Computer Communications 33 (2010) 365–371

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

SCTP for robust and flexible IP anycast services

Tim Stevens *, Daan Pareit, Filip De Turck, Ingrid Moerman, Bart Dhoedt, Piet Demeester

Ghent University—IBBT, Department of Information Technology (INTEC), Gaston Crommenlaan 8, Bus 201, 9050 Ghent, Belgium

ARTICLE INFO

Article history:
Received 3 August 2009
Received in revised form 1 October 2009
Accepted 2 October 2009
Available online 13 October 2009

Keywords:
Anycast
SCTP
Overlay network
Stateful communications

ABSTRACT

IP anycast is a powerful network layer mechanism that can be used for transparent communications between clients and a distributed service infrastructure. Unfortunately, large-scale deployment of IP anycast would cause a number of severe problems, including excessive routing table growth and potential routing instability. In order to solve these problems, a number of overlay network architectures have been proposed over the last years. In this paper we show that the robustness of anycast services provided via such anycast architectures can be significantly improved by using SCTP transport layer facilities. More specifically, the proposed approach adds the following important features to existing anycast overlays: Robustness to anycast overlay node failure or network reconfiguration, seamless anycast service delivery to mobile clients, and true stateful anycast communications over an entirely stateless infrastructure. Furthermore, we argue that the number of overlay nodes can be drastically reduced in comparison with the earlier architectures, without degrading service quality or increasing the end-to-end path stretch.

1. Introduction

IP anycast [1] enables communications between a sender and a single node out of a group of potential receivers, usually the nearest group member based on network distance. In practice, network layer anycast is realized by assigning the same IP address to all members of the anycast group, whereupon the routing infrastructure is reconfigured to take into account all potential anycast target locations. From the sender's perspective, communicating with an anycast group (member) is indistinguishable from traditional unicast communications. This, combined with the fact that anycast groups can support an arbitrarily large number of members (by design), makes IP anycast a powerful network layer tool for distributed service provisioning.

Despite these promising features, present operational use of IP anycast in the Internet is essentially limited to DNS root server replication [2]. This is because IP anycast introduces serious network problems, including routing scalability issues and potential routing instability caused by joining or leaving anycast group members. Over the past decade, several network layer solutions have been proposed that—at least partially—address these anycast issues. One of the first proposals is the Global IP Anycast (GIA) framework due to Katabi and Wroclawski [3], where anycast routing scalability is achieved by introducing a GIA-specific address prefix and adding extra anycast logic to routers that distinguishes between popular and unpopular anycast destinations. More recently, Ballani and Francis [4] have proposed the Proxy IP Anycast Service (PIAS). PIAS is an overlay infrastructure providing anycast routing scalability, network stability, and full transparency towards clients. In short, PIAS edge nodes attract anycast traffic and silently tunnel anycast packets towards their final destinations based on anycast destination IP address and TCP port information. The Architecture for Scalable and Transparent Anycast Services (ASTAS) proposed by Stevens et al. [5] elaborates on PIAS by introducing a fine-grained target selection mechanism that improves resource utilization efficiency, albeit at the expense of adding extra complexity to the overlay nodes.

Even though PIAS and its derivative ASTAS enable practical use of IP anycast for large scale deployments and a virtually unlimited number of coexisting anycast services, there are still a number of limitations that prevent ideal service provisioning:

(1) The mechanism is non-resistant to network configuration changes or overlay node failures.
(2) PIAS and ASTAS lack support for stateful services on mobile clients, even when Mobile IP (MIP) [6] is used.
(3) The overlay edge nodes introduce packet forwarding bottlenecks.

In this paper, we show how the interaction between an anycast overlay infrastructure and the *Stream Control Transmission Protocol* (SCTP) [7] provides an elegant and natural solution for the above mentioned operational shortcomings related to anycast communications. The proposed solution relies on overlay nodes *only* during session initialization, but supports true stateful, flexible, and robust anycast services. Moreover, overlay nodes do not maintain

* Corresponding author. Tel.: +32 9 3314900.
E-mail addresses: tim.stevens@intec.ugent.be (T. Stevens), daan.pareit@intec.ugent.be (D. Pareit).

0140-3664/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2009.10.002

individual session state, which significantly improves overall scalability of the proposed architecture.

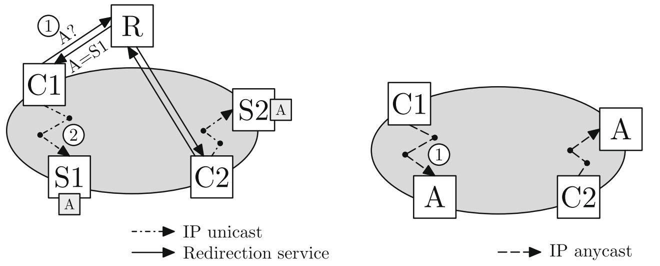
The remainder of this paper is structured as follows. Section 2 reviews existing anycast architectures and differentiates between network and application layer anycasting mechanisms. New design goals to improve IP anycast-based service provisioning are discussed in Section 3. In Section 4, we reveal our architectural anycast solution based on SCTP. The discussion in Section 5 revisits the initial design objectives and comments on minor limitations. Finally, Section 6 concludes the paper.

