICDCS*, San Francisco, CA, May 1997, pp. 160–168.
- [5] S. Michel, "Adaptive web caching: Towards a new caching architecture," in *Proceedings of 3rd International Caching Workshop*, June 1998, pp. 2041–2046.
- [6] P. Krishnan, D. Raz, and Y. Shavitt, "The cache location problem," in *IEEE/ACM Transactions on Networking*, vol. 8(5), October 2000, pp. 568–582.
- [7] R. Prasad and H. Wu, "Minimum-cost gateway deployment in cellular wi-fi networks," in *IEEE Consumer Communications and Networking Conference*, Las Vegas, 2006.
- [8] D. Awduche and et. al., "Overview and principles of Internet traffic engineering," IETF RFC 3272, May 2002.
- [9] A. Medina, N. Taft, K. Salamatian, S. Bhattacharyya, and C. Diot, "Traffic matrix estimation: Existing techniques and new directions," in *Proceedings of the ACM SIGCOMM*, 2002.
- [10] Y. Vardi, "Network tomography: Estimating source-destination traffic intensities from link data," *J. of the American Statistical Association*, pp. 365–377, 1996.
- [11] B. Li and et. al., "On the optimal placement of Web proxies in the Internet," in *Proceedings of IEEE INFOCOM*, March 1999, pp. 1282–1290.
- [12] X. Jia and et. al., "Optimal placement of proxies of replicated web servers in the Internet," in *International Conference on Web Information Systems Engineering*, 2000.
- [13] S. Jamin and et. al., "Constrained mirror placement on the Internet," in *Proceedings of IEEE INFOCOM*, April 2001, pp. 31–40.
- [14] L. Qiu, V. Padmanabhan, and G. Voelker, "On the placement of web server replicas," in *Proceedings of IEEE INFOCOM*, vol. 3, 2001, pp. 1587–1596.
- [15] P. Radoslavov, R. Govindan, and D. Estrin, "Topology-informed Internet replica placement," in *Proceedings of WCW*, June 2001.
- [16] Y. Ahmad and U. Cetintemel, "Network-aware query processing for stream-based applications," in *Proceeding of VLDB*, 2004.
- [17] U. Srivastava, K. Munagala, and J. Widom, "Operator placement for in-network stream query processing," in *Proceeding of PODS*, 2005.
- [18] T. Nakano and T. Suda, "Self-organizing network services with evolutionary adaptation," in *IEEE Transactions on Neural Networks*, vol. 16(5), September 2005.
- [19] K. Papagiannaki, N. Taft, Z. Zhang, and C. Diot, "Long-term forecasting of Internet backbone traffic: Observations and initial models," in *Proceedings of the IEEE INFOCOM*, March 2003.
- [20] Z. Jianping, L. Keqin, and W. Zhimei, "Selection algorithms for anycast relay routing," in *IEEE International Conference on Performance, Computing, and Communications*, 2004, pp. 21 – 27.
- [21] Z. Fei, S. Bhattacharjee, E. W. Zegura, and M. H. Ammar, "A novel server selection technique for improving the response time of a replicated service," in *Proceedings of IEEE INFOCOM*, April 1998.
- [22] E. W. Zegura, M. H. Ammar, Z. Fei, and S. Bhattacharjee, "Application-layer anycasting: A server selection architecture and use in a replicated web service," in *IEEE/ACM Trans. on Networking*, vol. 8, August 2000, pp. 455–466.
- [23] Y. Hou and et. al., "On energy provisioning and relay node placement for wireless sensor networks," in *IEEE Transactions on Wireless Communications*, vol. 4, September 2005, pp. 2579–2590.
- [24] X. Kenan, H. Hassanein, and G. Takahara, "Relay node deployment strategies in heterogeneous wireless sensor networks: multiple-hop communication case," in *Proceedings of Sensor and Ad Hoc Communications and Networks*, September 2005, pp. 575–585.
- [25] A. Kashyap, S. Fangting, and M. Shayman, "Relay placement for minimizing congestion in wireless backbone networks," in *Proceedings of Wireless Communications and Networking Conference*, vol. 1, April 2006, pp. 159–164.
- [26] R. Cohen and G. Nakibly, "On the computational complexity and effectiveness of N-hub shortest path routing," in *Proceedings of the IEEE INFOCOM*, Honk Kong, March 2004.
- [27] R. M. Karp, "Online algorithms versus offline algorithms: How much is it worth to know the future?" International Computer Science Institute, Technical Report TR-92-044, 1992.
- [28] J. Lin and J. S. Vitter, "e-approximations with minimum packing constraint violation," in *Proceedings of the ACM Symposium on Theory of Computing*, 1992, pp. 771–782.
- [29] Y. Azar and O. Regev, "Strongly polynomial algorithms for the unsplittable flow problem," in *Proceedings of the 8th Conference on Integer Programming and Combinatorial Optimization (IPCO)*, 2001, pp. 15–29.
- [30] B. Awerbuch, Y. Azar, and S. Plotkin, "Throughput competitive online routing," in *Proc. 34th IEEE Symp. on Found. of Comp. Science*, 1993, pp. 32–40.
- [31] K. Kar, M. Kodialam, and T. Lakshman, "Minimum interference routing of bandwidth guaranteed tunnels with MPLS traffic engineering applications," in *Proceedings of IEEE INFOCOM*, 2000.
- [32] S. Suri, M. Waldvogel, and P. Warkhede, "Profile based routing: A new framework for MPLS traffic engineering," in *Quality of Future Internet Services, Lecture Notes in Computer Science*, vol. 2156, September 2001.
- [33] A. Barabasi, R. Albert, and H. Jeong, "Scale-free characteristics of random networks: the topology of the world-wide web," in *Physica A: Statistical Mechanics and Its Applications*, vol. 281, June 2006, pp. 69–77.

