

A Collaborative P2P Scheme for NAT Traversal Server Discovery based on Topological Information

Rubén Cuevas^{a,*} Ángel Cuevas^a Albert Cabellos-Aparicio^b
Loránd Jakab^b Carmen Guerrero^a

^a*Universidad Carlos III de Madrid. Departamento de Ingeniería Telemática*

^b*Universitat Politècnica de Catalunya, D. d'Arquitectura de Computadors*

Abstract

In the current Internet picture more than 70% of the hosts are located behind Network Address Translators (NATs). This is not a problem in the client/server paradigm. However, the Internet has evolved and nowadays the major portion of the traffic is due to peer-to-peer (p2p) applications, and in such scenario, two hosts behind NATs (NATed hosts) cannot establish direct communications. The easiest way to solve this problem is by using a third entity, called Relay, that forwards the traffic between the NATed hosts. Although many efforts have been devoted to avoid the use of Relays, they are still needed in many situations. Hence, the selection of a *suitable* Relay becomes critical to many p2p applications. In this paper we propose the *Gradual Proximity Algorithm (GPA)*: a simple algorithm that guarantees the selection of the topologically closest Relay. We present a measurement-based analysis showing that the GPA minimizes both the delay of the relayed communication and the transit traffic generated by the Relay, being a QoS-aware and ISP-friendly solution. Furthermore, the paper presents the Peer-to-Peer NAT Traversal Architecture (P2P-NTA), which is a global, distributed and collaborative solution, based on the GPA. This architecture addresses the Relay discovery/selection problem. We have performed large-scale simulations based on real measurements, which validate our proposal. The results demonstrate that the P2P-NTA performs similarly to direct communications with reasonable deployments of p2p applications. In fact, only 5% of the communications experience an extra delay that may degrade the QoS due to the fact of using Relays. Furthermore, the amount of extra transit traffic generated is only 6%. We also show that the P2P-NTA largely outperforms other proposals where the QoS degradation affects up to more than 50% of the communications, and the extra traffic generated goes further than 85%.

Key words: p2p, Relay, NAT, AS, simulation, measurements

1 Introduction

The rapid reduction of the IPv4 address space [1] has stimulated the widespread deployment of Network Address Translators (NATs) [2] on the Internet. Furthermore, companies' network security policies include the utilization of NATs and/or firewalls in order to hide the network topology and control both inbound and outbound traffic. As a result, recent studies state that more than 73% of Internet end-hosts are located behind NATs or firewalls [3]. In the following we will use the term *NATed* to designate entities behind NAT.

NATs were designed for client/server applications. However, in the last decade peer-to-peer (p2p) applications such as VoIP (e.g., Skype), online games, P2P file sharing (e.g., BitTorrent), P2P streaming (e.g., PPLive) have become tremendously popular and nowadays they are responsible for the largest share of Internet traffic [4]. Unfortunately, p2p communications cannot be directly established through NATs. This is because NATs do not allow inbound connections unless they are manually configured to do so. Researchers have designed a set of techniques to provide NAT Traversal capabilities to the NATed hosts [5–10]. However, there are a large number of cases (e.g. symmetric NAT) [11] where these techniques do not work. In these cases the communication must be established using a third non-NATed entity that we call Relay (also known as NAT Traversal Server). In such a scenario, both end-hosts communicate through this Relay ($end-host1 \leftrightarrow Relay \leftrightarrow end-host2$), that forwards the traffic between them.

As a consequence, the selection of an appropriate Relay becomes a key issue that has a direct impact on the communication delay and, in addition, it may avoid extra-costs for the ISP that hosts the Relay. In particular: (i) a bad choice of the Relay may increase the delay, leading to an undesirable QoS experienced in delay-sensitive communications such as multimedia (VoIP, online gaming, etc); (ii) if the Relay is not carefully selected, it increases the total transit traffic of the ISP that hosts it. It must be noted that some p2p applications with a large number of NATed users, such as BitTorrent or eMule, generate a large amount of traffic that may produce an increase of the ISP's costs. Therefore, it is critical to define a Relay selection algorithm that eliminates, or at least alleviates, these negative effects.

This paper presents the *Gradual Proximity Algorithm* (GPA). GPA is a Relay selection algorithm that chooses the closest topological Relay among the

* Corresponding author.

Email addresses: rcuevas@it.uc3m.es (Rubén Cuevas), acrumin@it.uc3m.es (Ángel Cuevas), acabello@ac.upc.edu (Albert Cabellos-Aparicio), ljakab@ac.upc.edu (Loránd Jakab), carmen.guerrero@uc3m.es (Carmen Guerrero).

available ones: first, it tries to find a Relay in the same Autonomous System (AS) as the NATed client. In case there is no Relay in that AS, a Relay within the same NATed node's country is selected; if this fails too, the GPA selects a Relay in the NATed node's continent (to avoid transcontinental links); finally if all the previous attempts fail, a random Relay is chosen. We rely on real measurements to demonstrate that the GPA minimizes the delay of the relayed communication. Also, this algorithm minimizes the extra transit traffic of the Relays' ISPs.

Moreover, this paper presents the Peer-to-Peer Nat Traversal Architecture (P2P-NTA). This is a wide area collaborative solution that addresses the problem of Relay discovery and selection by using the GPA. Basically, the P2P-NTA is a distributed solution where Relays form a Distributed Hash Table (DHT) to store their location information: AS, country and continent. The NATed nodes are connected to a Relay that belongs to the DHT. In the case when the node is connected to a distant Relay (e.g. on a different continent), the former asks the latter for a closer Relay. The Relay then runs the GPA: (i) it queries the DHT to find a Relay in the same AS as the client; (ii) if there is none, it queries for a Relay in the same country; (iii) if no Relay is found in the same country, it asks for a Relay on the same continent. After that, the Relays found are sent to the NATed host that connects to the closest one with enough available bandwidth.

Therefore, if we assume that the DHT is well populated and there is at least one Relay per AS, the relayed communications would experience a latency similar to the direct counterpart. This is because we are just adding an additional intra-AS delay. Moreover, all relayed communications follow the same AS-path than the direct one, producing no extra transit costs to the ISPs.

In order to validate the P2P-NTA architecture we have developed the P2P-NTA simulator that uses the real Internet AS-topology and real end-to-end latencies. This data has been obtained from the iPlane project [12,13]. In particular, this project provides AS connectivity, IP prefixes announced in the default-free zone and delay measurements between pairs of Points of Presence (PoPs) in the Internet. This dataset has allowed us to run very large scale simulations involving thousands of real ISPs and several thousands of end-hosts. The results assess the validity of our proposal: its performance is similar to that of direct communication when considering a reasonably sized deployment. Less than 5% of communications suffer from an extra delay that may affect the QoS. Moreover, the P2P-NTA generates an almost negligible amount of transit traffic (6%), thus confirming that it is an ISP friendly solution. On the other hand, the P2P-NTA clearly outperforms other Relay selection algorithms such as Random Relay Selection [14] or Pre-Established Relay [15]. In these proposals, more than 50% of the communications suffer from an extra delay that may affect the QoS, whereas even in the best case over 85% of extra

transit traffic is generated.

The proposed solution is based on the collaboration of nodes participating in the p2p application (e.g. PPLive, BitTorrent) where many of these end-hosts may act as Relays. Typically, any member of the p2p network with a public IP address is a potential Relay candidate. This leads to a big amount of potential Relays, thus becoming a Relay implies low cost.