2. Background

Similar to multicast, anycast communications can be realized in two fundamentally different ways: Either at the application layer or at the network layer. Fig. 1 depicts both anycasting techniques. Application layer anycast consists of two successive phases: First, a higher layer redirection mechanism (indicated by *R* in Fig. 1(a)) resolves the anycast service identified by *A* to a regular (unicast) IP address. Once the client C1 receives the IP address, the server S1 is contacted directly using unicast communications (step 2). Intermediate routers forward IP packets over the shortest path to their destination. Most often, the redirection service *R* is implemented using the Domain Name System (DNS) infrastructure. Zegura et al. [8] show how application layer anycasting can be used for successful web server replication. Freedman et al. [9] further optimize end-to-end latency for DNS-based anycasting through their OASIS infrastructure that selects target servers based on a mapping of the Internet to geographical coordinates. Although application layer anycast is the easiest to deploy, it has the disadvantage that target selection and connection establishment cannot be combined in an inseparable, one-step operation. Once a unicast IP address is discovered, the redirection mechanism can be bypassed for subsequent service requests, eventually disrupting load-balancing mechanisms or centralized management tasks (e.g., accounting).

Native IP anycast [10] immediately forwards IP packets targeted at an anycast group to the nearest group member, as depicted in Fig. 1(b). This simple property makes it an appealing packet delivery mode in the context of distributed service provisioning, because *the network* automatically selects a target server for each initiated service request. From the client perspective, communicating with the group is no different from unicast communications. Unfortunately, the benefits of IP anycast are largely overshadowed by the severe network problems it introduces, which immediately explains its limited use in today's Internet. IP anycast suffers from:

(1) *Excessive routing table growth*: Contrary to IP unicast, routes to anycast destinations cannot be aggregated because anycast group members may be scattered all over the Internet, i.e., there is no correlation between anycast IP address and physical location. Hence, routing tables grow linearly with the number of globally deployed anycast groups.



(a) Application layer anycast (b) Network layer anycast

Fig. 1. Application and network layer anycasting side by side.

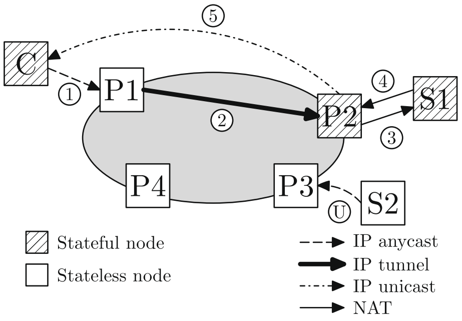


Fig. 2. PIAS [4] data path.

(2) *Frequent route updates*: Each joining and leaving anycast group member potentially triggers a cascade of routing table updates in a large network area. If the target joining and/or leaving frequency is too high, this results in network instability.

The Proxy IP Anycast Service (PIAS) architecture proposed by Ballani and Francis [4] eliminates these anycast deployment problems in an elegant way and opens the door for global IP anycast deployment. Additionally, PIAS enables to select a target server based on several criteria (e.g., server load) instead of only minimizing end-to-end path length. The basic working principles of PIAS are depicted in Fig. 2. PIAS consists of inbound (e.g., P1) and outbound (e.g., P2) overlay nodes called *proxies*. Proxies behave just as regular routers for unicast traffic, but anycast traffic is handled differently. Furthermore, PIAS assumes that all anycast addresses are grouped in a small number of IP address ranges. By making this assumption, it is easy to configure the PIAS proxies to capture all anycast traffic. This is achieved by advertising a route to these anycast IP ranges through the proxies. In short, setting up a session via PIAS takes place in five steps (See Fig. 2):

(1) An anycast packet is attracted by the nearest proxy P1.
(2) The inbound proxy P1 tunnels (IP-in-IP [11]) this packet to a suitable outbound proxy P2 in the neighborhood of an actual target server for the requested service.[1]
(3) The outbound proxy decapsulates the tunneled packet and initiates a connection with the target server S1 on behalf of the client C (using Network Address Translation (NAT) [12]).
(4) S1 sends a return packet to P2, thinking P2 is the actual client (due to NAT).
(5) P2 performs NAT on the return packet and forwards it directly to the client C (via unicast routing).

New anycast target servers join by sending a registration message to their nearest proxy, again by using native anycast. Server status updates are sent in a similar way (step U in Fig. 2).

Stevens et al. [5] have extended PIAS to the Architecture for Scalable and Transparent Anycast Services (ASTAS) in order to enable fine grained control over outbound proxy selection in inbound proxies, and to enhance support for stateful communications. The ASTAS data path is depicted in Fig. 3. Differences between the ASTAS and PIAS data path are the following:

(1) ASTAS inbound proxies maintain session state to enable per session outbound proxy selection.

[1] Appropriate outbound proxies are located via a distributed registry. This is not shown in Fig. 2.