- [34] A. Medina, A. Lakhina, I. Matta, and J. Byers, "BRITE: An approach to universal topology generation," in *Proceedings of MASCOTS*, 2001.
- [35] N. Spring, R. Mahajan, and D. Wetherall, "Measuring ISP topologies with rocketfuel," in *Proceedings of the ACM SIGCOMM*, August 2002.
- [36] R. Mahajan, N. Spring, D. Wetherall, and T. Anderson, "Inferring link weights using end-to-end measurements," in *ACM SIGCOMM Internet Measurement Workshop (IMW)*, November 2002.
- [37] M. Berkelaar, "Lp_solve software," ftp.es.tue.nl/pub/lp_solve.
- [38] J. Kowalski and B. Warfield, "Modeling traffic demand between nodes in a telecommunications network," in *Proceedings of ATNAC*, 1995.
- [39] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg, "Fast accurate computation of large-scale IP traffic matrices from link loads," in *Proceedings of ACM SIGMETRICS*, June 2003.
- [40] D. Katz and et. al., "Traffic engineering (TE) extensions to OSPF version 2," IETF RFC 3630, September 2003.
- [41] A. Srinivasan, *Approximation Algorithms via Randomized Rounding: a Survey. Lectures on Approximation and Randomized Algorithms*. PWN, Warszawa: Polish Scientific Publishers, 1999.

APPENDIX A

A PROOF FOR THE APPROXIMATION RATIO OF ALGORITHM V-A.1 FOR THE PLACEMENT PROBLEM

We now prove that Algorithm V-A.1 guarantees that the number of service gateways for every $s \in S$ does not exceed k_s , and that the load imposed on every link does not exceed 1, and that the admitted bandwidth is not smaller than $\Omega(\max\{(y^*)^2/|E|, y^*/\sqrt{|E|}\})$, where y^* is the bandwidth admitted by an optimal solution for the relaxed problem. The proof draws on the proof presented in [41] for the Unsplittable Flow Problem. The main difference is that here we also need to prove that the number of service gateways for every $s \in S$ in the solution does not exceed k_s . In order to focus on our contribution we avoid reiterating some technical details in the proof. These details are fully presented in [41]. However, all information which is crucial to the understanding of the proof is included here.

Let x_{if}^* , X_f^* and h_i^{s*} be the values assigned to the corresponding variables by an optimal solution to the relaxed problem. Let L_e be a random variable representing the load on link $e \in E$ after the randomized rounding procedure. Let \mathcal{H}_s , where $s \in S$, be the sets of nodes used as hubs in the final solution.

As in [41], we need to find well-behaved estimators for each of the following two "bad" events: edge e is overloaded, namely $L_e > 1$, and the objective function is not smaller than $\Omega(\max\{(y^*)^2/|E|, y^*/\sqrt{|E|}\})$. Let these two bad events be denoted as \mathcal{E}_e and \mathcal{E}_y , respectively. Due to the nature of our problem, we also need to find a well-behaved estimator for additional $|S|$ "bad" events, namely, that more than k_s hubs for service s are chosen ($|\mathcal{H}_s| > k_s$). Let these events be denoted by \mathcal{E}_h^s .