In short, the main contributions of this work are:

- *The Gradual Proximity Algorithm*: This is a lightweight and simple algorithm that allows finding the topologically closest Relay from the available ones. This allows to minimize the relayed communication delay and avoids to generate extra transit traffic at the Relay's ISP.
- *The Peer-to-Peer NAT Traversal Architecture (P2P-NTA)*: This is a globally distributed and collaborative architecture that solves the problem of Relay discovery and selection. For this purpose it uses a DHT to register and retrieve the location information of the Relays, and implements the GPA's selection procedure. This lightweight architecture inherits the advantages of the GPA.
- *The P2P-NTA simulator*: This is an Internet-scale simulator that allows us to evaluate the architecture considering thousands of real ISPs and tens of thousands of final users. Since it uses real AS-maps and real end-to-end delays the results are relevant. It must be noted that this simulator can be adapted to evaluate other large-scale solutions in the Internet, so we release it to the scientific community¹.

2 Smart Relay Selection: the Gradual Proximity Algorithm

In this section we first discuss why Relays are a must in the current Internet. Next, we clarify why selecting a topologically closer Relay is efficient and we describe the Gradual Proximity Algorithm. Finally, we present a measurement-based analysis that validates the proposed algorithm.

- *Why we need Relays:*

The main reason is that currently, around the 73% of the users in the Internet are located behind NAT [3]. It has been shown that even using very sophisticated techniques [7–10], there are some types of NATs (e.g Symmetric NAT), widely deployed [11], that can be hardly traversed. In addition, not all the applications implement these NAT traversal techniques. This leads to the conclusion that Relays are a necessity of today's Internet.

¹ The simulator is available at <http://personals.ac.upc.edu/p2pntat>

- *Why selecting a topologically close Relay is efficient:*

When two nodes (A and B) that are connected behind a NAT want to communicate through a relay they establish two different connections: $A \leftrightarrow \text{Relay} \leftrightarrow B$. Some previous overlay routing proposals suggested to select the Relay randomly [14], or from a pre-established pool [15]. These selection algorithms may obtain an unsuitable Relay that increases the communication delay and the transit traffic of the ISP that hosts the Relay. Indeed, it is possible that two end-users located in the same ISP choose a Relay from a different one, or even from a different continent.

We claim that selecting a topologically close Relay reduces both the delay of the communications and the transit traffic of the ISP that hosts the Relay. The Internet is structured into Autonomous Systems (AS), hence the closest Relay, in terms of hops, is usually located in the same AS as the node itself. Unfortunately, not all of the ASes may host a Relay. For this case we have defined the *Gradual Proximity selection Algorithm (GPA)*, that defines different degrees of proximity in the current Internet scheme (Alg. 1).

Algorithm 1 Gradual Proximity Algorithm

```
/* Input Parameters*/
userAS, userCountry, userContinent
/* Output Parameters*/
Relay
/* Algorithm*/
if  $\exists$  Relay on the same AS then
    Choose the closest one with enough bandwidth among them
else if  $\exists$  Relay on the same country then
    Choose the closest one with enough bandwidth among them
else if  $\exists$  Relay on the same continent then
    Choose the closest one with enough bandwidth among them
else
    Choose a random Relay with enough bandwidth
end if
return Relay
```

Next, we explain the consequences that the GPA has in terms of delay and transit traffic cost:

- *Delay:* The idea behind our algorithm is to use the shortest possible AS-Path between the two end-hosts, the main rationale being the following: (i) If we use a Relay in the same AS, the AS-Path will be the same as for direct communication. Then, GPA adds just some extra hops inside the AS at IP level; (ii) If we choose a Relay in the same country we are probably adding one AS-hop toward the Relay; (iii) If we select a Relay on the same continent we add some hops to the AS-Path, but we avoid transcontinental links (e.g. transoceanic) that have a very high propagation delay.

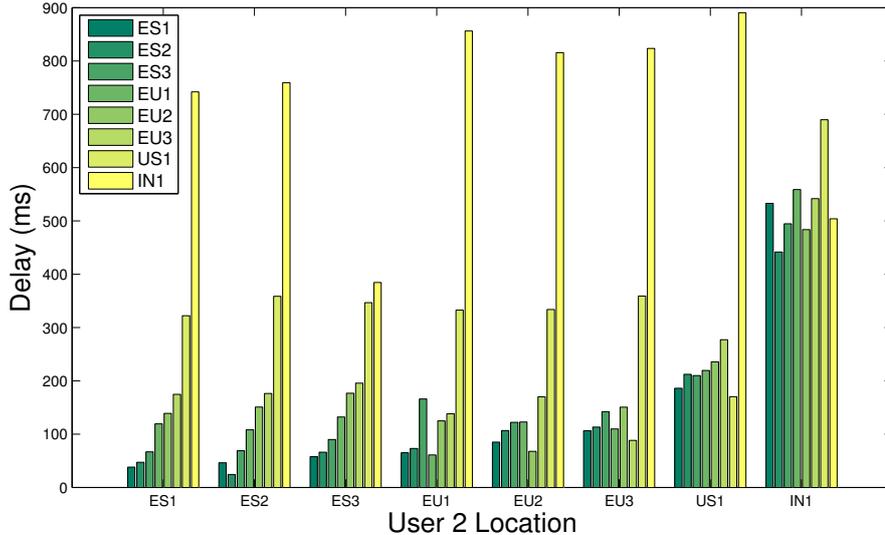


Fig. 1. Relayed Communications RTT ($u1$ located in ES1)

To validate the performance of the proposed algorithm in terms of delay, we have performed a set of live experiments. We have deployed measurement boxes in different ASes: 3 in Spain: ES1, ES2, ES3; 3 in three different European countries: EU1 (Italy), EU2 (Norway), EU3 (Greece); 1 of them located in US, US1; and the last one located in India, IN1. We use the notation $u1$ and $u2$ for the end-users involved in the communication and R for the Relay. The location of $u1$ is fixed to ES1 while we iterate $u2$ and R among all the possible locations. Figure 1 shows the RTT values of the relayed communications. The x-axis refers to the location of $u2$, while the bars show the resulting RTT values as a function of the location of the Relay.

As the figure shows the topological distance between the Relay and the end-users has a significant impact on the communication delay. We can also see that, for a given $u2$ location, the latency increases as assumed by the GPA: same AS < same country < same continent < different continent. The results also suggest that there is a positive relation between the topological distance and the delay.

To further validate our algorithm we have measured the end-to-end delay of a large set of end-users. The measurements come from iPlane [12,13], which is a scalable service providing accurate predictions of Internet path performance for overlay services and Internet-scale simulations. To achieve these goals, the iPlane project uses hundreds of vantage points distributed across the Internet for measurements, and they update their dataset daily.

In particular, we have measured the one-way delay for: (i) end-users located in the same AS; (ii) end-users located in the same country but different ASes; (iii) end-users located on the same continent but different

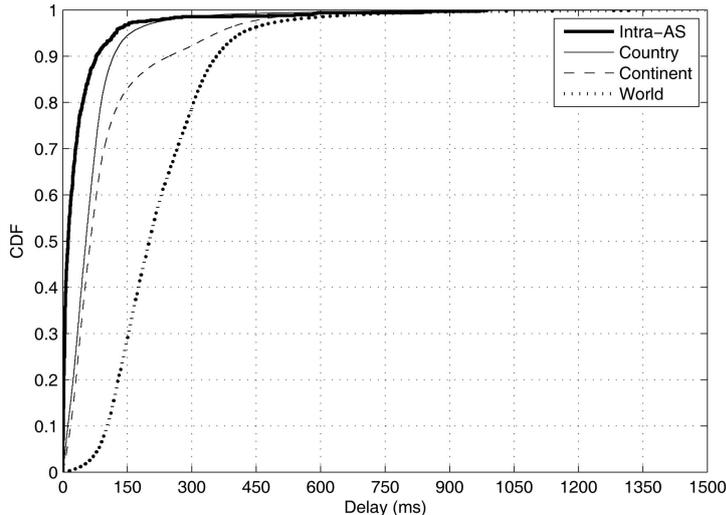


Fig. 2. One way Delay for different topological end-user locations

country; (iv) end-users located on different continents. Our dataset contains 200k end-to-end delays. Figure 2 shows the Experimental Cumulative Distribution Function (ECDF) of the delay for the different cases. By comparing these distributions we can see that the delay is strongly related to the topological distance between the end-users. If we consider the maximum one-way delay recommended by the ITU-T for voice communications (150 ms) [16], we can conclude that more than 95% of the communications are below that threshold if both end-users belong to the same AS. This percentage is still over 90% when the users are located in the same country but in different ASes. When the users are located on the same continent, the percentage drops to 83%. Finally the delay is severely impacted (40%) if the users are on different continents, and the end-to-end path includes trans-continental links.