The first two estimators are identical to those presented in [41]. We include them here for the sake of completeness. If all the demands are less than 1/2, we have:

$$E[L_e] = \sum_{i,f} x_{if}^* \cdot z_{if}^e \cdot t'(f) \leq 2/\gamma,$$

and

$$Pr(\mathcal{E}_e) = Pr\left(\sum_{i,f} x_{if} \cdot z_{if}^e \cdot t'(f) \geq 2\right),$$

where $t'(f) = 2t(f)$. Hence, a possible estimator for \mathcal{E}_e is:

$$\chi_1^e = \frac{\prod_{i,f} (1 + \delta)^{x_{if} \cdot z_{if}^e \cdot t'(f)}}{(1 + \delta)^{\mu(1+\delta)}}$$

where $\mu = 2/\gamma$ and $\delta = \gamma - 1$. We know that $E[\chi_1^e] \leq G(\mu, \delta)$, where

$$G(\mu, \delta) \doteq \left(\frac{e^\delta}{(1 + \delta)^{(1+\delta)}}\right)^\mu.$$

If all the demands are greater than 1/2, we have:

$$Pr(\mathcal{E}_e) = Pr\left(\sum_{i,f} x_{if} \cdot z_{if}^e \geq 2\right).$$

Hence, a possible estimator for \mathcal{E}_e is:

$$\chi_2^e = \Psi_2\left(\frac{x_{if} z_{if}^e}{\gamma} : \forall i, f\right)$$

where

$$\Psi_2(z) = \sum_{1 \leq i_1 < i_2 \leq n} z_{i_1} z_{i_2}, \text{ for } (z_1, z_2, \dots, z_n) \in \mathcal{R}^n$$

We know that $E[\chi_2^e] \leq 2/\gamma^2$.

Let us now consider the routing of only $U \subseteq F$. We construct a proper estimator for the bad event where the bandwidth of the admitted traffic is less than $y_U^*(1-1/e)/(2\gamma)$, where $y_U^* = \sum_{f \in U} t(f) \cdot X_f^*$. By [41], a possible estimator for this bad event is:

$$\chi_U^y = \frac{\prod_{i=1}^n (1 - \delta_1)^{t(f)}}{(1 - \delta_1)^{\mu_1(1-\delta_1)}},$$

where $\mu_1 = y_U^*(1 - 1/e)/\gamma$ and $\delta_1 = 1/2$. We know that $E[\chi_U^y] \leq H(\mu, \delta)$, where $H(\mu, \delta) \doteq e^{-\mu\delta^2/2}$.

Next, we find a well-behaved estimators χ_s^h for the last bad events. For every $i \in V$, let H_i^s be a random variable whose value is 1 if i is chosen as a hub and 0 otherwise. Note that these random variables are independent of each other, since each route for each flow is chosen independently of the others.

We now find an upper bound on the expected value of each of these random variables:

$$\begin{aligned} E[H_i^s] &= Pr(H_i^s = 1) = 1 - \prod_{f \in \mathcal{F}_s} (1 - x_{if}^*/\gamma) \\ &\leq 1 - \prod_{f \in \mathcal{F}_s} (1 - h_i^{s*}/\gamma) = 1 - (1 - h_i^{s*}/\gamma)^{|\mathcal{F}_s|} \\ &\leq |\mathcal{F}_s| h_i^{s*}/\gamma. \end{aligned}$$

The last inequality follows from the fact that for all $x \in [0, 1]$, $1 - (1-x)^n \leq nx$. We can now calculate the expected number of hubs \mathcal{H}_s :

$$\begin{aligned} E[\mathcal{H}_s] &= E\left[\sum_i H_i^s\right] = \sum_i E[H_i^s] \leq \sum_i |\mathcal{F}_s| h_i^{s*}/\gamma \\ &\leq |\mathcal{F}_s| k_s/\gamma. \end{aligned}$$

Using similar considerations to those presented in [41] for constructing χ_1^e , we conclude that

$$\chi_s^h = \frac{\prod_i (1 + \delta_2)^{H_i^s}}{(1 + \delta_2)^{\mu_2(1+\delta_2)}}, \text{ and } E[\chi_s^h] \leq G(\mu_2, \delta_2),$$

where $\mu_2 = \frac{k_s |\mathcal{F}_s|}{\gamma} \geq E[\mathcal{H}_s]$, $\delta_2 = \frac{\gamma}{|\mathcal{F}_s|} - 1$.

We define χ_{sU}^h as a well-behaved estimator for \mathcal{E}_h^s while only the subset $U \subseteq \mathcal{F}$ is considered for routing. It is obvious that $E[\chi_{sU}^h] \leq E[\chi_{sU}^h]$.

Before proceeding the approximation ratio of the algorithm, we present the following theorem from [41].

Theorem 1: Let E_1, E_2, \dots, E_t be events and r, s be non-negative integers with $r + s \leq t$ such that:

- E_1, E_2, \dots, E_r are all increasing, with respective well-behaved estimators g_1, g_2, \dots, g_r ,
- E_{r+1}, \dots, E_{r+s} are all decreasing, with respective well-behaved estimators g_{r+1}, \dots, g_{r+s} ,
- E_{r+s+1}, \dots, E_t are arbitrary events, with respective proper estimators $g_{r+s+1}, g_2, \dots, g_t$,
- all E_i and g_i are completely determined by \vec{X} .