It is also worth to note that while ITU recommends 150ms as a quality threshold for voice communications, empirical experiments run by Cisco show that there is a negligible difference in voice quality Mean Opinion Score (MOS) when a 200ms threshold is used [17]. Throughout the paper we will use a 200ms delay threshold to differentiate between acceptable and non-acceptable quality voice communications. In particular, in this dataset if we consider the Cisco’s threshold: 98%, 96%, 87% and 50% of the communications are below 200ms when users are located in the same AS, same country, same continent and different continents respectively.

Therefore, the obtained results confirm the suitability of GPA as Relay selection algorithm that effectively minimizes the end-to-end communication delay.

- *Transit Traffic*: In this subsection we evaluate the extra traffic caused by the GPA algorithm. ISPs usually pay both for inbound and outbound traffic flowing through transit links, while the cost of traffic transmitted through

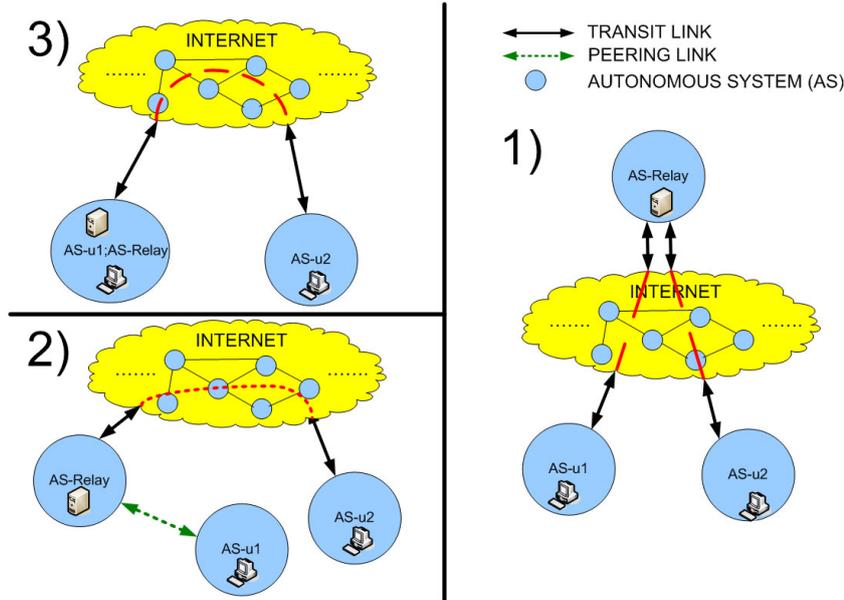


Fig. 3. AS Transit

peering links is typically free [18]. In order to better understand how the GPA avoids the transit traffic, let us focus on the case where $u1$ and $u2$ are located in different countries and ASes. In this case, *what is the extra cost that the Relay's ISP has to pay compared to the case of direct communication?*:

- (1) If we choose a Relay in a country and AS other than the two end-hosts, the Relay uses transit links to communicate with both users. Therefore, the ISP where the Relay is located has to pay for the outbound and inbound traffic of $u1$ and $u2$.
- (2) If we select a Relay in a different AS, but in the same country as $u1$, the Relay uses a peering link to communicate with $u1$ ² whereas it uses a transit link to communicate with $u2$. Thus, the cost for the ISP where the Relay is located is half than in the previous case.
- (3) If we select a Relay in the same AS as either $u1$ or $u2$, the relayed AS-path is the same than the direct one, thus the ISP does not incur in any extra cost.

Figure 3 shows the different scenarios explained above. As a result, the Gradual Proximity Algorithm always selects a Relay that minimizes the ISP transit traffic, and therefore can be considered as an ISP-friendly algorithm.

In a nutshell, we have demonstrated that our GPA leads to a reduction in the communication delays while minimizing the transit traffic costs compared to other proposals [15,14].

Finally, it must be highlighted that these benefits could have a direct im-

² Access ISPs within the same country typically establish peering agreements.

pact in currently deployed applications with millions of users. On the one hand, VoIP applications such as Skype use real-time communications that are delay-sensitive, therefore GPA would improve the quality of the communications. On the other hand, the majority of users of p2p file sharing applications such as eMule or BitTorrent are located behind NATs³, thus requiring a Relay. Moreover, these applications produce a large amount of traffic, and this increases the costs for ISPs. In this scenario, the GPA reduces significantly the relayed transit traffic and the costs for ISPs with regard to other proposals.

3 The P2P NAT Traversal Architecture (P2P-NTA)

In this section we present the *P2P Nat Traversal Architecture (P2P-NTA)*. This is a collaborative, distributed and wide area application that implements the Gradual Proximity Algorithm as described in section 2. First, we detail the proposed architecture and its functionality.

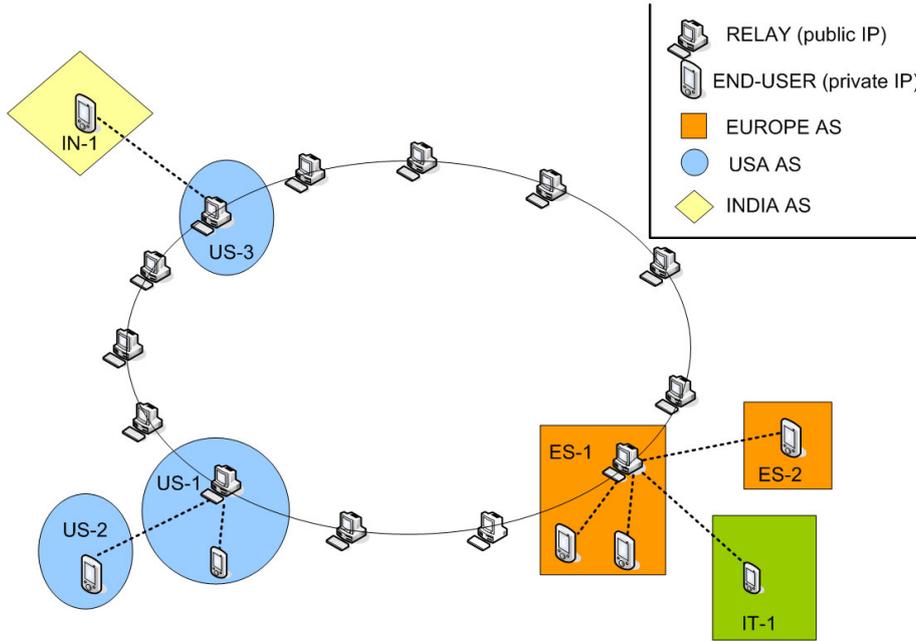


Fig. 4. P2P-NTA deployment

Physical Architecture: the *P2P-NTA* consists of two different types of nodes, *clients* and *Relays*. Nodes located behind NAT are clients, while nodes having a public IP address and enough available bandwidth typically become Relays. The latter form a Distributed Hash Table (DHT) where they regis-

³ We have conducted a large-scale crawling of BitTorrent, demonstrating that more than half of the users are located behind NATs. Due to lack of space we do not provide the details of this experiment in this paper.

ter their location information. Each client has an associated Relay for NAT-Traversal capabilities. Figure 4 depicts the physical architecture.