Then, if

$$1 - \left(\prod_{i=1}^r (1 - E[g_i]) \right) + 1 - \left(\prod_{i=r+1}^{r+s} (1 - E[g_i]) \right) + \sum_{i=r+s+1}^t E[g_i] < 1$$

holds, we can efficiently construct a deterministic assignment for \vec{X} under which none of E_1, E_2, \dots, E_t holds. (Empty products are taken to be 1. If there is g_i such that $E[g_i] > 1$, then the entire product is equal to 0.) ■

We shall use the above theorem and the well-behaved estimator for the “bad” events mentioned above to prove the following.

Theorem 2: A deterministic placement of k service gateways that facilitates an admitted traffic of no less than $\Omega(\max\{(y^*)^2/|E|, y^*/\sqrt{|E|}\})$ can be found efficiently. ■

Proof

We start by showing the $\Omega((y^*)^2/|E|)$ bound. Let F_0 be the subset of flows for which traffic demand is at most $1/2$ and $F_1 = F \setminus F_0$. Let $y_{F_0}^*$ be the optimal objective function when only the subset $U \subseteq F$ is considered for routing. We first assume that $y_{F_1}^* \leq y_{F_0}^*$, i.e., $y_{F_0}^* \geq y^*/2$. Since the events \mathcal{E}_h and $\mathcal{E}_e \forall e \in E$ are increasing and the event \mathcal{E}_y is decreasing, in order to avoid these events we must have,

$$1 - \left(\left(1 - \prod_s E'[\chi_{sF_0}^h] \prod_e (1 - E'[\chi_2^e]) \right) + E[\chi_{F_0}^y] \right) \leq 1,$$

where $E'(\cdot) \doteq \min\{E(\cdot), 1\}$.

Since

$$E[\chi_{F_0}^y] \leq H(y_{F_0}^*(1-1/e)/\gamma, 1/2) \leq H(y^*(1-1/e)/(2\gamma), 1/2),$$

for a suitably large constant c it can be shown that $\gamma = c|E|/y^*$ satisfies the above inequality. Thus, from Theorem 1 follows that if $y_{F_1}^* \leq y_{F_0}^*$, we can efficiently select feasible paths for F_0 with objective function value $\Omega((y_{F_0}^*)^2/|E|) = \Omega((y^*)^2/|E|)$.

Next, we consider the case where $y_{F_1}^* \geq y_{F_0}^*$. As in the previous case, in order to avoid all of the “bad” events we must have

$$1 - \left(\left(1 - \prod_s E'[\chi_{sF_1}^h] \prod_e (1 - E'[\chi_2^e]) \right) + E[\chi_{F_1}^y] \right) \leq 1.$$

Again, for a suitably large constant c , and from the same considerations as above, it can be shown that $\gamma = c|E|/y^*$ satisfies

the above inequality. Thus, from Theorem 1 follows that if $y_{F_1}^* \geq y_{F_0}^*$, we can efficiently select feasible paths for F_1 with objective function value $\Omega((y_{F_1}^*)^2/|E|) = \Omega((y^*)^2/|E|)$.

To conclude the proof we need to show the $\Omega(y^*/\sqrt{|E|})$ bound. If $y^* \geq \sqrt{|E|}$, this immediately follows from the $\Omega((y^*)^2/|E|)$ bound. If $y^* < \sqrt{|E|}$, we simply choose to admit a flow f for which $t(f) = 1$ holds. □

PLACE
PHOTO
HERE

Reuven Cohen (M'93, SM'99) received the B.Sc., M.Sc. and Ph.D. degrees in Computer Science from the Technion - Israel Institute of Technology, completing his Ph.D. studies in 1991. From 1991 to 1993, he was with the IBM T.J. Watson Research Center, working on protocols for high speed networks. Since 1993, he has been a professor in the Department of Computer Science at the Technion. He has also been a consultant for numerous companies, mainly in the context of protocols and architectures for broadband access networks. Dr.

Cohen has served as an editor of the IEEE/ACM Transactions on Networking, and the ACM/Kluwer Journal on Wireless Networks (WINET). Dr. Cohen is a senior member of the IEEE and heads the Israeli chapter of the IEEE Communications Society.

PLACE
PHOTO
HERE

Gabi Nakibly (S'04) received the B. Sc. in Information Systems engineering (summa cum laude) and M. Sc. in Computer Science from the Technion - Israel Institute of Technology, Haifa, Israel, in 1999 and 2004, respectively. Since 2005, he has been a Ph.D. student in Computer Science Department in the Technion, working on QoS routing and traffic engineering.