Bootstrapping: When turned on, the node checks if it is a Relay (it has a public IP address) or a client (it is behind a NAT). If it is a Relay it joins the DHT. For this purpose, it computes its *Peer-ID* as $hash(IP\ address)$. This *Peer-ID* indicates the position of the node in the DHT. After that, the Relay contacts any member of the DHT (previously known, well-known peers, Bootstrapping server,...) to join the *P2P-NTA*. On the other hand, if the node is a client it needs to attach to a Relay belonging to the *P2P-NTA*. If it is the first time the node joins the p2p, it will contact a bootstrapping server or a well-known DHT peer, if not, it will connect to any Relay known in the past. After that, the client checks if this Relay belongs to its own AS and if it provides a delay below a pre-established threshold. In this case, the client keeps this Relay, otherwise it finds a closer one as described below (Relay Look-up Procedure).

Registration of the Relay's Location Information: Once the Relay has joined the DHT, it computes its $ASkey = hash(ASnumber)$, $countrykey = hash(countryID)$ and $continentkey = hash(continentID)$. Then it stores the following tuples: $\langle AS\ key, Relay\ IP \rangle$, $\langle country\ key, Relay\ IP \rangle$ and $\langle continent\ key, Relay\ IP \rangle$ in the DHT. In particular these tuples are stored on the nodes with the closest IDs to the AS key (AS key's responsible node), country key (country key's responsible node) and continent key (continent key's responsible node) respectively.

Relay Look-up Procedure (based on GPA): The client can eventually detect that its current Relay is located in another ISP, or that it is too distant (e.g. the RTT towards the Relay is above a predefined threshold). This triggers the lookup procedure for a closer Relay: the client sends a message to its current Relay, including its publicly visible IP address. In turn, the Relay discovers the client's AS, country and continent and with this information, it computes an *AS key*, a *country Key* and a *continent Key*. Next, the Relay applies the GPA: (i) First, it sends a query to the DHT looking for the *AS key*. If the query succeeds, the Relay obtains a list with all the IP addresses of the Relays located in the same AS as the client. (ii) If this first query fails, the Relay sends a second query looking for the *country Key*, if this query succeeds the Relay retrieves the list of IP addresses of Relays located in the same country than the client. (iii) If the query fails again, the Relay sends a third query asking for the *continent Key*. The Relay, in turn, forwards this information to the client. Thus, the client obtains a list of, either all the Relays located in its AS (if the first query succeeded), all the Relays located in its country (if the second query succeeded) or all the Relays located on its continent (if the third query succeeded). If none of the queries succeed the client keeps its current Relay (instead of selecting a random one as explained in section 2).

It could be possible that the number of Relays located in a given AS, country or continent is too large (hundreds or even thousands). In this case the responsible node would not answer with the full list but rather a limited number (e.g. 50) selected at random.

Relay Selection: The client has to select a Relay from those included in the list. In order to select the best one in terms of available bandwidth and delay, it contacts first the closest one and solicits to join. If that given Relay has enough available bandwidth, it accepts the client. Otherwise the client is rejected and tries the second closest Relay, repeating the process until it gets accepted.

Communication Establishment: After the Relay selection algorithm each user is bind to a specific Relay. When two users ($u1$ and $u2$) want to establish a communication they use standard mechanisms, such as ICE [7], to exchange information about their respective Relays ($R1$ and $R2$). With this information the users can test both paths ($u1-R1-u2$ and $u1-R2-u2$) and choose the best Relay, in terms of delay, to establish their connection.

Geolocation Procedure: In our solution, the Relays must be capable of identifying the country, continent and AS number associated to the client's IP address. For this purpose different public services [19] and databases [20,21] can be used.

Churn and Replication: The P2P-NTA users are, in fact, end-users that may join and leave the system at any moment. This phenomenon is known as churn. In the P2P-NTA, when a Relay leaves the system gracefully, it notifies the necessary DHT nodes and removes the registered information contacting the nodes responsible of its AS, country and continent. Furthermore, it informs its neighbors and, if necessary, reassigns its stored information (AS, country and continent keys) to them. If the Relay leaves the system abruptly, it leaves the DHT with inconsistent information. To deal with such cases we have defined the following mechanisms:

- *Tuple timer:* The responsible node of an AS, country or continent tuples uses an expiration timer. The Relays must update the tuples before the timer expires, otherwise the responsible node removes the tuple. This way, if a Relay leaves the system abruptly, the tuples related to it would automatically expire after a certain amount of time.
- *Replication:* The responsible node replicates the stored information in R replicas into the DHT. These are nodes with the i th ($i \in [2, R+1]$) closest IDs to the given key. Thus, if the responsible node unexpectedly leaves the system, it is not affected at all, since the first replica takes the responsibility of its keys. In addition, the use of replicas enable load balancing mechanisms [22–25] to share the load among all of them.

Finally, when a Relay leaves the system, its clients must select a new one by triggering the *Relay Selection* algorithm as described above. The worst case happens when a Relay, which is forwarding traffic from two different users, leaves the system abruptly. Since each user is bound to a given Relay, and they have agreed on using one of them by applying the *Communication Establishment* procedure, they can switch immediately to the other one and resume their communication.

4 Related Work

Overlay Routing: It is commonly accepted by the research community that with the current AS-based routing (BGP [26]), the direct route between two end-hosts may be suboptimal in terms of delay. Hence, several solutions have been proposed that, using an overlay routing approach may reduce this delay in some cases. These solutions use one or multiple intermediate overlay relay nodes in order to shorten this AS-path. Solutions such as RON (where the Relays are selected from a static pool) [15] or SORS (where the Relays are selected at random) [14] are intended to improve general IP routing. Other solutions such as ASAP [27] are designed only for VoIP applications. In particular, ASAP is a complex architecture to find Relays that, according to their results, reduces the delay in some cases. However, it is not clear what the signaling overhead produced by the proposed system is, and it has the disadvantage of relying on bootstrapping servers and cluster representatives (called surrogate nodes) that are single points of failure. Unlike the P2P-NTA, none of these solutions consider the NAT Traversal problem. In addition, these solutions produce extra transit traffic at the ISPs where the Relays are located, imposing extra costs to these companies.

Relay/Super Node Selection: To the best of the authors' knowledge there are very few proposals that address the problem of Relay selection in case of hosts behind NATs. The P2PSIP Working Group (WG) [28] of the IETF is currently designing a p2p version of the SIP framework of protocols, where the users are able to establish communications among them without using any rendezvous server such as SIP Proxies or SIP Registrars. For this purpose, they use a DHT. One of their major challenges is how to solve the problem of NAT Traversal. In [29], the authors propose a lightweight mechanism to discover Relays within the P2PSIP DHT. Although the protocol is promising, it does not consider any kind of location information for selecting a Relay, and this incurs the costs already mentioned in this paper. We believe that the P2P-NTA is a good candidate solution to be considered by the P2PSIP WG to solve the problem of NAT-Traversal server discovery⁴.

⁴ Due to lack of space we cannot explain the details of how to specifically implement

Authors in [30] propose VIP, a p2p communication platform for NAT Traversal. In their solution, the nodes use ICE [7] and Hole Punching [8–10] in order to traverse the NATs. Basically, the nodes learn their available IP addresses/ports and register them into a DHT, so their *buddies* can easily access this information. Some of these VIP nodes act as Relays for those which are behind NATs. Unfortunately, the paper does not specify how a VIP node discovers one of these Relays.

Finally, some p2p applications such as Skype select as *super nodes* those that show stability and have enough available bandwidth. These *super nodes* act as Relays for that specific application. Skype is a proprietary application, and it is unknown how the Relay is selected. Nevertheless, some researchers have reverse engineered it [27,31,32] and have found that Skype uses Relays even if direct communications are possible. In particular the clients try to establish the connection through different Relays (sometimes dozens of them) before selecting one. It is also known that the Relays are not randomly selected and that the selection is AS-unaware [27,32].

Locality Solutions for P2P Applications: Over the last years ISPs are experiencing an extra transit traffic due to p2p applications. Furthermore, some ISPs have started to throttle traffic from some applications such as BitTorrent [33,34]. As a reaction to this problem, recently some works have appeared describing locality solutions to keep the traffic of p2p applications within the local ISP as much as possible [35,36]. To the best of our knowledge, there is no previous work that addresses this issue for relayed communications. Thus, we believe that our proposal is the first contribution regarding transit traffic reduction for relayed communications in the Internet.

5 The P2P-NTA Simulator

This section describes the P2P-NTA simulator used to validate the proposal. This is an iterative simulator, implementing a Chord DHT with users deployed in the real Internet topology. Moreover, it is using real latencies to account for the communication delays among the nodes.

The foundation of the simulator is the iPlane [12,13] platform (described in section 2). It builds the topology based on this dataset, which contains AS connectivity, the IP prefixes announced in the BGP default-free zone and delay measurements between pairs of Points of Presence (PoPs) in the Internet. Each PoP, as defined by iPlane, is a set of IP addresses with low latency among them. The simulator uses the PoPs to build the Internet-topology, where we consider

P2P-NTA within the P2PSIP architecture.

55.000 PoPs and their actual point of attachment. Note that an AS may contain more than one PoP. The P2P-NTA simulator considers these PoPs as the access routers of ISPs and therefore, it deploys the Relays randomly among them. We consider 4 cases: 100, 1000, 10000 and 25000 Relays.

Then, for each iteration, the P2P-NTA chooses two different random users from two different PoPs. The users are chosen according to the following criteria, in order: within the same country, within the same continent or from different continents. With this set of experiments we aim to show the performance of our proposal in different scenarios. Nevertheless it is important to remark that the first case, where the users are chosen within the same country, is the most common one, especially in VoIP applications.

The users query the Chord network deployed among the Relays that run the GPA selection algorithm. The P2P-NTA simulator implements a highly scalable Chord network and it is able to route the query towards its destination, and provide the path, number of hops, and latency. In order to simulate very-large Chord networks, the P2P-NTA uses a steady-state approach, and only simulates this p2p network after it has stabilized. We assume that during the simulation no churn is observed, and iteratively generate the finger table for each node in turn, knowing a priori the full list of the nodes in the overlay. After the finger tables are generated, queries can be routed by the simulator using this topology. It is worth noting here that we validated our implementation of the Chord protocol in steady state with OpenChord 1.0.5⁵. Specifically we created a p2p network using OpenChord and waited until the network converged. Next, we compared the finger tables of the P2P-NTA simulator and OpenChord nodes, which were found identical.

Finally, once the query has finished, and both users have agreed on communicating using a given Relay with the help of the GPA, the simulator computes both the direct and the relayed delay. In order to estimate these latencies we rely also on the iPlane platform. Unfortunately, iPlane does not provide all possible delay pairs. When iPlane provides the delay between a pair of nodes, P2P-NTA uses the iPlane delay. This corresponds to about 65% of the pairs in our simulations. For the other 35% pairs, we developed a latency estimator that fills the iPlane gaps. The estimator correlates the geographical distance between IP addresses as reported by the MaxMind database⁶ with the latency, based on a training dataset from iPlane (see details in [37])

Finally it is worth to note that iPlane provides the delay between PoPs, but not the delay between the PoP and the end-user (i.e. the access link). This part of the end-to-end delay is estimated using the dataset provided in [38], that measures the median access link speed for different countries. Hence, the

⁵ <http://open-chord.sourceforge.net>

⁶ Available from <http://www.maxmind.com>.

user is geolocated [20]⁷ and the access link latency is estimated.

6 Evaluation

In this section we present the obtained results from our large-scale simulations in terms of: (i) delay; (ii) ISP-friendliness; (iii) overhead produced by the P2P-NTA (lookup latency and Relay load).

- *Simulation Set-up*

We have used the P2P-NTA simulator described in section 5 to simulate our proposal with different deployments: 100, 1000, 10000 and 25000 Relays. Furthermore, we have also simulated the random Relay selection algorithm for the same number of Relays and the pre-established Relay selection algorithm for a fixed pool of 1000 Relays. These solutions are equivalent to SORS [14] and RON [15] respectively.

Based on the iPlane dataset we consider 55.000 real PoPs for locating the end-users and among them, we randomly choose our deployment scenarios. Table 1 describes them in detail:

Scenarios	Users	100 Relays	1000 Relays	10000 Relays	25000 Relays
AS	15815	84	820	4996	10361
Countries	203	26	92	149	190
Continents	6	6	6	6	6

Table 1

ASes, countries and continents considered by the users and the different deployment scenarios.

We have simulated, for each solution and number of Relays: (i) 30k communications in the *Intra-Country* scenario: the communication is established between hosts located in the same country; (ii) 30k communications in the *Intra-Continent* scenario: same continent (but different countries); (iii) 30k communications in the *Inter-Continent* scenario: different continents. In total we have simulated roughly 700k communications to evaluate our solution.

For each communication we calculate the direct and the relayed delay. Also, we geolocate (AS, country and continent) the two users ($u1$ and $u2$) and the Relay (R) involved in the communication in order to estimate the *ISP-friendliness*

⁷ This database is open source and has an 99.8% accuracy at country level, 75% accuracy at city level (within a range of 25 miles), 22% accuracy at more than 25 miles, and 3% that the IP is not covered by such database.

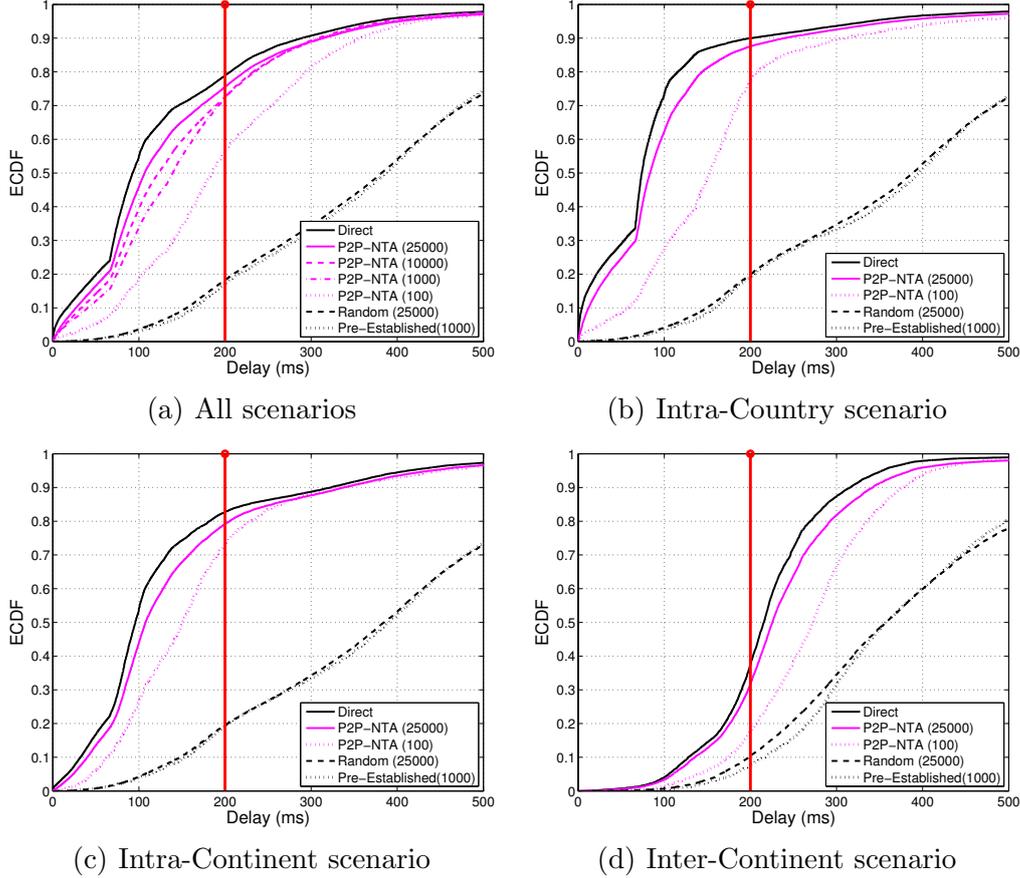


Fig. 5. Relayed Communications Delay

for the different solutions. In addition, we calculate the load supported by each Relay in terms of number of relayed communications. Finally, we compute the Relay look-up latency for each communication.

- *Communication Delay*

We have computed the ECDF of the one-way delay for each solution (P2P-NTA, SORS and RON), deployment (100, 1000, 10000 and 25000 Relays) and scenario (Intra-Country, Intra-Continent and Inter-Continent). Fig 5 summarizes the obtained results:

- Figure 5(a) shows the ECDF for the delay for the 90k communications. We consider the Cisco's 200 ms quality threshold described in [17]. That is, we consider that 200 ms is the maximum tolerable one-way delay for voice communications with acceptable QoS. As expected, the direct communications present the lowest delay, and 79% of them are below the aforementioned threshold. The P2P-NTA slightly increases the direct communication delay under the largest considered deployment. For instance, in the case of

25000 Relays⁸, 75.5% of the communications are below the 200 ms threshold. This means that less than 4% of the communications would suffer from QoS degradation due to the use of Relays compared to the direct one. As the deployment decreases, the number of communications below the threshold slowly decreases (72.6% for 10000 Relays and 72.2% for 1000 Relays) up to 56.3% in the 100 Relays case. However, even in such small deployments, the P2P-NTA clearly outperforms other proposed algorithms. The figure shows that SORS-like⁹ (random) and RON-like (pre-established) selection algorithms can only keep the communication delay below the 200 ms threshold in 18.3% and 17.2% of the cases respectively. These values are less than 1/2 of our proposal's worst case (100 Relays).

- Figures 5(b), 5(c), 5(d), depict the result for the Intra-Country, Intra-Continent and Inter-Continent scenarios respectively. For clarity we just plot the deployments of 25000 and 100 Relays for the P2P-NTA solution. The rest of deployments lay between these two curves. In those cases, the amount of communications which are below the quality threshold between the direct and the P2P-NTA (25000 Relays) is always smaller than 6%. Hence, independently of the type of communication considered (short, medium or long distance) we can conclude that our solution causes a minimum QoS degradation.

Moreover, the P2P-NTA outperforms other Relay selection algorithms. Even if we compare our solution using the smallest deployment (100 Relays), the number of communications below 200 ms is 4 times larger in the Intra-Country and Intra-Continent case, and almost 2 times larger in the Inter-Continent case.

Finally, and as a side-result, it is worth to note the problems that long distance delay-sensitive communications (e.g. VoIP) may experience in the current Internet. Fig 5(d) shows that even in the direct communication case, just 1/3 of the communications are below the 200 ms threshold.

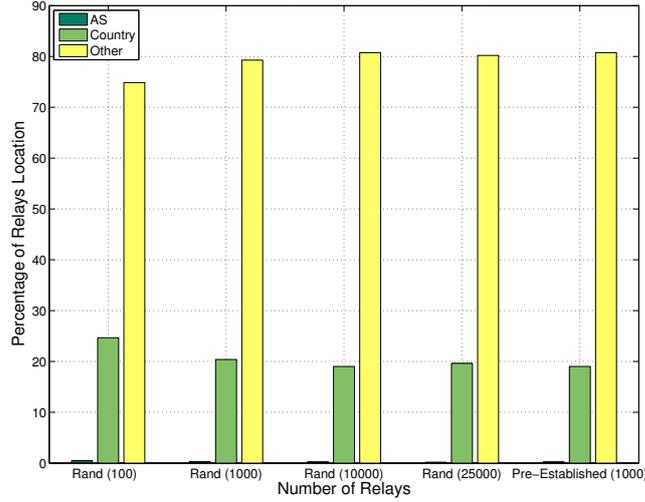
In short, we can conclude that in terms of end-to-end communication delay our solution largely outperforms other previous works, and for (p2p applications) reasonable deployments the performance is similar to the direct communications.

- *ISP-Friendliness*

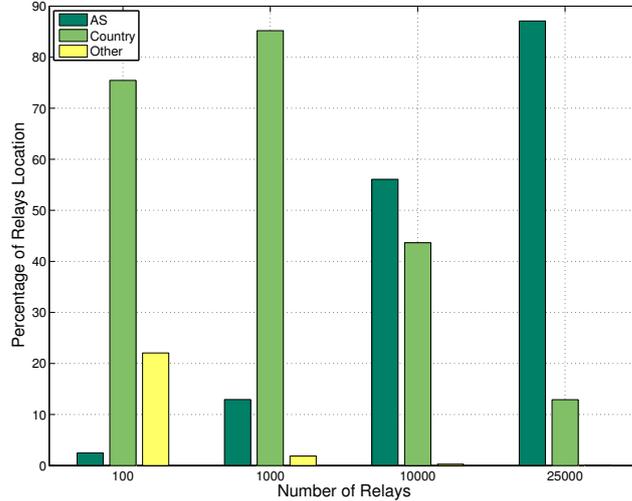
As stated in section 4, ISPs have recently shown their concerns because of the large amount of traffic generated by p2p applications (e.g., BitTorrent). Figure 6 supports our initial hypothesis, and shows that the P2P-NTA is ISP-friendly

⁸ It must be highlighted that 25000 Relays is actually a small deployment if we consider current p2p applications such as Skype or KAD that include millions of concurrent users.

⁹ Note that the random selection algorithm performs similarly regardless of the number of Relays. Thus, we only depict one case.



(a) SORS(random) and RON (pre-established)



(b) P2P-NTA

Fig. 6. ISP friendliness. Percentage of Relays located in the same AS, same country or different country than the users.

since it minimizes the relayed traffic transmitted through transit links. This figure represents the percentage of communications (out of the 90k) in which the Relay has been selected in: (A) the AS of one of the two end-users, thus leading to zero cost; (B) an AS in the same country of one of the two hosts, thus using for free a peering link to communicate; (C) any other case where the Relay uses transit links to communicate with both end-hosts.

In figure 6(a) we present the results for SORS-like (random) and RON-like (pre-established) selection algorithms, whereas figure 6(b) depicts the results

for the different deployments of our solution. We can see that both, RON-like and SORS-like, (in the best case) use less than 1% and 25% of Relays located in the same AS (case A) and country (case B) than $u1$ or $u2$, respectively. Thus, using in more than 73% of the cases transit links towards both $u1$ and $u2$ (case C).

On the other hand, and as expected, the performance of our solution is affected by the deployment. In the case of 25000 Relays, 87% of the communications are established through a Relay located in the same AS than one of the hosts (case A) while the remaining 13% use a Relay in the same country (case B). If we consider the minimum deployment case with just 100 Relays, our solution still performs quite well since just around the 20% of connections are established through a Relay outside the ISP and country of both hosts (case C).

To further understand the results, we consider relative transit costs: $cost = 0$ for case A; $cost = 1/2$ for case B; $cost = 1$ for case C. Recall that in case B, the Relay uses a transit link to communicate with one of the hosts, whereas in case C it uses transit links for both hosts. Furthermore, case A is free of cost since the Relay is located in the same AS than one of the end-users. Table 2 shows the average cost of the communications for each solution. We observe that RON-like and SORS-like selection algorithms are close to the maximum cost (between the 87-90% of the maximum possible cost), whereas the P2P-NTA produces almost no transit traffic cost (6%) for the largest considered deployment. The cost increases as we reduce the deployment. However, even in the case of minimum deployment (100 Relays), our solution reduces around 30% of the transit traffic compared to the other proposals.

	100 Relays	1000 Relays	10000 Relays	25000 Relays
P2P-NTA	0.60	0.44	0.22	0.06
SORS-like	0.87	0.89	0.90	0.90
RON-like	-	0.89	-	-

Table 2

Average Transit Traffic Cost of the Communications (Max = 1, Min = 0)

P2P-NTA Overhead

In this section we evaluate the different aspects of the overhead introduced by our solution. First, we evaluate the Relay lookup latency. Next, we evaluate the number of communications transmitted through the Relay nodes, and we compare them to a SORS-like approach.

- *Relay lookup latency*: We define the lookup latency as the time to search through the P2P-NTA and retrieve a list of Relays. We have measured the Relay lookup latency for all the communications (90k) and the different

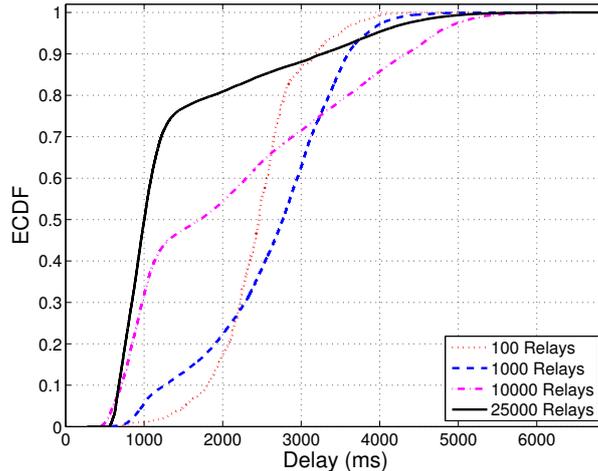


Fig. 7. Relay Look-up Latency

deployments. Fig 7 summarizes the results. It shows the ECDF of the Relay lookup latency for the different deployments. As we can observe, the lookup latency is in the order of a few seconds and decreases as we augment the number of Relays. Although this could seem a high value, it is not affecting the QoS of the communications. Note that the P2P-NTA launches the lookup procedure when the application (e.g. Skype) starts. Thus, when the user desires to establish a communication the Relay has been already selected. Then, this does not add any extra delay to the standard connection establishment protocol (e.g. [7]) in our solution.

- *Number of Supported Communications per Relay:* We have computed the number of communications supported by each Relay, for both solutions, P2P-NTA and SORS-like (random selection), considering the different deployments. The resultant ECDFs are presented in Fig 8. First, we observe that the load produced by our system is similar to that generated by a random Relay selection algorithm. Since a random selection is expected to produce a fair distribution, we can claim that the P2P-NTA is a fair solution in terms of Relay usage. Actually, for large deployments (25000) the curves are completely overlapped. Hence, the higher the deployment, the more fair the solution is, while still keeping lightweight.

7 Conclusions

Our work starts from the premise that relayed communications cannot be avoided in today's Internet. Based on real measurements we have shown that using a topologically closer relay has a critical impact on the QoS of the communications and the costs of ISPs. In order to reduce this impact we have introduced: (i) the Gradual Proximity algorithm (GPA), that finds the topologically closest available relay and (ii) the Peer-to-Peer NAT Traver-

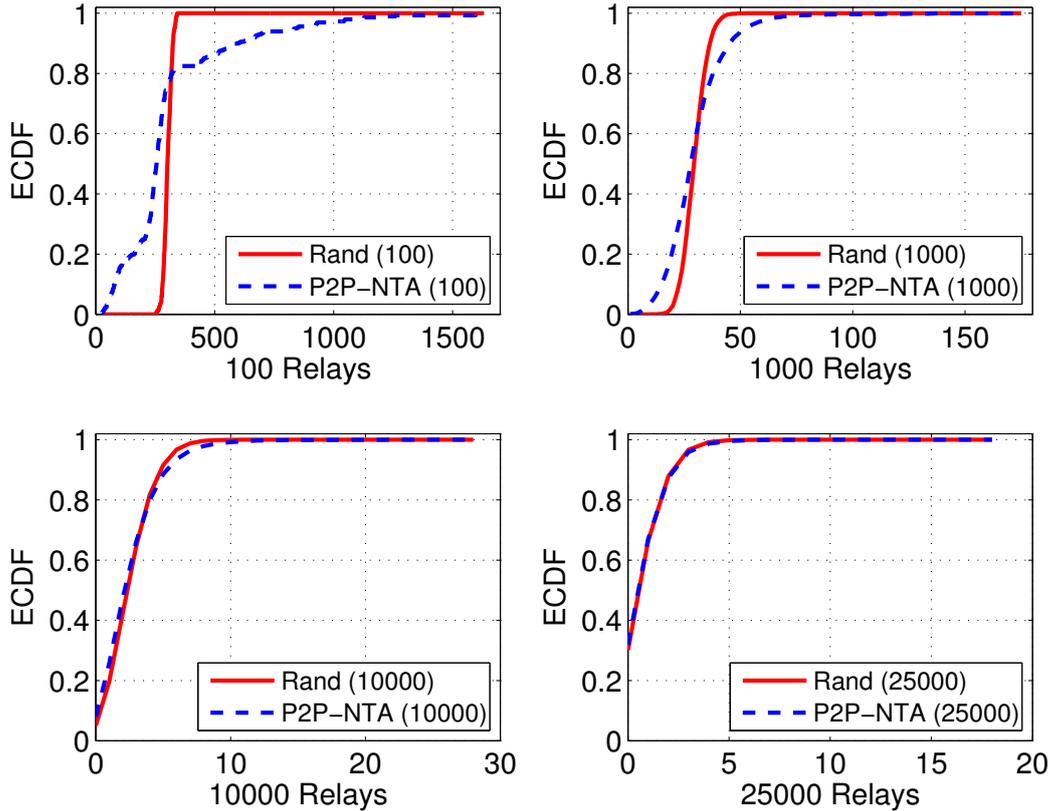


Fig. 8. Number of relayed communications per Relay

sal Architecture (P2P-NTA), which is a lightweight distributed architecture - based on a DHT - that implements the GPA. We have carried out large-scale simulations using the Internet real topology associated with real delays. The obtained results show that our proposal largely outperforms previous works in the literature. Furthermore, the P2P-NTN exhibits performance levels comparable to direct communication when used by a resonable deployment of a p2p application.

References

- [1] I. Van Beijnum, "IPv4 Address Consumption," *The Internet Protocol Journal*, vol. 10, no. 3, pp. 22–28, 2007.
- [2] K. Egevang and P. Francis, "RFC 1631 The IP Network Address Translator (NAT)," Internet Engineering Task Force, May 1994.
- [3] M. Cadaco and M. Freedman, "Illuminati - Opportunistic Network and Web Measurement," Sep. 2007, <http://illuminati.coralcdn.org/stats/>.
- [4] "The Impact of P2P File Sharing, Voice over IP, Skype, Joost, Instant Messaging, One-Click Hosting and Media Streaming such as YouTube on the

Internet,” 2007, <http://www.ipoque.com/resources/internet-studies/internet-study-2007>.

- [5] J. Rosenberg, J. Weinberger, C. Huitema, and R. Mahy, “STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs),” Internet Engineering Task Force, 2003.
- [6] J. Rosenberg, R. Mahy, and P. Matthews, “Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN),” Internet Engineering Task Force, Apr. 2009.
- [7] J. Rosenberg, “Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols,” Internet Engineering Task Force, Oct. 2007.
- [8] B. Ford, P. Srisuresh, and D. Kegel, “Peer-to-peer communication across network address translators,” in *ATEC '05: Proceedings of the annual conference on USENIX Annual Technical Conference*. USENIX Association, 2005, pp. 13–23.
- [9] S. Guha and P. Francis, “Characterization and measurement of TCP traversal through NATs and firewalls,” in *IMC '05: Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement*. USENIX Association, 2005, pp. 18–18.
- [10] A. Biggadike, D. Ferullo, G. Wilson, and A. Perrig, “NATBlaster: Establishing TCP connections between hosts behind NATs,” in *ACM SIGCOMM Asia Workshop*. ACM, Apr. 2005.
- [11] Y. Huang, T. Z. Fu, D.-M. Chiu, J. C. Lui, and C. Huang, “Challenges, design and analysis of a large-scale p2p-vod system,” in *ACM SIGCOMM 2008 conference on Data communication*. ACM, 2008, pp. 375–388.
- [12] H. V. Madhyastha, T. Isdal, M. Piatek, C. Dixon, T. Anderson, A. Krishnamurthy, and A. Venkataramani, “iPlane: an information plane for distributed services,” in *OSDI '06: Proceedings of the 7th symposium on Operating systems design and implementation*. USENIX Association, 2006, pp. 367–380.
- [13] iPlane, “ iPlane: An Information Plane for Distributed Services,” <http://iplane.cs.washington.edu/>.
- [14] K. P. Gummadi, H. V. Madhyastha, S. D. Gribble, H. M. Levy, and D. Wetherall, “Improving the reliability of internet paths with one-hop source routing,” in *OSDI'04: Proceedings of the 6th conference on Symposium on Operating Systems Design & Implementation*. USENIX Association, 2004.
- [15] D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris, “Resilient overlay networks,” *SIGOPS Oper. Syst. Rev.*, vol. 35, no. 5, pp. 131–145, 2001.
- [16] “One-Way Transmission Time,” 1996, ITU Rec. G.114, International Telecommunications Union.

- [17] *Enterprise QoS Solution Reference Network Design Guide*. CISCO, 2005.
- [18] W.B.Norton, “The Evolution of the U.S. Internet Peering Ecosystem,” Equinix white papers, 2004.
- [19] WHOIS service. <http://www.netdemon.net/tutorials/whois.txt>.
- [20] MaxMind. <http://www.maxmind.com/>.
- [21] G. Siganos, J. M. Pujol, and P. Rodriguez, “Monitoring the Bittorrent Monitors: A Bird’s Eye View.” in *PAM’09: Proceedings of the Passive Active Measurement conference*, ser. Lecture Notes in Computer Science, vol. 5448. Springer, 2009, pp. 175–184.
- [22] P. B. Godfrey and I. Stoica, “Heterogeneity and load balance in distributed hash tables,” in *IN PROC. OF IEEE INFOCOM*, 2005.
- [23] S. Surana, B. Godfrey, K. Lakshminarayanan, R. Karp, and I. Stoica, “Load balancing in dynamic structured peer-to-peer systems,” *Perform. Eval.*, vol. 63, no. 3, pp. 217–240, 2006.
- [24] D. R. Karger and M. Ruhl, “Simple efficient load balancing algorithms for peer-to-peer systems,” in *SPAA ’04: Proceedings of the sixteenth annual ACM symposium on Parallelism in algorithms and architectures*. ACM, 2004, pp. 36–43.
- [25] R. Cuevas, M. Urueña, and A. Banchs, “Routing fairness in Chord: Analysis and Enhancement,” in *In proceedings INFOCOM 2009. 28th IEEE International Conference on Computer Communications*. IEEE, 2009.
- [26] Y. Rekhter, T. Li, and S. Hares, “A Border Gateway Protocol 4 (BGP-4),” Internet Engineering Task Force, Jan. 2006.
- [27] S. Ren, L. Guo, and X. Zhang, “Asap: an as-aware peer-relay protocol for high quality voip,” in *ICDCS ’06: Proceedings of the 26th IEEE International Conference on Distributed Computing Systems*. IEEE Computer Society, 2006.
- [28] P2PSIP Working Group. <http://www.ietf.org/html.charters/p2psip-charter.html/>.
- [29] J. XingFeng, “A mechanism to discover STUN/TURN nodes in P2PSIP. draft-jiang-p2psip-stun-turn-discovery-00,” Internet Engineering Task Force, 2007.
- [30] X. Wang and Q. Deng, “VIP: A P2P Communication Platform for NAT Traversal,” in *Parallel and Distributed Processing and Applications, Third International Symposium, ISPA 2005*, 2005, pp. 1001–1011.
- [31] S. A. Baset and H. G. Schulzrinne, “An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol,” in *In proceedings INFOCOM 2006. 25th IEEE International Conference on Computer Communications*. IEEE, 2006, pp. 1–11.

- [32] K. Woogyun, S. Baset, and H. Schulzrinne, "Skype relay calls: Measurements and experiments," in *In proceedings INFOCOM Workshops 2008, . 27th IEEE International Conference on Computer Communications*. IEEE, 2008, pp. pp.1–6.
- [33] M. Dischinger, A. Mislove, A. Haeberlen, and K. P. Gummadi, "Detecting Bittorrent Blocking," in *IMC '08: Proceedings of the 8th ACM SIGCOMM conference on Internet measurement*. ACM, 2008, pp. 3–8.
- [34] "Vuze Network Status Monitor," Azureus, http://azureus.sourceforge.net/plugin_details.php?plugin=aznetmon.
- [35] H. Xie, R. Y. Yang, A. Krishnamurthy, Y. G. Liu, and A. Silberschatz, "P4p: Provider portal for applications," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 4, pp. 351–362, 2008.
- [36] D. R. Choffnes and F. E. Bustamante, "Taming the torrent: a practical approach to reducing cross-isp traffic in peer-to-peer systems," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 4, pp. 363–374, 2008.
- [37] A. Cabellos-Aparicio, D. Saucez, O. Bonaventure, and J. Domingo-Pascual, "Validation of a LISP simulator," UPC, Tech. Rep. UPC-DAC-RR-CBA-2009-8, 2009.
- [38] "Ookla's speedtest throughput measures," <https://confluence.slac.stanford.edu/display/IEPM/Ookla%27s+Speedtest+Throughput+Measures>.